

**Ângela Cristina Marques de Oliveira**

**METODOLOGIA PARA O DESENVOLVIMENTO DE UMA  
APLICAÇÃO DE MANIPULAÇÃO DE EXERCÍCIOS E  
DOCUMENTOS ASSOCIADOS**



**Departamento de Ciência de Computadores  
da Faculdade de Ciências da Universidade do Porto**

**2006**



**Ângela Cristina Marques de Oliveira**

**METODOLOGIA PARA O DESENVOLVIMENTO DE UMA  
APLICAÇÃO DE MANIPULAÇÃO DE EXERCÍCIOS E  
DOCUMENTOS ASSOCIADOS**



Tese submetida à FCUP para obtenção do grau de  
Mestre em Informática, área de especialização em  
Ciência de Computadores



*“O que conseguimos obter quando alcançamos os nossos objectivos não é tão importante como aquilo em que nos tornámos ao alcançá-los.”*

**Zig Ziglar**, orador norte-americano



## *Índice Geral*

<b>AGRADECIMENTOS</b>	<b>III</b>
<b>RESUMO</b>	<b>V</b>
<b>ABSTRACT</b>	<b>VII</b>
<b>ÍNDICE DE TEXTO</b>	<b>IX</b>
<b>ÍNDICE DE FIGURAS</b>	<b>XI</b>
<b>ÍNDICE DE TABELAS</b>	<b>XIII</b>
<b>SIGLAS/ ABREVIATURAS</b>	<b>XV</b>
<b>CAPÍTULO 1 - INTRODUÇÃO</b>	<b>1</b>
<b>CAPÍTULO 2 – MÉTODOS DE INVESTIGAÇÃO</b>	<b>9</b>
<b>CAPÍTULO 3 - O QUE EXISTE NO CONTEXTO DO TEMA A DESENVOLVER?</b>	<b>17</b>
<b>CAPÍTULO 4 – METODOLOGIA E MODELAÇÃO DA INFORMAÇÃO</b>	<b>37</b>
<b>CAPÍTULO 5 - DESENVOLVIMENTO DO PROTÓTIPO GEREXA</b>	<b>69</b>
<b>CAPÍTULO 6 – CONCLUSÕES E TRABALHO FUTURO</b>	<b>91</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>93</b>



## **Agradecimentos**

Na caminhada da vida, recolhemos estímulo e incentivo de pequenos gestos e palavras que parecem insignificantes...

À minha orientadora, Professora Nelma Moreira, por toda a disponibilidade, apoio e incentivo.

Aos meus pais,

À minha irmã,

Ao Luís

À Sara,

Apoiaram-me incondicionalmente em mais esta etapa da minha caminhada, mas o que se pode dizer das pessoas que mais amamos...

A todos os meus amigos que me deram força para continuar, cada vez que um obstáculo surgia.

A todos os meus colegas e amigos de trabalho, que me fizeram muitas vezes não perder o rumo.

A todos o meu mais sincero OBRIGADA.



## Resumo

Durante o seu percurso, os professores vão acumulando matérias de diferentes temas. Geralmente a produção de um documento de avaliação, torna-se às vezes complicado pela forma dispersa dos exercícios.

Assim, com a evolução das tecnologias surgem novos desafios na área da educação. É fundamental conceber soluções de e-Learning que flexibilizem o acesso aos recursos de aprendizagem.

Neste trabalho, apresenta-se primeiro uma metodologia que originou a definição de uma linguagem e finalmente um protótipo de uma aplicação, que assenta nos princípios básicos de desenvolvimento das aplicações de e-Learning, mas com a introdução de funcionalidades que permitirão aos professores manter a sua informação organizada e disponível.

O objectivo é tornar mais cómodo e eficiente o trabalho de produção de material didáctico e de avaliação.

O recurso à modelação de uma linguagem como o XML, trás inovação relativamente às bases de dados estruturadas e tradicionais, ao mesmo tempo que permite garantir que a informação continua a ter uma forma estruturada mas com uma semelhança maior com os documentos reais.

Combinando a flexibilidade do XML, com PYTHON, uma linguagem fácil de utilizar, orientada a objectos e multiplataforma, permite uma excelente concretização neste tipo de aplicações.



## **Abstract**

Through the years, teachers accumulate a considerable amount of documents to support their activity, such as drills, assignments, homework's or exams. The basic content of those documents are exercises, that normally are kept disperse and unorganized.

Computers and new Web technologies can be of valuable help in manipulating and organizing those learning resources.

In this work, first we created a methodology that it originated a language definition and create a prototype that is seats in the beginner base from E-learning, but with entry of functionalities that will be allowing teachers hold information available and organized.

The objective is to become more efficient the manipulation of exercises and exams.

XML language allows keeping structure of information with bigger similarity with real documents in opposition to the traditional databases.

Combine XML flexibility and an oriented multiplatform programming language Python, that it is easy to use, it allows an excellent specify in this type of applications.



# Índice de Texto

<b>CAPÍTULO 1</b>	<b>- INTRODUÇÃO</b>	<b>1</b>
1.1 -	MOTIVAÇÃO	1
1.2 -	DEFINIÇÃO DO PROBLEMA	3
1.3 -	OBJECTIVOS	4
1.4 -	ORGANIZAÇÃO DA DISSERTAÇÃO	6
<b>CAPÍTULO 2</b>	<b>- MÉTODOS DE INVESTIGAÇÃO</b>	<b>9</b>
2.1 -	CONCEITOS	9
2.2 -	MÉTODOS DE INVESTIGAÇÃO	11
2.2.1	<i>Métodos lógicos</i>	11
2.2.1.1	Método Lógico Dedutivo	12
2.2.1.2	Método Hipotético Dedutivo	12
2.2.1.3	Método Lógico Indutivo	13
2.2.2	<i>Método Histórico</i>	13
2.2.3	<i>Método Sintético</i>	14
2.2.4	<i>Método Empírico</i>	14
2.2.5	<i>Método de Experimentação Científica</i>	14
2.3 -	ABORDAGEM E MÉTODO (S) SELECIONADO (S)	15
<b>CAPÍTULO 3</b>	<b>- O QUE EXISTE NO CONTEXTO DO TEMA A DESENVOLVER?</b>	<b>17</b>
3.1 -	APLICAÇÕES RELACIONADAS	18
3.1.1	- <i>WebDuc</i>	18
3.1.2	- <i>GedEx Web</i>	19
3.1.3	- <i>TestGIP</i>	22
3.1.4	- <i>SAEL</i>	24
3.1.5	- <i>easyTest</i>	24
3.1.6	- <i>SEATT</i>	25
3.1.7	- <i>“Sistema para el diseño, generación, aplicación y evaluación de exámenes basado en plataforma Internet” – SGA Ex</i>	26
3.1.8	- <i>NetSupport School®</i>	26
3.1.9	- <i>Perguntas de Escolha Múltipla On-line</i>	27
3.1.10	- <i>Advanced eLearning Builder®</i>	29
3.1.11	- <i>FORMARE®</i>	30
3.1.12	- <i>MOODLE</i>	32
3.1.13	- <i>IMS Global Learning Consortium Inc. ®</i>	33
3.2 -	CONCLUSÕES	34
<b>CAPÍTULO 4</b>	<b>- METODOLOGIA E MODELAÇÃO DA INFORMAÇÃO</b>	<b>37</b>
4.1 -	METODOLOGIA	37
4.2 -	MODULAÇÃO DA INFORMAÇÃO	40
4.2.1	- <i>Escolha múltipla</i>	40
4.2.2	- <i>Resposta múltipla</i>	40
4.2.3	- <i>Correspondência de colunas</i>	41
4.2.4	- <i>Verdadeiro/Falso</i>	41
4.2.5	- <i>Resposta calculada</i>	42
4.2.6	- <i>Desenvolvimento</i>	42
4.3 -	TECNOLOGIAS XML	43
4.4 -	A LINGUAGEM XML GEREXA	46

4.4.1	- <i>Temas</i>	50
4.4.2	<i>Exercícios</i>	51
4.4.2.1	Escolha Múltipla	53
4.4.2.2	Resposta Múltipla	55
4.4.2.3	Correspondência de Colunas	56
4.4.2.4	Verdadeiro/Falso	57
4.4.2.5	Desenvolvimento	58
4.4.2.6	Resposta Calculada	58
4.4.3	<i>MetaDados</i>	65
4.4.4	- <i>Aluno</i>	66
4.4.5	- <i>Professor</i>	67
4.4.6	- <i>Teste</i>	67
4.5 -	- CONCLUSÃO	68
<b>CAPÍTULO 5 - DESENVOLVIMENTO DO PROTÓTIPO GEREXA</b>		<b>69</b>
5.1 -	LINGUAGEM DE SUPORTE	69
5.1.1	<i>O Python</i>	69
5.1.1.1	DOM & SAX	71
5.1.1.2	DOM	71
5.1.1.3	SAX	71
5.1.1.4	DOM vs SAX	72
5.1.1.5	ELEMENTTREE	72
5.2 -	DESCRIÇÃO DO PROTÓTIPO GEREXA	74
5.2.1	<i>Funcionalidades do Protótipo</i>	74
5.2.2	<i>Modelo de Classes do Protótipo</i>	75
5.2.2.1	Os Temas	76
5.2.2.2	Os Exercícios	83
5.2.2.3	Os Alunos/ Os Professores	87
5.2.2.4	Outros Aspectos	89
<b>CAPÍTULO 6 – CONCLUSÕES E TRABALHO FUTURO</b>		<b>91</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>		<b>93</b>

# Índice de Figuras

Figura 3-1 – Autenticação.....	20
Figura 3-2 – Escolha de Conteúdo .....	20
Figura 3-3 – Manipulação de perguntas .....	20
Figura 3-4 – Selecção de perguntas.....	21
Figura 3-5 – Escolha do tipo de pergunta.....	21
Figura 3-6 – Inserção de dados adicionais.....	21
Figura 3-7 – Interface do TestGIP .....	22
Figura 3-8 – Áreas do Interface.....	23
Figura 3-9 – Resultado produzido .....	23
Figura 3-10 – Interface do Módulo Professor – “Desenhador de Provas”.....	27
Figura 3-11 – Esquema de funcionamento da aplicação Perguntas de Escolha Múltipla On-line.....	28
Figura 3-12 – exemplo do uso do interface da aplicação - Perguntas de Escolha Múltipla On-line.....	28
Figura 3-13 – Interface de construção de teste .....	29
Figura 3-14 – Interface de elaboração de teste diagnóstico.....	30
Figura 3-15 – Um exemplo de teste produzido pelo FORMARE.....	31
Figura 3-16 – <a href="http://e-matematica.pt">http://e-matematica.pt</a> .....	32
Figura 3-17 – Utilizadores da plataforma Moodle.....	32
Figura 4-1 – Metodologia GerExa.....	37
Figura 4-2 – Representação em árvore do elemento “node” .....	50
Figura 4-3 - Representação em árvore do elemento “exercício” .....	51
Figura 4-4- Representação em árvore do elemento “escolhamúltipla” .....	54
Figura 4-5- Representação em árvore do elemento “respostamúltipla” .....	55
Figura 4-6- Representação em árvore do “elemento correspondência de colunas” .....	56
Figura 4-7- Representação em árvore do elemento “respostacalculada”.....	58
Figura 4-8- Representação em árvore do elemento “meta” (metadados) .....	65
Figura 4-9- Representação em árvore do elemento “aluno” .....	67
Figura 4-10- Representação em árvore do elemento “teste” .....	67
Figura 5-1 – Diagrama de Classes do protótipo GerExa .....	75
Figura 5-2 – Representação de uma árvore de “Temas”.....	76
Figura 5-3 – Estrutura inicial para manipulação de “Temas” .....	77
Figura 5-4 – Inserção de novo “tema pasta” .....	78
Figura 5-5 – Resultado da inserção de um “tema pasta” .....	79
Figura 5-6 – Resultado da inserção de um “tema folha” .....	79
Figura 5-7 – Procedimento de edição de um “tema pasta” .....	80
Figura 5-8 – Resultado da edição de um “tema pasta” .....	81
Figura 5-9 – Estrutura com vários níveis e sub-níveis.....	81
Figura 5-10 – Procedimento para eliminar um tema .....	82
Figura 5-11 – Estrutura em árvore resultante da eliminação de um tema (pasta).....	82
Figura 5-12 – Módulo de Manipulação de Exercícios.....	83
Figura 5-13 – Definição dos parâmetros de um exercício .....	84
Figura 5-14 – Inserção de uma questão de escolha múltipla com 3 opções .....	85
Figura 5-15 – Listagem de Exercícios .....	85
Figura 5-16 – Listagem do exercício “matemática1.xml” .....	86
Figura 5-17 – Página inicial do protótipo “GerExa” .....	87
Figura 5-18 – Módulo de Administração - Alunos.....	87
Figura 5-19 – Procura de Alunos pelo campo Instituição.....	88



## Índice de Tabelas

Tabela 3-1 – Tabela resumo das aplicações estudadas	35
Tabela 5-1– Tabela resumo DOM vs SAX	72



## Siglas/ Abreviaturas

API	Application Programming Interface
API JDBC	Java Database Connectivity Application Programming Interface
ASP	Active Server Pages
CSS	Cascading Style Sheets
DOM	Document Object Model
DTD	Document Type Definition
FLASH	Tecnologia utilizada no WEB que permite a criação de animações vectoriais.
HTML	HyperText Markup Language
http	HyperText Transfer Protocol
JAVA	Linguagem de Programação
MySQL	Sistema de Gestão de Bases de Dados Open Source
OLE	Object Linking and Embedding
PHP	Linguagem de Programação WEB
PYTHON	Linguagem de Programação
SAX	Simple API for XML
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
SQL2000	Sistema de Gestão de Bases de Dados proprietário da Microsoft
UNIX	Sistema operativo Multi-utilizador
W3C	World Wide Web <i>Consortium</i>
WEB	Termo pelo qual é mais conhecida a World Wide Web
WWW	World Wide Web
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformations
UML	Unified Modelling Language



## Capítulo 1 - Introdução

### 1.1 - Motivação

O ensino de uma determinada matéria, por parte de um docente, obriga à avaliação da mesma. Todos os docentes se deparam com inúmeros períodos de avaliação, os quais implicam a realização de testes de avaliação.

Com o avanço tecnológico, o aparecimento do ensino assistido por computador e do ensino à distância as necessidades e grau de exigência dos professores são cada vez maiores, tornando a automatização de tarefas num processo não só útil mas também necessário, de forma a implementar um ambiente de ensino com conteúdos dinâmicos.

É notória a importância que vem sendo dada ao longo do tempo, à forma como a avaliação dos alunos é conduzida e aos factores que podem contribuir para melhorar o desempenho dos alunos, dando especial destaque à avaliação via WEB, como refere Carla Langsch [1], no seu artigo “Avaliação no Ensino à Distância via WEB”. Assim é fundamental desenvolver aplicações que auxiliem o professor na “ádua e dura” tarefa que é a avaliação.

O recurso a meios informáticos, para leccionar todo o tipo de matérias, bem como o ensino à distância, têm vindo a ganhar cada vez mais relevância no meio académico e torna-se necessário adaptar os métodos, técnicas e tecnologias tradicionais para desenvolver aplicações não convencionais e evoluídas, de forma a automatizar tanto quanto possível as tarefas de “rotina”.

## **CAPÍTULO 1**

Pode dizer-se que hoje em dia, existem três tipos de ensino: o ensino tradicional (presencial), E-learning e um misto dos dois tipos referidos, cada um dos quais com as suas vantagens e desvantagens inerentes [2] e é fundamental a adaptação a novas aplicações que permitam uma melhor integração em qualquer dos tipos de ensino referidos.

Os objectivos em que se devem centrar as aplicações são: reutilização, acessibilidade, e durabilidade, tendo em conta a integração de material didáctico e os ambientes virtuais de educação [2].

Assim, não só a produção de material pedagógico como a avaliação, tem que ter por parte dos professores uma nova preocupação e ir de encontro às novas formas de ensino.

A preocupação maior, hoje em dia, reside na procura de soluções para uma adequada integração da tecnologia com a pedagogia, existindo várias organizações a trabalhar neste campo, entre elas, a IMS Global Learning Consortium [3].

Mas o estudo das estratégias de e-learning está cada vez mais em foco, como refere Romiszowski [4], a visão das plataformas que implementam e-learning, tem que ser encarada segundo diversas perspectivas, podendo ser uma actividade individual ou de grupo, que pode ocorrer em tempo-real ou permitindo uma escolha flexível do tempo. No seu artigo, o autor, mostra que é importante desenvolver sistemas de e-learning, onde os papéis dos professores e alunos sejam amplamente salientados e analisados. No que diz respeito ao professor ao se transformar num “professor on-line”, este necessita de desenvolver competências no domínio das novas tecnologias de ensino. No que diz respeito ao aluno devido à quantidade de informação disponibilizada, este necessita de desenvolver novas competências de aprendizagem.

Assim ao formar uma geração sintonizada com um novo modelo de aprendizagem, é necessário construir o suporte para essa inovação, disponibilizando ferramentas, o mais ajustadas, possível, a essas mesmas necessidades.

## **1.2 - Definição do Problema**

Um problema que geralmente se apresenta nas instituições de educação é a elaboração e avaliação dos testes de avaliação, processo que exige aos professores bastante tempo.

Outro aspecto é que, o professor, ao longo da sua carreira vai, realizando inúmeras provas de avaliação, que apesar de hoje em dia na grande maioria dos casos ficarem armazenadas em formato digital, se encontram distribuídas por vários documentos, não tendo, muitas vezes noção da quantidade e diversidade de exercícios que foram acumulados ao longo do tempo.

Considerando, ainda que existem vários professores que leccionam a mesma matéria, poderemos aumentar o número de exercícios disponíveis. Será vantajoso utilizar soluções informáticas, que permitam reunir o material, como refere Jamie Dinkelacker, no seu artigo “Placing Our Stuff So We Can Find It Later: A Meta-Learning Essencial”[5], e na analogia que faz com um bando de pássaros em voo, mostrando que a coordenação e cooperação existente pode ser usada para modelar sistemas de dados, usando uma organização nos ficheiros, colocando-os onde possam ser facilmente acedidos e eliminar a confusão criada por cópias desnecessárias ou ficheiros repetidos, bem como a definição de nomes para pastas e sub-pastas onde os ficheiros devem ser guardados.

Assim, seguindo a perspectiva de Dinkelacker [5], desenvolver uma aplicação, que obedeça a regras, onde os exercícios são catalogados por conteúdos ou matérias, sempre acessíveis uma vez que o arquivo dos mesmos é efectuado de forma estruturada, evitando assim a existência de exercícios repetidos e posteriormente utilizados para elaborar testes, numa aplicação que permita liberdade na construção dos mesmos, representará sem dúvida um sistema potente de ajuda aos professores no desempenho das suas funções de avaliação.

### **1.3 - Objectivos**

O objectivo desta tese pode ser dividido em dois sub-objectivos:

O primeiro é o desenho de uma metodologia para a modelação de uma linguagem XML que permita a manipulação de exercícios e testes de avaliação.

Embora haja já desenvolvido, nesta área um vasto trabalho, com especial destaque para o IMS Global Learning [3], o que se pretende é uma linguagem mais específica de acordo com a nossa realidade em termos de exercícios e testes.

O segundo é o desenvolvimento de um protótipo de uma aplicação, a que se deu o nome de GerExa, que permita armazenar exercícios e posteriormente gerar testes de avaliação com base nos exercícios guardados. Tendo em conta que esta ferramenta poderá ser posteriormente integrada num sistema de E-learning, para além das tarefas associadas ao professor consideram-se também algumas tarefas correspondentes aos alunos.

Assim, no que diz respeito ao segundo sub-objectivo, poderão considerar-se os seguintes pontos, como objectivos gerais da aplicação:

- Facultar ao professor uma ferramenta eficiente e atractiva, para guardar os exercícios e gerar testes de avaliação;
- Reduzir o tempo dispendido com a realização de testes de avaliação;
- Facilitar a organização da informação aos professores que leccionam a mesma disciplina;
- Facultar aos alunos o acesso a exercícios de treino;

- Permitir aos alunos a resolução on-line de testes de avaliação;
- Produzir a correcção automática dos testes de avaliação.

## **1.4 - Organização da Dissertação**

Esta dissertação encontra-se organizada em 6 Capítulos, que se descrevem, a seguir, de forma sucinta:

Neste Capítulo e nas secções anteriores foi abordado o problema a desenvolver, bem como a motivação e os objectivos a atingir.

No Capítulo 2 – “Métodos de Investigação ”, referem-se conceitos gerais sobre o envolvimento da investigação científica, bem como os diferentes métodos que existem. Neste capítulo foi seleccionado o método usado nesta dissertação.

No Capítulo 3 – “O que existe no contexto da Aplicação a desenvolver?”, diz respeito a uma pesquisa das aplicações existentes na área do problema a investigar, no final do capítulo foi elaborada uma tabela com as principais características de cada uma das aplicações pesquisadas bem como a tecnologia associada.

No Capítulo 4 – “Metodologia e Modelação para a manipulação de Exercícios e Testes”:

Apresenta a uma metodologia para a manipulação de documentos estruturados: exercícios e testes.

Apresenta a modulação da informação subjacente à metodologia definida, que origina a definição de uma linguagem XML.

No Capítulo 5 – É apresentada a descrição do protótipo, que permite testar a funcionalidade e aplicabilidade da metodologia e da linguagem XML – GerExa.

Bem como apresentado um estudo referente às linguagens de suporte ao desenvolvimento do protótipo.

O Capítulo 6, faz referência às conclusões, da aplicabilidade da metodologia definida, da linguagem GerExa e do protótipo GerExa, bem como o Trabalho Futuro a desenvolver nas três vertentes apresentadas nesta dissertação: metodologia, linguagem XML e protótipo.

## **CAPÍTULO 1**

## Capítulo 2 – Métodos de Investigação

### 2.1 - Conceitos

“O homem nunca actua directamente sobre as coisas. Existe sempre um intermediário, um instrumento entre ele e os seus actos. Isto também acontece quando se investiga cientificamente, quando se pretende conhecer algo.” [6]

Quando se fala em metodologias de investigação surgem conceitos como ciência, conhecimento, teorias, hipóteses, entre outros.

Mas o que se pode dizer da ciência?

- A ciência é um conjunto de conhecimentos que se obtêm mediante a observação, a partir dos quais se deduzem princípios e leis gerais [7].

-Pode ser definida como, um conhecimento racional, exacto e que se pode validar, através de investigação científica [8].

A actividade ou investigação científica é caracterizada por ser um processo de acumulação de conhecimento que comporta teorias ainda em fase de desenvolvimento.

A actividade científica pode ser definida como uma abordagem com capacidade de explicativa, de previsão, de classificação, descrição e interpretação, comportando assim várias verdades provisórias, novas descobertas.

A especificação de uma metodologia científica, para além de assegurar que a investigação é pode ser reproduzida, fornece ferramentas ao investigador para abordar uma dada problemática de uma forma sistemática e objectiva.

## CAPÍTULO 2

Uma investigação científica é definida como um “trabalho empreendido metodologicamente, quando surge um problema, para o qual se procura a solução adequada de natureza científica” [9].

Um trabalho científico, é caracterizado por “uma investigação científica e pelo tratamento das questões abordadas metodologicamente” [9].

O estudo de métodos científicos passa por quatro etapas: Formulação do Problema, Formulação de Hipóteses, Verificação da Hipótese e Generalização e formulação de teorias.

Segundo Albert Einstein, a formulação de um problema é muitas vezes mais importante que a sua solução, a qual pode ser uma mera questão de cálculos matemáticos ou uma questão experimental. Fazer novas perguntas e considerar novas possibilidades para resolver problemas antigos, requer imaginação criativa e indica o verdadeiro progresso da Ciência.

A primeira etapa tem assim especial relevância uma vez que a formulação do problema, tem que ser efectuada com algum cuidado, para que haja respostas para o problema em questão.

Segundo Graviel Salomon, um problema que define o tipo de investigação científica a desenvolver, que pode ser:

- **Pesquisa Descritiva** – o objectivo deste tipo pesquisa é definir o problema, de modo a proporcionar novas perspectivas sobre o tema;
- **Pesquisa Aplicada** - o objectivo é aplicar teorias e modelos na descoberta de soluções;
- **Pesquisa Pura ou Teórica** – o objectivo é apresentar teorias ou modelos indo para além da descrição de problemas.

Para a questão da formulação das hipóteses, segundo Michael Myers [10], os métodos de investigação científica podem ser classificados de várias formas, mas o mais comum é fazer a distinção abordagens qualitativas e quantitativas.

- **Abordagem Quantitativa** - foi desenvolvida nas ciências da natureza, para proceder a estudo de fenómenos naturais.  
Na abordagem quantitativa, são definidas hipóteses, com base na teoria. Essas hipóteses terão que ser quantificadas através de variáveis que possam sofrer um estudo, estatístico a fim de validar ou não as hipóteses formuladas. Este tipo de abordagem procura enumerar e avaliar todos eventos.
- **Abordagem Qualitativa** – São formuladas as hipóteses com a preocupação, de efectuar uma medição objectiva dos resultados e ao contrário da abordagem quantitativa, parte de questões mais amplas que se vão desenvolvendo à medida que a investigação prossegue.

## ***2.2 - Métodos de Investigação***

Os métodos de investigação científica, podem ser diferenciados em dois grupos, os métodos lógicos e os métodos empíricos [11].

### **2.2.1 Métodos lógicos**

Os métodos lógicos são métodos que aplicam os conceitos de dedução, análise e síntese. Assim pode falar-se em métodos como:

## **CAPÍTULO 2**

### **2.2.1.1 Método Lógico Dedutivo**

Neste método aplicam-se princípios descobertos, a casos particulares, ou seja a partir de princípios conhecidos, encontram-se novos princípios, ou consequências desconhecidas de princípios já conhecidos.

Neste caso, parte-se de teorias ou leis aceites e procede-se à sua aplicação numa dada investigação.

Pode falar-se em método dedutivo directo e método indutivo indirecto, no primeiro tem-se uma conclusão imediata, no segundo a conclusão a que se chega é formal.

#### **Exemplo (Método Dedutivo Directo):**

“Os livros são cultura”

“Em consequência, algumas manifestações culturais são livros”.

#### **Exemplo (Método Dedutivo Indirecto):**

“Os ingleses são pontuais”

“Charles é inglês”

“Logo, Charles é pontual”.

### **2.2.1.2 Método Hipotético Dedutivo**

O investigador propõe uma hipótese baseada na refutação das teorias existentes e formulação de teorias mais abrangentes.

Este método usa procedimentos indutivos e dedutivos para validar a hipótese.

Através deste método o conhecimento evolui, por intermédio da eliminação do que é falso e da confirmação do que não é refutado.

Através das inferências lógico dedutivas pode chegar-se a conclusões particulares a partir da hipótese, que posteriormente se pode comprovar experimentalmente.

### 2.2.1.3 Método Lógico Indutivo

Este método parte de constatações particulares tendo por objectivo chegar a teorias e leis genéricas. Permite a formulação de hipóteses, investigação de leis científicas e demonstrações.

A indução pode ser completa, ou incompleta.

A indução só pode ser completa, caso se conheçam todos os elementos que formam o objecto de investigação.

A indução é incompleta quando o objecto de investigação não pode ser numerado e estudado na sua totalidade, sendo necessário avaliar o problema a través de uma amostra representativa que permita fazer generalizações.

#### **Exemplo (Método Indutivo Completo):**

Se o objecto de estudo for a avaliação do rendimento de um conjunto de alunos, a indução pode ser completa, uma vez que o objecto de estudo é relativamente pequeno e podem ser avaliados todos os elementos.

#### **Exemplo (Método Indutivo incompleto):**

Se o objecto de estudo for: “Os jovens gostam de *música house*”, teria que ser estudada a preferência de um conjunto representativo de jovens e a conclusão será tirada com base nas repetições que se verificarem. A maior ou menor probabilidade na aplicação do método, reside no número de casos analisados e as conclusões não podem ser tomadas como demonstrações mas sim como possibilidade de uma verdade.

### 2.2.2 Método Histórico

Este método está relacionado com o conhecimento das diferentes etapas dos objectos na sua sucessão cronológica. Analisa-se a sua trajectória concreta e o seu condicionamento aos diferentes períodos da história.

### **2.2.3 Método Sintético**

Neste método, relacionam-se factores aparentemente isolados e é formulada uma teoria que unifica os diversos elementos. Consiste numa reunião racional de vários elementos dispersos numa nova hipótese que terá que ser demonstrada.

### **2.2.4 Método Empírico**

A definição destes métodos assenta na percepção directa do objecto de investigação e do problema formulado.

### **2.2.5 Método de Experimentação Científica**

O investigador cria modelos e reproduz condições. Este método deve seguir as regras seguintes:

1. O fenómeno a tratar deve ser devidamente isolado para um estudo mais concreto;
2. A experiência deve repetir-se, verificando as mesmas circunstâncias para comprovar os resultados;
3. As condições de realização da experiência devem ser alteradas para averiguar de que forma o fenómeno é afectado;
4. A experiência deve durar o tempo suficiente até comprovar o objectivo esperado.

### **2.3 -Abordagem e Método (s) Seleccionado (s)**

É complicado, escolher um método, como método ideal e caminho único para realizar uma investigação, uma vez que alguns métodos se complementam ou estão relacionados.

Considerando o tema e a problemática que se pretendem abordar o método de investigação científica: hipotético-dedutivo, já que é possível formular hipóteses que se podem analisar dedutiva ou indutivamente.

Este método permite a evolução do conhecimento através da eliminação do que é falso e da confirmação de novas teorias mais abrangentes.

Assim, o método hipotético-dedutivo define que uma investigação deve começar por identificar o problema e uma questão alvo da investigação, em seguida, deve ser proposta uma teoria, expressa através de hipóteses passíveis de serem testadas

O problema a abordar foi especificado no capítulo 1. Mediante a motivação e os objectivos traçados respectivamente na secção 1.2 e 1.3 do capítulo 1, será elaborado no capítulo 3, seguinte um estudo sobre o que já existe no âmbito da aplicação a desenvolver, de forma a levantar questões sobre os aspectos a estudar e analisar a forma como essas questões foram até ao momento abordadas sendo a base de estudo para a proposta de solução do problema definido.

## **CAPÍTULO 2**

## **Capítulo 3 - O que existe no contexto do Tema a desenvolver?**

A aprendizagem e todo o seu envolvimento: técnicas e métodos, tem vindo a ser bastante explorado, nomeadamente no que diz respeito ao ensino à distância e à WEB.

O contexto educativo da WEB, disponibiliza uma nova plataforma, que estimula tanto o ensino como a aprendizagem. Assim o âmbito educativo da WEB, não é apenas mais uma ferramenta de ensino mas surge como alternativa ao ensino tradicional [12].

Este tipo de aplicações tem características especiais, de forma a proporcionar aos alunos ambientes de trabalho estimulantes.

Nesta perspectiva é considerado de fulcral importância o papel do professor, também ele necessita de dispor de aplicações que vão de encontro às exigências estabelecidas por esta nova abordagem na educação.

Neste sentido, foi elaborada uma investigação, sobre as aplicações existentes na WEB, que contemplassem não só a vertente do aluno mas também desenvolvessem a actividade do professor no que diz respeito à inserção e utilização do material didáctico.

### **3.1 -Aplicações Relacionadas**

Existem disponíveis, na WEB, inúmeras ferramentas que referem a geração de testes, geralmente associadas a plataformas de e-learning, existindo diversas aplicações comerciais e aplicações de software livre (não proprietário a maior parte das vezes).

Neste capítulo vão ser analisadas essas aplicações sobre o ponto de vista das suas funcionalidades e também sobre os ambientes de desenvolvimento.

#### **3.1.1 – WebDuc**

A aplicação Webduc [13], é uma proposta de uma ferramenta de avaliação formativa, desenvolvida em linguagem PHP, associada ao motor de bases de dados MySQL, ou seja, usando software não proprietário, e encontra-se dividida, em dois módulos diferentes: o módulo do aluno e o módulo do professor.

O interesse específico é no módulo do professor que nesta aplicação, permite:

- **Inserir questões** – nesta opção o professor poderá inserir questões dos quatro tipos disponibilizados (questões de desenvolvimento, relacionamento de colunas, escolha múltipla e verdadeiro/falso), não havendo limite para o número de questões inseridas.
- **Visualizar questões** – permite visualizar e eliminar as questões de um dos tipos de questão, seleccionado;
- **Questões por avaliação** – nesta opção o professor pode seleccionar o número de questões de cada tipo, para um dado momento de avaliação;
- **Visualizar avaliação** – permite ao professor visualizar os momentos de avaliação gerados;

- **Inserir respostas** – nesta opção o professor coloca as respostas às questões de escolha múltipla, relacionamento de colunas e verdadeiro/falso, para posteriormente serem automaticamente corrigidas;
- **Verificar respostas** – o professor tem acesso às respostas de cada um dos alunos e caso a avaliação contenha apenas questões do tipo escolha múltipla, correspondência de colunas e verdadeiro/falso, aparecem automaticamente corrigidas, tendo o professor que corrigir apenas, as respostas às questões de desenvolvimento.

### **3.1.2 – GedEx Web**

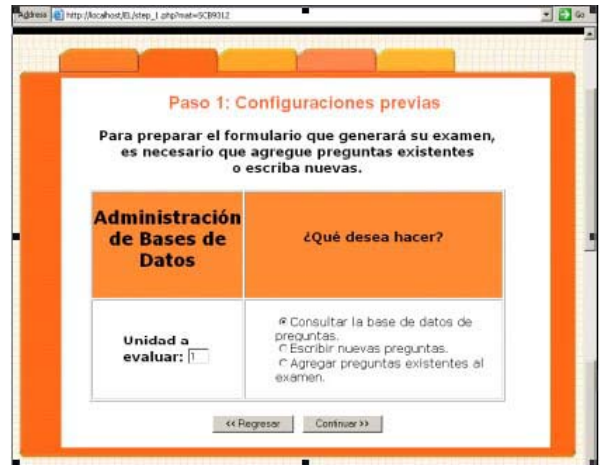
A aplicação GedEx Web [14], foi criada com o objectivo, da elaboração de testes para a sua posterior publicação, de forma a facilitar, ao professor a sua tarefa de elaboração de testes e que o aluno tenha acesso aos mesmos, independentemente da distância.

Esta aplicação disponibiliza um conjunto de funcionalidades, que funcionam como etapas a seguir, para produzir um documento HTML, que será publicado na WEB e depois de ser resolvido pelos alunos, será enviado por correio electrónico para o professor.

A aplicação foi desenvolvida em PHP e MySQL, recorrendo ao Flash Macromedia, folhas de estilo CSS, para produzir um formulário final em HTML.

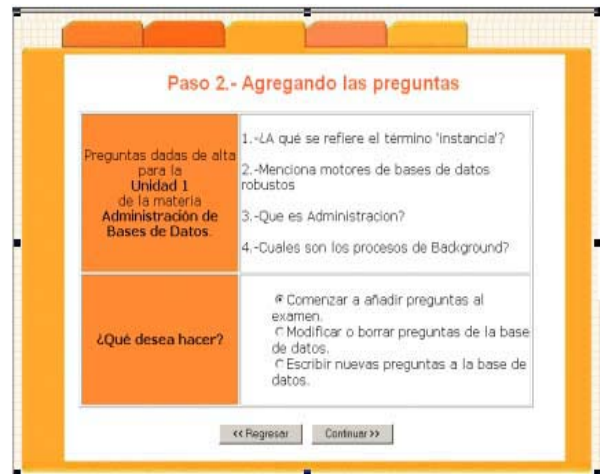
Assim como etapas/funcionalidades desta aplicação tem-se:

**Proceso de autentificación (etapa 1)** – o professor tem que inserir login e password, de forma que apenas ele acede à base de dados das questões.



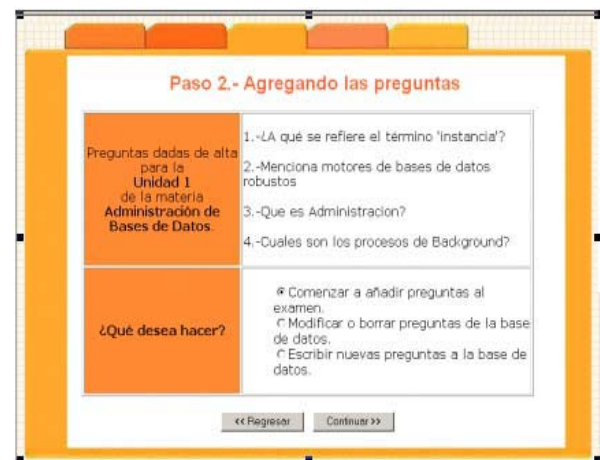
**Figura 3-1 – Autentificación**

**Selección de materias (etapa 2)** – permite seleccionar un determinado conteúdo, para posterior manipulação de questões.



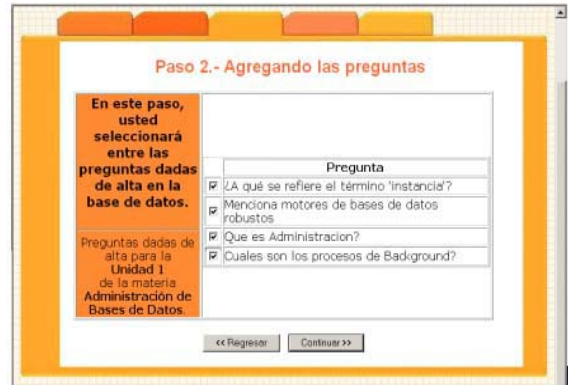
**Figura 3-2 – Escolha de Conteúdo**

**Manipulação de questões (etapa 3)** - Permite a consulta, inserção, alteração e eliminação de questões do conteúdo escolhido.



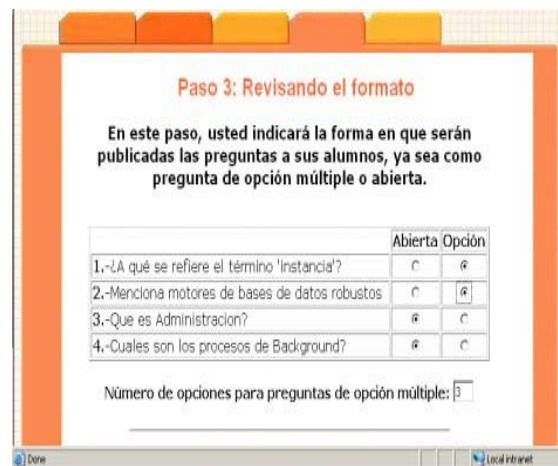
**Figura 3-3 – Manipulação de perguntas**

**Agregação de perguntas (etapa 4)** – permite seleccionar as questões a colocar no exame que se está a realizar.



**Figura 3-4 – Selección de preguntas**

**Indicação do tipo de pergunta (etapa 5)** – nesta etapa o professor escolhe se a questão vai ser de desenvolvimento ou de escolha múltipla, indicando no último caso, o número de opções que irão ter as respostas. Este número será igual, para todas as questões do mesmo exame.



**Figura 3-5 – Escolha do tipo de pergunta**

**Agregação de opções e dados adicionais (etapa 6)** – nesta etapa colocam-se as escolhas possíveis para as respostas de escolha múltipla.



**Figura 3-6 – Inserção de dados adicionais**

Geração do documento (etapa 7) – a aplicação gera um formulário HTML.

### 3.1.3 – TestGIP

TestGIP [15], é uma ferramenta de apoio ao professor, desenvolvida em Microsoft Visual Basic 6.0, com o objectivo de elaborar, desenvolver e avaliar, testes de escolha múltipla com documentos de multimédia e dispõe de três módulos: TestGIP-Professor, TesteGIP-Aluno e TestGIP\_Notas.

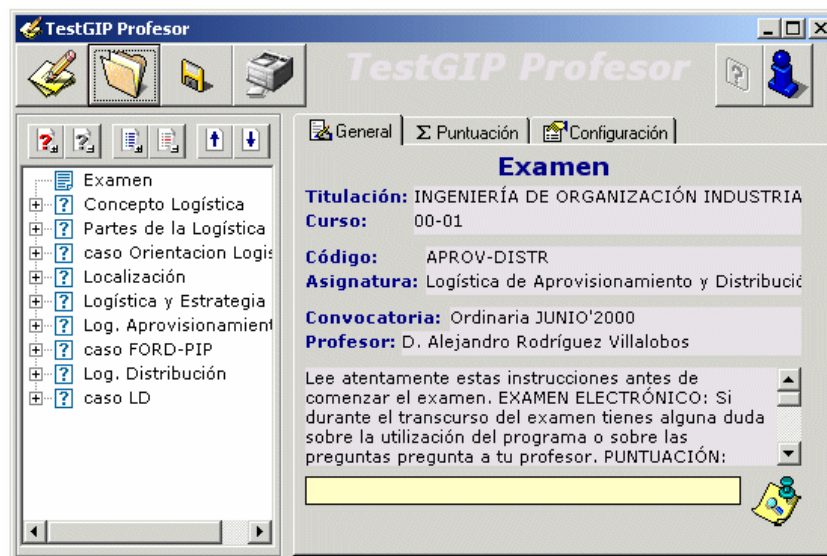
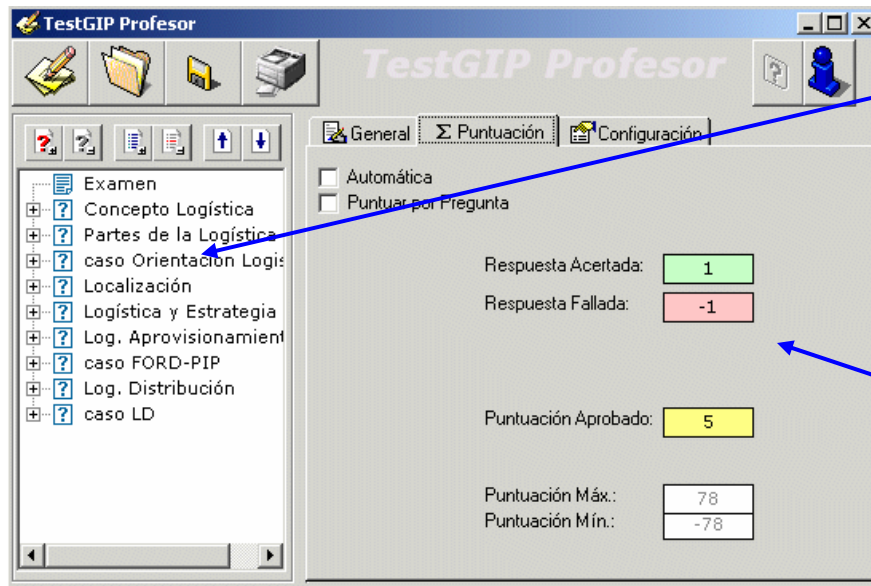


Figura 3-7 – Interface do TestGIP

No módulo TestGIP-Professor, que é o objecto de análise, o professor, pode realizar testes, que podem ser completamente novos ou com base na modificação de um exame já existente.

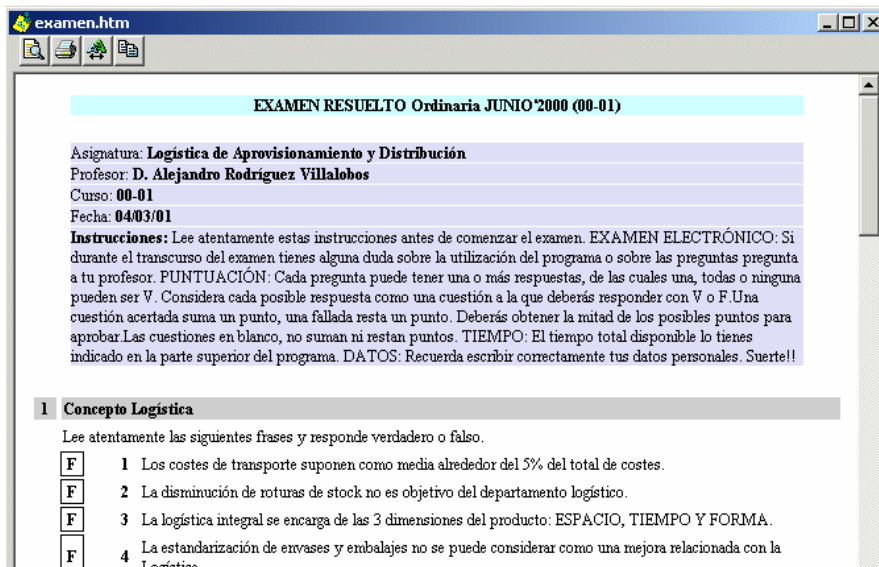
São disponibilizadas duas áreas, contendo respectivamente, perguntas e respostas (área 1) e o exame (área 2), representadas na *Figura 3-8*.



Na área 1, é possível adicionar e eliminar perguntas e respostas, modificar a sequência das perguntas do teste.

Na área 2, é possível modificar as instruções gerais do teste e configurar duração, cotação, e penalizações.

**Figura 3-8 – Áreas do Interface**



Quando é terminada a elaboração do teste, *Figura 3-9*, este é guardado, para posterior distribuição aos alunos.

**Figura 3-9 – Resultado produzido**

### **3.1.4 – SAEL**

SAEL [16], é um sistema de avaliação electrónico como ferramenta para a actividade didáctico-pedagógica do professor, foi desenvolvida em Delphi Developer 2.01 e entre as suas funcionalidades têm-se:

- **Matrícula de Alunos** – permite registar e matricular os alunos nas diferentes disciplinas;
- **Registo de Professores** – permite o registo dos dados dos professores;
- **Registo de Cursos** – permite a inserção dos cursos existentes na instituição;
- **Registo de Disciplinas** – permite a inserção das disciplinas dos diferentes cursos;
- **Registo de Assuntos** – permite a inserção dos conteúdos leccionados nas diferentes disciplinas;
- **Elaboração de Testes** – permite inserir tempo de duração do teste, número de questões, tipo de teste (questões subjectivas, objectivas ou mistas) e permite a escolha de questões já existentes ou a elaboração de questões novas recorrendo a edição gráfica e de textos com editores disponibilizados, pela aplicação.

### **3.1.5 – easyTest**

EasyTest [17], é uma ferramenta, desenvolvida em JAVA, que permite a partir um conjunto de perguntas e respectivas respostas, tantas quantas o professor queira, gerar  $n$  testes.

À medida que o professor vai construindo o teste, com respostas de escolha múltipla, vai indicando qual a resposta correcta para cada pergunta.

A geração dos testes é completamente aleatória, tanto para a ordem das perguntas, como para a ordem das respostas, para que, com o mesmo conjunto de perguntas e respectivas respostas se possa gerar um teste diferente por aluno.

Todos os testes são gerados no mesmo documento HTML e possuem um número único, que será posteriormente associado a um aluno.

Cada teste, possui uma tabela com todas as questões, onde o aluno deverá colocar, qual a resposta certa para cada questão, de modo a que os testes possam ser corrigidos automaticamente.

### **3.1.6 – SEATT**

SEATT [18], é uma ferramenta, desenvolvida em JAVA, e usa API JDBC, para aceder a uma base de dados onde estão armazenadas as perguntas e respectivas opções de resposta, bem como, a opção de resposta correcta.

As perguntas podem ser armazenadas por área de conhecimento e/ou grau de dificuldade, para permitir a escolha, de um destes itens ou ambos para gerar os testes.

Esta aplicação é completamente automática na geração de testes.

### **3.1.7 - “Sistema para el diseño, generación, aplicación y evaluación de exámenes basado en plataforma Internet” – SGA Ex**

SGA Ex [19] é um sistema integral, o qual pretende ser uma ferramenta para os professores de todos os níveis de ensino, de modo a ajudá-los no processo de avaliação dos alunos.

Esta ferramenta implementa a construção, geração e classificação de testes, com perguntas do tipo: escolha múltipla, correspondência e verdadeiro/falso.

Para além desta funcionalidade a ferramenta disponibiliza, o acesso aos alunos, para realização de determinado teste, que o professor pode disponibilizar, escolhendo uma hora para o efeito. A plataforma foi implementada em PHP, sendo por esse motivo um sistema multi-plataforma.

### **3.1.8 – NetSupport School®**

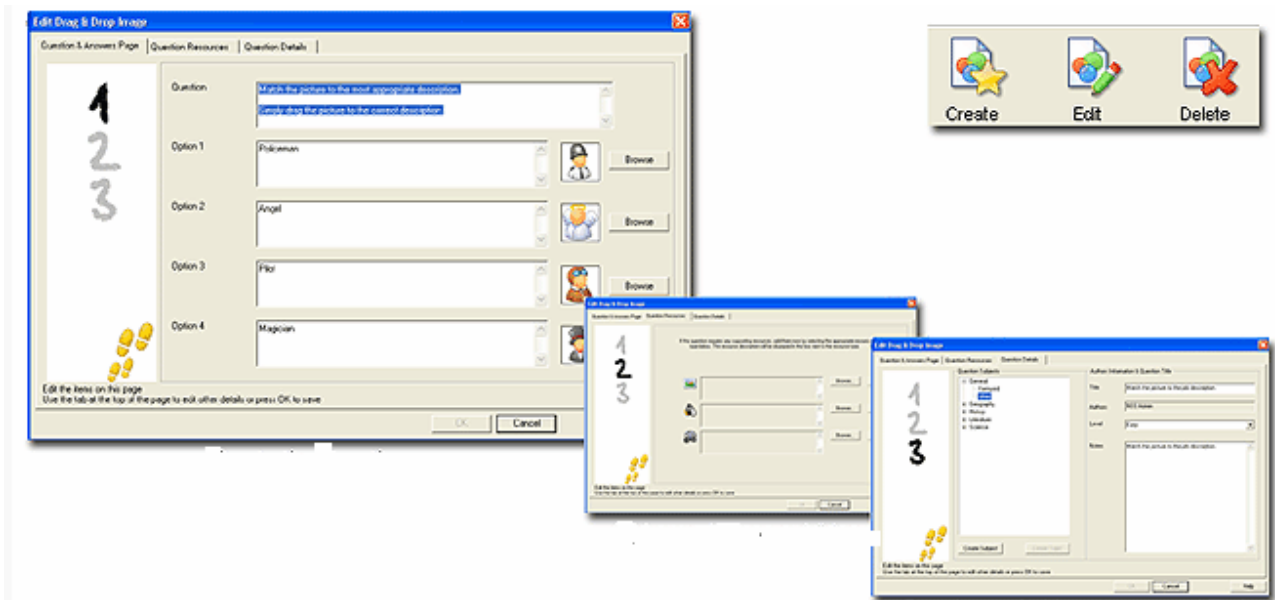
NetSupport School® [20], é uma potente ferramenta de formação comercial, que beneficia os professores que utilizam uma rede informática, serve para ensinar aos alunos o funcionamento de aplicações informáticas.

O professor, pode visualizar no seu monitor, a turma a que está a dar uma aula, ou apenas um aluno, e acompanhá-los à medida que vão realizando os exercícios nos seus postos de trabalho, de modo semelhante ao funcionamento de um laboratório de idiomas.

NetSupport é composto por dois módulos distintos:

- O Programa Professor (chamado de programa Controlo);
- O Programa Aluno (chamado de programa Cliente).

Esta aplicação disponibiliza também um módulo para a elaboração de testes de avaliação. Este módulo inclui, um “*Desenhador de Provas*”, intuitivo que permite ao professor configurar os seus testes, com perguntas de formato texto, fotografias, áudio e vídeo, assim podem, nesta opção, ser criadas novas perguntas, novas provas/testes, manter perguntas e testes armazenados.



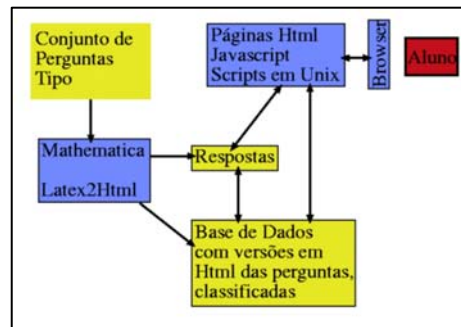
**Figura 3-10 – Interface do Módulo Professor – “Desenhador de Provas”**

### **3.1.9 - Perguntas de Escolha Múltipla On-line**

Perguntas de Escolha Múltipla on-line [21], é um gerador aleatório de perguntas, combinado com o conversor latex2html [22].

## CAPÍTULO 3

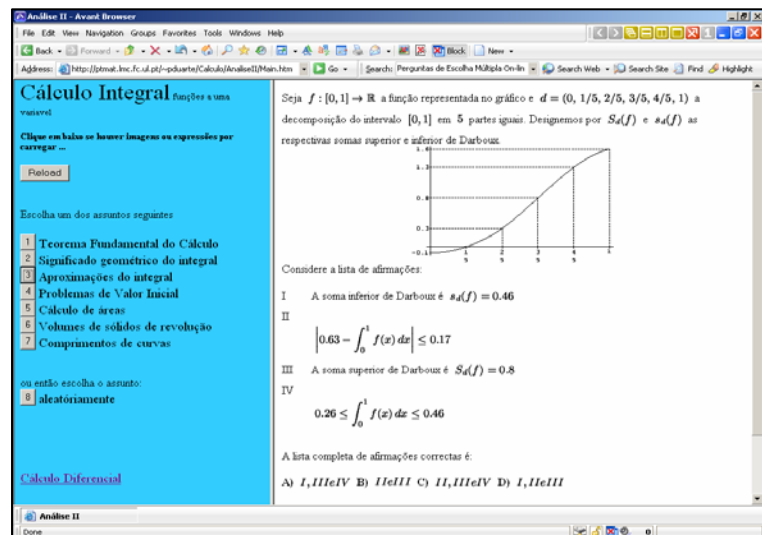
O funcionamento da aplicação está ilustrado na *Figura 3-11*:



**Figura 3-11 – Esquema de funcionamento da aplicação Perguntas de Escolha Múltipla On-line**

As perguntas são de escolha múltipla, e encontram-se numa base de dados organizadas por tema, para produzir uma questão em formato HTML, pode escolher-se pergunta um único tema, dentro dos temas disponíveis ou gerar aleatoriamente a pergunta. Finalmente, através de um JAVASCRIPT, são verificadas as respostas que também se encontram guardadas em outra base de dados.

Um exemplo do funcionamento desta aplicação está ilustrado na *Figura 3-12*, com exercícios de matemática – cálculo integral.



**Figura 3-12 – exemplo do uso do interface da aplicação - Perguntas de Escolha Múltipla On-line**

### 3.1.10 – Advanced eLearning Builder®

A aplicação comercial, Advanced eLearning Builder [23], foi desenvolvida para criar todo o material necessário para o funcionamento de uma plataforma de e-learning.

No que diz respeito a elaboração de testes, encontram-se nesta aplicação, vários tipos de exercícios: correspondência de colunas, escolha múltipla e desenvolvimento, onde se pode incluir texto, gráficos, objectos de multimédia (OLE) e ainda botões, para dotar os testes do aspecto mais atractivo possível, permitindo ainda mudança de tipo de letra, cores e alterações de tamanho.

Apenas com um clique de rato, as questões podem ser as adicionadas aos testes, tornando o uso da aplicação simples, para os professores que pretendam criar os seus testes, uma vez que a aplicação também disponibiliza modelos de testes, podem ser editados num processador de texto.

Na *Figura 3 -13*, pode observar-se a construção de um teste, nesta aplicação.

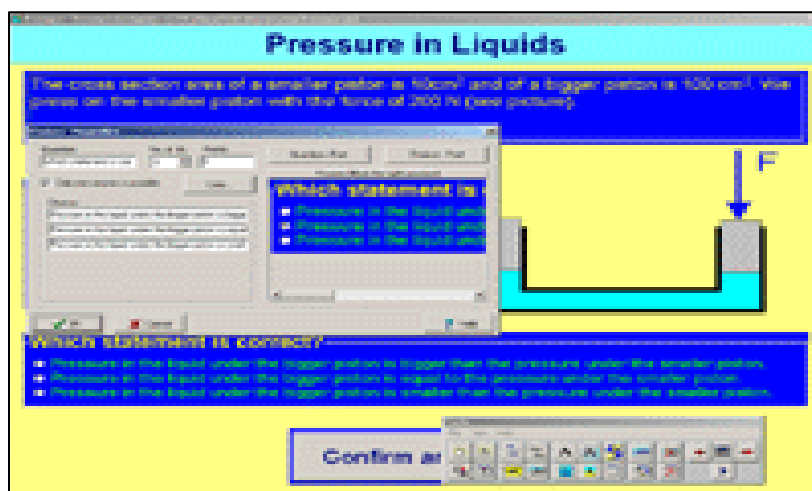


Figura 3-13 – Interface de construção de teste

### 3.1.11 - FORMARE®

A aplicação FORMARE ® [24], é uma aplicação comercial desenvolvida pela PT Inovação, como mais um exemplo de uma potente plataforma de e-learning, desenvolvida em Microsoft SQL2000 e ASP.

Esta aplicação também permite gerar automaticamente os testes de formação e avaliação, que como se pode observar na *Figura 3.14*, os testes podem ser de vários tipos.

Para cada teste, associam-se perguntas, que podem ser de escolha múltipla, verdadeira/falsa, desenvolvimento ou ainda para completar frases, sendo a correcção de todas elas com excepção das de desenvolvimento de correcção automática.

Como se pode observar, também, pela *Figura 3-14*, neste interface é possível, criar, editar e apagar perguntas.

Designação do questionário:

**Grupos de Perguntas** ⓘ  
Não existem grupos de perguntas associados a este questionário

Designação do grupo:

**Perguntas**

Grupo	Texto da Pergunta	Tipo	Peso	
<input type="checkbox"/>	O formare suporta soluções de ...	Escolha Múltipla	1	<a href="#">Editar</a> ✓ ↗
<input type="checkbox"/>	Quais dos seguintes benefícios...	Escolha Múltipla	1	<a href="#">Editar</a> ✓ ↗
<input type="checkbox"/>	Complete e seguinte frase:	Completar Frase	1	<a href="#">Editar</a> ✓ ↗
<input type="checkbox"/>	Relacione os campos seguintes	Relacionamento	1	<a href="#">Editar</a> ✓ ↗
<input type="checkbox"/>	Em que consiste o serviço de a...	Desenvolvimento	1	<a href="#">Editar</a> ✓ ↗
<input type="checkbox"/>	O anexo técnico do formare des...	Escolha Múltipla	1	<a href="#">Editar</a> ✓ ↗

Figura 3-14 – Interface de elaboração de teste diagnóstico

Um exemplo de teste produzido por esta aplicação pode ser observado, na *Figura 3-15*.

**Q-ecurso**  
**q-formare**

o formare suporta soluções da FE, em amb. internet/intranet e difusão de conteúdos mult/hipermédia (8, 7%)

Verdadeiro  
 Falso

completar a frase: (25%)

as 3 meninas [dropdown], [dropdown] e [dropdown] gostam muito de [dropdown] e de [dropdown]

benefícios q não fazem parte do formare (16, 3%)

redução de custos  
 descontos viagens  
 disponibilidade  
 roi elevado

relacionar campos (16, 3%)

Coluna A		Coluna B
tem acesso ao cronograma da acção	[dropdown]	<b>A</b> português
definir áreas temáticas	[dropdown]	<b>b</b> aluno/F/C
criar questionários correcção automática	[dropdown]	<b>c</b> scorm1.2
perfis utilizador	[dropdown]	<b>d</b> aluno
língua oficial	[dropdown]	<b>f</b> Formador
norma	[dropdown]	<b>e</b> coordenador

em que consiste o serviço de alojamento (25%)

[text area]

o anexo técnico do Formare descreve a versão 3.0 do formare (8, 7%)

Verdadeiro  
 Falso

Submeter Questionário Cancelar

**Figura 3-15 – Um exemplo de teste produzido pelo FORMARE**

**3.1.12 - MOODLE**

O Moodle [25], é um software para gestão de aprendizagem e trabalho permitindo a criação on-line, páginas de disciplinas, grupos de trabalho e comunidades de aprendizagem.

É *Open Source* e desenvolvido em PHP com recurso a base de dados.

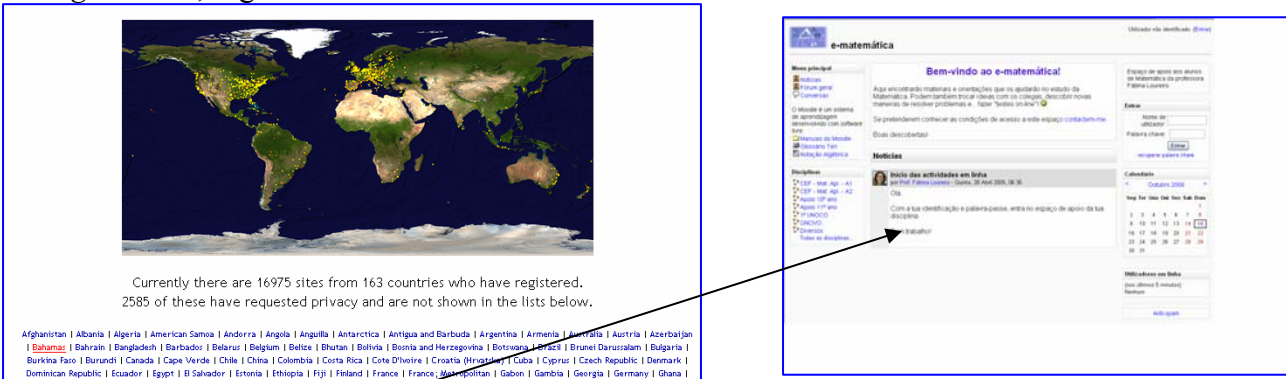
Possui os habituais perfis de utilizador: aluno, professor e administrador e as funções desenvolvidas para plataformas de E-learning:

- Inserção de conteúdos;
- Produção de documentos;
- Chats;
- Fóruns;
- Avaliação, etc.

É uma plataforma que pretende um constante desenvolvimento, para isso foi criado mais um perfil de utilizador: programadores. Para este tipo de utilizador é disponibilizado um conjunto de manuais sobre o estilo de programação do Moodle, bem como a sua arquitectura e estilo de interface.

Foi desenvolvida em várias línguas e é utilizada por um grande número de utilizadores, como se ilustra na *Figura 3.17*, e um dos sites que usam esta plataforma em Portugal ilustra-se na

*Figura 3.16*, seguinte:



The figure consists of two parts. On the left is a world map with yellow dots indicating registered sites across various countries. Below the map is a list of 163 countries. On the right is a screenshot of the Moodle interface for 'e-matematica', showing a login page with fields for name and password, and a calendar. An arrow points from the 'Portugal' entry in the country list to the Moodle interface.

Currently there are 16975 sites from 163 countries who have registered. 2585 of these have requested privacy and are not shown in the lists below.

Afghanistan | Albania | Algeria | American Samoa | Andorra | Angola | Antarctica | Antigua and Barbuda | Argentina | Armenia | Australia | Austria | Azerbaijan | **Bahamas** | Bahrain | Bangladesh | Barbados | Belarus | Belgium | Belize | Benin | Bolivia | Bosnia and Herzegovina | Botswana | Brazil | Brunei Darussalam | Bulgaria | Burkina Faso | Burundi | Canada | Cape Verde | Chile | China | Colombia | Costa Rica | Cote D'Ivoire | Croatia (Republic of) | Cuba | Cyprus | Czech Republic | Denmark | Dominican Republic | Ecuador | Egypt | El Salvador | Estonia | Ethiopia | Fiji | Finland | France | France (Metropolitan) | Gabon | Gambia | Georgia | Germany | Ghana | Greece | Guam | Guatemala | Guyana | Honduras | Hong Kong | Hungary | Iceland | India | Indonesia | Iran (Islamic Republic of) | Ireland | Israel | Italy | Jamaica | Japan | Jordan | Kazakhstan | Kenya | Korea, Republic of | Kosovo | Kuwait | Kyrgyzstan | Lao People's Democratic Republic | Latvia | Lebanon | Lithuania | Luxembourg | Macedonia, Former Yugoslav Republic of | Malaysia | Mali | Malta | Mexico | Moldova, Republic of | Monaco | Mongolia | Morocco | Mozambique | Myanmar | Namibia | Nepal | Netherlands | Netherlands Antilles | New Zealand | Nicaragua | Nigeria | Northern Mariana Islands | Norway | Oman | Pakistan | Palestine | Panama | Papua New Guinea | Paraguay | Peru | Philippines | Poland | **Portugal** | Puerto Rico | Qatar | Reunion | Romania | Russian Federation | Saint Lucia | Saint Vincent and the Grenadines | Samoa | Saudi Arabia | Senegal | Serbia and Montenegro | Singapore | Slovakia (Slovak Republic) | Slovenia | Solomon Islands | South Africa | Spain | Sri Lanka | Sudan | Suriname | Sweden | Switzerland | Syrian Arab Republic | Taiwan | Tajikistan | Tanzania | United Republic of | Thailand | Togo | Tokelau | Trinidad and Tobago | Tunisia | Turkey | Turkmenistan | Ukraine | United Arab Emirates | United Kingdom | United States of America | Uruguay | Uzbekistan | Venezuela | Vietnam | Virgin Islands (U.S.) | Wallis | Yemen | Zambia | Zimbabwe | SHOW ALL

**Figura 3-16 – <http://e-matematica.pt>**

### **3.1.13 IMS Global Learning Consortium Inc. ®**

O IMS Global Learning Consortium Inc. ® [3] é um consórcio global do qual fazem parte membros de organizações educacionais, comerciais e governamentais com o objectivo de desenvolver uma arquitectura “*open-source*” para o ensino on-line.

Possuem como objectivo estabelecer um padrão das tecnologias para as aplicações de e-Learning, com o intuito de estabelecer referências, concentrando-se na especificação de modelos que determinam padrões para o formato de conteúdos, sintaxe e semântica dos meta dados.

As vantagens da utilização de conceitos como “*learning objects*” estão fundamentadas principalmente pela utilização de modelos padrão. Já que os “*learning objects*” são compostos por texto, imagens, sons e outras informações de acordo com o contexto em que são utilizados.

Dada a diversidade do seu conteúdo importantes organizações como IEEE e o IMS Global Learning Consortium Inc. ®, propuseram o padrão Learning Object Metadata (LOM), que é um padrão que descreve os “*learning objects*”, permitindo assim, que estes sejam classificados, reutilizados e localizados por ferramentas de pesquisa.

Existe uma vasta especificação no que diz respeito aos exercícios e que foi atentamente analisada.

**3.2 - Conclusões**

A tabela seguinte, ilustra o resumo das aplicações referidas anteriormente, retirando da descrição das mesmas, as características mais relevantes para o desenvolvimento do trabalho proposto, com excepção para o consórcio IMS Global Learning Inc.® [3], que não é comparável define um conjunto de especificações que poderão ser utilizadas em diversas aplicações tendo em conta a sua vasta dimensão.

<b>Nome da Aplicação</b>	<b>Ambiente de Desenvolvimento</b>	<b>Tipo de Questões</b>	<b>Académica/ Comercial</b>
<b>Webduc</b>	- PHP - MySql - Apache Server	-Desenvolvimento -Correspondência de Colunas -Escolha Múltipla -Verdadeiro/Falso	Académica
<b>GedEx Web</b>	- HTML - CSS - PHP - Macromedia Flash 5.0 - Apache Server - MySql	-Desenvolvimento -Escolha Múltipla	Académica
<b>TestGIP</b>	- Microsoft Visual Basic 6.0	-Escolha Múltipla	Académica
<b>SAEL</b>	- Delphi Developer 2.01	-Desenvolvimento -Correspondência de Colunas -Escolha Múltipla -Verdadeiro/Falso	Académica
<b>EasyTest</b>	- JAVA - HTML	-Escolha Múltipla	Académica
<b>SEATT</b>	- JAVA - API JDBC - HTML - Microsoft Access	- Escolha Múltipla	Académica

<b>SGA Ex</b>	- PHP	-Correspondência de Colunas -Escolha Múltipla -Verdadeiro/Falso	Académica
<b>MOODLE</b>	- PHP	-Desenvolvimento -Completar frases -Escolha Múltipla -Verdadeiro/Falso	Académico
<b>NetSupport School®</b>	- Software proprietário	-Desenvolvimento -Correspondência de Colunas -Escolha Múltipla -Verdadeiro/Falso	Comercial
<b>Advanced eLearning Builder®</b>	- Software proprietário	-Desenvolvimento -Correspondência de Colunas -Escolha Múltipla -Verdadeiro/Falso	Comercial
<b>FORMARE®</b>	- Software proprietário	-Desenvolvimento -Completar frases -Escolha Múltipla -Verdadeiro/Falso	Comercial

**Tabela 3-1 – Tabela resumo das aplicações estudadas**

Da tabela resumo (*Tabela 3.1*), importa sobretudo ter em consideração as aplicações que implementam geração de testes de avaliação. Na sua maioria, verifica-se o recurso a bases de dados, quer para armazenar exercícios, quer testes de avaliação gerados.

Implementar uma solução alternativa, especialmente no que diz respeito ao armazenamento é um dos objectivos deste trabalho.

Tendo em conta os documentos produzidos para a implementação do projecto Ganesh [26], e em especial a citação que refere que uma base de dados “...*não é suficiente flexível para permitir a gestão de grandes quantidades de informação estruturada, especialmente de exercícios; implica a manutenção de demasiados formulários para gestão de dados que dificultam alterações à estrutura dos dados...*” [27], para além da estrutura rígida a que as bases de dados obedecem, surge tal como é referido na documentação do projecto uma alternativa, que é o uso de documentos estruturados.

## **CAPÍTULO 3**

## Capítulo 4 – Metodologia e Modelação da Informação

Neste capítulo vai ser ilustrada a metodologia e a modelação da informação para manipulação de documentos de apoio às aulas (fichas de trabalho) e documentos de treino e avaliação. Quer à metodologia, quer à linguagem XML, assim como ao protótipo deu-se o nome, respectivamente: Metodologia GerExa, Linguagem XML GerExa e Protótipo GerExa. Será efectuada a apresentação detalhada da linguagem XML, desenvolvida, e discutidas as ferramentas XML que darão suporte ao desenvolvimento do protótipo apresentado no Capítulo 5.

Os testes de avaliação e treino, bem como as fichas de trabalho são documentos centrados na manipulação de exercícios.

### 4.1 -Metodologia

A metodologia está ilustrada na *Figura 4-1* seguinte:



Figura 4-1 – Metodologia GerExa

## **CAPÍTULO 4**

A ideia desta metodologia assenta, num conjunto de exercícios que devem estar agrupados por temas, considerando as seguintes etapas:

**Definição de Temas** – Definir um conjunto de temas de acordo com os exercícios a manipular.

**Exemplo:** Para exercícios de Probabilidade, convém definir como tema principal “Probabilidades” e segundo os capítulos em que esta matéria se divida considerar sub-temas necessários: Probabilidade Simples, Probabilidade Condicionada, etc.

A definição e classificação de temas tem que permitir a adequação do número de sub-temas de acordo com as preferências de cada professor.

**Atribuição de Tipo de Exercício** – Existe uma diversidade de tipos de exercícios, que permitem correcção automática, como sejam escolha múltipla, resposta calculada, verdadeiro/falso e alguns tipos de exercícios de linguagens de programação. Para além destes, existem os exercícios de desenvolvimento, que devem ser considerados.

**Construção de Exercício** – Cada tipo de exercícios, deve permitir uma construção adequada de acordo com as suas especificações particulares. Cada exercício está incluído num determinado tema.

Nos exercícios de correcção automática, será inserida a quando da construção do exercício qual a(s) resposta(s) certa(s) no caso dos exercícios de escolha múltipla ou resposta múltipla, a resposta verdadeira no caso de exercícios Verdadeiro/Falso, etc.

**Gerar documentos de avaliação ou apoio** – Com base nos exercícios existentes construir os diferentes tipos de documentos: testes de avaliação ou treino e fichas de trabalho.

Esta construção terá que ser o mais flexível possível, poderá ser:

**Completamente automática** – com base em escolhas efectuadas pelo professor, por exemplo o tema ou temas, ou o tipo de exercícios, o documento é gerado;

**Assistida** – a escolha poderá ser efectuada exercício a exercício;

**Semi-automática** - permitir gerar automaticamente um conjunto e efectuar uma escolha do conjunto produzido.

**Correcção** – O aluno poderá responder a testes de avaliação e treino, que são os únicos documentos submetidos a correcção:

**Correcção Automática** - A classificação das respostas às questões que não necessitem da supervisão do professor, é produzida de imediato.

**Correcção Assistida** – O professor poderá efectuar a correcção das questões que não permitirem correcção automática de modo a completar a avaliação do aluno.

Definidas as etapas a considerar para a definição da linguagem, é necessário saber exactamente o tipo de informação a ser manipulada, nomeadamente no que diz respeito aos exercícios.

### **4.2 - Modulação da Informação**

Para além da informação relativa aos intervenientes: alunos e professores, surge em destaque a modelação dos exercícios que são a base de desenvolvimento da linguagem e que darão origem à geração de documentos.

Assim e dado que existe uma quantidade significativa de informação associada aos exercícios, a guardar e documentos a produzir, é necessário organizar e distinguir claramente cada tipo de exercício de forma a esquematizar e estruturar a informação relativa.

Assim, para a modulação desta informação e depois da análise efectuada no capítulo 2, consideram-se os seguintes tipos de exercícios e respectivos exemplos:

#### **4.2.1 - Escolha múltipla**

Neste tipo de exercício o objectivo é seleccionar uma das hipóteses apresentadas, apenas uma está correcta.

*Exemplo:* Num programa em C a função primeira função a executar é:

- a) `return()`
- b) `#include`
- c) `void()`
- d) `main()`

#### **4.2.2 - Resposta múltipla**

Neste tipo de exercício o objectivo é seleccionar uma ou mais hipóteses apresentadas, uma ou mais poderão estar correctas.

**Exemplo:** São cidades de Portugal:

- a) Porto
- b) Madrid
- c) Faro
- d) Salamanca
- e) Paris

### 4.2.3 - Correspondência de colunas

O objectivo, neste tipo de exercício é efectuar a correspondência entre as opções da coluna 1 e 2, ou seja, cada hipótese da coluna 1 tem uma hipótese correspondente na coluna 2.

**Exemplo:** Qual a ordem de cada uma das funções:

<b>Coluna 1</b>	<b>Coluna 2</b>	Coluna1	Coluna2
1 – $n \log n$	1. $O(n^3)$	<b>1</b>	
2 – $2n^2+n$	2. $O(1)$	<b>2</b>	
3 – c, com c constante	3. $O(n^2)$	<b>3</b>	

### 4.2.4 - Verdadeiro/Falso

Como o próprio nome indica, neste tipo de exercícios pretende-se o valor lógico de uma afirmação.

**Exemplo:** Indique se é verdadeira ou falsa a seguinte afirmação:

“A amostra em estatística representa um subconjunto da população a estudar.”

### **4.2.5 - Resposta calculada**

Para este tipo de exercícios, existe uma resposta imediata de cálculo.

*Exemplo:* Considere a matriz  $M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$

Calcule o valor próprio dominante da matriz M.

### **4.2.6 - Desenvolvimento**

As questões de desenvolvimento, são as únicas para as quais se torna difícil a correcção automática.

*Exemplo:* Considere a matriz  $A = \begin{bmatrix} 4 & 1 & 2 \\ 0 & 1 & 0 \\ 8 & 4 & 5 \end{bmatrix}$  e o vector  $b = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$ .

Resolva o sistema  $Ax=b$ , utilizando o Método de Eliminação de Gauss.

Este exemplo pode ser aproveitado, para discutir outra questão, a considerada na metodologia. Se poder ser executado o método de Eliminação de Gauss, este exercício podia ser classificado como um exercício de resposta calculada. Assim, dependendo do tipo de temas a tratar podem ser definidas aplicações auxiliares que executem automaticamente determinadas funções, como sejam os exercícios que envolvam algoritmos para cálculo de uma solução.

Tendo em conta as fases estipuladas para a metodologia e a modelação da informação, no que diz respeito aos exercícios, vão ser tecidas algumas considerações sobre as tecnologias a adoptar, nomeadamente as vantagens e desvantagens destas abordagens e os motivos da escolha das mesmas.

### **4.3 - Tecnologias XML**

Tendo em conta que um dos objectivos do trabalho é a disponibilização da informação na WEB, a escolha das tecnologias terá obedecer aos requisitos exigidos pela mesma.

Para a formatação e exibição de documentos na WEB, a linguagem mais divulgada é o HTML, que no entanto dado o crescimento da utilização da Internet e o recurso cada vez mais frequente a aplicações, a linguagem começou a ter lacunas no que diz respeito à troca de informação.

Esta evolução deu origem a que um grupo de organizações denominado W3C, apresenta-se uma nova linguagem que permite troca de dados e que pode ser executada em diferentes plataformas de forma unificada - o XML (eXtensible Markup Language), que deriva de uma outra linguagem mais abrangente o SGML (Standard Generalized Markup Language) [28] que é utilizada para lidar com documentos de texto e essencialmente, permite descrever a estrutura lógica de documentos através da definição de gramáticas, permitindo a sua classificação hierárquica. No entanto trata-se de uma linguagem muito extensa para colocar na WEB.

O XML [29, 30, 31, 32], surge como objectivo do desenvolvimento de uma linguagem que elimine as limitações do SGML, usando no entanto todas as suas potencialidades, tendo em conta a principal função de troca de informações na WEB.

## **CAPÍTULO 4**

Num documento XML, os dados são estruturados independentemente das aplicações ou plataformas em que são utilizados.

Como se pode observar pelos exemplos, apresentados na modelação da linguagem, Secção 2 do Capítulo 4, o objectivo é construir uma estrutura que para além de suportar os tipos de exercícios definidos, permita que eles contenham os vários elementos, como sejam fórmulas matemáticas e imagens, por exemplo.

Assim, considerando o tipo de informação a tratar, e as principais características do XML, quais os fundamentos da escolha?

- Representação estruturada dos dados:

Em XML é possível criar uma linguagem adaptada a cada tipo de necessidade, sem descurar um aspecto fundamental que é a existência de uma estrutura de dados. Na maior parte das aplicações essa estrutura obtêm-se com o recurso a bases de dados, no entanto o XML providencia-a mediante um conjunto de regras simples, mas que são suficientes para garantir a estruturação da informação.

- Portabilidade:

A portabilidade é uma das características fundamentais do XML, já que a linguagem pode ser usada independente da plataforma ou sistema operativo. É importante poder favorecer a transacção de documentos, que obedeçam a uma estrutura previamente definida, podendo assim integrar numa aplicação informação proveniente de várias fontes.

- Sem ambiguidades:

Um documento XML, pode ser definido de tal forma, que não haja duas maneiras de interpretar os nomes, a ordem e a hierarquia. No caso da metodologia a desenvolver, existem vários tipos de exercícios e documentos a produzir, pelo que é necessário que estejam univocamente estruturados.

- Flexibilidade:

Podem ser definidas a partir de um mesmo documento, diferentes formas de visualização dos dados, uma vez que a linguagem XML define apenas a estrutura de dados e não a forma como estes vão ser apresentados, podendo assim variar consoante as necessidades de cada aplicação. Como a informação, sobre a forma de apresentação dos documentos, está definida fora do documento XML, essa configuração pode ser usada para gerar documentos distintos e a sua alteração é mais fácil.

- Desenvolvimento de aplicações flexíveis para a WEB:

Os documentos XML são, facilmente manipulados por aplicações. Podem ser apresentados de imediato ou estar sujeitos a um determinado processamento.

No caso do protótipo GerExa, para além da inserção de informação será necessário efectuar consultas, gerar novos documentos, etc., pelo que a flexibilidade e o comportamento do XML na WEB são factores bastante relevantes.

- Fácil distribuição na WEB:

Os documentos XML, possuem um formato aberto em texto, que pode ser distribuído por HTTP, protocolo standard da WEB.

Considerando os aspectos referidos anteriormente, o XML será usado para estruturar a informação, fazendo uso das tecnologias de manipulação que lhe estão associadas, como sejam: DOM, SAX, ELEMENT TREE, CSS, XSL e XSLT.

### **4.4 - A linguagem XML GerExa**

Com base na modulação da informação elaborada na secção 2 do Capítulo 4, construiu-se uma linguagem XML, que como já foi referido na introdução do Capítulo 4, se designou GerExa. Apesar de existirem várias referências a linguagens na mesma área, especialmente definidas pela IMS [3] e pela SCORM (Shareable Content Object Reference Model) [33], são linguagens muito vastas, tornando-se mais simples manipular uma linguagem elaborada à “medida” para a realidade a tratar.

Para descrever uma linguagem XML, usa-se um esquema ou seja uma gramática. Vão ser dados exemplos de esquemas utilizados actualmente.

Considere-se o exemplo seguinte, que representa em XML os dados de matrícula de um aluno:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  <matricula>
    <dadosAluno>
      <numero>1047</ numero >
      <nome>João Sebastião Oliveira</nome>
      <curso>Engenharia Informática </curso>
      <anolectivo>2005/2006</ anolectivo >
    </dadosAluno>
  </matricula>
```

Analisem-se as representações possíveis, para os esquemas que descrevem os documentos XML, entre elas, o XMLSchema [34], o RELAX NG [35] e os DTD's [36]:

Assim, para esquematizar o ficheiro XML anterior, usando o primeiro tipo de representação que é o XMLSchema, tem-se:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xml:lang="PT">
  <xsd:element name="matricula">
    <xsd:complexType>
      <xsd:element name="dadosAluno">
        <xsd:complexType>
          <xsd:sequence ">
            <xsd:element name="numero" type="xs:string"/>
            <xsd:element name="nome" type="xs:string"/>
            <xsd:element name="curso" type="xs:string"/>
            <xsd:element name="anolectivo" type="xs:string"/>
          </xsd:sequence >
        </xsd:complexType >
      </xsd:complexType>
    </xsd:element>
  </xs:schema>
```

## CAPÍTULO 4

A segunda esquematização, é o RELAX NG, e para o ficheiro XML exemplo tem-se:

```
<element name="matricula">
  <element name="dadosAluno">
    <element name="numero">
      <data type="token"/></element >
    <element name="nome">
      <data type="token"/></element >
    <element name="curso">
      <data type="token"/></element >
    <element name="anolectivo">
      <data type="token"/></element >
  </element>
</element>
```

Finalmente, a última forma de representação escolhida são os DTD's, para o exemplo tem-se:

```
<!ELEMENT matricula (dadosAluno)>
<!ELEMENT dadosAluno (numero, nome, curso, anolectivo)>
<!ELEMENT numero (#PCDATA)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT curso (#PCDATA)>
<!ELEMENT anolectivo (#PCDATA)>
```

Note-se que cada um destes esquemas têm expressividades diferentes, podendo uns exprimir aspectos que os outros não representam.

A linguagem XML GerExa, será descrita usando DTDs ("Document Type Declaration") [36], para cada uma das suas componentes: Temas, Exercícios e Testes.

A escolha dos DTD's, surge para apresentar de forma sucinta o que se pretende para cada um dos documentos XML a estruturar, sem especificar tipos de dados, uma vez que todas as validações de tipos irão posteriormente ser efectuadas, por uma linguagem de programação.

O DTD, permite fazer uma declaração dos elementos que podem existir num determinado documento, bem como quais os atributos de cada elemento, que são os aspectos mais relevantes para a definição da linguagem.

Embora os outros esquemas permitam maior expressividade e flexibilidade optou-se pela simplicidade e pela existência de um maior número de ferramentas disponíveis para validação.

### 4.4.1 – Temas

Os exercícios devem pertencer a um tema (unidade de matéria), por exemplo de acordo com a ACM Computing Classification System [37]. Cada exercício deve obedecer a uma classificação que na linguagem XML GerExa, cada utilizador define, considerando o número de níveis que considerar necessário.

Cada tema poderá ser nó principal sem sub-temas, onde apenas se incluem exercícios, ou ter uma estrutura ilimitada de sub-temas.

O DTD seguinte definido pelo ACM, representa essa estrutura:

```
<!ELEMENT node (isComposedby|isRelatedTo|hasNote)*>
<!ATTLIST node id CDATA #IMPLIED
              label CDATA #IMPLIED>
<!ELEMENT isComposedBy (node+)>
<!ELEMENT isRelatedTo (node+)>
<!ELEMENT hasNote (#PCDATA)>
<!ATTLIST hasNote type CDATA #REQUIRED>
```

E é Ilustrado na *Figura 4-2*, seguinte:

Neste DTD, esta representado um esquema de nós e sub-nós.

Assim um nó representado no DTD, por “*node*”, terá apenas como atributos o “*label*” que representa o nome do nó e o “*id*”, um identificador.

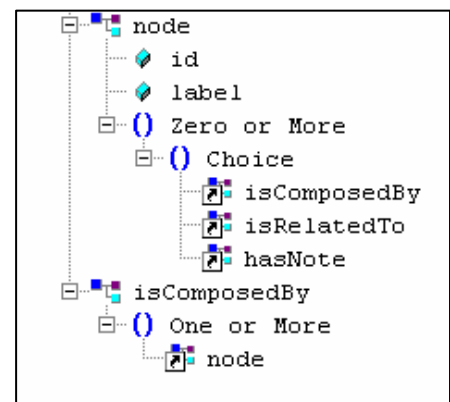


Figura 4-2 – Representação em árvore do elemento “node”

A representação de sub-nó, no DTD, será identificada por “*isComposedBy*”.

A possível relação com outros nós será identificada por “*isRelatedTo*” e a referência a anotações ficará representada em “*hasNote*”.

#### 4.4.2 Exercícios

O aspecto mais importante da definição da linguagem XML, diz respeito aos exercícios.

Depois de analisadas as possibilidades ou tipos de exercício, considerou-se o esquema de DTD, ilustrado na *Figura 4-3*:

Cada exercício terá um “*id\_exercicio*” a identificá-lo, terá o “*id\_node*”, que representa o tema ao qual pertence e estará associado a metadados, representados pelo “*id\_md*”.

Terá ter um enunciado opcional, que contenha texto, fórmulas matemáticas, fragmentos de programas descritos numa linguagem e/ou imagem, e várias questões. Caso haja um enunciado comum às questões, isso corresponde, aos muito utilizados, exercícios com alíneas.

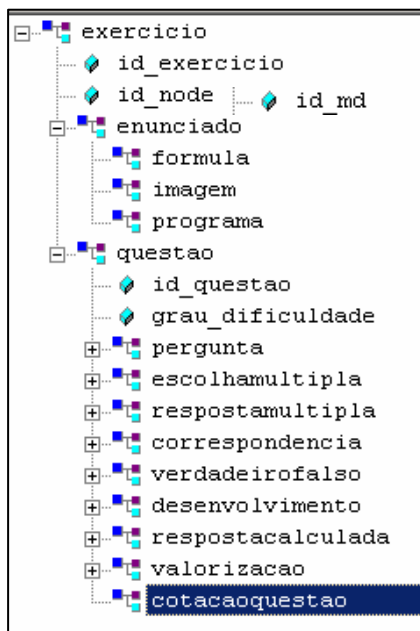


Figura 4-3 - Representação em árvore do elemento “exercício”

## CAPÍTULO 4

O enunciado não é obrigatório uma vez que cada questão tem um elemento pergunta, que pode dispensar o uso de enunciado, como se exemplifica a seguir:

```
<enunciado> Considere o lançamento de duas moedas.</enunciado>
  <questao id="1" grau_dificuldade="2">
    <pergunta> Qual a probabilidade de sair cara? </pergunta>
  </questao>
```

Para este mesmo enunciado poderia ter-se outra questão, por exemplo:

```
<questao id="2" grau_dificuldade="2" >
  <pergunta> Qual a probabilidade de sair coroa? </pergunta>
</questao>
```

Mas pode optar-se por colocar tudo, dentro do elemento pergunta, prescindindo do enunciado como por exemplo:

```
<pergunta>
  No lançamento de duas moedas qual a probabilidade de sair coroa?
</pergunta>
```

O extracto do DTD - exercício que ilustra a situação descrita, é o seguinte:

```
<!ELEMENT exercicio ((enunciado)?,(questao)+)>
<!ATTLIST exercicio
  id_exercicio ID #REQUIRED
  id_node IDREF #REQUIRED
  id_md IDREF #REQUIRED>
<!ELEMENT enunciado (#PCDATA|formula|imagem|programa)*>
```

Cada questão tem associado um tipo, do conjunto de tipos de questão, que foram definidos com base no estudo preliminar efectuado na secção 2, do Capítulo 4.

Considera-se também relevante atribuir a cada questão um grau de dificuldade, que poderá ser usado como critério para a produção dos documentos de avaliação e de treino.

Sendo a parte do DTD correspondente, a seguinte:

```
<!ELEMENT questao
(pergunta,(escolhamultipla|respostamultipla|correspondencia|verdadeirofalso|
desenvolvimento|respostacalculada),(valorizacao)?,(cotacaoquestao)?)>
<!ATTLIST questao
    id_questao ID #REQUIRED
    grau_dificuldade (1|2|3|4) "2">
<!ELEMENT pergunta (#PCDATA|formula|imagem|programa)*>
```

Cada um dos tipos de questão definidos é tratado de forma separada, como se descreve nos parágrafos seguintes e foi referido na definição da metodologia Secção 1 do Capítulo 4.

#### **4.4.2.1 Escolha Múltipla**

Para a escolha múltipla, estipulou-se como aceitável a inserção mínima de três opções, sendo esta opção posteriormente configurável, podendo conter para além do habitual texto, fórmulas matemáticas e/ou imagens, tendo cada uma das opções um identificador único.

Como existe uma única opção correcta, ficou especificado que a resposta dessa opção terá o valor lógico um e todas as outras terão o valor lógico zero.

## CAPÍTULO 4

A utilidade desta escolha serve para efeitos de correcção automática dos exercícios, uma vez que para cada exercício é indicada a resposta correcta.

A estrutura deste tipo de questão ilustra-se na *Figura 4-4*, seguinte:

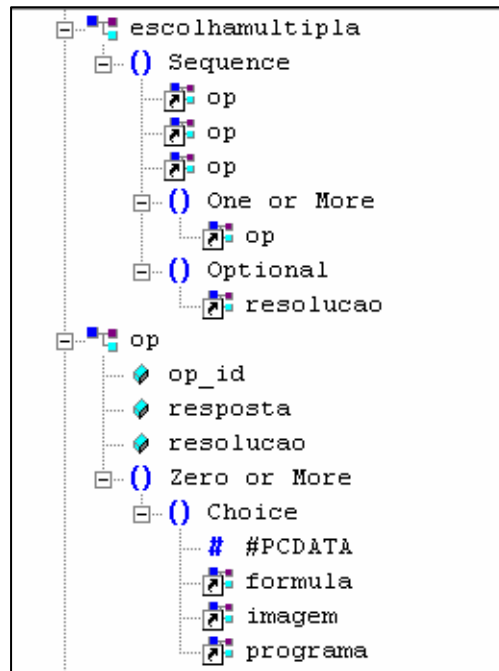


Figura 4-4– Representação em árvore do elemento “escolhamultipla”

O DTD que representa a estrutura ilustrada é o seguinte:

```
<!ELEMENT escolhamultipla (op,op,op,(op)+,(resolucao)?)>
<!ELEMENT op (#PCDATA|formula|imagem|programa)*>
<!ATTLIST op
  op_id (1|2|3|4|5) "1"
  resposta (0|1) "0"
  resolucao (0|1) #IMPLIED>
```

#### 4.4.2.2 Resposta Múltipla

A estrutura seguinte, diz respeito ao elemento resposta múltipla, que representa o tipo de questão com o mesmo nome. Neste tipo de questões podem ser inseridas no mínimo três alíneas.

Tal como nos exercícios de escolha múltipla, nos exercícios de resposta múltipla o número de opções será configurável.

Também para este tipo de exercício, foi considerado, o facto de se terem respostas verdadeiras e respostas falsas, seguindo esquema das questões de escolha múltipla, as respostas verdadeiras são assinaladas com o valor lógico um e as falsas com o valor lógico 0, tanto para indicar a(s) resposta(s) correctas como também para facilitar a correcção automática.

A ilustração deste tipo de questão, é efectuada na *Figura 4-5*:

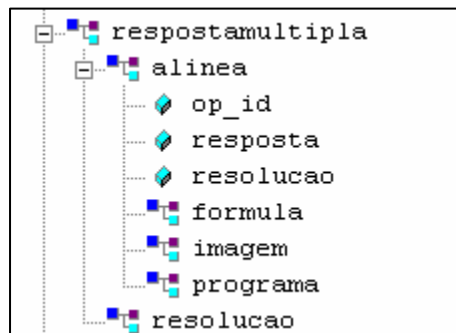


Figura 4-5- Representação em árvore do elemento “respostamultipla”

O respectivo DTD será:

```

<!ELEMENT respostamultipla (alinea,alinea,alinea,(alinea)+,(resolucao)?)>
<!ELEMENT alinea (#PCDATA|formula|imagem|programa)*>
<!ATTLIST alinea
  op_id (1|2|3|4|5|6|7|8) "1"
  resposta (0|1) "0"
  resolucao (0|1) #IMPLIED>
  
```

### 4.4.2.3 Correspondência de Colunas

O elemento, que representa as questões do tipo correspondência de colunas, pode ter no mínimo três colunas e no máximo de cinco.

A lógica de funcionamento deste tipo de exercício, implica que a escolha do mesmo número de colunas 1 e colunas 2, para efectuar a correspondência entre elas.

Assim, para a coluna 2 foi definido um atributo identificador, para que se possa fixar a numeração de uma coluna e efectuar a correspondência com os valores da outra.

Assim sendo o campo resolução considerado para a resposta do aluno, aparece apenas na coluna 1.

Em concordância com as duas situações anteriores, também em cada coluna pode existir texto, imagens, fórmulas ou programas.

A *Figura 4-6* seguinte ilustra, representa o esquema em árvore da situação descrita:

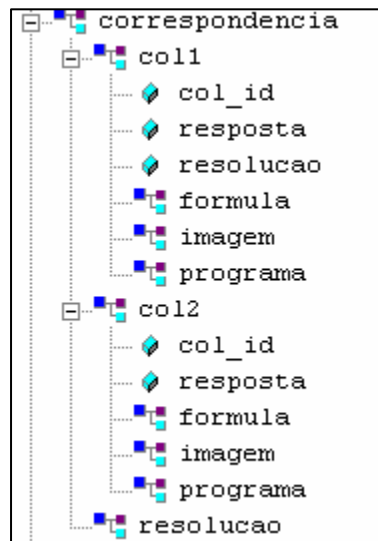


Figura 4-6- Representação em árvore do “elemento correspondência de colunas”

O respectivo DTD será:

```

<!ELEMENT correspondencia
(col1,col1,col1,(col1)?,(col1)?,col2,col2,col2,(col2)?,(col2)?,(resolucao)?)>
<!ELEMENT col1 (#PCDATA|formula|imagem|programa)*>
<!ATTLIST col1
    col_id (1|2|3|4|5) "1"
    resposta (1|2|3|4|5) "1"
    resolucao (1|2|3|4|5) #IMPLIED>
<!ELEMENT col2 (#PCDATA|formula|imagem|programa)*>
<!ATTLIST col2
    col_id (1|2|3|4|5) "1"
    resposta (1|2|3|4|5) "1">
    
```

#### **4.4.2.4 Verdadeiro/Falso**

O elemento que representa as questões do tipo verdadeiro/falso, apenas tem atributos, uma vez que como a pergunta está definida como um elemento da questão, só é necessário considerar a resposta correcta e a resolução do aluno, obedecendo ao facto de a esta questão só poder responder se é verdadeira ou falsa.

O DTD correspondente será:

```

<!ELEMENT verdadeirofalso EMPTY>
<!ATTLIST verdadeirofalso
    resposta (V|F) "F"
    resolucao (V|F) #IMPLIED>
    
```

#### 4.4.2.5 Desenvolvimento

Quanto às questões de desenvolvimento a limitação é maior, uma vez que torna-se difícil senão impossível definir critérios que facilitem a correção automática, assim estes exercícios contêm apenas um elemento que permite a inserção da resposta do aluno e outro que permitirá a posterior correção por parte do professor. Como se pode observar no DTD seguinte:

```
<!ELEMENT desenvolvimento ((respostadesenv)?,(resolucaodesenv)?)>  
<!ELEMENT respostadesenv (#PCDATA|formula|imagem|programa)*>  
<!ELEMENT resolucaodesenv (#PCDATA|formula|imagem|programa)*>
```

#### 4.4.2.6 Resposta Calculada

Por último, a estrutura seguinte, representa as questões de resposta calculada que necessitam dos elementos para a solução e resposta do aluno, que como já foi referido, apenas aparecerá nos testes.

A *Figura 4-7* seguinte, permite observar a estrutura em árvore deste tipo de questão:

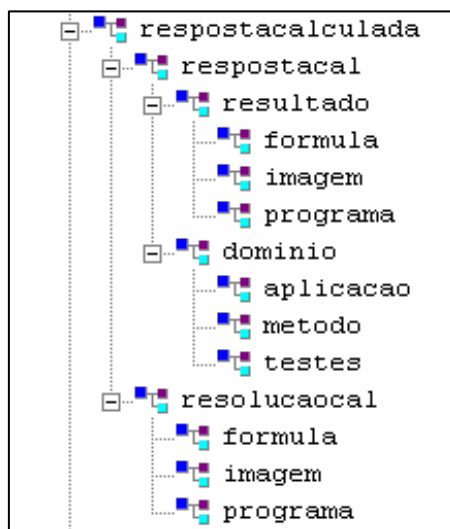


Figura 4-7- Representação em árvore do elemento “respostacalculada”

A resposta a este tipo de questão, pode ser pensada futuramente, como resposta produzida ou calculada por uma aplicação auxiliar.

Por exemplo, pode ter-se uma pergunta sobre programação do tipo “Dado o seguinte código, qual o OUTPUT produzido?”, e recorrer-se à execução de programa específico que produza o OUTPUT, e o compare com a resposta dada pelo aluno.

De seguida, apresenta-se o respectivo DTD:

```
<!ELEMENT respostacalculada ((respostacal)?,(resolucaocal)?)>
<!ELEMENT respostacal (resultado|(dominio)?)>
<!ELEMENT resolucaocal (#PCDATA|formula|imagem|programa)*>
<!ELEMENT resultado (#PCDATA|formula|imagem|programa)*>
<!ELEMENT dominio (aplicacao,metodo,testes)>
<!ELEMENT aplicacao (#PCDATA)>
<!ELEMENT metodo (#PCDATA)>
<!ELEMENT testes ANY>
```

Para os exercícios, foram ainda definidos, mais alguns elementos que se referem à cotação que o professor pode atribuir, a cada questão.

O elemento valorização inclui a cotação da questão e a classificação que o aluno obteve num documento de avaliação.

No elemento cotação, foram considerados os elementos, valor e valor desconto, o primeiro refere-se à percentagem de cada opção, no caso dos exercícios de escolha múltipla e resposta múltipla, nos restantes essa percentagem será 100%, uma vez que a pergunta é única; o segundo refere-se ao valor a descontar por cada resposta mal assinalada.

## **CAPÍTULO 4**

Como se ilustra no DTD:

```
<!ELEMENT valorizacao (cotacao,(classificacao?))>  
<!ELEMENT cotacao (valor,valordesconto)>  
<!ELEMENT classificacao (#PCDATA)>  
<!ELEMENT valor (#PCDATA)>  
<!ELEMENT valordesconto (#PCDATA)>  
<!ELEMENT cotacaoquestao (#PCDATA)>
```

Faltam apenas os elementos genéricos do exercício, representados no DTD, são:

```
<!ELEMENT formula (#PCDATA)>  
<!ELEMENT imagem (#PCDATA)>  
<!ELEMENT programa (#PCDATA)>
```

Assim o DTD completo do elemento exercício será:

```

<?xml version="1.0" encoding="UTF-16LE"?>
<!ELEMENT exercicio ((enunciado)?,(questao)+)>
<!ATTLIST exercicio
    id_exercicio ID #REQUIRED
    id_node IDREF #REQUIRED
    id_md IDREF #REQUIRED>
<!ELEMENT enunciado (#PCDATA|formula|imagem|programa)*>
<!ELEMENT questao
(pergunta,(escolhamultipla|respostamultipla|correspondencia|verdadeirofalso|desenvolvimento|respostacalculada),(valorizacao)?,(cotacaoquestao)?)>
<!ATTLIST questao
    id_questao ID #REQUIRED
    grau_dificuldade (1|2|3|4) "2">
<!ELEMENT pergunta (#PCDATA|formula|imagem|programa)*>
<!ELEMENT escolhamultipla (op,op,op,(op)+,(resolucao)?)>
<!ELEMENT op (#PCDATA|formula|imagem|programa)*>
<!ATTLIST op
    op_id (1|2|3|4|5) "1"
    resposta (0|1) "0"
    resolucao (0|1) #IMPLIED>
<!ELEMENT respostamultipla (alinea,alinea,alinea,(alinea)+,(resolucao)?)>
<!ELEMENT alinea (#PCDATA|formula|imagem|programa)*>
<!ATTLIST alinea
    op_id (1|2|3|4|5|6|7|8) "1"
    resposta (0|1) "0"
    resolucao (0|1) #IMPLIED>
<!ELEMENT correspondencia (col1,col1,col1,(col1)?,(col1)?,col2,col2,col2,(col2)?,(col2)?,(resolucao)?)>
<!ELEMENT col1 (#PCDATA|formula|imagem|programa)*>
<!ATTLIST col1
    col_id (1|2|3|4|5) "1"
    resposta (1|2|3|4|5) "1"
    resolucao (1|2|3|4|5) #IMPLIED>

```

```
<!ELEMENT col2 (#PCDATA|formula|imagem|programa)*>
<!ATTLIST col2
  col_id (1|2|3|4|5) "1"
  resposta (1|2|3|4|5) "1"><!ELEMENT verdadeirofalso EMPTY>
<!ATTLIST verdadeirofalso
  resposta (V|F) "F"
  resolucao (V|F) #IMPLIED>
<!ELEMENT desenvolvimento ((respostadesenv)?,(resolucaodesenv)?>
<!ELEMENT respostadesenv (#PCDATA|formula|imagem|programa)*>
<!ELEMENT resolucaodesenv (#PCDATA|formula|imagem|programa)*>
<!ELEMENT respostacalculada ((respostacal)?,(resolucaocal)?>
<!ELEMENT respostacal (resultado|(dominio)?>
<!ELEMENT resolucaocal (#PCDATA|formula|imagem|programa)*>
<!ELEMENT resultado (#PCDATA|formula|imagem|programa)*>
<!ELEMENT dominio (aplicacao,metodo,testes)>
<!ELEMENT aplicacao (#PCDATA)>
<!ELEMENT metodo (#PCDATA)>
<!ELEMENT testes ANY>
<!ELEMENT valorizacao (cotacao,(classificacao)?>
<!ELEMENT cotacao (valor,valor desconto)>
<!ELEMENT classificacao (#PCDATA)>
<!ELEMENT valor (#PCDATA)>
<!ELEMENT valor desconto (#PCDATA)>
<!ELEMENT cotacaoquestao (#PCDATA)>
<!ELEMENT formula (#PCDATA)>
<!ELEMENT imagem (#PCDATA)>
<!ELEMENT programa (#PCDATA)>
```

De seguida apresentam-se dois exemplos, de como fica um exercício em XML usando o DTD anterior:

**Exercício de desenvolvimento:**

```
<exercício id=1 id_node=1.1 id_md=1>
```

```
  <enunciado>
```

Considere a matriz  $A = \begin{bmatrix} 4 & 1 & 2 \\ 0 & 1 & 0 \\ 8 & 4 & 5 \end{bmatrix}$  e o vector  $b = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$

```
  </enunciado>
```

```
  <questao id="1" grau_dificuldade="2">
```

```
    <pergunta> “Resolva o sistema  $Ax=b$ , utilizando o Método de Eliminação de Gauss e Factorização LU.”
```

```
    </pergunta>
```

```
    <valorizacao>
```

```
      <cotacao>
```

```
        <valor/>100
```

```
        <valordesconto/>0
```

```
      </cotacao>
```

```
    </valorizacao>
```

```
  </questao>
```

```
</exercício>
```

Note-se que o elemento enunciado teria que ter um elemento fórmula para representar a matriz e o vector, o conteúdo desses elementos será abordado no Capítulo 5.

**Exercício de Escolha Múltipla**

```
<exercício id=1 id_node=2.1 id_md=3>
  <enunciado>
    “Escolha a resposta correcta:”
  </enunciado>
  <questao id="1" grau_dificuldade="2">
    <pergunta>
      “Em C um programa começa com a função:”
    </pergunta>
    <escolha_multipla>
      <op op_id="1" resposta="0">
        “return()”
      </op>
      <op op_id="2" resposta="0">
        “#include”
      </op>
      <op op_id="3" resposta="0">
        “void()”
      </op>
      <op op_id="4" resposta="1">
        “main()”
      </op>
      <op op_id="5" resposta="0" resolucao="0">
        “#declare”
      </op>
    </escolha_multipla>
    <valorizacao>
      <cotacao>
        <valor/>20
        <valordesconto/>10
      </cotacao>
    </valorizacao>
  </questao></exercício>
```

### 4.4.3 MetaDados

Como já foi referido anteriormente, e se pode observar na *Figura 4-8*, aos exercícios estão associados os metadados com informação específica e relevante sobre o autor, datas criação, modificação, disponibilização, disciplina em que o exercício está a ser aplicado, ou a que se aplica, etc.

A definição destes metadados tem como referência os elementos normalizados do DUBLIN CORE [38], essa normalização prende-se com critérios de consistência com as comunidades que usam elementos metadata.

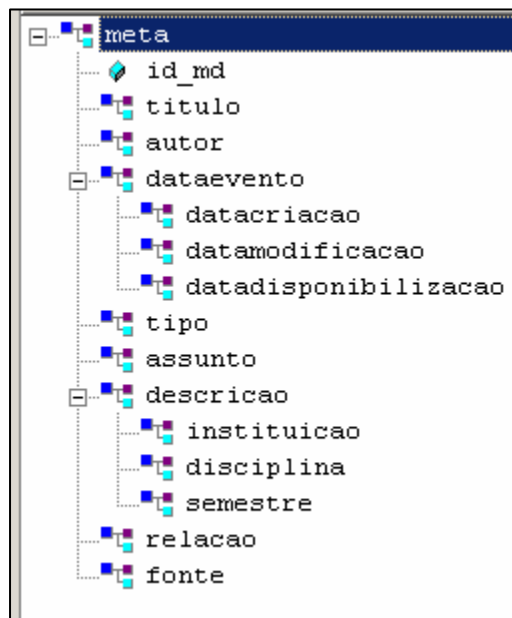


Figura 4-8- Representação em árvore do elemento “meta” (metadados)

O DTD seguinte ilustra a situação referida anteriormente:

```
<!ELEMENT meta (titulo,autor,dataevento,tipo,assunto,descricao,(relacao?),(fonte?))>
<!ATTLIST meta
    id_md ID #REQUIRED>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT dataevento (datacriacao|datamodificacao|datadisponibilizacao)>
<!ELEMENT datacriacao (#PCDATA)>
<!ELEMENT datamodificacao (#PCDATA)>
<!ELEMENT datadisponibilizacao (#PCDATA)>
<!ELEMENT assunto (#PCDATA)>
<!ELEMENT descricao (instituicao|disciplina|semestre)>
<!ELEMENT instituicao (#PCDATA)>
<!ELEMENT disciplina (#PCDATA)>
<!ELEMENT semestre (#PCDATA)>
<!ELEMENT relacao (#PCDATA)>
<!ELEMENT fonte (#PCDATA)>
```

### **4.4.4 - Aluno**

O elemento aluno, é um elemento significativo na definição da metodologia, uma vez que os documentos produzidos, sendo alguns de avaliação, devem ser realizados por alunos. Assim, para a linguagem XML GerExa, foram considerados alguns atributos básicos que identificam um aluno, bem como os elementos que permitem a validação do aluno num sistema WEB: login e password. Situação que se ilustra na *Figura 4-9*, seguinte:

```

<!ELEMENT aluno
(nome,instituicao,curso,ano,login,password)>
<!ATTLIST aluno numero ID #REQUIRED>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT instituicao (#PCDATA)>
<!ELEMENT curso (#PCDATA)>
<!ELEMENT ano (#PCDATA)>
<!ELEMENT login (#PCDATA)>
<!ELEMENT password (#PCDATA)>
    
```

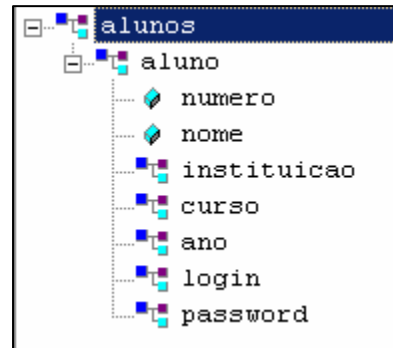


Figura 4-9– Representação em árvore do elemento “aluno”

#### 4.4.5 - Professor

O elemento Professor será análogo ao elemento Aluno com exceção dos elementos ano e curso.

#### 4.4.6 - Teste

O elemento teste permite modelar a informação relativa a um documentos de avaliação. Na *Figura 4-10*, pode observar-se essa mesma representação:

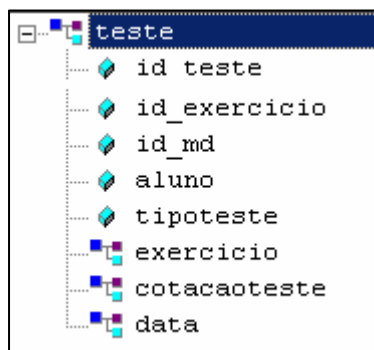


Figura 4-10- Representação em árvore do elemento “teste”

O elemento teste é composto por vários exercícios, e deve ser de um determinado tipo. Tipo esse que será: Exercícios de Treino, Ficha de Trabalho, Frequência, Exame de Época Normal, Exame de Recuso ou ainda Exame Especial.

## **CAPÍTULO 4**

O tipo de documento condiciona a cotação final do teste, uma vez que por exemplo em fichas de trabalho esta poderá não aparecer.

Seguidamente o DTD que ilustra a situação apresentada na *Figura 4-10*:

```
<!ELEMENT teste ((exercicio+), cotacaoteste?, data?)>
<!ATTLIST teste
  id_teste ID #REQUIRED
  id_exercicio IDREFS #REQUIRED
  id_md IDREF #REQUIRED
  aluno IDREF #IMPLIED
  tipoteste (Treino|Ficha|Frequencia|ExameN|ExameRec|ExameE) "Frequencia">
<!ELEMENT data (#PCDATA)>
<!ELEMENT cotacaoteste (#PCDATA)>
```

Os testes, tal como os exercícios, têm associado um conjunto de informação relevante, que não faz parte do conteúdo de um teste, pelo que aos testes também ficarão associados metadados.

### **4.5 -- Conclusão**

Neste Capítulo foi definida a Metodologia GerExa, a modulação da informação e que deram origem à linguagem GerExa apresentada. No Capítulo seguinte e com base nos aspectos definidos neste Capítulo será apresentado o Protótipo GerExa.

## Capítulo 5 - Desenvolvimento do Protótipo GerExa

Neste capítulo vão ser tecidas considerações sobre o desenvolvimento de um protótipo que exemplifica a metodologia e a linguagem definidas.

A primeira parte prende-se com a escolha das linguagens de suporte utilizadas e justificação dessa mesma escolha.

Na segunda parte são apresentadas as funcionalidades desejáveis numa futura aplicação que implemente quer a metodologia quer a linguagem. Nesta parte são referidas as funcionalidades de um modo geral e explicadas as funcionalidades que foram implementadas.

### **5.1 -Linguagem de suporte**

A linguagem de programação de suporte à aplicação que foi escolhida é o Python, pelos motivos que se passam a descrever.

#### **5.1.1 O Python**

Python [42, 43, 44], é uma linguagem de programação interpretada, interactiva e orientada a objectos (ainda que suporte outros paradigmas, como a programação funcional).

As características desta linguagem, que mais se destacam são:

- É uma linguagem com uma sintaxe simples;
- Os programas em Python são:
  - Fáceis de compreender;
  - Legíveis;
  - Fáceis de manter;

## **CAPÍTULO 5**

- Fáceis de completar;
- É uma linguagem orientada a objectos;
- É uma linguagem interpretada;
- Multiplataforma.

Hoje em dia é usada em diversas áreas, como servidores de aplicação e computação gráfica. Está disponível como linguagem *script* em aplicações como OpenOffice [45] (Python UNO Bridge) [46] e pode ser utilizado em *Stored Procedures* na base de dados PostgreSQL [47] (PL/Python) e está disponível para as mais diversas plataformas, desde Unix [48] (Linux, FreeBSD, Solaris, MacOS X etc.), Windows, versões antigas de MacOS.

Algumas das desvantagens desta linguagem são:

- Um pouco “lenta” (mas pode-se re-implementar o que está lento em C, ou em outra linguagem de programação);
- Não é tão utilizada como o Java, PERL ou PHP.

O Python será usado no desenvolvimento desta aplicação, uma vez que satisfaz os requisitos de programação necessários e surge como uma experiência no uso de uma linguagem “nova”, para a WEB.

Os exercícios e documentos produzidos pelo protótipo terão que ficar armazenados em ficheiros XML. Para efectuar o processamento desses documentos existem duas API's padrão: o DOM (Document Object Model) e SAX (Simple API for XML), para além de outras. Na linguagem de programação Python trabalha-se eficazmente com ambas.

### **5.1.1.1 DOM & SAX**

O DOM e o SAX [44, 49, 50, 51] são duas API's que permitem interpretar, gerar e extrair dados de documentos XML, cada uma delas com finalidades diferentes.

### **5.1.1.2 DOM**

O DOM, funciona, construindo a partir de um documento XML uma estrutura hierárquica de objectos, ou seja, constrói uma árvore que permite a navegação na estrutura do documento.

Serve para:

- Criar um novo documento XML;
- Percorrer a árvore XML construída;
- Permite no que diz respeito a elementos, atributos, etc.:
- Modificar;
- Remover;
- Adicionar.

### **5.1.1.3 SAX**

O SAX, é considerado mais simples que o DOM, uma vez que disponibiliza métodos que respondem a eventos que vão sendo produzidos à medida que o documento é processado.

O funcionamento do SAX é basicamente a resposta a eventos durante a leitura do documento, alguns desses eventos podem ser:

- Início de um documento;
- Fim de um documento;
- Início de um determinado elemento;
- Fim de um determinado elemento;
- etc.

**5.1.1.4 DOM vs SAX**

DOM	SAX
<ul style="list-style-type: none"><li>- Gasta mais memória em documentos grandes e/ou com uma hierarquia mais extensa;</li><li>- Disponibiliza funções que permitem maior controlo sobre os dados;</li><li>- É ideal para manipular recursivamente as árvores XML.</li></ul>	<ul style="list-style-type: none"><li>- Gasta menos memória;</li><li>- É mais simples o que implica menos recursos;</li><li>- Não permite a validação de referências cruzadas;</li><li>- É ideal para lidar com aplicações, que não manipulam toda a árvore de objectos e para ler um documento de forma sequencial;</li></ul>

**Tabela 5-1**– Tabela resumo DOM vs SAX

Considerando, a *Tabela 5-1*, o DOM e o SAX, poderão ser utilizados no protótipo GerExa tendo em conta as vantagens e desvantagens inerentes a cada um.

**5.1.1.5 ELEMENTTREE**

Para a linguagem Python, existem outras tecnologias para manipular documentos estruturados XML especialmente dirigidas para as características da referida linguagem.

Em particular considerou-se o ELEMENTTREE [52, 53].

Esta tecnologia está centrada na estrutura de dados que representam os documentos XML. Como o nome ELEMENTTREE indica, esta estrutura de dados é uma hierarquia dos objectos, onde cada um desses objectos representa um elemento do XML.

Trata como dicionários de Python os atributos de XML e oferece o acesso aos nós da árvore de XML usando os objectos especializados de Python, que não são baseados em DOM. Cada nó do elemento pode ser acedido usando o protocolo do dicionário de Python e o protocolo da lista para alcançar os seus nós filhos.

## **5.2 -Descrição do Protótipo GerExa**

### **5.2.1 Funcionalidades do Protótipo**

O primeiro passo para a implementação de qualquer aplicação passa por definir qual(ais) as funcionalidade(s) que se pretende disponibilizar.

Neste caso específico, o protótipo deverá implementar a metodologia e a linguagem GerExa, definidas.

Assim as funcionalidades que se pretende que o protótipo disponibilize, são:

Para o Professor:

- Inserir, Editar, Apagar:
  - Sistema de Classificação;
  - Exercícios;
- Gerar Aleatoriamente e Gerar Assistido
  - Fichas de Exercícios
  - Testes de Avaliação;
- Corrigir e resolver Testes;

Para o Aluno:

- Resolver testes de avaliação e treino.

### 5.2.2 Modelo de Classes do Protótipo

Para que a especificação do que se pretende fique mais clara, foi elaborado em UML (Unified Modeling Language) [54], que é uma linguagem de modelação, que permite a visualização de diagramas, que ilustram uma aplicação.

É uma linguagem de modelação orientada aos objectos e independente das linguagens de programação.

Para ilustrar, de forma gráfica as classes a considerar neste desenvolvimento, foi elaborado o seguinte modelo de classes, que é um modelo de domínio, por essa razão não são incluídos no modelo os métodos referentes a cada classe:

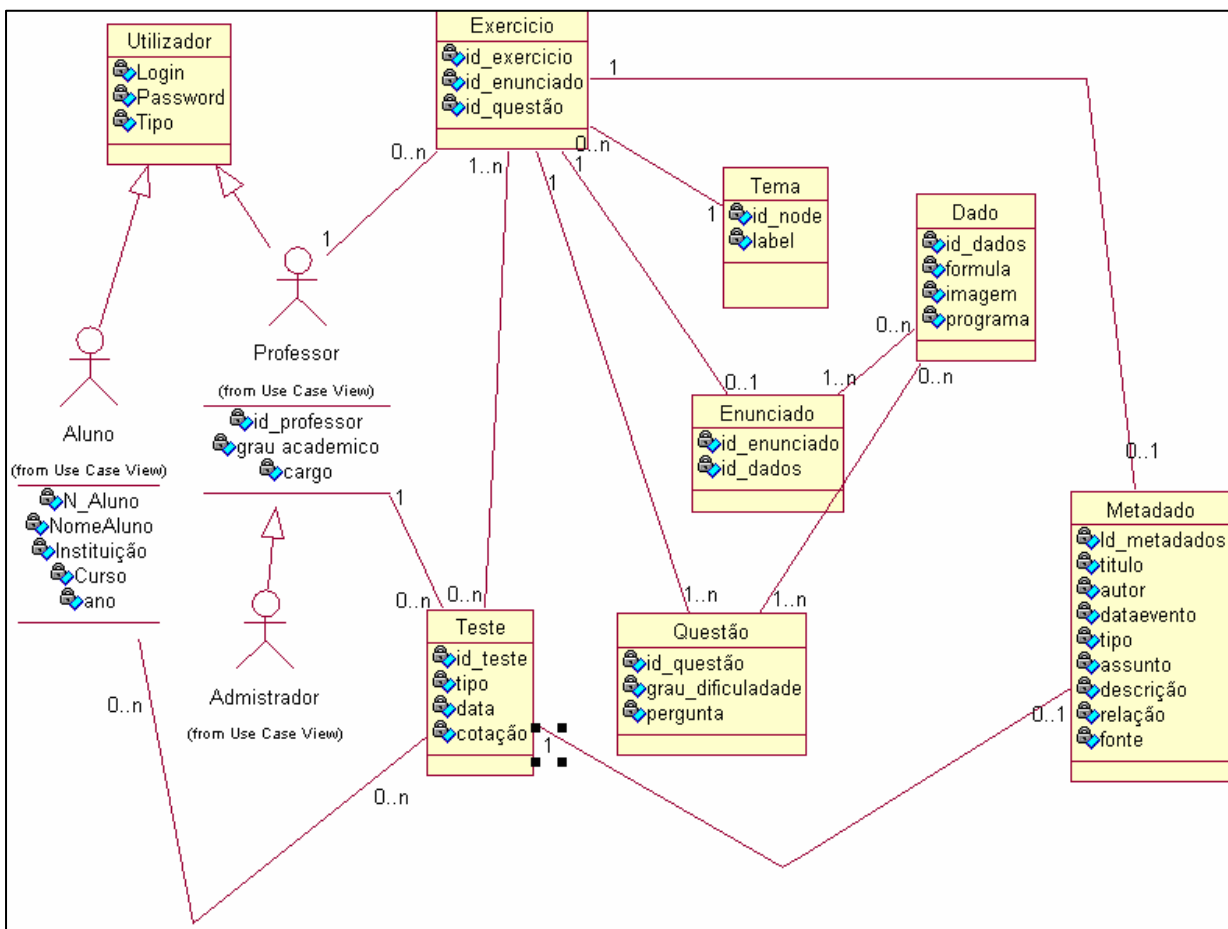


Figura 5-1 – Diagrama de Classes do protótipo GerExa

Tendo como base as funcionalidades e as classes definidas, de seguida apresentam-se as funcionalidades implementadas neste protótipo, começando pelas funcionalidades relacionadas com a classificação e divisão dos exercícios.

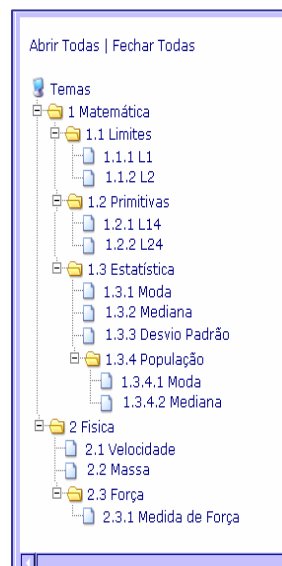
### 5.2.2.1 Os Temas

A classe temas representa o sistema de classificação dos exercícios.

O primeiro tema, ou tema raiz será designado “Temas”, para qualquer situação a considerar.

Todos os outros níveis serão sub-temas do “Temas”; para o caso ilustrado na *Figura 5.2*, foram considerados dois temas de nível 1: Matemática e Física, podendo a qualquer momento, ser inserido, mais um tema e para cada um deles os sub-temas.

Assim, foi estipulado que existiria o nível 0 – Temas, e conseqüentemente, já criado um ficheiro XML, onde vão ser inseridos todos os níveis.



**Figura 5-2 – Representação de uma árvore de “Temas”**

De acordo com o ACM [37], quando se estabelece a classificação de documentos, esta deve permitir a gestão de diferentes níveis. Baseado nesse pressuposto, a árvore resultante pode ter tantos temas, quantos o utilizador considerar necessários.

O protótipo GerExa, permite inserir o número de temas e sub-temas que o utilizador pretender.

### Como funciona a implementação desta estrutura no protótipo, para a manipulação de “temas”?

O layout apresenta uma árvore inicialmente vazia, com um nível de topo designado “Temas”, a partir do qual se pode efectuar a inserção de novos temas, como se ilustra na *Figura 5-3*:

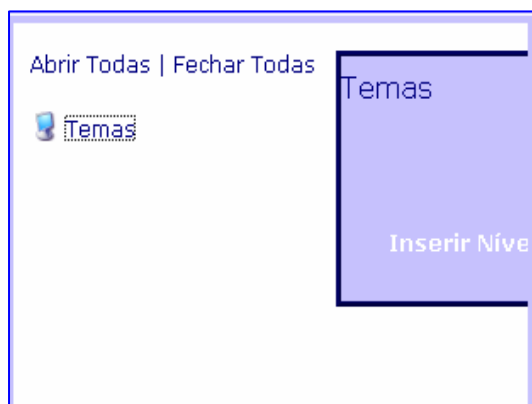


Figura 5-3 – Estrutura inicial para manipulação de “Temas”

Esta estrutura, vai ser semelhante ao Explorer de um sistema de ficheiros, como o Explorer do Windows, com a qual a maioria dos utilizadores está familiarizada.



A estrutura em árvore foi construída, recorrendo a um *javascript* - “*dtree.js*” [55] já existente, ao qual foi acrescentado o código necessário para alcançar a solução pretendida.

## **CAPÍTULO 5**

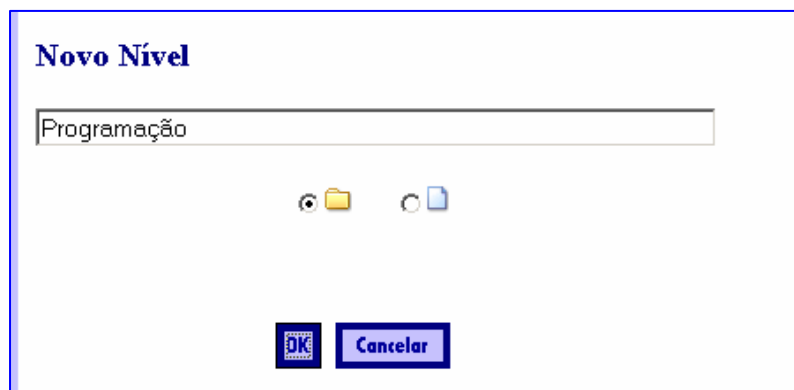
Toda a manipulação de ficheiros XML é efectuada com Python, neste caso foi utilizado SAX. É produzida uma lista com todos os “temas” e respectivos “sub-temas” que com recurso a um *javascript- arvore.js* auxiliar criado para manipular o resultado dessa lista e será produzida a árvore, usando o *javascript - dtree.js*.

Todas as alterações ao código original estão assinaladas no mesmo, havendo uma menção específica em comentário no local do código onde novos blocos de código foram adicionados.

A manipulação dos “temas” a nível de inserção, edição e eliminação com o PYTHON, foi efectuada recorrendo ao ELEMENT TREE.

Na parte do protótipo que o utilizador dispõe, o passo seguinte é inserir novos “temas” na árvore, que podem ser “temas folha” (“temas” que não possuem “sub-temas”), representados por  ou “temas pasta” (“temas” que vão conter “sub-temas”), representados por .

Nas figuras seguintes, *Figura 5-4* e *5-5*, é ilustrada a inserção de novos “temas”, como “temas pasta”:



**Figura 5-4 – Inserção de novo “tema pasta”**

Cujo resultado, na estrutura em árvore é:

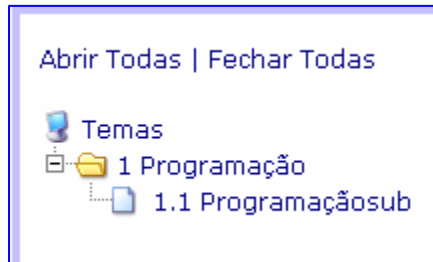



Figura 5-5 – Resultado da inserção de um “tema pasta”

Na inserção de um “tema pasta” é criado em simultâneo um “tema folha” com o mesmo nome acrescido do sufixo “sub”, ou seja, a concatenação das *strings* “nome do tema” + “sub”, uma vez que foi estipulado que não existem níveis pasta, que não tenham pelo menos um “tema folha”. Este nome gerado pelo protótipo pode posteriormente ser editado e alterado.

Um “tema folha” é inserido seleccionando a imagem  , ilustrada na *Figura 5-4* anterior e o resultado na estrutura em árvore é ilustrado na *Figura 5-6*:

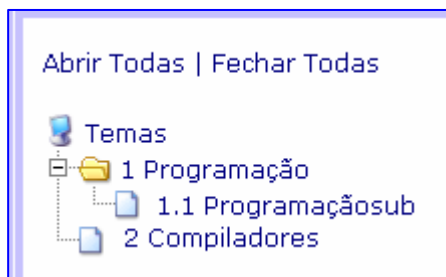


Figura 5-6 – Resultado da inserção de um “tema folha”

## **CAPÍTULO 5**

Com os “temas” criados é possível efectuar sobre eles um conjunto de operações. Existem, no entanto, diferenças na manipulação dos “temas folha” e dos “temas pasta”, dada a sua respectiva definição. As operações permitidas, para cada um são:

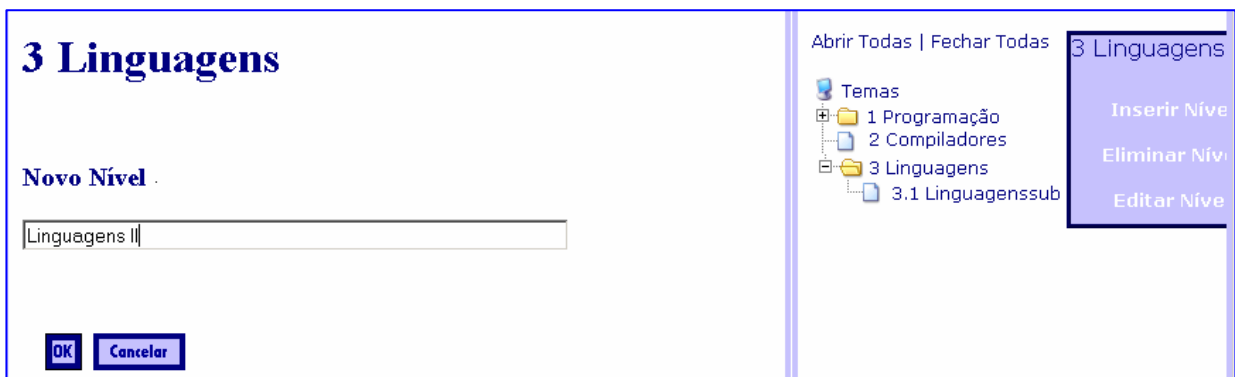
- “Temas folha”:

- Editar;
- Eliminar;

- “Temas pasta”:

- Editar;
- Eliminar;
- Inserir temas (operação idêntica à ilustrada na *Figura 5.5*).

Na *Figura 5-7* seguinte, ilustra-se a edição de um “tema pasta”:



**Figura 5-7 – Procedimento de edição de um “tema pasta”**

O resultado da alteração do nome do nível de “Linguagens” para “Linguagens II”, pode observar-se na *Figura 5-8*, seguinte:

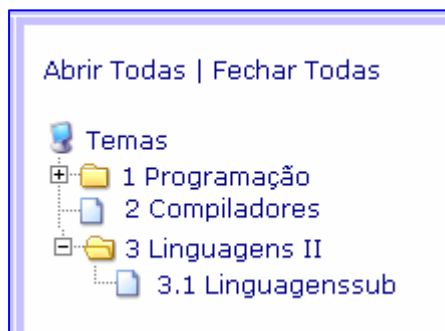


Figura 5-8 – Resultado da edição de um “tema pasta”

A edição de um nome de um “tema folha” é idêntica e o resultado da alteração do sub-tema “Linguagenssub”, criado aquando da inserção do tema “Linguagens” pode ser visualizada na *Figura 5-9*:

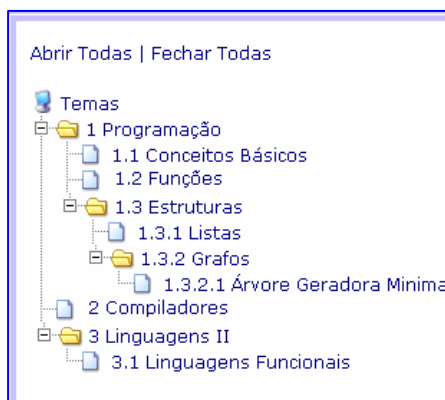
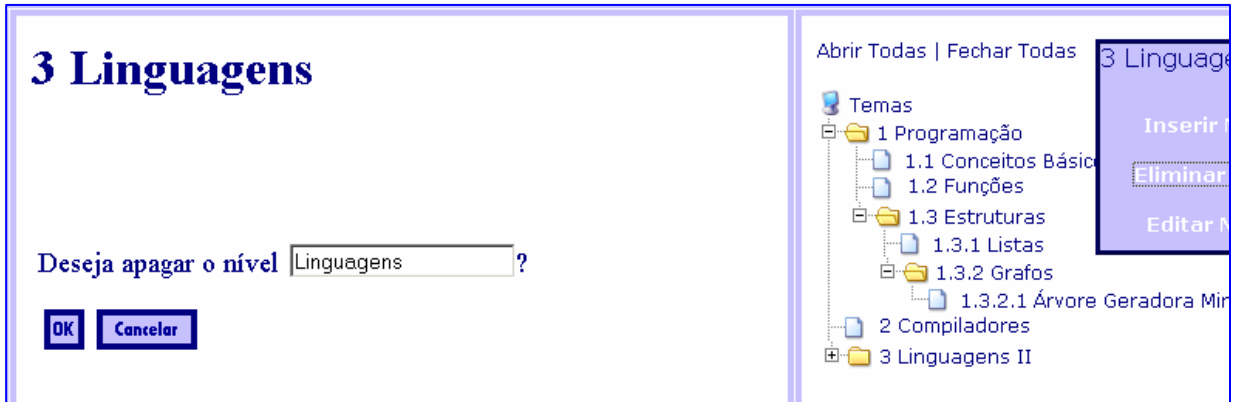


Figura 5-9 – Estrutura com vários níveis e sub-níveis

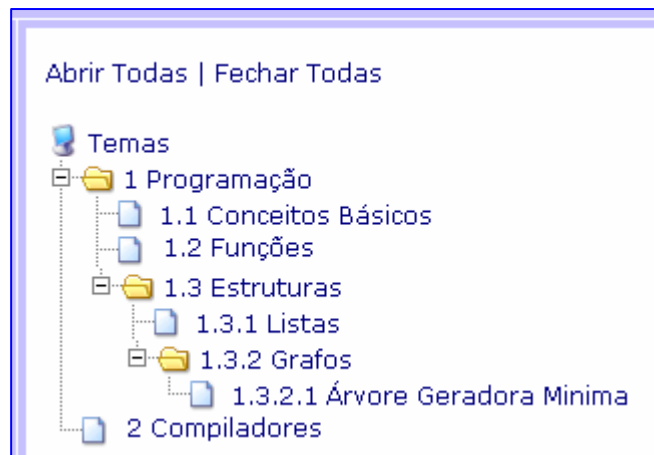
Com os níveis “folha” e “pasta” é possível ainda efectuar a operação eliminar. O procedimento é o mesmo em ambos os casos, a diferença é o resultado da operação: no primeiro caso elimina apenas o tema em questão, no segundo caso, elimina o tema e todos os sub-temas que ele tiver.

Na *Figura 5-10*, pode observar-se a operação eliminar sobre o “tema pasta”, “Linguagens II”. Uma vez que se trata de um tema, com um sub-tema, “Linguagens Funcionais”, o segundo também será eliminado.



**Figura 5-10 – Procedimento para eliminar um tema**

O resultado da operação encontra-se ilustrado na *Figura 5-11*, seguinte:



**Figura 5-11 – Estrutura em árvore resultante da eliminação de um tema (pasta)**

A estrutura em árvore gerada, foi pensada, para ser suficientemente flexível de modo a efectuar a manipulação de diversas matérias e permitir uma organização personalizada das disciplinas e respectivos conteúdos leccionados por cada professor.

### 5.2.2.2 Os Exercícios

Os exercícios são um dos elementos fundamentais do protótipo. Assim, é necessário contemplar as funções básicas de manipulação:

- Inserção;
- Edição;
- Eliminação;
- Consulta.

Para efectuar essa manipulação foi considerada uma árvore semelhante à definida para os temas, mas contemplando as operações que dizem respeito aos exercícios, como se ilustra na *Figura 5-12*, seguinte:



Figura 5-12 – Módulo de Manipulação de Exercícios

## **CAPÍTULO 5**

Na definição da linguagem GerExa foi definido que os exercícios seriam inseridos apenas em “nos folha”, cada exercício será inserido na numa directoria com o nome Exercícios e terá como primeira parte do nome a indicação do tema a que pertence.

Outra questão definida na linguagem GerExa, foi que para cada exercício poderia existir um enunciado comum a um conjunto de questões do mesmo tipo, ou do mesmo tipo mas com número distinto de escolhas ou ainda de diferentes tipos, como se ilustra na *Figura 5-13*, para a escolha múltipla com 3 opções e para a escolha múltipla com 4 opções. É ainda possível definir o número de questões dos tipos referidos.

**Enunciando**

Quantas questões pretende introduzir para o enunciado especificado?

Escolha múltipla 3 opções

Escolha múltipla 4 opções

0  
1  
2  
3  
4  
5

Seguinte Cancelar

**Figura 5-13 – Definição dos parâmetros de um exercício**

Note-se que não é obrigatória a inserção do enunciado, uma vez que para cada questão foi definido um campo “pergunta”.

O passo seguinte será a inserção dos dados da questão, bem como a respectiva correcção. No caso de questões de escolha múltipla, é assinalada a resposta verdadeira a quando da inserção dos dados do exercício, como se ilustra na *Figura 5-14*, seguinte.

The image shows a dialog box titled "Pergunta" (Question). It contains a text area with the question: "No lançamento de um dado, qual a probabilidade de sair seis?". Below the text area are three radio button options, each with a text input field:
 

- 1) 0
- 2) 1/6 (This option is selected, indicated by a filled radio button)
- 3) 1

 At the bottom of the dialog box are two buttons: "OK" and "Cancelar".

Figura 5-14 – Inserção de uma questão de escolha múltipla com 3 opções

De forma a consultar todos os exercícios inseridos, existe uma lista onde são apresentados os exercícios de cada tema, como se ilustra na *Figura 5-15*:

The image shows a list of exercises. At the top, there is a dropdown menu currently displaying "matematica1.xml" and a "Listar" button. Below the dropdown, a list of two items is shown: "matematica1.xml" and "matematica2.xml". The "matematica1.xml" item is highlighted with a blue selection bar.

Figura 5-15 – Listagem de Exercícios

O resultado da listagem de um exercício seleccionado na lista da figura anterior será:

```
Enunciado
"Considere uma moeda."

Pergunta
"Qual a probabilidade de sair cara?"

Opções
"1/3"
"1/4"
"1"
"1/2"

Pergunta
"Qual a probabilidade de sair coroa?"

Opções
"1/3"
"1/4"
"1"
"1/2"
```

**Figura 5-16 – Listagem do exercício “matemática1.xml”**

As operações de edição e eliminação são semelhantes à operação de consulta. Esta operação é substituída respectivamente por editar e eliminar. Ou seja, após a listagem de um determinado exercício ele pode ser editado ou eliminado.

### 5.2.2.3 Os Alunos/ Os Professores

De acordo com as funcionalidades definidas para o protótipo na secção 5.2.1, terá que existir uma área para o aluno e outra área para o professor.

O acesso a cada uma das áreas, designadas “GerExa Prof” e “GerExa Stud”, será efectuada por meio de *login* e *password*, como se ilustra na *Figura 5-17*, seguinte:

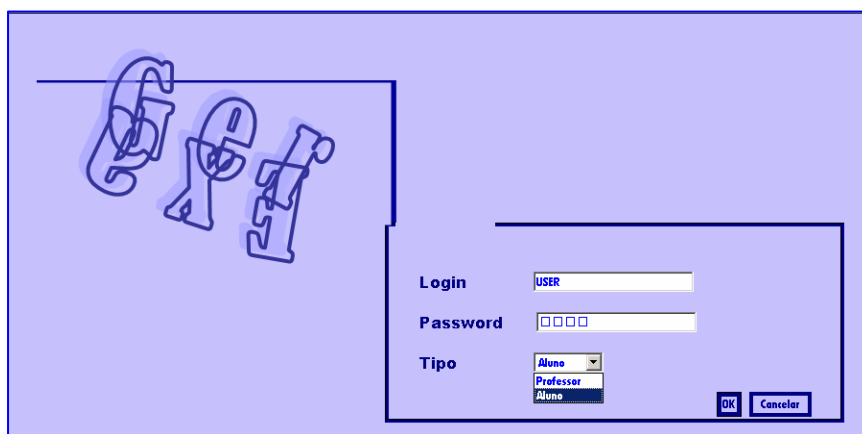
A screenshot of the initial login page for the GerExa prototype. The page has a light blue background. On the left, there is a stylized logo for 'GEREXA' in a 3D, metallic font. On the right, there is a login form with the following fields: 'Login' with a text input containing 'USER'; 'Password' with a masked input (four squares); and 'Tipo' with a dropdown menu showing 'Aluno' selected, and 'Professor' and 'Aluno' as other options. At the bottom right of the form are 'OK' and 'Cancelar' buttons.

Figura 5-17 – Página inicial do protótipo “GerExa”

Foi considerado, de acordo com a metodologia e linguagem definidas que o professor terá as funções de administrador do sistema. Assim vai dispor de um módulo de administração ilustrado na *Figura 5-18*, seguinte:

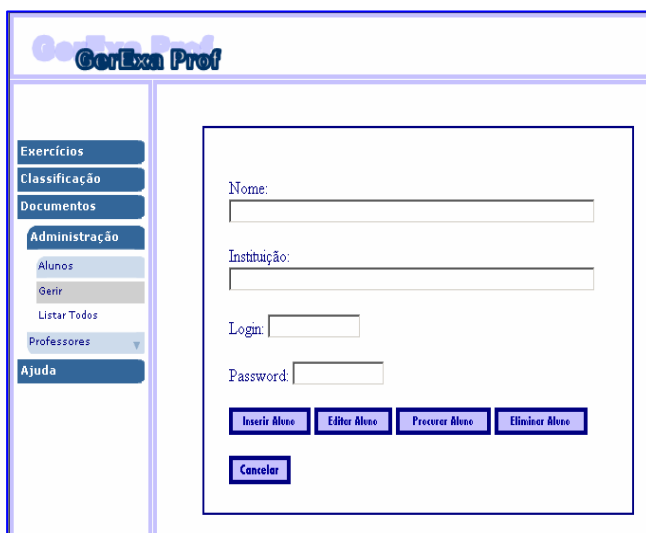
A screenshot of the 'GerExa Prof' administration module for students. The page has a light blue header with the 'GerExa Prof' logo. On the left, there is a vertical navigation menu with buttons for 'Exercícios', 'Classificação', 'Documentos', 'Administração', 'Alunos', 'Gerir', 'Listar Todos', 'Professores', and 'Ajuda'. The main content area contains a form for adding a student with fields for 'Nome:', 'Instituição:', 'Login:', and 'Password:'. Below the form are four buttons: 'Inserir Aluno', 'Editar Aluno', 'Procurar Aluno', and 'Eliminar Aluno', along with a 'Cancelar' button.

Figura 5-18 – Módulo de Administração - Alunos

## CAPÍTULO 5

A partir deste módulo o professor pode proceder à inserção de alunos, à edição dos respectivos dados, efectuar procuras e eliminar os alunos.

A “Procura de Alunos” pode ser efectuada por qualquer um dos campos com excepção da “password”, na *Figura5-19* seguinte, é ilustrada uma procura por instituição.

Login	Nome	Instituição
<a href="#">angela</a>	Angela Oliveira	EST
<a href="#">vasco</a>	vasco miguel	EST
<a href="#">anbbbbbb</a>	Ana	EST
<a href="#">anattt</a>	Ana maria	EST

**Figura 5-19 – Procura de Alunos pelo campo Instituição**

Uma procura sem qualquer dado inserido nos campos permite a listagem dos campos Login, Nome e Instituição de todos os alunos.

De forma a mostrar, a flexibilidade do protótipo e as potencialidades da combinação do XML e PYTHON, para os Alunos/Professores a informação foi tratada e apresentada de maneira distinta dos temas, a parte correspondente à gestão foi construída usando ELEMENT TREE e CSS.

No que diz respeito à opção do menu “Listar Todos”, foi implementada recorrendo ao SAX e permite a listagem de todos os elementos do Alunos.

Para os professores as operações seriam idênticas.

#### 5.2.2.4 Outros Aspectos

Ainda do protótipo, um outro aspecto que foi testado de forma independente foi o da inserção de fórmulas matemáticas.

Para as implementar e de modo a que fiquem inseridas em ficheiros XML e possam ser utilizadas e visualizadas recorreu-se MathML [39]. Para usar o MathML podem-se usar várias ferramentas, Stand-Alone ou em JavaScript. No nosso caso recorreu-se ao uso de um JavaScript - ASCIIMathML.js [40] em HTML. É contudo necessário ter instalado o MathPlayer[41]. É utilizada uma linguagem simbólica, disponibilizada ao utilizador, por meio de uma ajuda onde se encontram todas as especificações dessa mesma linguagem.

Em termos do ficheiro XML, onde o exercício é guardado, ficará guardada a especificação simbólica da linguagem. Em termos de visualização, aparece a respectiva fórmula matemática.

#### Exemplo 1:

Se o utilizador pretender inserir a matriz,  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ , a respectiva linguagem simbólica será:

`\[[1,2],[3,4]]`.

**Exemplo 2:**

Se o utilizador pretender, inserir a equação  $f(x,y) = x^{y+1} + \sqrt{\frac{x+1}{y-1}}$  a respectiva linguagem simbólica seria `f(x,y) = x^(y+1)+sqrt((x+1)/(y-1))`.

## Capítulo 6 – Conclusões e Trabalho Futuro

Um protótipo como o GerExa, pretende tornar mais cómodo e eficiente o trabalho de produção de material didáctico e de avaliação, podendo ser integrado em diferentes plataformas e integrar diferentes aplicações, por exemplo, compiladores.

O recurso à modelação de uma linguagem como o XML, trás inovação relativamente às bases de dados estruturadas e tradicionais, ao mesmo tempo que permite garantir que a informação continua a ter uma forma estruturada evidenciando a semelhança maior com os documentos reais.

No entanto, ainda há muito a acrescentar a este trabalho, nomeadamente no que diz respeito à produção e resolução de testes.

Uma das grandes vantagens do uso de XML é a estrutura dos documentos, assim uma funcionalidade interessante a implementar seria a inserção de documentos: exercícios e testes, que tivessem a mesma estrutura, definida pela linguagem GerExa, por “*Upload*”.

Há melhorias a implementar na interface, nomeadamente permitir a personalização do mesmo por parte de cada utilizador.

Dependo das amplitude da aplicação construir um módulo de administração para atribuir as permissões de acesso aos professores consoante as disciplinas leccionas.

## **CAPÍTULO 6**

Aos alunos conceder o acesso às disciplinas a que se encontrem inscritos em cada ano lectivo.

Construir a árvore de manipulação de temas sem recorrer ao *JavaScript* que foi reutilizado para o efeito, permitindo uma maior personalização da mesma.

Acrescentar novos tipos de questões que permitam a correção automática, como por exemplo as que envolvam o uso de programas de computador.

Tornar a inserção de fórmulas matemáticas mais “amigável” para o utilizador, bem como a inserção de imagens ou outros objectos de outras aplicações, por exemplo um gráfico proveniente do Excel.

Elaborar estatísticas para o aluno e professor.

- Para o aluno, estatísticas de evolução na resolução dos testes de treino e fichas de avaliação.
- Para o professor, estatísticas do número de alunos que responde aos testes treino e fichas de avaliação, bem como, as estatísticas da avaliação dos alunos.

## Referências Bibliográficas

- [1] Carla Wandscheer Krieger Langsch, Avaliação no Ensino à Distância via WEB, 1999.
- [2] Rubén Peredo Valderrama & Leonid B. Sheremetov & Cutberto García Tamayo & Jonatan Larios Tapía - Generador de Exámenes Utilizando Componentes de Software Programables Reutilizables Inteligentes para Materiales Multimedia en Espacios Virtuales de Aprendizaje (EVA), Simposio Latinoamericano y del Caribe, La Educación, La Ciencia y la Cultura en la Sociedad de la Información..19 de Fevereiro de 2002.
- [3] IMS Global Learning Consortium , Disponível por WWW em <http://www.imsproject.org/> Setembro de 2006.
- [4] Alexander Romiszowski, “O Futuro do E-Learning como inovação educacional: Factores que influenciam o sucesso ou o fracasso de projectos.”, Em The official science magazine of the Brazilian Association of Education at a Distance, publicado em 07/11/2003.
- [5] Jamie Dinkelacker “Placing Our Stuff So We Can Find It Later: A Meta-Learning Essential”, artigo publicado, IEEE Computer Society, Learning Technology, Disponível por WWW em [http://ltf.ieee.org/learn\\_tech/issues/october2003](http://ltf.ieee.org/learn_tech/issues/october2003), Outubro de 2003.
- [6] Cerro, A. L., Bervian, P.A., “Metodologia Científica, 4ª ed., Makron Books, 1996.
- [7] Citação de Mario Bunge, Disponível por WWW em [http://es.wikipedia.org/wiki/Mario\\_Bunge](http://es.wikipedia.org/wiki/Mario_Bunge), Maio de 2005.

- [8] Citação de James Trefil, Disponível por WWW em [http://es.wikipedia.org/wiki/James\\_Trefil](http://es.wikipedia.org/wiki/James_Trefil), Maio de 2005.
- [9] Salomon, Graviel, Distributed Cognitions: Psychological and Educational Considerations (Learning in Doing - Social, Cognitive, and Computational Perspectives), Cambridge University Press, ISBN: 0521574234, 1996.
- [10] Citação de Michael D. Myers, Disponível por WWW em <http://www.qual.auckland.ac.nz/MDMyers/research.htm>, Maio de 2005.
- [11] Sampieri, R. H., Collado, C. F., Lucio, P.B, “Metodologia de la Investigación”, McGraw Hill, 1996
- [12] Fetherston, T. “Pedagogical Challenges for the World Web” em “World Conference on Educational Multimedia Hypermedia & Telecommunications, 1999.
- [13] Webduc, [Ferramenta de Avaliação Formativa, Disponível por WWW em http://www.cinted.ufrgs.br/renote/mar2004/artigos/15-webduc.pdf](http://www.cinted.ufrgs.br/renote/mar2004/artigos/15-webduc.pdf), <http://www.teleduc.cl/>, Junho de 2004.
- [14] GedEx Web, Aplicação para Elaboração de Testes, Disponível por WWW em <http://www.itq.edu.mx/vidatec/espacio/int/gedex.htm>, Junho de 2004.
- [15] TestGIP, Ferramenta de Apoio ao Professor, Disponível por WWW em <http://www.softonic.com/ie/26561/TestGIP>, Junho de 2004.
- [16] SAEL, Sistema de Avaliação Electrónico- Ferramenta Didáctico-Pedagógica, Disponível por WWW em <http://www.c5.cl/ieinvestiga/actas/tise97/trabajos/trabajo13/>, Junho de 2004.

- [17] EasyTest, Ferramenta para Geração de Testes, Disponível por WWW em <http://www.inf-cr.uclm.es/mpolo/easytest.htm> e [http://ttt.upv.es/~arodrigu/TestGIP/TestGIP\\_Profesor.htm](http://ttt.upv.es/~arodrigu/TestGIP/TestGIP_Profesor.htm), Junho de 2004.
- [18] SEATT, Ferramenta de Armazenamento de Perguntas, Disponível por WWW em <http://www.edutec.es/edutec01/comunic/EMP05.html>, Junho de 2004.
- [19] “Sistema para el diseño, generación, aplicación y evaluación de exámenes basado en plataforma Internet”, artigo apresentado no Simposio Internacional, México (2002), Disponível por WWW em <http://www.somece.org.mx/memorias/2002/Grupo3/Garrido.doc>, Junho de 2004.
- [20] NetSupportSchool®, Ferramenta de Formação, Disponível por WWW em [http://www.netsupportinc.com/nss/netsupport\\_school\\_overview.htm](http://www.netsupportinc.com/nss/netsupport_school_overview.htm), e [http://www.netsupport-inc.com/nss/netsupport\\_school\\_testing\\_modules.htm](http://www.netsupport-inc.com/nss/netsupport_school_testing_modules.htm).
- [21] Escolha Múltipla On-line, Gerador Aleatório de Perguntas, Disponível por WWW em [http://ptmat.lmc.fc.ul.pt/~pduarte/Teaching/gap\\_online.html](http://ptmat.lmc.fc.ul.pt/~pduarte/Teaching/gap_online.html) e <http://ptmat.lmc.fc.ul.pt/~pduarte/Calculo/AnaliseII/Main.html>, Junho de 2004.
- [22] LaTeX, conjunto de macros com base num sistema de tipografia, Disponível por WWW em <http://www.latex2html.org/>, Junho de 2004.
- [23] Advanced eLearning Builder®, Aplicação para desenvolvimento de material para plataformas e-learning, Disponível por WWW em <http://www.eduiq.com/elearning.htm>, Junho de 2004.
- [24] FORMARE®, Plataforma de e-Learning da PT Inovação, Disponível por WWW em [http://www.ptinovacao.pt/produtos/plataformas/foI\\_FORMARE.pdf](http://www.ptinovacao.pt/produtos/plataformas/foI_FORMARE.pdf), Junho de 2004.

- [25] MOODLE, Software para Gestão de Aprendizagem, Disponível por WWW em [www.moodle.com](http://www.moodle.com), Setembro de 2006.
- [26] Projecto *Ganesh*, Disponível por WWW em <http://www.ncc.up.pt/~zp/ganesh/>, Setembro de 2004.
- [27] “Trabalho a Fazer – Projecto *Ganesh*”, Disponível por WWW em <http://www.ncc.up.pt/~zp/ganesh/afazer.html>, Setembro de 2004.
- [28] Eric Herwinjnen, “Practical SGML”, Kluwer Academic Publishers, 1994
- [29] Anderson, Richard et al., “Professional XML”, Wrox Press, 2000
- [30] J.Ramalho, P. Henriques, “XML & XSL”, FCA, 2002
- [31] Pedro Coelho, “XML – A nova Linguagem da WEB”, FCA, 2000
- [32] A. Dangui, “Perspectivas sobre o Desenvolvimento de Aplicações Baseadas em Java e XML”, 2003
- [33] SCORM, Conjunto de Padrões e Especificações para conteúdos, tecnologias e serviços para e-Learning, Disponível por WWW em <http://www.adlnet.org/technologies/SCORM/index.cfm>, Dezembro 2004.
- [34] XMLSchema, Disponível por WWW em <http://www.w3.org/XML/Schema>, Janeiro de 2005.
- [35] RELAX NG, Disponível por WWW em <http://www.relaxng.org>, Janeiro de 2005.

- [36] DTD, Disponível por WWW em <http://www.w3.org/TR/REC-html40/sgml/dtd.htm>, Janeiro de 2005.
- [37] ACM Computing Classification System, Disponível por WWW em <http://www.acm.org>, Fevereiro de 2005.
- [38] DUBLIN CORE, Disponível por WWW em <http://www.dublincore.org/documents/dces>, Fevereiro de 2005.
- [39] W3C, Disponível por WWW em <http://www.w3.org/Math/implementations.html>, Março 2005
- [40] Código javascript: MathDemo, Disponível por WWW em <http://www1.chapman.edu/~jipsen/mathml/asciimathdemo.xml>, Outubro de 2005.
- [41] MathPlayer, Disponível por WWW em, <http://www.dessci.com/en/products/mathplayer/>, Outubro de 2005.
- [42] P. Morais, J. N. Pires, “Python“, FCA, 2002.
- [43] A. Downey, J. Elkner, C. Meyers, “How to Think Like a Computer Scientist – Learning with Python“, Green Tea Press, 2002.
- [44] C.A.Jones; F. L, Drake, Jr., “Python & XML”, O’ Reilly, 2002.
- [45] OpenOffice, Disponível por WWW em <http://www.openoffice.org/>, Março de 2005.
- [46] Python UNO Bridge, <http://udk.openoffice.org/python/python-bridge.html>, Março de 2005.

- [47] PostgreSQL, <http://www.postgresql.org/>, Abril de 2005.
- [48] Unix, <http://www.unix.org/>, Abril de 2005
- [49] DOM vs SAX, Disponível por WWW em, <http://developerlife.com/saxvsdom/default.htm>, Abril de 2005.
- [50] DOM & SAX, Linux Magazine, “XML processing with Python”, 2001
- [51] DOM & SAX, Disponível por WWW em <http://www.devx.com/xml/Article/16922/1763/page/2>, Abril de 2005.
- [52] ELEMENT TREE, Disponível por WWW em <http://effbot.org/zone/elementtree.htm>, Abril de 2005.
- [53] ELEMENT TREE, Disponível por WWW em <http://www.xml.com/pub/a/2003/02/12/py-xml.html>, Abril de 2005.
- [54] - UNIFIED MODELING LANGUAGE (UML), Disponível por WWW em <http://www.uml.org>, Julho de 2005.
- [55] Código javascript, Disponível por WWW em [www.destroydrop.com/javascript/tree/](http://www.destroydrop.com/javascript/tree/), Julho de 2005.