

# GerExa: Plataforma Integrada para a Organização, Geração e Avaliação de Exercícios e Testes

Ângela Oliveira, Escola Superior de Tecnologias, Castelo Branco  
email:angelaoliveira@est.ipcb.pt

Nelma Moreira, DCC-FC& LIACC, Universidade do Porto  
email:nam@ncc.up.pt

**Resumo** Com o advento da sociedade digital aparecem novos paradigmas na área da educação. É fundamental conceber soluções de e-Learning que flexibilizem o acesso aos recursos de aprendizagem. Assim, neste trabalho, apresenta-se o desenho duma plataforma que assenta nos princípios básicos de desenvolvimento das aplicações de e-Learning, mas com agregação de funcionalidades que permitirão aos professores manter a sua informação organizada e disponível. Para tal definiu-se uma linguagem XML para a manipulação de exercícios e de testes, que serve de base para o desenvolvimento duma aplicação que permita aos professores a edição de exercícios de diferentes disciplinas, para posterior elaboração, aleatória ou assistida, de documentos de apoio às aulas e de avaliação, e sua resolução e correcção. Também os alunos poderão, gerar testes de treino e de avaliação. Pretende-se ainda que a maior parte dos exercícios sejam de correcção automática, para permitir uma avaliação interactiva e mais frequente.

**Palavras-Chave:** e-Learning, XML, geração de testes

## 1 Introdução

O XML (eXtensible Markup Language) é uma linguagem para representação de informação estruturada e uma das suas grandes vantagens é o facto de ser independente de ferramentas e plataformas [10,3]. Hoje em dia, e devido ao grande volume de informação e às inúmeras aplicações que manipulam documentos de texto, surge a necessidade de utilizar uma linguagem que de forma simples permita a sua manipulação e armazenamento uniformizados, tendo como base os seus conteúdos. Precisamente, o XML é um padrão aberto que fornece um conjunto de regras para descrever o conteúdo de documentos, bem como a sua estrutura lógica, de modo a que possam ser interpretados e/ou manipulados, quer por diferentes utilizadores, quer por diferentes aplicações [14].

Neste artigo descreve-se a modulação da informação em XML para uma plataforma integrada para gerar e organizar documentos de apoio ao ensino,

tais como fichas de trabalho e testes, que posteriormente, serão resolvidos e corrigidos. Na secção seguinte apresentamos a motivação do desenvolvimento de uma aplicação deste tipo. Na Secção 3 fazemos uma análise da informação que se pretende tratar. A Secção 4 é constituída pela descrição da linguagem XML que denominamos GerExa e que define a estrutura da informação. Na Secção 5 é apresentado o desenho da plataforma GerExa, actualmente em desenvolvimento. Alguns comentários finais são discutidos na Secção 6.

## 2 Motivação

Os professores para o exercício da sua função de ensino, manipulam uma vasta informação, fazendo uso de documentos estruturados, nomeadamente documentos para o apoio às aulas (por exemplo, fichas de exercícios) e documentos de avaliação. Para que os alunos possam ter acesso a uma diversidade considerável de exercícios, o professor necessita de reunir, muitas vezes durante anos, um vasto conjunto de material, que na maioria das vezes se encontra organizado em documentos distintos e dispersos.

Tendo em conta a evolução do ensino e a motivação dos alunos para o recurso a novas tecnologias, nomeadamente a tecnologia WEB, é fundamental o professor evoluir também nesse sentido, disponibilizando informação nesse tipo de ferramentas. Assim surge a proposta de organização da informação que diz respeito ao processo de elaboração de fichas de exercícios e testes de avaliação formativa e sumativa. Estes três tipos de documentos vão conter em comum exercícios, que se encontram organizados por conteúdo, num primeiro nível, estando os conteúdos organizados segundo módulos e estes segundo disciplinas. Os aspectos mais relevantes desta informação prendem-se com os exercícios e testes, uma vez que a ideia subjacente é o desenvolvimento de uma aplicação, que permita aos professores guardarem todos os exercícios de uma determinada disciplina, de forma organizada, para posteriormente poderem produzir os diversos elementos de apoio às aulas e de avaliação, através de geração aleatória ou assistida, desses mesmos documentos. De forma semelhante os alunos poderão, gerar testes de treino, bem como responder a testes de avaliação produzidos pelo professor. Para que seja possível uma avaliação interactiva e mais momentos de avaliação, pretende-se que a maior parte dos exercícios permitam correção automática. Por isso, e pelo menos numa primeira fase, serão construídos exercícios de resposta simples, p.e, de escolha múltipla ou resposta calculada.

Com o advento da sociedade digital aparecem novos paradigmas na área da educação e da formação, tal como a formação ao longo da vida, como factor fundamental na estabilidade do indivíduo no mercado de trabalho. É fundamental conceber soluções de e-Learning que flexibilizem o acesso aos recursos de aprendizagem implantando estratégias pedagógicas adequadas a uma aprendizagem mais eficaz [13]. A plataforma GerExa aqui apresentada assenta nos princípios básicos de desenvolvimento das aplicações de e-Learning, mas com uma visão alternativa mais vocacionada para as necessidades do ensino tradicional, ou seja, com a agregação de funcionalidades que permitirão aos docentes

manter a sua informação organizada e sempre disponível. Neste caso, organização da informação, prende-se com a implementação de uma estrutura que permita guardar os exercícios e os testes de forma simples e de fácil acesso, aspectos que se encontram no XML como alternativa às tradicionais bases de dados que são usadas na maior parte das plataformas de e-Learning, por exemplo, NetSupport School© [5], Advanced e-Learning Builder© [2] e FORMARE© [6] e plataformas que implementam geração de testes. Também existem já plataformas de geração de testes que usam XML, como é o caso da IMS Global Learning Consortium, Inc. © [4], cujo software proprietário implementa uma plataforma de e-Learning. Uma das características do GerExa é precisamente ser desenvolvido usando especificações abertas, independente da plataforma e software livre, e ainda, ir estar disponível sob uma licença GPL (GNU Public Licence).

### 3 Análise da Informação

A informação base a manipular será a dos exercícios. Os testes, fichas de trabalho e resoluções dos alunos serão constituídas por conjuntos de exercícios com diversos tipos de conteúdos (e/ou respostas dos alunos, métodos de resolução, cotações, etc). Os exercícios para além de informações referentes às disciplinas (módulos e conteúdos, docente, etc.), diferem essencialmente no tipo. Os tipos a considerar são os seguintes:

#### 1. Escolha múltipla

Exemplo de um exercício da disciplina de Programação:

*Escolha a resposta correcta:*

Em C um programa começa com a função:

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

#### 2. Resposta múltipla

Exemplo de um exercício da disciplina de Geografia:

*Seleccione a(s) resposta(s) correctas:*

São cidades de Portugal:

- a) Porto
- b) Salamanca
- c) Madrid
- d) Paris
- e) Faro
- f) Guarda
- g) Vigo
- h) Guimarães.

#### 3. Correspondência de colunas

Exemplo de um exercício da disciplina Algoritmos:

*Estabeleça a correspondência de colunas.*

Qual a ordem de cada uma das funções:

Coluna 1	Coluna 2
1. $n \log n$	1. $O(n)$
2. $2n^2 + n$	2. $O(1)$
3. $c$ , $c$ constante	3. $O(n^2)$

Coluna1	Coluna2

#### 4. Verdadeiro ou Falso

Exemplo de um exercício de Matemática Discreta:

Se  $A$  e  $B$  conjuntos, é verdade que  $\overline{A \cap B} = \overline{A} \cup \overline{B}$ ?

#### 5. Resposta calculada

Exemplo de um exercício da disciplina Álgebra:

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

Calcule o valor próprio dominante da matriz  $M$ .

#### 6. Desenvolvimento

Exemplo de um exercício da disciplina Análise Numérica:

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 e Factorização LU.

Como se pode observar pelos exemplos, pretende-se que nos enunciados e resoluções dos exercícios se possam incluir elementos como fórmulas matemáticas, imagens ou (fragmentos de) programas de linguagens de programação.

Para cada tipo de exercício é necessário ainda definir a informação que permitirá a correção automática, isto é, como é representada a sua solução, a resolução do aluno e como será (eventualmente) calculada a cotação. Os casos mais complexos, são obviamente para os exercícios de **resposta calculada**. Aqui, podem-se englobar exercícios mais simples em que a resposta é só um valor, até exercícios que envolvam a comparação de dois programas de computador. Para estes exercícios poder-se-á usar o tipo de abordagem proposta em [11]. Para os exercícios de **desenvolvimento** não está previsto qualquer tipo de correção automática. No entanto, no caso do exemplo acima apresentado para exercícios de desenvolvimento, pode-se prever a sua correção (semi)-automática, uma vez que se poderá implementar o método referido e comparar a solução com a resolução do aluno.

## 4 A Linguagem GerExa

Com base na informação referida na secção anterior, construiu-se uma linguagem XML, denominada GerExa. A linguagem será descrita usando DTDs ("Document Type Declaration") [3], sendo indicado um fragmentos dos DTDs para cada

uma das suas componentes. A escolha deste esquema de especificação, em detrimento de, por exemplo, o XML Schema [9] ou RelaxNG [7], prende-se com sua maior simplicidade e, também, ser o mais conhecido e implementado. A possibilidade de especificar tipos de estruturas de dados e o modo como se podem compor as diversas especificações, serão vantagens dos outros dois esquemas referidos. E, embora as expressividades associadas sejam diferentes [12], pretendemos fazer uma especificação da linguagem GerExa usando o RelaxNG. A sua simplicidade (especialmente na notação compacta) e clara formalização teórica, tornam-o um forte candidato de substituição dos DTDs. Uma das vantagens que se pretende explorar (para além das acima referidas) é a maior flexibilidade entre o uso de elementos e atributos.

#### 4.1 Exercícios

O fragmento de DTD seguinte especifica o elemento `exercicio`:

```
<!ELEMENT exercicio ((enunciado)?,(questao)+)>
<!ATTLIST exercicio id_conteudo IDREF #REQUIRED
                    id_exercicio ID #REQUIRED >
<!ELEMENT enunciado (#PCDATA|formula|imagem|programa)*>
<!ELEMENT questao (pergunta,(escolha_multipla|resposta_multipla|
correspondencia|verdadeiro_falso|desenvolvimento|
resposta_calculada),valorizacao,(cotacao_questao?))>
<!ATTLIST questao id ID #REQUIRED
                    grau_dificuldade (1|2|3|4) "2">
<!ELEMENT pergunta (#PCDATA|formula|imagem|programa)*>
<!ELEMENT cotacao_questao (#PCDATA)>
<!ELEMENT valorizacao (cotacao,(classificacao?))>
<!ELEMENT cotacao (valor,valor desconto)>
<!ELEMENT classificacao (#PCDATA)>
<!ELEMENT valor (#PCDATA)>
<!ELEMENT valor desconto (#PCDATA)>
```

Cada elemento `exercicio` tem um elemento `enunciado`, que é opcional, e vários elementos `questao`. O `enunciado` pode conter texto, fórmulas matemáticas, fragmentos de programas ou imagens. As questões correspondem aos exercícios com alíneas. O `enunciado` não é obrigatório uma vez que cada questão tem um elemento `pergunta`, que pode dispensar o uso de `enunciado`. Por exemplo:

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

Para este mesmo `enunciado` poderia haver mais questões, por exemplo:

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

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

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

As fórmulas matemáticas serão guardadas usando a linguagem MathML [8], mas aqui iremos apenas indicar que correspondem a um elemento `formula`. Para as imagens o conteúdo do elemento `imagem` será uma indicação da localização da imagem. Os programas poderão ser acedidos como as imagens ou incluídos no elemento, mas em qualquer dos casos deverá ser indicada a linguagem de programação, p.e:

```
<!ELEMENT programa (#PCDATA)>
<!ATTLIST programa codigo (CDATA) #IMPLIED
  linguagem (CDATA) #REQUIRED
  uri (0|1) '0' #REQUIRED>
```

Cada exercício foi hierarquizado, na estrutura referida anteriormente, ou seja o exercício pertence a um determinado conteúdo, que está incluído num módulo de uma determinada disciplina. Considera-se como atributo o identificador do conteúdo ao qual pertencem. Esta hierarquia poderá ser demasiado rígida. A nível de especificação XML seria simples alterar: bastava considerar que o exercício teria um elemento `disciplina`, que teria opcionalmente um elemento `modulo`, e este por sua vez um elemento `conteudo`. Cada um destes elementos teria apenas um atributo com o respectivo identificador.

Cada questão pode ser dum dos tipos seguintes, que por abuso de linguagem, iremos designar por *tipo de exercício*:

### Escolha múltipla

Neste tipo de exercício existem várias opções das quais apenas uma está correcta. Estipulou-se a inserção mínima de três opções e máxima de cinco, sendo este número configurável na aplicação. Como existe uma única opção correcta, a resposta correcta tem o valor do atributo `resposta` igual a 1 e para as restantes esse valor será 0.

A respectiva estrutura é a seguinte:

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

O atributo `resolucao` será usado para guardar a resolução do aluno.

### Resposta múltipla

Neste tipo de exercícios, decidiu-se que poderiam ser inseridas no mínimo duas alíneas e no máximo oito, e tal como nos exercícios de escolha múltipla, este valor será configurável na aplicação. Cada uma das alíneas pode ser verdadeira ou falsa, a que corresponderá o valor do atributo `resposta`, 1 ou 0, respectivamente.

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

Como no caso anterior, o atributo `resolucao` será usado para guardar a resolução do aluno. Note-se que a nível de especificação XML os elementos `escolha_multipla` e `resposta_multipla` são basicamente idênticos. E isto aconteceria se tivesse sido usado outro esquema, p.e, XML Schema ou RelaxNG. No entanto a sua separação permitirá um tratamento diferente pela aplicação de verificação e correção automática.

Nestes dois casos, o número de opções (ou de alíneas) podia também ter ficado por especificar usando-se, por exemplo, o operador `+`. Esta especificação, contudo, está de acordo com o interface usado para a geração dos exercícios. Caso seja necessário, poder-se-á fazer uma especificação mais genérica, que valide os documentos obtidos pela especificação aqui adoptada e outros mais gerais.

### Correspondência

Nestes exercícios é necessário fazer a correspondência de linhas entre duas colunas com igual número de linhas. Considera-se um mínimo três linhas e um máximo cinco. Cada linha tem um atributo identificador. A coluna 2 tem uma numeração fixa. Para a coluna 1 o atributo `resposta`, indica, para cada linha, qual a linha correspondente na coluna 2. Também o atributo `resolucao` apenas é necessário na coluna 1.

```
<!ELEMENT correspondencia (col1,col1,col1,(col1)?,(col1)?,
col2,col2,col2,(col2)?,(col2)?,(resolucao)?)>
<!ELEMENT col1 (#PCDATA|formula|imagem|programa)*>
<!ATTLIST col1
    col1_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 col2_id (1|2|3|4|5) "1">
```

Mais uma vez, não é possível garantir, nesta especificação, que o número de elementos col1 e col2 é o mesmo. Para tal teria de se ter diversos elementos: correspondencia3, correspondencia4, etc.

### Verdadeiro-Falso

Para este tipo de exercício apenas é necessário indicar se a pergunta é verdadeira ou falsa.

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

### Resposta Calculada

Nesta categoria englobam-se todos os exercícios cuja resposta pode ser obtida através de um cálculo efectuado por computador. No caso mais simples, apenas poderá ser definido no elemento resultado, o valor da resposta. Mas, no caso geral, deverá ser dado no elemento dominio o método e os dados necessários.

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

### Desenvolvimento

Como já se disse, para estes exercícios não se prevê nenhum tipo de correção automática. Contudo pode ser vantajoso que o professor indique qual seria a resposta correcta, para uma eventual indicação aos alunos, no caso de testes de treino, ou para uma possível correção depois de um teste de avaliação.

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

Para cada elemento questao consideram-se ainda elementos que permitam, num teste, o cálculo de uma classificação. Estes elementos referem-se à cotação que o professor pode atribuir a cada questão e a classificação obtida pelo aluno.

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



O elemento `cotacao` é constituído pelos elementos `valor` e `valordesconto`. 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.

## 4.2 Disciplinas

Uma disciplina é constituída por módulos e cada módulo por conteúdos. A definição de uma disciplina pode ainda incluir alguma informação geral como o título ou designação, o curso ou cursos a que pertence, ano ou anos curriculares da disciplina, a escolaridade, etc. Temos os seguintes elementos:

```
<!ELEMENT disciplina (titulo,(curso)+,(anolectivo)?,
(escolaridade)?,(docente)*,(modulo)+>
<!ATTLIST disciplina id_disciplina ID #REQUIRED
                    ref_docente IDREFS #IMPLIED>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT curso (nome,ano)>
<!ATTLIST curso instituicao_id IDREF #REQUIRED>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT ano (#PCDATA)>
<!ELEMENT anolectivo (#PCDATA)>
<!ELEMENT escolaridade (#PCDATA)>
<!ELEMENT modulo (conteudo)+>
<!ATTLIST modulo id_modulo ID #REQUIRED
                    tema CDATA #REQUIRED >
<!ELEMENT conteudo (#PCDATA)>
<!ATTLIST conteudo id_conteudo ID #REQUIRED>
```

Cada disciplina tem associado um ou mais docentes e uma referência à instituição, aos quais se associam os elementos seguintes:

```
<!ELEMENT docente (nome_doc)>
<!ATTLIST docente id_docente ID #REQUIRED>
<!ELEMENT nome_doc (#PCDATA)>
<!ELEMENT instituicao (#PCDATA)>
<!ATTLIST instituicao id_instituicao ID #REQUIRED>
```

Finalmente, é necessária alguma informação para identificar os alunos:

```
<!ELEMENT aluno (nome_aluno,numero_aluno)>
<!ATTLIST aluno id_aluno ID #REQUIRED>
<!ELEMENT nome_aluno (#PCDATA)>
<!ELEMENT numero_aluno (#PCDATA)>
```

A informação das disciplinas será agregada num elemento de topo da forma:

```
<!ELEMENT disciplinas (disciplina)+>
```

E, analogamente, para a informação dos docentes, dos alunos e das instituições.

### 4.3 Testes

Um teste é um conjunto de exercícios que poderá ter ou não uma identificação da data em que se realiza. O elemento `cotacao_teste` será usado para guardar a cotação da resolução de um aluno.

```
<!ELEMENT teste (data?,(exercicio)+,cotacao_teste?)>
<!ATTLIST teste id_teste ID #REQUIRED
                id_disciplina IDREFS #REQUIRED
                id_instituicao IDREF #IMPLIED
                aluno IDREF #IMPLIED>
                tipo_teste (Treino|Ficha_Trabalho|Frequencia|
                            ExameN|ExameR|ExameEsp) "Frequencia">
<!ELEMENT data (#PCDATA)>
<!ELEMENT cotacao_teste (#PCDATA)>
```

## 5 A Aplicação GerExa

A especificação apresentada na secção anterior está a servir de base para o desenvolvimento da aplicação GerExa. A aplicação permitirá ao professor:

- Inserir, editar e apagar exercícios e disciplinas,
- Gerar testes de forma aleatoria ou assistida,
- Corrigir ou verificar testes,

e, permitirá ao aluno gerar e resolver testes de treino e resolver os testes de avaliação.

Para a inserção de um exercício será necessário seleccionar a disciplina, o respectivo módulo e o conteúdo a que o exercício pertence (que caso não existam, terão de ser introduzidos). Opcionalmente será introduzido o enunciado do exercício, e em seguida as várias questões. Para cada uma, será escolhido o tipo e mediante esse tipo, serão apresentados formulários para o preenchimento da informação correspondente.

Cada exercício é guardado num documento de tipo `exercicio`, i.e, num documento cujo elemento de topo é `exercicio`. A informação das disciplinas é guardada num documento de tipo `disciplinas`.

Os testes serão gerados pela aplicação, a partir dos exercícios anteriormente introduzidos (ver Figura 1).

Para o caso do aluno, a selecção dos exercícios é efectuada de forma completamente aleatória, dentro de um conteúdo, de um módulo de uma disciplina, ou vários conteúdos ou módulos de uma mesma disciplina.

O professor pode também optar pela geração automática ou por uma geração assistida. No primeiro caso, escolhe um conjunto de exercícios, a partir do qual pretende seleccionar um número menor. Depois de ter um conjunto de questões seleccionadas, pode alterar por exemplo, a ordem, retirar, acrescentar questões, etc.

A implementação está a ser considerada de modo a que, a geração dos testes seja o mais dinâmica possível, considerando-se futuramente a hipótese de alguns exercícios poderem ser gerados por aplicações externas à aplicação GerExa.

Cada teste é guardado num documento de tipo `teste`, sendo feita uma cópia dos vários exercícios que o compõem. Embora, possa aqui haver alguma redundância de informação, diminui a complexidade da gestão, nomeadamente, no caso de se pretender apagar exercícios. Notar que a ênfase da aplicação é na organização dos exercícios, sendo os testes considerados documentos autónomos.

**Figura 1.** Interface para a geração de testes

Depois de gerado um teste, pode ser respondido por um ou mais alunos, e/ou transformado para ficheiros HTML,  $\text{\LaTeX}$ , PDF, etc. Em particular, tanto o enunciado como as resoluções dos alunos poderão ser impressas em papel.

Os testes já resolvidos pelos alunos poderão ser consultados pelos professores para correção de perguntas de desenvolvimento (ou resposta calculada).

Cada teste resolvido por um aluno será guardado num documento XML. Durante as fases de resolução e correção esse documento conterá toda a informação do teste, podendo depois ser apenas armazenada a informação referente à resolução do aluno. Presentemente, não está definida qual a especificação destes últimos.

Para a introdução de fórmulas matemáticas, quer na inserção de exercícios quer na resolução dos alunos, está a ser usado actualmente uma aplicação javascript `ASCIIMathML` [1], a qual através de um mecanismo de importação permite converter para `MathML`. Na aplicação ao seleccionar o botão `Inserir Fórmulas Matemáticas`, aparece uma janela de texto que permite inserir as fórmulas, através de uma linguagem simbólica.

## 6 Conclusões

Uma aplicação como o GerExa, pretende tornar mais cómodo e eficiente o trabalho de produção de material didáctico e de avaliação. O recurso a uma linguagem como o XML, trás inovação e flexibilidade relativamente às bases de dados estruturadas tradicionais, mantendo a garantia de continuar a ter uma forma estruturada, mas mais próxima dos documentos usuais. A especificação apresentada pode contudo ser melhorada permitindo uma maior flexibilidade na construção dos exercícios, de que salientamos:

- mecanismos de geração de exercícios semelhantes (p.e, que difiram no valor de uma quantidade)
- os mecanismos de correção automática de exercícios de resposta calculada

Pretendemos ainda que o prótipo, agora em fase de desenvolvimento, seja testado e avaliado por um conjunto de professores e alunos.

## Agradecimentos

Agradecemos aos revisores anónimos os comentários e sugestões que permitiram melhorar este trabalho.

## Referências

1. Asciiathdemo. <http://www1.chapman.edu/~jipsen/mathml/asciiathdemo.xml>.
2. Edulq.com. <http://www.eduiq.com/elearning.htm>.
3. Especificação de xml. <http://www.w3.org/TR/REC-xml>.
4. Ims. [http://www.imsglobal.org/question/qti\\_item\\_v2p0pd/xmlbinding.html](http://www.imsglobal.org/question/qti_item_v2p0pd/xmlbinding.html).
5. Netsupport incorporated. [http://www.netsupport-inc.com/nss/netsupport\\_school\\_overview.htm](http://www.netsupport-inc.com/nss/netsupport_school_overview.htm).
6. Pt inovação. [http://www.ptinovacao.pt/produtos/plataformas/fof\\_FORMARE.pdf](http://www.ptinovacao.pt/produtos/plataformas/fof_FORMARE.pdf).
7. Relax ng. <http://www.relaxng.org/>.
8. W3c math. <http://www.w3.org/Math/implementations.html>.
9. Xml schema. <http://www.w3.org/XML/Schema>.
10. José Carlos Ramalho e Pedro Henriques. *XML & XSL*. FCA, 2002.
11. José Paulo Leal and Nelma Moreira. Using matching for automatic assessment in computer science learning environments. In Francisco Restivo and Lúgia Ribeiro, editors, *Web-Based Learning Environments*. Merlin 2000 Project, FEUP, June 2000.
12. Dongwon Lee, Murali Mani, and Makoto Murata. Reasoning about xml schema languages using formal language theory. Technical report, IBM Almaden Research Center, 2000.
13. Jorge Reis Lima. *e-Learning e e-Conteúdos*. FCA, 2003.
14. Artur Afonso Sousa. *Bases de Dados, Web e XML*. FCA, 2000.