

UNIVERSIDADE DE LISBOA  
Faculdade de Ciências  
Departamento de Informática



**AUTOMATIC TEACHING OF BASIC  
PROGRAMMING TECHNIQUES**

**Diogo Gonçalo Ferreira Santos**

**PROJECTO**

**MESTRADO EM ENGENHARIA INFORMÁTICA**  
Especialização em Engenharia de Software

2014



UNIVERSIDADE DE LISBOA  
Faculdade de Ciências  
Departamento de Informática



**AUTOMATIC TEACHING OF BASIC  
PROGRAMMING TECHNIQUES**

**Diogo Gonçalo Ferreira Santos**

**PROJECTO**

**MESTRADO EM ENGENHARIA INFORMÁTICA**  
Especialização em Engenharia de Software

Dissertação orientada pelo Prof. Doutor Thibault Nicolas Langlois

2014



## Resumo

Este trabalho visa, por meio de novas evoluções no campo da educação *online*, providenciar um método de ensino das linguagens de programação mais fácil a pessoas sem experiência prévia no campo, assim como ajudar professores a ensinar e avaliar grupos inteiros.

Para este fim vai ter em conta experiência de outros projectos já efectuados na medida do *e-learning*, mais concretamente *Massive Open Online Courses*, principalmente na plataforma Duolingo, pela sua abordagem interessante que não imita simplesmente o ensino presencial como é a tendência mas que faz uso das tecnologias que agora estão à sua disposição.

O método de ensino que irá ser aplicado neste trabalho difere dos outros métodos e é baseado no Duolingo: aprendizagem efectuada unicamente através de exercícios. Estes exercícios irão aumentar de dificuldade, adaptar-se conforme as necessidades educativas de cada aluno e aperfeiçoar-se ao aprender com as respostas dadas. Desta maneira envolvente espera-se que os eventuais alunos sintam menos dificuldades do que em outros métodos mais passivos de aprendizagem.

**Palavras-chave:** GWAP, Computação humana, Programação, e-learning, MOOC, Educação

## Abstract

This work aims to provide a different approach to online teaching of programming languages, making use of the new evolutions in the field of on-line education, as well as help teachers to teach and evaluate whole groups.

For this purpose the experiences taken from other e-learning projects, namely Massive Open Online Courses, have been taken into account, specially the learning platform Duolingo. This platform's approach is interesting because it does not simply copy classroom learning, as is the tendency of others, but it takes advantage of the available technologies.

The teaching method that will be applied to this project, which differs from other methods and is based on Duolingo, is that learning will happen solely through exercises, which will evolve in difficulty and adapt depending on the difficulty that students face and improving itself using the answers given. With this focus on being engaging it is expected that students will feel more motivated than in other more passive methods of learning.

**Keywords:** GWAP, Human Computation, Programming, e-learning, MOOC, Education

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Objectivos . . . . .	1
1.3	Contribuições . . . . .	1
1.4	Linhas de código . . . . .	2
1.4.1	Frontend . . . . .	2
1.4.2	Editor . . . . .	2
1.4.3	Backend . . . . .	3
1.5	Estrutura do documento . . . . .	3
<b>2</b>	<b>Trabalho relacionado</b>	<b>5</b>
2.1	Ensino assistido por computador . . . . .	5
2.2	Ensino da programação assistido por computador . . . . .	5
2.3	Massive Open Online Courses . . . . .	7
2.4	Análise dos MOOCs recentes . . . . .	7
2.5	Ensino da programação no contexto dos MOOCs . . . . .	10
2.6	O caso Duolingo . . . . .	11
<b>3</b>	<b>Análise</b>	<b>13</b>
3.1	Objectivos . . . . .	13
3.2	Contornos do sistema . . . . .	13
<b>4</b>	<b>Desenho</b>	<b>15</b>
4.1	Lições . . . . .	15
4.2	Questões . . . . .	15
4.3	Tags . . . . .	15
4.4	Sessões . . . . .	16
4.5	Tipos de perguntas feitas ao aluno . . . . .	16
4.6	Casos de Uso . . . . .	17
4.6.1	Registo de conta . . . . .	17
4.6.2	Autenticação do utilizador . . . . .	17

4.6.3	Mudança da linguagem de programação activa . . . . .	18
4.6.4	Resolução de exercícios . . . . .	18
<b>5</b>	<b>Implementação</b>	<b>21</b>
5.1	Componentes Principais do Projecto . . . . .	21
5.2	Backend . . . . .	22
5.2.1	Base de Dados . . . . .	22
5.2.2	Representação interna dos dados . . . . .	22
5.2.3	Tags . . . . .	23
5.2.4	Ordem das questões . . . . .	23
5.3	Frontend . . . . .	24
5.3.1	Apresentação . . . . .	24
5.3.2	Apresentação das Questões . . . . .	26
5.3.3	Modos de Lição . . . . .	28
5.4	Editor . . . . .	28
5.4.1	Estatísticas Guardadas . . . . .	28
5.4.2	Tradução para múltiplas línguas . . . . .	29
<b>6</b>	<b>Teste de Usabilidade</b>	<b>33</b>
6.1	Inquéritos . . . . .	33
6.2	Conclusões . . . . .	34
<b>7</b>	<b>Trabalho Futuro</b>	<b>37</b>
7.1	Traduções . . . . .	37
7.2	Pontuações . . . . .	37
7.3	Dicas . . . . .	38
7.4	Cache . . . . .	38
7.5	Ordem das questões . . . . .	38
7.6	Tipos de questões . . . . .	38
7.7	Navegadores . . . . .	39
7.8	Melhorar distribuidor de carga . . . . .	39
7.9	Sincronização . . . . .	39
7.10	Segurança . . . . .	39
7.11	Progresso numa lição . . . . .	40
7.12	Clarificar fim da lição . . . . .	40
7.13	Comentários e feedback . . . . .	40
7.14	Indicação da comunicação entre frontend-backend . . . . .	40
<b>8</b>	<b>Conclusão</b>	<b>41</b>

---

<b>A</b>	<b>Manual Instalação</b>	<b>43</b>
A.1	Pré-requisitos . . . . .	43
A.1.1	Instalação dos pré-requisitos . . . . .	43
A.2	Configuração . . . . .	44
A.2.1	Back-end . . . . .	44
A.2.2	Editor . . . . .	46
A.2.3	Front-end . . . . .	46
A.3	Instalação dos componentes do Moxo . . . . .	46
A.4	Execução . . . . .	46
<b>B</b>	<b>Diagrama da Base de Dados</b>	<b>49</b>
<b>C</b>	<b>Diagrama de classes do backend</b>	<b>51</b>
<b>D</b>	<b>Lição “Mais Listas”</b>	<b>53</b>
	<b>Bibliografia</b>	<b>58</b>



# Capítulo 1

## Introdução

### 1.1 Motivação

No século XXI a tecnologia tem vindo a ter um papel cada vez mais importante na vida das pessoas. Saber programar traz uma mais valia no sentido de entender melhor o funcionamento dos programas a serem utilizados, melhorar a capacidade lógica do indivíduo e torná-lo mais apelativo aos olhos de um potencial empregador [1].

O maior problema até agora é que a aprendizagem da programação requeria um certo nível de dedicação e as abordagens existentes nem sempre tratavam esta aprendizagem com a devida seriedade, tal como linguagens didácticas que depois de aprendidas não davam ao aluno as ferramentas necessárias para construir algo de nível de produção.

### 1.2 Objectivos

O ensino *online* tem sofrido uma recente revolução no qual a tendência é para continuar a mudar a maneira tradicional de aprendizagem [2].

O objectivo deste projecto é o de utilizar técnicas baseadas nestas novas técnicas de ensino *online*, com ênfase na plataforma de ensino Duolingo. Torna o ensino das linguagens de programação mais facilitada do que fora até agora para pessoas sem conhecimentos prévios na área. Ou seja, difundir a aprendizagem de linguagens de programação a pessoas com pouco ou nenhum conhecimento.

Este projecto procura também facilitar o trabalho dos professores, disponibilizando-lhes uma ferramenta para facilitar a distribuição massiva do material e para monitorizar o progresso dos alunos.

### 1.3 Contribuições

Para este projecto foram criados três componentes: o back-end em java, o editor de perguntas e o front-end.

As bibliotecas externas utilizadas pelo back-end:

- NanoHTTPD — Para permitir actuar como um servidor HTTP  
<http://nanohttpd.com/>
- Java Ini Parser — Usado para processar o ficheiro de opções de formato INI  
<http://www.xinotes.net/notes/note/407/>
- JSON Parser — Usado na comunicação entre o back-end e o front-end  
<http://www.json.org/java/index.html>
- JSOUP — Para fazer a filtragem de certas tags HTML  
<http://jsoup.org/>

No front-end foi utilizado código baseado em Using CORS para fazer a comunicação entre javascript e o servidor java.

Também foi utilizada a biblioteca JQuery em certas secções para produzir código mais compacto e legível tanto no front-end como no editor, e foi utilizado JQuery UI para produzir sugestões no preenchimento de certos campos.

O professor Thibault foi o responsável pelo preenchimento da base de dados com lições, pela coordenação das sessões de testes junto do Liceu Francês Charles LePierre e dos inquéritos sobre estes.

## 1.4 Linhas de código

Segue-se a contagem das linhas de código produzidas no âmbito deste projecto. Estatísticas obtidas com a ferramenta CLOC. *Ficheiros* refere-se ao número de ficheiros com essa linguagem, as restantes colunas referem-se ao número de linhas pertencente ao tipo especificado.

### 1.4.1 Frontend

Linguagem	Ficheiros	Em branco	Comentários	Código
PHP	10	148	96	1364
Javascript	1	8	12	77
CSS	2	3	10	71

### 1.4.2 Editor

Linguagem	Ficheiros	Em branco	Comentários	Código
PHP	33	532	431	3914
Javascript	1	12	48	68
CSS	3	18	23	209

### 1.4.3 Backend

Linguagem	Ficheiros	Em branco	Comentários	Código
Java	35	738	1199	4122

## 1.5 Estrutura do documento

Este documento está organizado da seguinte forma:

- Na Introdução (capítulo 1) refere-se qual é o âmbito deste projecto e estabelece o tema sobre o qual este revolverá.
- Em Trabalho relacionado (capítulo 2) fala-se sobre o trabalho já existente na área da educação online.
- Em Análise (capítulo 3) vai se definir as bases pretendidas do sistema.
- Em Desenho (capítulo 4) mencionam-se os conceitos utilizados no desenvolvimento do sistema.
- Em Implementação (capítulo 5) referem-se os detalhes e conceitos aplicados na implementação do sistema.
- Em Teste de Usabilidade (capítulo 6) discute-se o teste de usabilidade efectuado no Liceu Francês Charles Lepierre.
- Em Trabalho Futuro (capítulo 7) listam-se os pontos do sistema que requerem melhoria para aumentarem a sua robustez.
- Em Conclusão (capítulo 8) faz-se um apanhado do trabalho.



# Capítulo 2

## Trabalho relacionado

### 2.1 Ensino assistido por computador

O ensino à distância é uma instituição com séculos de existência, os primeiros registos são datados de 1728 [3]. Este tem desenvolvido um papel importante no sentido de complementar o ensino tradicional. Uma das suas mais fortes características é o seu aspecto flexível, tem vindo a adaptar-se às tecnologias ao longo do tempo, e com o advento da electrónica generalizada este veio para a categoria de *e-learning* [3].

O *e-learning* pode ser definido como qualquer aprendizagem efectuada com tecnologias de informação e comunicação. É a distribuição de ensino e suas actividades relacionadas, acesso a comunidades e eventos via média electrónica tais como a *Internet*, intranet, CD-ROM, cassetes de vídeo, DVDs, TV, telemóveis, etc [4]. Combina várias formas de aprendizagem como o auto-estudo, exercícios em grupo ou individuais, testes, simulações, seminários virtuais, entre outros. Esta definição engloba ensino ministrado à distância ou na sala de aula.

A *Internet* teve um grande impacto no ensino à distância. Este impacto foi maior com a *web 2.0* e possibilitou o aparecimento de uma nova filosofia de ensino social onde se aprende não através da abordagem clássica (passivamente com o professor como fonte primária de conhecimento), mas onde agora existe partilha de informação entre várias pessoas numa rede, há trabalho colaborativo e discussão de ideias [5, 6]. Este ensino colaborativo assistido por computador tem como ideologia que uma das melhores maneiras de aprender algo é ensinando-o a outros.

### 2.2 Ensino da programação assistido por computador

Existe um grande número de linguagens dirigidas ao ensino da programação. Estas linguagens são bastante diversas, e variam entre o tipo de linguagem em que se baseiam e o conhecimento e idade do seu público-alvo. [7]

De entre estas linguagens podem destacar-se três exemplos históricos: Pascal, BASIC

e Logo. A linguagem Pascal foi concebida tendo em mente a educação e por volta da década de 80 foi bastante utilizada nas aulas de informática para ensinar boas práticas, programação estruturada e estruturas de dados [7, 8].

Quanto à linguagem de programação BASIC é possível perceber pelo acrónimo *Beginner's All-purpose Symbolic Instruction Code* a sua filosofia de facilidade de uso. Este ficou famoso pela introdução dos primeiros microcomputadores em meados dos anos 70 e continuou a sua popularidade até finais dos anos 80 com numerosas revistas a publicar código fonte BASIC [9, 10, 11]. A linguagem foi continuada pela Microsoft através do aparecimento do Visual Basic (sendo a sintaxe apenas a única familiaridade), o VBScript e o Visual Basic .NET [12].

Outra linguagem mencionável é o Logo, com o slogan “uma linguagem para aprendizagem”, concebida para o ensino de conceitos matemáticos a crianças por via de programação. As semelhanças da sua sintaxe com linguagem com uma língua natural são uma razão forte para o seu uso no ensino. Tal como uma linguagem natural esta também pode ser aprendida em pequenos incrementos, ao começar por conceitos concretos e pela criação de gráficos (uma das características mais fortemente associada à linguagem, os *turtle graphics* [13]) passando para conceitos mais abstractos [14].

De entre as linguagens de ensino à programação que têm como público alvo crianças existem dois grupos que se distinguem: Editores de programação gráficos (Blockly<sup>1</sup>, Scratch<sup>2</sup>) e edição textual de alto nível (Guido van Robot<sup>3</sup> baseado em Python, e Hackety Hack<sup>4</sup> que ensina Ruby). Nos editores de programação gráfica tentam reduzir o esforço inicial de memorizar código para em vez disso se focarem nos essenciais da programação, adiando o problema para mais tarde uma vez que depois os alunos que usaram editores gráficos sentirão maior atrito ao passarem para linguagens textuais. A desvantagem de começar logo com uma linguagem textual é que pode-se dar uma maior fricção inicial, o que poderá causar desmotivação.

Algo que existe em comum em todas as linguagens é o *feedback* imediato que providenciam e a gratificação que prometem. Muitos têm lições associadas à sua linguagem [15, 16], estes ensinamentos prestados têm um tema relacionado com jogos assim como uma conotação lúdica para prender a atenção dos utilizadores durante uma lição ao mesmo tempo que usam também o tema de construção de um jogo como um meio para prender a atenção ao longo de todo o ensino da linguagem [17, 18].

---

<sup>1</sup><https://code.google.com/p/blockly/>

<sup>2</sup><http://scratch.mit.edu/>

<sup>3</sup><http://gvr.sourceforge.net/>

<sup>4</sup><http://hackety.com/>

## 2.3 Massive Open Online Courses

Os *Massive Open Online Courses* (MOOC) são a evolução dos cursos por correspondência para o meio tecnológico da *Internet* [19].

São cursos *online*, disponibilizados gratuitamente ou com inscrição paga. Alguns destes oferecem certificação mediante o pagamento de taxas. Comparativamente a um curso tradicional existem professores associados, datas de início e de fim, participantes, aulas, exercícios e interação com os colegas [20].

No entanto, MOOCs são um subconjunto do *e-learning* e distinguem-se de outros métodos como a distribuição linear de conhecimento por que se focam na aprendizagem colaborativa. Um MOOC é um evento onde pessoas que se interessam por um assunto podem reunir-se e aprender apoiando-se mutuamente [20].

Este termo foi usado pela primeira vez em 2008 por Dave Cormier como resultado de um curso intitulado “*Connectivism and Connective Knowledge*” no qual 25 alunos fizeram-no para obtenção de crédito curricular e outros 2200 por livre acesso [21].

## 2.4 Análise dos MOOCs recentes

Existe um número crescente de plataformas que disponibilizam MOOCs, fazendo-o através de associações com Universidades. Por exemplo, três das plataformas mais conhecidas, edX<sup>5</sup> têm ligações com o MIT e a Universidade de *Havard*, Udacity<sup>6</sup> com *Stanford*, e Coursera<sup>7</sup> com bastantes ligações, entre elas *Stanford*, *Princeton*, Universidade de *Michigan* e Universidade de *Pennsylvania* [22]. Os MOOCs, devido à suas parcerias e derivados instrutores, têm um nível de educação predominantemente ao nível do ensino superior.

Fazem forte uso de vídeo (Figura 2.1), disponibilizando muitas vezes alternativa textual como transcrição completa ou subtítulos separados [23]. Quanto ao estilo de gravação existem muitos estilos variados, ficando à discrição do professor e variam entre: professores que se gravam a dar uma aula tradicional numa sala de aulas a alunos presentes, professores que se filmam a falar para a câmara, que apenas gravam o ecrã sincronizado com a sua voz a explicar a matéria, ou uma mistura dos dois com uma gravação do ecrã com o material e também a sua cara no canto [24]. É de referir que os professores envolvidos indicaram que ao filmarem as suas aulas têm se tornado mais conscientes destas e têm consciente ou sub-conscientemente tomado passos para melhorar as suas aulas (incluindo as suas aulas presenciais) [24].

Mesmo que um vídeo possa parecer um meio arrastado e entediante, o que é certo é que o *feedback* dos alunos indica que mesmo tendo disponibilizado quer pelo professor quer pela *Internet*, existe uma preferência por ver a matéria a aparecer gradualmente no

---

<sup>5</sup><https://www.edx.org/>

<sup>6</sup><https://www.udacity.com/>

<sup>7</sup><https://www.coursera.org/>

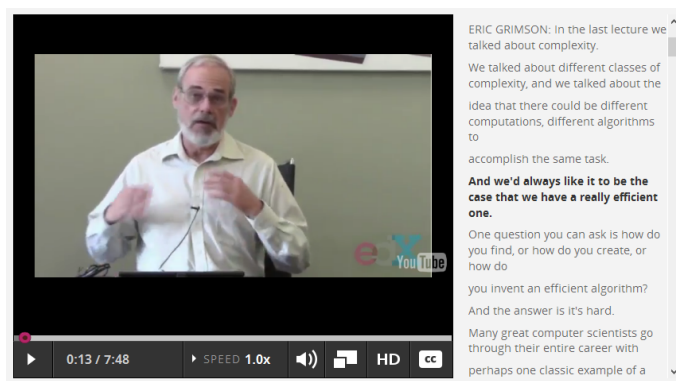


Figura 2.1: Aula de programação no edX, com transcrição à direita

ecrã à medida que é explicado ao invés da alternativa que é ter a matéria toda à frente [25].

A duração dos vídeos varia entre 5 minutos a uma hora [26], mas vídeos mais breves são mais benéficos para não perder a atenção dos alunos, sendo a duração mais usual entre 8 a 12 minutos [25]. Para além disto geralmente é possível alterar a velocidade de reprodução dos vídeos.

Existem também a parte das questões, dado mais ênfase por uns do que outros devido ao facto que esta é uma área muito recente e portanto existe muito espaço para experimentar diferentes medidas e também existem diferentes necessidades e possibilidades consoante o tipo de matéria leccionada. Estas questões têm importância variada: entre questionários leves a meio dos vídeos das lições para cimentar o conhecimento recente antes de prosseguir, a exercícios cotados (Figura 2.2) que contam para a nota de modo a garantir o compromisso dos alunos perante os exames finais para elegibilidade a receber o certificado de passagem emitido normalmente pelo professor responsável pelo curso. O número normal de perguntas varia entre 4 a 10, havendo uma preocupação em não amontoar demasiada matéria diferente mas disponibilizar exercícios assim que a matéria relacionada foi leccionada.

O tipo de perguntas utilizado varia entre os tradicionais métodos de escolha múltipla com várias opções correctas possíveis, botões de opção com só uma escolha seleccionável, resposta aberta até aos métodos que fazem vantagem do meio tecnológico em que se inserem como expressões matemáticas e químicas que são interpretadas enquanto escritas, código fonte com sintaxe realçada. Quanto à avaliação a maioria das perguntas são do tipo linear, permitindo um *feedback* imediato. No que toca às perguntas de resposta aberta os seguintes métodos são utilizados para avaliação: ou existe uma comparação com todas as respostas válidas previstas que o aluno pode inserir (se bem que é difícil prever casos limite, como por exemplo alunos que se queixam de ter respostas certas mas terem sido marcadas erradas por terminarem a resposta com um ponto final) ou existe o *peer-reviewing* [27] no qual a avaliação é feita pelos colegas através de um guia dado pelo

**Question 3**

After the following assignment statement has been executed, which expression(s) produce the letter 'g'?

```
dance_style = 'Vogue'
```

Your Answer	Score
<input checked="" type="checkbox"/> <code>dance_style[-3]</code>	0.25
<input type="checkbox"/> <code>dance_style[-4]</code>	0.25
<input type="checkbox"/> <code>dance_style[1]</code>	0.25
<input checked="" type="checkbox"/> <code>dance_style[2]</code>	0.25
Total	1.00 / 1.00

**Question Explanation**

Positive indices count from the left-hand side with the first character at index 0, the second at index 1, and so on.  
Negative indices count from the right-hand side with the last character at index -1, the second last at index -2, and so on.  
You can, of course, try running these.

Figura 2.2: Pergunta no Coursera, com cotação e explicação

professor, guia este que também é utilizado no método seguinte, a auto-avaliação, na qual o próprio aluno no fim vai analisar se o seu trabalho está consoante esse guia. Outro método é a análise via Inteligência Artificial o qual permite que um programa tente fazer essa tarefa de verificar se o trabalho está em conformidade com o guia do professor. Existem muitas queixas dos alunos quanto a este método por ser muito falível e deixar muito a desejar [28].

Também é usual em algumas perguntas permitir ver a resposta e haver uma explicação da questão para perguntas respondidas de modo errado.

O outro aspecto de um MOOC e talvez o que mais o diferencia dos outros meios de aprendizagem na *Internet*, é o aspecto social. Enquanto que material de aprendizagem pode ser obtido com mais ou menos facilidade pela *Internet*, o ponto forte de um MOOC é que junta pessoas com o mesmo interesse: o de aprender sobre a matéria da classe. É nesse grupo de pessoas que a aprendizagem é reforçada, ao partilhar dúvidas e conhecimento com outras pessoas. Outro dos objectivos de um MOOC é o de efectuar ligações entre pessoas que vão para além da classe, tal como no ensino tradicional [20, 23].

Quanto à creditação, alguns cursos têm datas limites para início e conclusão e datas limites para conclusão dos exercícios, sendo possível a inscrição nestes cursos fora das datas mas sem possibilidade de creditação na conclusão do curso [26]. Também existem cursos com creditação mas sem datas limites. Existe nos utilizadores uma opinião dividida quanto a esta restrição de tempo, sendo que para uns é um bom motivador para manter a disciplina e o ímpeto e não deixar cair em procrastinação, enquanto que para outros é um factor limitador cuja vida profissional ou pessoal faz dar preferência por soluções com horários mais flexíveis [29]. A creditação mencionada pode ou não existir, ser um documento dado pelo docente responsável pela disciplina, ou em certos casos especiais (através do pagamento de uma taxa e a execução de um exame supervisionado) crédito

curricular reconhecido em instituições selectas [26].

O resultado de uma classe destas não é reconhecido em instituições de ensino (salvo especiais excepções), pois este é um campo recente, havendo desconfiança sobre a sua fiabilidade. O próprio ensino à distância sempre teve uma conotação inferior ao ensino presencial.

Algo que difere largamente entre os MOOCs e o ensino tradicional é o enorme número de alunos inscritos que não chega a terminar o curso, mais de 90%. Isto pode dever-se a um variado número de factores: alunos que não conseguem regular o seu ensino, apenas foram ver sobre o que é o curso uma vez que as inscrições são normalmente grátis e não existe nenhum impedimento à inscrição impulsiva, não estavam preparados para o rigor de ensino académico, obtiveram o que quiseram do curso e não se preocuparam em acabá-lo, tomam uma posição passiva e não participam activamente, etc [30]. Este alto nível de desistências já causou situações de desentendimento entre as plataformas fornecedoras de MOOCs e as instituições associadas [31] e só irá contribuir mais para a descredibilização do meio enquanto não se investigar este fenómeno e conseqüentemente possíveis soluções.

Também existe a questão da falsificação de respostas. Normalmente os alunos são encorajados a procurar material na *Internet* caso tenham dúvidas ao responder à matéria, mas as únicas tentativa feitas para evitar a cópia de respostas em várias plataformas de MOOC são, para além dos termos de acordo aceite pelos utilizadores ao se registarem, a concordância com um código de honra [32] que estipula que não deverão discutir perguntas de exame nem disponibilizar soluções e também a vigilância dos fóruns por comentários que vão contra este código. Perguntas sobre exercícios normais são permitidos, apenas os pertinentes ao exame final da classe é que são restritos. Esta questão da falsificação serve também para diminuir a reputação dos MOOCs, pois é muito fácil obter respostas ou buscar soluções sem poder haver um meio eficaz de combater esta situação. Uma alternativa encontrada foi o de disponibilizar exames supervisionados, no entanto estes não são gratuitos.

## 2.5 Ensino da programação no contexto dos MOOCs

O ensino é efectuado recorrendo frequentemente a vídeos intercalados com questionários sobre a matéria acabada de dar. Os tópicos iniciais são de natureza técnica, como instalar e preparar as ferramentas necessárias para programar e correr o código ensinado. Os professores filmam o ecrã, onde escrevem o código referente ao conceito a ensinar. Geralmente se é um conceito avançado englobam-no num problema concreto para ajudar a visualizar a conexão entre o teórico e o prático. Enquanto vão escrevendo o código, vão fazendo comentários sobre a utilidade e a lógica do programa ser feito desta maneira, sem pressa ao escrever o código todo, mas em vez disso focando-se em comentar a ideia

subjacente ao escrever o programa.

Existe uma grande predominância entre os professores das várias plataformas de MOOC para utilizar o Python como a linguagem de programação para ensinar os conceitos necessários nas disciplinas de introdução à programação.

Os exercícios de classes de programação costumam usar o mesmo formato usual que outros tipos de matérias - perguntas sobre programação com resposta, por exemplo, de escolha múltipla.

Algumas plataformas (como o Udacity) utilizam interpretadores remotos para analisar códigos de resposta dos exercícios, envolvendo programação e código concreto nas suas respostas. Esta vertente, comparando com a alternativa de respostas genéricas de escolha múltipla, envolve mais os utilizadores na programação tornando a aprendizagem mais prática mas esperando já um nível de entendimento de conceitos que o aluno pode não ter percebido da aula, e é difícil de inferir com detalhe onde o aluno tem mais dificuldade, enquanto que nas respostas simples de escolha múltipla afere-se directamente os conceitos teóricos e consegue-se quantificar o entendimento do utilizador em cada um destes.

## 2.6 O caso Duolingo

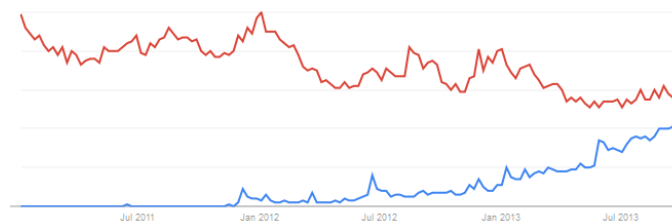


Figura 2.3: Volume de pesquisas no Google sobre o Duolingo (a azul) versus Rosetta Stone (a vermelho), um outro famoso produto de ensino de linguagens, entre Janeiro de 2011 a Setembro de 2013[33]

O Duolingo, em vez de tentar imitar um curso tradicional, faz pleno uso do novo meio tecnológico em que opera (a *Internet*) e das suas vantagens. Não se comporta como um MOOC, uma vez que não tem professores e nem o fulcro da aprendizagem é a discussão com colegas mas sim a resolução de perguntas e a tradução de textos genuínos. Este encaixa na definição mais abrangente de *e-learning* mas não é um MOOC.

Os objectivos do Duolingo são o de ensinar uma língua de maneira gratuita enquanto se traduzem documentos de forma *crowdsourced*, sendo este o seu método de rendimento e a maneira que consideraram mais fiável para tornar o registo aberto a todos [34].

O Duolingo utiliza uma aproximação do tipo lúdica ao ensino através de recompensas e níveis a alcançar. Permite também utilizar estes níveis arbitrários para fomentar competitividade entre amigos, enquanto no *backend* o Duolingo utiliza uma abordagem

*data-driven* ao analisar as questões nas quais o utilizador tem mais dificuldade e os tipos de erro que comete para personalizar futuras sessões de perguntas.

### Select the correct image

“the newspaper”

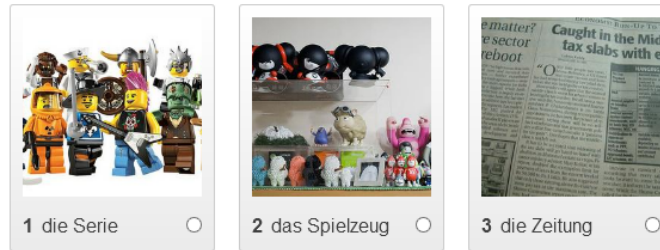


Figura 2.4: Exemplo de questão inicial para ensinar conceito novo no Duolingo

# Capítulo 3

## Análise

### 3.1 Objectivos

O objectivo deste projecto é o de tirar partido dos métodos destes recentes desenvolvimentos na educação, nomeadamente na sua vertente *online*, para a propagação do ensino da programação a pessoas sem conhecimento prévio. Num mundo cada vez mais tecnológico ter este tipo de conhecimento é uma vantagem: fica-se a conhecer mais sobre um meio que se utiliza tão frequentemente e pratica-se a lógica. No entanto a sua aprendizagem nem sempre é muito facilitada, o que cria uma espécie de barreira em que só quem é realmente dedicado e tem proficiência é que tem sucesso.

Este projecto visa utilizar métodos inspirados no site de *e-learning* Duolingo. A sua aproximação envolvente e gradual é ideal para manter o utilizador envolvido na sua aprendizagem e a utilização de *crowdsourcing* aperfeiçoará a matéria para as necessidades que os alunos enfrentarem; espera-se que este projecto torne o ensino de programação mais alargado e acessível.

### 3.2 Contornos do sistema

O sistema irá ensinar linguagem de programação usadas em produção. Não são utilizadas linguagens de programação educacionais (desenvolvidas para a educação de conceitos de programação) nem será desenvolvida uma linguagem nova expressamente para o ensino da matéria. As linguagens previstas para serem ensinadas são o C, Java e Python, e põe-se ainda a possibilidade da linguagem Haskell.

Não se vai utilizar vídeo nem o conceito de aulas utilizado nos MOOCs. Toda a aprendizagem irá ser feita recorrendo ao uso de perguntas de crescente dificuldade. Usando essas perguntas ir-se-à localizar os conceitos que o aluno ainda não entendeu bem e adaptar as perguntas futuras nesse utilizador para irem de encontro às suas necessidades educacionais. Fora das perguntas poder-se-à disponibilizar páginas com instruções sobre como instalar as ferramentas necessárias a obter um ambiente de produção para testar os con-

ceitos ensinados, para que o ensino não seja só teórico.

O sistema também terá a componente social tal como está presente no Duolingo enquanto pilar da ideologia MOOC, onde os alunos poderão comentar as questões, colocar dúvidas, responder a dúvidas de outros utilizadores e também reportar problemas em questões (como perguntas inválidas ou respostas não considerada certas).



Figura 3.1: Acesso às discussões em cada pergunta no Duolingo

O sistema estará acessível pela *Internet* com possibilidade de vários utilizadores conectados simultaneamente.

Será implementado com um *front-end* em HTML e JavaScript e *back-end* em Java com uma possível ligação a uma base de dados numa versão mais avançada. Pretende-se manter a complexidade reduzida inicialmente para focar o desenvolvimento nas questões envolvendo mais a interação e a funcionalidade e pondo as questões de desempenho e escalabilidade para depois.

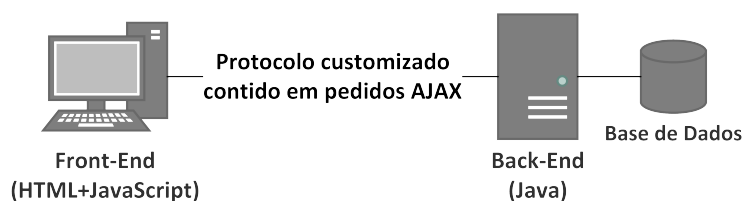


Figura 3.2: Arquitectura do sistema

A conectividade entre o *front-end* e o *back-end* será feita por AJAX através do qual o *back-end* irá transmitir instruções de controlo sobre a exibição e os dados dinâmicos, enquanto que o *front-end* irá conter a parte estática do visual e também o interpretador dos comandos vindos do *back-end*. Devido a este esquema de controlo remoto irá ser necessário aplicar camadas de segurança sobre esta comunicação para colmatar possíveis ataques maliciosos.

# Capítulo 4

## Desenho

Neste capítulo são mencionados os conceitos utilizados na implementação do sistema.

### 4.1 Lições

As lições são a unidade maior do sistema. São conjuntos de perguntas agrupados sob um tema em comum. Uma lição pode ter um conjunto de lições como requerimento, nesse caso a lição só poderá ser feita caso todas as lições em requerimento tenham sido completadas.

As lições iniciais de uma linguagem (lições disponíveis logo à partida) são as lições que não têm nenhum requerimento.

### 4.2 Questões

As questões são os elementos que constituem uma lição, e cada questão pode ter associada a si múltiplas respostas possíveis, dentro destas havendo um conjunto de respostas marcado como correcto.

O uso dos campos da questão são possivelmente interpretados diferentemente dependendo do tipo da questão.

Secções do enunciado podem ser marcados para apenas serem visíveis no modo de ensino e omitidos no modo de avaliação.

### 4.3 Tags

Tags são conceitos genéricos utilizados para marcar questões, utilizadores ou lições. O uso mais comum de tags é o de marcar o conceito que uma questão representa e o conceito que desbloqueia, método pelo qual se define a ordem das questões.

Tags associadas a lições e utilizadores são apelidadas de metatags e servem para atribuir notas e gerir utilizadores (Secção 5.4.1).

## 4.4 Sessões

Sessão é o conceito de uma lição efectuada por um utilizador. É utilizado para se poder guardar informação detalhada de uma lição feita por um utilizador, mesmo que este repita a mesma lição múltiplas vezes, podendo-se consultar, por exemplo, quando é que o utilizador terminou pela primeira vez a lição, ou qual foi o melhor que conseguiu responder à lição.

## 4.5 Tipos de perguntas feitas ao aluno

As perguntas serão feitas com dificuldade crescente. Cada tipo de pergunta pode ter vários níveis de dificuldade. Não é linear se escolha múltipla é mais fácil do que resposta aberta, a dificuldade irá consistir num misto entre o enunciado, conhecimento do aluno e respostas disponíveis.

O tipo de perguntas a serem colocadas serão de escolha múltipla (com ou sem múltiplas respostas correctas) onde se poderá por exemplo perguntar sobre uma propriedade da linguagem e as respostas disponíveis serão limitadas a verdadeiro / falso, ou poder-se-á mostrar código e pedir ao utilizado que escolha qual é o resultado de correr o código, ou ainda mostrar código com uma linha oculta e perguntar qual é a linha necessária para obter um certo output.

Outro tipo de perguntas é o de resposta aberta, que irá comportar-se como a escolha múltipla, excepto nas perguntas de verdadeiro ou falso, tendo o utilizador que achar a resposta correcta sem ter a ajuda de uma selecção limitada. Este tipo de perguntas dará a possibilidade de guardar as respostas dos utilizadores para perguntas futuras, e ganhar o conhecimento automático de confusões que utilizadores estejam a fazer, utilizando essas confusões para criar perguntas com rasteiras comprovadas.

Um tipo intermédio é a resposta aberta com selecção, na qual uma parte da resposta é pergunta aberta e outra parte (a mais difícil) é uma selecção entre várias possibilidades. Este tipo de perguntas irá possibilitar uma transição de escolha múltipla para resposta completamente aberta sem no entanto oprimir o utilizador com conceitos em que não esteja ainda à vontade ajudando-o aí com a escolha múltipla.

Outro tipo de questões é o de ordenação de código. Ao dar um objectivo ao utilizador e o código que o atinge mas desordenado, o utilizador tem de pôr o código na ordem correcta. Este tipo é ideal para utilizadores que não estejam à vontade para escrever código ainda, providenciando a vantagem que linguagens gráficas têm. Para aumentar a dificuldade pode-se dar mais linhas de código por ordenar do que as necessárias (com alterações pequenas mas relevantes que levarão o utilizador a revelar o seu conhecimento na sua distinção) ou também linhas em branco por preencher pelo utilizador.

As perguntas serão associadas apenas a uma lição (nó na hierarquia de conceitos). Cada pergunta tem um “tag” associado que é o tema principal da pergunta e servirá

também para as ordenar dentro de uma lição. Contêm o enunciado e as respostas que se podem seleccionar.

Dos tipos de perguntas contemplados apenas a escolha múltipla e ordenação de código foram implementados (Secção 7.6).

## 4.6 Casos de Uso

### 4.6.1 Registo de conta

**Sumário:** Utilizador regista-se no sistema

**Actor:** Utilizador

**Pré-condições:** O utilizador deseja utilizar o sistema e ainda não tem conta registada

**Pós-condições:** O utilizador tem uma conta associada a si registada no sistema

**Fluxo Principal:**

1. Utilizador dirige-se ao site do projecto.
2. Escolhe um nome de ecrã pelo qual será conhecido e uma password.
3. Depois de preencher os campos com os dados clica no botão de registo.
4. Selecciona qual a linguagem de programação que vai começar a aprender.

**Fluxo Alternativo:**

(3) Utilização de e-mail

- O utilizador pode ainda indicar um e-mail que será utilizado para fins de recuperação de password e notificações.

### 4.6.2 Autenticação do utilizador

**Sumário:** Utilizador autentica-se com uma conta que já criou previamente

**Actor:** Utilizador

**Pré-condições:** O utilizador já criou uma conta anteriormente

**Pós-condições:** O utilizador encontra-se autenticado perante sistema

**Fluxo Principal:**

1. Utilizador dirige-se ao site do projecto.
2. Introduce o nome de utilizador e a password associados à sua conta.
3. Depois de preencher os campos clica no botão de *login*.

### 4.6.3 Mudança da linguagem de programação activa

**Sumário:** Utilizador altera qual a linguagem de programação que está a aprender

**Actor:** Utilizador

**Pré-condições:** O utilizador já criou uma conta anteriormente e está autenticado

**Pós-condições:** O sistema apresenta exercícios relativos à linguagem agora escolhida, sem esquecer o progresso na linguagem anterior

**Fluxo Principal:**

1. Utilizador escolhe item de mudança da linguagem do menu.
2. Escolhe nova linguagem de programação.

### 4.6.4 Resolução de exercícios

**Sumário:** Utilizador resolve um conjunto de exercícios relativos à linguagem de programação escolhida pelo utilizador

**Actor:** Utilizador

**Pré-condições:** O utilizador está registado e autenticado

**Pós-condições:** O utilizador concluiu os exercícios, aprendendo sobre a linguagem

**Fluxo Principal:**

1. O utilizador escolhe de entre as matérias na hierarquia de conceitos que lhe está disponível.
2. Para cada pergunta responder à questão
3. Ao responder um número suficiente de questões sem errar demasiadas considera-se essa matéria como aprendida, desbloqueando outras matérias.

**Fluxo Alternativo:**

(2) Resposta errada

- No caso da resposta estar errada o sistema apresenta a solução correcta assim como uma explicação.
- Futuras perguntas serão ajustadas para focarem na pergunta que o utilizador mostrou dificuldade.

(2) Interacção social

- O utilizador poderá discutir a pergunta com outros utilizadores utilizando o botão de discussão visível após responder à questão actual, sem interromper a série de perguntas

(3) Insucesso na resolução

- Caso o utilizador falhe um número suficiente de perguntas a tentativa termina e o utilizador terá de a reiniciar novamente do princípio.



# Capítulo 5

## Implementação

Neste capítulo refere-se os detalhes e conceitos aplicados na implementação do sistema. Quando pertinente referencia-se em rodapé as linhas de código relevantes ao assunto a ser mencionado.

### 5.1 Componentes Principais do Projecto

Existem três principais componentes independentes no Moxo:

Um **backend**, que contém a lógica do sistema e proporciona acesso à base de dados.

Um **frontend**, a interface web pela qual os utilizadores vão aceder e usufruir do sistema.

Um **editor**, que providencia acesso à base de dados para utilizadores privilegiados, como por exemplo um professor, disponibilizando facilidade à criação de perguntas e gestão e monitorização dos utilizadores e seus resultados.

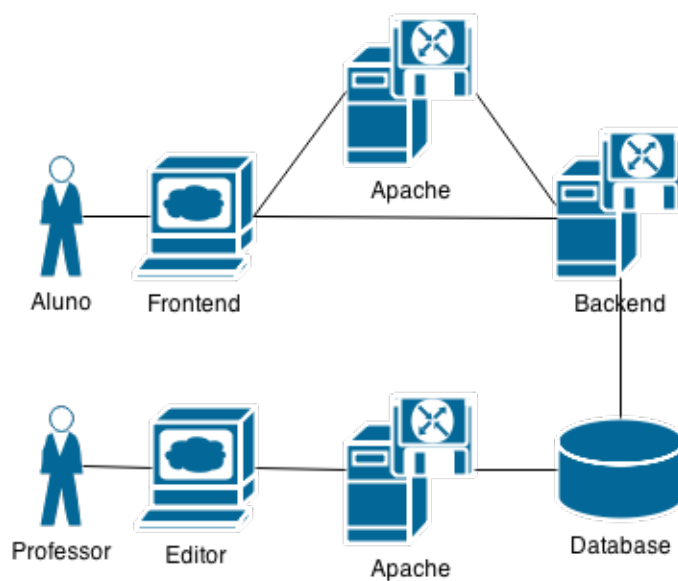


Figura 5.1: Interação dos utilizadores com os componentes.

O editor foi desenhado para ser independente do backend e funcionar como uma espécie de frontend para a base de dados.

O frontend nunca se liga directamente à base de dados, usa sempre o backend como intermediário. O servidor web Apache faz pedidos ocasionais ao backend para propósitos de balanceamento de carga e verificação de redireccionamento de página (como no caso do utilizador ainda não ter escolhido uma linguagem de programação).

## 5.2 Backend

### 5.2.1 Base de Dados

Inicialmente para guardar os dados das questões foram utilizados ficheiros de formato *xml*. Com a necessidade de manter um maior número de dados foram então implementadas conexões genéricas para bases de dados, testados para funcionar para múltiplos tipos de bases de dados (MySQL e SQLite). Após essa mudança para a utilização de bases de dados passou-se a utilizar proeminentemente bases de dados do tipo MySQL, mas em teoria pode-se utilizar qualquer tipo suportado por um conector JDBC.

Devido às múltiplas mudanças que a base de dados veio a sofrer durante o desenvolvimento do projecto, veio-se a considerar se seria melhor usar um modelo entidade-atributo-valor, mas as mudanças não eram frequentes o suficiente para justificar o uso desse modelo, além do esforço extra em código para compensar a dispersão dos dados.

### 5.2.2 Representação interna dos dados

O back-end não guarda os dados directos, apenas referências à base de dados pelo id, funcionando como um *thin client*, garantindo que os dados na base de dados e no servidor java estão sempre consistentes.

O único ponto de inconsistência é onde não se conseguiu sincronizar os dados, na representação interna das questões de uma lição. Se depois de iniciar o back-end houver uma modificação na ordem ou tags principais das perguntas então esta alteração só será observada depois de reiniciar o back-end. Isto deve-se ao facto de que gerar a árvore interna das perguntas de uma lição é um processo dispendioso em termos de múltiplos pedidos à base de dados, então em vez de gerar essa árvore cada vez que uma lição é mencionada gere-se apenas uma vez e guarda-se numa cache<sup>1</sup>.

Também nos deparámos com um problema na persistência da ligação do back-end à base de dados que se desconectava após grandes periodos de tempo (durante a noite). Isto deveu-se a que inicialmente utilizou-se uma única ligação persistente à base de dados<sup>2</sup>,

---

<sup>1</sup>`moxo-backend/src/moxo/Lesson.java:30`

<sup>2</sup>`moxo-backend/src/moxo/MoxoDatabase.java`

que várias fontes na internet indicaram como má prática, tendo sido a ligação mudada para se efectuar a cada novo pedido necessário<sup>3</sup>.

### 5.2.3 Tags

As tags são uma componente importante, ajudando a organizar, marcar e ordenar os vários elementos da base de dados. São utilizadas nas questões e nas lições para marcar a ordem destas.

As metatags são tags utilizadas em contexto especial, podem ser usadas apenas como comentário para associar a uma questão, lição ou utilizador, ou se fôr uma tag com comportamento associado pode afectar a quem está atribuída.

Por exemplo a tag 'modo avaliação' que força o utilizador com a tag a só poder efectuar sessões de perguntas em modo de avaliação.

### 5.2.4 Ordem das questões

Para cada lição é gerada uma árvore interna (Figura 5.2) para ser usada para escolher a próxima questão. Cada nó da árvore representa um par tag/dificuldade e cada nó tem uma referência ao nó anterior (para o caso de ser necessário repetir um conceito) e referências para os nós seguintes, que têm o mesmo tag mas dificuldade superior (com valor do nó actual mais um) e tags desbloqueadas por questões pertencentes ao nó corrente.

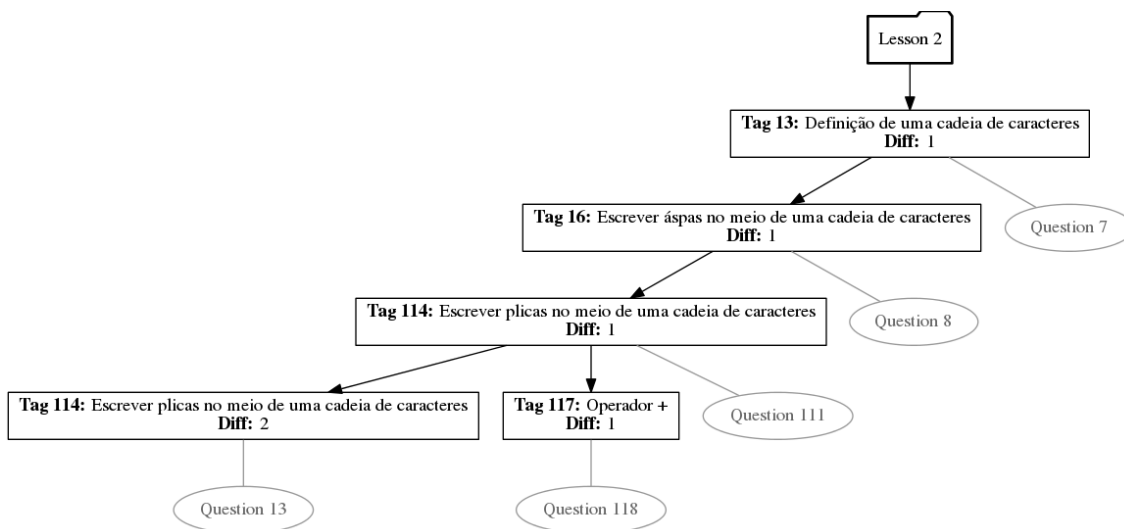


Figura 5.2: Árvore de estados de uma lição

A árvore de uma certa lição é gerada da primeira vez que se acede a essa lição na base de dados<sup>4</sup>. Como se acede frequentemente às lições e seria dispendioso estar sempre

<sup>3</sup>moxo-backend/src/moxo/MoxoSystem.java

<sup>4</sup>moxo-backend/src/moxo/Lesson.java:41

a actualizar esta árvore (especialmente porque é constante caso não haja alterações às lições/questões) é utilizado uma cache<sup>5</sup> que guarda as árvores geradas passando a ser estas utilizadas ao longo da utilização do sistema. A desvantagem é que caso haja uma alteração na estrutura das questões que requeresse a actualização da árvore seria necessário reiniciar o sistema, pois este continuaria a utilizar a contida na cache que estaria desactualizada (7.4).

Quando o utilizador responde correctamente a uma questão pertencente a um nó (questão com o mesmo tag e mesma dificuldade do nó) os nós seguintes são desbloqueados<sup>6</sup>. Para escolher qual a questão a apresentar é escolhido uma questão aleatoriamente de entre todos os nós desbloqueados<sup>7</sup>.

No caso de haver duas questões com a mesma tag e a mesma dificuldade, então essas questões são interpretadas como variantes e ao responder correctamente não é necessário apresentar a outra por ser redundante.

Quando o utilizador responde erradamente volta-se a obter uma questão aleatoria dos nós desbloqueados. Préviamente o utilizador era retornado ao nó anterior, o que causava repetição de perguntas que o utilizador já tinha respondido correctamente, causando repetitividade e sentido de perda de progresso, o que se tornou prejudicial pois era desmotivador e em vez de prestar mais atenção à matéria que possivelmente ajudava à questão seguinte os utilizadores focaram-se mais na memorização para poderem avançar.

Para depuração destas representações existe a opção de gerar um ficheiro *.dot*<sup>8</sup> com a representação gráfica destas árvores consoante forem geradas, possibilitando uma maneira fácil de verificar visualmente a correcção da ordem das questões.

Existe um limite máximo de questões que um utilizador pode responder até falhar a lição e ter que a repetir. Para obter esse número limite usou-se o tamanho do ramo maior da árvore de representação interna<sup>9</sup> mais 20% de folga.

## 5.3 Frontend

### 5.3.1 Apresentação

O frontend é a interface gráfica usada pelo utilizador para interagir com o sistema.

Este foi implementado com PHP e Javascript, permitindo o acesso via browser para alargar o acesso a qualquer plataforma com browser compatível. Foi testado principalmente com o Firefox. Não foi utilizado código específico do browser, por isso não se

---

<sup>5</sup>moxo-backend/src/moxo/MoxoSystem.java:693

<sup>6</sup>moxo-backend/src/moxo/LessonSession.java:175

<sup>7</sup>moxo-backend/src/moxo/LessonSession.java:105

<sup>8</sup>Formato utilizado pelo programa Graphviz para gerar grafos especificados por linguagem DOT.

<sup>9</sup>moxo-backend/src/moxo/LessonSession.java:65

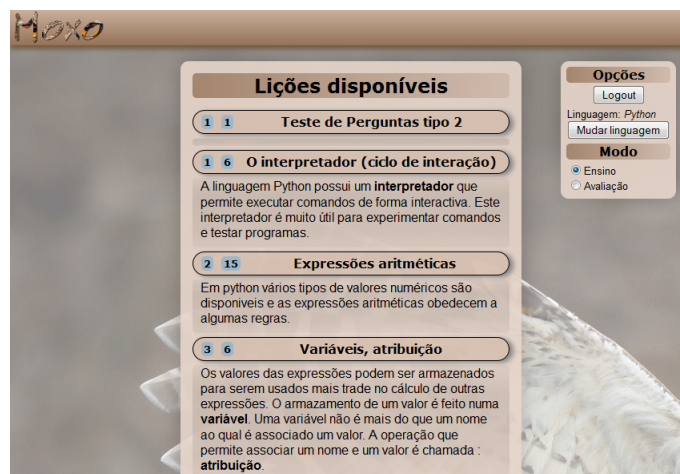


Figura 5.3: Página de selecção de lição

prevê grandes incompatibilidades mas será necessário efectuar testes para outros browsers que poderão vir a ser utilizados (7.7).

O frontend serve como um cliente leve, na medida em que apenas serve a interface ao utilizador. Todos os dados a receber ou a introduzir no sistema são tratados pelo backend, servindo o frontend de intermediário.

O frontend pode ser definido para distribuir a carga entre múltiplos backends, usando o código em `moxo-frontend/server_selector.php`. Para definição dos backends disponíveis existe o ficheiro de configuração `frontend/backend-servers.csv` que contém uma lista dos servidores disponíveis e um índice de prioridade com o qual é calculado qual o servidor a utilizar, um servidor com índice de prioridade 3 receberá três vezes mais utilizadores um de índice 1. Para a escolha obtém-se o servidor com menor número de utilizadores dividido pelo índice de prioridade<sup>10</sup>, contactando-se para isso todos os servidores backend da lista para se obter o número de utilizadores e a versão, e descartar os servidores que não respondem. Depois de ser escolhido o servidor é assinalado<sup>11</sup> ao utilizador até que este reinicie o browser ou termine a sessão no sistema. Também foi previsto a possibilidade de existir múltiplas versões: o frontend evita escolher backends com versões mais recentes que a sua para evitar incompatibilidades de protocolos.

A comunicação entre o frontend e o backend é efectuada usando *Cross-origin resource sharing*<sup>12</sup>, que na sua essência são pedidos AJAX mas com permissões a domínios externos, necessário para o caso de o backend não se encontrar hospedado no mesmo local do frontend. Estes pedidos são processados no backend usando a biblioteca NanoHttpd, e utiliza-se o formato JSON para a estruturação dos dados. As mensagens são transmitidas principalmente em modo POST, e têm no mínimo três campos: *op* com a operação

<sup>10</sup>`moxo-frontend/server_selector.php:53`

<sup>11</sup>`moxo-frontend/server_selector.php:87`

<sup>12</sup>`moxo-frontend/scripts/javascript.js:2`

que se deseja efectuar *user\_id* com o número do utilizador a efectuar o pedido (0 se fôr um pedido sem necessidade de autenticação), e *session\_id* com o número da sessão para verificar se o utilizador que está a efectuar o pedido se encontra autenticado<sup>13</sup>.

Para um caso de uso de um utilizador ter efectuado o login e ir efectuar uma sessão de perguntas, a comunicação entre o frontend e o backend é o seguinte: Ao entrar na página de selecção de lição, o frontend pede qual a linguagem seleccionada pelo utilizador<sup>14</sup>, caso este não tenha seleccionado nenhuma ainda este é redireccionado para a página de selecção, senão manda outro pedido para obter as lições disponíveis ao utilizador<sup>15</sup>. As lições são apresentadas ao utilizador, e ao escolher uma a sua escolha é reportada ao backend<sup>16</sup>. Passa-se para a página de apresentação das questões (`moxo-frontend/question_session.php`) e obtém-se a informação constante da sessão de questões (título, número máximo de questões) com um pedido<sup>17</sup>, após o qual entra-se num *loop* até ao final da sessão, onde-se obtém a questão actual<sup>18</sup> e enviando a resposta do utilizador, recebendo ao mesmo tempo a verificação da resposta, a correcção caso esteja errada e a notificação de informação sobre se a sessão terminou e se com sucesso<sup>19</sup>.

### 5.3.2 Apresentação das Questões

Cada pergunta pode tomar um de múltiplos formatos, previstos inicialmente, tendo no final sido implementado dois formatos: Escolha múltipla (com múltiplas respostas) e ordenação de código.

No início da implementação utilizou-se escolha múltipla como formato principal devido à sua versatilidade e possibilidade de representar a maioria das perguntas genéricas. Este formato apresenta as várias escolhas possíveis verticalmente, podendo o utilizador escolher de entre as respostas possíveis apresentadas qual é, ou quais são, as correctas.

Mais adiante foram introduzidas alterações para acomodar um novo tipo, assim como futuros tipos. Estas alterações foram principalmente mudar na base de dados onde antes se guardavam referências que dependiam do tipo da pergunta (como por exemplo no registo das respostas apresentadas ao utilizador na tabela *useranswer*) passou-se a guardar os dados em formato JSON, sendo depois tratado diferente dependendo do tipo de questão associado.

O novo tipo de questão implementado foi a ordenação de código (Figura 5.4), escolhido para mostrar a possibilidade de tipos de perguntas diferentes coexistirem de acordo com a natureza da matéria que se visava ensinar. Na ordenação é apresentado ao utili-

---

<sup>13</sup>Quando se efectua a autenticação é se indicado um número aleatório atribuído à sessão actual, sendo este número o esperado receber com cada pedido durante a sessão do utilizador.

<sup>14</sup>`moxo-backend/src/ui/HttpUI.java:212`

<sup>15</sup>`moxo-backend/src/ui/HttpUI.java:205`

<sup>16</sup>`moxo-backend/src/ui/HttpUI.java:219`

<sup>17</sup>`moxo-backend/src/ui/HttpUI.java:228`

<sup>18</sup>`moxo-backend/src/ui/HttpUI.java:235`

<sup>19</sup>`moxo-backend/src/ui/HttpUI.java:242`

(35, 185) 1 6

### Mais repetições

Uma vez que sabemos construir listas podemos explorar a instrução `for` (para) que permite executar operações com todos os elementos de uma lista. Por exemplo:

```
a = range(6)
for n in a:
    print n
```

vai imprimir os valores de 0 a 5. Como funciona? Os valores da lista vão ser colocados na variável `n` (um de cada vez) e a instrução `print n` será executada. A instrução `print` vai portanto ser executada tantas vezes como o número de elementos na lista `a`.

Defina uma função que imprime a tabuada dos 7 (note que a função não tem parâmetros logo colocamos `()` a seguir ao nome da função):

```
>>> print n * 7
>>> v = range(10)
def tabuadaDos7():
>>> for n in range(10):
>>> for n in v:
>>> print v * 7
```

Submeter resposta

Figura 5.4: Exemplo de questão com ordenação de código.

zador duas colunas, uma com linhas de código desordenadas, e outra coluna para onde o aluno terá de arrastar as linhas na ordem certa (podendo o autor da questão especificar mais do que uma ordem correcta). Também previsto para este tipo de perguntas, mas não chegado a ser implementado, foi a possibilidade de já haver linhas pré-preenchidas na posição correcta, e também a existência de linhas que permitissem input textual do utilizador.

Após o utilizador submeter a sua resposta é-lhe indicado se está correcta, ou caso contrário apresenta a solução, indicando onde o utilizador esteve certo e onde errou se possível. Esta última situação actualmente somente implementada para escolha múltipla, no caso de errar uma ordenação de código é apresentada uma das soluções, visto que é possível haver múltiplas soluções diversificadas e não será trivial identificar qual a que o utilizador estaria a tentar atingir.

### 5.3.3 Modos de Lição

Existem dois modos com que se pode fazer uma lição: modo de ensino e modo de avaliação. Estes modos foram criados com o intuito de expandir a utilidade do sistema para várias situações.

O modo de avaliação na implementação actual apenas difere do modo de ensino por omitir secções no corpo do enunciado que o professor tenha marcado como só para ensino.

Também é possível forçar o modo com que um utilizador pode efectuar uma lição atribuindo-lhe uma tag especial “Modo ensino” ou “Modo avaliação” para o modo respectivo. No futuro pretende-se que esta função seja útil para permitir ao professor um maior controlo sobre a matéria apresentada, caso este o necessite.

## 5.4 Editor

O editor foi implementado como uma ajuda preciosa para auxiliar os professores na criação das lições/questões assim como a consulta do progresso dos alunos e manutenção geral da base de dados.

Para além da funcionalidade base de criar e modificar lições e suas componentes, também permite pré-visualizar as questões mostrando estatísticas sobre essa questão, tal como qual a resposta mais escolhida (Figura 5.5).

### 5.4.1 Estatísticas Guardadas

Parte dos objectivos do projecto é ajudar os docentes a gerir o progresso dos alunos, para este efeito são guardadas várias informações sobre as respostas dadas.

Para além de outros valores mais utilitários necessários para manter o progresso do utilizador são também guardados dados temporais, quando foi respondido e quanto tempo demorou a responder<sup>20</sup>, assim como dados para reconstituír a sessão, como a ordem das respostas possíveis apresentadas ao utilizador e em que modo (Secção 5.3.3) o utilizador efectuou a sessão.

Também estava previsto a atribuição de pontuações às lições para dar ao utilizador um feedback passível de ser comparado com resultados de outros utilizadores (ou com o seu de tentativas anteriores), com os dados guardados é possível implementar esse sistema (e corrigi-lo se necessário) com efeitos retroactivos, pois é possível reconstruír de maneira suficiente as condições em que o utilizador efectuou a resposta.

Como um dos objectivos deste projecto é ajudar os professores a diagnosticar problemas com os alunos, foi desenvolvido no lado do Editor uma tabela que permite verificar visualmente (assim como alterar) o progresso dos alunos pelas lições, assim como observar para uma lição em específico o número de vezes que as suas questões foram respon-

---

<sup>20</sup>`moxo-frontent/question_session.php:265`; coluna *timer* da tabela *useranswer*.

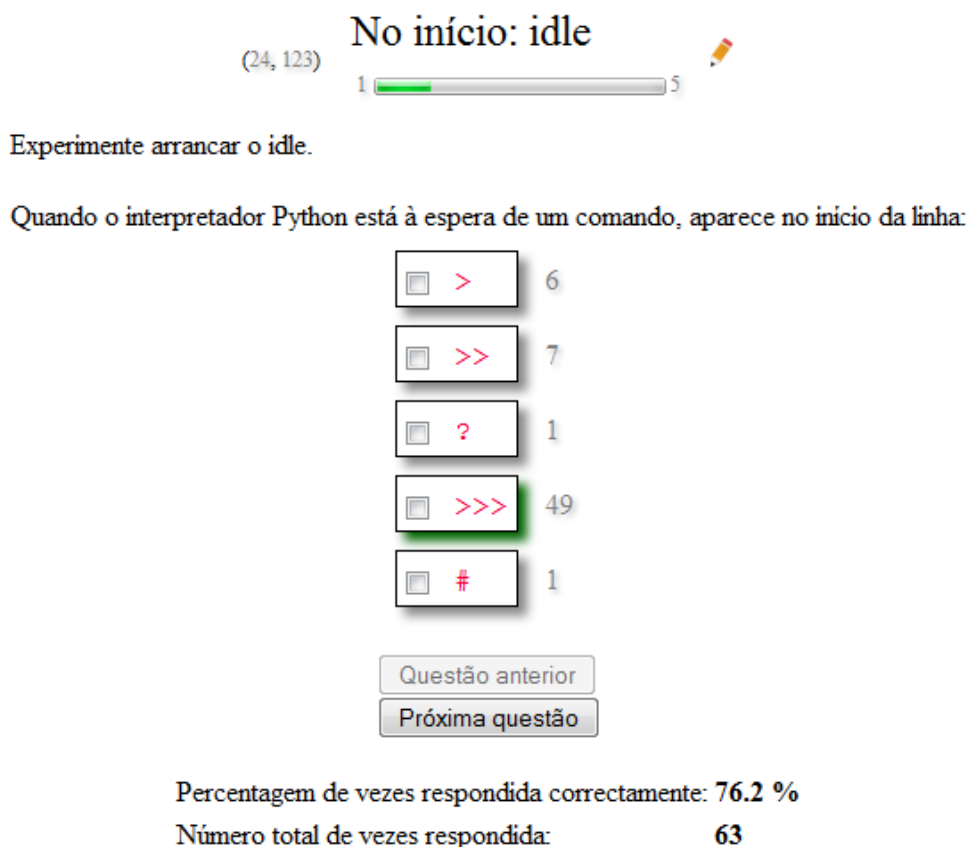


Figura 5.5: Prévisualização de uma questão com estatísticas visíveis.

didadas assim como o nível de correcção dessas tentativas (Figura 5.6). Também permite outras acções como atribuir tags a utilizadores e filtrar a lista de utilizadores por tags, para permitir uma visualização selectiva, como no caso de se querer consultar só uma turma<sup>21</sup>.

### 5.4.2 Tradução para múltiplas línguas

De modo a que o projecto pudesse abranger o maior público possível o sistema foi adaptado cedo para poder lidar com múltiplas traduções de línguas naturais.

Para este efeito, na base de dados todos os campos que são supostos conter texto passível de ser traduzido (somente campos de tipo *string*) passaram a ser uma referência para uma única tabela *translation* que contém todas as traduções em todas as linguagens naturais disponíveis, por exemplo, o valor textual de uma tag é um campo que pode ser traduzido, sendo então uma referência para a coluna *id* na tabela *translation*, sendo necessário depois escolher das múltiplas linguagens naturais disponíveis (coluna *language*, referente à coluna *id* da tabela *naturallanguage*) qual a desejada, as duas mais comuns

<sup>21</sup>Assumindo que o professor atribuiu previamente uma tag representando a turma aos alunos.

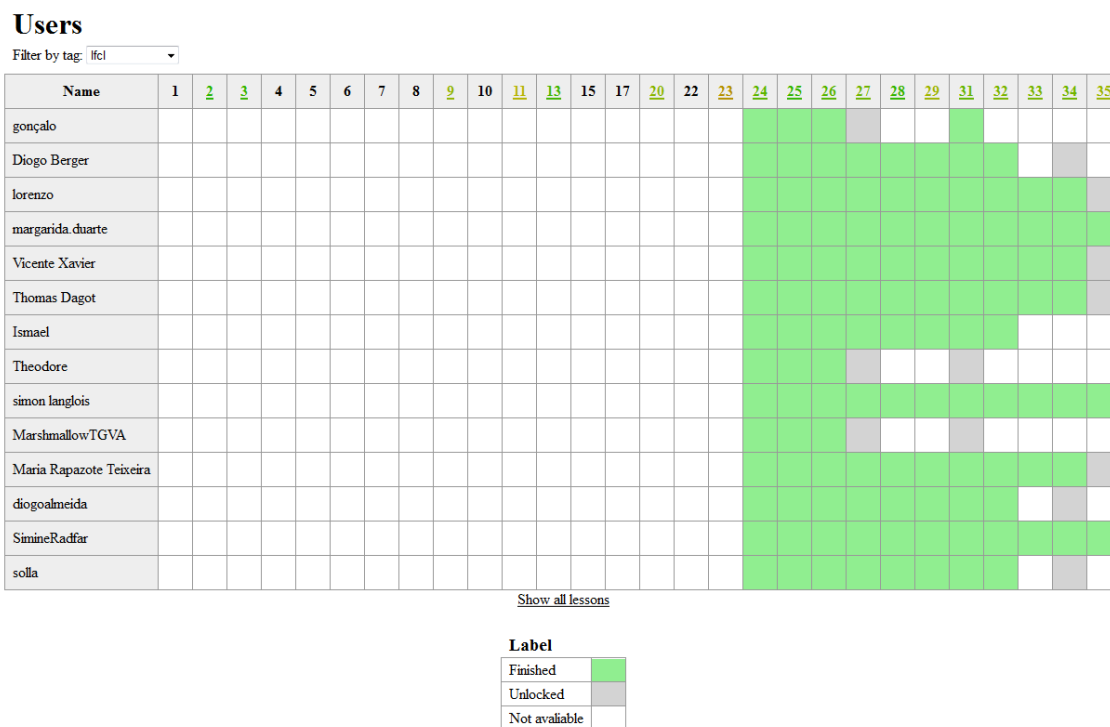


Figura 5.6: Secção de estatísticas de utilizadores no Editor

sendo a linguagem por defeito<sup>22</sup> ou a linguagem natural escolhida pelo utilizador (a coluna *natlang\_selected* na tabela *user*).

O Editor apresenta o conteúdo da base de dados de forma compatível com as múltiplas linguagens naturais, sendo possível escolher a linguagem natural com que consultar os dados e editar as traduções dessa linguagem, sendo possível também de gerir as linguagens naturais disponíveis, eliminando-as ou criando novas (Figura 5.7).

O comportamento na criação de uma nova linguagem natural é de copiar todas as traduções existentes de uma linguagem natural já introduzida no sistema à escolha. Ao criar um novo elemento que contém campos traduzíveis (tags, lição ou questão) a tradução é copiada para todas as linguagens naturais disponíveis.

Estes comportamentos dão-se ao facto de ser preferível mostrar traduções parciais do que mostrar campos vazios ou omitir questões/lições por falta de tradução.

Este sistema de traduções ainda é apenas uma mera base, havendo ainda o seguinte em falta para ter o sistema completamente multi-língua: tradução dos componentes estáticos (páginas html, editor e frontend), assistência à tradução no editor (marcar as traduções pré-definidas, percentagem de tradução completa) (Secção 7.1).





<sup>22</sup>A linguagem por defeito é a linguagem com *id* 1

```

88b      d88      ,ad8888ba, 8b      d8      ,ad8888ba,
888b      d888      d8"'"      "8b Y8,      ,8P      d8"'"      "8b
88`8b      d8'88      d8'      `8b `8b d8'      d8'      `8b
88 `8b      d8' 88      88      88 Y88P      88      88
88 `8b d8' 88      88      88 d88b      88      88
88 `8b d8' 88      Y8,      ,8P      ,8P      Y8,      Y8,      ,8P
88 `888' 88      Y8a.      .a8P      d8'      `8b      Y8a.      .a8P
88      `8' 88      `Y8888Y"' 8P      Y8      `Y8888Y"'
    
```

Index

## Traduções

ID	Linguagem	Actions
1	Português	 
2	English	 

[+New translation](#)

Figura 5.7: Secção de edição de linguagens naturais no Editor



# Capítulo 6

## Teste de Usabilidade

Para pôr o sistema sob teste num contexto de uso real foi efectuada uma sessão de testes junto do Liceu Francês Charles Lepierre com a inestimável ajuda da professora Christine Ladeira, envolvendo alunos voluntários com idades compreendidas entres os 12 e os 14 anos, matriculados no equivalente do 7º e 8º ano lectivo Português, o qual ocorreu ao longo de quatro dias.

Uma novidade inserida nestas sessões de testes foi o facto de ter sido utilizado o Python como a linguagem a ser ensinada ao grupo de crianças, quando o usual é usar-se linguagens de aprendizagem mais limitadas.

Este teste foi representativo do caso de utilização mais comum: público-alvo jovem e sem conhecimentos prévios de programação.

Foi concebido para testar vários aspectos do programa como por exemplo: se a sua funcionalidade estava a um nível de permitir a sua utilização, testar se o sistema suscitava algum interesse por parte dos alunos, e se era possível utilizar o mesmo como auxílio no ensino da linguagem Python a crianças.

### 6.1 Inquéritos

Para obter a percepção com que os alunos ficaram acerca do sistema, no fim da sessão de testes foi pedido aos alunos para preencherem um inquérito qualificando a sua experiência com o sistema.

O inquérito foi feito tendo como base a *System Usability Scale*, que consiste numa ferramenta que permitiu medir a usabilidade deste produto.

O resultado do inquérito foi o seguinte (com a coluna 1 correspondendo a não concordo nada e 5 com concordo muito):

Nº	Questão	1	2	3	4	5
1	Já sabia alguma coisa de programação antes?	4	7	2	0	0
2	Acha que programar é difícil?	0	4	4	4	1
3	Acha que programar é divertido?	0	1	2	6	4
4	Considera que aprendeu muito ou pouco sobre o Python?	0	0	4	2	7
5	Gostou de usar o Moxo?	0	0	2	2	9
6	Gostaria de ter acesso a mais lições para fazer em casa?	0	2	4	3	4
7	Eu voltaria a usar o Moxo.	1	1	3	6	2
8	Achei que a utilização era demasiado complexa.	0	7	3	2	1
9	Achei o Moxo fácil de usar.	1	1	2	6	3
10	Penso que preciso de um adulto para usar o Moxo.	3	4	4	2	0
11	Penso que os meus amigos usariam facilmente o Moxo.	0	2	7	3	1
12	Penso que ainda tenho que aprender muitas coisas antes de conseguir sozinho usar o Moxo.	1	6	5	0	1
Mostrando dois fragmentos de código equivalentes em Scratch e em Python:						
Nº		Sim	Não	Mais ou Menos		
13	Percebe a versão Scratch?	2	3	7		
14	Percebe a versão Python?	11	1	0		
Nº		Scratch	Python	Não há diferença		
15	A versão mais fácil de entender é:	1	10	1		
16	A versão mais bonita é:	3	7	2		
17	Se hoje fosse o primeiro dia de programação preferia usar:	3	6	3		

Com estes resultados conseguem-se distinguir dois grupos: um que conseguiu efectuar os testes com relativa facilidade e outro que encontrou mais dificuldades. Isto nota-se nos resultados do inquérito. Ao observar as questões onde não há consenso, existem opiniões divididas em dois polos distintos como por exemplo na questão número 4, na qual as opiniões se dividem entre as pontuações 3 e 5. Ainda assim houve um consenso generalizado acerca da facilidade e entretenimento do sistema.

Quanto à comparação entre uma possível alternativa gráfica (Scratch) os alunos mostraram uma preferência quanto ao Python, o que pode ter sido influenciado pelo facto do inquérito ter sido efectuada imediatamente após o final da sessão de testes, depois dos alunos já terem investido na linguagem Python.

## 6.2 Conclusões

Para cada dia foram preparadas entre duas a quatro lições, com os alunos a conseguir completar as lições de maneira atempada havendo sempre tempo em excesso para preencher com lições externas ao sistema, apresentando conceitos e exercícios no quadro. No primeiro dia a maioria conseguiu acabar as lições planeadas para os primeiros três dias.

Concluimos ainda através destes testes que não houve rejeição da linguagem Python por parte dos alunos ainda que a matéria inclua texto livre e identações, isto apesar da norma ser usar linguagens como o Scratch por este evitar os pontos mencionados.

Um potencial ponto fraco do sistema foi identificado, os alunos não interagem com as questões apresentadas, preferindo a memorização da resposta. As questões com mais complexidade pediam ao utilizador para experimentar o código no interpretador, o que os alunos só fizeram sob insistência. Para colmatar este problema passou-se a não voltar à questão anterior quando se errava, continuando-se entre as questões já desbloqueadas, reduzindo o número de perguntas repetidas e o grau aparente de dificuldade. Outro aspecto existente no sistema é a possibilidade de especificar duplicados de questão<sup>1</sup>, questões iguais entre si em que ao responder uma as outras são consideradas já respondidas, com isto é possível introduzir múltiplas variantes que diferem em pormenores que requerem ao utilizador prestar mais atenção à questão apresentada.

Também previsto para trabalho futuro é a implementação de dicas (Secção 7.3) que servem para estes casos em que os utilizadores necessitem de uma informação adicional para os ajudar a interpretar o enunciado, assim como um sistema de comentários onde os utilizadores têm a possibilidade de comentar entre si opiniões e dúvidas sobre as perguntas (Secção 7.13).

Outros elementos a que nos foi chamado a atenção foram algumas facetas da interface em que houve confusão, tal como quando o utilizador, quando lhe é mostrado a resposta correcta à sua tentativa anterior (Figura 6.1), este tenta erroneamente alterar a sua resposta nesse ecrã em vez de correctamente avançar para a próxima questão (Secção 7.11). Também a espera pela comunicação entre o frontend e o backend não está sinalizada por isso a página fica como que parada por uns segundos (Secção 7.14).

No entanto estas foram questões que os utilizadores superaram e que não tiveram impacto significativo na experiência.

Apesar da experiência geralmente positiva, os alunos ainda se mostraram dependentes de ajuda externa para interpretar as questões, o que mostra a dependência do sistema na qualidade das lições criadas pelos docentes. Para colmatar este problema existe previsto um sistema de comentários que permitirá os utilizadores deixarem questões ou interagirem com os outros utilizadores e os professores.

---

<sup>1</sup>Para este efeito basta especificar o tag e dificuldade igual ao das questões que lhe são equivalentes.

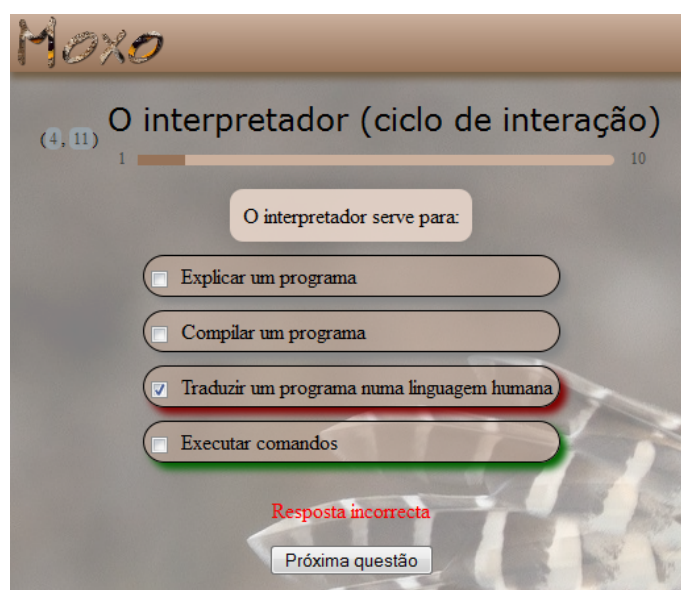


Figura 6.1: Exemplo de questão respondida errada com correcção apresentada

# Capítulo 7

## Trabalho Futuro

O sistema no seu estado actual apresenta-se usável com resultados positivos (Secção 6), mas devido a limitações de tempo e prioritização de certas características sobre outras levaram à possibilidade de os seguintes pontos serem passíveis de melhoria para uma solidificação do sistema.

### 7.1 Traduções

O sistema de múltiplas linguagens está implementado no editor, mas para ser utilizáveis é preciso no frontend ser possível especificar qual a linguagem natural de preferência e o backend utilizar essa linguagem na obtenção de material passível de ser traduzido (como lições e questões). Também é necessário permitir e efectuar a tradução dos elementos estáticos (as páginas web) no editor como no frontend.

Para facilitar o trabalho de quem traduz é possível modificar a base de dados para marcar as traduções que precisam de ser modificadas por terem sido copiadas de outra linguagem. Para este efeito adicionaria-se uma coluna *verified* na tabela *translation* com o valor padrão 0, servindo esta coluna para diferenciar as traduções já editadas das com valor copiado, e no código onde se altera o valor para uma tradução definir também o valor de *verified* para essa tradução para 1, para marcar como actualizada.

### 7.2 Pontuações

Como medida para fomentar a competitividade entre os utilizadores, assim como providenciar feedback sobre o progresso, foi prevista a possibilidade de serem atribuídas pontuações às lições completadas. Esta pontuação iria ter em conta o número de respostas erradas em relação ao número de respostas totais, tendo também em conta possíveis penalidades pela consulta de dicas.

### 7.3 Dicas

Para ajudar os utilizadores que se mostrassem com dificuldade em responder uma determinada questão, pôs-se a hipótese de criar um sistema de dicas que desse a opção ao docente de disponibilizar informação adicional sobre o problema possivelmente desvendando a solução, caso o utilizador não estivesse em modo avaliação e em troca de uma pontuação final mais reduzida.

### 7.4 Cache

Para a ordem das perguntas é preciso gerar uma árvore interna agrupando as questões por combinação de tag e dificuldade. Sendo esta operação efectuada quando se acede a uma lição e sendo mais dispendiosa do que um acesso, o cálculo da árvore só é efectuado da primeira vez e guardado numa cache que é utilizada para os restantes acessos. A limitação ocorrente desta implementação é que se houver uma mudança na estrutura das questões de uma lição a cache irá ficar desactualizada, a maneira actual de proceder é reiniciar o backend.

Para melhorar este ponto falta implementar um novo comando no backend para esvaziar a cache, e no editor fazer uso desse comando quando se efectua uma acção que invalide a cache (por exemplo na criação de uma nova pergunta).

### 7.5 Ordem das questões

Para seleccionar a próxima questão a apresentar ao utilizador são agregados todos os nós desbloqueados da árvore interna das questões da lição actual e é então escolhido uma questão de entre todas as questões desses nós.

Mas assim se houver um nó que tenha múltiplas questões, e como um nó é uma combinação de tag e dificuldade essas questões são então tratadas como equivalentes entre si, as questões desse nó têm maior probabilidade de ser escolhido do que todos os outros nós que só tenham uma questão associada, havendo então uma tendência no selector aleatório de questões para nós com maior número de questões.

Para resolver este problema basta escolher primeiro um nó aleatório de entre os disponíveis em vez de considerá-los todos ao mesmo tempo, restaurando assim o equilíbrio esperado entre os nós.

### 7.6 Tipos de questões

Apenas foram implementados dois tipos de questões pois o primeiro tipo (escolha múltipla) foi suficientemente genérico para representar qualquer pergunta necessária e porque a

criação do segundo tipo (ordenação de código) generalizou a implementação dos tipos de questões ao se usar campos JSON que permitem conter representações de qualquer tipo de valor em vez de estar fixo ao tipo específico de uma questão.

Deste modo, criar um novo tipo de questão é apenas fazer uso dos campos genéricos existentes na base de dados.

## 7.7 Navegadores

O principal método de interface com o sistema é via navegador web. A totalidade dos testes foram efectuados em Firefox, mas para haver um acesso generalizado do sistema será necessário haver testes com outros navegadores, especialmente porque o frontend faz uso de *compartilhamento de recursos cross-origem*, podendo a implementação não ser generalizada o suficiente para haver compatibilidade entre navegadores.

## 7.8 Melhorar distribuidor de carga

O frontend permite distribuir os utilizadores entre uma lista de backends. No entanto, este balanceamento demora algum tempo pois para cada utilizador que vai efectuar login espera por resposta de todos os backends da lista para ter a noção da carga entre si e escolher o ideal. Para colmatar esta lentidão o servidor é associado ao utilizador até que este faça logout ou saia do navegador, mas podia se melhorar mais fazendo só um pedido de informação aos backends para todos os utilizadores novos, guardando a informação de quantos utilizadores e para que servidor mandar, e recalculando caso esse número de utilizadores tenha sido enviado ou tenha passado certo tempo.

Também é preciso testar esta funcionalidade mais exaustivamente para verificar o seu funcionamento correcto.

## 7.9 Sincronização

Outro aspecto também a considerar é que se houver mais do que um backend activo tendo cada um a sua própria base de dados então é preciso haver maneira de sincronizar estas, pois é esperado que as bases de dados sejam iguais independentemente do backend.

Sem sincronização então os dados obtidos irão depender do backend que o distribuidor de carga do frontend escolheu para se ligar.

## 7.10 Segurança

Desde o princípio que a segurança foi de forma conscientemente posta em segundo plano para não interferir no desenvolvimento dos componentes mais relevantes aos objectivos.

Existe sistema interno de utilizadores, com possibilidade de criação de novos utilizadores, mas não existe opção de mudar a password da conta e a senha é guardada na base de dados em texto puro.

O acesso ao editor também foi livre tendo sido feitos backups frequentes à base de dados para reduzir o perigo de modificações maliciosas.

## 7.11 Progresso numa lição

Para dar ao utilizador feedback sobre o progresso numa sessão de questões existe sob o título da lição uma barra indicadora do número actual de tentativas e o número máximo, mas durante a sessão de testes verificou-se confusão sobre a interpretação dos valores, sendo necessário rever uma representação mais intuitiva do progresso.

## 7.12 Clarificar fim da lição

A lição quando acaba mostra apenas um pequeno texto que não chama a atenção indicando o sucesso da lição, é necessário tornar o final da lição mais visível, e também existe casos em que no fim de uma lição com sucesso a mensagem indica erroneamente que a lição falhou.

## 7.13 Comentários e feedback

Foi previsto a possibilidade de utilizadores poderem comentar nas questões, assim como lerem os comentários deixados por outros utilizadores nessa questão (só após terem respondido à questão, para evitar problemas de haver soluções postas nos comentários), isto possibilitaria interactividade entre os alunos e com o professor, possibilitando correcções a questões que necessitassem clarificação ou que tivessem erros. O tipo *comentário* já se encontra implementado na base de dados e no backend, falta implementar no frontend.

## 7.14 Indicação da comunicação entre frontend-backend

Actualmente não existe nenhuma indicação visual de que um pedido feito pelo utilizador está a ser processado, o utilizador tem que esperar para ver o resultado da interacção, o que causa os utilizadores a repetirem acções por assumirem que a acção não foi tomada.

Para evitar este problema bastaria cada vez que está a ocorrer comunicação entre o frontend e o backend colocar um *throbber*<sup>1</sup> na página onde o utilizador está.

---

<sup>1</sup>Nome do elemento gráfico que indica que está a ocorrer uma acção em plano de fundo, como por exemplo sob a forma de um círculo a girar.

# Capítulo 8

## Conclusão

Cremos que foram alcançados os objectivos a que nos propusemos no início deste trabalho na medida em que as experiências a que levámos a cabo *in loco* levaram-nos a concluir que o uso de ensino automatizado consegue ajudar um professor a fazer passar a matéria a um público mais abrangente, reduzindo a sua necessidade presencial e conseguindo-se concentrar nos alunos com mais dificuldades.

Concluimos portanto que o ensino está cada vez mais à disposição de toda a população e de todas as faixas etárias, tendo em atenção os rápidos desenvolvimentos nesta área levamos a crer que num futuro a curto/médio prazo poderemos constatar o aumento do ensino à distancia fazendo uso da tecnologia actual.



# Apêndice A

## Manual Instalação

### A.1 Pré-requisitos

O Moxo está dividido nos seguintes componentes:

- back-end* Servidor java que gere e obtém os dados na base de dados
- editor* Assistente à criação de questões e visualizador de dados
- front-end* Site onde os utilizadores interagem com o sistema

Os requisitos mínimos<sup>1</sup> de cada componente são:

- back-end
  - Java 1.6.0\_31
- front-end e editor
  - PHP 5.3.10 com os seguintes pacotes:
    - \* pdo\_mysql
    - \* pdo\_sqlite *Caso se pretenda usar bases de dados do tipo SQLite*
  - MySQL 5.5.37
  - Apache 2.2.22

#### A.1.1 Instalação dos pré-requisitos

##### Windows

Basta instalar o software Xampp, e durante a instalação certificar que as opções Apache e MySQL estão seleccionadas.

O Xampp agrupa os pacotes necessários. Para correr basta executar *xampp-control.exe*; e na secção *Actions* premir *Start* para o Apache e MySQL.

---

<sup>1</sup>Estas são as versões mais antigas em que o sistema funcionou, é possível mas não garantido que funcione noutras versões.

## Linux

Instalar os seguintes pacotes usando o gestor de pacotes que preferir:

- php5-common
- libapache2-mod-php5
- php5-cli
- php5-curl
- php5-mysql
- php5-sqlite
- mysql-server
- mysql-client

## A.2 Configuração

Os ficheiros de configuração estão em formato *INI*, sendo pares de definição / valores separados por um sinal de igual (=).

### A.2.1 Back-end

Em *moxo\_install/server/*, na mesma pasta do *jar* do servidor java está o ficheiro *serversettings.ini* com as configurações a serem utilizadas pelo servidor java. O nome de ficheiro pode ser um qualquer, será depois indicado ao servidor utilizando a opção da linha de comandos *-c*.

As definições possíveis de especificar são os seguintes. Os valores indicados são os aplicados no caso de essa definição não se encontrar no ficheiro de configuração indicado ao servidor ou caso não tenha sido especificado um ficheiro de configuração:

Definição	Tipo de valor	Valor por omissão	Descrição
SEED	Inteiro	Aleatório	A semente a ser utilizado pelo gerador de números aleatórios.
DEFAULT_NAT_LANG_ID	Inteiro	1	O número do id da linguagem natural a ser considerada como a por omissão.
DEFAULT_PORT	Inteiro	8080	O número do porto em que o servidor vai correr.
DB_JDBC_DRIVER	String	com.mysql.jdbc.Driver	Nome da class em Java contendo o driver para a base de dados em utilização. O outro único valor válido é "org.sqlite.JDBC" para SQLite.
DB_TYPE_STRING	String	mysql	O nome do tipo da base de dados a utilizar. O outro único valor válido é "sqlite".
DB_PATH	String	//127.0.0.1:3306/moxo	URI para a base de dados, incluindo ip, porto e nome da base de dados.
DB_AUTH	Booleano	true	Se é preciso enviar autenticação para aceder à base de dados. MySQL requer, mas não é preciso no caso do sqlite.
DB_USER	String	root	O nome de utilizador com o qual aceder à base de dados.
DB_PASS	String		A password com a qual aceder à base de dados.
DRAW_QUESTION_TREES	Booleano	false	O servidor tem a possibilidade de desenhar a representação das árvores internas das questões numa lição com a sua ordem, esta opção indica se é para produzir os ficheiros <i>.dot</i> contendo as representações.
QUESTION_TREES_DIR	String	null	A pasta para onde guardar os ficheiros <i>.dot</i> contendo as representações internas das lições. Guarda no directório corrente (.) se não for especificado.
LOG_ENABLED	Boolean	true	Se o log está activamente a ser produzido.
LOG_FILE_PATH	String	log.txt	Caminho para um ficheiro onde as mensagens de log irão ser guardadas.
LOG_FILE_LEVEL	Level	Level.ALL	O nível de detalhe guardado no ficheiro. Ver <code>java.util.logging.Level</code> para uma lista dos níveis disponíveis.
LOG_CMD_LEVEL	Level	Level.ALL	O nível de detalhe das mensagens apresentadas na linha de comandos.
DEFAULT_MODE	Inteiro	1	O modo das lições seleccionado por omissão. 1 para ensino, 2 para avaliação.
FORCE_MODE_1_TAG	Inteiro	1	O id da metatag que representa que um utilizador está forçado a fazer as lições em modo ensino.
FORCE_MODE_2_TAG	Inteiro	2	O id da metatag que representa que um utilizador está forçado a fazer as lições em modo avaliação.

## A.2.2 Editor

Em *moxo\_install/web/editor/settings.ini* estão as configurações do editor (os valores são do tipo String e têm de ser rodeados por aspas ("")):

Definição	Valor de exemplo	Descrição
db_type	"mysql"	O tipo de base de dados a que se vai aceder.
db_path	"127.0.0.1"	O caminho (uri) para a base de dados.
db_name	"moxo"	O nome da base de dados a aceder.
db_user	"root"	O nome a utilizar para o acesso à base de dados.
db_pass	""	A password a utilizar para o acesso à base de dados.

## A.2.3 Front-end

Em *moxo\_install/web/frontend/backend-servers.csv* há um ficheiro que contém uma lista de endereços de servidores java e valores de prioridade (quanto menor o valor maior é a prioridade de utilização, e um servidor com prioridade 1 é utilizado duas vezes mais que um com prioridade 2).

Exemplo de um ficheiro que indica que estão dois servidores a correr localmente nos portos 8080 e 8081, tendo o porto 8080 o dobro de prioridade:

```
http://127.0.0.1:8080, 1
http://127.0.0.1:8081, 2
```

## A.3 Instalação dos componentes do Moxo

Usando o arquivo *moxo\_install-<data>.tar* que contém a pasta *moxo\_install* com os ficheiros necessários à instalação do Moxo num sistema com os requisitos mínimos já satisfeitos:

- Copiar a pasta *moxo\_install/web/* para dentro da pasta de acesso público do Apache (ex.: `/var/www/html/`).
- Copiar a pasta *moxo\_install/server/* para a nova localização do servidor java.
- Inicializar a base de dados com o ficheiro *db\_structure.sql* que contém a estrutura base e os valores iniciais. Isto pode ser feito com o seguinte comando:

```
mysql --user=user --password=password nome_bd < db_structure.sql
```

## A.4 Execução

Correr primeiro o Apache (ex.: `/etc/init.d/apache2 start`) e a base de dados MySQL (ex.: `/usr/sbin/mysqld`).

Correr o back-end com o seguinte comando:

```
java -jar moxoserver_versão.jar
```

Existem também os seguintes parâmetros possíveis de especificar ao correr o back-end:

- c *FICHEIRO*      Especifica um ficheiro contendo as configurações a utilizar (mais informações na secção A.2.1).
- p *PORTO*        Especifica o porto em que correr o servidor, sobrepondo-se a qualquer valor definido no ficheiro de configuração.



# **Apêndice B**

## **Diagrama da Base de Dados**

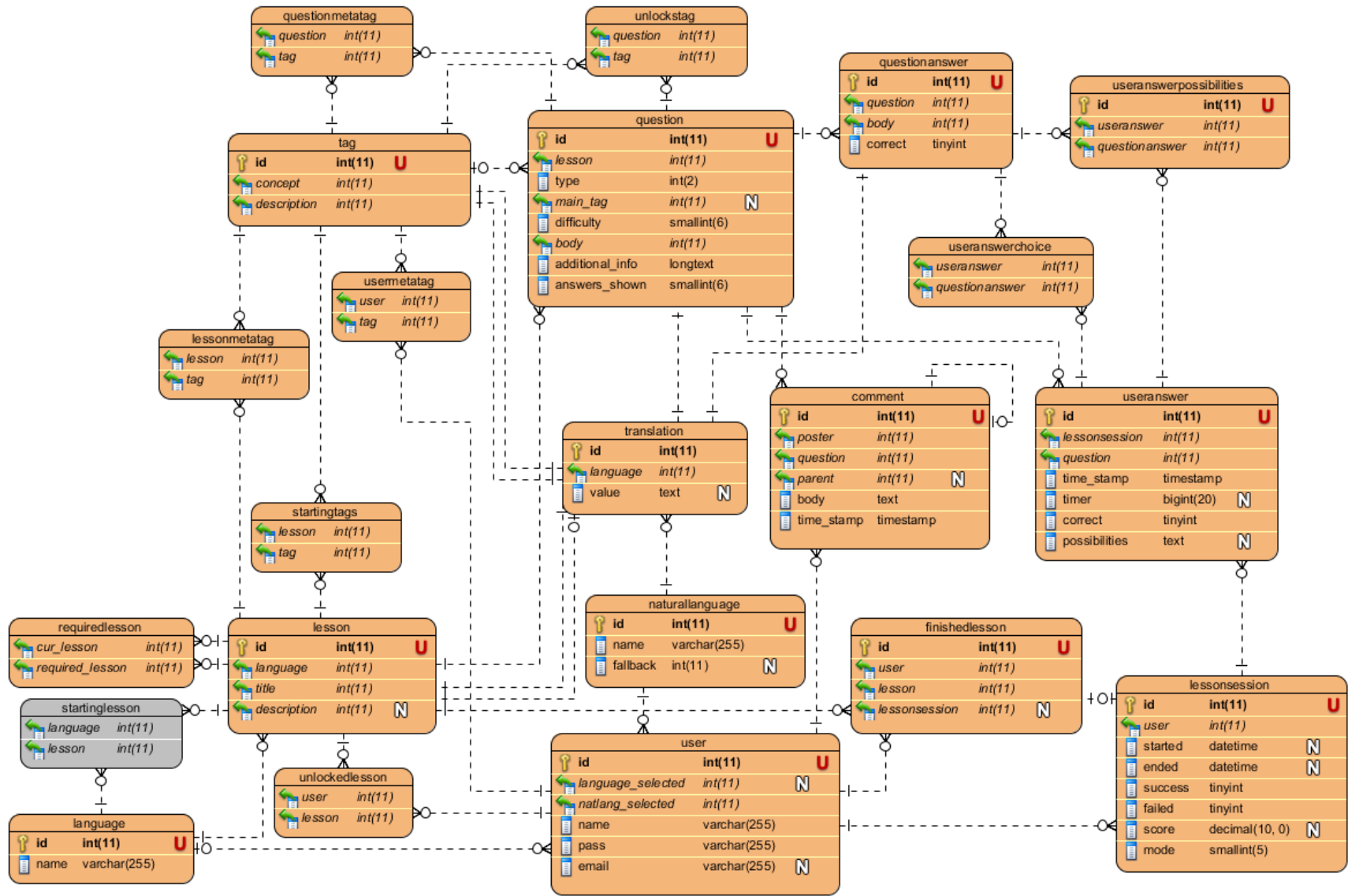
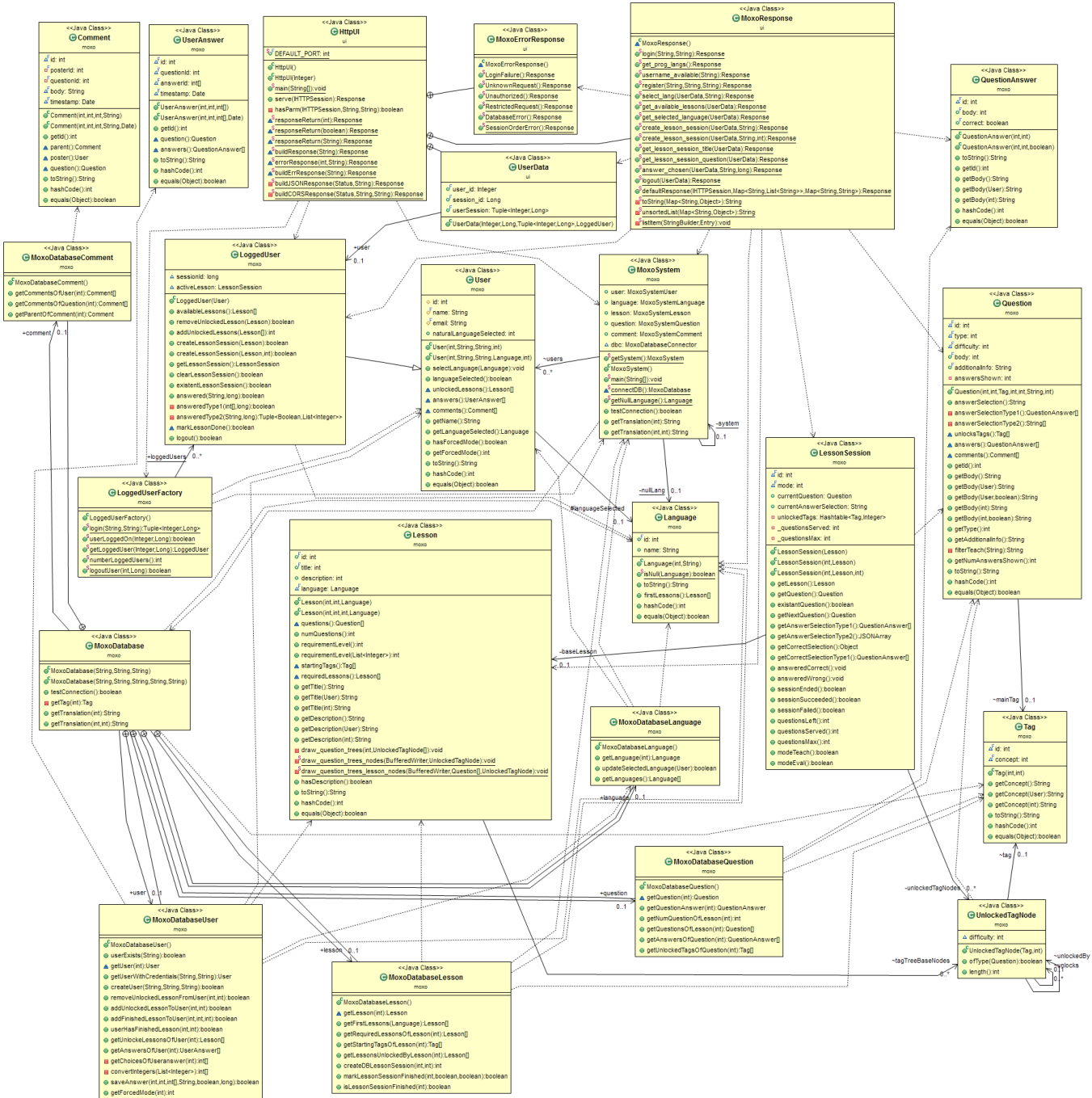


Figura B.1: Diagrama entidade-relação da base de dados

## **Apêndice C**

### **Diagrama de classes do backend**



# Apêndice D

## Lição “Mais Listas”

Seguem-se as três questões associadas à lição “Mais Listas”, com as respostas correctas assinaladas a verde.

### Mais listas !

Numa lista pode-se guardar valores não numericos. Por exemplo se queremos guardar o nome de uma pessoa, o seu numero de telefone e o seu endereço de email podemos escrever:

```
>>> contacto = ["Felisberto Caramulo", 967213456,
"felizberto.caramulo@gmail.com"]
```

A seguir podemos obter cada elemento da lista da seguinte forma:

Para imprimir o nome da pessoa:

```
>>> print contacto[0]
Felisberto Caramulo
```

Para imprimir o seu endereço de email:

```
>>> print contacto[2]
felizberto.caramulo@gmail.com
```

É portanto possível obter qualquer elemento de uma lista, basta colocar a seguir ao nome da variável a posição do elemento desejado entre [] e sabendo o primeiro está na posição 0.

Tendo a lista `a = ["benfica", 1, "sporting", 3]`, indique as expressões acompanhadas do resultado correcto. (Pode experimentar no idle antes de responder !)

`a[0]` resultado: "benfica"

`a[1]` resultado: 1

`a[3] + 2` resultado: 5

`a[1]` resultado: "benfica"

`a[3]` resultado: "sporting"

## Mais listas !

Tendo a listas `a = [22, 5, 1, 4, 0, 34.5, 0, 8]`, indique as expressões escritas correctamente e acompanhadas do resultado correcto:

`a[0] + a[2]` resultado: 23

`a{0} + a{2}` resultado: 23

`a[1] * a[3]` resultado: 20

`a[7] / a[3]` resultado: 2

`a[6] / a[3]` resultado: 2

## Mais listas !

Um elemento de uma lista pode ser uma lista. Por exemplo podemos guardar numa lista as notas de todos os alunos de uma turma:

```
>>> turma = [[12,13,14], [15,16,17], [3,5]]
```

Neste caso a turma tem 3 alunos o primeiro teve como notas [12, 13, 14] o segundo [15, 16, 17] e o terceiro [3, 5].

Qual é o valor de `turma[1]` ?

[15, 16, 17]

[12, 13, 14]

[3, 5]

# Bibliografia

- [1] Cal Flyn. Computer programming: who is teaching our children to code? - telegraph. <http://www.telegraph.co.uk/education/9600921/Computer-programming-who-is-teaching-our-children-to-code.html>, October 2012. [Online; accessed 10-October-2013].
- [2] Anant Agarwal. Online universities: it's time for teachers to join the revolution — education — the observer. <http://www.theguardian.com/education/2013/jun/15/university-education-online-mooc>, June 2013. [Online; accessed 10-October-2013].
- [3] Börje Holmberg. The evolution of the character and practice of distance education. *Open learning*, 10(2):47–53, 1995.
- [4] Wikipedia. E-learning — wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=E-learning&oldid=576314085>, 2013. [Online; accessed 10-October-2013].
- [5] Tony Karrer. Understanding e-learning 2.0. <http://www.astd.org/Publications/Newsletters/Learning-Circuits/Learning-Circuits-Archives/2007/07/Understanding-E-Learning-20>, July 2007. [Online; accessed 10-October-2013].
- [6] Stephen Downes. E-learning 2.0 stephen's web. <http://www.downes.ca/post/31741>, October 2005. [Online; accessed 10-October-2013].
- [7] Wikipedia. List of educational programming languages — wikipedia, the free encyclopedia. [http://en.wikipedia.org/w/index.php?title=List\\_of\\_educational\\_programming\\_languages&oldid=576568033](http://en.wikipedia.org/w/index.php?title=List_of_educational_programming_languages&oldid=576568033), 2013. [Online; accessed 11-October-2013].
- [8] Wikipedia. Pascal (programming language) — wikipedia, the free encyclopedia. [http://en.wikipedia.org/w/index.php?title=Pascal\\_\(programming\\_language\)&oldid=576285968](http://en.wikipedia.org/w/index.php?title=Pascal_(programming_language)&oldid=576285968), 2013. [Online; accessed 11-October-2013].

- [9] David H. Ahl. Basic computer games. <http://www.atariarchives.org/basicgames/>, 1978.
- [10] David H. Ahl. More basic computer games. <http://www.atariarchives.org/morebasicgames/>, 1979.
- [11] David H. Ahl. Big computer games. <http://www.atariarchives.org/bigcomputergames/>, 1984.
- [12] Wikipedia. Microsoft basic — wikipedia, the free encyclopedia, 2013. [Online; accessed 11-October-2013].
- [13] David D. Thornburg. Friends of the turtle: On logo and turtles. [Online; accessed 9-November-2013].
- [14] Steven Kiryakoza. The logo programming language, December 1997. [Online; accessed 9-November-2013].
- [15] Mama - an educational 3d programming language - eytam robotics. <http://www.eytam.com/mama>. [Online; accessed 11-October-2013].
- [16] Kidsruby.com. <http://www.kidsruby.com/about>. [Online; accessed 11-October-2013].
- [17] Create ios, android and flash games with stencyl. <http://www.stencyl.com/>. [Online; accessed 11-October-2013].
- [18] Robomind.net - welcome to robomind.net, the new way to learn programming. <http://www.robomind.net/pt/index.html>. [Online; accessed 11-October-2013].
- [19] Juliana Marques. A short history of moocs and distance learning. <http://mooconewsandreviews.com/a-short-history-of-moocs-and-distance-learning/>, April 2013. [Online; accessed 11-October-2013].
- [20] Dave Cormier. What is a mooc? <http://www.youtube.com/watch?v=eW3gMGqcZQc#!>, December 2010. [Online; accessed 11-October-2013].
- [21] Dave Cormier. The cck08 mooc – connectivism course, 1/4 way. <http://davecormier.com/edblog/2008/10/02/the-cck08-mooc-connectivism-course-14-way/>, October 2008. [Online; accessed 11-October-2013].

- [22] Jeffrey R. Young. Mooc provider edx more than doubles its university partners. <http://chronicle.com/blogs/wiredcampus/mooc-provider-edx-more-than-doubles-its-university-partners/43917>, May 2013. [Online; accessed 11-October-2013].
- [23] Willem van Valkenburg. Oeb12: 4 distinct factors of moocs. <http://www.e-learn.nl/2012/12/01/oeb12-4-distinct-factors-of-moocs>, December 2012. [Online; accessed 11-October-2013].
- [24] The New York Times. Welcome to the brave new world of moocs (massive open online courses). <http://www.youtube.com/watch?v=eW3gMGqcZQc#!>, January 2013. [Online; accessed 11-October-2013].
- [25] Laura Pappano. The year of the mooc. *The New York Times*, 2(12):2012, 2012.
- [26] Catherine Round. The best mooc provider: A review of coursera, udacity and edx. <http://www.skilledup.com/blog/the-best-mooc-provider-a-review-of-coursera-udacity-and-edx/>, February 2013. [Online; accessed 11-October-2013].
- [27] Coursera. Coursera help — how do peer assessments work? <http://help.coursera.org/customer/portal/articles/1163294-how-do-peer-assessments-work->. [Online; accessed 11-October-2013].
- [28] Scott Freeman and John W Parks. How accurate is peer grading? *CBE-Life Sciences Education*, 9(4):482–488, 2010.
- [29] Jenny Mackness, Sui Mak, and Roy Williams. The ideals and reality of participating in a mooc. In *Networked Learning Conference*, pages 266–275. University of Lancaster, 2010.
- [30] Ry Rivard. Measuring the mooc dropout rate. <http://www.insidehighered.com/news/2013/03/08/researchers-explore-who-taking-moocs-and-why-so-many-drop-out>, March 2013. [Online; accessed 11-October-2013].
- [31] Jon Fingas. San jose state university suspends udacity online course trials. <http://www.engadget.com/2013/07/19/san-jose-state-university-suspends-udacity-online-course-trials/>, July 2013. [Online; accessed 11-October-2013].

- [32] Coursera. Honor code — coursera. <https://www.coursera.org/about/honorcode>. [Online; accessed 11-October-2013].
- [33] Google. Google trends - interesse em pesquisa web do google: duolingo, rosetta stone - a nível mundial, 01/2011 - 09/2013. <http://www.google.com/trends/explore?q=duolingo#q=duolingo%2C%20rosetta%20stone&date=1%2F2011%2033m&cmpt=q>. [Online; accessed 9-October-2013].
- [34] Ignacio Garcia. Learning a language for free while translating the web. does duolingo work? *International Journal of English Linguistics*, 3(1):p19, 2013.