

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Uma Aplicação Social Utilizando Estilos de Aprendizagem

Luís Filipe Correia Gonçalves Varandas

VERSÃO DEFINITIVA

Relatório de Dissertação
Mestrado Integrado em Engenharia Informática e Computação

Orientador: Doutor Francisco José de Oliveira Restivo (Prof. Associado da FEUP)

Março de 2009

Uma Aplicação Social Utilizando Estilos de Aprendizagem

Luís Filipe Correia Gonçalves Varandas

Relatório de Dissertação
Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Doutor Luís Paulo Gonçalves dos Reis (Professor Auxiliar da FEUP)

Arguente: Doutor Feliz Alberto Ribeiro Gouveia (Professor Titular da UFP)

Vogal: Doutor Francisco José de Oliveira Restivo (Professor Associado da FEUP)

20 de Março de 2009

Resumo

A *Web 2.0* veio trazer uma nova dinâmica à Internet. De entre todos os mecanismos que potenciam este paradigma, as redes sociais merecem especial importância, quer pela sua utilização massiva, principalmente por parte dos internautas mais jovens, quer pelas potencialidades que encerra. Torna-se premente explorar esta rede no sentido de melhor se poderem vir a utilizar as suas potencialidades e por conseguinte, poder fornecer uma mais valia a todos os que dela se servem diariamente. A iniciativa OpenSocial, uma API baseada em *standards* abertos, surgida pelas mãos da Google e dos seus parceiros, veio potenciar este modelo e facilitar o trabalho dos programadores que podem desta forma ver as suas aplicações, com poucas ou nenhuma alteração, correr em qualquer plataforma que implemente a API.

Hoje em dia o processo de ensino/aprendizagem passa por uma grande transformação. É urgente uma mudança de métodos e de estratégias que aproximem o aluno à escola e a tornem mais interessante. Só desta forma se consegue cativar a atenção dos estudantes que hoje em dia encontram outras formas de conhecimento mais estimulantes fora do ambiente escolar. Se por um lado a estratégia passa por aliar a escola à tecnologia, traze-la para a sala de aula, envolve-la em todo o processo de ensino, por outro é também importante conhecer melhor os alunos, as suas aptidões, aferir sobre as suas preferências, conhecer a sua estrutura cerebral, a sua forma de pensar e principalmente, conhecer a forma como mais facilmente adquirem a informação. Neste sentido, existem diversos estudos já realizados, principalmente ao nível do ensino superior, onde se tenta compreender melhor os alunos de modo a que o professor operacionalize estratégias com vista à prossecução de um dos seus principais objectivos, o de otimizar o processo de ensino/aprendizagem.

Aliando as redes sociais aos estilos de aprendizagem, torna-se possível a aplicação de questionários à escala global, conseguindo-se desta forma obter uma amostra mais representativa, apesar de mais divergente em termos etários. Esta divergência justifica-se plenamente nos dias que correm pois o processo de ensino/aprendizagem deixou de fazer parte apenas da formação inicial de um indivíduo para se passar a processar ao longo da vida. No entanto, a aplicação de um questionário nestes termos só é viável se os utilizadores destas redes colaborarem, e para isso, torna-se necessário desenvolver estratégias para que as pessoas o realizem de forma espontânea e que o façam conhecer aos seus amigos.

Para além da utilidade de uma aplicação com estas características, a sua implementação numa ou em mais plataformas OpenSocial torna possível a aquisição de novos conhecimentos sobre os mecanismos e as potencialidades das tecnologias emergentes na área das redes sociais. Foi com este propósito desenvolvido um sistema que permite ministrar questionários sobre estilos de aprendizagem e guardar os resultados obtidos numa base de dados para posterior análise. Para tornar aliciante a utilização da aplicação e ao mesmo tempo tirar partido do ambiente onde foi integrada, desenvolveu-se um esquema de atribuição de cores de acordo com os resultados obtidos no questionário e que se pensa poder vir a contribuir para uma rápida propagação da aplicação, fornecendo ainda a vantagem de se tornar um mecanismo aceitável para encontrar pessoas com algumas afinidades.

Abstract

Web 2.0 has brought a new dynamics to the Internet. Amongst all the mechanisms that exploit this paradigm, the social networks deserve special attention, either by its massive use, mainly by younger people, either by the potentialities that it offers. It is therefore necessary to explore this network so that it is possible to use its potentialities and therefore, to provide one increased value to those who daily use it. The OpenSocial initiative, an API based on open *standards*, lunched by Google and its partners, came to improve this model and to facilitate the programmer's work who can, this way, see its applications, with few or none alterations, to run in any platform which implements this API.

Nowadays, the education/learning process is suffering a great transformation. It is urgent a change of methods and strategies that approach the student to the school in order to turn it into a more interesting place. Only this way it is possible to captivate student's attention, who nowadays find more stimulating ways of knowledge out of the school environment. If in one hand the strategy passes by allying the school to technology, bringing it into the schoolroom, involving it in all the teaching process, on the other hand it is also important to know the students better, their aptitudes, to assess on their preferences, to know their cerebral structure, their way of thinking and mainly, to know the way as more easily they acquire the information. In this way, there are already a lot of studies, mainly at superior education level, trying to better understanding the students, leading teachers to develop strategies to achieve one of its main goals, improving the teaching/learning process.

By allying the social networks to learning styles, it becomes possible the application of questionnaires at a global scale, obtaining, this way, a more representative sample, although more divergent in age scale. This divergence is totally justified nowadays, once the teaching/learning process is no longer a part of only the initial formation of an individual, occurring throughout all their life. However, the application of a questionnaire in these terms is only viable if the users of these networks collaborate, and for that it becomes necessary to develop strategies so that people can do it in a spontaneous way and that they can share it with their friends.

Beyond the utility of an application with these characteristics, its implementation in one or more OpenSocial platforms makes possible the acquisition of new knowledge about the mechanisms and the potentialities of the emergent technologies in the area of the social networks. It was with this intention, developed a system that allows to apply questionnaires on learning styles and to keep the results obtained in a database to later analysis. In order to make the use of the application enjoyable and at the same time to take part of the environment where it was integrated, was developed a scheme of colours according to the results gotten in the questionnaire and that it is thought to be able to increase the propagation of the application, supplying still the advantage of becoming an acceptable mechanism to find people with some affinities.

Agradecimentos

Deixo aqui uma palavra de apreço ao professor Francisco Restivo, pela sua disponibilidade empenhamento e incentivo ao longo de toda esta cruzada.

Queria também agradecer à minha família que sempre me apoiou e que tornou possível a conclusão deste trabalho.

Luís Varandas

Índice

1	Introdução.....	1
1.1	Contexto/Enquadramento.....	2
1.2	Motivação e Objectivos.....	2
1.3	Estrutura da Dissertação.....	3
2	Enquadramento Tecnológico.....	4
2.1	Introdução.....	4
2.2	Google Gadgets.....	5
2.3	OpenSocial.....	5
2.3.1	Social Gadgets.....	6
2.4	Google Gadget Editor.....	6
2.5	Google Data APIs.....	7
2.6	Google Charts API.....	7
2.7	Google Desktop API.....	7
2.8	Google App Engine.....	7
2.8.1	Linguagens utilizadas.....	7
2.8.2	Persistência de dados.....	8
2.8.3	Autenticação.....	8
2.8.4	Serviços disponibilizados.....	8
2.8.5	Ambiente de desenvolvimento.....	8
2.8.6	Ambiente de produção.....	9
2.8.7	Limitações.....	9
2.8.8	Consola de Administração.....	9
2.8.9	Acesso e utilização da Google App Engine.....	9
2.9	As Redes sociais.....	10
2.10	Resumo e Conclusões.....	11
3	Estilos de aprendizagem.....	13
3.1	Análise de questionários.....	14
3.2	Resumo e Conclusões.....	15
4	Implementação.....	18
4.1	Contexto.....	18
4.2	Características dos utilizadores.....	18
4.3	Requisitos funcionais.....	19
4.3.1	Aplicação social.....	19
4.3.2	Administração.....	19
4.3.3	Requisitos não funcionais.....	20
4.4	Visão geral.....	20
4.5	Funcionamento da aplicação social.....	21
4.6	Arquitectura.....	21
4.7	Tecnologias.....	24

5	Detalhes e Resultados da Implementação do Site de Administração.....	26
5.1	O SDK Google App Engine.....	26
5.2	Configuração da aplicação.....	26
5.3	O <i>framework</i> Webapp.....	28
5.4	Controlo de acessos.....	29
5.5	Satisfação dos pedidos externos.....	30
5.6	Persistência de dados.....	31
5.7	Comunicação com a aplicação social.....	31
5.8	<i>Layout</i> do site de administração.....	32
5.9	Funcionalidade disponibilizada.....	33
5.9.1	Importação de um questionário.....	33
5.9.2	Visualização de um questionário previamente gravado.....	34
5.9.3	Seleção do questionário a disponibilizar.....	35
5.9.4	Estatísticas.....	35
5.10	<i>Update</i> da aplicação.....	36
5.11	Resumo e Conclusões.....	37
6	Detalhes e Resultados da Implementação da Aplicação Social.....	38
6.1	Ferramentas e processo de desenvolvimento.....	38
6.2	Estrutura de uma aplicação social no Hi5.....	39
6.3	Alojamento e testes.....	40
6.4	Funcionamento da aplicação.....	40
6.5	Acesso a dados externos.....	44
6.6	Persistência de dados do OpenSocial.....	45
6.7	As <i>views</i> do Hi5.....	46
6.8	Propagação da aplicação.....	47
6.9	Resumo e Conclusões.....	47
7	Conclusões e Trabalho Futuro.....	48
7.1	Satisfação dos Objectivos.....	48
7.2	Trabalho Futuro.....	50
	Referências.....	52

Lista de Figuras

Figura 4.1: Visão Geral.....	20
Figura 4.2: Funcionamento da aplicação social.....	21
Figura 4.3: Arquitectura lógica.....	22
Figura 4.4: Arquitectura física.....	23
Figura 5.1: Simulação local da autenticação.....	29
Figura 5.2: Google Accounts.....	30
Figura 5.3: <i>Layout</i> do site de administração.....	32
Figura 5.4: Importação de um questionário.....	33
Figura 5.5: Visualização de um questionário.....	34
Figura 5.6: Selecção do questionário activo.....	35
Figura 5.7: Opção para visualização de estatísticas.....	36
Figura 6.1: Página de entrada da aplicação social.....	41
Figura 6.2: Página com informação das cores atribuídas.....	41
Figura 6.3: Página para resposta ao questionário.....	42
Figura 6.4: Página de resultados do teste.....	43
Figura 6.5: Visualização da cor atribuída de acordo com os resultados do teste.....	44
Figura 6.6: Vista da página Preview.....	47

Lista de Tabelas

Tabela 3.1: Questionários para avaliação dos estilos de aprendizagem.....	16
Tabela 3.2: Alguns questionários disponíveis na Internet.....	17

Abreviaturas e Símbolos

API	Application Programming Interface
CGI	Common Gateway Interface
DHTML	Dynamic HyperText Markup Language
DTD	Document Type Declaration
FEUP	Faculdade de Engenharia da Universidade do Porto
GGE	Google Gadget Editor
GQL	Google Query Language
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IIS	Internet Information Services
JSON	JavaScript Object Notation
REST	Representational State Transfer
RPC	Remote Procedure Calls
RSS	Really Simple Syndication
SDK	Software Development Kit
SMS	Short Message Service
SQL	Structured Query Language
URL	Uniform Resource Locator
VARK	Visual, Aural, Read/write and Kinaesthetic
WWW	World Wide Web
XML	eXtensible Markup Language
YAML	YAML Ain't Markup Language

1 Introdução

A *Internet* tem vindo a adquirir uma importância crescente na vida de todos os cidadãos. A velocidade de acesso cada vez maior e o número de ofertas por parte de diversos operadores a custos cada vez mais reduzidos, potenciaram a sua utilização, sendo hoje em dia não só um meio imprescindível para o desenvolvimento profissional e pessoal de qualquer cidadão, mas também como potenciador de um relacionamento social, sem nunca esquecer o seu carácter lúdico. As gerações mais novas conviveram desde sempre com estas novas tecnologias e sentem-se completamente à vontade quando lidam com elas. Fica até a sensação de que é algo que lhes é inato. Para além disso, a maior parte dos jovens sentem-se motivados e são muito curiosos sobre todas as novidades que vão surgindo neste meio.

A *World Wide Web* veio disponibilizar um enorme manancial de informação a qualquer cidadão com acesso à *Internet*. São cada vez mais, e mais completos os serviços disponibilizados nesta rede, o que muitas vezes se traduz numa melhoria assinalável da qualidade de vida, pois vieram agilizar muitos processos que tradicionalmente implicavam custos tanto financeiros como de tempo. Veja-se por exemplo a entrega de declarações ao estado ou o pagamento de serviços.

A *Web 2.0* revolucionou a forma como a *Internet* era encarada e proporcionou a dinamização da sua utilização a uma franja cada vez mais alargada de internautas. O “hiper-espaço” foi povoado por *wikis*, *blogs*, sites para partilha de vídeo e redes sociais. Ao utilizador foi dada a possibilidade de deixar de ser um mero espectador podendo passar a ter um papel mais activo. Cada um pode actualmente dar o seu contributo, tornar as suas ideias visíveis ao mundo, fazer valer o seu ponto de vista, mas, ao mesmo tempo, sujeitar-se à crítica e à supervisão de todos. A *Web* tornou-se um local propício à partilha generalizada, tanto de materiais como de experiências.

Dentro deste novo formato de *Web*, talvez uma das áreas que mais tem vindo a crescer seja a das redes sociais, que constituem um enorme manancial de informação que poderá alimentar a *Web 3.0*. Estas redes encontram os seus adeptos, maioritariamente entre o público mais jovem. Estes ocupam actualmente grande parte do seu tempo livre em frente ao computador, sendo a maior parte gasta a navegar na *Internet*.

Existe por isso necessidade de investigar processos que tirem partido desta nova tendência, de modo a proporcionarem uma mais valia na formação dos jovens. O tipo de ensino que se praticava a algumas décadas atrás não é praticável nos tempos que correm. Existe uma enorme necessidade de alterar os processos de ensino e de encontrar estratégias para motivar os jovens nas salas de aula. Não se pode travar uma batalha contra as novas tecnologias e as novas tendências, antes pelo contrário, existe todo o interesse em nos juntarmos a elas e aproveitá-las

da melhor forma para que o seu carácter lúdico e social se transforme também numa mais valia para todos, alunos e professores.

1.1 Contexto/Enquadramento

Surgiram recentemente uma série de novos termos e conceitos relacionados com as redes sociais de que são exemplo os *gadgets*, os *widgets* e o OpenSocial¹. O OpenSocial representa um novo conceito no desenvolvimento de aplicações a integrar em redes sociais. A sua vantagem reside no facto de ser possível a integração do mesmo programa em diversas plataformas, sem haver necessidade de reescrever código. Neste contexto surgiu a proposta da Faculdade de Engenharia da Universidade do Porto que, no âmbito de uma dissertação do Mestrado Integrado em Engenharia Informática e Computação, veio propor a construção de uma aplicação social que utilizando o OpenSocial e as tecnologias mais recentemente disponibilizadas, afira sobre as preferências quanto aos estilos de aprendizagem dos utilizadores deste tipo de redes.

1.2 Motivação e Objectivos

Os dias em que o professor expunha a matéria a um conjunto de alunos que se mantinham atentos e a absorviam, uns melhor que outros, na medida das possibilidades de cada um, pertencem ao passado. É de todo o interesse alterar os processos de ensino/aprendizagem de modo a que estes vão de encontro às necessidades e expectativas dos alunos. Para isso é necessário tentar perceber como estes melhor adquirem os conhecimentos e quais as suas preferências relativamente à forma como esses conhecimentos são adquiridos.

Por outro lado, é fundamental descobrir qual a melhor forma de tirar partido das tecnologias emergentes, tentando perceber o modo como estas se poderão tornar uma mais valia no processo de Ensino/Aprendizagem. Esta dissertação tem como primeiro objectivo a exploração dessas tecnologias, explorando o modo do seu funcionamento e as suas potencialidades e tentar perceber a sua dinâmica. Neste sentido, o desafio prende-se com o desenvolvimento de uma aplicação que seja instalada numa rede social, que explore as principais características desta rede e que ao mesmo tempo recolha informação, utilizando um questionário sobre estilos de aprendizagem com vista a melhor adequar o processo de Ensino/Aprendizagem a um novo público que apesar de ser muito heterogéneo possui muitos pontos de confluência.

A aplicação a desenvolver deverá possibilitar a utilização de diversos questionários que poderão ser desenvolvidos e carregados para depois serem disponibilizados *on-line*. Os resultados desses questionários serão recolhidos numa base de dados e utilizados posteriormente para análises e estatísticas.

Para que o questionário seja respondido por um número considerável de internautas terá que ser desenvolvido um qualquer artifício que o torne apelativo o suficiente para que um utilizador após o ter concluído o dê a conhecer aos seus amigos e desta forma a aplicação seja instalada por um número crescente de utilizadores. O cumprimento desse objectivo poderá passar pela transformação da aplicação numa espécie de jogo social de modo a que quem o jogue tenha interesse em que um seu amigo também o venha a jogar. Pensa-se que só desta forma se obterá um grande número de utilizadores e consequentemente uma amostra suficientemente alargada de modo a que se possam tirar conclusões da aplicação do questionário.

Na realização deste trabalho tornou-se fundamental uma abordagem aos Estilos de Aprendizagem, primeiramente para a aquisição de uma percepção abrangente daquilo em que

1 <http://www.opensocial.org/>

Introdução

consistem e posteriormente para proceder à escolha de um questionário para aferição desses estilos e que será utilizado pela aplicação a desenvolver.

As tecnologias envolvidas na criação de redes sociais deverão ser analisadas e realizados testes e experiências no sentido de poderem vir a ser tomadas decisões conscientes e fundamentadas na implementação do produto final.

1.3 Estrutura da Dissertação

No capítulo 2 será efectuada uma exploração das tecnologias actualmente disponíveis para a criação de aplicações sociais no sentido de aferir o seu estado actual, de perceber as suas diferenças, vantagens e desvantagens de modo a que se adopte um ou vários desses processos e tecnologias na prossecução do objectivo final.

Segue-se um capítulo onde será feita uma pequena abordagem à temática dos estilos de aprendizagem, para se perceber em que consistem e a forma como poderão ser utilizados.

O capítulo 4 é dedicado à implementação da aplicação onde se descrevem os requisitos e a sua arquitectura.

Nos capítulos 5 e 6 são apresentados alguns detalhes e resultados da implementação do sistema desenvolvido.

Por fim, no capítulo 7 é feito um balanço final, enunciadas possíveis melhorias e dadas algumas ideias relativamente a investigações e desenvolvimentos futuros.

2 Enquadramento Tecnológico

O OpenSocial consiste num *standard* aberto que permite desenvolver aplicações de cariz social para um número crescente de sites que o suportam. O seu aparecimento ocorreu em Novembro de 2007. Este *standard* veio possibilitar a criação de aplicações que podem correr em diversas redes sociais como o MySpace, Plaxo, Orkut, iGoogle, Hi5 entre muitos outros[Ope09], sem necessidade de as reescrever, o que facilita muito a vida dos programadores que pretendam instalar as suas aplicações em mais do que uma rede social.

A API OpenSocial² encontra-se actualmente na versão 0.8.1. Esta versão consiste numa revisão efectuada em Agosto de 2008 da versão 0.7 e veio disponibilizar novos modelos de programação adicionando protocolos *server-to-server* (REST e RPC). A futura versão 0.9 promete uma maior simplicidade e uma maior velocidade. O OpenSocial foi uma iniciativa da Google, que contou com a parceria de outras empresas dedicadas à vertente social da *Web*. Para que se perceba melhor o que é o OpenSocial e o contexto em que se encontra inserido, será feita neste capítulo uma breve abordagem às principais tecnologias e ferramentas disponibilizadas maioritariamente pela Google, algumas das quais relacionadas com o OpenSocial outras que poderão tornar-se úteis tanto na implementação da própria aplicação social como no seu *backoffice*.

2.1 Introdução

Para se perceber melhor o que é descrito a seguir é necessário clarificar dois conceitos importantes. O conceito de *container* e o conceito de *sandbox*. Neste contexto, um Gadget Container será um site que suporta a integração de *gadgets* e um OpenSocial Container será a designação atribuída a um site onde seja possível integrar e executar aplicações que recorram à API OpenSocial. O termo *container* poderá ser utilizado indistintamente do seu tipo, de acordo com o contexto em que se encontre inserido.

Uma *sandbox* representa um ambiente protegido, onde podem ser testadas aplicações em condições semelhantes àquelas encontradas no ambiente final onde serão integradas.

2 <http://code.google.com/apis/opensocial/>

2.2 Google Gadgets

Os Google Gadgets³ são mini-aplicações escritas em HTML e JavaScript, Flash ou Silverlight inseridas num bloco de XML[Goo09a]. Podem correr em sites e plataformas da família Google, como no iGoogle, Google Maps, Google Toolbar ou Orkut, em produtos e sites de terceiros, ou ainda em qualquer página *Web*.

Qualquer utilizador pode a qualquer momento instalar no seu ambiente um *gadget* a partir de uma lista de *gadgets* disponibilizados, configurá-lo ou remove-lo caso assim o pretenda.

Um *gadget* pode ser visualizado a partir de várias localizações dentro da aplicação que o acolhe, sendo por isso conveniente possuir alguma flexibilidade que permita o seu redimensionamento, tornando possível a sua utilização em qualquer uma das vistas disponíveis.

O programador pode considerar a criação de mais que uma *view* do mesmo *gadget* de modo a que este tire maior partido do local onde será mostrado. Neste sentido, podemos, por exemplo, construir uma *Home View*, que será apresentada na página principal e uma *Canvas View*, que proporciona um maior espaço permitindo assim fazer um aproveitamento mais otimizado de todas as suas funcionalidades[Goo09b].

Um *gadget* pode possuir uma componente social, por intermédio da utilização da API OpenSocial, que lhe permite interagir com outros utilizadores que figurem na lista de contactos do utilizador que instalou o *gadget*. Este tipo de *gadget* pode ser designado de Social Gadget.

Para o desenvolvimento deste tipo de aplicações a Google disponibiliza uma ferramenta on-line, designada de GGE⁴ (Google Gadget Editor).

Existem duas APIs disponíveis que podem ser utilizadas no desenvolvimento de *gadgets*, a Legacy Gadgets API e a *gadgets.** API. Antes de iniciar a construção de um *gadget* é necessário, em primeiro lugar, decidir qual das duas APIs se irá utilizar, o que vai depender do local onde se pretende disponibilizar esse mesmo *gadget*. A segunda API é sempre preferível à Legacy Gadgets API, pois representa uma evolução desta e oferece um conjunto de funcionalidades adicionais[Goo09c]. No entanto, nem todos os ambientes que suportam *gadgets* são compatíveis com esta nova API.

2.3 OpenSocial

O OpenSocial é um conjunto de APIs construídas com base em *standards* abertos com o objectivo de fornecer uma tecnologia não proprietária para a construção de aplicações sociais, baseadas nos Google Gadgets. Esta iniciativa, foi lançada em Novembro de 2007 no Google Campfire One⁵ pela Google, que se juntou a outros parceiros que operam na área das redes sociais, como o MySpace, Friendster, Hi5 entre outros, com o intuito de fazer frente ao domínio do Facebook como plataforma social. Apareceram assim os Social Gadgets que podem ser disponibilizados em *containers* que suportem a API OpenSocial.

Desta forma, usando *standards* abertos e amplamente utilizados como o HTML, JavaScript e XML, podem ser construídas aplicações sociais que consigam correr em diversos OpenSocial *containers*[Roo08a], sendo para isso necessário realizar poucas ou nenhuma alteração, o que evita a aprendizagem de diversas linguagens e tecnologias específicas se pretendermos desenvolver uma aplicação para diversas plataformas.

O número destes *containers* cresce cada vez mais por entre os sites sociais existentes na *Web*. A lista completa de *containers* OpenSocial pode ser encontrada na OpenSocial Community Wiki⁶ mantida pela Google.

Muitos destes *containers* disponibilizam ao programador *sandboxes* para que a tarefa de desenvolvimento das aplicações que recorrem a esta API, se faça de uma forma mais facilitada.

3 <http://code.google.com/apis/gadgets/>

4 <http://code.google.com/apis/gadgets/docs/legacy/gg.html#GGE>

5 <http://code.google.com/campfire/>

6 <http://wiki.opensocial.org/index.php>

2.3.1 Social Gadgets

Um Social Gadget é um *gadget* que utiliza a API OpenSocial, possibilitando desta forma o acesso à informação dos utilizadores do *container* em que se encontra instalado e proporcionando a interação entre utilizadores desse mesmo *gadget*.

Para se transformar um Google Gadget num OpenSocial Gadget, apenas é necessário adicionar uma nova tag: `<Require feature="opensocial-0.8" />`

É possível, e de uma forma bastante simples, testar um Social Gadget recorrendo à rede social Orkut e ao GGE. Basta para isso seguir os seguintes passos:

1. Gravar o *gadget* utilizando o GGE. O *gadget* ao ser gravado no GGE ficará disponível na *Web*, a partir de um URL externo.
2. Criar uma conta na rede social Orkut ou utilizar uma conta existente;
3. Efectuar um pedido para aceder à *sandbox* do Orkut⁷;
4. Depois de ser concedido o acesso à *sandbox*, adicionar o *gadget* anteriormente criado, bastando para isso entrar na área de edição de aplicações e indicar o URL onde ele se encontra alojado.

As alterações que forem efectuadas no GGE, após serem gravadas, ficarão disponíveis a partir de um URL externo, o que permite o seu acesso pela *sandbox* do Orkut, podendo desta forma serem testadas.

Saliente-se no entanto que deverá haver algum cuidado relativamente aos mecanismos de cache implementados que podem dificultar o processo de *debug* de uma aplicação, pois as alterações efectuadas poderão não ficar de imediato disponíveis, havendo por isso que ter alguns cuidados.

2.4 Google Gadget Editor

O GGE é uma ferramenta *Web* disponibilizada pela Google e constitui o único editor de *gadgets* que permite de uma forma relativamente fácil, proceder à edição e teste deste tipo de aplicações.

É possível gravar os *gadgets* alojando-os no ambiente do GGE, sendo esta uma condição necessária para publicar uma aplicação a partir desta ferramenta. É então possível efectuar a publicação do *gadget* desenvolvido, de modo a que este apareça na página pessoal do iGoogle e aí testá-lo convenientemente. Depois do código se encontrar bem testado e caso se queira disponibilizar o *gadget* a um maior número de utilizadores, existe a possibilidade de o publicar na directoria do iGoogle, não sendo este um passo necessário para quem pretenda apenas utilizar e partilhar o *gadget* com a sua lista de amigos. Existe ainda a possibilidade de publicar o *gadget* numa qualquer página *Web*. Para isso e após escolher a opção correcta a partir do menu do GGE e de proceder a algumas configurações, basta copiar o código gerado e colá-lo na página à qual se destina.

Podemos testar um *gadget* utilizando apenas a opção *preview* do editor. Nesta vista é possível observar o comportamento da aplicação.

Apesar de ser uma boa ferramenta para testar *gadgets*, este editor não suporta a API *gadgets.**, nem os Social Gadgets. Isto quer dizer que não é possível ver o comportamento de um *gadget* social por intermédio da opção de *preview*, incluída no editor. Para testarmos a aplicação será necessário importá-la para um *container* que suporte o OpenSocial, como é o caso do Orkut.

7 <http://sandbox.orkut.com/SandboxSignup.aspx>

2.5 Google Data APIs

As Google Data APIs⁸ permitem a uma aplicação OpenSocial, assim como a uma qualquer aplicação cliente interagir com diversos serviços disponibilizados pela Google[Bla08]. Estas APIs utilizam um protocolo baseado no Atom Publishing Protocol (AtomPub), que usa o *standard* Atom Syndication Format para representar os dados e HTML para proceder à comunicação. O protocolo Google Data API estende o AtomPub para processar *queries*, para processos de autenticação, pedidos *batch* e fornecer formatos de output alternativos (JSON, RSS)[Goo09d].

Muitos serviços da Google suportam o protocolo Google Data API, o que permite que a aplicação cliente interaja com esses serviços como é o caso de serviços relacionados com Bases de dados, *blogs*, calendários, contactos, entre muitos outros[Goo09d]. No entanto em muitos destes casos o acesso a estas funcionalidades está dependente de uma autenticação.

2.6 Google Charts API

Utilizando a Google Charts API⁹, é possível transformar dados estruturados em gráficos, de modo a apresentar esses mesmos dados de uma forma mais clara e intuitiva. O pedido é efectuado por intermédio de um URL que é chamado com os parâmetros necessários à construção do gráfico. É retornada uma imagem com o formato .png que será mostrada na página do cliente.

2.7 Google Desktop API

Esta API¹⁰ permite a criação de um tipo de *gadgets* que corre tanto no Desktop como na *home page* do iGoogle. Este tipo de *gadgets* são comumente chamados de Google Desktop Gadgets. A criação destes *gadgets* pode ser feita com a ajuda do Google Desktop SDK¹¹, uma ferramenta disponibilizada pela Google.

2.8 Google App Engine

A plataforma Google App Engine¹² foi lançada pela Google em Abril de 2008 no Google Campfire One, e proporciona aos programadores a possibilidade de alojarem os seus sites na infra-estrutura do Google, sem se terem que preocupar com a manutenção de servidores *Web* ou de Bases de Dados. Após a criação de uma conta na Google App Engine, cada programador passa a dispor de 500MB de armazenamento, 5 milhões de *page views* por mês e a possibilidade de registar até 10 aplicações. A partir de Fevereiro de 2009, passou a ser possível a compra de recursos adicionais.

2.8.1 Linguagens utilizadas

O Python¹³ é para já a única linguagem disponível para os utilizadores da plataforma, sendo a versão actualmente suportada a 2.5.2. No entanto está previsto um alargamento a outras

8 <http://code.google.com/apis/gdata/>

9 <http://code.google.com/apis/chart/>

10 <http://code.google.com/apis/desktop/>

11 <http://desktop.google.com/downloadsdksubmit>

12 <http://code.google.com/appengine/>

13 <http://www.python.org/>

linguagens. A plataforma inclui a livreria *standard* do Python com algumas limitações relacionadas com o ambiente, como é o caso dos métodos de leitura e escrita de ficheiros que não são suportados. Pode ser feito o *update* de outras quaisquer livrerias desde que escritas em Python.

A App Engine integra um *framework Web*, designado de Webapp, de modo a facilitar o desenvolvimento de aplicações *Web* em Python. Suporta ainda outros *frameworks*, escritos em Python, como o Django¹⁴.

2.8.2 Persistência de dados

A persistência de dados é realizada com recurso a uma base de dados distribuída e escalonável, designada de Datastore. Esta base de dados baseia-se no BigTable[Cha06], um sistema de armazenamento distribuído de alta performance proprietário da Google que serve de suporte aos seus produtos como é o caso do Google Earth e do Google Finance.

A Datastore não suporta o modelo relacional, o que obriga a um desenho e manuseamento dos dados diferente daquele que é habitual encontrar nas bases de dados relacionais. Todos os objectos guardados na base de dados, ou seja, todas as entidades, possuem um nome e um conjunto de propriedades. Uma propriedade pode ser de um dos tipos de dados suportados ou ser uma referência para uma outra entidade, possibilitando desta forma a construção de relações de 1 para muitos e de muitos para muitos. A modelação dos dados é feita com recurso a classes Python sendo as propriedades descritas pelos atributos das classes. Cada entidade concreta é uma instância da classe que lhe serve de modelo. Para facilitar o manuseamento dos dados neste tipo de base de dados, é disponibilizada uma linguagem com uma sintaxe semelhante ao SQL, a GQL, mas bastante mais limitada. A GQL não suporta a clausula JOIN por questões de performance, limitando o retorno de dados a apenas uma tabela de cada vez. Permite apenas clausulas WHERE muito simples e existe um número máximo de 1000 linhas que podem ser devolvidas por cada consulta à base de dados.

2.8.3 Autenticação

A App Engine permite a uma aplicação desenvolvida nesta plataforma, e por intermédio de uma API, a sua integração com o sistema de autenticação da Google. Desta forma, torna possível à aplicação, distinguir automaticamente 3 tipos de utilizadores: os utilizadores não autenticados, os utilizadores autenticados e os administradores que constituem o conjunto de pessoas que pertence à equipa de desenvolvimento ou que nela foram incluídas.

2.8.4 Serviços disponibilizados

Foram também disponibilizados pela plataforma diversas APIs que permitem o acesso a serviços de grande importância, como *Web services*, ou acesso a outros dados através do protocolo HTTP ou HTTPS, envio de *e-mails*, manutenção de caches e manipulação de imagens.

2.8.5 Ambiente de desenvolvimento

O desenvolvimento e teste das aplicações pode ser feito localmente e em qualquer plataforma (Windows, Mac Os X, Linux e outras), por intermédio de um SDK *open source*, o App Engine Software Development Kit, disponível para download no site da Google¹⁵, e que

¹⁴ <http://www.djangoproject.com/>

¹⁵ <http://code.google.com/intl/pt/appengine/downloads.html>

simula todas as operações necessárias, incluindo a autenticação de utilizadores. Para isso este SDK inclui um *Web server* e uma versão local da Datastore. Oferece ainda suporte ao envio de *e-mails*, entre outras funcionalidades que proporcionam uma completa autonomia ao programador.

O App Engine Software Development Kit necessita da versão 2.5 do Python instalada na máquina local para poder correr. Caso ainda não esteja instalada, esse deverá ser o primeiro passo antes da instalação do SDK.

2.8.6 Ambiente de produção

O *update* da aplicação para o servidor da Google é efectuado por linha de comando, por intermédio de uma aplicação fornecida também pelo SDK.

2.8.7 Limitações

Para além das cotas relativas ao espaço de armazenamento (500MB) e às *page views*(5 milhões por mês), são impostas outras limitações relativamente aos recursos disponibilizados. A maior parte destas limitações são estabelecidas numa base diária, ou seja, os valores atingidos pela aplicação são limpos ao fim de 24 horas. Caso o limite imposto seja atingido, o recurso deixa de estar disponível até que seja limpo, ou seja, até que se complete o período de tempo equivalente a um dia, período esse que começa sempre às zero horas.

Um inconveniente importante da Google App Engine prende-se com o tempo de resposta que é exigido na satisfação do pedido de uma página. Se o pedido não é satisfeito dentro de um pequeno intervalo de tempo que se limita a apenas alguns segundos, a App Engine devolve um erro. Este espaço de tempo adquire um carácter dinâmico, podendo ser encurtado, caso o número de *timeouts* seja frequente.

2.8.8 Consola de Administração

Uma consola de administração é também disponibilizada, tanto para o desenvolvimento local, como para a versão de produção, tendo no entanto estas duas consolas funcionalidades distintas, mas ajustadas às diferentes necessidades de cada um dos ambientes. Enquanto que a consola local apenas permite a visualização, edição e criação de entidades, a consola de produção para além destas operações permite a visualização de diversas estatísticas relacionadas com a utilização da aplicação e de informações relativas ao estado de utilização das diversas quotas impostas. Possibilita ainda a consulta dos diversos tipos de *logs* que são gerados e a consulta da informação dos índices da base de dados. Para além disso, possibilita a edição do título da aplicação, a gestão das contas dos utilizadores, o controlo de versões, e a consulta do histórico das operações realizadas na própria consola.

2.8.9 Acesso e utilização da Google App Engine

Para correr uma aplicação na Google App Engine é necessário ter uma conta no Google, aceder à Google App Engine¹⁶ e criar uma nova aplicação. Antes de criar a aplicação, é obrigatório obter um código que é recebido por SMS. Obtido este código e completado o registo da aplicação, basta fazer o seu upload utilizando um utilitário do SDK para que fique disponível a todos os que a pretendam visualizar.

¹⁶ <http://appengine.google.com>

2.9 As Redes sociais

O OpenSocial partiu de uma parceria liderada pela Google que contava com diversas redes sociais, como a LinkedIn, o Hi5 ou o MySpace, e que pretendia fazer frente ao Facebook que resolveu manter-se afastado deste *standard*, proporcionando aos programadores um ambiente proprietário para o desenvolvimento de aplicações.

O Orkut, a rede social da Google, tem uma representação muito pequena em Portugal, apesar de a maioria dos seus utilizadores falarem a língua portuguesa. Em Fevereiro de 2009, o Brasil liderava o ranking dos países que mais utilizam esta rede social, de acordo com dados divulgados no site do Orkut¹⁷. Por outro lado, o Hi5 goza de uma elevada popularidade em Portugal para além de se encontrar actualmente posicionado entre os mais populares sites sociais¹⁸.

O Hi5 tem vindo a trabalhar com afinco na causa OpenSocial. Em Novembro de 2007 iniciou a aventura nesta área, disponibilizando aos programadores uma *sandbox*¹⁹ que lhes permitia testarem as suas aplicações. Nesta altura, as funcionalidades de uma aplicação social no Hi5 estavam limitadas à obtenção de dados públicos do utilizador e da sua lista de amigos. Para além disso, não eram dadas garantias do seu funcionamento no browser da Microsoft, pois foi apenas testada com o Firefox.

Para instalar novas aplicações na *sandbox*, é necessário efectuar primeiro o registo no Hi5 e posteriormente efectuar também o registo como programador. Para adicionar e testar uma aplicação no ambiente do Hi5, basta depois fornecer o URL dessa aplicação, que terá que se encontrar alojada num qualquer servidor que permita o acesso externo.

A partir de Março de 2008 passou a ser possível convidar os amigos a instalar uma aplicação. Foi criada uma interface *standard* que uniformiza e facilita o envio de convites a todos os membros da lista de amigos, podendo cada utilizador receber apenas um convite para uma aplicação do mesmo amigo. Nesta data passou a estar também disponível o envio de notificações a quem possuísse a mesma aplicação instalada ou a qualquer membro da lista de amigos. Outra funcionalidade disponibilizada foi a possibilidade de envio de *e-mails* para todos os utilizadores que possuíssem a aplicação instalada e que tivessem dado permissões para isso. O limite de *e-mails* foi fixado em 1 mensagem por aplicação, por utilizador, por dia. As *Friend Updates* passara a estar disponíveis para as aplicações, o que permite a um utilizador receber, por exemplo, a informação de que um seu amigo instalou determinado programa. Foi também nesta data que a plataforma ficou acessível ao público contando com 53 aplicações aprovadas.

O Hi5 foi o primeiro parceiro da Google a disponibilizar a versão 0.8 do OpenSocial, o que aconteceu em Agosto de 2008, altura em que ficou disponível para os programadores uma versão beta, tendo sido garantida, no entanto, a retro compatibilidade com a versão 0.7. Esta versão implementa uma API baseada na RESTfull Data API que se baseia no protocolo REST para comunicações *server-to-server*, e que vai permitir que as aplicações OpenSocial corram também no *desktop*, em telemóveis e em outros ambientes sem suporte de JavaScript. Estas aplicações passam assim a poder aceder à informação dos perfis dos utilizadores e às relações existentes entre eles na rede social, e para além disso, a poder aceder e publicar acções na rede social e ainda aceder e escrever informação relacionada com a aplicação no perfil do utilizador, na forma de pares chave-valor. A implementação da versão 0.8 do OpenSocial passou a ser disponibilizada na plataforma de produção em Outubro de 2008. Desde essa data têm vindo a ser feitas melhorias, introduzidas novas funcionalidades, como os OpenSocial Templates, e corrigidos alguns *bugs*. Aos programadores é disponibilizado um espaço para os auxiliar no desenvolvimento de aplicações para este *container*²⁰.

17 <http://www.orkut.co.in>

18 <http://www.ebizmba.com/articles/social-networking>

19 <http://sandbox.hi5.com>

20 <http://www.hi5networks.com/developer/>

2.10 Resumo e Conclusões

Os Google Gadgets podem ser divididos em dois tipos, os Google Desktop Gadgets e os Google Internet Gadgets. Ambos os tipos de *gadgets* são baseados em JavaScript, ao que se acrescenta a linguagem XML. Os primeiros têm em vista o *desktop*, apesar de também poderem funcionar na *home page* do iGoogle. Estes *gadgets* contam com um ambiente de desenvolvimento especialmente criado para a sua construção, o Google Desktop SDK. Os Google Internet Gadgets são orientados para a utilização na *Internet* e correm em páginas *Web*. As novidades mais recentes relativamente aos *gadgets* têm a ver com o OpenSocial, uma API que estende a sua funcionalidade aos dados disponíveis nas redes sociais. É este tipo de *gadget* que maior interesse desperta actualmente.

Apesar da ambicionada standardização do OpenSocial, existem ainda algumas diferenças entre os diversos *containers* que é necessário ter em conta. Um dos pontos a rever antes da submissão da aplicação a um novo *container* prende-se com as vistas ou *views* que cada um suporta. O OpenSocial define 4 *views* possíveis: Profile, Canvas, Preview e Home. Cada uma destas *views* tem como objectivo adequar a aplicação a determinados tipos de visualização e aos locais e circunstâncias em que elas são mostradas. Nem todos os *containers* suportam todas estas *views*. O mais habitual é suportarem apenas algumas delas como o caso do Orkut que suporta as *views* Profile e Canvas. Por outro lado, o Hi5, por exemplo, já suporta as *views* Profile, Preview e Canvas. Para além disso, as características de cada *view* podem variar entre *containers*, tomando especificidades diferentes, tais como a dimensão máxima ou restrições a elas associadas. Para além disso, a implementação da API OpenSocial por parte de cada um dos *containers* pode ser diferente, podendo levar a que algumas funcionalidades não sejam suportadas ou seja limitado o acesso a alguns dos dados dos utilizadores. Encontram-se disponíveis para consulta as matrizes²¹ dos testes de conformidade que foram realizados para alguns dos *containers* que implementam a API OpenSocial e que podem constituir uma preciosa ajuda no desenvolvimento de aplicações sociais para um desses *containers* e mais ainda se a aplicação de destinar a múltiplos *containers*.

As facilidades oferecidas aos programadores que desenvolvem aplicações sociais necessitam também de ser melhoradas. Existe a necessidade de criação de um ambiente de desenvolvimento que torne o processo de construção e testes mais facilitado.

O GGE poderá facilitar a tarefa do teste de *gadgets*, não só devido à funcionalidade fornecida pela opção *preview*, que se torna útil para *gadgets* sem a componente social, mas porque ao ser gravado, o *gadget* fica disponível por intermédio de um URL externo, que possibilita o seu acesso por parte de um *container* OpenSocial. Por outro lado, questões relacionadas com a formatação do texto, com a falta de qualquer tipo de assistente que auxilie na escrita de código e principalmente com alguns problemas relacionados com a cache, tornam a sua usabilidade bastante baixa. No entanto, pode ser utilizado livremente qualquer editor de texto para criar um *gadget*. Disponibilizá-lo depois num qualquer servidor *Web* será o suficiente para que um *container* OpenSocial aceda ao ficheiro e corra a aplicação.

A utilização da App Engine permite a criação e manutenção de uma aplicação *Web* incluindo o acesso e manutenção de dados sem necessidade de manter um servidor *Web* próprio ou de um gestor de Bases de Dados. A base de dados que integra é anunciada como sendo uma base de dados distribuída de alta performance e escalonável, o que lhe permite ir acompanhando as necessidades da aplicação. Em alternativa poderia ser utilizada uma base de dados relacional como o MySQL em conjugação com, por exemplo, o PHP e alojar a aplicação em qualquer servidor *Web*. No entanto, a utilização da App Engine não só constitui um desafio mais aliciante pela sua novidade, como, apesar das desvantagens, principalmente as relacionadas com o sistema de cotas, aparenta bastantes benefícios em relação a qualquer outro sistema, nomeadamente ao nível do alojamento gratuito, da performance, das facilidades de gestão e das funcionalidades oferecidas das quais se destacam as seguintes:

²¹ <http://opensocial-compliance.appspot.com/>

Enquadramento Tecnológico

- Servidor de páginas *Web* dinâmicas;
- Armazenamento persistente (App Engine Datastore);
- Escalonamento e *load balancing*;
- APIs para autenticação e envio de *e-mails*;
- Um *Framework Web* (Webapp);
- Simulação do ambiente da Google App Engine, no computador local, por intermédio do SDK App Engine Software Development Kit;
- Controlo de versões;
- Consola de administração com uma interface *Web-based*.

Será por isso esta a plataforma escolhida para a construção do *backoffice* da aplicação social como também para o alojamento desta aplicação.

Quanto ao *container*, o Orkut seria à partida o alvo mais fácil da aplicação OpenSocial a desenvolver, uma vez que se está a utilizar a tecnologia criada pela Google. No entanto, o lema do OpenSocial é o de desenvolver para 1 e utilizar em muitos. Utilizando uma rede social que não o Orkut, este lema pode ser posto à prova. O Hi5 é a rede social mais utilizada em Portugal. Segundo um estudo da Markttest[Mar09] realizado em Janeiro de 2009 e publicado no site da Agência Financeira, o Hi5 foi mesmo o site com o maior número de *page views* em 2008, rondando os 6,5 mil milhões em Portugal. Neste estudo, o google.pt foi o site que recebeu mais utilizadores únicos. Dados revelados no site do Hi5²², mostram que 30% dos seus utilizadores pertencem à faixa etária dos 13 aos 18 anos, representando a maior fatia os utilizadores entre os 18 e os 24 anos, com 45%. Para além dos seus 80 milhões de utilizadores registados, esta rede social é a preferida entre os portugueses, tornando-se por isso a principal razão por ter sido escolhida como *container* da aplicação social a desenvolver.

22 <http://www.hi5networks.com/developer/newsblog.html>

3 Estilos de aprendizagem

As preferências no que toca à forma como cada indivíduo adquire novos conhecimentos não são coincidentes. Também a forma como esses conhecimentos são assimilados por cada um, difere de pessoa para pessoa. Uns adquirem mais facilmente conhecimentos através da leitura, enquanto outros preferem ouvir alguém fazer uma explanação oral sobre o mesmo assunto. Alguns aprendem melhor fazendo, enquanto que outros optam por uma reflexão aprofundada sobre a matéria em estudo.

Existem muitas formas de adquirir e processar informação. Faz por isso todo o sentido conhecer o modo como os alunos melhor adquirem conhecimentos para que o processo de ensino/aprendizagem seja efectivo. No que concerne ao ensino presencial, este conhecimento é importante não para a utilização de técnicas de ensino que vão de encontro ao estilo de aprendizagem de cada indivíduo, pois tal seria impraticável numa sala de aula, mas no sentido de poder fornecer ao professor um maior conhecimento sobre os seus alunos, conhecimento esse que poderá utilizar na aplicação dos métodos que melhor se enquadram na sua estratégia de ensino. Para que esta estratégia seja bem sucedida é importante que os estilos de ensino empregues, cubram durante a maior parte do tempo as preferências da maioria dos alunos. Por outro lado, também é importante “treinar” os alunos em todos os estilos de aprendizagem, pois seguramente será necessário recorrerem a todos eles ao longo da sua vida profissional e pessoal. Por esse motivo, e segundo Felder[Fel05], a estratégia do professor deverá passar por, em alguns momentos, contrariar o estilo de aprendizagem predominante dos alunos, no sentido de os munir de uma maior polivalência no que diz respeito aos métodos de aprendizagem, e em outras alturas optar por uma estratégia de ensino que vá de encontro às preferências da maioria da turma. Esta opção poderá ser tomada em função dos conteúdos a abordar ou mesmo em função da disposição psicológica da turma em determinado momento.

A administração de metodologias que vão de encontro ao estilo de aprendizagem de cada aluno pode ser facilmente efectuada em cursos de *e-learning*. Ao contrário do que acontece num sistema de ensino presencial em que os alunos se encontram numa sala de aulas e todos têm que assistir às mesmas orientações do professor, num sistema de *e-learning*, os processos de ensino podem ser individualizados. Torna-se fácil, neste caso, disponibilizar diferentes metodologias que podem ser escolhidas pelos formandos de acordo com as suas preferências pessoais.

Apesar de esta ser uma estratégia controversa, devido essencialmente ao facto de não se estarem a preparar os alunos para a sua vida profissional e mesmo pessoal, em que serão sujeitos a vários estilos de aprendizagem e para os quais deveriam estar igualmente aptos, temos ainda que ter em consideração que, na sociedade em que vivemos, e principalmente no âmbito profissional, as pessoas deparam-se frequentemente com situações em que se vêem obrigadas a

assimilar e a processar uma grande quantidade de conteúdos e de práticas num curto espaço de tempo. Neste contexto, seria importante que cada indivíduo aprendesse segundo o método que lhe fosse mais favorável pois a aquisição dos novos conhecimentos seria efectuada de uma forma mais rápida. Não se pretende no entanto realizar uma análise profunda desta problemática, mas sim de fornecer os instrumentos que possam apoiar estudos mais abrangentes desenvolvidos por especialistas na área da educação.

No sentido de ensaiar a aplicação com um questionário que aferisse as preferências dos utilizadores sobre estilos de aprendizagem, e como a elaboração desse questionário sai fora do âmbito deste trabalho, houve a necessidade de adoptar um questionário já existente, desenvolvido por alguém, com provas dadas nesta área.

3.1 Análise de questionários

Existem inúmeros questionários elaborados com o objectivo de aferir sobre os estilos de aprendizagem de cada indivíduo e que foram já objecto de análises, validações e estudos diversos por especialistas.

Pretende-se, acima de tudo, encontrar um questionário simples e fácil de responder mas também que tenha algum reconhecimento pela comunidade científica.

Como se pode constatar em Felder[Fel05], os modelos de estilos de aprendizagem desenvolvidos e que foram objecto de um maior número de estudos na área da educação, foram: o modelo de Kolb; o modelo de Felder and Silverman; o modelo de Herrmann, o modelo de Dunn e Dunn e o modelo de Carl Jung, Theory of Psychological Type, operacionalizado por Isabel Briggs Mayers e pela sua mãe, Katharine Cook Briggs nos anos 40 por intermédio do Mayer-Briggs Type Indicator;

O Mayer-Briggs Type Indicator permite efectuar a análise de um indivíduo, explorando a ligação existente entre personalidade, temperamento, estilo de aprendizagem e escolhas profissionais[Tan04]. A utilização deste instrumento encontra-se disponível on-line para compra, e permite categorizar indivíduos nos seguintes tipos: *extraverts/introverts, sensors/intuitors, thinkers/feelers, judgers/perceivers*.

Kolb desenvolveu um modelo de processamento cognitivo baseado na sua teoria Experiential Learning[Kol84], que consiste num processo cíclico que envolve quatro escalas: *Concrete Experience, Reflective Observation, Abstract Conceptualization* e *Active Experimentation*. O binómio *Concrete Experience/Abstract Conceptualization* tem a ver com a forma como a informação é adquirida e o binómio *Active Experimentation/Reflective Observation* tem a ver com a forma como essa informação é interiorizada[Fel05]. A teoria de Kolb baseou-se no trabalho de proeminentes figuras da área das ciências da educação como John Dewey, Kurt Lewin, Jean Piaget, William James, Carl Jung, Paulo Freire, Carl Rogers entre outros[Kol05]. Baseado neste modelo, Kolb criou e aperfeiçoou um instrumento designado de Learning Style Inventory, um teste que pretende aferir o estilo de aprendizagem predominante de cada indivíduo que a ele se submete. Este teste é composto por 12 questões. Cada questão consiste numa frase que o utilizador terá que completar, tendo para isso quatro opções possíveis. O Learning Style Inventory é vendido pela HayGroup.

Um outro instrumento que é por vezes utilizado para avaliar as preferências na escala do modelo de Kolb é, segundo Felder[Fel05], o Learning Type Measure, desenvolvido por Bernice McCarthy e que pode ser adquirido on-line. Este questionário é composto por 26 perguntas, repartidas por dois grupos. O primeiro é constituído por 15 questões, cada uma com 4 itens para ordenar por ordem de preferência. No segundo grupo, composto por 11 perguntas é solicitada a escolha de uma de duas opções possíveis para cada questão. O tempo previsto para a conclusão do teste é de 30 minutos.

O Index of Learning Styles é um questionário elaborado tendo em conta o modelo de Felder and Silverman e avalia as preferências de estilos de aprendizagem nas dimensões *Sensing/Intuiting, Visual/Verbal, Active/Reflective, e Sequential/Global*. Este questionário é de

utilização gratuita e encontra-se disponível on-line. O questionário é composto por 44 afirmações que terão de ser completadas com uma de duas opções possíveis.

O modelo de Herrmann utiliza o Herrmann Brain Dominance Instrument, que associa as preferências de aprendizagem com as funções específicas de cada um dos quadrantes do cérebro. Este instrumento utiliza 120 itens o que parece ser um número excessivo de questões. Para além disso, a avaliação dos resultados do teste é bastante difícil[Bro03].

O Productivity Environmental Preference Scale, construído com base no modelo de Rita Dunn e Kenneth Dunn, pretende avaliar as preferências ao nível dos Elementos Ambientais (som, luz, temperatura e design), dos Elementos Emocionais (motivação, persistência, responsabilidade e necessidade de estruturação), dos Elementos Sociais (trabalhar sozinho, com parceiros, com um adulto ou outra combinação), e dos Elementos Físicos (forças, entradas, hora do dia, necessidade de mobilidade). Este questionário possui 100 questões[Dun79].

Para além dos modelos de estilos de aprendizagem referidos em [Fel05], e que deram origem a ferramentas de avaliação desses mesmos estilos, existem muitos outros questionários como o VARK ou o Learning Styles Questionnaire que pretendem igualmente aferir aspectos relacionados com as preferências de cada indivíduo relativamente ao modo como melhor adquirem novos conhecimentos.

O VARK Inventory foi desenvolvido por Neil Fleming em 1987 e ocupa-se apenas de uma parte do Myers-Briggs Personality Type Indicator, nomeadamente da parte relacionada com as preferências de cada indivíduo relativamente ao estilo de aprendizagem, ligando-as mais com aspectos do domínio sensorial. VARK é um acrónimo para *Visual, Aural, Read/write and Kinesthetic*. São disponibilizadas na *Internet*²³ diversas versões deste questionário, entre as quais encontra-se uma para jovens entre os 12 e os 18 anos, também disponível na versão portuguesa. Este questionário específico é composto por 16 questões, podendo o utilizador seleccionar por cada pergunta, uma ou mais respostas dentro das quatro opções disponíveis.

O Learning Styles Questionnaire é uma ferramenta desenvolvida por Peter Honey e Alan Mumford que pretende medir, por intermédio de um questionário constituído por 80 perguntas na sua versão completa, as preferências de aprendizagem dos alunos relativamente a 4 escalas: *Activist, Reflector, Theorist* e *Pragmatist*. Para responder às questões, o avaliado pode indicar uma de duas opções para cada pergunta.

Na tabela 3.1 é apresentado um resumo, dos questionários apresentados até ao momento.

3.2 Resumo e Conclusões

O Mayer-Briggs Type Indicator, o Herrmann Brain Dominance Instrument, o Dunn and Dunn Productivity Environmental Preference Survey e o Honey and Mumford Learning Styles Questionnaire apresentam um número excessivo de perguntas o que praticamente inviabiliza a sua utilização de forma espontânea na *Internet*.

O Learning Style Inventory, o Learning Type Measure e o VARK Learning Style Test apresentam questionários mais fáceis de responder pela quantidade de perguntas que apresentam. O modelo de Kolb, talvez o mais conceituado e o que tem sido objecto de um maior número de estudos, é também um dos mais pequenos e fáceis de responder, apresentando apenas 12 questões. Por tudo isto, afigurava-se o modelo ideal par utilizar com a aplicação a desenvolver. No entanto, nenhum destes modelos é de acesso gratuito.

Por fim, o Index of Learning Styles, também um dos mais conhecidos e conceituados, utiliza um questionário que, apesar de ter um número considerável de perguntas, as suas respostas são bastante simples, tendo o utilizador que escolher uma de entre duas respostas possíveis. Este questionário apresenta ainda a vantagem de se encontrar disponível na *Internet*²⁴, onde podemos encontrar também as cotações das respostas. Este material pode ser utilizado sem

23 <http://www.vark-learn.com>

24 <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/ILS.pdf>

Estilos de aprendizagem

qualquer custo por qualquer pessoa que pretenda aferir os seus estilos de aprendizagem preferenciais e ainda, por estudantes e instituições de ensino desde que não utilizado com fins comerciais.

Tabela 3.1: Questionários para avaliação dos estilos de aprendizagem

<i>Instrumento de Avaliação</i>	<i>Autores</i>	<i>Categorização</i>	<i>Composição do Questionário</i>
Mayer-Briggs Type Indicator ²⁵	Isabel Briggs Mayers e Katharine Cook Briggs	<i>Extraverts/Introverts, Sensors/Intuitors Thinkers/Feelers, Judgers/Perceivers</i>	88 perguntas 2 respostas possíveis
Learning Style Inventory ²⁶	David A. Kolb	<i>Diverger (Concrete/Reflective) Assimilators (Abstract/Reflective) Convergents (Abstract/Active) Accommodators (Concrete/Active)</i>	12 perguntas 4 respostas possíveis
Learning Type Measure ²⁷	Bernice McCarthy	<i>Analytic, Common Sense, Dynamic, Innovative</i>	15 perguntas 4 itens para ordenar
		Como a informação é processada	11 perguntas 2 respostas possíveis
Index of Learning Styles ²⁸	Richard M. Felder e Barbara A. Soloman	<i>Sensing/Intuiting, Visual/Verbal Active/Reflective, Sequential/Global</i>	44 perguntas e 2 respostas possíveis
Herrmann Brain Dominance Instrument ²⁹	William Herrmann	<i>Analytical thinking, Sequential thinking Interpersonal thinking, Imaginative thinking</i>	120 perguntas
Dunn and Dunn Productivity Environmental Preference Survey	Rita Dunn e Kenneth Dunn	<i>Environmental preferences Emotional factors preferences Sociological factors preferences Physical preferences</i>	100 perguntas
VARK Learning Style Test ³⁰	Neil Fleming	<i>Visual learners, Auditory learners, Reading/writing-preference learners, Kinaesthetic learners or tactile learners</i>	16 perguntas 4 respostas possíveis
Honey and Mumford Learning Styles Questionnaire ³¹	Peter Honey e Alan Mumford	<i>Activist, Reflector, Theorist, Pragmatist</i>	80 Perguntas 2 opções possíveis (verdadeiro ou falso)

Na *Internet* existem ainda inúmeros questionários, a maioria de utilização gratuita e que se propõem avaliar diversos aspectos relativos aos estilos de aprendizagem. Alguns deles encontram-se esquematizados na tabela 3.2.

25 <http://www.knowyourtype.com>

26 <http://www.haygroup.com>

27 <http://www.aboutlearning.com>

28 <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>

29 <http://www.hbdi.com/WholeBrainProductsAndServices/assessment.cfm>

30 <http://www.vark-learn.com/english/page.asp?p=questionnaire>

31 <http://www.peterhoney.com>

Estilos de aprendizagem

Tabela 3.2: Alguns questionários disponíveis na *Internet*

<i>Instrumento de Avaliação</i>	<i>Categorização</i>	<i>Composição do Questionário</i>
Paragon Learning Style Inventory ³²	<i>Extrovert/Introvert, Sensate/Intuitive Feeler/Thinker, Judger/Perceiver</i>	52 Perguntas 2 opções possíveis
Memletics Learning Styles Inventory ³³	<i>Visual, Aural, Verbal, Physical, Logical, Social, Solitary</i>	70 Perguntas 3 opções disponíveis
Perceptual Learning-Style Preference Questionnaire ³⁴	<i>Visual, Tactile, Auditory, Group, Kinesthetic, Individual</i>	30 Perguntas 5 opções possíveis
The Personal Learning Styles Inventory For Students ³⁵	<i>Visual, Auditory, Kinesthetic</i>	36 perguntas 2 opções possíveis
Jewler and Gardner Personal Style Inventory Questionnaire ³⁶	<i>Introversion/Extroversion, Intuition/Sensing, Thinking/Feeling, Perceiving/Judging</i>	40 perguntas 5 opções possíveis
O'Brien Preferred Learning Modality Questionnaire ³⁷	<i>Visual, Auditory, Kinesthetic</i>	30 Perguntas 3 opções
Learning Styles Questionnaire adapted from Colin Rose(1987) ³⁸	<i>Visual, Auditory, Kinesthetic</i>	9 Perguntas 3 opções de escolha
Learning Style Inventory from Pennsylvania State University ³⁹	<i>Auditory, Visual, Tactile</i>	24 perguntas 3 escolhas possíveis
Jung Typology Test ⁴⁰	<i>Introverted/Extrovert, Sensing/Intuitive Thinking/Feeling, Judging/Perceiving</i>	72 perguntas 2 opções possíveis
Catherine Jester DVC Learning Style Survey for College ⁴¹	<i>Visual/ Verbal, Visual/ Nonverbal Tactile/ Kinesthetic, Auditory/ Verbal</i>	32 perguntas 3 opções de escolha
Howard Gardner's Learning Styles Test ⁴²	<i>Linguistic, Logical- Mathematical, Musical, Bodily- Kinesthetic, Spatial- Visual, Interpersonal, Intrapersonal</i>	>16 anos 70 perguntas 4 opções de escolha >8 e <16 35 perguntas 4 opções de escolha

Face à vasta quantidade de questionários disponíveis, torna-se essencial que se disponibilize uma aplicação destinada não só à utilização de apenas um questionário, mas que seja facilmente adaptável a um leque mais alargado de testes, de modo a que não fiquemos presos a apenas a um, e que poderia até vir a revelar-se não ser o mais apropriado. Desta forma, será utilizado para efeitos de demonstração do funcionamento da aplicação o Index of Learning Styles Questionnaire de Richard M. Felder e Barbara A. Soloman por ser sobejamente conhecido e amplamente utilizado, e para além disso, de utilização gratuita, não impedindo que posteriormente seja substituído por qualquer outro que se venha a revelar mais eficaz.

32 <http://www.oswego.edu/plsi/plsi48a.htm>

33 <http://www.learning-styles-online.com/inventory/questions.asp>

34 <http://lookingahead.heinle.com/filing/l-styles.htm>

35 http://www.howtolearn.com/lsinventory_student.html

36 <http://www.e-radiography.net/articles/learningstyles/questionnaire.htm>

37 <http://www.e-radiography.net/articles/learningstyles/modalityq.htm>

38 <http://www.chaminade.org/inspire/learnstl.htm>

39 <http://www.personal.psu.edu/bxb11/LSI/LSI.htm>

40 <http://www.humanmetrics.com/cgi-win/JTypes2.asp>

41 <http://www.metamath.com/lswweb/dvcllearn.htm>

42 <http://www.businessballs.com/howardgardnermultipleintelligences.htm#multiple%20intelligences%20tests>

4 Implementação

4.1 Contexto

Esta dissertação apresenta como produto final, uma aplicação a instalar na rede social hi5, que consiste numa ferramenta para aplicação de questionários de estilos de aprendizagem. Os resultados desses questionários deverão ser apresentados de uma forma gráfica para uma melhor compreensão. A aplicação deve utilizar a API OpenSocial e servir-se de mecanismos que procurem incentivar a sua instalação e disseminar a sua utilização pelos utilizadores da rede social. Para a gestão dos dados apresentados ao utilizador e gestão dos resultados obtidos nos questionários torna-se necessário desenvolver um site que funcione como *backoffice* da aplicação social.

4.2 Características dos utilizadores

A aplicação social destina-se a todos os utilizadores de redes sociais, em especial aos utilizadores com conta no Hi5. Devido à heterogeneidade do público-alvo, impõe-se que a aplicação possua uma interface extremamente amigável e que não levante dúvidas quanto à sua utilização.

Por outro lado, a componente de administração terá um leque mais restrito de utilizadores que se limitará aos administradores da aplicação que tanto poderão ser pessoas com conhecimentos na área da informática, como indivíduos de áreas relacionadas com a educação e com poucos ou nenhuns conhecimentos das novas tecnologias. A pensar neste último tipo de utilizador, deverá também haver algum cuidado na elaboração deste site, tornando-o o mais intuitivo e fácil de utilizar quanto possível. No entanto, e enquanto não for efectuado um editor de questionários que coloque a sua inserção e edição ao alcance de qualquer tipo de utilizador, este processo poderá ter que contar com a supervisão de alguém com alguns conhecimentos de XML.

4.3 Requisitos funcionais

4.3.1 Aplicação social

- Leitura de toda a informação referente ao questionário activo no momento, a partir de um servidor remoto.
- Leitura dos resultados obtidos num questionário, por um determinado utilizador do Hi5, a partir de um servidor remoto.
- Gestão das respostas dadas ainda não submetidas e controlo do estado do questionário relativamente à pergunta actualmente a ser respondida por intermédio de, por exemplo, uma barra de progresso.
- Cálculo dos resultados do questionário.
- Desenho de gráfico do tipo radar com os resultados obtidos em cada teste.
- Desenho de escalas com os resultados, de acordo com o questionário carregado na aplicação e as respostas dos utilizadores.
- Implementação de algoritmo de atribuição de cores ao utilizador, em função das respostas dadas ao questionário, de modo a que a utilizadores com respostas semelhantes seja atribuída a mesma cor.
- Leitura da informação do perfil do utilizador no Hi5.
- Escrita da informação relativa à cor atribuída pela realização do questionário.
- Visualização da cor de todos os utilizadores que já tenham realizado o questionário e pertençam à lista de amigos do utilizador actual.

4.3.2 Administração

Como suporte ao funcionamento da aplicação social, tornou-se necessária a implementação de uma aplicação para administração dos seus conteúdos e que deverá obedecer aos requisitos a seguir apresentados.

- Controlo de acesso, de modo a permitir apenas a entrada de utilizadores autorizados.
- Leitura de um questionário a partir de um ficheiro XML que deverá estar em conformidade com o DTD definido para o efeito.
- Escrita na base de dados de um ficheiro no formato XML contendo o questionário.
- Consulta de questionários gravados previamente na base de dados.
- Alteração de um questionário existente na base de dados.
- Remoção de um questionário gravado na base de dados.
- Selecção do questionário activo, ou seja, o que irá ser apresentado na aplicação social.
- Consulta do número de utilizadores que já responderam ao questionário.
- Somatório global dos valores obtidos para cada estilo de aprendizagem de cada questionário.

4.3.3 Requisitos não funcionais

- Devido à heterogeneidade do público-alvo, impõe-se que a aplicação possua uma interface extremamente amigável e que não levante dúvidas quanto à sua utilização.
- Deve ter-se em especial consideração o tempo de resposta da aplicação.
- Os utilizadores da aplicação social devem interagir com a aplicação por intermédio de uma interface *Web-based*, integrada no ambiente do Hi5, tendo que para isso proceder à sua instalação.
- Não deve ser necessária a instalação de qualquer *plug-in* no computador do utilizador, necessitando este de possuir apenas um browser e acesso à Internet.
- A aplicação deverá funcionar nos seguintes browsers: Mozilla FireFox, Internet Explorer e Apple Safari e possuir formas de incentivar o utilizador a proceder à sua instalação.
- O processo de selecção das respostas do questionário deve ser muito intuitivo e estudado de modo a que não leve os utilizadores a seleccionarem a mesma questão por inércia. Devem por isso ser implementados mecanismos que tentem igualar o esforço necessário para seleccionar qualquer uma das respostas.
- Os resultados do questionário deverão ser apresentados de forma gráfica de modo a se tornarem mais claros e de compreensão mais fácil e rápida.
- Deverá ser implementado um mecanismo que incentive o utilizador a dar a conhecer a aplicação aos seus amigos, utilizando as potencialidades das redes sociais e da API OpenSocial.
- Deve implementar mecanismos que tornem possível a utilização da aplicação com diversos questionários diferentes.

4.4 Visão geral

O sistema é composto por dois componentes principais: a aplicação social a integrar no Hi5 e que utiliza o Google Charts API e a plataforma Google App Engine, através da qual é possível a persistência de dados. Por sua vez o site de administração, comunica com a Google App Engine, plataforma onde se encontra alojado.

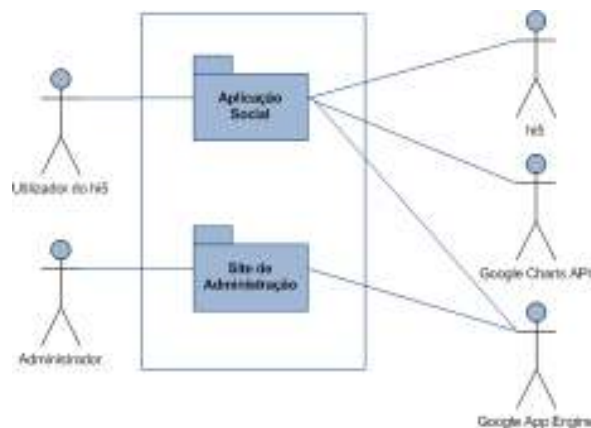


Figura 4.1: Visão Geral

4.5 Funcionamento da aplicação social

O funcionamento da aplicação social descreve-se sucintamente por intermédio de um diagrama de actividades.

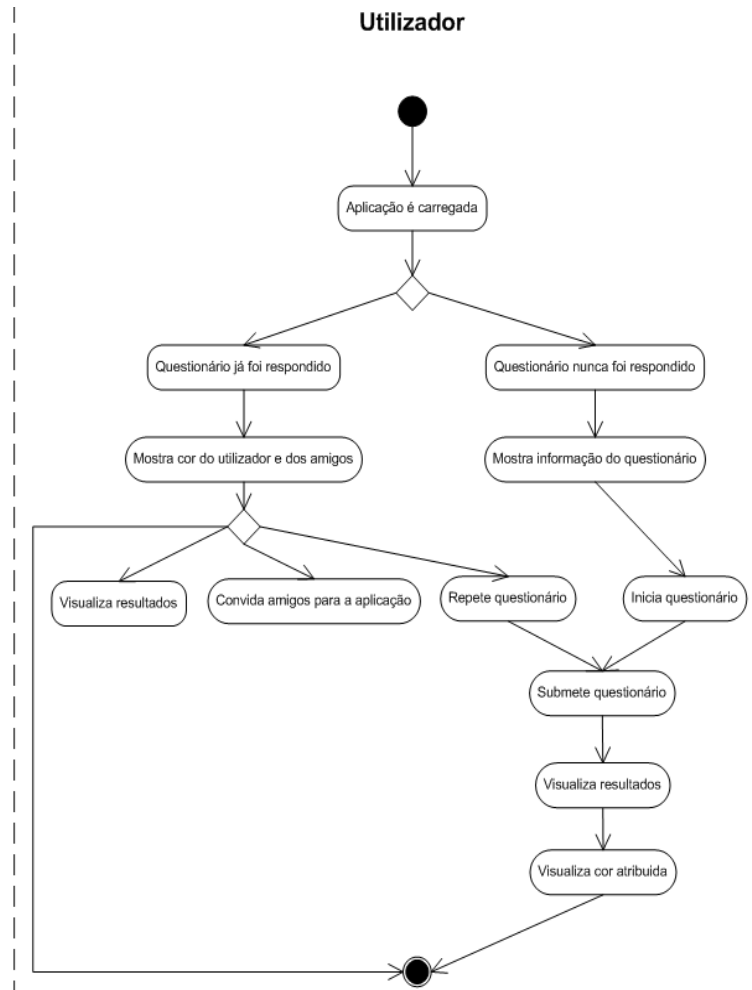


Figura 4.2: Funcionamento da aplicação social

4.6 Arquitectura

No desenvolvimento do sistema podemos considerar dois componentes principais, o primeiro destinado a toda a gestão de informação utilizada pela aplicação social e o segundo a própria aplicação social. O primeiro componente pode ainda ser dividido em dois sub-componentes, o primeiro destinado à gestão dos conteúdos por intermédio de um administrador, fornecendo para isso uma interface *Web* e o segundo destinado a prover a aplicação social com os dados de que necessita e para gravar os dados por ela gerados.

Na Datastore é armazenada a informação relativa aos questionários e respectivos resultados. Os questionários são inseridos na base de dados por intermédio de ficheiros XML.

Implementação

A autenticação e o controlo de acessos são efectuados de modo a restringir os utilizadores da componente de administração aos utilizadores autorizados. O acesso à aplicação social está dependente de autenticação na rede social.



Figura 4.3: Arquitectura lógica

Neste sistema podem ser identificados alguns componentes e relações existentes entre eles.

Aplicação Social

Aplicação acessível aos utilizadores da rede social Hi5, constituída por um ficheiro em formato XML que utiliza HTML e JavaScript para a apresentação dos dados, instalada num *container* OpenSocial (Hi5) e que corre no browser do cliente. Esta aplicação interage com a Google App Engine para leitura e escrita de dados na Datastore. Estes dados são transferidos no formato JSON. Faz ainda pedidos ao Google Charts. Por intermédio da API OpenSocial e realiza operações de leitura e escrita nos perfis dos utilizadores do Hi5.

Site de administração

Este site é desenvolvido com recurso à plataforma Google App Engine. O seu acesso é restrito apenas aos administradores, que são utilizadores registados no Google Accounts e para além disso pertencem à lista de programadores da aplicação.

Google App Engine

Qualquer aplicação desenvolvida com base na App Engine, é criada e configurada a partir de um ficheiro designado de `app.yaml` que define como os pedidos HTTP são satisfeitos de acordo com um sistema de mapeamento de URLs. Estas aplicações são desenvolvidas em Python. Para facilitar o seu desenvolvimento foi utilizado o framework Webapp que se encontra integrado na plataforma.

Implementação

Web Server

O site de administração e a aplicação social são servidos a partir da plataforma Google App Engine que integra um *Web Server*, o que permite o alojamento gratuito e liberta o programador das preocupações inerentes à manutenção de servidores.

Datastore

A Datastore, integrada na App Engine, permite que as aplicações desenvolvidas com recurso a esta plataforma, tenham um meio de persistência de dados. É aqui que são armazenadas as informações relativas aos questionários e às respostas dos utilizadores.

Google Accounts

O acesso ao sistema de contas do Google torna-se disponível por intermédio de uma API. Desta forma, é possível restringir o acesso às aplicações de uma forma bastante simples.

Google Charts

Esta API foi utilizada para o desenho do gráfico que é apresentado ao utilizador após a conclusão de um questionário.

Consola de administração

A Google App Engine disponibiliza uma consola de administração com controlo de acessos e que permite a visualização e manipulação dos dados gravados na Base de Dados da plataforma, ou seja, na Datastore. Para além das operações relacionadas com a manutenção dos dados, esta consola é utilizada para visualização de informações relativas às cotas de utilização e para a consulta de *logs*. É também possível, a partir dela, gerir alguns aspectos relacionados com a aplicação, como por exemplo, o controlo de versões.

O relacionamento entre estes componentes é mostrado na figura 4.4.

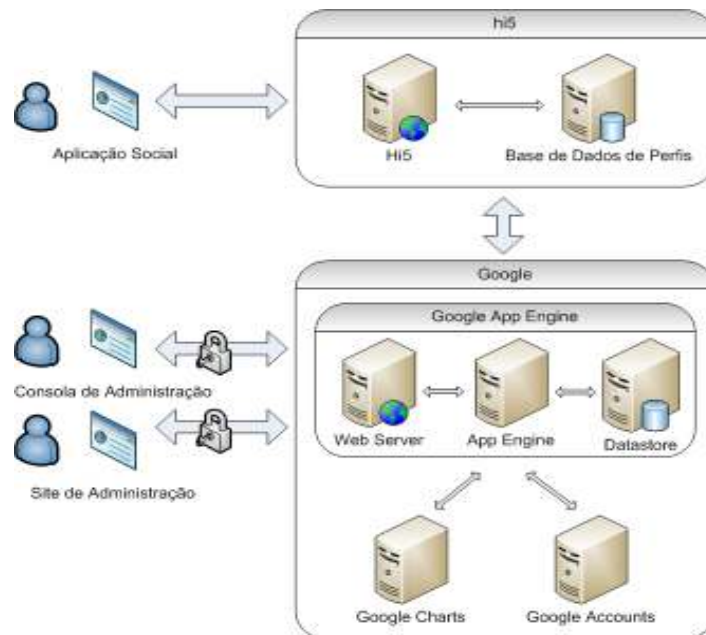


Figura 4.4: Arquitectura física

4.7 Tecnologias

Google App Engine

A necessidade de existir um módulo que servisse de apoio à aplicação social levou à escolha das ferramentas para o seu desenvolvimento. Uma possibilidade seria a utilização de uma linguagem como o PHP, juntamente com uma base de dados para a persistência de dados como poderia ser o caso do MySQL. Seguindo estas escolhas seria necessário ainda encontrar alojamento para o site a desenvolver e também para a aplicação social. Por outro lado, a Google App Engine, surgida em Abril de 2008, suscitou bastante interesse, quer pela sua novidade, que por aquilo que prometia, ou seja, um ambiente completo de desenvolvimento integrado e espaço de alojamento para até 10 aplicações. Para além disso, integra APIs de grande utilidade, como a Google Accounts API.

Python

O Python é uma das linguagens que pode ser utilizada como linguagem de *scripting* para aplicações *Web*. Para facilitar a criação deste tipo de aplicações podem ser utilizados *frameworks*. A sua versão 3.0 foi lançada em Dezembro de 2008. No entanto, esta versão não é compatível com as versões 2.x[Pyt08]. Foi por isso utilizada a versão 2.5, tal como indicado na documentação da Google App Engine[Goo09e]. A utilização do Python prendeu-se com a escolha da plataforma App Engine, visto ser esta a única linguagem por ela suportada até ao momento.

XML

O XML foi utilizado como formato de entrada para leitura e gravação dos questionários na Datastore, por intermédio do Site de Administração. A sua escolha prendeu-se com a facilidade que existe em criar um questionário recorrendo a este formato e à facilidade de compreensão da sua constituição. Por outro lado, com a ajuda de um DTD, pode-se facilmente verificar a validade do questionário a inserir na base de dados. A utilização de um ficheiro XML veio evitar a construção de um editor de questionários, que poderá constituir uma optimização a implementar de futuro.

HTML

O HTML é a linguagem base de qualquer página *Web*, tendo sido utilizado na construção tanto do site de administração como da aplicação social.

JavaScript

O JavaScript, utilizado essencialmente na aplicação social, possibilita a construção de uma aplicação dinâmica, que corra do lado do cliente.

AJAX

Pode ser encarado como uma técnica que permite evitar que a cada pedido de dados ao servidor, toda a página tenha que ser novamente carregada. Esta técnica possibilita as trocas de informação com o servidor sendo actualizadas apenas partes da página. Desta forma são evitados tempos de espera por parte do utilizador, passando o funcionamento da aplicação a assemelhar-se mais com o de uma aplicação tradicional a correr num ambiente cliente-servidor.

JSON (JavaScript Object Notation)

O JSON⁴³ é um formato de texto, independente de qualquer linguagem de programação, que permite a troca de informação entre aplicações. Este é um formato alternativo ao formato XML. A geração e *parsing* de dados no formato JSON é efectuado com recurso a bibliotecas disponíveis para a maior parte das linguagens de programação. Neste caso foi utilizada a biblioteca json.py. Em relação ao XML, o JSON é mais fácil de ler e manusear devido à sua forma de representação da informação. O formato JSON foi utilizado nas transferências de dados entre a aplicação social e a Datastore.

Google Charts API

A Google disponibiliza um grande número de APIs que facilitam a vida aos programadores e tornam as aplicações mais ricas e com um maior número funcionalidades úteis, sem grande esforço de codificação. A Charts API é uma dessas APIs disponibilizada pela Google, que permite a construção de gráficos, completamente parametrizáveis, bastando para tal fornecer um conjunto de argumentos, de acordo com o resultado que se pretende obter.

Google Accounts

Sistema de contas da Google que pode ser acedido a partir de uma API escrita em Python, integrada no AppEngine e que possibilita a integração das contas da Google com a aplicação.

OpenSocial API

É um conjunto de APIs que permite a construção de aplicações sociais em diversos sites[Goo09f]. Os sites que implementam esta API adquirem a designação de OpenSocial *container*, tendo vindo a crescer o número de sites que se associaram a esta iniciativa. A versão 0.8 permite já a comunicação *server-to-server* por intermédio dos protocolos REST e RPC, o que possibilita a criação de aplicações sociais para desktop e telemóveis. Apesar de se pretender que uma aplicação OpenSocial corra em qualquer *container* OpenSocial, existem diferenças de implementação (implementações parciais) ou questões relacionadas com a protecção aos dados que impedem que a mesma aplicação corra em *containers* diferentes. No entanto, para que isso aconteça, é geralmente necessário proceder a pequenas alterações.

Internet Explorer, Firefox e Safari

Estes foram os browsers utilizados para testar o sistema desenvolvido, tendo-se efectuado testes com o Internet Explorer 7, Mozilla Firefox 3 e Apple Safari 3.2.1.

43 <http://www.json.org/>

5 Detalhes e Resultados da Implementação do Site de Administração

Neste capítulo são explicados alguns detalhes de implementação do site de administração assim como alguns pormenores relativos à utilização da App Engine.

5.1 O SDK Google App Engine

O desenvolvimento de aplicações utilizando a Google APP Engine é efectuado recorrendo ao App Engine Software Development Kit⁴⁴ que se encontra disponível para *download* no site da Google. Existem versões para Windows, Mac OS X e Linux, tendo sido a versão utilizada a versão Windows. Para se poder utilizar este SDK, é necessário descarregar e instalar a versão 2.5 do Python⁴⁵, de acordo com indicações da Google.

Após a instalação do Python e do SDK, passam-se a ter na máquina local todas as ferramentas necessárias para o desenvolvimento de aplicações *Web* que venham a correr na Google App Engine, nomeadamente um *Web server*, uma versão local da base de dados, uma versão local da consola de administração, a possibilidade de testar o sistema de autenticação, a possibilidade de envio de *e-mails* e a possibilidade de comunicar com servidores na Internet, de modo a obter dados externos. No desenvolvimento desta aplicação, apenas não foi utilizada a funcionalidade de envio de *e-mails*.

5.2 Configuração da aplicação

Para que o *Web Server* de desenvolvimento possa correr, é necessário criar previamente um ficheiro de configuração denominado de `app.yaml` que utiliza o formato YAML⁴⁶.

44 <http://code.google.com/appengine/downloads.html>

45 <http://www.python.org/download/releases/2.5/>

46 <http://www.yaml.org/>

Detalhes e Resultados da Implementação do Site de Administração

Passa-se a transcrever o ficheiro `app.yaml` criado para a aplicação:

```
application: learningstyle-sapp
version: 1
runtime: python
api_version: 1
handlers:
- url: /static
  static_dir: static
- url: /sapp/*
  script: sappa.py
- url: /*
  script: lssappadmin.py
  login: required
```

Os elementos “application”, “version”, “runtime” e “handlers” são de utilização obrigatória e são descritos de seguida:

application

Nome escolhido para designar a aplicação.

version

Versão actual da aplicação. A App Engine mantém uma cópia de todas as versões que forem submetidas ao sistema de produção. O administrador ou o programador, poderá assim testar as diversas versões no ambiente real e seleccionar aquela que se tornará pública, utilizando para o efeito a consola de administração.

runtime

Actualmente o único valor permitido é “python”, a única linguagem actualmente permitida na Google App Engine.

api_version

Actualmente o único valor possível é 1. Este valor está relacionado com a versão *runtime* do Python.

handlers

Compreende uma lista de URLs. Neste caso, foram definidos 3 *handlers* entre dois tipos possíveis, os estáticos (“static_dir”) e os dinâmicos (“script”). O primeiro, o único do tipo estático, é utilizado, como o próprio nome indica, para servir conteúdos estáticos e corresponde a uma pasta física, criada dentro da pasta da aplicação. É aqui que se encontra alojado o ficheiro XML que constitui a aplicação social, assim como os ficheiros JavaScript, CSS e imagens diversas. O segundo *handler* é utilizado para a satisfação de pedidos por parte da aplicação social de forma que estes não estejam sujeitos a um processo de autenticação, o que inviabilizaria a sua utilização. O terceiro é o utilizado por todas as restantes páginas do site de administração. Como se pode ver pela linha “login: required”, todos os pedidos efectuados por intermédio do script “lssappadmin.py”, estão sujeitos a autenticação. Desta forma, na aplicação apenas é necessário diferenciar aqueles que se encontram autenticados mas que não possuem permissão para aceder aos conteúdos da aplicação e os administradores ou programadores que têm acesso total a todas as opções.

De acordo com o definido no ficheiro `app.yaml`, qualquer pedido por parte da aplicação cujo URL coincida com a expressão regular `/sapp/*`, é servido pelo ficheiro `sappa.py`. Se por outro lado coincidir com a expressão `/*`, é servido pelo ficheiro “lssappadmin.py”.

Para por a correr o *Web Server* utiliza-se o comando `dev_appserver.py` que integra o SDK, fornecendo como parâmetro o nome da aplicação, “learningstyle-sapp”, conforme definido no

ficheiro `app.yaml`. O servidor ficará assim a correr na porta 8080, porta essa que caso necessário poderá ser alterada.

5.3 O *framework* Webapp

Para facilitar o desenvolvimento do site de administração foi utilizado o *framework* Webapp que já vem instalado na App Engine. De acordo com as regras definidas para a utilização deste *framework*, foi criado o ficheiro “`lssappadmin.py`”, do qual se retirou o seguinte fragmento de código, onde são mostradas as componentes essenciais para o correcto funcionamento de uma aplicação que utilize o Webapp.

```
...
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app
...
import loadteste
import gravateste
import layout
import limpabd
import listateste
import estatisticas
import questoes
import activarquestionario
import resultados

class MainPage(webapp.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/html'
        #...
        #Autenticação
        #...

application = webapp.WSGIApplication([('/', MainPage),
                                     ('/load', loadteste.Load),
                                     ('/save', gravateste.Grava),
                                     ('/clearall', limpabd.Limpa),
                                     ('/list', listateste.Lista),
                                     ('/stat', estatisticas.Main),
                                     ('/questions', questoes.Questoes),
                                     ('/results', resultados.Resultados),
                                     ('/activo', activarquestionario.Activa),
                                     ],)

def main():
    run_wsgi_app(application)

if __name__ == "__main__":
    main()
```

São importados os módulos “webapp” e “run_wsgi_app” e os ficheiros Python que fazem parte da aplicação e serão servidos pelo *script* “`lssappadmin.py`”.

A aplicação é uma instância de “WSGIApplication”.

O construtor da classe “WSGIApplication” recebe como parâmetros uma lista de tuplos constituídos por uma *path* e pela classe responsável por processar a resposta.

A função `run_wsgi_app` recebe como argumento uma instância da aplicação e faz com que esta corra no servidor.

A instância “WSGIApplication” reencaminha os pedidos para cada uma das classes de acordo com o mapeamento definido. Por exemplo, quando é feito um pedido HTTP GET ao URL `/`, é criada uma instância da classe `RequestHandler`, associada à classe `MainPage`. É então

chamado o método correspondente à acção do pedido (`get()`, `post()`, `trace()`, ...), que no caso acima será o método `get()`.

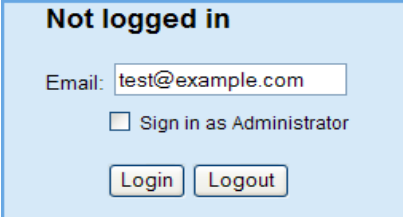
5.4 Controlo de acessos

Para impedir o acesso a utilizadores não autorizados, ou seja, de modo a permitir o acesso à aplicação apenas aos administradores, é acrescentado ao método `get()` o código seguinte.

```
if self.request.get('action') == 'logout':
    self.redirect(users.create_logout_url('/'))
elif users.is_current_user_admin():
    html = self._handleLoadtesteList("")
    html = layout._handleHtmlPageLayout(''.join(html), '1', '2')
    self.response.out.write(''.join(html))
else:
    html = layout._handleHtml_P0('Utilizador não autorizado.')
    html = layout._handleHtmlPageLayout(''.join(html), '0', '2')
    self.response.out.write(''.join(html))
```

O primeiro *if* apenas efectua o *logout*, caso o utilizador tome essa iniciativa por intermédio da interface do site. Neste caso, o parâmetro *action*, contido no URL será igual a “logout”. No segundo *if* é verificado se o utilizador actual é um administrador. Em caso afirmativo, é efectuado o processamento definido em “`_handleLoadtesteList`” e em “`layout._handleHtmlPageLayout`” e por fim é enviada a resposta para o *browser*, caso contrário o utilizador é informado que não possui permissões para aceder ao site.

No ambiente de desenvolvimento, a autenticação é efectuada por intermédio de uma página local que oferece a possibilidade de inserção de um qualquer endereço de *e-mail*. Para que esse endereço seja considerado pertencente a um administrador, basta activar a caixa de selecção disponível para o efeito. Desta forma é possível testar a aplicação a 3 níveis: sem autenticação; com um utilizador comum com conta no Google e com um utilizador com perfil de administrador. Um esquema mais complexo de permissões, terá que ser implementado manualmente. Para este caso concreto, os níveis de acesso disponibilizados cobrem todas as necessidades.



A imagem mostra uma caixa de diálogo de autenticação com o título "Not logged in". Dentro da caixa, há um campo de texto rotulado "Email:" com o valor "test@example.com". Abaixo do campo, há uma caixa de seleção desactivada rotulada "Sign in as Administrator". Na base da caixa, há dois botões: "Login" e "Logout".

Figura 5.1: Simulação local da autenticação

A chamada à página de *login* é feita de forma automática, sempre que não existe nenhum utilizador logado com uma conta Google, de acordo com o que foi definido no ficheiro `app.yaml` e para o URL `/*`.

Quando em modo de produção, ou seja a correr na App Engine e não no servidor local, a página representada na figura 5.1 é substituída pela página de autenticação do Google tal como se pode ver na figura 5.2.

The screenshot shows the Google Accounts sign-in interface. At the top left is the Google Accounts logo. The main heading is "Estilos de aprendizagem uses Google Accounts for Sign In." Below this, a disclaimer states: "Google is not affiliated with the contents of Estilos de aprendizagem or its owners. If you sign in, Google will share your email address with Estilos de aprendizagem, but not your password or any other personal information." Another line of text says: "Estilos de aprendizagem may use your email address to personalize your experience on their website." On the right side, there is a sign-in box with the text "Sign in with your Google Account". It contains two input fields for "Email:" and "Password:". Below these is a checkbox labeled "Remember me on this computer." and a "Sign in" button. A link "[I cannot access my account](#)" is located below the sign-in box. At the bottom right, there is a box with the text "Don't have a Google Account?" and a link "[Create an account now](#)". The footer of the page contains the copyright notice "©2009 Google" and links for "Google Home", "Terms of Service", "Privacy Policy", and "Help".

Figura 5.2: Google Accounts

Após o utilizador ter inserido os seus dados, é redireccionado de novo para o URL da aplicação.

5.5 Satisfação dos pedidos externos

O outro script referido no ficheiro `app.yaml` é o `sappa.py`. Neste são mapeados os URLs que satisfazem os pedidos efectuados por parte da aplicação social, nomeadamente, o pedido de envio do questionário activo no momento, o pedido dos resultados obtidos por um determinado utilizador e o pedido de gravação desses mesmos resultados. Como é óbvio, aqui não é feito qualquer controlo de acessos de modo a que todos consigam aceder à informação. O ficheiro `sappa.py` é listado de seguida.

```
import wsgiref.handlers
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app
import questoes
import resultados

application = webapp.WSGIApplication([
    ('/sapp/questions', questoes.Questoes),
    ('/sapp/results', resultados.Resultados)
])

def main():
    run_wsgi_app(application)
if __name__ == "__main__":
    main()
```

5.6 Persistência de dados

Foi criado no ficheiro `db_model.py`, o modelo da base de dados a utilizar pelo sistema. As tradicionais tabelas do modelo relacional dão lugar a classes Python passando a designar-se de entidades, cada uma constituída por um nome e um conjunto de propriedades descritas pelos atributos da classe, sendo os dados guardados em instâncias dessa classe. Foram então definidas 6 classes que modelam a base de dados.

```
class Teste(db.Model):
    nomeTeste = db.StringProperty(required=True)
    dscTeste = db.StringProperty()
    dscLonga = db.StringProperty()
    activo = db.StringProperty()

class LearningStyle(db.Model):
    nomeTeste = db.StringProperty(required=True)
    idLS = db.StringProperty(required=True)
    nomeLS = db.StringProperty(required=True)
    dscCurtaLS = db.StringProperty(required=True)

class Pergunta(db.Model):
    nomeTeste = db.StringProperty(required=True)
    numPergunta = db.IntegerProperty(required=True)
    dscPergunta = db.StringProperty(required=True)

class Resposta(db.Model):
    nomeTeste = db.StringProperty(required=True)
    numPergunta = db.IntegerProperty(required=True)
    numResposta = db.StringProperty(required=True)
    dscResposta = db.StringProperty(required=True)

class Peso(db.Model):
    nomeTeste = db.StringProperty(required=True)
    numPergunta = db.IntegerProperty(required=True)
    numResposta = db.StringProperty(required=True)
    idLS = db.StringProperty(required=True)
    peso = db.FloatProperty(required=True)

class Resultado(db.Model):
    idUtilizador = db.StringProperty(required=True)
    utilizador = db.StringProperty()
    nomeTeste = db.StringProperty(required=True)
    idLS = db.StringProperty(required=True)
    pontos = db.FloatProperty(required=True, default=0.0)
```

A App Engine cria na pasta da raiz da aplicação, um ficheiro com o nome `index.yaml` onde são definidos os índices utilizados pela base de dados. Este ficheiro é gerado e alterado automaticamente sempre que haja necessidade de criar um novo índice. Os índices são criados automaticamente ao se executar uma *query* simples à base de dados. No entanto em alguns tipos de *queries* será necessário acrescentar manualmente índices a este ficheiro para que seja possível executá-las. Isto acontece, por exemplo em *queries* que necessitem de ordenar os resultados por mais que uma propriedade[Goo09g].

5.7 Comunicação com a aplicação social

A comunicação com a aplicação social processa-se por intermédio dos ficheiros `questoes.py` e `resultados.py`. Estes ficheiros fornecem o acesso ao questionário activo no momento, por parte da aplicação social, e permitem ler e escrever os resultados obtidos por cada utilizador que o responde.

Detalhes e Resultados da Implementação do Site de Administração

Foi utilizada a livreria json.py⁴⁷, de modo a tornar possível a transferência de dados no formato JSON entre a aplicação de administração e a aplicação social. Este foi o formato utilizado por tornar mais fácil o manuseamento dos dados.

5.8 *Layout* do site de administração

Foi definida uma estrutura para todo o site de administração, construída com base em funções definidas no ficheiro layout.py. Desta forma, foi possível manter o mesmo aspecto em todas as páginas sem que fosse necessário repetir o mesmo código por todas elas.



Figura 5.3: *Layout* do site de administração

Todas as páginas utilizam as funções definidas neste ficheiro de modo a manter 4 áreas distintas ao longo de toda a navegação pelo site. A área com o número 4 destina-se aos conteúdos específicos de cada uma das funcionalidades disponibilizadas. Desta forma torna-se bastante simples proceder a qualquer alteração do *layout*, bastando para isso alterar apenas o código numa única localização. A estrutura do site é ilustrada na figura 5.3.

O site foi desenhado de modo a ser correctamente visualizado (sem recorrer à barra de deslocamento horizontal) numa resolução mínima de 1024x768.

⁴⁷ Copyright (C) 2005 Patrick D. Logan

5.9 Funcionalidade disponibilizada

5.9.1 Importação de um questionário

Actualmente, a única forma de introdução de um novo questionário na base de dados é através de um ficheiro XML. O ficheiro XML a criar com o questionário deverá obedecer ao seguinte DTD.

```
<?xml version="1.0"?>
<!DOCTYPE teste[
<!ELEMENT teste (nome, dscteste, dsclonga, learningstyles, perguntas)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT dscteste (#PCDATA)>
<!ELEMENT dsclonga (#PCDATA)>
<!ELEMENT learningstyles (learningstyle+)>
<!ELEMENT learningstyle (#PCDATA)>
<!ATTLIST learningstyle
    id ID #REQUIRED
    ls_dsc_curta CDATA #REQUIRED>
<!ELEMENT perguntas (pergunta+)>
<!ELEMENT pergunta (resposta+)>
<!ATTLIST pergunta p_num CDATA #REQUIRED>
<!ATTLIST pergunta p_desc CDATA #REQUIRED>
<!ELEMENT resposta (r_val+)>
<!ATTLIST resposta
    r_num CDATA #REQUIRED
    r_desc CDATA #REQUIRED>
<!ELEMENT r_val (#PCDATA)>
<!ATTLIST r_val
    ls_id IDREF #REQUIRED]>
```

Para o teste ser carregado, bastará indicar o URL correspondente à sua localização a partir da opção “Carregar Questionário”, como se pode ver na figura 5.4.

Uma aplicação Social Utilizando Estilos de Aprendizagem

Administração Utilizador: l.correia.go Logout

Carregar Questionário

Visualizar Questionário

Questionário Activo

Estatísticas

Insira o URL do ficheiro com o XML do teste que pretende carregar e uma descrição:

URL

Descrição

Carregar ficheiro

powered by Google App Engine

Figura 5.4: Importação de um questionário

Detalhes e Resultados da Implementação do Site de Administração

O questionário é carregado para o site e mostrado ao administrador. Se tudo estiver correcto, este dará a ordem de gravação para a base de dados. O questionário só será gravado caso ainda não exista, ou existindo, ainda não tenha sido respondido. Se o questionário já foi respondido por pelo menos 1 utilizador, e se a estrutura do questionário a inserir corresponder à estrutura do questionário já existente na base de dados, é perguntado ao administrador se o pretende substituir pelo novo. Se este confirmar a operação, apenas as perguntas e as respostas serão gravadas. Esta operação é permitida para que seja possível corrigir, por exemplo, erros ortográficos ou sintácticos que possam existir.

5.9.2 Visualização de um questionário previamente gravado

Apesar de ser possível visualizar, editar e remover conteúdos da base de dados por intermédio da consola de administração da App Engine, não é viável a visualização de todo um questionário por esse processo. Foi por isso necessário fornecer uma opção para uma visualização mais fácil de todos os dados que o constituem. Essa informação fica assim disponível, seleccionando a opção “Visualizar questionário” a partir do menu lateral, que dará acesso à escolha do questionário pretendido. Após a visualização do questionário, é possível proceder à sua remoção, isto se ele ainda não tiver sido respondido por nenhum utilizador da aplicação social.

Uma aplicação Social Utilizando Estilos de Aprendizagem

Administração Utilizador: l.correia.go Logout

Carregar Questionário
Visualizar Questionário
Questionário Activo
Estatísticas

Seleccione o questionário que pretende visualizar:

Questionários:

ILSQ - Index of Learning Styles Questionnaire de Richard M. Felder e Barbara A. Soloman

Descrição: Este teste vai ajudá-lo a conhecer-se melhor e poderá dar-lhe indicações precisas de como aprender mais, melhor e mais depressa.

Estilos de aprendizagem a avaliar:

- **Global** - Prefere perceber primeiro o problema como um todo.
- **Verbal** - Prefere sons, palavras escritas ou faladas e fórmulas.
- **Intuitivo** - Prefere memórias e pensamentos.
- **Reflexivo** - Prefere a introspecção.
- **Sequencial** - Prefere progressões lógicas com passos incrementais.
- **Visual** - Prefere imagens, diagramas, gráficos e demonstrações.
- **Sensitivo** - Prefere imagens, sons e sensações físicas.
- **Activo** - Prefere actividades físicas ou discussão de ideias.

	G l o b a l	V e r b a l	I n t u i t i v o	R e f l e x i v o	S e q u e n c i a l	V i s u a l	S e n s i t i v o	A c t i v o
1. Compreendo melhor quando								
a) tento fazer.	-	-	-	-	-	-	-	1.0
b) penso.	-	-	-	1.0	-	-	-	-
2. Preferiria ser considerado								
a) realista.	-	-	-	-	-	-	1.0	-
b) inovador.	-	-	1.0	-	-	-	-	-

Figura 5.5: Visualização de um questionário

5.9.3 Selecção do questionário a disponibilizar

A selecção de questionários é feita por intermédio de uma lista com os questionários existentes na base de dados, através da qual o administrador selecciona aquele que pretende aplicar. Assim que se procede à selecção e confirmação do questionário pretendido, será dada a indicação de que a operação foi correctamente processada e este ficará disponível para os utilizadores da aplicação social. A página para selecção do questionário a disponibilizar pode ser vista na figura 5.6,

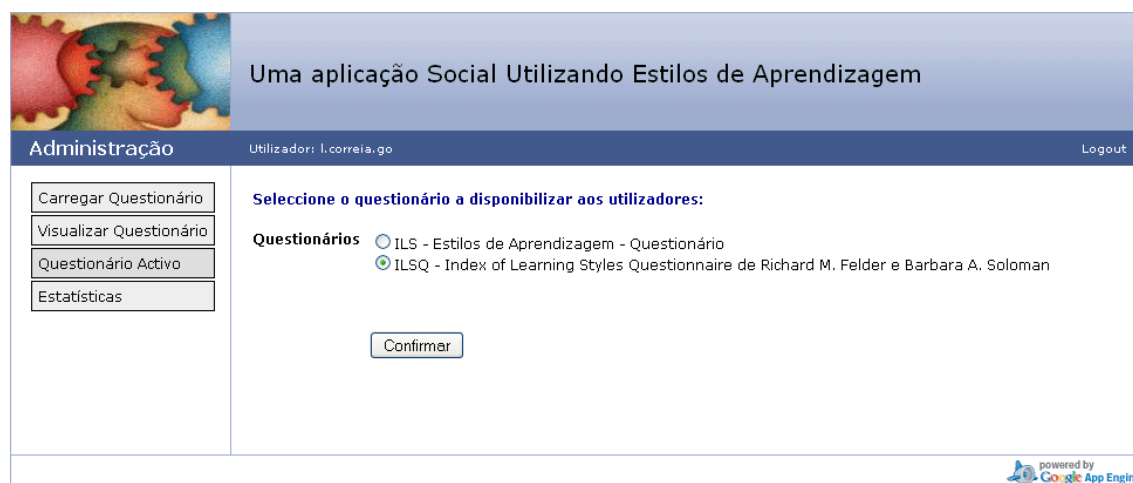


Figura 5.6: Selecção do questionário activo

5.9.4 Estatísticas

Actualmente apenas é possível a visualização do número de indivíduos que já realizaram determinado questionário e o somatório da pontuação obtida para cada um dos parâmetros avaliados no teste, como se pode ver na figura 5.7.

Uma aplicação Social Utilizando Estilos de Aprendizagem

Administração Utilizador: l.correia.go Logout

Carregar Questionário
Visualizar Questionário
Questionário Activo
Estatísticas

Selecione o questionário para o qual pretende ver as estatísticas:

Questionários Consultar

ILSQ - Index of Learning Styles Questionnaire de Richard M. Felder e Barbara A. Soloman

	G l o b a l	V e r b a l	I n t u i t i v o	R e f l e x i v o	S e q u e n c i a l	V i s u a l	S e n s i t i v o	A c t i v o
Pontuação	36.0	40.0	36.0	49.0	41.0	37.0	41.0	28.0

Responderam ao questionário 7 indivíduos.

powered by Google App Engine

Figura 5.7: Opção para visualização de estatísticas

Estas estatísticas poderão vir a ser alargadas de acordo com a informação que posteriormente se ache pertinente analisar.

5.10 Update da aplicação

Para disponibilizar o site na Internet, é necessário fazer o seu *update* para o servidor de produção, isto é, para a App Engine. Isto é feito por intermédio do comando `appcfg.py`. Antes de realizar o *update*, é necessário criar uma conta na Google App Engine e fornecer um contacto telefónico para que seja enviado por SMS um código de acesso. Esse código permite então a criação da aplicação.

O nome da aplicação a registar deverá ser o mesmo que consta em “Application”, no ficheiro `app.yaml`. A aplicação foi assim registada com o nome “learningstyle-sapp”.

Após o registo da aplicação e assumindo que esta se encontra localmente na pasta `d:\lsapp\`, para fazer o seu *update* basta digitar na linha de comando:

```
appcfg.py update d:\lsapp\learningstyle-sapp
```

Desta forma a aplicação fica disponível no endereço <http://learningstyle-sapp.appspot.com/>

5.11 Resumo e Conclusões

O SDK apesar de ser uma excelente ajuda na construção de aplicações que correram na Google App Engine, não impõe localmente limitações relativas a *time-outs* e a cotas. Isto leva a que problemas relacionados essencialmente com os *time-outs* apenas sejam detectados após o seu upload para a App Engine.

6 Detalhes e Resultados da Implementação da Aplicação Social

A aplicação social desenvolvida, denominada de “Estilos de Aprendizagem”, foi criada tendo em vista o *container* OpenSocial do Hi5. Esta corre totalmente do lado do cliente, havendo no entanto necessidade de aceder a servidores de dados para a leitura e gravação de informação. Utiliza ainda serviços externos para o desenho do gráfico, a apresentar aos utilizadores após a conclusão do questionário.

6.1 Ferramentas e processo de desenvolvimento

Após testada, concluiu-se que a ferramenta disponibilizada pela Google para a criação e teste de *gadgets*, o GGE não constituía uma opção viável, tanto pela sua baixa usabilidade como pela impossibilidade de testar toda a componente social.

Não foi encontrada qualquer ferramenta integrada que possibilitasse o desenvolvimento e teste integral de uma aplicação social. Foi então analisada a estrutura de um *gadget* social de modo a tomar uma decisão quanto ao processo de desenvolvimento a seguir. Constatou-se que um *gadget* é um ficheiro XML com a seguinte estrutura.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Module>
  <ModulePrefs title="Estilos de Aprendizagem">
  </ModulePrefs>
  <Content type="html">
    <![CDATA[
      <!-- Código da aplicação - HTML e JavaScript -->
    ]]>
  </Content>
</Module>
```

Para transformar um *gadget* num *gadget* social basta adicionar a *ModulePrefs*, a linha:

```
<Require feature="opensocial-0.8"/>
```

Neste caso concreto, o *gadget* passará a utilizar a versão 0.8 da API OpenSocial.

Detalhes e Resultados da Implementação da Aplicação Social

Como um *gadget* não é mais que um ficheiro escrito em HTML e JavaScript, inserido num bloco XML, constatou-se ser possível, retirando o bloco de XML, desenvolver praticamente toda a aplicação como sendo um simples ficheiro HTML.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<title>Estilos de Aprendizagem</title>
</head>
<body>
  <!-- Código da aplicação - HTML e JavaScript -->
</body>
</html>
```

Apesar de não ser igualmente possível proceder ao teste da componente social, este processo permite um desenvolvimento local da aplicação com todas as vantagens que daí advêm.

Decidiu-se então pelo desenvolvimento local durante a fase inicial, recorrendo apenas ao editor Notepad++, e ao *Web server* IIS.

Para o teste da componente social, recorreu-se à aplicação Dev App⁴⁸, que pode ser instalada na *sandbox* do Hi5. Desta forma, tornou-se possível a obtenção de resultados concretos da utilização da API OpenSocial no *container* ao qual se destina a aplicação. Após testado, foi sendo incorporado na aplicação desenvolvida localmente.

Para todas as componentes da aplicação poderem ser testadas em conjunto, tornou-se necessário copiar o código incluído no `<body>` do ficheiro HTML, para o ficheiro XML. Procedendo depois a pequenas alterações, ficando a aplicação pronta a ser testada no *container* do Hi5, ou em outro qualquer *gadget container*, caso assim se deseje. Com a ajuda do comando `appcfg.py`, torna-se bastante fácil o *update* da aplicação para a App Engine podendo desta forma ser testada. Este processo, apesar de mais moroso, permite testar o funcionamento completo da aplicação, tendo sido utilizado essencialmente durante a fase final do desenvolvimento.

6.2 Estrutura de uma aplicação social no Hi5

Para que uma aplicação corra no Hi5 é necessário adicionar ao `ModulePrefs` informação relativa ao *e-mail* do autor, uma descrição curta e outra mais pormenorizada, a localização para um *thumbnail* e para o um *icon* que identificará a aplicação.

Tal como pode ser visto em “Anatomy of a Hi5 Application”[hi508], o Hi5 possui 3 *views* possíveis para a aplicação. Existe por isso a necessidade de construir três páginas para cada uma dessas *views*, nomeadamente a página Profile, a página Preview e a página Canvas, sendo esta última a que possibilita a criação de páginas de maior dimensão e neste projecto a que permite o acesso a todas as funcionalidades. Cada uma destas páginas possui dimensões máximas bem definidas, estabelecidas por cada *container*.

O XML que constitui a aplicação social desenvolvida já com referência a todas as *views*, é apresentado de seguida.

48 <http://osda.appspot.com/>

Detalhes e Resultados da Implementação da Aplicação Social

```
<?xml version="1.0" encoding="UTF-8"?>
<Module>
  <ModulePrefs
    title="Estilos de Aprendizagem"
    summary="Conheça o seu estilo de aprendizagem predominante."
    description="Saiba como poderá aprender mais depressa
      lendo os conselhos que lhe são dados no final do teste
      e conheça os amigos com as mesmas preferências."
    author_email="l.correia.go@gmail.com"
    thumbnail="http://learningstyle-sapp.appspot.com/static/thumb.gif"
    icon="http://learningstyle-sapp.appspot.com/static/favicon.gif">
    <Require feature="dynamic-height"/>
    <Require feature="opensocial-0.8" />
    <Require feature="views" />
  </ModulePrefs>
  <Content type="html" view="profile">
    <![CDATA[
      <!-- Codigo correspondente à profile view - HTML e JavaScript -->
    ]]>
  </Content>
  <Content type="html" view="preview">
    <![CDATA[
      <!-- Codigo correspondente à preview view - HTML e JavaScript -->
    ]]>
  </Content>
  <Content type="html" view="canvas">
    <![CDATA[
      <!-- Codigo correspondente à canvas view - HTML e JavaScript -->
    ]]>
  </Content>
</Module>
```

6.3 Alojamento e testes

A aplicação social ficou alojada na Google App Engine, mais concretamente, na pasta /static da aplicação “learningstyle-sapp”. Desta forma o processo de teste da aplicação social na *sandbox* do Hi5 ficou bastante simplificado.

Para testar um qualquer *gadget* social na *sandbox* do Hi5 apenas é necessário efectuar o registo como programador a partir de uma conta previamente criada e adicionar a aplicação indicando o URL correspondente à sua localização, que neste caso é:

```
http://learningstyle-sapp.appspot.com/static/ls-sapp.xml
```

Uma vez instalada a aplicação na *sandbox* do Hi5, para testar qualquer alteração efectuada localmente apenas é necessário realizar o seu *update* para a App Engine, por intermédio do comando:

```
dev_appserver.py D:\lsapp\learningstyle-sapp
```

As alterações ficam então disponíveis no ambiente de desenvolvimento do Hi5.

6.4 Funcionamento da aplicação

Ao ser acedida, a aplicação lê a informação do perfil do utilizador actual, armazenada no Hi5 e verifica se este já realizou o questionário. Caso ainda não o tenha feito, será mostrada uma página onde consta uma breve descrição sobre a aplicação e a sua finalidade, como se pode ver na figura 6.1.

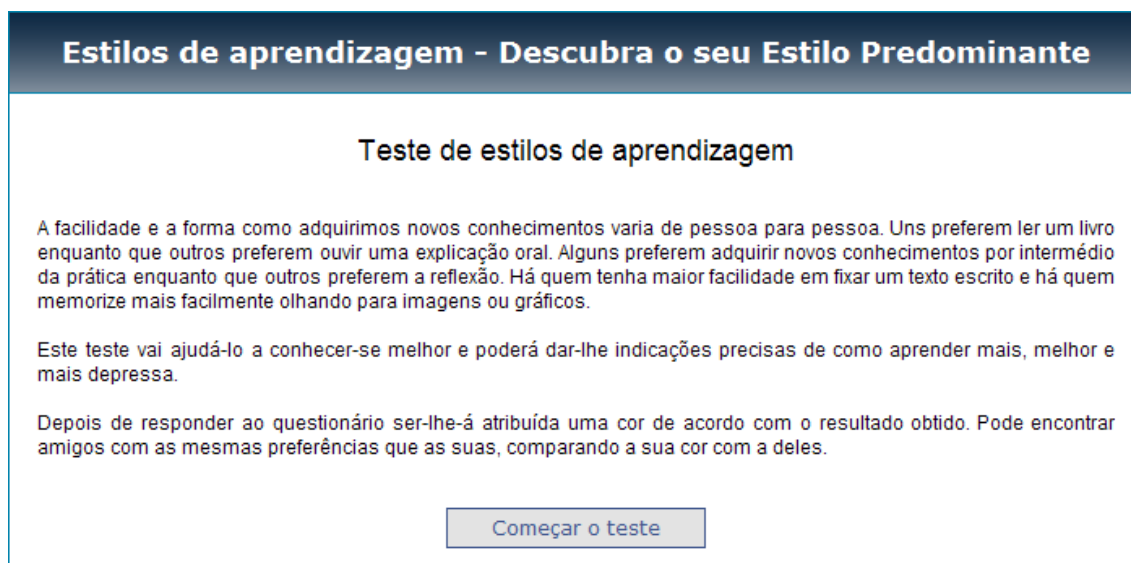


Figura 6.1: Página de entrada da aplicação social

Caso já tenha realizado o questionário, será mostrada uma página onde consta a cor atribuída ao utilizador, de acordo com os resultados obtidos, assim como a cor dos seus amigos que também já tenham respondido ao questionário como pode ser visto na figura 6.2. A partir desta janela torna-se possível visualizar os resultados detalhados do questionário previamente efectuado ou repetir esse questionário, caso assim o pretenda. Neste caso, os novos resultados substituirão os antigos previamente gravados na base de dados.

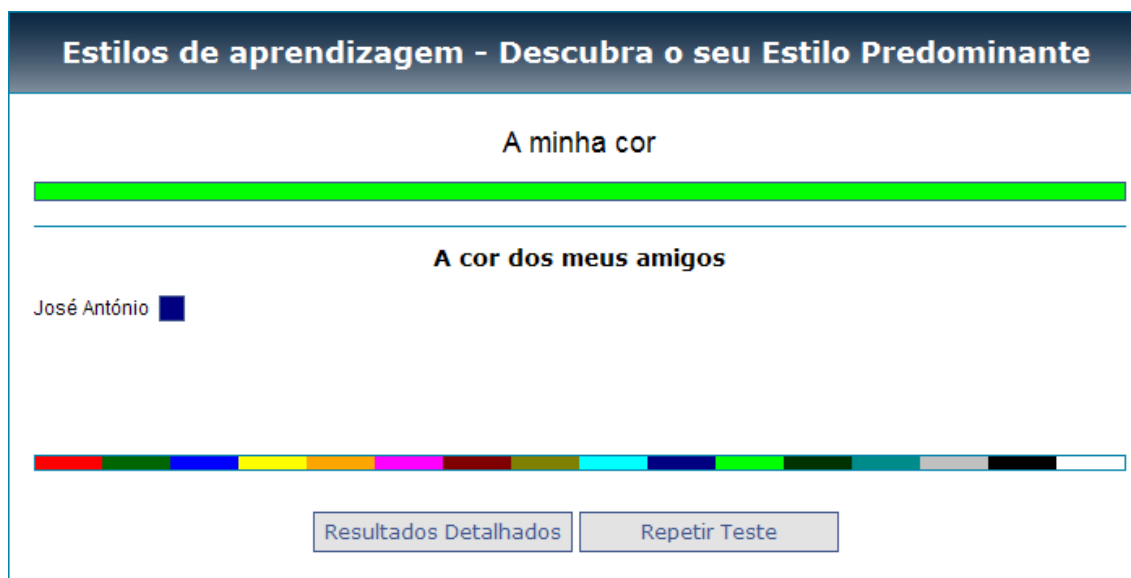


Figura 6.2: Página com informação das cores atribuídas

A oportunidade oferecida ao utilizador de realizar de novo o questionário, prende-se com o facto de se pensar que à primeira o possa fazer de uma forma pouco séria, escolhendo aleatoriamente as respostas para mais rapidamente o concluir e ver que tipo de resultados são

Detalhes e Resultados da Implementação da Aplicação Social

fornecidos e só então tomarem consciência de que poderá haver vantagens em responder de uma forma consciente.

Na realização do questionário, as perguntas são apresentadas ao utilizador uma de cada vez, como se pode ver na figura 6.3. O questionário é, no entanto, previamente carregado para a aplicação na sua totalidade. Isto para além de possibilitar o redimensionamento da página e da barra de progresso de acordo com o número de perguntas do questionário, facilita a implementação de toda a funcionalidade relativa à selecção e alteração das respostas.

Estilos de aprendizagem - Descubra o seu Estilo Predominante

Index of Learning Styles Questionnaire de Richard M. Felder e Barbara A. Soloman

39 - Para entretenimento, prefiro

a) ver televisão.

b) ler um livro.

89%

Pergunta Anterior Próxima Pergunta Cancelar

Figura 6.3: Página para resposta ao questionário

A barra de progresso à direita da página indica o posicionamento relativo da resposta que está a ser visionada em relação a todas as respostas do questionário.

A resposta a uma questão é feita clicando em cima da opção que se pretende seleccionar, obrigando depois o utilizador a clicar no botão “Próxima Pergunta” ou no botão “Pergunta Anterior” para poder visualizar uma nova pergunta. Pensa-se que desta forma o utilizador será menos tendencioso na sua escolha, pois o trabalho que tem a seleccionar uma ou outra é praticamente o mesmo.

O utilizador apenas poderá submeter o questionário se tiver respondido a todas as perguntas.

Os resultados do questionário são mostrados na forma de um gráfico do tipo radar e de diversas escalas, de acordo com os estilos de aprendizagem aferidos. São igualmente apresentadas as descrições resumidas de cada estilo, como se pode ver na figura 6.3.

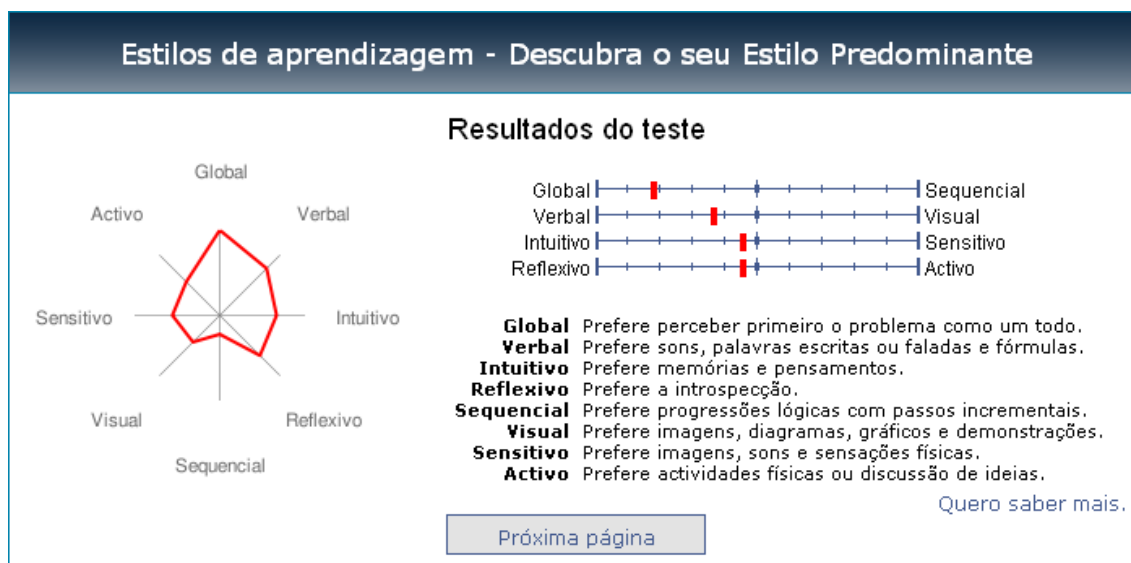


Figura 6.4: Página de resultados do teste

Para a criação do gráfico é utilizada a API Google Charts. O tipo de gráfico e a sua configuração são definidos passando os parâmetros adequados, juntamente com os dados relativos aos resultados obtidos no questionário. Esses resultados são convertidos previamente numa percentagem de modo a que o programa se adeque a qualquer questionário e a qualquer tipo de valores. O número de eixos do gráfico depende do número de estilos do teste que são também passados como parâmetros no URL atrás referido.

As escalas são igualmente dinâmicas, criadas em função dos estilos de aprendizagem do questionário e de acordo com os resultados obtidos no mesmo. Pelo motivo enunciado para os gráficos, antes de serem apresentados nas escalas, os valores são convertidos em percentagens. O número de escalas varia com os estilos de aprendizagem a avaliar. Ao contrário dos gráficos, não existe nenhuma API para a construção de escalas, tendo sido a sua criação desenvolvida especificamente para esta aplicação.

A posição tanto dos eixos do gráfico como das escalas, está dependente da ordem dos IDs dos estilos de aprendizagem no ficheiro XML que possui os dados do questionário e que é utilizado para a sua inserção na base de dados. Estes IDs são convertidos nos identificadores internos dos estilos e é por eles que estes são ordenados. Neste caso os estilos de aprendizagem foram ordenados no ficheiro XML como é apresentado de seguida.

```
<learningstyle id="ils_1_global" ls_dsc_curta="Prefere perceber primeiro
o problema como um todo.">Global</learningstyle>
<learningstyle id="ils_2_verbal" ls_dsc_curta="Prefere sons, palavras
escritas ou faladas e fórmulas.">Verbal</learningstyle>
<learningstyle id="ils_3_intuitivo" ls_dsc_curta="Prefere memórias e
pensamentos.">Intuitivo</learningstyle>
<learningstyle id="ils_4_reflexivo" ls_dsc_curta="Prefere a
introspecção.">Reflexivo</learningstyle>
<learningstyle id="ils_5_sequencial" ls_dsc_curta="Prefere progressões
lógicas com passos incrementais.">Sequencial</learningstyle>
<learningstyle id="ils_6_visual" ls_dsc_curta="Prefere imagens,
diagramas, gráficos e demonstrações.">Visual</learningstyle>
<learningstyle id="ils_7_sensitivo" ls_dsc_curta="Prefere imagens, sons e
sensações físicas.">Sensitivo</learningstyle>
<learningstyle id="ils_8_activo" ls_dsc_curta="Prefere actividades
físicas ou discussão de ideias.">Activo</learningstyle>
```

Detalhes e Resultados da Implementação da Aplicação Social

Clicando no botão “próxima página” da figura 6.4, acedemos à componente social da aplicação. Aí é apresentada a cor que o utilizador obteve, de acordo com o questionário realizado. A atribuição de uma cor processa-se do modo que se passa a explicar. Na aplicação existe um *array* com 16 cores. Cada elemento do *array* inclui um código, o nome e a própria cor, como se pode ver do fragmento de código mostrado a seguir.

```
cores = [{"id": "1357", "nomecor": "Vermelho", "cod": "#ff0000"},  
        {"id": "2357", "nomecor": "Verde", "cod": "#006600"}, ...]
```

Cada cor é atribuída de acordo com a posição em relação ao ponto médio das escalas, para cada um dos estilos de aprendizagem. Veja-se um exemplo: Se um utilizador após ter realizado o teste, os seus resultados apontam todos para o lado esquerdo das escalas, ou seja, os seus estilos predominantes são o Global, o Verbal, o Intuitivo e o reflexivo, então a cor a atribuir será o vermelho. Se por outro lado fosse Sequencial, Verbal, Intuitivo e Reflexivo, a sua cor já seria o Verde. Para 4 escalas, o número máximo de escalas permitidas pela aplicação, e que representam 8 estilos, existem 16 possíveis combinações, que é o número de cores do *array*.

As cores não possuem qualquer conotação. Este esquema simples de atribuição de cores foi adoptado em detrimento de outros mais complexos que fizessem variar a cor de acordo com uma determinada gama de modo a conseguir uma variação da cor em função da variação do resultado exacto, como por exemplo, vermelho mais claro ou mais escuro dependendo dos resultados serem mais à esquerda ou mais à direita dentro do mesmo grupo. Isto porque a ideia não consiste em criar um processo rigoroso de atribuição de cores, mas apenas estimular os utilizadores a realizarem o questionário e além disso, encontrarem pessoas com as quais tenham algumas afinidades. Isto só é possível se tivermos um esquema de cores muito limitado. As cores utilizadas foram escolhidas dentro das mais usuais, de modo a lhe podermos chamar um nome simples, como vermelho, azul, verde-claro, verde-escuro ou amarelo, evitando nomes menos usuais e não consensuais. Desta forma é mais fácil um indivíduo dizer com clareza, qual a cor que lhe foi atribuída.

Estilos de aprendizagem - Descubra o seu Estilo Predominante

Resultados do teste

A sua cor é o: ■


De acordo com os resultados obtidos no questionário, foi-lhe atribuída uma cor que no seu caso é o **Vermelho**.

Convide os seus amigos a realizarem este questionário e encontre aqueles que possuem as mesmas preferências. Basta verificar se a cor deles é igual à sua. Para convidar amigos clique no link **Convidar amigos**, que se encontra por cima da aplicação.

Amigos que já realizaram o teste

Ainda não existem amigos que já tenham realizado o teste. Convide-os para esta aplicação de forma a poder comparar a cor deles com a sua.

Amigos com a mesma cor significa amigos com preferências semelhantes.



[Início](#)

Figura 6.5: Visualização da cor atribuída de acordo com os resultados do teste

Detalhes e Resultados da Implementação da Aplicação Social

No final da realização do questionário, a cor obtida é gravada no perfil do utilizador, juntamente com o código do questionário realizado. Isto permite a utilização de questionários diferentes por parte da aplicação. Esta informação poderia ser gravada na base de dados juntamente com os resultados obtidos no teste. No entanto, ao guardá-la no perfil de cada utilizador, vem facilitar o seu acesso, principalmente quando necessitamos de aceder a essa informação para cada um dos perfis dos amigos de cada utilizador.

6.5 Acesso a dados externos

Para ler o questionário a partir da base de dados da App Engine, é utilizado o método “gadgets.io.RequestParameters” para efectuar um pedido à API JSON de modo a obter toda a informação com ele relacionada no formato JSON. A resposta é criada no ficheiro Python de acordo com o URL passado em gadget.io.makeRequest. Quando a informação é devolvida, é tratada pelo método “onLoadQuestionsFromDB”.

```
function loadQuestions() {
    var params = {};
    params[gadgets.io.RequestParameters.CONTENT_TYPE] =
        gadgets.io.ContentType.JSON;
    var url = GLOBAL_BASE_URL + 'questions';
    gadgets.io.makeRequest(url, onLoadQuestionsFromDB, params );
}
```

O processo de leitura dos resultados obtidos no teste é semelhante a este, diferindo apenas na constituição do URL.

Para escrever na base de dados, é utilizado o método “makeRequest” para efectuar um pedido do tipo POST onde são enviados os dados a gravar, que por sua vez serão gravados por intermédio do Python correspondente ao mapeamento do URL no ficheiro sappa.py. O fragmento do código que realiza esta operação é apresentado de seguida.

```
...
var params = {};
params[gadgets.io.RequestParameters.METHOD] = gadgets.io.MethodType.POST;
post_data = gadgets.io.encodeValues({
    'id_utilizador' : sender_id,
    'nome_teste' : nomeTeste,
    'id_lS_pontos' : aux_a });
params[gadgets.io.RequestParameters.POST_DATA] = post_data;
var url = GLOBAL_BASE_URL + 'questions';
gadgets.io.makeRequest(url, getGrafico(res, "grafico", "escala",
"descricoes"), params);
...
```

6.6 Persistência de dados do OpenSocial

O OpenSocial permite a leitura e gravação de dados do utilizador sem recorrer a servidores externos. O conjunto de funções que o torna possível tem a designação de OpenSocial's Persistence API[Roo08b]. Foi com recurso a esta API que se tornou possível gravar as cores no perfil de cada utilizador e posteriormente ler essas cores quer a que foi atribuída ao utilizador

Detalhes e Resultados da Implementação da Aplicação Social

logado, quer as cores de todos os seus amigos que tenham entretanto realizado o questionário e que mantenham a aplicação instalada.

O processo de escrita dados relacionados com a aplicação no perfil do utilizador é bastante simples, apresentando-se de seguida o fragmento de código utilizado com esse intuito.

```
...
var req = opensocial.newDataRequest();
req.add(req.newUpdatePersonAppDataRequest("OWNER", teste[0].nomeTeste +
                                          "Color", color));
req.add(req.newUpdatePersonAppDataRequest("OWNER", teste[0].nomeTeste +
                                          "ColorName", colorName));
req.send(handlePostInfoSocial);
```

A função “HandlePostInfoSocial”, que é executada após a gravação ou tentativa de gravação dos dados sociais, poderá ser utilizada para qualquer processamento necessário, como por exemplo, para o tratamento de erros.

A leitura dos dados dos amigos do utilizador actual é no entanto um pouco mais complexa como se pode ver pelo código seguinte

```
function requestMyDataFriends() {
    var params = {};
    params[opensocial.DataRequest.PeopleRequestFields.MAX] = 100;
    params[opensocial.DataRequest.PeopleRequestFields.FILTER] =
        opensocial.DataRequest.FilterType.HAS_APP;
    var req = opensocial.newDataRequest();
    var idSpec =
        opensocial.newIdSpec({"userId":"OWNER","groupId":"FRIENDS"});
    req.add(req.newFetchPeopleRequest(idSpec, params), "ownerFriends");
    req.add(req.newFetchPersonAppDataRequest(idSpec, teste[0].nomeTeste +
                                             'Color', params), 'cor');
    req.send(handleRequestMyDataFriends);
}
```

O tratamento da informação devolvida é efectuado em `handleRequestMyDataFriends`. Foram encontrados alguns problemas na implementação do método “`newFetchPersonAppDataRequest`” cujo comportamento, relativamente aos dados devolvidos não foi o esperado.

6.7 As views do Hi5

O Hi5 possui 3 *views* possíveis para a aplicação. Existe por isso a necessidade de construir três páginas para cada uma dessas *views*, nomeadamente a página Profile, a página Preview e a página Canvas, sendo esta última a de maior dimensão e neste projecto a que permite o acesso a todas as funcionalidades da aplicação. Estas páginas possuem dimensões máximas bem definidas e que podem ser encontradas em “Anatomy of a Hi5 Application”[hi508].

Para que a aplicação pudesse ser disponibilizada para cada uma das *views*, para além da página Canvas foram implementadas também a página Profile e a página Preview, diferindo estas duas, apenas na dimensão. Estas páginas limitam-se a apresentar a aplicação, resumindo-se a funcionalidade disponibilizada a um botão que conduz o utilizador à página Canvas. Na figura 6.6, pode ser vista a página Preview, tal como deverá aparecer a qualquer utilizador, antes de proceder à instalação da aplicação.



Figura 6.6: Vista da página Preview

6.8 Propagação da aplicação

O Hi5 possibilita que cada utilizador convide os seus amigos a utilizarem uma qualquer aplicação que tenham instalado, oferecendo para isso uma interface *standard*. Por esse motivo não se justificou a criação de qualquer interface proprietária para a realização da mesma tarefa, nem se sentiu necessidade de implementar qualquer outro processo recorrendo à API OpenSocial, para a divulgação da aplicação.

6.9 Resumo e Conclusões

A implementação desta aplicação social tornou-se mais complicada pela não existência de um ambiente de desenvolvimento local que permitisse testar todas as funcionalidades em conjunto. Foram sentidos alguns problemas relativamente à OpenSocial's Persistence API, responsável pela leitura e escrita de dados no perfil dos utilizadores, quando se tentou ler os dados dos amigos, ou seja, quando se tentou ler as cores atribuídas aos amigos de determinado utilizador. Apesar de ser previsível que ao pedir essa informação, os resultados se apresentassem já relacionados, ou seja, fossem devolvidos os pares, (utilizador, cor) de tal forma que para cada par a cor correspondesse ao utilizador, tal não se verificou. Na resposta, cada utilizador estava realmente associado a uma cor, mas essa cor poderia não ser a sua. Isto implicou um trabalho mais elaborado na leitura desta informação.

7 Conclusões e Trabalho Futuro

7.1 Satisfação dos Objectivos

O crescimento das redes sociais abriu um mundo de novas oportunidades e colocou ao alcance de todos os que desenvolvem aplicações para a *Web*, um número crescente de potenciais utilizadores. Munido das ferramentas próprias, cabe ao programador cativar esse publico de modo a que as suas aplicações tenham sucesso.

Para tirar partido destes novos ambientes surgiu a ideia da elaboração de uma aplicação que indagasse acerca das preferências sobre estilos de aprendizagem dos utilizadores destas redes e ao mesmo tempo tirasse partido das tecnologias emergentes associadas a esse meio.

Foi em primeiro lugar efectuado um estudo acerca da temática que envolve os estilos de aprendizagem, de modo a se obterem os conhecimentos mínimos necessários para a elaboração da aplicação social. Havia a necessidade de utilizar um questionário que aferisse de uma forma credível e séria os estilos de aprendizagem dos internautas. A possibilidade desse questionário ser desenvolvido de raiz foi posta de parte pois para isso seriam necessários conhecimentos técnicos especializados nessa área. Foi por isso efectuada uma pesquisa para encontrar o questionário certo para ser utilizado com a aplicação. Um dos questionários mais conhecidos mundialmente e que chegou a ser uma hipótese séria, foi o Learning Style Inventory de David A. Kolb, baseado na sua teoria, Experiential Learning. No entanto, resultado da análise efectuada, foi encontrado um outro questionário igualmente credível e amplamente utilizado, o Index of Learning Styles desenvolvido por Richard M. Felder e Barbara A. Soloman, que apesar de possuir um maior número de questões, é bastante fácil de responder. Para além disso, este questionário encontra-se acessível na internet para utilização livre tanto por parte de quem quiser aferir as suas próprias preferências a nível dos estilos de aprendizagem, como para a sua aplicação para fins científicos e não comerciais. Foi então este o questionário escolhido de entre todos os analisados.

A ideia de construir uma aplicação que possibilitasse a aplicação de apenas um questionário é bastante redutora. Rapidamente se concluiu que havia todo o interesse que fosse possível a aplicação de qualquer tipo de questionário, desde que obedecesse a determinados pressupostos. Desta forma, seria fácil desenvolver um novo questionário, ou mesmo escolher um outro de entre os já existentes e aplicá-lo. Foi então pensada uma estrutura que comportasse questionários com um qualquer número de perguntas, um qualquer número de respostas e para cada resposta possibilitasse a atribuição de um valor numérico para cada um dos estilos de aprendizagem a aferir. Este esquema matricial vem assim cobrir um elevado número de casos.

Conclusões e Trabalho Futuro

Para a gestão dos questionários foi criada uma aplicação *Web* de acesso restrito, funcionando como *backoffice* da aplicação social. Este site de administração passou a fornecer mecanismos não só para inserção e manuseamento de questionários, mas também para a visualização de informações relativas às respostas obtidas pela aplicação dos mesmos.

Decidiu-se pela utilização de um ficheiro XML para criar e carregar os questionários na base de dados. Com recurso a um DTD torna-se possível efectuar a sua validação, antes de os inserir na aplicação. A implementação de um editor de questionários foi posta de parte para esta fase, devido essencialmente a limitações de tempo.

A criação do site de administração implicou algumas decisões relativamente às tecnologias a utilizar na sua concepção. Se por um lado o mais óbvio seria a utilização de PHP juntamente com uma base de dados relacional, como por exemplo o MySQL, por outro achou-se por bem indagar sobre novas ferramentas existentes para a concretização da mesma tarefa. Uma dessas tecnologias que chamou a atenção não só pela sua novidade mas também pelas características anunciadas foi a Google App Engine que acabou por ser a plataforma utilizada tanto para o desenvolvimento e alojamento do site de administração como para o alojamento da aplicação social.

A aplicação social desenvolvida é um Social Gadget que utiliza a nova versão 0.8 da API OpenSocial, criada tendo em vista o Hi5. Esta aplicação disponibiliza um questionário sobre estilos de aprendizagem e apresenta ao utilizador os resultados nele obtidos. Pensando na facilidade de leitura dos resultados, estes são apresentados por intermédio de um gráfico do tipo radar. Para além disso, as pontuações obtidas são também esquematicamente apresentadas em diversas escalas, que variam em número de acordo com o questionário carregado na aplicação. Desta forma a percepção dos resultados processa-se de forma imediata e sem grande esforço. Uma curta descrição é também apresentada para que cada um fique a perceber melhor os resultados obtidos. Para além disso é dada a possibilidade aos mais interessados de aceder a informação mais completa sobre os estilos de aprendizagem. A apresentação dos resultados processa-se de forma dinâmica, variando de acordo com o questionário disponibilizado e por sua vez, com os parâmetros a avaliar.

O principal desafio na elaboração da aplicação social, consistiu em descobrir um meio de tornar aliciente a resposta ao questionário, assim como incentivar as pessoas a darem-no a conhecer aos seus amigos, de modo a que este seja respondido pelo maior número possível de internautas. Quanto ao incentivo à propagação da aplicação, foi pensado num sistema que tira partido do ambiente onde esta está inserida e conseqüentemente da API OpenSocial e que consiste num sistema de atribuição de cores aos utilizadores. A cada utilizador é atribuída uma cor após a realização do questionário e de acordo com os resultados nele obtidos. Se dois amigos possuírem resultados semelhantes, aos dois será atribuída a mesma cor. Isto quer dizer que podem ter algo mais em comum para além das mesmas preferências relativamente aos estilos de aprendizagem. Pensa-se que este sistema possa vir a contribuir para a rápida difusão da aplicação. Poderá também contribuir para que o utilizador responda de uma forma mais consciente às questões que lhe são colocadas de modo a poder realizar comparações mais credíveis com os elementos da sua lista de amigos.

No momento presente, este sistema não foi ainda testado, encontrando-se em fase de validação por parte da equipa do Hi5. Só após a sua aprovação poderá integrar a biblioteca de aplicações, sendo esta a única possibilidade de tornar o programa acessível a todos os utilizadores desta rede social.

Relativamente às tecnologias utilizadas, constatou-se que a Google App Engine é uma boa plataforma para desenvolvimento de aplicações *Web*, encerrando um conjunto de funcionalidades e serviços que a tornam completa evitando a necessidade de recorrer a quaisquer recursos externos. No entanto, existe a necessidade de adaptação a um modelo não relacional e proprietário de base de dados, que apesar disso, é anunciado como muito eficiente e ajustável às necessidades crescentes de recursos por parte das aplicações que o utilizam. O sistema de cotas imposto pela App Engine, pode ser um factor determinante na adopção ou não desta plataforma. O principal problema manifesta-se ao nível do tempo de resposta exigido por

cada pedido http, que pode variar ainda com a carga do sistema. Se por um lado este factor pode contribuir para que o programador optimize a sua aplicação de modo a conseguir tempos mais baixos de resposta, por outro lado pode inviabilizar a sua utilização, principalmente no caso de sites que necessitem de realizar algum processamento *batch*. O SDK disponibilizado aos programadores para o desenvolvimento de aplicações para esta plataforma, que constitui uma preciosa ajuda, não possui no entanto implementado qualquer sistema de cotas ou de limitações o que leva a que problemas de *timeouts* só se venham a detectar quando a aplicação se encontra em ambiente de produção.

Para o desenvolvimento da aplicação social revelaram-se mais dificuldades, pois não foi encontrada qualquer ferramenta que permitisse o desenvolvimento deste tipo de programas e que ao mesmo tempo possibilitasse um ambiente local de testes o que viria agilizar todo o processo. O editor de *gadgets* GGE disponibilizado pela Google, apesar de ser uma ferramenta útil para testar pequenos *gadgets*, não se revelou uma opção viável para o desenvolvimento de aplicações e ao mesmo tempo não permite o teste de toda a componente social. Por isso optou-se por desenvolver a aplicação localmente, mantendo a componente social *hardcoded* durante este processo. Esta componente foi testada com recurso à aplicação Dev App desenvolvida para o efeito e que é possível instalar no Hi5. Na fase final de desenvolvimento passou-se a testar a aplicação já no Hi5, havendo para isso a necessidade de efectuar o *update* de todo o código para o servidor externo, ou seja para a App Engine, sempre que se efectuava qualquer alteração.

O resultado final foi um sistema funcional e fácil de utilizar que se adapta a um grande número de questionários, que permite aferir diversas preferências dos utilizadores, de acordo com o questionário a aplicar e que implementa um mecanismo que se pensa vir a contribuir para a sua rápida difusão pela rede social.

7.2 Trabalho Futuro

O preenchimento de questionários é sempre uma tarefa pouco aliciante à qual a maioria das pessoas tenta fugir, a não ser, claro, quando essas pessoas são as principais interessadas nos seus resultados. Qualquer questionário sério, que contribua para um melhor auto-conhecimento e para uma melhoria das prestações de cada indivíduo, quer a nível profissional quer a nível pessoal, representa por si só um incentivo à sua realização. No entanto, a componente visual é cada vez mais importante nos dias que correm. Qualquer aplicação que seja graficamente mais apelativa atrairá mais a atenção dos internautas. Este seria um dos aspectos a ter em conta para uma nova versão, tanto ao nível da apresentação dos conteúdos como da própria interacção com o programa, como por exemplo, nas respostas ao questionário, apesar de se achar bastante simples e prático o método já disponibilizado. Devido ao facto de ser oferecida a possibilidade ao utilizador de não responder a todas as perguntas de uma forma sequencial, podendo deixar respostas para mais tarde, poderiam fazer parte desse melhoramento gráfico mecanismos auxiliares de modo a indicar o número de respostas já dadas ou melhor ainda, quais as perguntas já respondidas e quais as que ainda se encontram por responder, assim como uma forma de navegar directamente para uma determinada pergunta.

O objectivo desta aplicação não é apenas o de possibilitar que as pessoas se conheçam melhor, mas sim dar a conhecer ao meio científico os resultados desses questionários para que possam ser analisados e para que possam vir a ser tiradas conclusões proveitosas desse conhecimento, tirando partido do crescente número de utilizadores das redes sociais.

Neste sentido, para além de ser fundamental tornar os questionários mais apelativos, é necessário dotá-los de processos que facilitem a sua divulgação ao maior número possível de pessoas que por sua vez o venham a responder. Foi por isso implementado um mecanismo de atribuição de cores segundo o qual se incentivam as pessoas a dar a conhecer o questionário aos seus amigos de modo a poderem descobrir possíveis afinidades. Apesar do mecanismo se encontrar implementado, existe mais uma vez a necessidade de tornar a apresentação desses

resultados mais apelativa. Outras funcionalidades a implementar poderiam passar pela visualização das cores dos amigos dos amigos, a possibilidade de agrupar amigos com a mesma cor ou a possibilidade de enviar automaticamente, após a realização do teste, uma mensagem a todos os amigos que obtiveram a mesma cor. Uma forma de universalizar a aplicação terá necessariamente que passar pela disponibilização de suporte multilíngue.

Um outro aspecto a melhorar tem a ver com as *views* disponibilizadas pelo *container*. No Hi5, para além da Canvas *view*, a *view* principal na qual a aplicação se apresenta com todas as suas funcionalidades, existem ainda a Preview *view* e a Profile *view*. Todas estas *views* têm dimensões máximas que têm que ser respeitadas e destinam-se a localizações diferentes no *Web site*. Na Canvas *view*, como é a principal, os programadores têm a possibilidade de usufruir de um tamanho maior para disponibilizar as funcionalidades da aplicação. A Preview *view*, possui dimensões mais reduzidas que a Canvas *view* sendo esta a *view* que é mostrada ao utilizador antes de dar a ordem de instalação da aplicação. Por fim, a Profile *view* destina-se à página de perfil do utilizador e é aquela para a qual é exigida a menor dimensão. Devido ao facto de se ter adoptado para a Canvas *view* uma dimensão superior ao limite máximo permitido para a Preview *view*, introduziu a necessidade de criar uma interface diferente para cada uma destas *views*. Foram por isso criadas duas interfaces minimalistas, onde se faz uma pequena apresentação da aplicação e se disponibiliza um botão que leva o utilizador à Canvas *view*. Como se compreende, esta não será a implementação ideal, existindo por isso a necessidade de as prover de um maior número de funcionalidades.

Relativamente ao site de administração, a principal melhoria passaria pela construção de um editor de questionários de modo a tornar mais fácil e sua inserção e edição. Este editor teria que ser suficientemente flexível para permitir a inserção de diferentes tipos de questionários.

O módulo de estatísticas fornece uma informação mínima, relativamente aos resultados obtidos pelos utilizadores. Esta área poderá ser complementada com outras estatísticas que se venham a tornar úteis.

O sistema desenvolvido poderia ser utilizado para administração de questionários com base em amostras populacionais mais selectivas. Para tal, torna-se indispensável poder disponibilizá-lo em servidores próprios, onde possa haver um maior controlo de quem tem acesso à aplicação. Neste sentido poderia ser interessante a implementação de um *container* OpenSocial que poderia servir não só esta aplicação mas também outras que pudessem vir a tirar partido desse ambiente.

Referências

- [Bla08] Ignacio Blanco. Leveraging Google Data APIs in OpenSocial Apps. Agosto de 2008. <http://code.google.com/intl/pt/apis/opensocial/articles/authsub.html>, visitado pela última vez em 2 de Março de 2009.
- [Bro03] Van den Broeck H., Vanderheyden, K., Cools E.. The field of cognitive styles: from a theoretical review to the construction of the cognitive style inventory. Vlerick Leuven Gent Management School, 2003.
- [Cha06] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, et al. Bigtable: A Distributed Storage System for Structured Data. Google, Inc., Novembro 2006.
- [Dun79] Rita S. Dunn and Kenneth J. Dunn. Learning styles/teaching styles: should they...can they...be matched?. in *Educational Leadership*, pp. 238-244. Janeiro 1979.
- [Fel05] Richard M. Felder, Rebecca Brent. Understanding Student Differences in *Journal of Engineering Education*, pp. 57-72, 2005.
- [Goo09a] Google. Gadgets-API Overview. <http://code.google.com/apis/gadgets/docs/overview.html>, visitado pela última vez em 2 de Março de 2009.
- [Goo09b] Google. Anatomy of an iGoogle Gadget. <http://code.google.com/intl/pt/apis/igoogle/docs/anatomy.html>, visitado pela última vez em 2 de Março de 2009.
- [Goo09c] Google. Which API Should I Use?. http://code.google.com/intl/pt/apis/gadgets/devguide_landing.html, visitado pela última vez em 2 de Março de 2009.
- [Goo09d] Google. What are the Google Data APIs?. <http://code.google.com/apis/gdata/>, visitado pela última vez em 2 de Março de 2009.
- [Goo09e] Google. The Python Runtime Environment. <http://code.google.com/appengine/docs/python/>, visitado pela última vez em 2 de Março de 2009.
- [Goo09f] Google. OpenSocial API Documentation. <http://code.google.com/apis/opensocial/docs/index.html>, visitado pela última vez em 2 de Março de 2009.
- [Goo09g] Google. Google App Engine - Queries and Indexes. <http://code.google.com/appengine/docs/python/datastore/queriesandindexes.html>, visitado pela última vez em 2 de Março de 2009.
- [hi508] hi5. Anatomy of a hi5 Application. <http://www.hi5networks.com/platform/wiki/Anatomy>, visitado pela última vez em 2 de Março de 2009.
- [Kol05] Alice Y. Kolb, David A. Kolb. *The Kolb Learning Style Inventory—Version 3.1 Technical Specifications*. Maio 2005.
- [Kol84] David A. Kolb. *Experience as the source of learning and development*. Prentice Hall, 1984.

Conclusões e Trabalho Futuro

- [Mar09] Marktest. Google.pt lidera em 2008, Janeiro 2009. http://www.marktest.pt/produtos_servicos/Netpanel/default.asp?c=1292&n=1961 visitado pela última vez em 2 de Março de 2009.
- [Ope09] OpenSocial.org. Container Information. http://wiki.opensocial.org/index.php?title=Main_Page#Container_Information, visitado pela última vez em 2 de Março de 2009.
- [Pyt08] Python. Python 3.0. <http://python.org/download/releases/3.0/>, visitado pela última vez em 2 de Março de 2009.
- [Roo08a] Arne Roomann-Kirruk and Lane LiaBraaten . OpenSocial Across Containers. Maio de 2008. <http://sites.google.com/site/io/opensocial-across-containers>, visitado pela última vez em 2 de Março de 2009.
- [Roo08b] Arne Roomann-Kurrik. The Persistence API. Agosto de 2008. <http://code.google.com/apis/opensocial/articles/persistence-0.8.html>, visitado pela última vez em 2 de Março de 2009.
- [Tan04] Kimberly Tanner, Deborah Allenz. Approaches to Biology Teaching and Learning: Learning Styles and the Problem of Instructional Selection-Engaging All Students in Science Courses. in American Society for Cell Biology, pp. 197-201, 2004.