

Universidade do Minho
Conselho de Cursos de Engenharia
Licenciatura em Engenharia Informática

Laboratórios de Informática IV - Projecto

Ano Lectivo de 2007/08



escola de engenharia



departamento de
informática

Gestão de Horários

Igor Brás (43143), Ricardo Vilas Boas (43184), Jorge Moreira (43096)

Julho, 2008

Data de Recepção	
Responsável	
Avaliação	
Observações	

Gestão de Horários

Igor Brás (43143)

Ricardo Vilas Boas (43184)

Jorge Moreira (43096)

Julho, 2008

Resumo

Este relatório visa documentar, relatar os procedimentos adoptados e as decisões tomadas na elaboração do projecto da disciplina de Laboratórios IV da Licenciatura em Engenharia Informática, "Gestão de horários".

Ao longo do documento serão abordados, de forma aprofundada, todos os conceitos envolvidos no desenvolvimento do trabalho.

Nos dias de hoje, é notória a necessidade de aplicações Web com elevado desempenho, seguras e a confiança na tecnologia é um factor fulcral. Neste contexto, surge o interesse pela **Web 2.0** uma vez que responde a todos esses requisitos, diminuindo a complexidade e tempo de desenvolvimento.

Face a isto, decidimos escolher um projecto que englobasse esta temática e dessa forma introduzir no nosso percurso académico um conjunto de tecnologias cada vez mais importantes no desenvolvimento de aplicações Web com elevada usabilidade e simplicidade. No entanto, esse desenvolvimento abrange vários níveis de complexidade, desde uma interacção ao nível do sistema de gestão de base de dados à estruturação da aplicação Web de fácil utilização, sendo por isso tão importante um conhecimento sólido nestas tecnologias para uma fácil inserção no mercado de trabalho.

A temática escolhida tem como objectivo permitir um estudo o mais abrangente possível utilizando a tecnologia **Web 2.0**. Para isso foi feita uma análise e caracterização do caso em estudo proposto, seguido da implementação que abrange o desenvolvimento Web e adicionalmente uma questão tão crítica como a da segurança.

Área de Aplicação: Desenvolvimento de uma aplicação Web com ligação a um sistema de gestão de base de dados MySQL.

Palavras-Chave: Bases de Dados Relacionais (MySQL), Web 2.0, PHP, Ajax, CSS, XHTML, JavaScript.

Conteúdo

Resumo	i
Conteúdo	ii
Lista de Figuras	iii
1 Introdução	1
1.1 Contextualização	1
1.2 Apresentação do Caso de Estudo	1
1.3 Motivação e Objectivos	2
1.4 Estrutura do Relatório	2
2 Tecnologias Aplicadas	3
2.1 Web 2.0	3
2.2 XHTML	4
2.3 PHP	4
2.4 Ajax	5
2.5 CSS	6
2.6 JavaScript	6
2.7 RSS	7
3 Sistema de Gestão de Base de Dados	8
4 Case-Study	12
4.1 Análise de Requisitos	12
4.2 Funcionalidades do Sistema	13
5 Detalhes da Implementação	15

5.1	Consulta	16
5.1.1	Diferenças de Visualização	20
5.1.2	Colisões de Horário	20
5.1.3	Horário Personalizado	21
5.2	Edição	21
5.2.1	Adicionar Aula	21
5.2.2	Editar aula	22
5.2.3	Eliminar Aula	22
5.3	Autenticação	22
5.4	Exportação	23
5.5	Impressão	25
5.6	RSS Feeds	25
5.7	Estilos com CSS	26
6	Resultados Obtidos	28
6.1	Consulta de horários	29
6.2	Autenticação e <i>Info bubble</i>	33
6.3	Edição	35
6.4	Mensagens de Aviso	37
6.5	Exportação	39
7	Conclusões e Trabalho Futuro	40
	Bibliografia	42
	Referências WWW	44
	Lista de Acrónimos	46

Lista de Figuras

4.1	Diagrama de Use-Case.	13
5.1	Diagrama da arquitectura.	15
5.2	Aula com mais do que uma hora.	17
6.1	Visualização de um horário(sem autenticação).	29
6.2	Visualização de um horário(com autenticação).	30
6.3	Horário de docente.	31
6.4	Horário por sala.	32
6.5	Interface de <i>Login</i>	33
6.6	<i>Info bubble</i>	34
6.7	<i>Edit bubble</i>	35
6.8	Interface para adição de uma aula.	36
6.9	Interface para remoção de uma aula.	36
6.10	Aviso de colisão (quando existem incompatibilidades no horário).	37
6.11	Mensagem de aviso (Exemplo 1).	38
6.12	Mensagem de aviso (Exemplo 2).	38
6.13	Interface de exportação do horário.	39

1 Introdução

1.1 Contextualização

Num mundo cada vez mais dominado pelas tecnologias de informação, é notória a necessidade de aplicações, seguras e com elevado desempenho, em que a tecnologia é fiável e o seu desenvolvimento envolve poucos recursos. O desenvolvimento e manutenção de aplicações oferece grandes desafios, como por exemplo, a estruturação da lógica da aplicação, a integração no sistema de gestão de base de dados e segurança.

Neste contexto, surge um conceito vasto e poderoso de desenvolvimento, **Web 2.0**, que utiliza várias ferramentas de desenvolvimento permitindo responder aos requisitos enumerados anteriormente e que para além disso, oferece ao utilizador final uma aplicação de grande usabilidade. Com esta tecnologia, o programador pode trabalhar os problemas, por natureza, bastante complexos, de forma estruturada, simplificando o processo de desenvolvimento deste género de aplicações, obtendo melhores resultados e melhor desempenho.

1.2 Apresentação do Caso de Estudo

Actualmente, os horários dos cursos de informática e dos docentes do departamento de informática são gerados para folhas Excel a partir de uma base de dados MySQL. A aplicação utilizada é bastante lenta, o que dificulta o processo de construção interactiva dos horários. Com este projecto numa primeira fase, pretende-se construir uma aplicação Web que permita visualizar os horários de forma atractiva para o utilizador **WWW** comum. Pretende-se também que haja a possibilidade da aplicação exportar os horários para um formato standard de eventos por forma a poderem ser importados numa agenda electrónica. A opção de edição, surge numa segunda fase, em que se cria a possibilidade de expansão da aplicação.

1.3 Motivação e Objectivos

Uma vez que as temáticas abordadas ao longo deste documento não foram abordadas ao longo do nosso percurso académico, o desenvolvimento deste projecto despertou-nos um grande interesse por podermos ter contacto com um conjunto de tecnologias que está em voga no desenvolvimento de aplicações Web e para podermos ganhar mais experiência no desenvolvimento de futuras aplicações.

Os objectivos de desenvolvimento deste trabalhos são os seguintes:

- Explorar o conceito de **Web 2.0** para elaboração de aplicações com elevada usabilidade.
- Exploração das várias tecnologias que fazem parte da **Web 2.0**, e necessárias ao desenvolvimento Web, tais como **XHTML**, **PHP**, **Ajax**, **CSS** e **JavaScript**.

O desenvolvimento deste projecto, permitirá o estudo de soluções tecnológicas existentes que permitam a implementação da aplicação em causa, fazendo deste projecto um ponto de referência a projectos futuros nesta área.

Em termos de implementação, deverá ser conseguida uma aplicação que realize as funcionalidades requeridas e onde sejam correctamente aplicadas as tecnologias necessárias, tendo sempre em vista as características de robustez, segurança e capacidade de evolução.

1.4 Estrutura do Relatório

Numa primeira fase são desdobradas noções básicas sobre **Web 2.0**, explicando o seu conceito, mais especificamente as tecnologias Web base a esse conceito, como **XHTML**, **PHP**, **Ajax**, **CSS** e **JavaScript**.

Depois de descrita a análise de requisitos da aplicação (envolvendo análise e caracterização do caso em estudo) é feita uma referência bastante vinculativa no que diz respeito à arquitectura da aplicação a desenvolver, permitindo perceber todos os componentes e tecnologias envolvidas neste projecto. Finalmente, é feita uma ilustração aos resultados obtidos e são retiradas as devidas conclusões.

2 Tecnologias Aplicadas

2.1 Web 2.0

O termo **Web 2.0** foi usado pela primeira vez em Outubro de 2004 pela *O'Reilly Media* e pela *MediaLive International* como nome de uma série de conferências sobre o tema, popularizando-se rapidamente a partir de então. Tratou-se de uma constatação, que as empresas que conseguiram manter-se através da crise da Internet possuíam características comuns entre si, o que criou uma série de conceitos agrupados que formam o **Web 2.0**.

As regras a que se refere *O'Reilly* já foram discutidas antes do surgimento do termo, sob outros nomes como *infore*, *the internet operating system* e *the open source paradigm shift* e são produto de um consenso entre empresas como Google, Amazon, Yahoo e Microsoft e estudiosos da Web (como *Tim O'Reilly*, *Vicent Cerf* e *Tim Berners-Lee*) e da consolidação do que realmente traz resultado na Internet. Segundo *Tim O'Reilly*, a regra mais importante seria desenvolver aplicativos que aproveitem os efeitos da rede que se tornam cada vez melhores, com a utilização do indivíduo comum, aproveitando a inteligência colectiva.

Iniciou-se o desenvolvimento de software que é usado na Internet e comercializado como serviços, pagos mensalmente. Além disso, mudou-se a forma de fazer software. Para que tudo funcionasse bem na Internet, foi necessário a união de várias tecnologias (como **Ajax**) que tornassem a experiência do utilizador mais rica, com interfaces rápidas e muito fáceis de usar. Definiu-se então que quanto mais simples e modular a programação fosse, melhor era o resultado. Assim é fácil tirar ou acrescentar uma funcionalidade ou compartilhar uma parte do software com outro software. Os módulos podem ser reutilizados em diversos tipos de software ou compartilhados para serem usados por programas de terceiros.

A Web 2.0 propõe uma experiência de uso semelhante à de aplicativos para desktop, frequentemente fazendo uso de uma combinação de tecnologias surgidas no final da década de 1990, que incluem **Web services APIs** (1998), **Ajax** (1998), **Web syndication** (1997), entre outras. Estas tecnologias aumentaram a velocidade e a facilidade de uso de aplicativos Web, sendo responsáveis por um aumento significativo no conteúdo (colaborativo ou meramente expositivo) existente na Internet. Permitiu ainda o desenvolvimento de interfaces ricas, completas e funcionais, sendo que alguns aplicativos Web, ainda em versão beta, são considerados por muitos como "desktops on-line", proporcionando ao utilizador um ambi-

ente de trabalho inteiramente baseado na **WWW**, acessível de qualquer computador com conexão à Internet. De forma particular, o **Ajax** permite ao utilizador não esperar que uma página Web recarregue ou que o processo seja terminado para continuar a usar o software. Cada informação é processada separadamente, de forma assíncrona, de forma que não é mais necessário recarregar a página a cada clique.

Na **Web 2.0** os programas de software funcionam pela Internet, não somente instalados no computador local, de forma que vários programas podem integrar-se formando uma grande plataforma. Por exemplo, os contactos do programa de e-mail podem ser usados no programa de agenda, ou pode-se criar um novo evento numa agenda através do programa de e-mail. Outro conceito da **Web 2.0** que interfere na programação chama-se "Beta perpétuo". Na **Web 2.0** acabaram-se os ciclos de lançamento de programas. Os programas são corrigidos, alterados e melhorados durante todo o tempo, e o utilizador participa deste processo dando sugestões, reportando erros e aproveitando as melhorias constantes. Em oposição ao que acontece com o software tradicional, em caixas, com instaladores e dependentes de um sistema operativo, aplicativos Web podem ser actualizados de forma constante, linear e independente da acção do utilizador final.

2.2 XHTML

O **XHTML**, ou "eXtensible Hypertext Markup Language", é uma reformulação da linguagem de marcação **HTML** baseada em **XML**. Combina as tags de marcação **HTML** com regras do **XML**. Este processo de padronização tem em vista a exibição de páginas Web em diversos dispositivos (televisão, telemóvel, etc). A intenção é melhorar a acessibilidade. O **XHTML** consegue ser interpretado por qualquer dispositivo, independentemente da plataforma utilizada, pois as marcações possuem sentido semântico para as máquinas. O **HTML** não consegue esta implementação. No entanto, não existem muitas diferenças entre o **HTML** e **XHTML**. Para verificar se uma página **XHTML** está bem construída, o melhor método é validar o código através de uma aplicação Web disponibilizada pela organização **W3C**.

2.3 PHP

PHP ou "Hypertext Preprocessor", é uma linguagem de programação interpretada, livre e muito utilizada para gerar conteúdo dinâmico na **WWW**. O **PHP** é uma poderosa linguagem orientada a objectos.

O **PHP 5**, trabalha com handlers. Caso se copie um objecto, na verdade copia-se um apontador, pois, caso haja alguma mudança na versão original do objecto, todas as outras também sofrem a alteração, o que não acontecia na **PHP 4**. Trata-se de uma linguagem extremamente modularizada, o que a torna ideal para instalação e uso em servidores Web.

É muito parecida, em tipos de dados, sintaxe e mesmo funções, com a linguagem **C** e com a **C++**. Pode ser, dependendo da configuração do servidor, embarcada no código **HTML**. Construir uma página dinâmica baseada em bases de dados é simples com **PHP**, pois, este prevê suporte a um grande número de bases de dados. O **PHP** tem suporte aos protocolos: IMAP, SNMP, NNTP, POP3, HTTP, LDAP, XML-RPC, SOAP. É ainda possível abrir sockets e interagir com outros protocolos. E as bibliotecas de terceiros expandem ainda mais estas funcionalidades. O seu propósito principal é de implementar soluções web velozes, simples e eficientes.

Características:

- Velocidade e robustez.
- Orientação ao objecto.
- Portabilidade - independência da plataforma.
- Tipagem fraca.
- Sintaxe similar à linguagem **C/C++** e o **PERL**

2.4 Ajax

Ajax, ou "Asynchronous Javascript And XML", é o uso sistemático de tecnologias providas por browsers, como **JavaScript** e **XML**, para tornar páginas mais interactivas com o utilizador, utilizando-se de solicitações assíncronas de informações. o **Ajax** não é somente um novo modelo, é também uma iniciativa na construção de aplicações Web mais dinâmicas e criativas. O **Ajax** não é uma tecnologia, são realmente várias tecnologias conhecidas trabalhando juntas, cada uma fazendo sua parte, oferecendo novas funcionalidades.

Ajax incorpora em seu modelo:

- Apresentação baseada em padrões, usando **XHTML** e **CSS**.
- Exposição e interacção dinâmica usando o **DOM**.
- Intercâmbio e manipulação de dados usando **XML** e **XSLT**.
- Recuperação assíncrona de dados usando o objecto **XMLHttpRequest**.
- **JavaScript** unindo-as em conjunto.

No modelo clássico de aplicação Web, a maioria das acções do utilizador na interface dispara uma solicitação **HTTP** para o servidor Web. O servidor processa algo — recuperando dados, realizando cálculos, conversando com vários sistemas — e então retorna

uma página **HTML** para o cliente. É um modelo adaptado do uso original da Web como um agente de hipertexto, porém o que faz a Web boa para hipertexto não necessariamente a faz boa para aplicações de software. Esta aproximação possui muito dos sentidos técnicos, mas não faz tudo que um utilizador experiente poderia fazer. Enquanto o servidor está a fazer o seu trabalho, o utilizador está em espera.

A maior vantagem das aplicações **Ajax** é que elas correm no próprio navegador Web. Então, para estar hábil a executar aplicações **Ajax**, basta possuir algum dos browsers modernos, ou seja, lançados após 2001.

2.5 CSS

CSS, ou "Cascading Style Sheets", é uma linguagem de estilo utilizada para definir a apresentação de documentos escritos numa linguagem de marcação, como **HTML** ou **XML**. O seu principal benefício é prover a separação entre o formato e o conteúdo de um documento.

Ao invés de colocar a formatação dentro do documento, o programador cria uma ligação para uma página que contém os estilos, procedendo de forma idêntica para todas as páginas de uma aplicação Web. Quando se quiser alterar a aparência da aplicação basta portanto modificar apenas um arquivo. Com a variação de actualizações dos browsers como Internet Explorer que ficou sem nova versão de 2001 a 2006, o suporte ao **CSS** pode variar. O Internet Explorer 6, por exemplo, tem suporte total a CSS1 e praticamente nulo a CSS2. Browsers mais modernos como Opera, Internet Explorer 7 e Mozilla Firefox tem suporte maior, inclusive até a CSS 3.

A necessidade da criação do **CSS** adveio do fato de o **HTML** aos poucos ter deixado de ser usado apenas para criação de conteúdo na Web, e portanto havia uma mistura de formatação e conteúdo textual dentro do código de uma mesma página. Contudo, na criação de uma grande aplicação Web, fica quase impossível manter uma identidade visual, bem como a produtividade do programador. É nesse ponto que entra o **CSS**.

2.6 JavaScript

JavaScript é uma linguagem de programação criada pela Netscape em 1995, para atender, principalmente, as seguintes necessidades:

- Validação de formulários no lado cliente (browser);
- Interação com a página. Assim, foi feita como uma linguagem de script. **JavaScript** tem sintaxe semelhante à do Java, mas é totalmente diferente no conceito e no uso.
- Oferece tipagem dinâmica - tipos de variáveis não são definidos;

- É interpretada, ao invés de compilada;
- Possui ótimas ferramentas padrão para listagens (como as linguagens de script, de modo geral);
- Oferece bom suporte a expressões regulares (característica também comum a linguagens de script).

A sua união com o **CSS** é conhecida como **DHTML**. Usando **Javascript**, é possível modificar dinamicamente os estilos dos elementos da página em **HTML**. Além de uso em browsers processando páginas **HTML** dinâmicas, o **JavaScript** é hoje usado também na construção do navegador Mozilla, o qual oferece para a criação de sistemas **GUI** todo um conjunto de ferramentas (na sua versão normal como navegador, sem a necessidade de nenhum software adicional), que incluem (e não apenas) um interpretador de **Javascript**, um comunicador **Javascript** <-> **C++** e um interpretador de **XUL**, linguagem criada para definir a interface gráfica de aplicações.

2.7 RSS

RSS é um subconjunto de "dialectos", **XML** que servem para agregar conteúdo ou "Web syndication", podendo ser acedido mediante programas ou sites agregadores. É usado principalmente em sites de notícias e blogs. A tecnologia do **RSS** permite aos utilizadores da Web inscreverem-se em sites que fornecem "feeds"(fontes) **RSS**. Estes são tipicamente sites que mudam ou actualizam o seu conteúdo regularmente. Para isso, são utilizados feeds **RSS** que recebem estas actualizações. Desta maneira o utilizador pode permanecer informado de diversas actualizações em diversos sites sem precisar visitá-los um a um.

Os feeds **RSS** oferecem conteúdo Web ou resumos de conteúdo juntamente com o link para as versões completas deste conteúdo. Esta informação é entregue como um arquivo **XML** chamado "RSS feed", "webfeed", "Atom", ou ainda canal **RSS**.

3 Sistema de Gestão de Base de Dados

Este capítulo será dedicado à base de dados que foi herdada. Depois de termos recebido uma cópia da base de dados tínhamos duas opções para a realização do trabalho.

- 1- Não alterar a base de dados;
- 2- Criar uma base de dados nova.

Estudamos as duas opções e resolvemos optar pela primeira. Se optássemos pela segunda opção teríamos que no final do trabalho criar uma *script* para migrar todos os dados da base de dados antiga para a nova. Como a base de dados é usada por várias aplicações decidimos então ficar pela base de dados que está actualmente em funcionamento.

Quando recebemos a cópia ficou definido que não poderíamos apagar nenhum campo das tabelas, mas que não haveria problema nenhum em adicionarmos mais campos e até criar novas tabelas.

A base de dados foi sendo alterada à medida que o trabalho era desenvolvido, pois dentro das nossas necessidades fomos adicionando campos e tabelas. Vamos em seguida apresentar a base de dados e as alterações que efectuamos.

Inicialmente a base de dados continha sete tabelas, sendo necessária a introdução de mais duas tabelas durante a realização do trabalho. A base de dados não segue qualquer tipo de normalização, pois antes de nós a herdamos sempre que as necessidades o pediam adicionava-se o campo pretendido.

Tabelas:

aula	curso
disciplina	docente
feriados	grupodisc
login	sala
semestre	

Em seguida apresentaremos as tabelas com os respectivos campos.

Tabela aula:

CAMPOS	TIPO	CAMPOS	TIPO
chave	INT	sem	VARCHAR
dia	SMALLINT	inicio	SMALLINT
dur	SMALLINT	tipo	VARCHAR
turno	SMALLINT	disc	VARCHAR
doc	VARCHAR	sala	VARCHAR
alt	DATE	obs	VARCHAR

chave - Chave da tabela; sem - Identificação do ano e do semestre; dia - Dia da aula, entre 2 e 6 (Segunda a Sexta); inicio - Hora de início da aula, entre 8 e 19; dur - Duração da aula; tipo - Tipo da aula se é teórica (T), teórico-prática (TP), prática (P) ou prática laboratorial (PL); turno - Numeração dos turno práticos; disc - Código da disciplina; doc - Código do docente; sala - Código da sala; alt - Data de quando a aula foi editada; obs - Observação acerca da aula.

Tabela curso:

CAMPOS	TIPO	CAMPOS	TIPO
grupo	VARCHAR	id	VARCHAR
nome	VARCHAR	ano	VARCHAR
menu	SMALLINT		

grupo - Grupo científico em que o curso está inserido; id - Identificador do curso; nome - Nome do Curso; ano - Ano do curso; menu - Activa e desactiva o registo no menu (Campo adicionado por nós).

Tabela disciplina:

CAMPOS	TIPO	CAMPOS	TIPO
sem	CHAR	abrev	VARCHAR
dep	VARCHAR	id	VARCHAR
nome	VARCHAR	resp	VARCHAR

sem - Identificação do semestre; abrev - Abreviatura do nome da disciplina; dep - Departamento a que a disciplina pertence; id - Identificação da disciplina; nome - Nome da disciplina; resp - Responsável pela disciplina.

Tabela docente:

CAMPOS	TIPO	CAMPOS	TIPO
email	VARCHAR	grupo	VARCHAR
nome	VARCHAR	num	VARCHAR
sigla	VARCHAR	menu	SMALLINT

email - E-mail do docente; grupo - Grupo em que a disciplina está inserido; nome - Nome do docente; num - Número do docente; sigla - Sigla do nome do docente; menu - Activa e desactiva o registo no menu (Campo adicionado por nós).

Tabela *feriados* (Tabela criada por nós):

CAMPOS	TIPO	CAMPOS	TIPO
id	INT	descr	VARCHAR
inicio	DATE	fim	DATE
anual	SMALLINT	-	-

id - Identificador do feriado; descr - Descrição do feriado; inicio - Início do feriado; fim - Fim do feriado; anual - Usado para identificar um feriado fixo ou móvel.

Tabela *grupodisc*:

CAMPOS	TIPO	CAMPOS	TIPO
descr	VARCHAR	id	VARCHAR

descr - Descrição do grupo; id - Identificador do grupo.

Tabela *login* (Tabela criada por nós):

CAMPOS	TIPO	CAMPOS	TIPO
user	CHAR	password	CHAR
nome	VARCHAR	priv	SMALLINT

user - Username do utilizador; password - Password do utilizador; nome - Nome do utilizador; priv - Privilégio do utilizador.

Tabela *sala*:

CAMPOS	TIPO	CAMPOS	TIPO
cap	SMALLINT	descr	VARCHAR
hor	TINYINT	id	VARCHAR
id	VARCHAR	obs	VARCHAR

cap - Capacidade da sala; descr - Descrição da sala; hor - Distingue as salas que são do DI (Departamento de Informática); id - Identificador da sala; obs - Observação da sala.

Tabela semestre:

CAMPOS	TIPO	CAMPOS	TIPO
ano	VARCHAR	data	DATE
id	VARCHAR	sem	CHAR
versao	SMALLINT	data_fim	DATE
data_inicio	DATE	menu	SMALLINT

ano - Ano do semestre; data - Data de criação do horário; id - Identificador do ano e semestre; sem - Identificador do semestre; versao - Versão do horário; data_fim - Data do final do semestre (Campo adicionado por nós); data_inicio - Data do Início do semestre (Campo adicionado por nós); menu - Activa e desactiva o registo no menu (Campo adicionado por nós).

4 Case-Study

Hoje em dia a web 2.0 está em crescente evolução, uma vez que é uma tecnologia que permite aos utilizadores fazer mais do que obter informação. Com esta tecnologia é permitido ao utilizador construir/alterar dinamicamente conteúdos online e mais do que isso, permite executar aplicações remotamente através do seu browser.

Como Case-Study para este projecto foi-nos proposto uma aplicação Web para gestão de horários com as operações típicas de um serviço deste género. De seguida é apresentada a análise de requisitos.

4.1 Análise de Requisitos

A análise de requisitos foi efectuada muito depois de nos termos ambientado a toda a tecnologia necessária. Depois de nos termos reunido com o nosso orientador, captámos os seguintes use-cases.

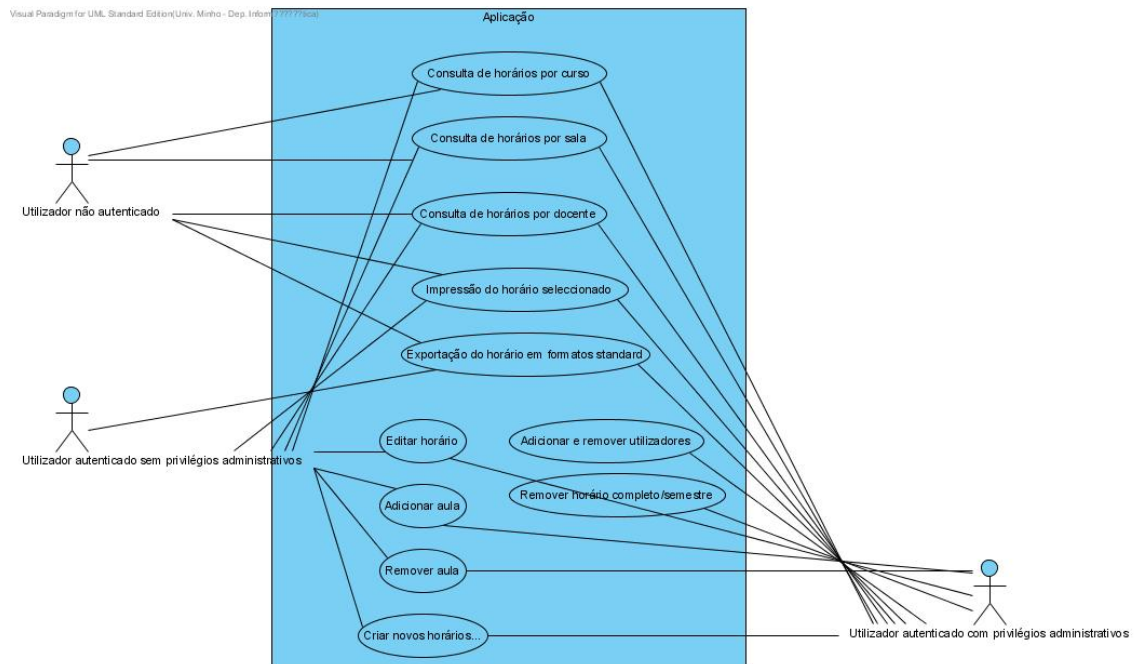


Figura 4.1: Diagrama de Use-Case.

4.2 Funcionalidades do Sistema

As funcionalidades do sistema representadas no diagrama são as seguintes:

- Consulta de horários por curso.
- Consulta de horários por sala.
- Consulta de horários por docente.
- Impressão do horário seleccionado.
- Exportação dos horários em formato standard.
- Edição de horários.
- Adição de aulas.
- Remover aulas.
- Criar novos horários.
- Adicionar e remover novos utilizadores.

- Remover horário completo.
- Remover semestre inteiro.

5 Detalhes da Implementação

A figura seguinte ilustra a arquitectura utilizada na realização do projecto, onde é possível ver todos os componentes e tecnologias empregues.

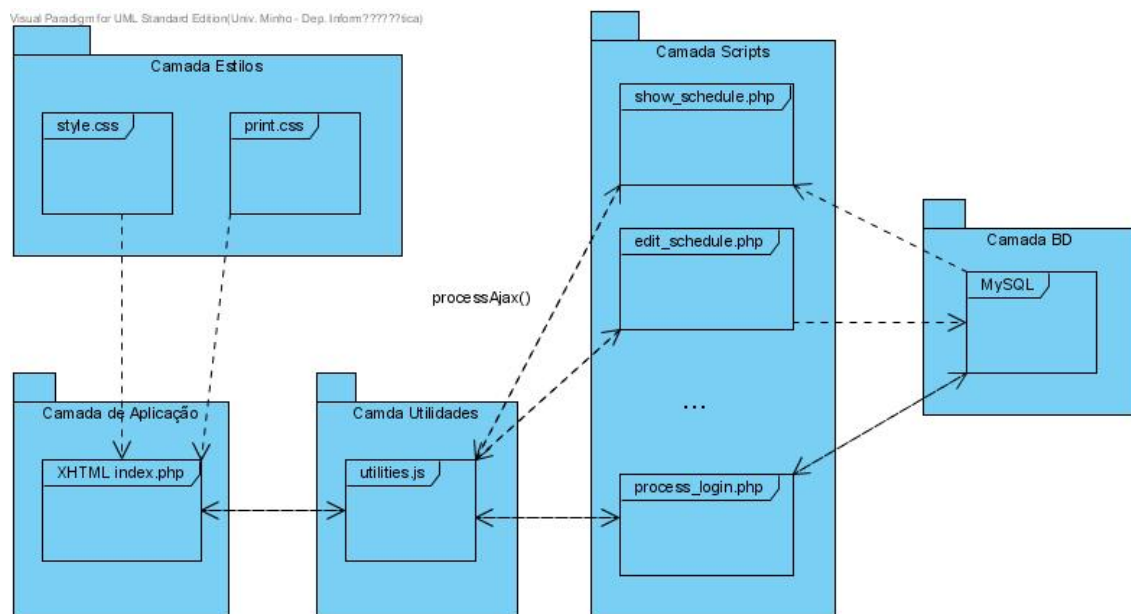


Figura 5.1: Diagrama da arquitectura.

Na figura acima, os componentes que constituem esta aplicação estão distribuídos por camadas de funcionalidade:

- Camada de Aplicação: Esta camada tem um único ficheiro, "index.php", que contém o código fonte XHTML da aplicação.
- Camada de Estilos (CSS) - É composta por dois ficheiros de configuração do estilo da aplicação no formato CSS:
 - "style.css", que configura o estilo base para a visualização no browser, desde a construção dos menus, à tabela que serve de suporte à visualização dos horários;
 - "print.css", configura o estilo para impressão dos horários, eliminando todos os elementos da página que não são desejados na impressão, tal como os menus ou o cabeçalho do horário;
- Camada Utilitária: - Esta camada possui um único ficheiro. "utilities.js". É um ficheiro em JavaScript que contém um conjunto de funções utilizadas pela aplicação para realizar as funcionalidades base, tal como enviar pedidos ajax ou imprimir o horário. Tem, também outra funcionalidade, a de servir de ligação entre o código XHTML da aplicação e o conjunto de scripts PHP.
- Camada de Scripts: - Esta camada é a camada que verdadeiramente realiza as funcionalidade da aplicação. É uma camada com um conjunto scripts na linguagem PHP que irão ser utilizados para gerar código HTML e aceder à base de dados.
- Camada da Base de Dados: - É a camada que possui a base de dados utilizada pela aplicação. Os dados da base de dados serão manipulados por scripts PHP, devido ao excelente suporte desta linguagem para base de dados MySQL.

5.1 Consulta

Um dos principais requisitos da aplicação, senão o mais importante, é a possibilidade da visualização dos horários nos principais browsers. Existem três tipos de horários a visualizar, cursos, docentes e salas. Nos horários do tipo curso será possível a visualização de todas as aulas de um determinado ano de um curso num semestre. Nos horário de tipo docente apenas é visualizado as aulas que um dado docente lecciona num semestre, e horários do tipo sala apenas é visualizado as aulas de um determinado semestre, cuja a sala de aulas (do departamento de informática) é comum entre elas.

Para um melhor suporte desta funcionalidade da nossa aplicação optámos pela utilização de tabelas HTML. Estes elementos HTML permitem a organização da informação em células:"<td> ... </td>", que se encontram agrupadas por linhas de tabela, chamadas "table

rows", "<tr><td>Célula 1</td> <td> Célula 2 </td></tr>", e por sua vez estas se encontram organizadas numa tabela: "<table> ... </table>". Cada aula será impressa numa única célula, havendo a possibilidade, se a duração da aula for superior a uma hora, a célula agrupar mais do que uma linha da tabela.

	PL TP1 (PRH) DI 0.02	TP I
	PL TP2 (PRH) DI A2	T I
SPCP		-----

Figura 5.2: Aula com mais do que uma hora.

Como se pode ver pela figura acima uma célula irá conter a seguinte informação sobre cada aula: abreviatura da disciplina (PL), o tipo de aula (TP), o turno (2), o docente (PRH) e a sala de aula. Se estivermos a visualizar um horário de um docente ou de uma sala de aulas à informação contida na célula é acrescentada o curso em que a disciplina é leccionada.

Para cada horário, será gerado dinamicamente um tabela HTML através de um script PHP (show_schedule.php). Este script obtém as aulas realizando um query à base de dados, baseado em três variáveis que indicam: o tipo (curso, docente ou sala), a identificação e semestre do horário. Estas variáveis podem ser variáveis de sessão, caso o utilizador esteja autenticado, ou cookies para caso de não existir autenticação. Será realizado por parte deste script um query diferente para cada tipo de horário, uma vez que a informação a extrair é diferente para os vários tipos de horário. Podemos de seguida ver um exemplo de um query:

```

1  $curso = 'EI3';
2  $semestre = '0708-2'
3
4  mysql_query("SELECT disciplina.abrev, aula.chave, aula.alt,
5  aula.doc, aula.tipo, aula.turno, aula.dia, aula.inicio, aula.dur,
6  aula.sala
7  FROM disciplina, aula
8  WHERE disciplina.curso = '$curso'
9  AND (disciplina.sem = '2' OR disciplina.sem = 'A')
10 AND aula.sem = '$semestre'
11 AND (aula.disc LIKE disciplina.id OR aula.disc LIKE
12 Concat('%',',', disciplina.id)
13 OR aula.disc LIKE Concat(disciplina.id,',','%')
14 OR aula.disc LIKE Concat('%',',',disciplina.id,',','%'))
15 ORDER BY aula.dia, aula.inicio, disciplina.abrev");

```

A figura acima demonstra como é possível extrair da base de dados todas as aulas do curso 'EI3' para o semestre '0708-2'. Note-se que as aulas podem ser semestrais ou anuais e, é possível que exista mais do que uma disciplina por aula sendo necessário realizar testes extra para verificar estes casos.

Toda a informação que foi recolhida sobre as aulas será guardada num array (*aulas[i][campo]*) com indices associativos que correspondem aos campos da tabela. Foi necessário guardar esta informação num array pois será futuramente manipulada para gerar a tabela com o horário.

Existem alguns passos intermédios antes da impressão das aulas na tabela. um desses passos é a geração automática dos turnos para o caso em que o campo 'turno' seja NULL. OS turnos são gerados pela ordem em que aparecem no horário neste caso ordenados por dia e hora, e aulas de tipos diferentes geram turnos diferentes, havendo independência entre eles.

```

1  if($tabela[$i]['turno'] == ""){
2    $turno_num = 1;
3    if(isset($turno[$abrev][$tipo])){
4      $turno_num = end($turno[$abrev][$tipo]) + 1;
5    }
6    $turno[$abrev][$tipo][$chave] = $turno_num;
7  }
8  else{
9    $turno[$abrev][$tipo][$chave] = $tabela[$i]['turno'];
10 }

```

Como se pode verificar na figura acima os turnos com valor NULL são guardados num array auxiliar (*turno[abrev][tipo][chave]*) ordenados por abreviatura, tipo e chave da aula,

sendo possível futuramente aceder a esta informação quando desejarmos conhecer o valor gerado para uma dada chave de aula.

Outro passo intermédio, e indispensável para o processo de construção da tabela foi verificar a largura máxima da tabela para cada dia da semana. Este cálculo é efectuado pois surgiu a necessidade de conhecer esta largura, uma vez que pode ser possível existirem mais do que uma aula à mesma hora sendo necessário adaptar a impressão da tabela a esta realidade. O cálculo desta largura é bastante simples, um array guarda para cada dia a largura máxima calculada ao percorrer todas as aulas e verificando a sua hora de início e duração.

A impressão da tabela é realizada linha a linha, percorrendo os dias da semana. Sendo as linhas da tabela correspondentes a hora, para cada dia é seleccionado o conjunto de aulas que têm início para a hora em questão (linha de tabela). Se a duração da aula for superior a 1 hora, essa célula ocupará mais do que uma linha (o número de linhas corresponde à duração da aula) que pode ser definida da seguinte forma:

```
1 <td colspan='$duração'> informação da aula... </td>
```

Em que o valor `colspan='$duração'` indica o número de linhas que essa célula ocupará. Existe o cuidado de na linha de tabela seguinte não imprimir esta aula, uma vez que já foi impressa, repetindo-se este processo até que a linha da tabela em questão seja superior à soma da hora de início com a sua duração.

A grande dificuldade na construção do algoritmo de impressão da tabela deve-se à existência de aulas em simultâneo. Para solucionar este problema foi calculado anteriormente a largura máxima do dia. Este valor serve para indicar quantos espaços em branco falta preencher a cada dia para uma determinada hora. Por exemplo, caso a largura máxima de um dia seja dois, este valor indica que para esse dia existe no máximo duas aulas em simultâneo. Para o caso em que o número de aulas seja igual à largura máxima não será necessário verificar o número de espaços em branco a imprimir, mas para o caso em que só exista uma aula é necessário preencher o resto do dia com os espaços em branco, o que neste caso será apenas um. Mas este cálculo não é suficiente para imprimir correctamente as aulas. Ainda existem casos em que os espaços em brancos não são correctamente calculados. São exemplo disso os casos em que existam aulas em simultâneo mas que a hora de início não seja coincidente e para esse hora não atinja o número de aulas máximo. Por exemplo, um dia que possua no máximo três aulas em simultâneo, mas que numa dada hora desse dia existem duas aulas simultâneas que não possuem a mesma hora de início. Neste caso o algoritmo não funciona, não sendo capaz de imprimir os espaços em branco. Este problema foi solucionado recorrendo a um contador que se inicia quando surge um novo dia e que vai adicionando ao seu valor as aulas que vão sendo impressas. Também verifica se existe alguma aula que foi impressa anteriormente e que se prolonga até à hora actual, sendo estas também adicionadas ao valor do contador. Os espaços em brancos a imprimir são calculados pela diferença entre a largura máxima e o contador. Com esta solução é

possível a correcta impressão do horários para qualquer número de aulas em simultâneo independentemente da hora em que se iniciam.

5.1.1 Diferenças de Visualização

O utilizador final da aplicação encontrará duas realidades distintas na visualização do horário, dependendo do tipo de autenticação que possui. Se o utilizador não possui autenticação, apenas terá acesso à visualização básica, sem a possibilidade de edição do horário e sem mensagens de aviso do seu estado actual. Na visualização básica é possível ver todas as aulas para um dado curso, docente ou sala num semestre. É possível a navegação entre os diferentes horários, uma vez que na informação disponível da aula é adicionado uma âncora na informação da sala e do docente que conduz ao seus respectivos horários. Este utilizador possui ainda as opções de exportação, para um dos vários formatos disponíveis: iCalendar, XML ou HTML, e impressão do horário. O utilizador sem autenticação possui ainda uma outra opção de visualização, a visualização de um horário personalizado. Se o utilizador está a assistir a aulas que não constam do ano em que está a frequentar, pode optar por adicionar ao horário aulas de anos anteriores ou posteriores ao horário que está a visualizar. É possível também, se o utilizador assim o desejar apenas visualizar o turno prático que se encontra a frequentar. Esta opção é um opção extra que consideramos bastante útil, uma vez que o utilizador poderá exportar o horário que acabou de criar.

Para o utilizador com autenticação todas as opções anteriores estão disponíveis, bem como as opções de edição e notificação do estado do horário. As opções de edição incluem adição, remoção e edição de aulas. E o sistema de notificação indica ao utilizador se existem colisões de docentes ou sala. Entende-se por colisão como alocação de um recurso numa mesma hora, ou seja, no nosso caso implica que o mesmo docente ou sala estejam alocados para mais do que uma aula à mesma hora.

5.1.2 Colisões de Horário

Para utilizadores autenticados, está disponível um sistema de notificação de colisões, que detecta colisões de salas e docentes que estejam escalonados para uma mesma hora. Este sistema chama a atenção do utilizador indicando uma mensagem de erro com a informação de que existem incompatibilidades no horário e imprime a aula a uma cor que se destaque das outras, normalmente a vermelho.

Este sistema realiza um query à base de dados por cada aula no horário verificando se durante a duração dessa aula não existe outra aula com o mesmo docente e/ou sala. Se encontrar alguma aula guarda a chave e o tipo de colisão num array para mais tarde poder ser visualizado pelo utilizador.

5.1.3 Horário Personalizado

Para os utilizadores mais exigentes, que necessitem de um horário à sua medida, é possível através da opção "Horário Personalizado", a criação de um horário criado pelo próprio utilizador. Esta opção tem como objectivo a criação de um horário personalizado que só contenha as disciplinas que o utilizador deseje, havendo a possibilidade de coexistir neste mesmo horário disciplinas de anos diferentes. A grande vantagem deste género de horário é que tal como todos os outros horários pode ser imprimido ou exportado, podendo o utilizador guardar sempre o horário que criou com a ajuda desta aplicação.

Este caso trata-se de um caso especial da visualização de horários, uma vez que não está limitado a um só ano ou à obrigatoriedade da visualização de todos os turnos de uma disciplina.

Esta funcionalidade foi obtida guardando as chaves das aulas a serem visualizadas em cookies (para o caso do utilizador não autenticado), ou em variáveis de sessão (para os utilizadores autenticados). Existe ainda a possibilidade de um utilizador não autenticado bloquear os cookies enviados pela aplicação. Para estes casos estará disponível uma janela com uma lista de aulas que o utilizador vai adicionando e no final submete o formulário para visualizar as aulas seleccionadas. Este tipo de utilizador não poderá construir o horário dinamicamente, uma vez que sempre que necessite de adicionar uma nova aula terá que repetir todo o processo, porque não existe a possibilidade de guardar as disciplinas que estão a ser visualizadas.

5.2 Edição

O modo de edição apenas se encontra disponível para utilizadores autenticados. Neste modo é possível a criação, remoção e edição de aulas, a criação e remoção de horários. O utilizador poderá aceder a estas funcionalidades através do menu 'Opções' ou clicando directamente na cadeira ou no espaço em branco que deseje adicionar uma nova aula.

5.2.1 Adicionar Aula

Esta opção tem como objectivo adicionar uma nova aula a um determinado horário. O utilizador tem ao seu dispor uma janela com os campos necessários à criação de uma aula, disciplina, turno, docente, sala, etc. Caso o utilizador pretenda adicionar uma nova aula a um horário do tipo curso, a aplicação automaticamente seleccionará apenas as disciplinas referentes a esse horário, o mesmo se passa se tivermos a visualizar um horário do tipo docente ou sala. Neste tipo de horários será seleccionado o docente e a sala respectivamente. Caso o utilizador aceda através de um click num espaço branco no horário a hora de início será seleccionada automaticamente. Os dados a serem adicionados então contidos num formulário que ao ser submetido irá desencadear um conjunto de funções em javascript

cujo o objectivo será ler o valor dos campos e construir uma string do género "nome=valor" a ser enviado utilizando o método "POST" do ajax ao script "add_schedule.php". Este script vai ler os valores contidos no array `$_POST[]` com a informação da nova aula e adiciona essa informação à base de dados. Se a inserção for bem sucedida é enviado código por parte do script com as funções de actualização de horário e notificação de adição da aula.

5.2.2 Editar aula

A edição é muito semelhante à adição possuindo um formulário com os campos já preenchidos com a informação relativa à aula a editar. E, apesar de o script PHP que realiza a actualização da base de dados ("update_schedule.php") ser diferente do utilizado pela adição e resultado é semelhante. Nesta opção ainda se destaca 2 funcionalidades extras que visam ajudar o utilizador a evitar erros. Uma dessas funcionalidades é um sistema que sugere valores à medida que o utilizador vai escrevendo no campo, podendo a qualquer altura seleccionar uma das sugestões evitando assim um maior número de erros a editar. Pode ainda haver a possibilidade de o utilizador introduzir um valor que não é o mais correcto, para estes casos existe um sistema de verificação dos valores que não permite que o formulário seja submetido enquanto os valores não se apresentarem correctos.

5.2.3 Eliminar Aula

A eliminação da aula é acedida através da opção 'Apagar' na caixa de informação da disciplina. Foi introduzido um sistema de segurança para prevenir eventuais acidentes, uma janela de confirmação é mostrada ao utilizador, para este confirmar que deseja apagar a aula. Esta janela envia a chave da aula para o script "delete_schedule.php" que lê este valor e apaga da base de dados a entrada que possui a chave passada:

```
1 mysql_query("DELETE FROM aula WHERE chave='$key'");
```

Se a eliminação for bem sucedida o script envia código JavaScript para ser executado com as funções de actualização do horário e notificação de eliminação de aula.

5.3 Autenticação

Esta aplicação tem como objectivo final a consulta dos horários por parte da comunidade académica, sendo necessário bloquear as opções de edição de forma a não permitir que qualquer utilizador altere a informação contida na base de dados. Para cumprir esse objectivo foi criado um sistema de autenticação, na qual um utilizador que deseje editar o horário apenas o conseguirá fazer após indicar um nome de utilizador e uma password válidos.

Para tal fim, foi utilizado variáveis de sessão de PHP. Estas variáveis permitem que o utilizador possua uma identificação única e impossível de ser replicada por outro utilizador uma vez que esta identificação é gerada automaticamente.

A autenticação é um processo simples, no qual verificamos se na base de dados existe um determinado utilizador verificando de seguida se a password introduzida corresponde à password do utilizador:

```
1  if(mysql_num_rows(mysql_query("SELECT user FROM login WHERE
2                                user=\`utilizador\`")) == 1){
3  $num = mysql_num_rows(mysql_query("SELECT user, nome FROM
4                                login WHERE user='utilizador'
5                                AND password=\`password\`"));
6  ($num > 0){ // ocorreu a autenticação... }
```

Uma vez o utilizador autenticado é definido um conjuntos de variáveis de sessão que irão servir para facilitar a gestão da sessão do utilizador, sendo algumas delas: \$_SESSION['auth'], que terá o valor "yes" para indicar que se trata de utilizador autenticado, \$_SESSION['user'], que guarda o nome do utilizador que será visualizado no menu, \$_SESSION['schedule_type'], \$_SESSION['schedule_id'], \$_SESSION['schedule_sem'], que contem os valores do tipo, identificação e semestre do horário actualmente a ser visualizado.

Quando o utilizador realiza o logout a sessão de utilizador, bem como as suas variáveis são destruídas e o seu valor apagado da memória:

```
1  session_destroy();
2  unset($_SESSION);
```

5.4 Exportação

Outra da funcionalidade base a ser implementada pela aplicação seria a exportação dos horários para um formato de eventos standard, com o objectivo de aceder às aulas em programas de gestão de eventos tal como o iCal para Mac OS X ou o Google Calendar. Devido a este requisito, decidimos utilizar como formado de exportação o formato iCalendar (.ics) que se trata de um standard de eventos. Um ficheiro iCalendar possui a seguinte especificação:

```
1 BEGIN:VCALENDAR
2 VERSION:2.0
3 PRODID://hacksw/handcal//NONSGML v1.0//EN
4 BEGIN:VEVENT
5 DTSTART:19970714T170000Z
6 DTEND:19970715T035959Z
7 SUMMARY:Bastille Day Party
8 END:VEVENT
9 END:VCALENDAR
```

Em que toda a informação de um calendário se encontra dentro das seguintes tags "BEGIN:VCALENDAR" e "END:VCALENDAR". Possui algumas propriedades como a versão do calendário: "VERSION:2.0", e a identificação e idioma da aplicação que gerou os eventos: "PRODID://Departamento de Informática - Universidade do Minho//Gestão Horários v1.0//PT". Os eventos do calendário são as aulas a exportar que se podem definir da seguinte forma:

```
1 BEGIN:VEVENT
2 TRANSP:OPAQUE
3 DTSTART;TZID=20080704T11000000
4 SUMMARY:CG TP2
5 CREATED:20080701T23000000
6 DTEND;TZID=Europe/Lisbon:20080704T13000000
7 LOCATION:DI 0.04
8 END:VEVENT
```

O evento acima indica uma aula TP de Computação Gráfica (CG) que começa as 11 horas do dia 4 de Julho de 2008 e termina as 13 horas desse mesmo dia. A localização do evento, neste caso é sala onde é leccionada a aula que será a sala DI 0.04.

Como funcionalidades extra de exportação aumentamos a capacidade de exportação para três formatos. Foi implementado ainda a exportação para o formato XML e para o formato HTML. Neste último formato é criado uma página web que contém um calendário dividido por meses ou semanas com as aulas leccionadas no horário. Acreditamos que este último formato será bastante útil para os utilizadores que não disponham de um programa que possa realizar a leitura de eventos.

Apresenta-se ainda algumas funcionalidades que achamos pertinentes incluir na exportação. Na janela de exportação o utilizador poderá ainda escolher a data de exportação e quais as disciplinas que deseja exportar. Na opção data de exportação poderá escolher o semestre inteiro e o script irá criar eventos (aulas) entre as datas contidas nos campos 'data_inicio' e 'data_fim' da tabela 'semestre' ou pode definir uma data de início e fim e os

eventos apenas serão gerados entre essas datas. A exportação possui ainda a particularidade de não criar eventos os dias em que não há aulas por motivo de fim-de-semana ou feriados. As datas que o script têm que "ignorar" encontram-se na tabela 'feriados'. Nesta tabela estão definidos feriados anuais (campo anual='1') e acontecimentos que se prolongam mais do que um dia como por exemplo férias da Páscoa. Estes dias são todos ignorados na exportação. Existe ainda a opção de seleccionar as disciplinas a exportar. Apenas as disciplinas seleccionadas serão exportadas, todas as outras serão ignoradas.

5.5 Impressão

A impressão dos horários é realizada usando o método pré-definido de JavaScript "window.print()", que permite a impressão por parte do sistema operativo da janela actualmente a ser visualizada no browser. A implementação desta função é apenas uma simples chamada na âncora com o valor "Imprimir", que se passa a demonstrar seguidamente:

```
1 <a href="javascript:window.print();" >Imprimir</a>
```

Este método imprime a página tal como está a ser visualizada, que não é o pretendido. Apenas queremos que a impressão contenha a tabela com o horário e um novo cabeçalho que pertença à tabela. Para realizar este objectivo foi adicionado ao ficheiro "index.php" um estilo CSS alternativo:

```
1 <link rel="stylesheet" type="text/css"  
2 href="themes/print.css" media="print" />
```

Este estilo alternativo configura a página para impressão, isto é conseguido uma vez que foi definido media="print", ou seja este estilo apenas será utilizado na impressão do horário. Para remover todos os elementos indesejados altera-se a propriedade "display" para o valor "none", o que irá ocultar o elemento em que esta propriedade seja assim definida:

```
1 #footer{ display: none; }
```

Com esta porção de código é possível ocultar o rodapé da aplicação de forma a não aparecer na impressão.

5.6 RSS Feeds

Uma das funcionalidades que nos pareceu óbvia facilitar ao utilizador da nossa aplicação foi fornecer uma método de se manter a par das alterações efectuadas nos horários. Por isso sugerimos a implementação de um sistema de notificação baseado em RSS Feeds. Este sistema consiste numa série de ficheiros XML com o seguinte formato:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <rss version="2.0">
3  <channel>
4  <title>Gestão de Horários - LEI3</title>
5  <link>http://www.di.uminho.pt/</link>
6  <webMaster>alcino@di.uminho.pt</webMaster>
7  <description>Gestão de Horários - Curso: EI3 Feed
8  </description>
9
10 <item>
11 <title>Aula PL TP4 Actualizada.</title>
12 <link>http://www.di.uminho.pt/horarios/feeds/feed_EI3.xml
13 </link>
14 <description>Alteração da hora de início das 14horas para
15 as 11horas.</description>
16 <guid isPermaLink="true">
17 http://www.di.uminho.pt/horarios/feeds/</guid>
18 <pubDate>Fri, 04 Jul 2008 04:00:00 GMT</pubDate>
19 </item>
20
21 </channel>
22 </rss>

```

O exemplo demonstrado acima indica uma alteração na hora de uma disciplina do curso EI3.

Existe um ficheiro deste género para cada horário a visualizar, permitindo ao utilizador subscrever notificações de vários horários diferentes. O processo de criação e alteração do conteúdos dos vários ficheiros .rss é conseguido através de um script em PHP ("process_feed.php") que a cada actualização na base de dados lê e escreve neste ficheiro as alterações efectuadas. O administrador não necessitará de fazer as alterações possuindo apenas um conjunto de opções que poderá editar de forma a otimizar a actualização da Feed. Entre as opções poderá seleccionar se pretende que haja notificação em todas as actualização efectuadas ou poderá seleccionar a que desejar, edição, remoção, etc...

5.7 Estilos com CSS

A utilização de CSS para definir o interface revelou-se bastante útil, uma vez que esta linguagem permitiu uma rápida construção de todos os elementos que constituem a aplicação. É possível com esta linguagem definir as propriedades de qualquer elemento HTML como se pode ver de seguida:

```

1 #menu{ background-color: #464646;
2     margin : 1px 0 0 0;
3     padding: 0;
4     width: 100%;
5     height: 23px;
6     border-bottom: 1px solid #58a7ff; }

```

O exemplo anterior demonstra como é possível definir um conjunto de propriedades para um elemento HTML, tais como a sua altura (height), largura (width) ou cor de fundo (background-color). Apesar de existirem grandes diferenças no suporte, por parte dos browsers, esta é uma linguagem standard para edição dos estilos de documentos HTML. De uma maneira geral todos os browsers tem um óptimo suporte para CSS, com a excepção do Internet Explorer (IE). Este facto causou alguns problemas à criação do interface com o utilizador, principalmente na criação dos menus. Os menus foram construídos à base de listas HTML ... e, sempre que um utilizador coloca o rato sobre algum item do menu este mostra um conjunto de opções referentes ao item seleccionado. Foi nesta parte que nos deparamos com graves problemas com o IE, uma vez que esta funcionalidade é conseguida através da alteração da das propriedades visuais do menu quando o rato se coloca sobre o elemento (elemento:hover). Neste caso trata-se de uma lista, mas o IE não possui este evento, não sendo capaz de alterar as suas propriedades. A solução para este problema foi resolvido recorrendo a uma script Open Source criado por Peter Nederlof que vem adicionar, através de um conjunto de funções em JavaScript, este evento ao browser:

```

1 function HoverElement(node, className, events) {
2     if(!node.hovers) node.hovers = {};
3     if(node.hovers[className]) return;
4     node.hovers[className] = true;
5     hookHoverEvent(node, events.on, function() {
6         node.className += ' ' + className; });
7     hookHoverEvent(node, events.off, function() {
8         node.className = node.className.replace(
9             new RegExp('\s+' + className, 'g'), ''); }); }
10
11 function hookHoverEvent(node, type, handler) {
12     node.attachEvent(type, handler);
13     hoverEvents[hoverEvents.length] = {
14         node:node, type:type, handler:handler }; }

```

Na figura acima vemos apenas uma pequena porção do script que recria a funcionalidade "elemento:hover" no IE. Após ultrapassar este obstáculo foi possível construir um menu que se comporta da mesma forma em todos os browsers, permitindo uma experiência semelhante de utilização desta aplicação em todos os browsers.

6 Resultados Obtidos

A aplicação Web desenvolvida gira à volta da consulta, edição e exportação de horários em formatos standard. Naturalmente toda a parte de segurança e autenticação de utilizadores faz parte da aplicação, havendo dois tipos de utilizadores com diferentes privilégios:

- Utilizadores com privilégios de edição e eliminação de aulas ou salas em que são docentes, para além das mencionadas anteriormente que não precisam de qualquer tipo de autenticação.
- Utilizadores com privilégios de, para além de todas as outras anteriormente mencionadas, actividades relacionadas com utilizadores (adição e remoção), e remoção total de horários.

Para além das funcionalidades primárias que eram principal objectivo do trabalho, foi adicionado **RSS feeds** por cada horário existente. Além da exportação para o formato **iCalendar**, também foi adicionada a possibilidade da exportação para os formatos **HTML** e **XML**, pois achamos extremamente úteis tais funcionalidades.

6.1 Consulta de horários

Gestão de Horários Departamento de Informática					
Cursos Docentes Salas Opções Ajuda	Login				
Curso: Mestrado de Informática – 1º ano Semestre: 2007/2008 – 2					
Versão: 01/07/2008					
Horas	2ª Feira	3ª Feira	4ª Feira	5ª Feira	6ª Feira
8-9					
9-10					
10-11					
11-12					
12-13					
13-14	EC T (ZZZ) DI 1.08	EL T (ZZZ) DI 0.04	SD T (ZZZ) DI 1.05	CPD T (ZZZ) DI 1.05	SSI T (ZZZ) DI 1.08
14-15					
15-16					
16-17					
17-18					
18-19					
19-20					

RSS Feed | Exportar | Imprimir

Copyright © 2008 Departamento de Informática – Universidade do Minho. Todos os direitos reservados.

Igor Brás | Jorge Moreira | Ricardo Vilas Boas

Figura 6.1: Visualização de um horário(sem autenticação).

Nesta imagem podemos ver o que um utilizador não autenticado pode ver/fazer. É possível fazer a consulta dos horários, por cursos, docentes e salas. Para além disso pode imprimir qualquer horário, subscrever **RSS** e exportar o horário para os vários formatos disponíveis.

Curso: Lic. Engenharia Informática - 1º ano
Semestre: 2007/2008 - 2

Versão: 01/07/2008

Horas	2ª Feira			3ª Feira			4ª Feira				5ª Feira				6ª Feira				
8-9	PI TP1 (JBB) CP1 103						TFM T CP2 102				TFM T CP2 102				TFM TP6 CP2 211				
9-10				Calc2 TP1 CP1 308	Log TP3 CP1 211	PI TP2 (MBB) DI A1	SC T (JMF) CP2 103				PI T (JBB) CP1 A5								
10-11	Calc2 T CP1 A4						Log T CP1 A5				SC T (JMF) CP2 103				LI2 PL5 (JMM) DI 0.03	Log TP7 CP1 308	PI TP5 (NFR) CP2 211	SC TP5 (JMF) CP1 211	
11-12				Calc2 TP2 CP1 307	Log TP4 CP1 211	PI TP3 (MBB) DI A1	Calc2 TP6 CP1 303	LI2 PL2 (RCM) DI 0.03	LI2 PL3 (JGR) DI 0.12	Log TP6 CP1 209	SC TP3 (JMF) CP2 111	LI2 PL4 (MPR) DI 0.03	PI TP4 (JBB) DI A1	SC TP4 (JMF) CP1 220	TFM TP5 CP1 307	PI T (JBB) CP2 102			
12-13	Log T CP1 A4																		
13-14																			
14-15	Log TP1 CP1 307	SC TP1 (AMP) CP1 308	TFM TP1 CP1 220	Calc2 TP3 CP1 308	Log TP5 CP1 311	TFM TP3 CP2 203									LI2 PL6 (JGR) DI 0.03	PI TP6 (NFR) CP1 305	SC TP6 (JMF) CP2 206		
15-16																			
16-17	Log TP2 CP1 307	SC TP2 (ANP) CP1 308	TFM TP2 CP2 210	Calc2 TP4 CP1 311	Calc2 TP5 CP1 313	LI2 PL1 (RCM) DI 1.09	TFM TP4 CP2 109												
17-18																			
18-19																			
19-20																			

[RSS Feed](#) | [Exportar](#) | [Imprimir](#)

Figura 6.2: Visualização de um horário (com autenticação).

Aqui é possível visualizar o conteúdo da aplicação quando um utilizador está autenticado, dependendo dos privilégios que possa ter poderá eliminar, editar e adicionar uma nova aula caso seja docente ou poderá eliminar um horário completo ou adicionar ou remover utilizadores do sistema, caso tenha privilégios administrativos.

Docente: [MAC] - Manuel Alcino Cunha
 Semestre: 2007/2008 - 2

Versão: 21/02/2008

Horas	2ª Feira	3ª Feira	4ª Feira	5ª Feira	6ª Feira
8-9					
9-10	PC [CC3] T (MAC,LSB) CP1 211			CP [E12] T (MAC) CP2 104	
10-11					
11-12		CP [E12] TP (MAC) CP2 111	PC [CC3] TP (MAC,LSB) DI A2,DI 0.11	CP [E12] TP (MAC) DI A2	
12-13					
13-14					
14-15					
15-16					
16-17					
17-18					
18-19					
19-20					

[RSS Feed](#) | [Exportar](#) | [Imprimir](#)

Figura 6.3: Horário de docente.

Está agora presente um exemplo da visualização de um horário de um docente. De notar que em qualquer aula, quer os docentes e salas estão "linkáveis", para apresentação dos respectivo horários.

Sala: DI A2 (Anf.)
Semestre: 2007/2008 - 2

Versão: 30/06/2008

Horas	2ª Feira	3ª Feira	4ª Feira	5ª Feira	6ª Feira
8-9				PI [CC1] TP (AMS) DI A2	
9-10					
10-11	Tec2 [MAPI] T (ZZZ) DI A2	PLC [CC2] TP (JJA) DI A2			
11-12		PL [E13] TP (PRH) DI A2	PC [CC3] TP (MAC,LSB) DI A2,DI 0.11	CP [E12] TP (MAC) DI A2	SEM [MI] T (PMC) DI A2
12-13					
13-14					
14-15		SRCR [E13] T (JMN) DI A2			
15-16					
16-17		PLC [CC2] TP (UCR) DI A2			SSD [MI] T (ZZZ) DI A2
17-18					
18-19					
19-20					

[RSS Feed](#) | [Exportar](#) | [Imprimir](#)

Figura 6.4: Horário por sala.

Está agora presente um exemplo da visualização de um horário de uma sala.

6.2 Autenticação e *Info bubble*

Gestão de Horários
Departamento de Informática

Cursos Docentes Salas Opções Ajuda Login

Curso: Lic. Engenharia Informática - 3º ano
Semestre: 2007/2008 - 2

Versão: 01/07/2008

Horas	2ª Feira	3ª Feira	4ª Feira	5ª Feira	6ª Feira	
8-9						
9-10		CC TP2 (ADC) DI 0.10		PL TP3 (PRH) DI 0.02	CC T (ADC) CP2 101	
10-11						
11-12	CC TP1 (ADC) DI 0.10	SRCR TP1 (PJN) DI 1.04	CC TP3 (ADC) DI 0.10	MEIO T CP2 104	CG T (ARF) CP2 101	
12-13						
13-14						
14-15	MEIO TP1 CP1 209	SRCR TP2 (CAR) DI 1.04	SRCR T (GMS) DI A2	PL T (PRH) CP1 A5	CG TP1 (ARF) DI 0.04	PL TP4 (PRH) DI 0.02
15-16						
16-17	MEIO TP2 CP1 101	SRCR TP3 (CAR) DI 1.04	CC TP4 (BAG) DI 0.10	CG TP1 (ARF) DI 0.04	CC TP5 (PNS) DI 0.10	SRCR TP4 (CAR) DI 1.04
17-18						
18-19						
19-20						

RSS Feed | Exportar | Imprimir

Copyright © 2008 Departamento de Informática - Universidade do Minho. Todos os direitos reservados. Igor Brás | Jorge Moreira | Ricardo Vilas Boas

Figura 6.5: Interface de *Login*.

A interface de *login*, é bastante simples, mas a autenticação é bastante segura como é pertinente em aplicações deste tipo.

Gestão de Horários
Departamento de Informática

Cursos Docentes Salas Opções Ajuda Bem-vindo: Igor Brás Logout

Curso: Lic. Ciências da Computação – 3º ano
Semestre: 2007/2008 – 2

Versão: 28/06/2008

Horas	2ª Feira	3ª Feira	4ª Feira	5ª Feira	6ª Feira
8-9					
9-10	PC T (MAC,LSB) CP1 211	SP T (LSB) CP1 307	TNC	TNC T P2 111	PA TP Lab.Mat
10-11			Computação Gráfica (TP) Docente: ARF Hora: 3ª Feira (11h-13h) Sala: DI 0.04	PA T P1 A3	CG T (ARF) CP2 101
11-12	SP TP (LSB) CP1 206	CG TP (ARF) DI 0.04			
12-13					
13-14					
14-15					AEC T (ZZZ) CP2 111
15-16					
16-17					
17-18					
18-19					
19-20					

RSS Feed | Exportar | Imprimir

Copyright © 2008 Departamento de Informática – Universidade do Minho. Todos os direitos reservados. Igor Brás | Jorge Moreira | Ricardo Vilas Boas

Figura 6.6: *Info bubble*.

Aqui temos presente uma mensagem de informação a que chamamos *Info Bubble*, onde apenas basta clicar numa qualquer aula para a *Info Bubble* aparecer com informação da aula e com as opções de edição ou remoção da aula.

6.3 Edição

Gestão de Horários
Departamento de Informática

[Cursos](#) [Docentes](#) [Salas](#) [Opções](#) [Ajuda](#)

Bem-vindo: Igor Brás [Logout](#)

Curso: Lic. Ciências da Computação - 3º ano
 Semestre: 2007/2008 - 2 Versão: 28/06/2008

Horas	2ª Feira	3ª Feira	4ª Feira	5ª Feira	6ª Feira
8-9					
9-10	PC T (MACLSB) CP1 211	SP T (LSB) CP1 307		TNC T P2 111	PA TP Lab.Mat
10-11					
11-12	SP TP (LSB) CP1 206	CG TP (ARF) DI 0,04		PA T P1 A3	CG T (ARF) CP2 101
12-13					
13-14					
14-15					
15-16					AEC T (ZZZ) CP2 111
16-17					
17-18					
18-19					
19-20					

[RSS Feed](#) | [Exportar](#) | [Imprimir](#)

Copyright © 2008 Departamento de Informática - Universidade do Minho. Todos os direitos reservados.
Igor Brás | Jorge Moreira | Ricardo Vilas Boas

Figura 6.7: *Edit bubble.*

Caso solicitemos a edição de uma aula é desta forma que procedemos à sua edição.

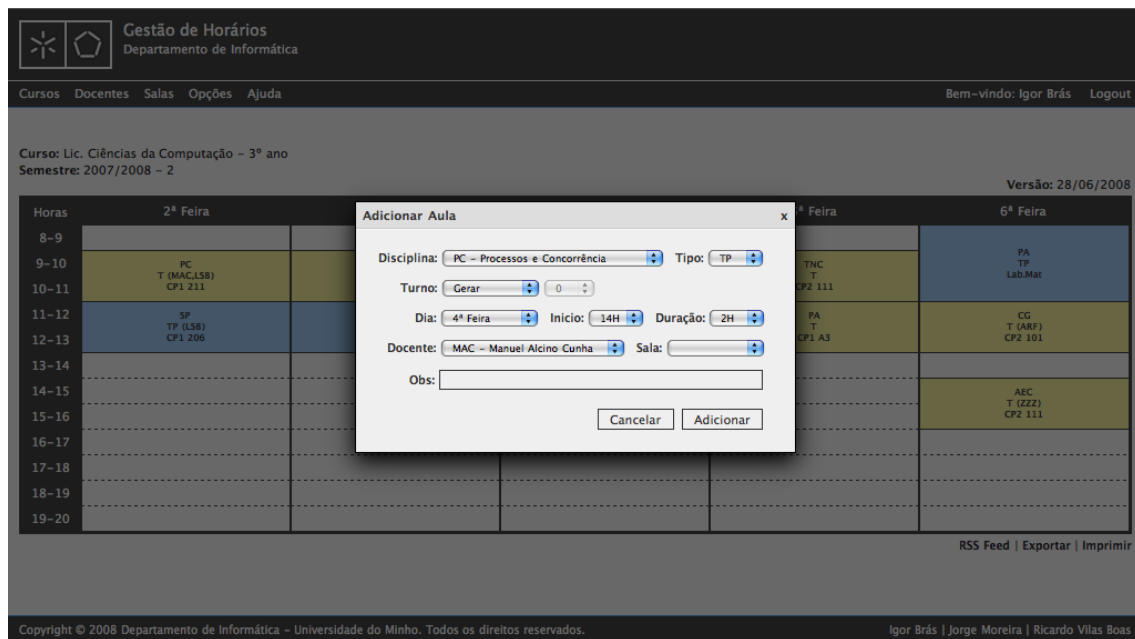


Figura 6.8: Interface para adição de uma aula.

Para adição de uma aula basta "clique" num espaço vazio do horário, introduzir os dados da aula e a aula é criada.

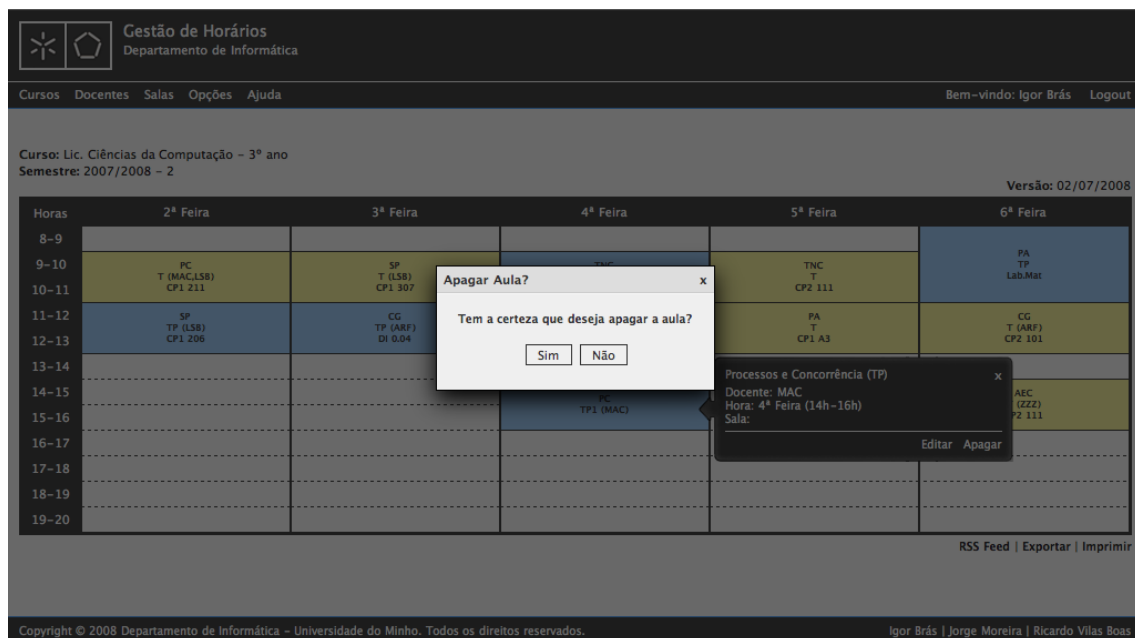


Figura 6.9: Interface para remoção de uma aula.

6.4 Mensagens de Aviso

Gestão de Horários
 Departamento de Informática

[Cursos](#) [Docentes](#) [Salas](#) [Opções](#) [Ajuda](#)

Bem-vindo: Igor Brás [Logout](#)

Curso: Lic. Engenharia Informática - 3º ano Existem incompatibilidades no horário!
 Semestre: 2007/2008 - 2 Versão: 01/07/2008

Horas	2ª Feira	3ª Feira	4ª Feira	5ª Feira	6ª Feira
8-9					
9-10		CC TP2 (ADC) DI 0.10	PL TP1 (PRH) DI 0.02	CC TP2 (ARF) DI 0.04	MEIO TP3 CP2 303
10-11				PL TP3 (PRH) DI 0.02	
11-12	CC TP1 (ADC) DI 0.10	SRCR TP1 (PJR) DI 1.04	CC TP3 (ADC) DI 0.10	PL TP2 (PRH) DI A2	CC T (ADC) CP2 101
12-13			CC TP7 (ARF) DI 0.04	MEIO TP4 CP2 303	CC T (ARF) CP2 101
13-14				MEIO T CP2 104	
14-15	MEIO TP1 CP1 209	SRCR TP2 (CAR) DI 1.04	SRCR T (JMN) DI A2		
15-16				PL T (PRH) CP1 A5	CC TP1 (ARF) DI 0.04
16-17	MEIO TP2 CP1 101	SRCR TP3 (CAR) DI 1.04	CC TP4 (EAD) DI 0.10	CG TP1 (ARF) DI 0.04	PL TP4 (PRH) DI 0.02
17-18				CC TP5 (PNS) DI 0.10	SRCR TP4 (CAR) DI 1.04
18-19					
19-20					

RSS Feed | Exportar | Imprimir

Copyright © 2008 Departamento de Informática - Universidade do Minho. Todos os direitos reservados.
Igor Brás | Jorge Moreira | Ricardo Vilas Boas

Figura 6.10: Aviso de colisão (quando existem incompatibilidades no horário).

Está representado aqui um caso de colisão do tipo do mesmo docente ter duas aulas há mesma hora, ou uma sala duplamente ocupada. Quando tal situações acontecem a aplicação Web detecta esses casos e apresenta uma mensagem de aviso.

Em baixo estão apresentados exemplos de mensagens de avisos, quando se efectuem determinadas acções.

Curso: Lic. Ciências da Computação – 3º ano
Semestre: 2007/2008 – 2

Aula criada com sucesso!

Versão: 02/07/2008

Horas	2ª Feira	3ª Feira	4ª Feira	5ª Feira	6ª Feira
8-9					PA TP Lab.Mat
9-10	PC T (MAC,LSB) CP1 211	SP T (LSB) CP1 307	TNC TP Lab.Mat	TNC T CP2 111	
10-11					
11-12	SP TP (LSB) CP1 206	CG TP (ARF) DI 0.04	PC TP (MAC,LSB) DI A2,DI 0.11	PA T CP1 A3	CG T (ARF) CP2 101
12-13					
13-14					
14-15			PC TP1 (MAC)		AEC T (ZZZ) CP2 111
15-16					
16-17					
17-18					
18-19					
19-20					

RSS Feed | Exportar | Imprimir

Figura 6.11: Mensagem de aviso (Exemplo 1).

Curso: Lic. Ciências da Computação – 3º ano
Semestre: 2007/2008 – 2

Aula 2544 apagada com sucesso!

Versão: 28/06/2008

Horas	2ª Feira	3ª Feira	4ª Feira	5ª Feira	6ª Feira
8-9					PA TP Lab.Mat
9-10	PC T (MAC,LSB) CP1 211	SP T (LSB) CP1 307	TNC TP Lab.Mat	TNC T CP2 111	
10-11					
11-12	SP TP (LSB) CP1 206	CG TP (ARF) DI 0.04	PC TP (MAC,LSB) DI A2,DI 0.11	PA T CP1 A3	CG T (ARF) CP2 101
12-13					
13-14					
14-15					AEC T (ZZZ) CP2 111
15-16					
16-17					
17-18					
18-19					
19-20					

RSS Feed | Exportar | Imprimir

Figura 6.12: Mensagem de aviso (Exemplo 2).

6.5 Exportação

The screenshot displays the 'Gestão de Horários' (Schedule Management) interface for the 'Lic. Engenharia Informática - 3º ano' (3rd year of the Informatics Engineering License) in the 'Semestre: 2007/2008 - 2' (2nd semester of 2007/2008). The interface shows a weekly schedule grid with columns for '2ª Feira', '3ª Feira', '4ª Feira', '5ª Feira', and '6ª Feira', and rows for hours from 8-9 to 19-20. A dialog box titled 'Exportar Horário' is overlaid on the grid, allowing users to export the schedule. The dialog box contains the following options:

- Formato: iCalendar XML HTML
- Data: Semestre Inteiro Definir Data
- Disciplinas: Todas Apenas as seguintes:

The dialog box also includes 'Cancelar' and 'Exportar' buttons. The interface footer includes 'Copyright © 2008 Departamento de Informática - Universidade do Minho. Todos os direitos reservados.' and 'Igor Brás | Jorge Moreira | Ricardo Vilas Boas'.

Figura 6.13: Interface de exportação do horário.

Nesta ultima imagem está representado a interface que ocorre quando é pedida a exportação. É possível exportar nos três formatos representados, o horário de semestre todo ou apenas de uma data a definir e de todas as disciplinas ou apenas as pretendidas.

7 Conclusões e Trabalho Futuro

De um modo geral, podemos afirmar que o grande objectivo subjacente na realização deste projecto prático, foi alcançado. Isto é, foram adquiridos conhecimentos importantes, numa área da tecnologia na qual não possuíamos ainda qualquer tipo de formação. Sendo uma área em forte desenvolvimento e com forte aplicação prática no mundo real, estamos certos que foi adquirida formação importante para colocar em prática a nível profissional futuro.

De modo um modo geral fizemos um excelente trabalho, conseguimos atingir com elevado sucesso os principais objectivos, pois estes foram alterados ao longo do projecto. Conseguimos fazer um trabalho muito útil para o futuro, pois como já referido anteriormente a aplicação usada neste momento para visualização dos horários é um pouco lenta e pouco interactiva, além do aspecto tratado da visualização conseguimos de uma forma interactiva editar o horário o que até aqui era conseguido através de inserção directa na base de dados.

De realçar, mas pela negativa, foi o problema que nos deparamos com a criação dos menus. Quisemos implementar uns menus, funcionais e também muito apelativos à vista de qualquer utilizador. Este menus tornaram-se um pouco complexos e surpreendentemente obtivemos resultados inesperados, como o não funcionamento (correctamente) no internet explorer. Mas mais uma vez lançamos o desafio e comprometemo-nos que na apresentação deste trabalho eles estarão a funcionar correctamente em todos os browsers.

Convém realçar que, embora o grande objectivo fosse o já indicado, tentamos colocar esses conhecimentos em prática para que da realização deste projecto surgisse algo mais para além de conhecimentos. Isto é, tudo foi feito para que, a aplicação Web desenvolvida, para além de utilizar as tecnologias já enumeradas e explicadas, fosse agradável ao utilizador, e foi nossa preocupação adicionar algumas funcionalidades que não era prioridade do projecto, como por exemplo, mais formatos de exportação e **RSS**.

Quanto à tecnologia empregue, apesar da dificuldade encontrada no início do projecto, foi aplicada de acordo com os padrões de desenvolvimento. De realçar o conceito **Web 2.0** que, não só do nosso ponto de vista mas de toda a comunidade Web, está e ainda irá revolucionar mais aplicações Web.

Como trabalho futuro poderiam ser adicionadas mais funcionalidades à aplicação, como por exemplo:

- Um dos aspectos principais de futuro seria tentar otimizar a base de dados, pois acreditamos que uma nova base de dados tornaria o nosso trabalho um pouco mais rápido.
- Expansão da aplicação para gestão dos horários de toda a Universidade, pois o tipo de implementação da aplicação permite uma fácil expansão.
- Possibilidade de importação através de ficheiros **XML**.
- Possibilidade de exportação para formato **CVS** (*MS Outlook*), para abranger mais utilizadores, apesar do standard ser **iCalendar**.

Como nota dominante no decorrer deste projecto, fica o facto de esta ser uma área com um leque enorme de potencialidades. Todo o conjunto de implementações existentes para as tecnologias usadas poderiam ser alvo de um estudo bastante mais aprofundado.

Bibliografia

- Lee Babin (2007)
Beginning Ajax with PHP
Apress
- Ryan Asleson, Nathaniel T. Schutta (2006)
Foundations of Ajax
Apress
- Steve Holzner (2006)
Ajax for Dummies
Wiley Publishing, Inc
- Eric A. Meyer (2006)
Cascading Style Sheets: The Definitive Guide
O'Reilly
- Richard Mansfield (2005)
CSS Web Design for Dummies
Wiley Publishing, Inc
- Bryan Pfaffenberger, Steven M. Schafer, Chuck White, Bill Karow (2004)
HTML, XHTML, and CSS Bible, 3rd Edition
Wiley Publishing, Inc
- Danny Goodman, Michael Morisson (2004)
JavaScript Bible, 5th Edition
Wiley Publishing, Inc
- Christian Heilmann (2006)
Beginning JavaScript with DOM Scripting and Ajax
Apress

- Tim Converse, Joyce Park, Clark Morgan (2004)
PHP and MySQL Bible
Wiley Publishing, Inc
- Janet Valade (2004)
PHP and MySQL for Dummies, 2nd Edition
Wiley Publishing, Inc
- Luke Welling, Laura Thomson (2005)
PHP & MySQL Web Development, 3r Edition
Sam's Publishing

Referências WWW

- 01 **<http://php.net/>**
Página oficial do PHP.
- 02 **<http://www.mysql.com/>**
Página oficial da base de dados utilizado neste projecto.
- 03 **<http://www.w3schools.com/>**
Página com diversos tutoriais sobre programação para web, abrangendo XHTML, CSS, PHP e JavaScript.
- 04 **<http://en.wikipedia.org/wiki/XHTML>**
Artigo da Wikipédia sobre XHTML.
- 05 **<http://www.cssbasics.com/>**
Página com uma boa introdução ao CSS.
- 06 **<http://sperling.com/examples/menuh/>**
Página com um tutorial sobre a construção de menus com CSS, usado nos menus da aplicação.
- 07 **<http://en.wikipedia.org/wiki/JavaScript>**
Artigo da Wikipédia sobre JavaScript.
- 08 **<http://www.javascript.com/>**
Página com inúmeros tutoriais sobre JavaScript.
- 09 **<http://www.ecma-international.org/publications/standards/Ecma-262.htm>**
Página com a especificação da linguagem ECMAScript (JavaScript).
- 10 **<http://www.webreference.com/programming/javascript/mk/column2/>**
Página com tutorial sobre "Drag and Drop" usando JavaScript.
- 11 **<http://www.w3.org/DOM/>**
Página da W3C com introdução ao DOM.
- 12 **<http://www.xul.fr/en-xml-rss.html>**
Página com uma pequena introdução ao formato RSS.

13 **<http://en.wikipedia.org/wiki/RSS>**

Artigo da Wikipédia sobre RSS.

14 **<http://www.wildbit.com/labs/modalbox/>**

Página com código fonte de como construir "Modal Box", usado na aplicação nas janelas de serviço.

15 **<http://en.wikipedia.org/wiki/iCalendar>**

Artigo da Wikipédia sobre o formato de eventos iCalendar.

16 **<http://www.ietf.org/rfc/rfc2445.txt>**

Texto com as especificações do formato iCalendar.

Lista de Acrónimos

BD Base de Datos

CSS Cascading Style Sheets

SQL Structured Query Language

XHTML Extensible Hypertext Markup Language

PHP Hypertext Preprocessor

Ajax Asynchronous Javascript And XML

WWW World Wide Web

HTML HyperText Markup Language

XML eXtensible Markup Language

RSS Really Simple Syndication

DOM Document Object Model

GUI Graphical User Interface