



**Escola Superior de
Tecnologia e de Gestão**
INSTITUTO POLITÉCNICO DE BRAGANÇA

**Pedro Gonçalves
Luís Vieira**

Portal AEPQ

Portal AEPQ

Relatório da Disciplina de Projecto
Engenharia Informática
Escola Superior de Tecnologia e de Gestão

Pedro Gonçalves, Luís Vieira
Setembro 2008

A Escola Superior de Tecnologia e Gestão não se responsabiliza pelas opiniões expressas neste relatório.

Certifico que li este relatório e que na minha opinião, é adequado no seu conteúdo e forma como demonstrador do trabalho desenvolvido no âmbito da disciplina de Projecto.

Paulo Alexandre Vara Alves Orientador

Certifico que li este relatório e que na minha opinião, é adequado no seu conteúdo e forma como demonstrador do trabalho desenvolvido no âmbito da disciplina de Projecto.

Arguente

Aceite para avaliação da disciplina de Projecto

Prefácio

O conteúdo deste relatório é a explicação do desenvolvimento de uma aplicação com fins académicos. Esta aplicação destina-se à criação de um portal para o agrupamento de Escolas Paulo Quintela, e numa segunda fase que consiste na investigação de várias formas de gestão de contas de utilizadores, bem como as suas sessões.

Este relatório contém também explicação de todos os métodos e linguagens usados na criação desta aplicação bem como uma explicação detalhada do seu funcionamento.

Agradecimentos

A realização deste projecto não seria possível sem a colaboração de algumas pessoas, às quais nos cabe agradecer:

- Ao Dr. Paulo Alves pela sua colaboração e ensinamentos, pois sem a sua preciosa ajuda nada, ou pouco, conseguiríamos fazer.
- À escola EB23 Paulo Quintela de Bragança, em particular aos professores Germano Lima, Acácio Lopes, Adriano Prada, e César Sá pela disponibilidade prestada na colaboração do novo sítio Web.
- Ao Evandro Pires Alves, pelo tempo disponibilizado e pela sua ajuda.

Conteúdo

1	Introdução	1
1.1	Definição de Objectivos	1
1.1.1	Sítio Web	1
1.1.2	Moodle.....	2
1.2	Requisitos	2
2	Sítio Web.....	3
2.1	Introdução.....	3
2.2	Instalação.....	5
2.3	Actualizar o Joomla!	6
3	Moodle.....	7
3.1	Introdução.....	7
3.2	Instalação.....	7
4	Desenvolvimento	9
4.1	Portal AEPQ.....	9
4.2	Integração do Joomla!/Moodle.....	11
4.2.1	Nível 1	11
4.2.2	Nível 2	15
4.2.3	Nível 3	19
Capítulo 5	25	
5	Conclusão.....	25
A	WampServer 2.0.....	28
A.1	Arquitectura.....	28

Lista de Figuras

Figura.1 – Diagrama da estrutura Hierárquico	3
Figura.2 – Diagrama de instalação das extensões do Joomla!	5
Figura.3 – FrontEnd Portal.....	10
Figura.4 – BackEnd Portal.....	10
Figura.5 – Configuração para utilização de uma base de dados externa no Moodle.	12
Figura.6 – Diagrama de representação de um acesso à base de dados.....	14
Figura.7 – Nova componente Joomla.....	15
Figura.8 – Pagina de Login CAS.....	20
Figura.9 – Browser inicial do Apache Tomcat.....	21
Figura.10 – Esquema Funcionamento CAS	21
Figura.11 – Ficheiro /etc/ldap/slapd.conf.....	23
Figura.12 – Esquema de autenticação.....	24
Figura.A1 – Significado da sigla Wamp5.....	28
Figura.A2 – Modelo Cliente-Servidor.....	29
Figura.A2 – Integração do PHP com o Apache.....	29

Capítulo 1

1 Introdução

1.1 Definição de Objectivos

Foram dois os principais objectivos deste projecto:

- A criação de um novo sítio Web para o Agrupamento de Escolas Paulo Quintela;
- Investigação de formas para integração do novo sítio Web com a plataforma Moodle;

1.1.1 Sítio Web

No início do projecto o objectivo definido foi o desenvolvimento de novas funcionalidades para o portal elaborado pelo nosso ex colega Duarte Câmara, de modo a integrar funcionalidades de uma Intranet através da disponibilização de serviços de workflow e de groupware.

Depois de uma primeira reunião com os professores responsáveis pelo sítio Web do Agrupamento de Escolas Paulo Quintela (AEPQ), chegou-se à conclusão que a melhor solução para o portal seria uma plataforma simples de utilizar e que tivesse já um historial ao nível da implementação em escolas. Depois de alguma pesquisa decidiu-se adoptar a plataforma Joomla! para a elaboração do sítio Web.

Esta foi uma escolha que se revelou bastante acertada uma vez que pouco tempo depois da nossa escolha a FCCN¹ decidiu criar um projecto-piloto com o Joomla!

Isto favoreceu o agrupamento uma vez que o tornou desde logo pioneira neste projecto-piloto, pois eram poucos os agrupamentos com esta envergadura a optarem por seguir este projecto.

¹ FCCN – Fundação para a Computação Científica Nacional

1.1.2 Moodle

A plataforma de E-Learning Moodle já tinha sido adoptada pelo AEPQ, estando esta plataforma em funcionamento e administrada por um dos professores do Agrupamento.

O que se pretendia com este projecto era que houvesse uma interligação da plataforma Joomla! com a plataforma Moodle, de maneira a que os utilizadores só precisassem de se registar no Joomla! e automaticamente terem também a mesma conta no Moodle. Simultaneamente, pretendia-se que fizessem apenas um *login* para as duas plataformas, isto é, ao efectuar o *login* no Joomla! que fosse usada a mesma sessão no Moodle.

1.2 Requisitos

Como requisito, foi necessária a instalação de um servidor Web, e foi adoptado o WampServer 2.0.

Consultar o Apêndice A para mais informações sobre o WampServer 2.0.

Capítulo 2

2 Sítio Web

2.1 Introdução

Como foi dito anteriormente, para a elaboração do novo sítio Web do AEPQ foi adoptada a plataforma Joomla!

Joomla! é um sistema de gestão de conteúdos Web, em inglês *Content Management System* (CMS), que integra ferramentas necessárias para criar, editar conteúdos em tempo real sem a necessidade de programação de código.

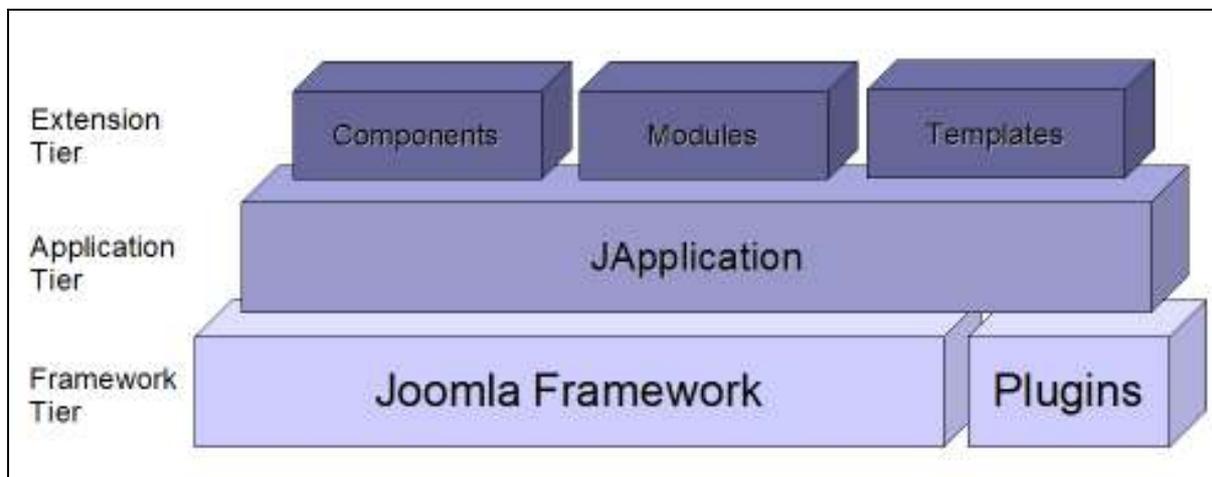


Figura.1 – Diagrama da estrutura Hierárquico

A plataforma Joomla! é cada vez mais utilizada por parte dos utilizadores que pretendem fazer um sítio Web. É de fácil instalação, permite a utilização de *Templates* de forma a alterar o visual do sítio, e a administração é feita de uma maneira simples.

O Joomla! é o CMS em maior expansão, sendo provavelmente o CMS mais procurado, com a maior comunidade e recursos disponíveis. A grande vantagem do Joomla! é a sua diversidade de extensões extras.

De referir também que o Joomla! é um projecto de código aberto, com licença GNU/GPL².

Devido a esse facto é que esta plataforma pode ser excepcional do ponto de vista do programador uma vez que a aplicação permite que o programador/administrador pode acrescentar praticamente todas as funcionalidades que lhe parecem necessárias, esta opção só possível devido extensões.

E o que são as extensões Joomla!?

Estas extensões são qualquer programa ou aplicação que possam aumentar o leque de funcionalidades do Joomla, podendo estas serem escolhidas entre as muitas disponibilizadas pela comunidade, ou então, se o programador preferir, pode criar uma de raiz instalando-a depois na sua plataforma.

As extensões então divididas em vários grupos como as componentes, plugins, módulos, idiomas e temas, sendo os temas e os idiomas um tipo de extensão mais ligada à customização do interface.

As restantes são mais as principais responsáveis pelo aumento das funcionalidades e criação de novas funcionalidades para o Joomla.

- **Componente:** é uma extensão que melhora o núcleo de funcionalidades do Joomla CMS, este tipo de extensão é criada quando se quer introduzir uma funcionalidade que seja vista no conteúdo principal do site, quer seja este no BackEnd ou então no FrontEnd;
- **Plugin:** este tipo de extensão é utilizada para estender as funcionalidades do site, como exemplo criar formas de introduzir conteúdos, usando o plugin de editor de texto, ou então criar formas de autenticação, etc.
- **Módulo:** é na sua maioria das vezes um programa que serve para criar todo o conteúdo em volta do nosso site, quer seja o aspecto, as estatísticas, as pesquisas, os destaques os menus, basicamente tudo o que rodeia o conteúdo principal.

² GNU/GPL - General Public License. Designação da licença para software livre

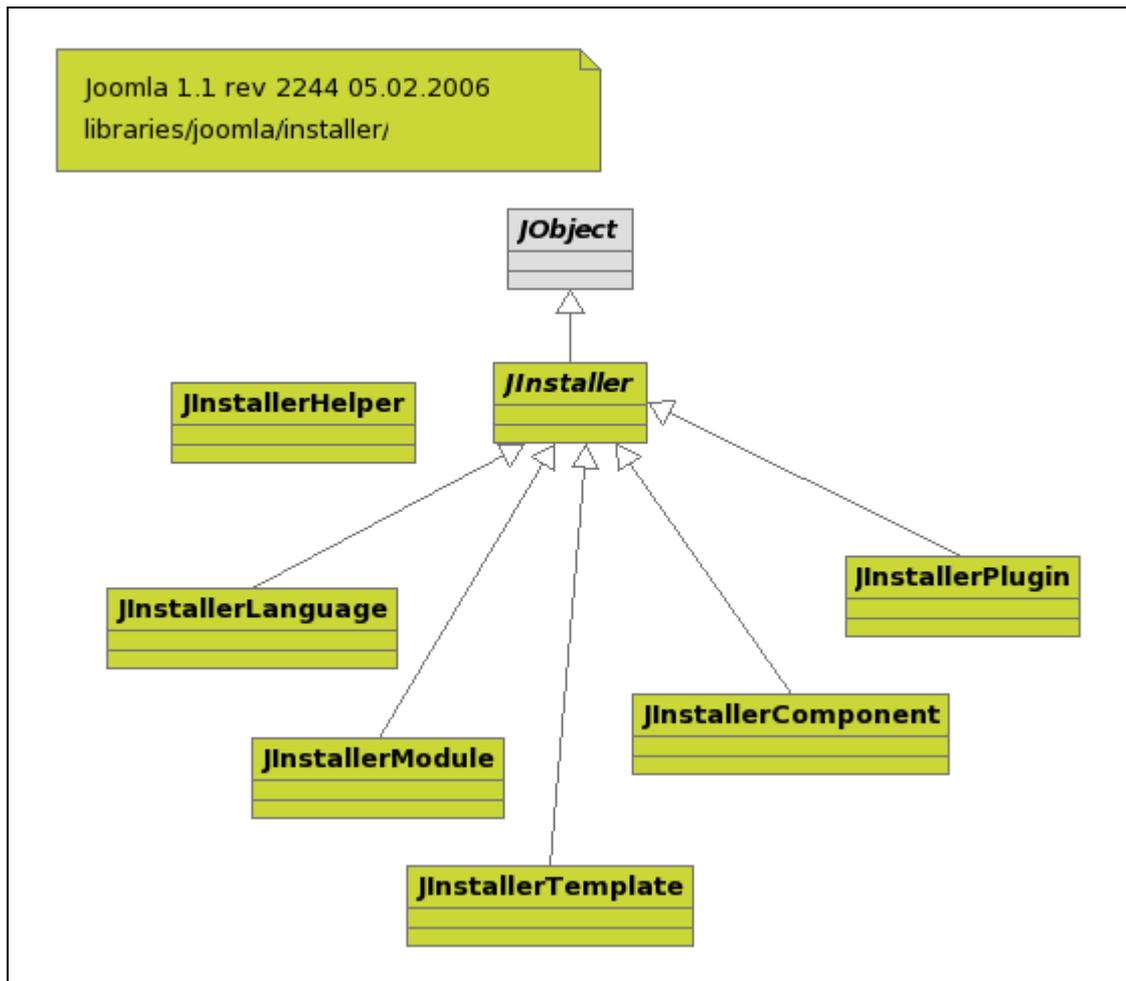


Figura.2 – Diagrama de instalação das extensões do Joomla!

Para obter mais informações sobre Joomla, consultar o anexo “Manual do Utilizador”.

2.2 Instalação

Antes de mais importa dizer que a instalação de que estamos a falar é uma instalação feita na nossa máquina, em “localhost”, e não num servidor externo. Como o AEPQ não tem servidor próprio, utiliza os recursos disponibilizados pela FCCN para as escolas alojarem o seu sítio na Internet.

O processo de instalação da plataforma Joomla! é muito simples. De seguida é exemplificado todo o processo.

1. Em primeiro lugar, tem de se descarregar o ficheiro de instalação. Este ficheiro pode ser encontrado em <http://www.joomlapt.com>.
2. Depois de descarregar o ficheiro, extrair para a pasta “c:\Wamp\www”.
3. Iniciar o WampServer.
4. De seguida colocar o seguinte endereço no navegador Web: “localhost”.
5. Seleccionar em “Your Projects” a pasta para onde foi extraído o ficheiro Joomla.

6. A partir deste ponto é visualizado um menu para a instalação guiada do Joomla, onde se configura o tipo de base de dados, o nome do servidor, nome de utilizador, palavra passe, e nome da base de dados.
7. Finalizada a instalação, o sítio Joomla! fica disponível em “localhost\

2.3 Actualizar o Joomla!

Como a plataforma Joomla! está em constante evolução, é necessário ir actualizando a plataforma.

O processo de actualização é muito simples, apenas tem-se de aceder ao sítio da comunidade Joomla! Português e descarregar o pacote de actualização referente à ultima versão.

De referir que só se pode actualizar para uma versão acima, por exemplo se a nossa plataforma for a versão 1.5.2 e a última versão for a 1.5.5 teremos de fazer a actualização faseada, primeiro actualizamos para a versão 1.5.4 e só depois para a versão 1.5.5.

Depois de descarregar o ficheiro o processo de actualização é necessário seguir os seguintes passos:

- Descompactar o ficheiro.
- Colocar os ficheiros resultantes do ficheiro anterior, na pasta do projecto.
- Quando pedir para confirmar a substituição dos ficheiros, escolher sim

Depois de efectuados estes passos, a plataforma fica actualizada.

Para verificar se a actualização foi bem sucedida, deve-se aceder ao BackEnd do sítio Web e verificar no canto superior direito se a versão actual é a aquela para a qual se pretendia actualizar o Joomla!.

Nota: o portal alojado na FCCN não é necessário fazer esta operação de actualização, isto porque a administração da FCCN faz a actualização do Joomla! sempre que haja uma nova versão.

Capítulo 3

3 Moodle

3.1 Introdução

O Moodle³ é um software livre e gratuito, com licença GNU/GPL, de apoio à aprendizagem executado num ambiente virtual.

Pode ser instalado em diversos ambientes, entre eles Windows, Unix, Linux, Mac OS. Podem também ser utilizados vários tipos de base de dados, como MySQL, PostgreSQL, Oracle, Access, Interbase ou ODBC. Está disponível em vários idiomas, incluindo o Português.

O conceito Moodle foi criado em 2001 pelo professor, e cientista computacional, Martin Dougiamas. É uma plataforma voltada para programadores e académicos da educação, e é constituído por um sistema de administração de actividades educacionais destinado à criação de comunidades *online* em ambientes virtuais, voltados para a aprendizagem colectiva.

Esta plataforma permite, de maneira simplificada, a um estudante ou a um professor integrar-se, estudando ou leccionando, num curso *online* à escolha.

A versão estável mais recente é a 1.9.2, do dia 11 de Julho de 2008.

3.2 Instalação

A instalação da plataforma Moodle é em tudo parecida à instalação da plataforma Joomla.

Para tal, é necessário fazer os seguintes passos:

1. Descarregar da Web o ficheiro de instalação do Moodle. Pode ser encontrado no seguinte endereço: <http://download.moodle.org/>.
2. Descomprimir o arquivo.

³ Moodle – Modular Object Oriented Dynamic Learning Environment

3. Transferir o directório, obtido no ponto anterior, directamente para o directório “www” do WampServer (“../wamp/www”).
4. Executar o script de instalação (“install.php”) ou seja, colocar o endereço localhost/moodle/install num navegador Web.

Depois de efectuar toda a configuração, e Moodle fica acessível no endereço “localhost/moodle”.

Capítulo 4

4 Desenvolvimento

4.1 Portal AEPQ

Para o agrupamento foi desenvolvido o portal com base no Joomla!.

Para criação deste sítio Web foi primeiro escolhida uma template e depois foi modificada consoante o tipo de instituição que a ia utilizar.

Depois de alguma pesquisa e reflexão, foi escolhida a template “Chamaeleon”. Em seguida procedeu-se à instalação d template, o qual foi alterado de modo a corresponder ao objectivo final de ser um sítio Web para uma instituição de ensino.

Todo o site foi reestruturado e o seu conteúdo proveniente do anterior site utilizado pelo agrupamento.

Esta opção foi adoptada para a transição de portal não causar uma mudança muito abrupta para os gestores do site, que não possuem formação específica na área das tecnologias de informação e comunicação.

Numa segunda fase do projecto, após ter sido testado exaustivamente, procedeu-se à sua publicação no endereço “<http://portal.eb23-paulo-quintela.rcts.pt/>”.

De modo a facilitar a gestão do portal foi feita uma sessão de formação no dia 5-Setembro-2008 com os administradores do portal (professores do agrupamento) e com o vice-presidente Acácio Lopes.

Nesta formação foi entregue o manual do utilizador⁴ onde explica como utilizar o FrontEnd e o BackEnd do portal.

⁴ Ver em Anexo

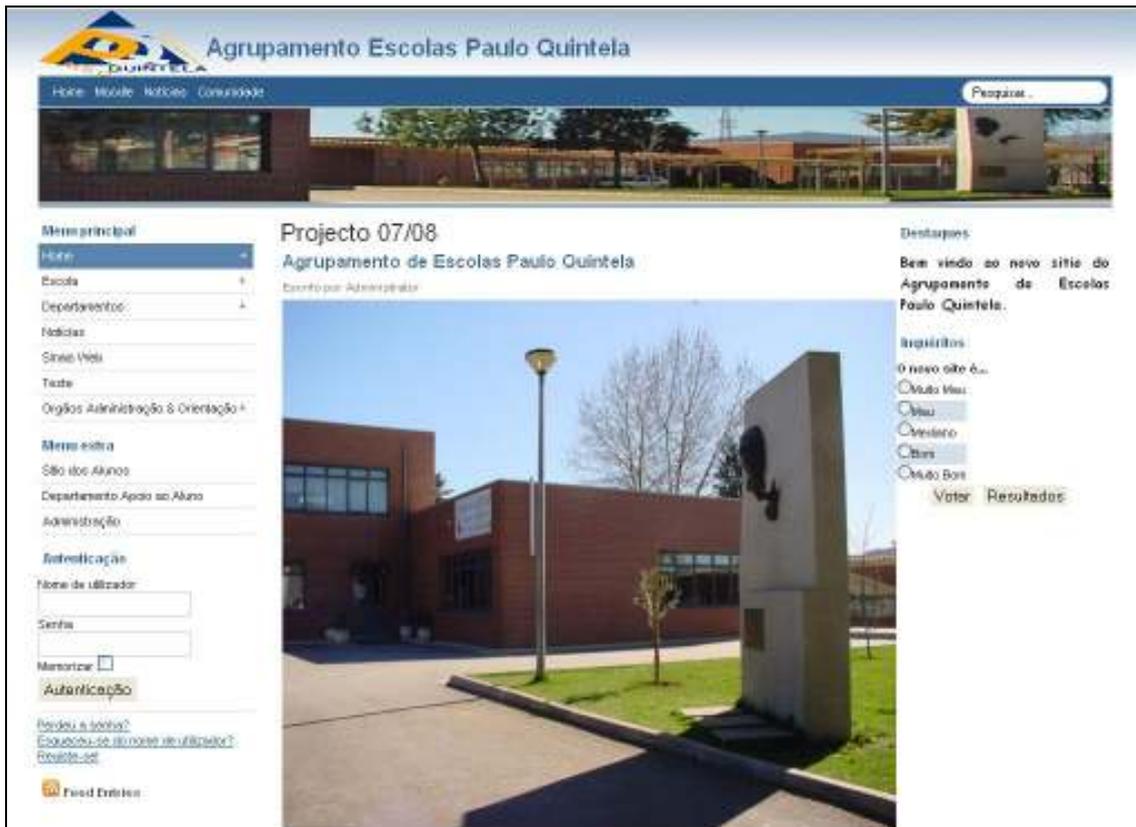


Figura.3 – FrontEnd Portal.



Figura.4 – BackEnd Portal.

Nesta formação além da entrega do manual, foi também explicado como estava organizada a página e demonstrado com exemplos práticos como utilizar as várias ferramentas que o Joomla! disponibiliza para o criação e alteração de conteúdos, gestão dos utilizadores, criação de menus e sua associação a conteúdos, bem como a explicação dos grupos de utilizadores disponíveis e quais os mais adequados a cada caso.

Posto isto, ficava então concluída a primeira fase do projecto, cujo objectivo era munir o agrupamento de um portal dinâmico, inovador e de fácil utilização, uma vez que seriam os próprios docentes a fazer a administração integral do sítio Web.

4.2 Integração do Joomla!/Moodle

Como já havia sido referido anteriormente, além do portal para o agrupamento haveria uma outra abordagem do projecto que consistia na investigação de métodos para criar uma interligação entre as duas plataformas que o agrupamento iria usar, ou seja, o Joomla! e o Moodle.

Esta interligação seria feita ao nível dos utilizadores, onde utilizando o portal como forma de contacto preferencial com o agrupamento, teriam depois acesso aos conteúdos disponibilizados no E-Learning.

A principal ideia era facilitar ao utilizador o acesso à informação, onde estaria incluído o melhoramento dos métodos de registo bem como os de autenticação.

Com a conclusão da parte de pesquisa foram considerados três níveis de integração, níveis estes, que apesar de serem diferentes, podiam ser considerados como evoluções uns dos outros, uma vez que com a conclusão de cada nível surgiam sempre algumas ideias para melhorar essas mesmas interligações entre as plataformas.

4.2.1 Nível 1

Na tentativa de começar a elaborar o primeiro nível deparou-se logo com o primeiro problema, pois as plataformas utilizavam bases de dados distintas, o que impossibilitava um utilizador registado só no portal aceder aos conteúdos do Moodle, e se este quisesse aceder a estes conteúdos teria de se registar nesta plataforma.

Depois de algum estudo do problema, e da análise do Moodle, descobriu-se uma forma de contornar o problema que consistia na configuração de autenticação dos utilizadores onde esta oferecia a possibilidade da autenticação ser feita numa base de dados externa, ou seja podia-se escolher a bases de dados do Joomla! e então a autenticação no Moodle era feita através da tabela de utilizadores registados no Joomla!.

Use uma base de dados externa

Este método usa uma tabela numa base de dados externa para verificar se um nome de utilizador e senha são válidos. Se for uma conta nova, a informação de outros campos pode ser também transferida para Moodle.

Servidor:
 Base de Dados:
 Usar apêndice da "base":
 Nome da Base de Dados:
 Utilizador da Base de Dados:
 Senha:
 Tabela:
 Campo do nome do utilizador:
 Campo de senha:
 Formato da senha:
 Codificação da base de dados externa:
 Comando de configuração de SQL:
 Depurador ADOdb:
 URL para alteração de senha:
 Script de sincronização com:
 Remover utilizador externo:

Endereço IP do computador que hospeda a base de dados de utilizadores:
 O tipo de base de dados (veja Documentação do ADOdb para mais pormenores)
 Estilo "Sybase" para proteger apóstrofos -- necessário para ORACLE, My SQL e outras bases de dados. Não usar para MySQL!
 Nome da própria base de dados:
 Nome de utilizador para aceder à base de dados:
 Senha para o utilizador acima:
 Nome da tabela na base de dados:
 Nome do campo que contém os nomes de utilizadores:
 Nome do campo que contém as senhas:
 Indique o modo que se está a usar no campo de senha. A criptografia MD5 é útil para trabalhar com outras aplicações como PostgreSQL.
 Codificação usada na base de dados externa:
 Comando SQL para configurar/alterar de base de dados, regulamentação usado para configurar a codificação de comunicação - exemplo para MySQL e PostgreSQL: `CHARACTER SET utf8`
 Depurador da ligação ADOdb para base de dados externas - usar quando receber página vazia durante a autenticação. Não é aconselhada para sítios em produção.
 Aqui pode especificar um local onde os utilizadores podem recuperar ou alterar a sua senha e nome de usuário caso se esqueçam dela. Isto será fornecido aos utilizadores como um botão na página de entrada e verificador e na sua página de utilizador. Se deixar este espaço em branco o botão não aparecerá.
 Especifique o que fazer com contas inibidas de utilizadores durante sincronização em massa quando o utilizador foi removido da fonte externa. Se utilizadores suspensos são automaticamente reativados se eles

Figura.5 – Configuração para utilização de uma base de dados externa no Moodle.

Nesta configuração é necessário indicar todos os parâmetros da base de dados externa, ou seja, o nome do servidor, o tipo de base de dados, o nome da base de dados bem como a tabela que seria usada para a autenticação.

Configurados estes campos é necessário ainda fornecer a este formulário o nome dos campos utilizados na autenticação, que no caso eram os campos *username* e *password*.

Para que fosse possível fazer este tipo de autenticação teve-se de contornar uma outra questão que se prendia com a encriptação das passwords. Enquanto que o Joomla! utiliza a encriptação Hash-Md5, o Moodle utiliza apenas Md5⁵ o que impossibilita a sincronização das tabelas de utilizadores. Posto isto, e uma vez que se estava a utilizar a base de dados do Joomla!, tal comparação nunca se iria verificar pois as *passwords* estavam encriptadas de maneira diferente.

Para contornar este problema foi então criada uma nova tabela na base de dados do Joomla! e criado um novo bloco de código que foi introduzido num dos ficheiros do Joomla! (“controller.php”), código este desenvolvido em PHP fazendo uso de classes para que a introdução de código no ficheiro existente fosse mínima, sendo assim necessário apenas a introdução de uma única linha de código que por sua vez invoca os ficheiros necessários para a execução do código pretendido.

Os novos ficheiros (“instancia.php”, “class.php”) apenas necessitavam de ser guardados nesta localização “Joomla\components\com_user”.

O código principal que se encontrava no ficheiro “class.php” permite que no caso do registo de um novo utilizador no Joomla! fosse criado um novo campo na nova tabela (“jos_pass”)

⁵ MD5 - Message-Digest algorithm 5

com o *username* e a sua *password*, esta já encriptada em Md5. Assim, torna-se possível a utilização desta tabela por parte do Moodle para fazer a autenticação dos utilizadores.

Sendo assim no ficheiro existente teria de se colocar o seguinte código:

```
require_once('instancia.php');
```

Que por sua vez iria chamar o ficheiro *instancia.php* com o seguinte código:

```
<?php
/**
 * @author Luis Vieira & Pedro Gonçalves
 * @copyright 2008
 */

    $us=$_POST['username'];
    $email=$_POST['email'];
    if(isset($_POST["b"]))
    {
        require_once('class.php');
        $data = new guardarPassMd5;
        $data->teste($us,$email);
    }

?>
```

Aqui fez-se o include do ficheiro *class.php* bem como a criação de uma instância da classe existente neste para que por fim se pudesse executar a função que continha o código pretendido:

```
<?php
    $bool=0;
    class guardarPassMd5
    {
        function teste($us,$email)
        {
            $flag=0;
            $count1=0;
            mysql_connect('localhost', 'root');
            mysql_select_db('apq');
            $query1="SELECT * FROM jos_users";
            $result1=mysql_query($query1);
            $cont2 = mysql_num_rows($result1);
            while($row5=mysql_fetch_object($result1))
            {
                if(($row5->username != $us)&&($row5->email !=
                $email)){
```

```

        $count1=$count1+1;
    }else{
        $flag=1;
        print("flagg");
    }
}
if($flag==0){
    $pw=$_POST['password2'];
    $passCR=md5($pw);
    mysql_select_db('apq');
    $query="INSERT INTO jos_pass (user, pass_md5)
VALUES ('$us','$passCR')";
    $result=mysql_query($query);
}
}
}
?>

```

Este segmento de código começa por armazenar a informação submetida no formulário de registo, fazendo de seguida as comparações necessárias para verificar se a informação podia ser armazenada, ou seja, testava se o username e o email já estavam a ser utilizados. Se esta informação estivesse presente num outro registo então nada era feito, e a informação era descartada. Caso contrário era feita então uma ligação à base de dados e armazenada na nova tabela o valor do campo *username* e *password*, estando a *password* guardada encriptada em Md5 para uma futura utilização por parte do Moodle.

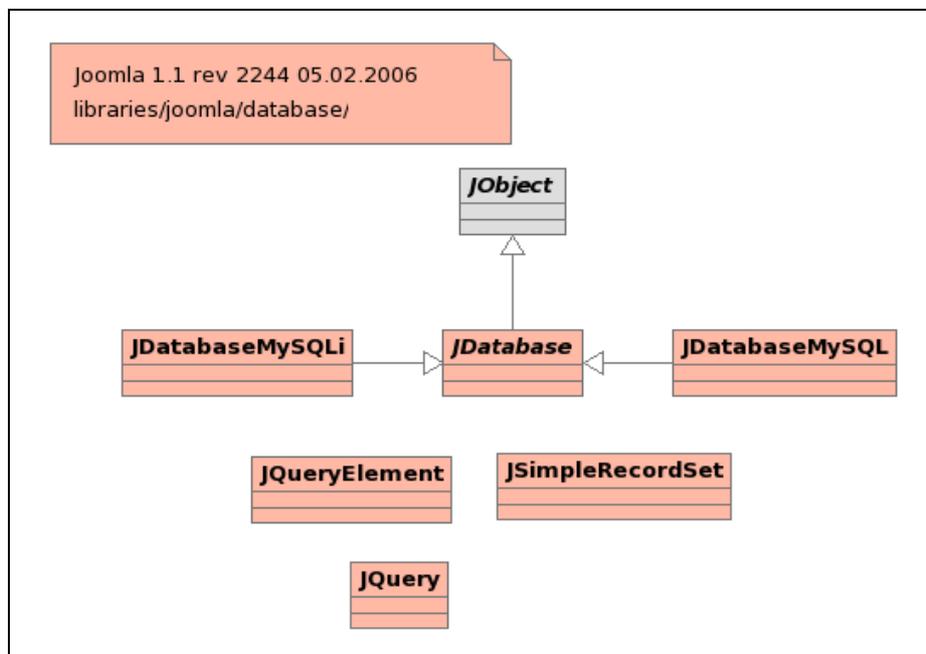


Figura.6 – Diagrama de representação de um acesso à base de dados.

Terminadas estas configurações torna-se possível aos utilizadores do Joomla! fazer login no Moodle sem que tivessem de fazer nenhum novo registo.

4.2.2 Nível 2

Apesar do nível 1 estar funcional pensou-se que se poderia melhorar, estudar uma forma de integração bidireccional desta integração.

Então partiu-se para uma nova forma de integração usando o código já criado no nível 1. Decidiu-se que cada plataforma deveria manter a sua base de dados inicial, onde seriam feitas as autenticações dos utilizadores nessas mesmas bases de dados, mas sempre pensando no utilizador e como ponto de partida o portal, ou seja, deveria ser possível o utilizador apenas se registar no portal e com esse mesmo registo ficar automaticamente registado no Moodle ficando apenas depois a faltar a sua activação e atribuição de uma categoria de utilizador que seria feita posteriormente pelo administrador da plataforma Moodle.

Após mais algum tempo de pesquisa descobriu-se uma das maiores potencialidades que o Joomla! pode oferecer ao utilizador mais avançado.

Trata-se da criação de componentes que posteriormente podem ser instaladas no Joomla! ficando a cargo do seu criador onde estas podem ser publicadas, se só na parte de administração, se no FrontEnd do portal ou então nos dois, tornando-se simples a configuração, pois apenas se tem de criar um ficheiro XML onde sejam especificadas essas opções.

Esta é uma enorme vantagem pois no caso do portal precisar de uma funcionalidade que ainda não tenha sido criada, o administrador pode por si criar uma componente que faça essa nova função pretendida.

As componentes são criadas como ficheiros de instalação o que permite serem instaladas no portal no caso de este, por exemplo, sofrer algum problema e tenha de ser reinstalado, sendo apenas depois necessário reinstalar a componente criada para o efeito desejado.

Posto isto, foi encontrada então a solução para o problema acima referido, onde seria criada uma componente que exportasse os utilizadores do Joomla! para a base de dados do Moodle e já com a sua password encriptada em Md5 uma vez que já tínhamos criado uma tabela anteriormente para esse efeito.

Esta componente ficou apenas disponível no BackEnd, ou seja na parte de administração, pois só assim o administrador teria o controlo dessa exportação.



Figura.7 – Nova componente Joomla

Antes propriamente da criação da componente, foi necessário desenvolver o código que esta utilizaria para desempenhar a função pretendida, que foi o seguinte:

```
<?php
/**
 * @author Luis Vieira & Pedro Gonçalves
 */

// impedir acesso directo
defined( '_JEXEC' ) or die ( 'Área de acesso restrito' );

JToolBarHelper::title(JText::_("M&ocirc;dule Exporta&ccedil;&atilde;o
de Utilizadores" ), 'user.png' );

print("<table>");
    print("<tr>");
        print("<td width='100%'>");
            print("*Este m&ocirc;dule extra joomla
tem como objectivo fazer a exporta&ccedil; dos utilizadores deste
plataforma para o base de dados do Moodle");
        print("</td>");
    print("</tr>");
    print("<tr>");
        print("<td width='100%' align='right'>");
            print("<form id='frmEXP' align='center'
name='frmEXP' action='$_PHP_SELF' method='post'>");
            print("<br>");
            print("<button name='EXP' type='submit'>
Exportar Novos Utilizadores </button>");
            print("</form>");
        print("</td>");
    print("</tr>");
print("</table>");

if(isset($_POST['EXP'])) {
    mysql_connect('localhost', 'root');
    $nov=0;
    mysql_select_db('apq');
    $query1="SELECT * FROM jos_users";
    $result1=mysql_query($query1);
    while($row1=mysql_fetch_object($result1))//for
($y=0 ;$y < $n; $y++)
    {
        //$row1=& $this->items[$y];
        $count=0;
        if ($row1->lastvisitDate == "0000-
00-00 00:00:00")
        {
```

```

mysql_select_db('moodle');
$query="SELECT * FROM mdl_user";
$result=mysql_query($query);
$cont = mysql_num_rows($result);
while ($row2 = mysql_fetch_object($result))
{
    if($row2->username != $row1->username)
    {
        $count=$count+1;
    }
}
if($count==$cont)
{
    mysql_connect('localhost', 'root');
    mysql_select_db('apq');
    $query2="SELECT * FROM jos_pass WHERE user='$row1->username'";
    $result2=mysql_query($query2);
    $row3 =mysql_fetch_object($result2);
    $nov=$nov+1;
    mysql_select_db('moodle');
    $query="INSERT INTO mdl_user
username,password,mnethostid,firstname,email) VALUES ('$row1-
>username','$row3->pass_md5','1','$row1->name','$row1->email')";
    $row2=mysql_query($query);
}
}
}
if($nov==0)
{
    print("***Nao existem novos utilizadores para exportar!");
}else{
    ?>
    <script>
        window.alert("Utilizadores exportados com sucesso!");
    </script>
    <?php
}
}
?>

```

O código acima colocado começa por estabelecer o aspecto da nova componente para que esta se adapte ao tipo de interface que tem o restante BackEnd.

Em seguida são criados os elementos que serão visíveis na componente por nós criada, a execução do código é efectuada quando o administrador pressiona o botão de exportação de utilizadores. A partir desse ponto é criada uma conexão à base de dados do Joomla! e escolhida a condição que define um novo utilizador, ou seja, todo o utilizador que nunca tenha feito um *login* no portal é considerado como um novo utilizador. Através da selecção do respectivo *username* e feita uma ligação a base de dados do Moodle onde se vai fazer mais uma condição que consiste em testar se o login já existe na tabela de utilizadores do Moodle. Caso não exista nenhum utilizador com o mesmo *login* da condição de teste, é criado então um novo utilizador com os mesmos parâmetros do utilizador do Joomla! sendo depois só necessário o administrador desta plataforma activar este novo membro e atribuir-lhe um estatuto, caso seja necessário.

No caso de a condição dar falso então significa que o utilizador já existe, ou seja, já tinha sido exportado numa sessão anterior e ainda não tinha feito nenhum login. Esta operação é efectuada para todos os novos utilizadores aquando do clique no botão de exportação de utilizadores existente na nova componente.

Após estas operações o utilizador já pode fazer *login* na plataforma Moodle sem necessitar de se registar novamente nesta, simplificando assim todo o processo de registo de utilizadores.

Falando novamente da criação da componente o programador apenas necessita de criar mais um ficheiro em XML, preferencialmente com o mesmo nome do ficheiro php com o código da componente e definir os parâmetros desta, da seguinte forma:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<install type="component" version="1.5.0">
  <name>Adicional</name>
  <creationDate>27.08.2008</creationDate>
  <author>Luis Vieira _ Pedro Gonçalves</author>
  <copyright>Projecto 2008</copyright>
  <authorEmail>vieira902@hotmail.com _ pedroerg@gmail.com</authorEmail>
  <authorUrl></authorUrl>
  <version>1.0 J1.5 RC4</version>
  <license>x</license>
  <description>Componente adicional de exportação de utilizadores para o moodle!</description>
  <administration>
    <menu>Exportar</menu>
    <files>
      <filename>adicional.php</filename>
    </files>
  </administration>
</install>
```

Neste ficheiro XML é definida e descrita a função da nova componente bem como a sua localização, neste caso como foi definido que a componente só estaria disponível na parte de administração para que tal fosse possível bastou apenas colocar entre tag's “<administration> </administration>” o carregamento do ficheiro adicional.php.

Caso se pretenda que a componente esteja também disponível no FrontEnd do portal basta apenas colocar entre as tag's “<files> </files>”.

Após os passos acima referidos apenas é preciso comprimir os dois ficheiros num formato ZIP, tornando-se assim num ficheiro de instalação pronto a ser carregado para a plataforma Joomla!.

Terminada assim a criação da componente é possível constatar que uma das grande potencialidades do Joomla! está na adição de componentes, uma vez que introduz o conceito de programação modular em plataformas opensource, possibilitando aos programadores estenderem as funcionalidades destes sistemas e partilharem facilmente entre si novas componentes.

4.2.3 Nível 3

Após a conclusão do nível de integração anterior foram estudadas várias formas de melhorar a autenticação no portal.

Uma das mais sugestivas foi o SSO (Single Sign On) que iria consistir no seguinte: um utilizador já registado no portal, e após a sua exportação para a plataforma Moodle, teria a possibilidade de efectuar a autenticação centralizada, permitindo o uso da mesma sessão para aceder ao Joomla! e ao Moodle.

Ou seja, tendo como ponto de partida o portal, o utilizador só necessitaria de fazer o *login* uma única vez para ter acesso às duas plataformas.

Para que esta nova solução fosse possível de ser implementada, tínhamos de utilizar novas aplicações para a gestão de sessões.

Numa primeira abordagem pensou-se na utilização de Shibboleth.

Este é um sistema que possibilita a autenticação do utilizador no portal, enviando ao sistema de destino os atributos necessários, sem que dados sensíveis (login/senha, chave privada, etc.) transitem por este último.

Mas após alguma pesquisa, verificou-se que o Shibboleth tem uma estrutura bastante complexa que seria incompatível com os recursos existentes nas escolas. Assim, foi estudado o CAS⁶ como alternativa, que é também um sistema de autenticação que permite fazer um *login* seguro numa instituição fazendo depois o *forward* da informação necessária para as restantes partes do sistema onde seja preciso a autenticação do utilizador.

⁶ CAS – Central Authentication Service

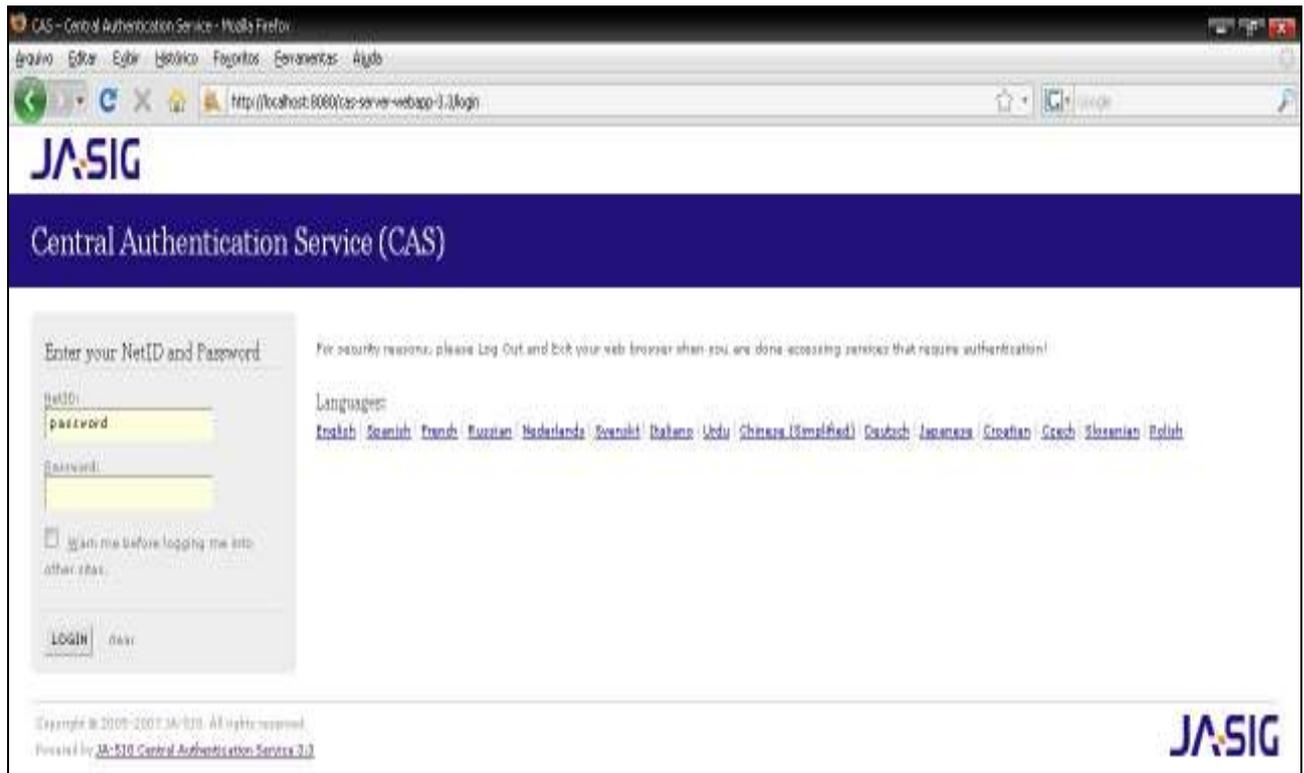


Figura.8 – Pagina de Login CAS.

Esta aplicação tem alguns pré-requisitos para o seu funcionamento, como o caso de ser baseado em JAVA.

Para satisfazer este requisito foi então instalado o **Tomcat 5.5**, que é um servidor de aplicações Java para a Web, tal como é requisitado no CAS.

Este é um software distribuído livremente e desenvolvido como código aberto dentro do conceituado projecto Apache.

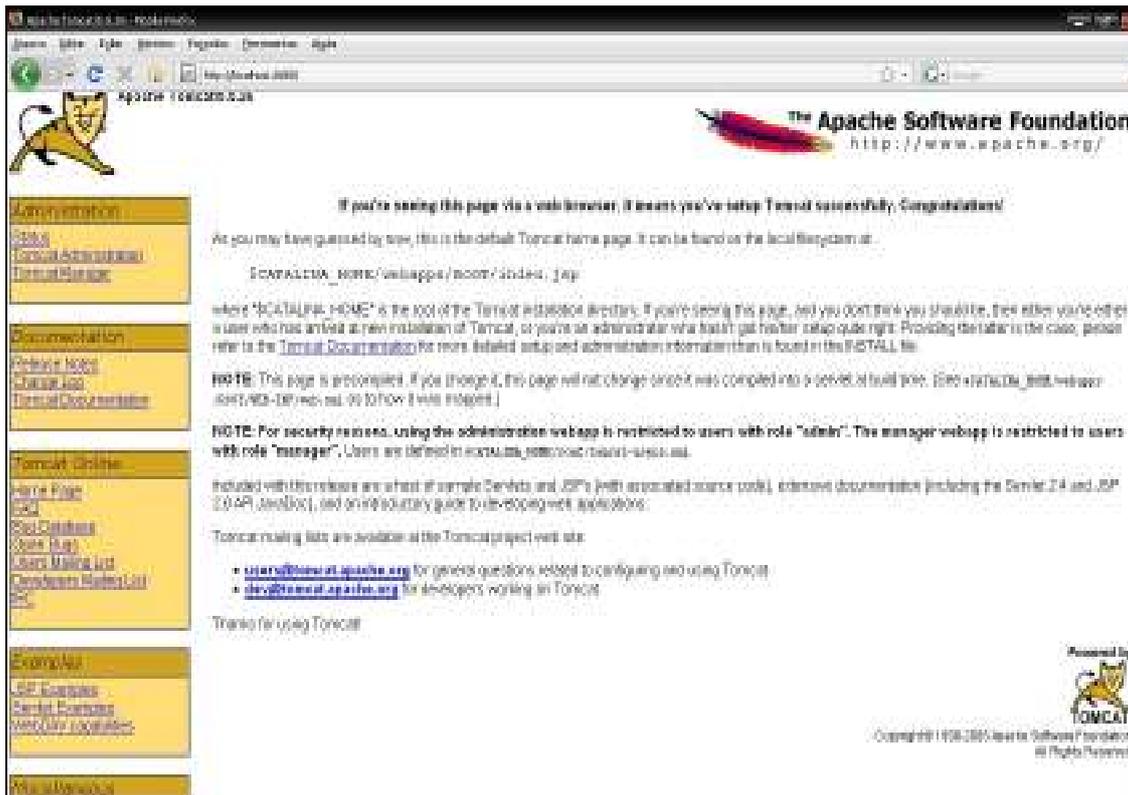


Figura.9 – Browser inicial do Apache Tomcat

O funcionamento do CAS é bastante simples, tem apenas de estar associado ao portal, neste caso foi instalada uma componente no Joomla! em que associa o portal ao CAS, e quando este efectua o *login* é redireccionado para o CAS onde faz a validação da autenticidade do utilizador, normalmente usando o *username* e *password*. Se a autenticação for bem sucedida o CAS volta a redireccionar para a aplicação.

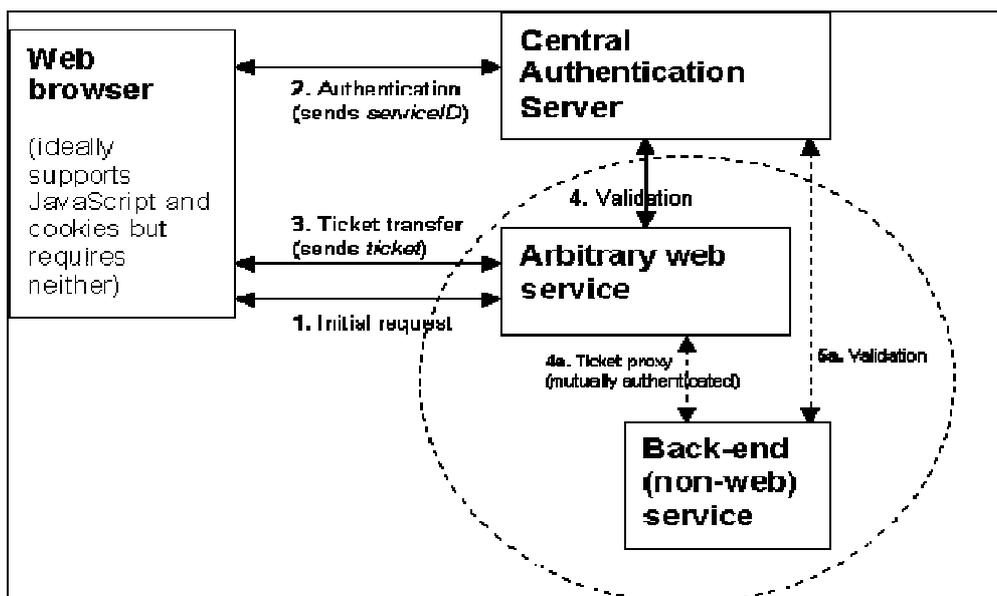


Figura.10 – Esquema Funcionamento CAS

Mas para que o CAS fizesse toda a gestão desde o login até às sessões, teria de se criar toda uma estrutura de ficheiros XML para que este se pudesse conectar às bases de dados do projecto, o que foi considerada uma solução inviável no tempo disponível para a conclusão do projecto.

Uma outra opção seria a utilização do LDAP⁷. Esta solução é muito utilizada ao nível das instituições de ensino superior, nomeadamente nos Estados Unidos, permitindo assim o uso central do login e da password para aceder a diversas aplicações Web de uma instituição.

Como o LDAP é um protocolo de rede que funciona sobre TCP/IP, permite organizar os recursos de rede de forma hierárquica, como uma árvore de directório, onde temos o directório raiz, em seguida a rede da empresa, o departamento e por fim o computador do funcionário e os recursos de rede (arquivos, impressoras, etc.) compartilhados por ele. A árvore de directório pode ser criada de acordo com as necessidades.

Ou seja o LDAP funcionaria como um repositório de dados onde seriam armazenados os dados necessários ao *login* de um utilizador, como por exemplo o seu username e a sua password.

Por norma o tráfego LDAP não é transmitido com segurança, daí se fazerem configurações sobre SSL, sendo neste ponto que se introduz o CAS, que foi referido acima, pois esta aplicação fornece-nos um serviço de autenticação seguro.

Existem diversas soluções baseadas no protocolo LDAP como é o Active Directory da que é compatível com LDAP e o OpenLDAP que é um projecto opensource.

Para se proceder à instalação do LDAP, é necessário proceder à configuração do protocolo, em que a primeira coisa a fazer é definir as configurações da base do directório onde se poderia optar por usar o modelo dcObjects (dc=dominio, dc=pt).

De seguida deve ser editado o ficheiro “/etc/ldap/slapd.conf” para adicionar toda a informação necessária desde o domínio, o tipo de BackEnd que o LDAP vai usar, quem é o super-administrador do directório bem como os índices para facilitar a pesquisa.

⁷ LDAP - Lightweight Directory Access Protocol

```

193.136.195.75 - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles

# Features to permit
#allow bind_v2

# Schema and objectClass definitions
include      /etc/ldap/schema/core.schema
include      /etc/ldap/schema/cosine.schema
include      /etc/ldap/schema/nis.schema
include      /etc/ldap/schema/inetorgperson.schema

# Where the pid file is put. The init.d script
# will not stop the server if you change this.
pidfile      /var/run/slapd/slapd.pid

# List of arguments that were passed to the server
argsfile     /var/run/slapd/slapd.args

# Read slapd.conf(5) for possible values
loglevel     0

# Where the dynamically loaded modules are stored
modulepath   /usr/lib/ldap
moduleload   back_bdb

# The maximum number of entries that is returned for a search operation
sizelimit    500

# The tool-threads parameter sets the actual amount of cpu's that is used
# for indexing.
tool-threads 1

#####
# Specific Backend Directives for bdb:
# Backend specific directives apply to this backend until another
# 'backend' directive occurs
backend      bdb
checkpoint  512 30

#####
# Specific Backend Directives for 'other':
# Backend specific directives apply to this backend until another

```

Connected to 193.136.195.75.

Figura.11 – Ficheiro /etc/ldap/slapd.conf.

De seguida edita-se o ficheiro ldap.conf onde se coloca o ip do HOST bem como os parâmetros para o teste na base de dados.

Após os passos anteriores é necessário inicializar o directório para de seguida se passar à criação da entrada principal deste, através do comando “slapadd”.

Com o LDAP configurado é necessário agora configurar ainda o modo de autenticação com o CAS.

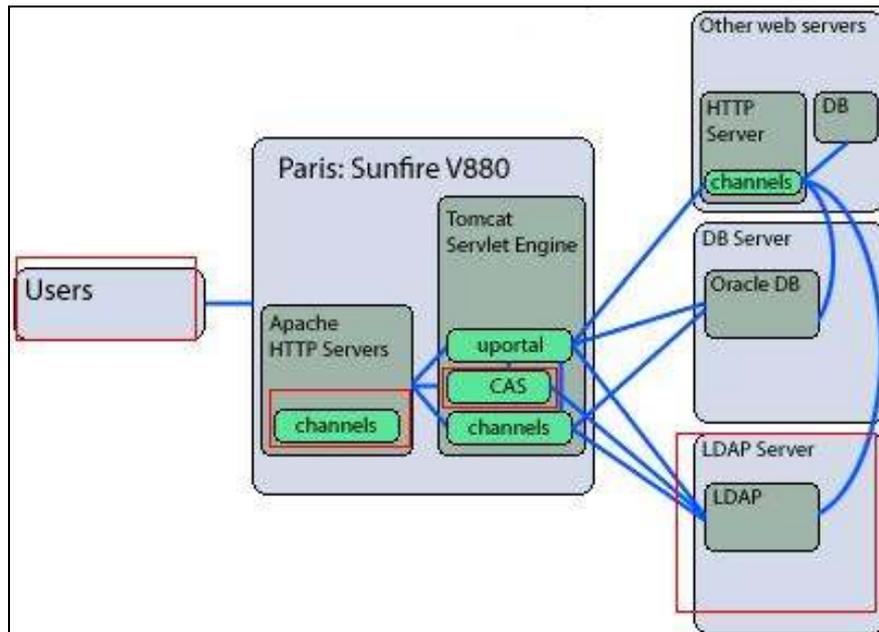


Figura.12 – Esquema de autenticação.

Para que a autenticação LDAP funcione temos de definir a sua configuração que assenta na especificação destes dois módulos:

- O método usado para aceder ao servidor LDAP;
- Atribuir um ou mais servidores LDAP;

Como os utilizadores vão ser armazenados por username e password, em que o método *fastbind* é modo de acesso directo.

O método *fastbind* pode ser usado juntamente com as directorias LDAP onde se pode conseguir os *Distinguished Names* dos utilizadores a partir do seu nome de login, ou seja neste modo o CAS tenta aceder às directorias do LDAP com o DN e a password fornecida no login.

Na segunda especificação temos apenas de indicar ao CAS qual o servidor LDAP que vai fazer a validação dos parâmetros que recebeu através do pedido de login.

Termina assim a configuração, qualquer utilizador registado necessita apenas de efectuar o login no Joomla podendo de seguida transitar de uma plataforma para outra sem que tenha de efectuar um novo login, uma vez que o CAS tem a variável de sessão armazenada e vai sempre fornecer esses dados à aplicação providenciando assim um login automático nas aplicações.

Capítulo 5

5 Conclusão

Em conclusão a este projecto, podemos primariamente dividi-lo em duas partes principais, a parte da elaboração do portal para o Agrupamentos de Escolas Paulo Quintela, e uma segunda parte de pesquisa sobre as várias formas de integração do portal com o E-Learning já existente no Agrupamento.

A primeira parte, que passou pela elaboração do portal, pensamos que foi atingida com sucesso uma vez que o portal foi desenvolvido e está já a ser utilizado.

Esta parte foi bastante produtiva pois além de munir o Agrupamento com uma excelente ferramenta OpenSource foi ainda tida em conta a parte de familiarização com a nova ferramenta de trabalho, através da formação dada aos professores do Agrupamento.

Esta nova ferramenta pensa-se que foi muito bem recebida pelo corpo docente uma vez que o feedback recebido foi bastante positivo, podendo-se então concluir que os objectivos propostos nesta primeira fase, que consequentemente é a mais importante, foram atingidos com sucesso.

Quanto à segunda parte do projecto, parte esta fundamentalmente de pesquisa e de estudo das diversas possibilidades de interligação do portal com o E-Learning, foram exploradas e implementas algumas possibilidades de integração do portal com a plataforma de E-Learning. Desta análise resultaram três níveis de interligação entre as plataformas, níveis que foram evoluindo uns a partir dos outros deixando assim várias opções de escolha.

O resultado final da pesquisa foram três níveis, níveis esses que não são distintos mas sim evoluções possíveis uns dos outros, logo à medida que se vai aumentando o nível está-se progressivamente a melhor a forma de comunicação entre as duas aplicações, bem como o grau de complexidade e, claro está, o grau de segurança.

Não podíamos deixar de fazer referência à excelente plataforma que é o Joomla!, bem como a toda a sua comunidade da qual fazemos também agora parte, pelo excelente software livre desenvolvido que permite que o utilizador o manipule e o adapte segundo as suas necessidades, liberdade esta que nos permitiu criar as componentes necessárias ao desenvolvimento deste projecto.

Gostaríamos ainda de salientar a possibilidade que nos foi dada de poder ajudar o Agrupamento do qual foi alvo o nosso trabalho.

Com este projecto pudemos aprofundar o nosso conhecimento em relação a tecnologias conhecidas bem como a outras desconhecidas por nós.

Bibliografia

- [MB 00] Monteiro, E., Boavida. F. Engenharia de Redes Informáticas. FCA – Editora de Informática, LDA – Agosto de 2000.
- [CLG07] Chopra, V. Li, S. Genender, J. Professional Apache TomCat 6. Wrox – Agosto de 2007.
- [URL1] <http://www.joomla.org/>
- [URL2] <http://forum.joomla.org/>
- [URL3] <http://www.joomlapt.com/>
- [URL4] <http://forum.joomlapt.com/>
- [URL5] <http://pt.wikipedia.org/wiki/Joomla>
- [URL6] <http://www.joomla.com.br/>

Apêndice A

A WampServer 2.0

O WampServer é um software publicado sob a licença GNU/GPL e desenvolvido pelo grupo PHP para ambientes Windows.

O lançamento da primeira versão (Wamp5 1.0.0) ocorreu no dia 13 de Julho de 2004, e neste momento a última versão é WampServer 2.0c (que inclui Apache 2.2.8, MySQL 5.0.51b e PHP 5.2.6).

Permite criar aplicações Web com Apache, PHP e MySQL. Para fazer a gestão da base de dados, o Wamp5 tem o phpMyAdmin e o SQLiteManager.

Na Figura.A1 pode-se ver o significado da sigla Wamp5.

Sigla	Significado
W	Windows - Sistema operativo da Microsoft
A	Apache - Servidor da Apache Software Foundation
M	MySQL - Banco de dados da MySQL AB
P5	PHP (Versão 5) - Linguagem de script da The PHP Group

Figura.A1 – Significado da sigla Wamp5.

A.1 Arquitectura

A arquitectura do Wamp5 baseia-se no modelo cliente-servidor. O utilizador requisita um ficheiro através de um endereço URL que se encontra alojado num computador remoto, por sua vez o servidor remoto verifica o pedido e devolve-lhe o pedido, através do envio do ficheiro ou de uma mensagem de erro.

A Figura.A2 representa o modelo cliente-servidor.

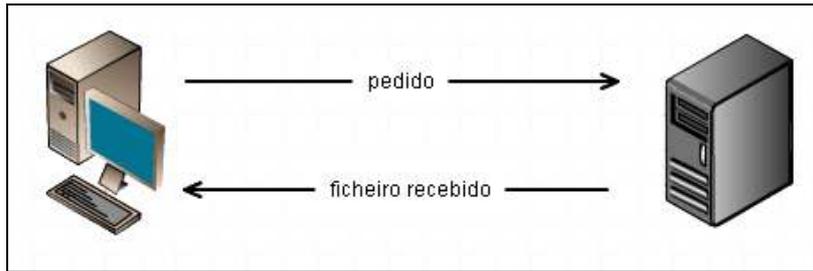


Figura.A2 – Modelo Cliente-Servidor.

A integração do PHP com o Apache http Server permite o processamento de pedidos de ficheiros “.php“, mas permite também o acesso a base de dados MySQL.

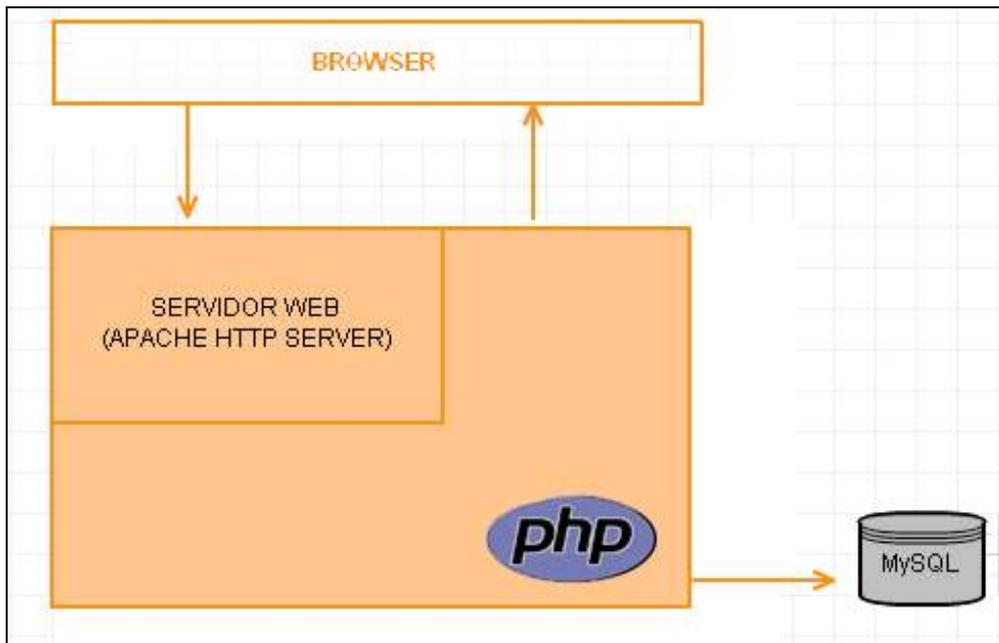


Figura.A2 – Integração do PHP com o Apache.

Anexos