

**Plataforma de Gestão de Conteúdos dos Cursos da
Escola de Comunicação, Artes e Tecnologia da
Informação**



Pedro Luís da Silva Correia n.º 20077537
Nuno Miguel Nóbrega e Silva Borges n.º 20076724



Universidade Lusófona de Humanidades e Tecnologias
Licenciatura em Engenharia Informática

Plataforma de Gestão de Conteúdos dos Cursos da Escola de Comunicação, Artes e Tecnologia da Informação

Pedro Luís da Silva Correia n.º 20077537
Nuno Miguel Nóbrega e Silva Borges n.º 20076724

Relatório submetido como requisito parcial
para obtenção do grau de licenciatura em
Engenharia Informática

Orientado por Pedro Malta



Universidade Lusófona de Humanidades e Tecnologias
Licenciatura em Engenharia Informática

Outubro de 2010

I. Resumo

O nosso trabalho consistiu em criar uma solução para optimizar a pesquisa de documentos existentes na ferramenta de *E-Learning* da Universidade Lusófona de Humanidades e Tecnologias.

A optimização consistiu em catalogar todos os documentos por palavras-chave, para uma melhor consulta por parte dos utilizadores, os alunos da Escola de Comunicação, Artes e Tecnologia da Informação (adiante designada por ECATI).

De forma a facilitar a gestão das palavras-chave foi implementado um *back office* para gestão das mesmas e da respectiva associação aos documentos.

O *back office* também permite gerir utilizadores e perfis que têm acesso à documentação, bem como gerar relatórios sobre a actividade dos docentes sobre os documentos.

Para o desenvolvimento deste projecto foram utilizadas um conjunto de ferramentas e tecnologias *Open Source*, as quais iremos abordar ao longo deste relatório.

Será apresentado o modelo de dados que foi criado para responder às necessidades do projecto e ainda o funcionamento geral da aplicação com alguns exemplos práticos.

Índice

| | | |
|-------|--|-----------|
| I. | Resumo..... | 1 |
| 1. | Introdução | 4 |
| 2. | Metodologia | 6 |
| 3. | Levantamento dos requisitos | 7 |
| 3.1 | <i>Características dos utilizadores</i> | 7 |
| 3.1.1 | <i>Utilizador Genérico.....</i> | 8 |
| 3.1.2 | <i>Administrador do sistema.....</i> | 8 |
| 3.1.3 | <i>Docente.....</i> | 8 |
| 3.1.4 | <i>Aluno</i> | 9 |
| 3.2 | <i>Especificação de requisitos funcionais</i> | 9 |
| 3.2.1 | <i>Gestão de acessos à aplicação</i> | 9 |
| 3.2.2 | <i>Gestão de utilizadores da aplicação.....</i> | 11 |
| 3.2.3 | <i>Gestão de conteúdos.....</i> | 13 |
| 3.2.4 | <i>Gestão de documentos</i> | 19 |
| 3.2.5 | <i>Obter relatórios.....</i> | 21 |
| 3.2.6 | <i>Pesquisa de documentos.....</i> | 22 |
| 3.3 | <i>Especificação de requisitos não funcionais.....</i> | 22 |
| 3.3.1 | <i>Requisitos de software</i> | 23 |
| 4. | Selecção de tecnologias | 24 |
| 4.1 | <i>MySQL</i> | 25 |
| 4.2 | <i>Apache</i> | 26 |
| 4.3 | <i>Python</i> | 26 |
| 4.4 | <i>Django.....</i> | 26 |
| 4.5 | <i>JQuery.....</i> | 27 |
| 4.6 | <i>Model-View-Controller.....</i> | 28 |
| 4.7 | <i>Asynchronous Javascript and XML</i> | 29 |
| 5. | Desenho do modelo de dados | 31 |
| 5.1 | <i>Relações importantes</i> | 34 |
| 6. | Funcionalidades principais | 36 |
| 6.1 | <i>Pesquisar documentos</i> | 36 |
| 6.2 | <i>Catalogar documentos por palavra-chave.....</i> | 36 |
| 7. | Próximos passos | 38 |
| 8. | Conclusão | 39 |
| 9. | Referências | 40 |
| 10. | Anexos | 41 |
| | <i>Anexo I – Manual de instalação.....</i> | 41 |
| | <i>Anexo II – Quick Guide</i> | 45 |

Índice de figuras

| | |
|---|----|
| Figura 1 – Metodologia utilizada | 6 |
| Figura 2 – Hierarquia de utilizadores | 7 |
| Figura 3 – Use Case Gestão de acessos à aplicação..... | 9 |
| Figura 4 – Use Case Gestão de utilizadores da aplicação | 11 |
| Figura 5 – Use Case Gestão de conteúdos | 13 |
| Figura 6 – Use Case Gestão de documentos | 19 |
| Figura 7 – Use Case Obter relatórios | 21 |
| Figura 8 – Use Case Pesquisa de documentos | 22 |
| Figura 9 – Modelo MVC | 28 |
| Figura 10 – Funcionamento AJAX..... | 29 |
| Figura 11 – Normalização de tabelas | 32 |
| Figura 12 – Modelo de dados..... | 33 |
| Figura 13 – Relação de associação das palavras-chave aos documentos | 34 |
| Figura 14 – Relação de associação de documentos às disciplinas | 34 |
| Figura 15 – Relação de associação de disciplinas aos utilizadores..... | 35 |
| Figura 16 – Pesquisa de documentos por palavra-chave..... | 36 |
| Figura 17 – Catalogar documentos por palavra-chave | 37 |

1. Introdução

Este projecto foi realizado no âmbito da disciplina Projecto Final de Curso e surge da necessidade de melhorar o funcionamento da ferramenta de *E-Learning* existente na Universidade Lusófona de Humanidades e Tecnologias (ULHT), onde são depositados documentos referentes às disciplinas dos cursos da ECATI.

Para a realização do nosso projecto foi-nos proposto a implementação de uma aplicação de gestão e consulta de documentos com matérias das disciplinas que pertencem ao curso de Engenharia de Informática, que permitisse:

- Catalogar todos os programas das disciplinas em termos de palavras-chave que caracterizem o curso:
 - Objecto cerne da disciplina;
 - Áreas subjacentes; e
 - Bibliografia.
- Permitir criar, consultar, alterar ou eliminar os documentos;
- Gerir perfis de utilizadores (docentes e alunos); e
- Obter relatórios por docente e ano/semestre/disciplina dos conteúdos.

Tendo em conta o que nos foi proposto, verificámos que seria necessário garantir que não existiriam perdas de versões ou sobreposição de documentos quando dois ou mais docentes se encontrarem a alterar o mesmo documento. Desta forma, decidimos que quando um documento se encontrava a ser alterado, ficaria indisponível para outro docente o alterar.

Esta funcionalidade não tem como objectivo o de criar versões de todas as alterações, mas sim garantir apenas que quando um documento se encontra a ser alterado por um determinado utilizador, apenas esse possa efectuar as alterações.

No decorrer do nosso projecto, decidimos manter a plataforma o mais genérica possível, não limitando a pesquisa de documento apenas às disciplinas do curso de Engenharia Informática. Desta forma, a nossa aplicação permite a pesquisa de documentos referentes às disciplinas de todos os cursos da ECATI.

Para o desenvolvimento da nossa aplicação, optámos por utilizar um conjunto de ferramentas e tecnologias *Open Source* que se encontram bem referenciadas e contam com um número elevado de implementações no mercado.

Escola de Comunicação, Artes e Tecnologias da Informação

O presente documento descreve o trabalho desenvolvido durante a implementação da plataforma de gestão de conteúdos, do seguinte modo:

- Capítulo 2 – Metodologia;
- Capítulo 3 – Levantamento dos Requisitos;
- Capítulo 4 – Selecção de Tecnologias;
- Capítulo 5 – Desenho do modelo de dados;
- Capítulo 6 – Funcionalidades principais;
- Capítulo 7 – Próximos passos;
- Capítulo 8 – Conclusão;
- Capítulo 9 – Referências; e
- Capítulo 10 – Anexos.

2. Metodologia

De modo a atingir os objectivos anteriormente descritos, foi utilizada a seguinte abordagem na realização do projecto:



Figura 1 – Metodologia utilizada

Seguidamente, descrevemos as actividades realizadas em cada uma das fases do projecto:

1. Definição: Levantamento de todos os requisitos da aplicação a desenvolver;
2. Escolha das tecnologias: Selecção das tecnologias a utilizar na implementação da nossa aplicação;
3. Desenho do modelo de dados: Definição do modelo de dados a implementar; e
4. Desenvolvimento: Implementação da aplicação, abrangendo todos os requisitos, e testes à aplicação.

3. Levantamento dos requisitos

Neste capítulo iremos apresentar os requisitos funcionais e não funcionais da aplicação. Estes serão descritos de uma forma informal, recorrendo a exemplos práticos sempre que se revele a sua necessidade.

3.1 Características dos utilizadores

Os potenciais utilizadores da aplicação foram identificados na tarefa de especificação de resultados do sistema, e agrupados de acordo com as suas características, ou perfil. Cada utilizador da aplicação, ao qual poderá estar associado a privilégios de utilização.

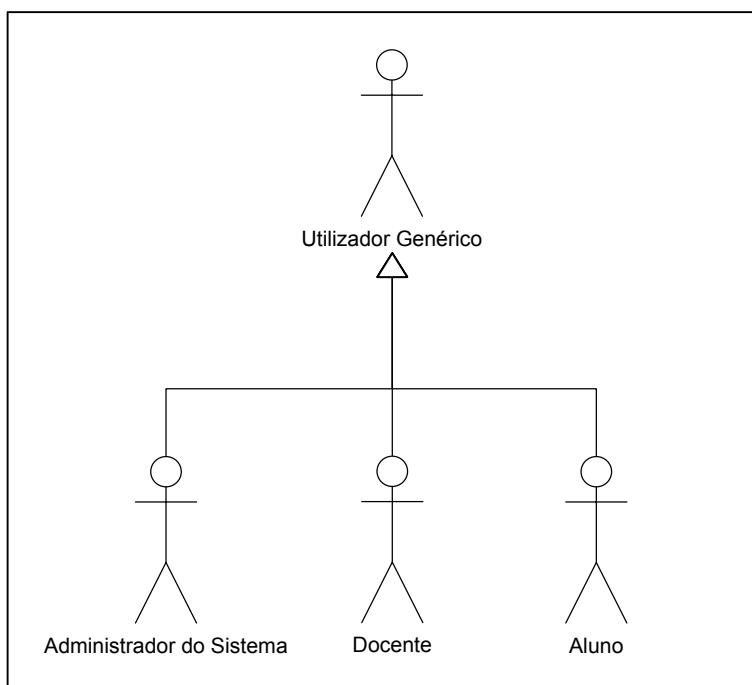


Figura 2 – Hierarquia de utilizadores

Utilizador genérico – Qualquer utilizador que tenha permissão para aceder à aplicação.

Administrador do sistema – Responsável pela manutenção dos utilizadores e gestão de acessos à aplicação.

Docente – Tem a responsabilidade de criar, alterar e eliminar documentos, bem como a gestão das palavras-chave que permitem a pesquisa dos mesmos. Este utilizador também pode obter relatórios dos conteúdos por docente e ano/semestre/disciplina.

Aluno – Tem apenas acesso a efectuar pesquisas dos documentos disponíveis na aplicação.

3.1.1 Utilizador Genérico

O utilizador genérico representa a entidade que tem funcionalidades comuns a qualquer utilizador da aplicação. Possui funcionalidades como:

- **Autorizar e autenticar**

Permite fazer login na aplicação. Esta funcionalidade é muito importante, já que é com ela que a segurança da aplicação é garantida, através da autenticação e autorização de utilizadores.

3.1.2 Administrador do sistema

O administrador de sistema terá a seu cargo as seguintes funções / funcionalidades:

- **Gerir utilizadores**

Criar, alterar e apagar utilizadores da aplicação.

- **Gerir perfis**

Criar, alterar e apagar perfis da aplicação.

- **Associar perfis a utilizadores**

Associar permissões a utilizadores.

3.1.3 Docente

O docente tem as seguintes funcionalidades ao seu dispor:

- **Gerir conteúdos**

Catalogar todos os programas das disciplinas em termos de palavras-chave que caracterizem os cursos da ECATI.

- **Gerir documentos**

Criar, alterar e apagar os documentos disponíveis na aplicação.

- **Obter relatórios**

Obter relatórios dos conteúdos por docente e ano/semestre/disciplina.

3.1.4 Aluno

O aluno tem as seguintes funcionalidades ao seu dispor:

- **Pesquisa de documentos**

Pesquisa de documentos através de palavras-chave disponibilizadas ao utilizador num modo de consulta dinâmico, onde o utilizador vai escrevendo sobre o que quer pesquisar, e a aplicação vai sugerindo as diversas palavras-chave que encontram disponíveis e os documentos correspondentes.

3.2 Especificação de requisitos funcionais

O presente capítulo consiste na descrição dos serviços que a aplicação deve oferecer, como deve reagir a certas entradas e como deve comportar-se em determinadas situações. Foi elaborado um conjunto de *use cases*, com o intuito de possibilitar uma melhor compreensão das funcionalidades da aplicação e da interacção dos diferentes utilizadores com a mesma.

3.2.1 Gestão de acessos à aplicação

A aplicação deve disponibilizar um conjunto de funcionalidades que permitam validar e autenticar o “utilizador genérico”, para que desta forma se possa disponibilizar as funcionalidades correspondentes ou impedir o seu acesso. O diagrama de *use cases* mostra o conjunto de funcionalidades da aplicação e a sua interacção com os diferentes utilizadores.

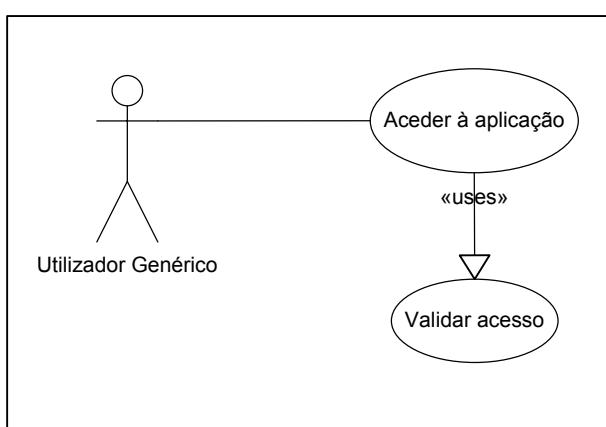


Figura 3 – Use Case Gestão de acessos à aplicação

Use case #1: Aceder à aplicação

Enquadramento: O utilizador genérico acede à aplicação via *web* e de acordo com o seu registo são disponibilizadas as funcionalidades.

Pré-condição:

1. O utilizador tem que estar ligado à *internet*.

Fluxo de eventos:

1. O utilizador insere o endereço da aplicação no seu *browser*.
 - a. **Uses** Validar acesso.
2. Inserção na aplicação da informação necessária para disponibilização das funcionalidades do utilizador que validou acesso; e
3. O utilizador tem acesso à suas funcionalidades via *web* com os dados e opções necessárias para executar as suas tarefas.

Pós-condição:

1. Disponibilização das funcionalidades do respectivo utilizador; e
2. Negação de acesso de entrada na aplicação.

Use case #2: Validar acesso

Enquadramento: Para garantir a segurança da aplicação e disponibilizar as funcionalidades correspondentes, cada utilizador necessita de validação para ter acesso à aplicação.

Pré-condição: N/A

Fluxo de eventos:

1. É mostrado ao utilizador um “ecrã de validação de acesso”, com os campos de *login* e *password* para serem preenchidos;
2. O utilizador introduz o seu *username* e a *password*; e
3. A aplicação verifica se existe algum utilizador com o *username* e *password* inseridos na base de dados da aplicação.

Pós-condição:

1. Validação do acesso e permissão de acesso às funcionalidades correspondentes; e
2. Acesso não validado.

3.2.2 Gestão de utilizadores da aplicação

A aplicação deve disponibilizar um conjunto de funcionalidades que permitam gerir os seus utilizadores. O seguinte diagrama de *use cases* demonstra o conjunto de funcionalidades da aplicação que possibilita a gestão de utilizadores:

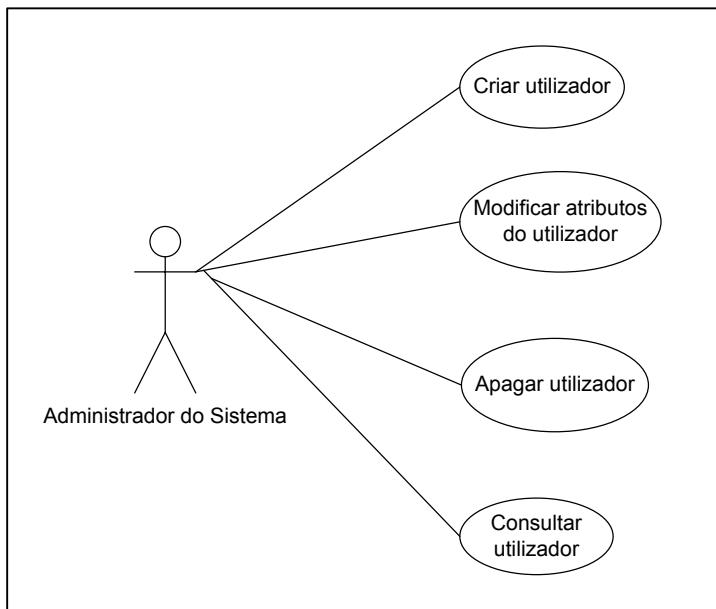


Figura 4 – Use Case Gestão de utilizadores da aplicação

Use case #3: Criar utilizador

Enquadramento: O administrador acede ao *back office* e selecciona a opção de criação de um novo registo de utilizador.

Pré-condição: Utilizador tem que estar validado na aplicação, ter o perfil de administrador da aplicação.

Fluxo de eventos:

- A aplicação mostra o ecrã para “adicionar utilizador”;
- O administrador introduz os dados requeridos no ecrã e selecciona a opção “Gravar”;
- A aplicação verifica se os dados foram inseridos correctamente; e
- Os dados do novo utilizador são guardados na base de dados da aplicação.

Pós-condição:

- Criação de utilizador com sucesso; e
- Falha na criação do novo utilizador.

Use case #4: Modificar atributos do utilizador

Enquadramento: O administrador da aplicação acede ao *back office* e lista os utilizadores existentes.

Pré-condição: Utilizador tem que estar validado na aplicação, ter o perfil de administrador da aplicação.

Fluxo de eventos:

1. A aplicação mostra a lista dos utilizadores existentes na aplicação;
2. O administrador selecciona o utilizador que pretende modificar os dados;
3. A aplicação mostra o ecrã de edição de utilizadores; e
4. O administrador introduz os dados requeridos no ecrã e selecciona a opção “Gravar”. Os novos atributos do utilizador são guardados na base de dados da aplicação.

Pós-condição:

1. Atributos do utilizador modificados com sucesso; e
2. Falha no registo das modificações.

Use case #5: Apagar utilizador

Enquadramento: O administrador acede ao *back office* e lista os utilizadores existentes na aplicação.

Pré-condição: Utilizador tem que estar validado na aplicação, ter o perfil de administrador da aplicação.

Fluxo de eventos:

1. A aplicação lista os utilizadores existentes;
2. O administrador da aplicação selecciona o utilizador que pretende apagar e escolhe a opção “Remover utilizadores seleccionados”;
3. A aplicação apresenta uma janela com o pedido de confirmação; e
4. O utilizador é removido da base de dados da aplicação.

Pós-condição:

1. Utilizador eliminado da aplicação com sucesso; e
2. Ocorrência de anomalia(s) na eliminação do utilizador.

Use case #6: Consultar utilizador

Enquadramento: O administrador da aplicação acede ao *back office* e selecciona a opção de consultar utilizador.

Pré-condição: Utilizador tem que estar validado na aplicação, ter o perfil de administrador da aplicação.

Fluxo de eventos:

1. A aplicação lista os utilizadores existentes; e
2. O administrador da aplicação selecciona a linha do utilizador que pretende consultar.

Pós-condição:

1. Consulta de utilizador efectuada com sucesso; e
2. Ocorrência de anomalias na consulta de utilizador.

3.2.3 Gestão de conteúdos

A aplicação deve disponibilizar um conjunto de funcionalidades que permitam gerir os conteúdos. O seguinte diagrama de *use cases* demonstra o conjunto de funcionalidades da aplicação que possibilita a gestão de conteúdos:

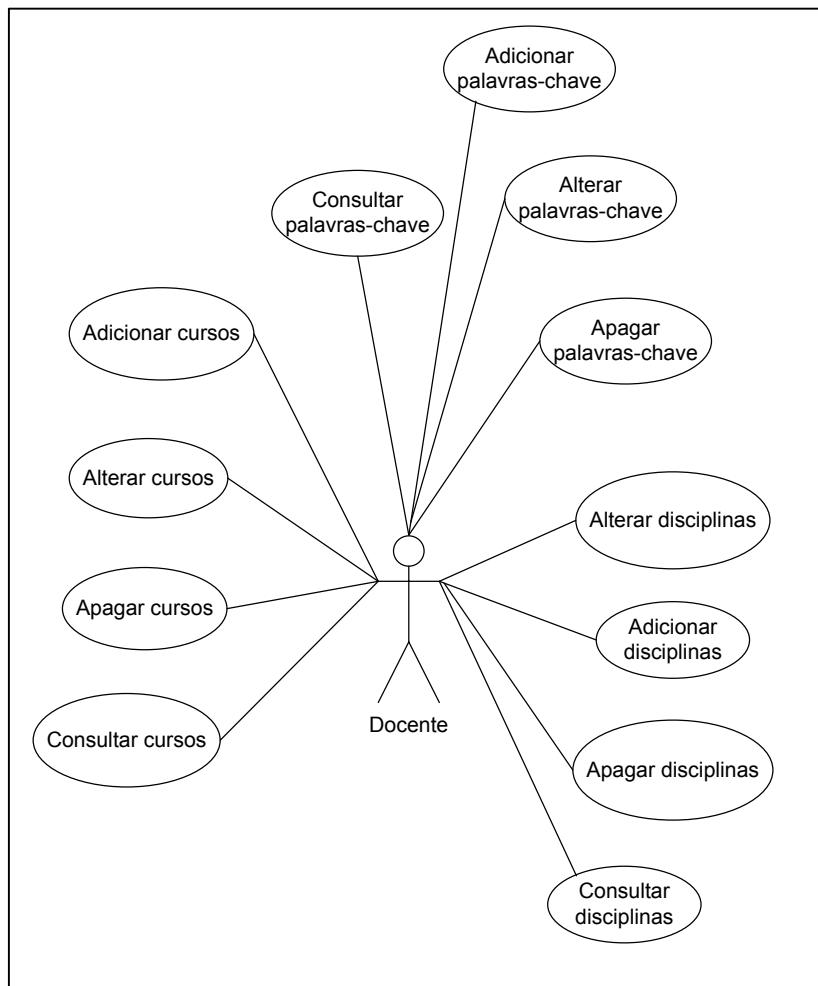


Figura 5 – Use Case Gestão de conteúdos

Use case #7: Adicionar palavras-chave

Enquadramento: O docente acede ao *back office* e selecciona a opção de adicionar uma nova palavra-chave.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação mostra o ecrã para “adicionar palavra-chave”;
2. O docente introduz os dados requeridos no ecrã e selecciona a opção “Gravar”;
3. A aplicação verifica se os dados foram inseridos correctamente; e
4. Os dados da nova palavra-chave são guardados na base de dados da aplicação.

Pós-condição:

1. Criação de palavra-chave com sucesso; e
2. Falha na criação da nova palavra-chave.

Use case #8: Alterar palavra-chave

Enquadramento: O docente acede ao *back office* e lista as palavras-chave existentes.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação mostra a lista de palavras-chave existentes na aplicação;
2. O docente selecciona a palavra-chave que pretende modificar;
3. A aplicação mostra o ecrã de edição de palavras-chave; e
4. O docente introduz os dados requeridos no ecrã e selecciona a opção “Gravar”. As alterações às palavras-chave são guardadas na base de dados da aplicação.

Pós-condição:

1. Palavras-chave modificadas com sucesso; e
2. Falha no registo das modificações.

Use case #9: Apagar palavra-chave

Enquadramento: O docente acede ao *back office* e lista as palavras-chave existentes.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação lista as palavras-chave existentes;
2. O docente selecciona a palavra-chave que pretende apagar e escolhe a opção “Remover chaves seleccionados”;
3. A aplicação apresenta uma janela com o pedido de confirmação; e
4. A palavra-chave é removida da base de dados da aplicação.

Pós-condição:

1. Palavra-chave eliminada da aplicação com sucesso; e
2. Ocorrência de anomalia(s) na eliminação da palavra-chave.

Use case #10: Consultar palavra-chave

Enquadramento: O docente acede ao *back office* e selecciona a opção de consultar palavras-chaves.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação lista as palavras-chave existentes; e
2. O docente selecciona a linha da palavra-chave que pretende consultar.

Pós-condição:

1. Consulta da palavra-chave efectuada com sucesso; e
2. Ocorrência de anomalias na consulta da palavra-chave.

Use case #11: Adicionar disciplinas

Enquadramento: O docente acede ao *back office* e selecciona a opção de adicionar uma nova disciplina.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação mostra o ecrã para “adicionar disciplinas”;
2. O docente introduz os dados requeridos no ecrã e selecciona a opção “Gravar”;
3. A aplicação verifica se os dados foram inseridos correctamente; e
4. Os dados da nova disciplina são guardados na base de dados da aplicação.

Pós-condição:

1. Criação de disciplinas com sucesso; e
2. Falha na criação da nova disciplina.

Use case #12: Alterar disciplinas

Enquadramento: O docente acede ao *back office* e lista as disciplinas existentes.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação mostra a lista de disciplinas existentes na aplicação;
2. O docente selecciona a disciplina que pretende modificar;
3. A aplicação mostra o ecrã de edição de disciplinas; e
4. O docente introduz os dados requeridos no ecrã e selecciona a opção “Gravar”. As alterações às disciplinas são guardadas na base de dados da aplicação.

Pós-condição:

1. Disciplinas modificadas com sucesso; e
2. Falha no registo das modificações.

Use case #13: Apagar disciplinas

Enquadramento: O docente acede ao *back office* e lista as disciplinas existentes.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação lista as disciplinas existentes;
2. O docente selecciona a disciplina que pretende apagar e escolhe a opção “Remover disciplinas seleccionados”;
3. A aplicação apresenta uma janela com o pedido de confirmação; e
4. A disciplina é removida da base de dados da aplicação.

Pós-condição:

1. Disciplina eliminada da aplicação com sucesso; e
2. Ocorrência de anomalia(s) na eliminação da disciplina.

Use case #14: Consultar disciplina

Enquadramento: O docente acede ao *back office* e selecciona a opção de consultar disciplinas.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação lista as disciplinas existentes; e
2. O docente selecciona a linha da disciplina que pretende consultar.

Pós-condição:

1. Consulta da disciplina efectuada com sucesso; e
2. Ocorrência de anomalias na consulta da disciplina.

Use case #15: Adicionar cursos

Enquadramento: O docente acede ao *back office* e selecciona a opção de adicionar um novo curso.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação mostra o ecrã para “adicionar cursos”;
2. O docente introduz os dados requeridos no ecrã e selecciona a opção “Gravar”;
3. A aplicação verifica se os dados foram inseridos correctamente; e
4. Os dados do novo curso são guardados na base de dados da aplicação.

Pós-condição:

1. Criação de cursos com sucesso; e
2. Falha na criação do novo curso.

Use case #16: Alterar cursos

Enquadramento: O docente acede ao *back office* e lista os cursos existentes.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação mostra a lista de cursos existentes na aplicação;
2. O docente selecciona o curso que pretende modificar;
3. A aplicação mostra o ecrã de edição de cursos; e
4. O docente introduz os dados requeridos no ecrã e selecciona a opção “Gravar”. As alterações aos cursos são guardadas na base de dados da aplicação.

Pós-condição:

1. Cursos modificados com sucesso; e
2. Falha no registo das modificações.

Use case #17: Apagar cursos

Enquadramento: O docente acede ao *back office* e lista os cursos existentes.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação lista os cursos existentes;
2. O docente selecciona o curso que pretende apagar e escolhe a opção “Remover cursos seleccionados”;
3. A aplicação apresenta uma janela com o pedido de confirmação; e
4. O curso é removido da base de dados da aplicação.

Pós-condição:

1. Curso eliminado da aplicação com sucesso; e
2. Ocorrência de anomalia(s) na eliminação do curso.

Use case #18: Consultar curso

Enquadramento: O docente acede ao *back office* e selecciona a opção de consultar cursos.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação lista os cursos existentes; e
2. O docente selecciona a linha do curso que pretende consultar.

Pós-condição:

1. Consulta do curso efectuada com sucesso; e
2. Ocorrência de anomalias na consulta do curso.

3.2.4 Gestão de documentos

A aplicação deve disponibilizar um conjunto de funcionalidades que permitam gerir os documentos a disponibilizar. O seguinte diagrama de *use cases* demonstra o conjunto de funcionalidades da aplicação que possibilita a gestão de documentos:

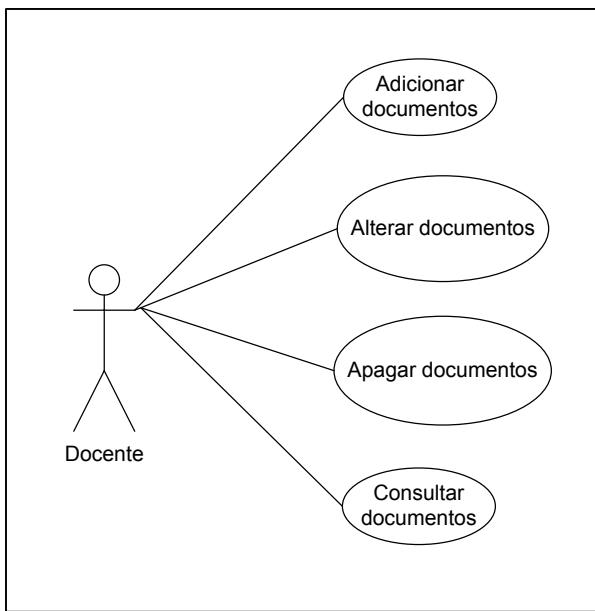


Figura 6 – Use Case Gestão de documentos

Use case #19: Adicionar documentos

Enquadramento: O docente acede ao *back office* e selecciona a opção de adicionar um novo ficheiro.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação mostra o ecrã para “adicionar ficheiros”;
2. O docente introduz os dados requeridos no ecrã e selecciona a opção “Gravar”;
3. A aplicação verifica se os dados foram inseridos correctamente; e
4. Os dados do novo ficheiro são guardados na base de dados da aplicação.

Pós-condição:

1. Criação de ficheiros com sucesso; e
2. Falha na criação do novo ficheiro.

Use case #20: Alterar documentos

Enquadramento: O docente acede ao *back office* e lista os ficheiros existentes.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação mostra a lista de cursos existentes na aplicação;
2. O docente selecciona o ficheiro que pretende modificar;
3. A aplicação mostra o ecrã de edição de ficheiros; e
4. O docente introduz os dados requeridos no ecrã e selecciona a opção “Gravar”. As alterações aos ficheiros são guardadas na base de dados da aplicação.

Pós-condição:

1. Ficheiros modificados com sucesso; e
2. Falha no registo das modificações.

Use case #21: Apagar documentos

Enquadramento: O docente acede ao *back office* e lista os ficheiros existentes.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação lista os ficheiros existentes;
2. O docente selecciona o curso que pretende apagar e escolhe a opção “Remover ficheiros seleccionados”;
3. A aplicação apresenta uma janela com o pedido de confirmação; e
4. O ficheiro é removido da base de dados da aplicação.

Pós-condição:

1. Ficheiro eliminado da aplicação com sucesso; e
2. Ocorrência de anomalia(s) na eliminação do ficheiro.

Use case #22: Consultar documentos

Enquadramento: O docente acede ao *back office* e selecciona a opção de consultar ficheiros.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação lista os ficheiros existentes; e
2. O docente selecciona a linha do ficheiro que pretende consultar.

Pós-condição:

1. Consulta do ficheiro efectuada com sucesso; e
2. Ocorrência de anomalias na consulta do ficheiro.

3.2.5 Obter relatórios

A aplicação deve disponibilizar um conjunto de funcionalidades que permitam obter relatórios dos conteúdos por docente e ano/semestre/disciplina. O seguinte diagrama de *use cases* demonstra o conjunto de funcionalidades da aplicação que possibilita a obtenção dos relatórios:

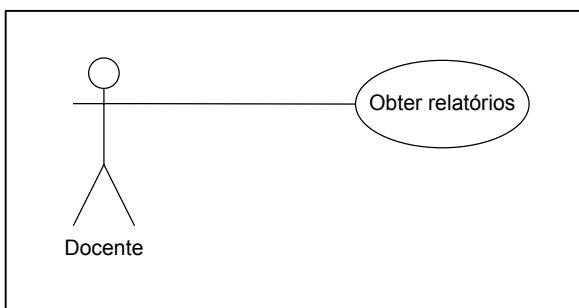


Figura 7 – Use Case Obter relatórios

Use case #23: Obter relatórios

Enquadramento: O docente acede ao *back office* e selecciona a opção de adicionar relatórios.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de docente da ECATI.

Fluxo de eventos:

1. A aplicação mostra o ecrã para “Gerar relatórios”;
2. O docente introduz os dados requeridos no ecrã e selecciona a opção “Gerar”; e
3. A aplicação gera o relatório e apresenta-o no ecrã.

Pós-condição:

1. Obtenção do relatório efectuada com sucesso; e
2. Ocorrência de anomalias na geração do relatório.

3.2.6 Pesquisa de documentos

A aplicação deve disponibilizar um conjunto de funcionalidades que permitam pesquisar os documentos através de palavras-chave disponibilizadas ao utilizador num modo de consulta dinâmico. O seguinte diagrama de *use cases* demonstra o conjunto de funcionalidades da aplicação que possibilita a pesquisa dos documentos:

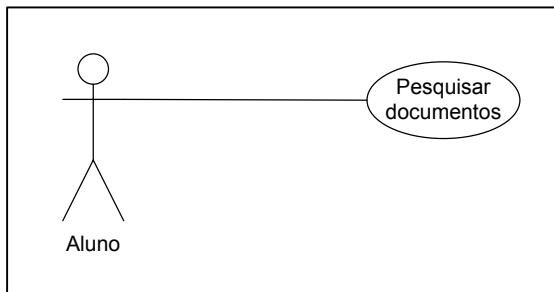


Figura 8 – Use Case Pesquisa de documentos

Use case #24: Pesquisar documentos

Enquadramento: O aluno acede à aplicação e selecciona a opção de pesquisa.

Pré-condição: Utilizador tem que estar validado na aplicação e ter o perfil de aluno da ECATI.

Fluxo de eventos:

1. O aluno insere a palavra-chave e selecciona a opção “Pesquisar”;
2. A aplicação lista os documentos existentes associados à palavra-chave inserida; e
3. O aluno selecciona a linha do documento que pretende consultar.

Pós-condição:

1. Consulta do documento efectuada com sucesso; e
2. Ocorrência de anomalias na consulta do documento.

3.3 Especificação de requisitos não funcionais

Alguns dos atributos de uma aplicação não podem ser definidos relativamente à sua funcionalidade. Apesar de ser difícil definir e medir objectivamente requisitos não funcionais, é útil identificá-los, para que possam ser considerados, pelo menos a um nível genérico, uma vez que também deles depende, em larga medida, a eficácia da aplicação.

3.3.1 Requisitos de software

A tabela seguinte possui uma estimativa para os requisitos de *software* necessários para efectuar um processamento optimizado da nossa aplicação.

| Nome do produto | Detalhe |
|-----------------|---|
| MySQL | Sistema de Gestão de Base de Dados |
| Apache | Servidor http |
| Django | <i>Framework Server Side</i> |
| Python | Linguagem de programação <i>Server Side</i> |
| JQuery | <i>Framework Client Side</i> |

4. Selecção de tecnologias

Este capítulo tem como objectivo explicar o porquê da escolha das tecnologias que utilizamos para a implementação da nossa aplicação.

A nossa aplicação permite aos alunos da ECATI pesquisar documentos através de um *browser*, pelo que se trata de uma aplicação *Web*.

Pretendemos escolher a que melhor se adequa ao nosso projecto para que à data estipulada para a entrega a nossa aplicação esteja disponível com a qualidade esperada.

Neste sentido, surgiu a questão de qual plataforma/linguagem de programação a utilizar para o seu desenvolvimento.

Para implementação da nossa aplicação decidimos utilizar a *framework* Django, uma vez que a utilização de *frameworks* tem um elevado número de vantagens, das quais são enumeradas algumas aqui:

- **Redução do tempo de desenvolvimento** – Após conhecer a *framework* permite aos programadores terminar os projectos mais rapidamente;
- **Níveis de abstracção** – Possibilita o acesso a uma base de dados, acesso a sistema de ficheiros e simplifica a aplicação de técnicas relacionadas com a web;
- **Código testado** – A comunidade envolvida na *framework* permite que o código seja testado em aplicações reais o que possibilita a diminuição de erros e falhas de segurança;
- **Estrutura organizada** – As *frameworks* estipulam uma estrutura ao nível da organização dos ficheiros que constituem a aplicação, o que permite aos programadores reduzirem o tempo a pensar onde vão guardar a informação;
- **Open Source** – A maioria das *frameworks* são distribuídas de uma forma *open source*, o que significa que não têm custos directos de utilização;
- **Extensibilidade** – Por norma o código da *framework* é extensível, ou seja, pode de forma relativamente rápida servir propósitos para os quais não foi inicialmente pensado;
- **Técnicas mais comuns incluídas** – As *web frameworks* vêm por norma com algumas técnicas mais comuns e/ou emergentes incluídas no seu core, como é o exemplo do AJAX cuja implementação e utilização é facilitada pela maioria das *frameworks*;
- **“User base”** – As *frameworks* têm uma grande quantidade de seguidores que criam comunidades activas à volta das mesmas, o que permite a discussão e decisão do percurso que estas tomem: novas funcionalidades, correcção de erros, etc.

Para o desenvolvimento da nossa aplicação utilizámos as seguintes tecnologias:

- **Sistema de Gestão de Base de Dados (SGBD):** MySQL;
- **Servidor HTTP:** Apache;
- **Linguagens Server Side:** Python, com utilização da *framework* Django;
- **Linguagens Client Side:** HTML e JavaScript, com utilização da *framework* JQuery;
- **Arquitectura de software:** Model-View-Controller (MVC); e
- **Modelo:** Asynchronous Javascript and XML (AJAX).

4.1 MySQL



O MySQL utiliza linguagem SQL como interface, é actualmente uma das bases de dados mais populares e que conta com mais de 10 milhões de instalações pelo mundo todo e é *Open Source*.

A base de dados MySQL tornou-se a mais popular base de dados *Open Source* do mundo, devido ao seu alto desempenho consistente, alta confiabilidade e facilidade de uso.

É utilizada pela maioria dos programadores que desenvolvem aplicações *Web*, bem como por muitas das maiores organizações mundiais nomeadamente o *Yahoo*, *Alcatel-Lucent*, *Google*, *Youtube* e entre outras.

O MySQL tem as seguintes características:

- Portabilidade – Suporta praticamente qualquer plataforma actual;
- Compatibilidade – Existem *drivers* ODBC, JDBC e .NET, módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, C#, Visual Basic, Python, Perl, PHP, ASP e Ruby;
- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de *hardware*;
- Facilidade de utilização;
- Software livre com base na GPL;
- Contempla a utilização de vários *Storage Engines*, como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV Solid;
- Suporta controlo transaccional;
- Suporta Triggers;
- Suporta Cursors (*Non-Scrollable* e *Non-Updatable*);
- Suporta Stored Procedures e Functions; e
- Replicação facilmente configurável.

4.2 Apache



O servidor HTTP Apache é o mais bem-sucedido servidor web *open source* e representa quase 75% dos servidores activos no mundo. O Apache é responsável por mais de uma dezena de projectos envolvendo tecnologias de transmissão via web, processamento de dados e execução de aplicativos.

O servidor Apache é compatível com o protocolo HTTP versão 1.1 e as suas funcionalidades são mantidas através de uma estrutura de módulos, permitindo inclusive que o utilizador escreva os seus próprios módulos recorrendo à API do software.

4.3 Python



O Python é uma linguagem de altíssimo nível (VHLL - *Very High Level Language*), de sintaxe moderna, orientada a objectos, interpretada via *bytecode*, com tipo de dados forte (não há conversões automáticas) e dinâmica (não há declaração de variáveis e elas podem conter diferentes objectos), modular e multi-plataforma, de fácil aprendizagem e de implementação livre.

O Python é um a linguagem simples e por isso de fácil aprendizagem, permite escrever programas com um bom grau de complexidade sem nunca colocar em risco a produtividade de quem esta a desenvolver. O Python apresenta por isso uma curva de aprendizagem interessante mesmo para quem nunca teve contacto com outras linguagens de programação.

A linguagem Python é utilizada tipicamente como linguagem de *scripting* para administração de sistemas e apresenta facilidade de integração com outras linguagens como por exemplo o C.

4.4 Django



O Django é uma *framework* de desenvolvimento rápido para web que usa como linguagem base o Python e que utiliza o padrão Model-View-Controller (MVC).

É um projecto *open source* e que utiliza o princípio *Don't Repeat Yourself* (DRY), onde faz com que o programador aproveite ao máximo o código já desenvolvido, evitando a repetição.

O Django intitule-se como “***The Web framework for perfectionists with deadlines***”, ou seja, permite-nos criar rapidamente aplicações web de alto desempenho e elegância sem nunca comprometer os prazos exigentes que nos são impostos.

É também utilizado a ferramenta Django-squeeze, que tem como objectivo reduzir o tamanho de ficheiro CSS, Javascript e HTML de modo a tornar os ficheiros que se encontram num ambiente de produção mais eficazes e reduzidos.

O tamanho reduzido irá representar menor tempo de *download* dos ficheiros necessários e menor tempo de interpretação dos mesmos. O Django-squeeze remove todos os espaços desnecessários e *tabs* que se encontram nos ficheiros.

4.5 JQuery



na lógica dos sistemas web.

O JQuery é uma *framework* que ajuda os programadores a contornarem os problemas de incompatibilidade dos *browsers* actuais, permitindo assim que este se concentre

Existem para o JQuery vários *plugins* que permitem adicionar efeitos especiais nas páginas web e também controlar os dados via AJAX.

Uma função em Javascript puro tem uma diminuição notável quando reescrita com esta *framework*.

As principais funcionalidades do JQuery são:

- Resolução da incompatibilidade entre *browsers*;
- Redução de código;
- Reutilização do código através de *plugins*;
- Utilização de uma vasta quantidade de *plugins* criados por outros programadores;
- Trabalha com AJAX e DOM; e
- Implementação segura de recursos do CSS1, CSS2 e CSS3.

4.6 Model-View-Controller

O Model-View-Controller (MVC) é um padrão de arquitectura de software utilizado pela maioria das linguagens web, e que visa separar a lógica de negócio da lógica de apresentação, permitindo por isso o desenvolvimento, testes e manutenção isolado de ambos.

A arquitectura de 3 camadas que está na figura seguinte representada é um modelo MVC:

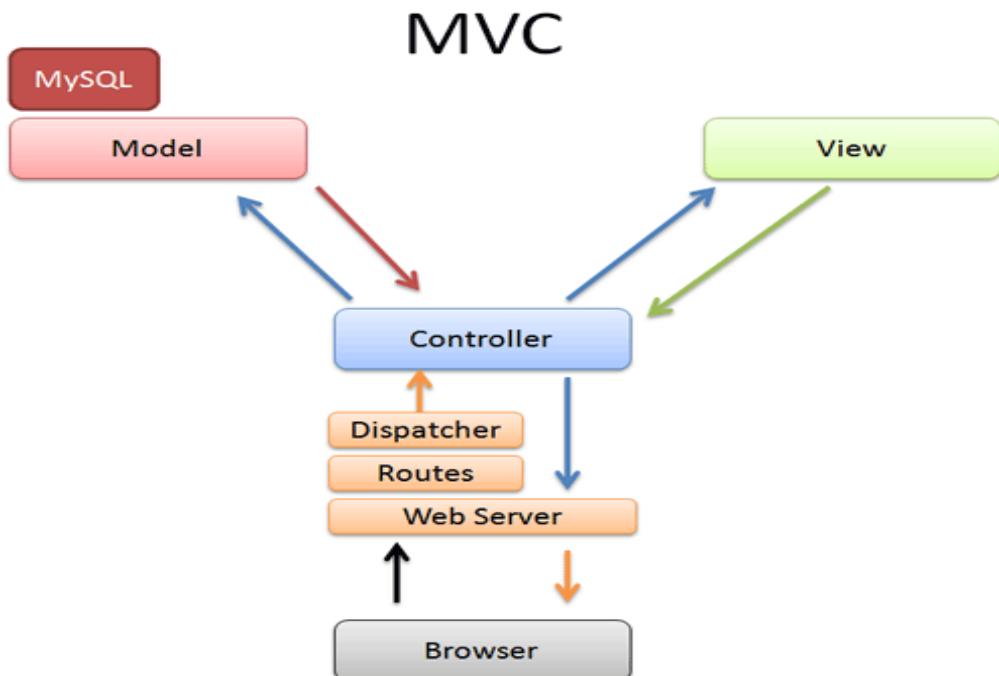


Figura 9 – Modelo MVC

O modelo (Model) é usado para definir e gerir o domínio da informação e notificar utilizadores sobre a mudança dos dados, e representa a informação que a aplicação opera. A lógica de negócio adiciona valor semântico aos dados, e quando há mudança de estado o modelo notifica os utilizadores.

A Visão (View) apresenta o modelo num formato adequado ao utilizador na saída de dados, neste caso o *browser*. Podem existir diferentes visões para um mesmo modelo, para diferentes propósitos.

O Controlador (Controller) recebe a entrada de dados e inicia a resposta ao utilizador ao invocar objectos do modelo, e por fim uma visão baseada na entrada. Ele também é responsável pela validação e filtragem da entrada de dados.

Um caso prático que implementa o modelo MVC é uma aplicação web em que a visão é um documento HMTL gerado pela aplicação. O Controlador recebe uma entrada GET ou POST após um estímulo de um utilizador decide como processá-la, invocando objectos do domínio para tratar a lógica de negócio, e por fim invocando uma visão para apresentar à saída.

4.7 Asynchronous Javascript and XML



O AJAX não é uma nova tecnologia que temos de aprender, mas sim uma nova maneira de pensar como vamos fazer utilizando tecnologias que já nos são familiares como o Javascript e XML.

O AJAX permite-nos criar páginas mais interactivas, com melhor resposta aos pedidos feitos e mais fáceis de utilizar. A utilização desta metodologia permite-nos fazer pedidos assíncronos ao servidor enviando apenas aquilo que nos interessa, podendo continuar a usar o resto da página enquanto esperamos pela resposta do servidor.

Uma das grandes vantagens é diminuir os tempos de carregamento das páginas web, porque apenas parte da página vai ser carregada ao contrário do método normal, em que todo o conteúdo da página é enviado e tem de ser reconstruído novamente na resposta do servidor.

Como funciona o AJAX:

- Utilizador faz um pedido via HTTP ao servidor;
- Na página existe código Javascript que se vai encarregar do pedido assíncrono ao servidor; e
- Quando o servidor responde ao código que criámos para fazer o pedido, vai dizer ao browser para refazer determinada parte da página.

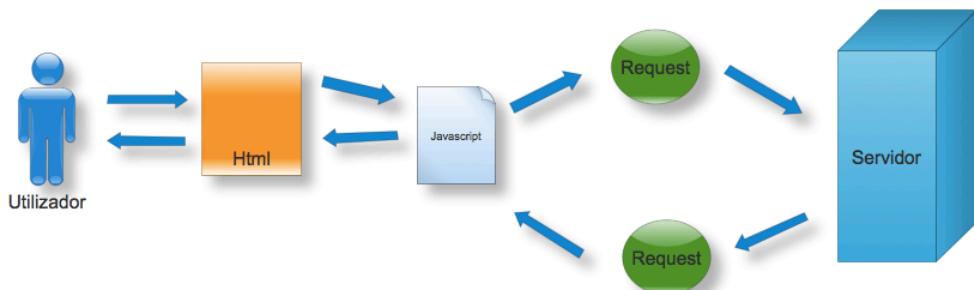


Figura 10 – Funcionamento AJAX

O AJAX incorpora no seu modelo:

- Exposição e interacção dinâmica usando o DOM;
- Intercâmbio e manipulação de dados usando XML e XSLT;
- Recuperação assíncrona de dados usando o objecto XMLHttpRequest e XMLHttpRequestResponse; e
- JavaScript fazendo a junção entre elementos.

5. Desenho do modelo de dados

Este capítulo tem por finalidade a descrição do desenho do modelo de dados do nosso projecto e determinar principalmente se o desenho cumpre com os requisitos anteriormente definidos, dando uma visão geral do sistema.

Nesta fase recorremos à linguagem UML para a modelação do sistema, combinando os diagramas disponibilizados pela linguagem, de forma a representar o comportamento dos objectos e classes que foram identificados como parte da aplicação.

No desenho do nosso modelo de dados utilizamos a normalização de tabelas, tendo como objectivo avaliar a qualidade do desenho das tabelas e transformá-lo num modelo de dados menos redundante e mais estável.

Na figura abaixo apresenta-se um resumo das várias fases da normalização de tabelas:

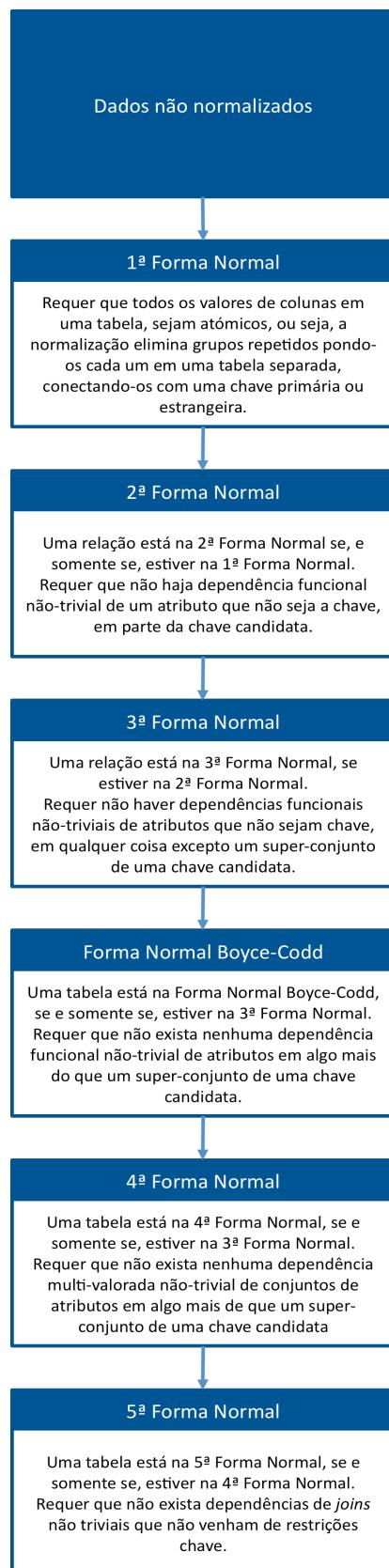


Figura 11 – Normalização de tabelas

O nosso modelo de dados encontra-se na 3^a Forma Normal da normalização de tabelas e está representado na figura seguinte:

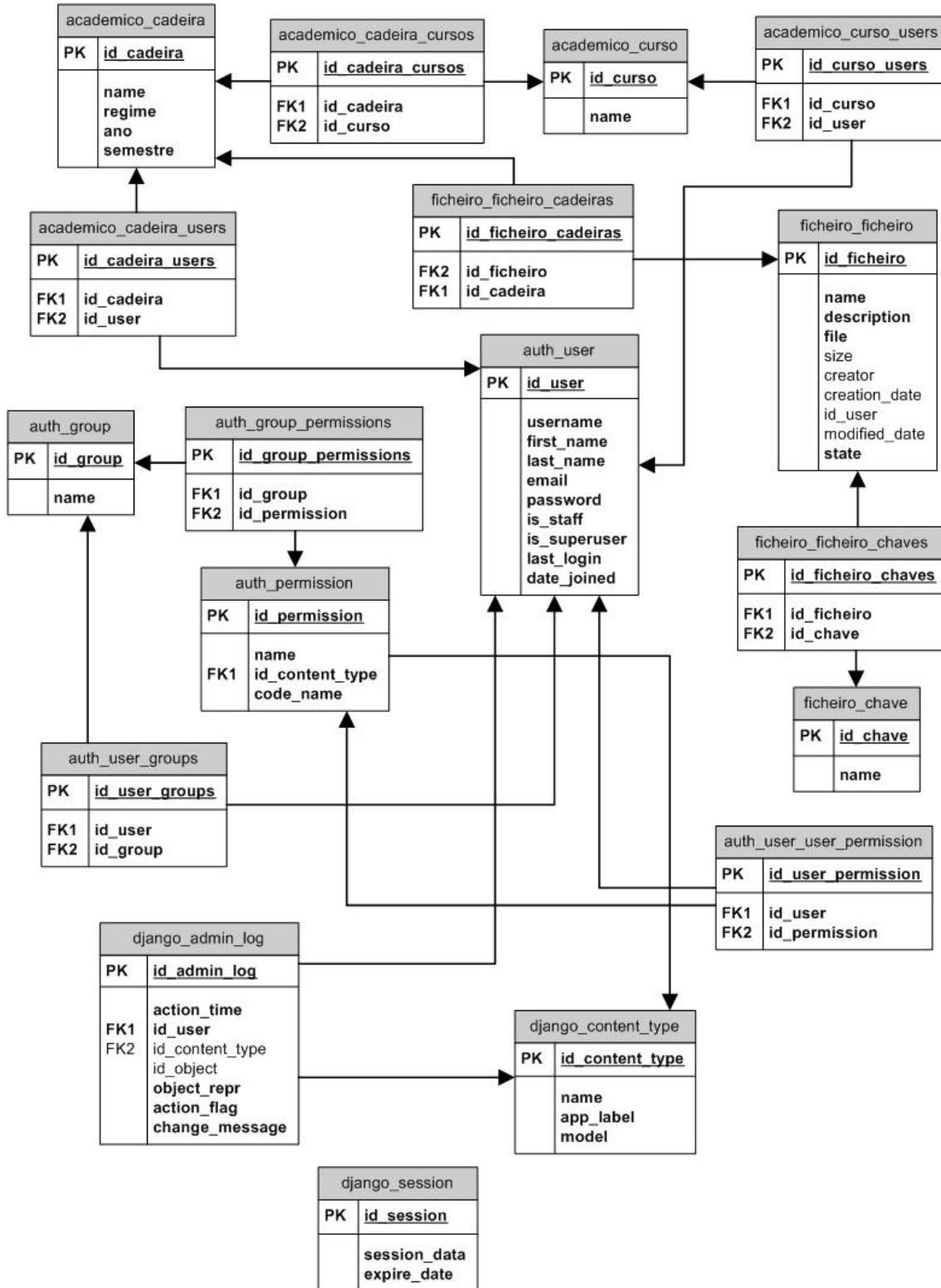


Figura 12 – Modelo de dados

5.1 Relações importantes

Este capítulo tem como objectivo salientar algumas tabelas e respectivas relações importantes para a nossa aplicação.

A relação que se segue representa a associação das palavras-chave aos documentos existentes. É através desta relação que é possível fazer as pesquisas dos documentos e respectiva visualização.

Esta relação é de n para n, o que significa que uma palavra-chave pode estar associada a vários documentos e os documentos podem estar associados a várias palavras-chave.

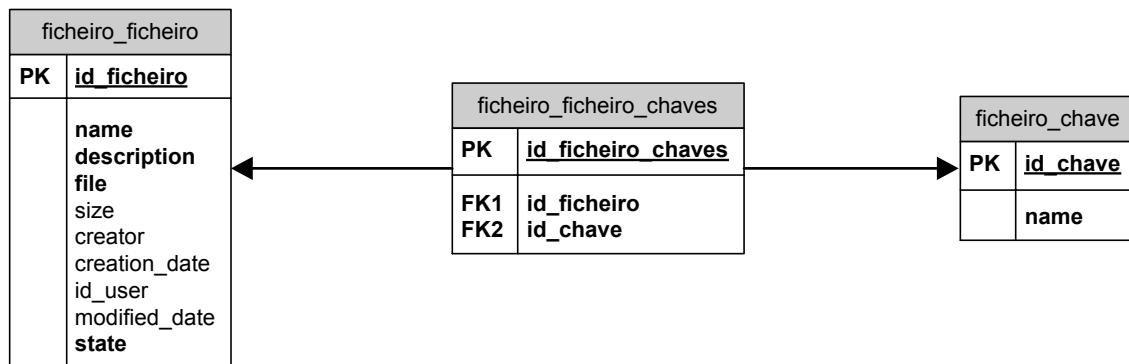


Figura 13 – Relação de associação das palavras-chave aos documentos

A relação seguinte representa a associação dos documentos às disciplinas. É através desta relação que o docente obtém os relatórios dos conteúdos por docente e ano/semestre/disciplina.

Esta relação é de n para n, o que significa que uma disciplina pode ter vários documentos associados e os documentos podem estar associados a várias disciplinas.

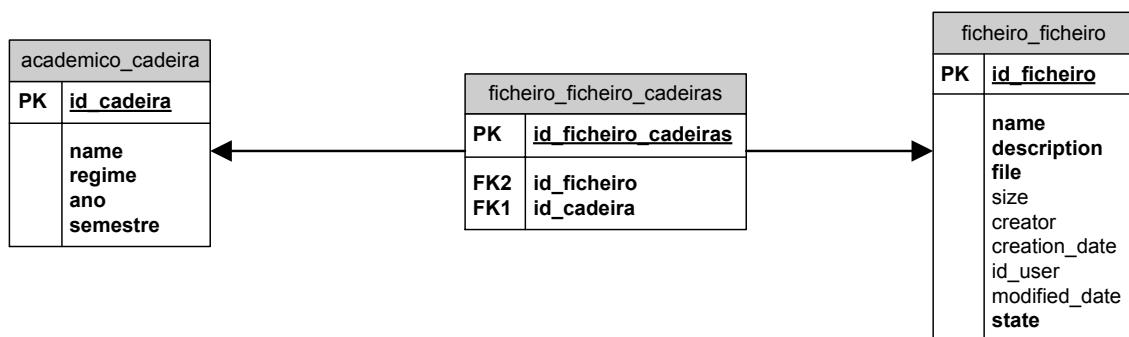


Figura 14 – Relação de associação de documentos às disciplinas

A relação abaixo representa a associação das disciplinas aos utilizadores. É através desta relação que os docentes fazem a gestão das disciplinas (adicionar, remover e modificar).

Esta relação é de n para n, o que significa que uma disciplina pode ter mais do que um utilizador (docente) associado e os utilizadores podem estar associados a várias disciplinas.

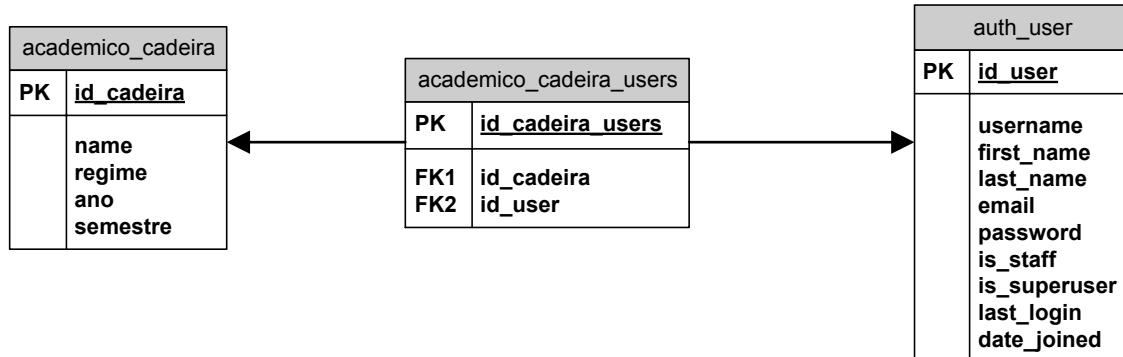


Figura 15 – Relação de associação de disciplinas aos utilizadores

6. Funcionalidades principais

Este capítulo tem como objectivo evidenciar a forma como foram desenvolvidas as funcionalidades de pesquisa de documentos e catalogação dos documentos por palavra-chave.

6.1 Pesquisar documentos

Um dos objectivos da nossa aplicação é de permitir aos alunos dos cursos da ECATI a pesquisa de documentos por palavra-chave.

Quando um aluno faz uma pesquisa de documentos, a aplicação faz uma chamada assíncrona ao servidor através de AJAX, verifica na tabela de chaves todos os valores que se enquadram no texto que o aluno está a inserir e mostra no ecrã todas as opções disponíveis.



Figura 16 – Pesquisa de documentos por palavra-chave

Adicionalmente ao que nos foi proposto, a aplicação também mostra no ecrã o nome dos documentos existentes no servidor que se enquadram no texto que o aluno está a inserir.

A aplicação permite também ao aluno inserir o texto que pretender mesmo que não esteja definido como palavra-chave na aplicação.

6.2 Catalogar documentos por palavra-chave

A nossa aplicação permite aos docentes das disciplinas dos cursos da ECATI associar palavras-chave aos documentos, de forma a facilitar a pesquisa dos mesmos.

Quando é adicionado um documento através da aplicação pode-se associar as palavras-chave anteriormente criadas.

É necessário ter em atenção o nome do documento inserido, uma vez que a aplicação também permite a pesquisa por este campo.



Administração da Plataforma de Gestão de Conteúdos dos Cursos da ECATI

Bem-vindo, admin. Modificar palavra-passe / Sair

História

Nome: Java em 24 horas

Descrição: Aprenda Java de uma fácil e divertida.

Ficheiro: Actualmente: uploads/frases.rtf
Modificar: [Choose File] No file chosen

Estado: Disponível

Chaves:

| Available keys | Selected keys |
|---------------------------------|--------------------|
| Microsoft .NET Base de Dados | Java Algoritmos |

Escolher todos Limpar tudo

Figura 17 – Catalogar documentos por palavra-chave

7. Próximos passos

Este capítulo descreve as próximas funcionalidades que se podem acrescentar à nossa aplicação.

As funcionalidades que podem ser adicionadas são:

- **Envio de notificações** – Permitir aos alunos receberem notificações, via e-mail ou directamente na página inicial, a avisar que foram inseridos documentos na aplicação ou efectuadas alterações; e
- **Comentar ficheiros** – Permitir aos alunos e docentes inserirem comentários aos documentos existentes na aplicação.

O nosso modelo de dados encontra-se preparado para estas funcionalidades, podendo ser apenas necessário acrescentar algum campo.

8. Conclusão

A utilização da internet como nova forma de interacção no processo educativo amplia a acção de comunicação entre o aluno e o docente. Desta forma, o acto de educar com o auxílio da internet proporciona a quebra de barreiras e remove o isolamento da sala de aula, acelerando a autonomia da aprendizagem dos alunos.

Assim sendo, os docentes do curso de Engenharia Informática da ECATI solicitaram uma solução por parte dos alunos que frequentam a disciplina de Projecto de Final de Curso para o desenvolvimento de um repositório de documentos com pesquisa fácil por palavra-chave.

Para desenvolvimento da plataforma de gestão de conteúdos dos cursos da ECATI utilizámos tecnologias *open source* que nos permitiu usufruir da ajuda das comunidades existentes que nos responderam a todas as dúvidas que surgiram ao longo deste projecto.

A tecnologia principal utilizada foi a *framework* Django que nos permitiu obter uma estrutura de trabalho organizada e facultou-nos tempo para nos dedicarmos às questões mais funcionais deste projecto.

O uso das tecnologias como a *framework* JQuery ajudou-nos a que a nossa aplicação fosse compatível com os vários *browsers* utilizados hoje em dia, sem ser necessário efectuar uma implementação específica para cada um deles.

O facto de termos estruturado o modelo de dados da nossa aplicação permite que no futuro seja possível adicionar com relativa facilidade novas funcionalidades.

Este projecto permitiu-nos utilizar e integrar um conjunto vasto de tecnologias *open source* disponíveis, o que aumentou o nosso conhecimento sobre as ferramentas actualmente existentes e permitiu-nos aumentar a nossa produtividade.

9. Referências

As referências utilizadas ao longo do desenvolvimento do nosso projecto foram as seguintes:

- Earle Castledine & Craig Sharkie (2010), *JQuery novice to ninja*, Sitepoint.
- Rebecca M. Riordan (2008), *Head First Ajax*, O'Reilly
- Michael Kofler (2005), *The Definitive guide to Mysql 5*, Third Edition, Apress
- Edgar F. Codd (1974), *Recent Investigations into Relational Data Base Systems*, IBM Research Report

Páginas Web utilizadas:

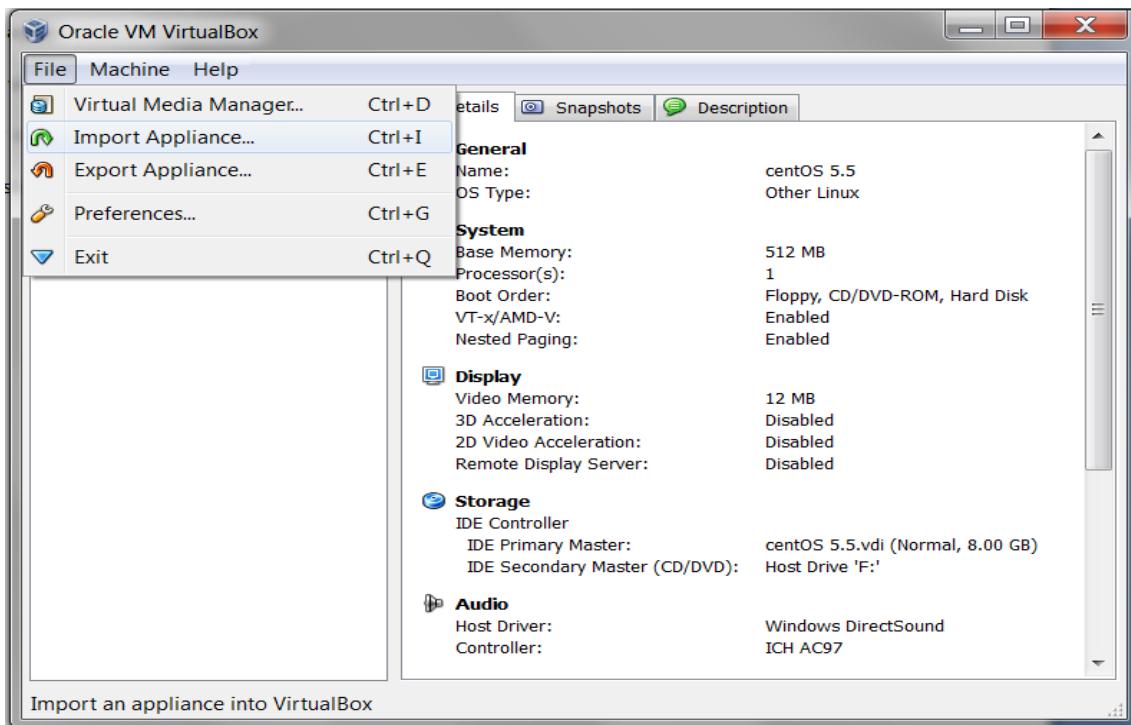
- http://pt.wikipedia.org/wiki/Normaliza%C3%A7%C3%A3o_de_dados
- <http://www.djangoproject.com/>
- <http://pt.wikipedia.org/wiki/MySQL>
- http://pt.wikipedia.org/wiki/Servidor_Apache
- [http://en.wikipedia.org/wiki/Python_\(programming_language\)](http://en.wikipedia.org/wiki/Python_(programming_language))
- <http://en.wikipedia.org/wiki/Model–View–Controller>
- http://www.macoratti.net/vbn_mvc.htm

10. Anexos

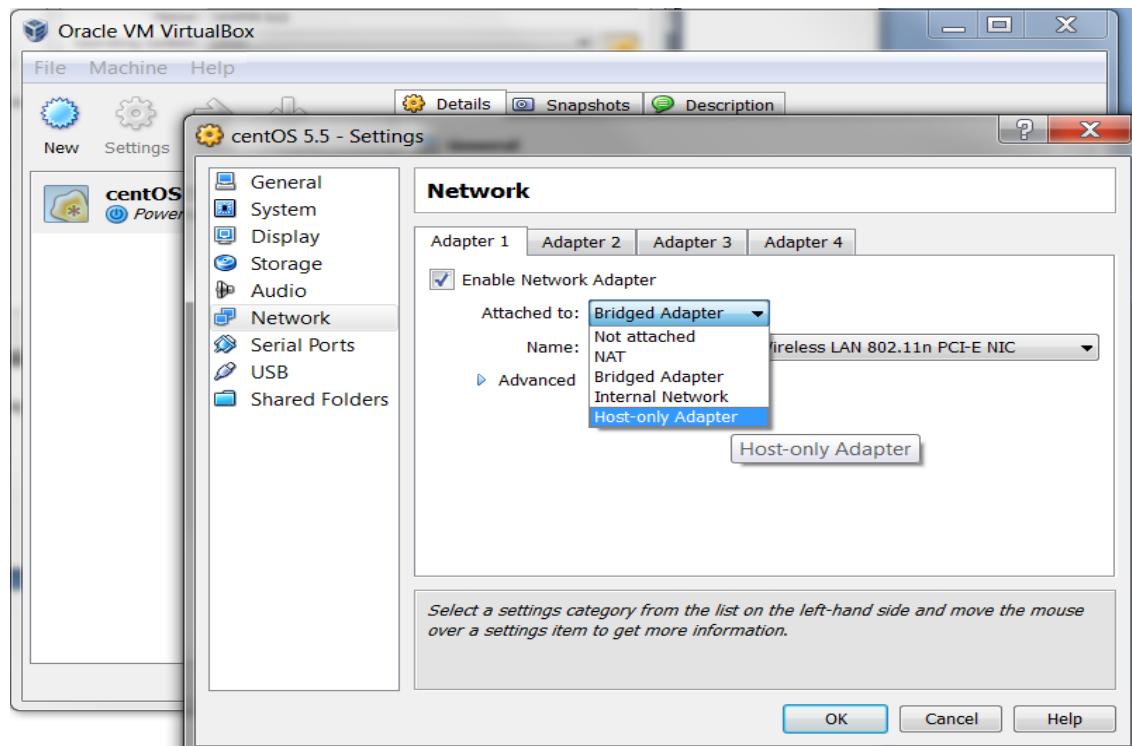
Anexo I – Manual de instalação

Para utilizar a nossa aplicação deve utilizar a imagem virtual:

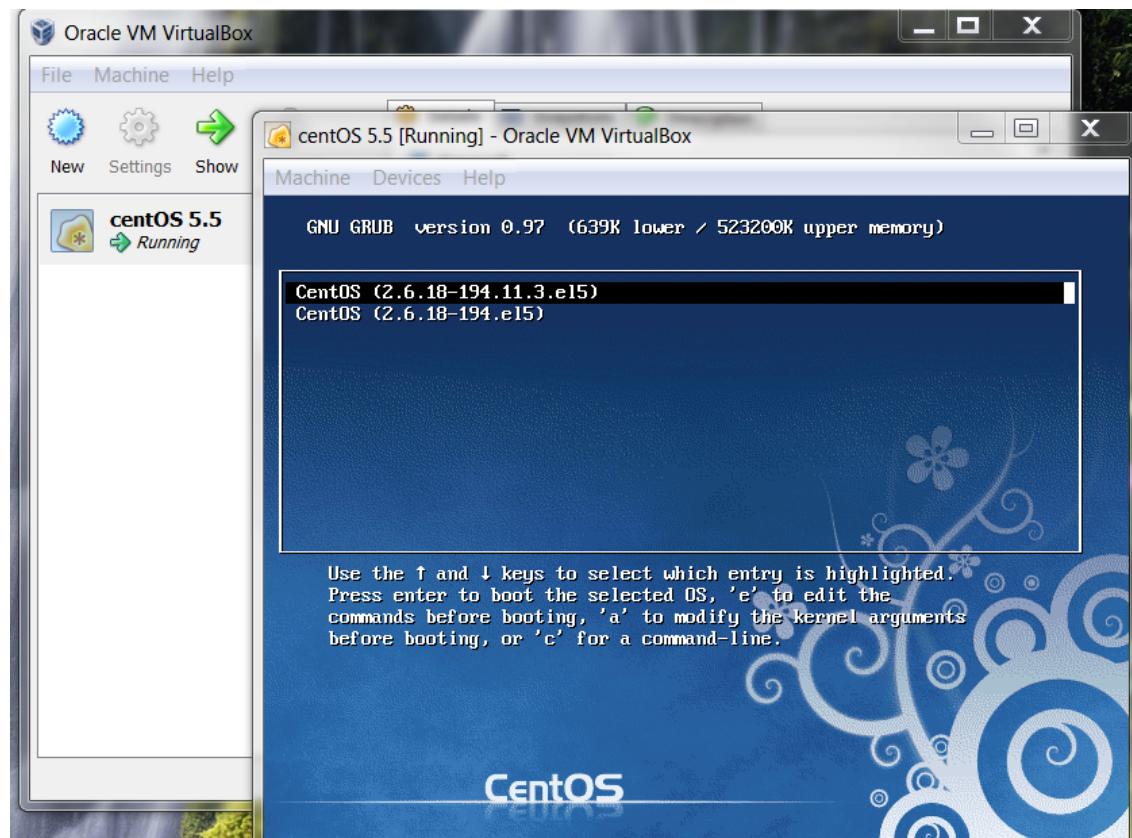
- Instalar o Virtual Box que se encontra no DVD
- Após instalado, importar a imagem da máquina virtual que se encontra no DVD



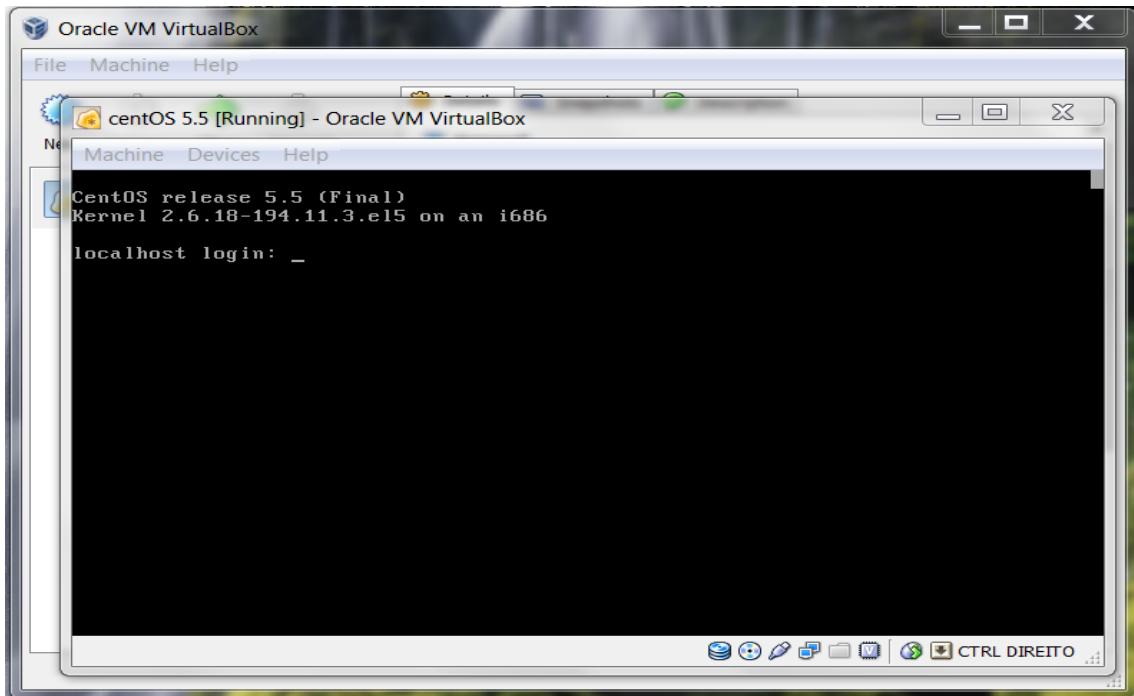
- Ir às *settings* da nova máquina virtual e confirmar o hardware



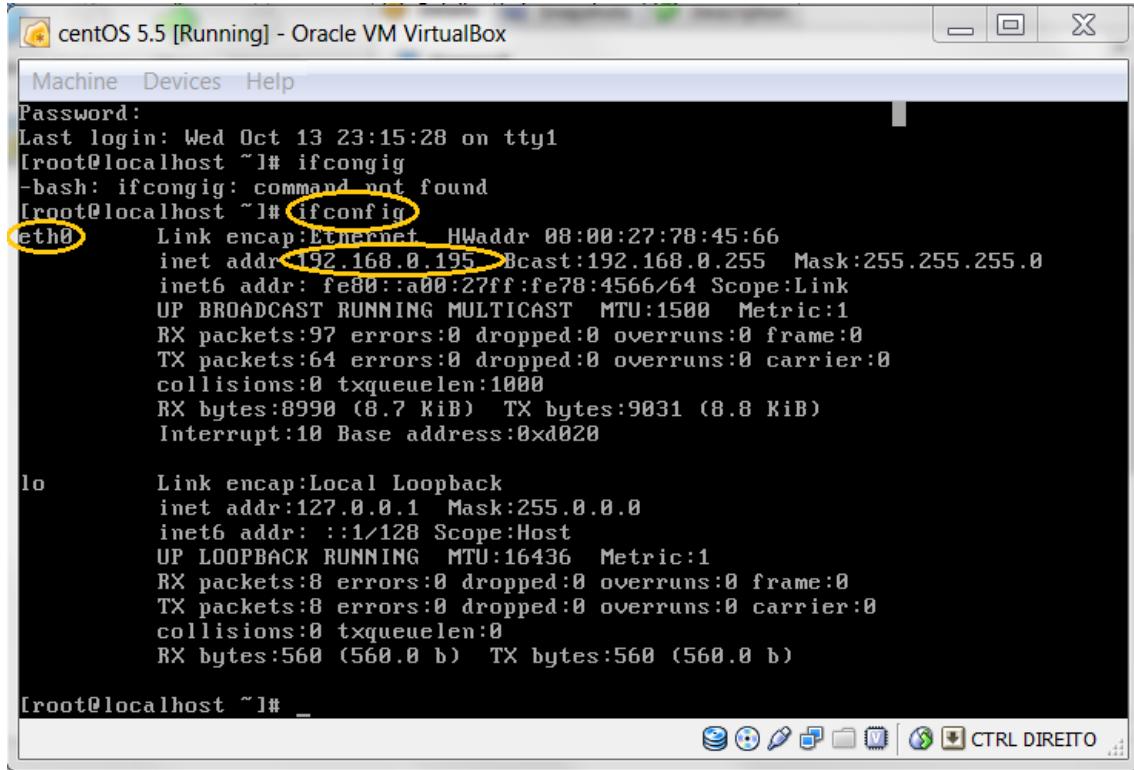
- Colocar o network adapter como “Host-only Adapter”
- Arrancar a máquina virtual



- Entrar como *root* na máquina
 - Login: root
 - Password: root123



- O *source* do projecto encontra-se em “/usr/share/xfile”
- Verificar qual o IP da máquina para poder usar um *browser* para aceder ao mesmo



```
centOS 5.5 [Running] - Oracle VM VirtualBox
Machine Devices Help
Password:
Last login: Wed Oct 13 23:15:28 on tty1
[root@localhost ~]# ifconig
-bash: ifconig: command not found
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:78:45:66
          inet addr:192.168.0.195 Bcast:192.168.0.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe78:4566/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:97 errors:0 dropped:0 overruns:0 frame:0
            TX packets:64 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:8990 (8.7 KiB) TX bytes:9031 (8.8 KiB)
            Interrupt:10 Base address:0xd020

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:8 errors:0 dropped:0 overruns:0 frame:0
            TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:560 (560.0 b) TX bytes:560 (560.0 b)

[root@localhost ~]# _
```

- Usar o *browser* para aceder à aplicação



- Para desligar a máquina basta fazer “poweroff” na linha de comandos

Anexo II – Quick Guide

1. Introdução

A plataforma de gestão de conteúdos dos cursos da ECATI foi desenvolvida de forma a proporcionar no futuro uma plataforma que responda às necessidades de todos os alunos da Escola de Comunicação, Artes e tecnologias da Informação. Considerada a importância de garantir uma ferramenta de repositório dos matérias didácticos de apoio as aulas a plataforma de gestão de conteúdos dos cursos da ECATI afirma-se como um opção adequada.

2. Funcionalidades Comuns

2.1 Adicionar dados

A criação de dados na plataforma pode ser feita de duas maneiras, utilizando o botão adicionar:



Figura 1 – Botão genérico para adicionar dados

Ou através do ecrã principal de cada funcionalidade carregando no botão de Adicionar “Funcionalidade”:

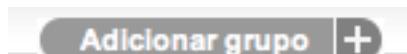


Figura 2 – Botão genérico para adicionar Grupo, Utilizadores, Cursos, Disciplinas, Palavras-Chave e Documentos

Depois de toda a informação ser inserida podemos utilizar um dos seguintes botões para guardar a infirmação:

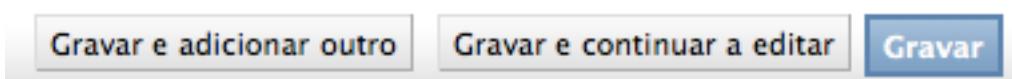


Figura 3 – Botão genérico para gravar os dados na aplicação

2.2 Modificar dados

A modificação de dados pode ser feita acendendo através do ecrã principal da gestão da plataforma de conteúdos, bastando para isso carregar no botão de modificar:



Figura 4 – Botão genérico de modificação de dados

Pode também fazer modificações através do ecrã da funcionalidade que esta a utilizar. Bastando para isso escolher os dados que pretende alterar. Para guardar as modificações efectuadas utilize um dos seguintes botões:



Figura 5 – Botão genérico para gravar os dados na aplicação

2.3 Remover dados

Para remover dados da plataforma de gestão, devemos aceder ao menu principal de cada funcionalidade. Quando nos encontramos no menu onde temos listados os dados que pretendemos eliminar, devemos então seleccionar os registos a serem eliminadas como demonstra a imagem seguinte:

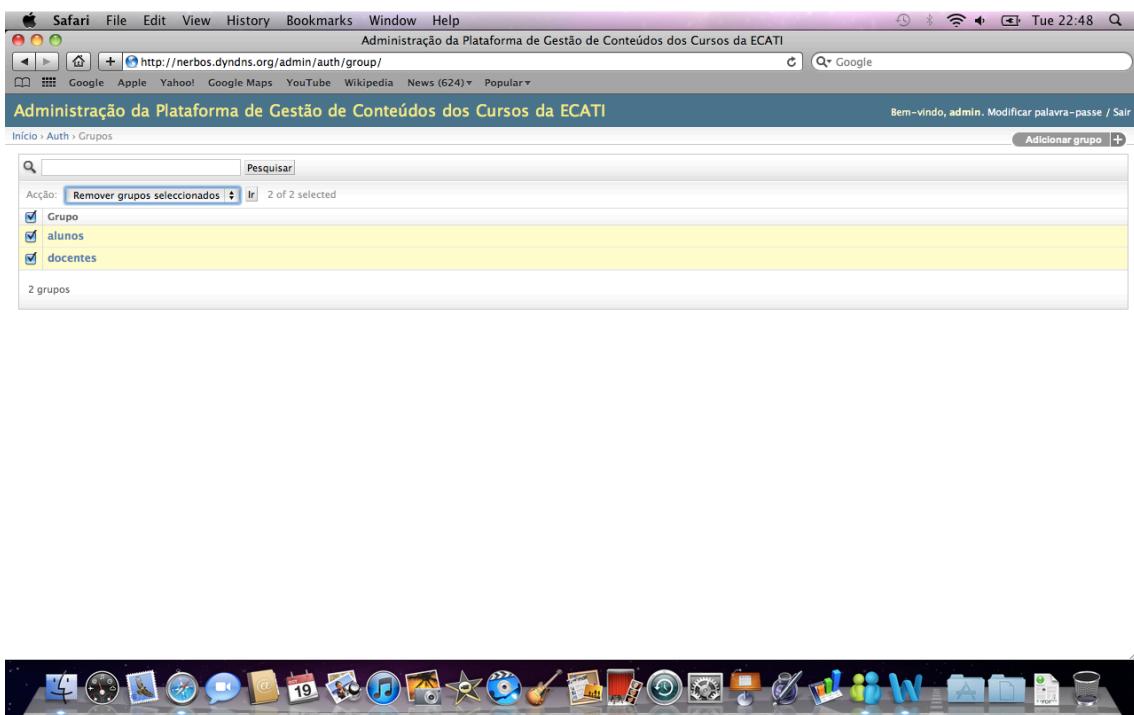


Figura 6 – Seleccionar os registo que pretende eliminar

Para remover os registo no campo acção deve seleccionar a opção de remover.

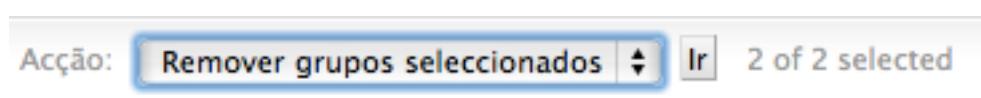


Figura 7 – Seleccionar a opção de remoção dos registo seleccionados

Podemos ainda remover os dados quando estamos a ver o seu detalhe, utilizando o botão:



Figura 8 – Botão para remover o registo que estamos a visualizar

3. Utilização da aplicação

Neste capítulo iremos descrever como utilizar a aplicação de gestão de conteúdos. Para aceder à aplicação é necessário ter acesso à internet e ter instalado no seu pc um browser. Para entrar na aplicação deverá utilizar o endereço do servidor onde se encontra instalada a aplicação.

Depois de inserir o endereço deverá surgir o seguinte ecrã:



Figura 9 – Ecrã principal da aplicação

3.1 Administração

Numa primeira utilização da aplicação, é necessário entrar com o utilizador de administração.

Dados *default* do administrador da aplicação:

- **User:** admin
- **Pass:** admin

Para isso devemos aceder ao menu de autenticação através do *link*:



Figura 10 – Opção que permite o utilizador ter acesso ao menu de administração

No menu de autenticação o utilizador deverá inserir os dados que foram fornecidos anteriormente:



Figura 11 – Menu de autenticação na aplicação

Depois de o *login* ter sido feito com sucesso, o utilizador administrador deverá ter acesso à opção de administração como ilustra a figura seguinte:

Escola de Comunicação, Artes e Tecnologias da Informação



Figura 12 – Menu com a opção de administração disponível

O utilizador deverá carregar na opção de administração, para ser redireccionado para o ecrã de administração. Estarão disponíveis as acções de criação, modificação e eliminação sobre as funcionalidades de:

- Gestão de Grupos;
- Gestão de Cursos;
- Gestão de Cadeiras;
- Gestão Palavras-chave;
- Gestão de Documentos; e
- Criação de Relatórios.

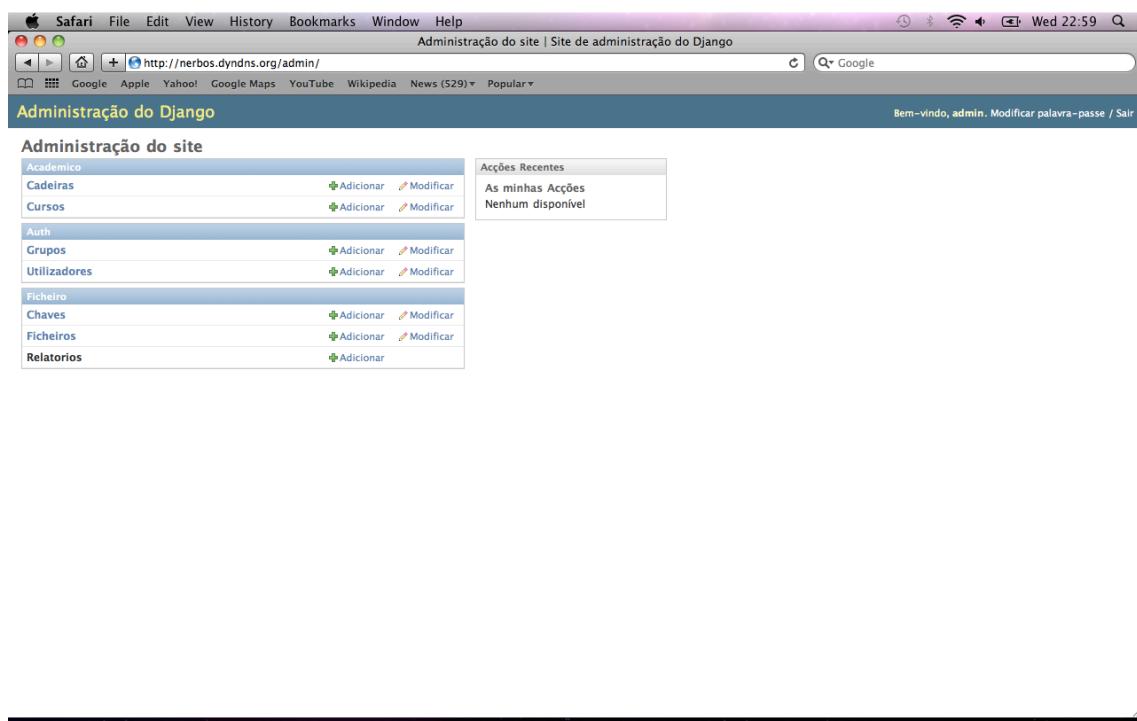


Figura 13 – Menu de administração

3.1.1 Gestão de grupos

O utilizador de administração deverá criar os grupos de utilização, e atribuir permissões de acesso. A estes grupos serão posteriormente associados os utilizadores da aplicação. Esta parametrização é importante pois permite controlar o acesso dos utilizadores as funcionalidades da aplicação.

É possível criar, modificar e eliminar um grupo, para isso deve usar as funcionalidades que foram descritas no capítulo “Funcionalidades Comuns”.

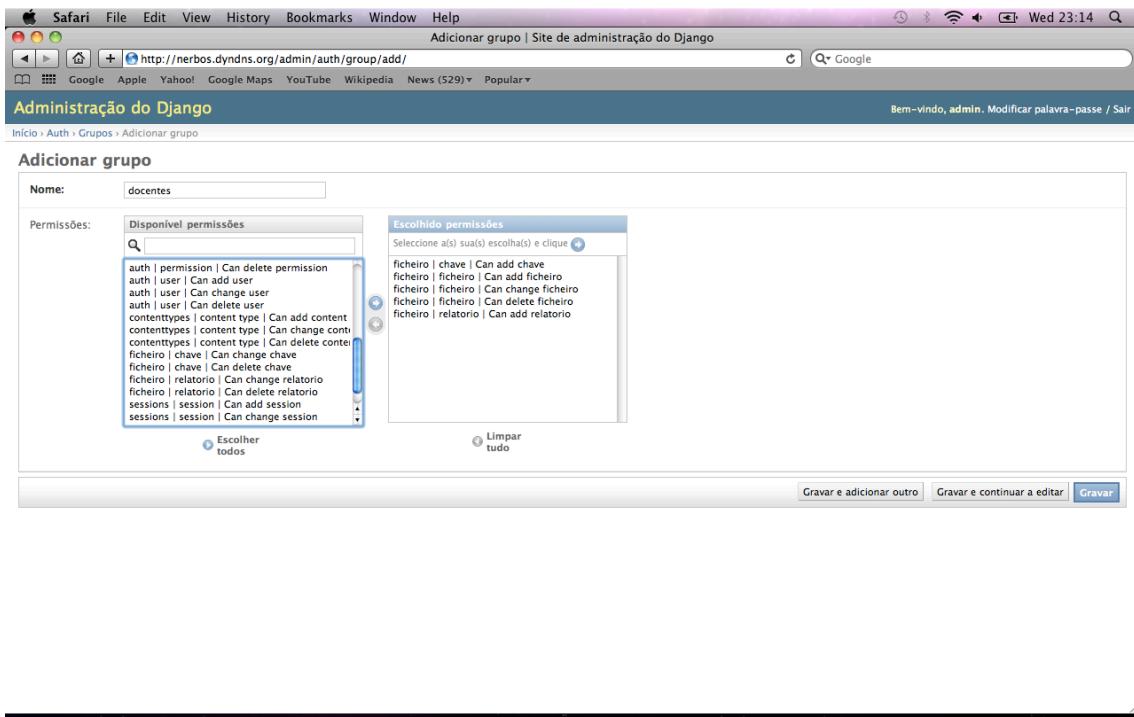


Figura 14 – Menu de detalhe de um grupo

3.1.2 Gestão de utilizadores

Depois de terem sido criados os grupos de utilizadores, pode começar a criar os utilizadores da aplicação.

Deve aceder ao menu de utilizadores através do ecrã principal de administração, carregando para isso na opção “Utilizadores” ou na opção de adicionar.

No ecrã de parametrização de um utilizador, os campos devem ser todos preenchidos e devem ser dadas a permissões de acesso à aplicação.

Escola de Comunicação, Artes e Tecnologias da Informação

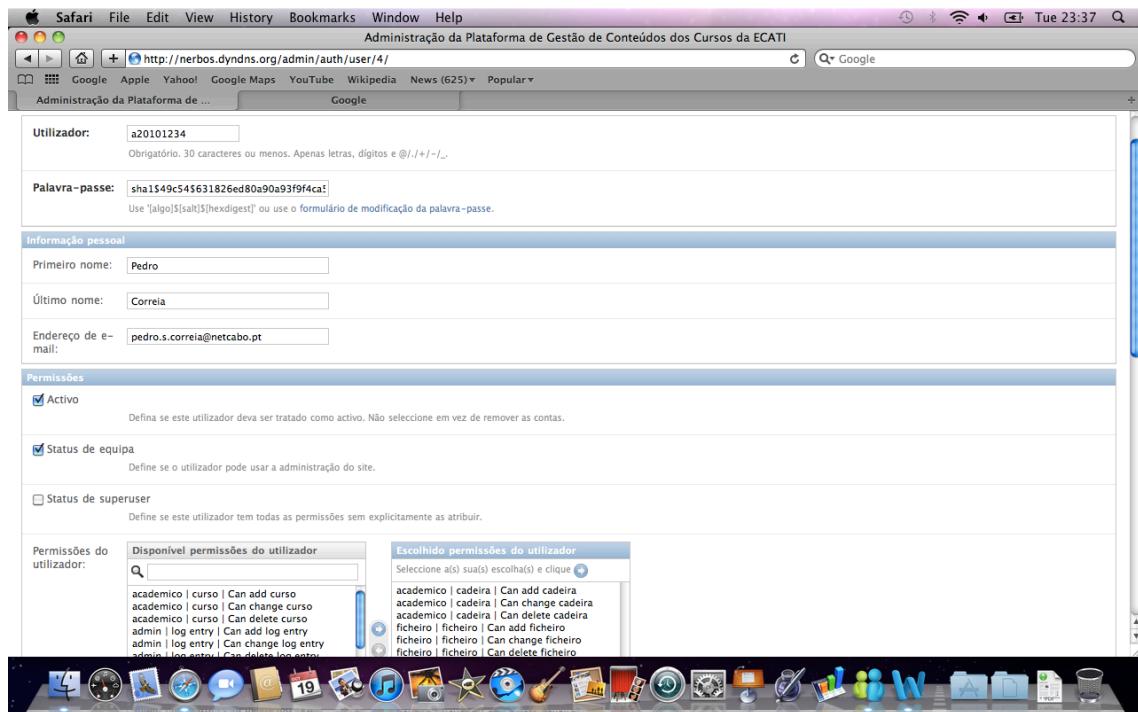


Figura 15 – Menu de detalhe de um utilizador

É neste momento que o utilizador também deve ser associado a um dos grupos criados no capítulo anterior.

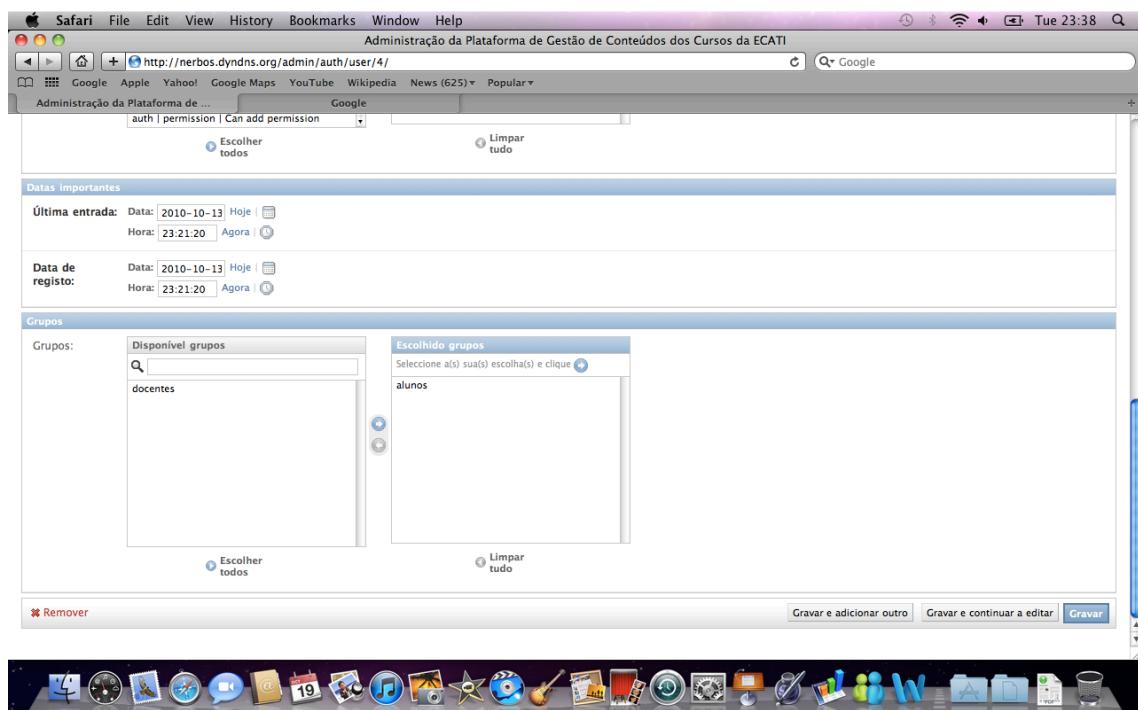


Figura 16 – Associar o utilizador a um grupo da aplicação

Escola de Comunicação, Artes e Tecnologias da Informação

O utilizador pode criar, modificar e eliminar um utilizador da plataforma de gestão de conteúdos, para isso deve usar as funcionalidades que foram descritas no capítulo “Funcionalidades Comuns”.

3.1.3 Gestão de cursos

Para inserir um curso na aplicação, pode aceder através do ecrã principal de administração e carregar na opção adicionar curso ou através do menu de cursos.

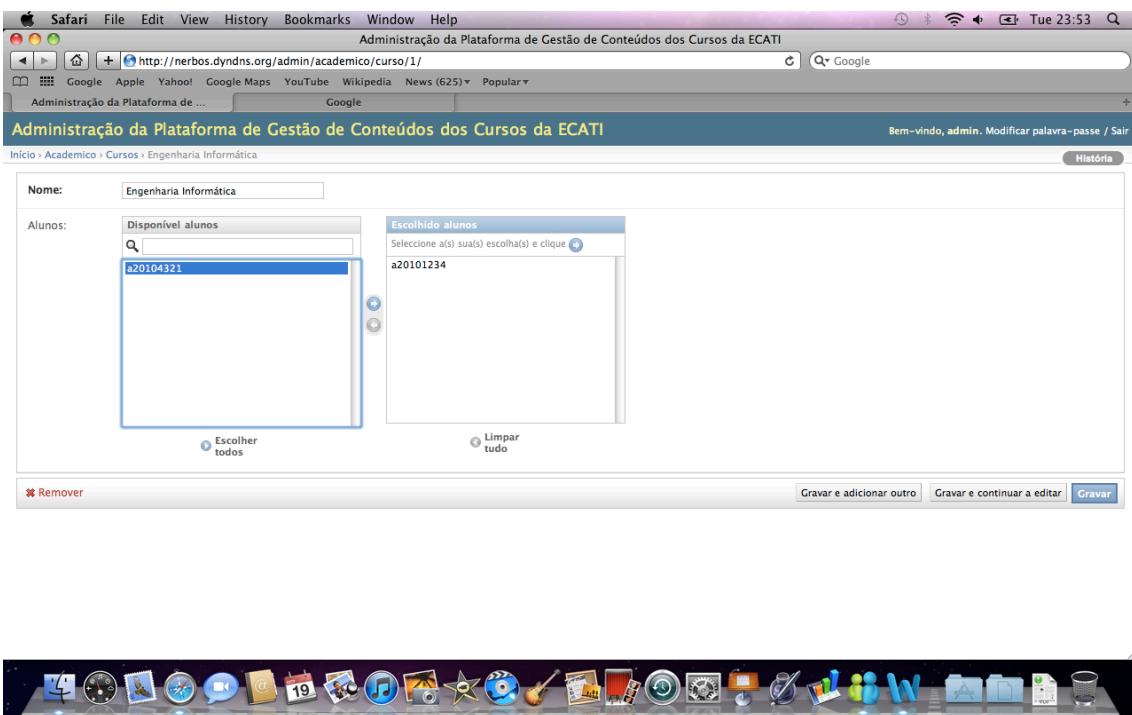


Figura 17 – Menu de detalhe de um Curso

O utilizador pode criar, modificar e eliminar um curso da plataforma de gestão de conteúdos, para isso deve usar as funcionalidades que foram descritas no capítulo “Funcionalidades Comuns”.

3.1.4 Gestão de disciplinas

Neste pode gerir as disciplinas dos cursos da universidade. Deverá aceder ao menu através do botão adicionar que se encontra no ecrã principal de administração, ou através do menu de disciplinas. No ecrã de parametrização da disciplina deve preencher todos os campos, como também associar a disciplina a um curso:

Escola de Comunicação, Artes e Tecnologias da Informação

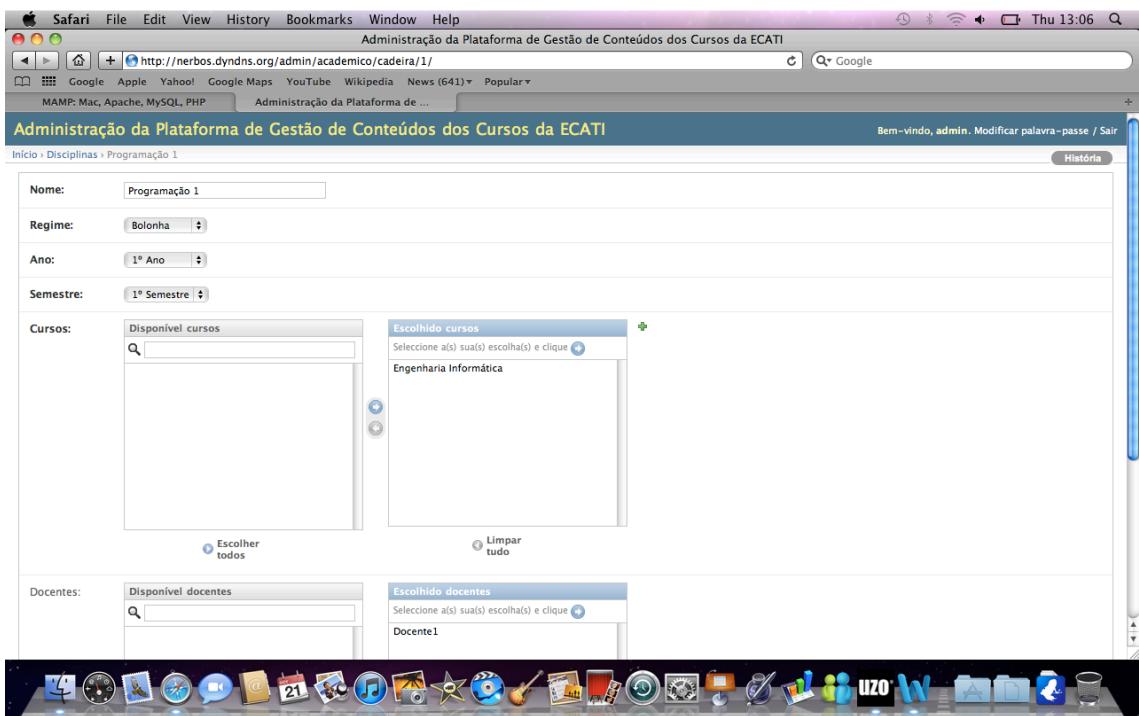


Figura 18 – Menu de detalhe de uma disciplina

As disciplinas podem ainda ser associadas aos docentes da universidade.

O utilizador pode criar, modificar e eliminar um disciplinas da plataforma de gestão de conteúdos, para isso deve usar as funcionalidades que foram descritas no capítulo “Funcionalidades Comuns”.

3.1.5 Gestão de chaves

Para gerir as palavras-chave deve aceder ao menu de gestão de palavras-chave através do menu principal da administração. Pode também aceder logo em modo de criação usando o botão adicionar que se encontra no ecrã principal da administração.

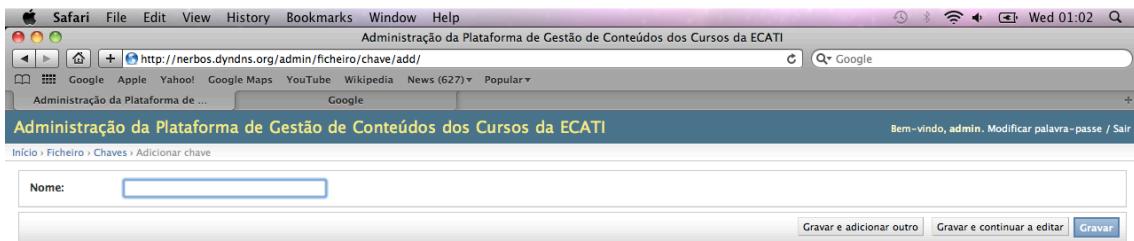


Figura 19 – Menu de detalhe de uma palavra-chave

O utilizador pode criar, modificar e eliminar uma palavra-chave da plataforma de gestão de conteúdos, para isso deve usar as funcionalidades que foram descritas no capítulo “Funcionalidades Comuns”.

3.1.6 Gestão de documentos

No menu de documentos o utilizador, pode fazer upload de documentos para o servidor e associar uma palavra-chave ao mesmo. Aceder ao menu de gestão de documentos através do ecrã principal de administração ou aceder directamente a criação de um documento carregando no botão adicionar que se encontra no ecrã principal de administração.

Escola de Comunicação, Artes e Tecnologias da Informação

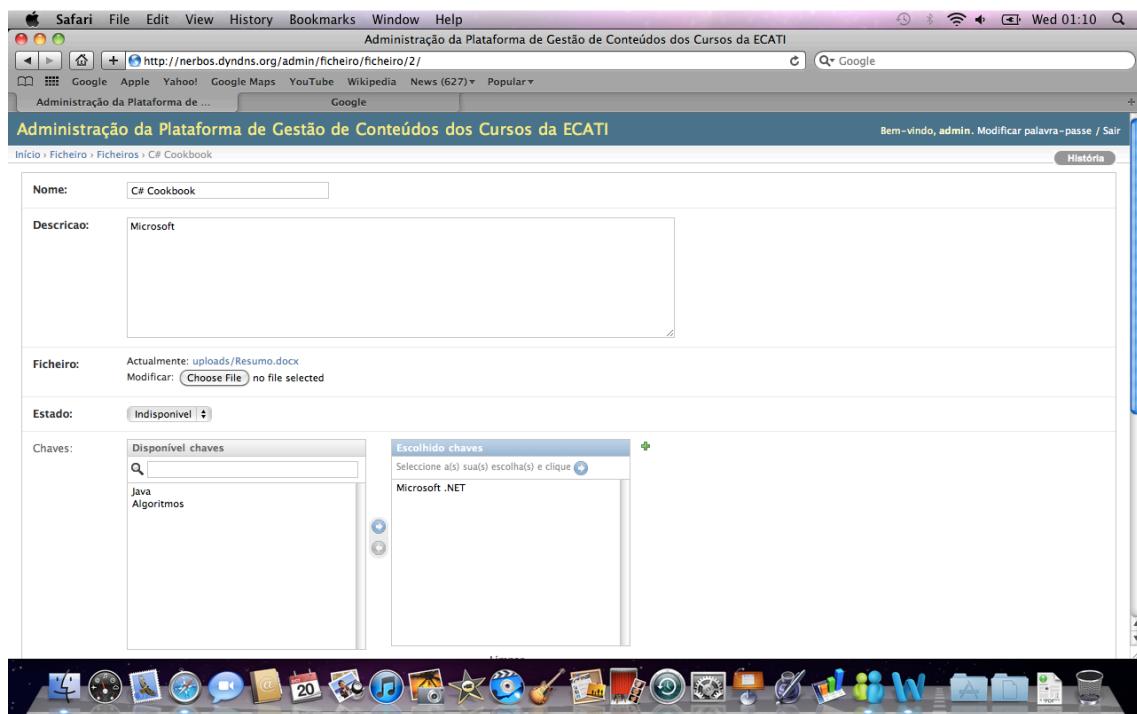


Figura 20 – Menu de detalhe de um documento

Na criação do documento este deve ser associado a uma cadeira.

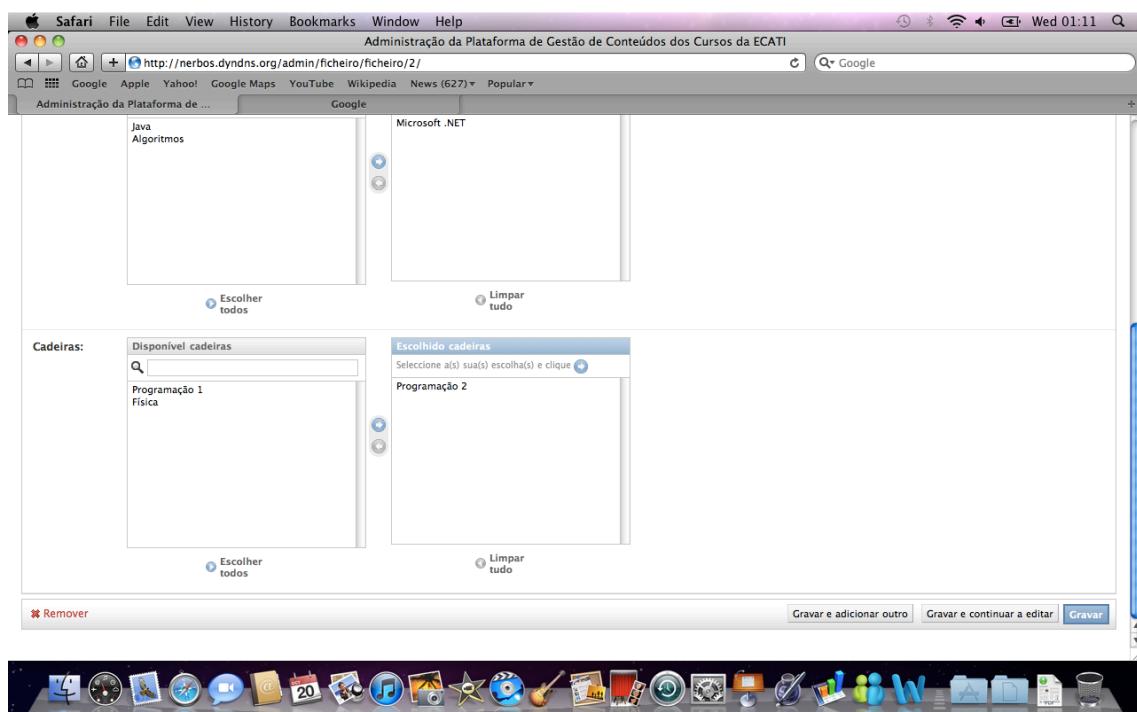


Figura 21 – Associar o ficheiro as disciplinas ou as palavras-chave

Atenção: O nome que é dado ao ficheiro no ecrã de parametrização, também será utilizado para a pesquisa por palavra-chave.

Quando um docente se encontra a alterar o documento que se encontra no servidor, deve colocar o estado do documento a indisponível. O ficheiro continua a estar disponível para consulta mas não é possível a outro docente alterar.

O utilizador pode criar, modificar e eliminar um documento da plataforma de gestão de conteúdos, para isso deve usar as funcionalidades que foram descritas no capítulo “Funcionalidades Comuns”.

3.1.7 Gerar relatórios

Para gerar um relatório os utilizadores devem aceder através do menu principal de administração e na opção de Relatórios carregar no botão de Gerar. No ecrã de relatório o utilizador é obrigado a preencher pelo menos o nome do ficheiro, o curso e o intervalo de tempo que pretendemos extrair.

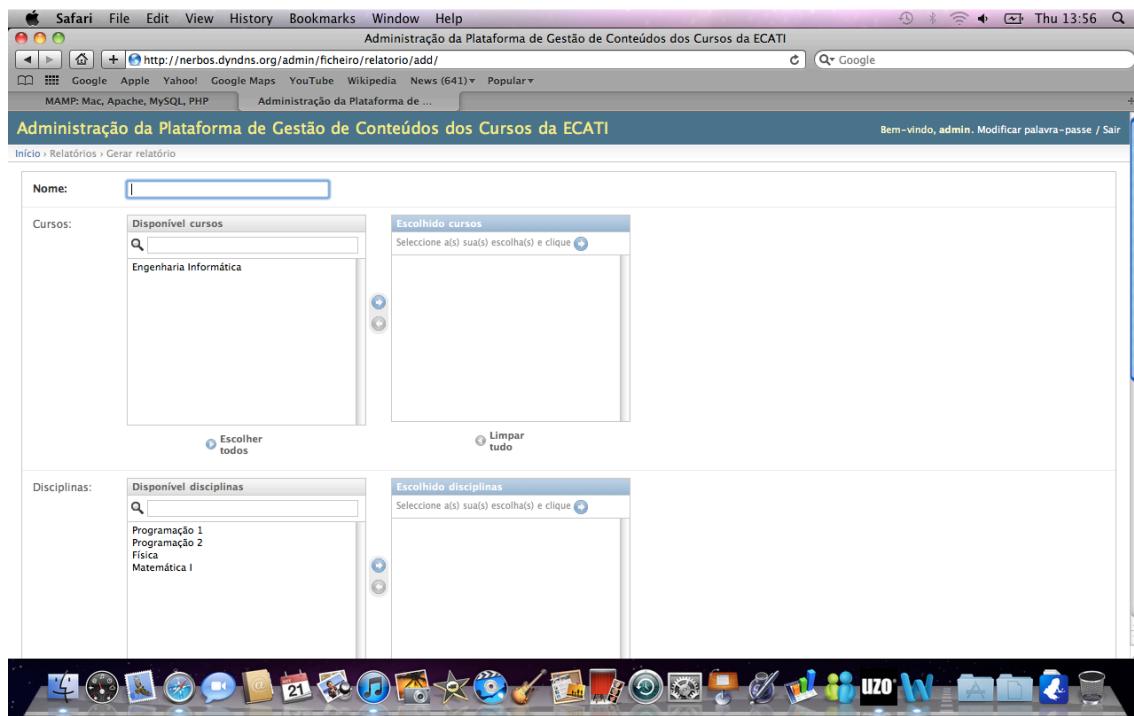


Figura 22 – Ecrã de detalhe do relatório

Para gerar o relatório carregar no botão gerar.



Escola de Comunicação, Artes e Tecnologias da Informação

A screenshot of a Mac OS X desktop with a Safari browser window open. The title bar says "Relatório" and the address bar shows "http://nerbos.dyndns.org/admin/ficheiro/relatorio/add/". The main content area displays a report titled "Relatório de Documentos" from the Universidade Lusófona. The report includes fields for "Curso: Engenharia Informática", "Disciplinas: Programação 1", "Docente: Docente1", "Nome: Java em 24 horas, criado em 2010-10-13 23:43:18", and "Total de ficheiros: 1 Ficheiro(s)". At the bottom are "Imprimir" and "Voltar" buttons. Below the browser is a dock with various application icons.

Figura 23 – Relatório em formato HTML

3.2 Pesquisa por palavra-chave

Para aceder à pesquisa por palavra-chave o utilizador deverá aceder através do ecrã principal da aplicação. Para isso deve carregar na opção pesquisa.



Escola de Comunicação, Artes e Tecnologias da Informação



Figura 24 – Menu de pesquisa por palavra-chave

Deve inserir no campo de pesquisa o conteúdo que pretende procurar.



Figura 25 – Sugestões da aplicação para o que está a ser inserido

Escola de Comunicação, Artes e Tecnologias da Informação

Quando o utilizador insere uma frase completa ou parcialmente completa que é uma chave que se encontra associada a um documento, vão sendo sugeridas varias opções como mostra a imagem anterior.

Mesmo que o utilizador insere uma pesquisa que não é uma palavra-chave, a pesquisa é feita pelo nome que foi inserido para o ficheiro.

Quando é inserida uma pesquisa valida, é retornada uma lista de ficheiros relacionados.



Figura 26 – Resultado da pesquisa efectuada

Poderá fazer o download do ficheiro carregando no *link* com o nome do ficheiro. Irá aceder a um menu que contem uma descrição pormenorizada do documento.

Escola de Comunicação, Artes e Tecnologias da Informação



Figura 27 – Detalhe do ficheiro

Para fazer *download* do ficheiro o utilizador deverá carregar no link “Fazer Download”.

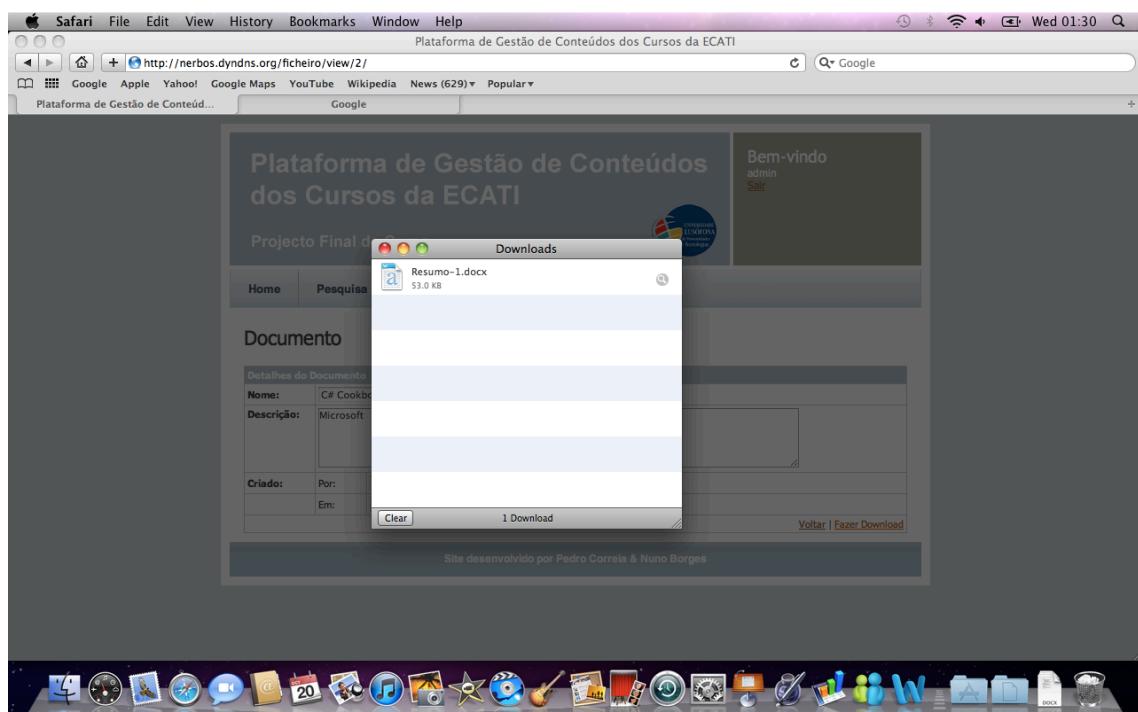


Figura 28 – Download do ficheiro

Poderá carregar no link “Voltar” para voltar ao ecrã da pesquisa que foi realizada.