

# Aplicação de gestão de uma biblioteca

Projecto final de curso, Licenciatura em Engenharia Informática

Alunos: Hugo Miguel Mota Rodrigues nº 2401632  
João António da Silva Lopes nº2401701

Orientador: Professor José Aser Castillo Lorenzo

Lisboa, Dezembro de 2008

**Índice**

Resumo .....	3
Abstract.....	4
Introdução .....	5
Enquadramento Teórico .....	6
Perfis de utilizador.....	6
Funcionalidades da aplicação .....	7
Tabelas e respectivos campos.....	8
Método.....	12
Imagem 1 – Diagrama de Casos de Uso.....	13
Imagem 2 – Diagrama Entidade Associação .....	14
Imagem 3 – Diagrama de estados do utilizador .....	15
Imagem 4 – Diagrama de estados do exemplar .....	15
Imagem 5 – Diagrama de estados do título .....	16
Imagem 6 – Diagrama de estados do empréstimo .....	16
Imagem 7 – Diagrama de estados da reserva .....	16
Imagem 8 – Matriz CRUD .....	17
Resultados.....	18
Imagem 9 – Login na aplicação.....	18
Imagem 10 – Inserir um utilizador .....	18
Imagem 11 – Pesquisa de utilizador .....	19
Imagem 12 – Listagem de utilizadores pesquisados .....	19
Imagem 13 – Alterar utilizador .....	20
Imagem 14 – Remover utilizador .....	20
Imagem 15 – Mensagem de utilizador removido .....	21
Conclusão .....	22
Bibliografia.....	23

## ***Resumo***

Este trabalho de fim de curso pretende construir uma aplicação que permite gerir de uma forma simples os recursos de uma biblioteca, sejam eles humanos (funcionários, utilizadores) ou materiais (livros). Esta aplicação automatiza os processos relevantes ao funcionamento de uma biblioteca, e permite agilizar e guardar os registos que vão sendo efectuados diariamente, desta forma a informação será facilmente introduzida, pesquisada, alterada e eliminada, mantendo uma organização estável e fiável dos dados existentes.

No desenvolvimento da aplicação foram utilizados diagramas UML para criação de modelos de suporte para a validação e compreensão do funcionamento da aplicação. A programação foi efectuada em HTML, Java Script, Java e SQL. Recorremos ao motor Oracle para armazenamento dos dados, e construímos a interface com o utilizador utilizando Servlets compatíveis com Tomcat, Oracle Container For Java (OC4J) ou outro Servlet container compatível com a norma. A ligação entre as Servlets e a Base de dados foi feita em JDBC. As servlets foram estruturadas de forma a separar a componente de criação de interface com o utilizador da componente de processamento.

***Abstract***

This final course work claims the construction of an application which allows, in a simple way, the managing of the resources of a library, whether they're human (employees, users) or materials (books). This application automatizes the relevant processes relatively to the working method of a library, and allows agility and keeps the registers that will be realized daily; if so, the information will be easily introduced, researched, changed and erased, maintaining a stable and reliable organisation of the existent data.

In the development of this application were used UML diagrams for the creation of supporting models for the validation and comprehension of the application functioning. The software was realized with HTML, Java Script, Java and SQL. We used the Oracle database engine to store data. The user interface was built using servlets that can run on Tomcat, Oracle container for Java (OC4J) or other servlet container compatible with the above. The connection between the servlets and the database was done with JDBC. The servlets were structured in order to divide the components of user interface and the components of processing.

## ***Introdução***

O tema deste trabalho de fim de curso surgiu da necessidade de querermos aprofundar os nossos conhecimentos na área de base de dados. Como tal imaginámos o desenvolvimento de uma aplicação de gestão para uma biblioteca.

Sentimos que actualmente e cada vez mais a informação é um recurso precioso a que se deve prestar extrema atenção, e por isso achamos que fazer uma gestão cuidada dessa mesma informação é uma mais valia em qualquer lado. Uma biblioteca não é excepção e daí termos pensado no desenvolvimento desta aplicação.

Assim sendo, o que pretendemos com a aplicação é organizar os dados referentes a uma biblioteca de uma forma simples e que ao mesmo tempo fosse o máximo possível isenta de erros. Isto inclui quatro operações fundamentais, inserir, alterar, pesquisar e eliminar dados, e é basicamente nisso que a aplicação se baseia.

Com estas quatro operações conseguimos fazer uma gestão dos utilizadores e do empréstimo de livros da biblioteca.

É importante referir que o acesso à aplicação é feito através de um nome de utilizador e de uma palavra-chave, ou seja, é sempre preciso um registo na base de dados para se poder utilizar a aplicação. O que nos leva a outra tema muito importante, a questão da segurança da aplicação e dos dados armazenados.

Como nem todos os utilizadores podem ter o mesmo tipo de acesso à aplicação foram criados três perfis de utilizador, leitor, funcionário e bibliotecário, cada um deles tem as suas funções e acessos restritos relacionados com o seu perfil, o que impede que os dados sejam manipulados e corrompidos por alguém que não deva ter acesso a eles.

Num campo mais técnico e em relação às tecnologias utilizadas no desenvolvimento da aplicação temos o uso de UML, que nos permitiu representar de uma forma simples e fiável a realidade que pretendemos implementar através de vários diagramas (Casos de Uso, Entidade Associação e Estados). O HTML e o Java Script foram usados no desenvolvimento dos ecrãs de interface com o utilizador. Optámos ainda por programar as servlets em Java para fazer a ligação à base de dados. E por fim, para a manipulação e obtenção dos dados armazenados na base de dados da Oracle usámos SQL.

## ***Enquadramento Teórico***

Com vista ao desenvolvimento de uma aplicação que gere uma biblioteca há alguns aspectos importantes a definir: nomeadamente a questão da segurança e dos privilégios de acesso às diferentes funcionalidades da aplicação, que são diferentes conforme o tipo de utilizador que a irá usar. Para responder a este requisito foram criados três perfis de utilizadores distintos, cada um com as suas particularidades e funções no que concerne à utilização da aplicação.

### **⇒ Perfis de utilizador:**

#### **✓ *Leitor:***

Um leitor poderá aceder à aplicação após estar registado na mesma, e poderá consultar os seus dados pessoais, verificar o seu histórico de empréstimos e/ou reservas e ainda fazer uma pesquisa mais aprofundada sobre os exemplares existentes na biblioteca.

#### **✓ *Funcionário:***

O funcionário terá um acesso extenso à aplicação após ter um registo na mesma, ou seja, poderá aceder a praticamente todas as funcionalidades com duas excepções, a inserção e a alteração de utilizadores e logins.

#### **✓ *Bibliotecário:***

É o utilizador com mais privilégios na aplicação, podendo aceder a todas as funcionalidades da mesma. É o responsável por inserir/alterar os dados de utilizadores e logins.

Esta diferenciação entre utilizadores permite-nos uma manipulação segura dos dados, visto que nem todos podem adicionar um exemplar ou alterar dados na aplicação, por exemplo.

Sendo assim, e com o perfil adequado a aplicação permite ao utilizador diversas funcionalidades.

**⇒ Funcionalidades da aplicação:****✓ O leitor pode:**

- a. Alterar dados leitor / login: alterar alguns dos seus dados pessoais e de login, nomeadamente a sua palavra passe de entrada no sistema.
- b. Consultar dados pessoais: consultar os seus dados pessoais.
- c. Pesquisar título: fazer uma pesquisa por título a um livro que deseje ter acesso a mais informação.
- d. Consultar histórico de empréstimos: consultar todos os empréstimos que já efectuou desde que se registou na aplicação.
- e. Consultar reservas: consultar as reservas de títulos que já efectuou desde o seu registo na aplicação.

**✓ O Funcionário / Bibliotecário pode:**

- a. Inserir autor: Inserir os dados de autores de determinados títulos.
- b. Alterar autor: alterar os dados de um autor já existente.
- c. Pesquisar autor: pesquisar por um autor através de determinadas palavras-chave.
- d. Eliminar autor: eliminar o registo de um determinado autor.
- e. Inserir tipo de autoria: inserir o tipo de autoria desejado.
- f. Alterar tipo de autoria: alterar os dados de um tipo de autoria.
- g. Eliminar tipo de autoria: eliminar um registo de um tipo de autoria.
- h. Pesquisar tipo de autoria: pesquisar os tipos de autoria disponíveis.
- i. Inserir autoria: inserir autoria.
- j. Alterar autoria: alterar os dados de uma determinada autoria.
- k. Pesquisar autoria: pesquisar as autorias existentes.
- l. Eliminar autoria: eliminar um registo de autoria.
- m. Inserir título: inserir os dados de um determinado título.
- n. Alterar título: alterar os dados de um determinado título.
- o. Pesquisar título: pesquisar os dados de um determinado título através de determinadas palavras-chave.
- p. Eliminar título: eliminar um determinado registo da tabela título.
- q. Inserir exemplar: inserir os dados de um determinado exemplar.
- r. Alterar exemplar: alterar os dados de um determinado exemplar.

- s. Pesquisar exemplar: pesquisar os dados de um determinado exemplar através dos campos disponíveis.
- t. Eliminar exemplar: eliminar um registo de um determinado exemplar.
- u. Pesquisar Utilizador: pesquisar os dados de um determinado utilizador.
- v. Registar empréstimo de exemplar: inserir os dados de empréstimo de um determinado exemplar.
- w. Receber devolução de exemplar: registar dados de devolução de um determinado exemplar.
- x. Pesquisar empréstimos de utilizador: pesquisar a lista de empréstimos de um determinado utilizador.
- y. Registar reserva de exemplar: inserir dados de reserva de um determinado exemplar.

✓ O Bibliotecário pode:

- a. Inserir utilizador / login: inserir os dados de um determinado utilizador de modo a criar um registo na base de dados.
- b. Alterar dados utilizador / login: alterar os dados pessoais / login de um determinado utilizador.
- c. Eliminar utilizador: eliminar um determinado utilizador.

A aplicação é baseada na relação entre as várias tabelas que contém os dados armazenados, e como tal é vital apresentar as tabelas e os seus respectivos campos e determinados tipos de dados de armazenamento.

⇒ **Tabelas e respectivos campos:**

✓ *Tabela autor:*

- a. ID\_AUTOR - Chave primária da tabela autor, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- b. NOME – Nome do autor. Tem uma capacidade máxima de até 80 caracteres.
- c. NACIONALIDADE – Nacionalidade do autor. Tem uma capacidade máxima de até 20 caracteres.



✓ *Tabela autoria:*

- a. ID\_TITULO – Chave estrangeira proveniente da tabela título, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- b. ID\_AUTOR – Chave estrangeira proveniente da tabela autor, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- c. ID\_TIPO\_AUTORIA – Chave estrangeira proveniente da tabela tipo de autoria, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.

✓ *Tabela tipo autoria:*

- a. ID\_TIPO\_AUTORIA – Chave primária da tabela tipo autoria, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- b. DESIGNAÇÃO – Designação da autoria. Tem uma capacidade máxima de até 100 caracteres.

✓ *Tabela Título:*

- a. ID\_TITULO – Chave primária da tabela título, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- b. TITULO – Título do livro. Tem uma capacidade máxima de até 100 caracteres.
- c. ISBN – ISBN do livro, é único e identifica o livro. Tem uma capacidade máxima de até 13 caracteres.
- d. EDITORA – Nome da editora que lançou o livro. Tem uma capacidade máxima de até 100 caracteres.
- e. NUMERO\_DE\_EDICAO – É o número da edição do livro. É um valor numérico com capacidade até 3 números.
- f. LOCAL\_DE\_EDICAO – Localidade onde o livro foi editado. Tem uma capacidade máxima de até 50 caracteres.
- g. DATA\_DE\_EDICAO – Data em que o livro foi editado. Campo do valor data, em que o formato da data deverá ser aaaa.mm.dd.
- h. DIMENSOES – Dimensões físicas do livro. Tem uma capacidade máxima de até 5 caracteres.

- i. NUMERO\_DE\_PAGINAS – Número de páginas que o livro possui. É um valor numérico com capacidade até 4 números.
- j. EXISTENCIA\_DE\_ILUSTRACOES – Campo de valor binário.
- k. EXISTENCIA\_DE\_ESTAMPAS – Campo de valor binário.
- l. EXISTENCIA\_DE\_GRAFICOS – Campo de valor binário.
- m. EXISTENCIA\_DE\_BIBLIOGRAFIA – Campo de valor binário.
- n. EXISTENCIA\_DE\_RESUMO – Campo de valor binário.
- o. RESUMO\_BIBLIOGRÁFICO – Breve resumo do livro. Tem uma capacidade máxima de até 4000 caracteres.

✓ *Tabela exemplar:*

- a. ID\_EXEMPLAR – Chave primária da tabela exemplar, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- b. ID\_TITULO – Chave estrangeira proveniente da tabela título, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- c. COTA – Indica onde o livro está arrumado. Tem uma capacidade máxima de até 10 caracteres.
- d. NUMERO\_INDEXACAO – É um valor numérico.
- e. NUMERO\_REGISTO – É um valor numérico.

✓ *Tabela Reserva:*

- a. ID\_RESERVA – Chave primária da tabela reserva, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- b. ID\_TITULO – Chave estrangeira proveniente da tabela título, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- c. ID\_UTILIZADOR – Chave estrangeira proveniente da tabela utilizador, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- d. DATA\_RESERVA – Data em que a reserva é efectuada, é um valor do tipo data e o formato da data é aaaa.mm.dd.

✓ *Tabela empréstimo:*

- a. ID\_EMPRESTIMO – Chave primária da tabela empréstimo, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.

- b. ID\_EXEMPLAR – Chave estrangeira proveniente da tabela exemplar, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- c. ID\_UTILIZADOR – Chave estrangeira proveniente da tabela utilizador, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- d. DATA\_EMPRESTIMO – Data em que o empréstimo é efectuado. Campo do valor data, em que o formato da data deverá ser aaaa.mm.dd.
- e. DATA\_ENTREGA – Data em que é prevista a entrega do exemplar. Campo do valor data, em que o formato da data deverá ser aaaa.mm.dd.
- f. DATA\_DEVOLUCAO – Data em que o exemplar é efectivamente devolvido. Campo do valor data, em que o formato da data deverá ser aaaa.mm.dd.

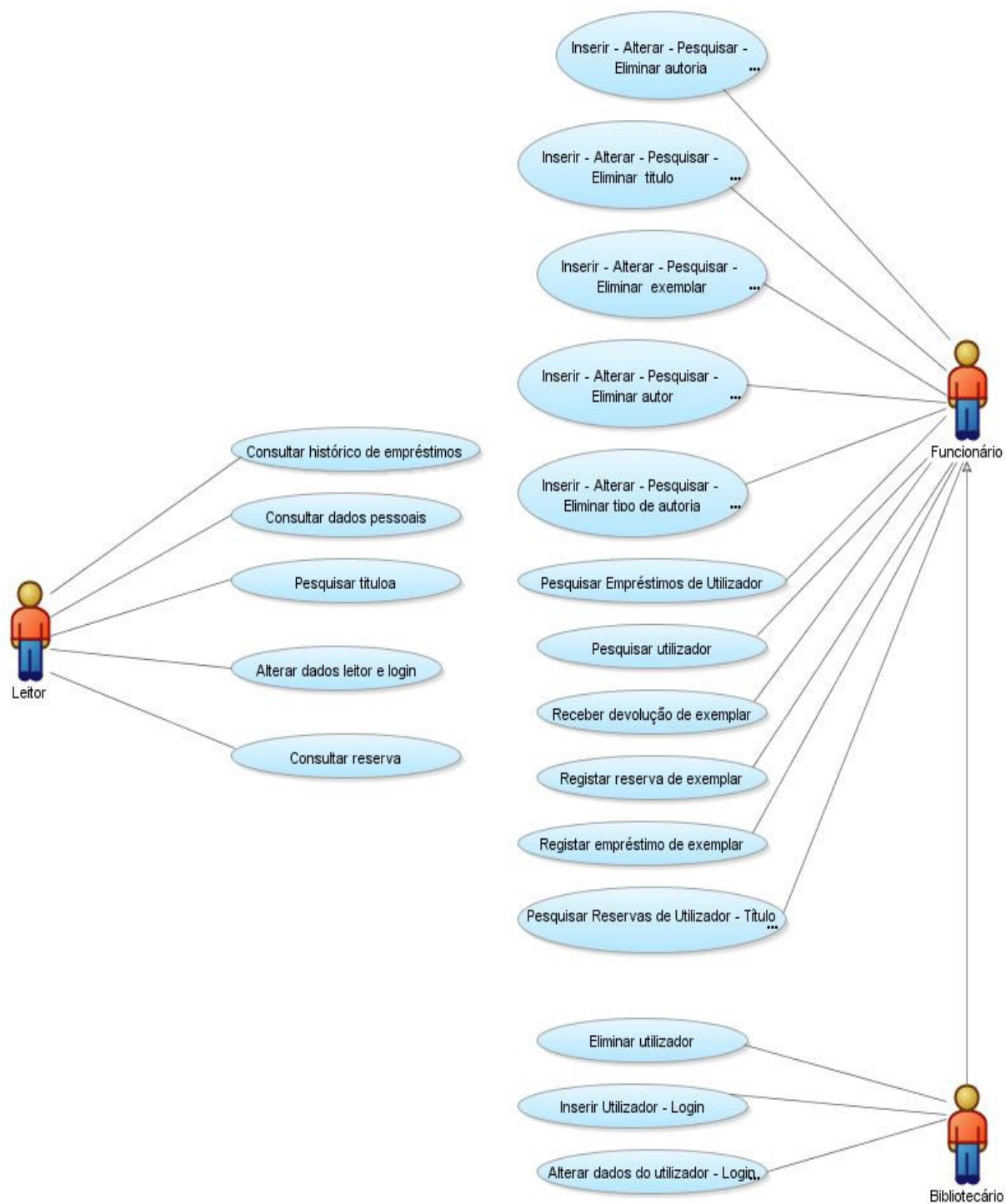
✓ *Tabela Utilizador:*

- a. ID\_UTILIZADOR – Chave primária da tabela utilizador, é de valor numérico e a sua capacidade máxima de preenchimento é de 8 espaços.
- b. PERFIL – Perfil que o utilizador irá ter na aplicação. É escolhido a partir de 3 opções disponíveis no ecrã (Leitor, Funcionário e Bibliotecário).
- c. PASSWORD – Campo de palavra passe do utilizador para entrar no sistema. Tem uma capacidade máxima de até 12 caracteres.
- d. NOME – Nome completo do utilizador. Tem uma capacidade máxima de até 80 caracteres.
- e. NUMERO\_BI – Número do Bilhete de Identidade do utilizador. Tem uma capacidade máxima de até 12 caracteres.
- f. MORADA – Morada do utilizador. Tem uma capacidade máxima de até 100 caracteres.
- g. TELEFONE – Número de contacto do utilizador. Tem uma capacidade máxima de até 12 caracteres.
- h. EMAIL – Endereço de correio electrónico do utilizador. Tem uma capacidade máxima de até 50 caracteres, e vai servir como nome de utilizador para fazer login na aplicação.
- i. NUMERO\_CONTRIBUINTE – Número do cartão de contribuinte do utilizador. Tem uma capacidade máxima de até 12 caracteres.
- j. ESTADO – Estado em que o utilizador se encontra no sistema. Campo com 2 valores à escolha: leitor: activo e inactivo.

## ***Método***

A abordagem ao trabalho foi bastante tradicional, pois começámos por fazer o levantamento de requisitos, que nos permitiu avaliar minimamente a realidade e construir os diagramas UML da nossa aplicação. Os diagramas começaram por ser elaborados no programa Sybase PowerDesigner, e nele foram desenvolvidos os diagramas de estado. Entretanto foi decidido que era melhor utilizar o programa JDeveloper para desenvolver o diagrama de Use Case e o diagrama Entidade Associação, uma vez que este possuía todas as funcionalidades que pretendíamos utilizar e era a mesma ferramenta onde a seguir procederíamos ao desenvolvimento das servlets. Isto permitia-nos economias de escala.

O diagrama de Casos de Uso contribuiu para a compreensão da aplicação, uma vez que mostra de uma forma geral todas as funcionalidades nela presentes e a forma como os actores (os tipos de utilizadores existentes) interagem com o sistema.



*Imagem 1 – Diagrama de Casos de Uso*

Já o diagrama Entidade Associação é um pouco mais aprofundado, uma vez que mostra a forma como as tabelas da base de dados se relacionam entre si.

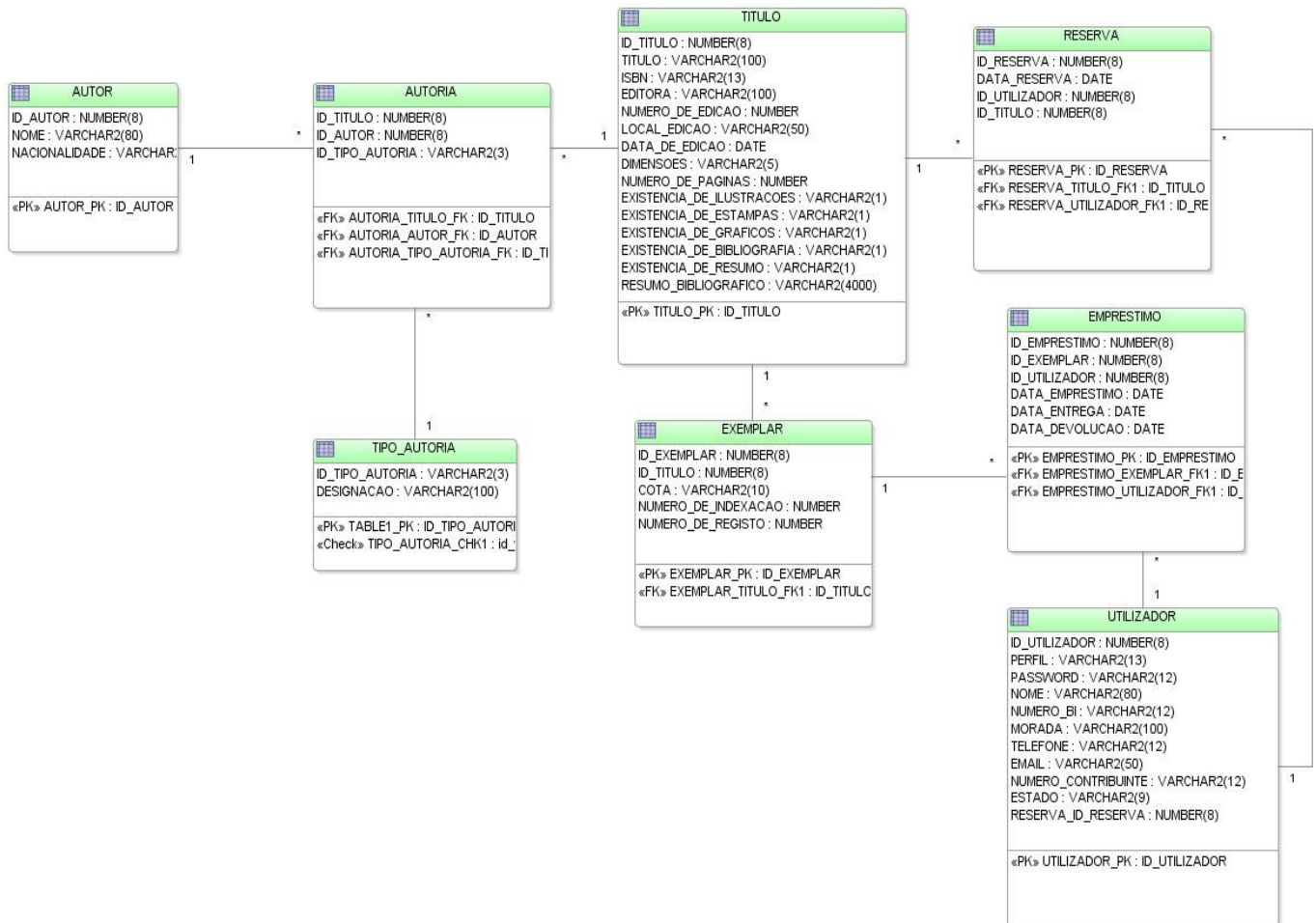


Imagem 2 – Diagrama Entidade Associação

Com os diagramas de estado verificámos o ciclo de vida das diversas entidades, ou seja, os estados por que as entidades passam ao longo de um determinado processo, e isso permite-nos ver a forma como se comportam e evoluem no sistema.

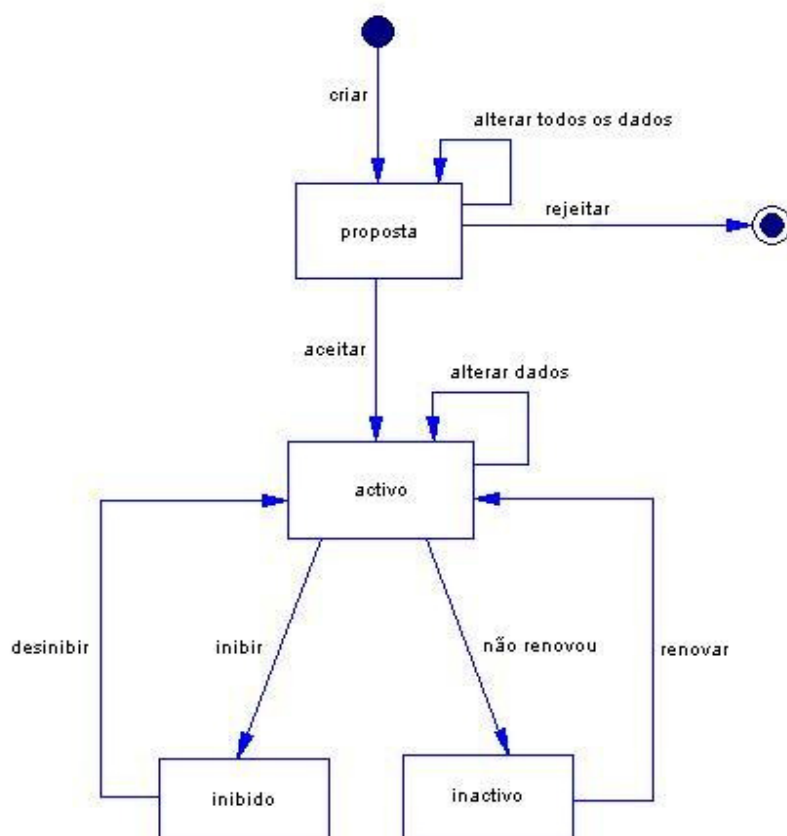


Imagem 3 – Diagrama de estados do utilizador

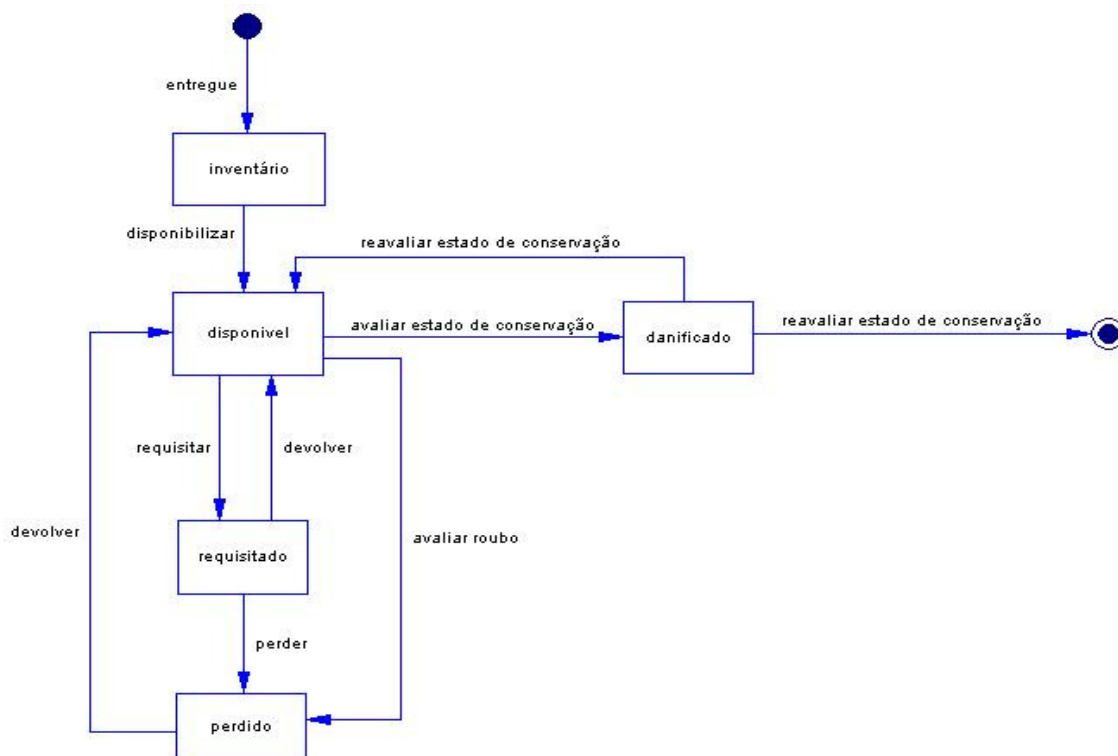
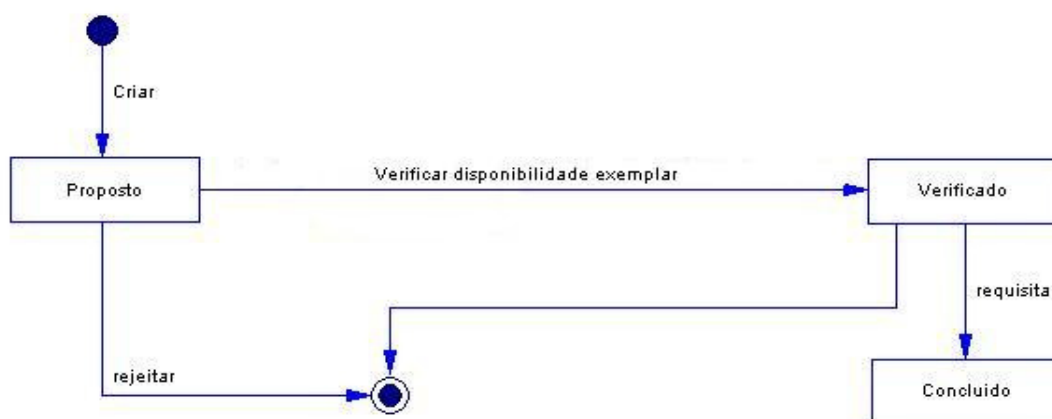


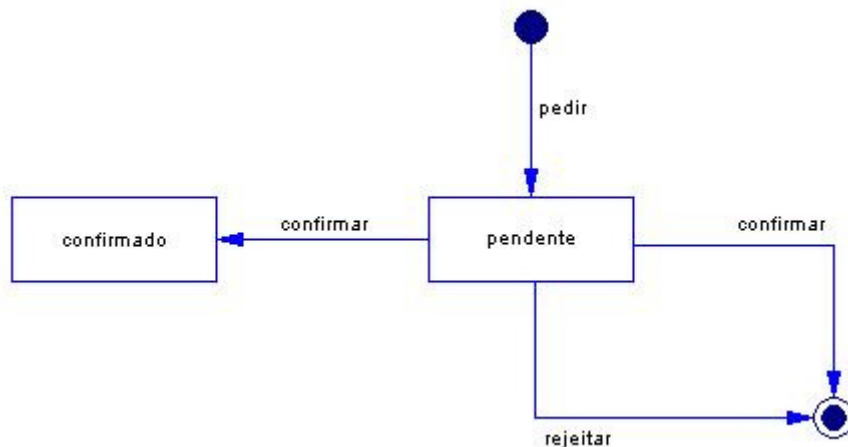
Imagem 4 – Diagrama de estados do exemplar



*Imagem 5 – Diagrama de estados do título*



*Imagem 6 – Diagrama de estados do empréstimo*



*Imagem 7 – Diagrama de estados da reserva*



O passo seguinte foi o de elaboração de uma matriz CRUD, que nos indica onde e em que situações um determinado processo de uma funcionalidade cria, actualiza, lê ou elimina um registo em uma ou mais tabelas.

Entidade	Acção	Autor	Tipo de Autoria	Autoria	Título	Reserva	Exemplar	Utilizador	Empréstimo
Leitor	Alterar dados leitor / Login							U	
	Consultar dados pessoais							R	
	Pesquisar Título				R		R		
	Consultar histórico de empréstimos								R
	Consultar reserva					R			
Funcionário / Bibliotecário	Inserir autor	C						R	
	Alterar autor	UR						R	
	Pesquisar autor	R							
	Eliminar autor	RD							
	Inserir tipo de autoria		C					R	
	Alterar tipo de autoria		UR					R	
	Pesquisar tipo de autoria		R						
	Eliminar tipo de autoria		RD						
	Inserir autoria			C				R	
	Alterar autoria			UR				R	
	Pesquisar autoria			R					
	Eliminar autoria			RD					
	Inserir título				C			R	
	Alterar título				UR			R	
	Pesquisar título	R	R	R	R		R		
	Eliminar título				RD				
	Inserir exemplar						C	R	
	Alterar exemplar						UR	R	
	Eliminar exemplar						RD		
	Pesquisar Reservas de Utilizador / Título				R	R			
	Pesquisar Exemplar						R		
	Pesquisar Empréstimos de Utilizador / Exemplar						R	R	R
	Pesquisar utilizador							R	
	Receber devolução de exemplar						U	R	UR
	Registar reserva de exemplar				R	C		R	
	Registar empréstimo de exemplar				R		RU	R	C
Bibliotecário	Inserir Utilizador / Login							CR	
	Alterar dados do utilizador / Login							UR	
	Eliminar utilizador							RD	

C - Criar / R - Ler / U - Actualizar / D - Eliminar

Imagem 8 – Matriz CRUD

Após esta parte de análise e definição, coube-nos passar para a parte em que pensámos e criámos os ecrãs de interface com o utilizador, usando o JDeveloper e a sua funcionalidade de modelação e escrita de HTML. Desta forma criámos e desenvolvemos o aspecto visual da aplicação e a forma como ela é apresentada ao utilizador, e aqui entrámos também já numa parte de programação em Java, que basicamente foi o passo seguinte a dar, programar as servlets e fazer a ligação à base de dados Oracle.

Com estas ligações criadas e as servlets a funcionarem foi possível então começar a desenvolver o SQL preciso para que a aplicação fosse buscar os elementos necessários à base de dados.

## Resultados

Após o desenvolvimento da aplicação, e apesar de algumas das funcionalidades ainda não estarem a funcionar em pleno, consideramos que o mais importante está terminado, e as seguintes imagens são a demonstração de algumas das funcionalidades da aplicação.



The screenshot shows a login interface with the title 'Login' in large bold text. Below the title, there are two input fields: 'Nome :' containing 'joaolopes@iol.pt' and 'Palavra chave :' containing four dots. At the bottom left of the form are two buttons: 'Limpar' and 'Validar'. In the top right corner, there is a small icon of a bookshelf.

*Imagem 9 – Login na aplicação*

Ecrã de login na aplicação, após preenchimento dos dados a aplicação verifica se são válidos e permite ou não o login.

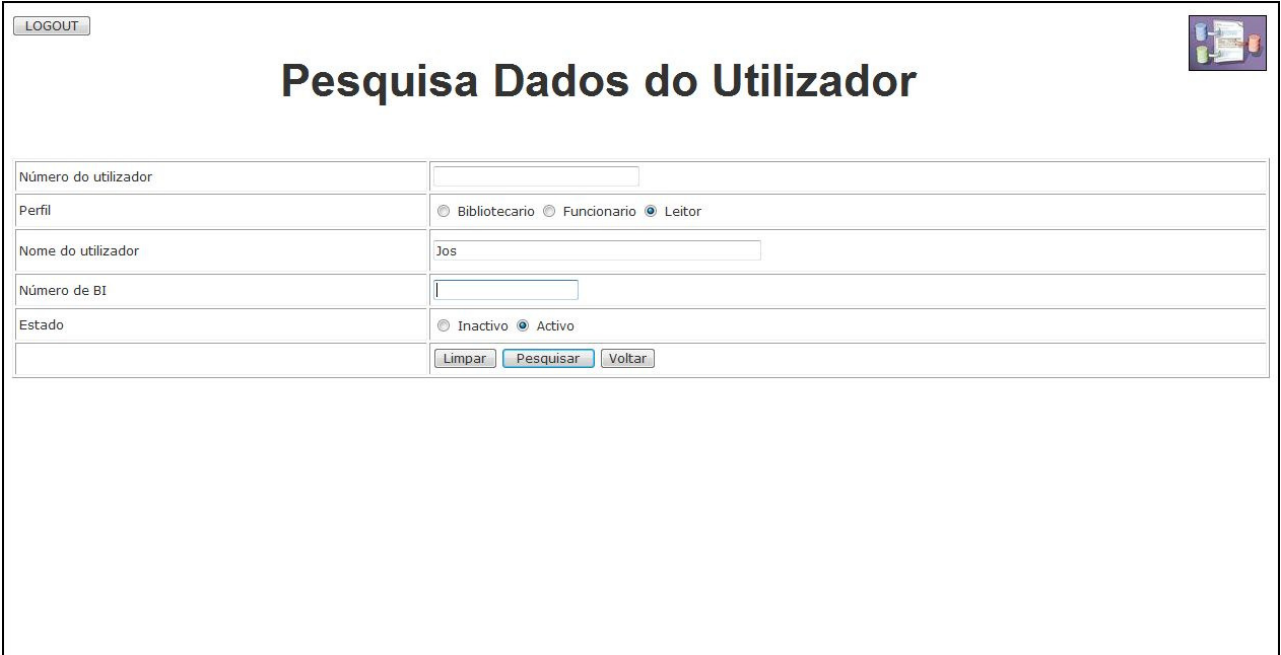


The screenshot shows a user data entry screen. At the top, a red message says 'Utilizador inserido com sucesso.' and there is a 'LOGOUT' button. The title 'Dados do Utilizador' is centered. Below the title is a form with the following fields: 'Número do utilizador' (empty), 'Perfil' (radio buttons for 'Bibliotecario', 'Funcionario', and 'Leitor', with 'Leitor' selected), 'Nome do utilizador' (containing 'José Antunes'), 'Número de BI' (containing '201928374'), 'Número de Contribuinte' (containing '310928374'), 'Morada' (containing 'Avenida dos aliados'), 'Telefone' (containing '213091827'), 'E-Mail' (containing 'jose@iol.pt'), 'Estado' (radio buttons for 'Inactivo' and 'Activo', with 'Activo' selected), and 'Password' (containing eight dots). At the bottom of the form are three buttons: 'Limpar', 'Gravar', and 'Voltar'. In the top right corner, there is a small icon of a bookshelf.

*Imagem 10 – Inserir um utilizador*

Para inserir um utilizador é necessário preencher os dados informativos em relação ao mesmo e pressionar o botão de 'Gravar', ficando o registo armazenado na base de

dados. É conveniente referir que o campo ‘E-Mail’ será usado como nome de utilizador aquando do login na aplicação, como podemos ver na imagem 9.



**Pesquisa Dados do Utilizador**

LOGOUT

Número do utilizador:

Perfil: ☐ Bibliotecario ☐ Funcionario ☒ Leitor

Nome do utilizador:

Número de BI:

Estado: ☐ Inactivo ☒ Activo

*Imagem 11 – Pesquisa de utilizador*

A pesquisa de utilizadores é efectuada através de várias informações da tabela do utilizador, basta preencher o critério pelo qual pretendemos efectuar a pesquisa e pressionar o botão ‘Pesquisar’.



**Listagem de Utilizadores**

Numero De Utilizador	Tipo de Perfil	Nome	Numero BI	Estado	Opções:
46	Leitor	José Antunes	201928374	Activo	<a href="#">Alterar</a> , <a href="#">Remover</a>

*Imagem 12 – Listagem de utilizadores pesquisados*

Como podemos ver, este é o resultado da pesquisa efectuada na imagem 11. É a partir de listagens de pesquisas que temos acesso às funcionalidades de alterar e eliminar da aplicação.

[LOGOUT](#) Utilizador alterado com sucesso.



## Alterar Utilizador

Número do utilizador	46
Perfil	Leitor
Nome do utilizador	José Antunes da Silva
Número de BI	201928375
Número de Contribuinte	310928377
Morada	Avenida dos aliados nº 25
Telefone	213091822
E-Mail	josesilva@iol.pt
Estado	Activo
Password	*****
<input type="button" value="Alterar"/> <input type="button" value="Voltar"/>	

*Imagem 13 – Alterar utilizador*

A informação do utilizador pode também ser alterada conforme o desejado, basta para isso que se altere o campo pretendido.

[LOGOUT](#)




## Remover Utilizador

Número do utilizador	46
Perfil	Leitor
Nome do utilizador	José Antunes da Silva
Número de BI	201928375
Número de Contribuinte	310928377
Morada	Avenida dos aliados nº 25
Telefone	213091822
E-Mail	josesilva@iol.pt
Estado	Activo
<input type="button" value="Eliminar"/> <input type="button" value="Voltar"/>	

*Imagem 14 – Remover utilizador*

O ecrã de remover utilizador não deixa alterar nenhuns dados, e permite-nos apenas confirmar se queremos mesmo eliminar o registo que seleccionámos.

[LOGOUT](#) Utilizador removido com sucesso.



## Home (Bibliotecário)

Inserir	Pesquisar
<input type="button" value="Autor"/>	<input type="button" value="Autor"/>
<input type="button" value="Tipo de Autoria"/>	<input type="button" value="Tipo de Autoria"/>
<input type="button" value="Autoria"/>	<input type="button" value="Autoria"/>
<input type="button" value="Título"/>	<input type="button" value="Título"/>
<input type="button" value="Utilizador"/>	<input type="button" value="Utilizador"/>
<input type="button" value="Exemplar"/>	<input type="button" value="Exemplar"/>
<input type="button" value="Reserva de Exemplar"/>	<input type="button" value="Reservas Utilizador/Título"/>
<input type="button" value="Empréstimo de Exemplar"/>	<input type="button" value="Empréstimos Utilizador/Exemplar"/>
<input type="button" value="Devolução de Exemplar"/>	

*Imagem 15 – Mensagem de utilizador removido*

Ao remover um registo o utilizador é direccionado para a homepage correspondente ao seu perfil (neste caso só funcionários e bibliotecários é que têm acesso a eliminar dados) e recebe uma mensagem no topo superior esquerdo do ecrã com a informação de que o registo escolhido foi de facto eliminado.

## ***Conclusão***

O trabalho demorou 2 anos a fazer, porque no primeiro ano andámos perdidos na análise, com levantamentos de funcionalidades, levantamento de requisitos e alterações no diagrama entidade associação. Durante esse tempo não assentámos em nenhum dos modelos propostos. Em Março de 2008 assentámos num modelo, que assumidamente seria uma simplificação da realidade analisada, mas que a representava minimamente. Após esta decisão o trabalho tornou-se produtivo, pois começamos a lidar com um número mais restrito de tabelas e de funcionalidades.

Depois de ultrapassada esta dificuldade, esbarrámos com o obstáculo seguinte: Servlets e a sua ligação à base de dados. Aqui tivemos a ajuda dos exemplos fornecidos pelo nosso orientador, mas ainda assim tivemos que os compreender e adaptar à nossa situação, o que demorou tempo, principalmente em debugging.

A realização deste trabalho obrigou-nos a rever e reflectir sobre os conhecimentos adquiridos ao longo do curso, o que nos permite agora avaliar as cadeiras com outra perspectiva. A tecnologia utilizada abriu os nossos horizontes, pois são ferramentas líderes de mercado que valorizam as nossas oportunidades de emprego.

Consideramos que as funcionalidades que foram implementadas respondem às necessidades de uma Biblioteca, como foi referido na introdução. Algumas precisam ainda de ser melhoradas, mas consideramos que o fundamental está a funcionar.

## ***Bibliografia***

LORENZO, José Aser – **Aplicação Modelo: Gestão Eventos**. Universidade Lusófona, Outubro 2007.

LORENZO, José Aser ; NUNES, Pedro ; MARTINS, Paulo Jorge – **Programação em Java – Aplicações Web**. Citeforma, Junho 2007.

PEREIRA, Alexandre ; POUPA, Carlos – **Linguagens Web**. Lisboa : Edições Sílabo, 2004. ISBN 972-618-320-0.

HUBBARD, JOHN R. – **Programação em Java**. Lisboa : McGraw-Hill, 2001. ISBN 972-773-141-4.

RODRIGUES, António – **Oracle 10g e 9i**. Lousã : FCA, Agosto 2005. ISBN 972-722-344-3.

PEREIRA, Alexandre ; POUPA, Carlos – **Como escrever uma tese, monografia ou livro científico usando o Word**. Lisboa : Edições Sílabo, 2006. ISBN 972-618-350-2.