

Universidade do Minho  
Licenciatura em Engenharia Informática  
2010



**Trabalho Prático de Base de Dados**

**BibliUM**

Ano Lectivo de 2010/2011

54738 **João Gomes**  
54745 **André Pimenta**  
54802 **Milton Nunes**  
54825 **Daniel Santos**

30 de Janeiro de 2011

## **Resumo**

Neste relatório são apresentados todos os passos da criação de da Base de Dados para o projecto BibliUM.

O projecto BibliUM com é descrito de forma mas especifica no relatório consiste numa base de dados para partilha de documentos por parte dos seus utilizadores.

Para a implementação deste mesmo projecto foram desenvolvidas uma aplicação em JAVA e outra em PHP, sendo que a aplicação de JAVA é responsável pelas tarefas de administração , e a aplicação em PHP por apresentar e disponibilizar todos os recursos ao utilizador.

No relatório são apresentados os principais comandos SQL utilizados pra a criação da base de dados em ORACLE, e o porque das opções tomadas na sua criação. Apresentamos também um manual de utilização para ambas as aplicações desenvolvidas sobre a base de dados.

# Conteúdo

<b>Conteúdo</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>iii</b>
0.1 Motivação e objectivos . . . . .	1
0.1.1 BibliUM . . . . .	1
0.2 Estrutura . . . . .	1
0.3 Introdução . . . . .	1
<b>1 Base de Dados e Normalização</b>	<b>2</b>
1.1 Restrições de integridade estrutural e referencial . . . . .	3
1.2 Tabelas SQL . . . . .	3
1.3 Sequences . . . . .	5
1.4 Views . . . . .	6
1.5 PL/SQL . . . . .	7
1.5.1 Functions . . . . .	7
1.5.2 Procedures . . . . .	7
1.5.3 Triggers . . . . .	7
<b>2 Desenvolvimento das aplicações</b>	<b>9</b>
2.1 Aplicação em JAVA . . . . .	9
2.1.1 Inserção de dados através do JDBC . . . . .	9
2.1.2 Querys efectuadas da aplicação JAVA . . . . .	10
2.1.3 Updates e deletes . . . . .	10
2.2 Aplicação em PHP . . . . .	10
2.2.1 Acesso á Base de Dados . . . . .	11
2.2.2 Principais querys . . . . .	11
2.2.3 Inserções de dados . . . . .	11
2.3 Ferramentas utilizadas . . . . .	12
<b>3 Manuais de Utilização</b>	<b>13</b>
3.1 Pagina Web . . . . .	13
3.1.1 Login . . . . .	13
3.1.2 Registar . . . . .	14
3.1.3 Pagina principal . . . . .	14
3.1.4 Upload de Documentos . . . . .	15
3.1.5 Conta do utilizadro . . . . .	16
3.1.6 Biblioteca do utilizador . . . . .	17

3.1.7	Pesquisa . . . . .	17
3.1.8	Pesquisa avançada . . . . .	18
3.1.9	Visualização de documentos . . . . .	19
3.2	Aplicação em JAVA . . . . .	19
<b>4</b>	<b>Conclusão</b>	<b>24</b>
<b>5</b>	<b>Bibliografia</b>	<b>25</b>
<b>6</b>	<b>Elementos do Grupo</b>	<b>26</b>

# Lista de Figuras

1.1	Modelo conceptual . . . . .	2
3.1	Entrada pagina Web . . . . .	13
3.2	Registar utilizador . . . . .	14
3.3	Pagina principal . . . . .	15
3.4	Upload de documentos . . . . .	16
3.5	Minha conta . . . . .	16
3.6	Minha biblioteca . . . . .	17
3.7	Pesquisa de documentos . . . . .	17
3.8	Pesquisa avançada . . . . .	18
3.9	Visulização de documentos . . . . .	19
3.10	Login . . . . .	19
3.11	Home . . . . .	20
3.12	Utilizadores . . . . .	21
3.13	Documentos . . . . .	21
3.14	Pais . . . . .	22
3.15	Autor . . . . .	22
3.16	Especificidade . . . . .	22
3.17	Universidade . . . . .	23
3.18	Grafico . . . . .	23
6.1	João Miguel . . . . .	26
6.2	André Pimenta . . . . .	26
6.3	Daniel Santos . . . . .	26
6.4	Milton Nunes . . . . .	26

## 0.1 Motivação e objectivos

Este projecto teve como principal motivação e objectivo o uso dos conhecimentos adquiridos ao longo do semestre na cadeira de Base de Dados, consolidando estes.

Foi nos permitido escolher um tema para o desenvolvimento de uma base de dados para o trabalho pratico. Com tal liberdade optamos pela escolha de algo que possa ser nos util e que esteja proximos do nosso ambiente de trabalho, desta forma o tema que optamos por escolher foi um sistema de partilha de documentos de caracter educativo, uma especie de biblioteca online.

Podemos então afirmar que a elaboração deste projecto foi bastante motivante para o grupo, inicialmente porque tivemos a hipotese de trabalhar num tema escolhido por nós, mas também porque nos permitio cimentar e melhorar o conhecimentos sobre base de dados.

### 0.1.1 BibliUM

O projecto escolhido por nós foi um sistema de gestão de partilha de documentos online, uma especie de biblioteca online.

O principal objectivo deste projecto a que chamamos de BibliUM é oferecer uma partilha de documentos aos utilizadores deste projecto. Consiste na criação de uma biblioteca universal onde todos nós podemos participar e contribuir, e ao mesmo tempo usufruir de material educativo que necessitemos.

BibliUm terá então como principal objecivo funcionar como uma biblioteca de partilha de documentos online, em que os contribuintes para esta biblioteca são os proprios utilizadores.

## 0.2 Estrutura

O presente relatório é constituído por 3 capítulos: **Capítulo I** é apresentada a forma como foi desenvolvida a base de dados, o **Capítulo II** onde são apresentadas as aplicações que irão utilizar a base de dados e por ultimo o **Capítulo III** onde se encontram os manuais de utilização das aplicações.

## 0.3 Introdução

Base de Dados é umas das areas mais importantes da toda a informatica. Cada vez mais se torna importante armazenar os dados e ter registos de certas acções que efectuamos, mas não só. Com uso de base de dados garantimos o acesso a determinados dados sempre que pertendeermos e assim tornamos a informação persistente, e está sim é uma das principais utilidades.

Apesar da grande utilidade que as base de dados nos oferecem estas necessitam de "interfaces" a funcionar sobre elas para que as suas potencialidades sejam exploradas.

Para o projecto BibliUM era imposto o uso de duas interfaces para a base de dados, uma como pagina web e outra como aplicação funcional, sendo que cada uma delas devem se ligar a mesma base de dados logo terão de estar em sincronia com os dados da mesma.

Com estas condições apresentadas anteriormente iremos apresentar em seguida todos os principais passos da criação da base de dados.

# Capítulo 1

## Base de Dados e Normalização

A base de dado foi criada usando o motor ORACLE. Este motor foi o que trabalhamos ao longo das aulas, logo já estávamos familiarizados com este, o que facilitou o criação da base de dados.

Sendo que o projecto BibliUM se destina a partilha de documentos por utilizadores, podes concluir que utilizadores e documentos irão ser as principais entidades da nossa base de dados.

Apesar de o sistema BibliUM funcionar em função dos seus utilizadores e dos documentos, esta informação só não chega. Daqui partimos para as restantes informações necessárias para o bom funcionamento da base de dados.

Após decidirmos quais os campos que eram relevantes para estas duas entidades iniciamos o processo de normalização e desenvolvemos as tabelas que se apresentam na terceira forma normal.

Apresentamos na figura a baixo o modelo conceptual da base de dados.

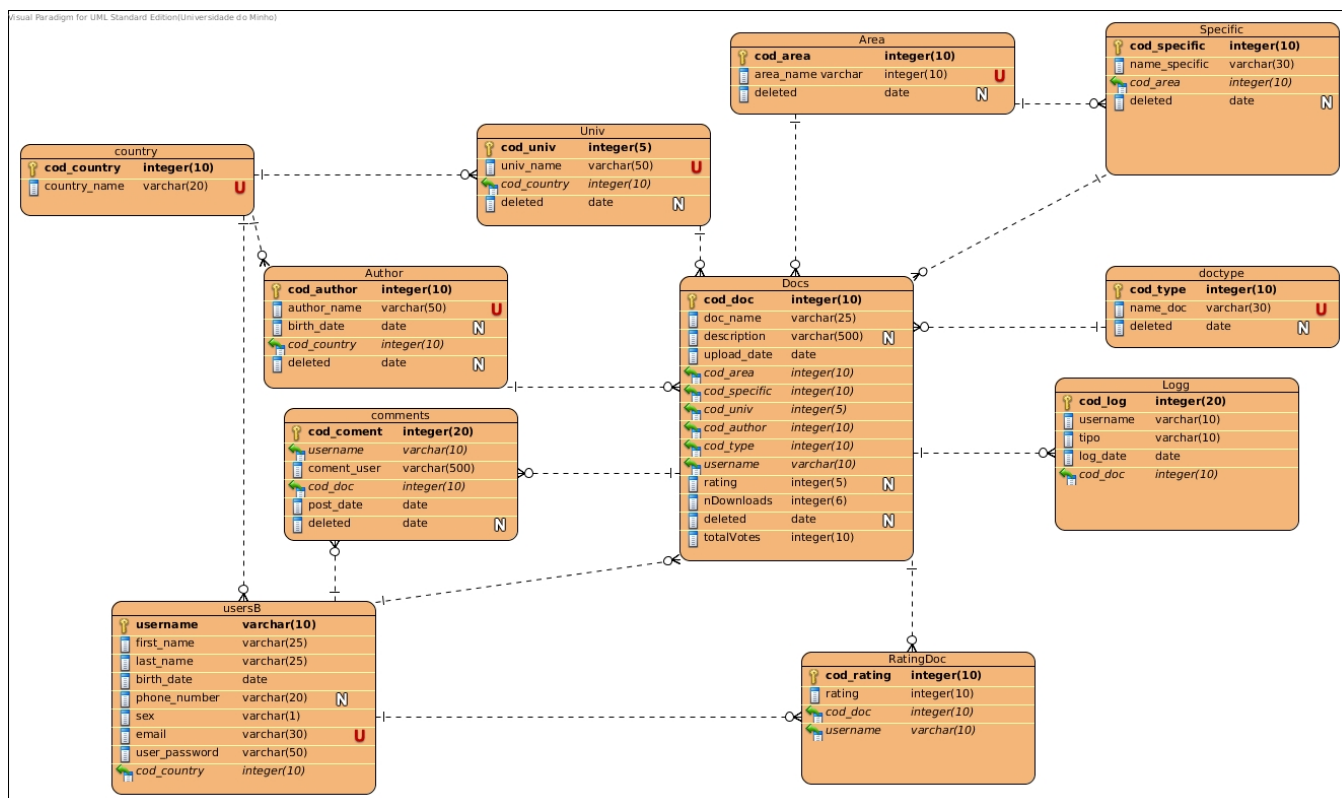


Figura 1.1: Modelo conceptual

## 1.1 Restrições de integridade estrutural e referencial

Como podemos ver no modelo conceptual apresentado anteriormente, estão identificadas as diferentes chaves primarias e chaves secundarias. Sendo que em quase todas as tabelas as chaves primarias são os codigos de identificação, excepção para a tabela userB em que a chave primaria é o username.

Podemos ver também identificadas as restrições 'unique' identificadas por um U no modelo, estas restrições são aplicadas a campos de tabelas que pretendemos que sejam unicos, um excelente exemplo é o campo email do utilizador. Este campo terá de ser unico, pois é a identificação de registo do utilizador, se um mail já se encontrar registado por um utilizador, então mais ninguém se poderá registar com este mesmo mail. Com este campo existem mais que como já foi referido estão identificados com a letra 'U' no modelo conceptual apresentado a cima.

Uma outra restrições importante é a restrição **not null**, esta restrição foi usada em todos os campos que pretendemos que nunca sejam nulos, que tenham de assumir obrigatoriamente um valor. No modelo conceptual apresentado assim os campos das tabelas que estão sujeitos a estas restrição não apresenta a letra 'N'. Os campos que apresentarem a letra 'N' significa que poderá ser nulos.

Estes são os campos que não têm um valor fundamental para o projecto (ex:telefone) , mas que no entanto não deixa de ser necessarios e de ter a sua importnacia.

Todas estas restrições têm e são respeitadas no projecto, pois são fundamentais para o funcionamento da base de dados.

## 1.2 Tabelas SQL

Os comandos SQL para a criação das tabelas são os seguintes:

Area representa categoria dos documentos (ex:Informatica,Biologia,etc).

```
CREATE TABLE area
(cod_ area number(10) PRIMARY KEY,
area_ name varchar(30) not null unique
deleted date
);
```

Specific representa as subcategorias dos documentos.

```
CREATE TABLE specific (
cod_ specific number(10) PRIMARY KEY,
name_ specific varchar(30) not null,
cod_ area number(10),
foreign key (cod_ area) references area(cod_ area),
deleted date
);
```

Doctype representa o tipo de documento guardado(ex:Relatorio, Livro,etc).

```
CREATE TABLE doctype
(cod_ type number(10) PRIMARY KEY,
name_ doc varchar(30) not null unique,
deleted date
);
```



Country contém os países.

```
CREATE TABLE country
(cod_country number(10) PRIMARY KEY,
country_name varchar(20) not null unique
);
```

A tabela author irá guarda os autores dos documentos.

```
CREATE TABLE author
(cod_author number(10) PRIMARY KEY ,
author_name varchar(50) not null unique,
birth_date date,
cod_country number(10) references country,
deleted date
);
```

A tabela Univ contém as universidade/instituições a que possivelmente possam pertencer os documentos.

```
CREATE TABLE univ
(cod_univ number(5) PRIMARY KEY,
univ_name varchar(50) not null,
cod_country number(10) references country,
deleted date
);
```

A tabela userB irá guardar toda a informação de registos dos utilizadores.

```
CREATE TABLE usersB
(username varchar(10) PRIMARY KEY,
first_name varchar(25) not null,
last_name varchar(25) not null,
birth_date date not null,
phone_number varchar(20),
sex varchar(1) not null,
email varchar(30) not null unique,
user_password varchar(50) not null,
cod_country number(10) references country
);
```

A tabela docs contém a informação de todos os docuemntos.

```
CREATE TABLE docs
(cod_doc number(10) PRIMARY KEY,
doc_name varchar(25) not null,
description varchar(500),
upload_date date not null,
cod_area number(10) references area not null,
cod_specific number(10) references specific,
cod_univ number(5) references univ,
```

```

cod_ author number(10) references author,
cod_ type number(10) references docType not null,
username varchar(10) references usersB,
rating number(5,2),
nDownloads number(6) not null,
deleted date
totalVotes number(10) not null
);

```

A tabela `logg` é responsável por guardar toda a actividade dos utilizadores na base de dados, caso um utilizador entre no BibliUM então será guardado o seu registo de entrada com a letra 'l', caso este faça upload de um documento com a letra 'l' e por fim caso este faça download será guardado com a letra 'd'.

```

CREATE TABLE logg
(cod_ log number(20) PRIMARY KEY,
username varchar(10) references usersB not null,
log_ date date not null,
tipo varchar(1) not null,
cod_ doc number(10) references docs
);

```

A tabela `ratingDoc` vai registar todas as avaliações feitas pelos utilizadores aos documentos.

```

create table ratingDoc (
cod_ rating number(10) primary key,
rating number(10) not null,
cod_ doc number(10) not null,
username varchar(10) not null,
foreign key (cod_ doc) references docs,
foreign key (username) references usersB);

```

A tabela `comments` irá guardar os comentarios feitos pelos diferentes utilizadores a um documento.

```

CREATE TABLE comments
(cod_ coment number(20) PRIMARY KEY,
username varchar(10) references usersB not null,
comment_ user varchar(500) not null,
cod_ doc number(10) references docs not null,
post_ date date not null
deleted date;
);

```

### 1.3 Sequences

O uso de Sequences na base de dados é muito importante. É através de sequences que iremos controlar/gerar ids automaticos.

No caso específico do nosso trabalho as sequences serão usadas para gerar automaticamente todas os co-

digos que representam chaves primarias de algumas tabelas automaticamente. Desta forma não teremos de nos preocupar com a actualização dos codigos pois estes serão gerados automaticamente. Apresentamos as Sequences usadas na nossa base de dados a seguir.

```
create sequence new_ area start with 1 increment by 1 cache 10;
create sequence new_ comments start with 1 increment by 1 cache 10;
create sequence new_ author start with 1 increment by 1 cache 10;
create sequence new_ country start with 1 increment by 1 cache 10;
create sequence new_ docs start with 1 increment by 1 cache 10;
create sequence new_ doctype start with 1 increment by 1 cache 10;
create sequence new_ logg start with 1 increment by 1 cache 10;
create sequence new_ specific start with 1 increment by 1 cache 10;
create sequence new_ univ start with 1 increment by 1 cache 10;
create sequence new_ rating start with 1 increment by 1 cache 10;
```

## 1.4 Views

As views podem ser vistas como tabelas virtuais ou uma especie de consulta armazenada, simplificando de certa forma a consualta de determinados campos.

O uso destas torna-se vantajoso pela sua simplicidade, segurança e manutenção. Com uma view conseguimos então organizar de forma mais eficaz e segura o acesso a determinados dados.

Na nossa base de dados foram criadas as seguintes views:

```
create or replace view documents as
select cod_ doc,doc_ name,description,rating,ndownloads ,upload_ date, usersb.first_ name, usersb.last_
name,
doctype.name_ doc,author_ name,area_ name,specific.name_ specific
, univ.univ_ name
from docs
join area using(cod_ area)
join doctype using(cod_ type)
join usersB using(username)
LEFT OUTER join specific using(cod_ specific)
LEFT OUTER join author using(cod_ author)
LEFT OUTER join univ using(cod_ univ)
;
```

A view documents têm extrema importancia na nossa base de dados pois permite-nos ter acesso a todos os dados necessarios relativos aos documentos.

```
create view mUtilizados as select doc_ name, nDownloads from docs order by nDownloads create
view mFicheiros as select username, count(*) from docs group by username order by count(*) create
view mDownloads as select username, count(*) from logg where tipo = 'd' group by username order by
count(*)
create or replace view authors as
select author_ name,birth_ date,country_ name from author
LEFT OUTER join country using(cod_ country);
```

A view authors permite nos ver os dados dos autores em apenas uma tabela (tabela virtual), sendo que temos logo acesso a todos os dados do autor do autor sem ter de aceder a varias tabelas.

## 1.5 PL/SQL

A utilização de PL/SQL é extremamente importante e como tal, também foi imposta a sua utilização no desenvolvimento da base de dados. Por estes motivos iremos apresentar alguns triggers, procedures e functions desenvolvidas na nossa base de dados.

### 1.5.1 Functions

Usamos funções com auxilires de codigo para determinadas acções.

As functions por nós desenvolvidas são apresentadas a seguir.

```
create or replace
function getMedia(cod_ d in varchar, rate in number) return number as
begin
declare media number;
votes number;
begin
media := 0;
select rating into media from docs where cod_ d = cod_ doc;
select totalVotes into votes from docs where cod_ d = cod_ doc;
return (media + rate) / votes;
end;
end getmedia;
```

### 1.5.2 Procedures

Tal como nas functions o uso de procedimentos vêm ajudar na melhor utilização e gestão da base de dados.

```
CREATE OR REPLACE PROCEDURE apagaSpec(" + data + "date, " + codigo + "number) BEGIN
UPDATE specific SET deleted = data WHERE cod_ area = codigo END apagaSpec;"
```

### 1.5.3 Triggers

O uso de triggers é muito importante, através deste conseguimos prever a corrupção através de dados mal formados, conseguimos gerar acções sobre a base de dados de forma automatica e simples.

Foi desta forma que criamos os seguintes Triggers para nos ajudar na gestão e controlo da base de dados.

```
create or replace
trigger duplicatedVote before
insert on ranting referencing new as newrow for each row
declare existe number ;
cod_ rate number;
oldrate number;
media number;
begin select max(cod_ rating) into cod_ rate from ratingdoc
where :newrow.cod_ doc=cod_ doc, :newrow.username, oldrate, :newrow.ranting);
if (cod_ rate is not null)
then select rating into oldrate from ratingdoc where cod_ rating=cod_ rate;
actualizaRateDup( :newrow.cod_ doc, :newrow.username, oldrate, :newrow.rating);
else update docs set totalVotes = totalVotes +1 where cod_ doc = :newrow.cod_ doc;
media:= getmedia(:newrow.cod_ doc, :newrow.cod_ doc;
end if;
end actRating;
```

```
create or replace
trigger actNDownloads after
insert on logg referencing new as newrow for each row
begin
if (:newrow.tipo='d') then
update docs set ndownloads = ndownloads + 1 where cod_ doc= :newrow.cod_ doc;
end if;
end actNDownloads;
create or replace
trigger actRating after
insert on ratingdoc referencing new as newrow for each row
declare media number;
begin
update docs set totalVotes = totalvotes + 1 where cod_ doc = :newrow.cod_ doc;
media := getmedia(:newrow.cod_ doc, :newrow.rating);
update docs set rating = media where cod_ doc = :newrow.cod_ doc;
end actRating;
```

## Capítulo 2

# Desenvolvimento das aplicações

### 2.1 Aplicação em JAVA

Para além da plataforma para o utilizador comum da nossa BibliUM, criamos também uma aplicação de gestão da mesma. Através da linguagem JAVA criamos um programa que permite a gestão total da base de dados do nosso projecto. Esta aplicação em JAVA permite todo o tipo de operações que um administrador da BibliUM necessita para fazer a gestão adequada dum projecto como este.

A aplicação permite a ligação a mais do que uma base de dados, sendo dada ao administrador a possibilidade de no arranque escolher qual a base de dados a que se pretende ligar. Oferece também a possibilidade de pesquisa (com a opção de filtrar informação caso assim o pretenda), eliminação, alteração e adição de dados à base de dados.

Para além destas funções básicas, o administrador pode consultar ainda alguns dados estatísticos, como o total de registos em cada tabela e analisar alguns gráficos gerados com base nesses dados retirados da consulta à base de dados.

A aplicação, como já foi referido, foi desenvolvida em JAVA, através da API Swing, e oferece um design simples e de fácil utilização por parte do administrador.

#### 2.1.1 Inserção de dados através do JDBC

Apresentamos um exemplo de inserções que foram feitas na aplicação em JAVA.

```
public void inserirUser(String user, String f_name, String L_name, String date, String sex, String mail,
String pass, String cod_country, String cod_doc) throws Exception
String sql = "INSERT INTO usersB VALUES ('"+user+"', '"+f_name+"', '"+L_name+"', '"+date+"',
 '"+sex+"', '"+mail+"', '"+pass+"', '"+cod_country+"', '"+cod_doc+"')";
PreparedStatement stmt = con.prepareStatement(sql);
stmt.executeQuery();
this.setChanged();
this.notifyObservers();
```

### 2.1.2 Querys efectuadas da aplicação JAVA

Na aplicação são efectuadas varias query's apresentamos o uso de uma em java.

```
public ResultSet getMedia() throws Exception
String sql = "create view media as select d.doc_ name, avg(ratd.rating) from docs d, ratingDoc ratd where
ratd.cod_ doc = d.cod_ doc group by d.doc_ name order by avg(ratd.rating)";
PreparedStatement stmt =con.prepareStatement(sql);
ResultSet resultSet = stmt.executeQuery();
return resultSet;
```

### 2.1.3 Updates e deletes

É muito importante não apagar os dados da base de dados por algumas razões, tentar recuperar dados, exemplos de historico entre outros.

Por estes motivos sempre que algo se pretenda apagar, vai ser ocultado. Vai ser o campo 'deleted date' que inicialmente se vai encontrar desactivado e caso seja e se pretende apagar algo, activa-se este campo, ficando na mesma na base de dados, mas no entanto não é apresentado aos utilizadores, caso se pretenda recuperar os dados é só voltar a colocar a null a data em que foi desactivado o objecto.

É possível e como tarefa de administração a edição de determinados dados, apresentamos um exemplo em baixo de updates realizados através do JDBC.

```
public void updateAuthor(String cod_ author, String fName, String data, String cod_ count) throws Ex-
ception
String sql = "UPDATE AUTHOR SET first_ name = '"+ fName + "', birth_ date = TO_ DATE('"+
data + "', 'RR.MM.DD'), cod_ country = '"+ cod_ count + "' WHERE cod_ author = '"+ cod_ author
+ "'";
PreparedStatement stmt =con.prepareStatement(sql);
stmt.executeQuery();
this.setChanged();
this.notifyObservers();
```

## 2.2 Aplicação em PHP

A aplicação em PHP é destinada aos utilizadores, todas as funções/necessidades que devem ser oferecidas a estes estão expostas nesta aplicação.

Desde já podemos dizer que a escolha sobre PHP para o desenvolvimento da aplicação web teve em conta a grande disponibilidade de recursos oferecidos por esta plataforma, mas também pela importância que têm hoje em dia o uso de uma linguagem web dinâmica.

Alguns dos recursos que decidimos usar foi o uso da framework PEAR.

Um dos principais recursos foi o pacote "HTML/QuickForm.php" que permite a criação dinâmica de formas. Formas estas que são criadas com a possibilidade de implementação de "regras". Vão ser estas regras que vão controlar todo o input da aplicação WEB, de forma a garantir que apenas serão passados dados correctos para a base de dados.

Outro pacote importante foi o "HTTP/Upload.php", que ajuda a gerir os uploads de documentos de forma eficaz.

Estes são alguns dos exemplos da utilidade da framework PEAR.  
 PHP foi usado juntamente com o oci8 para o acesso á base de dados.  
 Apresentamos em seguida algum código desenvolvido na aplicação PHP.

### 2.2.1 Acesso á Base de Dados

O acesso á base de dados é bastante simples. Usando a biblioteca oci8 usamos as seguintes linhas de código:

```
<?php
$conn = oci_connect('pimz', 'pimz53', 'localhost/XE');
if (!$conn)
    $e = oci_error();
trigger_error(htmlentities($e['message'], ENT_QUOTES), E_USER_ERROR);

?>
```

### 2.2.2 Principais queries

O projecto BibliUM contém varias queries, no entanto deixamos apenas aqui algumas como referencia do seu funcionamento e apenas como referencia.

```
$sqlID="SELECT * FROM documents ORDER BY upload_date DESC";
SELECT * FROM documents ORDER BY rating DESC"
"SELECT doc_name,cod_doc,upload_date FROM docs WHERE username=:b1 AND cod_type=:b2
AND deleted is null ORDER BY $ordem
SELECT * FROM authors WHERE author_name=:b1
"SELECT username,comment_user,post_date FROM comments where cod_doc=".$ _GET['id']."order
by post_date DESC"
SELECT count(*) FROM documents where doc_name LIKE :b1 OR description LIKE :b1
```

### 2.2.3 Inserções de dados

Na pagina web apenas são permitidas três inserções por parte dos utilizadores, registar um autor, fazer upload e consecutivamente registar o documento na base de dados e como é obvio o seu proprio registo de utilizador.

Apresentamos então essas inserções nas linhas que se seguem.

```
$s = oci_parse($conn,"insert into author values (new_author.NEXTVAL,:b1,TO_DATE(:b2,'YYYY/MM/DD'),:b3,
returning cod_author into :b4");
oci_bind_by_name($s,":b1",($form->exportValue('author')));
oci_bind_by_name($s,":b2",dataS($form->exportValue('date')));
oci_bind_by_name($s,":b3",$cods_country[($form->exportValue('country'))]);
oci_bind_by_name($s,":b4",$cod_author);
oci_execute($s, OCI_DEFAULT);
oci_commit($conn);
oci_free_statement($s);
```



```

$s = oci_parse($ conn,"insert into docs values(new_docs.NEXTVAL,:b1,:b2,TO_DATE(:data,'YYYY/MM/DD'),:cod_ doc into :b9");
oci_bind_by_name($ s, ":b1",($ form->exportValue('name')));
oci_bind_by_name($ s, ":b2",($ form->exportValue('description')));
oci_bind_by_name($ s, ":b3",$ cods_ area[$ a]);
oci_bind_by_name($ s, ":b4",($ cods_ specific[$ a]/$ e));
oci_bind_by_name($ s, ":b5",($ cods_ univ[$ form->exportValue('univ')]));
oci_bind_by_name($ s, ":b6",$ cod_ author);
oci_bind_by_name($ s, ":b7",($ cods_ type[$ form->exportValue('doc_ type')]));
oci_bind_by_name($ s, ":b8",$ _ SESSION['nome']);
oci_bind_by_name($ s, ":b9",$ id);
oci_bind_by_name($ s, ":data",date("Y/m/d"));
oci_execute($ s, OCI_DEFAULT);
oci_commit($ conn);
oci_free_statement($ s);

```

```

$s = oci_parse($ conn,"INSERT into usersB values (:b1,:b2,:b3 ,TO_DATE(:b4,'YYYY/MM/DD')
,:b5,:b6,:b7,:b8,:b9)");
oci_bind_by_name($ s, ":b1",($ form->exportValue('username')));
oci_bind_by_name($ s, ":b2",($ form->exportValue('firstname')));
oci_bind_by_name($ s, ":b3",($ form->exportValue('lastname')));
oci_bind_by_name($ s, ":b4",dataS($ form->exportValue('date')));
oci_bind_by_name($ s, ":b5",($ form->exportValue('phoneN')));
oci_bind_by_name($ s,":b6",($ form->exportValue('sex')));
oci_bind_by_name($ s,":b7",($ form->exportValue('email')));
oci_bind_by_name($ s, ":b8",md5($ form->exportValue('password')));
oci_bind_by_name($ s,":b9",$ cods_ country[( $ form->exportValue('country'))]);
oci_execute($ s, OCI_DEFAULT);
oci_commit($ conn);

```

## 2.3 Ferramentas utilizadas

As ferramentas utilizadas pelo grupo foram as seguintes:

**ORACLE** - Motor da base de dados;

**ORACLE SQL developer** - Utilizado para o desenvolvimento do código SQL;

**NetBeans** - IDE de código aberto utilizado para desenvolver a aplicação em JAVA;

**Geany** - Desenvolvimento do código PHP e relatório.

## Capítulo 3

# Manuais de Utilização

### 3.1 Pagina Web

A aplicação web no nosso projecto torna-se um instrumento fundamental, pois é através desta que os utilizadores poderão usar o BibliUM, esta foi criada usando PHP como plataforma de desenvolvimento. Estando a aplicação web destinada ao utilizadores todas as funcionalidades possíveis serão apresentadas nestas, para o usufruimento dos utilizadores, estando a administração do **BibliUM** restringida á aplicação desenvolvida em JAVA.

Desta forma iremos apresentar as principais funcionalidades apresentadas aos utilizadores.

#### 3.1.1 Login

A aplicação web apenas poderá ser usada mediante um pré-registo que posteriormente irá dar acesso ás funcionalidades desta. O login é efectuado através do "username" escolhido no registo do utilizador que terá de combinar posteriormente com a palavra pass escolhida pelo mesmo.



Figura 3.1: Entrada pagina Web

### 3.1.2 Registar

Como já referimos para o uso do **BibliUM** é necessário um registo.

Este irá conter os dados pessoais de cada utilizador, juntamente com a sua combinação de acesso à aplicação web. Existem campos que são obrigatórios o seu preenchimento e outros que são opção do utilizadore regista-los.

Os campos necessários para o registo de um utilizador são expostos na imagem que se segue.

The screenshot shows a web browser window with the address bar displaying 'localhost/register\_user.php'. The page title is 'BibliUM' and the main heading is 'Criar conta'. The registration form includes the following fields and options:

- \*username:
- \*Primeiro nome:
- \*Ultimo nome:
- \*Password:
- Confirmar password:
- \*Email:
- Numero de telefone:
- Sexo: ☒ Masculino ☐ Feminino
- Localidade:  Escolha um pais
- Data de nascimento:  01  Jan  1900

Buttons:

\* Campos obrigatórios

At the bottom of the page, there are links for 'Autores' and 'Contactar administração'.

Figura 3.2: Registar utilizador

### 3.1.3 Pagina principal

A "home page" está destinada a apresentar as principais funcionalidades e objetivos do projeto.

Como já referimos o principal objetivo do **BibliUM** é disponibilizar recursos aos utilizadores. Para tornar este acesso mais eficaz e organizado é apresentado na "home page" do lado esquerdo a lista de categorias dos documentos que se apresentam disponíveis ao utilizador, e do lado direito a lista de formatos dos recursos, estes dois campos poderão ser cruzados entre si para que seja possível ver os recursos "mais populares", "os mais recentes" e os que possuem "melhor classificados pelos utilizadores" de cada categoria e formato.

Na página principal também podemos realizar pesquisas de documentos, aceder à nossa conta, fazer uploads e gerir a nossa biblioteca.

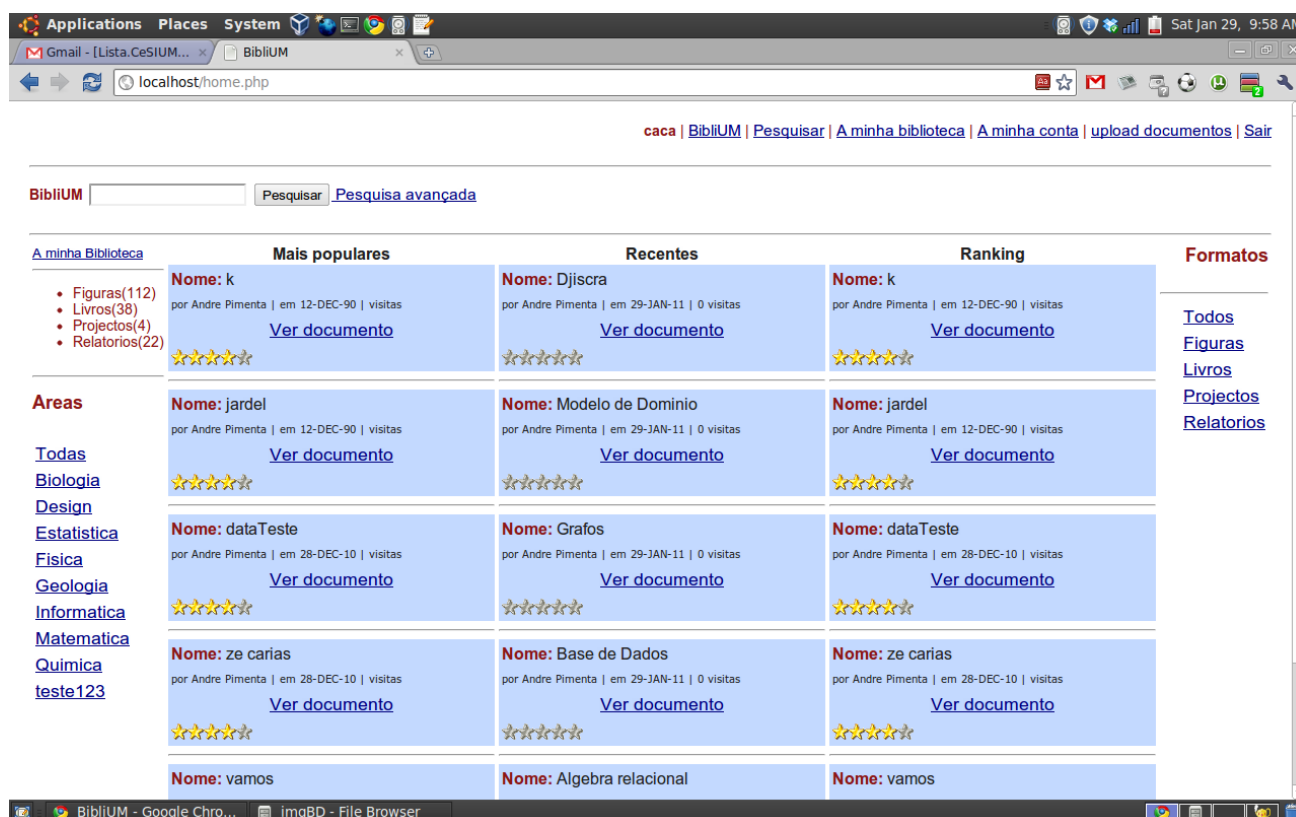


Figura 3.3: Pagina principal

### 3.1.4 Upload de Documentos

É permitido e de certa forma essencial para o funcionamento do **BibliUM** a partilha de recursos. Para que esta mesma partilha se realize cada utilizador poderá fazer upload de documentos, sendo que este estará limitado ao tamanho máximo de 25MB.

Ao fazer upload de qualquer documento temos uma lista de campos a preencher para podermos categorizar e identificar da melhor maneira cada arquivo. Estes campos são apresentados na imagem que se apresenta a seguir.

Dos campos que são apresentados podemos dar destaque ao campo autor, que não é de carácter obrigatório, mas que poderá ser o próprio utilizador o autor e então confirma que o documento lhe pertence ou outra pessoa, e neste caso deverá preencher os campos que lhe são destinados.

The screenshot shows a web browser window with the address bar displaying 'localhost/upload.php'. The page title is 'Upload documentos'. At the top, there is a navigation bar with links: 'caca | BibliUM | Pesquisar | A minha biblioteca | A minha conta | upload documentos | Sair'. The form contains the following fields and controls:

- \*Nome do documento:** A text input field.
- Da sua autoria?** A checkbox with the text '(Se seleccionar esta opção deve ignorar os campos autor)'.
- Autor:** A text input field.
- Data de nascimento(Autor):** Three dropdown menus for 'Dia', 'Mês', and 'Ano'.
- Localidade(Autor):** A dropdown menu labeled 'Escolha um País'.
- \*Tipo do documento:** A dropdown menu labeled 'Escolha um tipo'.
- Universidade:** A dropdown menu labeled 'Escolha uma Universidade'.
- \*Categoria:** A dropdown menu labeled 'Escolha uma area'.
- Descrição:** A large text area.
- \*Ficheiro a carregar:** A 'Choose File' button, followed by the text 'No file chosen'.
- Guardar:** A button to submit the form.

At the bottom of the form, there is a red asterisk and the text '\* Campos obrigatórios'.

Figura 3.4: Upload de documentos

### 3.1.5 Conta do utilizadro

O utilizador poderá alterar os seus dados pessoais e de acesso ao **BibliUM**. Na imagem que se apresenta a seguir poderemos ver quais os campos com premissões para serem alterados.

The screenshot shows a web browser window with the address bar displaying 'localhost/myAccount.php'. The page title is 'Minha conta'. At the top, there is a navigation bar with links: 'caca | BibliUM | Pesquisar | A minha biblioteca | A minha conta | upload documentos | Sair'. The form is divided into two sections:

- Alterar password:**
  - \*Password actual:** A text input field.
  - \*Nova password:** A text input field.
  - Confirmar nova password:** A text input field.
  - Alterar:** A button to submit the password change.
- Editar informações pessoais:**
  - \*Primeiro nome:** A text input field with the value 'Andre'.
  - \*Ultimo nome:** A text input field with the value 'Pimenta'.
  - \*Email:** A text input field with the value 'apri@di.uminho.pt'.
  - Numero de telefone:** A text input field.
  - Sexo:** Two radio buttons, 'Masculino' (selected) and 'Feminino'.
  - Localidade:** A dropdown menu with the value 'Espanha'.
  - Data de nascimento:** Three dropdown menus for 'Dia', 'Mês', and 'Ano', with values '01', 'Jan', and '2000' respectively.
  - Alterar:** A button to submit the personal information changes.
  - Limpar:** A button to clear the form.

At the bottom of the form, there is a red asterisk and the text '\* Campos obrigatórios'.

Figura 3.5: Minha conta

### 3.1.6 Biblioteca do utilizador

Cada utilizador poderá alojar os documentos que pretenda e que queira disponibilizar para partilha. Têm ainda a possibilidade de remover estes caso o pretenda.

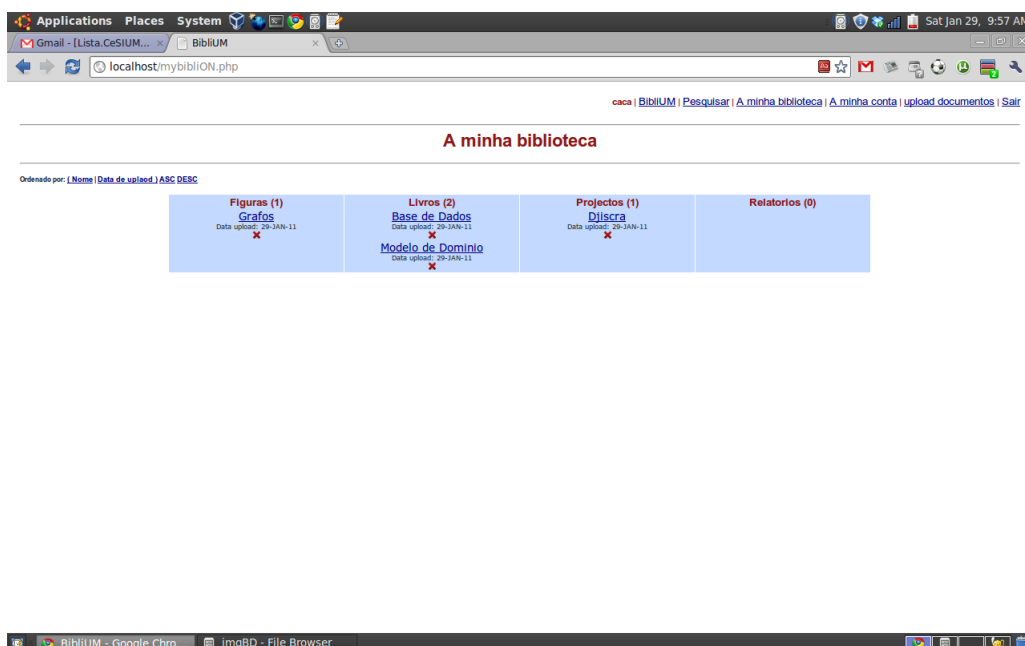


Figura 3.6: Minha biblioteca

### 3.1.7 Pesquisa

A pesquisa será uma das principais funções oferecidas ao utilizador. Com a pesquisa poderemos ter acesso aos recursos que pretendemos. Também é oferecida uma pesquisa avançada que será apresentada mais adiante.

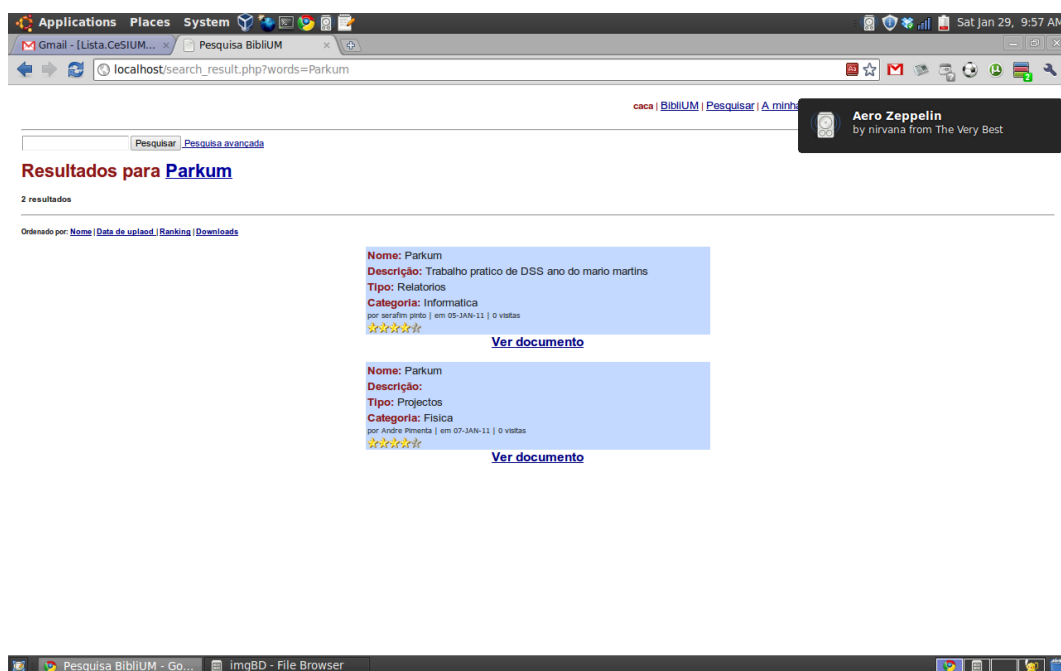


Figura 3.7: Pesquisa de documentos

### 3.1.8 Pesquisa avançada

Através da pesquisa avançada podemos procurar os recursos que pretendemos de forma minuciosa. Este recurso é de extrema importancia para o utilizador pois permite que este encontre os recursos que pretende de forma organizada e coerênte.

Applications Places System Sat Jan 29, 10:00 AM

Gmail - [Lista.Cesium...] Pesquisa avançada

localhost/advanced\_search.php

caca | BibliUM | Pesquisar | A minha biblioteca | A minha conta | upload documentos | Sair

## Pesquisa avançada

Com as palavras

Sem as palavras

Nome ficheiro

Autor

Categoria: Escolha uma area ▼

Universidade Escolha uma Universidade ▼

Data upload entre 01 ▼ Jan ▼ 2001 ▼  
e 29 ▼ Jan ▼ 2011 ▼

Tipo de documento Escolher Tipo de documento ▼

Pesquisar

Limpar

Pesquisa avançada - G... imgBD - File Browser home.png

Figura 3.8: Pesquisa avançada

### 3.1.9 Visualização de documentos

Cada documento que o utilizador pretenda usar é apresentado de forma detalha caso este pretenda. Na visualização de documentos é apresentado todos os detalhes sobre cada documento e ainda a possibilidade de o utilizador "classificar" este documento, dando a sua opinião.

É ainda apresentado um sistema de comentarios. Aqui o documento poderá ser comentado e discutido por cada utilizador.

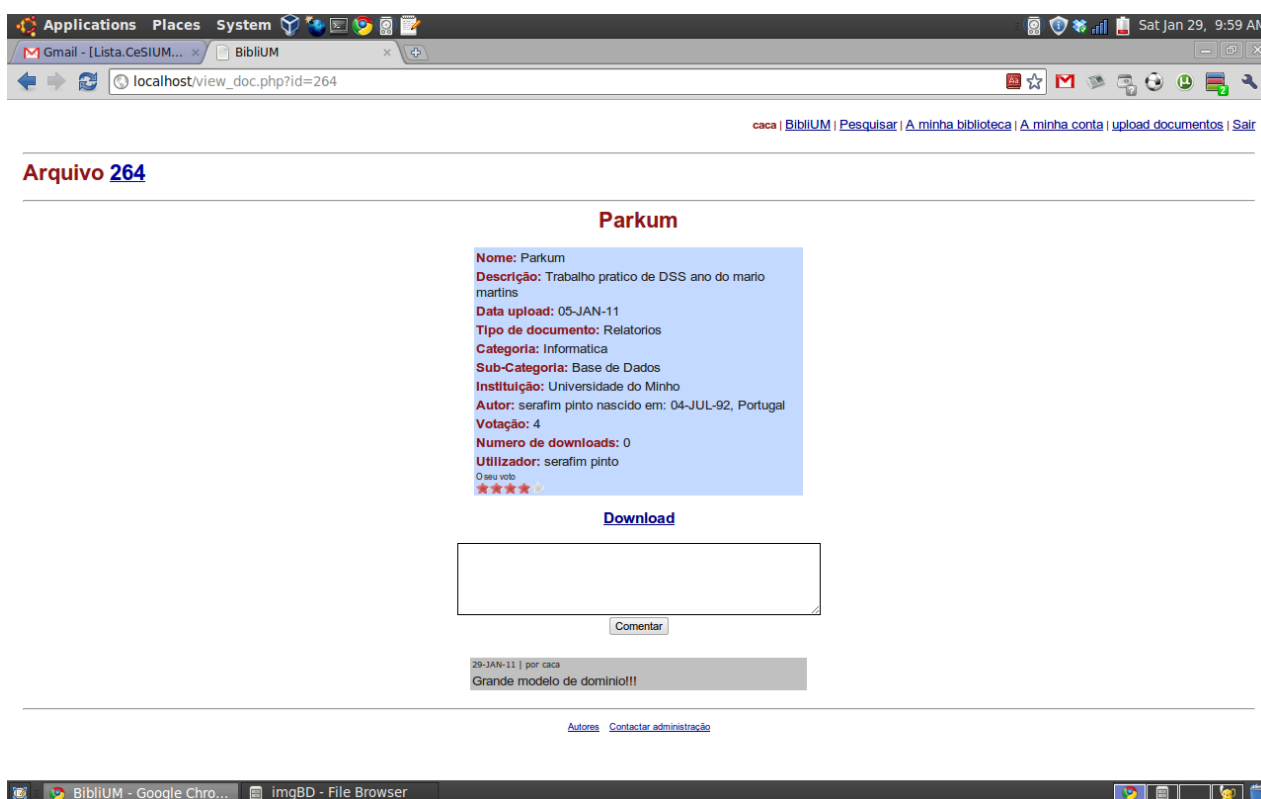


Figura 3.9: Visulazação de documentos

## 3.2 Aplicação em JAVA

Para que seja ainda mais fácil a sua utilização decidimos criar uma pequeno manual que simplifique ainda mais as capacidades que a nossa aplicação de administração da **BibliUM**.

A primeira interface a que o administrador é sujeito, trata-se do login:



Figura 3.10: Login



Ao fazer o login o administrador faz o login directo à base de dados, escolhendo assim a que base de dados se pretende ligar.

Na imagem abaixo temos o ambiente principal da nossa aplicação. No primeiro separador temos alguns dados estatísticos e o acesso a alguns gráficos. Para além disso, podemos aceder a cada uma dos separadores como é possível ver na parte superior da imagem. Cada separador (exceptuando o qual nos encontramos) corresponde a cada uma das tabelas da nossa base de dados.



Figura 3.11: Home

Nas imagens seguintes temos os separadores que nos permitem fazer as pesquisas sobre os dados das nossas tabelas. A aplicação encontra-se assim dividida de forma a simplificar os processos de pesquisa e gestão ao administrador da **BibliUM**.

As pesquisas têm a particularidade de, depende do tipo de registo, poderem ser feitas com o auxílio de filtros. Por exemplo, podemos pesquisar apenas autores portugueses.

Os botões Ver e Adicionar oferece ao administrador a possibilidade de ver um dos registos seleccionados na lista e adicionar um novo registo, respectivamente.

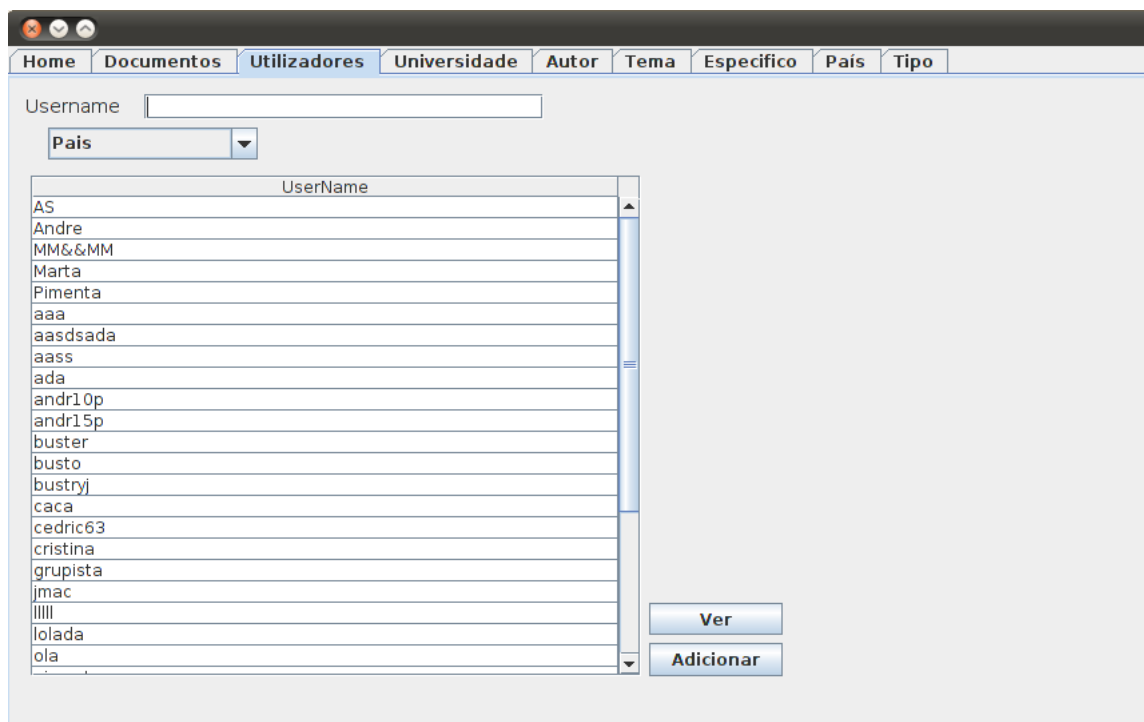


Figura 3.12: Utilizadores

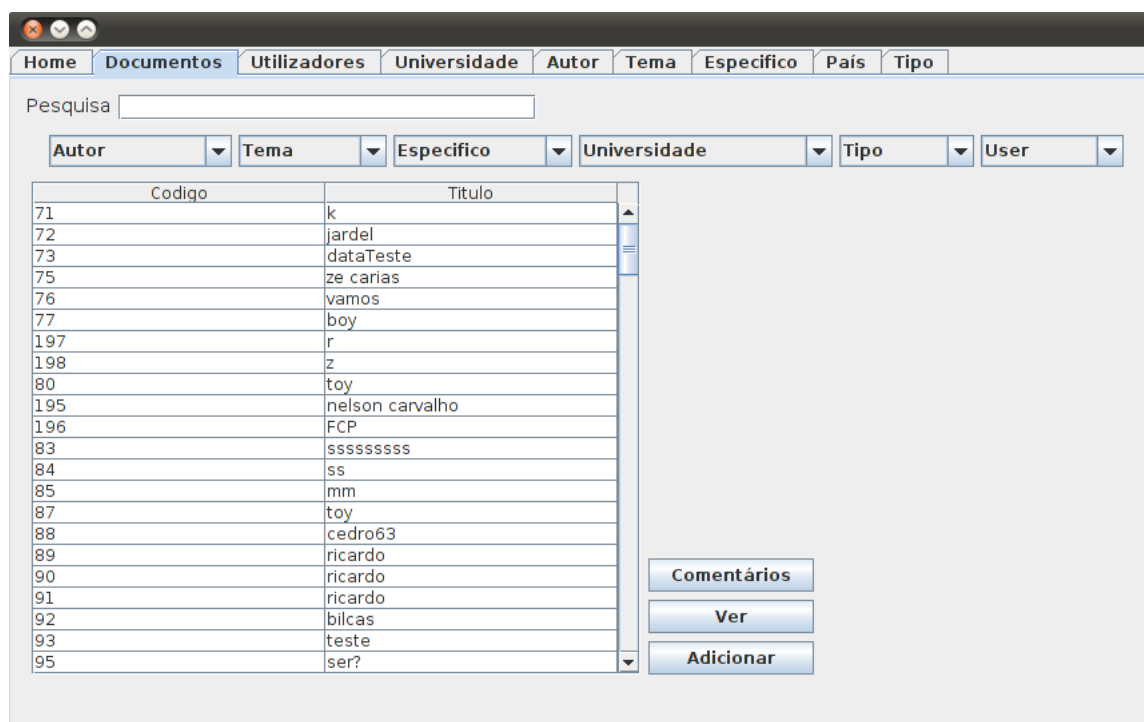


Figura 3.13: Documentos

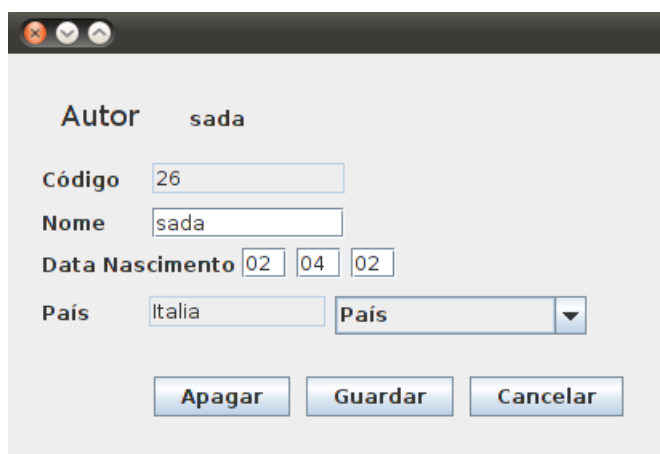
O separador dos documentos (ver imagem acima) têm ainda o botão comentários que permite ao utilizador ver e gerir os comentários do documento seleccionado na tabela, caso este os tenha.

As imagens seguintes mostram a opção de ver um registo de cada um dos separadores do nosso ambiente principal. Nestas janelas temos a opção de editar os dados do registo escolhido, bem como apaga-lo.



A screenshot of a software window titled 'País' (Country). It contains a header with 'País' and 'Brasil'. Below this are three input fields: 'Código' (Code) with the value '3', 'Nome' (Name) with the value 'Brasil', and 'País' (Country) with the value 'Brasil'. At the bottom are three buttons: 'Apagar' (Delete), 'Guardar' (Save), and 'Cancelar' (Cancel).

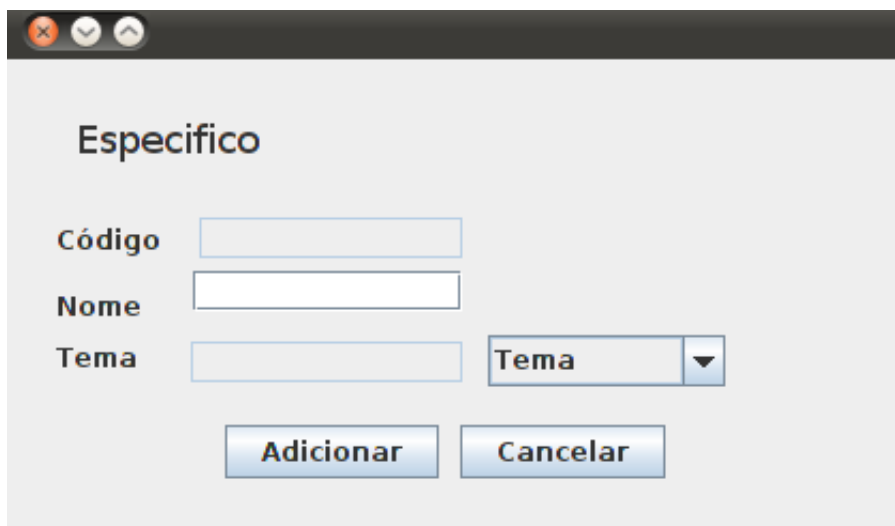
Figura 3.14: País



A screenshot of a software window titled 'Autor' (Author). It contains a header with 'Autor' and 'sada'. Below this are five input fields: 'Código' (Code) with the value '26', 'Nome' (Name) with the value 'sada', 'Data Nascimento' (Date of Birth) with the value '02/04/02', 'País' (Country) with the value 'Italia', and a dropdown menu for 'País' (Country) with a downward arrow. At the bottom are three buttons: 'Apagar' (Delete), 'Guardar' (Save), and 'Cancelar' (Cancel).

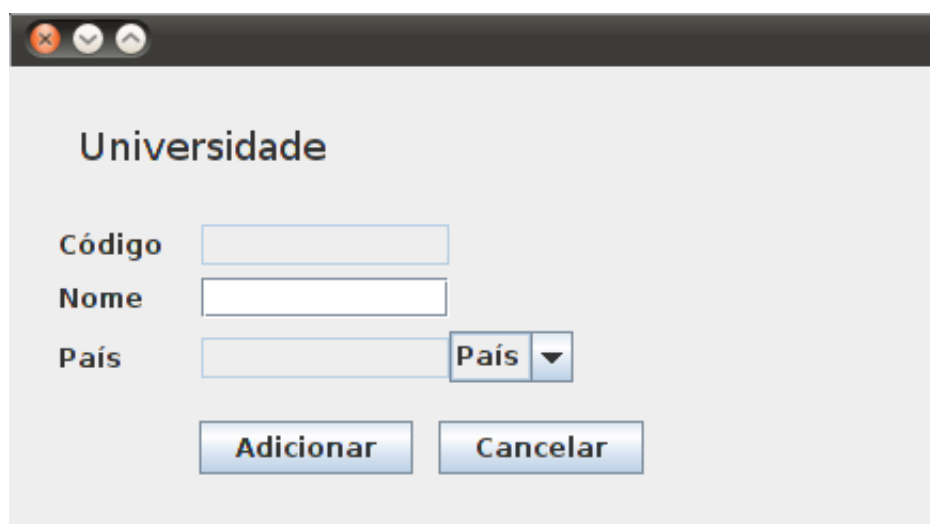
Figura 3.15: Autor

No caso do administrador optar por adicionar um determinado registo (Através do botão Adicionar referido acima), a mesma janela é inicializada mas vazia, para que o administrador insira os dados e adicione o registo à base de dados, como ilustram as imagens seguintes:



A screenshot of a software window titled 'Especifico' (Specific). It contains three input fields: 'Código' (Code), 'Nome' (Name), and 'Tema' (Topic). At the bottom are two buttons: 'Adicionar' (Add) and 'Cancelar' (Cancel).

Figura 3.16: Especificidade



The image shows a web application window titled "Universidade". It contains three input fields: "Código", "Nome", and "País". The "País" field is accompanied by a dropdown menu labeled "País". Below the input fields are two buttons: "Adicionar" and "Cancelar".

Figura 3.17: Universidade

Por fim, e como já havia sido referido mais acima, a nossa aplicação oferece também ao administrador a consulta de 4 gráficos elaborados através de dados estatísticos da base de dados, como exemplifica a imagem seguinte.

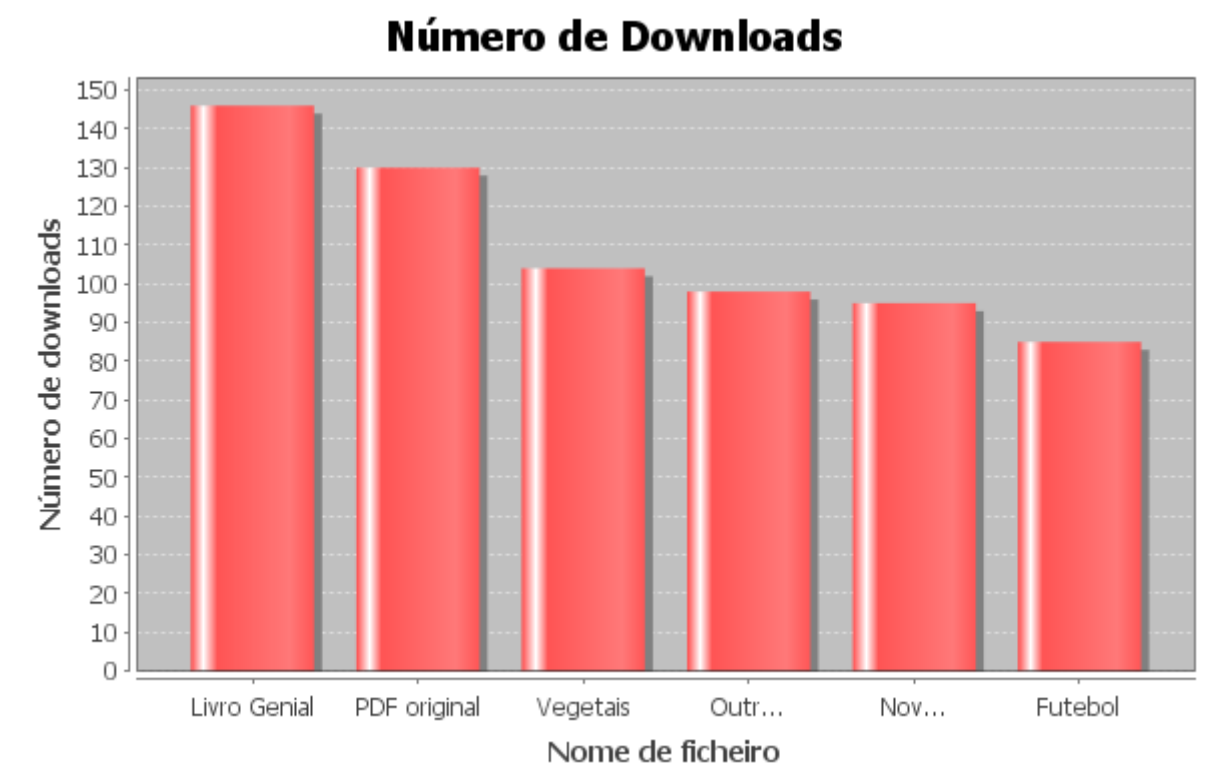


Figura 3.18: Grafico

## Capítulo 4

# Conclusão

Da elaboração do **BibliUM** podem ser retiradas varias conclusões. Uma das principais conclusões é a extrema utilidade que as base de dados nos oferecem.

Com a utilização de base de dados conseguimos usufruir da permanencia de dados. No entanto apesar de ser a principal característica das base de dados podemos ver outros aspectos importantes.

Com uma base de dados bem estruturada o acesso e controlo de dados torna-se muito facil e agradável de utilizar as base de dados, pois o seu manuseamento torna-se bastante simples. Tendo então uma base de dados bem estruturada e organizada é possível criar grandes projecto de forma simplificada e ao mesmo tempo permite a criação de projecto de enorme utilidade.

No entanto apesar de falarmos de base de dados bem estruturadas e organizadas elas não 'nascem' simplesmente desta forma, precisam de passar por um processo de normalização e só após a passagem por este processo se poderá dizer que a base de dados se encontra bem estruturada.

Outro ponto bastante importante é o uso de PL/SQL que permite e oferece uma melhoria no controlo e funcionalidades da base de dados. Com PL/SQL podemos implementar varias funcionalidades que não são oferecidas pelas estruturas da base de dados, podemos mesmo criar procesos que satisfaçam necessidades exclusivas à nossa base de dados. Apesar do projecto se basear fundamentalmente na construção de uma base de dados e da exploração das suas capacidades foram usadas outras plataformas de desenvolvimento de software, nomeadamente PHP e JAVA como já foi referido. Neste ponto podemos dizer que a utilização de destas plataformas especialmente PHP também foi muito util, pois adquirimos e desenvolvemos conhecimentos sobre estas.

De uma forma geral assumimos que este trabalho veio trazer uma grande noção sobre o que é usar/criar aplicações com o uso de uma base de dados, mas também as potencialidades desta.

Relativamente ao projecto achamos que este ficou algo que podemos considerar interessante. A **BibliUM** permite a criação de uma biblioteca onde toda a gente podia aceder de forma simples e contribuir para esta, a ideia será tal como outros projectos dedicados a partilha de informação, a utilização deste projecto para partilha e uniformização da informação por parte dos utilizadores, que neste caso serão 'recursos' academicos.

## Capítulo 5

# Bibliografia

- Machado, José. “Bases de Dados“. 2009.
- Begg, Carolyn. Connolly, Thomas. “Database Systems: A Practical Approach to Design, Implementation and Management“. 5th edition, 2010.
- <http://www.java2s.com/Code/Java/Chart/CatalogChart.htm>
- <http://www.techonthenet.com/oracle/>
- <http://www.w3schools.com/php/default.asp>
- <http://academicjava.com/JavaTutorial/Swing.html>

## Capítulo 6

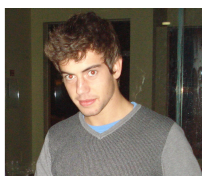
# Elementos do Grupo



**Figura 6.1:** João Miguel



**Figura 6.2:** André Pimenta



**Figura 6.3:** Daniel Santos



**Figura 6.4:** Milton Nunes