



**Licenciatura em Eng.<sup>a</sup> Informática**

**Projecto Final de Curso – 2010/2011**

***Aplicação de Gestão Espaço em Disco para bases de dados ORACLE***

**Alunos:**

**Ricardo Alvarez nº 20083549**

**José Pedro Esteves Correia nº20080612**

**Orientador: Professor José Aser Castillo Lorenzo**

**Lisboa, Agosto de 2011**

## **Abstract**

The objective of this work is to build up an application for Remote Oracle Database Administrators (DBA) ORACLE. It will improve the response time and mobility for executing tasks related to database storage management, which will give a better mobility in what concerns the implementation of tasks associated to the management of the space in disk of the data base ORACLE. For this purpose and using new technologies, we developed an application for the mobile Android platform that will enable a DBA to manage the database disk space using a mobile phone or a tablet PC running Android operating system.

We used a server, WAMPSEVER, which is not only responsible for the PHP and MySQL, but also for the management of the ORACLE database. We also had in mind that some of the statments (SQL) sent to ORACLE might take too long to be performed. Having this fact in mind, we have created an auxiliary table in a MySQL database which allows the management of the long running SQL statements, sending the information about the state of the statments to the Android system. This table is also used as a log for registering all SQL statements executed by the remote DBA. As last product, this table lets us know all the statments SQL requested to ORACLE.

## Índice

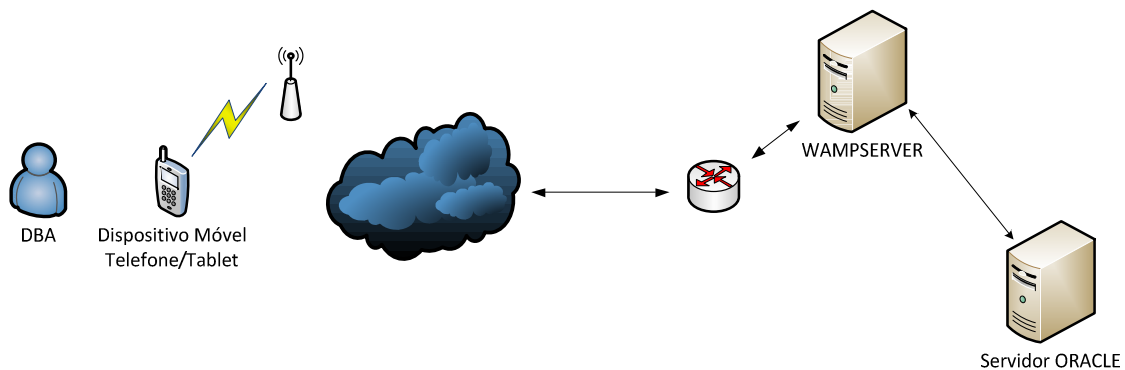
Abstract.....	2
1 - Introdução.....	4
2 – Requisitos .....	5
3 – Estruturas físicas e lógicas de armazenamento em disco <i>ORACLE</i> .....	7
4 - Casos de Uso .....	8
5 – Diagramas UML .....	9
6 – Metodologia.....	11
7 - Descrição da aplicação – Parte 1 Android .....	13
8 - Descrição da aplicação – Parte 2 Servidor PHP .....	15
9 – Classes .....	23
10 – Índice Figuras .....	37
11 – Bibliografia.....	38
12 - Índice Remissivo .....	39

# 1 - Introdução

O tema que escolhemos foi “Ferramenta WEB para Administração de uma BD ORACLE” e mais especificamente a “Gestão de espaço em disco das BD ORACLE”, tendo como objectivo o desenvolvimento de uma aplicação WEB com recurso à tecnologia PHP.

Quando começamos o levantamento de requisitos verificámos, que o principal requisito funcional era a aplicação ser destinada a administradores de base de dados ORACLE, que precisam de ter acesso às base de dados em qualquer altura e em qualquer lugar. Como faríamos para dar este acesso aos DBA?

Decidimos então abordar o problema de outra forma, em vez de uma plataforma WEB tradicional, fazer uma aplicação que “corresse” num telefone ou num PDA, para tal desenvolvemos a aplicação na plataforma Android, o que nos permitiu solucionar o principal requisito.



**Fig. 1 - Diagrama Geral da Aplicação**

Para efectuar este desenvolvimento deparámo-nos com várias dificuldades, tais como:

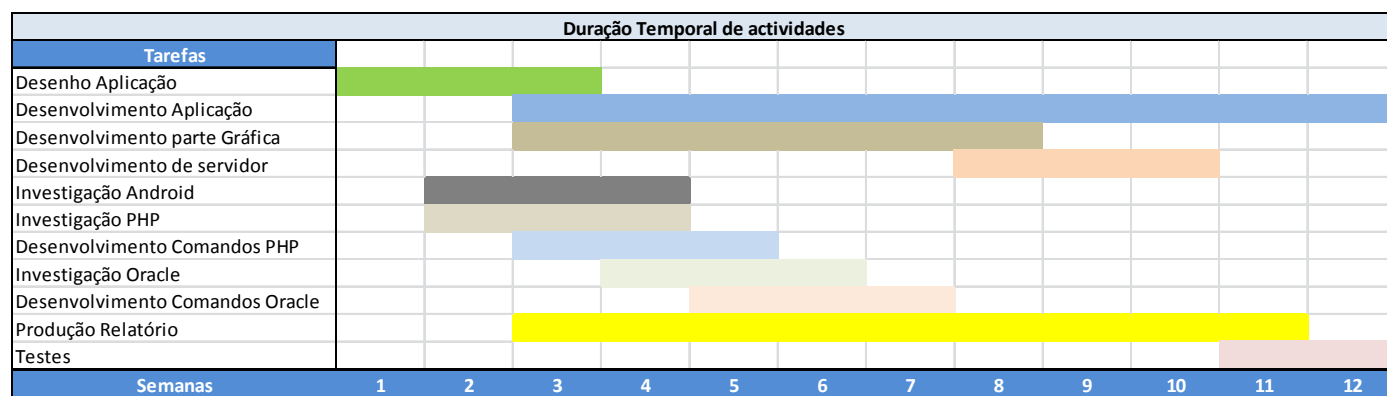
- Ambientes distintos, Android Java, PHP e SQL para ORACLE.
- Criação de um servidor que nos permitisse executar os comandos PHP.
- Comunicações entre o ORACLE e o dispositivo móvel.
- Tratamento de comandos SQL de longa duração.

No desenvolvimento da aplicação Android socorremo-nos dos apontamentos das aulas do professor Alexandre Pereira, que nos permitiu o começar a escrever a aplicação base, embora à medida que fomos incorporando novos objectos tivemos necessidade de aprender a manusear os mesmos, pelo que tivemos que consultar exaustivamente vários sites especializados, tais como o <http://android.developer.com>, <http://stackoverflow.com>, etc..

Para a criação do servidor recorremos ao WAMPSEVER que é uma aplicação freeware que nos permitiu a execução de páginas PHP e ligação às bases de dados ORACLE e MySQL, também nesta fase encontrámos algumas dificuldades, tais como o desenvolvimento das páginas PHP e a forma de ligação às bases de dados ORACLE.

Para a execução dos comandos ORACLE recorremos ao livro do professor José Aser “Administração de Bases de Dados Oracle para programadores – Gestão de Espaço em Disco.”, notas fornecidas pelo professor e consultas no site [www.oracle.com](http://www.oracle.com).

Outra dificuldade adicional que surgiu foi o tratamento de comandos SQL com tempos de execução longo, para isso tivemos que utilizar uma base de dados suplementar (em MySQL), que ficou com a tarefa de informar o dispositivo móvel do estado dos comandos enviados ao ORACLE.



**Fig. 2 - Duração Temporal de Actividades**

## 2 – Requisitos

### **Stakeholders**

- Utilizadores *ORACLE* com permissões *sysdba*<sup>1</sup>.

### **Funcionais**

- Plataforma Android para permitir a mobilidade dos DBA
- Interface claro e eficaz

### **Não Funcionais**

- Equipamento para o servidor PHP e *MySQL*, com alto desempenho por forma a permitir múltiplos acessos em simultâneo.

---

<sup>1</sup> Utilizador que tem permissões para realização de todas as tarefas inerentes à base de dados

### 3 – Estruturas físicas e lógicas de armazenamento em disco *ORACLE*

O SGBD<sup>2</sup> da *ORACLE* armazena toda a informação da base de dados em disco em *tablespaces*<sup>3</sup> (*datafiles*<sup>4</sup>) que podem ser constituídos por blocos de disco (Sistema Operativo) de 2,4,8,16 ou 32K, pelo que cada *tablespace* é um bloco *ORACLE*.

O SGBD criou esta organização do disco em *tablespaces* para lhe permitir ser o próprio a fazer a gestão e optimização do espaço em disco.

Assim como podemos ver na fig.1 os *tablespaces* contêm todos os objectos geridos pelo SGBD, tais como tabelas, índices, *Schemas*, etc.. Por sua vez um *tablespace* é constituído por *segments* e *extents* que são alocações contíguas de blocos *ORACLE*.

A alocação de espaço de uma tabela ou índice no *tablespace* é feita pelo comando CREATE TABLE ou CREATE INDEX.

Desta forma e no intuito de optimizar o acesso aos discos normalmente na criação das base de dados devem ser criadas *tablespaces* para tabelas com pequeno volume de dados, *tablespaces* para tabelas com médio volume e *tablespaces* para tabelas com um grande volume de informação.

Assim quando criamos uma *table* é criado no *tablespace* um *segment* que irá criar um ou vários *extents* consoante o crescimento do objecto.

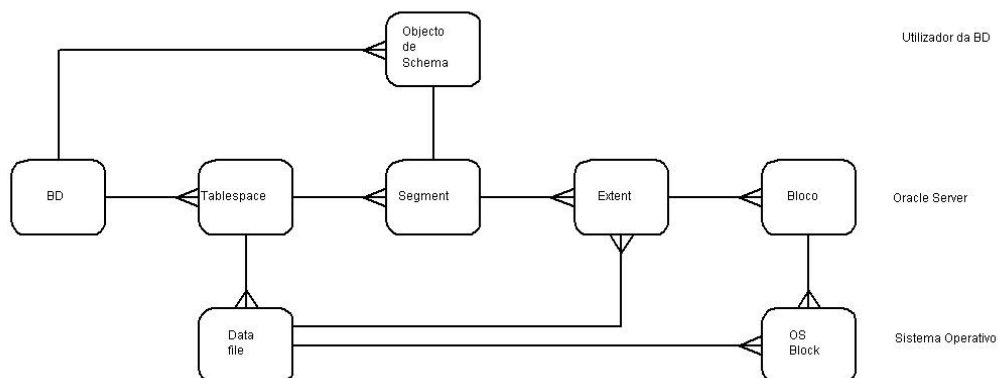


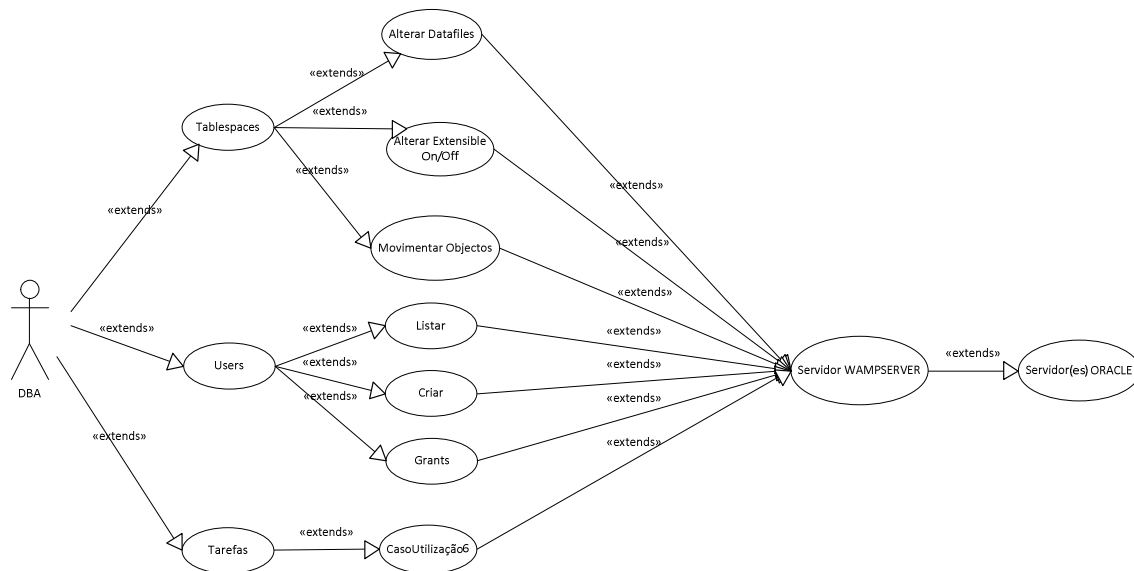
Fig. 3 – Estrutura Física do *ORACLE*

<sup>2</sup> Sistema de gestão de base dados

<sup>3</sup> Unidade lógica de armazenamento

<sup>4</sup> Estrutura física

## 4 - Casos de Uso



**Fig. 4 - Caso Uso de Gestão Espaço em Disco Oracle**



# 5 – Diagramas UML

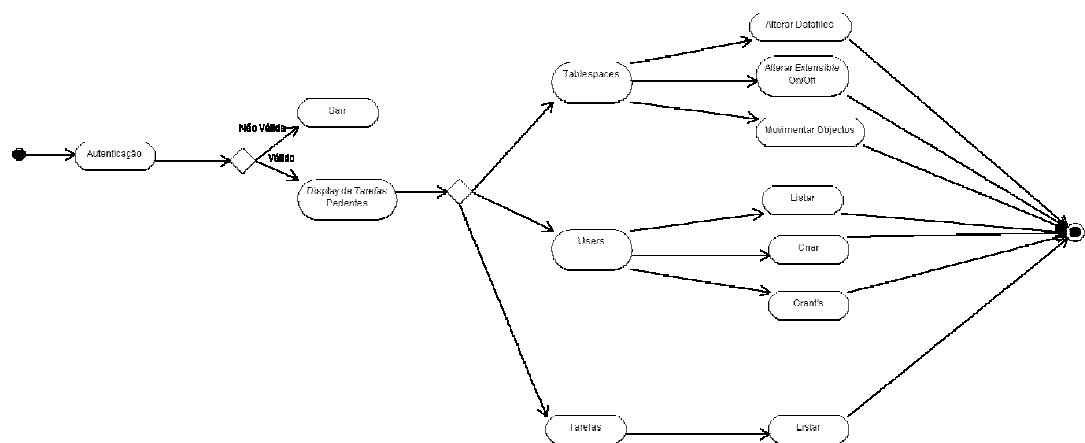


Fig. 5 - Diagrama de Actividades

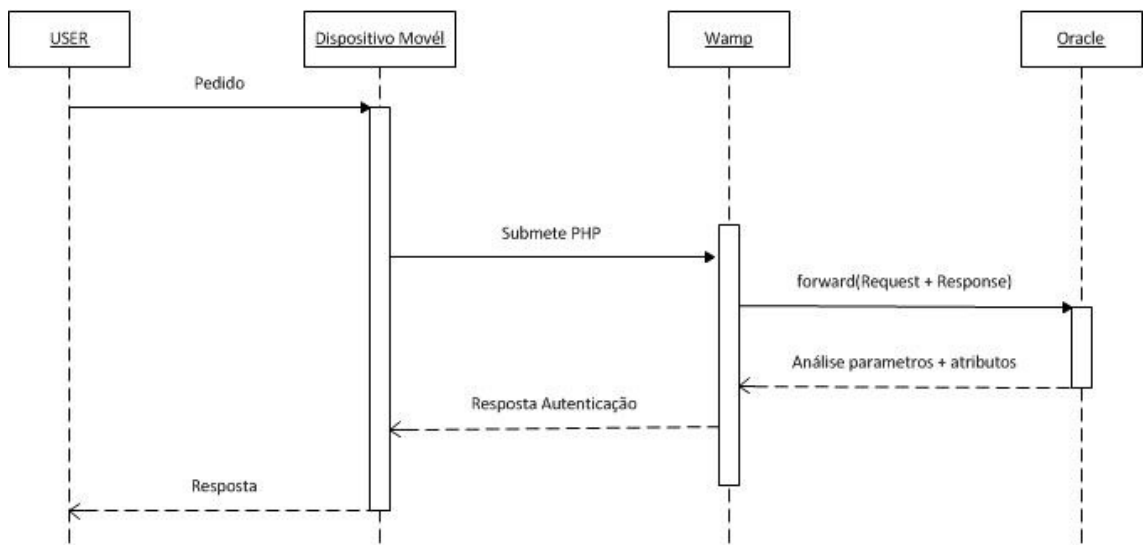
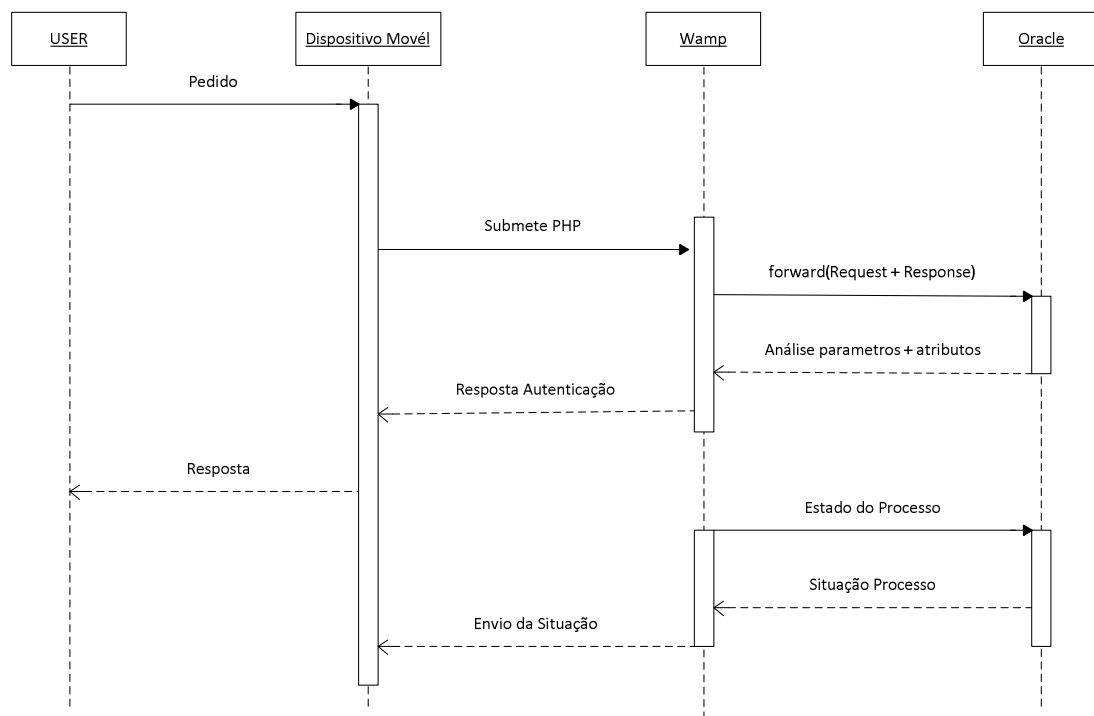


Fig. 6 - Diagrama de Sequência de Autenticação



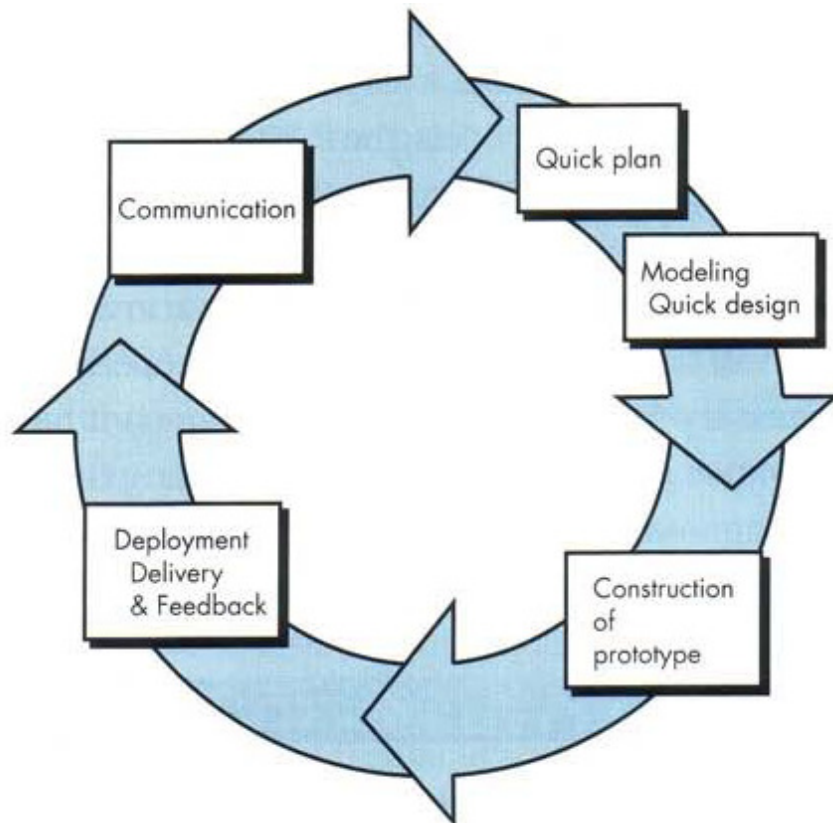
**Fig. 7 - Diagrama de Sequência de Queries Longos**

## 6 – Metodologia

A aplicação a que nos propusemos desenvolver é uma aplicação que se destina a um público-alvo muito especial, estamos a falar nomeadamente de *DBA* de bases de dados *ORACLE*, pelo que optámos por uma metodologia “*software prototyping*”, que nos traz benefícios como ter rapidamente um feedback dos utilizadores após a publicação de um protótipo, pelo que consideramos que a melhor situação seria a de desenvolver um protótipo com as funcionalidades básicas e a partir do feedback que iremos ter dos utilizadores criar então um modelo funcional em que os erros estarão corrigidos e novas funcionalidades incorporadas.

Para o desenvolvimento do protótipo inicial partimos dos requisitos propostos pelo nosso orientador professor José Aser, aos quais adicionámos algumas funcionalidades novas.

Devido ao requisitos não funcionais não serem definidos com muito detalhe, escolheu-se como paradigma de desenvolvimento de protótipo. O protótipo permite que se desenvolva rapidamente e que se baseie maioritariamente na comunicação. Este paradigma permite que seja desenvolvida uma solução rapidamente e adequada para grande parte da engenharia de software. Desde que determinada a regra de interacção desde o início, esta abordagem demonstra-se relevante. Como passo inicial, este paradigma encaminha-nos para uma solução final com qualidade assegurada e com total parecer dos Stakeholders.



5

**Fig. 8- Paradigma de Prototipagem**

---

<sup>5</sup> Robert Pressman

## 7 - Descrição da aplicação – Parte 1 Android

Como falamos de bases de dados, a primeira funcionalidade que faz parte da aplicação é de pedir os dados para autenticação nas bases de dados. Como estamos a desenvolver uma aplicação para DBA o utilizador em questão deve ter privilégios de administração *sysdba*. A aplicação começará por apresentar um aviso reportando os últimos comandos finalizados.

O trabalho ficou dividido em duas partes, a parte cliente (Android) que faz o interface com o utilizador e a segunda parte que faz a comunicação com as bases de dados *ORACLE*.

Funcionalidades da aplicação Android

A aplicação foi dividida em três grandes áreas funcionais:

7.1 - Tablespaces

7.2 - Users

7.3 – Tarefas

### 7.1 Tablespaces

#### 7.1.1 – Listar Tablespaces

Faz o display de uma *datagrid* onde aparecem todas as tablespaces que fazem parte da base de dados *ORACLE* que estamos a gerir. Clicando num item fará o display dos datafiles que fazem parte da tablespace escolhida, neste novo ecrã aparecem no menu duas novas funcionalidades:

Extensible OFF

Alterar Datafile

##### 7.1.1.1 - Extensible OFF

Ao clicar neste comando, o datafile fica com o atributo extensible igual a OFF.

##### 7.1.1.2 - Alterar Datafile

Permite alterar o tamanho ocupado em disco pelo *datafile* segundo os valores definidos pelo DBA, este comando é um comando que pode levar muito tempo a ser executado, pelo que após o pedido de execução o Android fica liberto para executar outras tarefas e será avisado posteriormente quando o mesmo tiver terminado.

#### 7.1.2 - Movimentar Objectos

Este comando permitirá ao DBA movimentar objectos de um tablespace para outro e desta forma racionalizar o espaço em disco.

## 7.2 Users

### 7.2.1 – Listar utilizadores

### 7.2.2 – Criar utilizadores

### 7.2.3 – Permissões aos utilizadores (Grant)

7.2.1 – Listar Utilizadores permite ao DBA obter uma lista de todos os utilizadores utilizando a mascara “%” ou conhecer os dados relativos a um determinado utilizador.

7.2.1 – Criar utilizadores permite criar novos utilizadores por forma a obter novos “Schemas”

### 7.2.3 – Permissões aos utilizadores (Grant)

Permite dar as opções necessárias consoante o que se pretende que façam.

## 7.3 Tarefas

### 7.3.1 – Listar tarefas

### 7.3.1 – Listar tarefas

Permite listar as tarefas já executadas ou em execução mediante a escolha que é feita através da selecção de “radio buttons”.

Esta funcionalidade para além de ser muito importante para conhecer o estado de execução das tarefas também permite ficar com um log de todas as tarefas pedidas ao *ORACLE*.

## 8 - Descrição da aplicação – Parte 2 Servidor PHP

Para o funcionamento da aplicação será necessário o recurso a um servidor que receba as informações recebidas do Android por *Post* e executar o correspondente ficheiro em PHP que executará os comandos SQL a enviar ao *ORACLE* necessários para cada uma das tarefas, assim como actualizar uma base de dados em *MySQL* que irá conter todos os comandos recebidos dos utilizadores, permitindo um log dos comandos e transmitir aos utilizadores o estado da execução dos comandos no *ORACLE*.

Neste caso, o servidor escolhido foi o WAMPSEVER que permite a execução de comandos PHP assim como o tratamento de bases de dados *ORACLE* e *MySQL*.

Dado que o número de ligações pode crescer muito rapidamente, é necessário que este servidor esteja alojado num sistema com ligações múltiplas e rápidas conexões ao exterior.

### 8.1 - Base de dados *MySQL*

Esta base de dados (TFC) contém uma só tabela “tarefa” que irá conter todos os pedidos de SQL efectuados pelos utilizadores.

tarefa

Campo	Tipo	Nulo	Defeito	Comentários
tarefa_id	int(11)	Não		Chave
user	varchar(30)	Não		user - corresponde ao user <i>ORACLE</i> com permissões DBA
desc_tarefa	varchar(150)	Sim	<i>NULL</i>	Descrição do tipo tarefa a ser executado
data_pedido	datetime	Sim	<i>NULL</i>	data/hora comando SQL
data_conclusao	datetime	Não		data/hora conclusão comando SQL
concluido	int(11)	Não		0=pendente / 1=concluido
Apresentar	bit(1)	Não	b'0'	Indicativo que permite que a tarefa seja apresentada no Android, quando o user chama a aplicação. 0=não apresenta / 1=apresenta
sql_statment	varchar(300)	Sim	<i>NULL</i>	Comando SQL enviado ao <i>ORACLE</i>
ip_dba	varchar(19)	Sim	<i>NULL</i>	Endereço publico de IP do utilizador

#### Índices:

Nome do Índice	Tipo	Único	Packed	Campo	Quantidade	Collation	Nulo	Comment
PRIMARY	BTREE	Sim	Não	tarefa_id	85	A		

### Exemplo dados contidos na tabela após funcionamento da aplicação.

**Máquina:** localhost

**Base de Dados:** tfc

**Data de Criação:** 08-Ago-2011 às 11:02

**Gerado por:** PHPMyAdmin 3.3.9 / *MySQL* 5.5.8-log

**Comando SQL:** SELECT \* FROM `tarefa` LIMIT 0, 30 ;

**Registos:** 30

tarefa_id	user	desc_tarefa	data_pedido	data_conclusao	concluido	Apresentar	sql_statement	ip_dba
1	sys	alterar database table	2011-07-15 16:54:22	0000-00-00 00:00:00	0	0	ALTER DATABASE DATAFILE	
2	sys	tarefa concluida	2011-07-17 13:03:06	2011-07-18 00:00:00	1	0	sem sql	
38	sys	Alterar Datafiles	2011-07-27 15:53:05	2011-07-27 15:53:06	1	1	'c:\ORACLE\oraadata\XE\TFC_01.dbf'	
39	sys	Alterar Datafiles	2011-07-27 15:55:07	2011-07-27 15:55:07	1	1	'c:\ORACLE\oraadata\XETFC_01.dbf'	
40	sys	Alterar Datafiles	2011-07-27 15:55:34	2011-07-27 15:55:35	1	1	'c:\ORACLE\oraadata\XETFC_01.dbf'	
41	sys	Alterar Datafiles	2011-07-27 15:56:31	2011-07-27 15:56:31	1	1	'c:\ORACLE\oraadata\XETFC_01.dbf'	
42	sys	Alterar Datafiles	2011-07-27 15:58:38	2011-07-27 15:58:38	1	1	'c:\ORACLE\oraadata\XETFC_01.dbf'	
43	sys	Alterar Datafiles	2011-07-27 15:59:46	2011-07-27 15:59:46	1	1	c:\ORACLE\oraadata\XETFC_01.dbf	



## 8.2 – Ficheiros PHP

Os ficheiros que passamos a descrever contêm os diversos métodos e instruções a fornecer à base de dados ORACLE, que ficou previamente identificada durante a autenticação do utilizador. De referir que cada um dos PHP envia e recebe dados para as bases de dados ORACLE, mas grava também um registo na tabela “tarefa” da base de dados MySQL, que permitirão ter um log das operações assim como conhecer o estado em que se encontram as tarefas.

**get\_tarefas.php** – recebe por Post “cod” que serve para fazer uma consulta selectiva à base de dados MySql (TFC).

Se o “cod”= 1 envia todas os registos da tabela “tarefa” que estão terminados ou seja campo concluído=1.

Se o “cod” = 0 envia todas os registos da tabela “tarefa” que estão pendentes ou seja campo concluído=0

Se o “cod” = % envia todas os registos da tabela “tarefa”.

**get\_objectos.php** – Extrai do ORACLE todas as tabelas e índices de um determinado utilizador.

**read\_users.php** – recebe por Post “user” e devolve os dados do ORACLE relativos a este user, se o “user” for igual a % devolve os dados relativos aos users da base de dados ORACLE. A base de dados de ORACLE será aquela em que o DBA se autenticou.

**alterar\_datafile.php** – recebe as seguintes variáveis por Post:

“dir” – contém o nome dos datafiles passíveis de alterar assim como o seu “path” completo.

“NewMax” – contém o novo valor máximo em MB do datafile

“OldMax” – contém o valor em MB actual do datafile

Coloca o atributo extensible do datafile em “ON” e aumenta o tamanho possível para o novo valor.

**alterar\_autoextend\_off.php** – recebe por Post “dir” que contém o nome do datafile e altera o seu atributo para “OFF”.

**criar\_users.php** – recebe por Post as seguintes variáveis:

“user” – nome do utilizador a criar

“userpwd” – password utilizador

“tablespace” – nome do tablespace a que o utilizador está afecto.

Seguidamente cria o utilizador assim como o respectivo “schema”.

**grant\_users.php** – Envia um comando SQL ao ORACLE com as permissões dadas a um utilizador.

**autenticar.php** – autenticar o utilizador (que forçosamente terá que ter privilégio SYSDBA).

**get\_tipos\_privilegios.php** – lista dos tipos de privilégios da tabela de sistema do ORACLE “system\_Privilege\_map”.

**get\_objectos.php** – lista dos objectos (Tables e índices) relativos a um determinado user.

### 8.3 – Comandos SQL para gestão das bases de dados ORACLE

Para que se possa fazer a gestão do espaço em disco nas bases de dados ORACLE é necessário criar para cada funcionalidade da aplicação um conjunto de comandos SQL, que serão executados através de ficheiros PHP, que passaremos a descrever:

#### Funcionalidade Autenticação

Ficheiro php: “autenticar.php”.

```
SELECT * FROM DBA_USERS WHERE USERNAME= $user;
```

Resposta: devolve “ok” se o user for encontrado ou “User Inexistente” se o user não existir.

#### Tab “Tablespaces”

##### Funcionalidade Listar TableSpaces

Ficheiro php: “read\_tablespaces.php”

```
SELECT TABLESPACE_NAME, CONTENTS, BIGFILE FROM  
DBA_TABLESPACES;
```

Resposta: Devolve todas as tablespaces

##### Funcionalidade Listar datafiles

Ficheiro php: “read\_datafile\_from\_tablespace.php”

```
SELECT FILE_NAME, AUTOEXTENSIBLE, INCREMENT_BY as  
INCREMENT_BYTES, round(MAXBYTES /1024/1024) as Max_Mb,  
ROUND(BYTES/1024/1024) as  
ALLOCATED_MB, ROUND(USER_BYTES/1024/1024) as USED_MB,  
ROUND((MAXBYTES-USER_BYTES)/1024/1024) as  
FREE_GROW_MB, ROUND((BYTES-USER_BYTES)/1024/1024) as  
FREE_IMMEDIATE_MB, TABLESPACE_NAME FROM DBA_DATA_FILES"  
WHERE TABLESPACE_NAME = ". $name;
```

Resposta: Devolve todos os datafiles referentes ao dataspace pedido.

##### Funcionalidade Autextensible ON/OFF

Ficheiro php: “alterar\_autoextend\_off.php”

```
ALTER DATABASE DATAFILE $data_file AUTOEXTEND OFF ;
```

Resposta: Altera o indicativo de autoextend para OFF.

### **Funcionalidade Alterar Datafile**

Ficheiro php: “alterar\_datafile.php”

```
ALTER DATABASE DATAFILE $data_file AUTOEXTEND ON NEXT  
$NewMaxSize MAXSIZE $OldMaxSize . "M";
```

Resposta: Altera as dimensões do datafile para as novas dimensões fornecidas.

### **Funcionalidade Movimentar Objectos**

Ficheiro php: “get\_objectos.php”

```
SELECT * FROM DBA_SEGMENTS WHERE OWNER = $cod AND  
(SEGMENT_TYPE = 'TABLE' or SEGMENT_TYPE = 'INDEX');
```

Resposta: Obtém todos os objectos que sejam do tipo TABLE ou INDEX

Ficheiro php: “read\_tablespaces.php”

```
SELECT TABLESPACE_NAME, CONTENTS, BIGFILE FROM  
DBA_TABLESPACES
```

Resposta: Obtêm todas as tablespaces que podem ser utilizadas como destino da movimentação.

Ficheiro php: “read\_users.php”

```
SELECT * FROM DBA_USERS;
```

Resposta: obtém todos os utilizadores do sistema.

Ficheiro php: “mover\_tabelas.php”

```
ALTER TABLE $user $objecto MOVE TABLESPACE $tablespace;
```

Resposta: Movimenta a tabela de um tablespace para outro.

Ficheiro php: “mover\_indices.php”

Resposta: Movimenta um índice para outro tablespace.

## **Tab “Users”**

### **Funcionalidade “Listar Users”**

Ficheiro php: “read\_users.php”

**SELECT \* from DBA\_USERS;**

ou

**SELECT \* from DBA\_USERS WHERE USERNAME LIKE \$name;**

Resposta: Obtém a lista dos utilizadores.

### **Funcionalidade “Criar Users”**

Ficheiro php: “read\_tablespaces.php”

**SELECT TABLESPACE\_NAME,CONTENTS,BIGFILE FROM  
DBA\_TABLESPACES;**

Resposta: Devolve todas as tablespaces

Ficheiro php: “criar\_users.php”

**CREATE USER \$user IDENTIFIED BY \$userpwd DEFAULT TABLESPACE  
\$tablespace ACCOUNT UNLOCK;**

Resposta: Criação de um novo utilizador.

### **Funcionalidade “Permissões Users”**

Ficheiro php: “read\_users.php”

**SELECT \* from DBA\_USERS; ou**

**SELECT \* from DBA\_USERS WHere USERNAME like \$name;**

Resposta: Obtém a lista dos utilizadores.

Ficheiro php: “get\_tipos\_privilegios.php”

**SELECT \* FROM system\_Privilege\_map WHERE property=0 ORDER BY name;**

Resposta: Obtém todos os tipos de permissões que podem ser atribuídos a um user.

Ficheiro php: “grant\_users.php”

**GRANT \$perm TO \$user ;**

Resposta: Faz um Grant das permissões ao user.

## **Tab “Tarefas”**

### **Funcionalidade “Listar Tarefas”**

Ficheiro php: “get\_tarefas.php”

**SELECT \* FROM tarefa WHERE concluido LIKE \$cod ORDER BY tarefa\_id  
DESC;**

Resposta: Obtém uma lista de todas as tarefas que se encontram só pendentes ou só as terminadas.

## 9 – Classes

### 9.1 - TFC.java

**Ficheiros layout** : `Createconnection.xml`

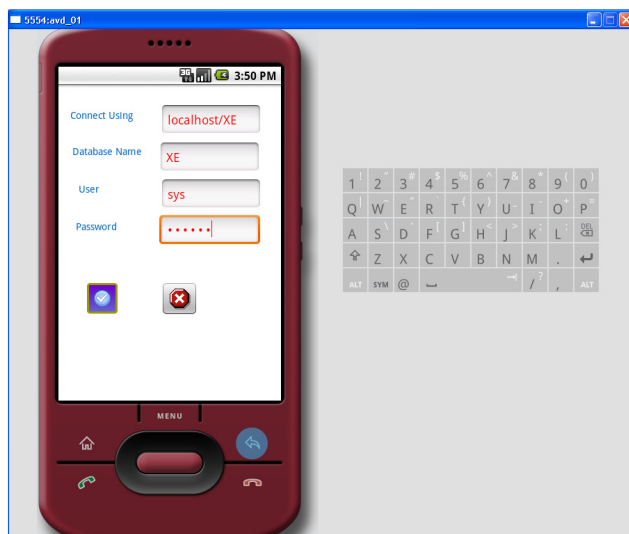
**Objectivo** : Pedir as credenciais ao utilizador e validar no caso de serem válidas e cria variáveis globais.

**Variáveis externas:**

`Glob_UserName` – nome utilizador  
`Glob_Database` – nome do database  
`Glob_ConnectAs` – nome da ligação

**Ficheiros PHP:**

`autenticar.PHP`



**Fig. 9 – Ecrã Principal**

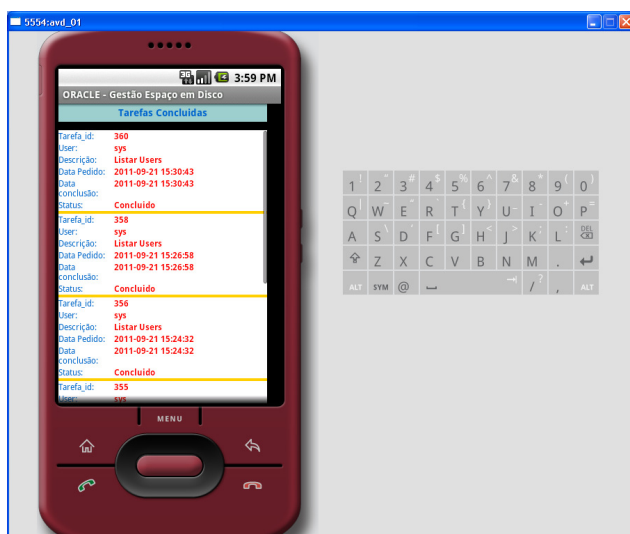
## 9.2 - DisplayAlertas.java

**Ficheiros layout** : `displayalertas.xml`

**Objectivo** : Fazer o display das tarefas terminadas que foram iniciadas durante a última execução da aplicação.

**Ficheiros PHP:**

`get_alertas.PHP`



**Fig. 10 – Ecrã de tarefas**



### 9.3 - TFC\_Main.java – main class

**Ficheiros layout :**

[main.xml](#) - layout ecrã principal

[menu\\_tfc.xml](#) - layout menu

**Objectivo :** Rotina principal que dá acesso a todas as funcionalidades da aplicação.

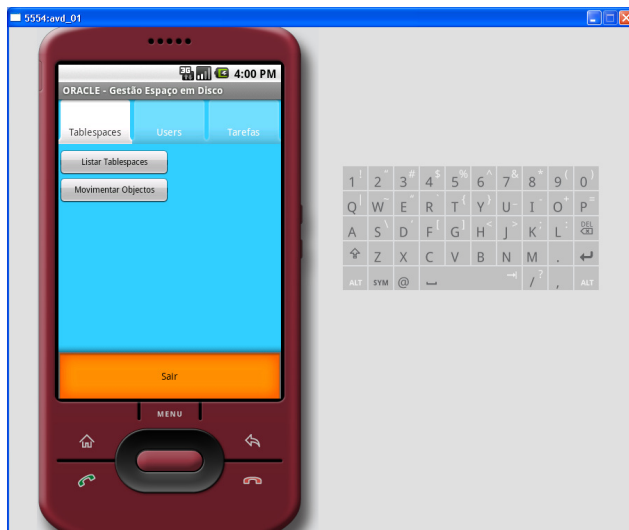
#### 9.3.1 - Descrição Tabs/Botões e menu

##### 9.3.1.1 - Tab “Tablespaces”

Trata de todos os processos relativos às tablespaces.

Botão - Listar Tablespaces - invoca a classe TableSpaces.java

Botão - Movimentar Objectos - invoca a classe  
MovimentarTableSpace.java.



**Fig. 11 - Tablespaces**

### 9.3.1.2 - Tab “Users”

Gere todos os processos relativos à manutenção dos utilizadores

- |                          |                                   |
|--------------------------|-----------------------------------|
| Botão – Listar Users     | - invoca a classe Users.java      |
| Botão – Criar Users      | - invoca a classe CriarUsers.java |
| Botão – Permissões Users | - invoca a classe CriarUsers.java |

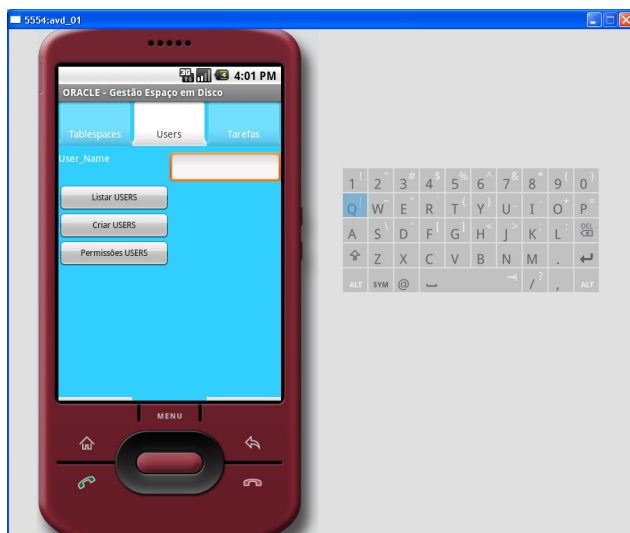


Fig. 12 - Users

#### 4.3.1.2.1 – Listar users

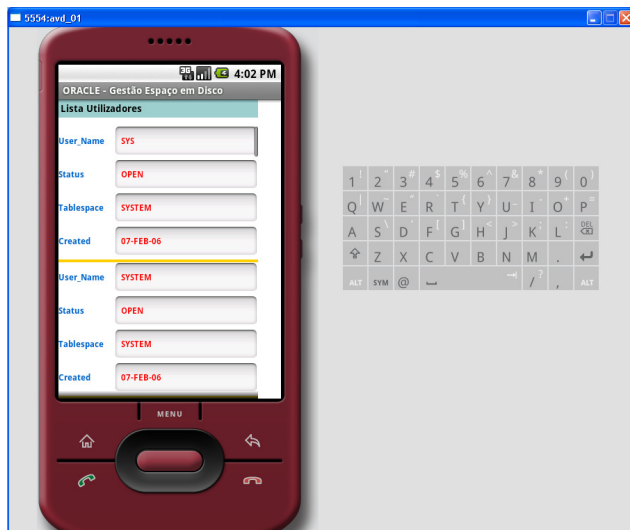
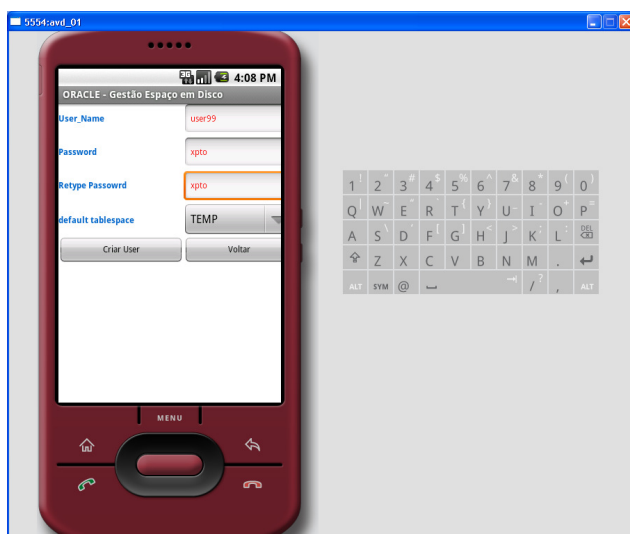


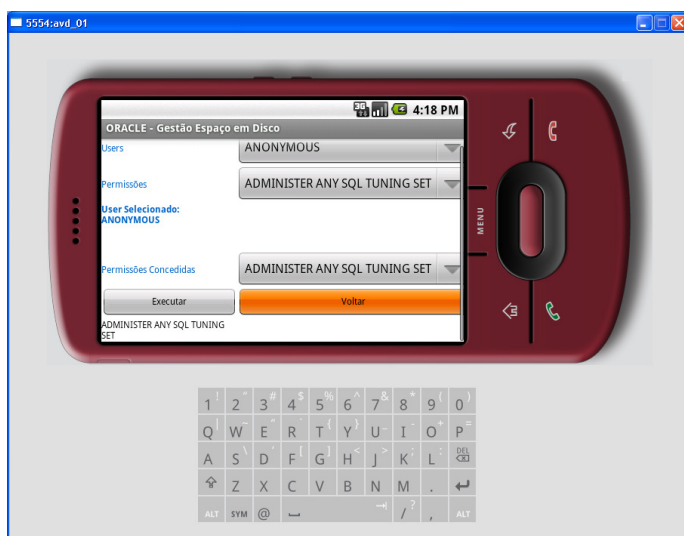
Fig. 13 - Listagem de users com máscara %

#### 4.3.1.2.2 – Criar users



**Fig. 14 - Criar users**

#### 4.3.1.2.3 – Permissões users

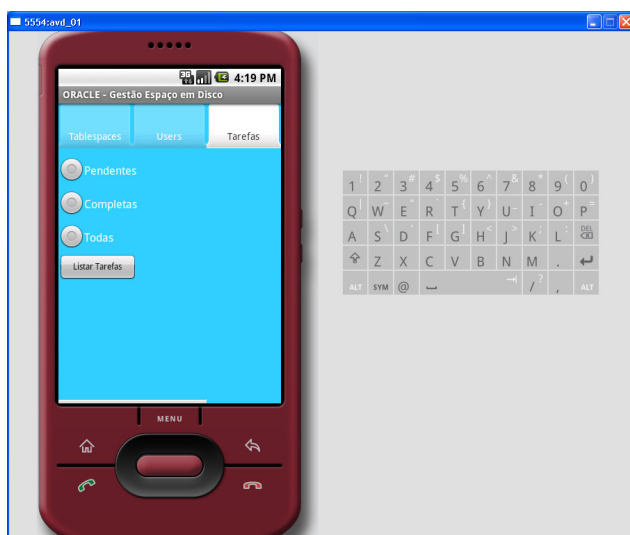


**Fig. 15 - Permissões users**

### 9.3.1.3 - Tab “Tarefas”

Consulta do log das operações, permite listar as operações efectuadas segundo o seu estado.

Botão – Listar Tarefas                      - Invoca a classe Tarefas.java



**Fig. 16 - Tarefas**

## 9.4 - TableSpaces.java

Ficheiros layout :

lista\_ts.xml - layout ecrã principal

row\_df.xml - details view

**Objectivo** : Lista todos os tablespaces e permite a selecção de um apresentando todos os datafiles que o compõem.

**Ficheiros PHP:**

read\_tablespaces.PHP

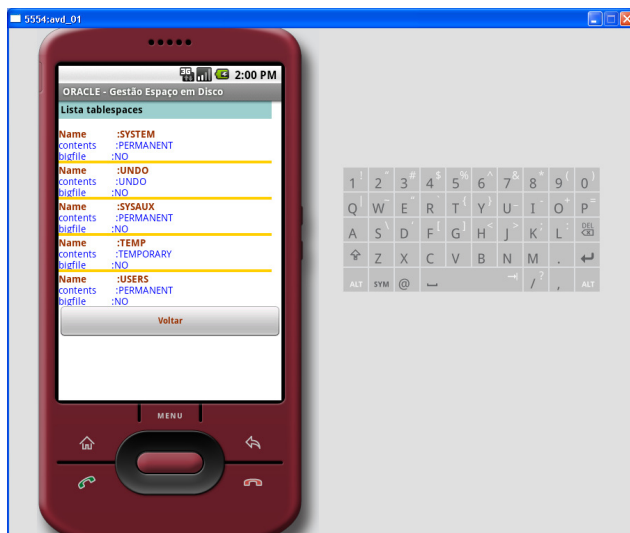


Fig. 17 - Lista dos tablespaces

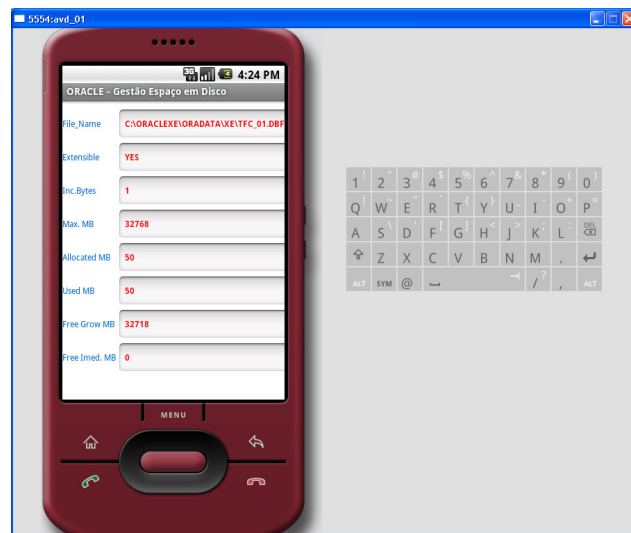
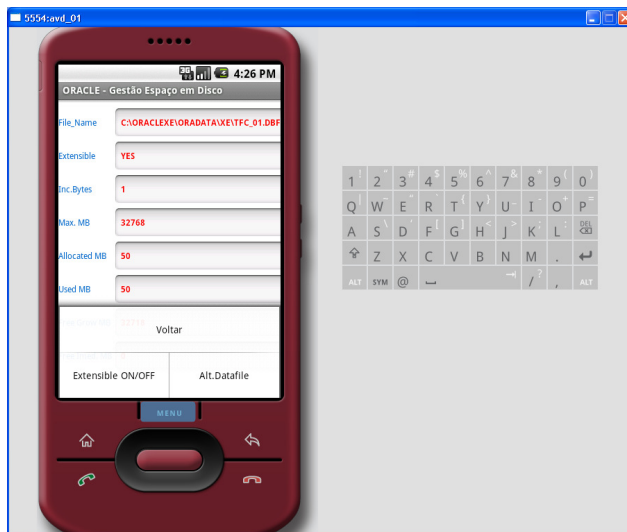


Fig. 18 - Lista dos datafiles

Quando se clica num tablespace é invocada a classe listaDataFiles.java que mostra os datafiles do tablespace em questão e permite efectuar as seguintes operações:



Botão “alt.Datafile” invoca a classe AlterarDataFile.java que permite a alteração do tamanho do datafile.

Botão “Extensible OFF” Permite alterar o atributo Extensible ON para OFF, este atributo passa automaticamente para ON quando se altera a dimensão do datafile.

**Fig. 19 - Display dos datafiles de um dataspace**

## 9.5 - AlterarDataFile.java

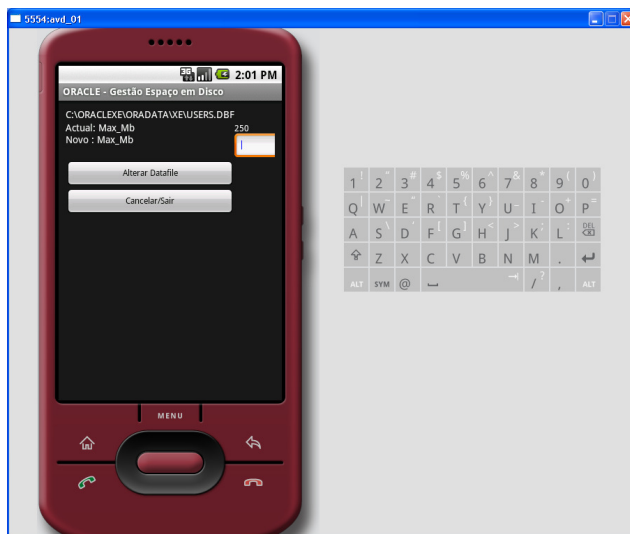
**Ficheiros layout :**

[alterardatafile.xml](#) - layout ecrã principal  
[menu\\_datafiles.xml](#) - details view

**Objectivo :** Faz o display do valor actual “Max\_Mb” do datafile e pede um novo valor do incremento do datafile.

**Ficheiros PHP:**

[alterar\\_datafile.PHP](#)



**Fig. 20 - Alterar datafile**

Botão “Alterar Datafile” – quando premido, pede a confirmação e no caso de ser afirmativa, faz um “post” ao servidor WAMP através de um “form” em PHP que executa os sql statment (descritos em baixo), que alteram o espaço em disco ocupado pelo datafile.

“ALTER DATABASE DATAFILE ‘C:\ORACLExe\oradata\xe\USERS.dbf’ MAXSIZE 250 M”

“ALTER DATABASE DATAFILE ‘C:\ORACLExe\oradata\xe\USERS.dbf’  
AUTOEXTEND ON NEXT 512 MAXSIZE 250 M”

## 9.6 - Users.java

### Ficheiros layout :

- `users.xml` - layout ecrã principal
- `menu_users.xml` - menu view
- `users_dialog.xml` - opções de diálogo

**Objectivo** : Permite listar todos os utilizadores ou um utilizador em particular e conhecer as suas opções.

Na textbox “user” podemos usar máscara (%) ou o nome exacto do utilizador, no caso de utilizarmos máscara faz o display de todos os users correspondentes.

### Ficheiros PHP:

`read_users.PHP`

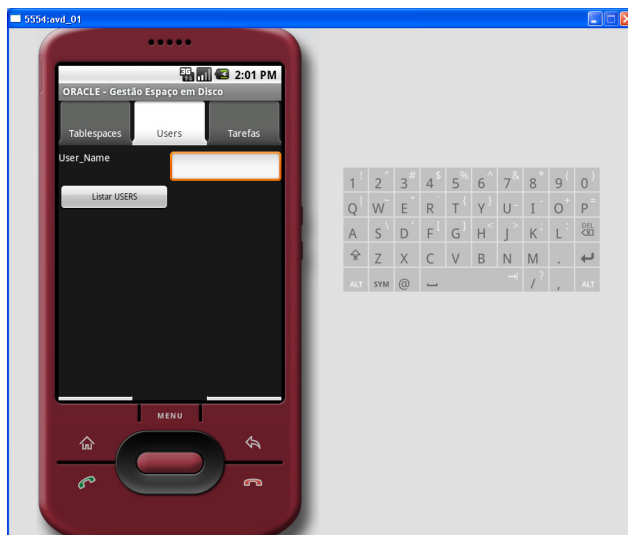


Fig. 21 - Escolha user a listar “%” todos

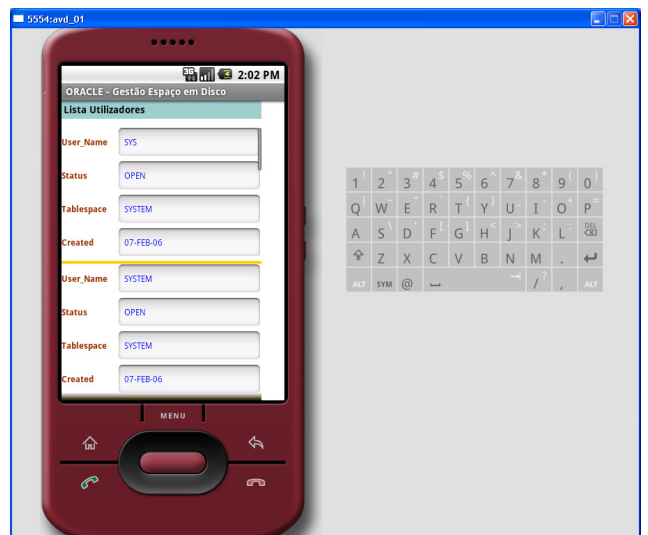


Fig. 22 - Display user(s) escolhido(s)



## 9.7 - CriarUsers.java

### Ficheiros layout :

[criar\\_users.xml](#)

- Layout Criação Users

[criar\\_users\\_grant.xml](#)

- Layout Permissões Users

**Objectivo** : Criar utilizadores na base de dados e dar permissões aos mesmos.

Esta classe executa duas funcionalidades consoante o parâmetro que recebe da classe que a chamou, assim se receber a variável externa TipoTarefa = “Grant” cria as permissões dos utilizadores, se receber “User” cria um novo utilizador.

### Ficheiros PHP:

[criar\\_users.PHP](#)

[get\\_tipos\\_privilegios.PHP](#)

[read\\_tablespaces.PHP](#)

[grant\\_users.PHP](#)

### 9.7.1 - Criar users

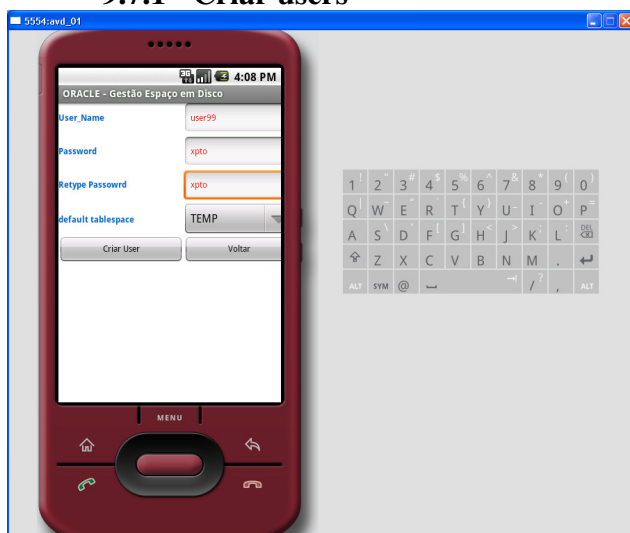
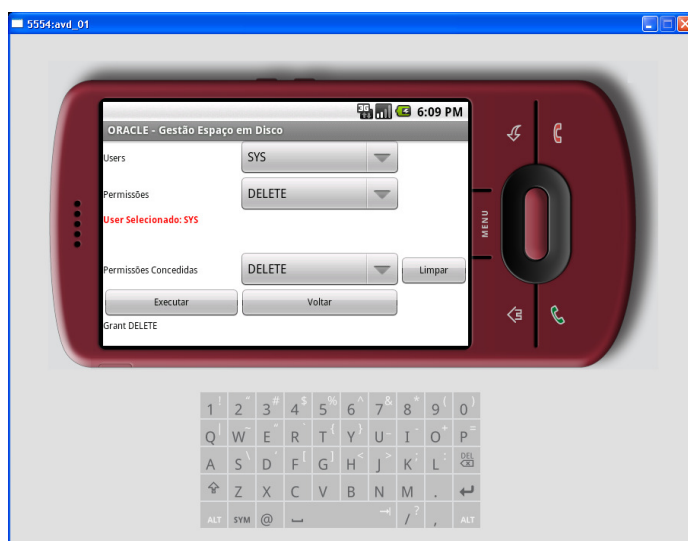


Fig. 23 - Criar users

## 9.7.2 - Permissões users



**Fig. 24 - Permissões users**

## 9.8 - Tarefas.java

**Ficheiros layout :**

`tarefas.xml` - layout ecrã principal  
`menu_tarefas.xml` - menu view

**Objectivo :** Permite listar todas as tarefas, executadas ou pendentes de execução.

**Ficheiros PHP:**

`get_tarefas.PHP`

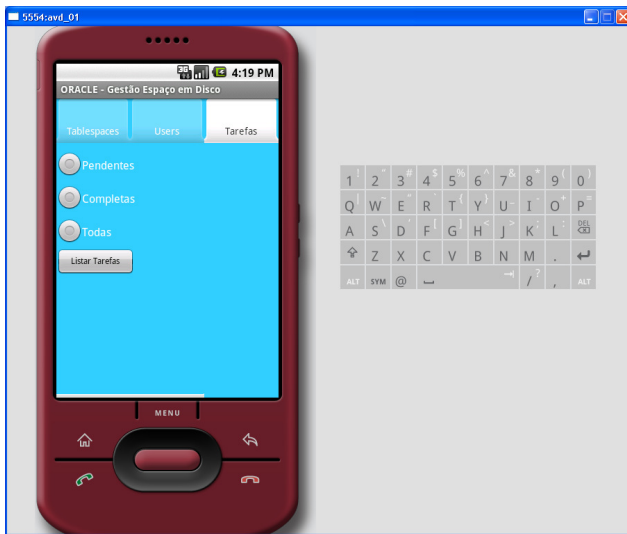


Fig. 25 – Escolha Tarefa por Estado

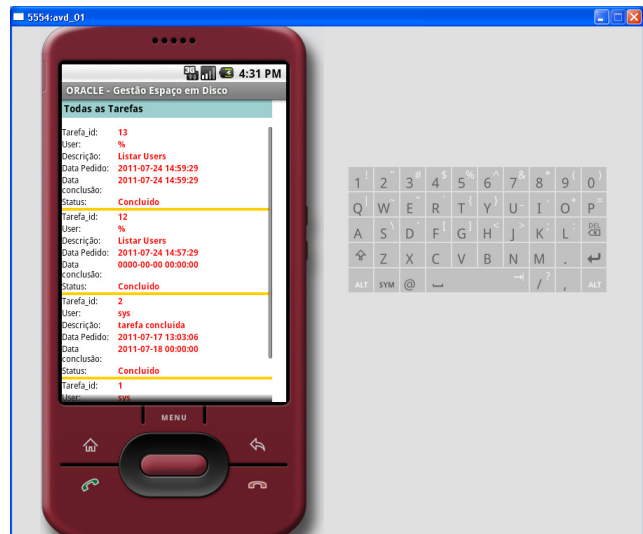


Fig. 26 - Display de Tarefas

## 9.9 – MovimentarTableSpaces.java

**Ficheiros layout :**  
[movimentar.xml](#) - Layout Criação Users

**Objectivo :** Permite mover tabelas e índices entre tablespaces.

**Ficheiros PHP:**  
[get\\_objectos.PHP](#)  
[read\\_tablespaces.PHP](#)  
[read\\_users.PHP](#)  
[mover\\_tabelas.PHP](#)

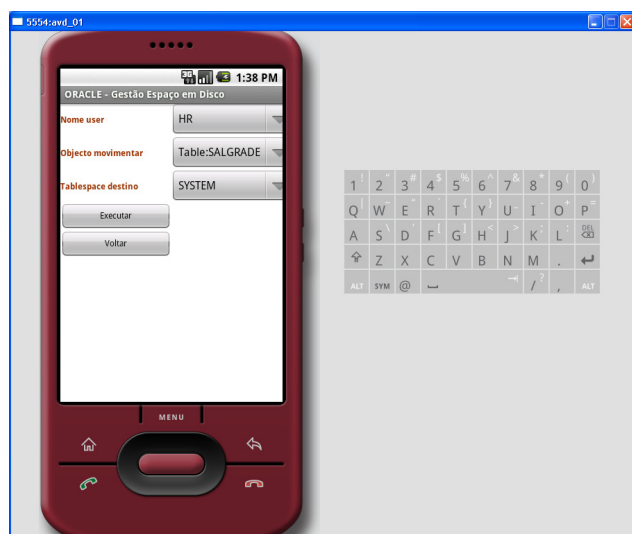


Fig. 27 - Mover Tablespaces

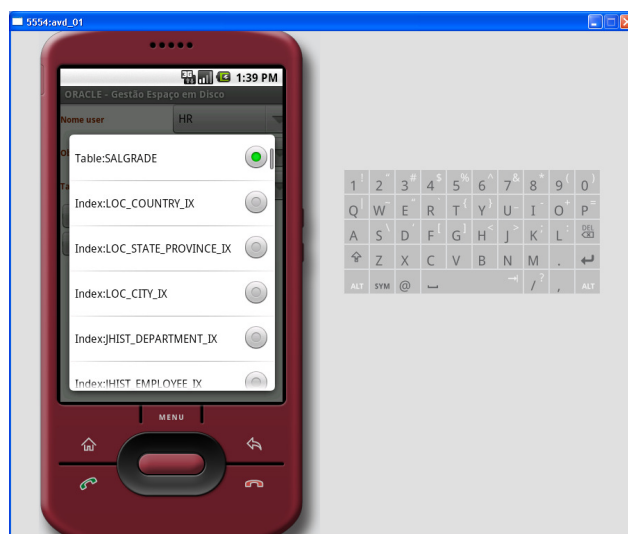


Fig. 28 - Display de opções

## 10 – Índice Figuras

Fig. 1 - Diagrama Geral da Aplicação .....	4
Fig. 2 - Duração Temporal de Actividades.....	5
Fig. 3 – Estrutura Física do <i>ORACLE</i> .....	7
Fig. 4 - Caso Uso de Gestão Espaço em Disco Oracle.....	8
Fig. 5 - Diagrama de Actividades.....	9
Fig. 6 - Diagrama de Sequência de Autenticação.....	9
Fig. 7 - Diagrama de Sequência de Queries Longos .....	10
Fig. 8- Paradigma de Prototipagem .....	12
Fig. 9 – Ecrã Principal .....	23
Fig. 10 – Ecrã de tarefas .....	24
Fig. 11 - Tablespaces .....	25
Fig. 12 - Users .....	26
Fig. 13 - Listagem de users com máscara % .....	26
Fig. 14 - Criar users .....	27
Fig. 15 - Permissões users .....	27
Fig. 16 - Tarefas .....	28
Fig. 17 - Lista dos tablespaces.....	29
Fig. 18 - Lista dos datafiles .....	29
Fig. 19 - Display dos datafiles de um dataspac.....	30
Fig. 20 - Alterar datafile .....	31
Fig. 21 - Escolha user a listar “%” todos.....	32
Fig. 22 - Display user(s) escolhido(s).....	32
Fig. 23 - Criar users .....	33
Fig. 24 - Permissões users .....	34
Fig. 25 – Escolha Tarefa por Estado .....	35
Fig. 26 - Display de Tarefas .....	35
Fig. 27 - Mover Tablespaces .....	36
Fig. 28 - Display de opções .....	36

## 11 – Bibliografia

Professor Alexandre Pereira Banha – **Artigo Android**

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

LORENZO, José Aser - **Administração de Bases de Dados Oracle para programadores – Gestão de Espaço em Disco**. Citeforma.

PRESSMAN, Richard – Software Engineering: a practitioner's approach. Sixth Edition. Edição Internacional, 2005

<http://developer.android.com>

<http://stackoverflow.com/>

## 12 - Índice Remissivo

<i>Android</i> .....	6, 13, 15, 38
<i>datafile</i> .....	13, 17, 19, 20, 30, 31
<i>datagrid</i> .....	13
<i>DBA</i> .....	17, 19, 20, 21
<i>DataBase Administrators</i> .....	6, 11, 13, 14, 15
<i>Grant</i> .....	14, 22, 33
<i>Java</i> .....	4, 38
<i>MySQL</i> .....	6, 15, 16, 17
<i>ORACLE</i> .....	1, 4, 5, 6, 7, 11, 13, 14, 15, 17, 18, 19
<i>PHP</i> ... ..	4, 5, 6, 15, 17, 19, 23, 24, 29, 31, 32, 33, 35, 36
<i>Post</i> .....	15, 17
<i>SGBD</i> .....	7
<i>SQL</i> .....	15, 16, 17, 19
<i>UML</i> .....	9
<i>WAMP</i> .....	31
<i>WAMPSEVER</i> .....	15