



UNIVERSIDADE DE ÉVORA

## **ESCOLA DE CIÊNCIAS E TECNOLOGIA**

DEPARTAMENTO DE INFORMÁTICA

### **O DBA Atual: Desafios e Soluções**

Manuel Joaquim Monteiro Ferreira

Orientação: Prof<sup>a</sup> Irene Pimenta Rodrigues

**Mestrado em Engenharia Informática**

Dissertação

Évora, 2018



UNIVERSIDADE DE ÉVORA

# **O DBA Atual: Desafios e Soluções**

Manuel Joaquim Monteiro Ferreira

Orientação: Prof<sup>a</sup> Irene Pimenta Rodrigues

**Mestrado em Engenharia Informática**

Dissertação

## Sumário

A área de sistemas da *Liberty Seguros S.A.*, é responsável, entre tantas outras plataformas, pela gestão das bases de dados Oracle do negócio em todos os seus vetores de intervenção, desde a segurança ao desempenho.

Atualmente no mercado e não apenas na nossa estrutura é solicitada a disponibilidade do serviço em modo 24 horas por dia, 7 dias por semana. Apesar de existirem equipas que acompanham o funcionamento da plataforma em permanência, essas equipas não têm todas as competências necessárias nem as ferramentas para efetuarem uma série de operações à distância. No âmbito em questão e fazendo eu parte desta equipa, pretendi criar uma aplicação que responda aos desafios que esta gestão coloca no dia-a-dia à nossa equipa e simultaneamente introduzir a possibilidade de alargar a solução a outras plataformas de base de dados.

## *The current DBA: challenges and solutions*

# Abstract

The infrastructure team of *Liberty Seguros S.A.* is responsible, among many other platforms, for managing the Oracle databases supporting the business, in all its vectors of intervention, from security to performance.

Currently in the market, and not only in our structure, the demand for availability of the service requested is 24 hours a day, 7 days a week. Although there are teams that accompany the operation of the platform at all times, these teams do not have all the necessary skills nor the tools to perform a series of remote operations. In the scope in question and being part of this team, we intend to create an application that responds to the challenges that this management puts in our day-to-day basis and simultaneously introduce the possibility of extending the solution to other database platforms.

## Acrónimos

**BD** – Base de dados

**BI** – Business Intelligence

**CPU** – Central Processing Unit

**DBMS** – Database Management System

**ETL** – Extract, Transform and Load

**ES6** – ECMAScript

**GDPR** – General Data Protection Regulation

**JS** - JavaScript

**JSON** – JavaScript Object Notation

**JSX** – JavaScript Syntax eXtension

**LDAP** – Lightweight Directory Access Protocol

**RDBMS** – Relational Database Management System

**REST** – Representational State Transfer

**SQL** – Structured Query Language

**LUW** – Linux Unix Windows

**UX** – User Experience



O DBA Atual: Desafios e Soluções

Universidade de Évora

Departamento de Informática

---

**UI** – User Interface

**XML** – Extensible Markup Language

# Lista de Conteúdo

Sumário .....	i
Abstract .....	ii
Acrónimos .....	iii
Lista de Conteúdo .....	v
Lista de Figuras .....	vii
1. Introdução .....	1
2. Gestão Remota de Bases de Dados .....	3
3. Estado da Arte .....	7
3.1 <i>Oracle Cloud Control</i> .....	7
3.2 <i>Microsoft SQL Server Management Studio (SSMS)</i> .....	8
3.3 <i>IBM Data Server Manager</i> .....	8
3.4 Servidores aplicativos ( <i>Backend</i> ) .....	9
3.5 Bases de dados .....	10
3.6 <i>Frontend</i> .....	11
4. A aplicação – <i>DBA Dashboard</i> .....	13
4.1 <i>Login Screen</i> .....	18
4.2 <i>Database Overview</i> .....	20

---

4.3 <i>Manage Database</i> .....	20
4.3.1 <i>Storage</i> .....	21
4.3.2 <i>Security</i> .....	22
4.3.2.1 <i>Grant Role</i> .....	22
4.3.2.2 <i>List Role</i> .....	22
4.3.2.3 <i>List User Role</i> .....	22
4.3.2.4 <i>Revoke Role</i> .....	23
4.3.3 <i>Sessions</i> .....	23
4.3.3.1 <i>Kill Session</i> .....	23
4.3.3.2 <i>Top Sessions</i> .....	23
4.3.4 <i>Users</i> .....	23
4.3.4.1 <i>Create User</i> .....	24
4.3.4.2 <i>Change User Status</i> .....	24
4.4 <i>Backend</i> .....	24
4.5 Base de Dados .....	25
5. Avaliação.....	29
6. Conclusões e trabalho futuro.....	32
7. Bibliografia .....	34
8. Anexos.....	36



## Lista de Figuras

Figura 1 - Diagrama da aplicação .....	13
Figura 2 - “DBA Dashboard Login” .....	18
Figura 3 - “Login Success” .....	19
Figura 4 - “Database List” .....	37
Figura 5 - “Database Overview” .....	38
Figura 6 - “Main Areas” .....	39
Figura 7 - “Storage Operations” .....	40
Figura 8 - “Security Operations” .....	41
Figura 9 - “Sessions Operations” .....	42
Figura 10 - “Users Operations” .....	43
Figura 11 - “Storage Summary” .....	44
Figura 12 - “Create Tablespace” .....	45
Figura 13 - “Failure Tablespace” .....	46
Figura 14 - “Success Tablespace” .....	47
Figura 15 - “Create Datafile” .....	48
Figura 16 - “Security Summary” .....	49
Figura 17 - “Grant Role” .....	50
Figura 18 - “Failure User Grant Role” .....	51
Figura 19 - “Failure Role Grant Role” .....	52
Figura 20 - “Role atribuído com sucesso” .....	53
Figura 21 - “List Role” .....	54
Figura 22 - “List User Role” .....	55
Figura 23 - “Revoke Role Screen” .....	56
Figura 24 - “Sucesso Revoke Role” .....	57

---

Figura 25 - “Sessions Summary” .....	58
Figura 26 - “Kill Session” .....	59
Figura 27 - “Failure Kill Session” .....	60
Figura 28 - “Success Kill Session” .....	61
Figura 29 - “Top Sessions” .....	62
Figura 30 - “Users Summary” .....	63
Figura 31 - “Create User” .....	64
Figura 32 - “Failure Create User” .....	65
Figura 33 - “Success Create User” .....	66
Figura 34 - “Change User Status” .....	67
Figura 35 - “Success Enable User” .....	68
Figura 36 - “Failure Change User” .....	69
Figura 37 - “Success Disable User” .....	70

# Capítulo 1

## 1. Introdução

Neste capítulo é evidenciado o quê e o porquê desta dissertação passando pelos principais pontos de motivação e expectativas de obtenção de resultados.

Atualmente a quantidade de dados que é recolhida e registada numa BD cresce a um ritmo quase exponencial o que requer vigilância constante sobre o crescimento de uma base de dados.

Exige-se que estas bases de dados tenham alta disponibilidade e desempenho. Que o acesso às mesmas seja controlado e auditado de uma forma cada vez mais exigente como podemos verificar pelo muito discutido GDPR com data final de implementação a 25 de maio de 2018 [1].

É esperado que estas bases de dados estejam disponíveis 24 horas por dia, 7 dias por semana. Para além de ser necessário manter todas as BDs de produção, é necessário manter todas as que suportam o ciclo de vida de produção. Este imenso número de exigências transporta uma grande pressão para as equipas, exigindo-lhes um contacto permanente com o sistema e uma rapidez de resposta eficaz para fazer frente a todas as solicitações. Neste sentido surge a necessidade de criar uma aplicação para dispositivos

móveis que permita endereçar alguns destes problemas e agilizar as operações.

Nesta tese foi desenvolvida uma aplicação para dispositivos móveis que visa o seguinte:

- Facilitar a intervenção dos elementos da equipa adicionando rapidez e eficiência aos processos de gestão;
- Tornar transparente a gestão da BD, removendo grande parte do aspeto técnico da mesma;
- Aproximar a equipa dos DBMS, permitindo que a distância ou dispositivo utilizado não seja um fator que cause entropia;
- Gerir bases de dados independentemente do ambiente em que estejam;
- Dotar a equipa de sistemas com ferramentas de gestão remota;
- Permitir a gestão de mais do que uma plataforma de DBMS;
- Uniformizar procedimentos ao estandardizar operações;
- Independência do sistema operativo.

## Capítulo 2

### 2. Gestão Remota de Bases de Dados

Num cenário atual, podemos estar sentados no sofá e comprar um produto de uma empresa americana para entregar em nossa casa e pagar na hora sem sequer nos levantarmos. Todo este processo é suportado por bases de dados, sejam elas de clientes ou de produtos. Operação que gera dados que serão colocados ainda em mais bases de dados, sejam de BI ou para um qualquer processo ETL.

O mundo está digital e está a crescer. Com o seu crescimento crescem também as bases de dados, em dimensão e em número. É hoje normal uma empresa de dimensão média ter dezenas de bases de dados com requisitos diferentes e com fins diversos.

O papel de administrador de bases de dados é cada vez mais distribuído, tendo de agir sobre uma multiplicidade muito grande de bases de dados e para todas ter de desempenhar as mesmas funções, todas elas com um nível elevado de responsabilidade, apesar de em alguns casos não terem o mesmo nível de prioridade. Para todas as bases de dados será necessário:

- A gestão do espaço e do crescimento da base de dados;
- A criação e/ou modificação dos objetos (tabelas, índices, etc...) que as equipas de desenvolvimento desenham;

- Garantir a segurança dos dados através da criação, controlo e monitorização de acessos, assim como a sua correta atribuição aos utilizadores;
- Acompanhar o desempenho da base de dados, procurando pontos de otimização;
- Evoluir o software da base de dados, assegurando que obedece às obrigações de *compliance* instituídas;
- Assegurar a continuidade de negócio através da criação de estratégias de salvaguarda e reposição da base de dados.

Para além de garantir todas as funções sobre as bases de dados, o administrador das bases de dados vê-se também confrontado com os desafios que uma grande empresa com presença nos mercados da aldeia global lhe apresenta. Como é o caso de manter bases de dados a funcionar em 24 horas por dia e 7 dias por semana, com diminutas janelas de manutenção e garantir o "*follow the sun*". Ou seja, a qualquer momento tem de ser garantido o suporte às bases de dados independentemente do local geográfico onde estão e, conseqüentemente, da hora do dia em que se encontram.

Esta exigência do negócio global vem colocar ainda mais *stress* sobre as equipas de administração já que se espera que as aplicações funcionem em permanência e que qualquer necessidade seja resolvida de imediato.

Para responder às pressões de funcionamento de 24 horas por dia e 7 dias por semana são criadas equipas que estejam disponíveis a qualquer momento, com as mesmas capacidades técnicas. Este crescimento das equipas acarreta maiores custos e menor agilidade das equipas, dada a dimensão das mesmas. Torna-se assim importante que se distribua esta

carga por outras equipas, delegando e simplificando algumas das tarefas que estas fazem e permitindo que as façam a qualquer momento.

Precisamente com o objetivo de simplificar e delegar funções em outros elementos de outras equipas, como a equipa de Operação, são disponibilizadas ferramentas centrais de gestão de bases de dados. Exemplos disso são o *"Enterprise Manager Cloud Control"* da Oracle, o *"SQL Server Management Studio"* da Microsoft ou o *"Data Server Manager"* da IBM. Estes sistemas permitem monitorizar e controlar toda a infraestrutura de bases de dados, desde a segurança ao desempenho, fornecendo inclusive algumas ferramentas de desenvolvimento.

É agora necessário tornar essas ferramentas acessíveis em qualquer parte, libertando os seus utilizadores dos constrangimentos de rede ou suporte físico (computador portátil) e fazendo usufruto dos seus aparelhos móveis, sejam *tablets* ou *smartphones*.

Este novo método de acesso trará agilidade às equipas e reduzirá o tempo de resposta para os problemas mais comuns e que não carecem de acompanhamento particular.

O recurso a tecnologias direcionadas para os dispositivos móveis, combinadas com as ferramentas já disponíveis no mercado, pode trazer essa flexibilidade e alavancar o trabalho das equipas de gestão das bases de dados, libertando tempo e recursos para outras tarefas com maior exigência de análise.

Tarefas que visem a manutenção de utilizadores, o ajuste de armazenamento ou a identificação de uma *"runaway query"* podem ser efetuadas através de

tais ferramentas, dado o pouco impacto que a sua implementação acarreta para a base de dados e não necessitarem de uma análise aprofundada.



## Capítulo 3

### 3. Estado da Arte

Muitos dos DBMS existentes no mercado têm consolas centrais de gestão. No entanto, nenhuma fornece, pelo menos "out of the box" a capacidade de gestão dessas bases de dados através de uma aplicação para dispositivo móvel.

Para as bases de dados proprietárias com maior dimensão como é o caso de *SQL Server*, *Oracle* e *DB2* temos as ferramentas abaixo.

#### 3.1 *Oracle Cloud Control*

O *Oracle Cloud Control* é uma plataforma centralizada de gestão de produtos *Oracle*. Permite gerir todas as bases de dados existentes em máquinas onde o agente do *Cloud Control* esteja instalado. O controlo das bases de dados é total mediante a apresentação de credenciais adequadas para a base de dados em questão, mas também de credenciais para o servidor onde está alojada a base de dados. Estas últimas são necessárias por exemplo, caso seja necessário reiniciar a BD. Dada a sua abrangência, uma vez que pode gerir outro *software* relacionado com BDs ou não (gestão de ciclo de vida aplicacional, gestão de *middleware*, etc... [7]), trata-se de um

*software* que consome bastantes recursos, é bastante complexo e tem uma curva de aprendizagem algo extensa.

A sua utilização é efetuada via *browser* o que lhe confere alguma agilidade já que não requer instalação de cliente nas máquinas dos utilizadores. Não disponibiliza aplicação para dispositivos móveis.

### **3.2 Microsoft SQL Server Management Studio (SSMS)**

O *Microsoft SQL Server Management Studio* é a ferramenta de gestão de bases de dados da *Microsoft*. A sua área de ação passa por toda a infraestrutura do SQL, desde o servidor à base de dados. Como a maioria dos *softwares* de gestão, este também pode monitorizar, administrar e configurar múltiplas instâncias de bases de dados. Fornece também a funcionalidade de criação de *scripts* e *queries*. A ferramenta está preparada para fazer tudo isto não só em bases de dados locais, mas também em base de dados na *Cloud* fazendo uso da infraestrutura *Azure* da *Microsoft*. [14]

### **3.3 IBM Data Server Manager**

Semelhante ao *Oracle Cloud Control*, esta consola de gestão centralizada permite administrar e monitorizar o DB2, oferecendo mecanismos de melhoria de desempenho e de gestão centralizada de bases de dados, inclusive sobre várias plataformas LUW (*Linux*, *Windows* e *Unix*) assim como z/OS. [13].

A utilização de aplicações móveis para gestão de plataformas não é algo de novo, existem aplicações móveis para gestão de *hardware*, exemplo disso é a *IBM Storage Mobile Dashboard* [12]. A possibilidade de utilização destas tecnologias sobre bases de dados, atualmente é diminuta.

No caso do *IBM Storage Mobile Dashboard*, é possível verificar o estado e performance da DS8000 através da utilização de métricas e gestão de eventos. Podemos antecipar problemas e verificar outros que já existem. É possível agendar uma intervenção de forma assíncrona ou solicitar ajuda externa (a substituição de hardware em falha é um desses casos).

Não é, no entanto, possível ter uma ação aprofundada sobre o sistema que estamos a gerir. Não é possível, por exemplo, criar uma LUN através da aplicação nos casos em que é necessário estender um determinado volume.

A opção de criar a aplicação descrita nesta dissertação nos moldes em que foi construída prende-se exatamente com esse *handicap* de utilização, uma vez que tenta preencher esse vazio, possibilitando ao utilizador final uma interação integrada com o sistema de base de dados alavancando o potencial da aplicação e permitindo que a aplicação seja efetivamente uma extensão remota das possibilidades existentes num ambiente local.

### **3.4 Servidores aplicativos (*Backend*)**

No mercado existem bastantes alternativas em servidores aplicativos para Java. Nos casos de servidores aplicativos comerciais e sujeitos a licenciamento temos na linha da frente o Weblogic Application Server da

Oracle, o Websphere Application Server da IBM e o JBOSS EAP da RedHat que apesar de ser open source tem associada uma licença para suporte.

Por outro lado, temos servidores aplicativos *open source* que suportam JAVA EE como o Tomcat da Apache e o Glassfish da Oracle.

Há uma imensa lista de possibilidades que tanto o Weblogic como o Websphere disponibilizam, desde o simples controlo de JMS até possibilidades de *load balancing* e escalabilidade horizontal. Estes servidores possibilitam controlar o mais ínfimo detalhe e ambos necessitam de licenciamento. Obviamente que estes dois fornecedores cumprem os requisitos técnicos para a execução do *backend* desenvolvido.

Também o mais acessível, o Tomcat da Apache, cumpre os requisitos com o atrativo de ter um ecossistema e comunidade muito forte de apoio ao Tomcat. A opção Tomcat foi escolhida como a mais adequada, permitindo manter as premissas da aplicação, leve e acessível.

### **3.5 Bases de dados**

Existe um largo espectro de bases de dados disponíveis e entre as melhores e mais usadas, as opções comerciais Oracle Database da Oracle, o DB2 da IBM e o SQL Server da Microsoft.

Outras opções, open source, são o MySQL, MariaDB e o PostgreSQL. Todas as bases de dados comerciais requerem licença caso não sejam versões express/lite o que acarreta limitações para as mesmas, tais como, espaço máximo ocupado e CPU alocado. A melhor escolha, dado o suporte que

possui e a sua implementação no mundo empresarial, seria a base de dados Oracle. A gestão desta base de dados é bastante complexa e, como tal, exige elevados conhecimentos técnicos, o que no fundo é o que a aplicação desenvolvida no âmbito da dissertação procura simplificar, complexidade essa que não é necessária para o desenvolvimento do projeto. Como alternativa *open source* e dada a sua escalabilidade e fácil integração a MariaDB seria uma escolha de primeira linha, ainda assim a escolha acaba por ficar no MySQL, base de dados génese da qual o MariaDB surgiu após a compra do MySQL por parte da Oracle.

Se por um lado existe um menor apoio da comunidade ao MySQL em detrimento do MariaDB, acabamos por ter um gigante como a Oracle a fornecer esse suporte.

Um outro lado positivo da escolha é a facilidade com que poderemos mudar entre MySQL e MariaDB, uma vez que, a sua génese é a mesma.

### **3.6 Frontend**

Entre as *frameworks/libraries* de desenvolvimento mobile mais utilizadas temos AngularJS, React (Native), Vue e Ionic. Sendo AngularJS a *framework* mais utilizada, tem-se vindo a esbater a diferença para o React (segundo pesquisa no Google Trends).

Apesar de AngularJS ser mais fácil de utilizar do que React, esta última e mais propriamente a sua extensão React Native é mais versátil uma vez que podemos utilizar a mesma base de código para Android e iOS e até mesmo na maioria da lógica para web, o que permite eliminar tempo de

desenvolvimento. Além disso tem um ecossistema maior (graças ao seu pai React). React Native utiliza componentes nativos o que o torna mais rápido por comparação com os seus concorrentes. Para os requisitos em mente, e apesar da curva de aprendizagem ser superior, o lema "learn once, write anywhere" ganha vantagem e, por consequência, o React Native.

## Capítulo 4

### 4. A aplicação – *DBA Dashboard*

#### Desenho da aplicação

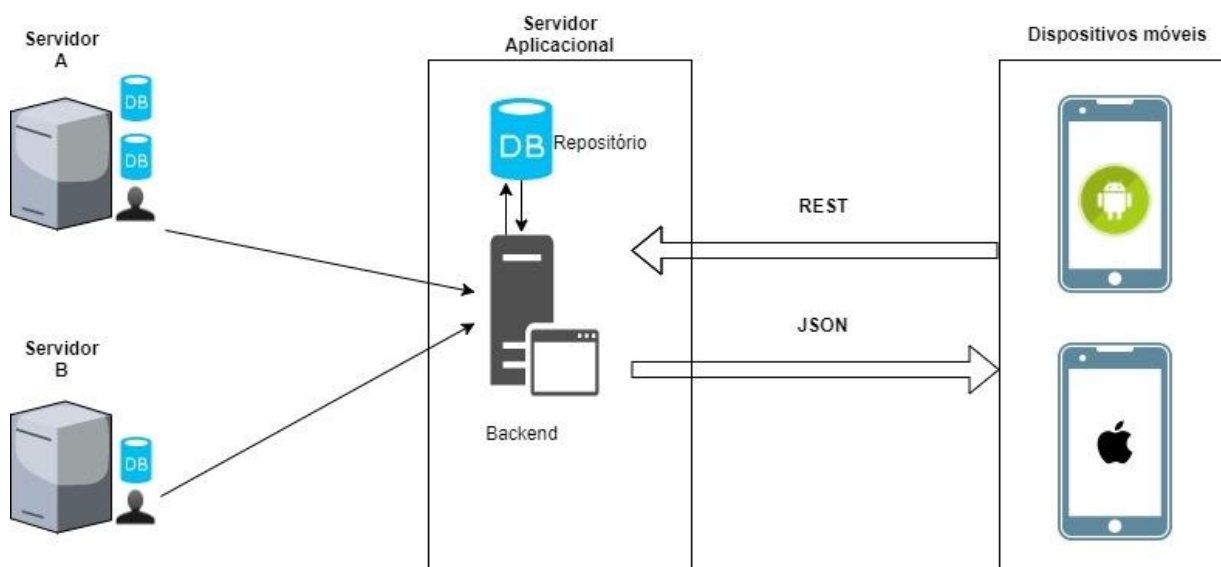


Figura 1 - Diagrama da aplicação

O foco da dissertação reside na aplicação "*DBA Dashboard*" construída com o objetivo de poder efetuar operações sobre bases de dados através de dispositivos móveis. A figura 1 apresenta o diagrama geral de fluxo de dados desta aplicação. A aplicação faz pedidos REST a um servidor aplicacional TOMCAT que tem o suporte de um repositório de dados

construído em MySQL. Após a execução dos pedidos é devolvida uma resposta em JSON para a aplicação móvel com o feedback da operação.

Para alcançar o objetivo descrito foram criados os seguintes componentes:

- Uma base de dados de repositório;
- Um *backend* para servir a aplicação;
- Um *frontend* para disponibilização de interface ao utilizador.

Para o funcionamento da aplicação necessitamos dos 3 componentes acima descritos.

A base de dados criada serve como repositório de informação que será colhida pelos agentes (a criação destes agentes não está contemplada na tese) presentes em cada máquina que aloja BDs. Estes irão recolher informação e popular o nosso repositório. Para além desta informação existe também informação relacionada com a nossa aplicação, como é o caso do controlo de acessos. Este controlo de acessos é referente à aplicação desenvolvida, o que significa que mesmo que exista alguma falha de permissões, possibilitando o acesso a mais operações do que o desejado, a validação dessas mesmas operações sobre a base de dados é sempre efetuada pela mesma contra a lista de privilégios do autor. Para isto assume-se que o utilizador criado na aplicação terá o mesmo *username* na base de dados.

O *backend* tem como função expor uma série de serviços à aplicação. Estes serviços poderão ser simplesmente de consulta de informação, mas também de execução de procedimentos. Todos os métodos estão expostos como serviços REST. Todas as respostas dadas pelo serviço são em formato JSON.



Finalmente, a aplicação disponibiliza através de interface gráfico os métodos que o *backend* nos proporciona.

Para a implementação da aplicação foram utilizadas plataformas que hoje em dia são referência no que se refere a dispositivos móveis. Desde o *backend* até ao *frontend*, a premissa é incluir a mais recente das tecnologias.

### ***React (Native)***

Para implementação da interface da aplicação foi utilizada uma biblioteca que tem uma larga utilização no mercado em empresas de enorme dimensão como o *Facebook*, o *Instagram* e o *Skype*, entre outras [3]. O *React* tem como objetivo efetivo a criação dessas mesmas interfaces de utilizador. [2]

O *React* é uma biblioteca de JS e tem como pontos fortes ser declarativa, eficiente e flexível. Permite criar componentes mais pequenos que gerem o seu próprio estado, possibilitando a construção de elementos mais complexos a partir destes elementos mais básicos que interagem com bibliotecas e *frameworks* já conhecidas.

Partindo das potencialidades do *React* e sendo o objetivo a construção de uma aplicação para dispositivos móveis, foi utilizado mais precisamente o *React Native* que nos permite criar aplicações utilizando apenas JS.

Com *React Native* podemos efectuar a disponibilização da aplicação tanto para IOS como para *Android* sem praticamente alterar código e podemos

também desenvolver mais rápido, uma vez que a alteração de código não implica uma recompilação do projeto, mas apenas um recarregar do mesmo.

Foi utilizada a versão 16 do *React* com a versão 0.50 de *Native*.

## ***Node.JS***

Para a execução do *React* recorri ao *Node.JS* uma vez que é bastante leve e eficiente, é essencial que assim seja para plataformas móveis. Permite também ter acesso a uma vasta coleção de pacotes *open source* feitos pela comunidade. A utilização destas bibliotecas como blocos de construção torna mais ágil a criação de aplicações, uma vez que se poderá tirar partido de otimizações já efetuadas pela comunidade.

## **REST/JSON**

A escolha entre os dois maiores protocolos de comunicação em *web services* recaiu sobre REST+JSON. Principalmente porque, para além de ser mais fácil para qualquer programador entender, logo mais fácil a partilha de conhecimento, é também mais eficiente e não necessita de manter o estado de comunicação com um cliente, permitindo maior flexibilidade e rapidez de utilização.

JSON acaba por ser a escolha óbvia dada a sua ligação ao *JavaScript*. Mais simples de entender que o XML, mais rápido de processar e direcionado para a troca de dados entre um *browser* e um servidor [8][9][10][11].

## ***JavaScript / JSX***

O *React* é uma biblioteca de JS e como tal a lógica da aplicação foi desenvolvida com recurso a esta linguagem, fazendo já uso da sintaxe ES6 como a utilização das declarações *let* e *const* [5].

Para a definição do aspeto da interface foi utilizado a extensão de sintaxe do JS, o JSX tal como requerido pelo *React Native* [6].

## ***Apache TOMCAT***

Foi utilizado o *Tomcat 8* da *Apache* [15] como servidor aplicacional. Para além de estar disseminado pela comunidade e com boa documentação e suporte, não necessita de licenciamento para este fim. Foi ponderada a utilização de *Websphere* e *Weblogic* (utilizado no *Oracle Cloud Control*) no entanto como requerem licenciamento ficaram colocados de lado uma vez que o Tomcat não tem este constrangimento e responde às necessidades do projeto.

## 4.1 *Login Screen*

O ecrã de *Login* apresenta os campos para o nome de utilizador e *password*. Este nome de utilizador e *password* são locais, apenas pertencem à aplicação e não estão ligados a nenhum serviço de LDAP.

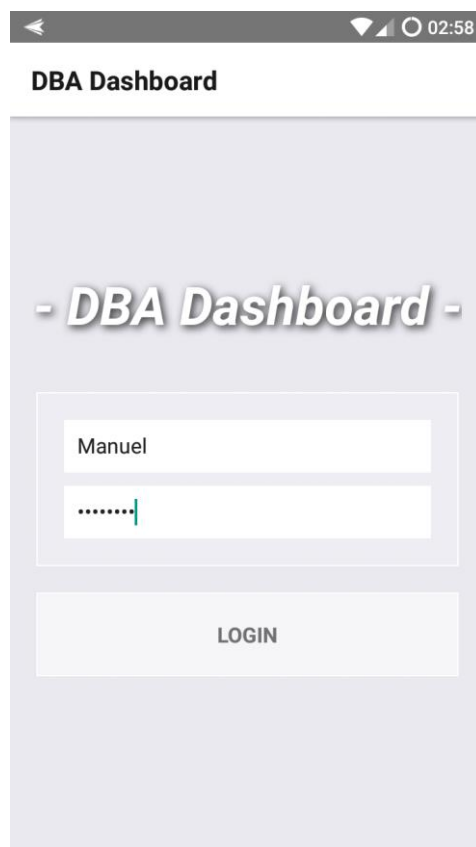
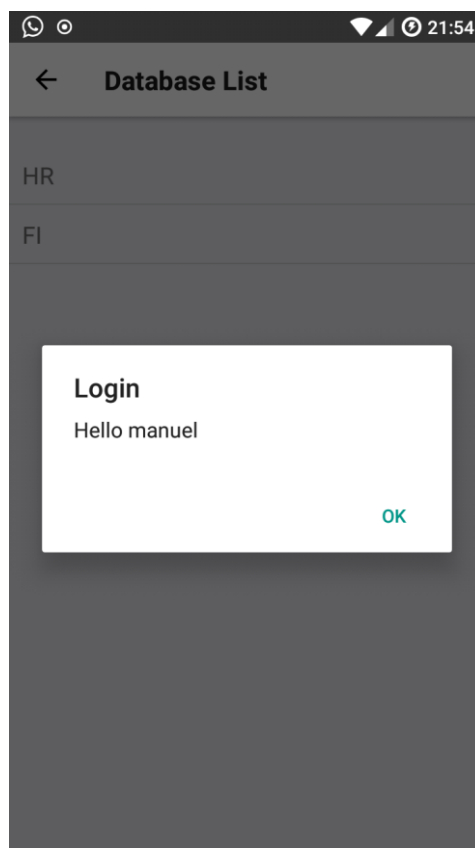


Figura 2 - “DBA Dashboard Login”



**Figura 3 - “Login Success”**

## **4.2 Database Overview**

O ecrã *Database Overview* (figura 4) e *Database List* (figura 5) apresentam a listagem das bases de dados disponíveis para gestão e os dados gerais de cada base de dados no que diz respeito aos recursos de máquina atribuídos.

## **4.3 Manage Database**

O ecrã *Manage Database* apresenta todas as áreas de intervenção sobre uma base de dados disponível na aplicação (figura 6, 7, 8, 9 e 10).

Para cada uma das áreas temos a listagem de operações por área passíveis de utilização na aplicação, nomeadamente:

- *Create Tablespace*
- *Create Datafile*
- *Grant Role*
- *List Role*
- *List User Role*
- *Revoke Role*
- *Kill Sessions*
- *Top Sessions*
- *Create User*
- *Change User Status*

### 4.3.1 *Storage*

O ecrã de *Storage* (figura 11) apresenta o sumário das opções relacionadas com o espaço utilizado pela BD selecionada, nomeadamente:

- Espaço Utilizado;
- Espaço Alocado;
- Espaço Livre;
- Gráfico circular que espelha a ocupação de espaço da BD.

#### 4.3.1.1 *Create Tablespace*

A opção "Create Tablespace" (figura 12, 13 e 14) permite-nos criar um *tablespace* na base de dados selecionada. Tem como input o nome do *tablespace*.

#### 4.3.1.2 *Create Datafile*

A opção "Create Datafile" (figura 15) permite-nos criar um *datafile* na base de dados selecionada. São listados para seleção os *tablespaces* existentes na base de dados.

### 4.3.2 *Security*

O ecrã de *Security* (figura 16) apresenta o sumário das opções relacionadas com a segurança na BD selecionada, nomeadamente:

- Gráfico circular que espelha o número de *roles* da BD e a número de utilizadores que os possuem.

#### 4.3.2.1 *Grant Role*

A operação "Grant Role" (figura 17, 18, 19 e 20) permite atribuir um *role*, existente na BD, a um utilizador.

#### 4.3.2.2 *List Role*

A operação "List Role" (figura 21) lista todos os *roles* existentes na BD.

#### 4.3.2.3 *List User Role*

A operação "List User Role" (figura 22) lista todos os *roles* pertencentes a um determinado utilizador. Caso o utilizador não exista na BD é retornado um erro.



#### **4.3.2.4 Revoke Role**

A operação "Revoke Role" (figura 23 e 24) permite remover um *role* de um utilizador.

#### **4.3.3 Sessions**

O ecrã de *Sessions* (figura 25) apresenta o sumário das opções relacionadas com as sessões existentes na BD selecionada, nomeadamente:

##### **4.3.3.1 Kill Session**

A operação "Kill Session" (figura 26, 27 e 28) permite terminar uma sessão em execução.

##### **4.3.3.2 Top Sessions**

A operação "Top Sessions" (figura 29) lista o top 10 de sessões em execução na BD selecionada.

#### **4.3.4 Users**

O ecrã de "*Users*" (figura 30) apresenta o sumário das opções relacionadas com utilizadores na BD selecionada.

#### **4.3.4.1 *Create User***

A operação "Create User" (figura 31, 32 e 33) permite criar um utilizador na BD seleccionada.

#### **4.3.4.2 *Change User Status***

A operação "Change User Status" (figura 34, 35, 36 e 37) permite desabilitar/habilitar um utilizador na BD seleccionada.

### **4.4 *Backend***

O *backend* foi criado em Java e corre sobre um servidor Tomcat. Com provas dadas, é um servidor aplicacional *open source*, com enorme presença no mercado e um grande ecossistema de *developers*. Foi utilizado o JDK 1.7

Foi desenhado para responder a pedidos REST com resposta em JSON.

É da responsabilidade do *backend* executar todas as tarefas de gestão do repositório e comunicação com os agentes que executarão as tarefas nos clientes, no caso, bases de dados.

## 4.5 Base de Dados

O RDBMS utilizado foi MySQL. Para além de ser *open source* é bastante utilizada e com largo suporte. À data de hoje é detida pela *Oracle* que tem inclusive uma versão otimizada desta BD, a *Aurora* e ainda um *fork* do MySQL, a MariaDB o que diz bem do espectro de utilização do RDBMS.

As tabelas que dão suporte ao repositório da aplicação são as seguintes:

- ***Databases***

A tabela "Databases" consiste em duas colunas, "PK\_ID" e "Name" que correspondem ao identificador numérico e nome da base de dados existentes no repositório da aplicação.

- **DB\_Info**

A tabela "DB\_Info" contém as informações relacionadas com uma base de dados do repositório. Consiste nas colunas "PK\_ID", "Databases\_PK\_ID", "CPU", "Memory" e "Status\_idStatus" que correspondem respetivamente ao identificador numérico da informação, identificador numérico da base de dados, CPU alocado à base de dados, memória alocada à base de dados e o identificador numérico relativo ao estado da base de dados.

- **DF\_Info**

A tabela "DF\_Info" contém as informações relacionadas com os *datafiles* de uma base de dados do repositório. Consiste nas colunas "idDF\_info", "DF\_NAME", "DF\_CURRENT\_SIZE", "DF\_MAX\_SIZE", "DF\_TS", "DF\_DG", "database\_id" e "DF\_USED\_SIZE" que correspondem respetivamente ao identificador numérico da informação, nome do *datafile*, tamanho atual do *datafile*, tamanho máximo do *datafile*, identificador do *tablespace* em que está o *datafile*, identificador do *diskgroup* onde está o *datafile*, identificador da base de dados ao qual o *datafile* pertence e espaço ocupado pelo *datafile*.

- **DG\_Info**

A tabela "DG\_Info" contém as informações relacionadas com os *diskgroups* disponíveis para as bases de dados. Consiste nas colunas "idDG", "DG\_NAME", "Current\_Occupation", "Max\_Occupation" que correspondem respetivamente ao identificador da informação, ao nome do *diskgroup*, à ocupação atual do *diskgroup* e à ocupação máxima do *diskgroup*.

- **TS\_Info**

A tabela "TS\_Info" contém as informações relacionadas com os *tablespaces* de uma base de dados do repositório. Consiste nas colunas "idTS\_Info", "TS\_Name", "TS\_Database" e "TS\_Size" que

correspondem respetivamente ao identificador numérico da informação, ao nome do *tablespace*, ao identificador numérico da base de dados onde o *tablespace* está criado e ao tamanho atual do *tablespace*.

- **Role**

A tabela "Role" contém as informações relacionadas com os *roles* de uma base de dados do repositório. Consiste nas colunas "idRole", "role\_Name" e "Role\_Database" que correspondem respetivamente ao identificador do *role*, ao *nome* do role e ao identificador numérico da base de dados à qual o *role* pertence.

- **Session\_Info**

A tabela "Session\_Info" contém as informações relacionadas com as sessões existentes numa base de dados. Consiste nas colunas "idSession\_Info", "Session\_ID", "Session\_Time", "Session\_SQLID", "Session\_User" e "database\_id" que correspondem respetivamente ao identificador da informação, ao identificador da sessão, ao tempo de execução da sessão no "*sqlid*" corrente, o "*sqlid*" em utilização na sessão, ao utilizador da sessão e ao identificador da base de dados onde a sessão se encontra em execução.

- **Status**

A tabela "Status" contém as informações relativas a um estado. Consiste nas colunas "idStatus" e "Status\_Description" que correspondem respetivamente ao identificador numérico da informação e à descrição do estado.

- **User**

A tabela "User" contém as informações relativas a um utilizador da base de dados. Consiste nas colunas "name", "username", "password", "register\_dt", "database\_id" e "status" que correspondem respetivamente ao nome, *login*, palavra passe, data de registo, identificador da base de dados e estado do utilizador.

- **User\_Role**

A tabela "User\_Role" contém as informações relativas ao *roles* atribuídos aos utilizadores. Consiste nas colunas "idUser\_Role", "idRole" e "username" que correspondem respetivamente ao identificador da informação, identificador do *role* e *login* do utilizador.

## Capítulo 5

# 5. Avaliação

Para avaliar a efetividade da aplicação, foram considerados dois tópicos. A performance e a utilização em condições reais.

Para efeitos de performance suportamos a avaliação sobre duas métricas: os tempos de resposta e a carga.

As medições dos tempos de resposta e carga acabam por ser despidiendos, uma vez que, no que se refere a carga, e por inerência de funções, a simultaneidade de utilizadores será mínima. O número de DBAs é reduzido para o caso em análise, assim como o número de elementos de outras equipas que possam utilizar a aplicação. Quanto a tempos de resposta também serão mínimos dado a transferência de dados, a sua execução no servidor aplicacional também o será, uma vez que, o tempo irá depender mais da execução da tarefa na base de dados final do que no servidor aplicacional. Exemplo disso será a criação de um *datafile*, operação simples do ponto de vista da sua chamada, mas que potencialmente terá um tempo elevado de execução no lado da base de dados, uma vez que, consoante o tamanho do *datafile* com o espaço inicial alocado, esta operação poderá ser mais demorada.

Como exemplos de utilização temos 3 cenários:

### **Cenário 1.**

Um processamento massivo de dados está a encher o *tablespace* temporário. De modo a que o processo responsável por esta carga não termine por falta de espaço, o que no limite poderia implicar o *rollback* de todo o processo, é necessário acrescentar um *datafile*. Uma vez que a equipa de administração não está disponível, a utilização da aplicação possibilita que essa criação seja efetuada no imediato reduzindo o risco de cancelamento do processo

### **Cenário 2.**

Um dos elementos da equipa de administração está a efetuar uma manutenção programada na base de dados e bloqueia o seu utilizador.

É imprescindível que o desbloqueie o mais rápido possível, uma vez que, a janela de manutenção é reduzida. Um dos elementos de prevenção ainda se encontra nos transportes públicos e sem a sua máquina por perto. Após contacto, pode aceder ao perfil do utilizador bloqueado e torná-lo disponível novamente, potenciando o cumprimento do prazo de intervenção.

### **Cenário 3.**

A equipa de administração está em reunião e apenas tem disponível o seu telefone móvel. Devido a imperativos aplicacionais é necessário atribuir um



determinado *role* a um utilizador e apenas utilizadores privilegiados, como um DBA, o pode fazer. Após acusar um alerta para esta situação, basta aceder à aplicação e atribuir o *role* desejado ao utilizador envolvido.

Em todos os cenários, o tempo de resposta é importante. A utilização da aplicação vem reduzir esse tempo. No primeiro cenário é inclusive solicitado conhecimento técnico para a criação de um *datafile* mas a complexidade é eliminada através do interface da aplicação.

## Capítulo 6

# 6. Conclusões e trabalho futuro

A aplicação responde às necessidades prementes, é simples de utilizar, disponível e ágil no que se refere à sua implementação nas duas plataformas móveis mais utilizadas: iOS e Android. Acrescenta rapidez e eficiência aos processos de gestão, facilitando a intervenção dos diferentes elementos das equipas de gestão de base de dados. Permite simplificar procedimentos e uma gestão remota eficiente.

A aplicação está totalmente funcional, cumprindo os requisitos idealizados e projetados. Poderá ser melhorada ao nível da *interface* gráfica requerendo um aprofundamento ao nível da UX/UI. A implementação de GraphQL[16] seria também interessante e a sua comparação com REST, o que no contexto do software utilizado na conceção da aplicação (React) faria sentido uma vez que também o *Facebook* utiliza ambas as tecnologias. GraphQL iria alavancar a interação da aplicação com outros DBMS introduzindo uma camada de abstração ao nível do motor de base de dados utilizado, o que permitiria uma maior flexibilidade e portabilidade.

Apesar da autenticação não ser um problema no que diz respeito a ações privilegiadas sobre uma base de dados, uma vez que todos os perfis criados são locais e a validação de acessos é sempre efetuada contra a base de dados independentemente da sua origem na aplicação, por questões de

auditoria, que é um requisito que nas empresas é bastante importante, poderia ter sido implementado um sistema mais robusto e persistente, com recurso a *tokens* ou *oAuth* por exemplo. E porque não *blockchain* para gestão de identidades?

Em conclusão, o *software* desenvolvido corresponde às expectativas e requisitos promovendo a produtividade da equipa, introduzindo abstração face ao conhecimento técnico e aumentando a disponibilidade do serviço, eliminando a necessidade de um recurso disponível em permanência para responder às necessidades que a aplicação veio colmatar.

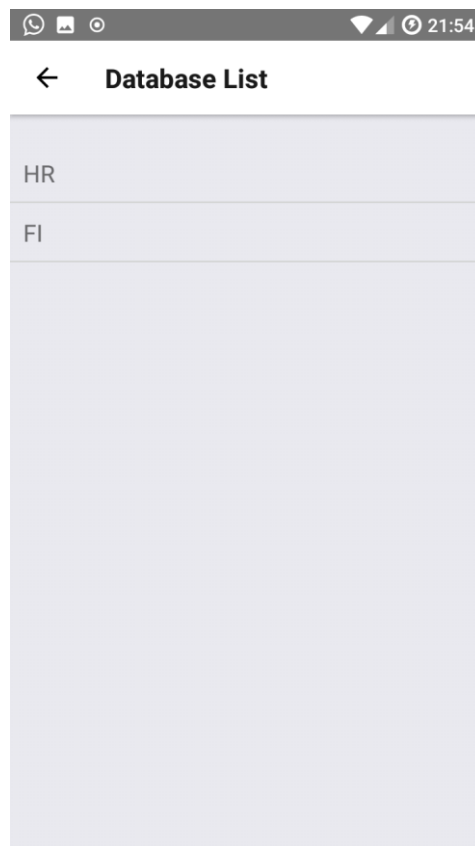
## 7. Bibliografia

- [1] EUGDPR. EUGDPR.com. <https://www.eugdpr.org/eugdpr.org.html>, (Online a 10 de Dezembro de 2017)
- [2] REACT. Reactjs.org. <https://reactjs.org/>, (Online a 10 de Dezembro de 2017)
- [3] REACT NATIVE. <https://facebook.github.io/react-native/showcase.html>, (Online a 10 de Dezembro de 2017)
- [4] FACEBOOK. Facebook code <https://code.facebook.com/projects/176988925806765/react/>, (Online a 10 de Dezembro de 2017)
- [5] Rausmayer, Axel. Exploring ES6 <http://exploringjs.com/es6.html>, Online a 10 de Dezembro de 2017)
- [6] REACT. Reactjs.org. <https://reactjs.org/docs/introducing-jsx.html>, (Online a 10 de Dezembro de 2017)
- [7] Oracle Cloud Control. Docs.oracle.com [https://docs.oracle.com/cd/E24628\\_01/doc.121/e25353/overview.htm#EMCON131](https://docs.oracle.com/cd/E24628_01/doc.121/e25353/overview.htm#EMCON131) (Online a 15 de Fevereiro de 2018)
- [8] SOAP. Stackify.com <https://stackify.com/soap-vs-rest/> (Online a 01 de Abril de 2018)

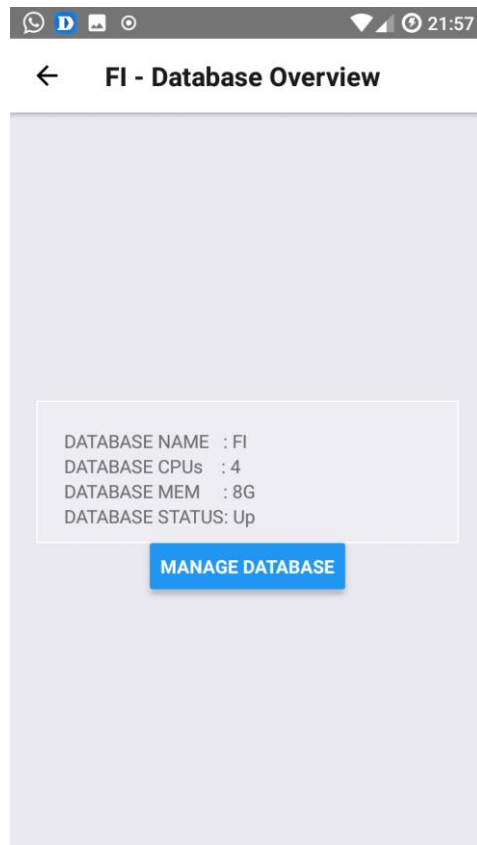
- [9] REST. Stormpath.com <https://stormpath.com/blog/rest-vs-soap>. (Online a 01 de Abril de 2018)
- [10] JSON and XML. W3schools.com [https://www.w3schools.com/js/js\\_json\\_xml.asp](https://www.w3schools.com/js/js_json_xml.asp). (Online a 01 de Abril de 2018)
- [11] JSON. Json.org <http://www.json.org/>. (Online a 01 de Abril de 2018)
- [12] TS4500. Ibm.com [https://www.ibm.com/developerworks/community/blogs/storage\\_redbooks/entry/try\\_out\\_the\\_ibm\\_ts4500\\_mobile\\_app?lang=en](https://www.ibm.com/developerworks/community/blogs/storage_redbooks/entry/try_out_the_ibm_ts4500_mobile_app?lang=en). (Online a 01 de Abril de 2018)
- [13] IBM Data Server Manager. Ibm.com <https://www.ibm.com/developerworks/downloads/im/dsm/index.html>. (Online a 01 de Abril de 2018)
- [14] SSMS. Docs.microsoft.com <https://docs.microsoft.com/enus/sql/ssms/sql-server-management-studio-ssms>. (Online a 01 de Abril de 2018)
- [15] Tomcat. Apache.org <http://tomcat.apache.org/>. (Online a 01 de Abril de 2018)
- [16] GraphQL. GraphQL.com <https://graphql.org/> (Online a 01 de Abril de 2018)

## 8. Anexos

## Figuras

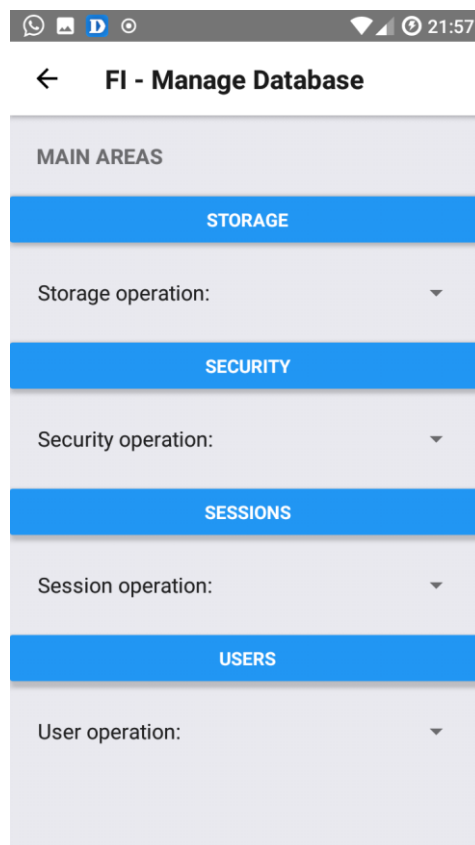


**Figura 4 - "Database List"**

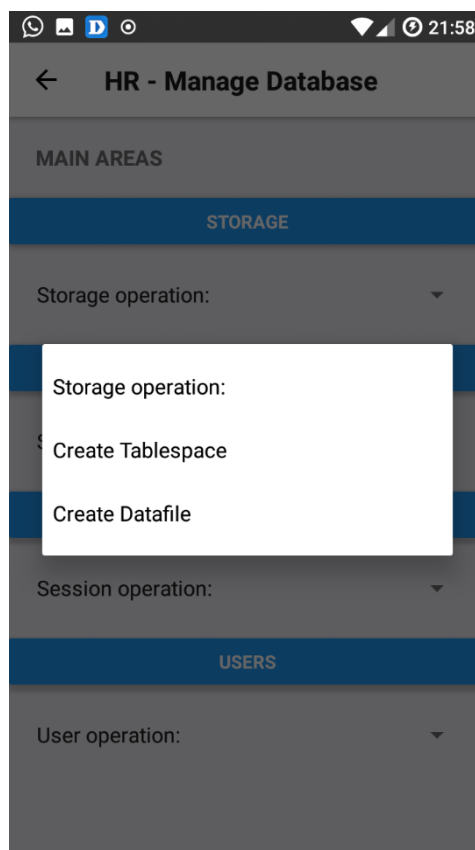


**Figura 5 - “Database Overview”**

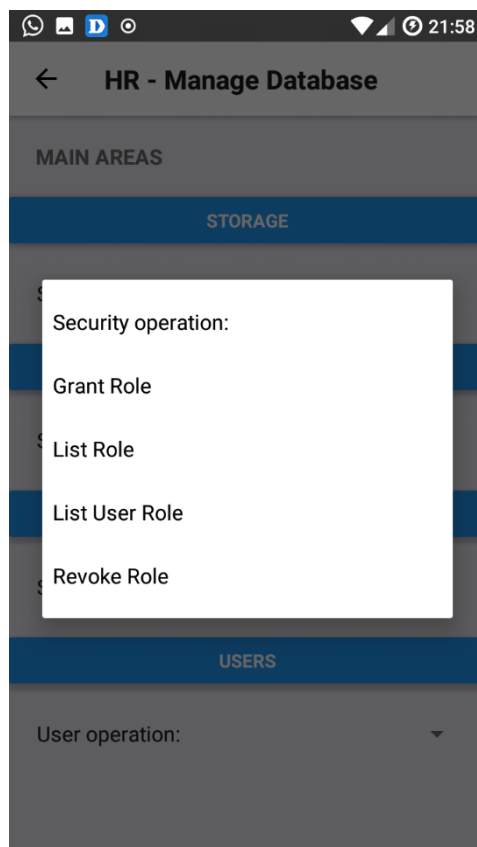




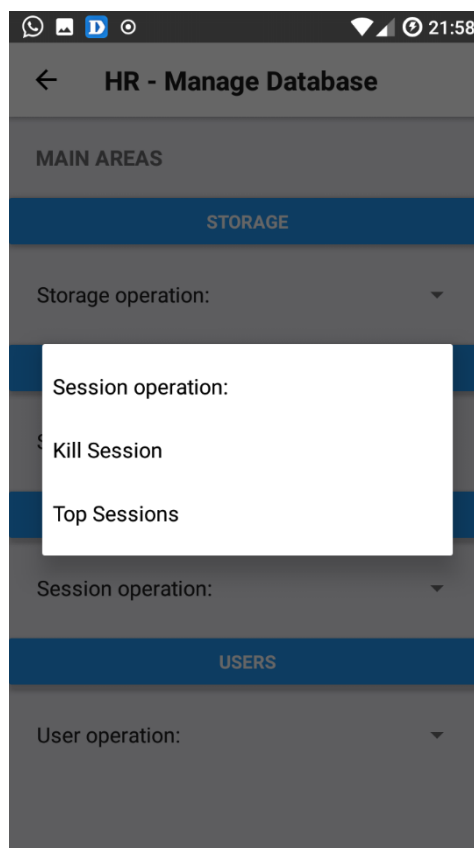
**Figura 6 - "Main Areas"**



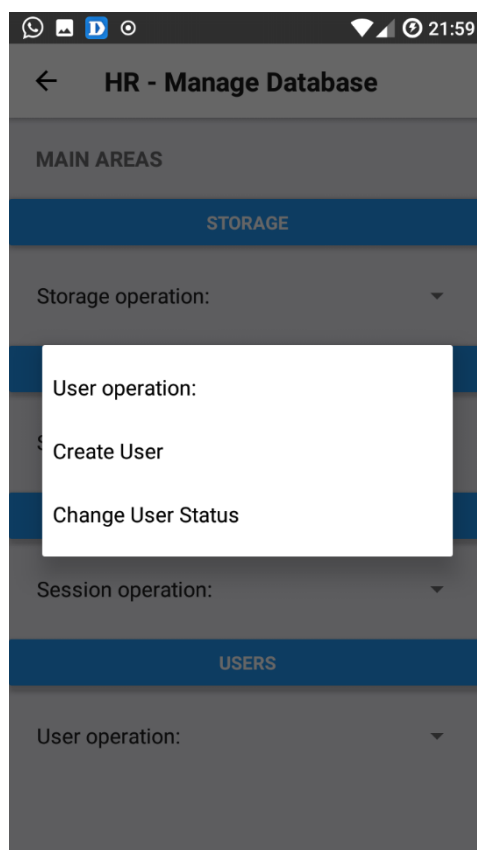
**Figura 7 - “Storage Operations”**



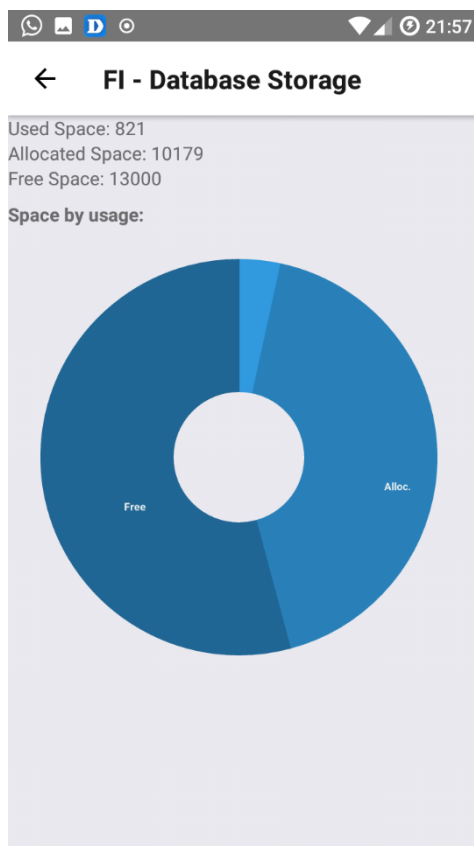
**Figura 8 - "Security Operations"**



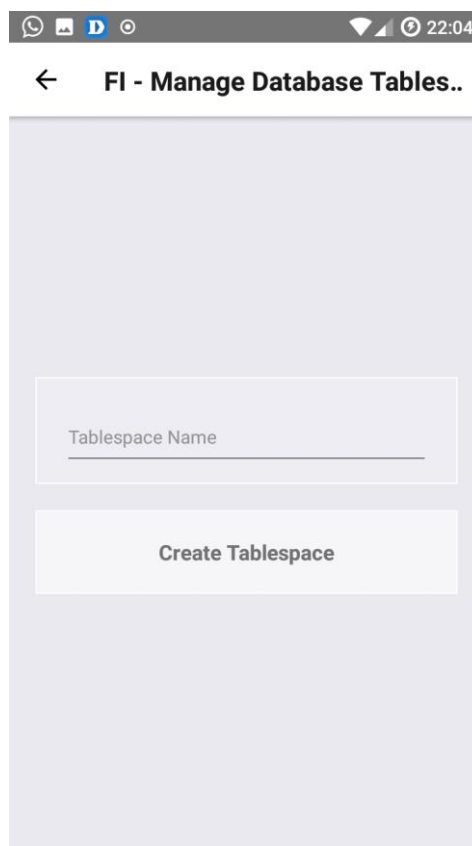
**Figura 9 - "Sessions Operations"**



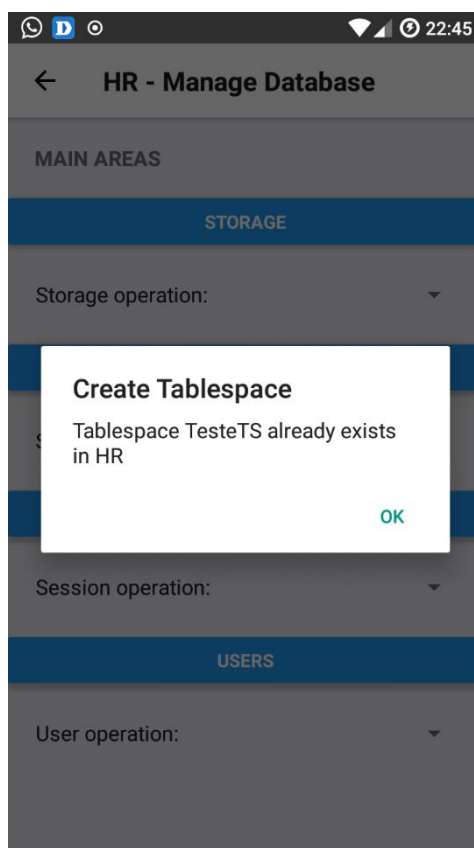
**Figura 10 - "Users Operations"**



**Figura 11 - “Storage Summary”**

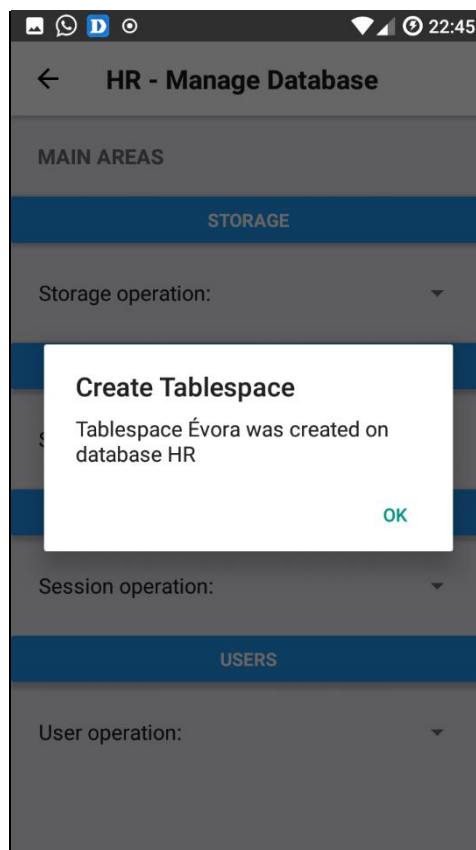


**Figura 12 - “Create Tablespace”**

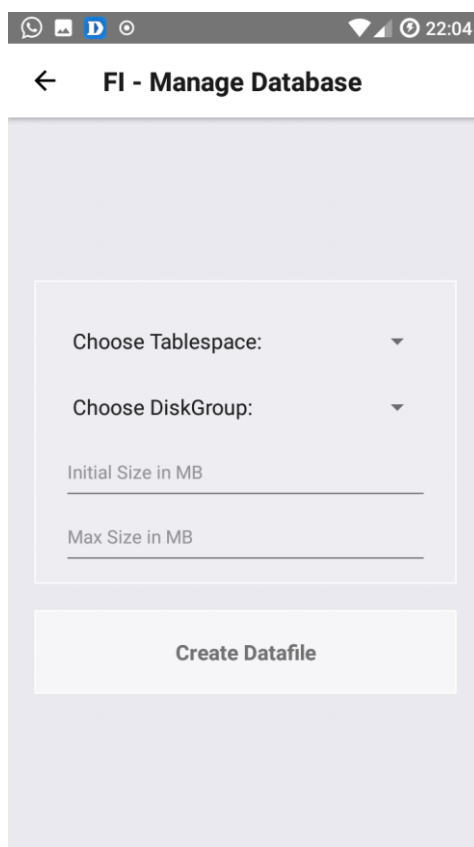


**Figura 13 - “Failure Tablespace”**

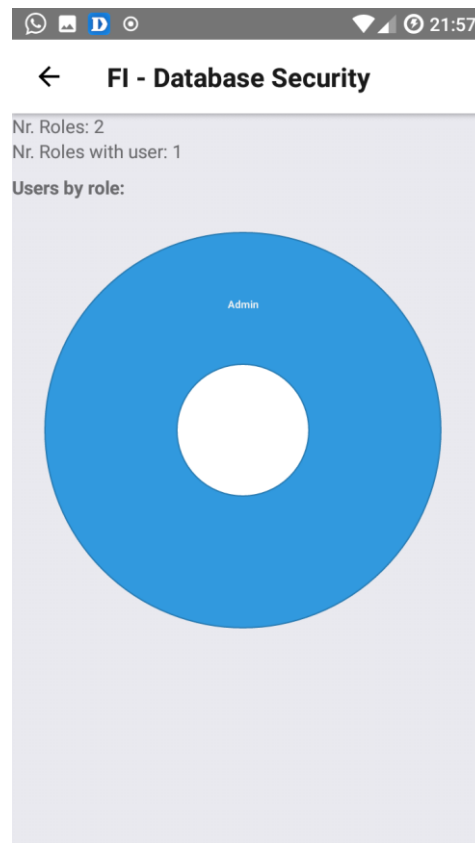




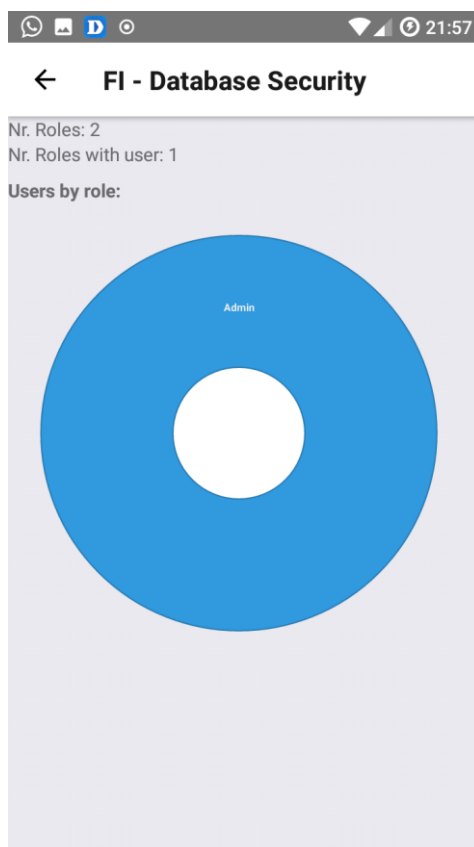
**Figura 14 - "Success Tablespace"**



**Figura 15 - "Create Datafile"**



**Figura 16 - “Security Summary”**



**Figura 17 - “Grant Role”**

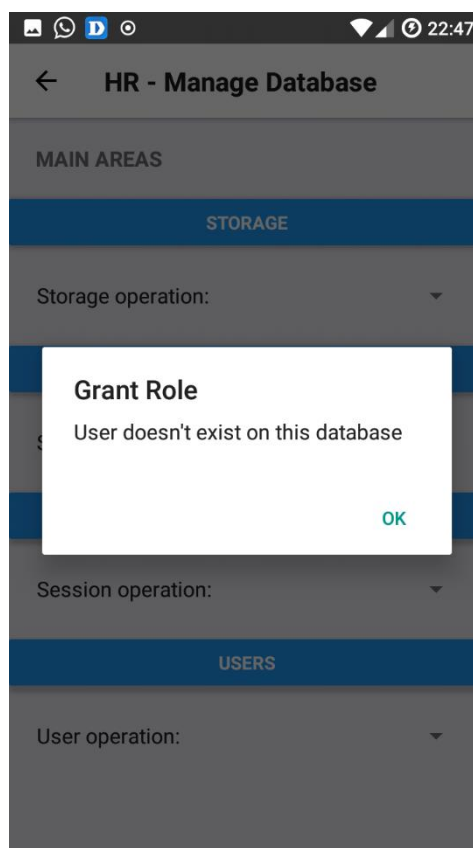
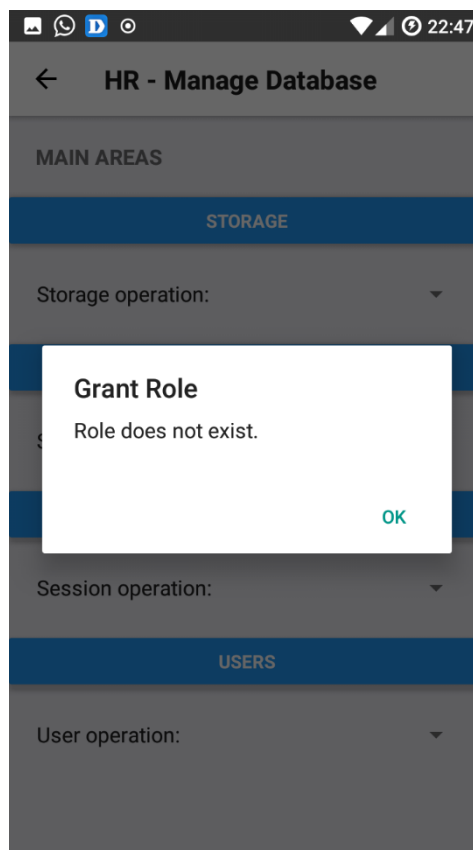
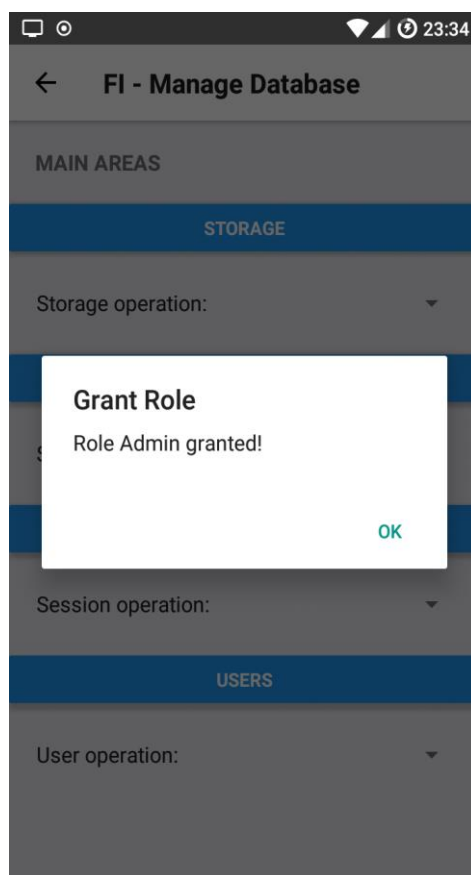


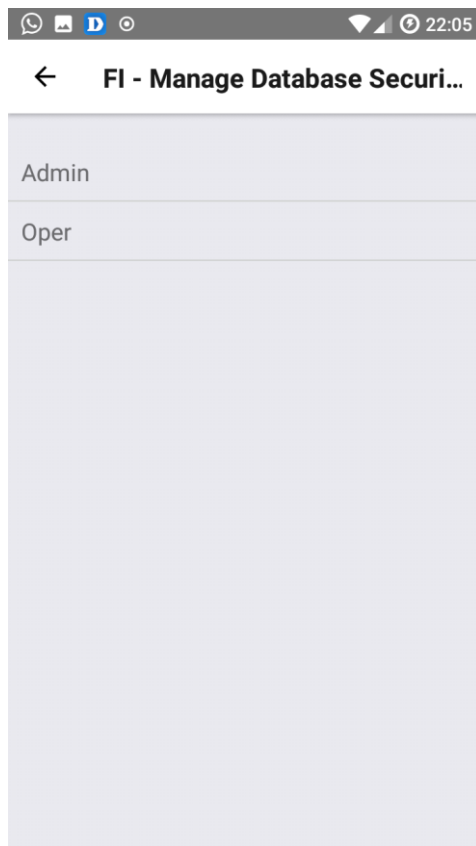
Figura 18 - "Failure User Grant Role"



**Figura 19 - "Failure Role Grant Role"**

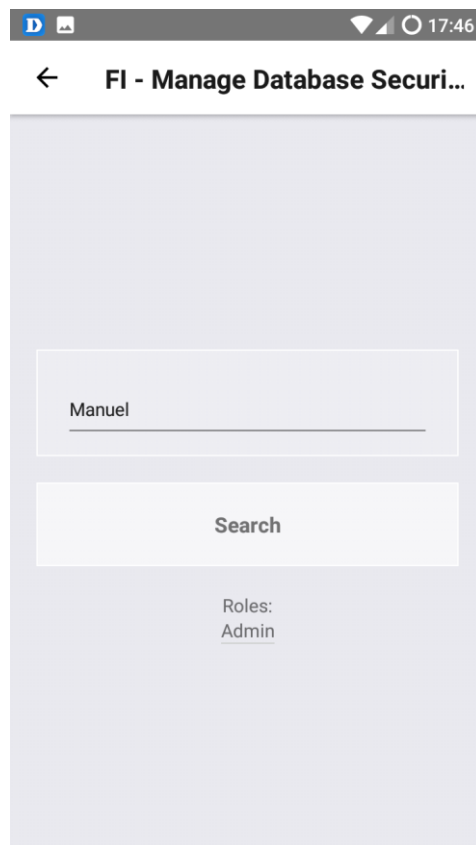


**Figura 20 - “Role atribuído com sucesso”**

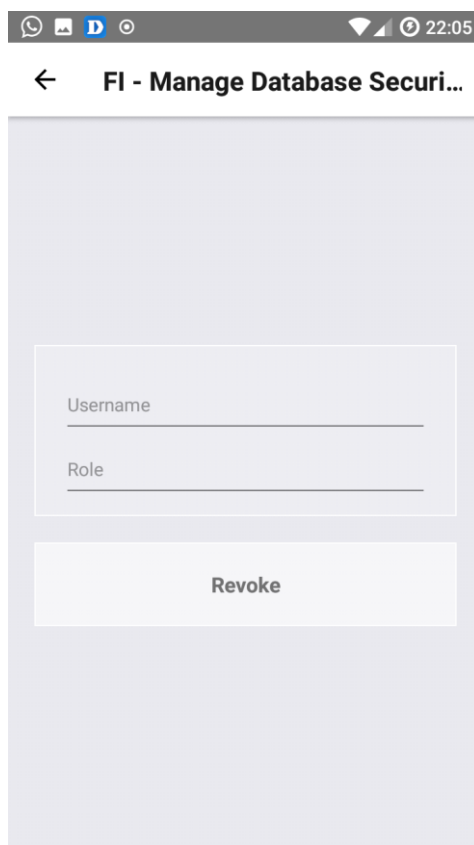


**Figura 21 - "List Role"**

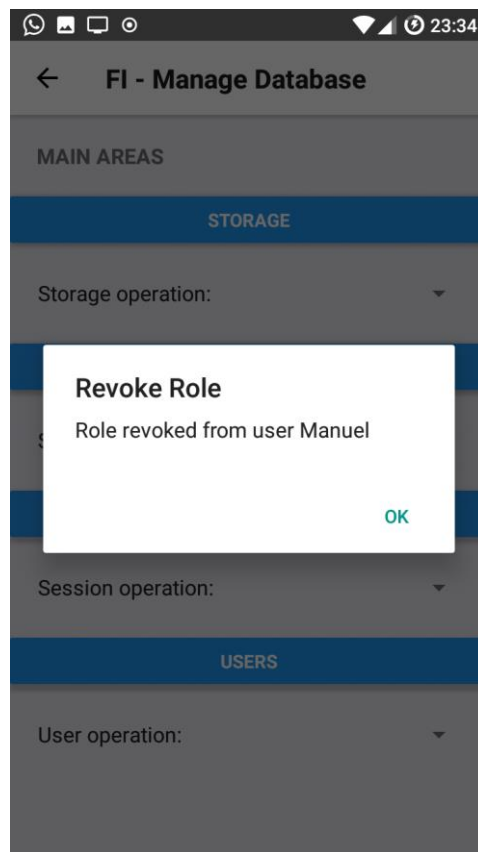




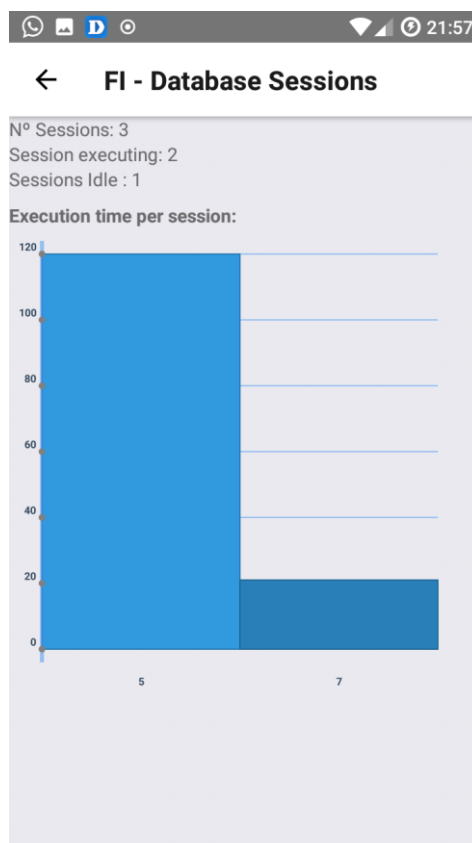
**Figura 22 - “List User Role”**



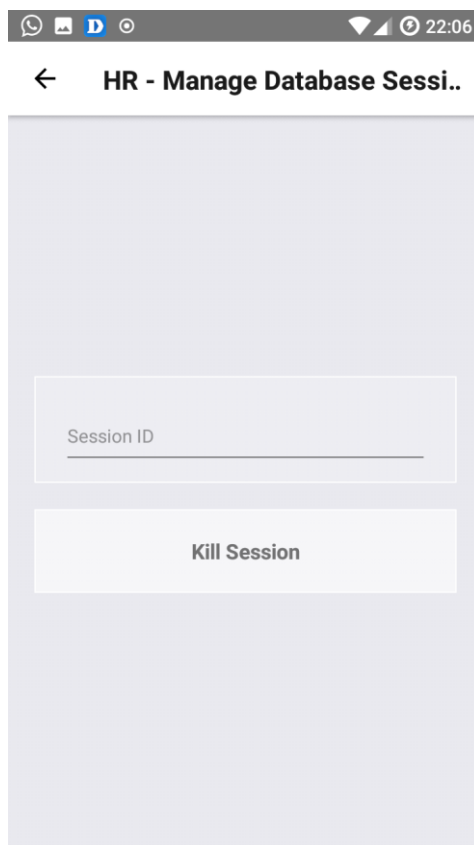
**Figura 23 – “Revoke Role Screen”**



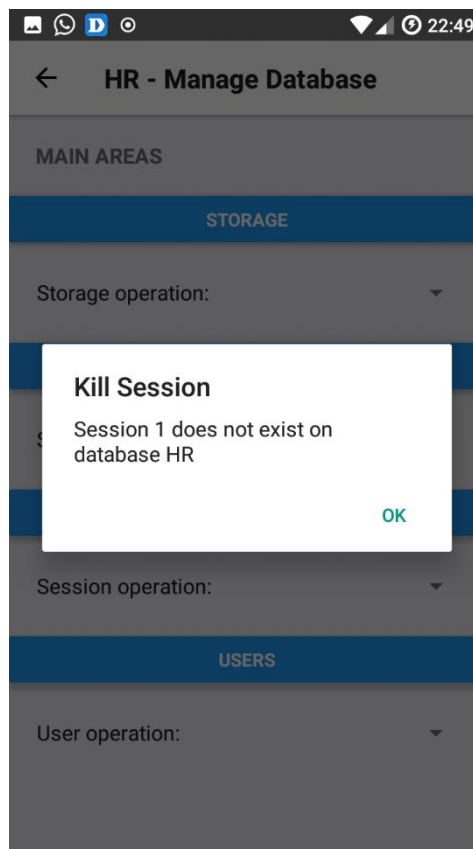
**Figura 24 - "Sucesso Revoke Role"**



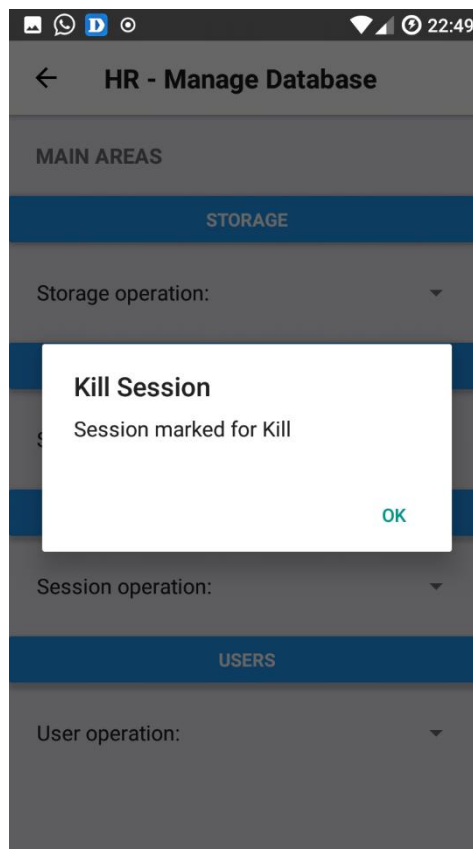
**Figura 25 - “Sessions Summary”**



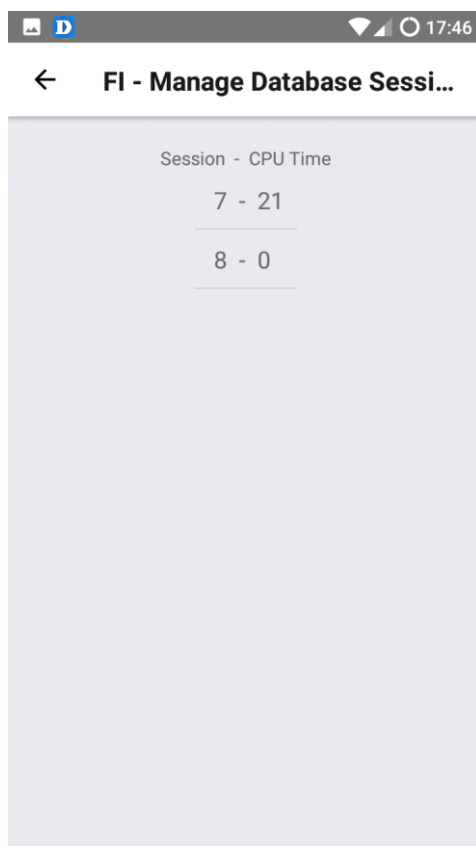
**Figura 26 - "Kill Session"**



**Figura 27 - "Failure Kill Session"**

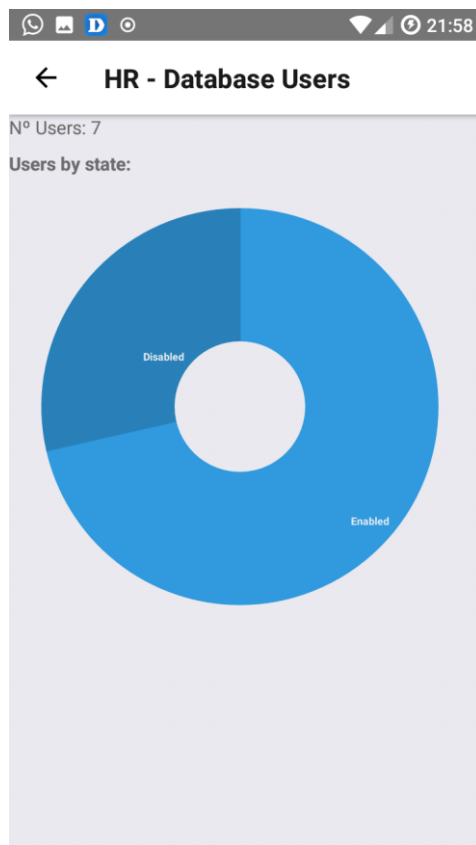


**Figura 28 - "Success Kill Session"**

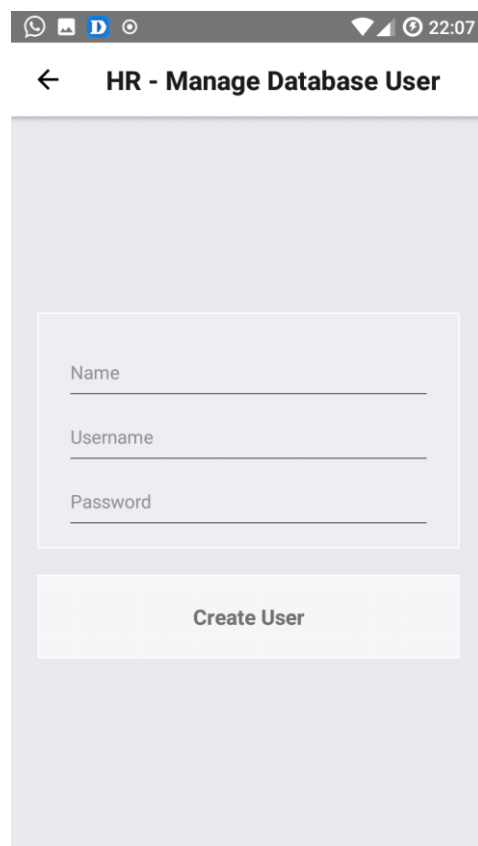


**Figura 29 - “Top Sessions”**



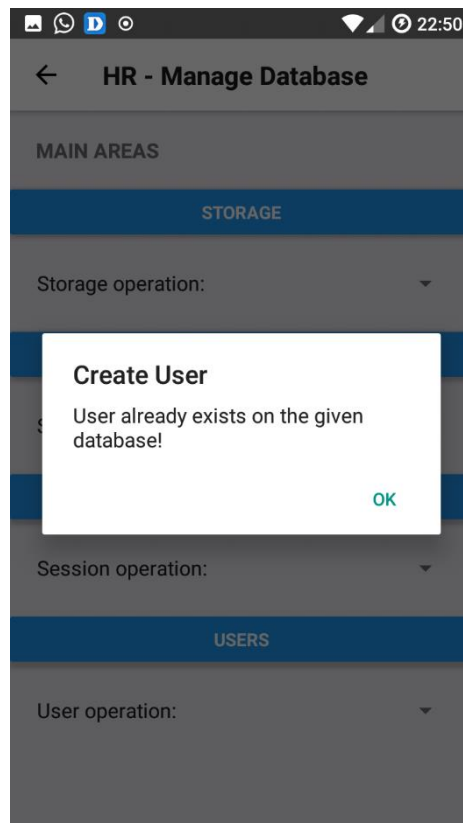


**Figura 30 - “Users Summary”**

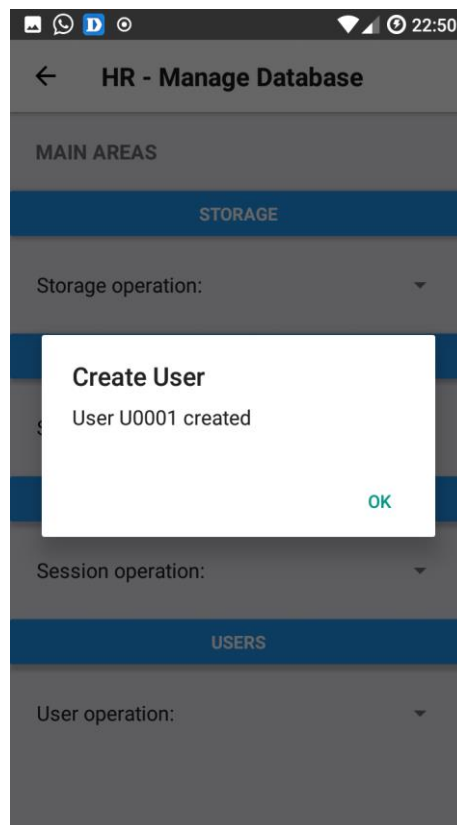


The image shows a mobile application interface for managing database users. At the top, there is a status bar with icons for WhatsApp, messages, a blue 'D' icon, and a battery icon, along with the time 22:07. Below the status bar is a navigation bar with a back arrow and the text "HR - Manage Database User". The main content area is a light gray rectangle containing a form with three input fields: "Name", "Username", and "Password". Below the form is a button labeled "Create User".

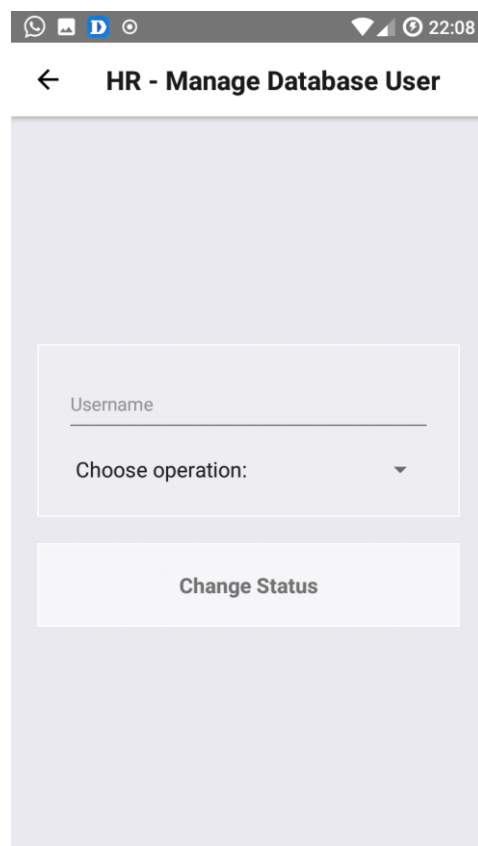
**Figura 31 - "Create User"**



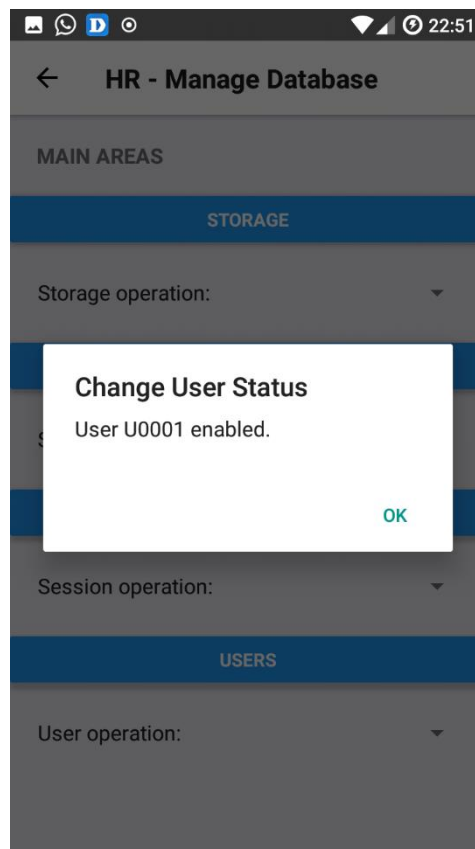
**Figura 32 - "Failure Create User"**



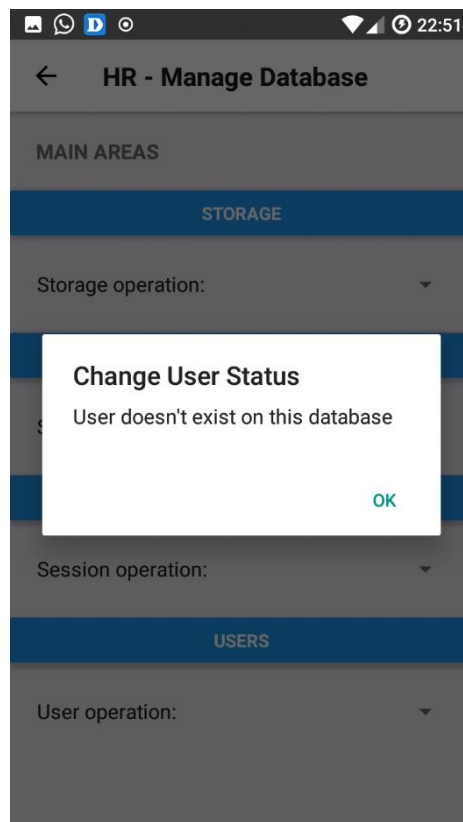
**Figura 33 - "Success Create User"**



**Figura 34 - "Change User Status"**



**Figura 35 - "Success Enable User"**



**Figura 36 - "Failure Change User"**

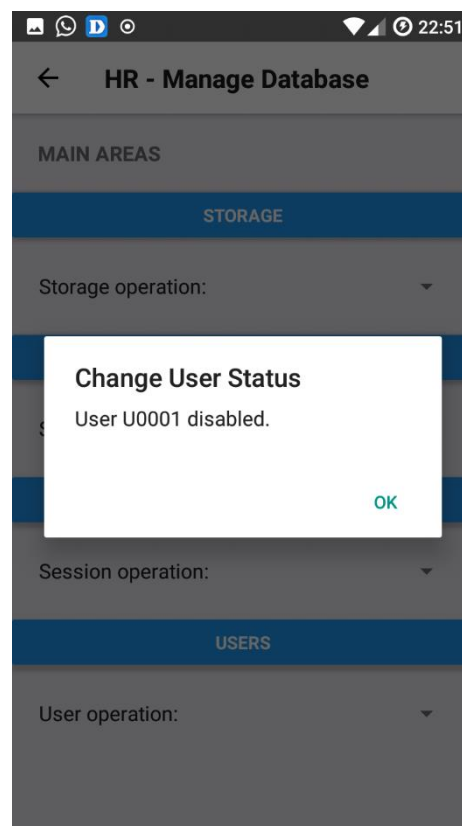


Figura 37 - "Success Disable User"