

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



**DESENVOLVIMENTO DE UM APLICATIVO PARA MONITORAMENTO DE
BANCO DE DADOS**

CELSO PEREIRA DO VALE JÚNIOR

GOIÂNIA
2020

CELSO PEREIRA DO VALE JÚNIOR

**DESENVOLVIMENTO DE UM APLICATIVO PARA MONITORAMENTO DE
BANCO DE DADOS**

Trabalho de Conclusão de Curso apresentado à Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Orientador(a):

Profa. Ma. Ludmilla Reis Pinheiro dos Santos

Banca examinadora:

Prof. Me. Eugênio Júlio Messala Cândido de
Carvalho

Prof. Me. Fernando Gonçalves Abadia

GOIÂNIA
2020

CELSO PEREIRA DO VALE JÚNIOR

**DESENVOLVIMENTO DE UM APLICATIVO PARA MONITORAMENTO DE
BANCO DE DADOS**

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharel em Ciência da Computação, em ____/____/_____.

Orientadora: Profa. Ma. Ludmilla Reis Pinheiro dos Santos

Profa. Ma. Ludmilla Reis Pinheiro dos Santos
Coordenadora de Trabalho de Conclusão de Curso

GOIÂNIA
2020

RESUMO

Com o aumento de dados nas empresas, surge a necessidade de aplicações de software para o armazenamento dos mesmos. Para um armazenamento seguro e eficaz, faz-se necessário o uso de um Sistema Gerenciador de Banco de Dados (SGBD). Os Administradores de Bancos de Dados são responsáveis por gerenciar, instalar, configurar, atualizar e monitorar estes bancos de dados. Este trabalho apresenta os requisitos, a implementação e a documentação do aplicativo DbMonitor, que permite o monitoramento de bancos de dados Oracle através de dispositivos móveis. Para a implementação foram utilizados o *framework* Flutter para a construção das interfaces, linguagem de programação *Dart*, banco de dados Firebase para armazenamento dos dados dos usuários, além da construção de uma Interface de Programação de Aplicações (*Application Programming Interface*, API) utilizando o *framework* .NET para a comunicação com os bancos de dados Oracle. O aplicativo permite o monitoramento em tempo real de diversas métricas disponibilizadas pelo Oracle, como o uso da Unidade central de processamento (*Central Processing Unit*, CPU) da instância, espaço utilizado pelas *tablespaces* e operações longas atuais.

Palavras-Chave: *Aplicativo DbMonitor, Flutter, Dart, Oracle, Firebase, .NET.*

ABSTRACT

With the increase of data in companies, there is a need for software applications to store them. For safe and effective storage, it is necessary to use a Database Management System (DBMS). Database Administrators are responsible for managing, installing, configuring, updating and monitoring these databases. This work presents the implementation, documentation and requirements of the DbMonitor application, which allows the monitoring of Oracle databases through mobile devices. For the implementation, the Flutter framework was used to build the interfaces, Dart programming language, Firebase database to store user data, in addition the building of an Application Programming Interface (API) using the .NET framework to communicate with Oracle databases. The application allows real-time monitoring of several metrics provided by Oracle, such as the Central Processing Unit (CPU) usage of the instance, space used by tablespaces and current long operations.

Keywords: *DbMonitor Application, Flutter, Dart, Oracle, Firebase, .NET.*

LISTA DE FIGURAS

Figura 1 – Arquitetura do Flutter.....	45
Figura 2 – Stateful e Stateless Widgets.	46
Figura 3 – Serviços do Firebase.....	48
Figura 4 – Modelo de e-mail para redefinição de senha.	49
Figura 5 – Cloud Firestore do aplicativo DbMonitor.	50
Figura 6 – Visão geral do Visual Studio Code.	51
Figura 7 – Visão geral do Visual Studio.	52
Figura 8 – Tela de Login.	54
Figura 9 – Tela de Cadastro.....	55
Figura 10 – Tela de Recuperação de Senha.....	56
Figura 11 – E-mail de recuperação de senha.....	56
Figura 12 – Tela de erro de e-mail não cadastrado.....	57
Figura 13 – Tela Principal.....	58
Figura 14 – Menu Lateral.	59
Figura 15 – Tela de Seleção de Banco de Dados.	60
Figura 16 – Tela de Cadastro de Banco de Dados.	61
Figura 17 – Telas de desempenho, parte 1.....	62
Figura 18 – Telas de desempenho, parte 2.....	63
Figura 19 – Telas de desempenho, parte 3.....	64
Figura 20 – Filtros da Tela de Desempenho.	65
Figura 21 – Tela de Notificações.....	66
Figura 22 – Tela de filtros de notificações.....	67
Figura 23 – Tela de <i>Tablespaces</i>	68
Figura 24 – Tela de <i>Longops</i>	69
Figura 25 – Tela de <i>Top SQL</i>	70
Figura 26 – Filtros de <i>Top SQL</i>	71
Figura 27 – Tela de <i>Advisor</i> para <i>DbCache</i> e <i>Memory Target</i>	72
Figura 28 – Tela de <i>Advisor</i> para <i>PGA</i> e <i>Hit Ratio</i>	73
Figura 29 – Tela de detalhes <i>Longops</i>	74
Figura 30 – Consulta de <i>Tablespaces</i> no Oracle	74

LISTA DE QUADROS

Quadro 1 – Requisitos de Usuários - Necessidades	18
Quadro 2 – RF 001: Cadastrar Usuário.....	19
Quadro 3 – RF 002: Realizar Login.....	20
Quadro 4 – RF 003: Recuperar senha.	20
Quadro 5 – RF 004: Cadastrar um banco de dados.....	21
Quadro 6 – RF 005: Atualizar o cadastro de um banco de dados.....	21
Quadro 7 – RF 006: Alterar o banco de dados atual.	22
Quadro 8 – RF 007: Visualizar a simulação do Advisor.	22
Quadro 9 – RF 008: Visualizar operações longas.....	23
Quadro 10 – RF 009: Visualizar as notificações.....	23
Quadro 11 – RF 010: Visualizar dados de desempenho do banco de dados.	24
Quadro 12 – RF 011: Visualizar as tablespaces.	24
Quadro 13 – RF 013: Visualizar operações de um período.	25
Quadro 14 – RF 014: Visualizar o SQL de uma operação.	25
Quadro 15 – RQ 001: Disponibilidade.....	26
Quadro 16 – RQ 002: Recuperação de erros.....	26
Quadro 17 – RQ 003: Linguagem de Programação.	27
Quadro 18 – RQ 004: Plataformas.	27
Quadro 19 – RQ 005: Segurança dos dados.	28
Quadro 20 – RQ 006: Responsividade.....	28

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
CPU	<i>Central Processing Unit</i>
CSU	Caso de Uso
DBA	<i>DataBase Administrator</i>
DD	Dicionário de Dados
RF	Requisitos Funcionais
RQ	Requisitos de Qualidade
SGBD	Sistema Gerenciador de Banco de Dados
SID	<i>System ID</i>

SUMÁRIO

1	INTRODUÇÃO.....	12
1.1	Objetivo geral	12
1.2	Objetivos específicos	12
1.3	Justificativa.....	13
1.4	Stakeholders	13
1.5	Visão geral do trabalho	13
2	DESCRIÇÃO GERAL DO APLICATIVO	14
2.1	Aspecto geral do aplicativo	14
2.1.1	<i>Interfaces do aplicativo.....</i>	14
2.1.2	<i>Interfaces do usuário.....</i>	16
2.1.3	<i>Interfaces de hardware.....</i>	16
2.1.4	<i>Interfaces de software.....</i>	16
2.2	Características dos usuários	16
2.3	Funcionalidades do aplicativo	16
2.4	Restrições/Limitações	17
3	DOCUMENTAÇÃO DO APLICATIVO.....	18
3.1	Requisitos de Usuários - Necessidades.....	18
3.2	Requisitos Funcionais	19
3.3	Requisitos de Qualidade	25
3.4	Diagrama de casos de uso.....	28
3.5	Casos de uso descritivos	29
3.6	Dicionário de dados do sistema	42
4	DESENVOLVIMENTO DO APLICATIVO DBMONITOR.....	44
4.1	<i>Framework Flutter</i>	44
4.1.1	<i>Camadas Arquiteturais.....</i>	44
4.1.2	<i>Widgets</i>	45
4.2	Linguagem de programação Dart.....	47
4.3	Firebase	47
4.3.1	<i>Autenticação usando Firebase.....</i>	48
4.3.2	<i>Firestore Cloud Firestore.....</i>	50
4.4	Visual Studio Code.....	50
4.5	API DbMonitor.....	51
4.5.1	<i>.NET Framework.....</i>	51
4.5.2	<i>Oracle Data Provider para .NET</i>	51

4.5.3	<i>Visual Studio Community</i>	52
5	IMPLEMENTAÇÕES E RESULTADOS	53
5.1	Telas do Aplicativo	53
5.1.1	Tela de Login	53
5.1.2	Tela de Cadastro	54
5.1.3	Tela de Recuperação de Senha	55
5.1.4	Tela Principal	57
5.1.5	Tela de Seleção de Banco de Dados	59
5.1.6	Tela de Cadastro de Banco de Dados	60
5.1.7	Tela de Desempenho	61
5.1.8	Tela de Notificações	65
5.1.9	Tela de <i>Tablespaces</i>	67
5.1.10	Tela de <i>Longops</i>	68
5.1.11	Tela de <i>Top SQL</i>	69
5.1.12	Tela de <i>Advisor</i>	71
5.1.13	Tela de Detalhes do SQL	73
5.2	Comparação da visualização dos dados	74
6	CONSIDERAÇÕES FINAIS	75
6.1	Trabalhos futuros	75
	REFERÊNCIAS	76
	APÊNDICE A – ENTREVISTA	78
	APÊNDICE B – BRIEFING	79
	ANEXO A – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA	80

1 INTRODUÇÃO

Com o crescente desenvolvimento tecnológico, amplia-se o acesso aos smartphones, que são equipados com processadores e memórias semelhantes aos utilizados em computadores, permitindo a construção de softwares robustos e performáticos com as comodidades que os dispositivos móveis oferecem (SARTORI, 2015; BEKOSKI, 2016).

Com o aumento de dados nas empresas, surgem aplicações de software juntamente com seus bancos de dados para armazenamento destes dados. Para um armazenamento seguro e eficaz, faz-se necessário o uso de um Sistema Gerenciador de Banco de Dados (SGBD). Segundo Silberschatz (1999, p. 21), um SGBD consiste em uma coleção de arquivos e programas que são interrelacionados e que permitem ao usuário acesso para consultas e alterações destes dados.

O profissional responsável por gerenciar, instalar, configurar, atualizar e monitorar os bancos de dados é o Administrador de Banco de Dados (*Database Administrator*, DBA) (ORACLE, 2015). O DBA tem a necessidade de constantemente monitorar os bancos de dados pelos quais é responsável para garantir o funcionamento ininterrupto destes.

Com base na necessidade de monitoramento destes recursos, este trabalho visa a criação de um aplicativo móvel, denominado DbMonitor, para realizar o monitoramento de bancos de dados.

1.1 Objetivo geral

Este trabalho tem o objetivo de desenvolver e documentar o aplicativo multiplataforma *DbMonitor*, para auxiliar no monitoramento de bancos de dados Oracle.

1.2 Objetivos específicos

- Elicitar os requisitos funcionais e de qualidade do sistema;
- Especificar dicionários de dados;
- Especificar casos de uso;
- Estudar tecnologias para implementação do aplicativo;

- Implementação do aplicativo DbMonitor utilizando a linguagem de programação Dart com o *framework* Flutter;
- Implementação da API DbMonitor utilizando o framework .NET;
- Realizar testes e a documentação do aplicativo DbMonitor.

1.3 Justificativa

O monitoramento de banco de dados é uma das atividades diárias de um DBA. A detecção rápida de falhas graves no banco de dados, como uso excessivo de recursos e operações muito longas, é de extrema importância, pois irá minimizar os efeitos negativos destas. Diante das situações expostas, o monitoramento dos bancos de dados com acesso aos dados de monitoramento em um aplicativo móvel se torna relevante.

1.4 Stakeholders

Stakeholders se referem a parte interessada para o desenvolvimento deste aplicativo, sendo os DBAs e os analistas de bancos de dados.

1.5 Visão geral do trabalho

Este trabalho está dividido em seis capítulos. A introdução, os objetivos gerais e específicos, problemática e justificativa foram apresentadas no capítulo 1.

O capítulo 2 aborda os aspectos gerais do aplicativo DbMonitor, sendo apresentadas as características dos usuários, interfaces do aplicativo e suas restrições/limitações.

O capítulo 3 descreve a documentação utilizada para o desenvolvimento do aplicativo.

O capítulo 4 descreve as tecnologias utilizadas para o desenvolvimento do aplicativo.

O capítulo 5 apresenta a implementação das telas do aplicativo juntamente com suas funcionalidades.

Finalmente, o capítulo 6 apresenta as considerações finais, dificuldades encontradas durante o desenvolvimento e sugestões para trabalhos futuros.

2 DESCRIÇÃO GERAL DO APLICATIVO

Este capítulo apresenta as interfaces do aplicativo DbMonitor, suas características, perfil dos usuários alvo e suas restrições/limitações.

2.1 Aspecto geral do aplicativo

O aplicativo DbMonitor foi construído utilizando o *framework Flutter* com o objetivo de facilitar o monitoramento de bancos de dados Oracle, compilando diversas informações acerca do desempenho destes.

Ele permite com que os usuários realizem o seu cadastro e, em seguida, realizem o cadastro dos bancos de dados que desejam monitorar.

2.1.1 Interfaces do aplicativo

O aplicativo é composto das seguintes telas:

- **Tela de login:** Nesta tela o usuário deverá informar o e-mail e a senha previamente cadastrados para acessar as outras funcionalidades;
- **Tela de cadastro:** Nesta tela o usuário realiza o cadastro, informando nome, e-mail, senha e confirmação de senha;
- **Tela de recuperação de senha:** Nesta tela o usuário informa o endereço de e-mail para que seja enviado um e-mail para a recuperação de senha;
- **Tela inicial:** Nesta tela são exibidas as informações de:
 - Nome do usuário autenticado;
 - E-mail do usuário autenticado;
 - Quantidade de notificações do banco de dados selecionado;
 - Nome do banco de dados selecionado.
- **Tela de cadastro de banco de dados:** Nesta tela o usuário cadastra os bancos de dados Oracle que serão acessados pelo aplicativo. O cadastro deve conter:
 - O nome do banco de dados;
 - O Host do banco de dados;
 - A porta do banco de dados;
 - O *System ID* (SID) do banco de dados;

- O usuário de acesso ao banco de dados;
 - A senha de acesso ao banco de dados.
- **Tela de seleção do banco de dados:** Nesta tela o usuário seleciona o banco de dados Oracle que o aplicativo DbMonitor irá acessar, dentre os cadastrados previamente;
 - **Tela de desempenho:** Nesta tela são exibidos os dados de desempenho dos últimos 45 minutos, para o banco de dados Oracle selecionado. O usuário poderá selecionar quais gráficos serão exibidos a partir da lista de opções no menu “Filtros”, conforme as categorias a seguir:
 - *CPU*: Uso de CPU da instância;
 - *Scheduler*: Uso de recursos do agendador de tarefas do Oracle;
 - *User I/O*: Tempo de espera para leitura e escrita de usuário;
 - *System I/O*: Tempo de espera para leitura e escrita do sistema;
 - *Concurrency*: Tempo de espera por concorrência de recursos;
 - *Application*: Tempo de espera por conta de aplicações;
 - *Commit*: Tempo de espera por *commit* de transações.
 - **Tela de notificações:** Nesta tela são exibidas notificações importantes geradas pelo banco de dados Oracle selecionado, podendo ser filtradas de acordo com o nível da notificação;
 - **Tela de tablespaces:** Nesta tela são listadas as *tablespaces* do banco de dados Oracle selecionado, exibindo o seu nome, status, tamanho alocado, tamanho utilizado, percentual alocado utilizado e percentual utilizado do máximo;
 - **Tela de Longops:** Nesta tela são exibidas as operações longas que estão ocorrendo no banco de dados Oracle selecionado;
 - **Tela de Top SQL:** Nesta tela são exibidas as consultas SQL executadas no banco de dados Oracle selecionado de acordo com os filtros de dias, quantidade mínima de execuções e módulo;
 - **Tela de Advisor:** Nesta tela são exibidos os dados do advisor para as categorias *DbCache*, *Memory Target*, *Program Global Area (PGA)* e *Hit Ratio* para o banco de dados Oracle selecionado.

2.1.2 Interfaces do usuário

O aplicativo DbMonitor foi construído para se adaptar as diferentes resoluções e tamanhos de telas, podendo ser utilizado dispositivos móveis que atendam as interfaces de *hardware* e *software*.

2.1.3 Interfaces de hardware

A instalação de aplicativos desenvolvidos com a utilização do *framework* Flutter pode ser realizada em dispositivos móveis Android que possuam processadores ARM ou dispositivos iOS a partir do iPhone 4S (FLUTTER, 2020).

2.1.4 Interfaces de software

O aplicativo pode ser instalado em dispositivos iOS e Android, tendo como requisito as versões:

- Android Jelly Bean, v16, 4.1.x ou mais recente;
- iOS versão 8.0 ou superior.

2.2 Características dos usuários

Os usuários que irão utilizar o aplicativo serão os Administradores de Banco de Dados que pretendem monitorar um ou mais bancos de dados Oracle.

Estes usuários poderão cadastrar os bancos de dados no aplicativo e monitorá-los em relação as diversas métricas que serão apresentadas no desenvolvimento deste trabalho.

2.3 Funcionalidades do aplicativo

O aplicativo DbMonitor permite com que o usuário:

- Realize o cadastro no aplicativo;
- Realize o cadastro dos bancos de dados a serem monitorados;
- Visualize os dados de desempenho destes bancos de dados;
- Visualize as notificações geradas por estes bancos de dados;
- Visualize as *tablespaces* destes bancos de dados;

- Visualize as operações longas destes bancos de dados;
- Visualize os *SQLs* executados em um período;
- Visualize os dados do *Advisor*.

2.4 Restrições/Limitações

O aplicativo DbMonitor possui as seguintes restrições:

- É necessária uma conexão com a internet para o acesso ao aplicativo, por conta do uso do banco de dados Firebase;
- É necessário que o banco de dados Oracle cadastrado aceite conexões do *host* que hospeda a API do DbMonitor.

3 DOCUMENTAÇÃO DO APLICATIVO

Este capítulo apresenta a documentação utilizada para o desenvolvimento do aplicativo *DbMonitor*. Esta documentação inclui: Requisitos de Usuários – Necessidades (RU), Requisitos Funcionais (RF), Requisitos de Qualidade (RQ) e Casos de Uso descritivos (CSU).

3.1 Requisitos de Usuários - Necessidades

Os requisitos foram elicitados com base na entrevista com o Bel. Augusto Custódio Mendes e no documento de *Briefing*, disponíveis respectivamente nos Apêndices A e B.

No Quadro 1 são apresentados os Requisitos de Usuários – Necessidades, elicitados da entrevista e do documento de *Briefing*.

Quadro 1 – Requisitos de Usuários - Necessidades

ID	Descrição	Fonte
RU 001	Usuário faz cadastro no sistema.	Documento <i>Briefing</i>
RU 002	Usuário faz login no sistema.	Documento <i>Briefing</i>
RU 003	Usuário recupera dados de login.	Documento <i>Briefing</i>
RU 004	Usuário mantém um banco de dados.	Documento <i>Briefing</i>
RU 005	Usuário visualiza informações relativas ao <i>Advisor</i> .	Documento <i>Briefing</i>
RU 006	Usuário visualiza as operações longas no banco de dados.	Entrevista
RU 007	Usuário visualiza o comando gerador de uma operação longa no banco de dados.	Entrevista
RU 008	Usuário visualiza as notificações recebidas.	Documento <i>Briefing</i>
RU 009	Usuário visualiza dados de desempenho do banco de dados.	Documento <i>Briefing</i>
RU 010	Usuário visualiza o status, tamanho e quantidade utilizada das <i>tablespaces</i> .	Documento <i>Briefing</i>
RU 011	Usuário visualiza dados dos SQLs mais recorrentes do banco de dados.	Entrevista

Fonte: Elaborado pelo autor.

3.2 Requisitos Funcionais

Nos Quadros 2 ao 14 são apresentados os requisitos funcionais do aplicativo DbMonitor.

Quadro 2 – RF 001: Cadastrar Usuário.

Identificador	Nome
RF 001	Cadastrar Usuário
Caso de Uso	Autor
CSU 001	Celso Pereira do Vale Júnior
Descrição	
O sistema deve permitir o cadastro do usuário que deverá informar e-mail e senha.	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar se o e-mail é válido. • Verificar se a senha possui letras e números com tamanho entre 8 e 12 caracteres. • Verificar se já existe um usuário cadastrado com este e-mail. 	
Dependência	Prioridade
RU 001, DD 001	Essencial

Fonte: Elaborado pelo autor.

Quadro 3 – RF 002: Realizar Login.

Identificador	Nome
RF 002	Realizar Login
Caso de Uso	Autor
CSU 002	Celso Pereira do Vale Júnior
Descrição	
O sistema deve permitir o login de usuários previamente cadastrados. Após informar o e-mail e senha, o usuário será redirecionado para a tela inicial do aplicativo.	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar se existe uma conta criada para o e-mail e a senha informados. 	
Dependência	Prioridade
RU 002, RF 001	Essencial

Fonte: Elaborado pelo autor.

Quadro 4 – RF 003: Recuperar senha.

Identificador	Nome
RF 003	Recuperar senha
Caso de Uso	Autor
CSU 003	Celso Pereira do Vale Júnior
Descrição	
O sistema deve permitir que o usuário recupere sua senha. Para isso deve informar o e-mail cadastrado.	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar se existe uma conta criada para o e-mail informado. 	
Dependência	Prioridade
RU 003, RF 001	Essencial

Fonte: Elaborado pelo autor.

Quadro 5 – RF 004: Cadastrar um banco de dados.

Identificador	Nome
RF 004	Cadastrar um banco de dados
Caso de Uso	Autor
CSU 004	Celso Pereira do Vale Júnior
Descrição	
O sistema deve permitir o cadastro de bancos de dados Oracle, informando o identificador, <i>host</i> , porta, usuário do banco, senha e SID.	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar se todos os campos foram preenchidos. • Verificar se os dados de conexão fornecidos são válidos. • Verificar a existência de um banco de dados com o mesmo identificador. 	
Dependência	Prioridade
RU 004, DD 002	Essencial

Fonte: Elaborado pelo autor.

Quadro 6 – RF 005: Atualizar o cadastro de um banco de dados.

Identificador	Nome
RF 005	Atualizar o cadastro de um banco de dados
Caso de Uso	Autor
CSU 005	Celso Pereira do Vale Júnior
Descrição	
O sistema deve permitir a atualização do cadastro de bancos de dados Oracle, informando o identificador, <i>host</i> , porta, usuário do banco, senha e SID.	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar se houve campos alterados. • Verificar se os dados de conexão fornecidos são válidos. • Verificar a existência de conflito no identificador do banco de dados. 	
Dependência	Prioridade
RU 005, RF 003	Essencial

Fonte: Elaborado pelo autor.

Quadro 7 – RF 006: Alterar o banco de dados atual.

Identificador	Nome
RF 006	Selecionar o banco de dados atual
Caso de Uso	Autor
CSU 006	Celso Pereira do Vale Júnior
Descrição	
O sistema deve permitir a seleção de um banco de dados Oracle previamente cadastrado para a utilização das outras funcionalidades do aplicativo.	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar se os dados de conexão fornecidos são válidos. 	
Dependência	Prioridade
RU 004, RF 003	Essencial

Fonte: Elaborado pelo autor.

Quadro 8 – RF 007: Visualizar a simulação do Advisor.

Identificador	Nome
RF 007	Visualizar a simulação do <i>Advisor</i>
Caso de Uso	Autor
CSU 008	Celso Pereira do Vale Júnior
Descrição	
O sistema deve permitir a consulta do <i>Advisor</i> para as categorias de <i>DbCache</i> , <i>Memory Target</i> , <i>Program Global Area</i> e <i>Hit Ratio</i> possibilitando filtrar uma das categorias e simular como as alterações em cada uma delas irá afetar o banco de dados Oracle selecionado.	
Critério de Verificação	
<ul style="list-style-type: none"> • Ter um banco de dados selecionado. 	
Dependência	Prioridade
RU 006	Essencial

Fonte: Elaborado pelo autor.

Quadro 9 – RF 008: Visualizar operações longas.

Identificador	Nome
RF 008	Visualizar operações longas
Caso de Uso	Autor
CSU 009	Celso Pereira do Vale Júnior
Descrição	
O sistema deve permitir a consulta das operações longas que estão acontecendo no banco de dados no momento, possibilitando a ordenação por horário de início, tempo decorrido e tempo estimado para o término.	
Critério de Verificação	
<ul style="list-style-type: none"> Ter um banco de dados selecionado. 	
Dependência	Prioridade
RU 007	Essencial

Fonte: Elaborado pelo autor.

Quadro 10 – RF 009: Visualizar as notificações.

Identificador	Nome
RF 009	Visualizar as notificações
Caso de Uso	Autor
CSU 011	Celso Pereira do Vale Júnior
Descrição	
O sistema deve permitir a listagem dos alertas do banco de dados Oracle selecionado.	
Critério de Verificação	
<ul style="list-style-type: none"> Verificar se todos os campos foram preenchidos; Verificar se já existe uma conta com aquela matrícula. Ter um banco de dados selecionado. 	
Dependência	Prioridade
RU 009	Essencial

Fonte: Elaborado pelo autor.

Quadro 11 – RF 010: Visualizar dados de desempenho do banco de dados.

Identificador	Nome
RF 010	Visualizar dados de desempenho do banco de dados
Caso de Uso	Autor
CSU 012	Celso Pereira do Vale Júnior
Descrição	
<p>O sistema deve permitir a consulta dos dados de desempenho do banco de dados Oracle selecionado, permitindo realizar um filtro da data e hora de início e fim do período analisado de acordo com a categoria deste medidor de desempenho. As categorias são CPU, User I/O, System I/O, Concurrency, Application, Commit, Configuration, Network e Outros.</p>	
Critério de Verificação	
<ul style="list-style-type: none"> Ter um banco de dados selecionado. 	
Dependência	Prioridade
RU 010	Essencial

Fonte: Elaborado pelo autor.

Quadro 12 – RF 011: Visualizar as tablespaces.

Identificador	Nome
RF 011	Visualizar as <i>tablespaces</i>
Caso de Uso	Autor
CSU 014	Celso Pereira do Vale Júnior
Descrição	
<p>O sistema deve permitir a visualização das <i>tablespaces</i> do banco de dados Oracle selecionado, listando o status (Ativa ou Inativa), o tamanho total e o tamanho utilizado.</p>	
Critério de Verificação	
<ul style="list-style-type: none"> Ter um banco de dados selecionado. 	
Dependência	Prioridade
RU 011	Essencial

Fonte: Elaborado pelo autor.

Quadro 13 – RF 013: Visualizar operações de um período.

Identificador	Nome
RF 013	Visualizar operações de um período
Caso de Uso	Autor
CSU 015	Celso Pereira do Vale Júnior
Descrição	
O sistema deve permitir a consulta de operações executadas no banco de dados em um determinado período, disponibilizando as métricas de quantidade de execuções, tuplas por execução, tempo decorrido por execução, total de tuplas do período e tempo decorrido total do período.	
Critério de Verificação	
<ul style="list-style-type: none"> Ter um banco de dados selecionado. 	
Dependência	Prioridade
RU 012	Essencial

Fonte: Elaborado pelo autor.

Quadro 14 – RF 014: Visualizar o SQL de uma operação.

Identificador	Nome
RF 014	Visualizar o SQL de uma operação
Caso de Uso	Autor
CSU 010	Celso Pereira do Vale Júnior
Descrição	
O sistema deve permitir obter a consulta SQL de uma determinada operação no banco de dados Oracle selecionado.	
Critério de Verificação	
<ul style="list-style-type: none"> Ter um banco de dados selecionado. 	
Dependência	Prioridade
RU 008, RU 012	Essencial

Fonte: Elaborado pelo autor.

3.3 Requisitos de Qualidade

Nos quadros 15 ao 20 são apresentados os requisitos de qualidade do aplicativo DbMonitor.

Quadro 15 – RQ 001: Disponibilidade

Identificador	Nome
RQ 001	Disponibilidade
Caso de Uso	Autor
	Celso Pereira do Vale Júnior
Descrição	
O sistema deve estar disponível mediante a conexão com o banco de dados Firebase e com ao menos um banco de dados Oracle cadastrado pelo usuário.	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar a conexão com os bancos de dados informados. 	
Dependência	Prioridade
	Essencial

Fonte: Elaborado pelo autor.

Quadro 16 – RQ 002: Recuperação de erros.

Identificador	Nome
RQ 002	Recuperação de erros
Caso de Uso	Autor
	Celso Pereira do Vale Júnior
Descrição	
O sistema deve ser capaz de exibir os erros em tempo de execução sem comprometer o funcionamento e a integridade dos dados.	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar o comportamento do aplicativo em dispositivos com diferentes tamanhos de tela. 	
Dependência	Prioridade
	Essencial

Fonte: Elaborado pelo autor.

Quadro 17 – RQ 003: Linguagem de Programação.

Identificador	Nome
RQ 003	Linguagem de Programação
Caso de Uso	Autor
	Celso Pereira do Vale Júnior
Descrição	
O sistema deve ser desenvolvido com a utilização da linguagem de programação <i>Dart</i> utilizando o framework <i>Flutter</i> .	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar o comportamento do aplicativo em dispositivos com diferentes tamanhos de tela. 	
Dependência	Prioridade
	Essencial

Fonte: Elaborado pelo autor.

Quadro 18 – RQ 004: Plataformas.

Identificador	Nome
RQ 004	Plataformas
Caso de Uso	Autor
	Celso Pereira do Vale Júnior
Descrição	
O sistema deve ser multiplataforma, podendo ser executado em aparelhos com sistema operacional iOS versão 8.0 e Android versão Jelly Bean, v16, 4.1.x ou mais recente.	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar o comportamento do aplicativo em dispositivos com diferentes tamanhos de tela. 	
Dependência	Prioridade
	Essencial

Fonte: Elaborado pelo autor.

Quadro 19 – RQ 005: Segurança dos dados.

Identificador	Nome
RQ 005	Segurança dos dados
Caso de Uso	Autor
	Celso Pereira do Vale Júnior
Descrição	
O sistema deve garantir a segurança dos dados dos usuários salvando a senha de acesso ao sistema, utilizando <i>hash</i> .	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar a <i>hash</i> salva através do console do <i>Firebase</i>. 	
Dependência	Prioridade
	Essencial

Fonte: Elaborado pelo autor.

Quadro 20 – RQ 006: Responsividade.

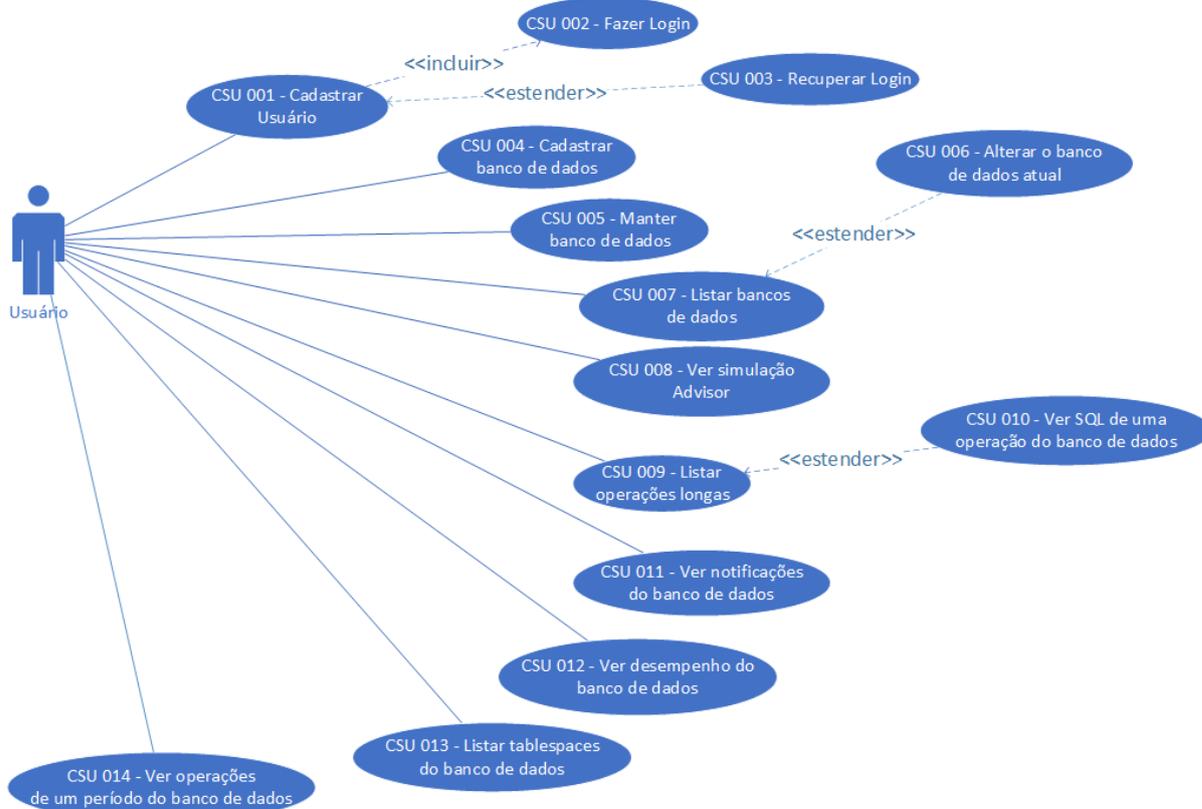
Identificador	Nome
RQ 006	Responsividade
Caso de Uso	Autor
	Celso Pereira do Vale Júnior
Descrição	
O sistema deve se adaptar a diferentes tamanhos de telas, sem comprometer a visualização dos dados.	
Critério de Verificação	
<ul style="list-style-type: none"> • Verificar o comportamento do aplicativo em dispositivos com resoluções distintas de tela. 	
Dependência	Prioridade
	Essencial

Fonte: Elaborado pelo autor.

3.4 Diagrama de casos de uso

Foi desenvolvido um diagrama de casos de uso para apresentar as funcionalidades do aplicativo, conforme Figura 1.

Figura 1: Diagrama de casos de uso.



Fonte: Elaborado pelo autor.

3.5 Casos de uso descritivos

A seguir são descritos os casos de uso apresentados na Figura 1.

CSU 001: Cadastrar Usuário.

Identificador	CSU 001
Nome	Cadastrar Usuário
Atores	Usuário
Requisitos	RF 001
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve o processo para o cadastro de um usuário.
Pré-condições	
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário abre o aplicativo; 2. O sistema exibe a tela de login; 3. O usuário seleciona a opção “Cadastrar”; 4. O sistema exibe a tela de cadastro; 5. O usuário preenche os campos; 6. O usuário seleciona a opção “Cadastrar”; 7. O sistema valida os dados informados; 8. O sistema armazena as informações no banco de dados <i>Firebase</i>; 9. O usuário é redirecionado para a tela inicial; 10. Fim do caso de uso.
Cenários alternativos	<p>(A1) Passos 5 ao 6 – O usuário cancela o cadastro</p> <ol style="list-style-type: none"> 1.a O usuário seleciona a opção “Cancelar” ou fecha o aplicativo; 2.a O sistema cancela o cadastro sem salvar dados no banco de dados <i>Firebase</i>; 3.a O sistema retorna ao passo 10 do cenário principal.
Cenário de exceção	<p>(E1) Passo 7 – O usuário informa um e-mail já cadastrado no sistema</p> <ol style="list-style-type: none"> 1.a O sistema apresenta a mensagem “E-mail já cadastrado”; 2.a O sistema retorna ao passo 5 do cenário principal. <p>(E2) Passo 7 – As senhas informadas pelo usuário são diferentes</p> <ol style="list-style-type: none"> 1.b O sistema apresenta a mensagem “As senhas informadas não conferem”; 2.b O sistema retorna ao passo 5 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor.

CSU 002: Fazer Login.

Identificador	CSU 002
Nome	Fazer Login
Atores	Usuário
Requisitos	RF 002
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve o processo para o login do usuário.
Pré-condições	
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário abre o aplicativo; 2. O sistema exibe a tela de login; 3. O usuário preenche os campos de e-mail e senha; 4. O usuário seleciona a opção “Login”; 5. O sistema valida os dados informados; 6. O usuário é redirecionado para a tela inicial; 7. Fim do caso de uso.
Cenários alternativos	<p>(A1) Passos 3 ao 4 – O usuário cancela o login</p> <ol style="list-style-type: none"> 1.a O usuário fecha o aplicativo; 2.a O sistema cancela o login; 3.a O sistema retorna ao passo 7 do cenário principal. <p>(A2) Passos 3 ao 4 – O usuário seleciona a opção de cadastro</p> <ol style="list-style-type: none"> 1.b O usuário seleciona a opção “Cadastrar”; 2.b O sistema retorna ao passo 7 do cenário principal.
Cenário de exceção	<p>(E1) Passo 5 – O usuário informa um e-mail não cadastrado no sistema ou uma senha diferente</p> <ol style="list-style-type: none"> 1.a O sistema apresenta a mensagem “E-mail ou senha inválidos”; 2.a O sistema retorna ao passo 3 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor.

CSU 003: Recuperar Login.

Identificador	CSU 003
Nome	Recuperar Login
Atores	Usuário
Requisitos	RF 002
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve o processo para recuperar dados de login do usuário.
Pré-condições	
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário abre o aplicativo; 2. O sistema exibe a tela de login; 3. O usuário seleciona a opção “Esqueci minha senha”; 4. O sistema exibe a tela de recuperação de senha; 5. O usuário preenche os dados; 6. O usuário seleciona a opção “Confirmar”; 7. O sistema valida os dados informados; 8. O sistema envia um e-mail com um link para a troca de senha; 9. O usuário acessa o link; 10. O sistema exibe uma página web; 11. O usuário informa uma nova senha; 12. O sistema exibe a mensagem “Senha alterada com sucesso”. 13. Fim do caso de uso.
Cenários alternativos	<p>(A1) Passos 5 ao 7 – O usuário cancela a recuperação de senha</p> <ol style="list-style-type: none"> 1.a O usuário seleciona a opção “Cancelar” ou fecha o aplicativo; 2.a O sistema retorna ao passo 13 do cenário principal. <p>(A2) Passos 8 ao 11 – O usuário desiste da recuperação de senha</p> <ol style="list-style-type: none"> 1.a O usuário não acessa o link enviado para o e-mail ou não informa a nova senha; 2.a O sistema retorna ao passo 13 do cenário principal.
Cenário de exceção	<p>(E1) Passo 5 – O usuário informa um e-mail não está cadastrado no sistema</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “E-mail não encontrado”; 2.a O sistema retorna ao passo 5 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor.

CSU 004: Cadastrar banco de dados.

Identificador	CSU 004
Nome	Cadastrar banco de dados
Atores	Usuário
Requisitos	RF 004
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve o cadastro de um banco de dados.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no sistema.
Pós-condições	<ul style="list-style-type: none"> • O usuário é redirecionado para a tela de seleção do banco de dados a ser acessado.
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário seleciona a opção de cadastrar um banco de dados; 2. O sistema exibe a tela de cadastro de banco de dados; 3. O usuário preenche os dados; 4. O usuário seleciona a opção “Confirmar”; 5. O sistema valida os dados informados; 6. O sistema armazena os dados do banco de dados a ser acessado no <i>Firebase</i>; 7. Fim do caso de uso.
Cenários alternativos	<p>(A1) Passos 3 ao 4 – O usuário cancela o cadastro</p> <ol style="list-style-type: none"> 1.a O usuário seleciona a opção “Cancelar” ou fecha o aplicativo; 2.a O sistema retorna ao passo 7 do cenário principal.
Cenário de exceção	<p>(E1) Passo 5 – A conexão com o banco de dados informado não foi bem sucedida</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “A conexão informada não é válida”; 2.a O sistema retorna ao passo 3 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor.

CSU 005: Manter banco de dados.

Identificador	CSU 005
Nome	Manter banco de dados
Atores	Usuário
Requisitos	RF 005
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve a edição e exclusão de um banco de dados.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no sistema.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário seleciona a opção de alterar o banco de dados; 2. O sistema exibe a tela de listagem de bancos de dados; 3. O usuário seleciona o banco de dados desejado; 4. O sistema exibe as opções de excluir e editar; 5. O usuário seleciona a opção de editar; 6. O sistema exibe a tela de edição de banco de dados; 7. O usuário altera os dados desejados; 8. O usuário seleciona a opção “Confirmar”; 9. O sistema valida os dados informados; 10. O usuário é redirecionado para a tela de seleção de banco de dados; 11. Fim do caso de uso.
Cenários alternativos	<p>(A1) Passo 5 – O usuário deseja excluir o banco de dados</p> <ol style="list-style-type: none"> 1.a O usuário seleciona a opção “Excluir”; 2.a O sistema exibe a mensagem de confirmação “Confirma a exclusão do banco de dados?”; 3.a O usuário seleciona a opção “Sim”; 4.a O sistema exclui logicamente o banco de dados; 5.a O usuário é redirecionado para a tela de seleção de banco de dados; 6.a O sistema é redirecionado para o passo 11 do cenário principal. <p>(A2) Passo 3.a – O usuário desiste de excluir o banco de dados</p> <ol style="list-style-type: none"> 1.b O usuário seleciona a opção “Não”; 2.b O usuário é redirecionado para a tela de seleção de banco de dados; 3.b O sistema é redirecionado para o passo 11 do cenário principal.

Cenário de exceção	(E1) Passo 9 – A conexão com o banco de dados informado não foi bem sucedida 1.a O sistema exibe a mensagem “A conexão informada não é válida”; 2.a O sistema retorna ao passo 7 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor.

CSU 006: Selecionar o banco de dados atual.

Identificador	CSU 006
Nome	Selecionar o banco de dados atual
Atores	Usuário
Requisitos	RF 006, RF 004
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve a seleção do banco de dados atual.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no sistema.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário seleciona a opção de selecionar o banco de dados atual; 2. O sistema exibe a tela de listagem de bancos de dados; 3. O usuário seleciona o banco de dados desejado; 4. Fim do caso de uso.
Cenários alternativos	(A1) Passo 3 – O usuário desiste de alterar o banco de dados 1.a O usuário fecha a tela do aplicativo; 2.a O sistema é redirecionado para o passo 4 do cenário principal.
Cenário de exceção	
Qualidades	

Fonte: Elaborado pelo autor.

CSU 008: Ver simulação *Advisor*.

Identificador	CSU 008
Nome	Ver simulação <i>Advisor</i>
Atores	Usuário
Requisitos	RF 007
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve o uso da tela de simulação do <i>advisor</i> para o banco de dados atual.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no sistema.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário seleciona a opção "<i>Advisor</i>"; 2. O sistema exibe a tela do <i>Advisor</i>; 3. O usuário seleciona uma das opções do filtro de categoria (<i>Cache Advisor</i>, <i>Memory Target</i>, <i>PGA</i> ou <i>Hit Ratio</i>); 4. O sistema exibe os dados do <i>Advisor</i> de acordo com o filtro selecionado; 5. Fim do caso de uso.
Cenários alternativos	<p>(A1) Passo 3 – O usuário não seleciona um filtro</p> <ol style="list-style-type: none"> 1.a O sistema exibe os dados do <i>Advisor</i> para a categoria <i>Cache Advisor</i>, categoria selecionada por padrão; 2.a O sistema é redirecionado para o passo 5 do cenário principal.
Cenário de exceção	<p>(E1) Passo 4 – A conexão com o banco de dados não foi bem sucedida</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem "A conexão com o banco de dados falhou"; 2.a O sistema retorna ao passo 5 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor.

CSU 009: Listar operações longas.

Identificador	CSU 009
Nome	Listar operações longas
Atores	Usuário
Requisitos	RF 008
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve o uso da tela de operações longas do banco de dados atual.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no sistema.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário seleciona a opção “Operações Longas”; 2. O sistema exibe a tela de operações longas; 3. O sistema exibe os dados de todas operações longas ativas no banco de dados; 4. Fim do caso de uso.
Cenários alternativos	<p>(A1) Passo 3 – O usuário realiza um filtro</p> <ol style="list-style-type: none"> 1.a O usuário preenche os dados de filtro; 2.a O usuário seleciona a opção “Filtrar”; 3.a O sistema exibe os dados de operações longas ativas no banco de dados que satisfazem os filtros selecionados; 4.a O sistema é redirecionado para o passo 4 do cenário principal.
Cenário de exceção	<p>(E1) Passo 3 – A conexão com o banco de dados não foi bem sucedida</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “A conexão com o banco de dados falhou”; 2.a O sistema retorna ao passo 4 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor.

CSU 010: Ver SQL de uma operação do banco de dados.

Identificador	CSU 010
Nome	Ver SQL de uma operação do banco de dados
Atores	Usuário
Requisitos	RF 014
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve o uso da tela de visualização do SQL de uma operação do banco de dados atual.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no sistema.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário seleciona a opção “Operações Longas”; 2. O sistema exibe a tela de operações longas; 3. O sistema exibe os dados de todas operações longas ativas no banco de dados; 4. O usuário seleciona a opção “Detalhes” de uma das operações listadas; 5. O sistema exibe a tela de SQL de uma operação; 6. Fim do caso de uso.
Cenários alternativos	<p>(A1) Passo 1 – O usuário seleciona uma operação da tela de “Operações do banco de dados”</p> <ol style="list-style-type: none"> 1.a O usuário seleciona a opção “Operações”; 2.a O sistema exibe a tela de operações por período; 3.a O sistema exibe os dados de todas operações do período selecionado do banco de dados; 4.a O usuário seleciona a opção “Detalhes” de uma das operações listadas; 5.a O sistema exibe a tela de SQL da operação selecionada; 6.a Fim do caso de uso.
Cenário de exceção	<p>(E1) Passo 3 – A conexão com o banco de dados não foi bem sucedida</p> <ol style="list-style-type: none"> 3.a O sistema exibe a mensagem “A conexão com o banco de dados falhou”; 4.a O sistema retorna ao passo 6 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor.

CSU 011: Ver notificações do banco de dados.

Identificador	CSU 011
Nome	Ver notificações do banco de dados
Atores	Usuário
Requisitos	RF 009
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve o uso da tela de notificações do banco de dados atual.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no sistema.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário seleciona a opção “Notificações”; 2. O sistema exibe a tela de notificações; 3. O sistema exibe as notificações do banco de dados; 4. Fim do caso de uso.
Cenários alternativos	
Cenário de exceção	<p>(E1) Passo 3 – A conexão com o banco de dados não foi bem sucedida</p> <ol style="list-style-type: none"> 5.a O sistema exibe a mensagem “A conexão com o banco de dados falhou”; 6.a O sistema retorna ao passo 4 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor.

CSU 012: Ver desempenho do banco de dados.

Identificador	CSU 012
Nome	Ver desempenho do banco de dados
Atores	Usuário
Requisitos	RF 010
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve o uso da tela de desempenho do banco de dados atual.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no sistema.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário seleciona a opção “Desempenho”; 2. O sistema exibe a tela de desempenho do banco de dados; 3. O sistema exibe os dados de desempenho do banco de dados para os últimos 45 minutos; 4. Fim do caso de uso.
Cenários alternativos	<p>(A1) Passo 3 – O usuário realiza um filtro</p> <ol style="list-style-type: none"> 1.a O usuário preenche os dados de filtro; 2.a O usuário seleciona a opção “Filtrar”; 3.a O sistema exibe os dados de desempenho do banco de dados para as categorias selecionadas; 4.a O sistema é redirecionado para o passo 4 do cenário principal.
Cenário de exceção	<p>(E1) Passo 3 – A conexão com o banco de dados não foi bem sucedida</p> <ol style="list-style-type: none"> 7.a O sistema exibe a mensagem “A conexão com o banco de dados falhou”; 8.a O sistema retorna ao passo 4 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor.

CSU 013: Listar tablespaces do banco de dados.

Identificador	CSU 013
Nome	Listar tablespaces do banco de dados
Atores	Usuário
Requisitos	RF 011
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve o uso da tela de listagem das <i>tablespaces</i> do banco de dados atual.
Pré-condições	<ul style="list-style-type: none"> • O usuário deve estar logado no sistema.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário seleciona a opção “Tablespaces”; 2. O sistema exibe a tela de listagem das <i>tablespaces</i> do banco de dados; 3. O sistema exibe os dados de <i>tablespaces</i> do banco de dados; 4. Fim do caso de uso.
Cenários alternativos	
Cenário de exceção	<p>(E1) Passo 3 – A conexão com o banco de dados não foi bem sucedida</p> <ol style="list-style-type: none"> 1.a O sistema exibe a mensagem “A conexão com o banco de dados falhou”; 2.a O sistema retorna ao passo 4 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor.

CSU 014: Ver operações de um período do banco de dados.

Identificador	CSU 014
Nome	Ver operações de um período do banco de dados
Atores	Usuário
Requisitos	RF 013
Responsável	Celso Pereira do Vale Júnior
Descrição/Resumo	Este caso de uso descreve o uso da tela de listagem das <i>tablespaces</i> do banco de dados atual.
Pré-condições	<ul style="list-style-type: none"> O usuário deve estar logado no sistema.
Pós-condições	
Cenário Principal	<ol style="list-style-type: none"> O usuário seleciona a opção “Operações”; O sistema exibe a tela de operações de um período; O sistema exibe os dados das operações do banco de dados dos últimos 45 minutos; Fim do caso de uso.
Cenários alternativos	<p>(A1) Passo 4 – O usuário realiza um filtro</p> <ol style="list-style-type: none"> O usuário preenche os dados de filtro; O usuário seleciona a opção “Filtrar”; O sistema exibe os dados das operações do banco de dados para o período selecionado no filtro; O sistema é redirecionado para o passo 4 do cenário principal.
Cenário de exceção	<p>(E1) Passo 3 – A conexão com o banco de dados não foi bem sucedida</p> <ol style="list-style-type: none"> O sistema exibe a mensagem “A conexão com o banco de dados falhou”; O sistema retorna ao passo 4 do cenário principal.
Qualidades	

Fonte: Elaborado pelo autor.

3.6 Dicionário de dados do sistema

DD 001: Dados do Usuário.

Identificador	Descrição			
DD 001	Dados do Usuário			
Identificador	Tamanho	Tipo	Formato	Descrição
Nome	50	String		Nome do usuário
E-mail	100	String	\w+@\w+\.\w+	E-mail do usuário
Senha	14	String		Senha do usuário

Fonte: Elaborado pelo autor.

DD 002: Dados do banco de dados.

Identificador	Descrição			
DD 002	Dados do banco de dados			
Identificador	Tamanho	Tipo	Formato	Descrição
Nome	50	String		Nome único do banco de dados no sistema
Host	100	String		<i>Host</i> do banco de dados
Porta	5	Inteiro		Porta do banco de dados
Usuário	50	String		Usuário do banco de dados
Senha	30	String		Senha do usuário do banco de dados
SID	10	String		Nome da instância do banco de dados

Fonte: Elaborado pelo autor.

4 DESENVOLVIMENTO DO APLICATIVO DBMONITOR

Neste capítulo são apresentadas as tecnologias utilizadas para o desenvolvimento do aplicativo DbMonitor. Para a construção do aplicativo foram utilizadas a linguagem de programação Dart e o *framework* Flutter e, para a comunicação com os bancos de dados Oracle, foi construída uma API utilizando o *.NET Framework*.

4.1 *Framework* Flutter

Flutter é um conjunto de ferramentas para construção de interfaces gráficas multiplataforma, desenvolvido pelo Google e lançado em maio de 2017. Foi projetado para possibilitar o reuso de código, projetando aplicativos para os sistemas operacionais iOS, Android e Web (FLUTTER, 2020).

Durante o desenvolvimento, os aplicativos são executados em uma máquina virtual que possibilita o *Hot Reload*, que reflete as alterações feitas no código instantaneamente sem a necessidade de recompilar o aplicativo (FLUTTER, 2020).

4.1.1 *Camadas Arquiteturais*

O Flutter foi projetado para trabalhar com camadas extensíveis, as quais dependem da camada subjacente. Essas camadas não possuem acesso privilegiado à camada inferior e qualquer parte do *framework* foi projetada para ser opcional e substituível (FLUTTER, 2020).

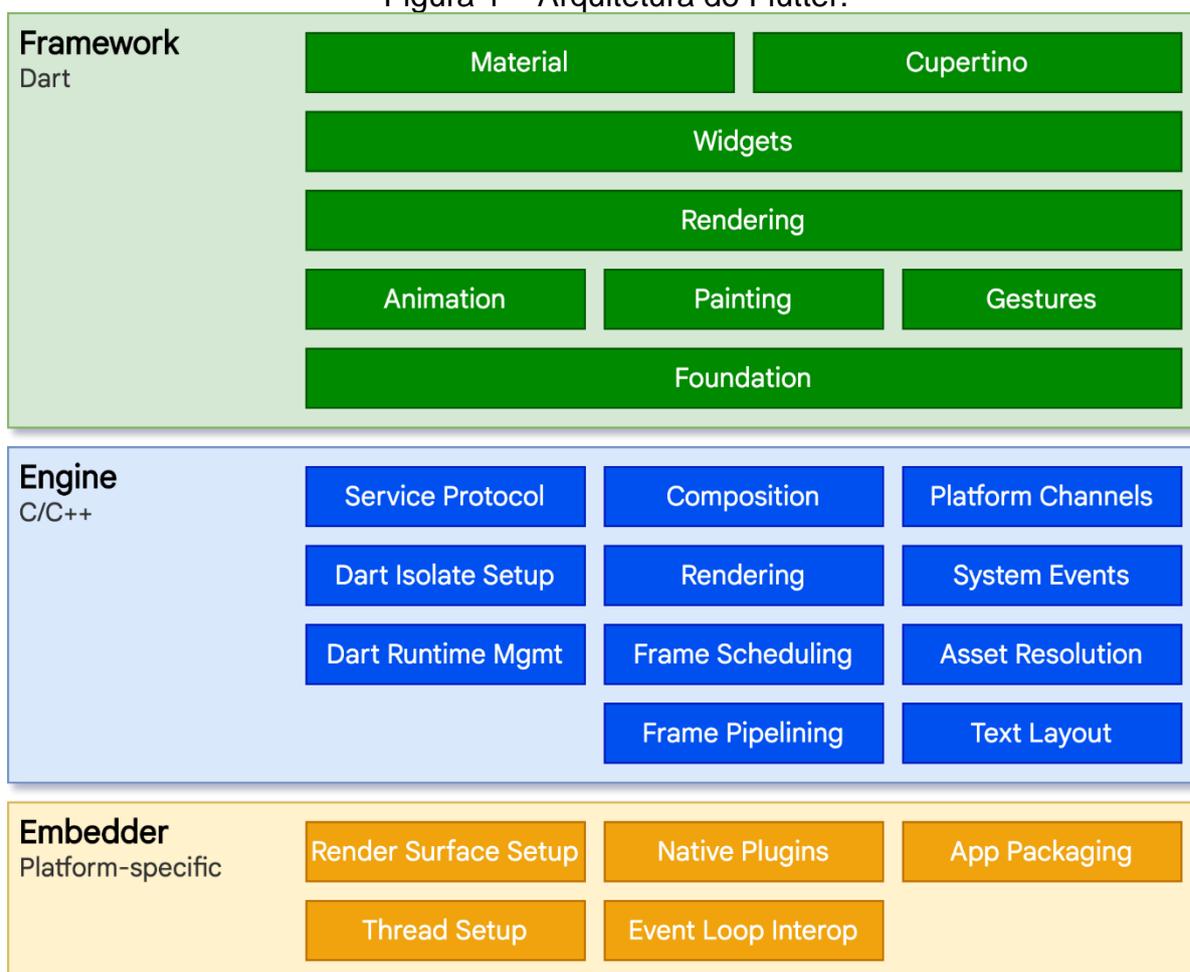
Na Figura 1 são apresentadas as camadas e suas funções, sendo a superior à camada do *framework*, construída em Dart e composta por:

- Bibliotecas *Material* e *Cupertino*, que implementam os estilos de *widgets* do Android e iOS, respectivamente;
- Camada *Widget*, que são os blocos de construção da interface do aplicativo;
- Camada de renderização, que provê a abstração para lidar com o *layout*;
- Classes responsáveis pela animação, cores e detecção de gestos;
- Classes base que oferecem abstrações comuns sobre outras classes base subjacentes.

Em seguida está a camada *Engine*, que é escrita em sua maior parte em C++, inclui a implementação de baixo nível da API central do Flutter, incluindo os gráficos, *layouts* de texto, manipulação de arquivos, rede, suporte à acessibilidade, dentre outros.

A última camada é o *Embedder*, que é o ponto de entrada para os sistemas operacionais específicos e gerencia o acesso aos recursos nativos destes.

Figura 1 – Arquitetura do Flutter.



Fonte: Flutter, 2020.

4.1.2 Widgets

No Flutter, os widgets são blocos básicos de construção da interface e cada um possui uma declaração imutável de uma parte da interface de usuário. Eles formam uma estrutura hierárquica, aninhados em seu Widget pai, carregando esta estrutura até o Widget raiz do aplicativo.

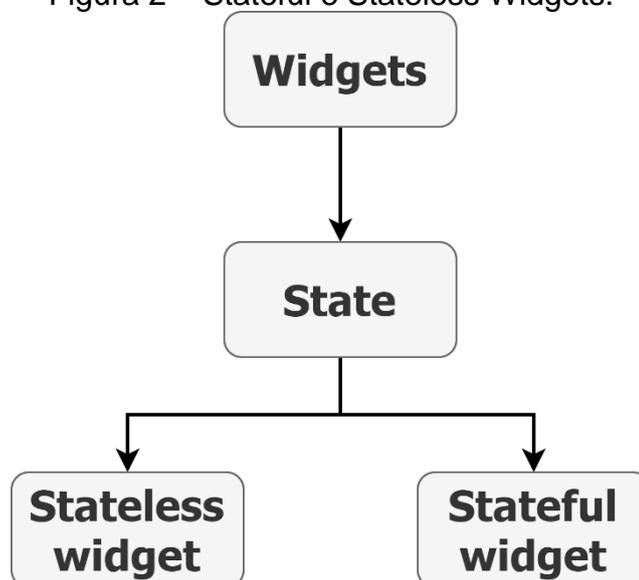
Em *frameworks* de interface tradicionais, o estado inicial da interface é descrito uma vez e então atualizado separadamente, em tempo de execução, em um código feito pelo programador. Por outro lado, o Flutter desacopla a interface do seu estado subjacente e o framework é responsável pela configuração para criar e atualizar a interface conforme necessário.

Esta implementação traz como principais benefícios:

- A extensibilidade ilimitada dos elementos da interface;
- Evita um gargalo de performance na construção da tela, pois não tem a necessidade de alternar entre o código Flutter e o código da plataforma;
- Desacoplamento do comportamento da aplicação de requisitos do sistema operacional.

Os Widgets são classificados como *Stateful* e *Stateless*, conforme mostra a Figura 2. Para os Widgets que não têm propriedades que se alteram ao longo do tempo e tem uma exibição imutável, são utilizados os *Stateless Widgets*. Já para aqueles que tem características que se alteram baseadas na interação do usuário, são utilizados os *Stateful Widgets*.

Figura 2 – Stateful e Stateless Widgets.



Fonte: Adaptado de KANSAL, 2020.

No aplicativo DbMonitor, os *Stateless Widget* são utilizados, por exemplo, na tela de login, onde as interações do usuário não afetam o que é exibido na tela. Já os *Stateful Widget* são utilizados, por exemplo, na tela de desempenho, onde a seleção dos filtros altera a exibição dos gráficos.

4.2 Linguagem de programação Dart

Dart é uma linguagem originalmente desenhada para ser utilizada em aplicações web, apresentada pelo Google em outubro de 2011, porém sua principal utilização é para o desenvolvimento de aplicações que utilizam o framework Flutter (SAMPAIO, 2015).

Dart tem a sintaxe baseada na linguagem C, porém sua compilação é baseada em códigos Javascript. Possui suporte aos principais conceitos da orientação a objetos, como interfaces, classes abstratas, tipos genéricos e tipos opcionais (SAMPAIO, 2015).

Além do uso em projetos Flutter, a Dart possibilita a criação de: (a) aplicações de execução direta via terminal de comandos; (b) aplicações *Google App Engine*, para execução na nuvem do Google; (c) *plugins*, para o navegador Google Chrome; e (d) aplicações web simples (SAMPAIO, 2015).

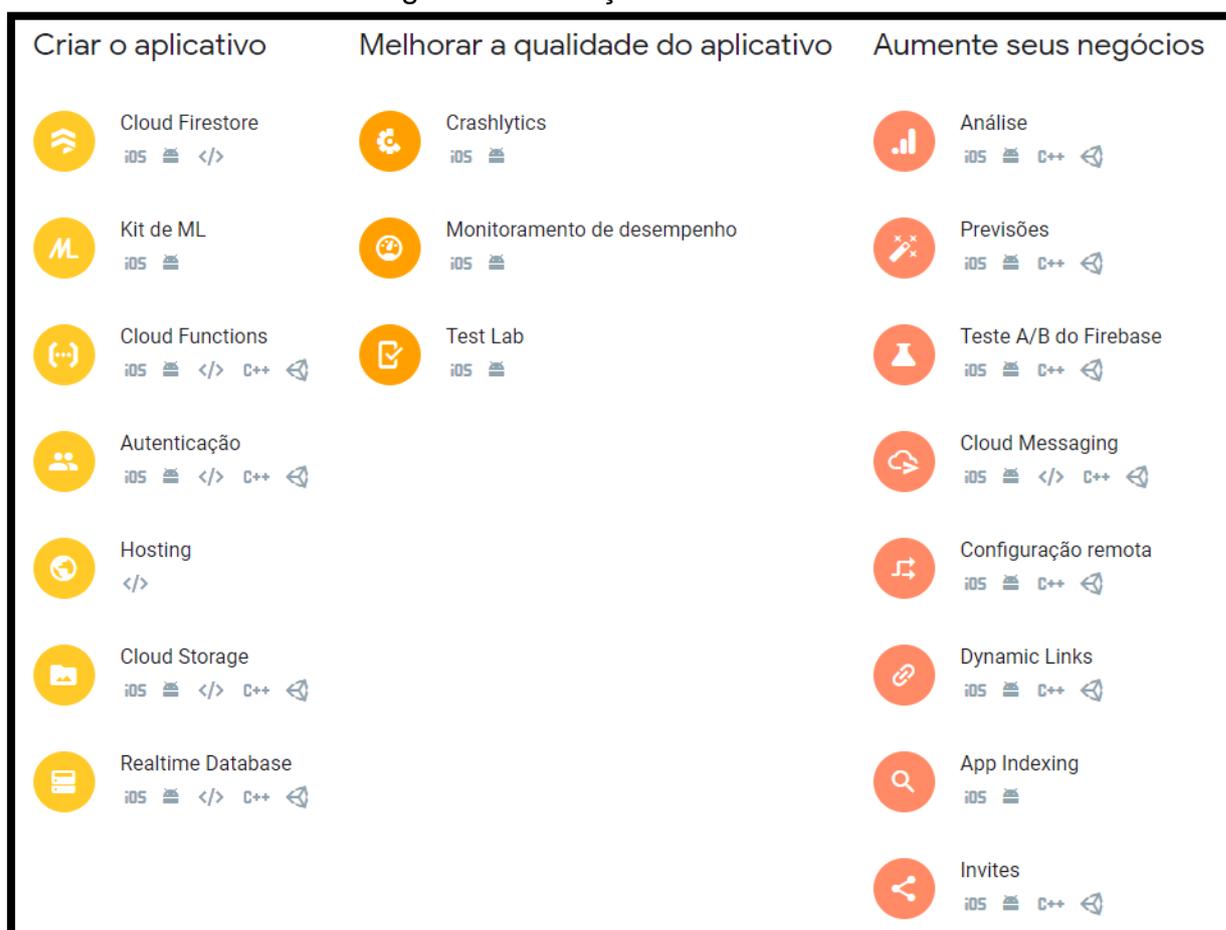
4.3 Firebase

Firebase é uma plataforma de desenvolvimento de aplicativos do tipo *Backend-as-a-Service (BaaS)* que provê serviços como banco de dados em tempo real, armazenamento em nuvem, autenticação e *machine learning* (FLUTTER, 2020).

No aplicativo DbMonitor, o Firebase é utilizado para realizar o cadastramento e autenticação dos usuários, recuperação de senhas e armazenamento em banco de dados, utilizando o plano gratuito.

A Figura 3 apresenta o catálogo de produtos que são disponibilizados pelo Firebase.

Figura 3 – Serviços do Firebase.



Fonte: Firebase, 2020.

4.3.1 Autenticação usando Firebase

O aplicativo DbMonitor utiliza o *Firebase Authentication* para realizar o cadastramento, autenticação e recuperação de senha dos usuários.

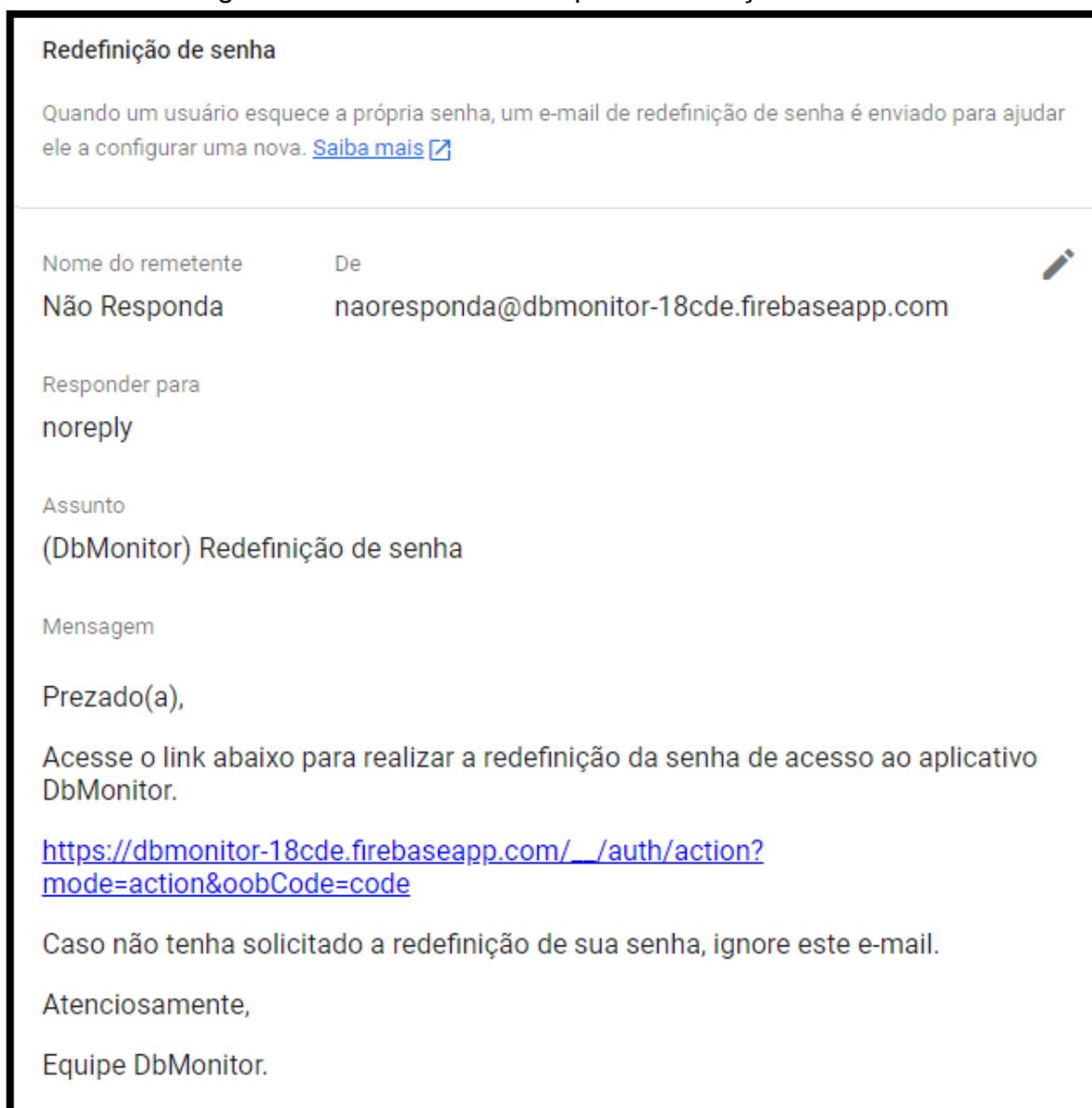
O Firebase oferece a possibilidade de autenticar usuários utilizando os serviços do próprio Firebase, através do endereço de e-mail e senha, número de telefone ou *link* por e-mail. Oferece, ainda, a possibilidade de utilizar uma conta existente dos sites Google, Facebook, Apple, Play Games, Game Center, GitHub, Microsoft, Twitter e Yahoo (FIREBASE, 2020). Dentre as opções de autenticação disponibilizadas, o aplicativo DbMonitor utiliza a autenticação por e-mail e senha.

O Firebase utiliza uma versão modificada do algoritmo Scrypt para gerar os *hashs* das senhas armazenadas (FIREBASE, 2020). O Scrypt é muito utilizado para aplicações em criptomoedas como uma alternativa ao SHA-256, ambos possuindo complexidade semelhante. O Scrypt evita o problema em que circuitos integrados

específicos para o processamento de *hashs* tenham vantagem de performance quando comparados ao processamento com CPUs e GPUs (MATT, 2018).

Após o cadastro dos usuários no aplicativo DbMonitor, é disponibilizada a possibilidade de recuperar a senha a partir de um e-mail enviado pelo *Firebase Authentication*. A Figura 4 apresenta o modelo de e-mail que será enviado ao usuário para alterar sua senha.

Figura 4 – Modelo de e-mail para redefinição de senha.



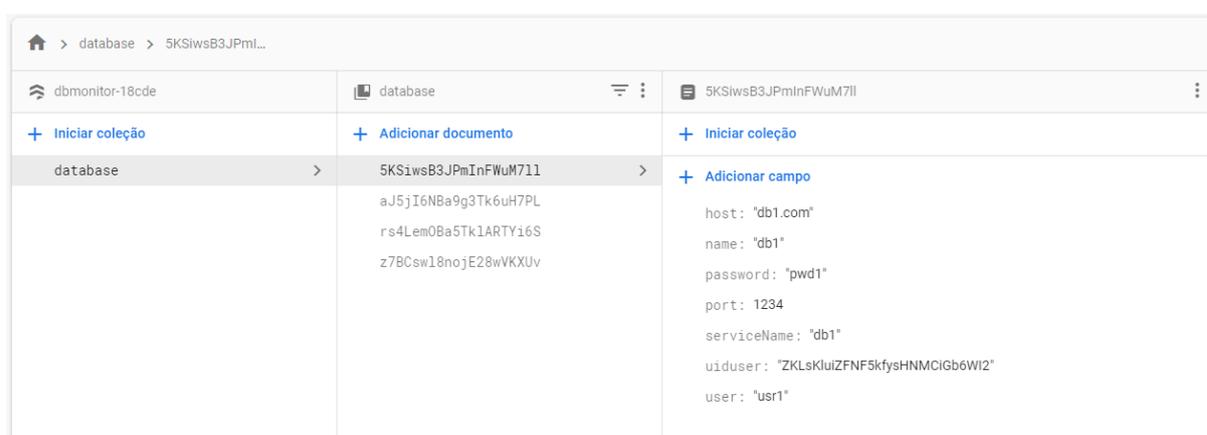
Fonte: Elaborado pelo autor.

4.3.2 Firebase Cloud Firestore

O *Cloud Firestore* é um banco de dados não relacional em nuvem disponibilizado no rol de serviços do *Firebase*. É utilizado, no aplicativo *DbMonitor*, para persistir os dados dos bancos de dados a serem acessados pelo aplicativo, conforme exibido na Figura 5.

O *Cloud Firestore* segue o modelo *NoSQL* para armazenar os dados em forma de mapeamento de chaves e valores, permitindo consultas expressivas, eficientes e flexíveis aos dados armazenados (FIREBASE, 2020).

Figura 5 – Cloud Firestore do aplicativo *DbMonitor*.

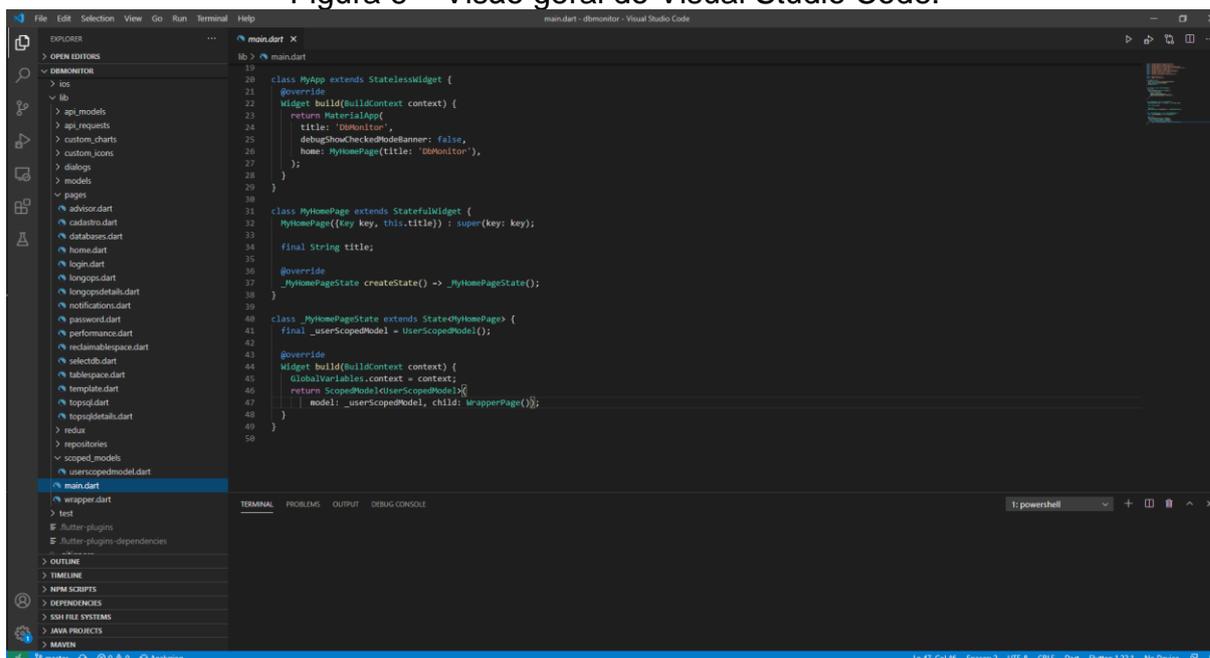


Fonte: Elaborado pelo autor.

4.4 Visual Studio Code

O Visual Studio Code foi escolhido para o desenvolvimento do aplicativo *DbMonitor* devido a elevada performance comparado ao *Android Studio* (PAIM, 2019), sendo que o Ambiente de Desenvolvimento Integrado (*Integrated Development Environment, IDE*) recomendada pelo site oficial do *Flutter*. A visão geral do *Visual Studio Code* é apresentada na Figura 6.

Figura 6 – Visão geral do Visual Studio Code.



Fonte: Elaborado pelo autor.

4.5 API DbMonitor

Para realizar a comunicação do aplicativo DbMonitor com os bancos de dados Oracle cadastrados neste, foi construída uma API utilizando o *.NET Framework* em conjunto com a biblioteca *Oracle Managed Data Access*.

4.5.1 *.NET Framework*

O *.NET Framework* (.NET) é uma tecnologia construída pela Microsoft que visa uma plataforma única para desenvolvimento e execução das aplicações, provendo um ambiente de desenvolvimento consistente e orientado a objetos (MICROSOFT, 2020).

O funcionamento do *.NET Framework* é semelhante à plataforma Java, na qual o programador deixa de escrever código específico para um sistema e passa a escrever para o *.NET* e, então, o código é executado sobre um Ambiente de Execução Independente de Linguagem (*Common Language Runtime, CLR*) (MICROSOFT, 2020).

4.5.2 *Oracle Data Provider para .NET*

O *Oracle Data Provider for .NET* (ODP.NET) provê acesso rápido e seguro de clientes *.NET* a bancos de dados Oracle (ORACLE, 2014).

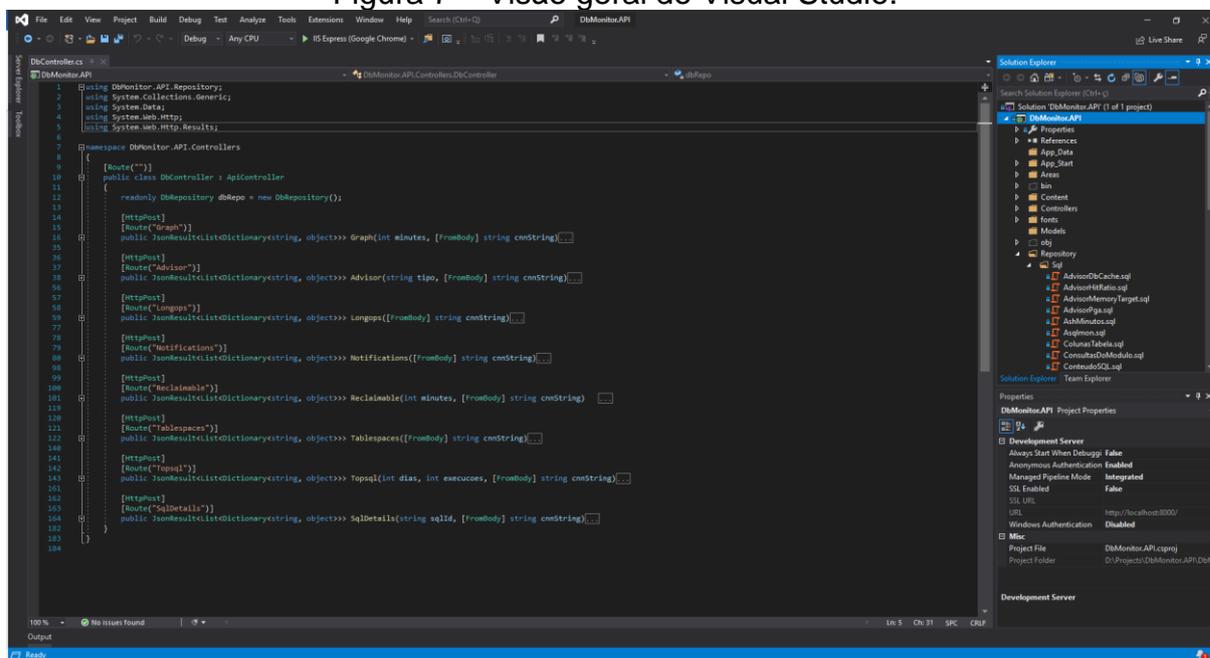
O ODP.NET oferece as bibliotecas gerenciadas e não gerenciadas, sendo a primeira um subconjunto da segunda. O .NET utilizada em todo o código possui funções gerenciadas, podendo assim, ser executado em ambientes restritivos, como a web (DEVART, 2020).

Para acesso aos bancos de dados Oracle na API, foram utilizadas as bibliotecas gerenciadas, onde os dados de conexão são repassados do aplicativo DbMonitor à API através de chamadas POST utilizando o protocolo *Hyper Text Transfer Protocol Secure* (HTTPS) para garantir a segurança do tráfego de informações.

4.5.3 Visual Studio Community

O Visual Studio foi criado pela Microsoft e pode ser definido como uma das melhores opções de IDE do mercado, especialmente para o desenvolvimento de aplicações baseadas na tecnologia .NET, onde oferece uma suíte de ferramentas integradas (IMPACTA, 2017). Uma tela de exemplo do Visual Studio pode ser vista na Figura 7.

Figura 7 – Visão geral do Visual Studio.



Fonte: Elaborado pelo autor.

5 IMPLEMENTAÇÕES E RESULTADOS

Neste capítulo são apresentadas as telas do aplicativo DbMonitor, descrevendo os seus componentes e detalhando as ações que podem ser realizadas pelos usuários, além de apresentar uma comparação entre a visualização dos dados nativa do Oracle e a visualização que o aplicativo oferece.

5.1 Telas do Aplicativo

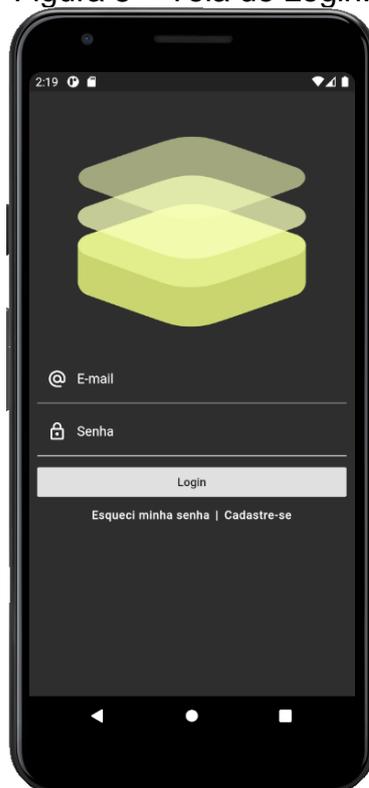
Esta seção apresenta as telas do aplicativo DbMonitor.

5.1.1 Tela de Login

A Figura 8 apresenta a tela inicial do aplicativo, onde o usuário tem a opção de realizar o *login*, recuperar sua senha ou cadastrar-se. Essa tela é composta pelos componentes:

- E-mail: Campo onde o usuário informa um e-mail previamente cadastrado;
- Senha: Campo onde o usuário informa a senha correspondente ao e-mail cadastrado;
- Login: Botão que ao ser pressionado irá realizar a validação dos campos de e-mail e senha, verificando se estes campos correspondem aos dados cadastrados. Caso os dados sejam validados com sucesso, o usuário será redirecionado para a Tela Principal e, caso contrário, será exibida uma mensagem informando o erro;
- Esqueci minha senha: Botão que irá redirecionar o usuário para a Tela de Recuperação de Senha;
- Cadastre-se: Botão que irá redirecionar o usuário para a Tela de Cadastro.

Figura 8 – Tela de Login.



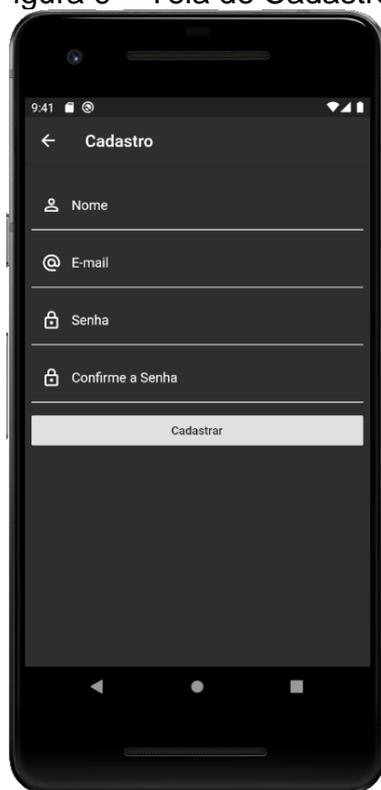
Fonte: Elaborado pelo autor.

5.1.2 Tela de Cadastro

A Figura 9 apresenta a tela onde o usuário realiza o seu cadastro no aplicativo DbMonitor, informando os seguintes dados:

- Nome: Campo onde o usuário informa o seu nome;
- E-mail: Campo onde o usuário informa o seu e-mail;
- Senha: Campo onde o usuário informa uma senha que contenha, no mínimo, 8 dígitos;
- Confirme a Senha: Campo onde o usuário informa a mesma senha que foi informada no campo “Senha”.

Figura 9 – Tela de Cadastro.



Fonte: Elaborado pelo autor.

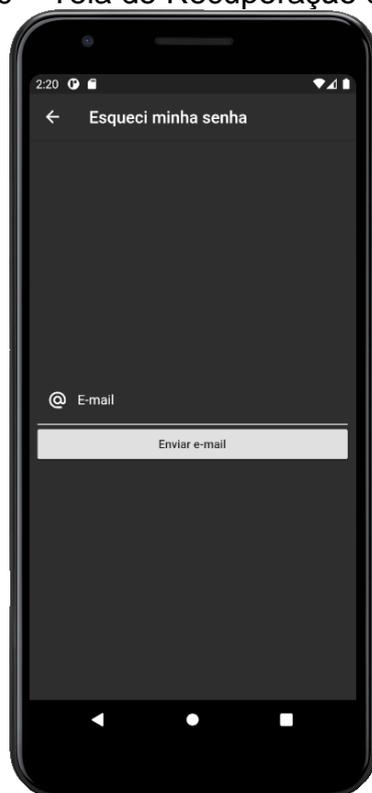
Ao pressionar o botão “Cadastrar”, o aplicativo irá validar se o e-mail informado já está cadastrando, exibindo uma mensagem de erro caso esteja.

O aplicativo também irá validar se o campo “Senha” corresponde ao campo “Confirme a Senha” e se possuem no mínimo 8 caracteres.

5.1.3 Tela de Recuperação de Senha

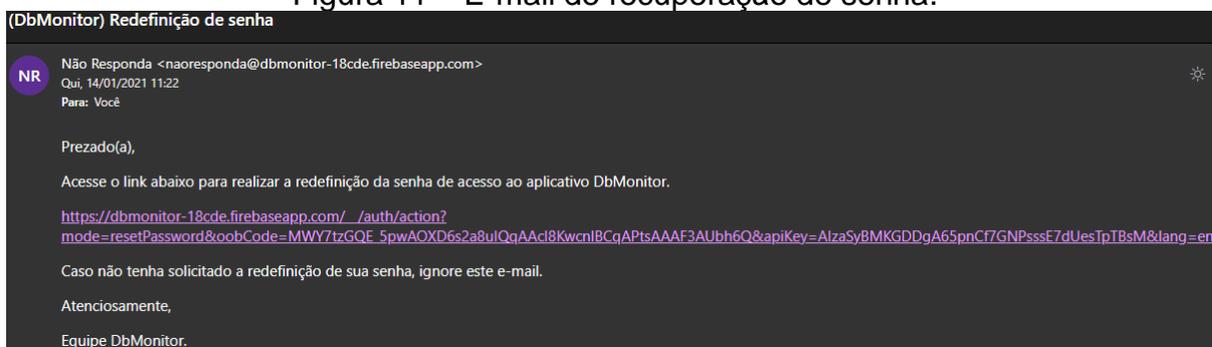
Nesta tela, Figura 10, o usuário deverá informar um e-mail previamente cadastrado no aplicativo e, ao pressionar o botão “Enviar e-mail”, o aplicativo irá validar se o e-mail corresponde a um e-mail cadastrado. Caso a validação obtenha êxito, o e-mail com as instruções de troca de senha será enviado como visto na Figura 11. Caso contrário, será exibida uma mensagem de erro como visto na Figura 12.

Figura 10 – Tela de Recuperação de Senha.



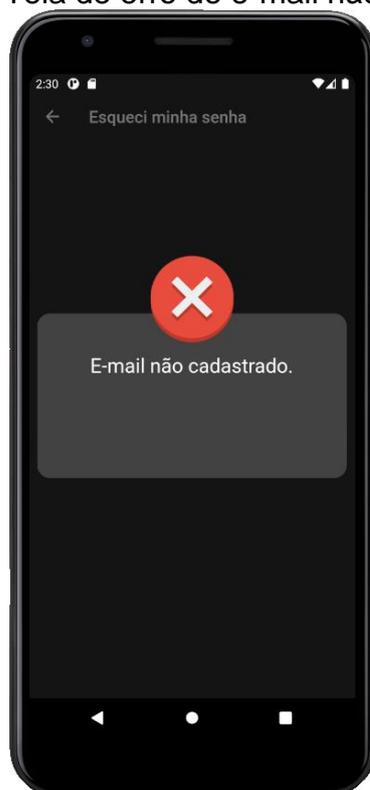
Fonte: Elaborado pelo autor.

Figura 11 – E-mail de recuperação de senha.



Fonte: Elaborado pelo autor.

Figura 12 – Tela de erro de e-mail não cadastrado.



Fonte: Elaborado pelo autor.

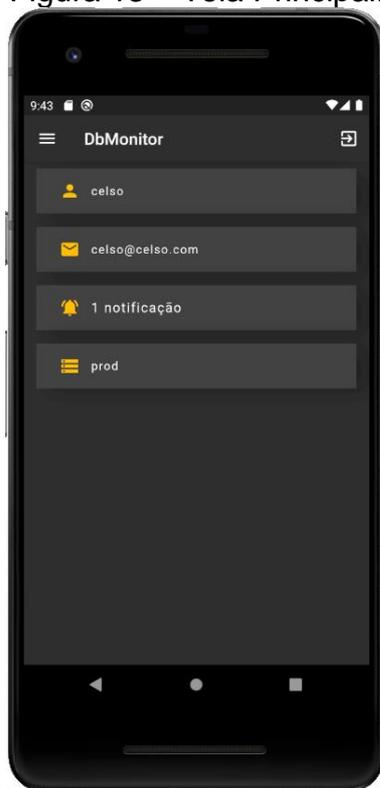
5.1.4 Tela Principal

Nesta tela são exibidas as seguintes informações, conforme mostra a Figura 13:

- Nome do usuário autenticado;
- E-mail do usuário autenticado;
- Quantidade de notificações do banco de dados selecionado, sendo exibido “0 notificações” caso nenhum banco de dados esteja selecionado;
- Nome do banco de dados selecionado, sendo exibido “Nenhum banco de dados selecionado” caso nenhum banco de dados esteja selecionado;

Ao pressionar as opções de quantidade de notificações ou nome do banco de dados selecionado, o usuário será redirecionado para a Tela de Seleção de Banco de Dados.

Figura 13 – Tela Principal.



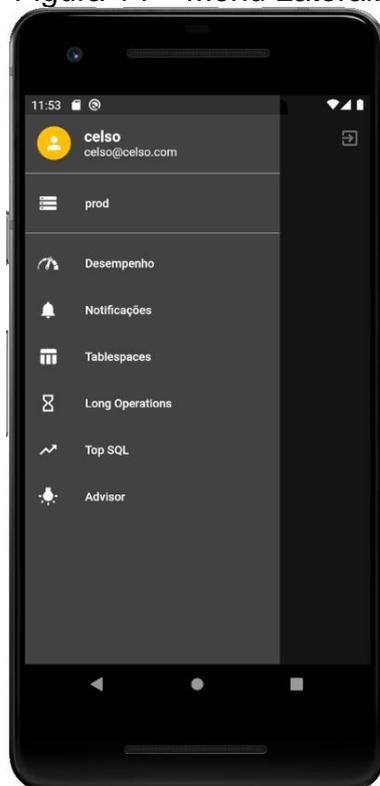
Fonte: Elaborado pelo autor.

Essa tela, Figura 13, contém com um menu lateral, acessado através do ícone no canto esquerdo superior, que será exibida em todas as telas, com exceção das telas de Login, Recuperação de Senha e Cadastro. Esse menu possibilita a navegação entre as telas do aplicativo, com as seguintes opções, conforme exibido na Figura 14:

- Nome e e-mail do usuário: Exibe o nome e e-mail do usuário autenticado e, ao ser pressionada, redireciona o usuário para a Tela Principal;
- Nome do banco de dados selecionado: Exibe o nome do banco de dados selecionado, sendo exibido “Nenhum banco de dados selecionado” caso nenhum banco de dados esteja selecionado. Ao pressionar esta opção, o usuário é redirecionado para a Tela de Seleção de Banco de Dados;
- Desempenho: O usuário será redirecionado para a Tela de Desempenho;
- Notificações: O usuário será redirecionado para a Tela de Notificações;
- *Tablespaces*: O usuário será redirecionado para a Tela de *Tablespaces*;
- *Long Operations*: O usuário será redirecionado para a Tela de *Long Operations*;

- *Top SQL*: O usuário será redirecionado para a Tela de *Top SQL*;
- *Advisor*: O usuário será redirecionado para a Tela de *Advisor*.

Figura 14 – Menu Lateral.



Fonte: Elaborado pelo autor.

5.1.5 Tela de Seleção de Banco de Dados

Esta tela permite que o usuário selecione o banco de dados que a aplicação irá realizar o monitoramento. Conforme exibido na Figura 15, são listados os bancos de dados cadastrados pelo usuário. O usuário poderá cadastrar um novo banco de dados pressionando o ícone “+”, o qual o irá redirecionar para a Tela de Cadastro de Banco de Dados.

Figura 15 – Tela de Seleção de Banco de Dados.



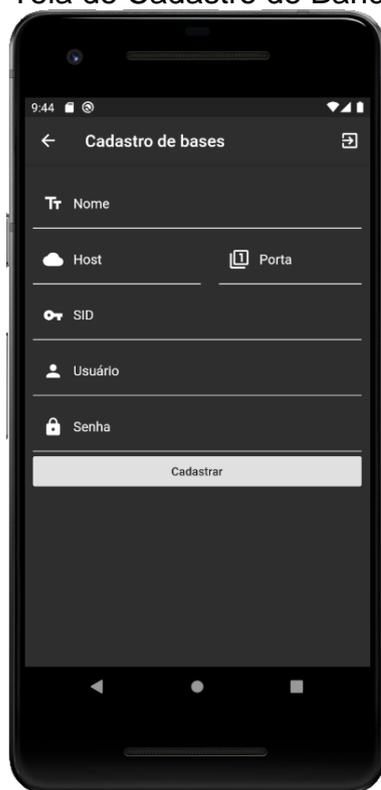
Fonte: Elaborado pelo autor.

5.1.6 Tela de Cadastro de Banco de Dados

Esta tela permite que o usuário cadastre um novo banco de dados para que a aplicação realize o monitoramento. Conforme a Figura 16, a tela possui os seguintes componentes:

- Nome: Nome do banco de dados na aplicação DbMonitor;
- Host: Nome ou endereço de host do banco de dados;
- Porta: Porta do banco de dados;
- SID: Nome da instância do banco de dados ou nome do serviço de rede do banco de dados;
- Usuário: Nome do usuário do banco de dados;
- Senha: Senha do usuário do banco de dados;
- Cadastrar: Botão que ao ser pressionado realiza a validação da tela, verificando se todos os campos foram preenchidos. Caso a validação resulte em êxito, o banco de dados é cadastrado e, caso contrário, exibe uma mensagem de erro.

Figura 16 – Tela de Cadastro de Banco de Dados.



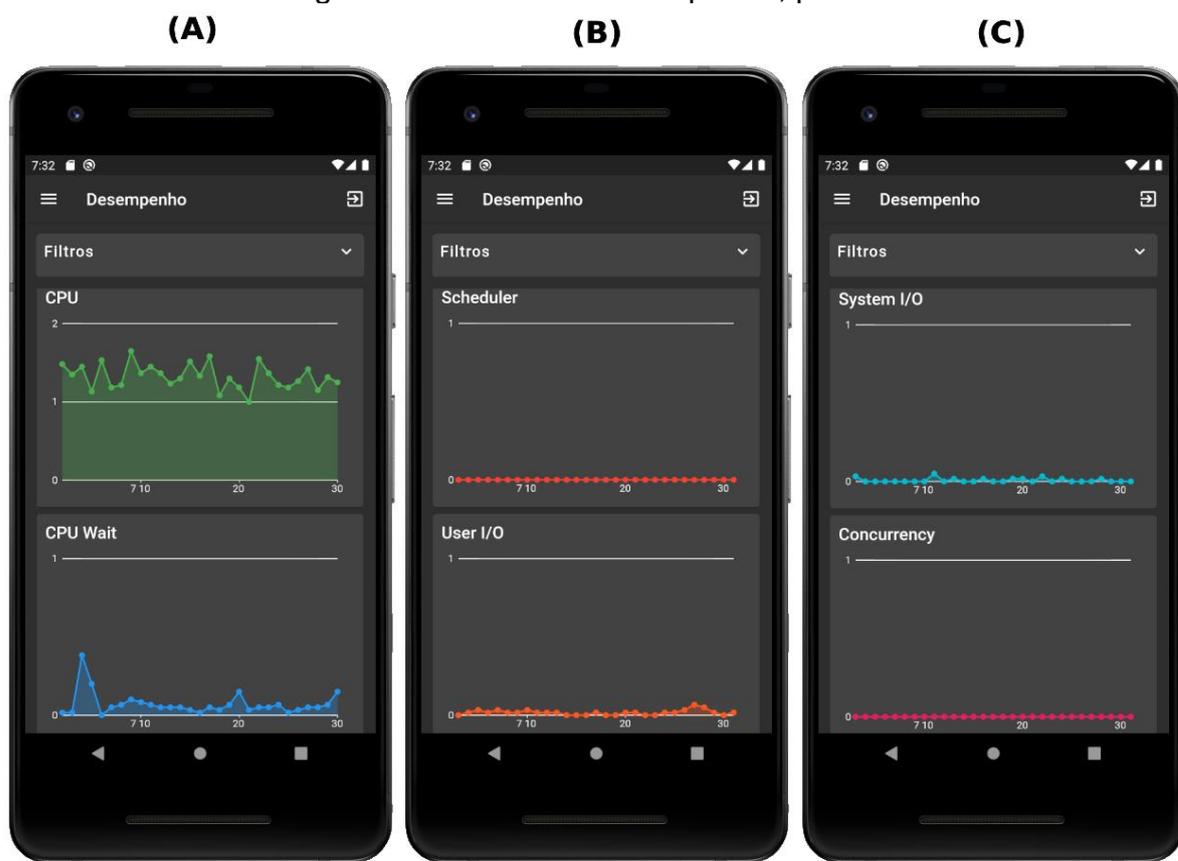
Fonte: Elaborado pelo autor.

5.1.7 Tela de Desempenho

Esta tela exibe os dados de desempenho do banco de dados selecionado, caso algum tenha sido selecionado.

Na Figura 17 são exibidos os gráficos para as categorias (a) *CPU* e *CPU Wait*, (b) *Scheduler* e *User I/O*, (c) *System I/O* e *Concurrency*.

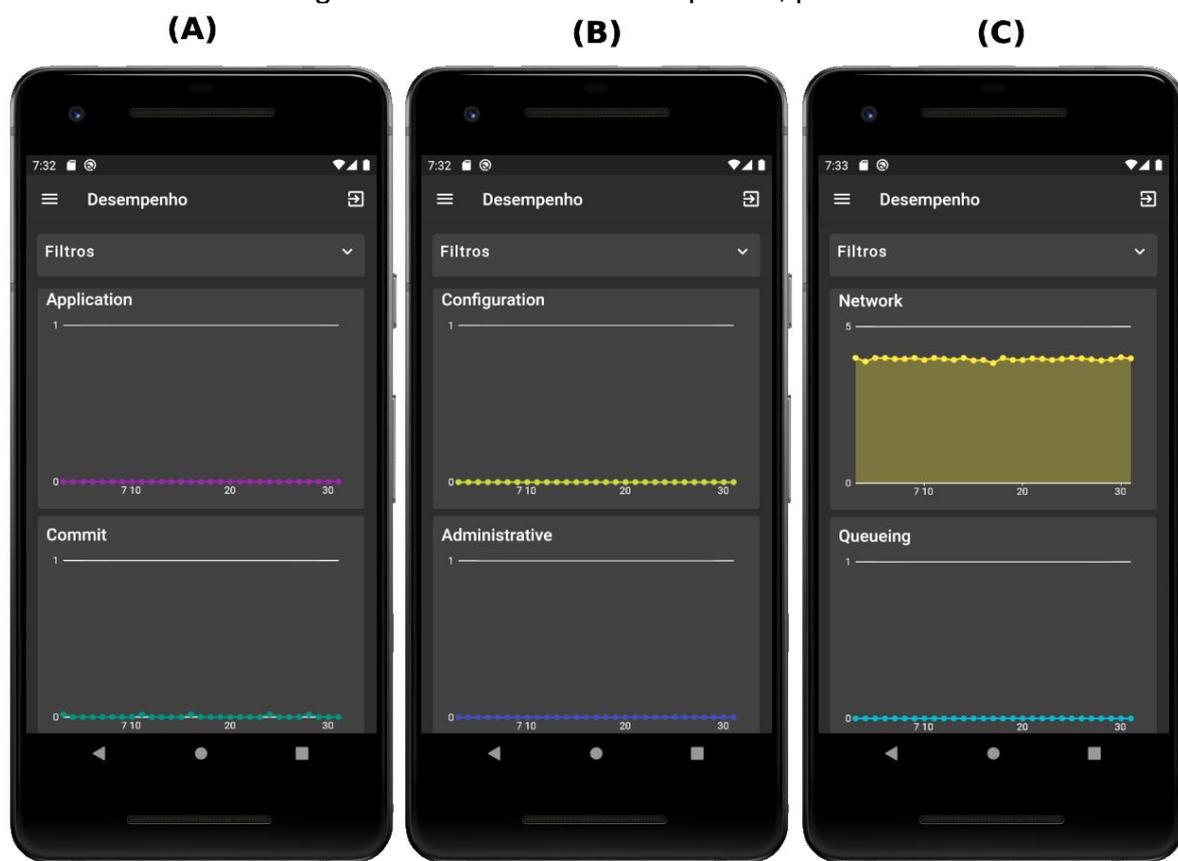
Figura 17 – Telas de desempenho, parte 1.



Fonte: Elaborado pelo autor.

Na Figura 18 são exibidos os gráficos para as categorias (a) *Application e Commit*, (b) *Configuration e Administrative*, (c) *Network e Queuing*.

Figura 18 – Telas de desempenho, parte 2.



Fonte: Elaborado pelo autor.

Na Figura 19 são exibidos os gráficos para as categorias *Cluster* e *Other*.

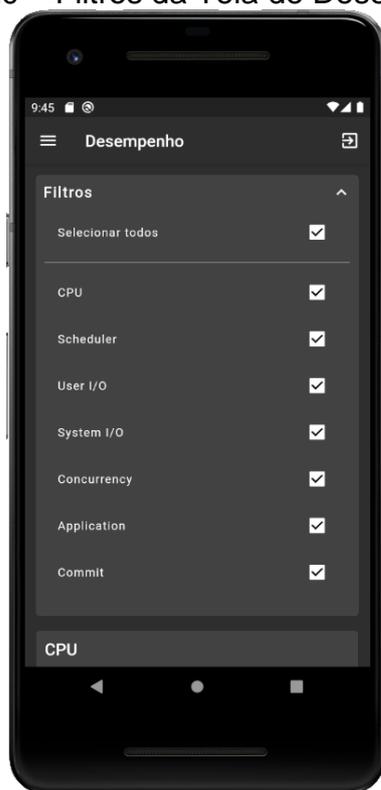
Figura 19 – Telas de desempenho, parte 3.



Fonte: Elaborado pelo autor.

A tela ainda permite a realização do filtro de quais gráficos serão exibidos, a partir da abertura do menu de filtros e seleção dos gráficos na lista exibida. A Figura 20 exibe o menu de filtros.

Figura 20 – Filtros da Tela de Desempenho.

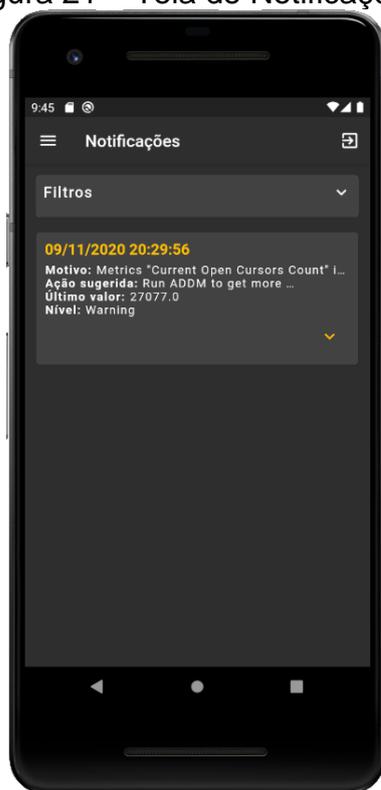


Fonte: Elaborado pelo autor.

5.1.8 Tela de Notificações

Esta tela, Figura 21, exibe as notificações geradas pelo banco de dados.

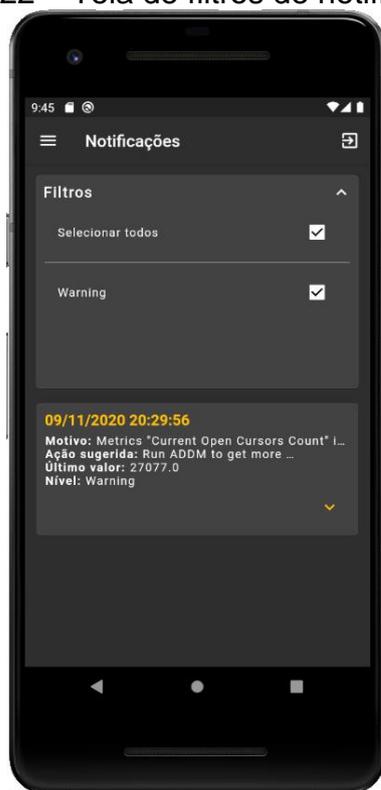
Figura 21 – Tela de Notificações.



Fonte: Elaborado pelo autor.

A Figura 22 exibe a tela de filtros de notificações, que permite a realização de filtros de acordo com o nível das notificações. A lista de filtros é construída dinamicamente de acordo com as opções disponíveis no banco.

Figura 22 – Tela de filtros de notificações.

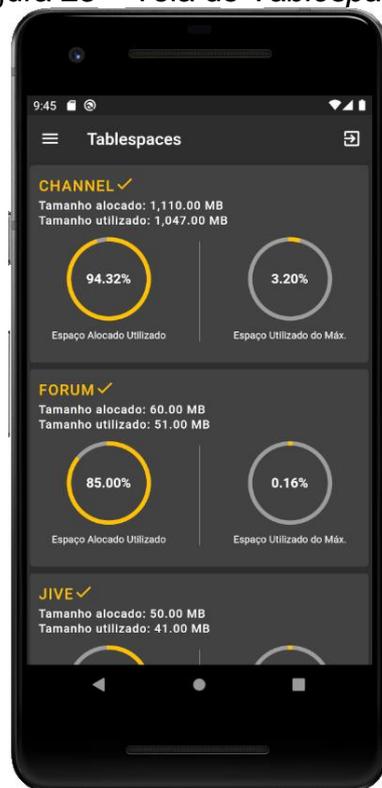


Fonte: Elaborado pelo autor.

5.1.9 Tela de *Tablespaces*

Esta tela, Figura 23, exibe uma lista das *tablespaces* que estão cadastradas no banco de dados selecionado, caso algum tenha sido selecionado. Os dados exibidos por esta tela são:

- Nome da *tablespace*;
- Status da *tablespace* (Ativa ou Inativa), indicado pelo ícone ao lado do nome da *tablespace*;
- Tamanho alocado para a *tablespace*;
- Tamanho utilizado pela *tablespace*;
- Porcentagem do espaço alocado utilizado;
- Porcentagem do espaço utilizado do máximo.

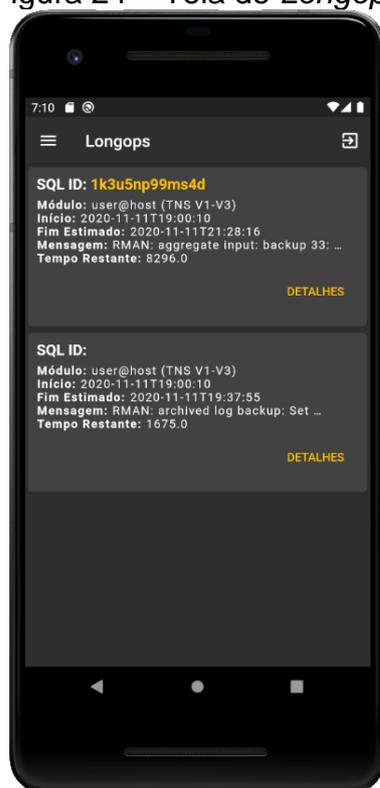
Figura 23 – Tela de *Tablespaces*.

Fonte: Elaborado pelo autor.

5.1.10 Tela de *Longops*

Esta tela, Figura 24, exibe as operações longas que estão ocorrendo no banco de dados selecionado no momento. Caso não tenha nenhum banco de dados selecionado, não serão exibidos nenhum dado. Ao clicar na opção “Detalhes”, o usuário é redirecionado para a Tela de Detalhes do SQL.

Figura 24 – Tela de Longops.

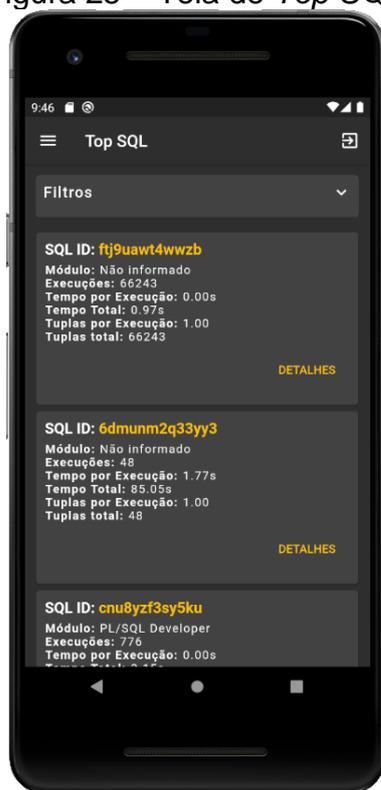


Fonte: Elaborado pelo autor.

5.1.11 Tela de Top SQL

Esta tela exibe as consultas SQL que satisfazem os filtros selecionados. A tela pode ser vista na Figura 25, enquanto os filtros podem ser vistos na Figura 26. Ao clicar na opção “Detalhes”, o usuário é redirecionado para a Tela de Detalhes do SQL.

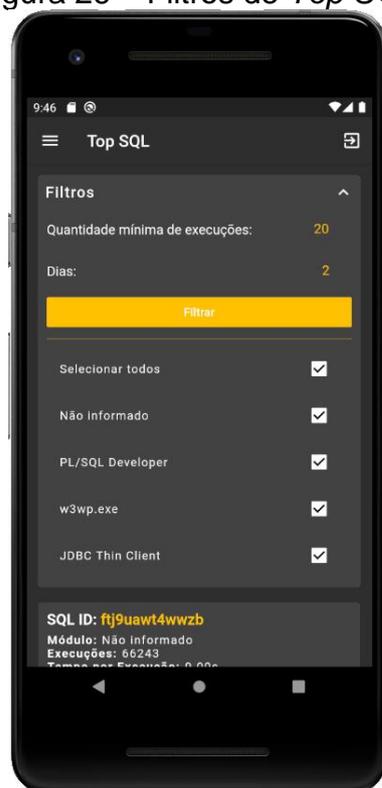
Figura 25 – Tela de Top SQL.



Fonte: Elaborado pelo autor.

Os filtros possíveis para esta tela são:

- Quantidade mínima de execuções do SQL, padrão 20;
- Quantidade de dias da amostragem, padrão 2;
- Módulo de execução do SQL. Os módulos são listados dinamicamente, de acordo com as opções disponíveis na lista de SQL.

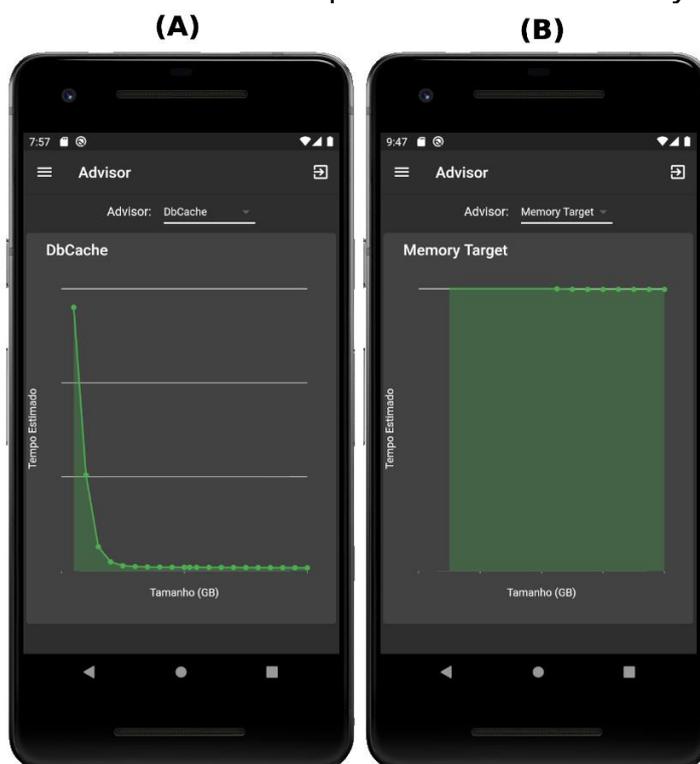
Figura 26 – Filtros de *Top SQL*.

Fonte: Elaborado pelo autor.

5.1.12 Tela de *Advisor*

Nesta tela são exibidos os dados do *Advisor*, selecionando as categorias disponíveis no menu de filtro.

A Figura 27 exhibe o *Advisor* para as categorias: (a) *DbCache* e (b) *Memory Target*.

Figura 27 – Tela de *Advisor* para *DbCache* e *Memory Target*.

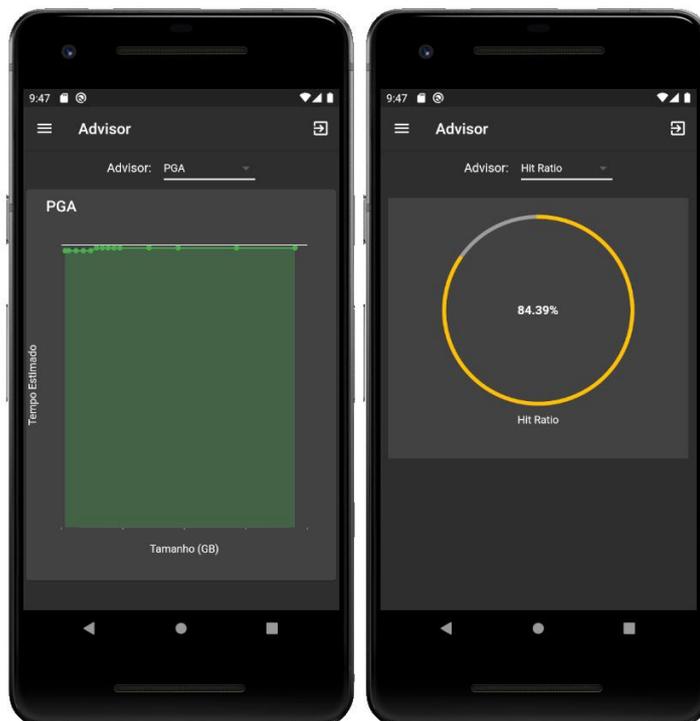
Fonte: Elaborado pelo autor.

A Figura 28 exibe o *Advisor* para as categorias: (a) *PGA* e (b) *Hit Ratio*.

Figura 28 – Tela de *Advisor* para *PGA* e *Hit Ratio*.

(A)

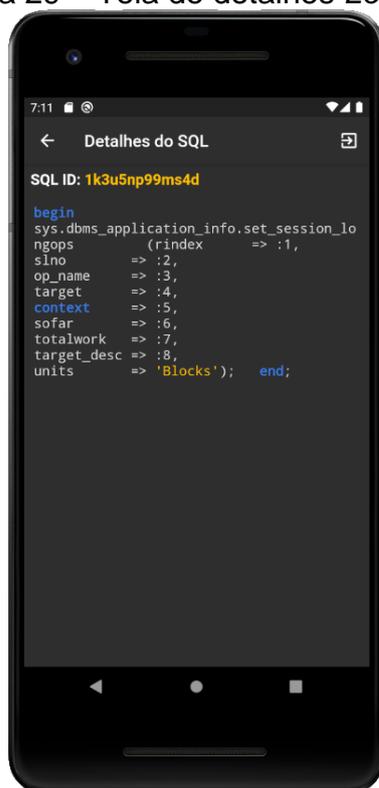
(B)



Fonte: Elaborado pelo autor.

5.1.13 Tela de Detalhes do SQL

Conforme mostrado na Figura 29, esta tela exibe o comando SQL. O identificador deste SQL é repassado pela tela antecessora, podendo ser a Tela de Longops, Figura 24, ou a Tela de Top SQL, Figura 25.

Figura 29 – Tela de detalhes *Longops*.

Fonte: Elaborado pelo autor.

5.2 Comparação da visualização dos dados

Um exemplo da melhoria na visualização dos dados pela utilização dos elementos visuais disponibilizados pelo Flutter, pode ser visto abaixo. A Figura 30 mostra como é a consulta dos dados de *tablespaces* realizada em um cliente Oracle.

Figura 30 – Consulta de *Tablespaces* no Oracle

TABLESPACE_NAME	SIZE_MB	FREE_MB	MAX_SIZE_MB	MAX_FREE_MB	FREE_PCT	USED_PCT	STATUS
CHANNEL	1110	63	32767	31720	96,8047120578631	3,19528794213691	ONLINE
FORUM	60	9	32767	32716	99,8443556016724	0,155644398327586	ONLINE
JIVE	50	9	32767	32726	99,8748741111484	0,125125888851588	ONLINE

Fonte: Elaborado pelo autor.

Na Figura 23 é exibida a mesma informação, porém utilizando o aplicativo DbMonitor. Podemos observar que a visualização dos dados no DbMonitor é mais intuitiva quando comparada a consulta realizada no cliente Oracle, onde estes dados são disponibilizados no formato de tabela.

6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento e a documentação do aplicativo DbMonitor apresentando a sua justificativa, descrição geral, tecnologias utilizadas para o desenvolvimento e seus resultados.

Para a elicitação dos requisitos, foi realizada e documentada uma entrevista com um DBA, e confeccionado o documento de *briefing*. Estes serviram de base para determinar as características do aplicativo e garantir que os requisitos levantados fossem persistentes para auxiliar no monitoramento de bancos de dados Oracle.

Com a utilização do *framework* Flutter, linguagem de programação Dart, banco de dados Firebase e a construção de uma API .NET, foi possível atender as necessidades levantadas para o aplicativo. O uso do Flutter possibilitou o desenvolvimento do aplicativo multiplataforma, utilizando apenas um código fonte escrito com a linguagem Dart, em tempo hábil e com sintaxe de fácil entendimento. Os serviços disponibilizados pelo Firebase possibilitaram o armazenamento dos dados de acesso aos bancos de dados Oracle e dos dados de usuários do DbMonitor, além da autenticação destes usuários. Com a construção da API .NET foi possível realizar o acesso seguro e eficaz a estes bancos de dados Oracle, utilizando a biblioteca *Oracle Managed Data Access*.

Os resultados obtidos foram satisfatórios, visto que o aplicativo possibilitou o monitoramento de bancos de dados Oracle a partir de cadastros simples dos usuários e dos dados de conexão com os bancos de dados. Além disto, possibilitou uma visualização mais intuitiva dos dados de desempenho quando comparada a forma de tabela que o Oracle disponibiliza estes dados.

6.1 Trabalhos futuros

- Possibilitar o cadastro de bancos de dados que utilizem outros SGBDs, como *MySQL* e *PostgreSQL*;
- Possibilitar a execução de *scripts SQL* dinâmicos criados pelo usuário, de maneira com que o resultado da execução seja comparado a valores pré-definidos no aplicativo para gerar alertas;
- Possibilitar a geração de relatórios de desempenho do banco de dados em um determinado período.

REFERÊNCIAS

BEKOSKI, Klemerson Chan. **DESENVOLVIMENTO DE UM APLICATIVO ANDROID PARA A CONSCIENTIZAÇÃO DO DESCARTE DE LIXO ELETRÔNICO**. 2016. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/7179/1/FB_COLIN_2016_1_03.pdf. Acesso em: 12 set. 2020.

DEVART. **Comparing Data Providers for Oracle**. 2020. Disponível em: <https://www.devart.com/dotconnect/oracle/articles/comparingproviders.html>. Acesso em: 18 set. 2020.

FIREBASE. **Firebase Authentication**. 2020. Disponível em: <https://firebase.google.com/docs/auth>. Acesso em: 12 set. 2020.

FIREBASE. **O Firebase ajuda as equipes de aplicativos para dispositivos móveis e da Web a alcançar o sucesso**. 2020a. Disponível em: <https://firebase.google.com/>. Acesso em: 14 set. 2020.

FLUTTER. **Technical overview**. 2020. Disponível em: <https://flutter.dev/docs/resources/technical-overview>. Acesso em: 14 set. 2020.

FLUTTER. **Segurança baseada no usuário**. 2020. Disponível em: <https://firebase.google.com/docs/database/security/user-security>. Acesso em: 14 set. 2020.

FLUTTER. **What is Flutter?** 2020. Disponível em: <https://flutter.dev/docs/resources/faq#what-is-flutter>. Acesso em: 11 set. 2020.

IMPACTA. **Quais as vantagens de usar Visual Studio ao criar apps?**. Disponível em: <https://www.impacta.com.br/blog/quais-vantagens-usar-visual-studio-desenvolver-aplicativos>. Acesso em: 15 set. 2020.

KANSAL, Aman. **Flutter in Focus: Stateful vs Stateless Widget**. 2020. Disponível em: <https://medium.com/flutter-for-you/flutter-in-focus-stateful-vs-stateless-widget-dd85570baf6b>. Acesso em: 09 set. 2020.

MATT. **Litecoin Script Algorithm Explained**. 2018. Disponível em: <https://www.mycryptopedia.com/litecoin-script-algorithm-explained>. Acesso em: 06 set. 2020.

MICROSOFT. **Overview of .NET Framework**. 2020. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/framework/get-started/overview>. Acesso em: 07 set. 2020.

ORACLE. **Getting Started with Database Administration**. 2015. Disponível em: https://docs.oracle.com/cd/E11882_01/server.112/e25494/dba.htm#ADMIN001. Acesso em: 10 set. 2020.

ORACLE. **Oracle Data Provider for .NET**. 2014. Disponível em: <https://docs.oracle.com/database/121/ODPNT/E17732-11.pdf>. Acesso em: 17 set. 2020.

PAIM, Leonardo. **Utilizando o VS Code para criar Aplicativos Flutter**. 2019. Disponível em: <https://medium.com/@leonardopaim/utilizando-o-vs-code-para-criar-aplicativos-flutter-windows-5074962ecfce>. Acesso em: 11 set. 2020.

SAMPAIO, Júlio. **Primeiros passos com a Google Dart**. 2015. Disponível em: <https://www.devmedia.com.br/primeiros-passos-com-a-google-dart/32954>. Acesso em: 12 set. 2020.

SARTORI, Guilherme; MOLINA, Leandro Ariel; LIMA, W. C. G. D. **Desenvolvimento De Um Sistema Microcontrolado De Baixo Custo Utilizando Smartphone Para Aplicações De Automação Residencial**. 2015. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/3884/1/CT_COELE_2014_2_09.pdf. Acesso em: 10 set. 2020.

APÊNDICE A – ENTREVISTA

Entrevistado: Augusto Custódio Mendes

Data: 10/09/2020

1 - Qual a sua profissão?

Administrador de Banco de dados Oracle.

2 - Qual a importância do monitoramento de banco de dados para você?

Evitar pausas por motivos simples, identificar processos não otimizados assim que iniciarem e garantir a segurança dos dados, réplicas e *backups*.

3 - Quais informações mais recorrentes você consulta em relação aos bancos de dados que é responsável?

Informações gerais das sessões, maiores *waits*, monitoramento de replicação e recuperação de desastres, degradação momentânea de performance etc.

4 - Atualmente você recebe notificações relacionadas aos bancos de dados que você é responsável?

Sim, apenas no portátil por Enterprise Manager ou *scripts* agendados e enviados por e-mail.

5 - Você tem acesso a informações dos bancos de dados que é responsável em seu *smartphone*?

Apenas alguns *scripts* criados para enviar mensagens por *Telegram*.

6 - Você emite relatórios periódicos relacionados aos bancos de dados que é responsável?

Sim, para monitorar a saúde geral da base de dados.

7 - Você utilizaria um aplicativo *mobile* que disponibilize informações relativas aos bancos de dados?

Sim, para agilizar a identificação de um problema e ter acesso rápido a alguma informação.

APÊNDICE B – BRIEFING

1 - Descrição do sistema.

É um aplicativo *mobile* que facilita no monitoramento de bancos de dados.

2 - Quais as funcionalidades do sistema?

Ele permite que:

- O DBA cadastre conexões com bancos de dados para serem monitorados;
- Seja gerado um gráfico com o uso de recursos do servidor do banco de dados;
- Seja emitida uma notificação com incidentes graves no banco de dados;
- Seja gerada uma lista de operações longas acontecendo no banco de dados;
- Seja gerada uma lista com o tamanho das *tablespaces* e porcentagem livre;
- Seja gerada uma lista do Advisor para DbCache, Memory Target, Program Global Area (PGA) e Hit Ratio;
- Seja gerada uma lista para os eventos de histograma;
- Seja gerada uma lista para espaço reutilizável em tabelas;
- Seja gerada uma lista de Top SQL do banco de dados.

3 - Como os usuários irão acessar o aplicativo?

O usuário deverá cadastrar uma conexão com o banco de dados Oracle que deseja monitorar e então poderá utilizar o aplicativo.

4 - Quem são os usuários?

Administradores de Bancos de Dados (DBA) e Analistas de Bancos de Dados.

5 - Quantas conexões poderão ser salvas?

Conexões ilimitadas.

6 - Quais informações sobre a conexão deverão ser fornecidas?

Deverá fornecer o *Host*, Usuário, Senha e Nome do Serviço.

ANEXO A – TERMO DE AUTORIZAÇÃO DE PUBLICAÇÃO DE PRODUÇÃO ACADÊMICA



GABINETE DO REITOR

Av. Universitária, 1069 ● Setor Universitário
Caixa Postal 86 ● CEP 74605-010
Goiânia ● Goiás ● Brasil
Fone: (62) 3946.1000
www.pucgoias.edu.br ● reitoria@pucgoias.edu.br

RESOLUÇÃO n° 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Celso Pereira do Vale Júnior
do Curso de Ciência da Computação, matrícula 2015.2.0028.0235-8,
telefone: (62) 98210-1655 e-mail celsoj_@hotmail.com, na qualidade de titular dos
direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do autor),
autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o
Trabalho de Conclusão de Curso intitulado
Desenvolvimento de um aplicativo para monitoramento de banco de dados
, gratuitamente, sem ressarcimento dos direitos autorais, por 5
(cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial
de computadores, no formato especificado (Texto (PDF); Imagem (GIF ou JPEG); Som
(WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da
área; para fins de leitura e/ou impressão pela internet, a título de divulgação da
produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 18 de fevereiro de 2021.

Assinatura do(s) autor(es):

Celso Pereira do Vale Júnior

Nome completo do autor: Celso Pereira do Vale Júnior

Assinatura do professor-orientador:

Ludmilla R. P. dos Santos

Nome completo do professor-orientador: Ludmilla Reis Pinheiro dos Santos