

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

JOÃO ANTONIO DE OLIVEIRA

**APLICATIVO ANDROID PARA O GERENCIAMENTO DE TAREFAS E
COMPROMISSOS PESSOAIS**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO
2020

JOÃO ANTONIO DE OLIVEIRA

**APLICATIVO ANDROID PARA O GERENCIAMENTO DE TAREFAS E
COMPROMISSOS PESSOAIS**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientadora: Prof.^a Andreia Scariot Beulke

PATO BRANCO
2020



Ministério da Educação
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
UTFPR - CAMPUS PATO BRANCO
DIRETORIA GERAL - CÂMPUS PATO BRANCO
DIR. DE GRAD. E EDUCAÇÃO PROFISSIONAL-PB
DEP. ACADÊMICO DE INFORMÁTICA DAINF-PB



TERMO DE APROVAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO

Aplicativo Android para o Gerenciamento de Tarefas e Compromissos Pessoais

POR

João Antonio de Oliveira

Este trabalho de conclusão de curso foi apresentado no dia 15 de julho de 2020, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Universidade Tecnológica Federal do Paraná. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Banca examinadora:

Profa. MSc Andreia Scariot Beulke
Professora orientadora

Prof. MSc Vinicius Pegorini
Professor convidado

Profa. Dra. Mariza Miola Dosciatti
Professora convidada

Prof. Dr. Edilson Pontarolo
Coordenador do Curso de Tecnologia em Análise e
Desenvolvimento de Sistemas

Profa. Dra. Mariza Miola Dosciatti
Responsável pela Atividade de Trabalho
de Conclusão de Curso



Documento assinado eletronicamente por **MARIZA MIOLA DOSCIATTI, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 16/07/2020, às 09:33, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **ANDREIA SCARIOT BEULKE, PROFESSOR(A) ORIENTADOR(A)**, em 16/07/2020, às 09:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **VINICIUS PEGORINI, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 16/07/2020, às 10:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **EDILSON PONTAROLO, COORDENADOR(A) DE CURSO/PROGRAMA**, em 16/07/2020, às 12:11, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.utfpr.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1525624** e o código CRC **055C6F21**.

RESUMO

A vida agitada e a falta de tempo são duas condições bastante presentes na vida das pessoas. A quantidade de compromissos e atividades a serem realizadas diariamente, seja de cunho pessoal ou profissional, são muitas e, equilibrar a vida pessoal e profissional pode ser uma tarefa difícil se não forem gerenciadas adequadamente. É preciso criar agenda, estabelecer prioridades e prazos, fazer planejamento e ter disciplina para atingir os objetivos estabelecidos e realizar as atividades que devem ser cumpridas. Isso pode causar dificuldade na organização das tarefas pessoais a serem realizadas. Visando auxiliar o gerenciamento do tempo que as pessoas têm para realizar suas tarefas, como resultado deste trabalho foi desenvolvido um aplicativo Android, hospedado localmente no dispositivo. Esse aplicativo permite fazer o cadastro das tarefas das pessoas informando a prioridade que pode ser emergencial, circunstancial e urgente. Além disso, também é possível categorizá-las em grupos. O aplicativo ainda possibilita ao usuário informar o local onde a tarefa será realizada por meio de um mapa. A notificação é padronizada para exibir a mensagem dez minutos antes de cada tarefa, contudo, o usuário poderá alterar o tempo para ser notificado.

Palavras-chave: Aplicação Android. Gerenciamento de tarefas pessoais. Agenda.

ABSTRACT

The hectic life and lack of time are two conditions very present in people's lives. The amount of appointments and activities to be carried out daily, whether personal or professional, are many and, balancing personal and professional life can be a difficult task if not managed properly. It is necessary to create a schedule, to establish priorities and deadlines, make planning and have discipline to achieve the established objectives and carry out the activities that must be fulfilled. This can cause difficulty in organizing the personal tasks to be done. In order to help manage the time that people have to perform their personal tasks, as a result of this work an Android application was developed, hosted locally on the device. This application allows you to register people's tasks, informing the priority that can be emergency, circumstantial and urgent. In addition, it is also possible to categorize them into groups. The application also allows the user to inform the location where the task will be performed using a map. The notification is standardized to display the message ten minutes before each task, however, the user can change the time to be notified.

Keywords: Android application. Management of personal tasks. Schedule.

LISTA DE FIGURAS

Figura 1 – Casos de uso	21
Figura 2 – Diagrama de Entidade e Relacionamento das tabelas do SQLite	25
Figura 3 – Entidade Compartilhamento desenvolvida no Firebase.....	25
Figura 4 – Botão de configuração.....	26
Figura 5 – Listagem da configuração de notificação.....	27
Figura 6 – Tela de edição de notificação	27
Figura 7 – Menu de navegação primário	28
Figura 8 – Listagem Tarefas	29
Figura 9 – Filtro por Status.....	29
Figura 10 – Tela de Tarefas	30
Figura 11 – Mapa de local da tarefa	30
Figura 12 - Notificação.....	31
Figura 13 – Tela de Login	32
Figura 14 – Listagem de Tarefas Compartilhadas.....	32
Figura 15 - Tela de Tarefa Compartilhada	33
Figura 16 – Backup e Restauração do banco de dados.....	33
Figura 17 – Tela de Relatórios	34
Figura 18 – Relatório de horas gastas.....	34
Figura 19 - Estrutura do código	35
Figura 20 – Estrutura das interfaces do projeto	36

LISTA DE QUADROS

Quadro 1 – Lista de ferramentas e tecnologias.....	17
Quadro 2 – Requisitos funcionais.....	20
Quadro 3 – Requisitos não funcionais.....	20
Quadro 4 – Operação de incluir registros no banco de dados	22
Quadro 5 – Operação de alterar registros no banco de dados	22
Quadro 6 – Operação de excluir registros no banco de dados.....	23
Quadro 7 – Operação de consultar registros no banco de dados	24

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
CETIC	Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação
ERP	<i>Enterprise Resource Planning</i>
IDE	<i>Integrated Development Environment</i>
JIT	<i>Just In Time</i>
MRT	<i>Materials Requirements Planning</i>
OBM	<i>Open-Book Management</i>
OPT	<i>Optimized Production Technology</i>
SQL	<i>Structured Query Language</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO	9
1.1 CONSIDERAÇÕES INICIAIS	9
1.2 OBJETIVOS	10
1.2.1 Objetivo Geral	10
1.2.2 Objetivos Específicos	10
1.3 JUSTIFICATIVA	11
1.4 ESTRUTURA DO TRABALHO	12
2 REFERENCIAL TEÓRICO	13
2.1 GESTÃO DO TEMPO	13
2.2 CLASSIFICAÇÃO DAS ATIVIDADES.....	14
2.3 USO DE TECNOLOGIAS NA GESTÃO DO TEMPO	15
3 MATERIAIS E MÉTODO	17
3.1 MATERIAIS	17
3.2 MÉTODO	17
4 RESULTADOS	19
4.1 ESCOPO DO SISTEMA	19
4.2 MODELAGEM DO SISTEMA	19
4.3 APRESENTAÇÃO DO SISTEMA.....	26
4 IMPLEMENTAÇÃO DO SISTEMA.....	35
5 CONCLUSÃO.....	49
REFERÊNCIAS	50

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais, os objetivos e a justificativa da realização deste trabalho. No final do capítulo é apresentada a estrutura do texto com a organização dos seus capítulos.

1.1 CONSIDERAÇÕES INICIAIS

A vida agitada e a falta de tempo são duas condições bastante presentes na vida das pessoas. A quantidade de atividades a serem realizadas, seja de cunho pessoal ou profissional, são muitas e equilibrar a vida pessoal e profissional pode ser uma tarefa difícil. É preciso criar agenda, estabelecer prioridades e prazos, fazer planejamento e ter disciplina para atingir os objetivos estabelecidos e realizar as atividades que devem ser cumpridas. Com a massificação do uso de aparelhos de celulares, existem aplicativos para atender às diversas necessidades do ser humano. Para isso, aplicativos de uso computacional ou de dispositivos móveis, pode auxiliar os usuários na organização para o devido cumprimento das suas atividades diárias.

Os usuários de celulares utilizam o aparelho para realizar diversos tipos de atividades, tais como: receber e enviar chamadas, tirar e compartilhar fotos, acessar redes sociais, enviar e receber mensagens, ouvir músicas, assistir vídeos, jogar, usar mapas, enviar e receber *e-mails*, acessar páginas *web*, buscar informações, entre outras (CETIC.BR, 2017a). Além disso, existem diversos aplicativos para dispositivos móveis com o objetivo de atender necessidades específicas dos usuários, como realizar compras e acessar contas bancárias. Sem contar o amplo uso que os *smartphones* vêm tendo na realização de tarefas profissionais.

Uma pesquisa realizada pelo TIC Domicílios e divulgada pelo Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (CETIC.BR) aponta que 49% da população utiliza apenas o celular para se comunicar e acessar conteúdo *on-line*, dispensando o uso de computadores (CETIC.BR, 2017b). A pesquisa mostra, ainda, que o número de pessoas que usam somente o computador para acessar a Internet está caindo, pois entre 2014 a 2017 o percentual variou de 24% a 4%, respectivamente. Isso mostra que o uso dos aparelhos celulares está se tornando cada vez mais comum entre as pessoas independentemente de faixa etária, grau de instrução, gênero, renda familiar e classe social.

Os indicadores de cada categoria apontados pelo CETIC.br (2017b) mostram que o percentual de pessoas com idade entre 10 e 60 anos que utilizam o celular é de 66% e 64%, respectivamente. O maior número de usuários por faixa etária é entre 16 a 59 anos (91% e

86%, respectivamente), sendo o maior número de usuários situados na faixa etária de 25 a 34 anos (94%). Com relação à renda familiar, 72% das pessoas com até um salário mínimo e 60% dos sem renda possuem aparelho celular. O maior número de usuários categorizados por renda familiar são os que recebem mais de dez salários mínimos. A pesquisa mostra que pessoas analfabetas ou que possuem apenas a educação infantil utilizam menos o celular (51%) e pessoas com ensino superior (97%) são as que mais utilizam o aparelho.

Diante do exposto, este trabalho propõe o desenvolvimento de um aplicativo para dispositivos móveis para gerenciar as tarefas e os compromissos dos usuários que podem ser de cunho pessoal, profissional ou acadêmico, por exemplo, sendo que cada tarefa pode ser categorizada em um grupo específico. A plataforma móvel foi escolhida por permitir uso independentemente da localização do usuário, com ou sem acesso à Internet, pois as informações de suas tarefas ou compromissos são salvas localmente. Contudo, o aplicativo possibilita a realização de *backup* na nuvem.

1.2 OBJETIVOS

O objetivo geral está relacionado com o resultado principal da realização deste trabalho e os objetivos específicos complementam o geral em termos de funcionalidades do sistema.

1.2.1 Objetivo Geral

Desenvolver um aplicativo para dispositivos móveis para gerenciar as tarefas e os compromissos dos usuários.

1.2.2 Objetivos Específicos

- Organizar a rotina diária das pessoas possibilitando o acesso rápido às tarefas e aos compromissos previamente cadastrados.
- Proporcionar o gerenciamento das tarefas com estimativa de tempo para a sua realização.
- Possibilitar o envio de alertas por meio de *pop-up* para cada tarefa e compromisso cadastrado.

- Compartilhar tarefas e compromissos entre os usuários.

1.3 JUSTIFICATIVA

A quantidade de tarefas e compromissos diários pode fazer com que as pessoas desperdicem tempo em atividades que, muitas vezes, são menos importantes que outras ou que deixem de realizar tarefas, inclusive importantes. Administrar os compromissos e as tarefas pessoais e profissionais é importante para auxiliar que todas as tarefas necessárias sejam realizadas e possa haver um controle de tempo e priorização para realização, se for o caso. E isso é possível quando se tem uma agenda planejada e organizada das atividades diárias.

Existem estudos que ajudam a definir as atividades em grupos para organizar o tempo e a prioridade de execução, aumentar a produtividade e o equilíbrio. Um desses estudos é realizado por Christian Barbosa (2012) que propõe um conceito para organizar as atividades na tríade do tempo, composta por esferas classificadas como importante, urgente e circunstancial. A esfera da importância refere-se às atividades que são significativas para a pessoa, a esfera da urgência reúne atividades que possuem tempo curto para sua execução ou período esgotado e a esfera das circunstâncias cobre as tarefas desnecessárias (BARBOSA, 2012).

Diante da necessidade de gerenciar as atividades e os compromissos e proporcionar melhoria na gestão do tempo, este trabalho propõe o desenvolvimento de um aplicativo *mobile* que permite que o usuário possa organizar a sua rotina diária.

O sistema permitirá um gerenciamento mais completo das atividades e compromissos, a fim de trazer uma melhoria no tempo necessário para a execução de cada tarefa. O usuário poderá verificar quanto tempo ele tem disponível para completar as tarefas ou quanto tempo tem de tarefas agendadas auxiliando na tomada de decisão para agendar nova tarefa.

Para o sistema proposto foi escolhido o desenvolvimento de um aplicativo *mobile* na plataforma Android, pois de acordo com Zuriarrain (2017) a plataforma Android ultrapassou o Windows e é o sistema operacional mais utilizado no mundo e, também, pela facilidade na utilização e pela mobilidade.

1.4 ESTRUTURA DO TRABALHO

Este texto está organizado em capítulos dos quais este é o primeiro e apresenta as considerações iniciais, objetivos e a justificativa do trabalho. O Capítulo 2 apresenta o referencial teórico. No Capítulo 3 são apresentados os materiais e o método utilizados para o desenvolvimento deste trabalho. No Capítulo 4 está o resultado da realização do trabalho, que é a modelagem e a implementação do aplicativo. Por fim, está a conclusão e as referências do trabalho.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta a fundamentação teórica deste trabalho que está dividida em três seções: a primeira explana sobre a gestão do tempo, a segunda apresenta metodologias de classificação das atividades e a terceira aborda o uso das tecnologias na gestão do tempo e das atividades.

2.1 GESTÃO DO TEMPO

Atualmente, as pessoas vivem em um ritmo cada vez mais acelerado, devido às diversas atividades que devem desenvolver. Essas atividades são relacionadas a tarefas domésticas, profissionais, acadêmicas, entre outras. As pessoas dividem seu tempo em funções para realizar todas as atividades programadas e, muitas vezes, acabam deixando de fazer alguma tarefa por falta de tempo.

De acordo com Lima e Jesus (2011) a administração do tempo é um assunto que vêm ganhando espaço e está sendo discutido no mundo corporativo, principalmente com o objetivo de aumentar a produtividade do indivíduo.

As pessoas realizam atividades que surgem repentinamente e que não contribuem para o sucesso profissional do indivíduo ou da empresa (IPOG, 2019). Um dos maiores desafios na gestão do tempo é distinguir as atividades que são urgentes das importantes e que isso deve-se ao fato do ritmo acelerado em que vivem (IPOG, 2019). A gestão do tempo ocorre quando há controle de tempo que o indivíduo gasta para realizar atividades específicas, tanto na vida pessoal quanto na profissional e, assim, aumentar a produtividade e a eficiência (MELO; FARIAS, 2015). Para isso, esses autores argumentam que a boa auto-gestão, disciplina, definir prioridades, executar uma tarefa de cada vez e, antes de começar uma tarefa, deve-se concluir outra que tenha sido iniciada, são características importantes para administrar o tempo.

Para Melo e Farias (2015) a procrastinação, uso indevido do tempo para realizar tarefas que não são importantes ou necessárias e a falta de planejamento são fatores que ocasionam problemas de gestão do tempo. Para esses autores, as pessoas estão ficando mais conscientes de que o gerenciamento eficaz do tempo é importante na realização de suas tarefas diárias.

2.2 CLASSIFICAÇÃO DAS ATIVIDADES

Além do planejamento adequado do tempo é necessário definir qual tarefa deve ser executada primeiro e, para isso, é importante diferenciar três elementos básicos que são: a urgência, a importância e a prioridade (MELO; FARIAS, 2015).

Covey (2005) classifica as atividades diárias em uma matriz do tempo, como: importantes e urgentes, importantes e não urgentes, não importantes e urgentes e, não importante e não urgentes. Conforme argumenta Covey (2005), basicamente uma tarefa é dividida em dois fatores (urgente e importante) sendo que urgente significa que uma tarefa deve ser executada de forma imediata e a importância tem a ver com resultados, pois exigem iniciativa e proatividade.

Barbosa (2012) critica a classificação das atividades simultaneamente em importantes e urgentes e desenvolveu um conceito chamado Tríade do Tempo, no qual o tempo é dividido em três esferas que são: urgentes, importantes e circunstancial. Para esse autor a principal diferença desse modelo é que uma atividade nunca é considerada importante e urgente ao mesmo tempo e que as três esferas juntas compõem a maneira de como o tempo é utilizado.

A esfera da importância é composta pelas atividades significativas que são realizadas a curto, médio ou longo prazo e que o que é importante tem tempo para ser realizado. Tudo o que é urgente tem tempo curto ou esgotado para ser realizado, geralmente envolvem atividades que não são previstas. Por outro lado, há as tarefas consideradas circunstanciais, ou seja, que não são necessárias e que não trazem resultados.

Barbosa (2012) corrobora com Covey (2005) ao afirmar que as tarefas consideradas importantes produzem resultados mais efetivos. No entanto, há discordância ao afirmar que toda tarefa importante pode ser ao mesmo tempo urgente, pois, para Barbosa (2012), as tarefas importantes têm prazo para serem cumpridas, caso contrário, são urgentes. Ainda, o autor diferencia as tarefas importantes das demais, quando as tarefas pessoais têm importância para a pessoa e nem sempre para os outros, proporcionam prazer ao serem executadas e, geralmente, são espontâneas. Por sua vez, tarefas urgentes devem ser executadas de forma imediata e causam algum tipo de problema caso não forem executadas. Uma tarefa que era importante e tornou-se urgente é quando havia um prazo de execução que não foi cumprido e a tarefa passa a ser realizada quando o indivíduo é cobrado ou pressionado. Dessa forma, quando existem muitas tarefas urgentes para serem realizadas, atrasa a realização das importantes e isso diminui a produtividade.

As circunstanciais ocorrem quando o indivíduo é levado a fazê-las independentemente de sua vontade, ou da situação em que se encontra. Barbosa (2012) identifica essas tarefas quando podem ser urgentes ou importantes para outras pessoas, mas nunca para o indivíduo, não geram resultados, geram situação de insatisfação ou quando são realizadas contra a vontade.

De acordo com a classificação do tempo proposta pela matriz de Covey (2005) e da tríade do tempo de Barbosa (2012), é interessante que o indivíduo classifique as atividades que precisa realizar adotando um critério. Para Barbosa (2012), cada tarefa a ser realizada deve ser analisada e classificada em uma esfera, pois isso ajuda a administrar o tempo e mudar os padrões. De acordo a pesquisa de Barbosa (2012) realizada com pessoas que classificaram as atividades na tríade do tempo, 69% conseguiram aumentar a quantidade de tarefas realizadas no dia e ainda colocaram atividades pessoais em seus planos. Assim, quando as pessoas fazem um planejamento de suas atividades de forma proativa buscam reduzir o urgente e realizam mais tarefas importantes.

2.3 USO DE TECNOLOGIAS NA GESTÃO DO TEMPO

As tecnologias são ferramentas que auxiliam os indivíduos na realização de suas tarefas, auxiliando no aumento da produtividade, seja na vida pessoal ou profissional. Quase todas as pessoas têm um *e-mail*, um celular ou um computador para auxiliar em suas rotinas diárias e na realização de suas atividades.

Existem muitas ferramentas que auxiliam os usuários à administrarem o tempo para a realização de suas tarefas. Na gestão empresarial há ferramentas como *Just In Time* (JIT), Kanban, *Enterprise Resource Planning* (ERP), *Materials Requirements Planning* (MRT), *Optimized Production Technology* (OPT), Trello, *Open-Book Management* (OBM), entre outras, que podem auxiliar a gestão empresarial no desenvolvimento de suas atividades com maior produtividade e eficácia. Há também aplicativos no formato de agenda que podem ser utilizados para administrar as atividades a serem realizadas.

Essas ferramentas variam em termos de funcionalidades e aplicabilidade. Algumas são mais direcionadas para empresas, outras podem ser utilizadas tanto para planejamentos pessoais quanto profissionais. Essas ferramentas permitem delegar tarefas, prazos, interagir em equipes, compartilhar tarefas, organizar projetos, por exemplo.

Para Barbosa (2012) o uso de uma ferramenta capaz de abordar a metodologia da administração do tempo é importante e necessário para produzir os resultados esperados,

como aumento de produtividade, por exemplo. O autor afirma que a metodologia ensina o modelo mental baseado em um modelo de administração do tempo, a tecnologia dá sustentação ao modelo empregado e a persistência para treinar o modelo.

Portanto, combinar as funcionalidades básicas de administração do tempo com uma metodologia que aborda o modelo mental (classificando as tarefas) e os elementos de execução de cada tarefa (tempo, dia, horário, pessoas, etc.) pode ser uma alternativa interessante para ter resultados positivos e melhor gerenciados.

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e o método utilizado para o desenvolvimento deste trabalho. Os materiais elencam as ferramentas e tecnologias utilizadas na modelagem e na programação do aplicativo. O método está relacionado a um modelo de processo que visa ordenar e estruturar o desenvolvimento do aplicativo.

3.1 MATERIAIS

O Quadro 1 representado a seguir, apresenta a lista das ferramentas e tecnologias que serão utilizadas para o desenvolvimento do sistema proposto neste trabalho.

Quadro 1 – Lista de ferramentas e tecnologias

Ferramenta / Tecnologia	Versão	Finalidade
Android SDK	26	Linguagem de Programação
Android Studio	3.3.1	<i>Integrated Development Environment (IDE)</i>
Astah Community	7.1.0	Modelagem do Sistema (casos de uso)
Balsamiq	3.5.17	Prototipagem de telas
Firebase	17.0.0	Ferramenta para gerenciamento de dados e autenticação
Java	11	Linguagem de Programação
MySQL Workbench	8.0	Diagrama de entidade e relacionamento
Room Persistence	1.1.1	Biblioteca de persistência de dados
SQLite	3.30.1	Servidor de banco de dados

Fonte: Autoria própria.

3.2 MÉTODO

A modelagem abordada para o desenvolvimento do aplicativo de gerenciamento de tarefas e compromissos gerais, foi dividida em várias etapas conforme o modelo sequencial linear descrito por Pressman e Maxim (2016). No modelo, as tarefas ocorrem sequencialmente, com suas fases bem definidas, sendo elas: comunicação, modelagem e implementação.

Na fase de comunicação foi realizado o levantamento de requisitos, que ocorreu por meio de pesquisas em softwares disponíveis no mercado com propósitos semelhantes ao do aplicativo a ser desenvolvido.

Levando em conta principalmente o *feedback* negativo e positivo dos usuários, para saber os principais pontos a serem atendidos no momento do desenvolvimento e, também, adicionar novas funcionalidades para serem agregadas no aplicativo.

Na etapa de modelagem foi realizada uma análise completa do projeto com a criação de diagramas de caso de uso e os das principais atividades propostas pelo aplicativo. E a partir dos resultados dessas atividades foram geradas as informações para ser realizada a modelagem do banco de dados por meio de entidades e relacionamentos.

Nesta etapa, o ponto relevante de consideração foi a facilitação máxima da manipulação e interação do usuário com seus compromissos. Outro ponto que ganhou bastante importância foi a sincronização de seus dados em nuvem, para manter a agenda sempre salva e atualizada.

A etapa da implementação do aplicativo consistiu na codificação de acordo com a especificação construída nas etapas anteriores. Os testes foram realizados informalmente, cada funcionalidade implementada era testada e, no final, foi realizado um teste geral para garantir o correto funcionamento do aplicativo desenvolvido.

4 RESULTADOS

Este capítulo apresenta o resultado deste trabalho que é o desenvolvimento de um aplicativo para dispositivos móveis que permite gerenciar as tarefas e os compromissos diários das pessoas.

4.1 ESCOPO DO SISTEMA

O sistema proposto neste trabalho visa proporcionar aos usuários um maior controle das suas atividades diárias.

O sistema permite que o usuário cadastre as tarefas, os grupos e a localização da tarefa. Ao cadastrar uma tarefa, o usuário poderá configurar a data de início e de término, o tempo estimado para a realização da tarefa, e o *status*. A notificação é recebida por *pop-up* no tempo definido pelo usuário

Ainda, poderá ser informado a periodicidade das tarefas, se são realizadas diariamente, semanalmente, quinzenalmente ou mensalmente. Ao iniciar uma tarefa ou compromisso será definido seu *status*, podendo estar em aberto, iniciada, em andamento, suspensão ou concluída. O sistema, ainda, permite que o usuário possa gerar relatórios em forma de gráfico das suas atividades, exibindo dados de acordo com o grupo das tarefas e o tempo gasto em cada grupo. Isso pode auxiliar o usuário no seu planejamento diário para decidir sobre agendar mais atividades ou compromissos em uma determinada data.

Quando uma tarefa é registrada, deverá ser vinculada a um grupo para organizá-las em categorias e auxiliar na visualização dos dados. No registro da atividade ou do compromisso, poderá ser inserida a localização por meio de um mapa.

O usuário poderá fazer o *backup* de suas atividades por meio de uma conta vinculada ao Gmail, assim seus dados serão armazenados no *Firebase Storage*.

4.2 MODELAGEM DO SISTEMA

O Quadro 2 apresenta a lista dos requisitos funcionais (RF) e o Quadro 3 a lista dos requisitos não-funcionais (RNF) do aplicativo.

Quadro 2 – Requisitos funcionais

Identificação	Nome	Descrição
RF 1	Manter tarefa ou compromisso	As tarefas e os compromissos são as atividades diárias do usuário. Ao registrar uma tarefa ou compromisso, deve ser informada a data e a hora de início e término, o tempo de realização e o grupo ao qual a tarefa pertence. Também deve ser personalizado o tempo desejado para notificação antes da tarefa ser iniciada. Quando a tarefa for registrada poderá ser informada a periodicidade desta tarefa, sendo ela diariamente, quinzenalmente ou mensalmente.
RF 2	Alterar <i>status</i>	O <i>status</i> de cada tarefa pode ser alterado conforme o andamento da tarefa (iniciada, cancelada, suspensa concluída).
RF 3	Definir prioridade	Ao registrar uma tarefa, o usuário deve definir a sua prioridade (circunstancial, urgente, importante).
RF 4	Manter grupo	Cada tarefa estará vinculada a um grupo de atividades (esporte, lazer, acadêmica, profissional, entre outros).
RF 5	Manter localização	Para cada tarefa é informada a localização de sua realização por meio de mapa.
RF 6	Compartilhar tarefa	Uma tarefa pode ser compartilhada com outras pessoas que já fizeram o uso do aplicativo, pelo seu <i>e-mail</i> .
RF 7	Personalizar notificação	Para cada tarefa registrada será possível personalizar o tempo antes do início da tarefa que o usuário deseja ser notificado por meio de um <i>pop-up</i> .
RF 8	Emitir relatórios	Será possível emitir relatórios em forma de gráfico, como, por exemplo, o tempo gasto em horas nas tarefas separadas por grupo.
RF 9	Filtrar tarefa por <i>status</i>	O aplicativo irá mostrar a lista de tarefa filtrada pelo seu <i>status</i> .
RF 10	Fazer e restaurar <i>backup</i>	Backup no <i>Firebase Storage</i> .

Fonte: Autoria própria.

O Quadro 3 exibe os requisitos não-funcionais identificados para o aplicativo.

Quadro 3 – Requisitos não funcionais

Identificação	Nome	Descrição
RNF01	Campos de preenchimento obrigatórios	Os campos que são de preenchimento obrigatório serão validados por meio de função do sistema.
RNF02	Campos com máscaras de entrada	Os campos que possuem caracteres especiais serão validados por meio de máscaras de entrada.
RNF03	Vínculo	Dados relacionados devem estar vinculados (um grupo de tarefas não pode ser removido se existem tarefas associadas a ele).

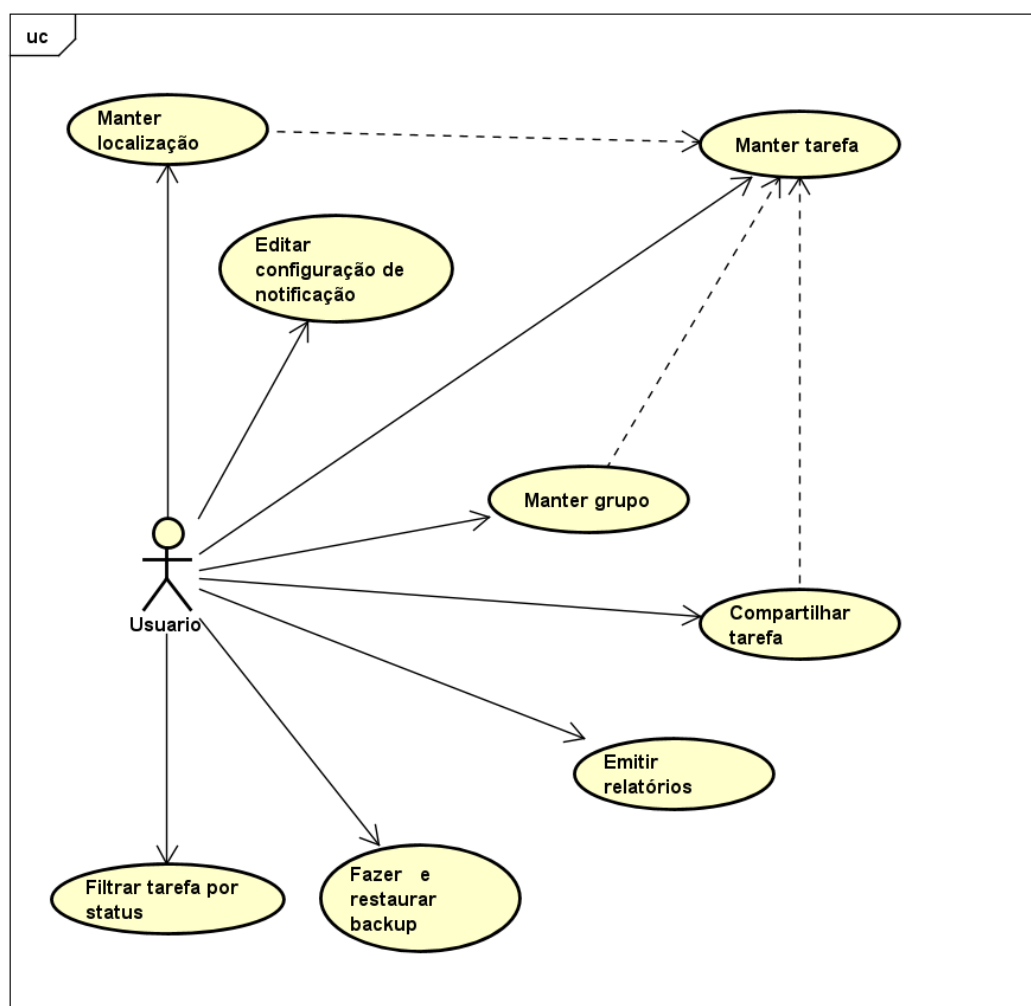
RNF04	Relatórios	A data inicial deve ser maior que a data final.
-------	------------	---

Fonte: Autoria própria.

O aplicativo tem apenas um usuário que é a pessoa que o utilizará para compor a sua agenda pessoal, elencando e categorizando as suas atividades (prioridades).

As funcionalidades que o usuário poderá desempenhar com o uso do aplicativo são mostradas na Figura 1 que representa o diagrama de casos de uso do aplicativo.

Figura 1 – Casos de uso



Fonte: Autoria própria.

Os quadros numerados de 4 a 7 apresentam a expansão dos casos de uso da Figura 1. A descrição dos casos de uso identificados como “Manter” incluem as operações de inserir, consultar, alterar e excluir dados de um registro do sistema. Essas operações apresentam, basicamente, o mesmo comportamento para todos os casos de uso que incluem essas operações.

O Quadro 4 apresenta a expansão do caso de uso para incluir um registro no banco de dados.

Quadro 4 – Operação de incluir registros no banco de dados

<p>Caso de uso: Inserir (refere-se à operação de inclusão nos casos de uso identificados como “manter”).</p> <p>Descrição: Inclusão dos dados cadastrais no sistema.</p> <p>Atores: Usuário.</p> <p>Pré-condição: Não há.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela para realizar os cadastros necessários. 2. Ator preenche os dados e clica no botão Salvar. O sistema insere as informações no banco de dados e mostra uma mensagem de confirmação da operação. <p>Pós-Condição: Registro inserido no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
1. Campos obrigatórios não preenchidos.	<p>1.1. O ator não preenche campos obrigatórios e clica no botão Salvar.</p> <p>1.2. O sistema valida as informações e exibe uma mensagem informando que campos obrigatórios não foram preenchidos.</p> <p>1.3. O sistema permanece na tela de inclusão, com os campos que já haviam sido preenchidos e destacando os campos sem preenchimento.</p>
2. Campos preenchidos com formato inválido.	<p>2.1. O ator preenche os campos de forma incorreta e clica no botão Salvar.</p> <p>2.2. O sistema valida as informações e exibe uma mensagem informando que os dados foram preenchidos de forma incorreta.</p> <p>2.3. O sistema permanece na tela de inclusão com os dados que já haviam sido informados e destacando os campos com dados no formato inválido.</p>

Fonte: Autoria própria.

O Quadro 5 apresenta a expansão do caso de uso para alterar um registro do banco de dados.

Quadro 5 – Operação de alterar registros no banco de dados

<p>Caso de uso: Alterar (refere-se à operação de alteração nos casos de uso identificados como “manter”).</p> <p>Descrição:</p>

<p>Alteração dos dados cadastrais no sistema.</p> <p>Atores: Usuário.</p> <p>Pré-condição: Registro estar cadastrado no sistema.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. O ator acessa a tela para visualização de dados do registro. 2. O sistema apresenta o registro selecionado para alteração. 3. Ator altera os dados do registro e clica no botão Salvar. 4. O sistema valida as informações e as salva no mesmo registro da tabela do banco de dados. <p>Pós-Condição: Registro alterado no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
1. Campos obrigatórios não preenchidos.	<p>1.1. O ator não informa dados obrigatórios e clica no botão Salvar.</p> <p>1.2. O sistema valida que não foram informados dados obrigatórios e exibe uma mensagem para o usuário sem salvar os dados no banco de dados.</p> <p>1.3. O sistema permanece na tela de alteração mantendo os dados informados anteriormente.</p>
2. Campos preenchidos com formato inválido.	<p>2.1. O ator altera os dados com um valor em formato inválido e clica no botão Salvar.</p> <p>2.2. O sistema valida informando que os dados não estão no formato esperado e exibe uma mensagem ao usuário sem salvar o registro no banco de dados.</p> <p>2.3. O sistema permanece na tela de inclusão mantendo as alterações realizadas.</p>

Fonte: Autoria própria.

O Quadro 6 apresenta a expansão do caso de uso para excluir um registro do banco de dados.

Quadro 6 – Operação de excluir registros no banco de dados

<p>Caso de uso: Excluir (refere-se à operação de exclusão nos casos de uso identificados como “manter”).</p> <p>Descrição: Exclusão de registros no sistema.</p> <p>Atores: Usuário.</p> <p>Pré-condição: Registro estar incluso no sistema.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela para a exclusão do registro. 2. Ator seleciona o registro para excluí-lo e clica no botão Excluir. 3. O sistema exclui o registro do banco de dados e exibe a lista de cadastros

atualizada. Pós-Condição: Registro excluído do banco de dados.	
Nome do fluxo alternativo (extensão)	Descrição
1. Exclusão de registro com vínculos no sistema.	1.1. Ator seleciona um registro que possui vínculos no sistema e clica no botão Excluir. 1.2 O sistema verifica que o registro possui vínculos, não exclui e exibe mensagem de alerta ao usuário.

Fonte: Autoria própria.

O Quadro 7 apresenta a expansão do caso de uso para consultar informações do banco de dados.

Quadro 7 – Operação de consultar registros no banco de dados

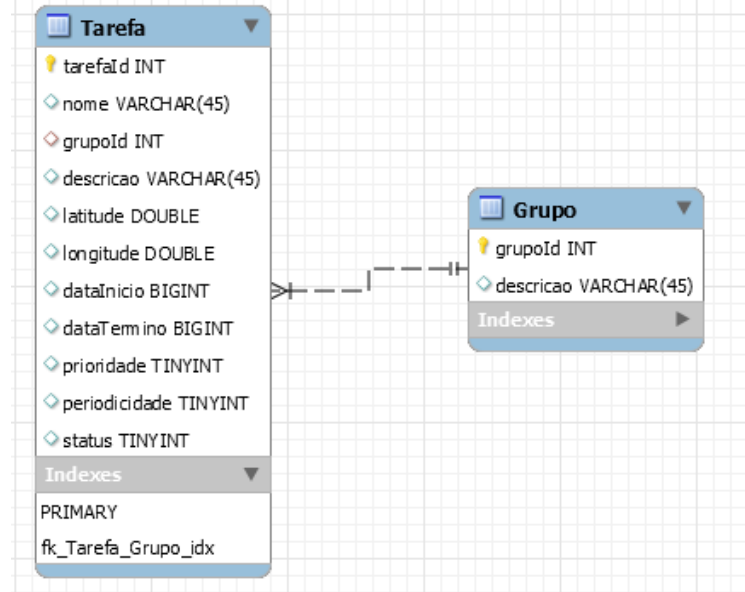
<p>Caso de uso: Consultar (refere-se à operação de consulta nos casos de uso identificados como “manter”).</p> <p>Descrição: Consultar informações cadastradas no sistema.</p> <p>Atores: Usuário.</p> <p>Pré-condição: Registro estar incluso no sistema.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. Ator acessa a tela para visualização de dados já cadastrados. 2. Ator indica quais dados ele pretende consultar por meio de filtros. 3. O sistema exibe os dados da consulta ao usuário. <p>Pós-Condição: Dados exibidos para o usuário.</p>

Fonte: Autoria própria.

A Figura 2 apresenta o diagrama de entidade e relacionamento do sistema. A entidade denominada Tarefa é a mais importante, pois é nela que ficarão armazenadas as informações da tarefa. Essas informações correspondem aos dados necessários para o registro da tarefa.

A Figura 3 representa a tabela de Compartilhamento desenvolvida no Firebase.

Figura 2 – Diagrama de Entidade e Relacionamento das tabelas do SQLite



Fonte: Autoria própria.

Figura 3 – Entidade Compartilhamento desenvolvida no Firebase



Fonte: Autoria própria.

Cada tarefa deve pertencer a um grupo (pessoal, acadêmica, profissional, entre outras), possuir uma localização (onde serão realizadas), ter um horário de início e de término (para estimar o tempo de realização) e uma data de início e de término.

Além disso, cada tarefa deve estar associada a uma prioridade (urgente, importante ou circunstancial), para categorizar as atividades e auxiliar o usuário na gestão do tempo e no gerenciamento de suas atividades.

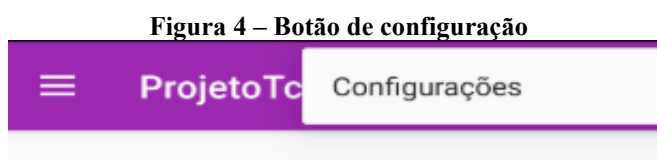
A periodicidade deve ser informada no caso de tarefas que se repetem diária, semanal, quinzenal ou mensalmente. O *status* representa o fluxo da tarefa, se foi iniciada, concluída, cancelada ou suspensa.

As entidades de tarefa e grupo foram desenvolvidas no SQLite, pois o acesso do banco de dados precisa ser feito local e individual para cada usuário, conforme a Figura 2.

Uma tarefa pode ser compartilhada com uma ou mais pessoas e as informações da tarefa compartilhada e das pessoas ficam armazenadas na entidade denominada Compartilhamento. Essa entidade foi desenvolvida no Firebase, pois o aplicativo não possui um lado servidor, e os dados das tarefas compartilhadas precisavam ser centralizadas para os usuários terem acesso. Devido a essas situações foi escolhido o Firebase para a integração, conforme a Figura 3.

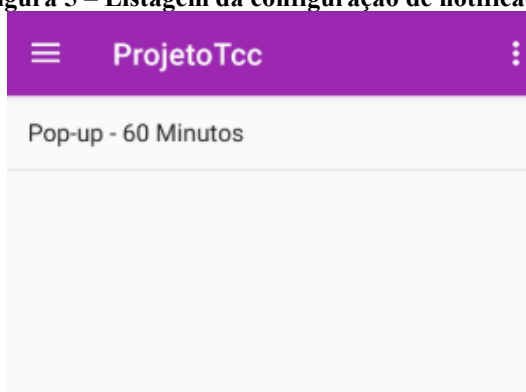
4.3 APRESENTAÇÃO DO SISTEMA

A Figura 4 ilustra o botão de configuração de notificações, que fica no canto superior direito da tela inicial, e nas demais telas do aplicativo. Ao clicar nesse botão, o aplicativo irá mostrar uma listagem das configurações do aplicativo.



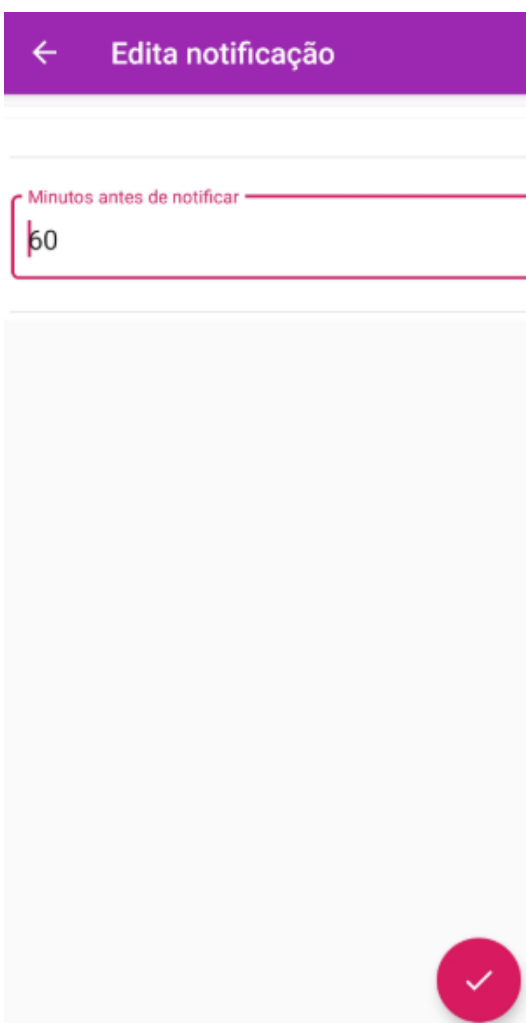
Fonte: Autoria própria.

Na Figura 5 é ilustrada a listagem de configurações, na qual é possível selecionar a configuração do tempo para ser notificado antes da realização de cada tarefa.

Figura 5 – Listagem da configuração de notificação

Fonte: Autoria própria.

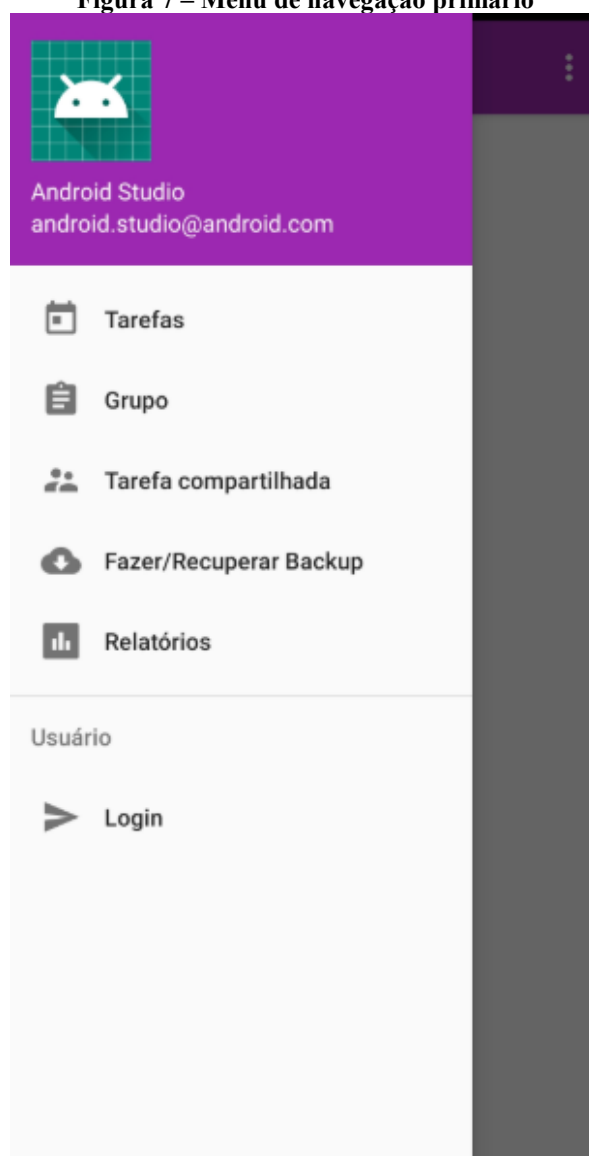
A Figura 6 representa a tela para a edição da configuração de notificação, na qual o usuário pode escolher quantos minutos ele deseja ser notificado antes do início da tarefa. Essa configuração é realizada em minutos.

Figura 6 – Tela de edição de notificação

Fonte: Autoria própria.

A Figura 7 ilustra o menu de navegação primário, que possibilita o acesso a todas as funções do aplicativo, ele fica disponível no botão localizado no canto superior esquerdo da tela. Esse menu contém as opções para o cadastro de tarefas, grupos e compartilhamento de tarefas, fazer e recuperar *backup*, gerar relatórios e permite realizar a autenticação do usuário no sistema.

Figura 7 – Menu de navegação primário



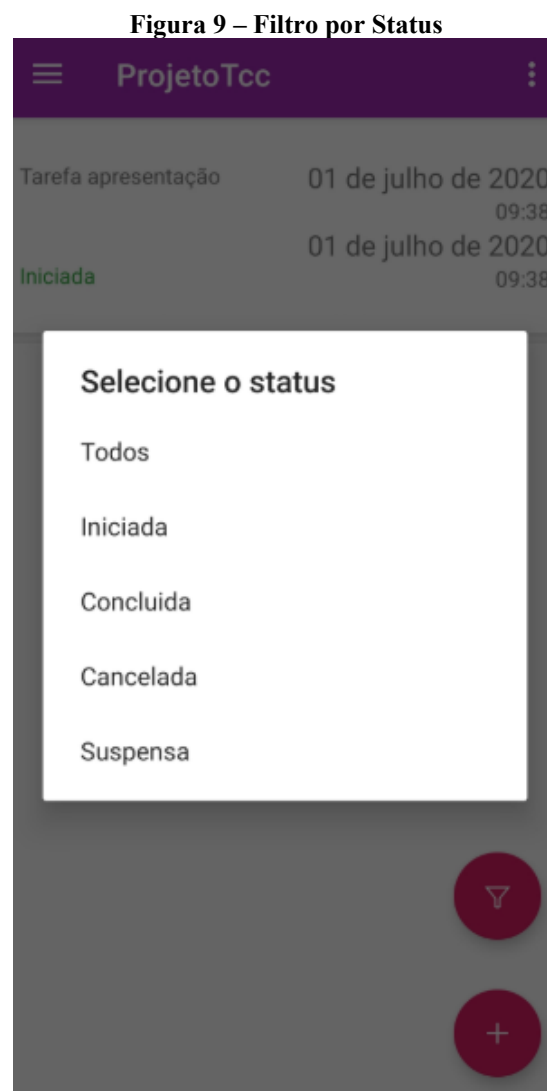
Fonte: Autoria própria.

A Figura 8 ilustra a listagem de tarefas cadastradas, na qual aparece as informações de nome, *status* e data e de início e término. Ao clicar no botão com o ícone de “+” é possível fazer a inclusão, edição e exclusão de uma nova tarefa e, ao clicar no botão com o ícone de filtro, é possível filtrar as tarefas pelo seu *status*, que pode ser iniciada, concluída, cancelada e suspensa.

Ao clicar no botão de filtro, da Figura 8, a tela da Figura 9 é exibida, na qual o usuário pode selecionar as tarefas que devem ser listadas pelo seu *status*.



Fonte: Autoria própria.



Fonte: Autoria própria.

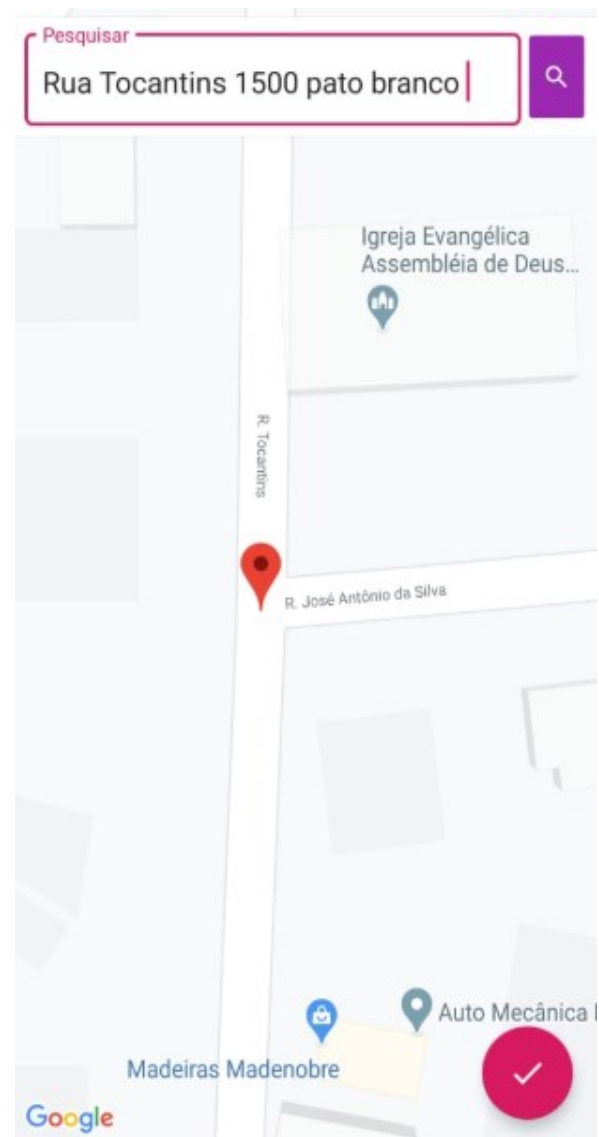
A Figura 10 exibe a tela de tarefas, na qual é possível fazer a inclusão e a edição de uma tarefa informando os dados de nome, descrição, data e hora de início, data e hora de término, grupo que essa tarefa pertence, *status*, prioridade (urgente, importante, circunstancial) e periodicidade (diariamente, semanal, quinzenal e mensal), a periodicidade informada através desta tela é apenas informativa para o usuário. Também é possível informar o local onde a tarefa ou o compromisso será realizado clicando no botão “Selecionar Local”.

A Figura 11 ilustra o mapa utilizado para selecionar o local onde a tarefa ou o compromisso será realizado. Nesse mapa é possível pesquisar o endereço ou selecionar o ponto por meio do clique no mapa em cima do local desejado.

Figura 10 – Tela de Tarefas

Fonte: Autoria própria.

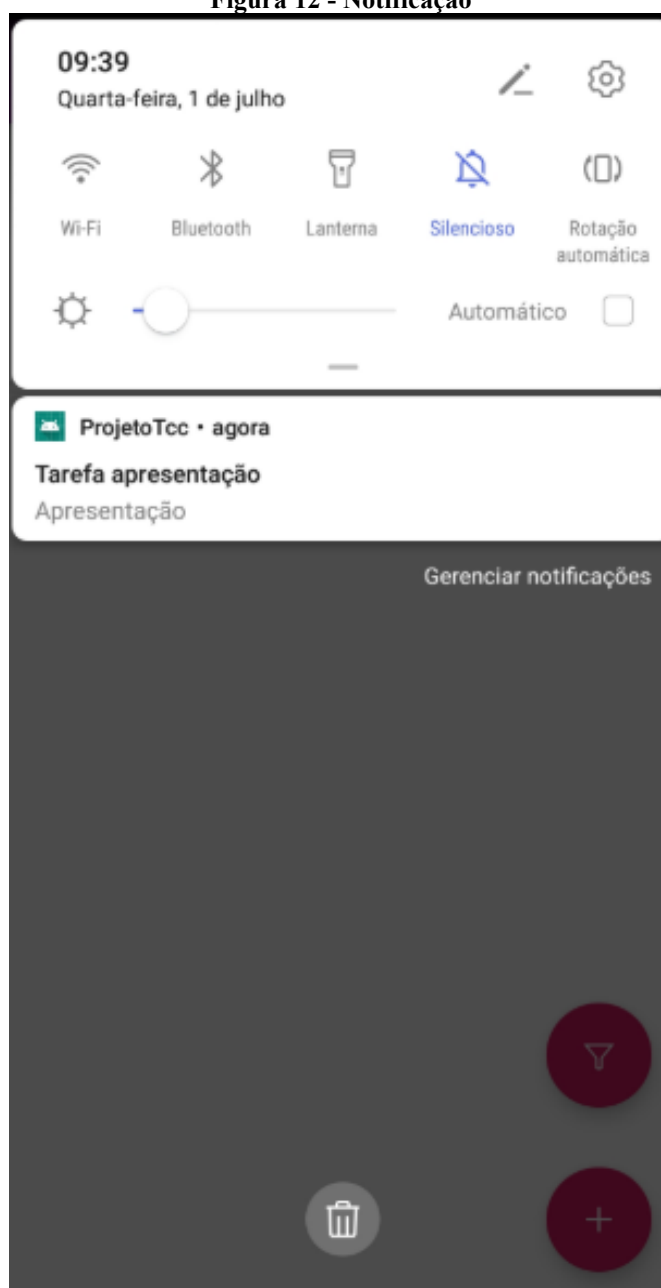
Figura 11 – Mapa de local da tarefa



Fonte: Autoria própria.

A Figura 12 ilustra a notificação recebida antes da tarefa ser iniciada. Por padrão, essa notificação aparece 10 minutos antes da data e da hora de início da tarefa. É possível fazer a configuração deste tempo pela tela de configuração ilustrada na Figura 4, o alerta será mostrado antes da hora de início da tarefa. Na notificação, são mostrados o nome e a descrição da tarefa.

Figura 12 - Notificação

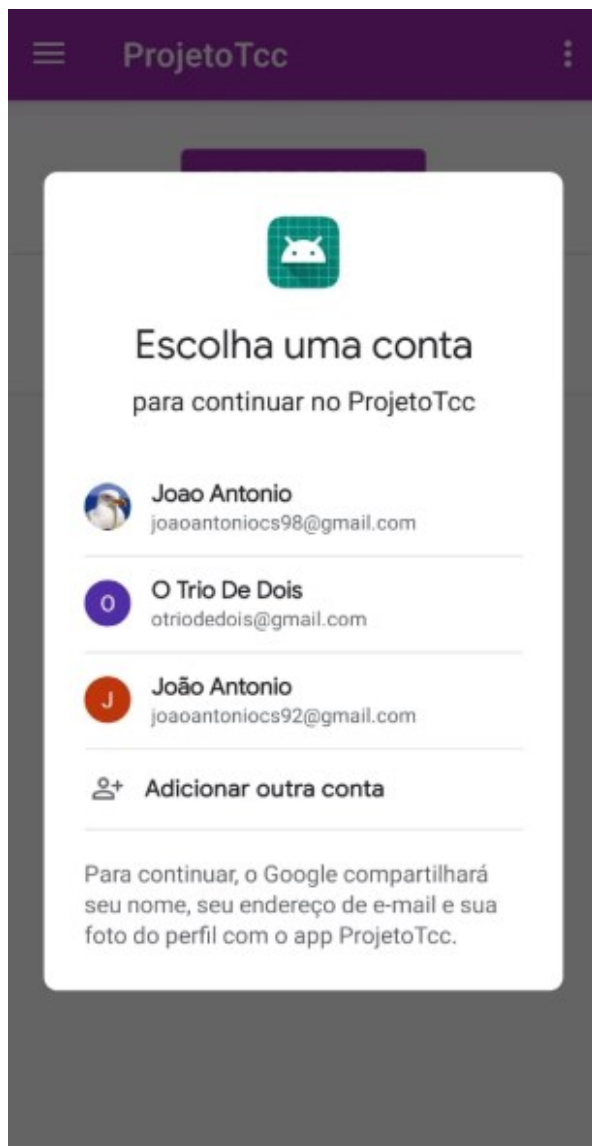


Fonte: Autoria própria.

A Figura 13 exibe a tela de autenticação do usuário para que ele tenha acesso às rotinas de tarefas compartilhadas.

A Figura 14 ilustra a lista de tarefas compartilhadas, na qual serão mostradas as tarefas que o usuário autenticado criou, ou as tarefas em que ele for convidado. Nessa tela, o usuário pode fazer a inclusão, edição e exclusão da tarefa e filtrar pelo *status*.

Figura 13 – Tela de Login



Fonte: Autoria própria.

Figura 14 – Listagem de Tarefas Compartilhadas



Fonte: Autoria própria.

A Figura 15 exibe a tela de tarefas compartilhadas, na qual é possível fazer a inclusão e edição de uma tarefa. Deve ser informado o local onde ela será realizada e o *e-mail* do convidado que participará da tarefa, o nome, a descrição, a data e hora de início, a data e hora de término, o grupo, o *status*, a prioridade e a periodicidade da tarefa.

Para que o email seja aceito o , usuário convidado já deve ter realizado a autenticação no aplicativo, caso contrário irá mostrar uma mensagem de alerta de que o *e-mail* do convidado não está disponível.

Figura 15 - Tela de Tarefa Compartilhada

← Edita tarefa compartilhada

+ SELECIONAR LOCAL

Email Do Convidado
joaoantoniocs92@gmail.com

Nome da tarefa
Teste

Descrição
Anajana

Data Inicio
27/06/2020 22:08

Data Final
27/06/2020 22:08

Grupo
Teste

Status
1 - Iniciada

Prioridade
1 - Urgente

Periodicidade

Fonte: Autoria própria.

A Figura 16 ilustra a tela de *backup* do banco de dados, na qual o usuário precisa estar autenticado para fazer o *backup* do banco atual salvando-o na nuvem, por meio da *Application Programming Interface* (API) do *Firebase Storage*, nele ficam armazenados apenas o último *backup* feito pelo usuário e também tem a opção de restaurar o banco de dados.

Figura 16 – Backup e Restauração do banco de dados.

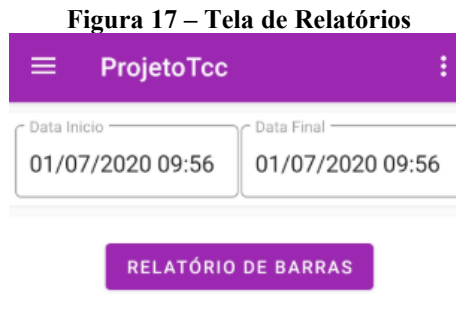
☰ ProjetoTcc ⋮

FAZER BACKUP

RESTAURAR BACKUP

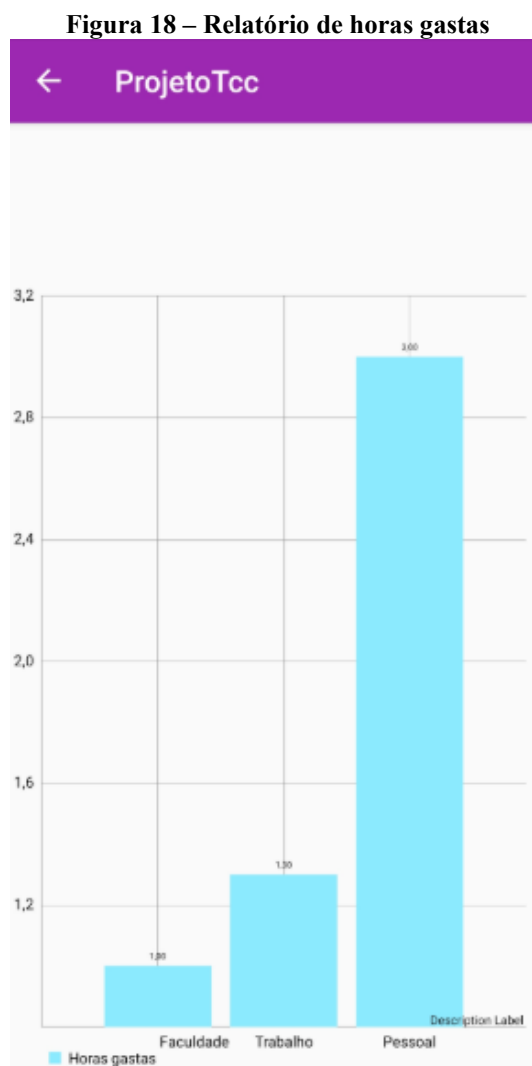
Fonte: Autoria própria.

A Figura 17 representa a tela de relatórios, na qual tem os filtros de data e hora de início e término. A data de início deve ser menor que a data de término, sendo que o aplicativo realiza essa validação.



Fonte: Autoria própria.

A Figura 18 ilustra o gráfico de horas gastas nas tarefas separadas por grupos.



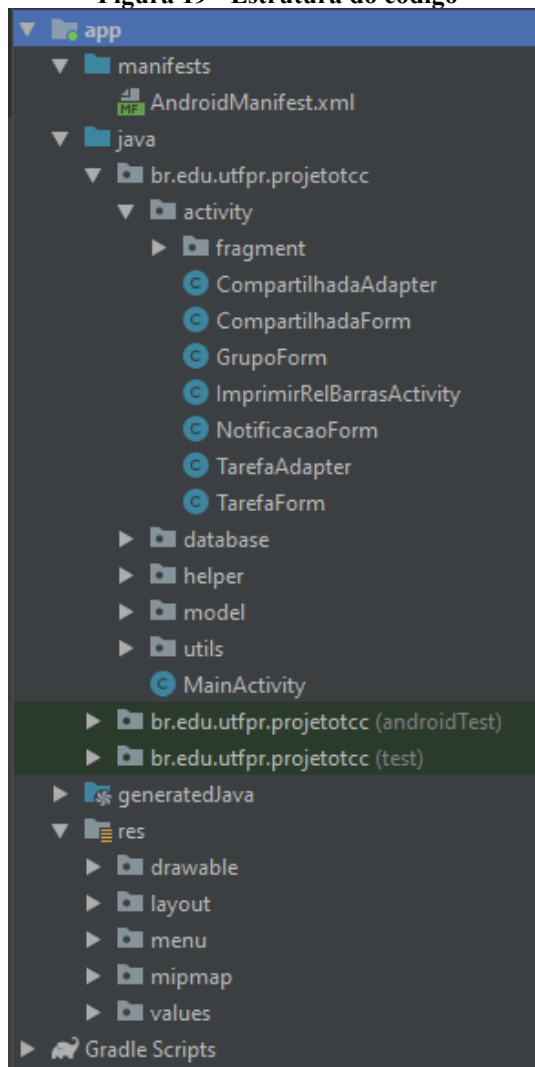
Fonte: Autoria própria.

4 IMPLEMENTAÇÃO DO SISTEMA

Serão apresentados, a seguir, alguns trechos de códigos do aplicativo, com o intuito de mostrar e exemplificar como foi realizada a organização e a implementação do aplicativo desenvolvido.

Os arquivos no projeto do Android, por padrão, são exibidos como modo Android, que faz com que o desenvolvedor tenha uma navegação intuitiva e facilitada, pois ela fica disponível com os padrões de *packages*, *Project*. A Figura 19 ilustra como a estrutura do aplicativo foi organizada.

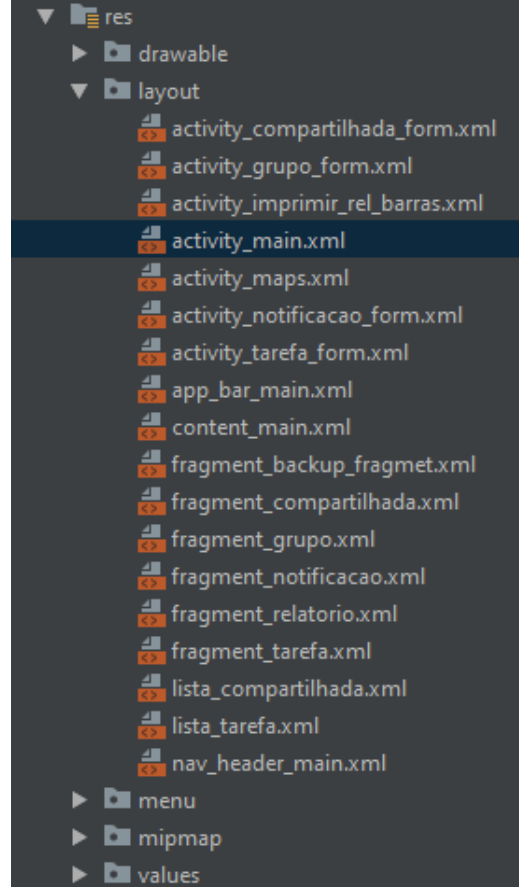
Figura 19 - Estrutura do código



Fonte: Autoria própria.

No diretório *manifest*, ficam localizados os arquivos responsáveis pela designação das permissões e configurações para o aplicativo, sendo assim, essencial para o seu funcionamento. A Figura 20 ilustra a pasta “*res*” do projeto, no qual fica armazenado a parte responsável pela aparência do aplicativo, as imagens ficam na pasta *drawable*, as telas ficam na pasta *layout*, os ícones no *mipmap* e os valores estáticos na pasta *values*.

Figura 20 – Estrutura das interfaces do projeto



Fonte: Autoria própria.

A Listagem 1 ilustra a codificação da função “*onNavigationItemSelected*”, da classe *MainActivity*, onde o usuário pode acessar as funcionalidades do aplicativo. Após a seleção de qual funcionalidade ele deseja, será direcionado para a respectiva tela.

Listagem 1 – Método responsável pela seleção do menu

```

156     public boolean onNavigationItemSelected(MenuÍtem item) {
157         int id = item.getItemId();
158
159
160         if (id == R.id.nav_agenda) {
161             TarefaFragment tarefaFragment = new TarefaFragment();
162             getSupportFragmentManager()

```

```
        .beginTransaction()
        .replace(R.id.conteudoFragment, tarefaFragment)
        .commit();
163     } else if (id == R.id.nav_grupo) {
164         GrupoFragment grupoFragment = new GrupoFragment();
165         getSupportFragmentManager()
            .beginTransaction()
            .replace(R.id.conteudoFragment, grupoFragment)
            .commit();
166     } else if (id == R.id.nav_compartilhada) {
167         CompartilhadaFragment compartilhadaFragment =
            new CompartilhadaFragment();
168         getSupportFragmentManager()
            .beginTransaction()
            .replace(R.id.conteudoFragment, compartilhadaFragment)
            .commit();
169     } else if (id == R.id.nav_login) {
170         try {
171             signIn();
172
173             nav_item_login.setVisible(false);
174             nav_item_logout.setVisible(true);
175         } catch (Exception ex) {
176             ex.printStackTrace();
177         }
178     } else if (id == R.id.nav_logout) {
179         try {
180             signOut();
181
182             nav_item_login.setVisible(true);
183             nav_item_logout.setVisible(false);
184         } catch (Exception ex) {
185
186         }
187     } else if (id == R.id.nav_backup) {
188         BackupFragment backupFragment = new BackupFragment();
189         getSupportFragmentManager()
            .beginTransaction()
            .replace(R.id.conteudoFragment, backupFragment)
            .commit();
190     } else if (id == R.id.nav_relatorios) {
191         RelatorioFragment relatorioFragment = new RelatorioFragment();
192         getSupportFragmentManager()
            .beginTransaction()
            .replace(R.id.conteudoFragment, relatorioFragment)
            .commit();
193     }
194
195     DrawerLayout drawer = findViewById(R.id.drawer_layout);
```

```

196         drawer.closeDrawer(GravityCompat.START);
197         return true;
198     }

```

Fonte: Autoria própria.

A Listagem 2 se refere a função “*signIn*”, da classe *MainActivity*, onde após autenticar-se com a conta do Google e obter sucesso, o usuário faz a chamada da função “*firebaseLogin*”, onde será criada uma autenticação com o Firebase Authentication, retornando nome, e-mail e a imagem do usuário.

Listagem 2 – Métodos responsáveis pelo Login e criação do usuário.

```

218
219     private void signIn() {
220         Intent intent = new Intent(Auth.GoogleSignInApi
221             .getSignInIntent(mGoogleApiClient));
222         startActivityForResult(intent, 1);
223     }
224
225     @Override
226     protected void onActivityResult(int requestCode, int resultCode,
227         @Nullable Intent data) {
228         super.onActivityResult(requestCode, resultCode, data);
229
230         if (requestCode == 1) {
231             GoogleSignInResult result = Auth.GoogleSignInApi
232                 .getSignInResultFromIntent(data);
233             if (result.isSuccess()) {
234                 GoogleSignInAccount account = result.getSignInAccount();
235                 firebaseLogin(account);
236             }
237         }
238     }
239
240     private void firebaseLogin(GoogleSignInAccount account) {
241         final AuthCredential credential = GoogleAuthProvider
242             .getCredential(account.getIdToken(), null);
243         mFirebaseAuth.signInWithCredential(credential)
244             .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
245                 @Override
246                 public void onComplete(@NonNull Task<AuthResult> task) {
247                     if (task.isSuccessful()) {
248                         mFirebaseUser = mFirebaseAuth.getCurrentUser();
249                         exibirDados(mFirebaseUser);
250                         alert("Usuário conectado");
251                     } else {
252                         alert("Falha na autenticação");
253                     }
254                 }
255             })

```

```

251         });
252     }

```

Fonte: Autoria própria.

A Listagem 3 refere-se à codificação da função do filtro na listagem das tarefas, chamando a função “*carregaTarefasFiltradas*”, presente na classe *TarefaFragment*, nela é informada o *id* do grupo que deseja ser filtrado, assim é retornado um cursor de apenas as tarefas com o determinado grupo.

Listagem 3 – Carregar Tarefas Filtradas

```

135     private void carregaTarefasFiltradas(View rootView, int idStatus) {
136         //Listagem das tarefas filtradas por status
137         Cursor tarefa = dao.findTarefasFiltradas(idStatus);
138         final TarefaAdapter tarefaAdapter =
139             new TarefaAdapter(getContext(), tarefa, idStatus);
140         listaTarefas.setAdapter(tarefaAdapter);

```

Fonte: Autoria própria.

A Listagem 4 apresenta a consulta em *Structured Query Language* (SQL) que retorna o cursor responsável de mostrar apenas as tarefas que forem do grupo selecionado.

Listagem 4 – Query responsável por carregar as Tarefas filtradas

```

38     @Query("SELECT * " +
39         "FROM tarefa " +
40         "WHERE EStatus = :idStatus")
41     Cursor findTarefasFiltradas(int idStatus);

```

Fonte: Autoria própria.

A Listagem 5 refere-se ao método “*criarNotificacao*”, que é responsável pela criação do *pop-up*. Para criar a notificação são calculados os minutos configurados para fazer o agendamento da notificação antes das tarefas terem início, esta notificação alerta o usuário antes da tarefa ser iniciada. Para fazer o agendamento da notificação foi utilizada a classe *AlarmManager* do próprio sistema operacional Android, onde ele é responsável por fazer o controle das notificações.

Listagem 5 – Método que cria a notificação

```

369     private void criarNotificacao(int comId) {
370         Notificacao notificacao =
371             (Notificacao) roomNotificacaoDao.findOne();
372         long diffNotificacao = new Long(notificacao.getTempo());
373         diffNotificacao *= 60;
374         diffNotificacao *= 1000;
375         Calendar date = dataInicial;
376         date.set(Calendar.SECOND, 0);
377         long diff = (date.getTimeInMillis() - diffNotificacao);
378
379         // AlarmService
380         Intent intent =
381             new Intent(TarefaForm.this, ServicoNotificacao.class);
382         intent.putExtra("comId", comId);
383         PendingIntent pendingIntent = PendingIntent
384             .getBroadcast(TarefaForm.this, comId, intent
385                 , 0);
386         AlarmManager alarmManager =
387             (AlarmManager) getSystemService(ALARM_SERVICE);
388
389         if ( diff > 0 ) {
390             alarmManager
391                 .setExact(AlarmManager.RTC_WAKEUP, diff, pendingIntent);
392         } else {
393             alarmManager
394                 .setExact(AlarmManager.RTC_WAKEUP, System
395                     .currentTimeMillis(), pendingIntent);
396         }
397     }

```

Fonte: Autoria própria.

A Listagem 6 se refere a criação do canal de notificação (*NotificationChannel*), este canal é obrigatório a partir da versão do Android 8.0+, nele são informados o nome do canal, a importância para ele ser mostrado para o usuário e a sua descrição. Esse canal é necessário para que a aplicação tenha acesso a notificar o sistema operacional.

Listagem 6 – Criar canal de notificação

```

392     private void criarCanalNotificacao() {
393         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
394             CharSequence nome = "ChannelNotificacao";
395             String descricao = "Descricao channel";
396             int importante = NotificationManager.IMPORTANCE_DEFAULT;
397         }

```

```

398     NotificationChannel channel =
           new NotificationChannel("ChannelNotificacao",
           nome, importante);
399     channel.setDescription(descricao);
400
401     NotificationManager notificationManager =
           getSystemService(NotificationManager.class);
402     notificationManager.createNotificationChannel(channel);
403     }
404 }

```

Fonte: Autoria própria.

Na Listagem 7 é exibido o método para pesquisar a latitude e a longitude no mapa. Por meio do texto digitado no campo de pesquisa do endereço, ele utiliza a classe *Geocoder* do Android, neste caso está sendo fixo um máximo de cinco resultados para esta pesquisa, mas utiliza-se somente o primeiro para obter o valor da latitude e da longitude para ser exibido no mapa.

Listagem 7 – Busca localização pelo endereço

```

79     public LatLng getLocationFromAddress(String strAddress) {
80         Geocoder coder = new Geocoder(this);
81         List<Address> address;
82         LatLng pl = null;
83
84         try {
85             address = coder.getFromLocationName(strAddress, 5);
86             if (address == null) {
87                 return null;
88             }
89             Address location = address.get(0);
90             location.getLatitude();
91             location.getLongitude();
92
93             pl =
           new LatLng(location.getLatitude(), location.getLongitude());
94
95             return pl;
96         } catch (Exception e) {
97             e.printStackTrace();
98         }
99         return pl;
100    }

```

Fonte: Autoria própria.

A Listagem 8 se refere o método “*getAddressFromLocation*”, da Classe *MapsActivity*, nela tem como função buscar a rua, cidade, bairro entre outras informações por meio da classe *Geocoder*, que obtém o valor da latitude e da longitude informada.

Listagem 8 – Busca endereço pela localização

```

103     public String getAddressFromLocation() {
104         Geocoder geocoder;
105         List<Address> addresses;
106         geocoder = new Geocoder(this, Locale.getDefault());
107
108         try {
109             addresses = geocoder.getFromLocation(latitude, longitude, 1);
110             String endereco = addresses.get(0).getAddressLine(0);
111             return endereco;
112         } catch (IOException e) {
113             e.printStackTrace();
114         }
115         return "";
116     }

```

Fonte: Autoria própria.

A Listagem 9 exibe a função “*getPosicaoAtual*”, que é responsável por buscar a localização atual do dispositivo por meio do GPS. É necessário verificar se o usuário concedeu a permissão para o aplicativo ter acesso ao GPS do dispositivo e, posteriormente obter a última localização conhecido do dispositivo por meio da classe *LocationManager*, transforma-se essa informação em um objeto *Location* do Android para obter acesso a longitude e latitude da localização. São exibidas no mapa a localização geográfica e o endereço desta localização.

Listagem 9 – Obtém a posição atual do GPS

```

139     public void getPosicaoAtual() {
140         mMap.clear();
141         LocationManager lm =
142             (LocationManager) getSystemService(Context.LOCATION_SERVICE);
143         if (checkSelfPermission
144             (Manifest.permission.ACCESS_FINE_LOCATION)
145             != PackageManager.PERMISSION_GRANTED
146             && checkSelfPermission
147             (Manifest.permission.ACCESS_COARSE_LOCATION)
148             != PackageManager.PERMISSION_GRANTED) {
149             Toast.makeText(this,
150                 "Você precisa permitir a permissão de " +
151                 "acesso do gps

```

```

        para melhor funcionamento do mapa!!",
        Toast.LENGTH_LONG). show ();
146         return;
147     }
148     Location location = lm.getLastKnownLocation
        (LocationManager.GPS_PROVIDER);
149     longitude = location.getLongitude();
150     latitude = location.getLatitude();
151
152     // Add marcador padrao e move a
        camera para localizacao atual do gps
153     LatLng atual = new LatLng(latitude, longitude);
154     mMap.addMarker(new MarkerOptions()
        .position(atual).title(getAddressFromLocation()));
155     mMap.moveCamera(CameraUpdateFactory.newLatLngZoom
        (atual, 20));
156 }

```

Fonte: Autoria própria.

A Listagem 10 se refere a verificação do e-mail do convidado, por meio da classe *FirebaseAuth* do *Firebase authentication*, nele chamamos o método *fetchSignInMethodsForEmail*, ele é responsável por verificar se o e-mail do convidado informado já está cadastrado na base de dados do aplicativo, se ele está cadastrado no banco de dados do aplicativo, a função de salvar é invocada para salvar a tarefa compartilhada.

Listagem 10 – Verificação se o e-mail do convidado já existe ou não

```

367 public void btSalvarOnClick(View view) {
368     //Verifica se email ja existe ou nao.
369     if(!etEmailConvidado.getText().toString().isEmpty()){
370     FirebaseAuth.getInstance()
        .fetchSignInMethodsForEmail(etEmailConvidado.getText().toString())
371         .addOnCompleteListener(
        new OnCompleteListener<SignInMethodQueryResult>() {
372             @Override
373             public void onComplete(@NonNull
        Task<SignInMethodQueryResult> task) {
374                 try {
375                     isNewUser = task.getResult()
        .getSignInMethods().isEmpty();
376                     salvarCompartilhada();
377                 } catch (Exception ex) {
378                     Toast.makeText(CompartilhadaForm.this,
        "E-mail do convidado não cadastrado
        ou inválido!", Toast.LENGTH_LONG). show ();
379                 }

```

```

380         }
381     });
382     }else{
383         Toast.makeText(CompartilhadaForm.this,
384             "E-mail do convidado não cadastrado ou inválido!",
385             Toast.LENGTH_LONG).show();
386     }
387 }

```

Fonte: Autoria própria.

Na Listagem 11 há o método de salvar a tarefa compartilhada da classe *CompartilhadaForm*, no qual obtém-se a classe de tarefa compartilhada por meio do método do *helper*. Após isso, são carregadas as informações do usuário autenticado para efetuar a ação de salvar. Inicialmente é verificado se esta tarefa possui um *id*, para identificar qual ação deve ser realizada, se é uma alteração de tarefa ou uma inclusão de uma nova. Após isso, o objeto é passado para a classe do Firebase para salvar no banco de dados.

Listagem 11 – Método de salvar Tarefa Compartilhada

```

387     private void salvarCompartilhada() {
388         Compartilhada compartilhada = helper.pegaCompartilhada(this);
389         FirebaseUser firebaseUser =
390             FirebaseAuth.getInstance().getCurrentUser();
391
392         if(!isNewUser){
393             if(compartilhada.getUid() != null){
394                 FirebaseHelper
395                     .getInstance()
396                     .getDatabaseReference()
397                     .child("Compartilhada")
398                     .child(compartilhada.getUid()).setValue(compartilhada);
399             }else{
400                 if(FirebaseAuth.getInstance() != null){
401                     compartilhada
402                         .setEmailPrincipal(firebaseUser.getEmail());
403                 }
404                 compartilhada.setUid(UUID.randomUUID().toString());
405                 FirebaseHelper.getInstance()
406                     .getDatabaseReference()
407                     .child("Compartilhada")
408                     .child(compartilhada.getUid())
409                     .setValue(compartilhada);
410             }
411         }
412         Toast.makeText(this,
413             "Sucesso!", Toast.LENGTH_LONG).show();
414     }
415 }

```

```

403     }else{
404         Toast.makeText(this,
                "E-mail do convidado não cadastrado ou inválido!",
                Toast.LENGTH_LONG).show();
405     }
406 }

```

Fonte: Autoria própria.

A Listagem 12 refere-se à codificação do botão imprimir relatório, que verifica se a data informada pelo usuário é válida. O intervalo das datas informadas é obtido e são exibidas apenas as tarefas que tem a data inicial entre este intervalo e, se for inválido, será mostrada uma mensagem de alerta para informar o usuário.

Listagem 12 – Botão para imprimir relatório

```

76     //Botão para imprimir relatorio de barras
77     MaterialButton btImprimirRelBarras =
            (MaterialButton) rootView
                .findViewById(R.id.btImprimirRelBarra);
78     btImprimirRelBarras.setOnClickListener(new View.OnClickListener() {
79         @Override
80         public void onClick(View view) {
81             if( (dataInicial != null && dataFinal != null)
82                 && (dataInicial.getTimeInMillis()
83                     <= dataFinal.getTimeInMillis()) ){
84                 Intent intent = new Intent(context,
85                     ImprimirRelBarrasActivity.class);
86                 intent.putExtra("dataInicial",
87                     dataInicial.getTimeInMillis());
88                 intent.putExtra("dataFinal",
89                     dataFinal.getTimeInMillis());
90                 startActivity(intent);
91             }else{
92                 Toast.makeText(getContext(),
93                     "A data inicial não pode ser maior
94                     que a data final!", Toast.LENGTH_LONG).show();
95             }
96         }
97     });

```

Fonte: Autoria própria.

A Listagem 13 exibe uma lista personalizada por meio da codificação em *Extensible Markup Language* (XML), para fazer a listagem das Tarefas, pelas *TextViews* e organizados com *RelativeLayout* e *LinearLayout*.

Listagem 13 – XML de lista de Tarefas

```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   android:orientation="vertical"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent">
5   <LinearLayout
6     android:layout_width="match_parent"
7     android:layout_height="138dp"
8     android:layout_gravity="center"
9     android:layout_marginBottom="5dp"
10    android:background="#FFF"
11    android:elevation="5dp"
12    android:orientation="horizontal"
13    android:paddingHorizontal="10dp"
14    android:paddingVertical="20dp">
15
16    <RelativeLayout
17      android:layout_width="0dp"
18      android:layout_height="match_parent"
19      android:layout_weight="50">
20
21      <LinearLayout
22        android:layout_width="match_parent"
23        android:layout_height="wrap_content"
24        android:layout_centerInParent="true"
25        android:orientation="vertical">
26
27        <LinearLayout
28          android:layout_width="145dp"
29          android:layout_height="wrap_content">
30
31          <TextView
32            android:id="@+id/tvNome"
33            android:layout_width="145dp"
34            android:layout_height="64dp"
35            android:text="Nome "
36            android:textSize="14sp" />
37        </LinearLayout>
38        <LinearLayout
39          android:layout_width="145dp"
40          android:layout_height="wrap_content">
41
42          <TextView
43            android:id="@+id/tvStatus"
44            android:layout_width="145dp"
45            android:layout_height="wrap_content"

```



```
95         android:text="Data Final "  
96         android:textSize="18sp"  
97         android:gravity="right"/>  
98  
99         <TextView  
100             android:id="@+id/tvHoraFinal"  
101             android:layout_width="match_parent"  
102             android:layout_height="wrap_content"  
103             android:text="Hora Final "  
104             android:textSize="14sp"  
105             android:gravity="right"/>  
106     </LinearLayout>  
107 </LinearLayout>  
108  
109     </RelativeLayout>  
110 </LinearLayout>  
111  
112 </LinearLayout>
```

Fonte: Autoria própria.

5 CONCLUSÃO

Este trabalho teve como objetivo propor uma solução para dispositivos móveis nativos do Android. Procedeu-se com a execução e explanação de fases de desenvolvimento, como o levantamento de requisitos, a análise e levantamento de requisitos, o projeto, a escolha de tecnologias e ferramentas para auxiliar no desenvolvimento, implementação e testes.

A solução foi desenvolvida como um aplicativo nativo Android em linguagem Java, com o Material Design na implementação de suas telas, ele fica hospedado local no dispositivo móvel do usuário e com a persistência de dados efetuada em um banco de dados relacional, que fica armazenado no dispositivo móvel também, utilizando como um serviço de computação em nuvem para armazenar alguns dados a mais de seu usuário.

A maior dificuldade no desenvolvimento do trabalho foi com relação à rotina de *backup* e *restore* do banco de dados, pois havia pouca documentação sobre e, também tem poucas APIs disponibilizadas para fazer a integração. Outra dificuldade encontrada, foi mostrar apenas as tarefas compartilhadas do usuário conectado, pois o Firebase não tem suporte para mais de um parâmetro na *query* para buscar a tarefa.

Constata-se que o projeto atingiu o seu objetivo, as funcionalidades implementadas e testadas comprovam a precisão e coerência dos requisitos coletados, assim sofrendo poucas alterações durante o desenvolvimento do projeto, sendo que todas foram bem aplicadas e bem-sucedidas.

Como trabalho futuro podem ser adicionadas novas funções, como a configuração do tipo de alerta que o usuário deseja receber, compartilhamento da tarefa em tempo real para usuários convidados e novos relatórios e gráficos. Além disso, pode ser realizada a conversão do aplicativo para uma aplicação híbrida, a fim de proporcionar um aplicativo para todos os dispositivos móveis disponíveis no mercado.

REFERÊNCIAS

BARBOSA, CHRISTIAN. **A tríade do tempo**. Rio de Janeiro: Sextante, 2012.

CECTIC.BR. **Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação**. 2017a. Disponível em: <https://cetic.br/tics/domicilios/2017/individuos/J4>. Acesso em: 01 abr. 2019.

CETIC.BR. **Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação**. 2017b. Disponível em: <https://www.cetic.br/tics/domicilios/2017/individuos/J2>. Acesso em: 01 abr. 2019.

COVEY, Stephen R. **Os 7 hábitos das pessoas altamente eficazes**. São Paulo: Best Seller, 2005.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software-8ª Edição**. McGraw Hill Brasil, 2016.

INSTITUTO DE PÓS-GRADUAÇÃO E GRADUAÇÃO. **Gestão do tempo: 8 dicas para aumentar a produtividade**. 2019. Disponível em <https://www.ipog.edu.br/arquivos/downloads/e-book-gestao-do-tempo-ipog-19107214.pdf>. Acesso em: 14 mai. 2019.

LIMA, Maria do Carmo F.; JESUS, Simone B. Administração do tempo: um estudo sobre a gestão eficaz do tempo como ferramenta para o aumento da produtividade e Work Life Balance. **GeSec Revista de Gestão e Secretariado**. ISSN: 2178-9010. v.2. n. 2, 2011. Disponível em: <https://www.revistagesec.org.br/secretariado/article/view/48/109>. Acesso em: 15 mai. 2019.

MELO, Edvonaide P. S.; FARIAS Ana Paula da S. Gerenciamento do tempo e da rotina de trabalho: um estudo de caso numa agência do banco Bradesco S/A. **XI Congresso Nacional em Excelência em Gestão**. ISSN 1984-9354. 2015. Disponível em <http://www.inovarse.org/node/4216>. Acesso em: 14 mai. 2019.

ZURIARRAIN, José Mendiola. **Android já é o sistema operacional mais usado do mundo**. 2017. Disponível em: https://brasil.elpais.com/brasil/2017/04/04/tecnologia/1491296467_396232.html. Acesso em: 27 abr. 2019.