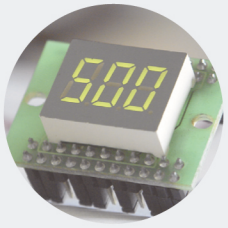
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES



DESENVOLVIMENTO DE UMA APLICAÇÃO ANDROID PARA TRATAMENTO DAS DESPESAS DOS COLABORADORES DE UMA EMPRESA

DIOGO ALBANO TEIXEIRA GOMES
novembro de 2021

POLITÉCNICO
DO PORTO

DESENVOLVIMENTO DE UMA APLICAÇÃO ANDROID PARA TRATAMENTO DAS DESPESAS DOS COLABORADORES DE UMA EMPRESA

Diogo Albano Teixeira Gomes

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2021

Relatório elaborado para satisfação parcial dos requisitos da Unidade Curricular de Tese/Dissertação do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Diogo Albano Teixeira Gomes, Nº1140247, 1140247@isep.ipp.pt
Orientação científica: António Avelino Amorim Marques, aav@isep.ipp.pt

Empresa: Instituto Eletrotécnico Português

Orientador: Marco Areosa, md@iep.pt

Departamento de Engenharia Eletrotécnica
Mestrado em Engenharia Eletrotécnica e de Computadores
Instituto Superior de Engenharia do Porto
18 de novembro de 2021

Agradecimentos

Desejo, primeiramente, agradecer ao meu orientador do ISEP, Engenheiro Avelino Marques, pela sua disponibilidade, o seu apoio e toda a partilha de conhecimento durante o desenvolvimento desta tese.

À minha namorada, Diana Santos, agradeço por toda a força e motivação dada durante todo o processo.

Por fim, à minha família e amigos, agradeço por toda a ajuda e por todo o apoio que me deram nesta etapa académica.

Resumo

Este projeto consiste no desenvolvimento de uma aplicação *Android*, que pretende dar resposta à necessidade de reformulação do sistema de submissão de despesas dos colaboradores do Instituto Eletrotécnico Português (IEP). Este sistema de submissão de despesas abrange uma média de 100 colaboradores todos os meses.

O processo atual de submissão de despesas consiste no preenchimento de uma folha de despesas com os diferentes valores das despesas do mês correspondente. Juntamente com a folha de despesas o colaborador tem de entregar as faturas na sede da empresa. A aplicação *Android* desenvolvida permitiu eliminar o registo em papel e poupar recursos financeiros consideráveis, com a eliminação da deslocação propositada dos colaboradores do IEP.

A solução permite a recolha de dados de forma estruturada, a aquisição fotográfica das faturas, o envio de todos os dados para um servidor da empresa, reduzindo assim o tempo dispensado pelos colaboradores neste processo.

Os dados recolhidos são guardados de forma organizada num diretório criado a partir dos dados inseridos inicialmente pelo utilizador. Estes dados permitem identificar de forma clara o colaborador da empresa, possibilitando uma melhoria no processo de validação das despesas por parte dos recursos humanos, sendo que, os dados de cada colaborador vão estar em numa pasta única, com um ficheiro com um nome único e as respetivas faturas com um nome indicativo do dia e tipo de despesa. Além disso, uma vez que os dados vão ser enviados para um servidor da empresa, os mesmo podem ser consultados em qualquer ponto do país, algo importante numa altura em que o teletrabalho é, atualmente, uma realidade para várias empresas.

A aplicação foi testada no IEP e concluiu-se que cumpre todos os objetivos. O teste foi realizado com 20 colaboradores no mês de outubro.

Palavras-Chave

Android, IEP, despesas, Android Studio, smartphone, aplicação

Abstract

This project consists of the development of an Android application, which aims to respond to the need to reformulate the expenses submission system of the employees of the Instituto Eletrotécnico Português (IEP). The system covers an average of 100 employees per month.

It should be noted that current submission process is based on filling in a sheet with the expenditure amounts for the corresponding month. Along with the expense sheet, the employee must hand the invoices to the company's headquarters. The developed Android application made it possible to do away with paper registration and save considerable financial resources, with the elimination of deliberate travel by IEP employees.

The solution allows the collection of data in a structured way, the photographic acquisition of invoices, the sending of all data to a company server, thus reducing the time spent by employees in this process.

The collected data are organized and stored in a directory created from the data initially entered by the user. These data allow for a clear identification of the company's employee, enabling an improvement in the process of validation of expenses by the human resources. Furthermore, the data of each employee will be in a single folder, with a file with a unique name and the respective invoices with a name indicating the day and type of expense. In addition, since the data will be sent to a company server, it can be consulted anywhere in the country, which is an added value in a society where teleworking is increasingly a reality for companies.

The application has been tested on the IEP and found to meet all goals. The test was carried out with 20 employees in the month of October.

Keywords

Android, IEP, expenses, Android Studio, smartphone, application

Índice

AGRADECIMENTOS ..... **I**

RESUMO ..... **III**

ABSTRACT ..... **V**

ÍNDICE ..... **VII**

ÍNDICE DE FIGURAS ..... **IX**

ÍNDICE DE TABELAS ..... **XI**

ACRÓNIMOS ..... **XIII**

1. INTRODUÇÃO **1**

1.1. INSTITUTO ELETROTÉCNICO PORTUGUÊS 2
1.2. CONTEXTUALIZAÇÃO 3
1.3. OBJETIVOS 3
1.4. CALENDARIZAÇÃO 4
1.5. ORGANIZAÇÃO DO RELATÓRIO 4

2. SISTEMA DE SUBMISSÃO DE DESPESAS DO IEP **7**

2.1. SISTEMA ATUAL DE SUBMISSÃO DE DESPESAS 7
2.1.1. Custos atuais com este sistema *10*
2.2. OPINIÃO DOS COLABORADORES 11
2.3. SOLUÇÃO A IMPLEMENTAR 15
2.3.1. Vantagens e desvantagens *16*

3. SISTEMA OPERATIVO *ANDROID* **17**

3.1. *ACTIVITY*(ATIVIDADE) 18
3.2. INTENT 18
3.3. OPÇÕES DE ARMAZENAMENTO 18
3.4. MANIFESTO DE UMA APLICAÇÃO 19
3.5. ANDROID STUDIO 20

4. SISTEMA DE SUBMISSÃO DE DESPESAS INFORMATIZADO PARA O SISTEMA *ANDROID* **21**

4.1. OBJETIVOS DA APLICAÇÃO 22
4.2. FUNCIONAMENTO DA APLICAÇÃO 22
4.3. TRATAMENTO DOS DADOS – EMPRESA 30
4.4. FUNÇÕES UTILIZADAS NA APLICAÇÃO 32
4.5. FLUXOGRAMA DA APLICAÇÃO ANDROID 42

4.6. DIFICULDADES ENCONTRADAS NO DESENVOLVIMENTO DA APLICAÇÃO 44

5. CONCLUSÕES 47

REFERÊNCIAS DOCUMENTAIS 49

ANEXO A. CÓDIGO DE ALGUMAS CLASSES CRIADAS 51

ANEXO B. CÓDIGO DE ALGUNS *LAYOUT'S* CRIADOS 69

***ANEXO C.* CÓDIGO DO *ANDROIDMANIFEST.XML* 81**

Índice de Figuras

Figura 1 – Folha de despesas de deslocação do IEP......9
Figura 2 – Inquérito realizado via *Google Forms* aos colaboradores do IEP. 12
Figura 3 – Gráfico da frequência com que os colaboradores submetem as despesas......12
Figura 4 – Gráfico relativo à deslocação para entrega das despesas no IEP. 13
Figura 5 – Gráfico relativo à deslocação propositada para entrega de despesas na empresa. 13
Figura 6 – Gráfico relativo à mais-valia de uma solução 100% digital......14
Figura 7 – Gráfico relativo ao tempo dispensado na entrega das despesas......15
Figura 8 – Imagem exemplificativa do *Android Studio.*.....20
Figura 9 – *Layout* inicial da aplicação......23
Figura 10 – Informação do colaborador......24
Figura 11 – Centro de custos. 25
Figura 12 – Menu central de despesas. 26
Figura 13 – Informação para preencher despesas. 26
Figura 14 – Informações da despesa. 28
Figura 15 – Tipo de despesa. 28
Figura 16 – Informação ao utilizador do envio da despesa com sucesso. 29
Figura 17 – Ficheiros criados pela aplicação......30
Figura 18 – Ecrã inicial da *cloud.*.....31
Figura 19 – Conta Despesas da *cloud.*.....31
Figura 20 – Ficheiro enviado pelo colaborador......31
Figura 21 – Aviso foto guardada. 36
Figura 22 – Aviso foto não guardada......36
Figura 23 – Fluxograma da aplicação *Android*. 43
Figura 24 – *Layout* Menu Central Despesas. 69
Figura 25 – *Layout* Informações do colaborador. 71
Figura 26 – *Layout* Informações da despesa......75

Índice de Tabelas

Tabela 1 – Calendarização do desenvolvimento da tese. 4
Tabela 2 – Limites máximos de cada despesa instituídos pela empresa. 10
Tabela 3 – Custos estimados da empresa com o atual sistema de submissão de despesas. 11
Tabela 4 – Vantagens e desvantagens da solução a implementar. 16

Acrónimos

API – *Application Programming Interface*

APK – *Android Application Pack*

DAF – Departamento Administrativo e Financeiro

IEP – Instituto Eletrotécnico Português

SDK – *Software Development Kit*

xml – *Extensible Markup Language*

1. INTRODUÇÃO

O Instituto Eletrotécnico Português (IEP) é uma empresa em constante crescimento. Atualmente com mais de 217 colaboradores distribuídos por todo o país [1]. Destes, mais de 100 prestam serviços externos, uns de forma permanente outros de forma pontual. Nesses serviços surgem inúmeras despesas que a empresa comparticipa, desde o Almoço, Jantar, Hotel, Estacionamento, Portagens, Combustíveis, entre outros.

Atualmente, existe um método arcaico de submissão destas despesas por parte dos colaboradores. Consiste no preenchimento de uma tabela, com as diferentes despesas, e posterior entrega na sede das respetivas faturas. Posteriormente, duas colaboradoras dos recursos humanos tratam de toda a informação, sendo por fim validadas pelos respetivos responsáveis técnicos.

No tratamento dos dados por parte dos recursos humanos, além da validação dos dados inseridos é necessário determinar o valor final a restituir ao colaborador por cada despesa, dado que cada uma tem um valor máximo.

Ao tomar conhecimento do procedimento habitual desse trabalho, surgiu a ideia conjunta de desenvolver uma aplicação *Android* que agilizasse e melhorasse o processo. A proposta consiste em registar as despesas num programa informático de fácil acesso e intuitivo. Esta aplicação permite preencher as despesas de forma organizada, anexar a fatura com uma

fotografia, identificá-las com um nome específico e irrepetível, armazenar de forma estruturada em uma pasta em um *smartphone/tablet*.

A aplicação terá, ainda, a possibilidade de enviar os dados comprimidos, para uma *cloud* do IEP.

No tratamento das despesas por parte dos recursos humanos é bastante vantajoso que toda a informação recolhida esteja convenientemente estruturada e armazenada, de forma que haja uma economia de recursos por um lado e por outro uma maior exatidão e menor risco de erro nos registos.

1.1. INSTITUTO ELETROTÉCNICO PORTUGUÊS

O Instituto Eletrotécnico Português, fundado em 1981, é uma empresa com solidas raízes, reconhecida no mercado como uma infraestrutura tecnológica que se pauta pela inovação, competência e qualidade [1].

Desde o seu início têm vido a disponibilizar no mercado soluções integradas nos domínios da inspeção, auditoria, ensaios, calibrações, formação, consultoria e inovação nos setores da Energia, Comunicações, Petróleo e Gás, Ambiente, Transportes, Indústria, Saúde e Serviços [1].

É constituída por equipas altamente qualificadas e experientes, que asseguram uma cobertura nacional. Estas têm alavancado o desempenho dos seus clientes, contribuindo para o aumento da qualidade e competitividade dos seus produtos, para a redução dos riscos associados às suas atividades e para a inovação dos seus processos [1].

O elevado número de reconhecimentos e acreditações, nacionais e internacionais, que acumulam ao longo dos anos são a prova evidente de isenção, do rigor e da excelência técnica que colocam ao serviço dos seus clientes em tudo o que fazem [1].

1.2. CONTEXTUALIZAÇÃO

Este projeto surgiu após várias sugestões dos colaboradores do IEP em que mostraram quanto tempo despendiam com todo este processo mensalmente. Além de todo o dinheiro gasto mensalmente com este processo, a transição de papel para digital é neste momento uma das prioridades da empresa.

Ao tomar conhecimento do procedimento existente, e no intuito de uma transição digital, surgiu o desejo de tornar todo o processo totalmente informatizado. Para isto foi proposto desenvolver uma aplicação para o sistema *Android* onde recolhesse de forma organizada todos os dados e fotografias necessárias e, por fim, que enviasse as mesmas para os recursos humanos para posterior tratamento.

Este novo método, 100% digital, permite reduzir o arquivo em papel, poupar aproximadamente 2018,75€ com deslocações à sede e tempo de trabalho. Sendo que a informação será toda organizada e estruturada, torna-se bastante vantajoso para posterior tratamento dos recursos humanos e responsáveis de departamento.

1.3. OBJETIVOS

O principal objetivo deste projeto é desenvolver uma aplicação para o Sistema *Android*, onde possa ser realizada toda a submissão de despesas por parte dos colaboradores, eliminando assim o processo arcaico de submissão de despesas.

A aplicação a desenvolver deve disponibilizar um formulário para introdução dos dados das despesas, permitir o registo fotográfico das respetivas faturas, fazer uma pré-validação dos valores das despesas, organizar toda a informação de forma estruturada e enviar a informação para os recursos humanos.

Para o cumprimento deste objetivo, sentiu-se a necessidade de o subdividir em múltiplas tarefas, a realizar antes de partir para o objetivo principal.

A fase inicial, teve como objetivo a familiarização com o processo existente utilizado no IEP. Numa segunda fase procedeu-se ao estudo e desenvolvimento de uma aplicação que cumprisse o objetivo proposto. As diferentes fases podem ser resumidas em:

- Estudo do processo atual (custos, vantagens, desvantagens);
- Levantamento da opinião dos colaboradores com realização de um inquérito;
- Estudar as vantagens e desvantagens de um processo 100% digital;
- Estudo do sistema *Android*, linguagem de programação e ferramentas de desenvolvimento;
- Desenvolvimento da aplicação;
- Teste e validação junto dos colaboradores.

1.4. CALENDARIZAÇÃO

O desenvolvimento desta tese começou em maio de 2021 e terminou em outubro de 2021, como se pode ver na tabela 1.

Tabela 1 – Calendarização do desenvolvimento da tese.

Meses	**Maio**				**Junho**				**Julho**				**Agosto**				**Setembro**				**Outubro**			
Semanas	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Estudo do sistema despesas																								
Estudo do Sistema Android																								
Desenvolvimento da APP																								
Teste da APP																								
Elaboração do Relatório																								

1.5. ORGANIZAÇÃO DO RELATÓRIO

No Capítulo 1 apresenta-se uma introdução ao projeto e a empresa onde se realizou, a sua contextualização, os seus objetivos, calendarização e organização do relatório. No capítulo seguinte, 2, é apresentado o atual sistema de despesas, custo estimado deste processo e o inquérito realizado aos colaboradores da empresa. No capítulo seguinte, o 3º, são apresentadas algumas noções sobre o Sistema *Android*, noções essas utilizadas na

aplicação desenvolvida. No capítulo seguinte, o 4º, é apresentado o desenvolvimento da aplicação *Android*. No último capítulo, o 5º, são reunidas as principais conclusões e perspetivados futuros desenvolvimentos.

2. SISTEMA DE SUBMISSÃO DE DESPESAS DO IEP

Antes de desenvolver uma solução para substituir o atual sistema de submissão de despesas do IEP é necessário entender profundamente o mesmo. Para isso estudou-se a folha de despesas, o limite máximo de cada despesa imposto pela empresa, opinião dos colaboradores realizando um inquérito aos mesmo. E por fim, analisar a solução a implementar, bem como as suas vantagens e desvantagens.

2.1. SISTEMA ATUAL DE SUBMISSÃO DE DESPESAS

O atual sistema de submissão de despesas consiste em que cada colaborador que teve despesas durante o mês corrente, preencher uma folha tipo, Figura 1, e anexar a fatura correspondente a cada despesa. A folha e as faturas devem ser entregues na sede da empresa para os colaboradores no respetivo distrito e distritos fronteiriços. Os restantes colaboradores devem enviar as despesas por correio.

Na empresa, duas colaboradoras dos recursos humanos tratam das despesas de cada colaborador. Esse tratamento consiste em verificar o valor de cada despesa com a respetiva fatura, esse valor deve ser igual ao da fatura e para posterior pagamento deve ser igual ou inferior aos limites máximos impostos pela empresa, Tabela 2. Além da questão dos

valores devem verificar o contribuinte de cada fatura, faturas com contribuinte errado ou sem contribuinte ficam inválidas.

Por fim, a contabilidade deduz o valor válido das despesas do mês na conta corrente do respetivo colaborador.

iep
instituto electrotécnico português

Direcção Administrativa e Financeira
Despesas de Deslocação

ANO_____ **MÊS**_____ **NOME** _____
TRABALHO EFECTUADO _____

	REFEIÇÕES + ALOJAMENTO				DESPESAS C/ VIATURA MATRICULA __-__-__					
DIA	ALMOÇO	JANTAR	P.ALM.	HOTEL	ESTAC.	LAVAGEM	PORTAG.	COMBUST.	OUTROS	OBSERV
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
TOTAL	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**	**0,00**

Obs: _____

Data: ___-___-___ Assinatura:_____

AUTORIZAÇÃO (assinatura e carimbo)_____

ÁREA	LOCALIZAÇÃO	PROJECTO

Conferido por:_____ Data:_____-____-_____

Subs. Almoço a descontar:_____

DOC. Nº _____
CONTAS A DEBITAR
_____ _____
_____ _____
_____ _____
_____ _____
CONTAS A CREDITAR
_____ _____

DAF004/0

Figura 1 – Folha de despesas de deslocação do IEP.

Tabela 2 – Limites máximos de cada despesa instituídos pela empresa.

Tipo de Despesa	Valor
Pequeno Almoço	3.50€/dia
Almoço	12,50€/dia
Jantar	17,50€/dia
Lavagem da viatura	10€/mês

2.1.1. CUSTOS ATUAIS COM ESTE SISTEMA

Este sistema de submissão de despesas, além de arcaico, tem encargos financeiros consideráveis para a empresa. Esses encargos são difíceis de calcular com exatidão, mas possíveis de estimar.

A forma de estimar estes cursos é saber quanto tempo cada colaborador gasta, em média, com a submissão das despesas. Para calcular esse tempo, contabilizamos o tempo de preencher a folha de despesas e o tempo de deslocação à sede para respetiva entrega.

Após um inquérito aos colaboradores, percebemos que cada colaborador gasta em média 30 minutos. Sendo que a maioria faz a submissão durante o horário de trabalho.

Com esta informação e tendo em conta outras despesas inerentes a este processo, estimamos o custo médio mensal, considerando 100 colaboradores que submetem as despesas, Tabela 3.

Tabela 3 – Custos estimados da empresa com o atual sistema de submissão de despesas.

	Valor ½ Hora por colaborador (médio)	Faturação ½ hora por cada trabalhador (médio)	Combustível gasto na deslocação à sede(25km) (médio)	Custo do envio das despesas por correio
Valor por Colaborador	2.5€	15€	2.5€	0.75€
Valor Total por despesa	250€	1500€	250€	18.75€ (25 colaboradores)
Valor Total	2018.75€			

2.2. OPINIÃO DOS COLABORADORES

Antes de iniciar o desenvolvimento da aplicação *Android*, decidiu-se realizar um inquérito aos colaboradores do IEP para entender quais as desvantagens do sistema atual e se um sistema 100% digital era vantajoso para os principais interessados no tema.

O inquérito, figura 2, foi realizado pela plataforma *Google Forms* e enviado via email para todos os colaboradores do Grupo IEP. Os mesmos tiveram 15 dias para responder ao inquérito, que não foi de caracter obrigatório.

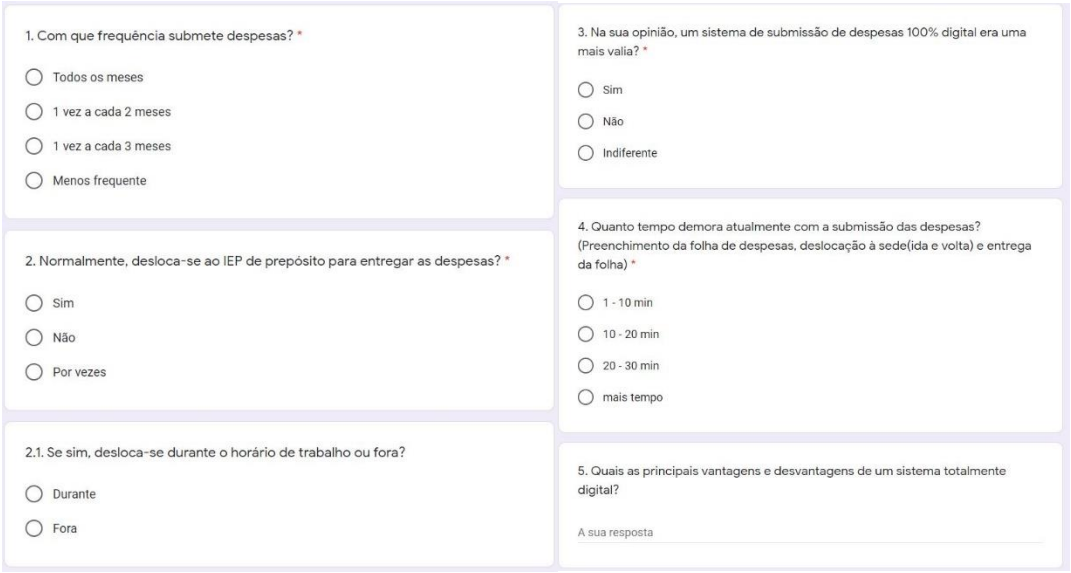


Figura 2 – Inquérito realizado via *Google Forms* aos colaboradores do IEP.

Tendo em conta que por volta de 100 colaboradores diferentes utilizam o sistema submissão de despesas, obteve-se 72 respostas, o que corresponde a 72%.

Para a pergunta "Com que frequência submete despesas", obtivemos o seguinte resultado, Figura 3:

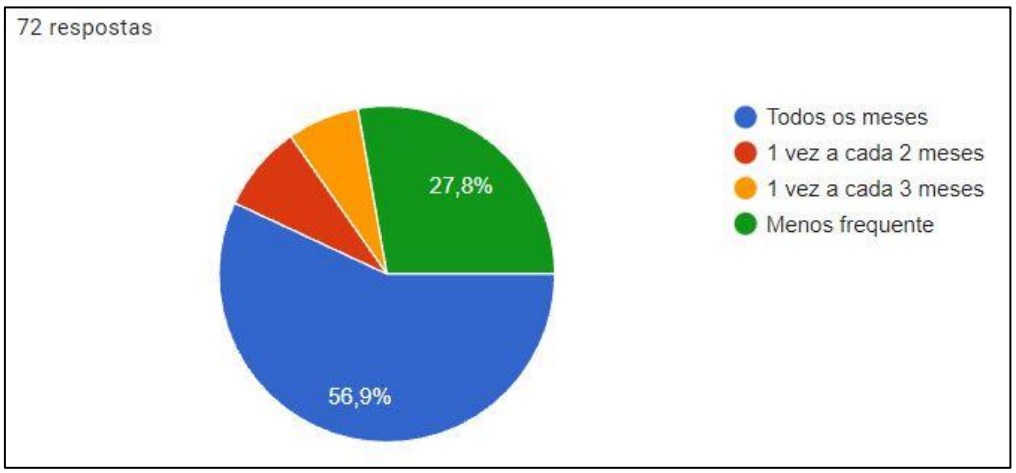


Figura 3 – Gráfico da frequência com que os colaboradores submetem as despesas.

Pelo gráfico, Figura 3, analisamos que mais de metade dos colaboradores, 56.9%, submete as despesas todos os meses, 8,3% submete uma vez a cada 2 meses, 6.9% submete uma vez a cada 3 meses e 27,8% submente de forma menos frequente.

Para a pergunta, “Normalmente, desloca-se ao IEP de propósito para entregar as despesas?”, obtivemos o seguinte resultado, Figura 4:

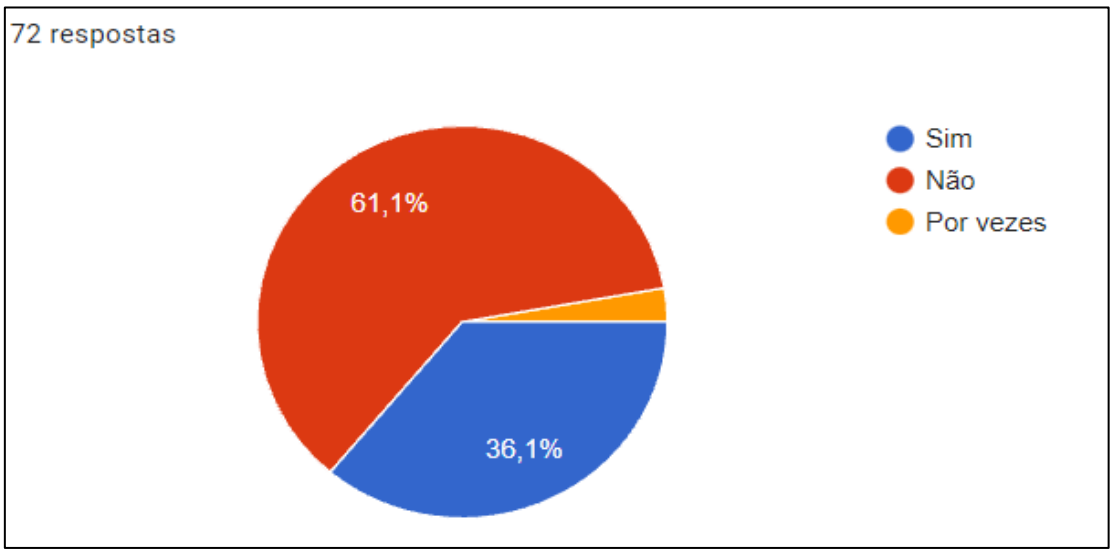


Figura 4 – Gráfico relativo à deslocação para entrega das despesas no IEP.

Pelo gráfico, Figura 4, analisamos que mais de metade dos colaboradores, 61.1%, não se desloca de propósito à empresa para entregar as despesas, 36.1% desloca-se de propósito e 2.8% por vezes desloca-se de propósito outra vezes não.

Para a pergunta, “Se te deslocas de propósito, deslocas-te durante o horário de trabalho ou fora?”, obtivemos o seguinte resultado, Figura 5:

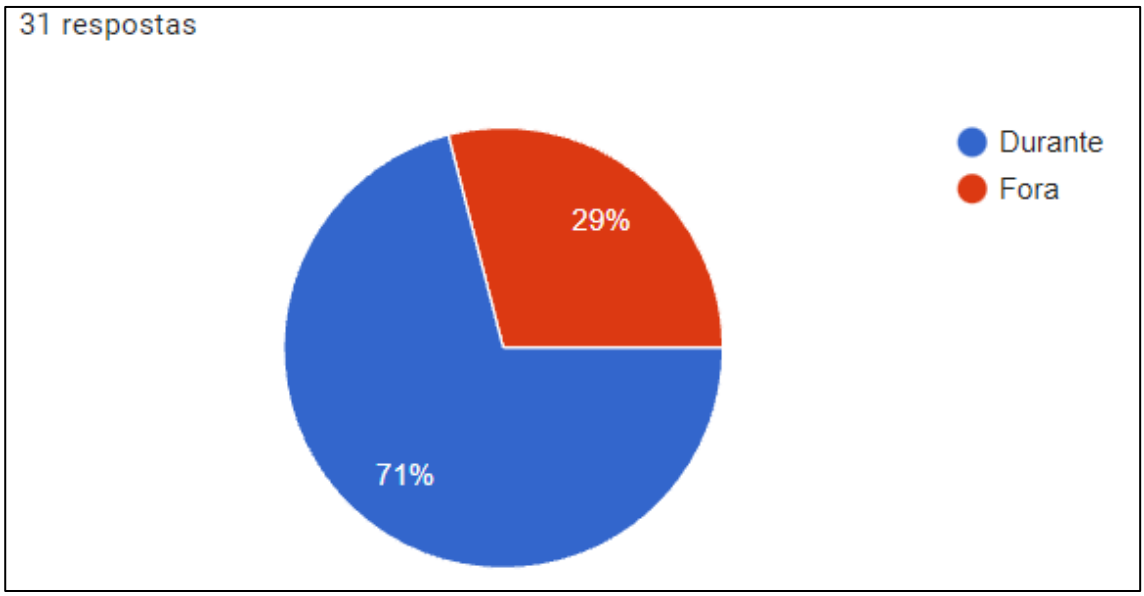


Figura 5 – Gráfico relativo à deslocação propositada para entrega de despesas na empresa.

Pelo gráfico, Figura5, analisamos que por volta de ¾ dos colaboradores, 71%, se descola durante o horário de trabalho e 29% desloca-se fora do horário de trabalho.

Para a pergunta, "Na sua opinião, um sistema de submissão de despesas 100% digital era uma mais-valia", obtivemos o seguinte resultado, Figura 6:

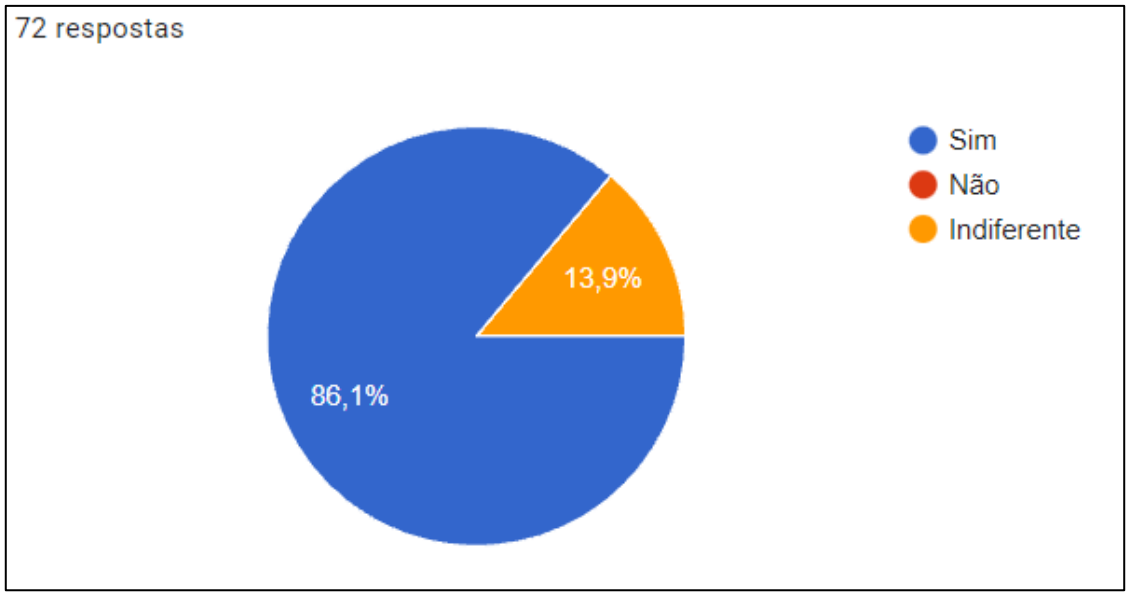


Figura 6 – Gráfico relativo à mais-valia de uma solução 100% digital.

Pelo gráfico, Figura 6, analisamos que mais de ¾ dos colaboradores, 86.1%, considera uma mais-valia e 13.9% acha indiferente haver um sistema novo.

Para a pergunta, "Quanto tempo demora atualmente com a submissão das despesas? (Preenchimento da folha de despesas, deslocação à sede (ida e volta) e entrega da folha)", obtivemos o seguinte resultado, Figura 7:

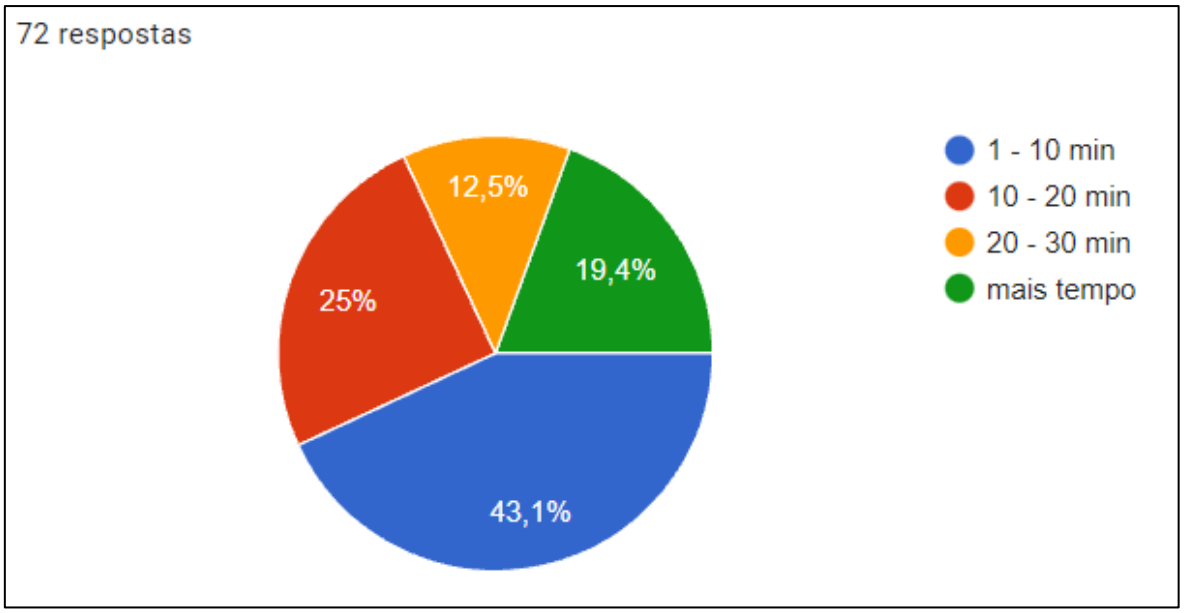


Figura 7 – Gráfico relativo ao tempo dispensado na entrega das despesas.

Pelo gráfico, Figura7, analisamos que mais de metade dos colaboradores, 56.9%, demora mais de 10 minutos, 43.1% demora entre 1 a 10 minutos, 25% demora entre 10 a 20 minutos, 12.5% demora entre 20 a 30 minutos e 19.4% demora mais de 30 minutos.

Para a pergunta de resposta livre, "Quais as principais vantagens e desvantagens de um sistema totalmente digital?", obteve-se 48 respostas diferentes, selecionou-se algumas respostas:

"Rapidez na entrega das despesas e simplificação do processo em termos administrativos."

"Não faz qualquer sentido manter o modelo atual com circulação de papéis para verificação e posterior validação. Acabar com papel (no próprio e no Departamento de Administração Financeiro (DAF)), simplificar o trabalho do DAF. Acelerar o processo de contabilização de despesas e reembolso."

"A meu ver, só traria vantagens, sobretudo a nível da logística de entrega das despesas."

"Evitar a deslocação à sede propositadamente para a entrega de despesas."

2.3. SOLUÇÃO A IMPLEMENTAR

Após análise do atual sistema de submissão de despesas, decidiu-se desenvolver uma aplicação para o Sistema *Android.* As principais funções da mesma consistem na recolha de todos os dados necessários, aquisição fotográfica das faturas, organização dos dados

numa pasta com um nome específico e irrepetível, e por fim, envio de um ficheiro comprimido (.zip) para uma *cloud* do IEP.

2.3.1. VANTAGENS E DESVANTAGENS

Após projetar a solução a implementar, e para compreender a utilidade da mesma analisaram-se algumas vantagens e desvantagens, como apresentado na Tabela 4.

Tabela 4 – Vantagens e desvantagens da solução a implementar.

Vantagens	Desvantagens
Eliminação do arquivo em papel	Ligação à internet para envio dos dados
Redução considerável dos custos devido à eliminação das deslocações dos colaboradores à sede, enviando as despesas remotamente	Necessidade de um smartphone para instalação da aplicação
Armazenamento de todos os dados de forma organizada	
Poupança ambiente	

Após analise da solução projetada, conclui-se que a mesma seria uma mais-valia para a empresa.

3. SISTEMA OPERATIVO *ANDROID*

Android é um sistema operativo baseado em *Linux*, a *Google* é atual detentora deste sistema. O mesmo é projetado principalmente para dispositivos móveis com *touch screen* [2].

Atualmente, é o sistema operativo móvel mais utilizado do mundo. Existem várias versões do sistema *Android*, a *Google*, disponibiliza constantemente, novas versões que corrigem pequenos erros e disponibilizam novas funcionalidades, a última versão disponibilizada é o *Android* 11 [3].

As aplicações, do sistema Android, utilizam a linguagem de programação *Java*. A ferramenta *Android Software development kit* (SDK) compila o código, gerando um *Android Application Pack* (APK). Estes arquivos contêm todo o conteúdo da aplicação, sendo estes utilizados para instalar as aplicações nos equipamentos com sistema *Android* [1].

3.1. *ACTIVITY*(ATIVIDADE)

Uma atividade é um componente do aplicativo que disponibiliza uma tela para o utilizador poder interagir. Cada atividade tem um *layout* associado que exibe uma interface ao utilizador.

Cada aplicação pode conter inúmeras atividades, sendo que cada atividade pode iniciar outra, com o objetivo de realizar determinadas ações, ao iniciar outra atividade a anterior é interrompida, mas o sistema conserva-a.

Para iniciar uma atividade pode-se utilizar o método *startActivaty()* passando uma *Intent*(este método será descrito posteriormente) que a identifica [3].

```
Intent intent = new Intent(this,
nomeActivaty.class);
StartActivity(nomeActivaty);
```

Caso se pretenda encerrar uma atividade deve-se utilizar o método *finish()*.

3.2. INTENT

A *Intent* é um objeto que tem como principal função solicitar uma ação de outro componente da aplicação.

Existem dois tipos de *Intents* [5]:

- Os *Intents* explícitos que especificam o componente a iniciar pelo nome, normalmente estes são utilizados para iniciar um componente no próprio aplicativo, visto que se sabe o nome explicito da classe da atividade ou serviço a iniciar;

- Os *Intents* implícitos que não especificam o componente a iniciar, mas sim uma ação geral a realizar, permite assim que um componente de outro aplicativo a processe.

A criação de um *Intent* consiste num objeto *Intent* que contém as informações necessárias para que o sistema determine o componente a iniciar [5].

3.3. OPÇÕES DE ARMAZENAMENTO

O sistema *Android* disponibiliza várias opções de armazenamento, como preferências compartilhadas e armazenamento comum [6].

As **preferências compartilhadas**, classe *SharedPreferences,* permite salvar e recuperar valores-chave. Por exemplo, caso o aplicativo contenha um campo para o utilizador inserir o seu nome e mais tarde, noutra atividade, seja necessário exibir esse nome, precisa-se de guardar esse campo na *SharedPreferences* para poder ser exibido.

O objeto *SharedPreferences* obtém-se utilizando-se um destes dois métodos [6]:

- *getSharedPreferences()* – caso seja necessário vários arquivos, identificando por nome cada um dos mesmos;
- *getPreferences()* – caso seja necessário somente um arquivo para a atividade.

Para gravar os valores deve-se [6]:

1) Utilizar o método *edit()* para obter um *SharedPreferences.Editor;*
2) Adicionar os valores com os métodos *putBoolean()* ou *putString();*
3) Confirmar os novos valores com o *commit().*

O **armazenamento** é permitido em todos os dispositivos com o sistema *Android*. Para guardar arquivos deve-se obter um *File* que representa o diretório adequado e utilizar o método *getExternalFilesDir()* fornecendo o tipo de diretório desejado, estes podem ser compartilhados com outros aplicativos, lidos e modificados pelo utilizador [6].

3.4. MANIFESTO DE UMA APLICAÇÃO

Todas as aplicações necessitam de um arquivo denominado *AndroidManifest.xml* no seu diretório de raiz. O arquivo inclui informações essências sobre a aplicação que transmite ao sistema *Android* [7].

O arquivo deve incluir, entre outras coisas [7]:

- O nome do pacote da aplicação, nome esse que identifica o aplicativo de forma exclusiva;
- Todas as permissões que o aplicativo necessita, como por exemplo caso necessite de interagir com o armazenamento externo;

- Declarar o nível mínimo da *Android Application Programming Interface* (API) exigido pela aplicação.

Por exemplo, para aplicação poder utilizar o armazenamento do dispositivo deve-se incluir a seguinte instrução no arquivo manifesto:

```
<uses-permission
android:name=”android.permission.WRITE_EXTERNAL_S
TORAGE”/>
```

3.5. ANDROID STUDIO

O *Android Studio* é a plataforma oficial para desenvolvimento de aplicações *Android*. Além do editor de código, o *Android Studio* disponibiliza inúmeros recursos para aumentar a produtividade do utilizador na criação de aplicações *Android*, como por exemplo:[8]

- Um sistema de compilação flexível baseado em *Gradle*;
- Um simulador rápido e com inúmeros recursos;
- Um ambiente único que possibilita o desenvolvimento para todos os dispositivos *Android*.

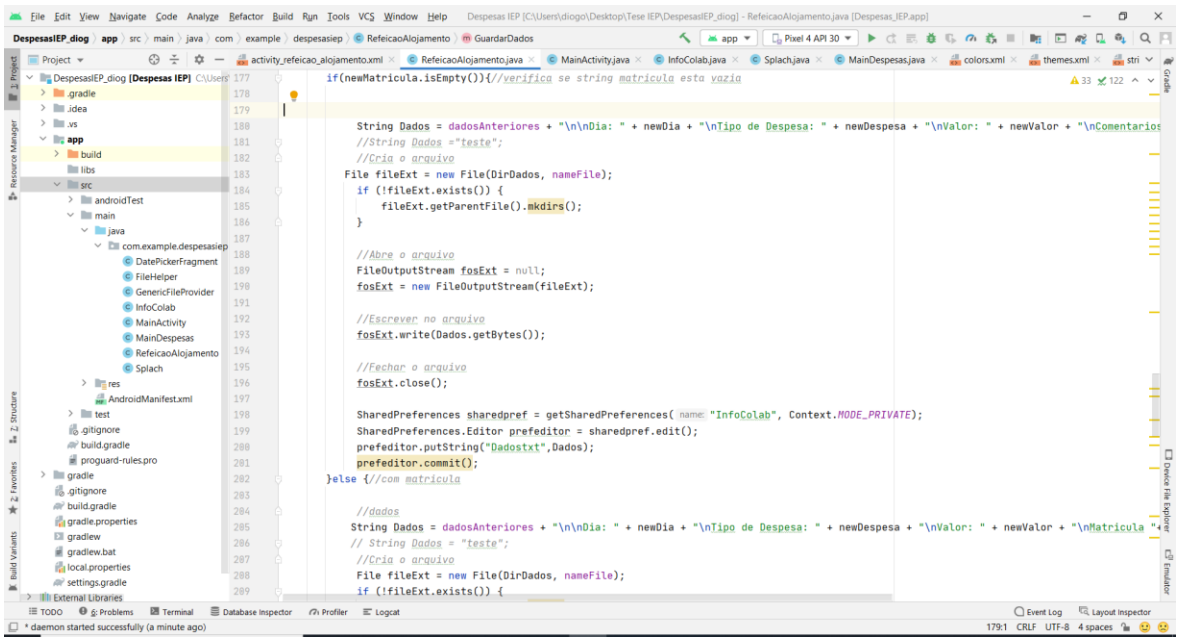


Figura 8 – Imagem exemplificativa do *Android Studio*.

4. SISTEMA DE SUBMISSÃO DE DESPESAS INFORMATIZADO PARA O SISTEMA *ANDROID*

Após se ter analisado, ao pormenor, todo o processo existente de submissão de despesas chegou-se à conclusão de desenvolver uma aplicação *Android*.

A aplicação *Android,* cujo nome será Despesas – IEP, foi desenvolvida recorrendo à plataforma *Android Studio* e terá como principais funções a recolha de todos os dados necessários, aquisição fotográfica das faturas, organização dos dados em uma pasta com um nome específico e irrepetível, e por fim, envio de um ficheiro comprimido (.zip) para uma *cloud* do IEP.

Para o seu desenvolvimento, necessitou-se de entender de forma clara todo o processo existente, projetar um novo sistema e, por fim, entender o sistema *Android*. Durante o seu desenvolvimento, houve inúmeras reuniões com toda a equipa de contabilidade e recursos

humanos para delinear todos os pormenores da aplicação. Além disso, a opinião dos colaboradores da empresa foi de extrema importância para todo o processo.

4.1. OBJETIVOS DA APLICAÇÃO

Antes de avançar para o desenvolvimento da aplicação necessitou-se de planear e definir detalhadamente a mesma. O sucesso da mesma depende de vários objetivos/pontos fundamentais, entre os quais:

- Aplicação fluída e de fácil utilização;
- Informação constante do que se esta a tratar;
- Acesso à camara do dispositivo;
- Possibilidade de ser instalada numa enorme variedade de dispositivos *Android;*
- Organização de todos os dados numa estrutura pré-definida;
- Envio dos dados remotamente e clicando apenas num botão.

4.2. FUNCIONAMENTO DA APLICAÇÃO

O funcionamento da aplicação é intuitivo e simplificado. Neste tópico serão apresentado vários *screenshots* para explicar de forma clara o seu funcionamento.

Para demonstração da aplicação irão ser utilizadas como exemplo as despesas de um colaborador no mês de outubro de 2021, onde teve 3 despesas, um jantar, um pequeno-almoço e um estacionamento.

Ao iniciar a aplicação, aparece um ecrã com a duas opções, iniciar uma nova folha de despesas ou obtermos explicações de como preencher as despesas na aplicação, como se pode ver Figura 9.

Figura 9 – *Layout* inicial da aplicação.

De seguida, ao clicar em "Nova Folha de Despesas", surge o menu de informações de colaborador, Figura 10. Este menu tem um conjunto de restrições que condicionam as decisões do utilizador, como por exemplo, só permitir letras no campo do nome, só permitir números no campo do número de trabalhador, ao clicar no campo "Mês-Ano" aparece um calendário para selecionar o mês e o ano, não permitindo outra configuração que não seja MM-AAAA. Relativamente ao campo "Centro de Custos", o utilizador tem um conjunto de opções pré-estabelecido pela empresa, Figura 11.

Figura 10 – Informação do colaborador.

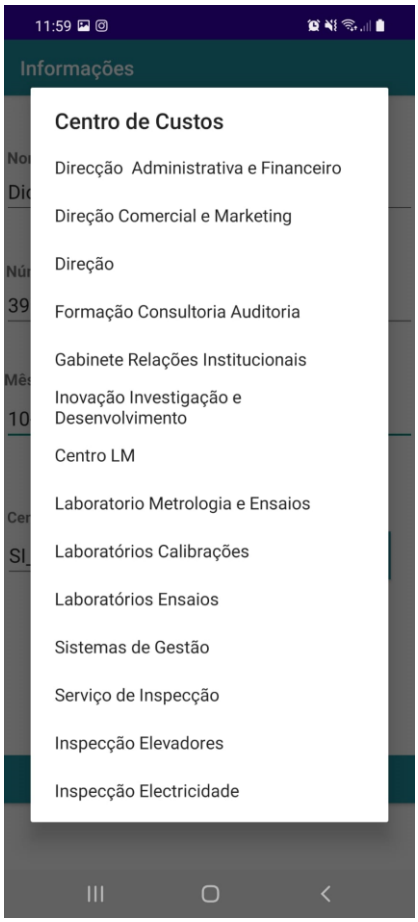


Figura 11 – Centro de custos.

De seguida, ao clicar em “Avançar”, surge o menu central, Figura 12. Este menu apresenta duas opções ao colaborador, inserir uma nova despesa ou enviar as despesas. A opção enviar despesas só é permitida quando o colaborador já preencheu todas as despesas do mês, o controle dessa condição é feito pelo *checkbox* “Já submeteu todas as despesas?”. Caso o utilizador clique no botão “Enviar despesas” por engano, aparece um aviso “Preencha por favor todas as despesas”, Figura 13.

Figura 12 – Menu central de despesas.

Figura 13 – Informação para preencher despesas.

De seguida, o utilizador deve clicar em “Nova Despesa”, ao fazê-lo surge o ecrã “Informações da despesa”, Figura 14. Este ecrã apresenta cinco campos de preenchimento, dia da despesa onde só permite a inserção de números, tipo de despesa onde o utilizador tem de escolher uma das opções disponível, Figura 15, valor da despesa onde só permite a inserção de números, matrícula da viatura, nas situações em que a despesa se refere à viatura, e por fim, um campo disponível para comentários adicionais.

Este ecrã, Figura 14, apresenta também um botão de seleção obrigatória, denominado “FOTO” onde o utilizador deve tirar uma fotografia à fatura em questão.

Após o preenchimento de todos os campos e respetiva fotografia anexada, o utilizador deve clicar no botão “SALVAR”. Ao clicar neste botão, é criada uma pasta com o nome tipo “Despesa_Nº de Funcionário_Nome do Colaborador_MM-AAAA_Centro de custos”, neste exemplo o nome da pasta criada é “Despesa_393_Diogo Gomes_10-2021_SI-IE” e posteriormente criado um ficheiro de texto (.txt) com todos os dados inseridos pelo utilizador com o mesmo nome tipo, as fotografias são guardadas com o formato (.jpg) e com o nome tipo *“*Despesa_Nº de Funcionário_DD-MM-AAAA_Tipo de Despesa*”,* neste exemplo o nome da fotografia é “Despesa_393_09-10-2021_Jantar” na mesma pasta. Para o utilizador a aplicação volta para o ecrã “Menu Central Despesas”, o utilizador deve clicar em “Nova Despesa” até preencher todas as despesas do mês em questão.

Figura 14 – Informações da despesa.

Figura 15 – Tipo de despesa.

De seguida, após terminar o preenchimento de todas as despesas, o utilizador deve proceder ao envio das despesas para a empresa, clicando no botão “Enviar Despesas”, sendo que tem de clicar na *chekbox* que se encontra nesse ecrã. Ao fazê-lo a aplicação vai compactar a pasta que foi criada, com a informação em documento de texto (.txt) e as fotografias das faturas, após o término da compactação o ficheiro (.zip) é enviado para a *cloud* do IEP. O utilizador recebe a informação que o ficheiro foi compactado com sucesso e enviado a partir de uma mensagem no ecrã, Figura 16. O utilizador não necessita de realizar mais nenhuma tarefa depois de enviar as despesas.

Figura 16 – Informação ao utilizador do envio da despesa com sucesso.

O utilizador caso pretenda ver o que foi criado, pode aceder ao diretório */Android/data/com.example.despesasiep/files* onde terá acesso as fotografias capturadas e ao ficheiro criado, como se pode ver na Figura 17.

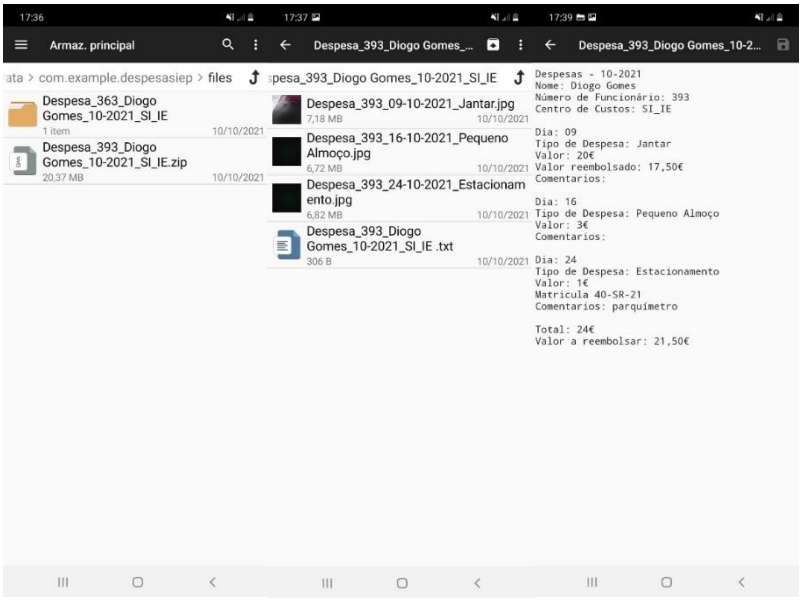


Figura 17 – Ficheiros criados pela aplicação.

4.3. TRATAMENTO DOS DADOS – EMPRESA

Após o dia 8 de cada mês (data-limite de submissão das despesas por parte dos colaboradores), o departamento financeiro necessita de tratar das despesas de cada colaborador. Esse tratamento consiste em validar os valores de cada despesa e verificar se os mesmos não ultrapassam os limites definidos pela empresa. Com o atual sistema a aplicação já faz uma pré verificação desses valores máximos, inserindo no ficheiro essa informação abaixo do valor indicado pelo colaborador.

O departamento financeiro recebe na *cloud* do IEP, cloud.iep.pt, Figura 18, o ficheiro comprimido (.zip) por parte do funcionário na pasta do mês correspondente, Figura 19. Este ficheiro compactado (.zip), com um nome irrepetível e indicativo do colaborador que enviou, contêm um ficheiro de texto (.txt), e as diferentes fotografias correspondentes a cada fatura, Figura 20.

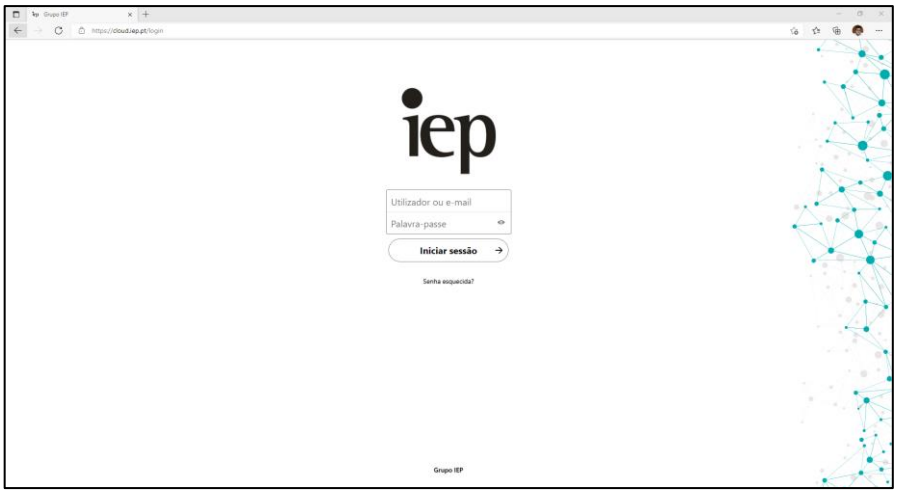


Figura 18 – Ecrã inicial da *cloud*.

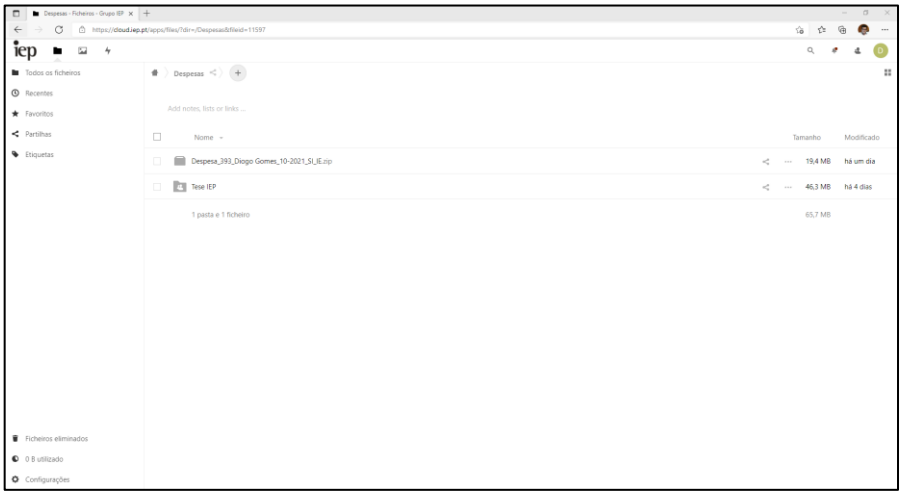


Figura 19 – Conta Despesas da *cloud*.

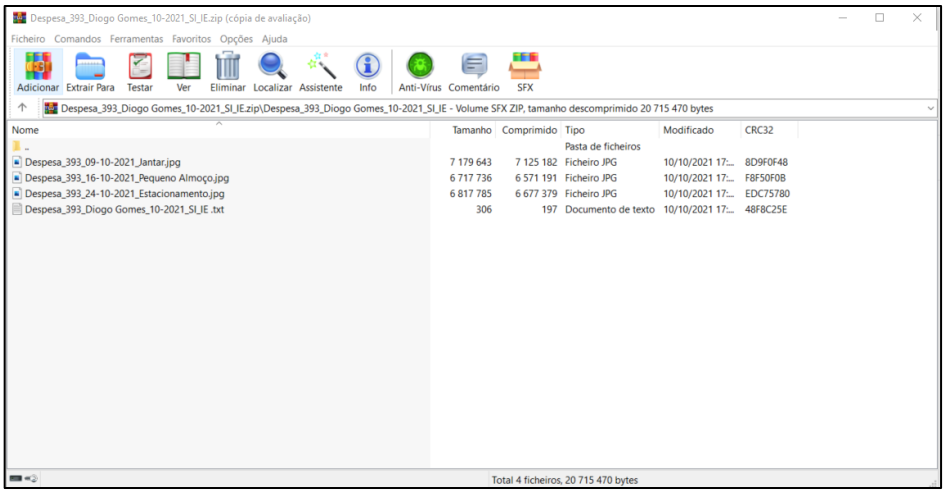


Figura 20 – Ficheiro enviado pelo colaborador.

Quando todas as despesas tiverem validadas, os responsáveis técnicos de cada colaborador dão a confirmação do pagamento das despesas aos colaboradores.

4.4. FUNÇÕES UTILIZADAS NA APLICAÇÃO

Como já foi referido a aplicação será exclusivamente desenvolvida para o sistema *Android*, dado que todos os equipamentos móveis utilizados pelo Grupo IEP utilizam esse sistema. Para o seu desenvolvimento utilizou-se o *Android Studio*, ferramenta disponibilizada pela *Google* para o desenvolvimento de aplicações *Android*.

Estas mesmas aplicações são divididas basicamente por classes *java*, que utiliza linguagem *java*, e *layouts* que utiliza linguagem *Extensible Markup Language (*xml). Cada atividade tem uma classe *java* e um *layout* associado, dado que a programação mais complexa se encontra nas classes *java* vai ser apresentado as principais funções criadas.

- **Dados inicias guardados em *shared preferences***

  A aplicação em várias funções utiliza os dados inicias inseridos pelo utilizador, o nome, o número do funcionário, o mês-ano e o centro de custos.

  Devido a isso, é necessário guardar esses valores para poderem ser utilizados a qualquer altura, para isso utiliza-se o método *SharedPreferences*, da seguinte forma:

```
SharedPreferences sharedpref =
getSharedPreferences("InfoColab",
Context.MODE_PRIVATE);
SharedPreferences.Editor prefeditor =
sharedpref.edit();

prefeditor.putString("nome",newNome); // guarda o
nome
prefeditor.putString("nfunc",newNfunc); // guarda
o número do funcionário
prefeditor.putString("mesref",newMesRef); //
guarda o mês e o ano em questão
prefeditor.putString("centrocustos",newCentroCust
os); // guarda o centro de custos selecionado
prefeditor.commit();
```

  Para utilizar estes dados, criou-se uma função “buscarDados” o mesmo método guardando cada dado numa nova *String*, da seguinte forma:

```
public void buscarDados (){
    SharedPreferences sharedpref =
getSharedPreferences("InfoColab",
Context.MODE_PRIVATE);

    nome = sharedpref.getString("nome","");
    nfunc = sharedpref.getString("nfunc","");
    mesref = sharedpref.getString("mesref","");
    centrocustos =
sharedpref.getString("centrocustos","");

}
```

- **Estrutura de armazenamento dos dados**

  O ficheiro de texto (.txt) e as fotografias capturadas são guardados numa pasta especifica consoante as informações iniciais que o utilizador introduz.

  Todos os dados são guardados no armazenamento interno do *smartphone* no diretório "/Android/data/com.example.despesasiep/files"

  Para armazenar os dados no diretório pretendido utiliza-se a seguinte função:

```
public File createFolder(){


String namePasta = "Despesa_" + newNfunc + "_" +
newNome + "_" + newMesRef + "_" +
newCentroCustos;

File file;
if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.Q) { // Android 10 - Q
    file = new File
(this.getExternalFilesDir(null), namePasta);
} else {
    file = new
File(Environment.getExternalStorageDirectory(),
namePasta);
}

if (!Dir.exists()) {
    Dir.mkdir();
}


    return Dir;
}
```

O ficheiro de texto (.txt) é criado da seguinte forma:

```
//Abre o arquivo
FileOutputStream fosExt = null;
fosExt = new FileOutputStream(fileExt);

//Escrever no arquivo
fosExt.write(Dados.getBytes()); // Dados - string
com a informação que pretendo guardar no arquivo

//Fechar o arquivo
fosExt.close();
```

- **Iniciar a câmara do dispositivo**

A aplicação necessita de aceder a câmara do dispositivo para o utilizador fotografar a fatura de cada despesa. Além disso consoante o que é realizado pelo utilizador na câmara a aplicação deve agir de forma diferente.

Para iniciar a camara deve ser utilizada a seguinte função:

```
public void startCamera(View view){

    buscarDados(); // dados para o nome da
fotografia

    //guardar numa string indicação por parte do
utilizador do tipo de despesa
    TextView TipoDespesas =
(TextView)findViewById(R.id.EditTipoDespesa);
    String newTipoDespesas =
TipoDespesas.getText().toString();

    //indica o diretorio para guardar imagem
    File DirDespesa = createFolder();
    // nome da despesa
    File image = new File(DirDespesa, "Despesa_"
+mesref + "_" + nfunc + "_" + newTipoDespesas +
".jpg");

    Uri photoURI =
FileProvider.getUriForFile(RefeicaoAlojamento.thi
s,
            BuildConfig.APPLICATION_ID +
".provider",
            image);

    Intent a = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);

    a.putExtra(MediaStore.EXTRA_OUTPUT,
photoURI);
    startActivityForResult(a,CODE_CA);
}
```

Consoante o sucesso ou não da ação fotográfica, o programa tem de realizar tarefas diferentes, caso o utilizador fotografe com sucesso a informação "Nº de fotos:" no ecrã deve ser atualizada e uma mensagem "Foto guardada", Figura 21, deve aparecer, caso contrário deve aparecer uma mensagem no ecrã "Foto NÃO guardada", Figura 22.

```
protected void onActivityResult(int requestCode,
int resultCode , Intent a) {

    super.onActivityResult(requestCode,
resultCode, a);
    if (requestCode == CODE_CA) {
        if (resultCode == RESULT_OK) {

  //incremento do n° de fotos
            ca++;
             Nfoto = "N° de Fotos:" +
Integer.toString(ca);

            TextView textview1;
            textview1 = (TextView)
findViewById(R.id.textView);
             textview1.setText(Nfoto);


Toast.makeText(RefeicaoAlojamento.this, "Foto
guardada", Toast.LENGTH_SHORT).show();
        }
        if (resultCode == RESULT_CANCELED) {

Toast.makeText(RefeicaoAlojamento.this, "Foto NÃO
guardada", Toast.LENGTH_LONG).show();
        }
    }
}
```

Figura 21 – Aviso foto guardada.

Figura 22 – Aviso foto não guardada.

- **Opções selecionáveis**

Durante a navegação pela aplicação, o utilizar depara-se por vezes com campos em que a sua escolha esta limitada. Por exemplo, não pode escrever o seu centro de custos ou o tipo de despesa. Deve sim selecionar uma das opções pré-definidas pela empresa.

No código da aplicação, essa função foi desenvolvida da seguinte forma:

```
public void startListaDespesas(View view) {

    buscarDados();

    AlertDialog.Builder builder = new
AlertDialog.Builder(this);

   // itemsDespesas é uma CharSequence[]
   builder.setItems(itemsDespesas, new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface
dialog, int opcao) {
            //consoante cada opção selecionada
coloca-a no ecrã
            if(opcao==0){
                EditText editText =
(EditText)findViewById(R.id.EditTipoDespesa);
                editText.setText("Almoço");
            }else
            if(opcao==1){
                EditText editText =
(EditText)findViewById(R.id.EditTipoDespesa);
                editText.setText("Jantar");
            }else
            if(opcao==2){
                EditText editText =
(EditText)findViewById(R.id.EditTipoDespesa);
                editText.setText("Pequeno
Almoço");
            }else
            if(opcao==3){
                EditText editText =
(EditText)findViewById(R.id.EditTipoDespesa);
                editText.setText("Hotel");
            }else
            if(opcao==4){
                EditText editText =
(EditText)findViewById(R.id.EditTipoDespesa);

editText.setText("Estacionamento");
            }else
            if(opcao==5){
                EditText editText =
(EditText)findViewById(R.id.EditTipoDespesa);
```

```
                editText.setText("Lavagem");
            }else
            if(opcao==6){
                EditText editText =
(EditText)findViewById(R.id.EditTipoDespesa);
                editText.setText("Portagem");
            }else
            if(opcao==7){
                EditText editText =
(EditText)findViewById(R.id.EditTipoDespesa);
                editText.setText("Combustivel");
            }else
            if(opcao==8){
                EditText editText =
(EditText)findViewById(R.id.EditTipoDespesa);
                editText.setText("Outros");
            }

        }
    });

    AlertDialog dialog = builder.create();
    dialog.show();
}
```

- **Comprimir uma pasta**

  Após o utilizador preencher todas as despesas deve enviar a pasta criada com os diferentes ficheiros, para isso é necessário comprimir a pasta antes do envio. O utilizador ao clicar no botão "Enviar Despesas", a aplicação antes de efetivamente enviar as despesas comprime a pasta da seguinte forma:

```
//diretório onde é guardado o ficheiro zip
zipPasta =
String.valueOf(this.getExternalFilesDir(null));
//nome da pasta
nameFile = zipPasta + "/Despesa_" + nfunc + "_" +
nome + "_" + mesref + "_" + centrocustos;

//nome do ficheiro zip
nomeZIP = "Despesa_" + nfunc + "_" + nome + "_" +
mesref + "_" + centrocustos + ".zip";

//informação relativa à checkBox
checkBox = (CheckBox)
findViewById(R.id.checkBox);

send = (Button) findViewById(R.id.send);
```

```
send.setOnClickListener(new
View.OnClickListener() {
    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    public void onClick(View v) {
        if (checkBox.isChecked()){// verifica se
checkBox está marcada
            //função zip
        if (FileHelper.zip(nameFile, zipPasta,
nomeZIP, checkBox.isChecked())) {
            Toast.makeText(MainDespesas.this,
"Ficheiro compactado com sucesso",
Toast.LENGTH_LONG).show();
        }
```

A função "zip" encontra-se numa classe diferente denominada "*FileHelper*".

```
public static boolean zip( String sourcePath,
String destinationPath, String
destinationFileName, Boolean includeParentFolder)
{
    new File(destinationPath ).mkdirs();
    FileOutputStream fileOutputStream ;
    ZipOutputStream zipOutputStream =  null;
    try{
        if (!destinationPath.endsWith("/"))
destinationPath+="/";
        String destination = destinationPath +
destinationFileName;
        File file = new File(destination);
        if (!file.exists()) file.createNewFile();

        fileOutputStream = new
FileOutputStream(file);
        zipOutputStream =  new
ZipOutputStream(new
BufferedOutputStream(fileOutputStream));

        if (includeParentFolder)
            parentPath=new
File(sourcePath).getParent() + "/";
        else
            parentPath=sourcePath;

        zipFile(zipOutputStream, sourcePath);

    }
    catch (IOException ioe){
        Log.d(TAG,ioe.getMessage());
        return false;
    }finally {
        if(zipOutputStream!=null)
            try {
                zipOutputStream.close();
            } catch(IOException e) {
```

```
            }
        }

        return true;

    }

    private static void zipFile(ZipOutputStream
    zipOutputStream, String sourcePath) throws
    IOException{

        java.io.File files = new
    java.io.File(sourcePath);
        java.io.File[] fileList = files.listFiles();

        String entryPath="";
        BufferedInputStream input;
        for (java.io.File file : fileList) {
            if (file.isDirectory()) {
                zipFile(zipOutputStream,
    file.getPath());
            } else {
                byte data[] = new byte[BUFFER_SIZE];
                FileInputStream fileInputStream = new
    FileInputStream(file.getPath());
                input = new
    BufferedInputStream(fileInputStream,
    BUFFER_SIZE);

    entryPath=file.getAbsolutePath().replace(
    parentPath,"");

                ZipEntry entry = new
    ZipEntry(entryPath);
                zipOutputStream.putNextEntry(entry);

                int count;
                while ((count = input.read(data, 0,
    BUFFER_SIZE)) != -1) {
                    zipOutputStream.write(data, 0,
    count);
                }
                input.close();
            }
        }
    }
```

- **Enviar ficheiro comprimido para *cloud* do IEP**

  Por fim, após todas as despesas preenchidas e ficheiro comprimido a aplicação envia para a *cloud* do IEP o ficheiro criado.

```
//conect com o sevidor, password
String credential =
Base64.getEncoder().encodeToString((password.getB
ytes());
OkHttpClient client = new OkHttpClient();

File fileExt = new File(zipPasta, nomeZIP);
if (!fileExt.exists()) {
    fileExt.getParentFile().mkdirs();
}

RequestBody body = new
MultipartBody.Builder().setType(MultipartBody.FOR
M)
        .addFormDataPart("file",
fileExt.getName(),

//tipo de ficheiro
RequestBody.create(MediaType.parse("text/zip"),
fileExt))
        .addFormDataPart("some-field", "some-
value")
        .build();

Request request = new Request.Builder()

.url("https://cloud.iep.pt/remote.php/dav/files/d
espesas/Despesas/" + nomeZIP)
        .put(body)
        .addHeader("Authorization", "Basic " +
credential)
        .build();

try {
    Response response =
client.newCall(request).execute();
    Log.e("NEXTCLOUD",
Objects.requireNonNull(response.body()).string())
;
} catch (IOException e) {
    e.printStackTrace();
}

    Toast.makeText(MainDespesas.this, "Ficheiro
enviado com sucesso", Toast.LENGTH_LONG).show();
```

- ***AndroidManifest.xml* da aplicação**

  Durante o desenvolvimento da aplicação foi necessário acrescentar diferentes permissões à aplicação. Para isso, indica-se no ficheiro *AndroidManifest.xml* as mesmas. As permissões utilizadas são:

```
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_S
TORAGE" />

<uses-permission
android:name="android.permission.CAMERA" />

<uses-permission
android:name="android.permission.INTERNET" />

<uses-permission
android:name="android.permission.ACCESS_NETWORK_S
TATE" />
```

A aplicação ocupa aproximadamente 13 *megabytes* no *smartphone* do utilizador.

4.5. FLUXOGRAMA DA APLICAÇÃO ANDROID

Para melhor compreensão da aplicação desenvolveu-se um fluxograma, Figura 23, na plataforma *https://app.diagrams.net/*.

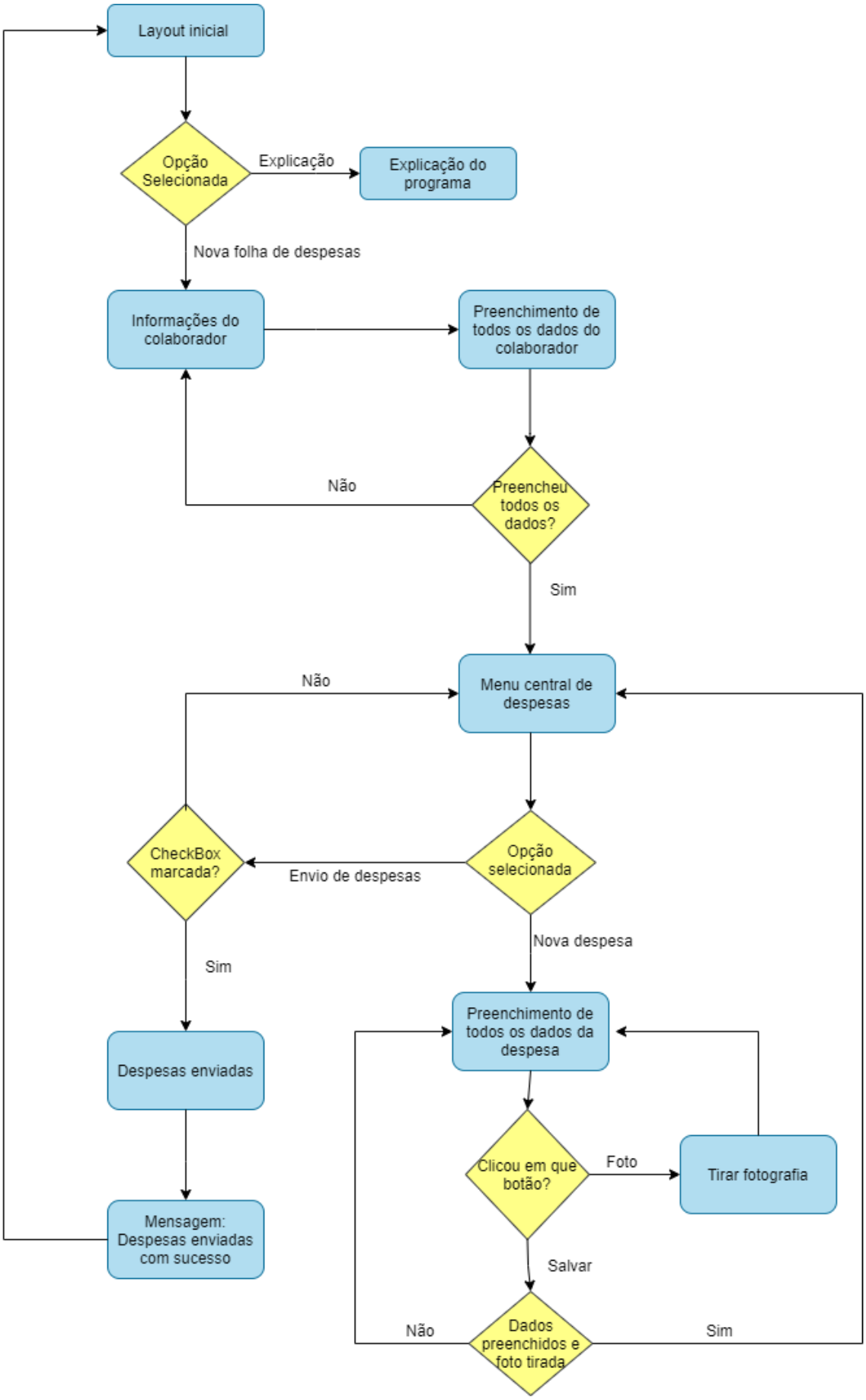


Figura 23 – Fluxograma da aplicação *Android*.

4.6. DIFICULDADES ENCONTRADAS NO DESENVOLVIMENTO DA APLICAÇÃO

Durante o desenvolvimento da aplicação para o sistema *Android*, existiram enumeras dificuldades sendo que as mais difíceis de serem ultrapassadas foram o armazenamento de dados no smartphone e a conexão com a *cloud* do IEP.

- **Armazenamento de dados no smartphone:**

  Nas primeiras versões do sistema *Android*, o acesso a arquivos era semelhante à forma de acesso num computador pessoal. Tendo a permissão de leitura ou escrita, era possível aceder a todos os arquivos em áreas de armazenamento comuns. Esses arquivos poderiam conter informações sensíveis. Além de obter informações pessoais, as aplicações podiam armazenar dados específicos em qualquer lugar na área comum de armazenamento, desta forma quando o utilizador desinstalasse uma aplicação a mesma deixava diversos dados para trás [9].

  A forma de armazenar dados mudou a partir do *Android* 10, inclusive, sendo que a partir desta versão, os dados criados pelas aplicações, são criados em pastas especificas para o efeito. Este facto dificultou o desenvolvimento da função de armazenamento de dados, pois toda a informação pesquisada utilizava código para versões anteriores.

  Nas versões anteriores ao *Android* 10 para criar algum ficheiro e guardá-lo no armazenamento interno o código a utilizar é:

```
file Dir = new
File(Environment.getExternalStorageDirectory(),
namePasta);
```

  Quando foram realizados os testes da aplicação, deparou-se que a mesma funcionava em versões inferiores ao *Android* 10 e em versões superiores ou iguais a aplicação dava erro fechando a aplicação. Isto porque nas versões *Android* 10 e posteriores o código a utilizar é:

```
File Dir = new
File(this.getExternalFilesDir(null), namePasta);
```

  Para garantir que a aplicação funcionava nas diferentes versões dos *smartphones* dos colaboradores (os *smartphones* do IEP têm sistema *Android* desde a versão 7

até a 11) fez-se uma condição "*if*", em que verificava a versão do sistema e, consoante a mesma, utilizava um método ou outro.

```
File file;
if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.Q) { // Android 10 - Q
    file = new File
(this.getExternalFilesDir(null), namePasta);
} else {
    file = new
File(Environment.getExternalStorageDirectory(),
namePasta);
}
```

Desta forma solucionou-se um problema encontrado durante o desenvolvimento da aplicação.

- **Envio de dados para a *cloud* do IEP**

A *cloud* do IEP, cloud.iep.pt, é utilizada para a receção dos dados enviados pela aplicação desenvolvida. A mesma é desenvolvida com base na *NextCloud*, que é uma plataforma de software cliente-servidor para criar e usar serviços de hospedagem de arquivos. É um software gratuito e de código aberto, onde qualquer empresa pode instalá-lo e operá-lo nos seus próprios servidores [10].

A plataforma *NextCloud* tem uma biblioteca própria para desenvolvedores de aplicações para o sistema *Android* interligarem as suas aplicações com a *cloud*.

Ao tentar utilizar essa biblioteca, percecionou-se a falta de documentação/informação sobre a mesma e a sua utilização, por parte dos desenvolvedores da mesma, e a sua utilização tornou-se difícil. Após inúmeras tentativas, e alguma pesquisa, achou-se por bem utilizar outra biblioteca, a *OkHttp3*, biblioteca assente em um cliente HTTP, que permitiu a conexão ao servidor e posterior envio de ficheiro de forma mais fácil e pratica.

5. CONCLUSÕES

Este projeto consistiu no desenvolvimento de uma aplicação *Android*, com o objetivo de reformular o sistema de submissão de despesas dos colaboradores do IEP, e assim, obter um sistema totalmente digital. Para esse efeito, houve a necessidade de estudar o sistema atual de submissão de despesas, entender os seus problemas e perceber como o processo podia ser melhorado.

O sistema atual de submissão de despesas consiste no preenchimento de uma folha de despesas, ao fim de cada mês, com os valores de cada despesa e o anexo da respetiva fatura. Este sistema, além de arcaico, trazia encargos financeiros à empresa consideráveis pois, obrigava deslocações dos trabalhadores à sede da empresa durante o horário de trabalho, para entrega das despesas. Por isso, o desenvolvimento de uma aplicação permite eliminar as deslocações dos trabalhadores à sede da empresa propositadas para submissão das despesas, melhorar o processo de validação por parte dos recursos humanos devido aos dados devidamente estruturados e eliminar o arquivo em papel.

Após o término do desenvolvimento da aplicação observa-se que a mesma cumpre todos os objetivos inicialmente propostos e, por isso, o IEP pretende que esta passe a ser utilizada por todos os seus colaboradores.

Ainda assim, após vários testes feitos à aplicação é possível identificar algumas melhorias que podem ser implementadas. Uma das propostas é a utilização de uma base de dados em vez de se guardar todos os dados em ficheiros de texto (.txt). Outra melhoria com utilidade, seria uma ligação inicial ao servidor onde o colaborador realizava um *login* e tinha acesso as suas despesas anteriores, bem como o saldo da sua conta corrente.

Em conclusão, o desenvolvimento deste trabalho traz benefícios relevantes para a empresa, pois otimiza um sistema que necessitava de um melhoramento considerável e contribui para a poupança de recursos financeiros todos os meses. Já para o aluno tratou-se também de uma mais-valia pois permitiu a aprendizagem de uma nova linguagem de programação, uma nova plataforma de trabalho, bem como aquisição de conhecimentos sobre um processo de gestão financeira de uma empresa e toda a sua complexidade, envolvendo um universo de 200 colaboradores.

Referências Documentais

[1] **IEP - Quem Somos.** [Online] [Consultado: 24 de setembro de 2021.] https://www.iep.pt/quem-somos/

[2] **Android - What is Android.** [Online] [Consultado: 20 de setembro de 2021.] https://www.android.com/what-is-android

[3] **Android - Android 11**. [Online] [Consultado: 27 de setembro de 2021.] https://www.android.com/android-11/

[4] **Android – Activities.** [Online] [Consultado: 12 de setembro de 2021.] https://developer.android.com/guide/components/activities.html.].

[5] **Android - Intents.** [Online] [Consultado: 15 de setembro de 2021.] https://developer.android.com/guide/components/intents-filters

[6] **Android - Data Storage.** [Online] [Consultado: 30 de setembro de 2021.] https://developer.android.com/training/data-storage

[7] **Android - App Manifest.** [Online] [Consultado: 29 de setembro de 2021.] https://developer.android.com/guide/topics/manifest/manifest-intro

[8] **Android - Android Studio.** [Online] [Consultado: 29 de setembro de 2021.] https://developer.android.com/studio

[9] **Android – Armazenamento de dados.** [Online] [Consultado: 26 de outubro de 2021] https://developer.android.com/training/data-storage#scoped-storage

[10] **NextCloud – About us**. [Online] [Consultado: 25 de outubro de 2021] https://nextcloud.com/about/

Anexo A. Código de algumas classes criadas

Classe: MainDespesas.java

```
package com.example.despesasiep;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;

import android.os.Build;
import android.os.Bundle;

import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;


import java.io.File;

import java.io.IOException;

import java.util.Base64;
import java.util.Objects;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;


public class MainDespesas extends
AppCompatActivity {

    String nome;
    String nfunc;
    String mesref;
    String centrocustos;
    String dadosAnteriores;
```

```
    String namePasta;
    String nameFileInic;
    String nameFile;
    String zipPasta;
    String nomeZIP;
    Button send;
    CheckBox checkBox;



    final static String TAG =
MainDespesas.class.getName();

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main_despesas);

        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().bui
ld();
        StrictMode.setThreadPolicy(policy);

        androidx.appcompat.app.ActionBar bar =
getSupportActionBar();
        bar.setTitle(("Menu Central Despesas"));
        bar.setBackgroundDrawable(new
ColorDrawable(Color.parseColor("#1D8E9E")));


        SharedPreferences sharedpref =
getSharedPreferences("InfoColab",
Context.MODE_PRIVATE);

        nome = sharedpref.getString("nome","");
        nfunc = sharedpref.getString("nfunc","");
        mesref =
sharedpref.getString("mesref","");
        centrocustos =
sharedpref.getString("centrocustos","");
        dadosAnteriores =
sharedpref.getString("Dadostxt","");
        nameFileInic =
sharedpref.getString("nameFile","");
        namePasta =
sharedpref.getString("namePasta","");

        //diretório onde é guadado o ficheiro zip
        zipPasta =
String.valueOf(this.getExternalFilesDir(null));
        //nome da pasta
        nameFile = zipPasta + "/Despesa_" + nfunc
+ "_" + nome + "_" + mesref + "_" + centrocustos;
        //npme do ficheiro zip
```

```
        nomeZIP = "Despesa_" + nfunc + "_" + nome
+ "_" + mesref + "_" + centrocustos + ".zip";

        //informação relaticaà checkBox
        checkBox = (CheckBox)
findViewById(R.id.checkBox);

        send = (Button) findViewById(R.id.send);
        send.setOnClickListener(new
View.OnClickListener() {
            @RequiresApi(api =
Build.VERSION_CODES.O)
            @Override
            public void onClick(View v) {
                if (checkBox.isChecked()){//
verifica se checkBox está marcada
                    //função zip
                if (FileHelper.zip(nameFile,
zipPasta, nomeZIP, checkBox.isChecked())) {

Toast.makeText(MainDespesas.this, "Ficheiro
compactado com sucesso",
Toast.LENGTH_LONG).show();
                }

                //conect com o sevidor, password
                String credential =
Base64.getEncoder().encodeToString(("despesas:l4$
B%n%$#MGq#IN7%%8ª~gZ8").getBytes());
                OkHttpClient client = new
OkHttpClient();

                File fileExt = new File(zipPasta,
nomeZIP);
                if (!fileExt.exists()) {

fileExt.getParentFile().mkdirs();
                }

                RequestBody body = new
MultipartBody.Builder().setType(MultipartBody.FOR
M)
                        .addFormDataPart("file",
fileExt.getName(),
                                //tipo de
fjichero

RequestBody.create(MediaType.parse("text/zip"),
fileExt))
                        .addFormDataPart("some-
field", "some-value")
                        .build();

                Request request = new
Request.Builder()
```

```
.url("https://cloud.iep.pt/remote.php/dav/files/despesas/Despesas/" + nomeZIP)
                        .put(body)

.addHeader("Authorization", "Basic " + credential)
                        .build();

                try {
                    Response response = client.newCall(request).execute();
                    Log.e("NEXTCLOUD", Objects.requireNonNull(response.body()).string());
                } catch (IOException e) {
                    e.printStackTrace();
                }


Toast.makeText(MainDespesas.this, "Ficheiro enviado com sucesso", Toast.LENGTH_LONG).show();


                }else{

Toast.makeText(MainDespesas.this, "Preencha por favor todas as despesas",
Toast.LENGTH_LONG).show();

                }
            }
        });


    }

    public void startTipoDespesa(View view) {
        Intent RefeicaoAlojamento = new Intent(this, RefeicaoAlojamento.class);
        startActivity(RefeicaoAlojamento);
    }
}
```

Classe: InfoColab.java

```
package com.example.despesasiep;

import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.app.DatePickerDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;

import android.os.Bundle;
import android.view.View;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.TextView;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Calendar;


public class InfoColab extends AppCompatActivity
{

    final CharSequence[] items = {"Direcção
Administrativa e Financeiro", "Direção Comercial
e Marketing", "Direção","Formação Consultoria
Auditoria","Gabinete Relações
Institucionais","Inovação Investigação e
Desenvolvimento","Centro LM","Laboratorio
Metrologia e Ensaios","Laboratórios Calibrações",
"Laboratórios Ensaios","Sistemas de
Gestão","Serviço de Inspecção","Inspecção
Elevadores","Inspecção Electricidade"};

    //Opçoes de centro de custos

    EditText EditMesRef;
    DatePickerDialog datePickerDialog;


    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_info_colab);
        androidx.appcompat.app.ActionBar bar =
getSupportActionBar();
        bar.setTitle(("Informações"));
```

```
        bar.setBackgroundDrawable(new
ColorDrawable(Color.parseColor("#1D8E9E")));

        // iniciar data
        EditMesRef = (EditText)
findViewById(R.id.EditMesRef);

        EditMesRef.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
               // classe calendário
                final Calendar c =
Calendar.getInstance();
                int mYear = c.get(Calendar.YEAR);
// ano
                int mMonth =
c.get(Calendar.MONTH); // mês
                int mDay =
c.get(Calendar.DAY_OF_MONTH); // dia
                // date picker dialog
                //
                datePickerDialog = new
DatePickerDialog(InfoColab.this,
                        new
DatePickerDialog.OnDateSetListener() {
                            @Override
                            public void
onDateSet(DatePicker view, int year,

int monthOfYear, int dayOfMonth) {

                                if(monthOfYear<9)
{

EditMesRef.setText("0" + (monthOfYear + 1) + "-"
+ year);
                                }else{

EditMesRef.setText((monthOfYear + 1) + "-" +
year);

                                }
                            }
                        }, mYear, mMonth, mDay);

                datePickerDialog.show();

            }
        });

    }
```

```
    public void startLista(View view) {

        AlertDialog.Builder builder = new
AlertDialog.Builder(this);

        builder.setTitle("Centro de Custos");


        builder.setItems(items, new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface
dialog, int opcao) {
                        //consoante cada opção
selecionada coloca-a no ecrã
                        if(opcao==0){
                            EditText editText =
(EditText)findViewById(R.id.EditCentroCustos);

editText.setText("DAF");
                        }else
                            if(opcao==1){
                                EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("DCM");
                            }
                            else
                            if(opcao==2){
                                EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("DIR");
                            }else
                            if(opcao==3){
                                EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("FCA");
                            }else
                            if(opcao==4){
                                EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("GRI");
                            }else
                            if(opcao==5){
                                EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("IID");
                            }else
                            if(opcao==6){
                                EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("LM");
```

```
                                }else
                                if(opcao==7){
                                    EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("LME");
                                }else
                                if(opcao==8){
                                    EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("LME_CL");
                                }else
                                if(opcao==9){
                                    EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("LME_EN");
                                }else
                                if(opcao==10){
                                    EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("SG");
                                }else
                                if(opcao==11){
                                    EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("SI");
                                }else
                                if(opcao==12){
                                    EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("SI_EV");
                                }else
                                if(opcao==13){
                                    EditText editText
= (EditText)findViewById(R.id.EditCentroCustos);

editText.setText("SI_IE");
                                }
                    }
            });

            AlertDialog dialog = builder.create();
            dialog.show();
        }



        //começar ecrã das Informações do colaborador
        public void startMainDespesas(View view)
throws IOException {
```

```
        TextView Nome =
(TextView)findViewById(R.id.EditNome);
        String newNome =
Nome.getText().toString();

        TextView Nfunc =
(TextView)findViewById(R.id.EditNFunc);
        String newNfunc =
Nfunc.getText().toString();

        TextView MesRef =
(TextView)findViewById(R.id.EditMesRef);
        String newMesRef =
MesRef.getText().toString();

        TextView CentroCustos =
(TextView)findViewById(R.id.EditCentroCustos);
        String newCentroCustos =
CentroCustos.getText().toString();

        String namePasta = "Despesa_" + newNfunc
+ "_" + newNome + "_" + newMesRef + "_" +
newCentroCustos;

        File Dir = new
File(this.getExternalFilesDir(null), namePasta);

        if (!Dir.exists()) {
            Dir.mkdir();
        }

        String nameFile = "Despesa_" + newNfunc +
"_" + newNome + "_" + newMesRef + "_" +
newCentroCustos + " .txt";

        String Dados = "Despesas - " + newMesRef
+ "\nNome: " + newNome + "\nNúmero de
Funcionário: " + newNfunc + "\nCentro de Custos:
" + newCentroCustos;

        //Cria o arquivo
        File fileExt = new File(Dir, nameFile);
        if (!fileExt.exists()) {
            fileExt.getParentFile().mkdirs();
        }

        //Abre o arquivo
        FileOutputStream fosExt = null;
        fosExt = new FileOutputStream(fileExt);

        //Escrever no arquivo
        fosExt.write(Dados.getBytes()); // Dados
- string com a informação que pretendo guardar no
arquivo
```

```
        //Fechar o arquivo
        fosExt.close();

        SharedPreferences sharedpref =
getSharedPreferences("InfoColab",
Context.MODE_PRIVATE);
        SharedPreferences.Editor prefeditor =
sharedpref.edit();

        prefeditor.putString("nome",newNome);
        prefeditor.putString("nfunc",newNfunc);
        prefeditor.putString("mesref",newMesRef);

prefeditor.putString("centrocustos",newCentroCust
os);
        prefeditor.putString("Dadostxt",Dados);

prefeditor.putString("namePasta",namePasta);

prefeditor.putString("nameFile",nameFile);
        prefeditor.commit();


        Intent MainDespesas = new Intent(this,
MainDespesas.class);
        startActivity(MainDespesas);
    }

}
```

Classe: RefeicaoAlojamento.java

```
package com.example.despesasiep;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.FileProvider;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;


import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

public class RefeicaoAlojamento extends
AppCompatActivity {


    final CharSequence[] itemsDespesas =
{"Almoço", "Jantar", "Pequeno Almoço", "Hotel",
"Estacionamento",
"Lavagem","Portagem","Combustivel", "Outros"};
    //dados  shared preferences
    String nome;
    String nfunc;
    String mesref;
    String centrocustos;
    String dadosAnteriores;
    String namePasta;
    String nameFile;



    //variaveis para as fotos
    String Nfoto;
    static  final int CODE_CA = 1;
    int ca =0;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_refeicao_alojamento);

        androidx.appcompat.app.ActionBar bar = getSupportActionBar();
        bar.setTitle(("Informações da despesa"));

        bar.setBackgroundDrawable(new ColorDrawable(Color.parseColor("#1D8E9E")));
    }

    public void startListaDespesas(View view) {

        buscarDados();

        AlertDialog.Builder builder = new AlertDialog.Builder(this);

       // itemsDespesas é uma CharSequence[]
       builder.setItems(itemsDespesas, new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int opcao) {
                //consoante cada opção selecionada coloca-a no ecrã
                if(opcao==0){
                    EditText editText = (EditText)findViewById(R.id.EditTipoDespesa);
                    editText.setText("Almoço");
                }else
                if(opcao==1){
                    EditText editText = (EditText)findViewById(R.id.EditTipoDespesa);
                    editText.setText("Jantar");
                }else
                if(opcao==2){
                    EditText editText = (EditText)findViewById(R.id.EditTipoDespesa);
                    editText.setText("Pequeno Almoço");
                }else
                if(opcao==3){
                    EditText editText = (EditText)findViewById(R.id.EditTipoDespesa);
                    editText.setText("Hotel");
                }else
                if(opcao==4){
                    EditText editText = (EditText)findViewById(R.id.EditTipoDespesa);

editText.setText("Estacionamento");
                }else
                if(opcao==5){
                    EditText editText =
```

```
(EditText)findViewById(R.id.EditTipoDespesa);
                    editText.setText("Lavagem");
                }else
                if(opcao==6){
                    EditText editText =
(EditText)findViewById(R.id.EditTipoDespesa);
                    editText.setText("Portagem");
                }else
                if(opcao==7){
                    EditText editText =
(EditText)findViewById(R.id.EditTipoDespesa);

editText.setText("Combustivel");
                }else
                if(opcao==8){
                    EditText editText =
(EditText)findViewById(R.id.EditTipoDespesa);
                    editText.setText("Outros");
                }

            }
        });

        AlertDialog dialog = builder.create();
        dialog.show();
    }

    public void buscarDados (){
        SharedPreferences sharedpref =
getSharedPreferences("InfoColab",
Context.MODE_PRIVATE);

        nome = sharedpref.getString("nome","");
        nfunc = sharedpref.getString("nfunc","");
        mesref =
sharedpref.getString("mesref","");
        centrocustos =
sharedpref.getString("centrocustos","");
        dadosAnteriores =
sharedpref.getString("Dadostxt","");
        nameFile =
sharedpref.getString("nameFile","");
        namePasta =
sharedpref.getString("namePasta","");

    }

    public File createFolder(){

            File file;
if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.Q) { // Android 10 - Q
    file = new File
(this.getExternalFilesDir(null), namePasta);
} else {
```

```
    file = new
File(Environment.getExternalStorageDirectory(),
namePasta);
}
            if (!Dir.exists()) {
                Dir.mkdir();
            }

            return Dir;

    }

    public void GuardarDados() throws IOException
{
        buscarDados();

        TextView Dia =
(TextView)findViewById(R.id.EditDia);
        String newDia = Dia.getText().toString();

        TextView Despesa =
(TextView)findViewById(R.id.EditTipoDespesa);
        String newDespesa =
Despesa.getText().toString();

        TextView Valor =
(TextView)findViewById(R.id.editTextValor);
        String newValor =
Valor.getText().toString();

        TextView Matricula =
(TextView)findViewById(R.id.editMatricula);
        String newMatricula =
Matricula.getText().toString();

        TextView Comentario =
(TextView)findViewById(R.id.editTextComentario);
        String newComentario =
Comentario.getText().toString();

        //cria pasta
        File DirDados = createFolder();

        //sem matricula
       if(newMatricula.isEmpty()){//verifica se
string matricula esta vazia


            String Dados = dadosAnteriores +
"\n\nDia: " + newDia + "\nTipo de Despesa: " +
newDespesa + "\nValor: " + newValor +
"\nComentarios: " + newComentario;
            //String Dados ="teste";
            //Cria o arquivo
          File fileExt = new File(DirDados,
nameFile);
```

```
            if (!fileExt.exists()) {
                fileExt.getParentFile().mkdirs();
            }

            //Abre o arquivo
            FileOutputStream fosExt = null;
            fosExt = new
FileOutputStream(fileExt);

            //Escrever no arquivo
            fosExt.write(Dados.getBytes());

            //Fechar o arquivo
            fosExt.close();

            SharedPreferences sharedpref =
getSharedPreferences("InfoColab",
Context.MODE_PRIVATE);
            SharedPreferences.Editor prefeditor =
sharedpref.edit();

prefeditor.putString("Dadostxt",Dados);
            prefeditor.commit();
       }else {//com matricula

            //dados
           String Dados = dadosAnteriores +
"\n\nDia: " + newDia + "\nTipo de Despesa: " +
newDespesa + "\nValor: " + newValor +
"\nMatricula "+newMatricula+ "\nComentarios: " +
newComentario;
           // String Dados = "teste";
            //Cria o arquivo
            File fileExt = new File(DirDados,
nameFile);
            if (!fileExt.exists()) {
                fileExt.getParentFile().mkdirs();
            }

            //Abre o arquivo
            FileOutputStream fosExt = null;
            fosExt = new
FileOutputStream(fileExt);

            //Escrever no arquivo
            fosExt.write(Dados.getBytes());

            //Fechar o arquivo
            fosExt.close();

            SharedPreferences sharedpref =
getSharedPreferences("InfoColab",
Context.MODE_PRIVATE);
            SharedPreferences.Editor prefeditor =
sharedpref.edit();
            prefeditor.putString("Dadostxt",
```

```
Dados);
            prefeditor.commit();

        }
    }



    public void startCamera(View view){
        buscarDados(); // dados para o nome da
fotografia

        //guardar numa string indicação por parte
do utilizador do tipo de despesa
        TextView TipoDespesas =
(TextView)findViewById(R.id.EditTipoDespesa);
        String newTipoDespesas =
TipoDespesas.getText().toString();

        TextView DiaDespesa =
(TextView)findViewById(R.id.EditDia);
        String newDiaDespesa =
DiaDespesa.getText().toString();

        //indica o diretorio para guardar imagem
        File DirDespesa = createFolder();
        // nome da despesa
        File image = new File(DirDespesa,
"Despesa_" + nfunc + "_" + newDiaDespesa +"-"+
mesref+ "_" + newTipoDespesas +  ".jpg");

        Uri photoURI =
FileProvider.getUriForFile(RefeicaoAlojamento.thi
s,
                BuildConfig.APPLICATION_ID +
".provider",
                image);

        Intent a = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);

        a.putExtra(MediaStore.EXTRA_OUTPUT,
photoURI);
        startActivityForResult(a,CODE_CA);
    }

    protected void onActivityResult(int
requestCode, int resultCode , Intent a) {

        super.onActivityResult(requestCode,
resultCode, a);
        if (requestCode == CODE_CA) {
            if (resultCode == RESULT_OK) {

                ca++;
                 Nfoto = "N° de Fotos:" +
```

```
Integer.toString(ca);

                TextView textview1;
                textview1 = (TextView)
findViewById(R.id.textView);
                 textview1.setText(Nfoto);


Toast.makeText(RefeicaoAlojamento.this, "Foto
guardada", Toast.LENGTH_SHORT).show();
            }
            if (resultCode == RESULT_CANCELED) {

Toast.makeText(RefeicaoAlojamento.this, "Foto NÃO
guardada", Toast.LENGTH_LONG).show();
            }
        }
    }

    public void startSalva(View view) throws
IOException {

        GuardarDados();
        Toast.makeText(RefeicaoAlojamento.this,
"Dados guardada", Toast.LENGTH_SHORT).show();


        Intent Salva = new Intent(this,
MainDespesas.class);
        startActivity(Salva);
    }

}
```

Anexo B. Código de alguns *layout's* criados

Layout: activity_main_despesas.xml

Figura 24 – *Layout* Menu Central Despesas.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res
/android"

xmlns:app="http://schemas.android.com/apk/res-
auto"

xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainDespesas">

    <Button
        android:id="@+id/button3"
        android:layout_width="match_parent"
```

```
        android:layout_height="178dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="21dp"
        android:onClick="startTipoDespesa"
        android:text="Nova Despesa"
        android:background="#1D8E9E"
        android:textColor="#FFFAFA"
        android:textSize="20sp"
        app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.501"

app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="124dp" />

    <Button
        android:id="@+id/send"
        android:layout_width="match_parent"
        android:layout_height="178dp"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"

        android:layout_centerInParent="true"
        android:layout_marginStart="0dp"
        android:layout_marginTop="268dp"
        android:layout_marginEnd="0dp"
        android:background="#1D8E9E"
        android:text="Enviar Despesas"
        android:textColor="#FFFAFA"
        android:textSize="20sp"
        app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.501"

app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="404dp" />

    <CheckBox
        android:id="@+id/checkBox"
        android:layout_width="369dp"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_centerInParent="true"
        android:layout_marginStart="19dp"
        android:layout_marginEnd="23dp"
        android:layout_marginBottom="66dp"
        android:text="Já preencheu todas as
despesas?"
        android:textSize="20sp" />
</RelativeLayout>
```

Layout: activity_info_colab.xml

Figura 25 – *Layout* Informações do colaborador.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res
/android"

xmlns:app="http://schemas.android.com/apk/res-
auto"

xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".InfoColab">

    <TextView
        android:id="@+id/TextNome"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
```

```
        android:layout_marginStart="3dp"
        android:layout_marginLeft="3dp"
        android:layout_marginTop="51dp"
        android:text="Nome:"
        android:textSize="16dp"
        android:textStyle="bold" />

    <EditText
        android:id="@+id/EditNome"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:layout_below="@+id/TextNome"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <TextView
        android:id="@+id/TextNFunc"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/EditNome"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="2dp"
        android:layout_marginLeft="2dp"
        android:layout_marginTop="44dp"
        android:layout_marginEnd="-1dp"
        android:text="Número de Funcionário"
        android:textSize="16dp"
        android:textStyle="bold" />

    <EditText
        android:id="@+id/EditNFunc"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="number"
        android:layout_below="@+id/TextNFunc"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />


    <TextView
        android:id="@+id/TextMesRef"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="0dp"
        android:layout_marginLeft="0dp"
        android:layout_marginTop="273dp"
        android:layout_marginEnd="0dp"
```

```
        android:text="Mês-Ano:"
        android:textSize="16dp"
        android:textStyle="bold" />

    <EditText
        android:id="@+id/EditMesRef"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/TextMesRef"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_marginStart="0dp"
        android:layout_marginLeft="-2dp"
        android:layout_marginTop="5dp"
        android:ems="10"
        android:hint="MM-AAAA"
        android:inputType="date" />

    <TextView
        android:id="@+id/TextCentroCustos"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/EditMesRef"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="3dp"
        android:layout_marginLeft="3dp"
        android:layout_marginTop="64dp"
        android:layout_marginEnd="0dp"
        android:text="Centro de Custos:"
        android:textStyle="bold" />

    <EditText
        android:id="@+id/EditCentroCustos"
        android:layout_width="298dp"
        android:layout_height="wrap_content"

android:layout_below="@+id/TextCentroCustos"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_marginStart="1dp"
        android:layout_marginLeft="1dp"
        android:layout_marginTop="5dp"
        android:clickable="false"
        android:cursorVisible="false"
        android:ems="10"
        android:focusable="false"
        android:focusableInTouchMode="false"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
android:layout_toEndOf="@id/EditCentroCustos"

android:layout_alignBottom="@id/EditCentroCustos"
        android:onClick="startLista"
        android:text="Opções"
        android:background="#1D8E9E"
        android:textColor="#FFFAFA"
        android:textSize="20sp"/>

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="67dp"
        android:text="Avançar"
        android:onClick="startMainDespesas"
        android:background="#1D8E9E"
        android:textColor="#FFFAFA"
        android:textSize="14sp"/>


</RelativeLayout>
```

Layout: activity_refeicao_alojamento.xml

Figura 26 – *Layout* Informações da despesa.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res
/android"

xmlns:app="http://schemas.android.com/apk/res-
auto"

xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".RefeicaoAlojamento">


    <TextView
        android:id="@+id/TextDia"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
```

```
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="0dp"
        android:layout_marginLeft="0dp"
        android:layout_marginTop="5dp"
        android:layout_marginEnd="2dp"
        android:text="Dia:"
        android:textSize="16dp"
        android:textStyle="bold" />

    <EditText
        android:id="@+id/EditDia"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/TextDia"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:ems="10"
        android:inputType="number" />

    <TextView
        android:id="@+id/TextTipoDespesa"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/EditDia"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentEnd="true"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="0dp"
        android:layout_marginLeft="0dp"
        android:layout_marginTop="0dp"
        android:layout_marginEnd="-2dp"
        android:text="Tipo De Despesa:"
        android:textStyle="bold" />

    <EditText
        android:id="@+id/EditTipoDespesa"
        android:layout_width="318dp"
        android:layout_height="wrap_content"

android:layout_below="@+id/TextTipoDespesa"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_marginStart="0dp"
        android:layout_marginLeft="1dp"
        android:layout_marginTop="5dp"
        android:clickable="false"
        android:cursorVisible="false"
        android:ems="10"
        android:focusable="false"
        android:focusableInTouchMode="false"
        android:inputType="textPersonName" />
```

```
    <Button
        android:id="@+id/button"
        android:layout_width="86dp"
        android:layout_height="wrap_content"

android:layout_alignBottom="@id/EditTipoDespesa"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="1dp"
        android:layout_marginEnd="5dp"
        android:layout_marginBottom="2dp"

android:layout_toEndOf="@id/EditTipoDespesa"
        android:background="#1D8E9E"
        android:onClick="startListaDespesas"
        android:text="Opções"
        android:textColor="#FFFAFA"
        android:textSize="14sp" />

    <TextView
        android:id="@+id/textValor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

android:layout_below="@+id/EditTipoDespesa"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="0dp"
        android:layout_marginTop="153dp"
        android:layout_marginEnd="-10dp"
        android:text="Valor:"
        android:textStyle="bold" />

    <EditText
        android:id="@+id/editTextValor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textValor"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="0dp"
        android:layout_marginTop="176dp"
        android:layout_marginEnd="0dp"
        android:ems="10"
        android:inputType="numberSigned|number"
/>

    <TextView
        android:id="@+id/textComentario"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editTextValor"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
```

```
        android:layout_centerVertical="true"
        android:layout_marginStart="0dp"
        android:layout_marginTop="305dp"
        android:layout_marginEnd="0dp"
        android:text="Comentário:"
        android:textStyle="bold" />

    <EditText
        android:id="@+id/editTextComentario"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

android:layout_below="@+id/textComentario"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="1dp"
        android:layout_marginTop="325dp"
        android:layout_marginEnd="-6dp"
        android:ems="10"
        android:inputType="textPersonName" />

    <Button
        android:id="@+id/button10"
        android:layout_width="196dp"
        android:layout_height="89dp"
        android:layout_alignBottom="@id/button11"
        android:layout_alignParentStart="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="2dp"
        android:layout_marginBottom="0dp"
        android:background="#1D8E9E"
        android:onClick="startCamera"
        android:text="Foto"
        android:textColor="#FFFAFA"
        android:textSize="20sp" />

    <Button
        android:id="@+id/button11"
        android:layout_width="196dp"
        android:layout_height="89dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="0dp"
        android:layout_marginBottom="0dp"
        android:background="#1D8E9E"
        android:onClick="startSalva"
        android:text="Salvar"
        android:textColor="#FFFAFA"
        android:textSize="20sp" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
```

```
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="94dp"
        android:text="N° de Fotos:"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/textViewMatricula"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="0dp"
        android:layout_marginTop="228dp"
        android:layout_marginEnd="-5dp"
        android:text="Maticula(se aplicavél)"
        android:textStyle="bold" />

    <EditText
        android:id="@+id/editMatricula"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="0dp"
        android:layout_marginTop="251dp"
        android:layout_marginEnd="-5dp"
        android:ems="10"
        android:inputType="textPersonName" />


</RelativeLayout>
```

Anexo C. Código do *AndroidManifest.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res
/android"
    package="com.example.despesasiep">

    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_S
TORAGE" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_ST
ORAGE" />
    <uses-permission
android:name="android.permission.CAMERA" />
    <uses-permission
android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.ACCESS_NETWORK_S
TATE" />


    <application
        android:allowBackup="true"
        android:icon="@drawable/logo"
        android:label="@string/app_name"

android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.DespesasIEP">
        <activity android:name=".MainActivity" />
        <activity
android:name=".RefeicaoAlojamento" />
        <activity android:name=".MainDespesas" />
        <activity android:name=".InfoColab" />
        <activity android:name=".Splach">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>

        <provider
            android:name=".GenericFileProvider"

android:authorities="${applicationId}.provider"
```

```
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data

android:name="android.support.FILE_PROVIDER_PATHS
"

android:resource="@xml/provider_paths" />
        </provider>
    </application>

</manifest>
```