

Computação Móvel

DayliStudent



Coimbra, dezembro de 2018

André Pascoal Bento 2017234390
apbento@student.dei.uc.pt

Jorge André Dias Silva 2017234557
jadsilva@student.dei.uc.pt

Rúben Filipe Ramos Alves 2012142647
rfalves@student.dei.uc.pt

Tabela de Versões

Tabela 1: Versões do Documento

Versão	Data	Nome	Descrição
1.0	27/12/2018	Equipa	Revisão final de todo o documento.
0.8	18/12/2018	Equipa	Revisão e melhoramento das seções elaboradas nas versões 0.3 e 0.4.
0.7	11/12/2018	Equipa	Revisão e melhoramento das seções elaboradas nas versões 0.1 e 0.2.
0.6	04/12/2018	Equipa	Finalização da tarefa anterior.
0.5	27/11/2012	Equipa	Inserção da documentação relativa à implementação da solução.
0.4	20/11/2018	Equipa	Finalização das tarefas anteriores e especificação da arquitetura da solução.
0.3	13/11/2018	Equipa	Estruturação e inserção dos protótipos de UI e Storyboarding.
0.2	6/11/2018	Equipa	Introdução, visão do projeto, Gantt Chart provisório, definição dos casos de uso e especificação dos requisitos.
0.1	30/10/2018	Equipa	Estruturação do Projeto.

Lista de Tabelas

Tabela 1: Versões do Documento	1
Tabela 2: Análise da Concorrência	10
Tabela 3: Elementos da equipa e respetivos cargos	13
Tabela 4: Intervenientes no projeto e respetivos contactos	13
Tabela 5: Cliente do Projeto	14
Tabela 6: Especificação dos Requisitos Funcionais	19
Tabela 7: Especificação detalhada dos Requisitos Funcionais	20
Tabela 8: Cenário #1 – Desempenho	22
Tabela 9: Cenário #2 -Disponibilidade	23
Tabela 10: Cenário #3 – Usabilidade	23
Tabela 11: Cenário #4 - Segurança	23
Tabela 12: Plano de Testes	33
Tabela 13: Resultados dos testes	36

Lista de Figuras

Figura 1: Quadro de Tarefas	15
Figura 2: Diagrama de Gantt (Previsão)	16
Figura 3: Diagrama de Gantt (Realidade)	16
Figura 4: Diagrama C4	17
Figura 5: Diagrama do modelo de dados da aplicação	18
Figura 6: Diagrama de Casos de Uso	20
Figura 7: Menu de Registo, Principal e Barra Lateral	25
Figura 8: Menu de Definições	25
Figura 9: Menu de Grupos, Notas Partilhadas e Criação de Notas	26
Figura 10: Menu de Notas, Partilhar Notas QR Code e Câmera	26
Figura 11: Menu Budget Notes, Criação de Carteira, Criação de uma BudgetNote e Detalhes de Carteira	28
Figura 12: Ecrã de Estatísticas	29
Figura 13: Ecrã de Créditos	29
Figura 14: Layouts da aplicação	32

Índice

Introdução	6
Enquadramento	6
Solução proposta	6
Público-Alvo	7
Competências a Explorar	7
Âmbito e Esboço do Documento	8
Visão Geral do Projeto	8
Descrição	8
Análise do Mercado	9
Discriminação do Segmento de Mercado	10
Análise da Concorrência	10
Objetivos	11
Restrições	11
Critérios e Sucesso	11
Organização do Projeto	12
Recursos	12
Papéis e Responsabilidades	12
Principais Entregáveis	13
Marcos de Alto Nível	13
Responsabilidades	13
Equipa	13
Contactos	13
Cliente	14
Gestão de Mudanças	14
Objetivo	14
Tipos de alterações	14
Papéis	14
Triggers	14
Etapas do Processo	14
Metodologia e planeamento	15
Planeamento	15

Arquitectura	17
Diagrama de componentes C4	17
Modelo de Dados	18
Análise de requisitos	19
Requisitos Funcionais	19
Casos de Uso	19
Atributos de Qualidade	21
Restrições	24
Mockup's	24
Desenvolvimento	29
Tarefas	29
Aplicação Móvel	30
Servidor	32
Testes	32
Referências	36

1. Introdução

O presente documento apresenta o trabalho envolvido no desenvolvimento do projeto, da unidade curricular Computação Móvel, ao longo do semestre. Através deste documento os utilizadores encontrarão, em detalhe, a descrição de todo o processo seguido na sua realização, estruturação e planeamento.

O objetivo principal foca-se em todo o processo de desenvolvimento na construção do projeto, o DayliStudent, e nos próximos subcapítulos serão apresentados o enquadramento, a solução proposta, o público alvo, as competências exploradas e a estrutura geral deste documento.

1.1. Enquadramento

Nos dias de hoje, de acordo com o Independent^[1], os dispositivos móveis multiplicam-se agora a um ritmo cinco vezes superior ao das pessoas, sendo que a população humana cresce a um ritmo de aproximadamente duas pessoas por segundo, e com isso aparecem oportunidades de negócio, pois podem-se explorar várias funcionalidades do mesmo em benefício do dia-a-dia do utilizador.

O tema “Gerir Poupanças” pode-se enquadrar dentro deste tema, pois como temos um dispositivo móvel sempre nos acompanhando, porque não utilizá-lo como nosso gestor? Qualquer pessoa na sua idade adulta, antigamente, para consultar as suas poupanças e dinheiro que tinha correntemente tinha que ir a uma agência bancária para conseguir fazer um extrato bancário do que tinha na conta, ou então tinha que contar “manualmente” todas as suas poupanças que tem armazenadas no seu mealheiro ou carteira. Para além disto, nunca conseguem verificar as suas estatísticas para verificarem se gastaram mais, por exemplo que o mês anterior, ou se conseguiram poupar mais.

Desta forma, surgiu a necessidade de criar uma solução que conseguisse resolver estes problemas, de uma forma automatizada e móvel, sem ter que requerer a “apontamentos em papel”. Essa solução será apresentada na secção seguinte.

1.2. Solução proposta

O DayliStudent, aplicação móvel direcionada para ambiente Android, surge com a finalidade de colmatar todas as dificuldades descritas anteriormente, numa única aplicação. Esta aplicação para além de conseguir registrar todos os ganhos/gastos do utilizador, permite que o utilizador consiga ter uma noção total dos seus ganhos/gastos e ter uma análise estatística que compare os mesmos relativamente com outros dias da semana.

Os utilizadores da primeira vez que iniciarem a aplicação terão que efetuar um pequeno registo na aplicação para posteriormente todos os dados serem associados à sua conta. Depois disto terão acesso a um *dashboard* principal, no qual irão encontrar um resumo de toda a informação relativa aos seus ganhos/gastos do dia, todas as suas carteiras e respetivas quantias, bem como notas que o utilizador tenha tirado durante o seu dia ou partilhado/recebido de outros utilizadores.

A partir deste momento o utilizador já pode começar a usufruir das funcionalidades da aplicação, como é o caso de inserir *Budget Notes* que irão permitir à aplicação começar a trabalhar para o utilizador, no que toca a tratamento de dados e estatísticas. Para isso o utilizador, através do acesso a um menu lateral, terá acesso a uma opção “Notas Budget”, que irá redirecionar o utilizador para um ecrã em que lhe são apresentadas duas secções, a de Movimentos e a de

Carteiras, bem como dois ícones no menu superior, cada um deles para anotar as *budget notes* e as suas carteiras, respetivamente.

Para anotar os ganhos/gastos o utilizador terá que selecionar a opção de tirar as *budget notes* para encontrar um ecrã que lhe pedirá toda a informação relativa à respetiva nota, como é o caso do título, o tipo de valores da nota (despesa ou ganho), a categoria em que se insere (p.ex. bar, cantina, propinas, etc.), a carteira a que se associa e o seu valor. Dentro desta opção também terá a funcionalidade de tirar uma fotografia, caso o utilizador queira registrar um talão com a respetiva nota, de maneira a especificar melhor os valores que a mesma possa conter. A opção de tirar *budget notes* só será possível de realizar caso o utilizador já possua, uma carteira inserida no sistema. Para isso o utilizador terá que clicar na opção “Criar carteira” dentro do ecrã das “Notas Budget” e localizada no canto superior direito do menu. Para a criação da mesma, serão requeridos alguns dados como é o caso do seu nome (pois pode ser um mealheiro ou então a sua carteira pessoal) e o valor nela contido. Posto isto e inseridos dados relativos a carteiras e a *budget notes* as secções relativas a cada uma delas serão preenchidas com os respetivos dados inseridos e irão atualizar uma área que irá notificar o utilizador do total ganho e gasto de todos os seus dados inseridos. Voltando ao Dashboard principal, o utilizador irá ver os valores relativos aos ganhos e gastos atualizados, bem como os dados da sua carteira.

Quando existem dados inseridos no sistema, durante vários dias, o sistema permite gerar gráficos que permitem ao utilizador verificar o seu total de gastos/ganhos e uma média de gastos durante um espaço de tempo de uma semana. Estes gráficos estão disponíveis através da opção no menu lateral, “Estatísticas”.

Para além destas funcionalidades principais, o utilizador também terá a funcionalidade de tirar notas textuais, quer sejam elas pessoais, ou de grupo. No que toca às notas pessoais o utilizador, terá um bloco de notas tradicionais dentro da sua aplicação, que lhe permitirá tirar qualquer tipo de notas textuais, removê-las, alterá-las e até mesmo partilhá-las através de QR Code com qualquer pessoa que tenha a mesma aplicação instalada. Quanto às notas de grupo o utilizador terá as mesmas funcionalidades que estão contempladas no bloco de notas pessoal, com a vantagem de ter a funcionalidade de conseguir partilhar as mesmas com outros utilizadores da aplicação com os quais o utilizador queira formar um grupo (quer seja ele de trabalho ou de estudo). Estas notas de grupo estarão devidamente organizadas e armazenadas num servidor externo à aplicação.

Por último a aplicação permitirá ao utilizador a sua customização, onde poderá alterar o nome e email registados previamente, poderá alterar o tema da aplicação (cores dos menus e das funcionalidades da aplicação), alterar as definições de envio das notas de grupo e o tamanho da letra da aplicação.

1.3. Público-Alvo

A aplicação tem como público-alvo as pessoas maiores de idade, nomeadamente estudantes universitários e classe trabalhadora, que possuem um dispositivo Android. O foco principal incide sobre os estudantes para que estes consigam gerir ao máximo as suas contas e poupanças juntamente com as suas notas diárias, uma vez que, estes não possuem tanto poder monetário e precisam de auxílio no que toca à gestão do mesmo.

1.4. Competências a Explorar

A aplicação desenvolvida tem como objetivo, para além de satisfazer as necessidades do público alvo, satisfazer todos os tópicos abordados durante o semestre na unidade curricular de

Computação Móvel. Para isso e para o correto funcionamento da mesma foram também aplicadas as boas práticas ensinadas na unidade, ligadas ao desenvolvimento Android e arquitetural.

Dito isto, os tópicos abordados no projeto e relacionados com a unidade curricular são:

- Actividades e fragmentos;
- App Bar (com items clicáveis);
- Navigation Drawer;
- ListView;
- RecyclerView;
- Integração com a Câmera;
- Gerador e scanner de QR Codes;
- Notificações na barra do dispositivo;
- Interação com File Manager;
- Interação com Bases de Dados;
- Cursor Adapter;
- Shared Preferences;
- Interação com Share Preferences;
- Comportamentos paralelos com threads;
- GSON (Objects into JSON and back);
- Retrofit API.

1.5. Âmbito e Esboço do Documento

Este documento está dividido em várias secções para melhor compreensão quer do mesmo, quer das várias fases do projeto que foi desenvolvido.

A **secção 1** apresenta uma pequena introdução da aplicação desenvolvida e um enquadramento ao seu tema.

A **secção 2** apresenta uma visão geral do projeto, onde se faz uma descrição detalhada do mesmo, menciona-se os seus objetivos, bem como as restrições e critérios de sucesso.

A **secção 3** explica a organização do projeto, onde fala-se dos recursos necessários para a realização do projeto, da equipa que desenvolverá a aplicação e quais as responsabilidades de cada elemento da mesma. Também se encontra mencionado qual o ciclo de vida adotado para o desenvolvimento do projeto.

Na **secção 4** especifica-se como iremos fazer a gestão de mudanças, ou seja, que tipo de mudanças podem ocorrer, o objetivo dessas mudanças e os passos a executar para efetuar tais mudanças. Por fim, na **secção 6** temos as referências.

2. Visão Geral do Projeto

2.1. Descrição

Este projeto foi-nos proposto no âmbito da disciplina de Computação Móvel do curso de Mestrado em design e Multimédia da Universidade de Coimbra, com a finalidade de aplicar os conhecimentos e técnicas adquiridas ao longo do semestre através do desenvolvimento de uma aplicação móvel para sistema operativo Android e simultaneamente resolver um problema real.

Através de uma reflexão entre os membros do grupo e sendo estes, atualmente, participantes na comunidade estudantil foi decidido atacar um dos problemas mais abundantes no seu dia-a-dia cujo tema principal é “Gestão Monetária”. Tendo como foco principal encontrar

uma solução para este tema o grupo decidiu criar o “DayliStudent”, uma aplicação móvel que permite aos seus utilizadores a oportunidade de encontrarem uma solução simples e prática para o auxílio de toda a sua gestão monetária.

Após o *download* e um pequeno registo na nossa aplicação os nossos utilizadores terão acesso a um conjunto diverso de funcionalidades desenvolvidas pelo grupo capazes de solucionar o problema mencionado anteriormente. Uma das funcionalidades principais consiste na criação de notas onde, um determinado utilizador, regista todos os ganhos ou despesas e através dos mesmos é gerada uma análise estatística que permite visualizar a linha temporal de despesas/ganhos e compará-las com outros dias da semana. Estas notas podem ser de divididas em dois tipos pessoais e de grupo, sendo que notas pessoais só são acessíveis pelo próprio utilizador e as de grupo podem ser partilhadas entre múltiplos utilizadores. Apesar de serem notas pessoais notas é permitido o envio das mesmas através de email para outro utilizador.

Qualquer informação a nível de registo, grupos, notas e até mesmo definições da própria aplicação, contém o correspondente menu de edição que permite aos nossos utilizadores alterarem os diversos dados introduzidos na nossa aplicação.

Por último, o sistema permite gerar gráficos que permitem ao utilizador verificar o seu total de gastos/ganhos e uma média de gastos durante um espaço de tempo de uma semana.

2.2. Análise do Mercado

Com a evolução rápida dos dispositivos móveis e especialmente com aparecimento dos primeiros smartphones, surgiu uma rápida procura de aplicações cujas funcionalidades permitem resolver problemas do dia-a-dia levando várias empresas a concentrarem-se no desenvolvimento móvel.

O mercado de aplicações direcionadas para os dispositivos móveis encontra-se numa fase de saturação onde um determinado utilizador facilmente encontra uma aplicação que o auxilie a resolver os seus diferentes problemas.

A Gartner prevê que menos de 0.01% das aplicações móveis serão consideradas sucessos financeiros no final do ano 2018. Também afirmam que apesar de existirem um vasto número de aplicações disponíveis no mercado e que estas levam as pessoas, especialmente desenvolvedores, a pensar que as aplicações móveis são o caminho para o sucesso e todas as suas riquezas associadas, tal não é verdade dizendo que a maior parte das aplicações não são rentáveis e nem estão desenhadas para tal, em vez disso, são desenhadas como meios publicitários ou foram construídas só por diversão.

Segundo um estudo realizado pela Distimo^[2] apenas 12% das aplicações, encontradas no Top 300 das mais transferidas na Google Play, são consideradas “novas”. Esta situação demonstra um problema grave para a adição de novas aplicações obrigando-as a serem mais originais, criativas e com melhorias a nível de funcionalidades quando comparadas com as já existentes.

Apesar de ser um mercado difícil para a introdução novas aplicações, há uma alta probabilidade de um novo produto devidamente desenvolvidos se torne viral e consequentemente, capaz de obter um crescimento rápido de adesão à mesma entre a comunidade que se pretende atingir.

2.3. Discriminação do Segmento de Mercado

De acordo com o blog “App Download and Usage Statistics (2018)”^[3] cerca de 65% da aplicação mais usadas encontram-se no domínio das redes sociais, entretenimento, música e multimédia deixando os restantes 35% para aplicações didáticas ou auxiliares de problemas do dia-a-dia. Também afirma que malta jovem até 44 anos passa mais tempo no telemóvel e tablet (até um mínimo de 64%) comparando com uso de desktops.

Outro ponto relevante é o facto de que segundo a LifeHacks^[4] o TOP 10 de aplicações orientadas ao auxílio de gestão monetária, metade delas são compatíveis apenas via web ou iOS, permitindo assim a entrada de novas aplicações a nível do sistema operativo Android.

Através dos dados anteriores, é possível concluir que a aplicação, DayliStudent, apesar de ser categorizada para apenas 35% dos utilizadores do mercado atual tem uma alta probabilidade de sucesso visto que como o público-alvo são adultos e jovens com atenção especial em estudantes, sendo estes que mais usam aplicações no seu dia-a-dia. Também é possível verificar que apesar de existirem aplicações dirigidas ao tema da Gestão Monetária estas encontram-se bastante dívidas em termos de exclusividade com diferentes plataformas, levando a “aberturas” para introdução de novas aplicações.

2.4. Análise da concorrência

Segundo LifeHacks é possível restringir a nossa concorrência principal para a aplicação Mint: Budget, Bills, Finance^[5] é a líder das aplicações financeiras presentes na GooglePlay e a Spendee^[6], atualmente na posição 5, que é a líder na plataforma Android.

A seguinte tabela demonstra os diferentes pontos fortes e fracos entre a nossa aplicação e a concorrência mencionada anteriormente:

Tabela 2: Análise da Concorrência

Pontos Fortes / Fracos	Plataforma	Preço	Público-Alvo	Serviços
DayliStudent	Android	Gratuita	Jovens	<ul style="list-style-type: none"> → Personal Notes → Unlimited wallets → Unlimited budgets → Share notes with others
Mint	Android (EUA e Canadá)	Gratuita	Adultos	<ul style="list-style-type: none"> → Unlimited wallets → Unlimited budgets
Spendee	Android	Gratuita 3.39€/Mês 25/99€/Ano	Adultos	<ul style="list-style-type: none"> → Bank accounts sync → Limit/Unlimited wallets → Limited/Unlimited budgets → Share Wallets with others → Backup and Sync (Premium exclusive)

Através da tabela anterior é possível verificar que, apesar de Mint ser a líder das aplicações financeiras, esta pode ser descartada como concorrente visto que apenas se encontra disponível nos Estados Unidos da América e Canadá. O mesmo não se aplica ao Spendee que se encontra disponível na Europa e para a plataforma Android, tornando-se assim o nosso maior concorrente.

A abordagem utilizada para distinguir o nosso produto quando comparado com o Spendee é o facto de que a nossa aplicação contém um sistema de bloco de notas privadas separado das notas relativas a despesas, enquanto Spendee apenas permite criar notas de despesas/ganhos. A DayliStudent é totalmente gratuita, evitando custos adicionais para fornecer funcionalidades extras aos seus utilizadores, que é uma mais valia especialmente quando nossos consumidores serão especialmente estudantes com pouca posse financeira. Outro fator relevante é que a nossa aplicação permite a criação de grupos de partilha do tipo *broadcast* onde é possível partilhar múltiplas notas em simultâneo com diferentes utilizadores.

2.5. Objetivos

Sendo DayliStudent um projeto de nível académico é possível dividir os objetivos em dois grupos distintos, objetivos educacionais e objetivos da própria aplicação.

- A nível educacional encontramos os seguintes objetivos:
- Utilização e consolidação de todos os conhecimentos adquiridos ao longo do semestre durante o desenvolvimento da aplicação móvel.
- Aperfeiçoamento das habilidades organizacionais dos membros da equipa.
- Contacto com ferramentas utilizadas no desenvolvimento de aplicações Android.

A nível da própria aplicação encontram-se os seguintes objetivos:

- Auxiliar a gestão monetária dos seus utilizadores de um modo simples e prático.

2.6. Restrições

As restrições impostas no projeto limitam-se apenas na etapa do seu desenvolvimento sendo que não existem restrições a nível financeiro e de hardware. A nível de restrições de desenvolvimento encontram-se as seguintes:

- A Aplicação é exclusiva para o sistema operativo Android.
- Certas funcionalidades necessitam de permissões adicionais que caso não sejam concedidas pelo utilizador, este pode não usufruir das mesmas.
- Curto período de desenvolvimento, visto que o projeto é de origem académica e elaborado no âmbito de uma disciplina semestral

2.7. Critérios e Sucesso

Para além dos critérios baseados em indicadores de desempenho individual de cada elemento da equipa, foram estipulados outros objetivos relativos exclusivamente ao processo de desenvolvimento da plataforma, tais como:

- Equipa fortemente organizada e estruturada

- Rápida curva de aprendizagem em relação aos conceitos necessários a aplicar no projeto
- Lifecycle adequado ao projeto e à equipa
- Cumprimento de todas as metas presentes no plano de desenvolvimento do projeto
- Implementação correta dos conceitos adquiridos durante o semestre

A respeito dos objetivos relativos ao produto final, a equipa estabeleceu como objetivo principal a obtenção de uma aplicação o mais próximo possível de um sistema completo, pronto a utilizar e que cumpra todos os requisitos.

3. Organização do Projeto

Na presente secção serão apresentados os aspetos com maior importância para o projeto com vista a que este seja bem estruturado, organizado e sucedido, contemplando todos os recursos necessários.

3.1. Recursos

Este projeto tem à sua disposição 3 elementos estudantes como recurso, que estarão divididos por vários cargos e funções: Gestor de Projeto, Gestor de Requisitos, *Developers* e *Testers*.

Os restantes recursos distinguem-se por ser três computadores pessoais, dois dispositivos móveis para testar a aplicação, um servidor declarado na *Google Cloud Engine*^[7] e uma sala de reuniões para que a equipa se possa juntar periodicamente com vista a debater as mais variadas fases de desenvolvimento do projeto.

3.2. Papéis e Responsabilidades

Os papéis e responsabilidades deste projeto serão divididos nos seguintes tópicos:

- **Developer - Front-End:** responsável por todo o desenvolvimento do projeto que será visível pelo utilizador (Interface *Web* e *Mobile*), bem como toda a estrutura das aplicações disponibilizadas para os mesmos.
- **Developer - Back-End:** responsável por todo o desenvolvimento do servidor, que irá ter a responsabilidade de gerir e providenciar os dados necessários à aplicação *front-end*.
- **Tester:** responsável pela realização de todos os testes possíveis e necessários para o bom funcionamento das aplicações e do *back-office*. Para além disso, terá de fornecer toda a documentação necessária a futuras correções de erros detetados.
- **Gestor de projeto:** esta função terá que aplicar os conhecimentos, as habilidades e as técnicas para elaboração de atividades relacionadas a um conjunto de objetivos pré-definidos, num certo prazo, com um certo custo e qualidade, através da mobilização de recursos técnicos e humanos.
- **Gestor de Requisitos:** responsável por analisar, acompanhar, definir prioridades e concordâncias sobre os requisitos. Também terá o controlo sobre mudanças e comunicação aos *stakeholders*.

3.3. Principais Entregáveis

O principal entregável deste projeto será a aplicação desenvolvida ao Professor Tiago Cruz, para avaliação na unidade curricular Computação Móvel. Com vista a que este fique o mais completo possível, será também entregue o repositório que contém todo o código fonte, quer da aplicação *front-end* quer da componente *back-end*, bem como toda a documentação relativa ao levantamento e definição de requisitos, à arquitetura e todo o processo de desenvolvimento deste projeto.

3.4. Marcos de Alto Nível

O desenvolvimento do projeto contará com várias metas definidas pela equipa:

- **Validação da ideia:** 31/10/2018
- **Sprints:**
 - 31/10/2018 - 06/11/2018;
 - 07/10/2018 - 14/11/2018;
 - 15/11/2018 - 29/11/2018;
 - 30/11/2018 - 13/12/2018;
 - 14/12/2018 - 28/12/2018.
- **Início do processo de desenvolvimento:** 15/11/2018
- **Entrega final do projeto:** 28/12/2018
- **Defesa final:** durante o mês de janeiro em dia a definir posteriormente.

4. Responsabilidades

Neste tópico são apresentados alguns dos intervenientes deste projeto, nomeadamente a equipa que irá proceder ao desenvolvimento do produto definido e o cliente que representa a entidade que requer o produto.

4.1. Equipa

A equipa, é composta por um total de 3 elementos estudantes no curso de Mestrado em Engenharia Informática, lecionado na Faculdade de Ciências e Tecnologias da Universidade de Coimbra. A Tabela 3 expõe todos os elementos e respetivos cargos:

Tabela 3: Elementos da equipa e respetivos cargos

Nome	Cargo
André Pascoal Bento	Developer Back-End, Front-End/Tester
Jorge André Dias Silva	Developer Back-End, Front-End/Gestor de Projeto
Rúben Filipe Ramos Alves	Designer/Tester

4.2. Contactos

Os contactos dos elementos da equipa e do cliente referido são os seguintes:

Tabela 4: Intervenientes no projeto e respetivos contactos

Nome	Contacto
André Pascoal Bento	apbento@student.dei.uc.pt

Jorge André Dias Silva	jadsilva@student.dei.uc.pt
Rúben Filipe Ramos Alves	rfalves@student.dei.uc.pt

4.3. Cliente

O cliente deste projeto é o docente responsável pela unidade curricular, Computação móvel, da própria Universidade de Coimbra, situada na Rua Sílvio Lima, Pólo II. A tabela 3 apresenta o cliente e o seu respetivo contacto:

Tabela 5: Cliente do Projeto

Nome	Contacto
Tiago José dos Santos Martins da Cruz	tjcruz@dei.uc.pt

5. Gestão de Mudanças

5.1. Objetivo

O sentido de existir um sistema de gestão de mudanças passa pela necessidade da tentativa de minimização do impacto de alterações, realizadas ao longo do projeto, no produto final.

5.2. Tipos de alterações

As alterações apenas provêm da equipa, uma vez que o cliente definiu definitivamente quais eram os requisitos finais da aplicação. Nas alterações por parte dos elementos da equipa, é necessário ter em conta possíveis mudanças de abordagens, alterações motivadas por custos e/ou metas e alocação de recursos.

5.3. Papéis

Em caso de mudanças, a equipa tem de desempenhar um conjunto de funções em prol da reestruturação do projeto. Desta feita, será necessário que o Gestor de Projeto notifique toda a equipa da alteração em questão, atualize o *backlog*/documento de requisitos e que a restante equipa se adapte a esta realidade.

5.4. Triggers

As modificações necessárias são despoletadas pela equipa de desenvolvimento, como já foi referido anteriormente, pois a equipa de desenvolvimento, durante o sprint pode identificar problemas ou soluções melhores para o que lhe foi pedido.

5.5. Etapas do Processo

Quando existir a necessidade de uma alteração, toda a equipa terá de ser notificada, o *backlog* e o documento de requisitos terão de ser atualizados e, por fim, os próximos sprints irão refletir essas mesmas mudanças.

6. Metodologia e planeamento

Considerando que este projeto está dividido em 4 fases (definição da ideia/requisitos, definição do modelo de dados, implementação e documentação), e que a duração do projeto é curta, optámos por usar uma metodologia Agile^[8], nomeadamente a SCRUM^[9], pois permite ter componentes do sistema funcionais de uma forma iterativa e aumentar a qualidade do projeto, uma vez que os vários componentes do sistema são testados à medida que vão sendo implementados.

Serão implementados *sprints* de 1 ou 2 semanas e, no final de cada *sprint*, existirá uma reunião com todos os elementos do grupo, de modo a apresentar todo o resultado do trabalho desenvolvido. Estas terão dois estados diferentes, dependendo da semana em que nos encontrarmos. Se a semana atual não coincidir com a reunião de final do sprint, a reunião da equipa terá como finalidade a análise do trabalho efetuado até à data e possíveis considerações para o desenvolvimento futuro. Caso coincida, a reunião da equipa será com vista à discussão dos seguintes aspetos:

1. Trabalho realizado no corrente *sprint*;
2. Trabalho a realizar no próximo *sprint*;
3. Avaliação dos riscos e das restrições.

Para além das sprints, também foi necessária a gestão dos recursos humanos para o correto funcionamento do seu desenvolvimento e da sua divisão de tarefas. Para isto foram utilizados os quadros do GitHub, pois assim consegue-se promover a organização e deteção de problemas numa fase inicial do projeto, pontos cruciais para o bom desenvolvimento do projeto.

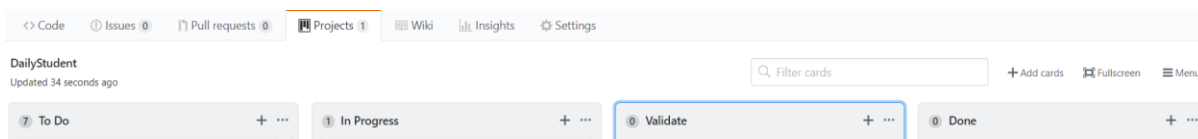


Figura 1: Quadro de Tarefas

6.1. Planeamento

Nesta secção irão ser apresentados os diagramas de Gantt inerentes a este projeto, sendo também feita a comparação da estimativa inicial com a realidade.

Para tal, decidimos optar por realizar duas abordagens de visualização, uma que recaí sobre a previsão, ou seja, consiste num diagrama em que os valores apresentados possuem bastante incerteza, e outra, em que os valores apresentados refletem exatamente o que se passou durante a execução deste projeto.

A primeira visualização foi criada com base numa atividade, levada a cabo por todo o grupo, onde foi debatido qual seria o tempo gasto por tarefa. Este tempo gasto é definido com base numa aproximação onde se encontram contemplados os valores de “*best case*”, “*worst case*”, “*most likely case*” e “*expected case*”. Os piores e os melhores casos foram dados com 2% e 98% de incerteza respetivamente. Para os diagramas de Gantt foram utilizados os valores que referem o melhor caso possível.

A figura seguinte representa o diagrama de Gantt parcial para a primeira visualização (previsão):

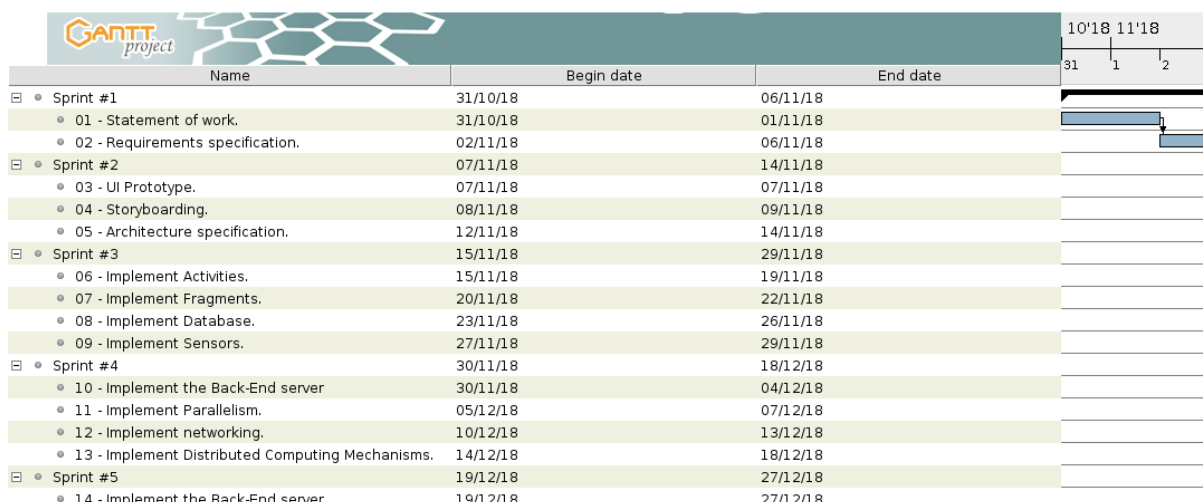


Figura 2: Diagrama de Gantt (Previsão)

Por outro lado, a figura seguinte representa o diagrama parcial de Gantt para a segunda visualização (realidade):

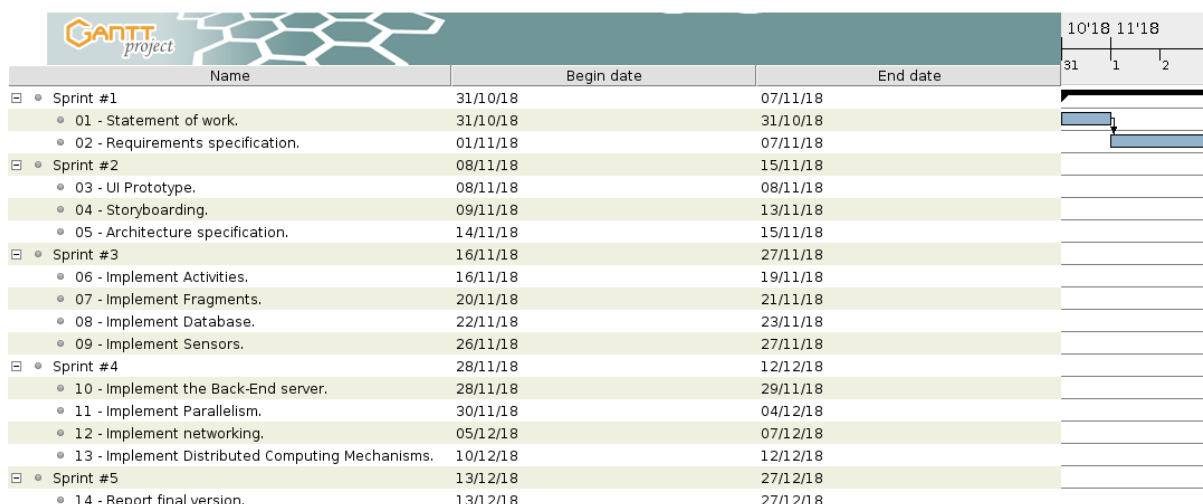


Figura 3: Diagrama de Gantt (Realidade)

Como podemos observar, através das duas figuras apresentadas anteriormente é possível constatar um certo desvio do que estava realmente previsto e estipulado, face ao que foi realmente executado. Isto deveu-se principalmente, à baixa experiência em estimativa existente à data pelos membros do grupo. Os desvios devem-se a gastos mais elevados no *Sprint #1*, em que no SOW (Statement of Work) foi gasto metade do tempo, no entanto na definição de requisitos foram gastos mais 2 dias. Relativamente ao segundo *Sprint* foi gasto mais tempo, cerca de dois dias, com a escrita e definição de requisitos. Quanto à implementação, foi reduzido em média um dia por cada *task*, tempo este que foi utilizado para realizar o último *Sprint* deste projeto, ou seja, o relatório final.

7. Arquitectura

Nesta secção é apresentada a arquitetura do sistema através de um Diagrama de Componentes C4 e de um modelo de dados. Foram escolhidos apenas estes dois porque permite uma visão completa de todo o sistema quer a nível de componentes, quer a nível da tipologia e morfologia dos dados.

7.1. Diagrama de componentes C4

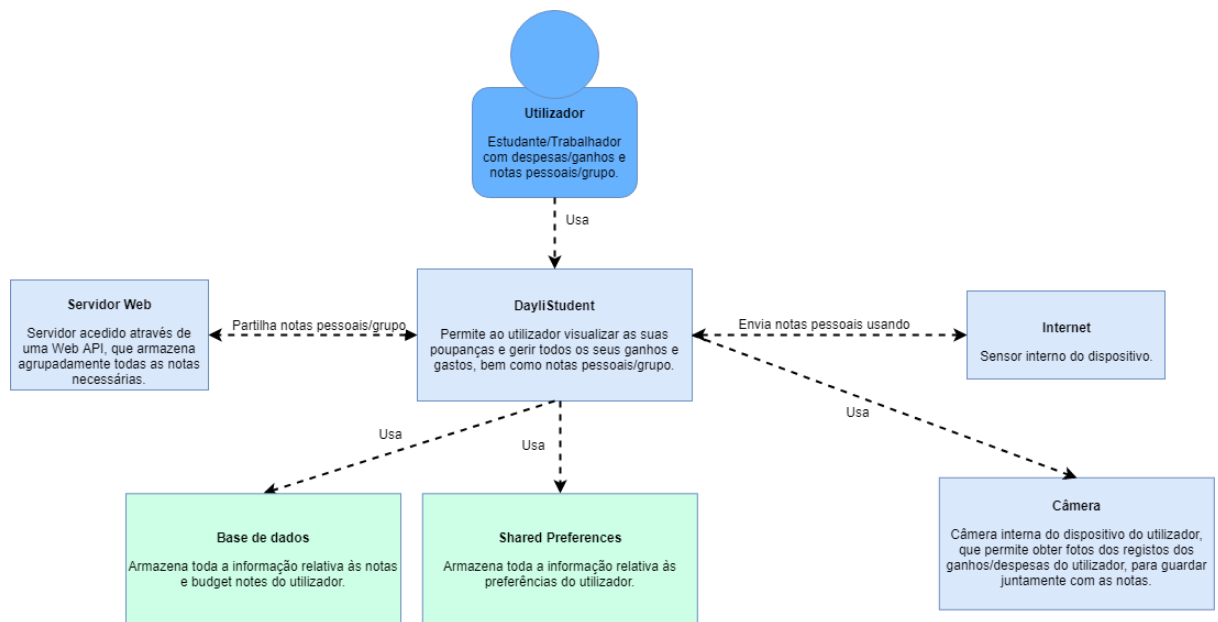


Figura 4: Diagrama C4

Como podemos verificar na Figura 4, a aplicação irá interagir com os componentes do dispositivo e com um servidor externo. Os componentes internos do dispositivo serão os mecanismos de armazenamento de dados (*Bases de dados* e *Shared Preferences*), para armazenar tudo o que esteja relacionado com os dados inseridos pelo utilizador e preferências do mesmo, a *Câmera*, para conseguir obter imagens para associar aos dados inseridos e para efetuar a leitura de um QR Code gerado por outra aplicação Daylistudent para receber notas pessoais partilhadas, e o sensor de internet do dispositivo, para conseguir transmitir as notas de grupo/pessoais através de um cliente mail instalado no dispositivo. O servidor web será utilizado para enviar notas de grupo entre os utilizadores que possuem a aplicação instalada e estão inseridos no mesmo grupo para o qual foram enviadas as notas.

7.2. Modelo de Dados

O modelo de dados mostra a estrutura lógica dos dados, incluindo as relações e restrições que determinam como os dados podem ser armazenados e acessados. Este modelo foi projetado com base nas regras e nos conceitos do modelo de dados mais abrangente.

A Figura 5, apresenta o diagrama de dados orientado a objetos definidos para o sistema.

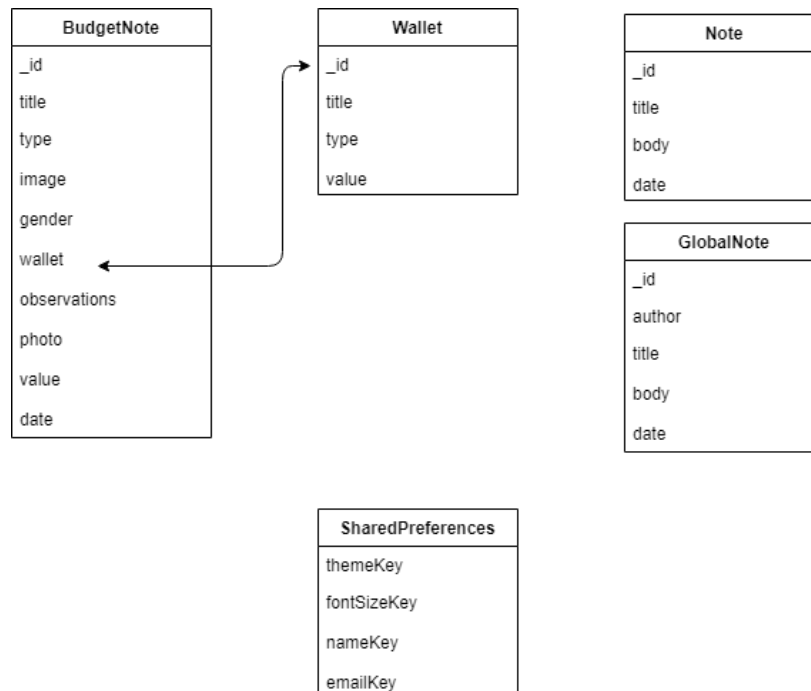


Figura 5: Diagrama do modelo de dados da aplicação

Como podemos ver na figura acima, existem tabelas para guardar os dados de várias entidades nomeadas como BudgetNotes, Wallets, Notes e Global Notes. Neste diagrama de representação dos dados podemos visualizar que uma BudgetNote estará associada sempre a uma Wallet através do seu id. Alguns atributos a destacar são, o **image**, guardado na tabela BudgetNote, que irá permitir associar as fotos tiradas na aplicação a uma determinada BudgetNote, o **type** que irá permitir à aplicação distinguir quais são as BudgetNotes consideradas despesas e quais as consideradas ganhos para futuras análises, e o **value**, que irá complementar todas as análises e estatísticas.

Também é possível observar que existe uma tabela SharedPreferences, sendo que será apenas utilizada para registar dados relativos às configurações do utilizador e da aplicação, como nomeadamente os dados relativos ao tema aplicado, ao tamanho da fonte e dos dados do utilizador (nome e e-mail).

8. Análise de requisitos

Nesta secção é apresentado o levantamento e a respetiva análise dos requisitos funcionais e dos atributos de qualidade do sistema. Esta secção é essencial para o desenvolvimento do projeto, uma vez que permite identificar os objetivos pretendidos para a aplicação, validar o sucesso e a concretização do projeto, ou seja, se a aplicação está conforme o que era suposto fazer.

8.1. Requisitos Funcionais

Os requisitos funcionais da aplicação, são as funcionalidades que a mesma apresenta e realiza. Na Tabela 6, apresenta-se a lista dos requisitos funcionais contemplados pela aplicação.

Tabela 6: Especificação dos Requisitos Funcionais

Requisitos Funcionais		
ID	Requisito	Prioridade
RF-1	Efetuar Registo	Médio
RF-2	Gerir Carteiras	Alto
RF-3	Gerir Budget Notes	Alto
RF-4	Tirar fotografia	Médio
RF-5	Eliminar notas	Médio
RF-6	Ver Estatísticas	Médio
RF-7	Gerir Notas Pessoais	Alto
RF-8	Gerir Notas de Grupo	Alto
RF-9	Partilhar Nota	Alto
RF-10	Gerir Definições	Baixo
RF-11	Ver Créditos	Médio
RF-12	Sair da Aplicação	Baixo
RF-13	Gerir Imagens	Baixo
RF-14	Gerir Registo	Baixo
RF-15	Gerir Preferências	Médio

8.2. Casos de Uso

Nesta secção, é apresentado o diagrama de casos de uso, que descreve as funcionalidades proposta para a aplicação que será projetada com base. Para melhor entendimento e contextualização das funcionalidades, foi realizado um diagrama de casos de uso que está apresentado na Figura 6.

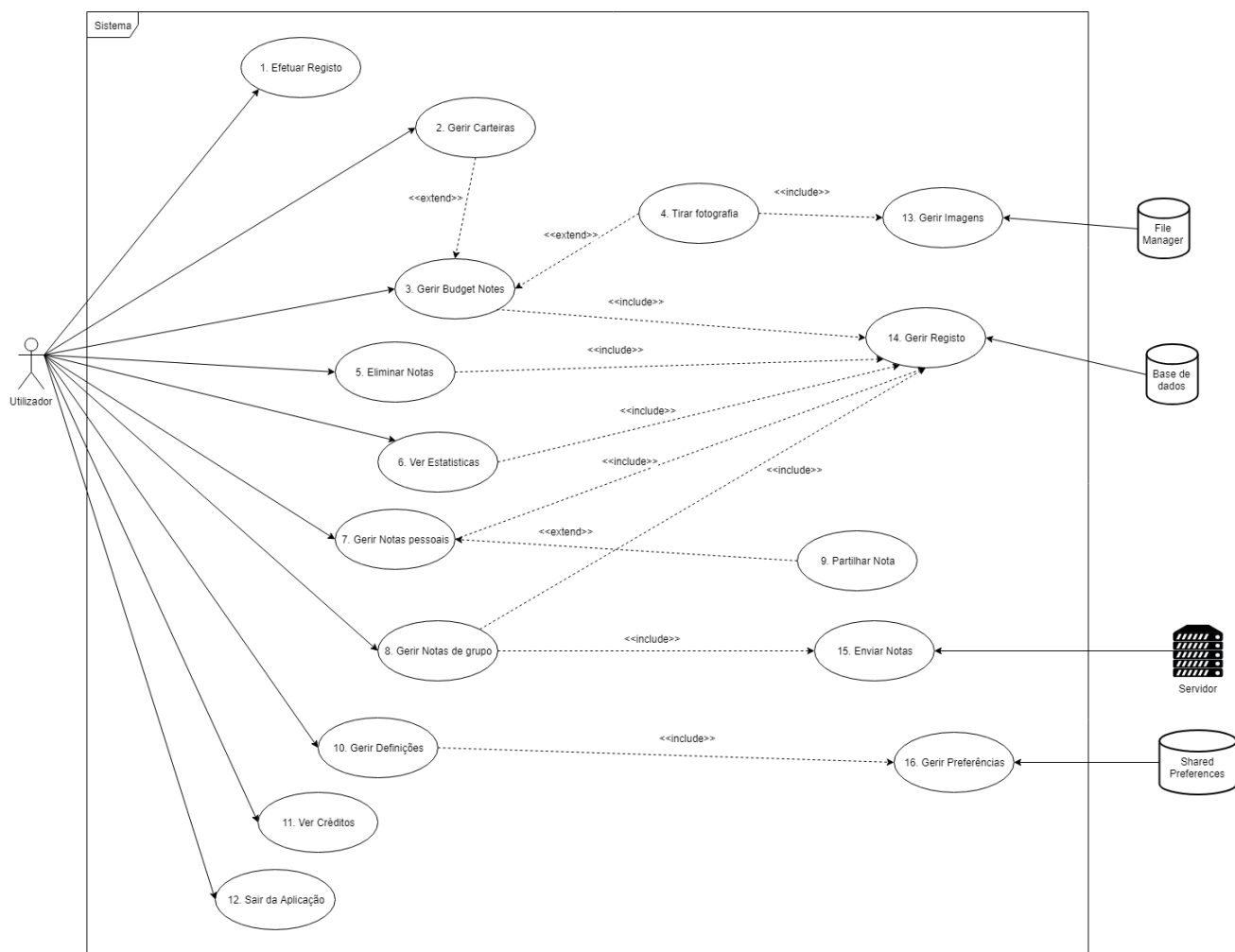


Figura 6: Diagrama de Casos de Uso

Na Tabela 7 apresenta-se uma breve descrição sobre cada um dos requisitos funcionais.

Tabela 7: Especificação detalhada dos Requisitos Funcionais

ID Requisito	Nome	Descrição
RF-1	Efetuar Registo	Como utilizador início a aplicação pela primeira vez no meu dispositivo, e forneço o meu nome de utilizador, e-mail e uma password, para que possa ser identificado em futuras ações.
RF-2	Gerir Carteiras	Como utilizador, já na página principal da aplicação, quero criar uma Wallet e que me sejam apresentadas todas as Wallets inseridas no sistema.
RF-3	Gerir Budget Notes	Como utilizador, já na página principal da aplicação, quero criar uma BudgetNote e que me sejam apresentadas todas as BudgetNotes inseridas no sistema.
RF-4	Tirar fotografia	Como utilizador, já na página de criação de uma

		BudgetNote, quero associar uma fotografia à mesma.
RF-5	Eliminar notas	Como utilizador, quando estou a visualizar todas as notas, quero poder apagar qualquer uma que tenha sido inserida.
RF-6	Ver Estatísticas	Como utilizador, quando estou na aplicação, quero poder verificar as estatísticas da mesma relativamente aos dados inseridos.
RF-7	Gerir Notas Pessoais	Como utilizador, já na aplicação, quero criar uma Nota Pessoal e que me sejam apresentadas todas as Notas Pessoais inseridas no sistema.
RF-8	Gerir Notas de Grupo	Como utilizador, já na aplicação, quero criar uma Nota de Grupo e que me sejam apresentadas todas as Notas de Grupo inseridas no sistema.
RF-9	Partilhar Nota	Como utilizador, e já com notas pessoais inseridas, quero partilhar uma determinada nota com outro utilizador, com a mesma aplicação instalada, através da geração/leitura de um QR Code..
RF-10	Gerir Definições	Como utilizador, já na aplicação, quero alterar as definições e que me sejam apresentadas todas as definições do sistema.
RF-11	Ver Créditos	Como utilizador, já na aplicação, quero visualizar os créditos da aplicação.
RF-12	Sair da Aplicação	Como utilizador, já na aplicação, quero encerrar a aplicação.
RF-13	Gerir Imagens	Como utilizador, já na página de edição de uma BudgetNote, quero alterar a fotografia à mesma, sendo para isso possível capturar novamente outra fotografia e armazená-la no dispositivo.
RF-14	Gerir Registo	Como utilizador, já na aplicação, quero alterar o meu registo no sistema e que me sejam apresentados os dados atualizados do meu registo.
RF-15	Enviar Notas	Como aplicação devo aceder ao servidor web para obter as notas do grupo no qual me insiro, e enviar as notas que o utilizador insere no grupo.
RF-16	Gerir Preferências	Como aplicação devo atualizar as minhas SharedPreferences, de forma a apresentar futuramente toda a informação correta ao utilizador.

8.3. Atributos de Qualidade

Os atributos de qualidade, ou requisitos não funcionais, são características relacionadas ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas. Estes requisitos dizem respeito a como as funcionalidades serão entregues ao utilizador do software. Normalmente estes atributos têm um forte impacto na arquitetura do sistema, visto que esta tem de ser modelada de forma a que seja dada resposta necessária a eles. De forma a perceber melhor o impacto e a importância que a cada um terá para a aplicação, estes estarão priorizados numa escala de três níveis: alta (H), média (M) e baixa (L).

Para que se consiga descrever de forma mais detalhada cada um deles, serão usados cenários, que permitem descrever as interações entre os utilizadores da aplicação e a própria aplicação. Em seguida serão apresentados os atributos de qualidade da aplicação, com a devida priorização e uma descrição.

- **Desempenho (H)** - A aplicação deve processar os pedidos do utilizador de forma rápida.
- **Disponibilidade (M)** - O servidor deve encontrar-se ativo e responder a pedidos em 99.9% do tempo total (*downtime* de 0 dias 8h:45m:35s).
- **Usabilidade (H)** - A aplicação deve ser perceptível, simples e fácil de utilizar, para que assim o utilizador consiga realizar as suas tarefas naturalmente.
- **Segurança (M)** - A aplicação deve encriptar os dados disponibilizados no *QR Code*.

Tabela 8: Cenário #1 – Desempenho

Atributo de Qualidade	Desempenho
Prioridade	Alta
Estímulo	Editar uma nota previamente criada com uma foto associada.
Fonte do Estímulo	Utilizador editar BudgetNote.
Condições do Ambiente	Condições Normais de funcionamento.
Elementos Arquiteturais	Aplicação DayliStudent e foto armazenada no dispositivo.
Resposta do sistema	Responde ao evento da forma esperada, apresentado os dados e a foto associada.
Medida da Resposta	A latência da resposta ao evento deve ser inferior a 1,5 segundos por cada edição de BudgetNote.

Tabela 9: Cenário #2 -Disponibilidade

Atributo de Qualidade	Disponibilidade
Prioridade	Média
Estímulo	O sistema e utilizadores.
Fonte do Estímulo	Acesso ao sistema back-end.
Condições do Ambiente	Condições Normais de funcionamento.
Elementos Arquiteturais	A plataforma.
Resposta do sistema	O sistema encontra-se disponível quando os utilizadores a ele tentam aceder.
Medida da Resposta	O sistema deve estar disponível anualmente 99.9% do tempo, ou seja, deve deter um <i>downtime</i> máximo de 0 dias 8h:45m:35s.

Tabela 10: Cenário #3 – Usabilidade

Atributo de Qualidade	Usabilidade
Prioridade	Alta
Estímulo	Utilizador realiza ações na aplicação de forma a utilizar qualquer uma das funcionalidades.
Fonte do Estímulo	Utilizador adicionar BudgetNote.
Condições do Ambiente	Condições normais de funcionamento.
Elementos Arquiteturais	Aplicação DayliStudent.
Resposta do sistema	O sistema responde à sequência de ações efetuadas pelo utilizador.
Medida da Resposta	O utilizador consegue aceder a qualquer funcionalidade em três ou menos cliques.

Tabela 11: Cenário #4 - Segurança

Atributo de Qualidade	Segurança
Prioridade	Média
Estímulo	Partilhar uma nota textual através de um QR Code.
Fonte do Estímulo	Utilizador partilhar uma nota textual.

Condições do Ambiente	Condições Normais de funcionamento.
Elementos Arquiteturais	Aplicação DayliStudent e nota textual armazenada no dispositivo.
Resposta do sistema	Responde ao evento da forma esperada, descriptando ou encriptando os dados recebidos ou enviados utilizando SHA-1.
Medida da Resposta	As notas transmitidas através do QR Code serão sempre encriptadas.

8.4. Restrições

Para este projeto, apenas temos duas restrições, pois apenas consideramos restrições, as decisões que não podemos alterar e as únicas que nos surgem é uma Restrições de Negócio e uma Restrição Técnica:

- **RN-1:** A versão final da aplicação tem de estar concluída a 28 de Dezembro de 2018.
- **RT-1:** A aplicação tem de ser direcionada para a API 19 ou superior do Android.

9. Mockup's

Neste capítulo são explicados em detalhe diversos percursos de navegação da aplicação e introduzidos todos os *Wireframe Diagrams* nos quais a aplicação desenvolvida foi baseada.

Quando um determinado utilizador inicializa a aplicação no seu dispositivo móvel pela primeira vez, será lhe apresentado um menu de registo de conta onde terá de preencher os devidos campos sobre os seus dados de *username*, email e *password* pretendidos.

Após preencher todos os campos e pressionar o botão de registo este será redirecionado para o ecrã principal onde encontrará um resumo de todas as suas informações relativamente aos lucros/despesas que teve no dia atual, do estado das suas carteiras e visualizar, por modo de duas listas distintas, as listas de todas as notas pessoais e partilhadas. A partir deste menu pode acessar outras funcionalidades usando os botões presentes na *app bar* ou através da barra lateral no qual pode acessar através de um simples deslizamento para a direita (*swipe right*).

Os seguintes *Wireframe Diagrams* representam o menu de registo, ecrã principal e da barra lateral:

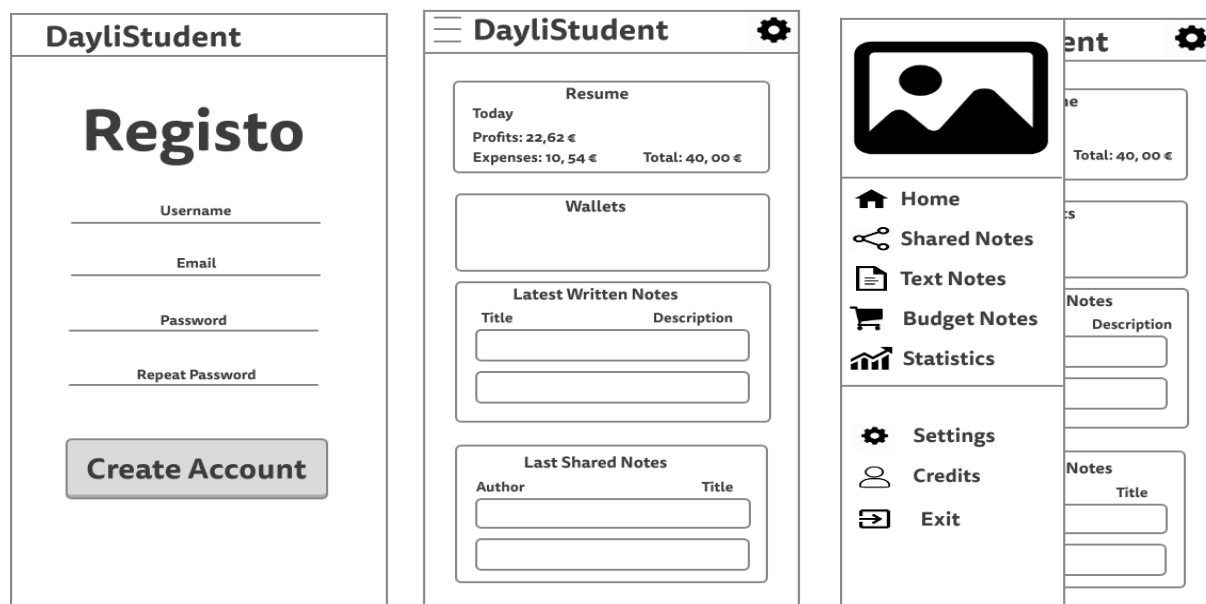


Figura 7: Menu de Registo, Principal e Barra Lateral

Ao pressionar o botão de definições na *app bar* ou na barra lateral o utilizador encontrará todo um menu de definições. Neste menu é possível alterar toda a informação pessoal como o email e username, alterar o tamanho de letra e o tema (esquema de cores) da aplicação. A seguinte imagem apresenta uma representação do menu de definições:



Figura 8: Menu de Definições

Na barra lateral, como primeira opção, o utilizador pode navegar para o menu de notas partilhadas (Shared Notes), onde lhe é apresentado um ecrã com a opção de entrar num grupo já existente ou criar um novo grupo. Para criar um novo grupo basta inserir o nome do grupo e pressionar o botão “Criar” enquanto que para entrar num grupo, o utilizador, tem de seleccioná-lo através de um dropdown menu seguido de premir o botão “Enter”. Ao juntar-se a um determinado grupo é apresentado um outro ecrã com uma lista de todas as notas partilhadas e

ao recorrer aos botões presentes na *app bar* o utilizador pode fazer *logout*, atualizar o ecrã ou criar uma nova nota partilhada.

Quando é seleccionada a opção de criação de uma nova nota é apresentado um novo ecrã, comum a criação de notas pessoais, onde apenas é necessário introduzir o título e o corpo da nota. Durante a criação de notas o utilizador pode seleccionar a nossa funcionalidade de enviar uma nota através do serviço de email presente na *app bar* da aplicação.

As seguintes imagens representam os diferentes ecrãs de grupo, ecrã de visualização de listas partilhadas e de criação de uma nova nota:

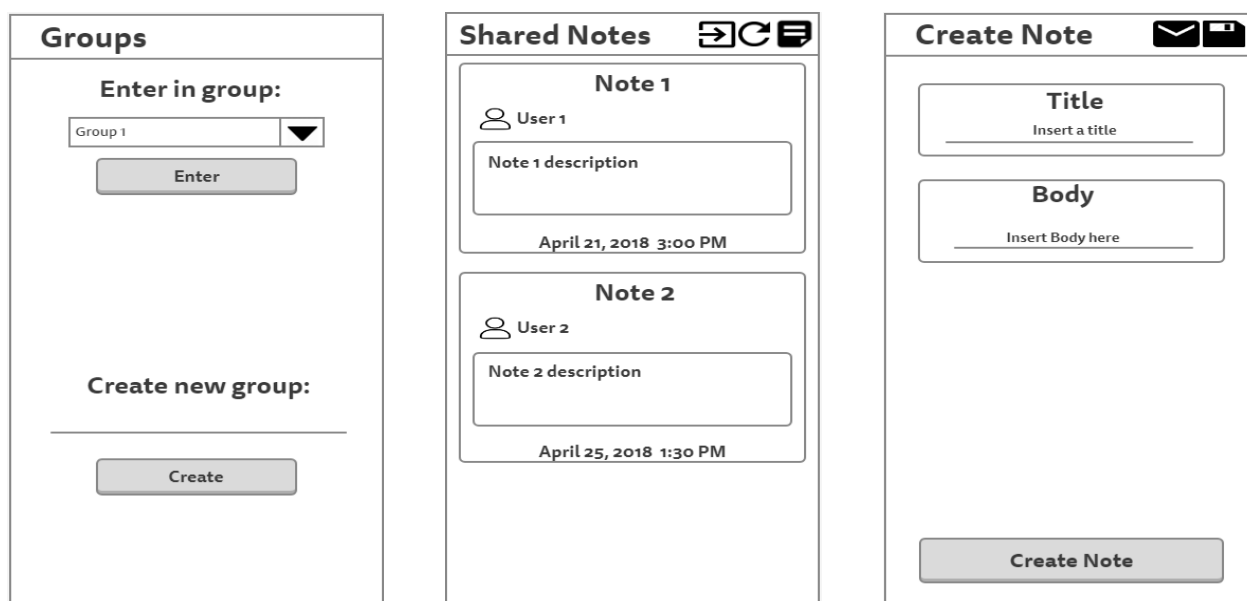


Figura 9: Menu de Grupos, Notas Partilhadas e Criação de Notas

A funcionalidade “Text Notes” presente na barra lateral do ecrã principal permite ao utilizador encontrar um ecrã com todas as suas notas privadas listadas, onde é possível editá-las, eliminá-las e partilhá-las através de um QR Code gerado automaticamente. Pode ainda usar a câmara para digitalizar um código gerado por um outro utilizador usando um botão na *app bar*.

Os seguintes *Wireframe Diagrams* representam o ecrã de listagem de notas pessoais, QR Code e uso da câmara:

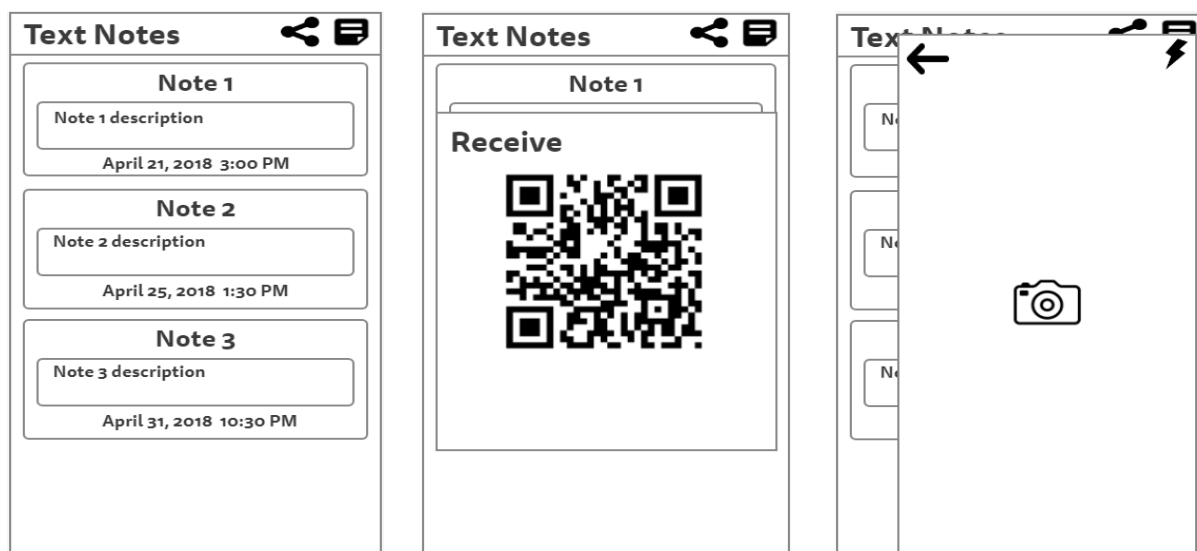


Figura 10: Menu de Notas, Partilhar Notas QR Code e Câmera

O utilizador ao seleccionar a opção “Budget Notes” na barra lateral é transferido para um ecrã onde lhe é possível visualizar o total de gastos e despesas, ver todas as notas relativamente aos seus ganhos/despesas registados/as e as suas carteiras. Usando as funcionalidades presentes na *app bar* é possível adicionar novas despesas/ganhos ou adicionar novas carteiras. Também é possível editar notas já existentes sendo o menu semelhante ao de criação da mesma e visualizar todas informações de uma carteira e suas notas associadas.

As seguintes imagens representam o ecrã de visualização geral, o menu de criação de uma carteira, menu de criação de uma Budget Note e o ecrã de visualização de informação de uma determinada carteira:

Manege Budget

Total = -45€

Movements

	Continete	35€
	Cinema	10€

Wallets

	Wallet 1	100€
	Wallet 2	250€

Create Wallet

Title
Insert a title

Wallet
Type: _____
Value: _____

Create Wallet

Create BudgetNote

Title
Insert a title

Account Characteristic
Value type:
Value: _____

Observations
Description _____

Photos

Create Note

Manage Wallets

Title
Wallet 1

Value: 20€
Type: Wallet

BudgetNotes From this Wallet:

Shopping 1	15€
Shopping 2	25€
Pizza	7€

Confirm

Figura 11: Menu Budget Notes, Criação de Carteira, Criação de uma BudgetNote e Detalhes de Carteira

Caso o utilizador aceda a funcionalidade de estatísticas, a aplicação prossegue para um ecrã onde o utilizador tem acesso a algumas informações relevantes sobre as suas despesas/ganhos, sendo elas:

- Total de gastos e ganhos
- Média de Gastos
- Gráficos com todas despesas e ganhos nos últimos dias

A seguinte imagem representa o ecrã de estatísticas:

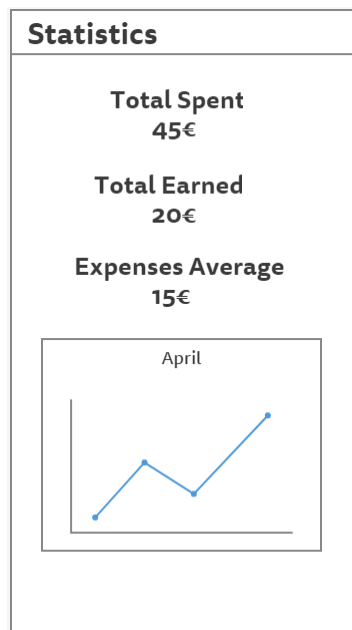


Figura 12: Ecrã de Estatísticas

A opção créditos situada na barra lateral prossegue para um ecrã onde é possível verificar os nomes dos membros de equipa responsáveis pelo desenvolvimento da aplicação, dayliStudent, e o brasão da Universidade de Coimbra.

A seguinte imagem representa o ecrã de créditos:

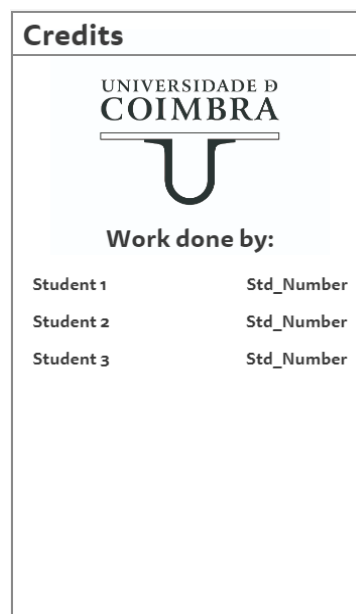


Figura 13: Ecrã de Créditos

Por fim, o utilizador pode sair da aplicação premindo a opção “Exit” que se encontra presente na barra lateral da aplicação.

10. Desenvolvimento

Neste capítulo irá ser apresentado todo o processo de desenvolvimento da aplicação e seu respetivo *back-end*. Sendo assim este capítulo estará dividido em dois subcapítulos, um para o servidor *back-end*, onde serão focados os aspetos mais técnicos relacionados com o serviço REST que comunica com a aplicação, e o outro que fala da própria aplicação móvel, onde são apresentadas mais ao pormenor todas as decisões tomadas, bem como as respetivas justificações.

10.1. Tarefas

Para cobrir o processo de desenvolvimento da aplicação, foi criado um repositório no GitHub de modo a alojar todo o código fonte, alguma documentação em formato *Markdown*^[10] e uma *board* do tipo *Kanban*^[11]. Nesta *board* foram registadas várias pistas (“*To Do*”, “*In Progress*” e “*Done*”) de forma a gerir todas as tarefas a serem realizadas durante o processo de desenvolvimento.

As *tasks* que alimentaram esta *board* provieram da definição de tarefas e devida estimação de tempo no início do projeto. À medida que cada *task* foi sendo realizada, esta foi circulando entre as pistas de modo a passar da primeira pista “*To Do*” até à terceira e última “*Done*”. As *tasks* de desenvolvimento foram divididas nas seguintes categorias:

- **Definição**
 - 01 - Statement of Work
 - 02 - Requirements specification
 - 03 - UI Prototype
 - 04 - StoryBoarding
 - 05 - Architecture Specification
- **Implementação**
 - 06 - Implement Activities.
 - 07 - Implement Fragments.
 - 08 - Implement Database.
 - 09 - Implement Sensors.
 - 10 - Implement the Back-End server.
 - 11 - Implement Parallelism.
 - 12 - Implement networking.
 - 13 - Implement Distributed Computing Mechanisms.

Na categoria da definição, englobamos todas as *tasks* relativas à definição de todo o *background* da aplicação, enquanto que na categoria de implementação, enquadramos todas as *tasks* relativas ao desenvolvimento técnico da mesma.

Para o desenvolvimento da aplicação *Front-End* foi escolhido o desenvolvimento nativo em Android, utilizando as linguagens de programação Java e Kotlin, com recurso ao Android Studio como ambiente de desenvolvimento. Por outro lado, para o desenvolvimento da aplicação *Back-End* foi escolhido a implementação de um servidor em *Flask*, utilizando a linguagem de programação Python, com recurso ao PyCharm como ambiente de desenvolvimento.

Cada uma das *tasks* referidas na implementação deu origem a uma branch, onde foi implementada a devida funcionalidade. Após a funcionalidade se encontrar implementada, a branch era fechada com um “*Pull request*”, o qual requer aprovação de todos os membros do grupo. Após a branch ser fechada era feito o *merge* automaticamente para a branch *master*. Esta abordagem permitiu a independência no desenvolvimento e a isolamento de funcionalidades.

Após o término do processo de desenvolvimento foi atribuída uma *tag* que referia a versão 1.0 da aplicação e onde é disponibilizado o APK para *download*.

10.2. Aplicação Móvel

Antes de qualquer desenvolvimento da aplicação móvel, tivemos que tomar algumas decisões técnicas, como por exemplo onde é que iríamos aplicar todas as características exigidas pelo cliente. As principais foram onde é que seriam aplicados fragmentos e os sensores. Para os fragmentos decidimos optar por colocar fragmentos na leitura do QR Code por parte da câmara, pois iria tornar a aplicação mais modular em termos dessa funcionalidade. No que toca aos sensores decidimos optar por integrar o sensor de internet para receber e enviar notas, pois achamos que seria uma funcionalidade essencial no dia-a-dia, quer de um trabalhador quer de um estudante.

Começando a aplicação, decidimos desenvolver um ecrã principal focado na disponibilização dos dados inseridos pelo utilizador na aplicação. Com esta abordagem permitimos que o utilizador tenha uma visão geral dos dados da aplicação sem que tenha que ir a cada categoria/funcionalidade verificar os mesmos. Os restantes ecrãs da aplicação são implementados como *Activities*.

Para a acessibilidade de funcionalidades optamos pela implementação de um menu *Navigation Drawer*, pois trata-se de um menu muito utilizado atualmente, que torna o ecrã principal mais limpo e as funcionalidades acessíveis num só local.

Depois do ecrã principal implementado passamos para o layout geral das atividades das funcionalidades que a aplicação iria ter. Com isto optámos por colocar como base, uma *App Bar* em cada uma que permitisse aceder a outras funcionalidades dentro da mesma, de forma a que o utilizador tivesse todas as funcionalidades possíveis dentro a atividade em que se encontrava, acessíveis no menu superior. Este menu superior variava um pouco de funcionalidade para funcionalidade, pois na funcionalidade de criar notas pessoais, o mesmo continha apenas as funcionalidades de adicionar uma nova nota e receber uma nota, enquanto que na funcionalidade de criar *budget notes* continha a opção de criar uma nova carteira e uma nova *budget note*. No geral este menu superior continha e média 2/3 botões, pois achamos que seria o limite ideal para o utilizador e para que o menu superior não ficasse sobrecarregado de ícones.

Como a maioria das *Activities* têm uma listagem de notas, quer sejam elas *budget* quer sejam elas textuais, optamos por aplicar uma *ListView* e uma *RecyclerView*, nas *budget notes* e nas notas de grupo respetivamente. Esta opção de utilizar dois mecanismos diferentes, para uma funcionalidade praticamente idêntica recaiu sobre a abrangência de conhecimentos no desenvolvimento da aplicação. Com isto o grupo conseguiu obter/aplicar os conhecimentos dos mecanismos aplicados.

A funcionalidade de tirar *budget notes* terá a funcionalidade de, sempre que necessário, tirar uma foto a qualquer registo importante para uma determinada nota, pois por vezes deparamos nos com a falta de tempo/paciência para discriminar os dados todos numa nota, e assim torna-se possível que o utilizador discrimine apenas os dados mais importantes e que os restantes sejam obtidos através de uma imagem. Essa imagem será possível de obter dentro da atividade de criação de uma *budget note* com o recurso à Câmara do dispositivo. Depois de tirada a imagem, esta será associada ao registo da nota e poderá ser acedida através do *File Manager* ou da galeria do dispositivo, caso o utilizador queira ver a imagem com mais detalhe. A câmara é aplicada ainda noutra funcionalidade, a partilha de notas pessoais. Ela integra-se quando um utilizador deseja receber uma nota pessoal de outro utilizador com a mesma aplicação instalada. Para que isso seja possível o utilizador seleciona a funcionalidade de receber uma nota pessoal, e a aplicação lança um *pop-up* com o um *Fragment* integrado, onde é possível visualizar a Câmara e

assim conseguir apontar a mesma para um QR Code gerado por outra aplicação DayliStudent. Depois disto a Câmera consegue detetar o conteúdo do respetivo QR Code e insere-o na base de dados da aplicação para futura visualização da mesma. O QR Code gerado e lido é realizado quando um utilizador cria uma nota pessoal e deseja que a mesma seja partilhada com outro utilizador, sendo que isso só será possível quando a nota estiver criada e o mesmo efetuar um *long press click* na respetiva, selecionando posteriormente a opção de partilha. Todo o conteúdo contido o QR Code gerado encontra-se encriptado para maior segurança dos dados da aplicação e para que a mesma não seja lida por outras aplicações.

Posto isto passamos para a implementação do armazenamento dos dados, em que passou pela integração de uma Base de dados, em SQLite, e de *SharedPreferences*. A base de dados serviu para armazenar tudo o que fosse referente às *budget notes* e notas pessoais, cada uma com os seus parâmetros e na sua devida tabela. Como era necessário apresentar os dados da base de dados em listas, utilizou-se o *Cursor Adapter* para que assim conseguíssemos ter um ponteiro que conseguisse controlar o que era necessário visualizar e em que tabela é que queríamos trabalhar. Quanto às *SharedPreferences* serviu para o armazenamento de dados relativos às definições do utilizador, como por exemplo o nome de utilizador, email e tema aplicado à aplicação.

Como é possível verificar apenas são guardadas na base de dados local as *budget notes* e as notas pessoais, pois as notas partilhadas estão armazenadas num servidor *web*. Este servidor é contactado pela aplicação através de pedidos HTTP, para que as nota/grupos sejam acessíveis na aplicação, recorrendo à biblioteca Retrofit. Essa decisão baseou-se no facto de a biblioteca apresentar uma abordagem que reduz bastante a lógica necessária para lidar com estes pedidos. Nestes pedidos englobamos algumas tecnologias que são necessárias para a correta implementação da comunicação da aplicação, como é o caso das *threads*, de objetos JSON e GSON.

Por fim, no que toca a notificações da aplicação na barra de dispositivos, decidimos implementar uma notificação sempre que o utilizador recebe uma nota partilhada, de forma a que o utilizador saiba sempre de quem, e que nota é que recebeu do QR Code.

Passando por estes vários pontos e especificações, chegamos ao fim do desenvolvimento da aplicação, sendo que alguns ecrãs da aplicação final podem ser vistos nas figuras abaixo mencionadas, e os restantes em anexo.

Figura 14: Layouts da aplicação

No final, e como dito anteriormente, foi gerada uma release da aplicação em versão 1.0, estando a mesma disponibilizada no seguinte [link](#).

10.3. Servidor

De igual modo, para o servidor, antes de se realizar qualquer implementação, foi tomado em consideração o que se iria utilizar como tecnologias. Visto que a necessidade era um simples servidor com operações CRUD (Create, Read, Update e Delete) sobre as notas globais, decidimos optar por realizar uma implementação com recurso a Flask, uma *framework* disponível para Python, que permite a exposição de endpoints com recursos do lado do servidor. Para o armazenamento de dados no servidor, foi utilizada uma base de dados, em *SQLite*, similar à existente na aplicação para persistir as notas globais.

Após implementação total do servidor, foi feito o *deploy* do mesmo para uma instância grátis que se encontra a correr na *Google Cloud Engine*. Foi também agregado um mecanismo de logs, de modo a monitorizar e a detetar falhas no próprio servidor.

11. Testes

Para que a aplicação garantisse os requisitos funcionais, para além dos testes habituais de um programador feitos durante o desenvolvimento, foram feitos testes de aceitação pelo grupo e por pessoas externas ao mesmo. Para isso foi realizado um plano de testes, apresentado na Tabela 12, que abrange todos os requisitos funcionais.

Tabela 12: Plano de Testes

Plano de testes			
Teste	ID Requisito Funcional	Descrição do teste	Resultado esperado
T-01	RF-01	O utilizador preenche todos os campos do registo.	Dados do utilizador registados nas <i>SharedPreferences</i> e apresentados no <i>Navigation Drawer</i> .
T-02	RF-01	O utilizador não preenche um dos campos do registo.	A aplicação apresenta uma mensagem de erro ao utilizador a informar que todos os campos devem ser preenchidos.
T-03	RF-01	O utilizador insere um email inválido.	A aplicação apresenta uma mensagem de erro ao utilizador a informar que o email inserido não cumpre com as normas.
T-04	RF-02	O utilizador preenche todos os campos de criação de uma carteira.	Criação e visualização da carteira, na listagem de carteiras e o montante atual da aplicação é atualizado.
T-05	RF-02	O utilizador não insere o nome da carteira a criar.	A aplicação apresenta uma mensagem de erro ao utilizador a informar que todos os

			campos devem ser preenchidos.
T-06	RF-02	O utilizador tenta inserir um montante inválido na carteira.	A aplicação não permite a inserção de montantes inválidos ou de caracteres que não sejam números.
T-07	RF-02	O utilizador tenta visualizar todas as notas associadas a uma carteira.	A aplicação apresenta uma listagem de todas as notas associadas à carteira.
T-08	RF-02	O utilizador apaga uma carteira.	A aplicação elimina carteira da listagem de carteiras.
T-09	RF-03	O utilizador preenche os dados todos do registo de uma <i>budget note</i> .	A aplicação insere a <i>budget note</i> , e atualiza a listagem das mesmas, bem como o montante atual da aplicação.
T-10	RF-03	O utilizador tenta inserir uma <i>budget note</i> sem uma carteira associada.	A aplicação apresenta uma mensagem de erro ao utilizador a informar que para completar a operação tem que ter associada uma carteira previamente criada.
T-11	RF-03	O utilizador tenta inserir um valor inválido na <i>budget note</i> .	A aplicação apresenta uma mensagem de erro ao utilizador a informar que o valor inserido não é válido.
T-12	RF-04	O utilizador tenta inserir uma <i>budget note</i> com uma imagem associada.	A aplicação cria a <i>budget note</i> e armazena/associa a imagem.
T-13	RF-04	O utilizador tenta visualizar uma imagem de uma <i>budget note</i> que foi previamente apagada.	A aplicação apresenta uma mensagem de erro a informar o utilizador de que a imagem já não se encontra disponível no sistema.
T-14	RF-05	O utilizador realiza um <i>long press click</i> para eliminar uma nota.	A aplicação elimina a nota e atualiza a respetiva listagem de notas.
T-15	RF-06	O utilizador depois de inserir <i>budget notes</i> , visualiza as estatísticas da app.	A aplicação apresenta as estatísticas baseada nos valores inseridos pelo utilizador.
T-16	RF-07	O utilizador preenche todos os campos de criação e guarda a nota pessoal.	Criação e visualização da nota pessoal, na listagem de notas pessoais.
T-17	RF-07	O utilizador não insere o	A aplicação apresenta uma mensagem de

		título da nota a criar.	erro ao utilizador a informar que todos os campos devem ser preenchidos.
T-18	RF-07	O utilizador seleciona a opção de enviar a nota pessoal por email.	A aplicação apresenta uma listagem de clientes email instalados nos dispositivos e com os quais é possível continuar a operação, apresentado posteriormente o título e assunto da nota, como assunto e corpo do email respetivamente.
T-19	RF-07	O utilizador apaga uma nota pessoal.	A aplicação elimina nota pessoal e atualiza a listagem de notas pessoais.
T-20	RF-07	O utilizador seleciona a partilha de uma nota pessoal.	A aplicação gera um QR Code para a partilha da mesma com outra aplicação do mesmo tipo.
T-21	RF-07	O utilizador seleciona a receção de uma nota pessoal.	A aplicação gera um <i>pop-up</i> com a câmara, para a leitura de um QR Code gerado por outra aplicação do mesmo tipo.
T-22	RF-07	O utilizador aponta a câmara do dispositivo para o ecrã de outro dispositivo, depois de ter selecionado a opção de receção de nota pessoal.	A aplicação efetua a leitura do QR Code e insere a respetiva nota partilhada na base de dados.
T-23	RF-08	O utilizador inicia a funcionalidade de gerir notas de grupo sem previamente ter se inserido num grupo.	A aplicação apresenta um ecrã com a possibilidade de se inserir num grupo previamente existente ou então de criar um grupo novo.
T-24	RF-08	O utilizador seleciona um grupo para se inserir.	A aplicação apresenta um novo ecrã com uma listagem das notas do grupo previamente selecionado.
T-25	RF-08	O utilizador insere um nome de um grupo para adicionar.	Criação e visualização do novo grupo na listagem.
T-26	RF-10	O utilizador acede às definições para alterar o tema da aplicação.	A aplicação altera as suas cores para as cores do tema selecionado pelo utilizador.
T-27	RF-10	O utilizador acede às definições para alterar o seu nome e email.	A aplicação altera os dados registados do utilizador para os inseridos pelo mesmo, e apresenta-os atualizados no <i>Navigation Drawer</i> .
T-28	RF-11	O utilizador acede à página dos créditos.	A aplicação apresenta os dados relativos ao projeto e aos criadores do mesmo.

T-29	RF-12	O utilizador seleciona sair da aplicação.	A aplicação encerra sem que os dados da mesma sejam removidos ou alterados.
------	-------	---	---

Depois de definido o plano de testes, foi reunido um grupo de amigos, ao qual foi dado o mesmo, e foi pedido que testassem a aplicação e fornecessem o *feedback* ao nível de possíveis anomalias que encontrassem. Deste modo, e depois do conjunto de testes ter sido realizado, obtivemos os resultados de cada teste, ou seja, se o teste correu como o esperado. Como podemos observar na Tabela 13, verificamos que os testes correram todos consoante o previsto. Este resultado era o esperado uma vez que durante o desenvolvimento iam sendo testadas as várias funcionalidades individualmente consoante a sua implementação na aplicação.

Tabela 13: Resultados dos testes

Resultados dos testes	
ID do teste	Resultado Correto
T-01	Sim
T-02	Sim
T-03	Sim
T-04	Sim
T-05	Sim
T-06	Sim
T-07	Sim
T-08	Sim
T-09	Sim
T-10	Sim
T-11	Sim
T-12	Sim
T-13	Sim
T-14	Sim
T-15	Sim
T-16	Sim
T-17	Sim

T-18	Sim
T-19	Sim
T-20	Sim
T-21	Sim
T-22	Sim
T-23	Sim
T-24	Sim
T-25	Sim
T-26	Sim
T-27	Sim
T-28	Sim
T-29	Sim

12. Referências

- [1] <https://www.dn.pt/ciencia/tecnologia/interior/ha-mais-dispositivos-moveis-ativos-que-pessoas-na-terra-4166852.html>
- [2] <https://www.tudocelular.com/economia-e-mercado/noticias/n27906/mercado-apps-saturado.html>
- [3] <http://www.businessofapps.com/data/app-statistics/>
- [4] <https://www.lifehack.org/articles/technology/best-15-money-management-apps-that-make-financial-planning-easy.html>
- [5] <https://play.google.com/store/apps/details?id=com.mint&hl=en>
- [6] <https://play.google.com/store/apps/details?id=com.cleevio.spendee>
- [7] <https://cloud.google.com/compute/docs/gcloud-compute/>
- [8] https://pt.wikipedia.org/wiki/Desenvolvimento_%C3%A1gil_de_software
- [9] [https://pt.wikipedia.org/wiki/Scrum_\(desenvolvimento_de_software\)](https://pt.wikipedia.org/wiki/Scrum_(desenvolvimento_de_software))
- [10] <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>
- [11] <https://github.com/kanboard/kanboard>