



ISEL

INSTITUTO SUPERIOR
DE ENGENHARIA DE LISBOA

Desenvolvimento de Aplicações Móveis

WALLET APP

Docente: Ana Correia

20 de junho de 2022

Devolvido por:

Ana Oliveira| A39275

Índice

Desenvolvimento de Aplicações Móveis.....	0
Índice.....	2
Índice de Imagens	3
Introdução.....	5
Desenvolvimento	6
1. Login e registo	7
2. Registo de Despesas e Receitas.....	9
3. Apresentação geral das economias.....	11
4. Gestão do Perfil	12
Wire-frames.....	15
Mock-ups	17
Base de Dados (Realtime Database).....	18
Diagrama UML	21
Produto Final.....	22
Conclusões	25
Bibliografia	26

Índice de Imagens

Figura 1 - Arquitetura Geral.....	6
Figura 2 - Diagramas de interação com Firebase - Registo e Login.....	8
Figura 3 - Realtime Database- Utilizadores.....	8
Figura 4 - Realtime Database- Movimentação	10
Figura 5 - Diagrama de interação com Firebase - Registo de movimentos	10
Figura 6 - Diagrama de interação com Firebase – Alterar nome e estado da subscrição	12
Figura 7 - Diagrama de interação com Firebase – Alterar foto de Perfil.....	13
Figura 8 - Notificação Agendada.....	14
Figura 9 - Notificação.....	14
Figura 10 – Wire-frames.....	16
Figura 11 - Mock-ups.....	17
Figura 12 - Estrutura principal da aplicação	18
Figura 13 - Guardar Utilizador na base de dados.....	19
Figura 14 - Guardar Movimento da base de dados.....	19
Figura 15 – Realtime Database – Visão geral	20
Figura 16 - Diagrama UML	21
Figura 17 - Wallet App Icon Launch.....	22
Figura 18 - Ecrãs de introdução Wallet App.....	22
Figura 19 - Wallet App - Registrar/Login	23

Figura 20 - Wallet App - Registrar	23
Figura 21 - Wallet App - Login	23
Figura 22 - Wallet App - Ecrã inicial.....	23
Figura 23 - Wallet App - Adicionar Despesa	23
Figura 24 - Wallet App - Adicionar Receita.....	23
Figura 25 - Wallet App - Perfil	24
Figura 26 - Wallet App - Ecrã inicial com despesas e receitas	24
Figura 27 - Wallet App - notificação Agendada.....	24
Figura 28 - Wallet App - Notificação	24

Introdução

As aplicações móveis consistem em software desenvolvido para dispositivos móveis, tais como os telemóveis, *tablets*, *smart Tvs*, entre outros. Estas aplicações surgiram como resposta ao mercado emergente dos telemóveis, no entanto com o aumento do mercado dos restantes produtos móveis ganharam, cada vez mais, maior popularidade.

No âmbito da unidade curricular Desenvolvimento de Aplicações Móveis foi proposta a criação e desenvolvimento de uma aplicação para o sistema operativo Android, centrada na experiência de utilizador e baseada num plano de negócio.

A aplicação Wallet permite que um utilizador gira as suas economias através do registo de despesas e rendimentos diários. A aplicação irá apresentar uma listagem de despesas e rendimentos para um determinado mês apresentando ao utilizador o saldo geral da conta.

No decorrer deste relatório serão detalhadas todas as fases de desenvolvimento desta aplicação, bem como apresentados todos os menus e esquemas de relação necessários para garantir a sua implementação.

O código fonte da aplicação apresentada está disponível no GitHub, através no seguinte link: <https://github.com/AnaSofOliveira/Android-Wallet-APP>. O produto final está disponível na Google Play Store e pode ser descarregado a partir do link <https://play.google.com/store/apps/details?id=com.a39275.wallet>, brevemente disponível.

Desenvolvimento

O processo de desenvolvimento foi iniciado pela definição das diversas funcionalidades pretendidas para o funcionamento da aplicação Wallet. Assim, o projeto foi subdividido nas seguintes funcionalidades:

- Login e registo;
- Registo de despesas e receitas;
- Apresentação geral das economias;
- Gestão do perfil;
- Agendamento de notificações.

Para que a informação de cada utilizador fosse persistente entre sessões, foi necessário interligar o sistema ao *Firebase*. Para isso, foi também necessário construir a arquitetura do projeto tornando evidente a comunicação entre diversos módulos do *Firebase*.

A figura 1 ilustra a arquitetura geral do projeto Wallet. Mais à frente irei desenvolver a comunicação entre cada uma das partes do sistema.

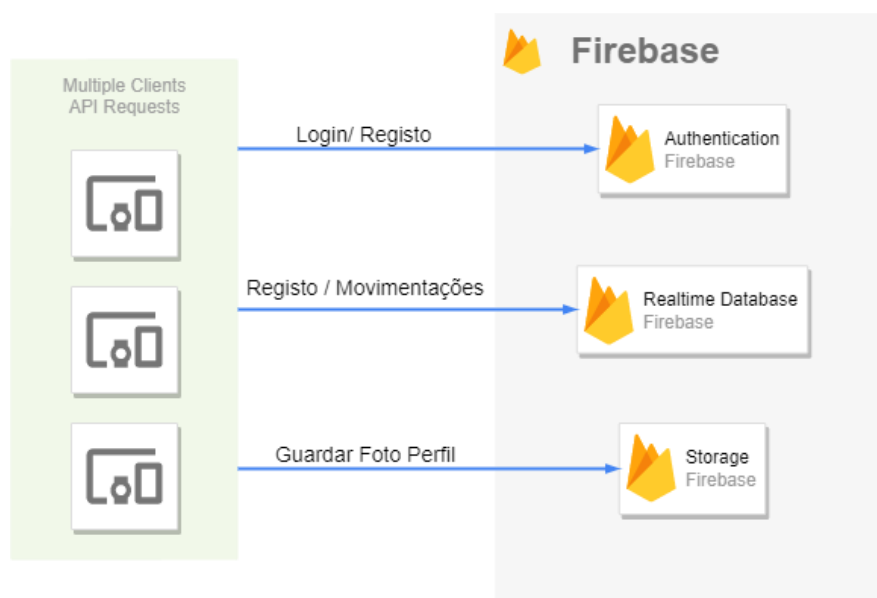


Figura 1 - Arquitetura Geral

1. Login e registo

O login e registo foi realizado em conjunto com a autenticação e a base de dados do *Firebase*. Sempre que um utilizador se regista, o *Firebase* gera um UID, código único, para cada utilizador. Através desse código é realizado o registo de toda a informação do utilizador no *Realtime Database* do *Firebase*.

Para cada utilizador são guardados os seguintes dados:

- Nome;
- E-mail;
- Localização da foto de perfil;
- Total Despesa;
- Total Receita;
- Estado subscrição premium.

No processo de registo o total de despesa e total de receita são €0.00, o estado da subscrição é falso e a localização da foto de perfil está vazia. Mais tarde, irei detalhar o processo para guardar a foto de perfil que irá editar o campo "*srcFoto*" no *Realtime Database* do *Firebase*.

Na figura 2, são apresentadas as interações necessárias para o registo e login do utilizador na Wallet APP.

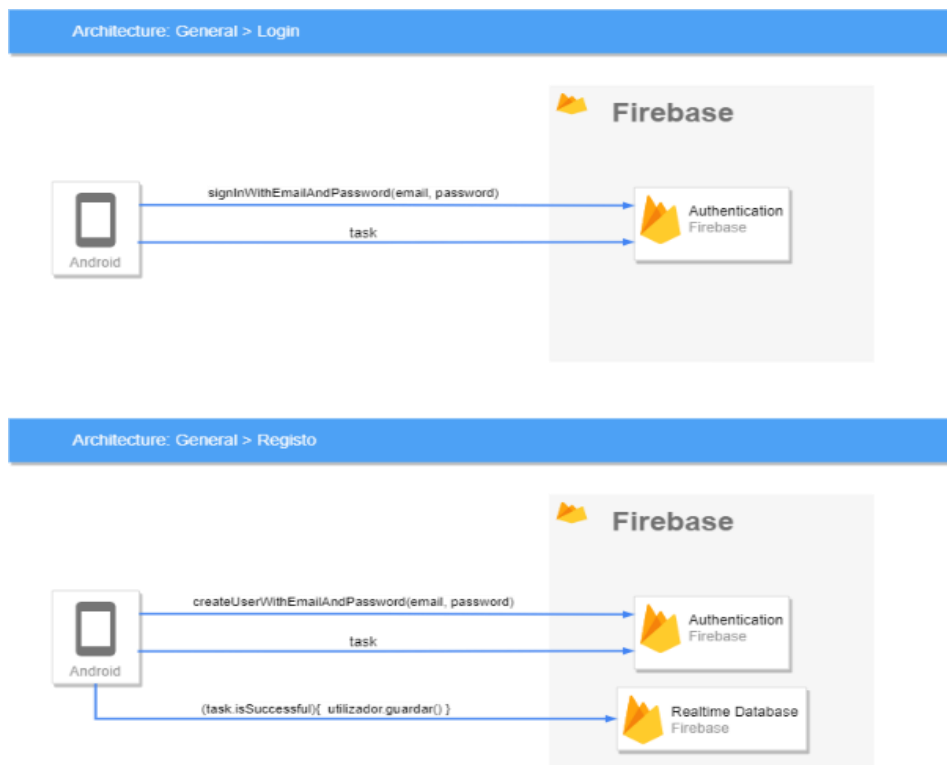


Figura 2 - Diagramas de interação com Firebase - Registo e Login

A estrutura armazenada no *Realtime Database* do *Firebase* é a apresentada na figura 3.

```

"utilizadores": {
  "ko8aWPNHI2T3qEeU2I4gGFdXBsO2": {
    "despesaTotal": 0,
    "email": "ana.as.ac@gmail.com",
    "nome": "Ana Sofia Oliveira",
    "premium": false,
    "receitaTotal": 0,
    "srcFoto": ""
  }
}

```

Figura 3 - Realtime Database- Utilizadores

2. Registo de Despesas e Receitas

Cada despesa ou receita é uma movimentação. Assim, é possível guardar informação relativa a cada movimento de forma genérica, mantendo apenas a informação específica tipo da movimentação. Desta forma, a responsabilidade no cálculo é requisito aplicacional.

Para cada movimento são guardados os seguintes dados:

- Categoria;
- Data;
- Descrição;
- Tipo (r – receita ou d – despesa);
- Valor.

Cada utilizador pode realizar vários movimentos. Movimentos esses que, após terem sido adicionados serão apresentados ao utilizador num resumo mensal. Assim, foi também necessário relacionar cada movimento, não só com o utilizador que realizou o movimento, mas também com o mês e o ano em que aquele movimento foi realizado. Desta forma, cada movimento fica registado no *Realtime Database* do *Firebase* conforme está apresentado na figura 4.

```

"movimentacao": {
  "ko8aWPNHI2T3qEeU2I4gGFdXBsO2": {
    "062022": {
      "-N4nCXs4Xw_SGXTb28Gs": {
        "categoria": "Salario",
        "data": "17/06/2022",
        "descricao": "Salario",
        "tipo": "r",
        "valor": 1000
      },
      "-N4nCaEwQ6y3RRYfeYKo": {
        "categoria": "Lazer",
        "data": "17/06/2022",
        "descricao": "Jantar Fora",
        "tipo": "d",
        "valor": 25
      }
    }
  }
}

```

Figura 4 - Realtime Database- Movimentação

Na figura 5 são apresentadas as interações necessárias para o registo dos movimentos no *Realtime Database* do *Firebase*.

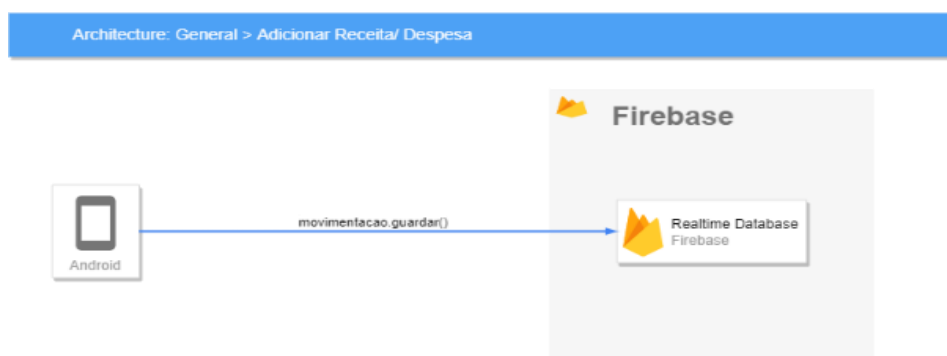


Figura 5 - Diagrama de interação com Firebase - Registo de movimentos

3. Apresentação geral das economias

Nesta fase, pretende-se que o utilizador consiga visualizar um resumo mensal das suas economias: saldo atual, lista de despesas e lista de receitas. Assim, foi construída uma interface gráfica que permitisse ao utilizador ter toda esta informação visível.

Esta interface contém a informação do saldo, calculado a partir da diferença entre o total de receitas e o total de despesas, registado dos dados do utilizador (Figura 3). Contém também um *Material Calendar View* com o mês selecionado e uma lista de movimentos obtida através do *AdapterMovimentacao*. O *AdapterMovimentacao* é responsável por, a partir de uma lista de movimentos, construir uma apresentação dinâmica de todos os movimentos dentro de um *RecyclerView*.

A interface permite ainda que o utilizador faça *swipe right* ou *swipe left* para avançar ou recuar no mês de consulta, respetivamente, atualizando as movimentações para o mês corrente na interface.

4. Gestão do Perfil

A gestão do perfil do utilizado é efetuado através do ecrã de Perfil. Nesta interface, o utilizador conseguirá atualizar a foto, o nome, o estado da sua subscrição ou terminar sessão. O campo e-mail também é apresentado ao utilizador, porém não é possível realizar a sua edição e alteração por decisão de negócio.

A atualização do nome é realizada através da alteração do campo textual e posterior *click* no botão "Guardar Perfil". Com isto, a aplicação irá alterar o *profile* da autenticação do *Firebase* e modificar o campo "nome" do utilizador no *Realtime Database* do *Firebase*, conforme apresentado na figura 6.

A atualização do estado da subscrição é idêntica à alteração do nome, sendo apenas necessário carregar no Premium *switch* e fazer *click* no botão "Guardar Perfil".



Figura 6 - Diagrama de interação com Firebase – Alterar nome e estado da subscrição

Em relação à alteração da imagem de perfil, ao carregar no botão "Alterar foto" irá ser aberta a galeria do dispositivo móvel, onde o utilizador poderá

selecionar a foto que pretende carregar. Após ter selecionada uma foto, é realizado o upload da imagem para o *Storage* do *Firebase*. Caso o upload decorra normalmente, é solicitado ao *Firebase* o URL da imagem, que será posteriormente guardado no *Realtime Database* do *Firebase*, no campo "*srcFoto*" dos dados do utilizador. A figura 7 ilustra todo este processo.

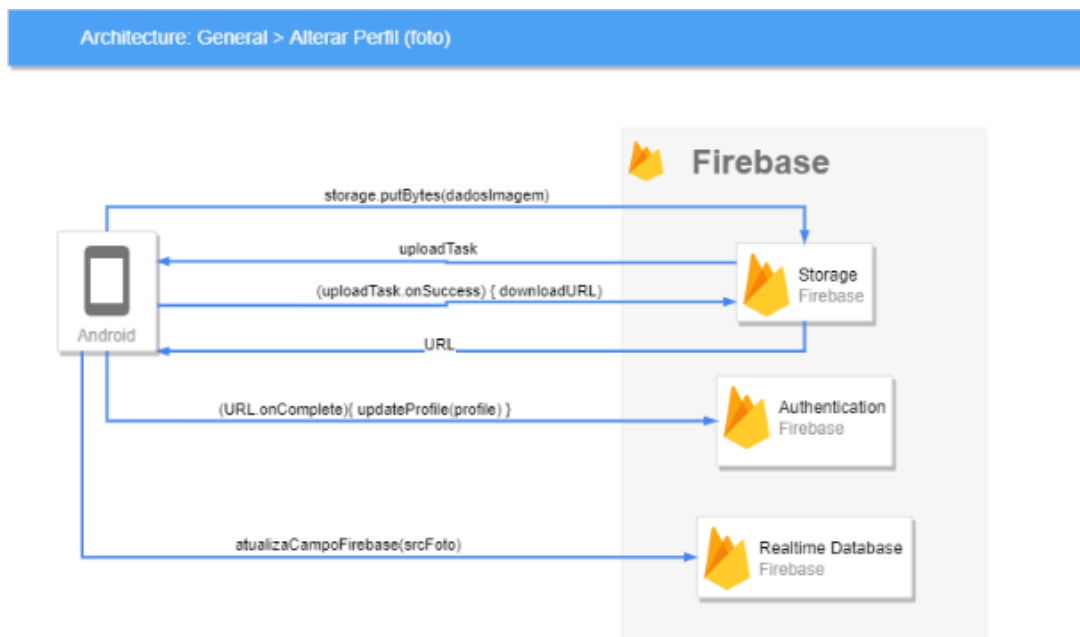


Figura 7 - Diagrama de interação com Firebase – Alterar foto de Perfil

5. Agendamento de notificações

O agendamento de notificações é uma funcionalidade premium sendo apenas possível realizar caso o utilizador tenha subscrito esta funcionalidade.

Para realizar o agendamento de notificações o utilizador necessita de introduzir a categoria, a descrição, a data e hora de alerta da descrição. Caso o utilizador programe a data de alerta para uma data e hora passadas, a notificação será imediatamente entregue. Caso contrário, a notificação será entregue na data programada independentemente de o utilizador estar ou não a utilizar a aplicação.

Ao carregar no alerta despoletado, o utilizador será encaminhado para o ecrã de gestão das economias onde poderá registar a sua despesa ou receita.

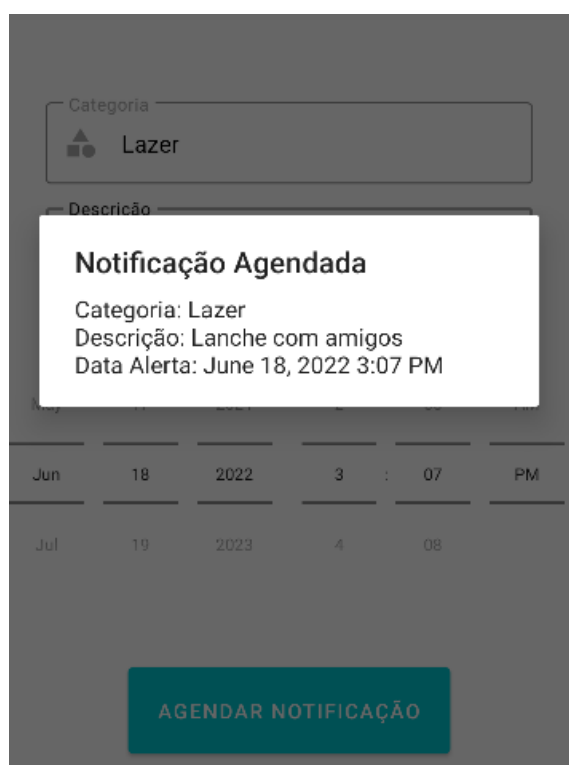
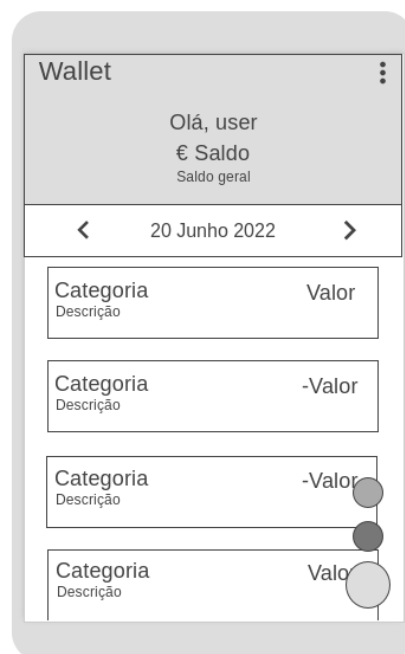
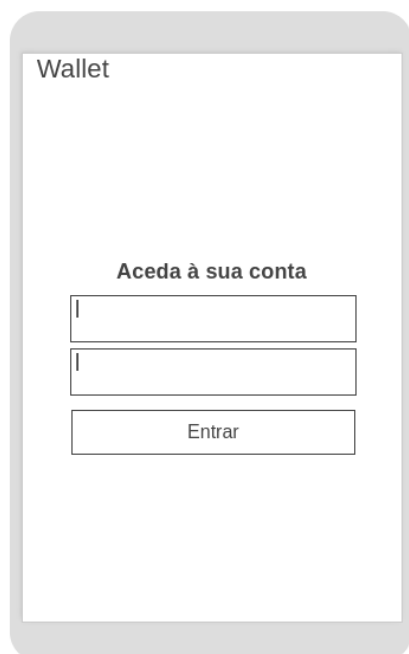
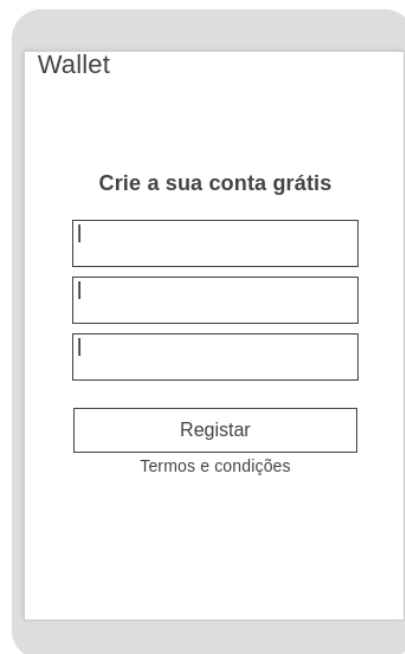
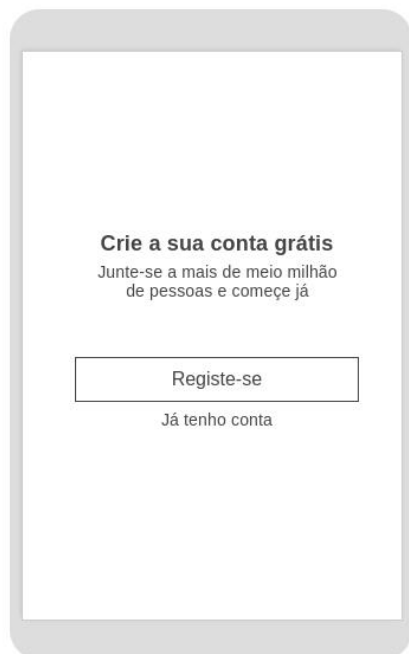


Figura 8 - Notificação Agendada



Figura 9 - Notificação

Wire-frames



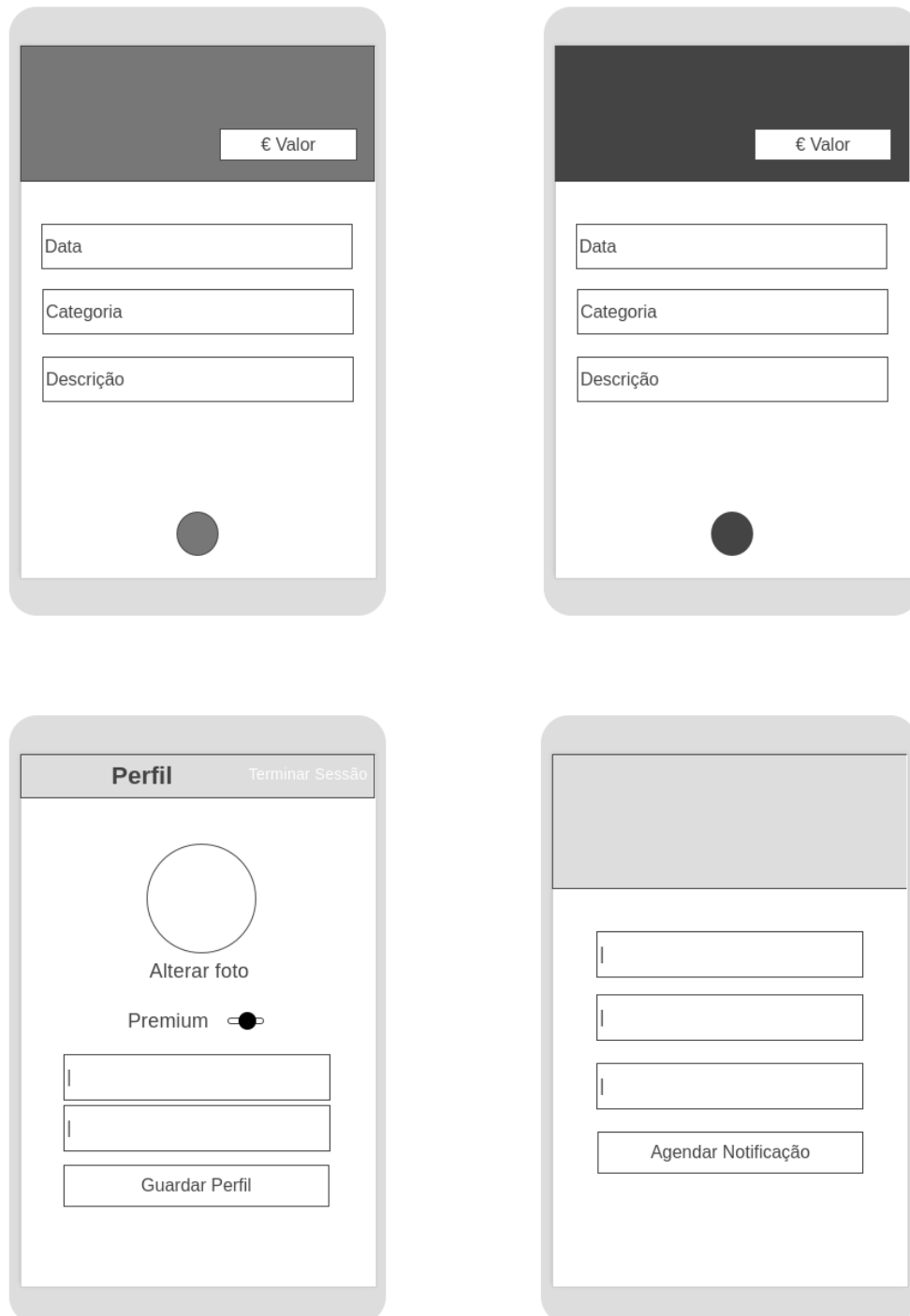


Figura 10 – Wire-frames

Mock-ups

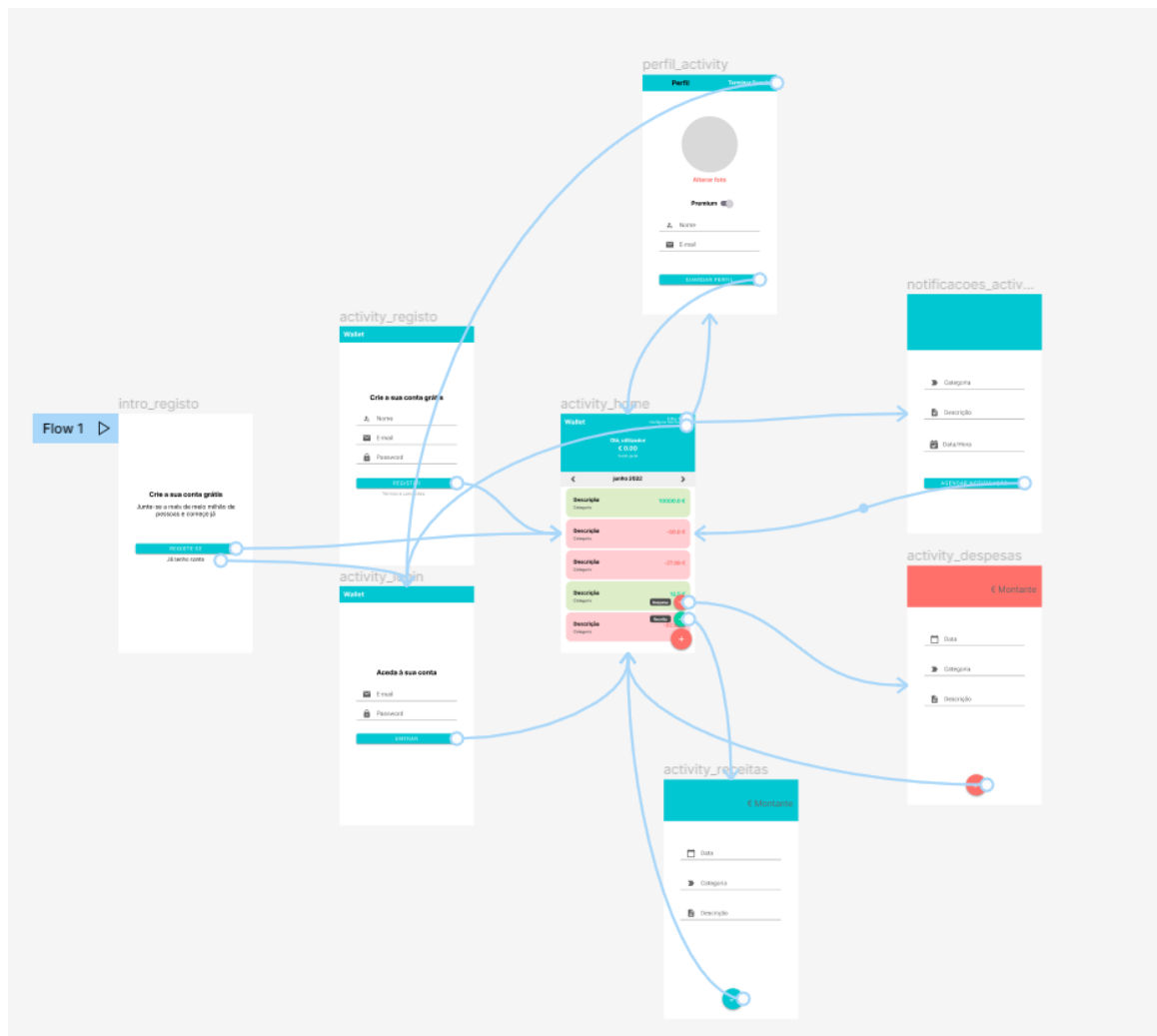


Figura 11 - Mock-ups

Base de Dados (Realtime Database)

Todas as informações necessárias à aplicação são armazenadas no *Realtime Database* do *Firebase*. O *Realtime Database* é uma base de dados *NoSQL*, que armazena e sincroniza os dados em tempo real partilhando-os entre todos os utilizadores conectados na aplicação. Esta base de dados está estruturada em formato JSON, permitindo maior flexibilidade na construção da base de dados da aplicação e proporcionando diferentes estruturas com diferentes tipos de dados. Assim, o *Realtime Database* do *Firebase* fornece uma maneira simples de armazenar e aceder aos dados.

Na Figura 12 é apresentado o esquema geral da base de dados que suporta a aplicação *Wallet*.

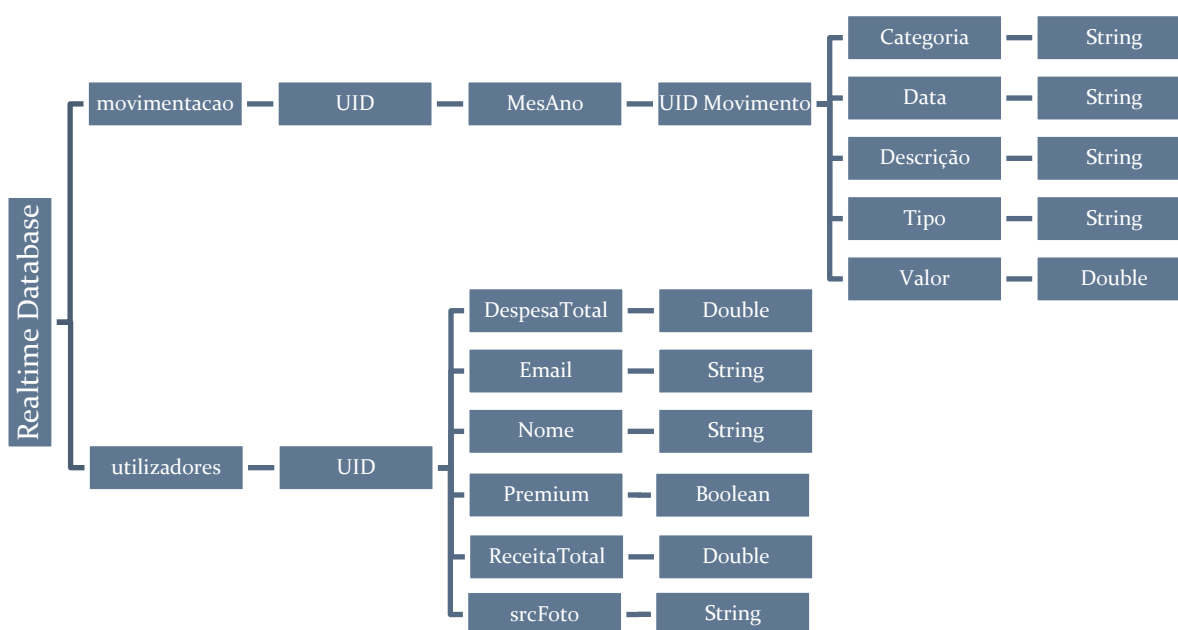


Figura 12 - Estrutura principal da aplicação

De forma a garantir a unicidade dos dados introduzidos na base de dados, *Firebase* disponibiliza um identificador único na autenticação – UID. Assim, este valor foi guardado no *Realtime Database*, de forma a discriminar os registos de cada utilizador. No caso das movimentações, foi necessário recorrer ao mecanismo

push do *Firebase*. Este mecanismo, permite que, sempre que seja adicionado um novo nó à base de dados, seja gerado uma chave única para introduzir os dados pretendidos, ficando com o seguinte formato:

/movimentação/UID Utilizador/MesAno/<unique-user-id>/<dados a inserir>

As figuras 13 e 14 apresentam o código usado para guardar os dados de cada utilizador e movimentação no *Realtime Database* do *Firebase*, utilizando os mecanismos acima referidos.

```
public void guardar(){
    DatabaseReference firebase = ConfiguracaoFirebase.getFirebaseDatabase();
    firebase.child("utilizadores")
        .child(this.idUtilizador)
        .setValue(this);
}
```

Figura 13 - Guardar Utilizador na base de dados

```
public void guardar(String mesAno){
    FirebaseUser utilizador = UtilizadorFirebase.getUtilizadorAtual();
    String idUtilizador = utilizador.getId();

    DatabaseReference firebase = ConfiguracaoFirebase.getFirebaseDatabase();
    firebase.child("movimentacao")
        .child(idUtilizador)
        .child(mesAno)
        .push()
        .setValue(this);
}
```

Figura 14 - Guardar Movimento da base de dados

Podemos ver, na figura 15, um *snapshot* dos dados guardados na base de dados do *Firebase* cumprindo com a estrutura acima indicada.

```
{
  "movimentacao": {
    "ko8aWPNHI2T3qEeU2I4gGFdXBsO2": {
      "062022": {
        "-N4nCXs4Xw_SGXTb28Gs": {
          "categoria": "Salario",
          "data": "17/06/2022",
          "descricao": "Salario",
          "tipo": "r",
          "valor": 1000
        },
        "-N4nCaEwQ6y3RRYfeYKo": {
          "categoria": "Lazer",
          "data": "17/06/2022",
          "descricao": "Jantar Fora",
          "tipo": "d",
          "valor": 25
        }
      }
    }
  },
  "utilizadores": {
    "ko8aWPNHI2T3qEeU2I4gGFdXBsO2": {
      "despesaTotal": 25,
      "email": "ana.as.ac@gmail.com",
      "nome": "Ana Sofia Oliveira",
      "premium": true,
      "receitaTotal": 1000,
      "srcFoto": "https://firebasestorage.googleapis.com/v0/b/wallet-
d9b2b.appspot.com/o/imagens%2Fperfil%2Fko8aWPNHI2T3qEeU2I4gGFdXBsO2.jpeg?alt=media&to
ken=c8daae22-0325-4c9f-af14-1c9eeb81401c"
    }
  }
}
```

Figura 15 – Realtime Database – Visão geral

Diagrama UML

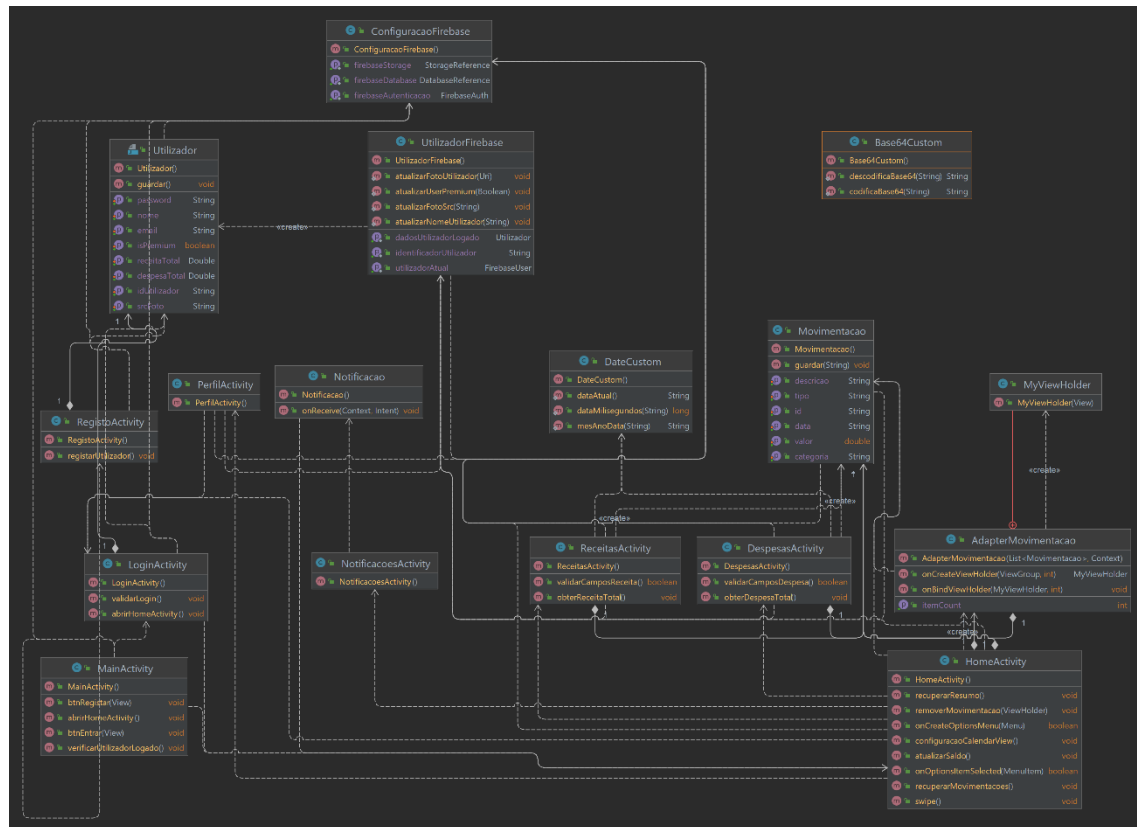


Figura 16 - Diagrama UML

Produto Final

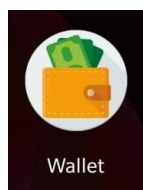


Figura 17 - Wallet App Icon Launch

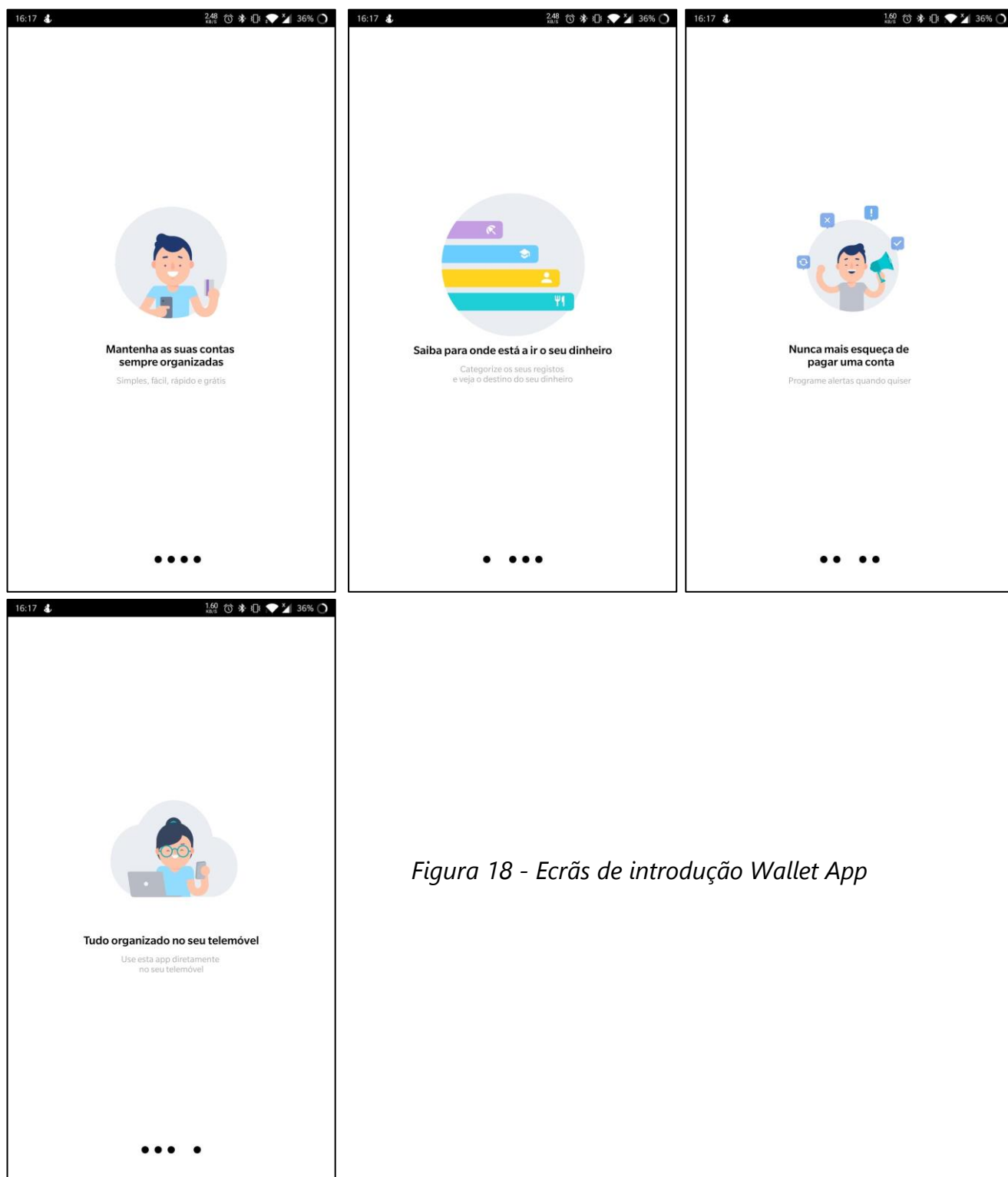


Figura 18 - Ecrãs de introdução Wallet App



Figura 19 - Wallet App -
Registrar/Login

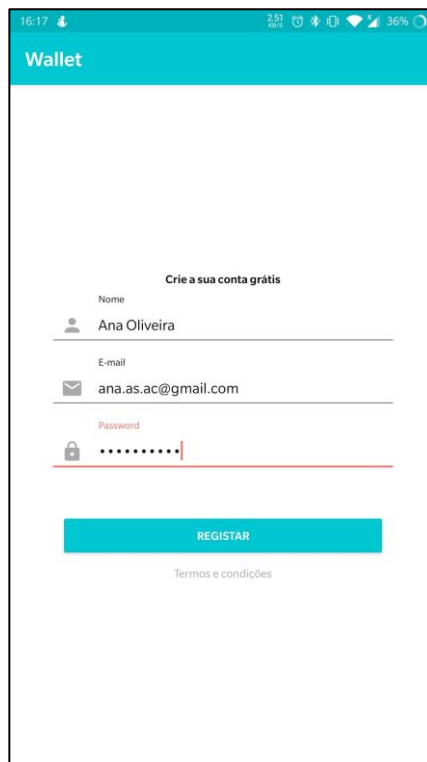


Figura 20 - Wallet App -
Registrar



Figura 21 - Wallet App - Login

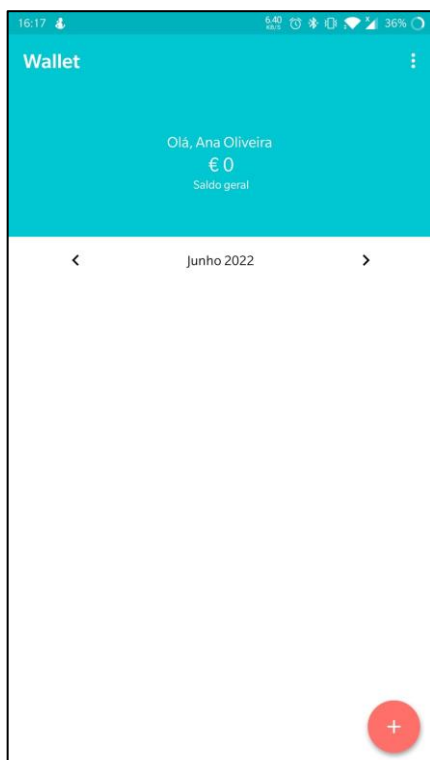


Figura 22 - Wallet App - Ecrã
inicial

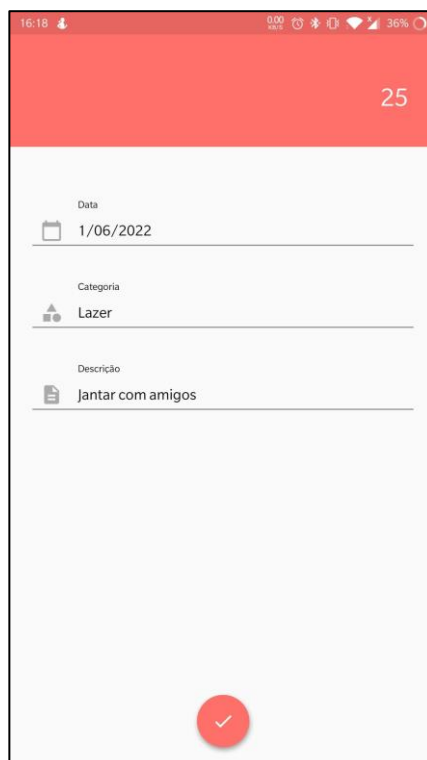


Figura 23 - Wallet App -
Adicionar Despesa

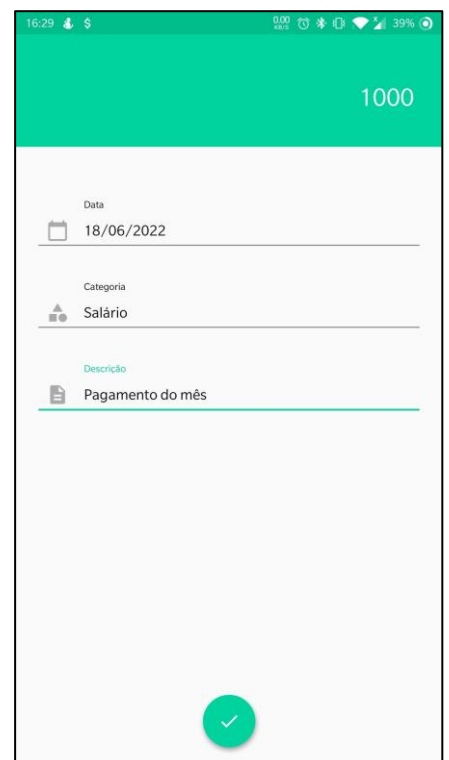


Figura 24 - Wallet App -
Adicionar Receita



Figura 26 - Wallet App - Ecrã inicial com despesas e receitas

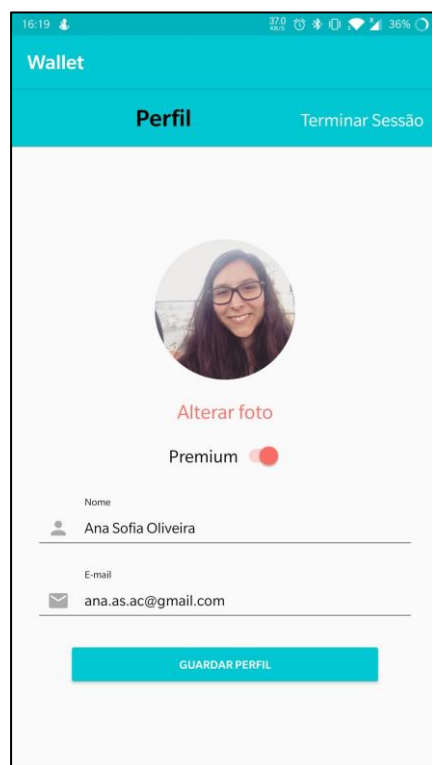


Figura 25 - Wallet App - Perfil

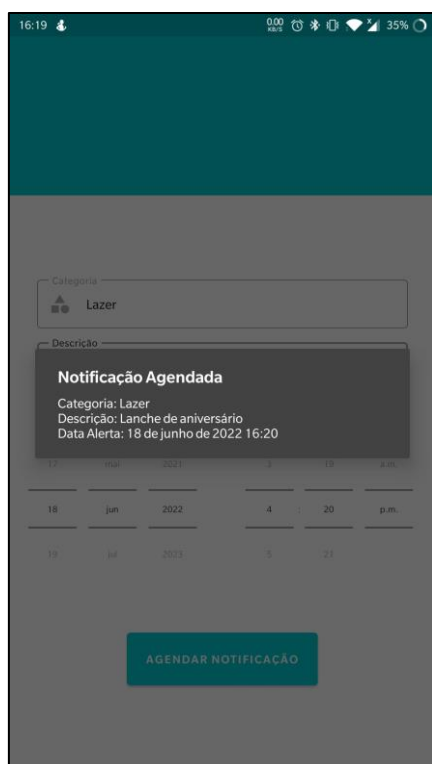


Figura 27 - Wallet App - notificação Agendada



Figura 28 - Wallet App - Notificação

Conclusões

O projeto desenvolvido permitiu melhorar e aumentar os conhecimentos das área de *User Interface* e *User Experience*(UI/UX), multimédia, design centrado no utilizador, bem como, o uso das plataformas *Firebase* de *Google* e *Android Studio*.

Através desta aplicação foi possível recolher alguns erros que, teoricamente, não seriam possíveis de captar, recolher feedback sobre diversos utilizadores, família e amigos. Durante a realização do projeto foram realizados diversos ajustes à interface gráfica, bem como ao funcionamento de algumas das funcionalidades.

O *Firebase* como *backend* permitiu automatizar o processo de construção de um servidor com motor da base de dados que permitisse manter o estado da aplicação entre os diversos utilizadores. Inicialmente, o desenvolvimento foi desafiante, porém com a consulta da documentação da *Google* tornou-se cada vez mais intuitivo.

Apenas uma das funcionalidades inicialmente programadas não foi implementada, a faturação da funcionalidade premium devido ao consumo de tempo exigido para compreender o lançamento de aplicação na *Google Play Console* com *features* pagas.

Considero que a aplicação se enquadra com o conceito proposto pelos docentes e cumpre o propósito para o qual inicialmente foi projetada. Todos os utilizadores a quem a aplicação foi apresentada não encontraram dificuldades de maior na sua utilização, apontando apenas alguns aspetos de design já modificados antes da apresentação deste produto final.

Bibliografia

- Slides fornecidos pelo docente;
- <https://firebase.google.com/docs>
- <https://github.com/heinrichreimer/material-intro>
- <https://github.com/prolificinteractive/material-calendarview>
- <https://github.com/bumptech/glide>
- <https://github.com/hdodenhof/CircleImageView>