



# *Happening@UBI: Partilha Rápida de Eventos da UBI*

Programação de Dispositivos Móveis  
Relatório do Trabalho de Grupo

Universidade da Beira Interior



*G-TEAM*

# Index

<b>Resumo</b>	<b>3</b>
<b>Introdução</b>	<b>4</b>
<b>Engenharia de Software</b>	<b>5</b>
3.1 - Ferramentas Utilizadas	5
Android Studio	5
Android	5
XML	5
Java	5
Git	6
Firebase	6
3.2 - Análise de Requisitos	6
Requisitos Funcionais	6
Requisitos Não Funcionais	7
3.3 - Diagrama Casos de Uso	7
3.3.1 - Diagrama de Atividade:	9
Registo:	10
Ver Feed:	10
Criar Evento:	10
<b>Implementação</b>	<b>11</b>
Escolhas de Implementação	11
Layout	11
Manual de Instalação	12
Manual de Utilização	12
<b>Reflexão Crítica e Problemas Encontrados</b>	<b>13</b>
Objetivos Propostos vs. Alcançados	13
Divisão de Trabalho pelos Elementos do Grupo	13
Problemas Encontrados	14
<b>Conclusões e Trabalho Futuro</b>	<b>15</b>
<b>Bibliografia</b>	<b>16</b>

# 1. Resumo

O tema deste projeto é o desenvolvimento de uma aplicação que permite docentes e não docentes da Universidade da Beira Interior partilhar e ver eventos que vão decorrer na mesma.

A equipa decidiu fazer um perfil para cada utilizador, este tem um *feed* em que todos os eventos que tenham ocorrido nas últimas 24 horas e todos os futuros eventos sejam apresentados por ordem cronológica.

O nosso objetivo era fazer uma aplicação parecida ao *Instagram* em que todos os utilizadores tinham um perfil público e que para além de cada utilizador poder falar com outros através de mensagens privadas também poderiam dizer se estariam presentes ou se estariam interessados em ir. No fim acabamos por descartar estas funcionalidades por ficarmos sem tempo e optamos apenas por ter um perfil privado para cada user e um *feed* em que este poderia ver todos os eventos das últimas 24 horas e eventos futuros.

## 2. Introdução

Este projeto tem como objetivo o desenvolvimento de uma aplicação que permite a partilha de eventos na *UBI*. No relatório explicamos alguns dos problemas que enfrentamos e como lidamos com eles. Falamos do processo de desenvolvimento da aplicação e como dividimos as tarefas pelos vários elementos do grupo.

O grupo é constituído por 5 elementos:

- Rúben Guilherme, com número de aluno 41059;
- Rúben Santos, com número de aluno 41308 ;
- Gonçalo Domingo, com número de aluno 41719 ;
- Paulo Duarte, com número de aluno 41853 ;
- Rui Barata, com número de aluno 41872 .

Este relatório está dividido em 4 capítulos principais:

- O primeiro capítulo (Resumo) apresenta o tema, como o abordamos e alguns resultados alcançados.
- O segundo capítulo (Introdução) descreve o problema a tratar, a constituição do grupo e uma breve descrição de todos os capítulos;
- O terceiro capítulo (Engenharia de *Software*) descreve as ferramentas utilizadas e o porquê de as escolhermos, enumera os requisitos funcionais e não funcionais e mostra os diagramas de caso de uso;
- O quarto capítulo (Implementação) explica as decisões tomadas na implementação da aplicação e os detalhes da mesma, o *layout*, o manual de instalação e de utilização.
- O quinto capítulo (Reflexão Crítica e Problemas Encontrados) neste capítulo falamos dos problemas encontrados e o que fizemos para os resolver, descreve também a divisão do trabalho pelos vários elementos do grupo e por fim uma reflexão crítica.

## 3. Engenharia de *Software*

### 3.1 - Ferramentas Utilizadas

#### ***Android Studio***

O *Android Studio* é um editor código que engloba várias funcionalidades, tais como ferramentas de correção de sintaxe da linguagem de programação, terminal ou o controlo *Git* incorporado, um rico editor de *layouts* com pré visualização destes mesmos, e um sistema de logs que ajuda no *debugging*. Neste programa foram utilizadas ferramentas de sincronização com os repositórios da plataforma *GitHub*, permitindo uma maior eficiência do projeto.

#### ***Android***

Android é um sistema operacional baseado no núcleo *Linux*, desenvolvido por um consórcio de desenvolvedores conhecido como *Open Handset Alliance*, sendo o principal colaborador do *Google*. Com uma interface de usuário baseada na manipulação direta, o Android é projetado principalmente para dispositivos móveis com tela sensível ao toque como smartphones e tablets; com interface específica para *TV*, carros (*Android Auto*) e relógios de pulso (*Android Wear*). O sistema operacional utiliza-se da tela sensível ao toque para que o usuário possa manipular objetos virtuais, assim como um teclado virtual. Apesar de ser principalmente utilizado em dispositivos com tela sensível ao toque, também é utilizado em consolas de videojogos, câmeras digitais, computadores e outros dispositivos eletrônicos.

#### ***XML***

*XML* é uma linguagem de marcação, e que tem como principal objetivo facilitar a troca de dados através da *internet*. Serve como facilitador da interação entre diversas linguagens de programação, sendo que estabelece um formato hierárquico para a criação de documentos. Usada na criação de *layouts* da aplicação.

#### ***Java***

*Java* é uma linguagem orientada a objetos diferente de outras linguagens de programação por ser “compilada para um *bytecode* que é interpretado por uma máquina virtual”. Começou a ser desenvolvida em 1991 pela *Sun Microsystems*, e desde então que tem vindo a crescer, até se tornar na linguagem de programação mais popular. Características como a orientação a objetos específicos do dia a dia e a sua independência da plataforma são

algumas das suas vantagens quando comparamos com outras linguagens de programação.

## **Git**

*Git* é uma ferramenta de controlo de versões, que serve para controlar as alterações feitas num computador local num projeto em que diversas pessoas participam ativamente. A sua principal função passa pela gestão de todo o código desenvolvido para uma aplicação, e a estabilização do mesmo, através da criação de diferentes versões dos desenvolvimentos realizados por diferentes elementos integrantes de um projeto.

## **Firebase**

*Firebase* é uma plataforma de desenvolvimento *mobile* (e *web*) adquirida pela *Google* em 2014. Com foco em ser um *back-end* completo e de fácil usabilidade, essa ferramenta disponibiliza diversos serviços diferentes que auxiliam no desenvolvimento e gerenciamento de aplicativos

## **3.2 - Análise de Requisitos**

Nesta seção será listado os requisitos funcionais e não funcionais da aplicação. Requisitos funcionais entendem-se como uma tarefa, atividade ou ação que o software deve desempenhar. Por sua vez, requisitos não funcionais englobam propriedades do sistema e respectivas limitações.

### **Requisitos Funcionais**

Os requisitos funcionais serão apresentados de seguida. Serão referenciados com a junção da abreviatura Requisito Funcional (*RF*) e um número indicador. Havendo de seguida os Requisitos Funcionais "Avançados", referenciados com a junção da abreviatura Requisito Funcional "Avançado" *RFA* e um número indicador.

RF01 - A aplicação deve permitir o registo de novos utilizadores.

RF02 - A aplicação deve suportar autenticação com as credenciais inseridas aquando do registo.

RF03 - Após autenticação, a aplicação deve mostrar todos os eventos ocorridos na *UBI* nas últimas 24 horas e todos os que ainda estão para acontecer.

RF04 - Após autenticação, a aplicação deve permitir a alteração de todos os dados de utilizador (exceto nome-de-utilizador) numa interface dedicada; deve também permitir apagar completamente o registo de um utilizador.

RF05 - Após autenticação, a aplicação deve permitir que um utilizador coloque um novo evento, inserindo um título e uma breve descrição,

RF06 - A inserção de imagens deve ser também possível, como adição ao título e descrição mencionados no ponto anterior.

### **“Avançados”**

RFA01 - O sistema suporta a partilha de gifs, para além de imagens e fotos.

RFA02 - O super utilizador pode eliminar eventos anteriores ou apagar utilizadores.

## **Requisitos Não Funcionais**

De seguida irão ser apresentados os Requisitos Não Funcionais, (*RNF*) com um número indicador.

RNF01 - A aplicação não irá permitir a criação de uma conta com um *email* já usado anteriormente.

RNF02 - A aplicação deverá implementar uma interface intuitiva.

RNF03 - A aplicação deverá ser completamente funcional a partir e incluído a versão *API 21* do *android*.

RNF04 - A aplicação deverá implementar um tema que se adeque às cores primárias do logótipo da universidade em questão.

## **3.3 - Diagrama Casos de Uso**

Nesta seção serão apresentados os casos de uso da aplicação. Define-se por casos de uso, os diagramas que identificam os atores numa interação e a sua descrição. O conjunto dos casos de uso deverá demonstrar todas as interações possíveis na plataforma. A aplicação conta com apenas 2 tipos de utilizadores: os utilizadores normais que utilizam a aplicação com o seu intuito, e os super utilizadores (*admins*) que fazem tudo o que os utilizadores fazem e mais algumas coisas.

### **Utilizador:**

Registar com uma conta: A partir de um *email* e de uma *password* vinda de um utilizador, este poderá se registar na aplicação;

Alteração da *password*: : Qualquer utilizador poderá alterar a sua palavra chave sempre que desejar;

Aceder com a conta da google : Qualquer utilizador que tenha uma conta *Google* poderá entrar na aplicação com a mesma;

Criação de eventos: Qualquer utilizador consegue criar um evento a partir de um nome, descrição, data, localização, horas e até uma imagem (opcional);

Visualização dos eventos criados por outros *users*: A página inicial apresentada ao utilizador permitirá que este visualize todos os eventos que estão a decorrer ou irão decorrer;

Acesso ao perfil: Qualquer utilizador poderá aceder ao seu perfil, seja para verificar os eventos criados ou até para mudar a sua foto de perfil e dados da conta;

### **Superutilizador:**

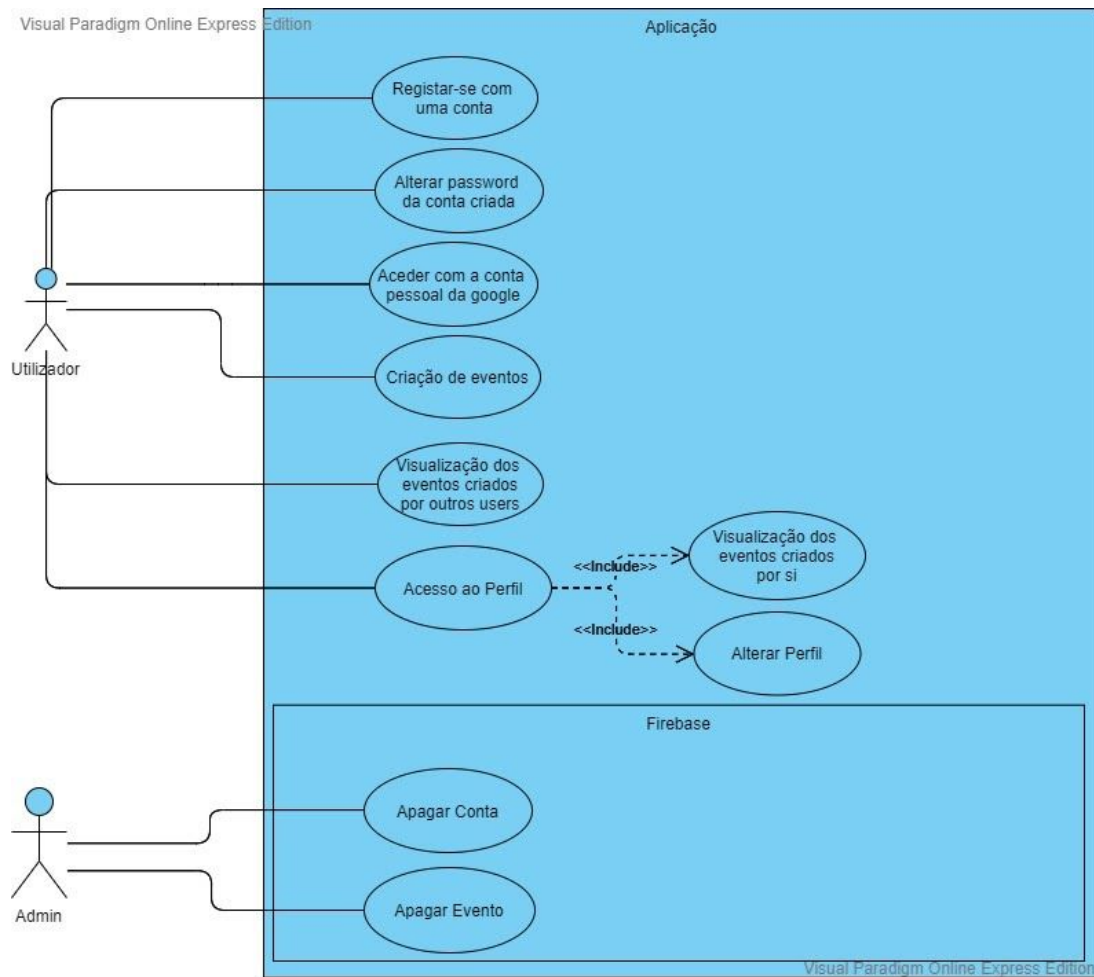
Para além de este conseguir fazer o que o utilizador normal faz, este utilizador tem acesso á *Firebase* e a partir daí poderá efetuar outras atividades como:

Apagar Conta: Este utilizador consegue apagar uma conta registada anteriormente;

Apagar Evento: Este utilizador consegue apagar um evento registado anteriormente;

| C \    \ \    \ ( ~ 3 ~ )    > ^ \    / ^ \    / / \ \  
wanna    ♪ /    \ ♪    / /    / ||    ( ( \    | | \ Kill    | ♪ \ ^ )    ||  
    ) / ♪    L / ( / myself | C \    \ \    \ ( ~ 3 ~ )    > ^ \    /  
    ^ \    / / \ \ wanna    ♪ /    \ ♪    / /    / ||    ( ( \    |  
| \ Kill    | ♪ \ ^ ) | C \    \ \    \ ( ~ 3 ~ )    > ^ \    / ^ \  
    / / \ \ wanna    ♪ /    \ ♪    / /    / ||    ( ( \    | | \ Kill  
| ♪ \ ^ )    ||    ) / ♪ | C \    \ \    \ ( ~ 3 ~ )    > ^ \    /  
    ^ \    / / \ \ wanna    ♪ /    \ ♪    / /    / ||    ( ( \    |  
| \ Kill    | ♪ \ ^ )    ||    ) / ♪    L / ( / myself | C \    \ \    \ ( ~ 3 ~ )  
    > ^ \    / ^ \    / / \ \ wanna    ♪ /    \ ♪    /  
    / /    / ||    ( ( \    | | \ Kill    | ♪ \ ^ )    ||    ) / ♪    L / ( / myself  
C \    \ \    \ ( ~ 3 ~ )    > ^ \    / ^ \    / / \ \  
wanna    ♪ /    \ ♪    / /    / ||    ( ( \    | | \ Kill    | ♪ \ ^ )    ||  
    ) / ♪    L / ( / myself    L / ( / myself    ||    ) / ♪    L / ( / myself

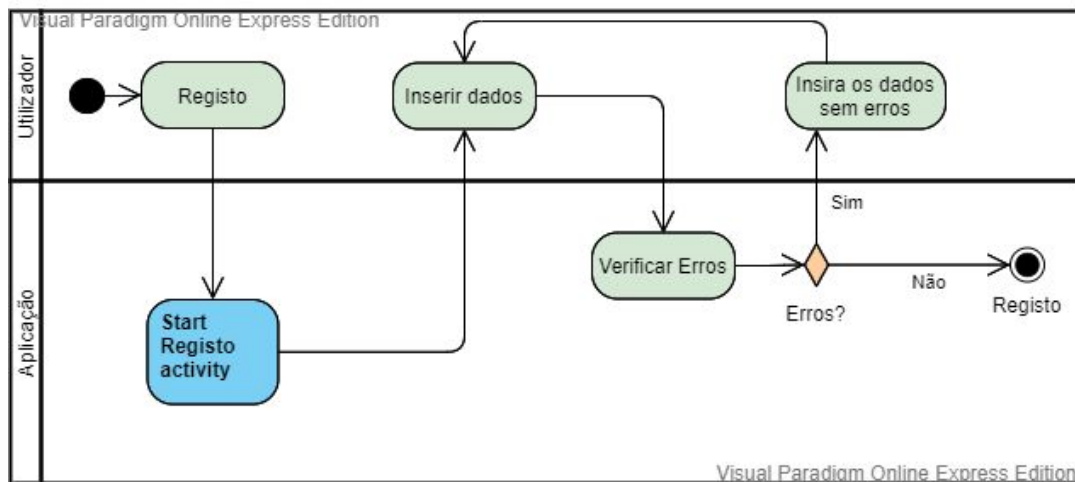




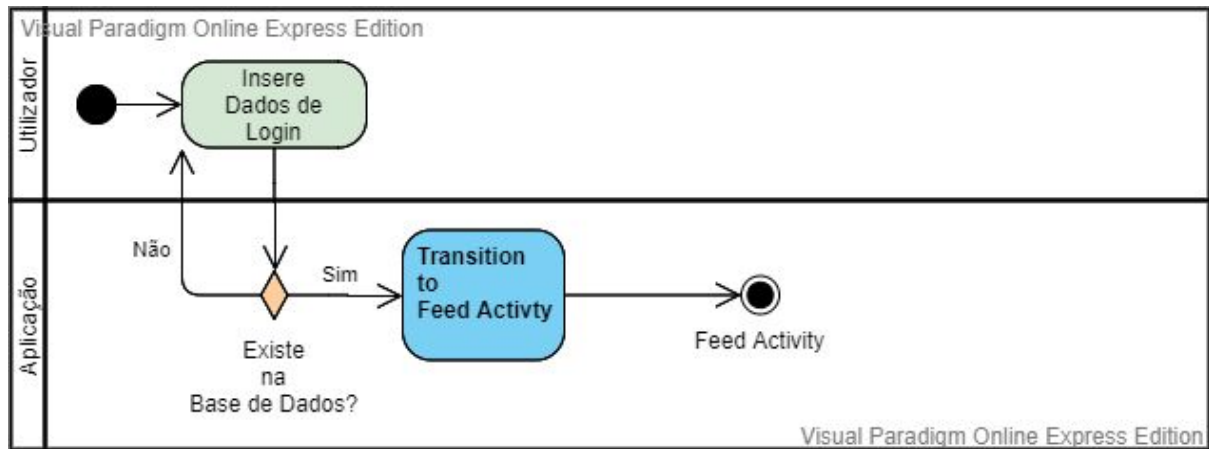
### 3.3.1 - Diagrama de Atividade:

Nesta secção serão apresentados os diagramas de atividade mais relevantes da aplicação. Os diagramas de atividade têm como função explicitar as atividades presentes num processo ou no processamento de dados.

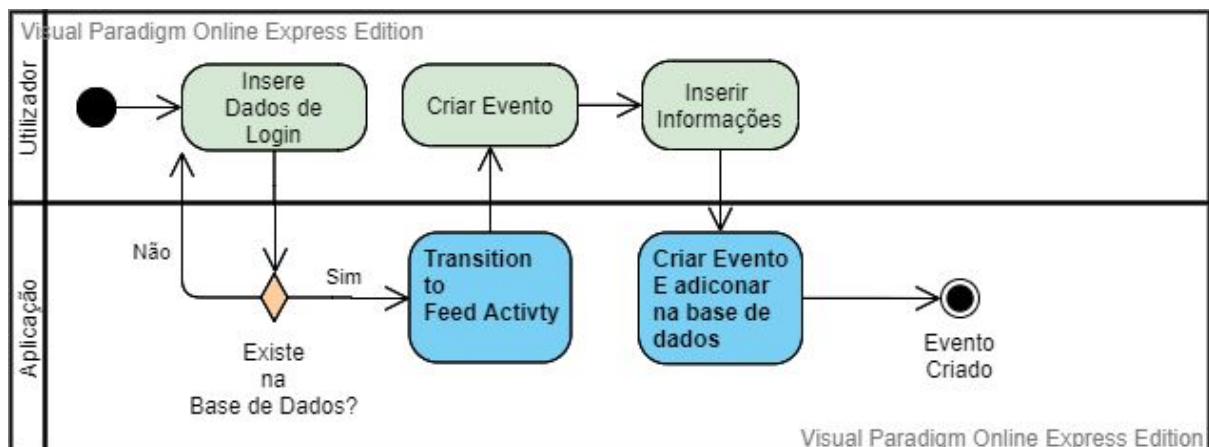
## Registo:



## Ver Feed:



## Criar Evento:



## 4. Implementação

### Escolhas de Implementação

Uma vez que tínhamos de fazer com que todos os eventos criados fossem partilhados por toda a comunidade de utilizadores da *Happening@UBI*, decidimos utilizar *Firebase* porque achámos que seria uma ferramenta bastante versátil e útil tendo em conta que precisávamos de utilizar uma base de dados remota.

### Layout



Para o *layout* da aplicação decidimos ter um *layout* intuitivo e que fosse simples. Para isso, olhamos para exemplos de sucesso nos casos das aplicações *Instagram* e *Snapchat*. Depois decidimos utilizar cores associadas à *UBI* como o azul utilizado no logotipo e tons modernos, permitindo ao utilizador focar-se nos aspetos importantes da aplicação.

## Manual de Instalação

Para instalar a aplicação, deve-se utilizar a *build variant release* e instalar no telemóvel ou dispositivo virtual através do *Android Studio*.

## Manual de Utilização

1. Iniciar a *app Happening@UBI*
2. Criar uma conta nova ou registar na *app* com a *Google*
3. Iniciada a sessão na *app*, o utilizador terá à sua disposição um *feed* com os eventos criados pelos diversos utilizadores. Clicando num evento, o utilizador verá as informações (texto e imagem se existir) mais detalhadas sobre esse mesmo evento.
4. No canto superior direito existem dois botões: o botão da esquerda é o botão que o utilizador deve premir se quiser criar um novo evento e o botão da direita é o botão que permitirá ao utilizador visualizar o seu perfil com os seus eventos criados.
5. No caso do utilizador querer criar um novo evento, então ser-lhe-á mostrado um formulário com os respetivos campos para que seja possível inserir as informações do evento e até imagens. Para publicar o evento, o utilizador deverá clicar no botão do canto superior direito. Caso o utilizador se arrependa de criar o evento poderá clicar no botão do canto superior esquerdo para que seja possível regressar ao *feed*.
6. No caso do utilizador querer visualizar a sua página de perfil, este poderá visualizar o seu *username*, a sua foto e os seus eventos criados. Clicando num evento, o utilizador verá as informações (texto e imagem se existir) mais detalhadas sobre esse mesmo evento. O utilizador se lhe for pertinente poderá clicar no botão que está localizado no canto superior direito onde poderá aceder às definições.
7. Se o utilizador quiser aceder às definições poderá alterar a foto de perfil, o nome de utilizador e a *password*, apagar a sua conta, mudar o idioma da aplicação e fazer *logout*.

## 5. Reflexão Crítica e Problemas Encontrados

### Objetivos Propostos vs. Alcançados

Dos objetivos propostos pelo professor Pedro Inácio para o nosso projeto *Happeing@UBI*, os objetivos que a equipa de desenvolvimento conseguiram alcançar foram:

- A aplicação deve permitir o registo de novos utilizadores;
- A aplicação deve suportar autenticação com as credenciais inseridas aquando do registo;
- Após autenticação, a aplicação deve mostrar todos os eventos ocorridos na *UBI* nas últimas 24 horas e todos os que ainda estão para acontecer;
- Após autenticação, a aplicação deve permitir a alteração de todos os dados de utilizador (exceto nome-de-utilizador) numa interface dedicada; deve também permitir apagar completamente o registo de um utilizador;
- Após autenticação, a aplicação deve permitir que um utilizador coloque um novo evento, inserindo um título e uma breve descrição;
- A inserção de imagens deve ser também possível, como adição ao título e descrição mencionados no ponto anterior.
- Autenticação com a *Google*.

Alguns objetivos que propusemos e não conseguimos alcançar devido ao tempo:

- Adicionar um super utilizador na aplicação.
- Adicionar um chat e mais interação entre utilizadores. (Tínhamos a infra-estrutura mas não tivemos tempo para implementar).

### Divisão de Trabalho pelos Elementos do Grupo

O trabalho foi dividido da seguinte forma:

Sistema de *login*:

- Rui Barata
- Gonçalo Domingos
- Rúben Santos

Sistema de registos:

- Rui Barata
- Rúben Santos
- Paulo Duarte

*Feed*:

- Rúben Guilherme
- Gonçalo Domingos

Criação de novos eventos:

- Rúben Guilherme
- Paulo Duarte

Página de perfil:

- Rúben Guilherme

Definições de perfil:

- Rui Barata
- Paulo Duarte

Definições da aplicação:

- Gonçalo Domingos

Páginas que mostram os eventos:

- Gonçalo Domingos

## Problemas Encontrados

Tivemos alguns problemas com o acesso à base de dados e com a criação de eventos em *real time*, além do *upload* de imagens. Para resolver isso decidimos utilizar a *Firebase* e as suas funcionalidades como a *Firebase Firestore* e a *Cloud Storage* da mesma.

Para adicionar o *login*, novamente decidimos utilizar a ferramenta de autenticação disponibilizada pela *Firebase* permitindo assim um fácil acesso à aplicação, além de autenticação com uma conta *Google*.

## 6. Conclusões e Trabalho Futuro

Com a finalização deste projeto pudemos melhorar inúmeras capacidades técnicas ao nível da programação de aplicações para *Android* como ao nível de *soft-skills*, tais como: o trabalho em grupo, a ajuda entre todos, a capacidade de analisar criticamente o trabalho que foi desenvolvido e até a nossa criatividade e liderança.

Assim sendo, temos pena por não termos conseguido criar todas as funcionalidades que tínhamos pensado inicialmente no tempo que nos foi dado, no entanto há a possibilidade de podermos continuar a melhorar a *app* de forma a que possamos finalmente finalizar a *app* tal e qual como tínhamos idealizado inicialmente.

## 7. Bibliografia

<https://code.tutsplus.com/tutorials/working-with-git-in-android-studio--cms-30514>  
<https://developer.android.com/guide>  
<https://developers.google.com/identity/sign-in/android/start-integrating>  
<https://stackoverflow.com/questions/19585815/select-multiple-images-from-android-gallery>  
<https://stackoverflow.com/questions/47214128/how-to-store-gmail-name-and-google-profile-picture-using-firebase-auth-and-fire>  
<https://stackoverflow.com/questions/42800349/forgot-password-in-firebase-for-android>  
<https://stackoverflow.com/questions/46799461/upload-profile-image-for-a-user-firebase>  
<https://stackoverflow.com/questions/38114689/how-to-delete-a-firebase-user-from-android-app>  
<https://www.youtube.com/watch?v=ts98gL1JCQU>  
[https://www.youtube.com/watch?v=18VcnYN5\\_LM](https://www.youtube.com/watch?v=18VcnYN5_LM)  
<https://www.youtube.com/watch?v=xgpLYwEmIO0>  
<https://www.youtube.com/watch?v=yKeULpr9fcg>  
<https://www.youtube.com/watch?v=TwHmrZxiPA8>  
<https://www.youtube.com/watch?v=jXtof6OUtcE>  
[https://www.youtube.com/watch?v=e5\\_C9e\\_gKOM](https://www.youtube.com/watch?v=e5_C9e_gKOM)  
<https://www.youtube.com/c/AndroidTrainingProgram>  
<https://medium.com/@iamspathan/firebase-authentication-adding-authentication-in-android-43c500c13ae7>