

# **Desenvolvimento em Android**

**Edney Vaz Pinto Lázaro - 37533**

**Trabalho realizado sob a orientação de**

**Nuno Gonçalves Rodrigues (IPB)**

**Johnnylee Bryan Marques da Rocha (IT Sector)**

**Licenciatura em Engenharia Informática**

**2019-2020**



# **Desenvolvimento em Android**

Relatório da UC de Projeto  
Licenciatura em Engenharia Informática  
Escola Superior de Tecnologia e Gestão

Edney Lázaro - 37533

2019-2020

A Escola Superior de Tecnologia e Gestão não se responsabiliza pelas opiniões expressas neste relatório.

Declaro que o trabalho descrito neste relatório é da minha autoria e é da minha vontade que o mesmo seja submetido a avaliação.

---

Edney Vaz Pinto Lázaro - 37533



# Dedicatória

Eu, Edney Vaz Pinto Lázaro, dedico a conclusão este Projeto a minha mãe, Eugenia Vaz Pinto Eusébio Lázaro, e ao meu pai, Eusébio Pereira Lázaro e a todos aqueles que me acompanharam nesta jornada académica.

# Agradecimentos

Venho por este meio dedicar, primeiramente, os meus maiores agradecimentos a minha mãe, Eugenia Vaz Pinto Eusébio Lázaro que sempre acreditou em mim e me apoiou incondicionalmente nesta jornada que decidi começar alguns anos atrás. Ao meu pai, Eusébio Pereira Lázaro que me ajudou financeiramente, sem o qual não seria possível eu estar tão perto do final da mesma. Serei sempre grato aos dois por me terem proporcionado esta oportunidade de poder seguir uma das minhas paixões.

Um agradecimento a toda a equipa da ITSector que infelizmente não pude conhecer todos pessoalmente, mas que tive o prazer de trabalhar com, sempre prontos a ajudar em qualquer dúvida que aparecia. A Marta Rodrigues por ter me dado a oportunidade de participar neste estágio. A Joana Pereira por me ter recebido no primeiro e me feito sentir logo em casa. E um especial agradecimento ao meu orientador Johnnylee Bryan Marques da Rocha por ter me ajudado durante toda esta nova experiência pela qual passei.



# Resumo

Este projeto foi proposto pela empresa ITSector, com o objetivo de manutenção, desenvolvimento e otimização de aplicações para plataformas moveis.

Com o avanço científico e sociológico dá-se surgimento de diversos problemas e muito deles sendo solucionados com o uso tecnológico. Nos dias atuais o uso da computação ubíqua é quase inevitável, e a obtenção de smartphones cresce cada vez mais. Com isso a demanda por aplicações novas e atualizadas torna-se maior igualmente.

Neste projeto o desafio apresentado é o desenvolvimento e otimização de aplicações, utilizando as plataformas Android Studio e também de tecnologias como Java, XML e Kotlin. Tendo uma primeira experiência em um espaço profissional.

A aplicação desenvolvida tem como alvo a área da banca e objetivo é atualizar algumas das funções já implementadas anteriormente e adicionar novas funções com o objetivo de melhorar e tornar mais intuitivo o uso da mesma pelos clientes.

**Palavras-chave (Tecnologias):** Java, Git

# Abstract

This project was proposed by the company ITSector, with the goal of maintenance, development and optimization of applications for mobile platforms.

With the scientific and sociological advances, several problems arise and many of them are solved with technological use. Nowadays the use of ubiquitous computing is almost inevitable, and the acquisition of smartphones grows more and more. As a result, the demand for new and updated applications becomes even greater.

In this project the challenge presented is the development and optimization of applications, using the Android Studio platforms and technologies such as Java, XML and Kotlin. Having a first experience in a professional space.

The developed application targets the banking area and aims to update some functions already implemented previously and adds new functions in order to improve and make it more intuitive for customers to use.

**Keywords (Technologies):** Java, Git

# Conteúdo

<b>Capítulo 1 .....</b>	<b>15</b>
<b>Introdução .....</b>	<b>155</b>
1.1 Enquadramento .....	155
1.2 Objetivos.....	166
1.3 Formação .....	16
1.4 Estrutura do Documento .....	17
<b>Capítulo 2.....</b>	<b>20</b>
<b>Contexto e Tecnologias/Ferramentas .....</b>	<b>20</b>
2.1 Contextualização.....	20
2.2 Tecnologias Utilizadas.....	20
<b>Capítulo 3.....</b>	<b>24</b>
<b>Abordagem/Análise/Modelação.....</b>	<b>24</b>
3.1 Análise de Requisitos.....	24
3.2 Engenharia de Software.....	26
<b>Capítulo 4.....</b>	<b>28</b>
<b>Desenvolvimento/Implementação .....</b>	<b>28</b>
4.1 Estrutura de um projeto em Android Studio.....	28
4.2 Arquitetura da Aplicação.....	30
<b>Capítulo 5.....</b>	<b>37</b>
<b>Descrição das Novas Funcionalidades.....</b>	<b>37</b>
5.1 Revogação de Cartão.....	37
5.2 Rastreamento de Cartão.....	38
<b>Capítulo 6.....</b>	<b>39</b>
<b>Conclusões .....</b>	<b>39</b>
6.2 Trabalho Futuro.....	40
<b>Bibliografia .....</b>	<b>41</b>

<b>Apêndice A.....</b>	<b>1</b>
------------------------	----------

# Lista de Figuras

Figura 1 - Java.....	21
Figura 2 – XML.....	21
Figura 3 – Android Studio.....	22
Figura 4 – Git. ....	22
Figura 5 – Representação da Revogação de Cartão. ....	26
Figura 6 – Representação do Rastreamento de chegada de novos cartões.....	26
Figura 7 – Pastas Referente a um projeto em Android Studio. ....	28
Figura 8 – Representação da Arquitetura MVC.....	30
Figura 9 – Ciclo de vida de uma Activity.....	31
Figura 10 – Representação do GitFlow.....	34

# Siglas

<b>ESTiG</b>	Escola Superior de Tecnologia e Gestão.
<b>API</b>	Application Programming Interface.
<b>JVM</b>	Java Virtual Machine
<b>XML</b>	Extensible Markup Language
<b>IDE</b>	Integrated Development Enviroment
<b>JVM</b>	Java Virtual Machine
<b>XML</b>	Extensible Markup Language
<b>NDK</b>	Native Development Kit
<b>UML</b>	Unified Modeling Language
<b>MVC</b>	Model-View-Controller

# Capítulo 1

## Introdução

Neste capítulo serão apresentadas as motivações, os objetivos, a formação e a organização do relatório para o desenvolvimento deste trabalho.

### 1.1 Enquadramento

Uma parte fundamental para a finalização da Licenciatura do curso de Engenharia Informática, é a realização de um projeto anual ou a realização de um estágio curricular, com o objetivo de testar e aprofundar os conhecimentos adquiridos durante os dois anos anteriores.

Este projeto foi proposto pela empresa ITSector, que é uma empresa de desenvolvimento de software especializada em transformação digital para instituições financeiras. Desde a sua criação em 2005, capacitam usuários de todo o mundo a se envolverem facilmente com aplicativos corporativos, mudando a vida das pessoas por meio de inovação e conhecimento especializado.

A proposta apresentada neste projeto é a implementação de novas funcionalidades em um projeto dirigido a área bancária tendo este sido iniciado anos atrás. Com o objetivo de manter e atualizar a mesma.

O referido projeto foi desenvolvido no formato de um estágio curricular, realizado nas instalações da empresa ITSector, em Lisboa e com a duração de 17 semanas.

## 1.2 Objetivos

O objetivo do projeto é desenvolver e otimizar uma aplicação aplicada à área bancária para o sistema Android, de modo a melhorar e adicionar novas formas do utilizador interagir com a mesma.

O projeto já havia sido inicializado aquando a minha chegada, por isso a maior parte das funções básicas já haviam sido implementadas.

Com isso foi-me dado o objetivo de ajudar a corrigir alguns bugs encontrados no código e a ajuda na implementação de novas maneiras do utilizador fazer uma tarefa de maneira mais simples.

## 1.3 Formação

Foi realizado um processo de integração no ambiente da empresa com a duração de um mês. Este processo foi dividido em dois módulos, o primeiro sendo uma introdução à criação de projetos no Android Studio e revendo os conceitos de programação orientada a objetos, e finalizado com a apresentação de um pequeno programa ao orientador. O segundo módulo, este mais avançado, foi realizado de modo a aprofundar ainda mais os conhecimentos e apresentando novos e mais avançados finalizando com um projeto que abrangia tudo o que foi aprendido durante a formação.

Durante o desenvolvimento do projeto os responsáveis estiveram sempre prontos a ajudar e a retirar toda e quaisquer dúvidas que foram aparecendo. Deste modo, proporcionando mais independência na resolução de problemas.

O projeto final da formação foi a realização de uma aplicação para o sistema Android que é ligada à nuvem que deixa o utilizador procurar por filmes populares e marcá-los como favoritos, fazem o uso de dados reais providos de uma API de terceiros. Para o desenvolvimento e implementação deste projeto foi utilizada a IDE Android Studio. Os tutoriais utilizados na formação foram Android Basics: User Input e Developing Android Apps ambas proporcionadas pela Udacity [1] que é um provedor de diversos cursos. O objetivo final foi a aquisição de conhecimentos direcionados ao sistema Android e ao seu funcionamento.



## 1.4 Estrutura do documento

Este relatório encontra-se dividido em 6 “Capítulos”, de modo que, o conteúdo esteja organizado de maneira coerente, facilitando a leitura e a compreensão.

Designação e Explicação dos capítulos:

- Capítulo 1: Introdução

Neste capítulo é feita uma breve abordagem relativamente às motivações, objetivos, formação e o enquadramento do projeto

- Capítulo 2: Contexto e Tecnologias/Ferramentas

Neste capítulo é feita a contextualização do problema e explicação das ferramentas e linguagem de programação utilizadas.

- Capítulo 3: Abordagem/ Análise/ Modelação

Neste capítulo é feita a abordagem, análise e modelação do que seria esperado da implementação das novas funcionalidades. Contem diagramas de casos de uso representando as novas funcionalidades e os intervenientes necessários.

- Capítulo 4: Desenvolvimento/Implementação

Neste capítulo é explicada a base do projeto e pela qual todas as novas funcionalidades devem seguir, não se desviando demasiado da estrutura já implementada. Tem como principal finalidade, a compreensão do funcionamento da arquitetura utilizada, bem como a estrutura do projeto

- Capítulo 5: Descrição das novas Funcionalidades

Neste capítulo é feita a explicação das novas funcionalidades implementadas.

- Capítulo 6: Conclusões

Neste capítulo é feita uma reflexão final do projeto, enunciando as várias etapas e dificuldades passadas para a finalização do mesmo. E finaliza com uma análise das possibilidades de trabalho futuro associado ao mesmo projeto.



# Capítulo 2

## Contexto e Tecnologias/Ferramentas

### 2.1 Contextualização

Como já mencionado no Capítulo 1, o principal objetivo deste projeto é o desenvolvimento e otimização de uma aplicação que tem sido desenvolvida já alguns anos, melhorando a experiência do utilizador.

O avanço das tecnologias a cada dia, exige que as aplicações sejam atualizadas frequentemente de modo a acompanharem a constante evolução dos smartphones, tornando assim acessível o uso de smartphones por quase toda a população.

Com isto dá-se a oportunidade de empresas criarem aplicações que irão tornar a realização de tarefas mundanas muito mais práticas com o uso destas aplicações, tal como pagar contas, verificar sados, ler livros, etc.

### 2.2 Tecnologias Utilizadas

Neste subcapítulo estão expostas uma pequena base das tecnologias utilizadas no desenvolvimento deste projeto, para que possamos entender o que foi utilizado para implementar as atualizações da aplicação.

## 2.2.1 Java



*Figura 1 - Java*

Java [2] é uma linguagem de programação orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Em 2008 o Java foi adquirido pela empresa Oracle Corporation. Diferente das linguagens de programação modernas, que são compiladas em código nativo, a linguagem Java é compilada para um bytecode que é interpretado pela JVM.

## 2.2.2 XML



*Figura 2 - XML*

O XML [3] (Extensible Markup Language) é uma linguagem de marcação que define um conjunto de regras para codificar documentos em um formato legível por máquina.

Os objetivos de design do XML enfatizam a simplicidade, generalidade e usabilidade na internet. É um formato de dados textuais com forte suporte via Unicode para diferentes idiomas humanos. Embora o design do XML se concentre nos documentos, a linguagem é amplamente usada para a representação de estruturas de dados arbitrárias, como aquelas usadas em serviços da web.

Existem vários sistemas de esquemas para auxiliar na definição de linguagens baseadas em XML, enquanto os programadores desenvolveram muitas interfaces de programação de aplicativos (API) para auxiliar no processamento de dados XML.

## 2.2.3 Android Studio



*Figura 3 - Android Studio*

O Android Studio [4] é um ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de apps para Android e é baseado no IntelliJ IDEA.

Além do editor de código e das ferramentas de desenvolvimento do IntelliJ, o Android Studio oferece ainda mais recursos para aumentar sua produtividade na compilação de apps Android, como:

- Emulador rápido
- Um ambiente unificado que possibilita o desenvolvimento para todos os dispositivos Android
- Frameworks e ferramentas de teste cheios de possibilidades
- Compatibilidade com C++ e NDK
- Modelos de código e integração com o GitHub para ajudar a criar recursos comuns de apps e importar exemplos de código

## 2.2.4 Git



*Figura 4 - Git*

Git [5] é um sistema de controle de versão distribuído para rastrear alterações no código fonte durante o desenvolvimento de software.

Permite desenvolver projetos em que uma ou mais pessoas podem contribuir simultaneamente no mesmo, mantendo-se uma versão base que será clonada, sendo o clone modificado pelos mesmos com a finalidade de ser implementadas novas funções que serão adicionadas à versão base caso seja aprovada.



# Capítulo 3

## Abordagem/Análise/Modelação

Neste capítulo foi feita uma análise à problemática mencionada nos capítulos anteriores, tentando achar a melhor forma de se implementar uma solução de maneira a tornar acessível ao utilizador.

A estrutura base do projeto já havia sido determinada aquando a minha chegada ao projeto. Este já havia sido iniciado e a estrutura consolidada sendo-me dada a tarefa de ajudar no solucionamento de bugs e implementação de novas funcionalidades, neste caso necessitando de uma modelagem que siga os parâmetros impostos pelo cliente.

De maneira a implementar estas novas funcionalidades foi-me dada uma orientação, devendo seguir na sua implementação de forma a manter a nova funcionalidade dentro dos parâmetros seguidos pela ITSector neste projeto em particular.

Neste capítulo irei apresentar uma pequena modelação das funcionalidades cujo qual participei na implementação.

### **3.1 Análise de Requisitos**

#### **3.1.1 Requisitos Técnicos**

O administrador tem acesso ao código fonte da aplicação, de modo a mantê-la livre de bugs e atualizá-la com novas funcionalidades que sejam convenientes.

O objetivo é tornar a aplicação o mais intuitiva possível para os utilizadores.

O utilizador deve conseguir usar a aplicação de uma forma fácil e cómoda, de modo a ter acesso à sua conta associada ao banco em causa.



## 3.2 Engenharia de Software

### 3.2.1 Linguagem UML

UML [6] (Unified Modeling Language) é uma linguagem padrão para a elaboração da estrutura de software. Ela pode ser usada para a visualização, a especificação, a construção e a documentação de artefactos que façam uso de sistemas complexos de software.

A UML não é uma linguagem de desenvolvimento, ou seja, ela não diz a ordem como o projeto será realizado, mas esta ajuda a visualizar o desenho e a comunicação entre os objetos.

Basicamente a UML permite que os desenvolvedores visualizem os produtos de seus trabalhos em um diagrama padronizado.

A linguagem UML nos apresenta com diversos diagramas que devem ser utilizados de modo a representar o sistema a ser montado, sendo o diagrama de Caso de Uso um dos utilizados para representar o sistema.

Um Diagrama de Casos de Uso é constituído por:

- Atores – são estes que se centram todas as ações tornando o sistema funcional. Deve ser externo ao sistema, podendo ser um usuário ou outro sistema que interaja com o sistema a ser criado.
- Caso de Uso – é uma especificação de um conjunto de ações executadas por um sistema que contém um resultado observável.
- Relacionamentos – auxiliam na descrição dos casos de uso, podendo ser: entre ator e caso de uso, entre atores e entre casos de uso.
  - Relacionamento de Inclusão: utilizado quando o comportamento se repete em mais de um caso de uso.
  - Relacionamento de Extensão: utilizado quando se deseja modelar um relacionamento alternativo.

### 3.2.2 Diagrama de Caso de Uso

Representação de caso de uso das funcionalidades pela qual fiz parte na implementação.

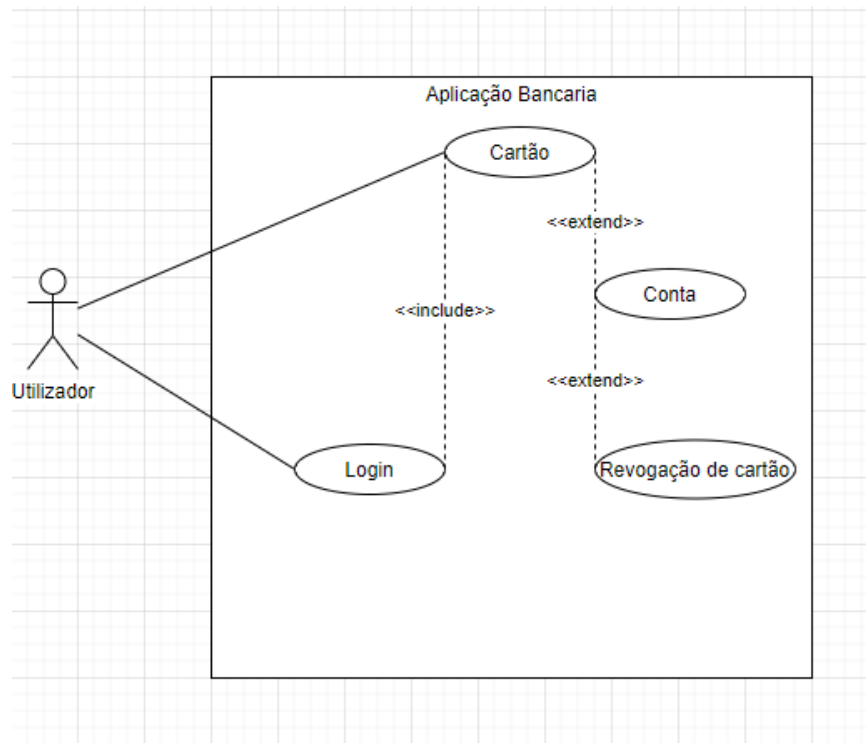


Figura 5 - Representação da Revogação de Cartão

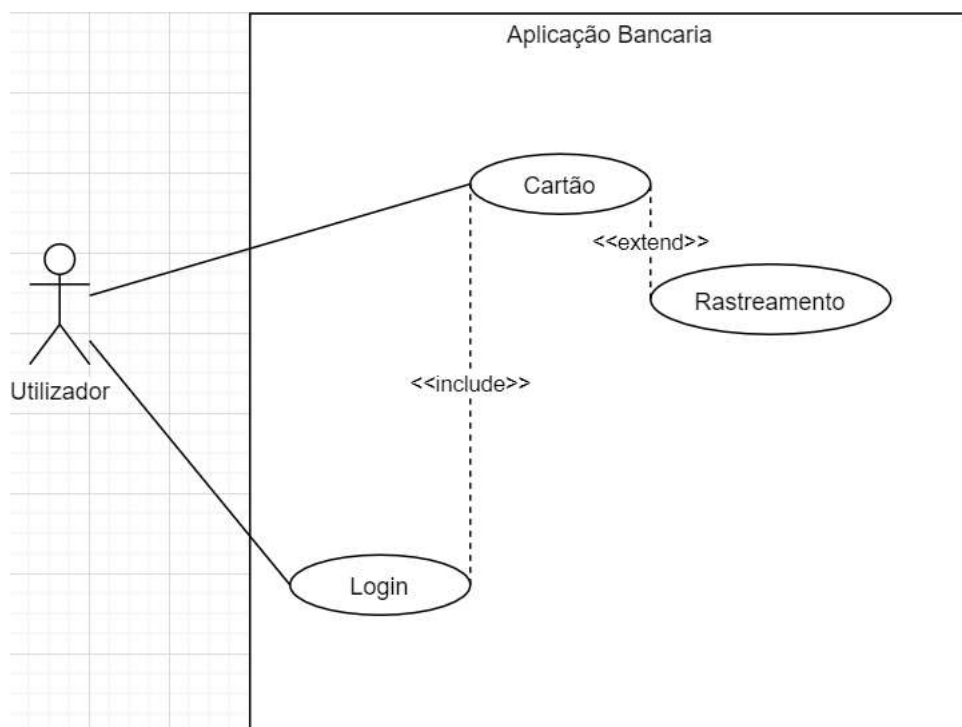


Figura 6 - Representação do Rastreamento de chegada de novos cartões

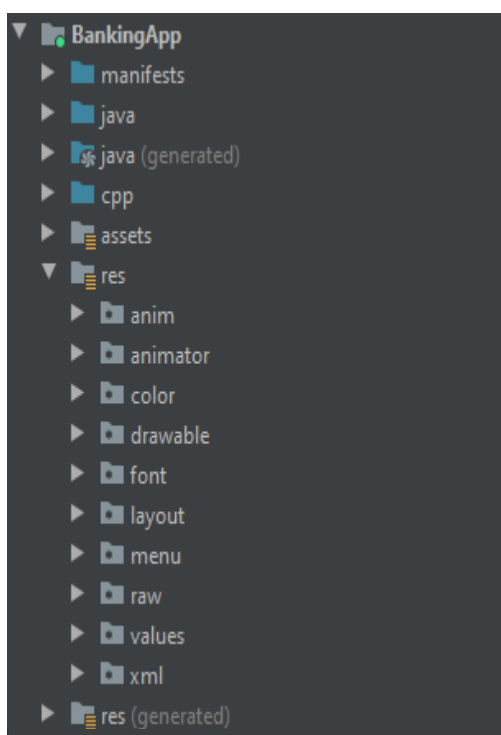


# Capítulo 4

## Desenvolvimento/Implementação

O projeto foi desenvolvido fazendo uso do ambiente de desenvolvimento Android Studio. De modo a introduzir a estrutura de um projeto em android irei apresentar uma explicação básica das pastas e elementos principais.

### 4.1 Estrutura de um projeto em Android Studio



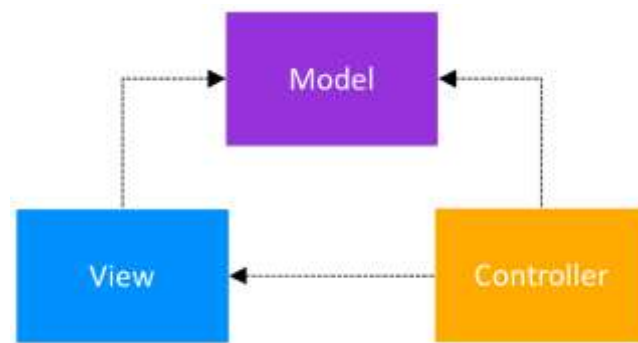
*Figura 7 - Pastas Referente a um projeto em Android Studio*

Na figura podemos ver as diferentes pastas referentes a um projeto em android. Nelas estão:

- manifests – contém os ficheiros AndroidManifest.xml
- java – contém os arquivos de código-fonte Java

- `cpp` – contém os arquivos de código-fonte C e C++ (opcional)
- `assets` – contém os arquivos que necessitam de ser compilados num arquivo `.apk`
- `res` – contém recursos de aplicativos
- `anim` – arquivos XML que definem as animações intermediárias
- `animator` – contém arquivos XML que definem as propriedades das animações
- `color` – contém arquivos XML que definem uma lista de estado de cores
- `drawable` – contém arquivos Bitmap (`.png`, `.jpg`, `.gif`, etc.) ou arquivos XML
- `font` – contém arquivos de fonte ou arquivos XML que incluem um elemento `<font-family>`
- `layout` – contém arquivos XML que definem a disposição da interface do utilizador
- `menu` – contém arquivos XML que definem os menus do aplicativo
- `raw` – contém arquivos arbitrários para salvar na forma bruta
- `values` – contém arquivos XML que contêm valores simples, como strings, números inteiros e cores.
- `xml` – contém arquivos arbitrários XML que podem ser lidos no ambiente de execução
- `Gradle Scripts` – contém arquivos de configuração da compilação do projeto

## 4.2 Arquitetura da Aplicação



*Figura 8 - Representação da Arquitetura MVC*

O MVC [7] (Model-View-Controller) é um padrão de design de software comumente usado para o desenvolvimento de interfaces de usuário que divide a lógica do programa relacionando em três elementos interconectados. Isso é feito para separar representações internas de informações das formas como as informações são apresentadas e aceitas pelo usuário.

### 4.2.1 Visão (View)

A Activity e o Fragment representam a camada View, que não lida com lógica de negócios e operações complexas.

A Activity é configurada no momento que é criada, após isso, ela é iniciada, e fica visível ao utilizador. Após a sua inicialização, a Activity passa por vários estados representados por métodos específicos para tal, sendo estes:

- onCreate – é usado para configurar a interface de usuário.
- onStart – é executado depois de a Activity ter sido enviada para o segundo plano.
- onResume – é acionado quando a Activity se inicia e quando é reiniciada. É sempre acionado quando a Activity volta ao primeiro plano.
- onPause – é acionado quando a Activity deixa o primeiro plano.
- onStop – é chamado quando a Activity não está mais visível para o utilizador.

- `onDestroy` – é chamado quando a Activity vai ser destruída. Sendo este a última chamada da Activity.

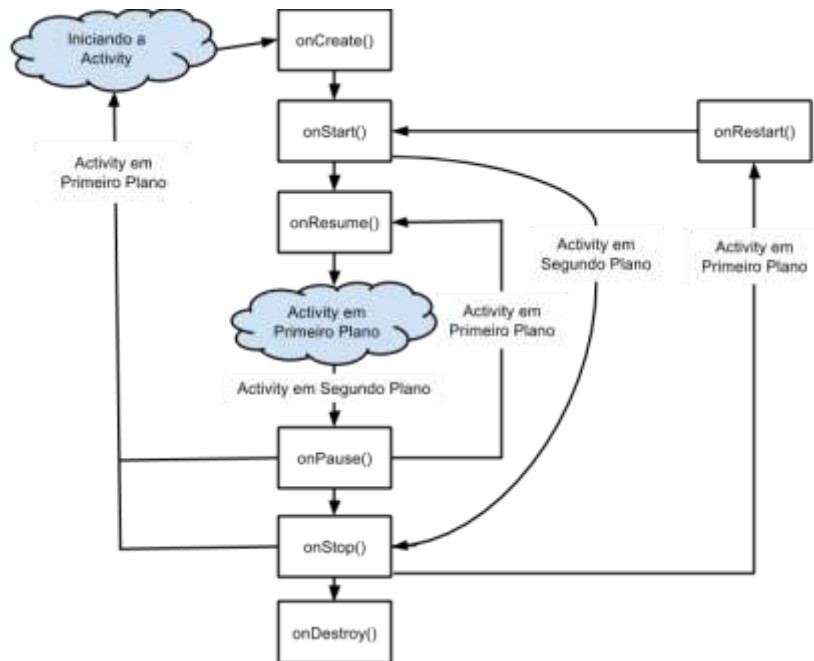


Figura 9 - Ciclo de vida de uma Activity

## 4.2.2 Controlador (Controller)

O controlador é a componente que controla a navegação do utilizador dentro da app, processando as ações e mudanças feitas pelo utilizador no domínio, alterando assim o que é apresentado ao utilizador dependendo da ação feita.

O controlador mais utilizado para tal neste projeto foi o `AsyncTask` [8] e um exemplo da sua implementação é esta.

---

```

private class DownloadFilesTask extends AsyncTask<URL,
Integer, Long> {
    protected Long doInBackground(URL... urls) {
        int count = urls.length;
        long totalSize = 0;
        for (int i = 0; i < count; i++) {
            totalSize += Downloader.downloadFile(urls[i]);
            publishProgress((int) ((i / (float) count) *
100));
            // Escape early if cancel() is called
            if (isCancelled()) break;
        }
    }
}
  
```

```

        return totalSize;
    }

    protected void onProgressUpdate(Integer... progress) {
        setProgressPercent(progress[0]);
    }

    protected void onPostExecute(Long result) {
        showDialog("Downloaded " + result + " bytes");
    }
}

```

---

### 4.2.3 Modelo (Model)

O modelo é a lógica dos dados mais estado mais negócios do aplicativo. São os cérebros da nossa aplicação, por assim dizer.

Não está vinculado à visualização ou controlador e, por isso, é reutilizável em muitos contextos.

Um exemplo de um modelo utilizado para as chamadas de banco de dados na aplicação é o RxJava [9].

Um simples exemplo da sua implementação é a seguinte.

---

```

package com.vogella.android.rxjava.simple;

import org.junit.Test;

import io.reactivex.Observable;

import static junit.framework.Assert.assertTrue;

public class RxJavaUnitTest {
    String result="";

    // Simple subscription to a fix value
    @Test
    public void returnAValue() {
        result = ""; Observable<String> observer =
        Observable.just("Hello"); //provides data
        observer.subscribe(s -> result=s); //Callable as
        subscriber
        assertTrue(result.equals("Hello"));
    }
}

```

---



O modelo MVC traz como benefício:

- Facilitar o reaproveitamento do código;
- Facilidade na manutenção e adição de recursos;
- Maior integração da equipa e/ou divisão de tarefas;
- Facilidade em manter o seu código sempre limpo.

O padrão MVC pode ser utilizado em vários tipos de projetos como, por exemplo, desktop, web e mobile.

### **4.3 Método de desenvolvimento**

De modo a manter um controle sobre a versão e o que é liberado ao utilizador, foram criados vários ambientes para testes das novas funcionalidades que são implementadas simultaneamente pelos diversos elementos da equipa.

Para tal o Android Studio oferece a possibilidade destes ambientes serem criados, alterando os ambientes consoante o tipo de operações se pretende realizar. Sendo estes desde a implementação a testes que devem ser realizados de modo a nova funcionalidade ser adicionada a versão atualmente disponível ao utilizador.

Fora do ambiente android foi criado um repositório em Azure DevOps (TFS), para o controle de tarefas da equipa. Este faz o uso do Git como ligação entre estes dois ambientes.

Recentemente foi adotada o uso do GitFlow para uma melhor organização dos ramos criados.

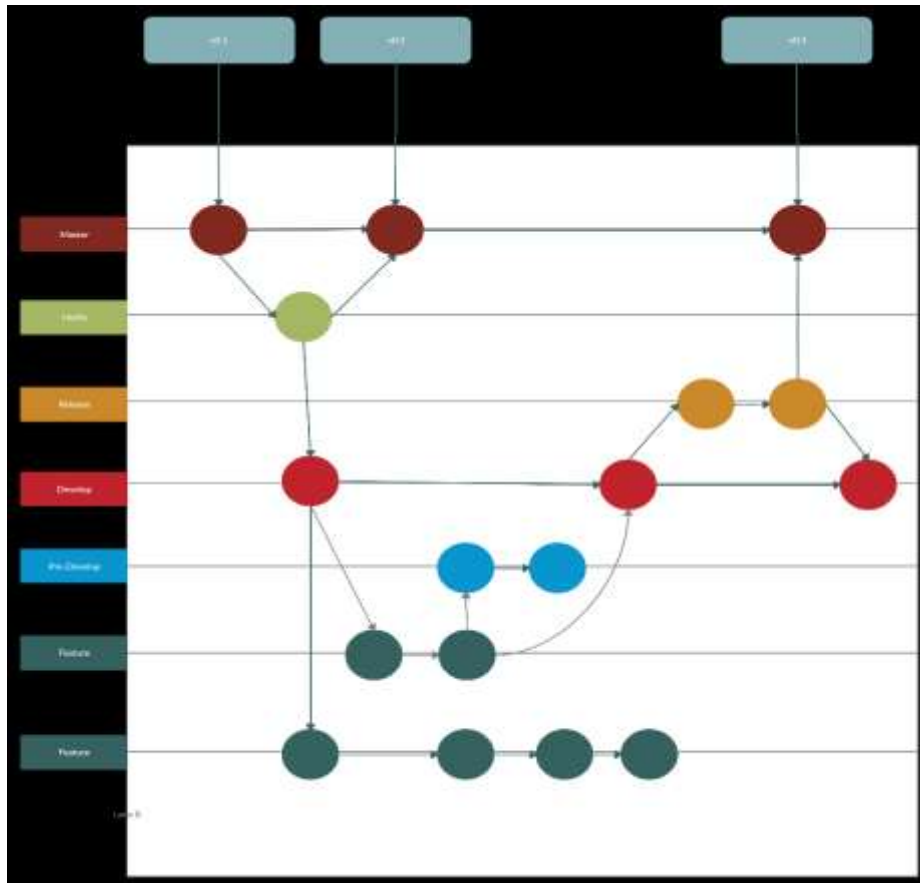


Figura 10 - GitFlow

### 4.3.1 GitFlow

GitFlow [10] é um modelo de ramificação para o Git, criado por Vincent Driessen. É utilizado para gerenciar os ramos melhor e de uma forma mais lógica.

- Master (Mestre) – é o ramo ativo do projeto. Este é um código de produção. É necessário ter cuidado ao confirmar este ramo.
- Hotfix – quando é necessária a resolução de algum problema no master, poderá se usar a ramificação Hotfix para tal, sendo estes de grande importância a sua resolução. Sendo submetido diretamente na Master.
- Develop (Desenvolvimento) – este ramo é um clone do ramo master, e é utilizado para o desenvolvimento de novos recursos e resolução de bugs.

- Release (Lançamento) – quando é terminado o desenvolvimento de novos recursos, estes necessitam de ser testados e é neste ramo que é feita o último teste antes dos novos recursos serem adicionados ao ramo master.
- Feature (Característica) – este ramo é um clone do ramo desenvolvimento, é nele que é feita a codificação dos novos recursos. Quando acabada a codificação esta é mesclada com o ramo desenvolvimento.

O GitFlow não resolve todos os problemas na ramificação. Mas oferece uma estrutura de ramificação mais lógica. E essa estrutura lógica de ramificação fornece um ambiente de trabalho mais eficiente ao longo do tempo.



# Capítulo 5

## Descrição das Novas Funcionalidades

Neste capítulo é feita, de forma generalizada, a demonstração das funcionalidades e bugs corrigidos e a sua descrição.

Tais funcionalidades estão incluídas ao sistema de autenticação (login) necessário para a utilização da aplicação, visto que se trata de contas ligadas a um banco sendo assim dobrada a necessidade de segurança.

Portanto o banco fornece aos utilizadores dados únicos que serão utilizados por um único utilizador ou mais, se assim for o desejo do mesmo.

### 5.1 Revogação de Cartão

Esta funcionalidade tem como objetivo a possibilidade do utilizador anular um cartão pedido recentemente, dentro do espaço de quinze dias.

A revogação é feita através da chamada de um ecrã feito para a confirmação do pedido de revogação, sendo chamado o fragmento de conclusão de revogação aquando da conclusão do fluxo de confirmação.

O fragmento de conclusão recebe um id como argumento, sendo este ligado à revogação em causa caso a revogação seja feita com sucesso. O fragmento também verifica se é necessário a emissão de um recibo, caso seja necessário existe um método no mesmo que chamará um serviço de modo a retornar um PDF com as informações de comprovação.

## **5.2 Rastreamento de Cartão.**

Esta funcionalidade, como o nome indica, serve para o rastreamento dos cartões aquando o seu pedido através da aplicação.

O utilizador tem a opção de escolher diversos tipos de cartões, cada um com os seus diversos benefícios.

Quando é feita a escolha do cartão, será enviado uma nova atividade em que serão apresentadas as estimativas de chegada do cartão ao local de escolha do utilizador.

É ainda oferecida a opção de ativação do mesmo cartão aquando a sua chegada através de um simples click.

## Capítulo 6

# Conclusões

Este projeto foi iniciado com uma fase de formação permitindo a introdução ao Android Studio e a recapitulação de conceitos de java apreendidos durante o curso. Esta fase foi de muita aprendizagem, pois os meus conceitos de programação de aplicações até então eram o mais básico possível. Sendo necessário a compreensão de conceitos mais avançados de modo a ser a avançar para a próxima etapa.

Aquando a finalização desta etapa deu-se início a compreensão do sistema já implementado, sendo necessária uma investigação intensiva, para a melhor compreensão e navegação do projeto que já fora iniciado aquando a minha chegada. Sendo assim me oferecido a oportunidade de compreender melhor a sua estrutura e ajudando para o melhor desenvolvimento das tarefas pedidas.

De seguida, iniciou-se o desenvolvimento. Neste período, foram poucas as dificuldades apresentadas, visto que as tarefas eram claras e objetivas. Foi necessário a adaptação à Azure DevOps de modo a receber as tarefas a serem desenvolvidas.

Sendo este o período de maior aprendizagem e consolidação de conhecimentos, e preparação para o futuro profissional.

Fiquei satisfeito com o trabalho desenvolvido durante este curto espaço de tempo. No que diz respeito à proposta inicial, penso ter conseguido alcançar todos os principais objetivos propostos, sendo estes o desenvolvimento e otimização de aplicativos móveis na plataforma Android.

## 6.1 Trabalho Futuro

Visto que com o passar do tempo as tecnologias vão evoluído rapidamente nunca será possível dizer que a aplicação esta completa, sendo este o motivo do projeto já ter anos de desenvolvimento. Existirá sempre um novo bug para solucionar ou uma nova funcionalidade a ser introduzida às já existentes.

Uma melhoria que poderia ser implementada seria a migração da aplicação parcial ou total para a linguagem Kotlin, que é muito mais simples que o Java, e torna o desenvolvimento muito mais simples para o desenvolvedor.



# Bibliografia

- [1] Udacity, “<https://www.udacity.com/>,” [Online].
- [2] JAVA, “[https://www.java.com/pt\\_BR/](https://www.java.com/pt_BR/),” [Online].
- [3] XML, “<https://www.tecmundo.com.br/programacao/1762-o-que-e-xml-.htm>,” [Online].
- [4] A. Studio, “<https://developer.android.com/about>,” [Online].
- [5] Git, “<https://git-scm.com/>,” [Online].
- [6] UML, “<https://www.atemomento.com.br/diagramas-uml/>,” [Online].
- [7] MVC, “8. <https://medium.com/upday-devs/android-architecture-patterns-part-1-model-view-controller-3baecef5f2b6>,” [Online].
- [8] AsyncTask, “<https://developer.android.com/reference/android/os/AsyncTask>,” [Online].
- [9] RxJava, “<https://medium.com/@nglauber/introdu%C3%A7%C3%A3o-ao-rx-java-com-kotlin-90c58d184c6b>,” [Online].
- [10] GitFlow, “<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>,” [Online].

- 11. <https://pt.stackoverflow.com/>
- 12. <https://www.youtube.com/?gl=PT>
- 13. <https://www.wikipedia.org/>
- 14. <https://www.itsector.pt/pt>



Apêndice A.

## Proposta Original do Projeto

## Android Development

**Supervisores:** Daniel Gomes/Victor Oliveira (ITSector)

**Orientador:** Nuno Rodrigues (IPB)

### 1 Objetivo

Neste projeto pretende-se manter, desenvolver e otimizar aplicativos móveis nas plataformas Android.

### 2 Detalhes

O avanço científico e sociológico tem permitido o aparecimento de diversos problemas e muitos deles são solucionados com o uso tecnológico. Nos dias atuais o uso da computação ubíqua é quase inevitável, e a obtenção de smartphones cresce a cada dia. Desta forma, o desenvolvimento de soluções móveis cresce de igual forma. O uso de aplicações para se comunicar, verificar a saúde, praticar exercícios físicos, são cada vez maiores.

Pretende-se, com este projeto, desenvolver e otimizar aplicações, tendo uma experiência mais profissional e aplicada, fazendo uso de plataformas Android Studio e também de tecnologias como Java e Kotlin.

Atuar na manutenção, construção e otimização dos aplicativos existentes, seguindo os padrões, especificações e layouts definidos. O aluno deve utilizar ferramentas e padrões para consumo de APIs e serviços REST.

### 3 Faseamento

Fase de avaliação de conhecimentos e revisão de conceitos teóricos e práticos.

Fase de integração no projeto.

Fase de atribuição de tarefas.

### 4 Logística

Dimensão da equipa (marcar com X): ☒ 1 aluno; ☐ 2 alunos

Recursos necessários (marcar com X): ☐ portátil pessoal ☒ fornecidos pelo proponente

Regime de trabalho pretendido (marcar com X): ☐ durante o tempo de Projeto no IPB ☒ a tempo inteiro em entidade externa ☐ a tempo parcial em entidade externa ☐ outro (descrever): \_\_\_\_\_

---