

APLICAÇÃO FORMAÇÃO

Carlos António Neto Pacheco
Diogo Miguel Azevedo
Samuel Correia Gonçalves



Departamento de Engenharia Informática e Telecomunicações

Escola Superior de Tecnologia e Gestão de Lamego

2020

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Projeto Final, do 3º ano, da Licenciatura em Engenharia Informática e Telecomunicações.

Candidatos: Carlos António Neto Pacheco, N°3782, estgl3782@estgl.ipv.pt

Diogo Miguel Azevedo, N°3788, estgl3788@estgl.ipv.pt

Samuel Correia Gonçalves, N°3847, estgl3847@estgl.ipv.pt

Orientador: Hazem Hadla, hhadla@estgl.ipv.pt

Coorientador: Fernando Mamede, fsantos@estgl.ipv.pt

Empresa: EgisRoad

Tutor: Rui Couto, rui.couto@egisportugal.pt



Departamento de Engenharia Informática e Telecomunicações

Escola Superior de Tecnologia e Gestão de Lamego

2020

Agradecimentos

Um agradecimento às nossas famílias pelo apoio, quando tomamos a decisão de prosseguir os estudos frequentando um curso de ensino superior.

Ao nosso tutor da empresa Egis Road, Rui Couto, pela disponibilidade e apoio durante a realização deste projeto.

Ao Professor Hazem Hadla por ter aceitado ser nosso orientador e ter demonstrado total disponibilidade para nos ajudar na realização deste projeto.

Por último, mas não menos importante, a todos os nossos colegas com quem convivemos durante o nosso percurso académico, com quem partilhamos experiências, vivências, dificuldades e sucessos.

Resumo

Este documento descreve sucintamente o projeto “App Formação”, foi realizado pelos candidatos acima mencionados, no âmbito da unidade curricular Projeto Final, que decorre no terceiro ano da Licenciatura em Engenharia Informática Telecomunicações da Escola Superior de Tecnologia e Gestão de Lamego.

Este projeto realizado no fim do ciclo de estudos tem como objetivo a criação de uma aplicação informática, que visa permitir aos colaboradores/trabalhadores da empresa obterem a sua formação para poderem fazerem os trabalhos na autoestrada A24. A formação engloba a tomada de conhecimento das regras de segurança para a realização de trabalhos na via e um questionário sobre as regras estabelecidas.

A aplicação foi elaborada recorrendo ao software de programação Android Studio, onde foi utilizada a linguagem de programação JAVA e posteriormente instalada num telemóvel, que utiliza o sistema operativo Android a partir da versão 8.0.

Palavras-Chave

Aplicação Android, Android Studio, Java.

Abstract

This document briefly describes the “App Formação” project, which was carried out by the above-mentioned candidates, in the course Final Project which takes place in the third year of the Degree in Computer Telecommunications Engineering at the School of Technology and Management of Lamego.

This project takes place at the end of the study cycle and aims to create an informatic application, which enables the company's employees / workers to support their training so that they can carry out the work on the A24 motorway. The training includes a knowledge of the safety rules for carrying out work on the road and a questionnaire about the rules established.

The application was developed using the Android Studio programming software, where the JAVA programming language was considered and later installed on a mobile phone, which operates using the Android operating system since version 8.0.

Keywords

Android Application, Android Studio, Java.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABELAS	XI
ACRÓNIMOS.....	XIII
1. INTRODUÇÃO	1
1.1. ORGANIZAÇÃO DO RELATÓRIO.....	2
1.2. CONTEXTUALIZAÇÃO	3
1.3. A EMPRESA	3
1.4. OBJETIVOS E BREVE DESCRIÇÃO DO PROJETO	4
1.5. CALENDARIZAÇÃO	5
2. ESTADO DA ARTE	7
2.1. SISTEMA OPERATIVO ANDROID.....	8
2.2. INTERFACE.....	10
2.3. GESTÃO DE MEMÓRIA	11
2.4. DESENVOLVIMENTO DO SISTEMA OPERATIVO.....	11
2.5. APLICAÇÕES ANDROID	12
2.6. O QUE É UMA LINGUAGEM DE PROGRAMAÇÃO	13
2.7. INTERPRETAÇÃO E COMPILAÇÃO	13
2.8. LINGUAGEM DE PROGRAMAÇÃO JAVA	14
2.9. UMA BREVE HISTÓRIA DA LINGUAGEM DE PROGRAMAÇÃO JAVA	15
3. ESTUDO PRELIMINAR DO PROJETO	18
3.1. ABORDAGEM INICIAL	19
3.2. MATERIAL USADO	20
3.3. ANDROID STUDIO	20
3.3.1 ESTRUTURA DE UM PROJETO.....	21
3.3.2 INTERFACE COM O UTILIZADOR.....	22
3.4. GOOGLE FORMS.....	23
3.5. FIREBASE	24
4. DESENVOLVIMENTO DO PROJETO	27

4.1.	LAYOUT	27
4.2.	DEFINIÇÃO DO LAYOUT E DO CÓDIGO DO LOGIN	28
4.3.	DEFINIÇÃO DO LAYOUT E DO CÓDIGO DO MENU.....	36
4.4.	LAYOUT DO SUBMENU TEÓRICA	46
4.5.	LAYOUT E CÓDIGO DO SUBMENU DOS VISITANTES	52
5.	CONCLUSÕES	56
6.	REFERÊNCIAS DOCUMENTAIS	59
ANEXOS		61

Índice de Figuras

Figura 1 - Logótipo Egis	3
Figura 2 - Logótipo Egis Road Portugal	4
Figura 3 - Logótipo Android	9
Figura 4 - Interface Android.....	10
Figura 5 - Logótipo Google.....	11
Figura 6 - Logótipo PlayStore.....	12
Figura 8 - Logótipo Java	16
Figura 7 - Logótipo Oracle.....	16
Figura 9 - Diagrama de blocos do projeto	19
Figura 10 - Logótipo do software Android Studio	20
Figura 11 - Organização da área Projeto do software Android Studio.....	21
Figura 12 - Janela principal do Android Studio	22
Figura 13 - Logótipo Google Forms	23
Figura 14 - Logótipo FireBase	24
Figura 15 - Paleta de cores da Empresa Egis Road.....	28
Figura 16 - Layout final da página de login	29
Figura 17 - Layout inicial para o menu	36
Figura 18 - Layout final do menu.....	42
Figura 19 - Layout final do submenu da Teórica	49
Figura 20- Google Forms Visitantes	53
Figura 21 - Layout Visitantes.....	54

Índice de Tabelas

Tabela 1 - Calenderização	5
Tabela 2 . Descrição do layout predefinido do Android Studio	23

Acrónimos

API – Application Programming Interface

SO – Sistema Operativo

RAM – Random Access Memory

SDK – Software development kit

APK – Android Package

ART – Android Runtime

JVM – Java Virtual Machine

SSL – Secure Sockets Layer

1. INTRODUÇÃO

Atualmente, vivemos num mundo onde o desenvolvimento e avanços tecnológicos nas diversas áreas de atividade de uma sociedade, são notícia quase diariamente, o que permitiu que surgissem diversos produtos e/ou equipamentos que nos facilitaram a vida quotidiana, desde a mais simples tarefa até às tarefas mais complexas.

Um dos produtos que teve um grande impacto na nossa sociedade foram os telemóveis e as suas funcionalidades. Consoante a evolução da tecnologia, os telemóveis também foram evoluindo, chegando a um ponto em que se tornaram imprescindíveis na nossa vida. Nos dias de hoje conseguimos realizar várias ações no conforto de nossa casa utilizando apenas o nosso telemóvel, desde ações simples como fazer uma chamada até ações mais complexas como uma transferência bancária, algo que seria impensável há alguns anos.

Neste sentido, o ramo da informática aplicado ao ramo dos dispositivos móveis fez com que evoluísse, através da implementação de aplicações móveis entre outros.

Seguindo esta tendência, a Empresa Egis Road, apostou no desenvolvimento desta aplicação, App de Formação, que junta as suas áreas de atividade com o ramo informático.

1.1. ORGANIZAÇÃO DO RELATÓRIO

Este documento está dividido em 5 partes principais: Introdução, Estado da Arte, Estudo preliminar do Projeto, Desenvolvimento do Projeto e Conclusões.

Neste 1º capítulo, não só, é feita uma breve introdução aos conceitos e métodos que serão mencionados e utilizados durante a realização deste projeto, mas também é apresentada a empresa responsável pelo projeto, os seus objetivos e calendarização das diferentes etapas que constituem o projeto.

No capítulo 2, Estado da Arte, é feita uma breve análise acerca do que já existe no mercado relativamente aplicações informáticas (Android), é apresentado um pouco da história do Sistema Operativo Android, uma breve abordagem às aplicações em android e uma breve abordagem à linguagem de programação Java.

No capítulo 3, Estudo Preliminar do Projeto, é apresentada a abordagem inicial e o material que foi utilizado na sua realização, assim como uma explicação do funcionamento do Android Studio.

No capítulo 4, Desenvolvimento do Projeto é apresentado o raciocínio adotado na realização da programação da aplicação Android, em Java.

Por fim no capítulo 5, Conclusões é feita a análise e reflexão sobre o trabalho realizado, a avaliação final da aplicação e a verificação se os objetivos, inicialmente propostos, foram ou não cumpridos, sendo também referidas melhorias ao mesmo.

1.2. CONTEXTUALIZAÇÃO

Este projeto surgiu do desejo de conceber uma aplicação informática que permitisse à empresa Egis Road oferecer uma formação remota aos seus colaboradores. Foi proposta então a conceção da App de Formação totalmente controlada remotamente através de uma aplicação Android instalada num dispositivo móvel

1.3. A EMPRESA

A Egis Group é um grupo de engenharia francesa que começou a sua atividade em 1970, está envolvida nas áreas de sistemas de infraestrutura e transporte, água e meio ambiente. Estando também envolvida no negócio de criação de projetos e operações para estradas e aeroportos [1].



Figura 1 - Logótipo Egis

Este grupo conta com muitas subsidiárias espalhadas pelo mundo inteiro, sendo uma delas a Egis Road Operation Portugal.

A Egis Road Operation Portugal foca a sua atividade na operação e manutenção de infraestruturas, gestão de concessões e ativos, mobilidade, *smart cities*, *smart roads* e implementação de soluções de energias renováveis [2].

A entidade é integralmente detida pelo Grupo Egis, encontrando-se presente em Portugal desde o ano 2000.

Tem atualmente no seu portfolio de serviços, entre outros:

- Operação e manutenção da autoestrada A24;
- Manutenção de equipamentos *Multi Lane Free Flow*;
- Manutenção de *Fixed Operating Equipment* (FOE) em infraestruturas rodoviárias;
- Gestão do ciclo de vida de ativos;
- Inspeções técnicas;
- Implementação de projetos fotovoltaicos;
- Outros serviços relacionados com a gestão de infraestruturas.



Figura 2 - Logótipo Egis Road Portugal

1.4. OBJETIVOS E BREVE DESCRIÇÃO DO PROJETO

O objetivo final deste projeto é a conceção de uma aplicação informática (Android), para isto subdividiu-se esse objetivo em algumas tarefas de realização mais simples, tais como:

- Estudo e conhecimento do ambiente de trabalho Android Studio;
- Estruturação e idealização do design da Aplicação;
- Estudo e elaboração do código necessário, em linguagem Java,

1.5. CALENDARIZAÇÃO

Nesta secção é demonstrada a calendarização das diferentes etapas do projeto, através da Tabela 1. Esta inclui um conjunto de tarefas que constituem o projeto, como por exemplo: o estudo do funcionamento do Android Studio, elaboração do design, elaboração da aplicação Android para dispositivos móveis e realização de testes ao sistema.

Tabela 1 - Calendarização

Etapas	Março	Abril	Maio	Junho
Estruturação e idealização do design da Aplicação. Estudo e conhecimento do ambiente Android Studio e das regras de segurança e do questionário.				
Início ao desenvolvimento do código da aplicação;				
Continuação do desenvolvimento do código. Criação do registo eletrónico e do certificado final;				
Junção de todas as partes pertencentes ao projeto (código, layout...); Realização de testes à aplicação				

2. ESTADO DA ARTE

Neste segmento é feita uma pesquisa e análise do que já existe no mercado até à data, relativamente ao que é proposto fazer no projeto.

Vai-se apresentar um pouco da história do Sistema Operativo Android e das aplicações para Android. No projeto é usada a linguagem de programação Java, sendo explicado o que é uma linguagem programação e a história da linguagem Java.

2.1. SISTEMA OPERATIVO ANDROID

O Android é um sistema operacional (SO) baseado no núcleo Linux, desenvolvido por um consórcio de desenvolvedores conhecido como *Open Handset Alliance*, sendo o principal colaborador a Google [3]. Com uma interface de utilizador baseada na manipulação direta, o Android é projetado principalmente para dispositivos móveis com ecrã sensível ao toque (*Touch Screen*) como smartphones e tablets; com interface específica para TV (Android TV), carro (Android Auto) e relógio de pulso (Android Wear). O sistema operacional utiliza o ecrã sensível ao toque para que o utilizador possa manipular objetos virtuais e também um teclado virtual. Apesar de ser principalmente utilizado em dispositivos com ecrã sensível ao toque, também é utilizado em consolas de videojogos, câmaras digitais, computadores e outros dispositivos eletrónicos.

O Android é o sistema operacional móvel mais utilizado do mundo, e, em 2013, possuía a maior percentagem das vendas mundiais de SO móveis [6]. Dispositivos com o sistema Android vendem mais dispositivos eletrónicos do que Windows, iOS e Mac OS X combinados. Em julho de 2013, a loja de aplicações Google Play possuía mais de 1 milhão de aplicações disponíveis, baixadas mais de 50 biliões de vezes. Uma pesquisa com programadores entre abril e maio de 2013 revelou que 71% dos programadores para sistemas móveis desenvolviam aplicações para Android. Na conferência anual Google I/O de 2014, a companhia revelou que existem mais de 1 bilião de utilizadores Android ativos. Em junho de 2013, este número era de 538 milhões [6]. O maior número de utilizadores no mundo também reflete no número de ataques de hackers no sistema, com cerca de 5.000 novos *malwares* sendo criados todos os dias para Android.

O código do sistema operacional é disponibilizado pela Google sob uma licença de código aberto, apesar de a maior parte dos dispositivos ser lançado com uma combinação de software livre e software privado. Inicialmente foi desenvolvido pela empresa Android, Inc., que foi fundada em Palo Alto, California em outubro de 2003 por Andy Rubin (cofundador da *Danger*), Rich Miner (cofundador da *Wildfire Communications, Inc.*), Nick Sears (ex-vice-presidente da *T-Mobile*), e Chris White para desenvolver, segundo Rubin, "dispositivos móveis mais inteligentes que estejam mais cientes das preferências e da localização do seu dono". Empresa que a Google dava suporte financeiramente. Foi comprada pela mesma em 2005 e revelado em 2007 junto com a fundação da Open

Handset Alliance — consórcio entre empresas de hardware, software e telecomunicações com o intuito de desenvolver a indústria de dispositivos móveis.

O Android é muito popular entre empresas de tecnologia que procuram um software pronto, de baixo custo e personalizável para dispositivos de alta tecnologia. A natureza do software de código aberto do sistema operacional tem encorajado uma grande comunidade de programadores e entusiastas a colocar uma fundação para o desenvolvimento de projetos feitos pela própria comunidade que adicionam recursos para utilizadores mais avançados, ou trazem o Android para dispositivos que inicialmente não foram lançados com a plataforma. O sucesso do SO fez dele um alvo para disputas de patente na chamada "guerra de smartphones" entre empresas de tecnologia.



Figura 3 - Logótipo Android

2.2. INTERFACE

As notificações são acedidas deslizando-se a barra do topo para baixo, notificações podem ser dispensadas individualmente simplesmente deslizando-as para o lado, e podem conter ainda funções adicionais, como retornar chamada e enviar mensagens. A interface padrão do usuário no Android é baseada na manipulação direta. A resposta para a manipulação é desenhada para ser imediata e produzir uma sensação de fluidez, utilizando-se constantemente da resposta háptica, ou seja, de resposta ao tato para informar o usuário sobre a conclusão do comando.

Dispositivos Android são direcionados ao ecrã inicial ou *homescreen* assim que são ligados. Lá encontramos a navegação primária e principal do sistema, e que é similar ao conceito de desktop utilizado nos computadores. São tipicamente compostas por ícones de aplicações instalados e *widgets*. Os ícones, quando selecionados, direcionam o usuário à aplicação correspondente, enquanto o *widget* mostra na própria tela inicial as informações, atualizando constantemente seu conteúdo, como previsão do tempo, emails não lidos ou notícias.

Presente no topo da tela está a status bar ou barra de estado, a qual mostra informações sobre o dispositivo e sua conectividade. Esta barra pode ser puxada para baixo para mostrar a tela de notificações onde aplicações mostram importantes informações ou atualizações, como um novo email ou uma nova mensagem. As notificações são persistentes e não saem da tela de notificações a não ser que o utilizador as leia, clicando nas notificações, ou as dispense, deslizando-as para o lado [4].

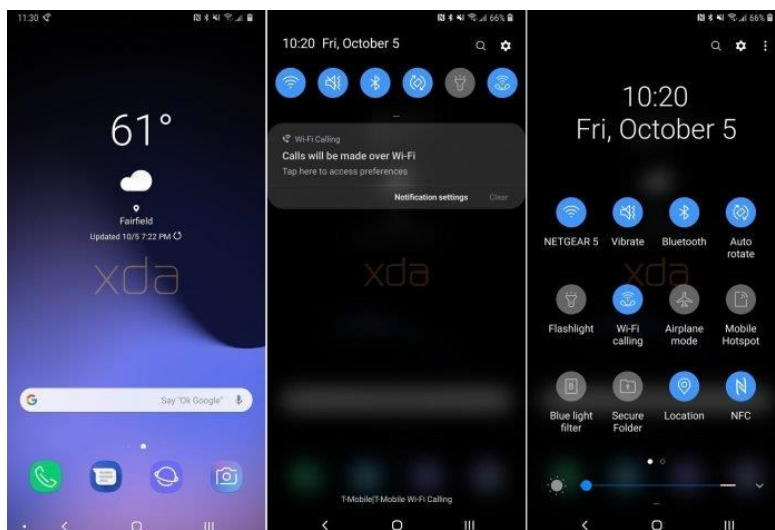


Figura 4 - Interface Android

2.3. GESTÃO DE MEMÓRIA

Já que os dispositivos Android são normalmente movidos a bateria, o sistema foi desenvolvido para gerir a quantidade de memória *Random Access Memory* (RAM) com o intuito de poupar a quantidade de energia gasta, em contraste com os computadores em que normalmente estão conectados à rede elétrica. Quando uma aplicação não está a ser usada, o sistema automaticamente suspende-o da memória; as aplicações suspensas não consomem nenhum recurso do sistema. Isso traz um benefício duplo, já que aumenta a responsividade do sistema, enquanto também não consome energia sem necessidade [4].

O Android gere aplicações na memória automaticamente: quando a memória está baixa, o sistema começa a dispensar aplicativos e processos que estão inativos há determinado tempo. Este processo é feito para ser invisível ao utilizador. Porém, a confusão sobre a regulação de memória no Android resultou que aplicações de terceiros que gerem a memória ficaram famosos na Google Play Store.

2.4. DESENVOLVIMENTO DO SISTEMA OPERATIVO

O SO móvel Android é desenvolvido em segredo pela Google até a última das modificações, quando estará pronto para ser lançado. Então, o código é lançado publicamente. O código só funcionará sem modificações em determinados aparelhos. O código é então adaptado a cada fabricante para que corra nos hardwares específicos, com ou sem modificações.

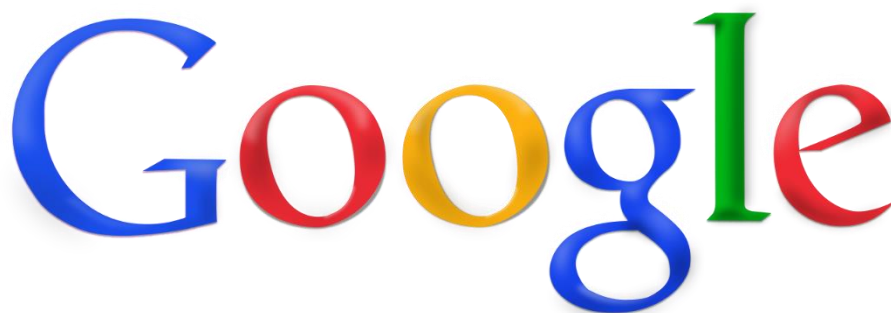


Figura 5 - Logótipo Google

2.5. APLICAÇÕES ANDROID

Aplicações ou apps que estendem funcionalidades dos dispositivos são desenvolvidos primariamente na linguagem Java, usando-se do sistema de desenvolvimento do software Android (SDK, Software development kit). O SDK inclui um compreensivo conjunto de ferramentas de desenvolvimento, que incluem um depurador, uma biblioteca (computação), um emulador baseado em QEMU que é um software livre que implementa um emulador de processador que permite uma virtualização completa de um sistema de um computador dentro de outro, documentação, códigos de exemplos e tutoriais. Outras ferramentas de desenvolvimento estão disponíveis, como o Kit de Desenvolvimento Nativo para aplicativos ou extensões em C ou C++, o Google App Inventor, um ambiente de desenvolvimento para novos desenvolvedores, e múltiplas plataformas de desenvolvimento cross-plataform [4].

O Android tem crescido na quantidade de aplicações de terceiros desenvolvidos para o sistema. Tais aplicativos podem ser adquiridos pelos utilizadores através do download e instalação do arquivo Android Package (APK) ou através do download utilizando um programa de alguma loja de aplicações virtuais como a PlayStore. A PlayStore é a loja primária dos utilizadores do Android. A Google PlayStore permite ao utilizador do Android navegar, comprar, fazer o download e atualizar as aplicações desenvolvidas pela Google ou por terceiros.

Devido à natureza aberta do Android, há também um alto número de aplicações de lojas de terceiros, tanto para disponibilizar aplicações para dispositivos não permitidos na PlayStore, quanto para colocar à disposição aplicações não permitidas na loja, ou aplicações que burlam o sistema de pagamento do Google, fazendo que saiam de graça para o utilizador.



Figura 6 - Logótipo PlayStore

2.6. O QUE É UMA LINGUAGEM DE PROGRAMAÇÃO

Uma linguagem de programação é uma estrutura padronizada para criar comandos para um computador. Podemos dizer que é um conjunto de regras usadas para definir um programa de computador. A linguagem de programação permite que um programador especifique precisamente como um computador vai utilizar os dados um computador, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias.

As Linguagens de programação também são usadas para construir algoritmos e fórmulas que para resolver problemas utilizando software. O conjunto de palavras, juntamente com essas regras, formam o código fonte de um software ou aplicações. Esse código fonte é depois traduzido para código de máquina, que é executado pelo microprocessador. Uma das principais propostas das linguagens de programação é que os programadores tenham uma maior produtividade, permitindo criar códigos mais facilmente do que quando comparado com a linguagem que um computador entende nativamente, o código de máquina [6].

No caso da plataforma Android, todos os telemóveis que contenham o sistema operacional do Google têm uma ferramenta intermediária que os ajuda a transformar uma linguagem de programação em código de máquina.

2.7. INTERPRETAÇÃO E COMPILAÇÃO

Uma linguagem de programação pode ser convertida, ou traduzida, em código de máquina por compilação ou interpretação. Em ambas, ocorre a tradução do código fonte para código de máquina. Se o método utilizado traduz todo o código do programa, para só depois executar o programa, então diz-se que o programa foi compilado e que o mecanismo utilizado para a tradução é um compilador.

A versão compilada do programa é armazenada de forma que o programa pode ser executado um número indefinido de vezes sem que seja necessária uma nova compilação, o que compensa o tempo gasto na compilação.

Se o código do programa é executado à medida que vai sendo traduzido, então diz-se que o programa foi interpretado e que o mecanismo utilizado para a tradução é um interpretador. Embora haja essa distinção entre linguagens interpretadas e compiladas, as coisas nem sempre são tão simples.

No caso do sistema operacional Android, o código fonte *Java* ou *Kotlin* é compilado para um código de máquina virtual chamado *bytecode*, que depois é traduzido novamente pela Android Runtime (ART) para código de máquina do Android.

2.8. LINGUAGEM DE PROGRAMAÇÃO JAVA

Para qualquer desenvolvedor de aplicações moveis em todo o mundo, a primeira e mais preferida linguagem de programação para uma aplicação Android é o Java, uma das razões pelas quais é simplesmente a linguagem de programação oficial para desenvolvimento de aplicações Android. É a linguagem mais suportada pelo Google e o que a maioria das aplicações da PlayStore são criados.

Mas porquê usar Java?

Entender um pouco da história da plataforma Java é essencial para entender os motivos que levaram ao seu sucesso. Os maiores problemas quando esta linguagem de programação foi criada eram ponteiros, gestão de memória, organização, falta de bibliotecas, ter de reescrever parte do código ao mudar de sistema operacional.

A linguagem Java resolve esses problemas, que até então apareciam com frequência nas outras linguagens. Alguns desses problemas foram particularmente resolvidos porque uma das grandes motivações para a criação da plataforma Java era de que essa linguagem fosse usada em pequenos dispositivos. Apesar disso a linguagem teve seu lançamento focado no uso em clientes web (browsers) para correr pequenas aplicações.

Atualmente esse não é o grande mercado do Java: apesar de ter sido idealizado com um propósito e lançado com outro, o Java ganhou destaque no lado do servidor [5].

2.9. UMA BREVE HISTÓRIA DA LINGUAGEM DE PROGRAMAÇÃO JAVA

A Sun Microsystems criou uma equipa (conhecido como Green Team) para desenvolver inovações tecnológicas em 1992 sendo mais tarde em 2009 adquirida pela Oracle, fortalecendo a marca. A Oracle sempre foi, junto com a IBM, uma das empresas que mais investiram e fizeram negócios através do uso da plataforma Java. Esta equipa foi liderada por James Gosling, considerado o pai do Java. A equipa voltou com a ideia de criar um interpretador para pequenos dispositivos, facilitando a reescrita de software para aparelhos eletrónicos, como televisão e aparelhos de TV a cabo.

A ideia não deu certo. Tentaram fechar diversos contratos com grandes fabricantes de eletrónicos, como Panasonic, mas não houve êxito devido ao conflito de interesses e custos. Hoje, sabemos que o Java domina o mercado de aplicações para telemóveis com mais de 2.5 bilhões de dispositivos compatíveis. Com o advento da web, a Sun Microsystems percebeu que poderia utilizar a ideia criada em 1992 para funcionar pequenas aplicações dentro do browser. A semelhança era que na internet havia uma grande quantidade de sistemas operacionais e browsers, e com isso seria grande vantagem poder programar numa única linguagem, independente da plataforma. Foi aí que o Java 1.0 foi lançado: focado em transformar o browser de apenas um cliente magro (thin client ou terminal burro) em uma aplicação que possa também realizar operações avançadas, e não apenas carregar html [6].

Java é uma linguagem de programação orientada a objetos. Diferente das linguagens de programação modernas, que são compiladas para código nativo, a linguagem Java é compilada para um *bytecode* que é interpretado por uma máquina virtual (*Java Virtual Machine*, mais conhecida pela sua abreviação JVM).

A linguagem Java foi projetada tendo em vista os seguintes objetivos:

Orientação a objetos - Baseado no modelo de Simular;

- Portabilidade - Independência de plataforma - "escreva uma vez, execute em qualquer lugar" ("*write once, run anywhere*");
- Recursos de Rede - Possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP;
- Segurança - Pode executar programas via rede com restrições de execução.

Além disso, podem-se destacar outras vantagens apresentadas pela linguagem:

- Sintaxe similar a C/C++
- Facilidades de Internacionalização - Suporta nativamente caracteres Unicode;
- Simplicidade na especificação, tanto da linguagem como do "ambiente" de execução (JVM);
- É distribuída com um vasto conjunto de bibliotecas (ou APIs);
- Possui facilidades para criação de programas distribuídos e multitarefa (múltiplas linhas de execução num mesmo programa);
- Deslocação de memória automática por processo de coletor de lixo;
- Carga Dinâmica de Código - Programas em Java são formados por uma coleção de classes armazenadas independentemente e que podem ser carregadas no momento de utilização.



Figura 8 - Logótipo Oracle



Figura 7 - Logótipo Java

3. ESTUDO PRELIMINAR DO PROJETO

Após ser feita a apresentação do projeto na sua forma integral por parte do orientador da empresa, foram discutidos qual seria a abordagem mais correta a fazer para o desenvolvimento do projeto proposto e quais seriam os softwares a serem utilizados para a concepção do projeto. Durante este capítulo é apresentada a abordagem que foi seguida para a realização deste projeto, quais os softwares utilizados, bem como uma explicação da razão pela opção da sua utilização.

3.1. ABORDAGEM INICIAL

Após discussão com o orientador da empresa foi definido que a aplicação a desenvolver teria de ser composta por duas partes fundamentais. A primeira parte é a teórica onde o utilizador carrega num botão e é redirecionado para uma página que contém a toda a informação disponibilizada pela empresa. A segunda parte é o exame, temos um segundo botão que redireciona os utilizadores para a parte do exame, onde terão de responder a questões relativas à parte teórica. Depois de concluído o exame as respostas por parte do utilizador são guardadas no ficheiro Excel. Estes dois botões estariam localizados num menu que é apresentado quando se inicializa a aplicação.

Desenvolvemos um diagrama blocos, apresentado na figura 9, no qual é apresentado uma noção do que se será realizado.

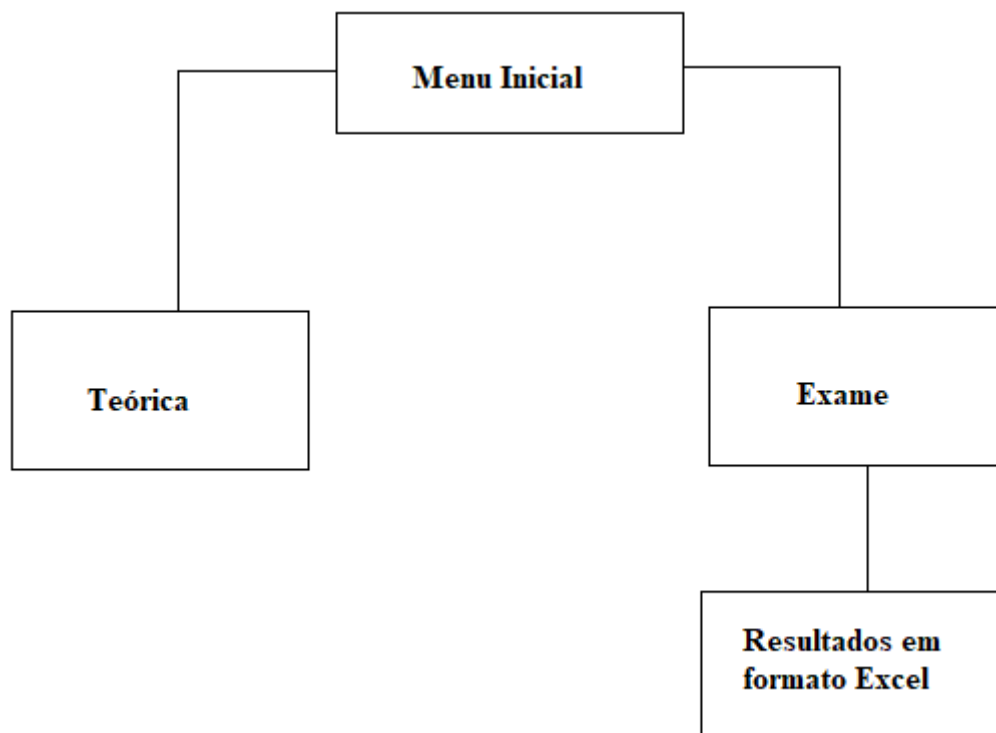


Figura 9 - Diagrama de blocos do projeto

3.2. MATERIAL USADO

Neste projeto além dos computadores e ligação à internet utilizamos dois programas que foram essenciais para a realização deste projeto, o Android Studio, e FireBase.

3.3. ANDROID STUDIO

Para a programação da aplicação Android foi utilizado o software de programação Android Studio, apresentado na Figura 10. Este software foi lançado em maio de 2013, o ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de aplicações Android é baseado em IntelliJ IDEA, que é um ambiente de desenvolvimento integrado escrito em Java para o desenvolvimento de software de computador. Por sua vez, IntelliJ IDEA é um Java IDE desenvolvido pela JetBrains [11].



Figura 10 - Logótipo do software Android Studio

O software está disponível gratuitamente [10] e é compatível com os sistemas operativos Windows, Linux e Mac OS X e substituiu o software Eclipse Android *Development Tools* como a IDE principal do Google para o desenvolvimento de aplicações Android. Além do editor de código e das ferramentas de desenvolvimento avançadas do *IntelliJ*, este software disponibiliza ainda mais recursos que facilitam a criação de aplicativos Android, tais como: modelos de código e integração com o GitHub que contribuem para criação de recursos comuns dos aplicativos, a possibilidade de importar exemplos de código; ferramentas e estruturas de teste cheias de possibilidades, ferramentas de verificação de código realizado para a detecção de problemas de desempenho e compatibilidade com outras versões; compatibilidade com C++ e NDK.

3.3.1 ESTRUTURA DE UM PROJETO

Ao ser criado um projeto, o software carrega automaticamente um ou mais módulos com arquivos de códigos-fonte e recursos necessários, como por exemplo algumas bibliotecas. Por padrão o Android Studio organiza estes arquivos na área Project, onde estão inseridos todos os arquivos referentes ao projeto criado. Esta área está organizada por pastas que possibilita o acesso rápido aos principais arquivos do projeto, como se pode observar na Figura 11.

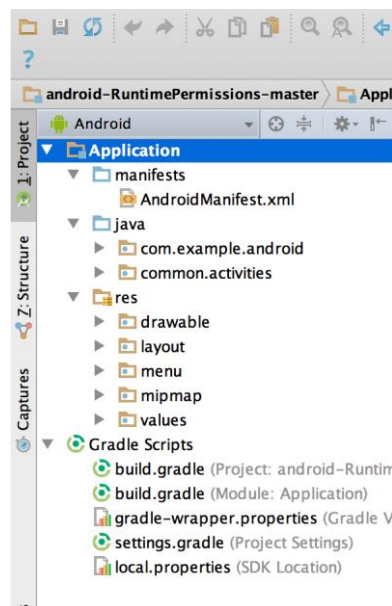


Figura 11 - Organização da área Projeto do software Android Studio

Os arquivos referidos são divididos em três diferentes tipos:

- Manifestos: que contém o arquivo AndroidManifest.xml;
- Java: que contém os códigos em linguagem Java necessários para o funcionamento da aplicação que está a ser desenvolvida;
- Recursos: contém ficheiros que não são código, como layouts, strings ou imagens.

É também possível personalizar a visualização dos arquivos do projeto com o objetivo de focar em aspetos específicos do desenvolvimento do projeto.

3.3.2 INTERFACE COM O UTILIZADOR

A janela principal do Android Studio, apresentada na Figura 12, é composta por diversas áreas, e está organizada segundo um layout predefinido, descrito na Tabela 2 mostrada em seguida, mas que não impede o programador de realizar as alterações que desejar

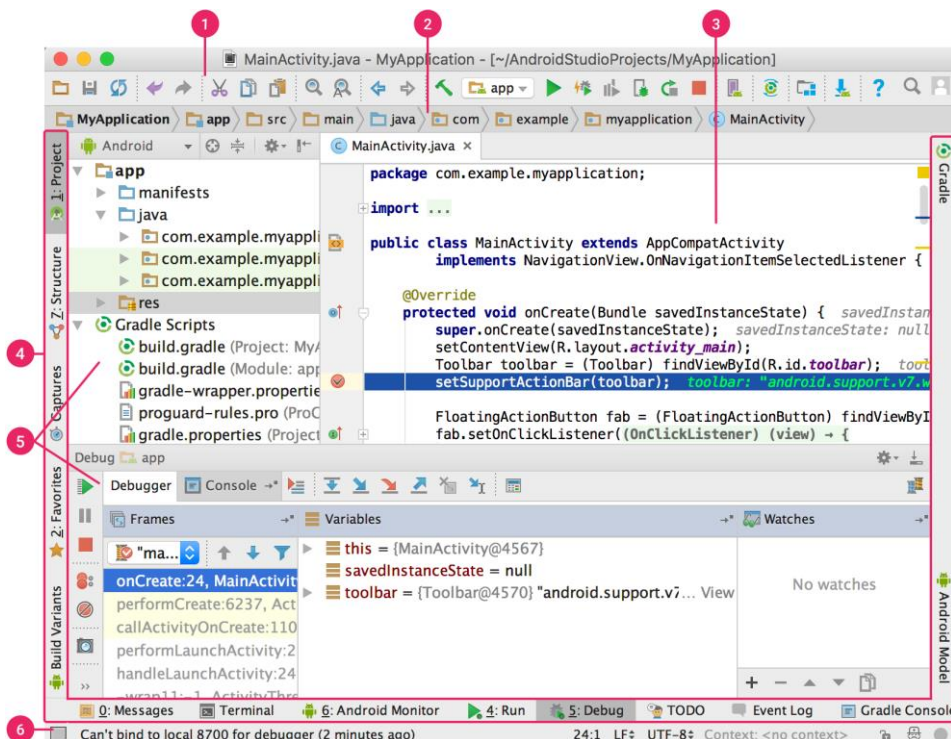


Figura 12 - Janela principal do Android Studio

Tabela 2 . Descrição do layout predefinido do Android Studio

1	Barra de ferramentas	Ferramentas que permitem a gestão da aplicação. As ações podem ser distinguidas em dois tipos: as ações simples, como abrir e guardar projetos, e as ações de compilação/simulação.
2	Barra de navegação	Permite navegar pelas diferentes pastas do projeto de forma simples, fornecendo uma visão compacta da área Project.
3	Janela de editor	Local onde é criado e/ou modificado o código dos ficheiros.
4	Barra de janela de ferramentas	Contém botões que permitem expandir/recolher a janela de cada ferramenta
5	Janela de ferramentas	Acesso a tarefas específicas para controlo do projeto.
6	Barra de <i>status</i>	Mensagens de erro ou de advertência relativas ao projeto ou ao próprio IDE.

3.4 GOOGLE FORMS

O Google Forms é uma aplicação de gestão de pesquisas lançada pela Google. Os utilizadores podem usar o Google Forms para pesquisar e guardar informações sobre outras pessoas e também pode ser usado para questionários e formulários de registo. As informações guardadas e os resultados do questionário serão transmitidos automaticamente e podem ser guardados e exportados no ficheiro do tipo Excel.



Figura 13 - Logótipo Google Forms

3.5 FIREBASE

Firebase é uma plataforma de desenvolvimento aplicações móveis e web adquirida pela Google em 2014. Com foco em ser um back-end completo e de fácil utilização, esta ferramenta disponibiliza diversos serviços diferentes que auxiliam no desenvolvimento e gerenciamento de aplicações [10].



Figura 14 - Logótipo FireBase

Para utilizar o Firebase, foi criada uma aplicação web facilitar a implementação. Nesta, o desenvolvedor adiciona um projeto e inclui os serviços que desejar. Nem todos os serviços são grátis, porém é possível criar um plano conforme as necessidades do desenvolvedor, caso seja necessário algo a mais do que já é oferecido gratuitamente.

O FireBase oferece vários serviços como por exemplo:

- *Realtime Database*: Banco de dados que sincroniza os dados com os dispositivos em tempo real.
- *Cloud Messaging*: Permite enviar mensagens para os utilizadores através da aplicação.
- *Storage*: Armazena dados do utilizador através de uma ligação segura e permite o compartilhamento dos mesmos.
- *Hosting*: Serviço para hospedagem de sites com certificado SSL (Secure Sockets Layer).
- *Notifications*: Envia notificações personalizadas para o utilizador.

O Firebase oferece serviços muito poderosos e de fácil implementação, e a maioria deles pode ser usada de forma gratuita, dentro dos vários serviços fornecidos pelo Firebase, foi usado para o nosso projeto o serviço de Authentication, que possibilita a autenticação através de contas do Google, Facebook, Twitter, Github ou um sistema de contas próprio que é o nosso caso.

4. DESENVOLVIMENTO DO PROJETO

Durante este capítulo são apresentados os vários layouts que foram utilizados na realização deste projeto, parte dos recursos utilizados foram fornecidos pela própria empresa – paleta de cores, links referentes ao site da empresa, ao LinkedIn, etc... - e uma explicação objetiva do código referente aos mesmos.

4.1. LAYOUT

Ao longo da execução deste projeto, surgiram várias ideias de como seria o layout da aplicação. Ideias essas que se centraram principalmente na paleta de cores usadas, bem como a posição dos vários objetos e textos utilizados nos mesmos.

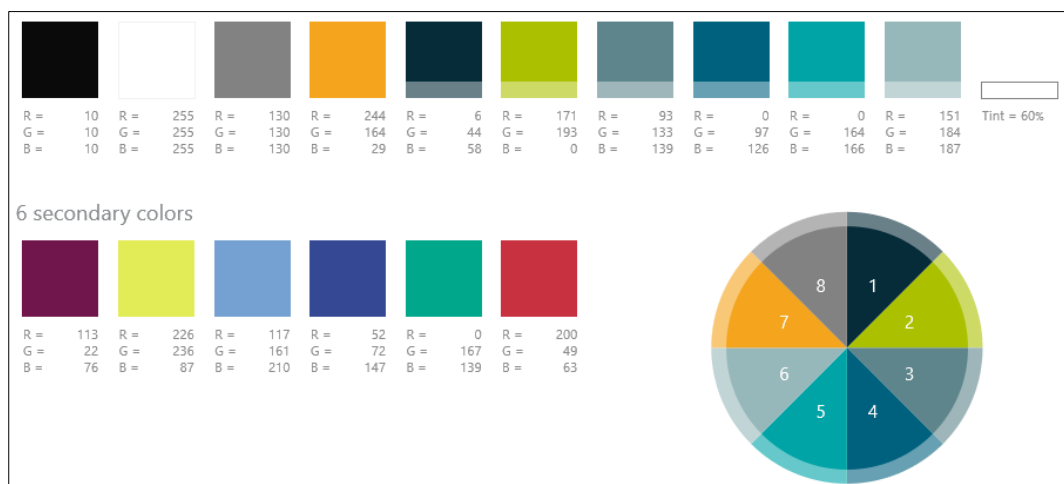


Figura 15 - Paleta de cores da Empresa Egis Road

4.2. DEFINIÇÃO DO LAYOUT E DO CÓDIGO DO LOGIN

Inicialmente a aplicação não tinha nenhum design definido para esta parte, visto que o login foi um elemento que não tinha sido idealizado. Durante a construção da aplicação, e depois de várias ideias trocadas com o tutor do projeto, tivemos a necessidade de criar um login, visto que, o objetivo desta aplicação é para uso interno da empresa, ou seja, ser uma aplicação na qual a empresa possui total controlo acerca dos seus utilizadores e não uma aplicação pública como todas as outras que existem na PlayStore – serviço de distribuição digital de aplicações da Google.

Ao fim de várias ideias discutidas, para este login, definimos que as cores principais seriam o branco (#FFFFFF) seria o verde (#FF55D394) de acordo com a paleta de cores definidas pela empresa (Figura 15) – e que teria um design simples e fácil de usar por parte dos trabalhadores da empresa. É um design que possui principalmente o botão para efetuar a autenticação por parte do utilizador, possui o nome da empresa e o nome da aplicação bem como duas caixas de texto nas quais o utilizador coloca o email e a palavra passe, ambos fornecidos pela empresa, visto que, como já tinha referido anteriormente não existe a opção de criação de conta por parte de quem faz o download da aplicação.

Para a execução das autenticações, foi criada uma database online cujo seu nome é Firebase, na qual é possível, por parte da empresa, atribuir um email e uma palavra passe aos trabalhadores da mesma, editar esses mesmos atributos ou até mesmo eliminá-los caso o trabalhador não frequente mais a empresa. Esta database terá de ser iniciada através de uma conta pertencente ao responsável pela formação dos trabalhadores, para que, unicamente esse mesmo responsável tenha acesso aos dados, quer sejam emails ou palavras passe, dos trabalhadores. Isto é, apenas esse responsável, poderá atribuir dados, editar e/ou eliminar os mesmos. Através deste processo, foi possível então fazer com que a aplicação fosse privada, ou seja, só quem possui os dados fornecidos pelo responsável é que tem acesso à aplicação.

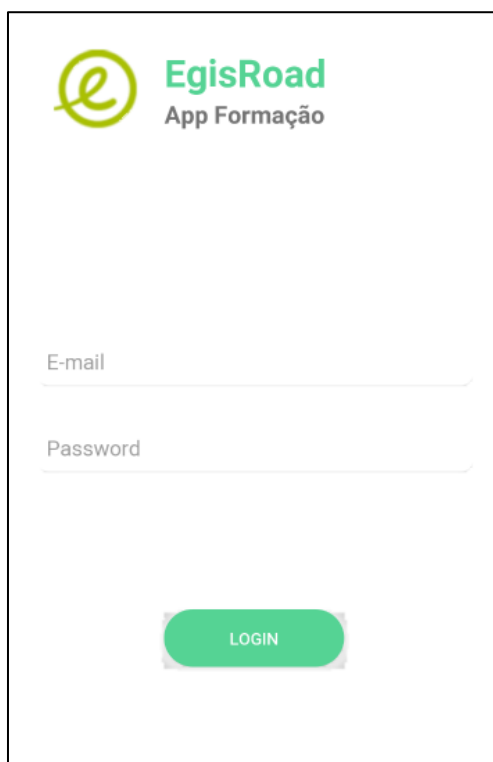
The image shows a login page for the 'EgisRoad App Formação'. At the top left is a logo consisting of a green '@' symbol. To its right, the text 'EgisRoad' is displayed in green, with 'App Formação' in a smaller, grey font below it. The page features two input fields: the first is labeled 'E-mail' and the second is labeled 'Password', both in a light grey font. Below these fields is a green, rounded rectangular button with the word 'LOGIN' in white capital letters.

Figura 16 - Layout final da página de login

Para chegarmos à estrutura final do layout, foi necessário executar alguns códigos referentes ao botão do login, ao nome da empresa e aplicação, que foram definidos como “*TextView*” bem como as caixas de texto, que neste caso foram definidas como “*EditText*”, visto que, é um tipo de texto no qual o utilizador da aplicação irá inserir o email e a palavra passe.

Primeiro foi inserida a imagem da empresa, com um “*ImageView*”, no canto superior esquerdo do layout e definimos o seu tamanho.

```
<ImageView
    android:id="@+id/imageView2"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginTop="12dp"
    android:src="@mipmap/egis2_foreground"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Depois definiu-se dois “*TextView*”, aos quais foram dados os nomes de “EgisRoad” e de “App Formação”. Tanto para o primeiro como para o segundo “*TextView*”, definiu-se várias funções android ilustrativas, tais como, a cor do texto – que mais uma vez foi escolhida a cor verde mencionada anteriormente- as margens, o tamanho do texto, o estilo de letra, e um id para que fosse mais fácil identificar esses mesmos “*TextView*” e para ser possível serem melhores organizados no design.

Para a posição da primeira, alinhamos á direita do logotipo da empresa inserido anteriormente.

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="120dp"
    android:layout_marginTop="52dp"
    android:layout_marginEnd="136dp"
    android:text="EgisRoad"
    android:textColor="@color/colorPrimary"
    android:textSize="32sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/imageView2"
    app:layout_constraintTop_toTopOf="parent" />
```


Para a posição da segunda, alinhamos com a base do primeiro texto.

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="118dp"
    android:layout_marginEnd="136dp"
    android:text="App Formação"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/imageView2"
    app:layout_constraintTop_toBottomOf="@+id/textView" />
```

De seguida foram criados dois “*EditText*” em que um é referente à inserção do email e outro referente à inserção da palavra passe. Mais uma vez, para os dois “*EditText*”, definiu-se várias funções android ilustrativas, tais como, o tamanho da caixa de texto, as margens, o tamanho do texto, o estilo de letra, um id para que fosse mais fácil identificar os “*EditText*”. Para que o utilizador possa introduzir o seu email, foi necessário declarar o método de entrada de texto, através de uma função android referente ao tipo de input do texto (*android:inputType="textEmailAddress"*), e o mesmo teve de ser feito para a palavra passe (*android:inputType="textPassword"*) para que a mesma apareça oculta.

```
<EditText
    android:id="@+id/email"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="24dp"
    android:layout_marginBottom="32dp"
    android:background="@drawable/rounded_box"
    android:backgroundTint="@android:color/white"
    android:elevation="10dp"
    android:ems="10"
    android:hint="E-mail"
    android:inputType="textEmailAddress"
    android:padding="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2"
    app:layout_constraintVertical_bias="0.29000002" />
```

Também foi criado uma barra de progresso com certos campos como a sua posição relacionada aos outros campos inseridos no layout, a sua cor e a sua visibilidade. Esta barra

irá permanecer invisível (*android:visibility="invisible"*) até que seja feita uma confirmação dos dados de login.

```
<ProgressBar
    android:id="@+id/progressBar"
    style="?android:attr/progressBarStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:indeterminateTint="@color/colorPrimary"
    android:visibility="invisible"
    app:layout_constraintBottom_toTopOf="@+id/login"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/password" />
```

Por fim, foi também criado o botão “login” para que o utilizador possa confirmar os dados inseridos e consequentemente iniciar sessão na aplicação. Para este mesmo botão foi, tal como para os anteriores objetos, definidas várias funções android, tais como, o tamanho do botão, as margens arredondadas, o tamanho do texto, o estilo de letra e um id para que fosse mais fácil identificar o mesmo na construção do código da aplicação. As funções implementadas têm como objetivo, tornar o botão mais apelativo, com um design simples e que tivesse contraste em relação a todo o layout.

```
<Button
    android:id="@+id/login"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="115dp"
    android:layout_marginEnd="24dp"
    android:background="@drawable/rounded_corners"
    android:text="Login"
    android:textColor="@android:color/white"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/password"/>
```

Relativamente à estrutura do código em si, primeiramente definiu-se as classes privadas para o “*EditText*”, para o “*Button*”, para a “*FirebaseAuth*” e para o “*ProgressBar*”.

```
package com.example.egisquiz;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class MainActivity extends AppCompatActivity {
    private EditText email, password;
    private Button login;
    private FirebaseAuth firebaseAuth;
    private ProgressBar progressBar;
```

Este código é criado automaticamente pelo Android Studio de modo a atribuir tudo o conteúdo desta página ao seu respetivo layout. Isto é criado para todas as páginas.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
```

A inicialização da autenticação da database online é feita através do comando “*firebaseAuth = FirebaseAuth.getInstance();*”, e é feita antes de o utilizador inserir os dados, pois os dados serão verificados quando o utilizador carregar no botão do login.

Com a criação do “*intent*” iremos fazer a mudança da página de Login para o menu principal da aplicação.

Se a sessão for confirmada (as credenciais estiverem corretas) ele irá iniciar o “*intent*” redirecionando o utilizador para o menu principal.

```

        email = findViewById(R.id.email);
        password = findViewById(R.id.password);
        login = findViewById(R.id.login);
        progressBar = findViewById(R.id.progressBar);
        // Initialize Firebase Auth
        firebaseAuth = FirebaseAuth.getInstance();
        final Intent intent = new Intent(this, Menu.class);

if (firebaseAuth.getCurrentUser() != null){
    startActivity(intent);
    finish();
    return;}

```

Na teórica o botão “login” foi criado juntamente com todos os outros objetos do layout, mas em termos de prática, para o botão ser um elemento funcional, foi utilizada uma função chamada “*login.setOnClickListener*”. Através deste comando o utilizador ao carregar neste botão pode acontecer várias coisas.

Primeiro caso os campos do e-mail e/ou password estejam vazios iram receber mensagens a dizer que será necessário preencher os respetivos campos. Sempre que clicarem no botão “Login” a “*progressBar*” irá mudar a sua visibilidade para visível.

```

login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (email.getText().toString().isEmpty()) {
            email.setError("Introduza o e-mail");
            return;
        } else {
            email.setError(null);
        }
        if (password.getText().toString().isEmpty()) {
            password.setError("Introduza a password");
            return;
        } else {
            password.setError(null);
        }
        progressBar.setVisibility(View.VISIBLE);
    }
}

```

Caso sejam introduzidos elementos nos campos do e-mail e da password ele irá ligar a database e verificar se este login foi criado. Caso exista eles irão receber uma mensagem de confirmação e a serão redirecionados para a página do menu principal.

Caso os dados sejam incorretos ele apenas irá apresentar uma mensagem de erro.

```
firebaseAuth.signInWithEmailAndPassword(email.getText().toString(), password.getText().toString()).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
    @Override  
    public void onComplete(@NonNull Task<AuthResult> task)  
    {  
        if (task.isSuccessful()) {  
            Toast.makeText(MainActivity.this, "Login efetuado com sucesso", Toast.LENGTH_SHORT).show();  
            startActivity(intent);  
            finish();  
        } else {  
            Toast.makeText(MainActivity.this, "As credencias estão incorretas", Toast.LENGTH_SHORT).show();  
        }  
        progressBar.setVisibility(View.INVISIBLE);  
    }  
});  
}
```

4.3. DEFINIÇÃO DO LAYOUT E DO CÓDIGO DO MENU

Neste subcapítulo irá ser discutido todos os aspetos referentes ao layout do menu, desde as várias ideias que surgiram para o design do mesmo, cores usadas, estrutura do menu, os vários botões bem como o código executável deste mesmo layout.

Tal como no layout anteriormente, surgiram várias ideias para o design do menu. Inicialmente o primeiro layout (Figura 17), era bastante “pesado”, visto que possuía bastantes objetos tais como botões, vários logos das redes sociais que depois foram retirados, vários textos referentes ao nome da empresa, ao nome da auto estrada na qual trabalham, número de telefone, etc...



Figura 17 - Layout inicial para o menu

Durante uma reunião que foi realizada com o tutor do projeto, ficou decidido que o layout teria de ser alterado. Essa alteração levou a que o design do layout final fosse baseado no layout do login. Isto é, foram utilizadas cores a partir da paleta de cores fornecida pela empresa. Foram retirados alguns ícones das redes sociais da empresa, visto que eram desnecessárias ficando só o LinkedIn e o site pertencente à empresa. Também foram retirados os números de telefone, o nome da auto estrada na qual trabalham, bem como o local onde a empresa está situada. Acrescentou-se um ponto importante, que foi exatamente o botão “*Logout*”, para que os utilizadores possam sair da sua conta sempre que optarem por isso.

Neste layout, foram criados quatro botões, dois “*TextView*”, bem como três “*ImageView*”, para os ícones do LinkedIn o site e o logotipo da empresa. Primeiramente, inserimos um “*ImageView*” no qual se inseriu o logotipo da empresa definiram-se os dois “*TextView*”, aos quais se deu os mesmos nomes que no layout anterior, logo o design do nome da empresa é igual nos dois layouts. De seguida, definiu-se os três botões, um para o utilizador poder ter acesso à teórica, um segundo para o utilizador aceder ao exame e um último botão para visitantes à empresa.

Este último botão, inicialmente não tinha sido idealizado, pelo que depois de várias reuniões com o tutor, ficou decidido que, a aplicação teria de ter uma área direcionada para pessoas que trabalham fora da empresa. Isto é, como o próprio nome indica, é um botão direcionado principalmente para visitantes à empresa. Esta funcionalidade da aplicação, tem como objetivo, comprovar que essas mesmas pessoas se comprometem a respeitar as regras de trabalho, assinando, no fim da visita, um pequeno panfleto fornecido pela empresa.

Em primeiro lugar, definiu-se a “*ImageView*” e os dois “*TextView*”, nas quais são utilizadas as mesmas funções/posições que foram definidas para o layout do login.

Para a posição do primeiro “*TextView*”, utilizou-se a mesma posição que no layout anteriormente sendo que, neste layout não existe o logotipo da empresa.

```
<ImageView
    android:id="@+id/imageView2"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginTop="12dp"
    android:src="@mipmap/egis2_foreground"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="120dp"
    android:layout_marginTop="52dp"
    android:layout_marginEnd="136dp"
    android:text="EgisRoad"
    android:textColor="@color/colorPrimary"
    android:textSize="32sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/imageView2"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="118dp"
    android:layout_marginEnd="136dp"
    android:text="App Formação"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@+id/imageView2"
    app:layout_constraintTop_toBottomOf="@+id/textView" />
```


De seguida foram implementados quatro botões, aos quais se deu os nomes de “Exame”, “Teórica”, “Visitantes” e por último o “Logout”.

Para o botão do “Exame” foram, tal como para o botão do login, definidas várias funções android ilustrativas, de forma a tornar o design do botão mais apelativo e simples para os utilizadores da aplicação. Neste botão, foi utilizado uma cor nova para o texto, cor essa que está identificada na paleta de cores da empresa (#062C3A).

Tal como o próprio nome indica, este botão está totalmente relacionado com a parte da aplicação dos conhecimentos adquiridos por parte dos utilizadores, que neste caso são os trabalhadores da empresa.

```
<Button
    android:id="@+id/button2"
    android:layout_width="170dp"
    android:layout_height="70dp"
    android:layout_gravity="center"
    android:layout_marginBottom="200dp"
    android:background="@drawable/button"
    android:text="Exame"
    android:textColor="#062C3A"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Depois criamos o botão “Teórica”, no qual, se utilizam as mesmas funções android ilustrativas que no anterior botão. Apenas existe uma diferença e que está relacionada aos ids atribuídos aos botões, isto é, cada botão possui o seu próprio id, o qual é atribuído por nós, para aquando da construção do código ser mais fácil identificar qual o botão que se quer utilizar em determinadas partes do código. Neste botão, é onde está toda a parte teórica que os trabalhadores necessitam de apreender para que possam posteriormente ter um exame positivo e assim poderem trabalhar na via/autoestrada. Foi também utilizada no texto a cor referida anteriormente (#062C3A).

```

<Button
    android:id="@+id/button2"
    android:layout_width="170dp"
    android:layout_height="70dp"
    android:layout_gravity="center"
    android:layout_marginBottom="200dp"
    android:background="@drawable/button"
    android:text="Exame"
    android:textColor="#062C3A"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

De seguida, criou-se o botão “Visitantes”. Todas as funções android ilustrativas aplicadas a este botão são iguais às funções implementadas nos dois botões anteriores, sendo que o id é diferente como já foi referido.

```

<Button
    android:id="@+id/button3"
    android:layout_width="170dp"
    android:layout_height="70dp"
    android:layout_gravity="center"
    android:layout_marginTop="20dp"
    android:background="@drawable/button"
    android:text="Visitantes"
    android:textColor="#062C3A"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.506"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button" />

```

Por último e em relação aos botões, criou-se um simples botão de logout, que como o próprio nome indica tem como objetivo encerrar a sessão do utilizador, caso o mesmo quera realizar essa opção. O design e as funções android ilustrativas, têm referência ao botão do login, ou seja, este botão possui a mesma letra, as margens arredondadas, o fundo do botão é verde, etc... A única diferença está simplesmente no uso da cor vermelha nas letras da palavra logout. Definiu-se essa cor, para que o botão tenha mais contraste em relação aos outros botões, isto é, para que o botão seja rapidamente identificado pelo utilizador.

```

<Button
    android:id="@+id/btnlogout"
    android:layout_width="120dp"
    android:layout_height="wrap_content"
    android:background="@drawable/rounded_corners"
    android:text="Logout"
    android:textColor="@android:color/holo_red_dark"
    app:layout_constraintBottom_toBottomOf="@+id/linearLayout"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button3" />

```

Por fim, foram também criados duas “*ImageView*” dentro de um “*LinearLayout*”. A única e simples função destas duas “*ImageView*” é apenas fazer com que o utilizador dependendo do ícone que carrega, é redirecionado para o site da “EgisRoadPortugal” ou para o site do LinkedIn da empresa. Para definir qual a imagem que iria ser apresentada em cada uma das “*ImageView*”, foi implementada uma função android para o ícone (“*app:srcCompat="@drawable/linkedin_logo"*”) e outra função android para o ícone da empresa (“*app:srcCompat="@mipmap/ic_egis"*”).

```

<ImageView
    android:id="@+id/imageViewSite"
    android:layout_width="0dp"
    android:layout_height="60dp"
    android:layout_weight="1"
    app:srcCompat="@mipmap/ic_egis1" />

```

```

<ImageView
    android:id="@+id/imageViewLinkedIn"
    android:layout_width="0dp"
    android:layout_height="60dp"
    android:layout_weight="1"
    app:srcCompat="@drawable/linkedin_logo" />

```

Como as duas “*ImageView*” foram implementadas dentro de um linear layout, todas as funções referentes às mesmas, tais como, a posição dos ícones relativamente a todo o layout, as margens em relação ao layout, foram implementadas não individualmente em cada “*ImageView*”, mas sim no próprio linear layout.

```

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="15dp"
    android:layout_marginEnd="15dp"
    android:layout_marginBottom="15dp"
    android:gravity="bottom"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent">

    <ImageView
        ... />
    <ImageView
        .../>
</LinearLayout>

```

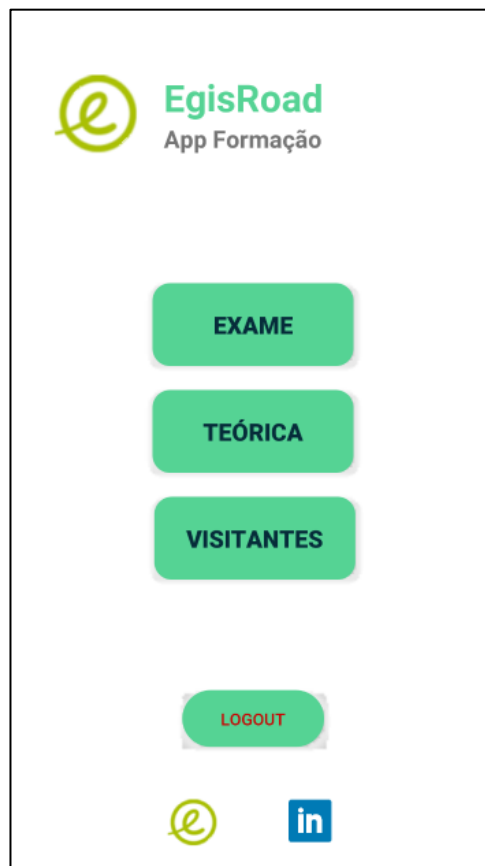


Figura 18 - Layout final do menu

Depois de criado o layout do menu, procedeu-se à construção do código relativo ao funcionamento botões.

Primeiramente definiram-se as classes privadas para o “*ImageView*”, para o “*Button*” e para a “*FirebaseAuth*”.

```
private ImageView botaoface;  
private ImageView botaolink;  
private Button buttonname;  
private FirebaseAuth.AuthStateListener mAuthStateListener;  
private Button mSignOut;
```

Primeiramente começou-se por definir os botões e atribuí-los os “*ID*’s” dos botões previamente criados no design, inicializaram-se esses mesmos botões e atribui-se um “*OnClickListener*” que irá servir para fazer algo quando se clica nele. Nestes dois botões definiu-se para fazer o redireccionamento para páginas diferentes.

```
Button b1 = findViewById(R.id.button);  
  
b1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(v.getContext(),  
RecyclerView3.class);  
        startActivity(intent);  
    }  
});  
  
Button button3 = findViewById(R.id.button3);  
  
button3.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent intvis = new Intent(v.getContext(),  
Visitantes.class);  
        startActivity(intvis);  
    }  
});
```

De seguida criou-se mais um botão que irá servir para fazer o “Logout”. Faz-se a atribuição do botão ao criado no design. Criou-se a função “*setupFirebaseListener*”.

```
mSignOut = (Button) findViewById(R.id.btnlogout);

setupFirebaseListener();

mSignOut.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        FirebaseAuth.getInstance().signOut();
    }
});
```

Na função “*setupFirebaseListener*” verifica se houve uma mudança do estado de autenticação (neste caso o *Logout*) e redireciona o usuário para o menu de Login enquanto que apresenta uma mensagem de confirmação.

De seguida, com a ajuda das “*Flags*”, limpou-se todas as “*Tasks*” que estejam associadas com este menu.

```
private void setupFirebaseListener() {
    mAuthStateListener = new FirebaseAuth.AuthStateListener() {
        @Override
        public void onAuthStateChanged(@NonNull FirebaseAuth
firebaseAuth) {
            FirebaseUser user = firebaseAuth.getCurrentUser();
            if (user != null) {
            } else {
                Toast.makeText(Menu.this, "Signed out",
Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(Menu.this,
MainActivity.class);
                intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
                startActivity(intent);
            }
        }
    };
}
```

Com estas duas funções inicializou-se e removeu-se os “*Listeners*” sempre que houver mudanças do estado de autenticação como um novo “Login” ou um “Logout”, isto é útil para não haver perdas de memória para que a app não “*crashe*”.

```
@Override
public void onStart() {
    super.onStart();

    FirebaseAuth.getInstance().addAuthStateListener(mAuthStateListener);
}

@Override
public void onStop() {
    super.onStop();
    if (mAuthStateListener != null) {

        FirebaseAuth.getInstance().removeAuthStateListener(mAuthStateListener);
    }
}
```

Por último faz-se a criação dos botões para redirecionar para o exame, para o menu das categorias que serviram para o utilizador puder fazer o estudo, para o menu do Visitante e para os links nos dados pela empresa.

```
buttonname = (Button) findViewById(R.id.button2);
buttonname.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(Intent.ACTION_VIEW,
Uri.parse("https://docs.google.com/forms/d/e/1FAIpQLScvLvTfsBjAKfr
o2t9Gaz1iG-XSDBk2JeZDzUswruCcTAXdZA/viewform?usp=sf_link")));
    }
});

botaoface = (ImageView) findViewById(R.id.imageViewSite);
botaoface.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.egisportugal.pt/pt")));
    }
});
```

```

botaolink = (ImageView) findViewById(R.id.imageViewLinkedIn);
botaolink.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(Intent.ACTION_VIEW,
Uri.parse("https://www.linkedin.com/company/egis-road-operation-
portugal")));
    }
})

```

4.4. LAYOUT DO SUBMENU TEÓRICA

Neste subcapítulo, irá ser feita como nos layouts anteriores, uma descrição de todos os passos relativos à construção do design deste submenu da teórica.

Este layout é constituído por uma função android chamada de “*ScrollView*” e por onze botões, os quais são os dez capítulos e um botão para regressar ao menu principal, para que o utilizador da aplicação, neste caso os trabalhadores da empresa, possam escolher qual o capítulo que querem estudar. De realçar, que foi também implementada a função android “*RelativeLayout*”, dentro do próprio “*ScrollView*”, de forma a que este mesmo “*RelativeLayout*” fosse responsável por conter todos os botões.

Outra das razões para usar o “*RelativeLayout*” é porque permite usar a função “*android:layout_below*” que permite posicionar os botões diretamente em baixo do botão que se escolhe.

Em primeiro lugar, começa-se por definir a função android “*ScrollView*”, no qual o objetivo do mesmo é simplesmente permitir ao utilizador procurar pelos capítulos que não apareçam no ecrã, isto é, por exemplo, no ecrã aparecem aproximadamente sete capítulos e por isso esta função foi criada, para que o utilizador possa ver os restantes navegando através de uma barra lateral, a qual foi criada através da função “*ScrollView*”. Nesta função, foi também implementado o código referente à orientação dessa mesma barra, pois tanto pode ser horizontal (da esquerda para a direita ou vice-versa) como pode ser vertical (de cima para baixo), sendo que neste caso a orientação escolhida foi vertical.


```
<ScrollView
    xmlns:android=http://schemas.android.com/apk/res/android
    xmlns:app=http://schemas.android.com/apk/res-auto
    xmlns:tools=http://schemas.android.com/tools
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimary"
    android:orientation="vertical"
    tools:context=".RecyclerView3">
```

De seguida, implementou-se o “*RelativeLayout*”, no qual, contém todos os botões existentes neste layout do submenu da teórica.

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="10dp">
```

Depois de criado o “*RelativeLayout*”, começou-se a criar os botões.

Começou-se pela criação dos botões para cada um dos capítulos da teórica. De realçar que para todos os botões existentes neste layout, foram utilizadas as mesmas funções android, sendo que a única diferença entre eles, está sobretudo nas imagens inseridas em cada um, os ids atribuídos a cada um dos botões bem como o título de cada capítulo. Para este primeiro capítulo, ou seja, segundo botão, foi inserida uma pequena imagem no próprio botão através do seguinte código android, (*android:drawableStart="@mipmap/ic_cap1"*), sendo que também foi implementada a seguinte função, (*android:background="@drawable/bordas"*), para que o botão tivesse um pequeno contraste em relação à cor de fundo e que os cantos fossem um pouco arredondados. Neste e nos restantes botões a cor usada tanto para o texto como para o código anterior referente às “bordas” do botão, é a mesma utilizada no texto dos botões do menu (#062C3A).

```

<Button
    android:id="@+id/b1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/bordas"
    android:drawableStart="@mipmap/ic_cap1"
    android:fontFamily="serif"
    android:gravity="center_vertical"
    android:padding="5dp"
    android:text="1. REALIZAÇÃO DOS \nTRABALHOS NA A24"
    android:textAllCaps="true"
    android:textColor="#062C3A"
    android:textSize="20sp" />

```

Ainda no primeiro capítulo, e quando o utilizador carregar nesse botão, o mesmo é redirecionado para a primeira página de teórica. Esta página, também possui a função “ScrollView”, visto que todos os componentes constituintes deste primeiro capítulo não estão “visíveis” para o utilizador e por isso o mesmo precisa de utilizar a barra lateral para poder visualizar a restante teórica desta mesma página. De realçar, que nesta página, mais propriamente no fundo da página, foram criados dois botões. A função destes dois botões é o de facilitar o trabalho por parte do utilizador. Isto é, um dos botões faz com que o utilizador possa regressar ao índice do submenu da teórica ou o outro botão faz com que o utilizador possa avançar para o segundo capítulo sem que seja necessário voltar ao índice.

```

<Button
    android:id="@+id/btncap2"
    android:layout_width="0dp"
    android:layout_height="50dp"
    android:background="@drawable/rounded_button"
    android:layout_weight="1"
    android:text="Capítulo 2"
    android:textSize="15sp" />

```



Figura 19 - Layout final do submenu da Teórica

Relativamente a todos os outros capítulos, a estrutura dos mesmo foi baseada no design do primeiro. Os únicos aspetos, nos quais os capítulos são diferentes, já tinha sido referido anteriormente que são principalmente as imagens utilizadas para cada capítulo, bem como o título dos mesmos. Cada capítulo, possui o seu respetivo texto, para que os utilizadores possam estudar. Também foram inseridas algumas imagens nesses textos.

Tal como no primeiro capítulo existe aqueles dois botões, os quais o utilizador pode avançar ou regressar ao índice, o mesmo acontece no último capítulo, ou seja, este último possui também dois botões, nos quais o utilizador pode retroceder um capítulo ou regressar ao índice do submenu da teórica. Todos os capítulos, desde o capítulo dois até ao capítulo nove, possuem três botões, em que o utilizador pode retroceder um capítulo, avançar um

capítulo ou até mesmo voltar ao índice, independentemente do capítulo em que o utilizador esteja.

Depois de criado o layout do submenu da teórica, procedeu-se à construção do código relativo ao funcionamento botões.

Primeiro começou-se por implementar um botão na “*Toolbar*” que irá servir como um “*Home Button*”, como mostra a Figura 19, para se fazer o redirecionamento para o menu principal. Definiu-se o seu título e a cor do seu “background”.

```
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setTitle("Índice");
getSupportActionBar().setBackgroundDrawable(new
ColorDrawable(getResources().getColor(R.color.colorPrimary)))
```

Para definir para que pagina para onde se redireciona, no “Manifest.xml”, definiu-se o “parentActivityName” para a página.

```
<activity
    android:name=".RecyclerView3"
    android:parentActivityName=".Menu"
    android:screenOrientation="nosensor" />
```

Por fim implementou-se dez botões e atribui-se o seu id a cada botão criado na parte do design e fez-se com que cada botão redirecionasse para um dos dez capítulos.

```
Button bt1 = findViewById(R.id.b1);
bt1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent in1 = new Intent(v.getContext(), capitulo1.class);
        startActivity(in1);
    }
});

-----

Button bt10 = findViewById(R.id.b10);
bt10.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent in10 = new Intent(v.getContext(),
capitulo10.class);
        startActivity(in10);
    }
});
```

A primeira parte do código habilita o botão na “Toolbar” que irá permitir com que possamos voltar ao menu das categorias.

A segunda parte do código é referente aos botões que nos irão possibilitar retroceder ou avançar nos capítulos.

As únicas diferenças entre os dez capítulos são que:

No 1º capítulo apenas se pode avançar e não retroceder; do capítulo 2 ao 9 pode-se avançar e retroceder; e no capítulo 10 apenas se pode retroceder, mas não avançar.

```
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setTitle("Capítulo 1");
getSupportActionBar().setBackgroundDrawable(new
ColorDrawable(getResources().getColor(R.color.colorPrimary)));

Button btncap = findViewById(R.id.btncap2);
btncap.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent in1 = new Intent(v.getContext(), capitulo2.class);
        startActivity(in1);
    }
});
```

O único capítulo com diferenças de código é o 3º capítulo. Criou-se quatro “Layouts” que irão agir como botões possibilitando a função de clicar neles. Ao clicar-se neles eles irão abrir/fechar assim mostrando/escondendo o texto presente neles.

```
public void expandableButton1(View view) {
    expandableLayout1 = (ExpandableRelativeLayout)
findViewById(R.id.expandableLayout1);
    expandableLayout1.toggle();
}

public void expandableButton2(View view) {
    expandableLayout2 = (ExpandableRelativeLayout)
findViewById(R.id.expandableLayout2);
    expandableLayout2.toggle();
}

public void expandableButton3(View view) {
    expandableLayout3 = (ExpandableRelativeLayout)
findViewById(R.id.expandableLayout3);
    expandableLayout3.toggle();
}
```

```
public void expandableButton4(View view) {
    expandableLayout4 = (ExpandableRelativeLayout)
findViewById(R.id.expandableLayout4);
    expandableLayout4.toggle();
}
```

4.5. LAYOUT E CÓDIGO DO SUBMENU DOS VISITANTES

Este layout, relativamente aos descritos anteriormente, é um pouco mais simples, visto que apenas possui texto acerca das regras de segurança fornecidas pela empresa e um botão, no qual o utilizador ao carregar nesse mesmo botão é redirecionada para uma página do Google Forms, na qual é pedido obrigatoriamente para o utilizador colocar os seus dados, relativos ao seu nome, ao seu email, bem como o nome da empresa à qual pertence.

Primeiramente, criou-se as caixas de texto nas quais foram inseridas todas as informações acerca das regras de segurança fornecidas pela empresa. Tal como nos layouts anteriores, foram utilizadas várias funções android para que, essas mesmas caixas de texto tivessem algum contraste relativamente ao fundo. A cor utilizada para o texto, já tinha sido referenciada acima pelo código #062C3A (cor azul escura).

```
TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:background="@drawable/bordas_cap"
    android:fontFamily="sans-serif-smallcaps"
    android:padding="5dp"
    android:text="Atravessar a autoestrada:"
    android:textColor="#D50000"
    android:textSize="30sp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/bordas_cap"
    android:fontFamily="sans-serif-smallcaps"
    android:padding="5dp"
    android:text="O atrevessamento pedestre da autoestrada é
perigoso pelo que deve de ser evitado.
    \n\nOs visitantes só poderão atravessar pedestremente a
aautoestrada, quando tal for estritamente necessário no âmbito da
visita.
```

```

\n\nEm caso de necessidade de atrevessamento ter em atenção:
\n  • Antes de atravessar olhar nos dois sentidos (Parar,
Olhar e Escutar);
\n  • Esperar para poder atravessar num intervalo suficiente
entre duas viaturas;
\n  • Não parar para apanhar objetos que possam ter caído;"
android:textColor="#000000"
android:textSize="20sp" />

```

De seguida, foi criado o botão, ao qual lhe foi dado o nome de “Tomei conhecimento das regras de segurança”. É um botão extremamente simples, em que o único objetivo do mesmo é redirecionar o utilizador para uma página do Google Forms definida. Nessa mesma página, é pedido ao utilizador que introduza o seu nome, o seu email e o nome da empresa à qual pertence. Esses dados são depois guardados num ficheiro Excel, em que o responsável da formação poderá posteriormente fazer o download do mesmo. Nesse ficheiro Excel, para além dos dados introduzidos pelo utilizador, irá também contar a respetiva data e hora em que os utilizadores, neste caso os visitantes, assinaram ou tomaram conhecimento das regras de segurança.

```

<Button
    android:id="@+id/btn_form"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/rounded_button"
    android:padding="5dp"
    android:text="Tomei conhecimento das regras de segurança"
    android:textSize="15sp" />

```

Figura 20- Google Forms Visitantes



Figura 21 - Layout Visitantes

Na parte do código primeiro começou-se por introduzir um “Home Button” como foi feito na parte das categorias e nos capítulos.

De seguida configurou-se um botão de modo a se fazer o redirecionamento para o “Form” de confirmação das regras lidas.

```
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setTitle("Area dos Visitantes");
getSupportActionBar().setBackgroundDrawable(new
ColorDrawable(getResources().getColor(R.color.colorPrimary)));

buttonform = (Button) findViewById(R.id.btn_form);
buttonform.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startActivity(new Intent(Intent.ACTION_VIEW,
Uri.parse("https://docs.google.com/forms/d/e/1FAIpQLSdC8q1NO-Zy-
a302160AHbU63jmdnxWV6k6OzEZf2B1Yer055g/viewform?usp=sf_link")));
    }
});
```


5. CONCLUSÕES

Nos últimos tempos o crescimento das aplicações para dispositivos móveis tem vindo a crescer exponencialmente e tem-se tornado num dos principais ramos da informática, e poder desenvolver um projeto final nesse contexto foi extremamente importante. Os objetivos propostos no início deste projeto foram cumpridos, sendo que a elaboração da aplicação Android revelou-se um desafio interessante, pois englobou a aprendizagem de uma nova linguagem de programação, a linguagem Java, e o contacto com softwares nunca utilizados como o Android Studio, o FireBase e o Google Forms.

A realização deste projeto foi uma experiência muita positiva que possibilitou adquirir novos conhecimentos de linguagens de programação e permitiu também obter uma metodologia de trabalho e saber lidar com a pressão e calendarização das tarefas necessárias para a conceção do produto final.

O projeto desenvolvido pode ser objeto de melhoramentos ao se integrar mais funcionalidades durante a execução da aplicação Android, como por exemplo, a possibilidade de serem adicionados novos botões para novas funcionalidades se assim for pretendido ou por exemplo adicionar animações na parte de teórica para tornar a aplicação mais interativa.

6. REFERÊNCIAS DOCUMENTAIS

- [1] [Online]. Available: <https://www.egis-group.com> [Accessed: 25-MAI-2020]
- [2] [Online]. Available: <https://www.egisportugal.pt/pt/> [Accessed: 25-MAI-2020]
- [3] [Online]. Available: <https://developer.android.com> [Accessed: 28-MAI-2020]
- [4] [Online]. Available: <https://www.javatpoint.com/history-of-java> [Accessed: 29-MAI-2020]
- [5] [Online]. Available: <https://www.freejavaguide.com/history.html> [Accessed: 29-MAI-2020]
- [6] [Online]. Available: <https://www.android.com> [Accessed: 31-MAI-2020]
- [7] [Online]. Available: <https://rockcontent.com/blog/linguagem-de-programacao/> [Accessed: 31-MAI-2020]
- [8] [Online]. Available: <https://stackoverflow.com> [Accessed: 01-JUL-2020]
- [9] [Online]. Available: <https://firebase.google.com> [Accessed: 24-JUN-2020]
- [10] [Online]. Available: <https://developer.android.com/studio> [Accessed: 30-JUN-2020]
- [11] [Online]. Available: <https://developer.android.com/studio/intro/> [Accessed: 14-JUN-2020]
- [12] [Online]. Available: <https://developer.android.com/samples> [Accessed: 19-JUN-2020]
- [13] [Online]. Available: <https://www.geeksforgeeks.org> [Accessed: 30-JUN-2020]

Anexos

Anexo1 – Código em linguagem Java elaborado no desenvolvimento da Aplicação Android

Anexo2 – APK da Aplicação