



Agrupamento de Escolas de Águas Santas

Curso Profissional de Técnico de Gestão e Programação de Sistemas Informáticos

Ano letivo 2021/2022

Turma H

RELATÓRIO

PROVA DE APTIDÃO PROFISSIONAL

SKYE



Afonso Almeida

Ricardo Mendes

Tiago Castro

março 2022

SKYE

2021/2022

Afonso Almeida
Ricardo Mendes
Tiago Castro

12.º H

Professores orientadores:

António Neves
Diana Almeida
Teotónio Silva

Agradecimentos

A elaboração deste projeto foi muito importante para mostrarmos as nossas capacidades, sobretudo daquilo que somos capazes de realizar em termos informáticos. Antes de mais, gostaríamos de agradecer aos professores orientadores por nos terem conduzido durante este percurso com vista à concretização deste projeto de aptidão profissional (PAP):

Gostaríamos de agradecer a todos os professores que nos ajudaram no desenvolvimento deste projeto, a realização deste relatório e a preparar a apresentação da PAP.

Agradecemos também às funcionárias da escola pela impressão e encadernação do relatório.

Descrição Sumária

Este projeto que nomeamos de “Skye” tem como objetivo mostrar um pouco do que poderá ser um carro no futuro em termos de utilidades e resposta a comandos de voz.

Índice

1. Introdução	1
1.1 Objetivos	1
1.2 Planeamento do projeto	1
2. Descrição Técnica	3
2.1 Recursos	3
2.2 Desenvolvimento	4
Planificação do projeto	4
Python – controlo do carro	5
Desenvolvimento dos scripts	5
Como funcionam os scripts?	6
HTML & CSS – criação do website	10
Index	10
Sobre	10
Manual	12
Galeria	13
Hospedagem do site da Skye na Firebase	13
Java – criação da aplicação para android	15
Desenvolvimento da aplicação utilizando Java	15
Activity	15
Ciclo de vida de uma Activity	15
Design da aplicação	17
Estrutura da Aplicação	18
Ficheiro <i>Manifest</i>	19
Design	19
Logotipo	19
Botões	19
MainActivity	21
Luzes e Piscas	22
Galeria do projeto	25
3. Reflexão crítica	30
Referências	31

Anexos	1
Anexo I	2
Script que controla a voz da Skye	2
Anexo II	9
Script que controla os mecanismos do carro	9
Anexo III	13
Script que controla os piscas do carro	13
Anexo IV	16
Código da página Index.html	16
Anexo V	19
Código da página Sobre.html	19
Anexo VI	23
Código da página manual.html	23
Anexo VII	27
Código da página galeria.html	27
Anexo VII	32
Código do ficheiro CSS das páginas	32
Anexo VIII	41
Código do ficheiro Java da app	41
Anexo IX	50
Código do ficheiro <i>manifest</i> da app	50
Anexo X	52
Código do ficheiro <i>xml</i> da app	52
Anexo XI	57
Glossário	57

Índice de figuras

_17dp8vu	
Figura 1 – Esquema da comunicação entre as diferentes plataformas/dispositivos utilizados	16
Figura 2 – Scripts finais	17
Figura 3 – Verificação constante dos campos da base de dados	17
Figura 4 – Exemplo de condição para alterar o ângulo do servo	18
Figura 5 – Verificação constante do campo “piscas”	18
Figura 6 – Exemplo de condição para ativas os piscas da Skye	18
Figura 7 – Definição de funções e pins utilizados no script ‘skye_voice.py’	19
Figura 8 – Definição de listas com strings objetivas para a Skye	19
Figura 9 – Request do input à firebase e decisão de acordo com as condições existentes	19
Figura 10 – Função que realiza as pesquisas na wikipédia	20
_17dp8vuFigura 11 – Printscreen da página principal do website	21
Figura 12 – Representação da programação no sobre	22
Figura 13 – Representação da skye no sobre	22
Figura 14 – Parte dos criadores no sobre	22
Figura 15 – Exemplo de imagem presente na galeria do website	24
Figura 16 – Adicionar app	25
Figura 17 – Escolha da web app	25
Figura 18 – Instalação das <i>tools</i> necessárias	25
Figura 19 – Lançamento dos ficheiros necessários na web	26
Figura 20 – Ciclo de vida de uma activity	28
Figura 21 – Interface da aplicação android	29
Figura 22 – Pastas principais de uma aplicação	30
Figura 23 – Permissões presentes na aplicação	30
Figura 24 – Logotipo da aplicação Android	31
Figura 25 – Interface da aplicação subdividida em grupos	31
Figura 26 – Código de um dos botões da aplicação	33
Figura 27 – Bibliotecas utilizadas para a criação da app android	33
Figura 28 – Controlo das luzes	34
Figura 29 –	36
Figura 30 – Código dos piscas esquerdo e direito	36
Figura 31 – Código dos quatro piscas	37
Figura 32 – Primeiro teste do olho da Skye(cor azul)	38
Figura 33 – Primeiro teste do olho da Skye(cor verde)	38
_17dp8vuFigura 34 – Demonstração da potência dos máximos da Skye	38
Figura 35 – Ricardo Mendes a descarnar fios para criar o circuito do olho da Skye	39
Figura 36 – Tiago Castro da testar o circuito de leds do chassi da Skye	39
Figura 37 – Programação do script de voz da Skye através de SSH	39
Figura 38 – Montagem do servo da Skye	40
Figura 39 – Moldagem do tubo de cola quente	40
Figura 40 – Tiago Castro a criar a aplicação para androids	40
Figura 41 – Ricardo Mendes a trabalhas no site	41

Figura 42 – Primeiro <i>boot</i> do raspberry	41_1ksv4uv
Figura 43 – Skye montada com as luzes(médios) ligadas	41
Figura 44 – Circuito quase completo da Skye	42

Índice de tabelas

Tabela 1 – Desenvolvimento do projeto em etapas	13
Tabela 2 – Preço e quantidade dos recursos utilizados no projeto	15
Tabela 3 – Transições de uma activity	30
Tabela 4 – Explicação do funcionamento de cada um dos botões da aplicação	34

Siglas e Acrónimos

pip – “Pip installs packages” ou “Pip installs python”

gtts – Google Text To Speech

HTML – Hypertext Markup Language

CSS – Cascade Style Sheet

API – application programming interface

IDE – integrated development ambient (ambiente de desenvolvimento integrado)

GPIO – general-purpose input/output (entrada/saida com um uso geral)

1. Introdução

O projeto consiste na realização de um carro semi inteligente, que responde a questões, é controlado por uma aplicação para android e tem um site para apresentação do projeto e explicação de dúvidas.

Algumas funções encontradas neste projeto são:

- Responder a perguntas predefinidas;
- Fazer perguntas simples na internet;
- Escrever/ler/eliminar notas criadas pelo utilizador;
- Ser controlado remotamente através da sua aplicação;
- Contar piadas...

Está programado com três tipos de linguagem (Python, Java e HTML).

1.1 Objetivos

Neste projeto tivemos como principal objetivo mostrar um pouco do futuro, aprofundar tipos de linguagem como HTML, novos tipos de linguagens tais como java e python e aprender mais sobre eletrónica. Além do nosso objetivo de aprofundar os nossos conhecimentos, quisemos criar um projeto diferente dos que estamos habituados a ver em projetos de PAP. Desta forma, escolhemos criar um carro semi inteligente, controlado através de uma aplicação android.

1.2 Planeamento do projeto

O projeto foi desenvolvido em várias fases como mostra a tabela 1:

	setembro	outubro	novembro	dezembro	janeiro	fevereiro	março
Etapa 1							
Etapa 2							
Etapa 3							
Etapa 4							
Etapa 5							
Etapa 6							

Tabela 1 – Desenvolvimento do projeto em etapas

Na primeira etapa (pesquisa) efetuamos a pesquisa e estudo de funções a usar na PAP, começamos a aprender a trabalhar com Python e terminamos as bases de Python.

Na segunda etapa (início do desenvolvimento do software) juntamos tarefas desenvolvidas a partir da pesquisa na Etapa 1, começamos a solidificar o HTML e a APP. Começamos o site da Skye e a aplicação móvel.

Na terceira etapa (recolha do material para a construção da estrutura da Skye), recolhemos o material necessário para a realização da parte física da Skye. Além do material, verificamos preços (quanto iríamos gastar). Por fim, começamos a comprar o material necessário.

Na quarta etapa (desenvolvimento da estrutura da Skye) começamos o desenvolvimento da estrutura do carro.

Na quinta etapa (início do desenvolvimento do relatório e da apresentação) iniciamos o desenvolvimento do relatório e preparação da apresentação da PAP.

Na sexta etapa (finalização do projeto na totalidade) finalizamos a estrutura da Skye, corrigimos erros encontrados durante o processo e finalizamos o relatório e a apresentação da PAP.

2. Descrição Técnica

2.1 Recursos

Para a concretização do projeto foram utilizados os seguintes materiais/equipamento/software
 Recursos utilizados (hardware):

Item	Componente	Preço Unitário	Quantidade	Subtotal
1	<i>Raspberry Pi 4</i>	<i>40€</i>	<i>1</i>	<i>40€</i>
2	<i>Pack de fios e outros componentes</i>	<i>35€</i>	<i>1</i>	<i>35€</i>
3	<i>LN 298</i>	<i>10€</i>	<i>1</i>	<i>10€</i>
4	<i>LEDs</i>	<i>0.20€</i>	<i>8</i>	<i>1.6€</i>
5	<i>LED RGB</i>	<i>0.60€</i>	<i>5</i>	<i>3€</i>
6	<i>Carro telecomandado</i>	<i>---</i>	<i>1</i>	<i>---</i>
7	<i>PowerBank</i>	<i>18.90€</i>	<i>1</i>	<i>18.90€</i>
8	<i>Tubo de cola quente</i>	<i>0.20€</i>	<i>1</i>	<i>0.20€</i>
			<i>Total</i>	<i>108.70€</i>

Tabela 2 – Preço e quantidade dos recursos utilizados no projeto

Recursos utilizados (software):

- Python – Linguagem de programação usada para assistente virtual
- HTML – Linguagem de programação usada para sites
- Java – Linguagem de programação usada para aplicação móvel
- Firebase - Base de dados para comunicação entre aplicação e carro
- Photoshop – Programa utilizador para criar o a interface da aplicação para android e para a criação do logotipo do projeto.
- Android Studio – Programa utilizado para desenvolver a aplicação para android.
- Canva – Plataforma utilizada para criar a apresentação eletrónica da PAP.
- Word – Programa utilizado para elaborar o relatório.

2.2 Desenvolvimento

Tal como referido anteriormente, o projeto foi desenvolvido com base num raspberry pi 4. Para que conseguíssemos trabalhar com o mesmo, precisávamos de aprender alguns conceitos, pelo facto de não termos conhecimentos suficientes para a realização do que queríamos alcançar.

Para construirmos o nosso projeto necessitámos, inicialmente, de conhecer nomeadamente as portas GPIO do raspberry, estas são essenciais para controlar o hardware da Skye. Precisámos também de nos familiarizar com a linguagem Python. A utilização de python foi essencial.

Planificação do projeto

No início, este projeto tinha sido pensado para ser apenas uma assistente virtual, porém, acabamos por decidir incorporar a assistente num carro controlável pelo telemóvel.

A nossa ideia inicial era controlar o carro completamente por *bluetooth*, mas devido a algumas complicações encontradas (controlo dos pinos GPIO do raspberry através do *bluetooth* e a conexão por *bluetooth* entre o dispositivo android e o raspberry), decidimos utilizar uma ferramenta disponibilizada pelo google chamada *Firebase*.

A Skye está constantemente a verificar os dados da base de dados, estes são alterados pela aplicação android e responde de acordo com os mesmos.

Para utilizar o projeto é necessário apenas uma conexão com a internet e um smartphone android. Visto que este projeto não tem um fim comercial, para conectar a Skye a uma rede *wifi* é preciso entrar no raspberry e fazer a ligação manualmente, através do ficheiro de configuração de rede wifi, ou se acedido pela interface do raspbian e realizar a conexão à internet através do mesmo.



Figura 1 – Esquema da comunicação entre as diferentes plataformas/dispositivos utilizados

A apresentação do desenvolvimento está organizada por linguagens de programação:

Python – controlo do carro

Após a primeira etapa decidimos utilizar os seguintes pacotes de python:

- os – Utilizado para guardar e apagar o ficheiro de voz da Skye;
- datetime – Utilizado para recolher a data atual;
- wikipedia – Utilizado para realizar pesquisas online sobre qualquer tema que o utilizador queira;
- gtts – Pacote da google que permite transformar texto em voz, é este pacote que permite à skye ter o poder da fala;
- random – Permite obter números aleatórios dentro de um intervalo definido;
- pyrebase – Permite ter acesso à base de dados em tempo real da plataforma Firebase, é este pacote que nos permite controlar a Skye através da aplicação móvel;
- pydub – Utilizado para reproduzir a voz da Skye, este pacote permite abrir ficheiros de audio;
- time – Utilizado para “parar” o programa durante um certo intervalo de tempo;
- multiprocessing – Utilizado para criar threads que colocam em execução uma parte do programa, sem o interromper.

Para o projeto funcionar corretamente, criamos três scripts que estão constantemente em execução no raspberry.

Um dos scripts está a controlar o motor e o servo do carro, permitindo que o carro se mova.

O outro script serve para a resposta da Skye. Este (script) permite que a Skye ouça e responda às questões feitas pelo utilizador.

O terceiro script serve para controlar os piscas do carro.

Desenvolvimento dos scripts

1º objetivo – encontrar uma biblioteca de python capaz de reproduzir uma *string* em forma de áudio. A google disponibiliza uma biblioteca chamada *Google Text to Speech* que serve perfeitamente o nosso propósito, pega numa *string* dada e transforma-a para um ficheiro de áudio.

2º objetivo – estruturar todo o código das perguntas e respostas reconhecidas pela Skye. Este foi o objetivo mais fácil, mas ao mesmo tempo o mais complicado, pois precisávamos de pensar em todas as funcionalidades que estariam disponíveis no projeto.

3º objetivo – criar todas as condições para comunicar com a base de dados e realizar as ações pretendidas, de acordo com o valor existente na base de dados. (como por exemplo quando o motor tem o valor ‘1’ o carro anda para a frente.)

Atualmente, o projeto funciona através de três scripts em python, um permite controlar o “sistema de voz”, outro permite controlar o mecanismo do carro e o terceiro permite controlar os piscas.

Como funcionam os scripts?

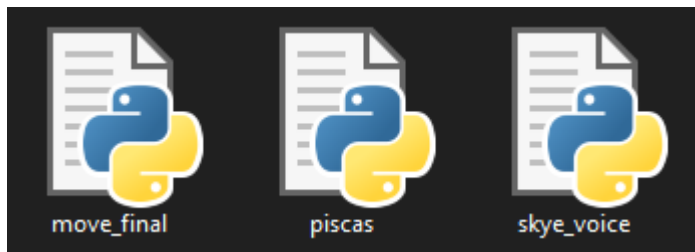


Figura 2 – Scripts finais

-move_final.py:

Neste script foram utilizados 5 pacotes:

- pyrebase
- gpiozero
- RPi.GPIO
- time
- multiprocessing

Quando este script está a correr, estão a ser verificados constantemente três campos da base de dados, o campo ‘servo’, ‘luzes’ e ‘motor’.

```
while True:
    servo1 = db.child("controlo").child("servo").get()
    luzes = db.child("controlo").child("luzes").get()
    motor = db.child("controlo").child("motor").get()
```

Figura 3 – Verificação constante dos campos da base de dados

De acordo com o valor de cada campo, vão ser lidas condições e de acordo com cada uma delas vão ser executadas diferentes tarefas.

```
if servo1.val() == 1:
    servo.angle=90
```

Figura 4 – Exemplo de condição para alterar o ângulo do servo

-piscas.py

Neste script foram utilizados 3 pacotes:

- pyrebase
- RPi.GPIO
- time

Quando este script está a correr, está a ser verificado constantemente um campo da base de dados, o campo ‘piscas’.

```
while True:
    pisca = db.child("controlo").child("piscas").get()
```

Figura 5 – Verificação constante do campo 'piscas'

De acordo com o valor de cada campo, vão ser lidas condições e de acordo com cada uma delas vão ser executadas diferentes tarefas.

```
while pisca.val() == 1:
    pisca = db.child("controlo").child("piscas").get()
    GPIO.output(26, GPIO.HIGH)
    pisca = db.child("controlo").child("piscas").get()
    time.sleep(tempo)
    pisca = db.child("controlo").child("piscas").get()
    GPIO.output(26, GPIO.LOW)
    pisca = db.child("controlo").child("piscas").get()
    time.sleep(tempo)
    pisca = db.child("controlo").child("piscas").get()
```

Figura 6 – Exemplo de condição para ativar os piscas da Skye

No caso do script dos piscas, a condição fica a ser executada até o valor ser alterado.

-skye_voice.py

Neste script foram utilizados 11 pacotes:

- os
- time
- datetime(time)
- wikipedia
- gtts(gTTS)
- random
- pyrebase
- pydub(AudioSegment)
- pydub.playback(play)
- RPi.GPIO
- multiprocessing

No início do script foram definidas algumas funções, a função de alguns pins e algumas variáveis necessárias para o funcionamento do programa.

```
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
verde = 27
azul = 22
GPIO.setup(verde,GPIO.OUT)
GPIO.setup(azul, GPIO.OUT)
intervalo = 0.2

def blueon():
    GPIO.output(azul, GPIO.HIGH)
```

Figura 7 – Definição de funções e pins utilizados no script 'skye_voice.py'

Foram também definidas algumas listas com diferentes *strings*, estas *strings* são utilizadas para otimizar o programa. As *strings* correspondem aos comandos a que a Skye vai responder ou a algumas das suas falas. Ao colocar estas listas todas no início do programa facilita a alteração das mesmas.

```
present_frase = ["O meu nome é Skye, prazer em conhecer-te!",
                "Chamo-me Skye, estou aqui para ajudar!",
                "Os meus criadores chamaram-me Skye, prazer em conhecer-te!"]

pesquisa = ["wikipédia",
            "pesquisa",
            "google"]
```

Figura 8 – Definição de listas com strings objetivas para a Skye

Após a definição, inicia-se um *loop* infinito, dentro deste *loop* estão várias condições. Cada uma das condições procura uma *string* específica no input de áudio disponível na firebase. Dependendo do input, a Skye vai responder de acordo com as informações obtidas online ou pedir mais informações ao utilizador para conseguir finalizar a tarefa requisitada.

```
while True:
    if coiso != 0:
        rand_continuacao = random.randint(0, 2)
        parar_blink = False
        speak(conti_frase[rand_continuacao])
        parar_blink = True

    coiso = 1
    flag = db.child("controlo").child("flag").get()
    abre = ""
    shut = ""
    date = date.today()

    choice1 = get_audio()
    print(choice1 + " || print do input da escolha")

    if pesquisa[0] in choice1 or pesquisa[1] in choice1 or pesquisa[2] in choice1:
        wikipedia_search()
```

Figura 9 – Request do input à firebase e decisão de acordo com as condições existentes.

Este script está estruturado da seguinte forma:

```
def wikipedia_search():
    speak("O que deseja pesquisar?")
    search = get_audio_ops()
    final = ""
    luz = multiprocessing.Process(target=blink, args=())
    luz.start()
    pesquisa = wikipedia.page(search).content
    contpfinal = 1
    ppos = 1
    while contpfinal <= 2:
        contpfinal = contpfinal + 1
        ppos = wikipedia.page(search).content.find(".", ppos + 1)
    i = 0

    luz.terminate()
    all_off()

    speak("Aqui vai o que eu encontrei na internet")

    while i <= ppos:
        print(pesquisa[i], end="")
        final = final + pesquisa[i]
        i = i + 1

    speak(final)
```

Figura 10 – Função que realiza as pesquisas na wikipédia

A função ‘wikipedia_search’ que começa por perguntar ao utilizador o que deseja pesquisar, em seguida, vai buscar o input à firebase (função ‘get_audio_ops’) e define e inicializa uma *thread*, esta *thread* permite que a função ‘blink’ e ‘wikipedia_search’ estejam a ser executadas ao mesmo tempo.

De seguida, o conteúdo da pesquisa é guardado na variável ‘pesquisa’ e é definida uma variável chamada ‘contpfinal’ (contagem ponto final), esta variável vai ser responsável por selecionar apenas as 2 primeiras frases do conteúdo da pesquisa. A variável ‘ppos’ (ponto posição) é responsável por guardar a posição dos pontos finais.

Após os pontos finais serem encontrados, a *thread* criada anteriormente é parada e a Skye avisa que já encontrou resultados e está prestes a dar ao utilizador o resultado da pesquisa.

O próximo ciclo *while* é responsável por selecionar os caracteres até ao segundo ponto final e guardá-los numa variável chamada ‘final’. Por último a *string* final é falada pela Skye.

HTML & CSS – criação do website

Para a realização do website, foram utilizadas as linguagens de programação habituais como HTML e CSS. Para alojar o website recorreremos à plataforma da google Firebase.

A estrutura do website está dividida em diferentes páginas.

Index

No início do website, o utilizador será automaticamente encaminhado para a página web “Index” na qual tentamos captar a atenção do utilizador para saber mais sobre o nosso projeto. Com o design escolhido acreditamos que despertará interesse ao utilizador.

Dentro do Index também existe um botão no qual há a possibilidade de instalar a aplicação para controlar a Skye apresentando uma interface muito intuitiva para que o utilizador tenha a maior das facilidades de a utilizar.

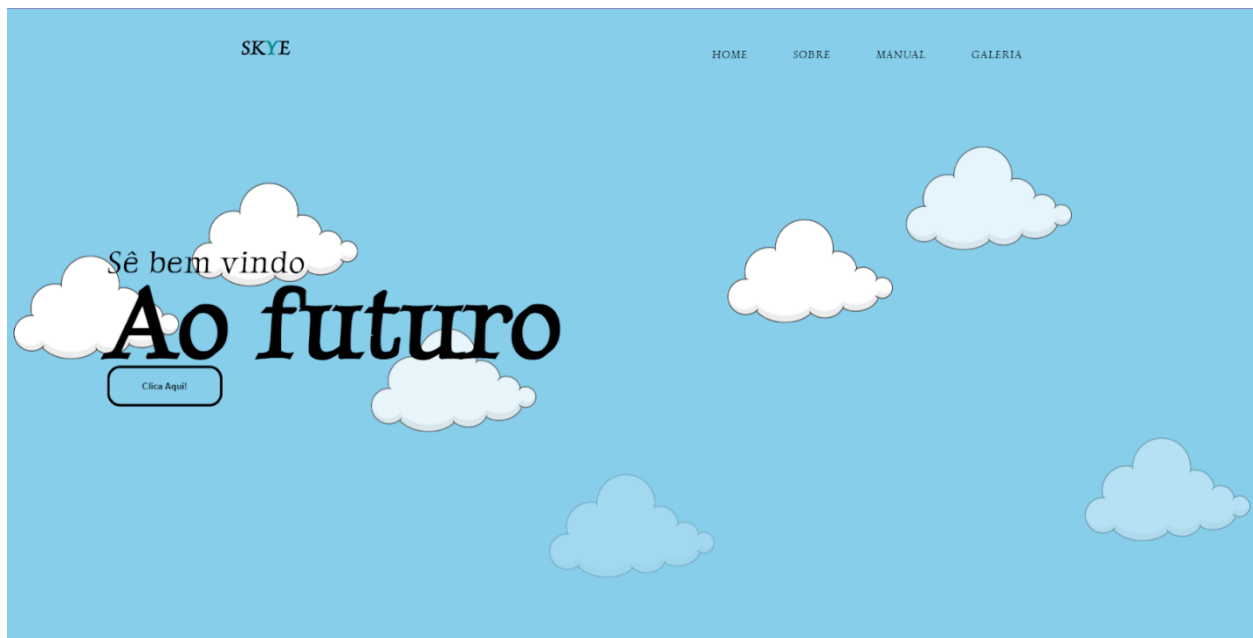


Figura 11 – Printscreen da página principal do website

Sobre

O sobre também apresenta uma estética intuitiva e apelativa para quem pretende visionar algo informativo sem perder o interesse. Para que isso fosse possível, optamos por utilizar um tom mais forte na parte das imagens e, por sua vez, um tipo de letra mais serifado para que o utilizador possa ter maior facilidade a ler a informação.

Esta página contém informação necessária e relativa ao projeto, ou seja:

Linguagens utilizadas

Nas linguagens utilizadas foi referido a utilização de, por exemplo, HTML para ser realizada a construção do website ou até mesmo de python para a assistência virtual...



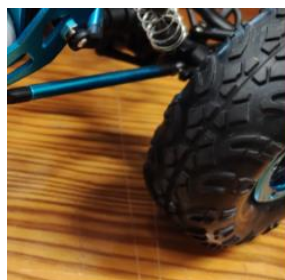
Programação

A primeira situação realizada neste projeto foi a finalização do código. Tanto a skye como a aplicação projetada para a skye foi realizada através de programação, sendo utilizado python para a skye e java para a aplicação android. Este site também foi construído a partir das linguagens de programação HTML, CSS e PHP.

Figura 12 - Representação da programação no sobre

O que é a Skye

Aqui falamos na generalidade, ou seja, o que a Skye faz, como é que foi projetada para ser anteriormente, os passos tomados, entre outros...



Skye

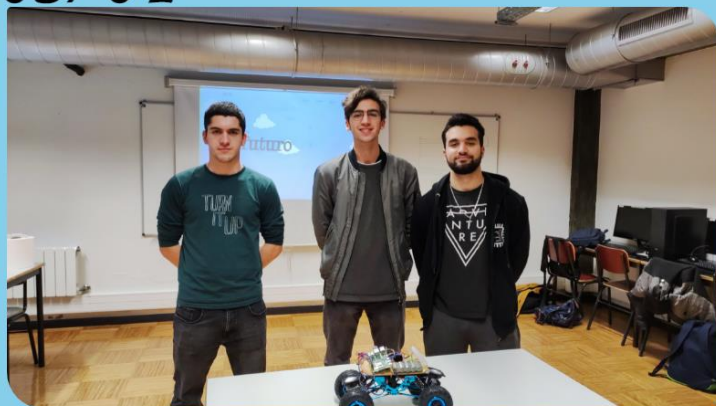
Este é o Físico da skye. O projeto no início foi projetado para ser somente um mero "cerebro", uma simples aplicação na qual seria a sua assistente virtual, no entanto, a inovação de colocar esta assistente dentro de um carro telecomandado foi uma ideia fantástica no qual floresceu este fantástico projeto.

Figura 13 - Representação da skye no sobre

Os seus criadores

Na parte dos criadores, como o nome indica, é mostrado na imagem os criadores da Skye e os respetivos nomes ao lado.

03/04



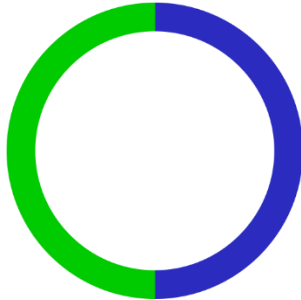
Os meus criadores!

Estes são os meus criadores:
 Afonso Almeida;
 Ricardo Mendes
 Tiago Castro
 E aquela ali em baixo sou eu, a Skye!

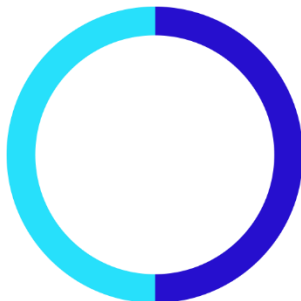
Figura 14 – Parte dos criadores no sobre

Manual

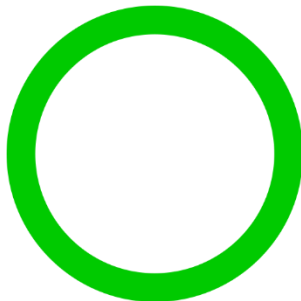
Caso o utilizador tenha algum tipo de dificuldade, o site tem também um manual de instruções com todas as funcionalidades da aplicação e os comportamentos que a Skye apresenta, como:



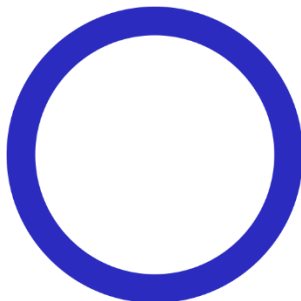
Quando a Skye pisca azul-escuro e verde, significa que a Skye está ocupada a realizar alguma ação, como por exemplo procurar informação na Internet para que seja possível responder ao utilizador.



Quando a Skye pisca azul-escuro e azul-claro, significa que a Skye está a comunicar com o exterior, ou seja, a responder à pergunta que o utilizador realizou.



Quando a Skye apresenta a cor verde, significa que está à espera de uma resposta entre operações, por exemplo, quando é dado a ordem de pesquisar na wikipedia, a mesma pergunta o que desejamos pesquisar e enquanto a skye não obter a resposta, a mesma irá estar à espera, ou seja, verde.



Quando a Skye está somente azul-escuro, significa que a Skye se encontra neutra, ou seja, não foi dada nenhuma ordem a realizar.

O manual também explica os botões utilizados na aplicação referido na parte do relatório da aplicação.

Galeria

Por fim, temos a última página HTML da Skye, a galeria. Nesta página podemos encontrar várias imagens como o comportamento da Skye, o trabalho realizado pelo grupo e algumas outras imagens como o desenvolvimento da aplicação, o código realizado em HTML, Python e Java, entre outras.

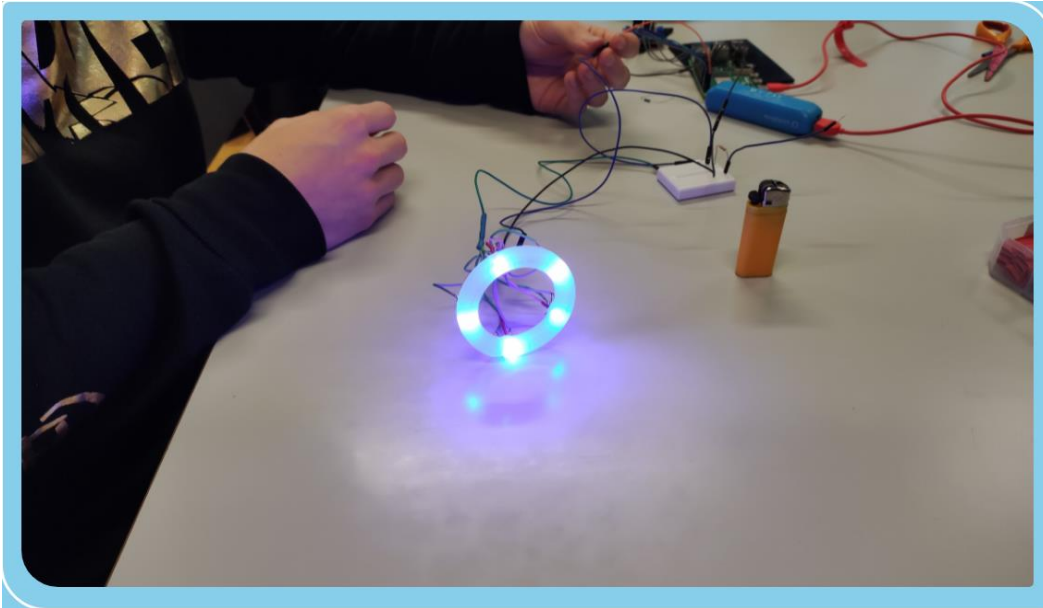


Figura 15 – Exemplo de imagem presente na galeria do website

Hospedagem do site da Skye na Firebase

Para o website estar na web, decidimos utilizar a mesma plataforma que utilizamos para o controlo do carro através da APP. Começamos por adicionar uma “app”, ou neste caso, começamos por alocar um “espaço” na web para que fosse possível qualquer utilizador aceder à nossa página web. Para isso, precisamos de completar alguns passos:

1. Aceder à firebase;
2. Clicar na aba “Adicionar app”:

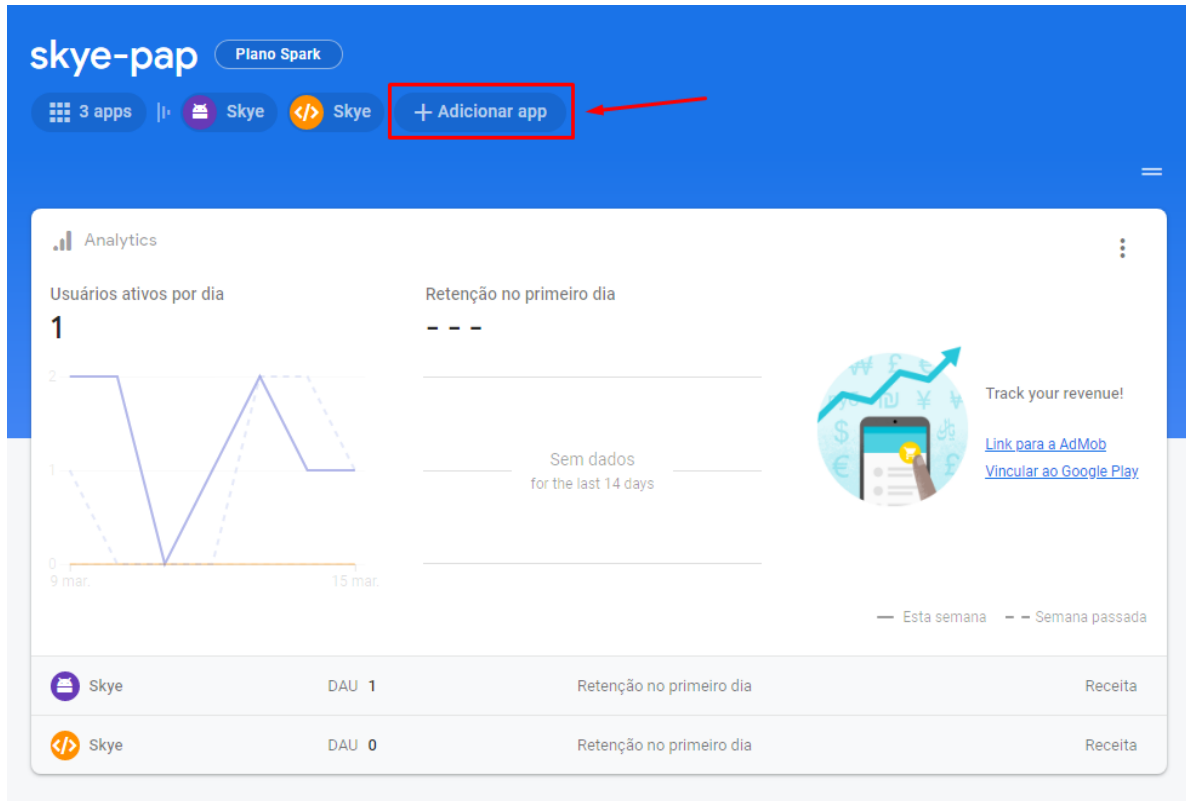


Figura 16 - Adicionar app

3. Escolher uma “web app” → Aqui iremos escolher o tipo de app que iremos adicionar



Figura 17 - Escolha da web app

4. Registrar app → Aqui iremos escolher o nome para o nosso website;
5. Instalar as tools necessárias no computador que irá ser o servidor;

```
C:\Skye\y>npm install -g firebase-tools
```

Figura 18 - Instalação das tools necessárias

6. Após o passo da instalação das *tools* necessárias, é preciso associar uma conta google ao hosting do website. Para isso, basta, no cmd, inserir “firebase login” e realizar o login na conta que irá alocar o website.
7. Por fim, basta dar “deploy” aos ficheiros presentes na pasta que foi denominada para “hostear” o website.

```
C:\Skye\y>firebase deploy

=== Deploying to 'skye-pap'...

i  deploying hosting
i  hosting[skye-pap]: beginning deploy...
i  hosting[skye-pap]: found 41 files in y
+  hosting[skye-pap]: file upload complete
i  hosting[skye-pap]: finalizing version...
+  hosting[skye-pap]: version finalized
i  hosting[skye-pap]: releasing new version...
+  hosting[skye-pap]: release complete

+  Deploy complete!
```

Figura 19 - Lançamento dos ficheiros necessários na web

O website da Skye pode ser visitado através do link ‘<https://skye-pap.web.app/>’.

Java – criação da aplicação para android

Para o desenvolvimento da aplicação utilizamos um software desenvolvido pela google, denominado de *Android Studio*.

Neste programa é possível programar em duas linguagens diferentes, sendo elas Java e Kotlin. Para o desenvolvimento da nossa aplicação, decidimos usar Java, pois já estávamos minimamente familiarizados com a mesma e conseguimos encontrar mais informação na internet sobre a mesma.

Quanto à interface gráfica, a programação da mesma é orientada para objetos e fica guardada num ficheiro *.xml*. As funcionalidades de cada botão e outros objetos são programadas no ficheiro Java.

Desenvolvimento da aplicação utilizando Java

Para o desenvolvimento da aplicação, foram utilizados vários componentes e funcionalidades do *Android Studio*:

Activity

Uma *activity* (ou ecrã), é uma coisa única focada no que o utilizador pode fazer. Quase todas elas possuem um ficheiro XML associado, neste ficheiro está presente a interface gráfica.

Ciclo de vida de uma Activity

O ciclo de vida de uma aplicação é controlado pelo sistema operativo. Assim que uma *activity* comece a ser executada, o sistema operativo manda-a para o topo da pilha (*stack*) tornando-se assim a *activity* principal. A *activity* anterior permanece em baixo na pilha e não é mostrada ao utilizador até voltar a estar em execução.

Uma *activity* pode assumir quatro estados:

- Em execução
- Interrompida
- Em segundo plano
- Destruída

Assim que uma *activity* é interrompida, o sistema operativo pode destruí-la caso seja necessário libertar memória do sistema.

Na figura a seguir está representado o ciclo de vida de uma *activity*.

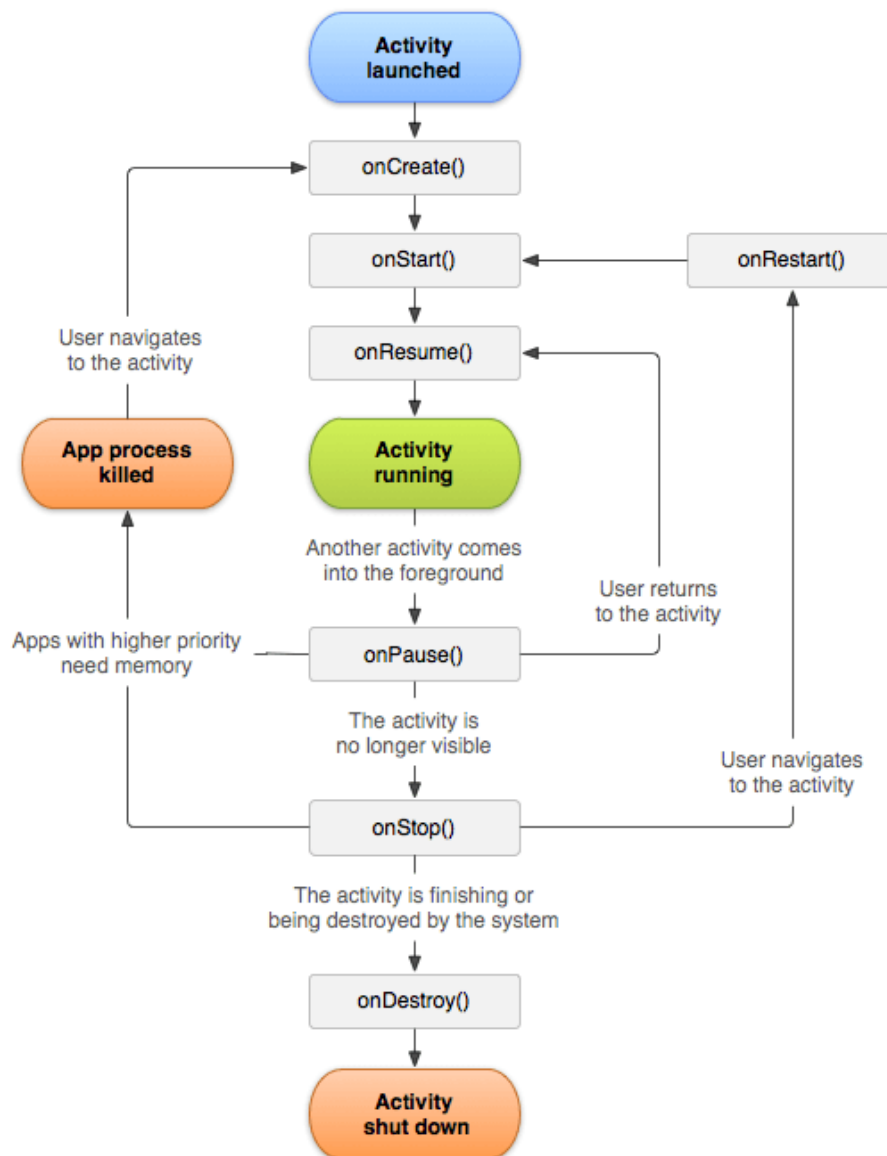


Figura 20 – Ciclo de vida de uma activity

Na tabela 3 são apresentados cada uma das transições entre o ciclo de vida de uma *activity*.

Método	Descrição	Próxima Transição
onCreate()	É chamado só uma vez quando a aplicação é executada	onStart()
onStart()	Executa o código que contém a interface gráfica (UI)	onResume() / onStop()
onRestart()	Restaura o estado da <i>activity</i> desde o momento em que ela foi interrompida	onStart()
onResume()	É chamado assim que a <i>activity</i> começa a interagir com o utilizador	onPause()
onPause()	É chamado quando a <i>activity</i> é interrompida	onResume() / onStop()
onStop()	É chamado quando outra <i>activity</i> está a ser executada.	onRestart() / onDestroy()
onDestroy()	É sempre o último método a ser chamado antes de uma <i>activity</i> ser encerrada.	Nenhum

Tabela 3 – Transições de uma *activity*

Design da aplicação

Após várias ideias e opiniões chegamos a um consenso quanto à interface da aplicação. Decidimos utilizar uma interface simples e intuitiva, para que qualquer pessoa possa utilizar a *Skye* sem qualquer complicação.

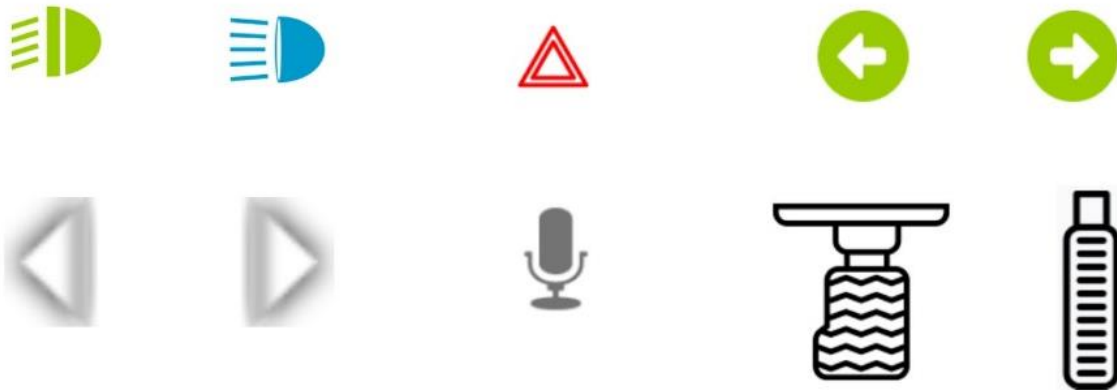


Figura 21 – Interface da aplicação android

O ciclo de vida de uma app é controlado no seu ficheiro Java respetivo, já no ficheiro XML está presente a interface do utilizador. O ficheiro Java irá conter as funcionalidades de cada botão, texto, imagem e etc.

Estrutura da Aplicação

Qualquer aplicação Android contém três pastas principais: manifests, java e res

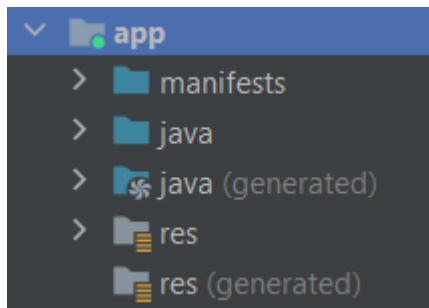


Figura 22 – Pastas principais de uma aplicação

- A pasta “manifests” contém o ficheiro AndroidManifest.xml, dentro deste ficheiro estão presentes todas as definições, configurações e permissões da aplicação.
- A pasta “java” contém todos os ficheiros Java como por exemplo: o ficheiro MainActivity.java.
- A pasta “res” contém todos os ficheiros XML, ou seja, Activities, imagens e animações.

Ficheiro *Manifest*

O AndroidManifest é um ficheiro do tipo XML onde estão escritas todas as informações essenciais que o sistema deve ter sobre a aplicação antes que esta seja executada.

Neste ficheiro são declaradas todas as permissões que a aplicação necessita para o seu funcionamento.

A nossa aplicação só necessita de permissão para usar a Internet como se pode ver na imagem abaixo.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Figura 23 – Permissões presentes na aplicação

Design

Durante o desenvolvimento da aplicação surgiram várias ideias em relação ao design, seja em termos de cores, posição dos botões ou forma dos botões. Isto tudo para que a interface gráfica fosse o mais intuitiva possível aos olhos do utilizador.

Logotipo

Numa aplicação, o logotipo é das coisas mais importantes pois é de ser a primeira coisa vista pelo utilizador antes de iniciar a aplicação.

A seguinte figura mostra o logotipo usado na aplicação.



Figura 24 – Logotipo da aplicação Android

Botões

A aplicação é composta por dez botões, todos eles com funcionalidades diferentes. É possível agrupar estes botões em três grupos distintos:

1. **Luzes;**
2. **Controlo do carro;**
3. **Comunicação.**

A seguinte figura mostra como podemos agrupar os botões dentro dos três grupos referidos em cima.

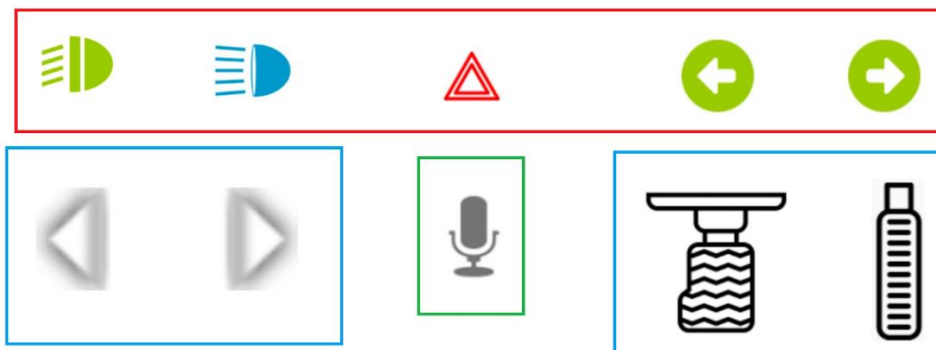




Figura 25 – Interface da aplicação subdividida em grupos

A seguinte tabela contém as funcionalidades de cada botão presente na aplicação.

Funcionalidades de cada botão	
	Microfone: Permite realizar comandos de voz para se comunicar com a Skye.
	Quatro Piscas: Liga/Desliga os Quatro Piscas/Luzes de emergência.
	Pisca Esquerdo: Liga/Desliga o pisca do lado esquerdo do carro.
	Pisca Direito: Liga/Desliga o pisca do lado direito do carro.
	Médios: Liga/Desliga as luzes de cruzamento (Médios).
	Máximos: Liga/Desliga as luzes de estrada (Máximos).
	Virar: Muda a direção do carro para a esquerda ou para a direita.



	<p>Acelerador: Faz com que o carro acelere e ande para a frente.</p>
	<p>Travão: Faz com que o carro reduza a velocidade ou ande em marcha atrás.</p>

Tabela 4 – Explicação do funcionamento de cada um dos botões da aplicação

A figura abaixo, mostra o código de um dos botões. O código é bastante semelhante em todos os botões sendo a principal diferença o ícone usado em cada um deles.

```
<ImageButton
    android:id="@+id/imageButton4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:backgroundTint="@color/white"
    android:rotationY="180"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.793"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.293"
    app:srcCompat="@drawable/dir"
    app:tint="@android:color/holo_green_light" />
```

Figura 26 – Código de um dos botões da aplicação

MainActivity

O MainActivity é o ficheiro java que contém todas as funcionalidades da aplicação. Tal como em várias outras linguagens de programação, temos de declarar as bibliotecas necessárias para o que queremos que o programa faça. Na seguinte figura, estão presentes todas as bibliotecas usadas na aplicação.

```
import static com.google.firebase.database.FirebaseDatabase.getInstance;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.view.MotionEvent;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.FirebaseApp;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.HashMap;
```

Figura 27 – Bibliotecas utilizadas para a criação da APP android

Luzes e Piscas

Como foi referido anteriormente, ao pressionar um botão qualquer ele envia um valor para a base de dados que depois é interpretado pelo *script* em Python. O mesmo acontece com os botões das luzes (médios e máximos) e dos piscas.

Não foi necessário usar a função *onTouchListener* pois não pretendemos que o valor volte a zero quando o botão deixa de ser pressionado. No entanto, é necessário que haja uma maneira de desligar as luzes e os piscas, para isto, foi usada uma estrutura *if* onde existe uma variável chamada “pos” cujo valor predefinido é zero. Ao pressionar o botão este valor vai ser alterado para “1” e ao pressionar o botão novamente o valor irá voltar a zero.

A figura a seguir demonstra esta estrutura *if*.

```
med.setOnClickListener(new View.OnClickListener() {
    int pos = 0;
    public void onClick(View v) {
        FirebaseDatabase database = FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-west1.firebaseio.com");
        DatabaseReference mDatabaseRef = database.getReference();
        if (pos == 0){
            mDatabaseRef.child("controlo").child("luzes").setValue(1);
            pos = 1;
        } else if (pos == 1){
            mDatabaseRef.child("controlo").child("luzes").setValue(0);
            pos = 0;
        }
    }
});
```

Figura 28 – Controlo das luzes

As luzes e os piscas foram programados da mesma maneira, sendo as únicas diferenças o valor enviado para a base de dados e a tabela para onde vão os mesmos.

As três seguintes figuras mostram o código para os botões das luzes e dos piscas.

```

med.setOnClickListener(new View.OnClickListener() {
    int pos = 0;
    public void onClick(View v) {
        FirebaseDatabase database = FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-west1.firebaseio.com");
        DatabaseReference mDatabaseRef = database.getReference();
        if (pos == 0){
            mDatabaseRef.child("controle").child("luzes").setValue(1);
            pos = 1;
        } else if (pos == 1){
            mDatabaseRef.child("controle").child("luzes").setValue(0);
            pos = 0;
        }
    }
});

max.setOnClickListener(new View.OnClickListener() {
    int pos = 0;
    public void onClick(View v) {
        FirebaseDatabase database = FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-west1.firebaseio.com");
        DatabaseReference mDatabaseRef = database.getReference();
        if (pos == 0){
            mDatabaseRef.child("controle").child("luzes").setValue(2);
            pos = 1;
        } else if (pos == 1){
            mDatabaseRef.child("controle").child("luzes").setValue(0);
            pos = 0;
        }
    }
});

```

Figura 29 – Código das luzes

```

pesq.setOnClickListener(new View.OnClickListener() {
    int pos = 0;
    public void onClick(View v) {
        FirebaseDatabase database = FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-west1.firebaseio.com");
        DatabaseReference mDatabaseRef = database.getReference();
        if (pos == 0){
            mDatabaseRef.child("controle").child("piscas").setValue(1);
            pos = 1;
        } else if (pos == 1){
            mDatabaseRef.child("controle").child("piscas").setValue(0);
            pos = 0;
        }
    }
});

pdir.setOnClickListener(new View.OnClickListener() {
    int pos = 0;
    public void onClick(View v) {
        FirebaseDatabase database = FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-west1.firebaseio.com");
        DatabaseReference mDatabaseRef = database.getReference();
        if (pos == 0){
            mDatabaseRef.child("controle").child("piscas").setValue(2);
            pos = 1;
        } else if (pos == 1){
            mDatabaseRef.child("controle").child("piscas").setValue(0);
            pos = 0;
        }
    }
});

```

Figura 30 – Código do pisca esquerdo e direito

```

haz.setOnClickListener(new View.OnClickListener() {
    int pos = 0;
    public void onClick(View v) {
        FirebaseDatabase database = FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-west1.firebaseio.com");
        DatabaseReference mDatabaseRef = database.getReference();
        if (pos == 0){
            mDatabaseRef.child("controlo").child("piscas").setValue(3);
            pos = 1;
        } else if (pos == 1){
            mDatabaseRef.child("controlo").child("piscas").setValue(0);
            pos = 0;
        }
    }
});
    
```

Figura 31 – Código dos quatro piscas

Galeria do projeto

Nesta parte colocamos algumas imagens captadas durante a criação do nosso projeto como prova do tempo e trabalho colocado no mesmo.

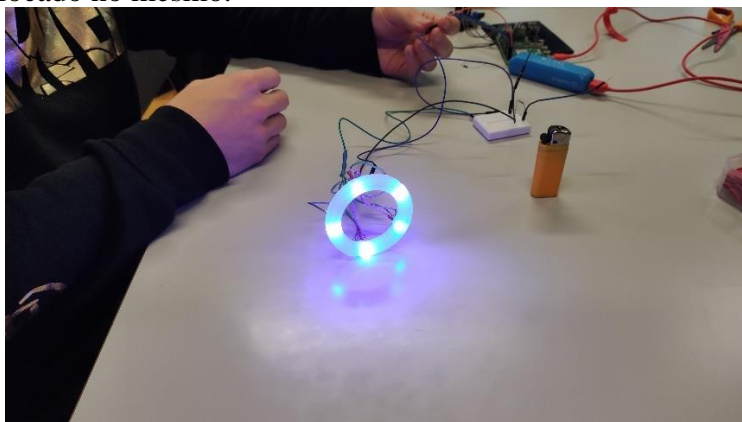


Figura 32 – primeiro teste do olho da Skye(cor azul)

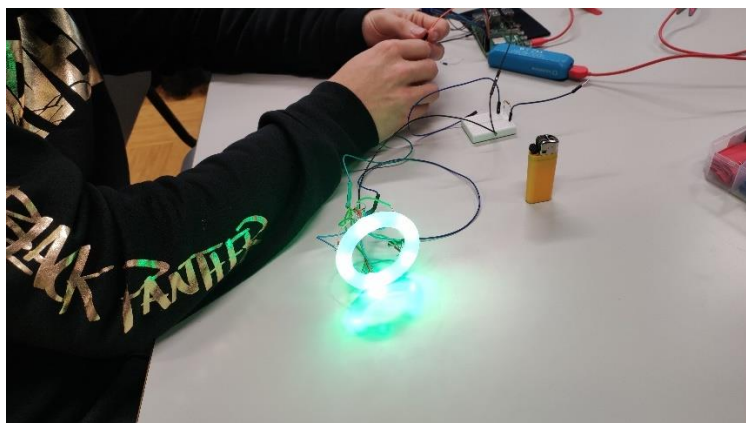


Figura 33 – primeiro teste do olho da Skye(cor verde)



Figura 34 – demonstração da potência dos máximos da Skye



Figura 35 – Ricardo Mendes a descarnar fios para criar o circuito do olho da Skye



Figura 36 – Tiago Castro a testar o circuito de leds do chassi da Skye

```

pi@skye:~/Desktop
GNU nano 3.2
teste_audio.py

import os
import time
from pydub import AudioSegment
from pydub.playback import play
from gtts import gTTS

bye_frase = ["Ok, vou voltar a dormir.", "Se precisar de mais algo, diga 'hey skye'"]

def speak (fala):
    tts = gTTS(text=fala, lang="pt-PT", slow=False)
    file = "voice.mp3"
    tts.save(file)
    voice_talk = AudioSegment.from_mp3(file)
    voice_talk_loud = voice_talk + 7
    play(voice_talk_loud)
    os.remove("voice.mp3")

a = input("fala")
speak(a)
    
```

Figura 37 – Programação do script de voz da Skye através de SSH

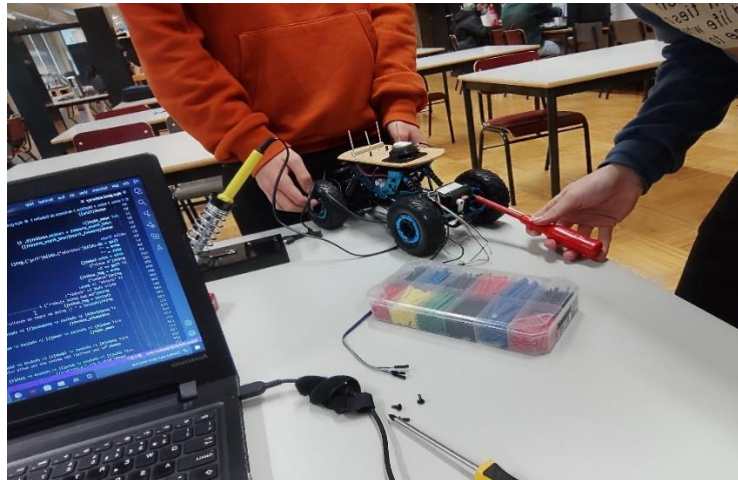


Figura 38 – Montagem do servo da Skye



Figura 39 – Moldagem do tubo de cola quente

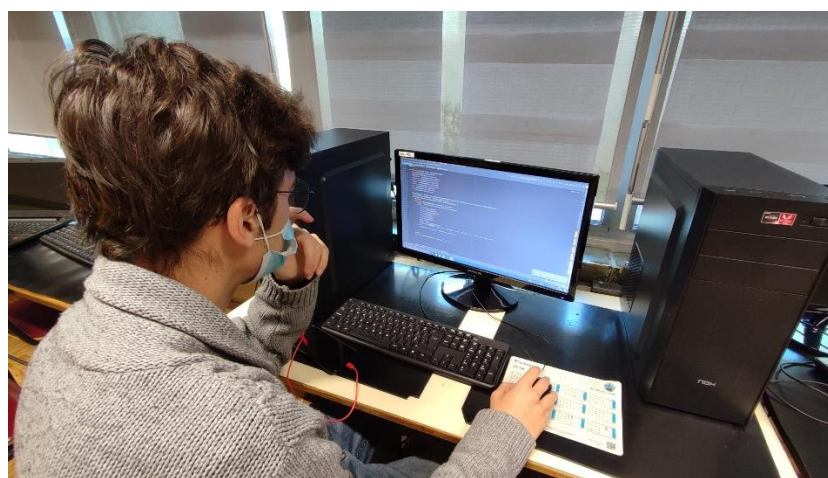


Figura 40 – Tiago Castro a criar a aplicação para android



Figura 41 – Ricardo Mendes a trabalhar no site



Figura 42 – Primeiro boot do raspberry

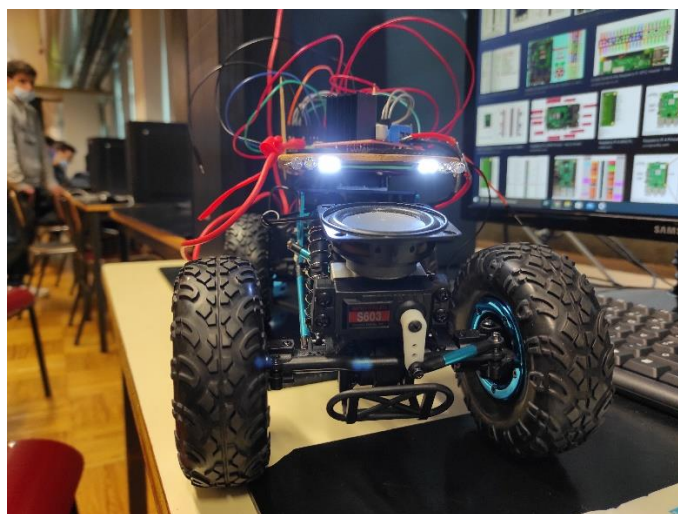


Figura 43 – Skye montada com as luzes (médios) ligadas

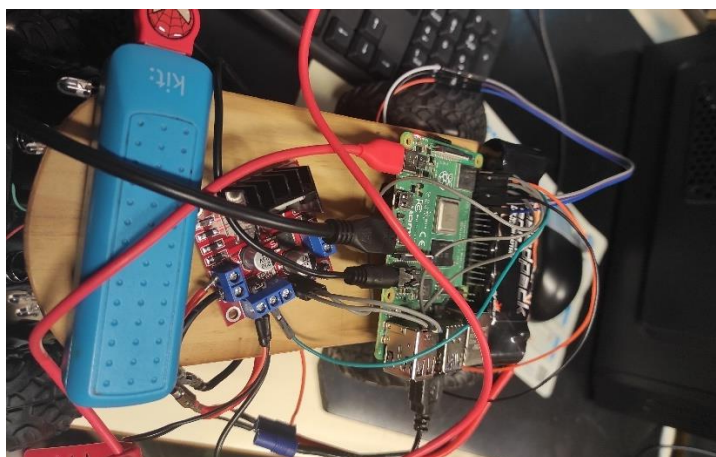


Figura 44 – circuito quase completo da Skye

3. Reflexão crítica

Este projeto foi, sem dúvida, o mais divertido que fizemos no nosso percurso escolar. Foi algo que nos possibilitou a criação de uma peça que verdadeiramente gostássemos, sem qualquer tipo de restrição.

A Skye deu-nos imenso trabalho, especialmente de pesquisa, pois engloba muitas matérias e áreas das quais não temos muitos conhecimentos, assim sendo, tivemos imenso trabalho para descobrir como concretizar o nosso objetivo.

A nossa maior dificuldade foi criar a aplicação para android, pois era uma área totalmente desconhecida.

Em termos futuros, uma melhoria a introduzir no nosso projeto será alterar o controlo do carro para Bluetooth, ao invés de WiFi.

Referências

AndroidStudio(1 de 10 de 2021). developer.android.com. Obtido de AndroidStudio:

<https://developer.android.com/studio>

<https://developer.android.com/reference/android/view/View.OnTouchListener>

<https://developer.android.com/reference/android/widget/Button>

Python(1 de 10 de 2021). python.org. Obtido de python.org: <https://www.python.org/>

Firebase (28 de 01 de 2021). firebase.google.com. Obtido de Firebase:

<https://firebase.google.com/>

<https://firebase.google.com/docs/database/android/read-and-write>

freecodecamp.org(28 de 01 de 2022). Obtido de youtube.com:

https://www.youtube.com/watch?v=fis26HvVDII&t=1236s&ab_channel=freeCodeCamp.org

https://www.youtube.com/watch?v=RcSHAkpwXAQ&t=0s&ab_channel=freeCodeCamp.org

W3Schools(13 de 02 de 2022). Obtido de w3schools.com:

https://www.w3schools.com/css/css3_flexbox_responsive.asp

https://www.w3schools.com/css/css_navbar.asp

Bootstrap(13 de 02 de 2022). Obtido de bootstrap.com: <https://getbootstrap.com/>

Code First With Hala(16 de 02 de 2022). Obtido de youtube.com:

<https://www.youtube.com/watch?v=8IWalfRUK1M&t=7s>

<https://www.youtube.com/watch?v=1DhvKCjG2NE>

<https://www.youtube.com/watch?v=H9cR1eoAtYM>

https://www.youtube.com/watch?v=sVwWEoDa_uY

JavaTPoit (26 de 02 de 2022). Javatpoint.com. Obtido de JavaTPoint:

<https://www.javatpoint.com/images/androidimages/Android-Activity-Lifecycle.png>

Anexos

Anexo I

Script que controla a voz da Skye

```
import os
import time
from datetime import date
import wikipedia
from gtts import gTTS
import random
import pyrebase
from pydub import AudioSegment
from pydub.playback import play
import RPi.GPIO as GPIO
import multiprocessing

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
verde = 27
azul = 22
GPIO.setup(verde,GPIO.OUT)
GPIO.setup(azul, GPIO.OUT)
intervalo = 0.2

def blueon():
    GPIO.output(azul, GPIO.HIGH)
def blueoff():
    GPIO.output(azul, GPIO.LOW)

def verdeon():
    GPIO.output(verde, GPIO.HIGH)
def verdeoff():
    GPIO.output(verde, GPIO.LOW)

def blink():
    while True:
        GPIO.output(azul,GPIO.HIGH)
        GPIO.output(verde, GPIO.LOW)
        time.sleep(intervalo)
        GPIO.output(azul, GPIO.LOW)
        GPIO.output(verde,GPIO.HIGH)
        time.sleep(intervalo)

def all_off():
    GPIO.output(verde, GPIO.LOW)
    GPIO.output(azul, GPIO.LOW)
```



```
def all_on():
    GPIO.output(verde, GPIO.HIGH)
    GPIO.output(azul, GPIO.HIGH)

def blink_blue():
    while True:
        GPIO.output(azul,GPIO.HIGH)
        GPIO.output(verde, GPIO.LOW)
        time.sleep(intervalo)
        GPIO.output(verde,GPIO.HIGH)
        time.sleep(intervalo)

firebaseConfig = {
informações secretas da config do firebase
}
firebase = pyrebase.initialize_app(firebaseConfig)
db = firebase.database()

bye_frase = ["Ok, vou voltar a dormir.",
            "Se precisar de mais algo, diga 'hey skye'!",
            "Parece que vou hibernar um bocado."]

present_frase = ["O meu nome é Skye, prazer em conhecer-te!",
                "Chamo-me Skye, estou aqui para ajudar!",
                "Os meus criadores chamaram-me Skye, prazer em conhecer-te!"]

pesquisa = ["wikipédia",
            "pesquisa",
            "google"]

nome = ["te chamas",
        "teu nome",
        "quem és"]

idade = ["anos tens",
        "tua idade",
        "és velha?",
        "és nova?"]

data = ["data de hoje",
        "dia de hoje",
        "dia é hoje",
```

```
"que dia é"]

conti_frase = ["Se precisar de mais alguma coisa estou aqui!",
               "Mais alguma coisa é só dizer!",
               "Estou disponivel se precisar"]

wikipedia.set_lang("pt")

def speak(fala):
    luz = multiprocessing.Process(target=blink_blue)
    luz.start()
    tts = gTTS(text=fala, lang="pt-PT", slow=False)
    file = "voice.mp3"
    tts.save(file)
    voice_talk = AudioSegment.from_mp3(file)
    voice_talk_loud = voice_talk + 7
    play(voice_talk_loud)
    os.remove("voice.mp3")
    luz.terminate()
    all_off()

def get_audio():
    all_off()
    blueon()
    flag = db.child("controlo").child("flag").get()
    while flag.val() == "True":
        flag = db.child("controlo").child("flag").get()

    voz = db.child("controlo").child("voz").get()
    retornar = voz.val()
    print(retornar)
    db.child("controlo").update({"flag": "True"})
    return retornar

def get_audio_ops():
    all_off()
    verdeon()
    flag = db.child("controlo").child("flag").get()
    while flag.val() == "True":
        flag = db.child("controlo").child("flag").get()
```

```
voz = db.child("controlo").child("voz").get()
retornar = voz.val()
print(retornar)
db.child("controlo").update({"flag":"True"})
return retornar

def wikipedia_search():
    speak("O que deseja pesquisar?")
    search = get_audio_ops()
    final = ""
    luz = multiprocessing.Process(target=blink, args=())
    luz.start()
    pesquisa = wikipedia.page(search).content
    contpfinal = 1
    ppos = 1
    while contpfinal <= 2:
        contpfinal = contpfinal + 1
        ppos = wikipedia.page(search).content.find(".", ppos + 1)
    i = 0

    luz.terminate()
    all_off()

    speak("Aqui vai o que eu encontrei na internet")

    luz = multiprocessing.Process(target=blink, args=())
    luz.start()

    while i <= ppos:
        print(pesquisa[i], end="")
        final = final + pesquisa[i]
        i = i + 1

    luz.terminate()

    speak(final)

def nome_skye():
    rand_frase_present = random.randint(0, 2)
    speak(present_frase[rand_frase_present])
```

```

speak("Em que posso ajudar?")

coiso = 0
while True:
    if coiso != 0:
        rand_continuacao = random.randint(0, 2)
        parar_blink = False
        speak(conti_frase[rand_continuacao])
        parar_blink = True

    coiso = 1
    flag = db.child("controlo").child("flag").get()
    abre = ""
    shut = ""
    date = date.today()

    choice1 = get_audio()
    print(choice1 + " || print do input da escolha")

    if pesquisa[0] in choice1 or pesquisa[1] in choice1 or pesquisa[2] in
choice1:
        wikipedia_search()

    elif nome[0] in choice1 or nome[1] in choice1 or nome[2] in choice1:
        nome_skye()

    elif idade[0] in choice1 or idade[1] in choice1 or idade[2] in choice1 or
idade[3] in choice1:
        speak("Eu sou imortal! Não penses que sou muito velha!")

    elif data[0] in choice1 or data[1] in choice1 or data[2] in choice1 or
data[3] in choice1:
        speak("Hoje é dia " + str(date))

    elif "iniciar protocolo de emergência" in choice1:
        speak("protocolo iniciado")
        exit()

    elif "criar uma nota" in choice1 or "cria uma nota" in choice1:
        speak("Que nome deseja dar à nota?")
        nome_nota = get_audio_ops()

```

```

speak("O que deseja colocar na nota " + nome_nota + "?")
data_nota = get_audio_ops()
db.child("notas").set({nome_nota:data_nota})
speak("Nota criada com sucesso com o nome " + nome_nota)

elif "ler uma nota" in choice1 or "lê uma nota" in choice1:
speak("Qual é o nome da nota que deseja ler?")
nome_ler_nota = get_audio_ops()
try:
    content_nota = db.child("notas").child(nome_ler_nota).get()
    speak("O conteúdo da nota é: " + content_nota.val())
except Exception as k:
    speak("Não existe nenhuma nota com esse nome...")

elif "remover uma nota" in choice1 or "apagar uma nota" in choice1:
speak("Qual é o nome da nota que deseja apagar?")
nome_delete = get_audio_ops()
try:
    db.child("notas").remove(nome_delete)
    speak("Nota apagada com sucesso")
except Exception as k:
    speak("Não existe nenhuma nota com esse nome...")

elif "estou triste" in choice1 or "conta-me uma piada" in choice1 or "ouvir
uma piada" in choice1:
    rand_nr = random.randint(0, 8)
    piada = db.child("piadas").child(rand_nr).get()

elif "cancelar" in choice1:
    break
else:
    if choice1 == "":
        rand_frase = random.randint(0, 2)
        print(bye_frase[rand_frase])
        shut = "dormir"
    else:
        speak("Lamento, mas não entendi o que disse")
        coisa = 0

```

Anexo II

Script que controla os mecanismos do carro

```
import pyrebase
from gpiozero import AngularServo
import RPi.GPIO as GPIO
import time
import multiprocessing

firebaseConfig = {
    informações secretas da config do firebase
}

servo = AngularServo(17, min_pulse_width=0.0006, max_pulse_width=0.0023)

firebase = pyrebase.initialize_app(firebaseConfig)

GPIO.setmode(GPIO.BCM)
Ena, In1, In2 = 2,3,4

GPIO.setup(Ena,GPIO.OUT)
GPIO.setup(In1,GPIO.OUT)
GPIO.setup(In2,GPIO.OUT)

GPIO.setup(10,GPIO.OUT) #maximos
GPIO.setup(9,GPIO.OUT) # stop
GPIO.setup(11,GPIO.OUT) #medios
GPIO.setup(5,GPIO.OUT) #medios

pwm = GPIO.PWM(Ena,100)
pwm.start(0)

db = firebase.database()

def all_off_light():
    GPIO.output(10, GPIO.LOW)
    GPIO.output(9, GPIO.LOW)
    GPIO.output(11, GPIO.LOW)
    GPIO.output(5, GPIO.LOW)

started = 0
off = False
```

```
while True:
    servo1 = db.child("controlo").child("servo").get()
    luzes = db.child("controlo").child("luzes").get()
    motor = db.child("controlo").child("motor").get()

    print(servo1.val())
    print(motor.val())
    print(luzes.val())

    if luzes.val() == 1: #medios
        all_off_light()
        GPIO.output(5,GPIO.HIGH)
        GPIO.output(11,GPIO.HIGH)
    elif luzes.val() == 2: #maximos
        all_off_light()
        GPIO.output(5,GPIO.HIGH)
        GPIO.output(11,GPIO.HIGH)
        GPIO.output(10,GPIO.HIGH)
    elif luzes.val() == 0:
        all_off_light()

    if motor.val() == 0:
        pwm.ChangeDutyCycle(0)

    if motor.val() == 2:
        GPIO.output(In1,GPIO.LOW)
        GPIO.output(In2,GPIO.HIGH)
        pwm.ChangeDutyCycle(100)

    if motor.val() == 1:
        GPIO.output(9,GPIO.HIGH)
        GPIO.output(In1,GPIO.HIGH)
        GPIO.output(In2,GPIO.LOW)
        pwm.ChangeDutyCycle(100)
    else:
        GPIO.output(9,GPIO.LOW)

    if servo1.val() == 1:
```



```
servo.angle=90

if servo1.val() == 2:
    servo.angle=-90

if servo1.val() == 0:
    servo.angle=0
```

Anexo III

Script que controla os piscas do carro

```
import pyrebase
import RPi.GPIO as GPIO
import time

firebaseConfig = {
informações secretas da config do firebase
}

firebase = pyrebase.initialize_app(firebaseConfig)

GPIO.setmode(GPIO.BCM)

GPIO.setup(19,GPIO.OUT) #pisca dir
GPIO.setup(26,GPIO.OUT) #pisca esq
tempo = 1

db = firebase.database()

while True:
    pisca = db.child("controlo").child("piscas").get()

    while pisca.val() == 1:
        pisca = db.child("controlo").child("piscas").get()
        GPIO.output(26, GPIO.HIGH)
        pisca = db.child("controlo").child("piscas").get()
        time.sleep(tempo)
        pisca = db.child("controlo").child("piscas").get()
        GPIO.output(26, GPIO.LOW)
        pisca = db.child("controlo").child("piscas").get()
        time.sleep(tempo)
        pisca = db.child("controlo").child("piscas").get()

    while pisca.val() == 2:
        pisca = db.child("controlo").child("piscas").get()
        GPIO.output(19, GPIO.HIGH)
        pisca = db.child("controlo").child("piscas").get()
        time.sleep(tempo)
        pisca = db.child("controlo").child("piscas").get()
        GPIO.output(19, GPIO.LOW)
        pisca = db.child("controlo").child("piscas").get()
        time.sleep(tempo)
        pisca = db.child("controlo").child("piscas").get()
```

```
while pisca.val() == 3:
    pisca = db.child("controlo").child("piscas").get()
    GPIO.output(19, GPIO.HIGH)
    GPIO.output(26, GPIO.HIGH)
    pisca = db.child("controlo").child("piscas").get()
    time.sleep(tempo)
    pisca = db.child("controlo").child("piscas").get()
    GPIO.output(19, GPIO.LOW)
    GPIO.output(26, GPIO.LOW)
    pisca = db.child("controlo").child("piscas").get()
    time.sleep(tempo)
    pisca = db.child("controlo").child("piscas").get()

while pisca.val() == 0:
    GPIO.output(19, GPIO.LOW)
    GPIO.output(26, GPIO.LOW)
    pisca = db.child("controlo").child("piscas").get()
```

Anexo IV

Código da página Index.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Skye</title>
  <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body>
  <header>
    <div class="container">
      <nav>
        <h1 class="skye"><a
href="index.php"><b>sk<span>y</span>e</b></a></h1>
        <ul>
          <li><a href="index.html">Home</a></li>
          <li><a href="sobre.html">Sobre</a></li>
          <li><a href="manual.html">Manual</a></li>
          <li><a href="galeria.html">galeria</a></li>
        </ul>
      </nav>
    </div>
  </header>
  <div class="why">
    <div class="teste">
      <div class="animacao">
        <small>Sê bem vindo</small>
        <h1>Ao futuro</h1>
        <button>Clica Aqui!</button>
      </div>
    </div>
  </div>
  <div>
    <div class="nuvem">
      
      
      
      
      
      
      
    </div>
  </div>
</body>

```

</html>

Anexo V

Código da página Sobre.html


```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>skye</title>

  <link rel="stylesheet" href="css/style.css">

  <style>
    h4{
      font-size: 28px;
      padding-bottom: 20px;
    }

    p{
      font-size: 14px;
    }

    .responsive {
      width: 100%;
      height: auto;
    }
  </style>
</head>
<body>

  <header>
    <div class="container">
      <nav>
        <h1 class="skye"><a
href="index.php"><b>sk<span>y</span>e</b></a></h1>
        <ul>
          <li><a href="index.html">Home</a></li>
          <li><a href="sobre.html">Sobre</a></li>
          <li><a href="manual.html">Manual</a></li>
          <li><a href="galeria.html">galeria</a></li>
        </ul>
      </nav>
    </div>
  </header>

  <section id="blog">

```

```

<div class="blog-box">
  <div class="blog-img">
    
  </div>
  <div class="blog-details">
    <h4>Programação</h4>
    <p>A primeira situação realizada neste projeto foi a
finalização do código. Tanto a skye como a aplicação projetada para a skye foi
realizada através de programação, sendo utilizado pyhton para a skye e java para
a aplicação android. Este site também foi construido a partir das linguagens de
programação HTML, CSS e PHP.</p>
  </div>
  <h1>01/04</h1>
</div>

<div class="blog-box">
  <div class="blog-img">
    
  </div>
  <div class="blog-details">
    <h4>Skye</h4>
    <p>Este é o Fisico da skye. O projeto no inicio foi projetado
para ser somente um mero "cerebro", uma simples aplicação na qual seria a sua
assistente virtual, no entanto, a inovação de colocar esta assistente dentro de
um carro telecomandado foi uma ideia fantástica no qual floresceu este fantástico
projeto.</p>
  </div>
  <h1>02/04</h1>
</div>

<div class="blog-box">
  <div class="blog-img">
    
  </div>
  <div class="blog-details">
    <h4>Os meus criadores!</h4>
    <p>Estes são os meus criadores: <br> Afonso Almeida; <br>
Ricardo Mendes <br> Tiago Castro <br> E aquela ali em baixo sou eu, a Skye!.</p>
  </div>
  <h1>03/04</h1>
</div>

<div class="blog-box">
  <div class="blog-img">

```

```

</div>
<div class="blog-details">
  <h4>aplicação</h4>
  <p>A skye contem uma aplicação muito intuitiva para que
qualquer um tenha a facilidade de trabalhar com a mesma.<br> A aplicação dispõe
de vários botões sendo eles os piscas, o botão de voz, o acelerador, o travão e
duas setinhas para direcionar a skye, cada um com a sua respetiva
funcionalidade.</p>
</div>
<h1>04/04</h1>
</div>
</section>
</body>
```

Anexo VI

Código da página manual.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Manual</title>
  <link rel="stylesheet" type="text/css" href="css/style.css">
</head>

<style>
  .center{
    margin-right: auto;
    margin-left: auto;
  }

  table{
    width: 100%;
    height: 100%;
    margin-top: 20px;
  }

  th, td {
    border: 1px solid;
    font-size: 20px;
  }

  td{
    width: 5%;
  }

  img{
    display: block;
    margin-left: auto;
    margin-right: auto;
    width: 100%;
    height: auto;
  }

  p{
    margin-right: auto;
    margin-left: auto;
  }
</style>
<body>
```

```

<header>
  <div class="container">
    <nav>
      <h1 class="skye"><a
href="index.php"><b>sk<span>y</span>e</b></a></h1>
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="sobre.html">Sobre</a></li>
        <li><a href="manual.html">Manual</a></li>
        <li><a href="galeria.html">galeria</a></li>
      </ul>
    </nav>
  </div>
</header>
<div class="container">
  <table class="center">
    <tr>
      <th>Imagem</th>
      <th>Função</th>
    </tr>
    <tr>
      <td></td>
      <td>A skye apresenta um pequeno olho a sua frente no qual pisca e, quando o mesmo reage ao utilizador piscando azul escuro/azul claro significa que a skye está a falar ou até mesmo a interagir com o utilizador.</td>
    </tr>
    <tr>
      <td></td>
      <td>A skye apresenta um pequeno olho a sua frente no qual pisca e, quando o mesmo reage ao utilizador piscando azul escuro/verde significa que a skye está a pensar ou a processar informação necessária para informar o utilizador.</td>
    </tr>
    <tr>
      <td></td>
      <td>Dentro da aplicação estão também presentes botões, cada um com a sua função. Estes dois botões apresentados servem para ativar os piscas, sendo o botão direito os piscas direitos e o botão esquerdo os piscas esquerdos, respetivamente.</td>
    </tr>
    <tr>
      <td></td>

```

```

        <td>Dentro da aplicação estão também presentes botões, cada um
        com a sua função.Estes dois botões apresentados servem para ativar os piscas,
        sendo o botão direito os piscas direitos e o botão esquerdo os piscas esquerdos,
        respetivamente.</td>
    </tr>
    <tr>
        <td></td>
        <td>Dentro da aplicação estão também presentes botões, cada um
        com a sua função.Estes dois botões apresentados servem para ativar os piscas,
        sendo o botão direito os piscas direitos e o botão esquerdo os piscas esquerdos,
        respetivamente.</td>
    </tr>
    <tr>
        <td></td>
        <td>Dentro da aplicação estão também presentes botões, cada um
        com a sua função.Estes dois botões apresentados servem para ativar os piscas,
        sendo o botão direito os piscas direitos e o botão esquerdo os piscas esquerdos,
        respetivamente.</td>
    </tr>
    <tr>
        <td></td>
        <td>Dentro da aplicação estão também presentes botões, cada um
        com a sua função.Estes dois botões apresentados servem para ativar os piscas,
        sendo o botão direito os piscas direitos e o botão esquerdo os piscas esquerdos,
        respetivamente.</td>
    </tr>
</table>
</div>
</body>
</html>

```

Anexo VII

Código da página galeria.html


```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="css/style.css">
  <title>skye</title>

  <style>
    body{
      background: linear-gradient(90deg, rgba(255,255,255,1) 0%,
      rgba(135,206,235,1) 15%, rgba(135,206,235,1) 85%, rgba(255,255,255,1) 100%);
    }
  </style>
</head>
<body>
<header>
  <div class="container">
    <nav>
      <h1 class="skye"><a
href="index.php"><b>sk<span>y</span>e</b></a></h1>
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="sobre.html">Sobre</a></li>
        <li><a href="manual.html">Manual</a></li>
        <li><a href="galeria.html">galeria</a></li>
      </ul>
    </nav>
  </div>
</header>
  <div class="post-img">
    
  </div>
  <h3 class="fig">figura 1</h3>
  <p class="desc">olá!</p>

  <div class="quadrado">
    <div class="post-img">
      
    </div>
    <h3 class="fig">figura 2</h3>
    <p class="desc">Descrição!</p>
  </div>

```

```
</div>

<div class="quadrado">
  <div class="post-img">
    
  </div>
  <h3 class="fig">figura 3</h3>
  <p class="desc">Descrição!</p>
</div>

<div class="post col-lg-6 col-md-6 col-12">
  <div class="post-img">
    
  </div>
  <h3 class="fig">figura 4</h3>
  <p class="desc">Descrição!</p>
</div>

<div class="post col-lg-6 col-md-6 col-12">
  <div class="post-img">
    
  </div>
  <h3 class="fig">figura 5</h3>
  <p class="desc">Descrição!</p>
</div>

<div class="post col-lg-6 col-md-6 col-12">
  <div class="post-img">
    
  </div>
  <h3 class="fig">figura 6</h3>
  <p class="desc">Descrição!</p>
</div>

<div class="post col-lg-6 col-md-6 col-12">
  <div class="post-img">
    
  </div>
  <h3 class="fig">figura 7</h3>
  <p class="desc">Descrição!</p>
</div>

<div class="post col-lg-6 col-md-6 col-12">
```

```

<div class="post-img">
  
</div>
<h3 class="fig">figura 8</h3>
<p class="desc">Descrição!</p>
</div>

<div class="post col-lg-6 col-md-6 col-12">
  <div class="post-img">
    
  </div>
  <h3 class="fig">figura 9</h3>
  <p class="desc">Descrição!</p>
</div>

<div class="post col-lg-6 col-md-6 col-12">
  <div class="post-img">
    
  </div>
  <h3 class="fig">figura 10</h3>
  <p class="desc">Descrição!</p>
</div>

<div class="post col-lg-6 col-md-6 col-12">
  <div class="post-img">
    
  </div>
  <h3 class="fig">figura 11</h3>
  <p class="desc">Descrição!</p>
</div>

<div class="post col-lg-6 col-md-6 col-12">
  <div class="post-img">
    
  </div>
  <h3 class="fig">figura 12</h3>
  <p class="desc">Descrição!</p>
</div>

<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"
integrity="sha384-
7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31e0z1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"
crossorigin="anonymous"></script>

```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-
QJHtvGhmr9XOIpI6YVutG+2Q0K9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"
crossorigin="anonymous"></script>
</body>
</html>
```

Anexo VII

Código do ficheiro CSS das páginas

```
@import
url('https://fonts.googleapis.com/css2?family=Luxurious+Roman&display=swap');

*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

html{
  font-size: 10px;
  font-family: "Luxurious Roman", sans-serif;
}

.yes{
  overflow-x: hidden;
}

body{
  background: rgb(135, 206, 235);
}

a{
  text-decoration: none;
  color: black;
}

/* _____ INICIO DA NAVBAR _____
*/

header{
  font-size: 10px;
  font-family: "Luxurious Roman", sans-serif;
  width: 100%;
  height: 15vh;
}

.container{
  max-width: 120rem;
  width: 90%;
  margin: 0 auto;
}
```

```
nav{
  padding-top: 1rem;
  display: flex;
  justify-content: space-between;
  align-items: center;
  text-transform: uppercase;
  font-size: 1.6rem;
}

.skye{
  font-size: 3rem;
  font-weight: 300;
  transform: translateX(-100rem);

  animation: slidein .5s forwards;
}

.skye span{
  color: darkcyan;
}

nav ul{
  display: flex;
}

nav ul li{
  list-style: none;
  transform: translateX(100rem);

  animation: slidein .5s forwards;
}

nav ul li:nth-child(1){
  animation-delay: 0s;
}

nav ul li:nth-child(2){
  animation-delay: .4s;
}

nav ul li:nth-child(3){
  animation-delay: .6s;
}
```

```
nav ul li:nth-child(4){
  animation-delay: .8s;
}

nav ul li a{
  padding: 1rem 0;
  margin: 0 7rem;
  position: relative;
  letter-spacing: 1px;
}

nav ul li a:last-child{
  margin-right: 0;
}

nav ul li a::before, nav ul li a::after{
  content: '';
  position: absolute;
  width: 100%;
  height: 2px;
  background-color: darkcyan;
  left: 0;
  transform: scaleX(0);
  transition: all .5s;
}

nav ul li a::before{
  top: 0;
  transform-origin: left;
}

nav li a::after{
  bottom: 0;
  transform-origin: right;
}

nav ul li a:hover::before, nav ul li a:hover::after{
  transform: scaleX(1);
}

@keyframes slidein {
  from{
```



```
    }
    to{
        transform: scaleX(1);
    }
}

@keyframes fadein{
    from{
        opacity: 0;
    }
    to{
        opacity: 1;
    }
}

a:hover{
    text-decoration: none;
    color:black;
}

/* _____ FIM DA NAVBAR _____ */

/* _____ INICIO DO LAYOUT _____ */

.why{
    width: 100%;
    height: 85vh;
    background-color: skyblue;
    background-size: cover;
    background-position: center;
    position: relative;
    overflow: hidden;
}

.animacao{
    animation: fadein 3s;
}

button{
    font-weight: bolder;
    color: black;
    padding: 20px 50px;
```

```
background: transparent;
border: 4px solid black;
border-radius: 20px;
outline: none;
cursor: pointer;
}

.teste{
  color: black;
  position: absolute;
  top: 40%;
  left: 8%;
  transform: translateY(-50%);
  z-index: 2;
}

h1{
  font-size: 160px;
  margin: 10px 0 30px;
  line-height: 80px;
}

small{
  font-size: 50px;
  margin: 10px 0 30px;
  line-height: 80px;
}

.nuvem img{
  width: 150px;
  animation: nuvem 7s linear infinite;
}

.nuvem{
  width: 100%;
  display: flex;
  align-items: center;
  justify-content: space-around;
  position: absolute;
  bottom: 0;
}

@keyframes nuvem{
```

```
0%{
  transform: translateY(0);
  opacity: 0;
  visibility: visible;
}
50%{
  opacity: 1;
}
70%{
  opacity: 1;
}
100%{
  transform: translateY(-80vh);
  opacity: 0;
}
}

.nuvem img:nth-child(1){
  animation-delay: 2s;
  visibility: hidden;
}
.nuvem img:nth-child(2){
  animation-delay: 1s;
  visibility: hidden;
}
.nuvem img:nth-child(3){
  animation-delay: 3s;
  visibility: hidden;
}
.nuvem img:nth-child(4){
  animation-delay: 5s;
  visibility: hidden;
}
.nuvem img:nth-child(5){
  animation-delay: 1.5s;
  visibility: hidden;
}
.nuvem img:nth-child(6){
  animation-delay: .5s;
  visibility: hidden;
}
.nuvem img:nth-child(7){
  animation-delay: 4.5s;
```

```
visibility: hidden;
}

/* _____ FIM DO LAYOUT _____ */

/* _____ INICIO DO SOBRE _____ */

#page-header .blog-header{
  background-image: url("img/banner/b1.jpg");
  width: 100%;
  height: 40vh;
  background-size: cover;
  display: flex;
  justify-content: center;
  text-align: center;
  flex-direction: column;
  padding: 14px;
}

#blog{
  padding: 150px 150px 0 150px;
}

#blog .blog-box{
  display: flex;
  align-items: center;
  width: 100%;
  position: relative;
  padding-bottom: 100px;
}

#blog .blog-img{
  width: 100%;
  height: auto;
}

#blog .blog-details{
  width: 100%;
  animation: fadein 7s;
  height: auto;
}

#blog .blog-box h1{
```

```
position: absolute;
top: -60px;
left: 2px;
font-size: 70px;
font-weight: 700;
z-index: -9;
}

/* _____ INICIO DA GALERIA
_____ */

.post-img{
margin: auto;
width: 60%;
border-radius: 10px 50px;
border: 3px solid white;
padding: 10px;
}

img{
width: 100%;
border-radius: 15px 50px;
padding: 10px;
flex: auto;
opacity: 1;
}

.fig, .desc{
text-align: center;
font-size: 25px;
}

.desc{
margin-bottom: 60px;
}
```

Anexo VIII

Código do ficheiro Java da app

```
package com.example.skye;

import static com.google.firebase.database.FirebaseDatabase.getInstance;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.view.MotionEvent;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.FirebaseApp;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.HashMap;

public class MainActivity extends AppCompatActivity {

    ImageView speechButton;
    EditText speecheText;
    ImageView direita;
    ImageView esquerda;
    ImageView travao;
    ImageView acc;
    ImageView med;
    ImageView max;
    ImageView pesq;
    ImageView pdir;
```

```

ImageView haz;
private static final int RECOGNIZER_RESULT = 1;

@Override

protected void onCreate (Bundle savedInstanceState){
    super.onCreate(savedInstanceState);
    getSupportActionBar().hide();
    setContentView(R.layout.activity_main);
    pesq =findViewById(R.id.imageButton4);
    pdir=findViewById(R.id.imageButton3);
    haz=findViewById(R.id.imageButton9);
    speechButton = findViewById(R.id.imageView);
    speechtext = findViewById(R.id.editText);
    direita = findViewById(R.id.imageView2);
    esquerda = findViewById(R.id.imageView3);
    travao = findViewById(R.id.imageButton8);
    acc = findViewById(R.id.imageView6);
    med= findViewById(R.id.imageView7);
    max=findViewById(R.id.imageView8);

    FirebaseDatabase database = FirebaseDatabase.getInstance("https://skye-
pap-default-rtdb.europe-west1.firebaseio.com");
    DatabaseReference mDatabaseRef = database.getReference();

    speechButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent speechIntent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
            speechIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
            speechIntent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Em que
posso ajudar?");
            startActivityForResult(speechIntent, RECOGNIZER_RESULT);

        }
    });
}

```



```
//<servo>

    direita.setOnTouchListener(new View.OnTouchListener() {
        @SuppressWarnings("ClickableViewAccessibility")
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            FirebaseDatabase database =
FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-
west1.firebaseio.com");
            DatabaseReference mDatabaseRef = database.getReference();
            if(event.getAction() == MotionEvent.ACTION_DOWN) {
                mDatabaseRef.child("controlo").child("servo").setValue(2);
                return true;
            }
            if(event.getAction() == MotionEvent.ACTION_UP) {
                mDatabaseRef.child("controlo").child("servo").setValue(0);
                return true;
            }

            return false;
        }
    });

    esquerda.setOnTouchListener(new View.OnTouchListener() {
        @SuppressWarnings("ClickableViewAccessibility")
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            FirebaseDatabase database =
FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-
west1.firebaseio.com");
            DatabaseReference mDatabaseRef = database.getReference();
            if(event.getAction() == MotionEvent.ACTION_DOWN) {
                mDatabaseRef.child("controlo").child("servo").setValue(1);
                return true;
            }
            if(event.getAction() == MotionEvent.ACTION_UP) {
                mDatabaseRef.child("controlo").child("servo").setValue(0);
                return true;
            }
        }
    });
```

```

    }

    return false;
}
});

</servo>

<motor>
    travao.setOnTouchListener(new View.OnTouchListener() {
        @SuppressWarnings("ClickableViewAccessibility")
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            FirebaseDatabase database =
FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-
west1.firebaseio.com");
            DatabaseReference mDatabaseRef = database.getReference();
            if(event.getAction() == MotionEvent.ACTION_DOWN) {
                mDatabaseRef.child("controlo").child("motor").setValue(1);
                mDatabaseRef.child("controlo").child("marcha").setValue(1);
                return true;
            }
            if(event.getAction() == MotionEvent.ACTION_UP) {
                mDatabaseRef.child("controlo").child("motor").setValue(0);
                mDatabaseRef.child("controlo").child("marcha").setValue(0);
                return true;
            }

            return false;
        }
    });

    acc.setOnTouchListener(new View.OnTouchListener() {
        @SuppressWarnings("ClickableViewAccessibility")
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            FirebaseDatabase database =
FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-
west1.firebaseio.com");
            DatabaseReference mDatabaseRef = database.getReference();
            if(event.getAction() == MotionEvent.ACTION_DOWN) {

```

```

        mDatabaseRef.child("controlo").child("motor").setValue(2);
        return true;
    }
    if(event.getAction() == MotionEvent.ACTION_UP) {
        mDatabaseRef.child("controlo").child("motor").setValue(0);
        return true;
    }

    return false;
}
});
//</motor>

//<luzes>
med.setOnClickListener(new View.OnClickListener() {
    int pos = 0;
    public void onClick(View v) {
        FirebaseDatabase database =
FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-
west1.firebaseio.app");
        DatabaseReference mDatabaseRef = database.getReference();
        if (pos == 0){
            mDatabaseRef.child("controlo").child("luzes").setValue(1);
            pos = 1;
        } else if (pos == 1){
            mDatabaseRef.child("controlo").child("luzes").setValue(0);
            pos = 0;
        }
    }
});

max.setOnClickListener(new View.OnClickListener() {
    int pos = 0;
    public void onClick(View v) {
        FirebaseDatabase database =
FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-
west1.firebaseio.app");
        DatabaseReference mDatabaseRef = database.getReference();
        if (pos == 0){
            mDatabaseRef.child("controlo").child("luzes").setValue(2);
            pos = 1;
        } else if (pos == 1){

```

```

        FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-west1.firebaseio.com").getReference().child("controlo").child("luzes").setValue(0);
        pos = 0;
    }
}
});

</luzes>

<piscas>
pesq.setOnClickListener(new View.OnClickListener() {
    int pos = 0;
    public void onClick(View v) {
        FirebaseDatabase database =
FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-west1.firebaseio.com");
        DatabaseReference mDatabaseRef = database.getReference();
        if (pos == 0){
            mDatabaseRef.child("controlo").child("piscas").setValue(1);
            pos = 1;
        } else if (pos == 1){
            mDatabaseRef.child("controlo").child("piscas").setValue(0);
            pos = 0;
        }
    }
});

pdir.setOnClickListener(new View.OnClickListener() {
    int pos = 0;
    public void onClick(View v) {
        FirebaseDatabase database =
FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-west1.firebaseio.com");
        DatabaseReference mDatabaseRef = database.getReference();
        if (pos == 0){
            mDatabaseRef.child("controlo").child("piscas").setValue(2);
            pos = 1;
        } else if (pos == 1){
            mDatabaseRef.child("controlo").child("piscas").setValue(0);
            pos = 0;
        }
    }
});

```

```

        haz.setOnClickListener(new View.OnClickListener() {
            int pos = 0;
            public void onClick(View v) {
                FirebaseDatabase database =
FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-
west1.firebaseio.com");
                DatabaseReference mDatabaseRef = database.getReference();
                if (pos == 0){
                    mDatabaseRef.child("controlo").child("piscas").setValue(3);
                    pos = 1;
                } else if (pos == 1){
                    mDatabaseRef.child("controlo").child("piscas").setValue(0);
                    pos = 0;
                }
            }
        });
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {

    if(requestCode == RECOGNIZER_RESULT && resultCode == RESULT_OK){

        ArrayList<String> matches =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
        speechtext.setText(matches.get(0).toString());
        FirebaseDatabase database =
FirebaseDatabase.getInstance("https://skye-pap-default-rtdb.europe-
west1.firebaseio.com");
        DatabaseReference mDatabaseRef = database.getReference();
        mDatabaseRef.child("controlo").child("voz").setValue(matches.get(0).t
oString());
        mDatabaseRef.child("controlo").child("flag").setValue("False");
    }

    super.onActivityResult(requestCode, resultCode, data);
}
}

```

```
public void onDestroy() {  
    super.onDestroy();  
    FirebaseDatabase database = FirebaseDatabase.getInstance("https://skye-  
pap-default-rtbd.euope-west1.firebaseio.app");  
    DatabaseReference mDatabaseRef = database.getReference();  
    mDatabaseRef.child("controlo").child("uso").setValue(0);  
}  
}
```

Anexo IX

Código do ficheiro *manifest* da app

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.skye">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Skye">

        <activity
            android:name=".MainActivity"
            android:screenOrientation="sensorLandscape"

            android:exported="true"
            tools:ignore="ExtraText">
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```


Anexo X

Código do ficheiro *xml* da app

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:backgroundTint="@android:color/holo_green_dark"
android:foregroundTint="@android:color/holo_green_light"
tools:context=".MainActivity">

<EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:visibility="gone"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.043" />

<ImageButton
    android:id="@+id/imageButton8"
    android:layout_width="117dp"
    android:layout_height="143dp"
    android:backgroundTint="@color/white"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.783"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.828"
    app:srcCompat="@drawable/travao"
    app:tint="@color/black"
    tools:ignore="MissingConstraints,SpeakableTextPresentCheck" />

<ImageButton
    android:id="@+id/imageButton3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:backgroundTint="@color/white"

```

```
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.956"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.153"  
app:srcCompat="@drawable/dir"  
app:tint="@android:color/holo_green_light" />
```

```
<ImageButton  
    android:id="@+id/imageButton4"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:backgroundTint="@color/white"  
    android:rotationY="180"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.772"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.153"  
    app:srcCompat="@drawable/dir"  
    app:tint="@android:color/holo_green_light" />
```

```
<ImageView  
    android:id="@+id/imageView8"  
    android:layout_width="73dp"  
    android:layout_height="70dp"  
    android:rotationY="180"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.261"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.149"  
    app:srcCompat="@drawable/max"  
    app:tint="@android:color/holo_blue_dark"  
    tools:ignore="MissingConstraints" />
```

```
<ImageView  
    android:id="@+id/imageView7"  
    android:layout_width="82dp"  
    android:layout_height="59dp"
```

```

android:rotationY="-180"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.066"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.144"
app:srcCompat="@drawable/med"
app:tint="@android:color/holo_green_light"
tools:ignore="MissingConstraints" />

```

```

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="85dp"
    android:layout_height="83dp"
    android:rotation="90"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.266"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.676"
    app:srcCompat="@android:drawable/arrow_up_float" />

```

```

<ImageView
    android:id="@+id/imageView3"
    android:layout_width="85dp"
    android:layout_height="83dp"
    android:rotation="-90"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.061"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.676"
    app:srcCompat="@android:drawable/arrow_up_float" />

```

```

<ImageButton
    android:id="@+id/imageButton9"
    android:layout_width="105dp"
    android:layout_height="109dp"
    android:backgroundTint="@color/white"
    app:layout_constraintBottom_toBottomOf="parent"

```

```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.109"
app:srcCompat="@drawable/haz" />

<ImageView
    android:id="@+id/imageView"
    android:layout_width="109dp"
    android:layout_height="98dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.503"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.677"
    app:srcCompat="@android:drawable/ic_btn_speak_now" />

<ImageView
    android:id="@+id/imageView6"
    android:layout_width="61dp"
    android:layout_height="134dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.958"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.801"
    app:srcCompat="@drawable/acc"
    tools:ignore="MissingConstraints" />

@drawable/travao
Button
parent
parent
0.841
parent
parent
0.911

</androidx.constraintlayout.widget.ConstraintLayout>

```

Anexo XI

Glossário

Python - Python é uma linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional e forte. Atualmente é uma das linguagens mais utilizadas e requisitadas, tendo vindo a aumentar a sua popularidade e tem como principal característica possibilitar uma fácil leitura do código.

HTML – HTML é uma linguagem de marcação que é utilizada para desenvolver páginas web.

Java – Java é uma linguagem orientada a objetos, estruturada, imperativa, genérica, funcional, reflexiva e concorrente. É uma das linguagens mais populares da atualidade. Pode ser utilizada para diferentes objetivos, sendo um deles a criação de aplicações para android.

Firebase – Firebase é uma plataforma criada pela Google para criar aplicações para smartphones e web. A partir dela podemos guardar dados, alojar websites, entre outros, gratuitamente.

Photoshop – Photoshop é um software desenvolvido pela Adobe Systems que permite manipular imagens e vídeos. É o editor de imagem bidimensional mais popular e mais utilizado do mercado.

Android Studio – é um IDE criado para desenvolver software dispositivos android. É baseado no software IntelliJ IDEA da JetBrains. Podemos desenvolver aplicações utilizando Java, Kotlin e C++.

Raspbian – Raspbian é um sistema operativo baseado em Debian otimizado para o hardware do Raspberry Pi.

Thread – processo que está em execução em simultâneo que determinado programa realiza

Bluetooth – bluetooth é uma rede sem fios, denominada de WPAN (*wireless personal area network*). Esta permite trocar informações entre dispositivos através de uma frequência de rádio de curto alcance.

Stack – uma *stack* é uma estrutura de dados linear utilizada para guardar uma coleção de objetos.

Activity – uma activity fornece a janela na qual a app desenha a própria interface.