Instituto Superior de Tecnologias Avançadas
Licenciatura em Engenharia Informática

# TRILHAR

Uma APP na interceção de um prisma desportivo e turístico

Realizado por:

João José da Conceição Martins, nº 30367

Coordenador – Professor Dr. Pedro Brandão

Orientadora – Professora Dra. Dulce Mourato

Lisboa, Ano Letivo 2022/2023

Instituto Superior de Tecnologias Avançadas
Licenciatura em Engenharia Informática

# TRILHAR

Uma APP na interceção de um prisma desportivo e turístico

Realizado por:

João José da Conceição Martins, nº 30367

Coordenador – Professor Dr. Pedro Brandão

Orientadora – Professora Dra. Dulce Mourato

Lisboa, Ano Letivo 2022/2023

# Agradecimentos

Gostaria de deixar um especial agradecimento ao impulsionador desta ideia, o meu pai, à minha prima Patrícia que desenvolveu mais a fundo o design e UI deste projeto e a todos os meus amigos envolvidos no desenvolvimento e no teste desta aplicação.

# Resumo

Este projeto tem como objetivo a criação de uma aplicação mobile no âmbito turístico e desportivo mais precisamente no ciclismo. A premissa de interligação destes dois mundos passa por ser possível disponibilizar ao utilizador uma aplicação que à medida que a atividade desportiva é realizada, simultaneamente, a atividade turística é praticada. A aplicação baseia-se na disponibilização de rotas de ciclismo, sendo coincidentemente turísticas, ou seja, existem pontos nessas mesmas rotas relevantes para a atividade turística.

Esta ideia partiu de um projeto familiar já existente. A Casa do Zé é um alojamento local localizado na vila do Soito, concelho do Sabugal, credenciado pelo Bikotel, marca que certifica a qualidade do serviço numa unidade hoteleira no âmbito do ciclismo. Esta zona, porventura, é uma zona rica de história com paisagens deslumbrantes, conta com inúmeras serras na área redundante, tais como a Serra de Malcata e a Serra do Homem de Pedra e com um rio que nasce e se estende por todo o seu território, o Rio Côa.

Todo este conjunto de fatores contribuíram para a implementação desta aplicação, o objetivo foi traçado a partir daí, como se poderia oferecer aos clientes uma experiência mais imersiva e dinâmica que juntasse o melhor dos dois mundos que esta zona tem para oferecer, por um lado, rotas desafiantes por inúmeros trilhos, por outro, a entrega de todas as paisagens e pontos repletos de história.

Palavras-Chave: App, Desporto, Ciclismo, Turismo, Flutter, Firebase.

# Abstract

This project aims to create a mobile application in the tourism and sports field, more precisely in cycling. The premise of interconnection of these two worlds is that it is possible to provide the user with an application that, as the sporting activity is carried out, simultaneously, the tourist activity is practiced. The application is based on the availability of cycling routes, which are coincidentally touristic, that is, there are points on these same routes that are relevant to tourist activity.

This idea came from an existing family project. A Casa do Zé is a local accommodation located in the village of Soito, municipality of Sabugal, accredited by Bikotel, a brand that certifies the quality of service in a hotel unit in the field of cycling. This area, perhaps, is an area rich in history with stunning landscapes, has numerous mountain ranges in the redundant area, such as Serra de Malcata and Serra do Homem de Pedra and with a river that rises and extends throughout its territory, the Côa River.

This whole set of factors contributed to the implementation of this application, the objective was traced from there, how could we offer customers a more immersive and dynamic experience that combined the best of both worlds that this area has to offer, on one hand, challenging routes along countless trails, on the other hand, the delivery of all the landscapes and places full of history.



Keywords: App, Sports, Cycling, Tourism, Flutter, Firebase.

# Abreviaturas

App – Aplicação
HTML - Hypertext Markup Language
CSS - Cascading Style Sheets
SQL - Structured Query Language
API - Application Programming Interface
UI – User Interface
SMS - Short Message Service
GPS - Global Positioning System
GPX - GPS Exchange Format
GNSS - Global Navigation Satellite System
BTT - Bicicleta de Todo-o-Terreno
FPC - Federação Portuguesa de Ciclismo

# Índice

*Introdução* ..... *10*
*Revisão de Literatura* ..... *11*
*Desenvolvimento Nativo* ..... *12*
*Desenvolvimento Híbrido* ..... *13*
*Flutter* ..... *15*
*Base de Dados* ..... *16*
*Base de Dados NoSQL* ..... *16*
*Firebase* ..... *17*
*Materiais e Métodos de Desenvolvimento* ..... *19*
*Calendarização* ..... *20*
*Script da Aplicação* ..... *21*
*Resultados e Discussão dos Resultados* ..... *22*
*Idealização* ..... *22*
*Desenho, Projeção e Implementação* ..... *24*
*Bugs e Testes no Terreno* ..... *34*
*Conclusão* ..... *36*
*Futuros Updates e Alterações* ..... *37*
*Bibliografia* ..... *38*

# Índice de Figuras

*Figura 1 - Gráfico comparativo entre as várias gerações de Intel vs. Apple* ..... *14*
*Figura 2 - Fluxograma representativo da aplicação* ..... *21*
*Figura 3 - Idealização do Ecrã Mapa* ..... *22*
*Figura 4 - Ecrã Boas-Vindas* ..... *24*
*Figura 5 - Ecrã Login* ..... *24*
*Figura 6 - Ecrã Registar* ..... *24*
*Figura 7 - Exemplo do código do botão Entrar* ..... *25*
*Figura 8 - Ecrã Perfil* ..... *26*
*Figura 9 - Ecrã Perfil (Expandido)* ..... *26*
*Figura 10 - Ecrã Definições* ..... *26*
*Figura 11 - Ecrã Percursos* ..... *27*
*Figura 12 - Ecrã Estrada* ..... *27*
*Figura 13 - Ecrã BTT* ..... *27*
*Figuras 14 - Parâmetros FPC* ..... *28*
*Figura 15 - Exemplo de código* ..... *28*
*Figura 16 - Ecrã Mapa* ..... *29*
*Figura 17 - Pop-up Ponto de Interesse* ..... *29*
*Figura 18 - Ecrã Ponto de Interesse* ..... *29*
*Figura 19 - Firebase Database* ..... *30*
*Figura 20 - Firebase Storage* ..... *30*
*Figura 21 - Exemplo de código do Geolocator* ..... *31*
*Figura 22 - Exemplo de código* ..... *31*
*Figura 23 - Exemplo de código* ..... *31*
*Figura 24 - Exemplo de código* ..... *32*
*Figura 25 - Ecrã Mapa (Recording)* ..... *32*
*Figura 26 - Ecrã Stats* ..... *32*

# Índice de Tabelas

*Tabela 1 - Calendarização das tarefas para o desenvolvimento da Aplicação .....20*

# Introdução

Este projeto tem como objetivo a construção de uma aplicação mobile no âmbito desportivo e turístico, que por sua vez, tem como finalidade a aglomeração de percursos de determinada região assim como todos os seus pontos turísticos presentes nesses suprarreferidos percursos. A tecnologia que foi utilizada para a sua construção, foi o *Flutter*, uma nova Framework da Google que promete revolucionar o mercado de aplicações híbridas.

A ideia surgiu através de uma parceria com a Casa do Zé, alojamento local localizado no Sabugal. O primeiro projeto a ser desenvolvido para esta empresa foi o seu site, também ele todo desenvolvido em *Flutter*, após isso, idealizou-se como se poderia divulgar de uma forma dinâmica quer os trilhos redundantes, quer os pontos turísticos da região, ora, sendo esta casa muito virada para o desporto, tentou-se juntar o melhor dos dois mundos, surgindo então a TRILHAR.

O objetivo principal será então a criação de uma aplicação em iOS e Android onde o foco principal será tornar acessível e documentado os pontos de interesse presentes em determinadas rotas de ciclismo e/ou pedestres, através do recurso GPS de uma determinada região.

Para além deste objetivo, também se irá criar outras funcionalidades que irão de encontro a complementar este foco principal:

- A Criação de um painel de utilizador singular onde conste informações sobre os acumulados, isto é, aquilo que o utilizador já acumulou em determinado empreendimento com o uso da aplicação (quilómetros, ganho de elevação, duração e médias).
- Criação de um painel Ranking, onde irá constar uma classificação amigável de todos os clientes que passaram no empreendimento e obtiveram os melhores resultados em determinado percurso e em determinado momento. (Com o objetivo de incentivar o regresso do cliente).
- Desenvolvimento da funcionalidade Live onde se pretende acompanhar o utilizador durante o percurso, e comunicar os devidos pontos de interesse através de notificações e/ou recurso a áudio.

# Revisão de Literatura

É notório que a utilização de aplicações, no prisma desportivo, tem vindo a ter cada vez mais impacto na maneira como são praticadas as diferentes modalidades. Poderá dizer-se, inclusive, que o uso tecnológico veio, efetivamente, revolucionar a maneira como essas modalidades são praticadas e partilhadas.

> O Desporto está a mudar. Na sociedade industrial, o desporto tinha como intuito principal a preparação dos indivíduos para o trabalho, para o esforço. Na era pósindustrial já não se verifica o mesmo. As inovações tecnológicas melhoram a eficácia laboral e libertam o indivíduo para atividades de lazer. O esforço é substituído pelo lazer, pela fruição e pelo hedonismo (Rodrigues, 2001). Apud (Valente et al., 2018, P28)

Exemplo do supradito é o Strava, uma aplicação que está já bastante presente no mundo dos desportistas (quer estejamos a falar de profissionais ou não).

> "We're the largest sports community in the world. Over 100 million athletes in 195 countries use Strava – so whatever your activity and goals, you'll have a community at your back (Strava, n.d.)."

O Strava (doravante designado por aplicação), assim como muitas outras semelhantes, veio proporcionar uma maneira fácil e interativa ao utilizador poder partilhar, com a comunidade, parâmetros e informações relevantes da sua atividade desportiva (a título de exemplo: totalidade de quilómetros percorridos, tempo, elevação acumulada, etc.) num determinado espaço de tempo: semanal, mensal ou até mesmo anual. Acresce que, através deste tipo de aplicações, possibilita-se ainda a partilha das rotas efetuadas pelo utilizador, sendo possível não só visualizá-las num mapa georreferenciado, como, saber quantos quilómetros têm as mesmas, o tempo que o utilizador demorou a efetuá-las, a frequência cardíaca atingida durante a atividade, entre outras informações de relevo.

Posto isto, é possível afirmar que com o crescimento exponencial que a aplicação tem vindo a ter desde 2009 (data do seu lançamento), existiu um consequente aumento da biblioteca de rotas e percursos que a aplicação disponibiliza, sendo certo que esta configura uma mais-valia para aqueles que gostam de competir entre si, visando realizar determinadas rotas num

melhor tempo que atletas anteriores, ou até mesmo para aqueles que apenas pretendem conhecer determinados trilhos.

> A Strava user might use data in particular ways during their physical activity (to monitor their pace or heart rate) or sometime shortly thereafter (to compare their efforts over time), but this very same data can have a different use value for other (Millington, 2016, p. 1).

(Couture, 2021, p. 7) especifica ainda sobre o mesmo tema: "Another user might use the same data to learn a new running route".

A TRILHAR, pretende seguir um pouco esta premissa do Strava, mas com a particularidade de serem percursos apenas introduzidos por administradores de empreendimentos turísticos, para assegurar a sua qualidade e relevância.

A maneira como a tecnologia e, mais especificamente, as aplicações mobile, vieram impulsionar este ramo, faz-nos questionar, numa perspetiva mais intrínseca do ramo informático, em que módulos e como é que estas são desenvolvidas.

Nesta senda, releva referir que, nos dias de hoje, existem duas principais formas de desenvolver aplicações móveis, nomeadamente:

- Uma forma nativa, que se consubstancia num desenvolvimento através da utilização das linguagens e frameworks de cada plataforma (Objective-C ou Swift, para iOS; e Java ou Kotlin, para Android); e
- Uma abordagem híbrida, podendo esta ramificar-se em dois ramos. Um deles resultante do recurso à utilização de uma WebView com o desenvolvimento em HTML5, CSS e JavaScript que, posteriormente, são convertidos e traduzidos por uma API nativa; o outro originário da utilização de frameworks especialmente desenvolvidas para esta multiplataforma, tais como o React Native, o Xamarin, ou o Flutter.

## Desenvolvimento Nativo

Recorrendo a uma abordagem nativa, é utilizada a linguagem base de cada plataforma. Desta maneira é possível utilizar todos os recursos do sistema operativo, assim como aceder, de forma inequívoca, ao hardware em questão. Assim, e conseguindo tirar total proveito da

simbiose software/hardware, a forma nativa acaba por oferecer a maior performance e estabilidade no âmbito do desenvolvimento de software mobile singular (uni plataforma).

> "The native mobile application runs on the operating system of the device and thus needs to be adapted for various target devices. Native, enables developers to utilize full capabilities of a device, thus have better performance (Ahmad et al., 2018, p. 1)."

Ainda que esta seja a forma mais eficiente, em termos de performance, de se desenvolver software mobile, não poderá deixar de se ressalvar que esta nem sempre é a melhor maneira de o fazer uma vez que, por ser uma abordagem singular, apresenta um custo mais elevado a todos os níveis.

> "However, developing a native application for multiple platforms requires expertise with the programming languages, Application Programming Interfaces (API) and Software Development Kits (SDKs) of each target platform, which demands additional resources (skills, time and costs) (Ahmad et al., 2018, p. 1)."

## Desenvolvimento Híbrido

A abordagem híbrida conta com a utilização de frameworks desenvolvidas, especificamente, para este propósito. A grande vantagem deste tipo de desenvolvimento é, indubitavelmente, a utilização de um único código para diversas plataformas (Windows, MacOS, Linux, iOS, Android, entre outras). Consequentemente, a aposta nesta área tem tido cada vez mais adesão por parte das grandes tecnológicas. Exemplo disso foi a aquisição, em 2015, pela Microsoft, da Xamarin. A Xamarin é uma empresa que, em 2013, se introduziu no mercado multiplataforma, com o desenvolvimento do próprio Xamarin (uma framework desenvolvida em C#). A esta seguiram-se o Facebook, (que, no mesmo ano, introduziu o React Native, baseado em JavaScript) e, por fim, a Google (que, em 2017, com a introdução do Flutter, que utiliza a sua própria linguagem de programação: o Dart).

Do *supra* exposto resulta que, contrariamente ao desenvolvimento nativo, este tipo de desenvolvimento acaba por não conseguir dar ao utilizador o desempenho máximo possível que o dispositivo consegue proporcionar. Isto sucede porque, por ter uma camada abstrata extra para conseguir traduzir o código para API's nativas, existe, consecutivamente, uma inevitável perda de performance.

Porém, no mundo em que vivemos atualmente, onde a performance dos dispositivos que levamos no nosso bolso já ultrapassa os que vivem nas nossas secretárias, essa dita perda de performance acaba por se tornar irrelevante, na maioria dos casos.

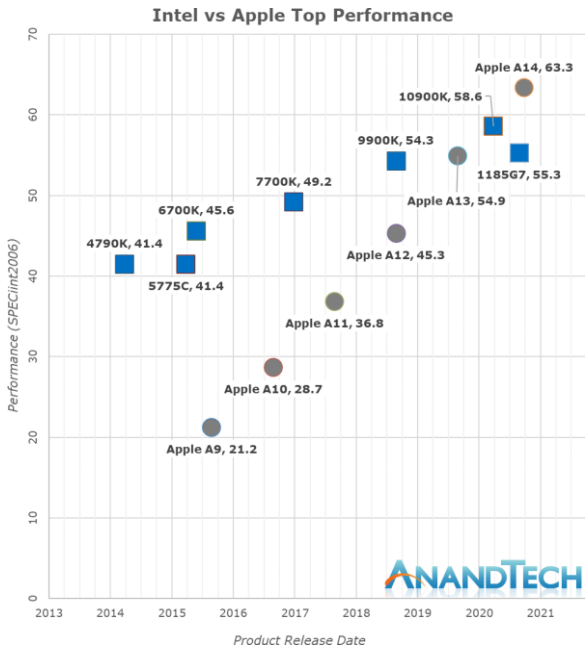


*Figura 1 - Gráfico comparativo entre as várias gerações de Intel vs. Apple (Frumusanu, 2020, p. 1).*

Neste gráfico podemos observar o crescimento linear que se verificou nos iPhones nos últimos anos, relativamente à não tão contínua Intel. É ainda possível identificar que, a empresa criada por Steve Jobs, ultrapassou a gigante fabricante de processadores, com a introdução do iPhone 11 (equipado com o Apple A13), diferenciando-se assim em 0.6 pontos do mais poderoso processador da Intel até à data criado (o i9-9900k). Esta diferença assentuou-se ainda mais aquando da introdução da geração seguinte, subindo de 0.6 para 4.7 pontos.

É desta forma que o desenvolvimento híbrido tem ganho mais sucesso a cada ano, com o hardware a quebrar constantes barreiras, o software torna-se cada vez mais capaz e eficiente.

Ora, posto isto, e assumindo que a única grande desvantagem deste tipo de desenvolvimento é esta ínfima perda de performance, e ainda, considerando todas as vantagens que este acarreta, tais como o desenvolvimento de um único código para diferentes plataformas, e o seu baixo custo comparado com a forma nativa, podemos assumir que esta abordagem tornasse bastante aliciante no que toca ao desenvolvimento multiplataforma.

## Flutter

Flutter foi a Framework utilizada para desenvolver este projeto. Esta escolha deveu-se não só ao facto desta permitir alcançar os dois ambientes pretendidos (nomeadamente, Android e IOS) com o desenvolvimento de um único código, sendo assim possível optimizar o esforço e o tempo de desenvolvimento, como também devido às suas funcionalidades únicas que o diferenciam dos seus concorrentes.

Uma das principais vantagens do Flutter é a sua capacidade de proporcionar uma experiência ao utilizador de alta qualidade, aproximando-se bastante de uma experiência nativa, considerada a renderização diretamente ao nível do sistema e aproveitando assim a aceleração por hardware. Estas características possibilitam a criação de animações suaves, transições rápidas e tempos de resposta curtos. Desta forma, fica garantida uma experiência de utilizador fluida e agradável. O Flutter conta com uma arquitetura orientada a widgets, que permitem a criação de interfaces ricas, fluidas, responsivas e escaláveis. Uma vez que os widgets são elementos de interface pré-construídos, personalizáveis e flexíveis, permitem a criação de interfaces esteticamente harmoniosas, que se adaptam a diferentes tamanhos de ecrã e dispositivos.

Não obstante esta grande vantagem dos widgets, já pré-construídos pela equipa do Flutter, de poderem ser totalmente personalizáveis pelo programador, acresce que estes podem ainda ser totalmente construídos do zero, possibilitando que a aplicação aufira mais complexidade e funcionalidades mais avançadas, caso seja necessário.

Sendo esta framework totalmente open-source, isso permite que exista uma comunidade ativa e uma ampla variedade de pacotes e bibliotecas disponíveis, facilitando a incorporação de recursos adicionais, tais como a integração de mapas, a autenticação de utilizadores, entre outros de igual relevo.

Outra grande vantagem presente no Dart é o hot-reload. Esta funcionalidade veio revolucionar a maneira como o código, o UI, e o estado da aplicação se relacionam. Explicitando, quando existe uma alteração no código da aplicação, ao invés de o flutter recompilar todo o código na sua integra, o que realmente acontece é que o flutter reconhece o pedaço de código que foi alterado e injeta apenas esse mesmo código. Desta maneira, mantém-se o estado da aplicação, resultando desta forma, durante o desenvolvimento de aplicações, em atualizações de milissegundos na UI.

> Any time you save a change to the source code, it is recompiled and the new version is loaded instantly your app picks up where you left off – in the same spot, with the same state, and same data we call it hot reloading, and it makes the development cycle ridiculously fast and frictionless (Payne, 2019, p. 24).

## Base de Dados

A utilização de uma base de dados acaba, ao fim e ao cabo, por ser um recurso necessário na construção desta aplicação, pelo simples facto de existir a necessidade de armazenar informação singular sobre os utilizadores e de atualizar elementos de forma imediata, eliminando-se assim a limitação de apenas se utilizarem updates da própria aplicação como método de atualização desses ditos elementos.

Esses elementos prefiguram, no fundo, as rotas, as informações sobre essas referidas rotas e os seus pontos de interesse inerentes. De forma a permitir ao administrador que introduza, edite e apague essas informações, tornou-se imperativo encontrar uma solução para este problema, o que apenas se tornou possível através da utilização de uma base de dados.

No contexto do desenvolvimento de aplicações, existem diferentes tipos de bases de dados, sendo as bases de dados relacionais e não relacionais (NoSQL) as mais comuns.

As bases de dados relacionais seguem um modelo baseado em tabelas, onde os dados são organizados em linhas e colunas. Este tipo de base de dados é adequado para estruturas de dados bem definidas e relações complexas entre entidades.

Além das bases de dados relacionais, surgiram as bases de dados NoSQL (Not Only SQL), que oferecem uma abordagem alternativa para armazenar e recuperar dados. Estas bases de dados NoSQL englobam diferentes modelos de dados, tais como documentos, chave-valor, colunas e grafos.

## Base de Dados NoSQL

As bases de dados NoSQL oferecem várias vantagens ao desenvolvimento de aplicações modernas, como a flexibilidade no esquema de dados, escalabilidade horizontal e alta performance. Estas bases de dados são especialmente adequadas para aplicações com requisitos de escalabilidade e processamento de grandes volumes de dados.

> "When we want to go beyond the possibilities that relational databases can offer us, for example, in cases where availability, performance, and scalability are essential then NoSQL databases can be the best choice (Sahatqija et al., 2018, p. 220)."

Além disso, o modelo de dados NoSQL pode ser facilmente adaptado a diferentes tipos de aplicações, incluindo aquelas em que a estrutura dos dados pode mudar com o tempo. Para este projeto que consiste em rotas de ciclismo, a utilização de uma base de dados NoSQL pode oferecer uma maior agilidade na manipulação e armazenamento dos dados relacionados às rotas e aos seus pontos de interesse.

## Firebase

O Firebase é um conjunto de ferramentas de desenvolvimento em cloud que são utilizadas essencialmente em apps mobile e são fornecidas, igualmente, pela Google.

Esta oferece uma variedade de serviços cloud, incluindo a autenticação de utilizadores, o armazenamento de ficheiros e, o mais relevante para este projeto, uma base de dados em tempo real.

O Firestore Database é um serviço de base de dados NoSQL, documental, que oferece diversas funcionalidades. Dentro da panóplia de funcionalidade oferecidas por este, podemos identificar o armazenamento de dados em tempo real, consultas e ordenação de dados, cache local (que melhora a velocidade de acesso aos dados e permite o funcionamento offline). Essa integração é especialmente relevante para o desenvolvimento de aplicações móveis, uma vez que permite o armazenamento e uma recuperação eficiente dos dados, em tempo real, relacionados às rotas de ciclismo e pontos de interesse.

> Firestore is a NoSQL document database that simplifies storing, syncing, and querying data for your mobile and web apps at global scale, Its client libraries provide live synchronization and offline support while its security features and integrations with the Firebase and Google Cloud platforms accelerate building truly serverless apps (Firebase, 2020).

A Cloud Storage é um serviço que funciona, tal como o nome indica, como uma cloud. Este permite que a aplicação faça updates e downloads de ficheiros quando necessário, sendo possível alojar diferentes tipos de ficheiros (áudio, texto, multimédia, entre outros formatos) e

disponibilizá-los aos utilizadores em tempo real. Ademais, ainda oferece a possibilidade de estes ficheiros ficarem guardados em cache, para que a aplicação consiga carregar estes recursos mais rapidamente.

> Firebase Storage provides secure document transfers and downloads for Firebase applications, regardless of network quality, and is supported by Google Cloud Storage, a capable, straightforward, and affordable object storage service. (Chatterjee et al., 2018).

O Firebase Authentication é um serviço que permite adicionar recursos de autenticação e identificação de utilizadores às aplicações de forma rápida e segura. A grande vantagem deste serviço é a possibilidade de implementar diferentes tipos de autenticação, tais como: autenticação por email e password, autenticação através de provedores de redes sociais (Google, Facebook, Twitter) e até mesmo autenticação por SMS. Além disso, o serviço ainda oferece funcionalidades bastante completas, como a gestão de utilizadores, emails de verificação de utilizadores e de redefinição de passwords e a possibilidade de integração com outros serviços presentes no Firebase. Esta é uma solução bastante poderosa, no sentido em que garante a segurança dos utilizadores e facilita bastante o trabalho dos programadores, visto ser uma infraestrutura totalmente desenvolvida pela Google.

> "Firebase Authentication makes authentication easy for end users and developers it allows you to focus on your users, and not the sign-in infrastructure to support them (Firebase, 2020)."

O Flutter possui suporte nativo ao Firebase, facilitando a ligação e interação entre a aplicação e todos estes serviços.

# Materiais e Métodos de Desenvolvimento

Primeiramente, foi utilizada uma metodologia estudo de caso para uma solução à medida, no sentido em que foi necessário o levantamento de requisitos exigidos para o desenho, construção e conceptualização da aplicação.

> "The case study is characterized by a thorough and exhaustive examination of one or a few items to enable a comprehensive and in-depth understanding of them." (Gil,2008)

Os recursos necessários foram, nomeadamente:

- As rotas e a demarcação das mesmas, em formato GPX. Para isso utilizou-se o Garmin Edge 1040, um computador de bordo para bicicletas, bastante avançado , equipado já com novas tecnologias GPS (tais como, e meramente a título de exemplo, o GNSS multibanda utilizado em situações de arvoredo denso e pouco sinal de cobertura GPS).

A recolha dos pontos realizou-se através da prospeção nas próprias rotas e da pesquisa no Google Maps. Já no que respeita aos critérios utilizados para definir os pontos de interesse, cumpre referir que foram os seguintes:

- A relevância, ou seja, a verificação da importância e/ou história que justifique a visita ao ponto;
- A proximidade, visando sempre garantir-se que os pontos escolhidos não se encontravam muito longe do trajeto pretendido;
- A facilidade de acesso à rota, objetivando-se que os pontos de interesse se encontrassem em locais revestidos de comodidades de acesso e oferta de serviços (Wifi, Comida, Água, etc).

Nesta senda, implementou-se uma metodologia experimental, bem como a observação participante, onde se colocaram à prova todas as funcionalidades e capacidades da aplicação, com a sua devida documentação, para que, posteriormente, se fizesse a correção de eventuais bugs.

> The main component of the experimental method is subjecting the study objects to the influence of particular variables under known and controlled conditions in order to observe the effects of the variable on the subject In these instances, the observational method differs from the experimental method in just one way: in

> experiments, the scientist makes provisions for some event to occur in order to observe what follows, in contrast to studies by observation, which only record events as they occur or have already occurred (Gil, 2008).

Este estudo de caso, realizado em primeira mão, realizou-se conjuntamente com o proprietário do alojamento local A Casa do Zé. A participação deste não se circunscreveu só à análise das funcionalidades necessárias para a realização deste projeto (visto que o Sr. Nuno é praticante de ciclismo e, por isso, detém já trilhos demarcados da região), como também foi crucial na investigação e recolha de pontos de interesse dos trilhos suprarreferidos.

## Calendarização

*Tabela 1 - Calendarização das tarefas para o desenvolvimento da Aplicação (Martins, 2023, p1).*

| | |
|---|---|
| Outubro 2022 - Dezembro 2022: | Apresentação e aceitação do tema<br>Desenvolvimento da Aplicação<br>Desenvolvimento do Pré-projecto |
| Janeiro 2023 - Março 2023: | Desenvolvimento da Aplicação<br>Testes no terreno<br>Correção de bugs |
| Abril 2023 - Junho 2023: | Entrega do trabalho final: parte prática e parte teórica baseada no anexo do Projeto Académico, com a declaração do Orientador e a declaração de autoria do aluno. |

# Script da Aplicação

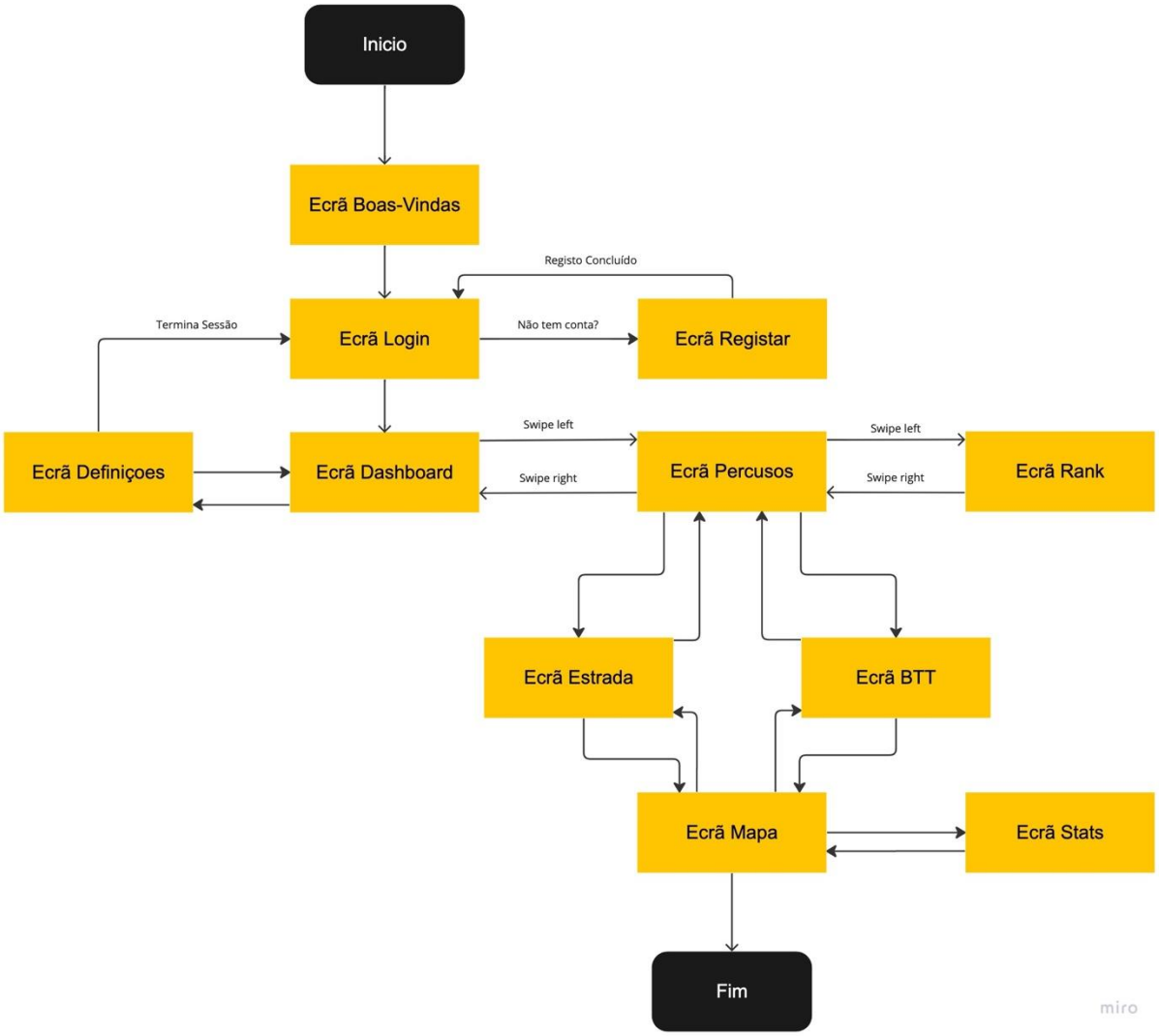


*Figura 2 - Script representativo da aplicação (Martins, 2023, p1).*

# Resultados e Discussão dos Resultados

## Idealização

Inicialmente, a aplicação de percursos com os ditos pontos de interesse seria apenas isso: uma aplicação onde se pudesse consultar as rotas de determinada região e da qual constassem os seus pontos turísticos, identificados num mapa georreferenciado (Figura 3). Sucede que, após uma análise mais detalhada, chegou-se à conclusão de que seria possível ir mais além e tentar explorar uma vertente mais interativa.

*Figura 3 - Idealização do Ecrã Mapa (Martins, 2023, p.1).*

Concluiu-se que seria relevante desenvolver uma abordagem na qual se introduzisse a funcionalidade de tracking na aplicação. Isto é, uma funcionalidade em que o utilizador conseguisse acompanhar a sua posição relativamente à rota, assim como a sua velocidade instantânea, velocidade média, tempo decorrido e a distância percorrida até ao momento. Ora, com a introdução desta nova funcionalidade (que permite obter a localização do utilizador em

tempo real), foi possível implementar o que viria a ser o segundo grande ponto de destaque desta aplicação: a visita guiada. Esta, tal como o nome indica, é uma funcionalidade que se destina à utilização de áudios para, não só anunciar a proximidade de determinado ponto de interesse, como principalmente, para expor, após a chegada do utilizador ao local, o contexto histórico daquele, os serviços disponíveis e outras informações relevantes sobre este dito ponto de interesse. Esta funcionalidade foi pensada no âmbito da utilização dos ciclistas, visto que não seria prático visualizar esta informação na própria aplicação durante a atividade. Por outro lado, e por uma questão de acessibilidade ou meramente preferencial, implementou-se, alternativamente, a utilização de notificações sonoras e vibratórias, para alertar o utilizador da proximidade a um ponto de interesse.

## Desenho, Projeção e Implementação

Inicialmente projetaram-se as funcionalidades mais básicas assim como os seus respetivos ecrãs, para isso utilizou-se o Figma, uma plataforma de desenho de interfaces, começou-se por criar uma página Boas-Vindas, onde o utilizador é saudado com uma mensagem explicativa da aplicação assim como um resumo da mesma (Figura 4), seguidamente é redirecionado para uma simples página de Login (Figura 5) onde poderá iniciar sessão ou com o seu email e password, com o Google ou Apple, ou se não tiver conta, criar uma através da página de registo (Figura 6).

*Figura 4 - Ecrã Boas-Vindas (Martins, 2023, p.1).*

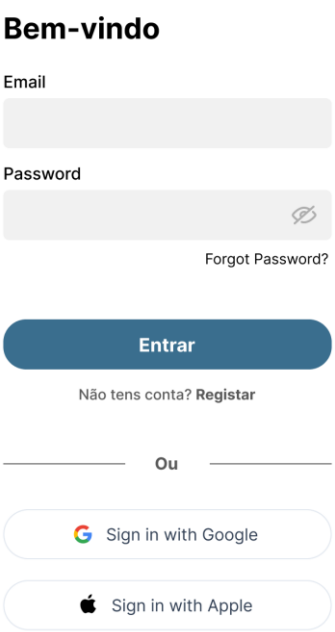


*Figura 5 - Ecrã Login (Martins, 2023, p.1).*

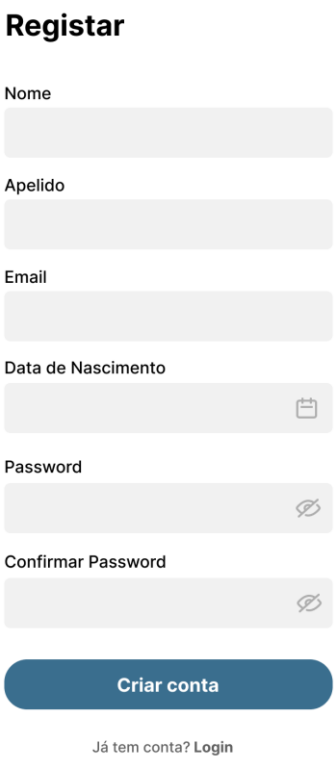


*Figura 6 - Ecrã Registar (Martins, 2023, p.1).*

Ora a realização desta funcionalidade implica a integração do Firebase Authentication (que trata de gerenciar todo o processo de autenticação de uma forma segura) e do Firebase Database (que trata de guardar toda a informação adicional sobre o utilizador) com o flutter, para isso basta criar um projeto no Firebase, escolher o Flutter como integração seguir os simples passos, associar a aplicação ao projeto, e começar a desenvolver o código (Figura 7).

```
RoundedButton(
  size: size,
  color: kColor6,
  text: 'Entrar',
  onPressed: () async {
    try {
      final existinguser = await _auth.signInWithEmailAndPassword(
        email: email,
        password: password,
      );
      if (existinguser.user != null) {
        Navigator.pushReplacement(context, CupertinoPageRoute(builder: ((context) => LoadingScreen())));
      }
    } on FirebaseAuthException catch (e) {
      switch (e.code) {
        case 'invalid-email':
          setState(() {
            errorcode = 'Email Inválido';
            _emailError = true;
          });
          print(e.code);
          break;
```

*Figura 7 - Exemplo do código do botão Entrar*
*(Martins, 2023, p.1).*

Este exemplo descreve como é contruído o botão Entrar assim como o que acontece quando é premido.

Este botão é renderizado através do widget Rounded Button, um widget personalizado, mas que herda grande parte da sua funcionalidade do Material Button, um widget pré-construído pela equipa do Flutter.

Este widget recebe 4 parâmetros, size, que é uma variável que foi determinada antes da execução do UI e recebe todos os parâmetros que se relacionem com tamanhos, rácios e definições do ecrã, este parâmetro no fundo vai determinar o tamanho do botão baseado no tamanho do ecrã, color, que é uma variável que define a cor do botão, e neste caso aponta para uma constante (kColor6), text, que define o texto presente no interior do botão, e por fim, o parâmetro onPressed, que recebe uma função e determina o comportamento da aplicação quando o botão é pressionado. Esta função é do tipo assíncrona pois a variável existinguser precisa de comunicar com o Firebase para efetuar o dito Login e, portanto, a aplicação tem de esperar que este passo seja concluído para que continue a execução do código. Após isso, se o existinguser existir de facto na base de dados, a aplicação continua para a página LoadingScreen (esta que se destina a carregar os dados do utilizador da base de dados), caso contrário, é determinado o erro: email inválido, password inválida, ou utilizador não encontrado e este é apresentado ao utilizador.

Seguidamente, após efetuar o login e as informações da base de dados forem carregadas é apresentado ao utilizador um dashboard (Figura 8) com as suas métricas acumuladas, assim como um menu inferior com acesso a duas outras páginas, Percursos e Ranking.

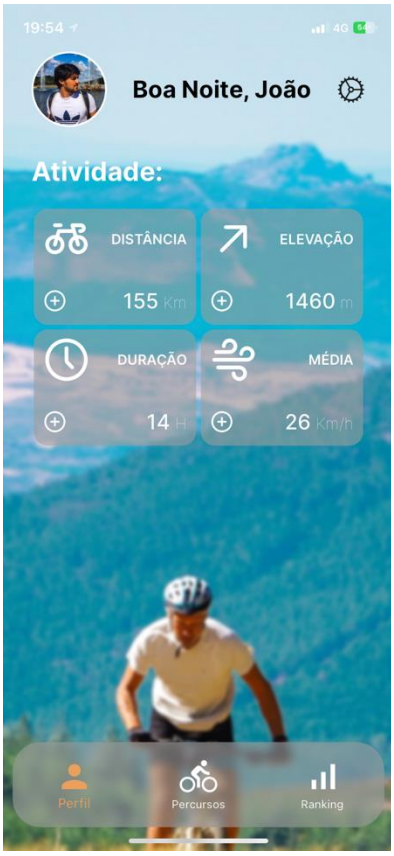


*Figura 8 - Ecrã Perfil (Martins, 2023, p.1).*

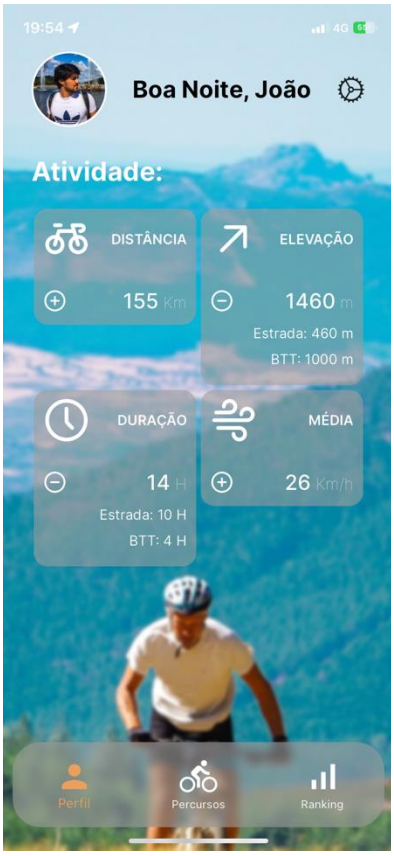


*Figura 9 - Ecrã Perfil (Expandido) (Martins, 2023, p.1).*

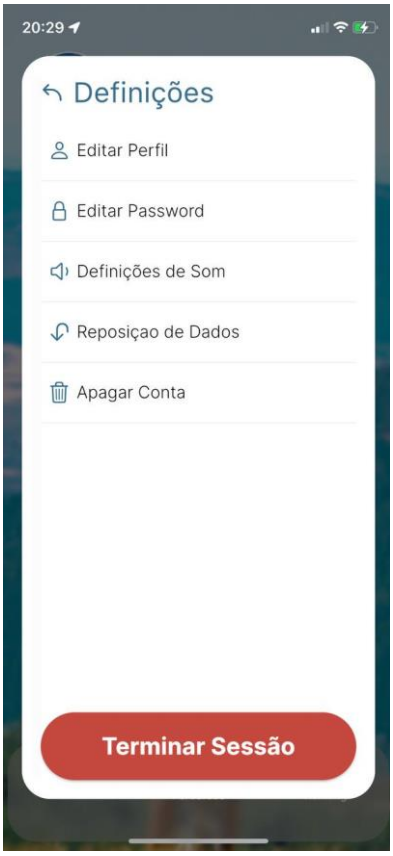


*Figura 10 - Ecrã Definições (Martins, 2023, p.1).*

A página Perfil/Dashboard (Figura 8) é o ponto de partida sempre que o utilizador abre a aplicação, aqui é possível visualizar todo um resumo referente á sua atividade neste local. Essa informação é disposta na forma de *cards*, os 4 cards presentes referem-se á distancia, elevação, duração e média acumuladas das duas atividades, ou seja, são medidas gerais referentes à soma da atividade acumulada de estrada com a atividade acumulada de btt, porém, é possível visualizar a atividade acumulada individual de cada modalidade simplesmente carregando no botao + que se encontra no canto inferior esquerdo de cada *card*, desta forma o *card* é expandido revelando as informações referentes ao acumulado das atividades individuais (Figura 9). Esta é a grande finalidade da página Perfil, mas para além disso, ainda conta com um botão de definições que quando premido, traz um novo menu (Figura 10) que permite ao

utilizador: alterar o seu perfil, editar a sua password, alterar definições de som, repor os dados da base de dados, apagar a sua conta, e por último, terminar sessão.

A página Percursos (Figura 11) apresenta dois grandes botões ilustrados com duas imagens elucidativas do tema em questão, uma aponta para os percursos de estrada (Figura 12) o outro para percursos de BTT (Figura 13).

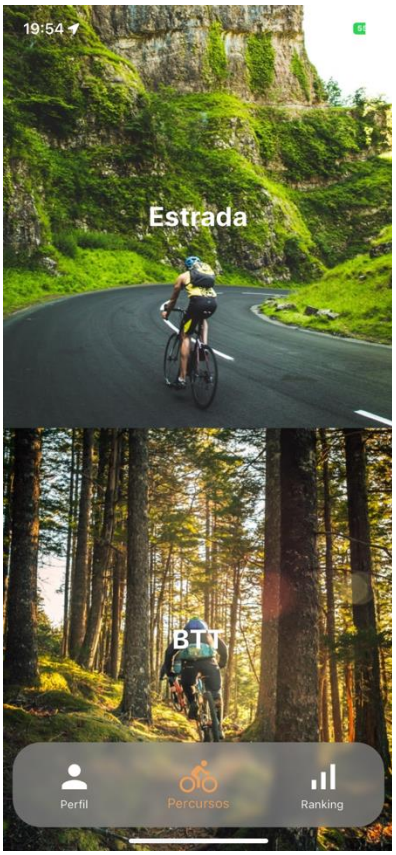


*Figura 11 - Ecrã Percursos (Martins, 2023, p.1).*

*Figura 12 - Ecrã Estrada (Martins, 2023, p.1).*

*Figura 13 - Ecrã BTT (Martins, 2023, p.1).*

Ambos estes botões levam o utilizador para ambientes muito semelhantes, à exceção da cor e do conteúdo, a diferença na cor foi pensada devidamente para dar ao utilizador uma experiência mais personalizada. Ambos os menus têm o intuito de listar os diferentes percursos catalogados até ao momento. Esta lista é composta por *cards, cards* esses que descrevem os pontos mais importantes de cada rota, uma imagem do local, o seu nome, o seu código, a sua distância, a sua elevação, e o seu nível, esta classificação referente à dificuldade tem como base um índice criado pela própria FPC (Federação Portuguesa de Ciclismo) (Figura 14).

3. Parâmetros e características por Grau de Dificuldade:

| VERDE | AZUL | VERMELHO | PRETO |
|---|---|---|---|
| Fácil | Moderado | Difícil | Muito Difícil |
| 4 a 6 pontos | 7 a 9 pontos | 10 a 12 pontos | 13 a 16 pontos |

4. Critérios de classificação.

A classificação destes percursos é determinada pela soma dos seguintes critérios:

- A distância do percurso
- O desnível positivo do percurso
- O tipo de piso do percurso
- Os aspetos técnicos do percurso

*Figura 14 - Parâmetros FPC*
*(Federação Portuguesa de Ciclismo, 2019, p. 17).*

A nível de programação este *card* é um widget desenvolvido inteiramente do zero, conta com alguns elementos pré-construídos, como ClipRRect, que basicamente encapsula qualquer elemento de UI atribuindo-lhe diferentes tipos de bordas, neste caso, bordas circulares na própria imagem, no canto superior esquerdo e direito, por sua vez, o TrackCard recebe o CachedNetworkImage, que tal como o nome indica, é um widget, que permite carregar imagens da web e guardá-las em cache para um acesso mais rápido e offline se for caso disso.

Como se pode constar, os parâmetros do Trackcard estão atribuídos a um document, este document refere-se a um snapshot da base de dados, o snapshot por sua vez é gerado por uma stream esta que atualiza a informação caso existir algum tipo de alteração na mesma, e é desta forma, que é possível obter informação da base de dados em tempo real. (Figura 15)

```
StreamBuilder(
  stream: FirebaseFirestore.instance.collection('tracks').where('id', isGreaterThanOrEqualTo: 'E').where('id', isLessThan: 'F').snapshots(),
  builder: ((context, snapshot) {
    if (!snapshot.hasData) return const Text('Loading...');
    return ListView.builder(
      scrollDirection: Axis.vertical,
      shrinkWrap: true,
      physics: const NeverScrollableScrollPhysics(),
      itemCount: snapshot.data!.docs.length,
      itemBuilder: ((context, index) {
        return _buildListItem(context, snapshot.data!.docs[index]);
```

```
Widget _buildListItem(BuildContext context, DocumentSnapshot document) {
  return ListTile(
    title: Container(
      padding: EdgeInsets.symmetric(horizontal: 10, vertical: 10),
      child: TrackCard(
        color: kColor6,
        trackid: document['id'],
        trackname: document['routename'],
        distance: document['distance'],
        difficulty: document['difficulty'],
        elevation: document['elevation'],
        image: CachedNetworkImage(
          fadeInDuration: Duration.zero,
          imageUrl: document['image'],
          height: 200,
          width: 400,
          fit: BoxFit.cover,
        ), // CachedNetworkImage
      ), // TrackCard
    ), // Container
  ); // ListTile
}
```

```
ClipRRect(
    borderRadius: BorderRadius.only(
      topLeft: Radius.circular(10),
      topRight: Radius.circular(10),
    ), // BorderRadius.only
    child: image), // ClipRRect
```

```
description: "description123"
difficulty: "Dificil"
distance: "59"
duration: "3"
elevation: "982"
id: "E02"
image: "https://firebasestorage.googleapis.com/v0/b/casadoze-271b2.appspot.com/o/Tracks%2FE02%2FE02.jpeg?alt=media&token=42af68c2-f3ee-4e56-be9c-1e08171a5ffa"
routename: "Rota do Côa"
```

*Figura 15 - Exemplo de código*
*(Martins, 2023, p.1).*

Após escolher uma rota, o ecrã seguinte destina-se a mostrar ao utilizador essa mesma rota num mapa georreferenciado com todos os seus pontos de interesse, assim como alguns pontos de água presentes ao longo do percurso. O API de georreferenciação escolhido para integrar este projeto foi o *openstreetmap* pois este é um mapa colaborativo. Ora, a grande vantagem desta funcionalidade é que mesmo para áreas rurais que não têm tanta atenção por parte das gigantes tecnológicas (Google Maps, Apple Maps, TomTom), qualquer utilizador da região consegue cartografar os seus trilhos.

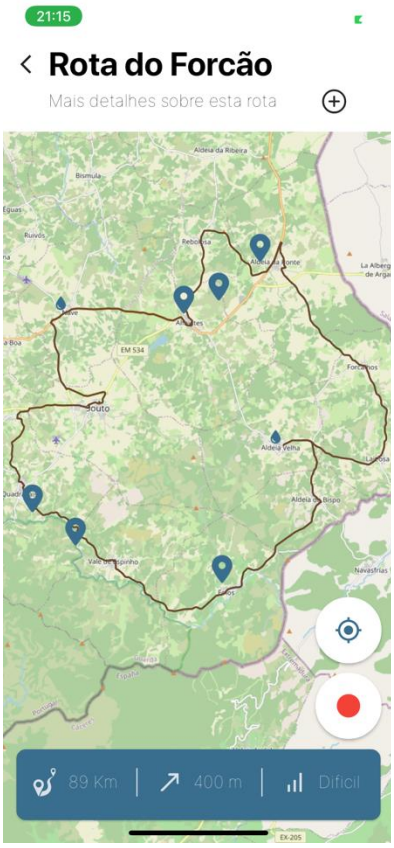


*Figura 16 - Ecrã Mapa (Martins, 2023, p.1).*

*Figura 17 - Pop-up Ponto de Interesse (Martins, 2023, p.1).*

*Figura 18 - Ecrã Ponto de Interesse (Martins, 2023, p.1).*

Neste ecrã (Figura 16) estão presentes um botão + que ao ser pressionado expande o cabeçalho revelando a história da rota, o dito mapa com o percurso da rota e os seus pontos de interesse, e por último no rodapé, informação dada posteriormente sobre a rota. No ecrã seguinte (Figura 17) é descrita a continuação da ação, ou seja, após o utilizador clicar sobre qualquer ponto de interesse, o mapa move-se definindo o mesmo como o seu centro e é apresentando um pop-up com informações superficiais sobre o ponto de interesse (uma pequena imagem, o nome, o desvio que é preciso fazer na rota para visitá-lo e uma pequena

descrição sobre o mesmo), caso o utilizador queira saber mais informações é possível carregar sobre Saber Mais para abrir uma página dedicada ao ponto de interesse (Figura 18), aqui é apresentada toda a informação que foi previamente exibida no pequeno pop-up mas de uma forma mais completa e detalhada, a pequena imagem sofre agora uma alteração para um carrossel onde é possibilitado ao utilizador fazer *swipe* para alternar entre as várias imagens presentes, a descrição sobre o local aparece toda na sua íntegra sendo possível fazer *scroll*, e por fim no rodapé estão representadas todas as comodidades presentes neste ponto de interesse.

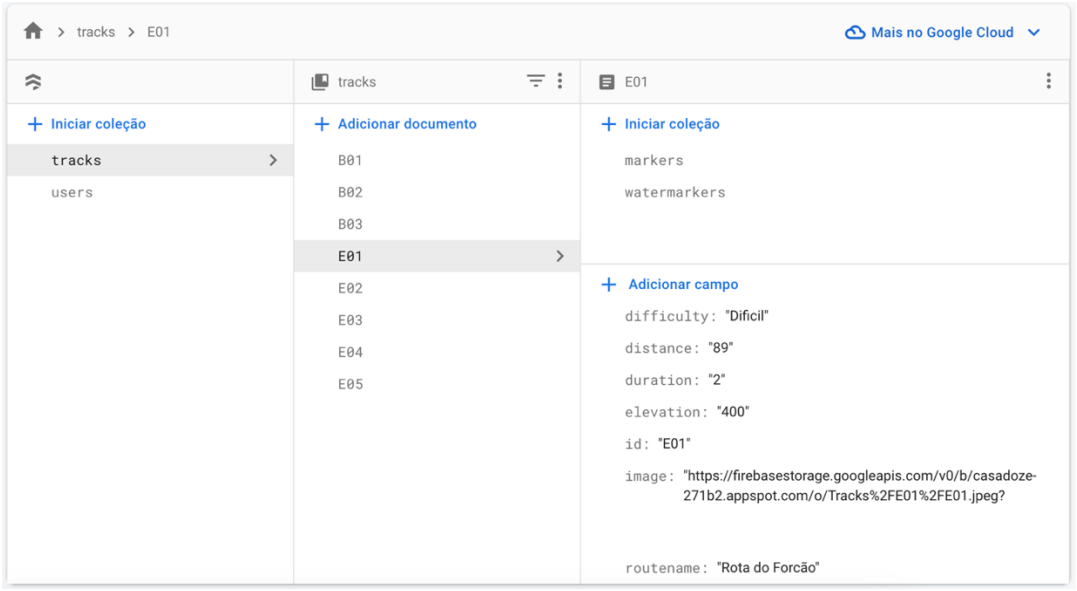


*Figura 19 - Firebase Database*
*(Martins, 2023, p.1).*

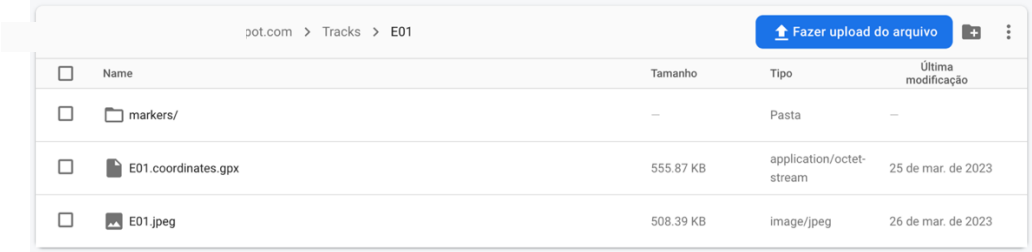


*Figura 20 - Firebase Storage*
*(Martins, 2023, p.1).*

Toda esta informação é carregada uma vez mais da Firebase Database (onde estão presentes as informações sobre a rota e os seus pontos de interesse) (Figura 19) e da Firebase Storage (onde estão presentes todas as imagens e ficheiros de GPS com as respetivas coordenadas) (Figura 20).

De seguida, depois de implementar estas funcionalidades mais básicas da aplicação, começou-se a desenvolver a vertente mais complexa da mesma. O primeiro passo seria

determinar a localização atual do utilizador, ora para isso, utilizou-se o package *geolocator*, uma biblioteca desenvolvida pela baseflow bastante completa nesta vertente. Com a utilização desta única biblioteca é possível determinar a localização atual do dispositivo, determinar a localização continua caso esta se altere, verificar se os serviços de localização do dispositivo estão ativos, e ainda, calcular a distância em metros entre duas coordenadas.

```
final positionStream = Geolocator.getPositionStream(locationSettings: LocationSettings(accuracy: LocationAccuracy.high)).asBroadcastStream();
```

*Figura 21 - Exemplo de código do Geolocator*
*(Martins, 2023, p.1).*

A implementação desta funcionalidade é possível com apenas uma linha de código (Figura 21). Nesta linha é possível definir várias propriedades através da *LocationSettings*, mas neste contexto só é relevante definir a *LocationAccuracy*, que neste caso deve ser *.high*.

```
//Gets current position
livelisten = positionStream.listen((position) {
  _position = position;
  notifyListeners();
});
```

*Figura 22 - Exemplo de código*
*(Martins, 2023, p.1).*

```
  ///When position changes
  recordinglisten = positionStream.listen((position) {
    speed = position.speed * 3.6;
    speedavglist.add(speed);
    speedavg = averageSpeed(speedavglist);
    distancelist.add(LatLng(position.latitude, position.longitude));
    distance = calcDistance(distancelist) * 0.001;
    globaldistance = distance;
    globalspeedavg = speedavg;
    notifyListeners();
  });
}
```

*Figura 23 - Exemplo de código*
*(Martins, 2023, p.1).*

A partir daqui, é possível adicionar serviços que usem *listeners* para ouvirem eventuais alterações na stream como é o caso do *livelisten* (Figura 22) e do *recordinglisten* (Figura 23), desta forma, sempre que a latitude e longitude do dispositivo alterarem, estas funções são atualizadas com os novos valores das mesmas sendo possível atualizar o UI do dispositivo, ou fazer cálculos matemáticos complexos como é o caso da função *calcDistance* (Figura 24), que usa a fórmula de *haversine* para calcular a distância entre dois pontos numa esfera.

```
double distanceBetween(
  double startLatitude,
  double startLongitude,
  double endLatitude,
  double endLongitude,
)
```

*package:geolocator/geolocator.dart*

Calculates the distance between the supplied coordinates in meters.

The distance between the coordinates is calculated using the Haversine formula (see https://en.wikipedia.org/wiki/Haversine_formula). The supplied coordinates [startLatitude], [startLongitude], [endLatitude] and [endLongitude] should be supplied in degrees.

*Figura 24 - Exemplo de código*
*(Martins, 2023, p.1)*

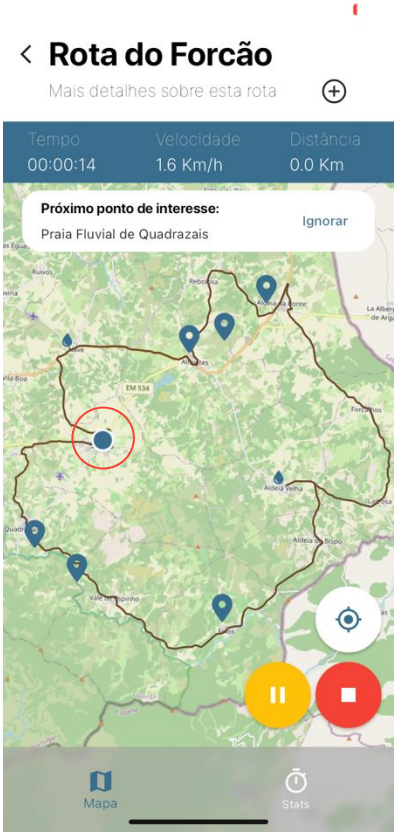


*Figura 25 - Ecrã Mapa (Recording)*
*(Martins, 2023, p.1).*

*Figura 26 - Ecrã Stats*
*(Martins, 2023, p.1).*

Ora todas estas funcionalidades estão inativas num primeiro momento, só após o utilizador pressionar um dos dois grandes botões presentes no ecrã do mapa (Figura 16) é que estas são acionadas. O primeiro botão dessa coluna aciona apenas o *livelisten*, com o objetivo de determinar a localização em tempo real do utilizador e mostrá-la no UI através de um pequeno ponto azul. O segundo botão para além de acionar esta funcionalidade, aciona também o *recordinglisten*, desta forma, a aplicação começa a gravar a atividade do utilizador, o UI por sua vez altera-se ligeiramente dando lugar a estas novas métricas (Figura 25) e é adicionado

um novo menu ao seu rodapé sendo possível alternar entre outro ecrã personalizado que mostra estas mesmas métricas, mas de uma forma mais simplificada e interativa (Figura 26). O botão que deu aso a toda esta transformação, também ele sofre uma metamorfose, originando dois botões no seu lugar, um de botão de *stop* e outro de *pause*, que tal como sugerem os seus próprios nomes servem para terminar a atividade e pausar a mesma, respetivamente.

Por fim, mas não menos importante, a última funcionalidade, e o grande foco desta aplicação, a Visita Guiada, foi a última peça a ser implementada devido à necessidade de toda esta estrutura estar já elaborada para que depois esta funcionalidade pudesse consumir os seus recursos. Esta funcionalidade tem como objetivo reproduzir alertas sonoros sobre o ponto em que o utilizador passa, determinou-se então, que um pouco antes de chegar ao local, mais precisamente a 500 metros, o utilizador ouve o seguinte alerta: A 500 metros, encontra-se o/a X local, sendo X, o nome desse ponto de interesse. Caso o utilizador faça o desvio para ir visitar esse ponto, a 40 metros do mesmo, é novamente ouvido outro alerta, mas desta vez, com uma descrição detalhada da área circundante assim como um contexto histórico, contrariamente, se o utilizador quiser ignorar este ponto, pode fazê-lo de forma manual, carregando sobre o botão ignorar (Figura 26), ou, e mais uma vez pensando numa abordagem *handsfree*, após andar 700 metros na direção contrária desse dito ponto a própria aplicação reconhece essa intenção e descarta por completo esse ponto passando para o próximo. Estes áudios foram gerados através de inteligência artificial e mais uma vez introduzidos na Firebase Storage.

## Bugs e Testes no Terreno

Ao longo do extenso desenvolvimento desta aplicação, ocorreram, como seria de esperar, vários bugs, desde problemas no UI, onde os elementos não se adaptavam a ecrãs mais pequenos, problemas na escrita e leitura da base de dados, até mesmo a simples erros de sintaxe, pois, ao longo destes oito meses, o próprio flutter sofreu vários updates, e com isso, algumas mudanças na sua própria sintaxe, mas, com a ajuda de recursos como o *Stackoverflow*, *Github* e outros programas semelhantes, foi possível ultrapassar todos estes obstáculos, conseguindo chegar a um protótipo funcional e consequentemente a uma versão alfa da aplicação. Apesar de, algumas funcionalidades projetadas, como é caso do login através do Google e Apple, ou de algumas opções no ecrã das definições não estarem para já funcionais devido a não serem uma prioridade e não ter havido tempo até à data de entrega deste projeto para as desenvolver, ainda assim, serão implementadas em futuras compilações.

Com a primeira versão alfa da aplicação pronta para ser testada no terreno, foi fundamental a deslocação ao concelho do Sabugal. Devido à minha ausência no local de estudo em questão, foi necessário usar terceiros como *testers*. O proprietário do alojamento local disponibilizou-se para este fim e, portanto, já no dispositivo do mesmo (iPhone SE 2020) foi-lhe disponibilizada a primeira versão alfa. A partir daqui, criou-se uma lista das funcionalidades necessárias a serem testadas no terreno, visto não ser possível testar as mesmas num ambiente de desenvolvimento.

Lista dos pontos a serem testados:

1. Depois de a aplicação obter a localização atual, determinar se essa localização corresponde à realidade.
2. Verificar se a linha a vermelho (que é a linha que segue o utilizador) se sobrepõe à linha castanha (linha do percurso).
3. Verificar se as métricas dadas pela aplicação (velocidade, distancia, etc.) são próximas da realidade. (Usar o GARMIN como meio de comparação).
4. Verificar se, bloqueando o dispositivo, a aplicação continua no background a gravar o percurso.
5. Usar uma aplicação que reproduza música (Ex: Spotify) e averiguar se, quando a aplicação reproduz um áudio, a música baixa automaticamente.
6. Averiguar se 500 metros é a distância ideal para avisar a proximidade do ponto de interesse.

7. Verificar se as duas vozes estão funcionais em cada ponto de interesse (a 500 metros e a 40 metros).
8. Testar uma vez o botão ignorar para se verificar se efetivamente a aplicação salta a leitura desse ponto de interesse.
9. Verificar o ecrã stats periodicamente para averiguar a sua veracidade.

Os resultados foram promissores, todos os pontos foram validados à exceção do ponto 6 e 7. No sexto ponto, o utilizador apontou que ouviu o sinal sonoro muito em cima do cruzamento onde é feito o desvio para o dito ponto de interesse. Já chegado ao local, e já em relação ao ponto 7, o utilizador relatou que não ouviu qualquer aviso sonoro referente a esse local. Desta forma concluímos que os dois raios definidos para acionarem os dois avisos sonoros, não são adequados.

# Conclusão

Com este projeto foi possível alcançar o resultado pretendido, o desenvolvimento de uma aplicação simplificada e dinâmica que facilitasse a prática de ciclismo juntamente com uma vertente turística em paralelo, desta forma, os futuros clientes que visitarem a Casa do Zé usufruirão de uma experiência inovadora e mais imersiva no que toca à prática de ciclismo na zona raiana.

A nova tecnologia desenvolvida pela Google, o Flutter, mostrou-se bastante capaz de acompanhar todas as funcionalidades que foram acrescentadas ao longo do desenvolvimento desta aplicação, de uma forma rápida devido a toda a sua panóplia de bibliotecas, e eficiente devido à sua linguagem poderosa, o Dart.

Este projeto, com o consequente desenvolvimento desta aplicação, surgiu mesmo após o término do curso: *The Complete Flutter Development Bootcamp with Dart*, curso esse fornecido pela plataforma Udemy, o que permitiu cimentar de uma forma bastante completa todas as matérias abordadas no suprarreferido curso, dada a complexidade da aplicação, dando lugar a um crescimento profissional muito positivo.

Futuramente, será explorada uma forma de globalizar a aplicação para dar a possibilidade a outras entidades turísticas de usufruírem deste serviço.

## Futuros Updates e Alterações

Nas próximas versões da aplicação serão implementadas várias alterações. O primeiro passo será incluir todas as funcionalidades descritas que até à data de entrega deste projeto não foram possíveis de concretizar, nomeadamente:

- Implementar o Login através de provedores de redes sociais (Google, Apple, Twitter).
- Implementar a confirmação de registo através de um email, funcionalidade que o próprio Firebase disponibiliza, para evitar ataques de DDOS.
- Terminar o terceiro ecrã Ranking onde o UI está praticamente acabado, faltando apenas implementar as funções na base de dados que devolvam à aplicação a informação do TOP 5 dos clientes com maior acumulado de quilómetros até ao momento, assim como, o TOP 5 de percursos mais frequentados.
- Implementar algumas definições em falta, tais como, Definições de Som, Reposição de Dados e Apagar Conta.
- Implementar os mapas em modo offline.
- Ajustar os raios em que os avisos sonoros são executados.

Após a implementação destes Updates irão surgir outros que foram pensados para dar uma melhor experiência ao utilizador, nomeadamente:

- Adicionar pequenas setas direcionais na linha da rota para um melhor entendimento do início e fim do respetivo percurso.
- Implementar no *recordinglisten*, um método que faça o UI acompanhar a posição do utilizador em relação ao mapa.
- Implementar a funcionalidade: pontos de paisagem, onde em determinado sítio voltado para determinado ponto cardeal o utilizador consiga observar uma paisagem deslumbrante.
- Criar um portal na Web especificamente para administradores, onde será possível adicionar, editar e remover novos percursos e os seus ditos pontos de interesse.

# Bibliografia


Andrei Frumusanu. (2020, November 10). *Apple Announces The Apple Silicon M1: Ditching x86 - What to Expect, Based on A14*. Anandtech.com; AnandTech. https://www.anandtech.com/show/16226/apple-silicon-m1-a14-deep-dive/4

Ahmad, A., Li, K., Feng, C., Asim, S. M., Yousif, A., & Ge, S. (2018). An Empirical Study of Investigating Mobile Applications Development Challenges. *IEEE Access*, *6*, 17711–17728. https://doi.org/10.1109/access.2018.2818724

*Build Documentation | Firebase Documentation*. (n.d.). Firebase. https://firebase.google.com/docs/build

Chatterjee, Nilanjan & Chakraborty, Souvik & Decosta, Aakash & Nath, Asoke. (2018). *Real-time Communication Application Based on Android Using Google Firebase.* IJARCSMS. 6.

Couture, J. (2020). Reflections from the "Strava-sphere": Kudos, community, and (self-) surveillance on a social network for athletes. *Qualitative Research in Sport, Exercise and Health*, 1–17. https://doi.org/10.1080/2159676x.2020.1836514

Federação Portuguesa de Ciclismo. (2019, May 7). *Regulamento de Homologação Percursos Cicláveis e Centros "Cyclin' Portugal."* https://www.cyclinportugal.pt/images/downloads/CyclinPortugal_Regulamento_2019.pdf

Gil, A. C. (2008). *Métodos e Técnicas de Pesquisa Social*. https://ayanrafael.files.wordpress.com/2011/08/gil-a-c-mc3a9todos-e-tc3a9cnicas-de-pesquisa-social.pdf

Martins, J. (2023). Diário de Investigação. Observação de Campo. (Não Impresso). Pp.1.

Payne, R. (2019). *Beginning app development with Flutter : create cross-platform mobile apps*. Apress.

Sahatqija, K., Ajdari, J., Zenuni, X., Raufi, B., & Ismaili, F. (2018). Comparison between relational and NOSQL databases. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. https://doi.org/10.23919/mipro.2018.8400041

Strava. (n.d.). *About Us*. www.strava.com. Retrieved June 30, 2023, from https://strava.com/about

Valente, C., Rui, D., Bértolo, J., Madeira, L., Presidente, C., Doutor, J., Jorge, C., Colaço, P., Vogal, C., Nuno, D., Da, M., Januário, S., Filipe, C., & (2018). *Dissertação elaborada com vista à obtenção do grau de Mestre em Gestão do Desporto*. https://www.repository.utl.pt/bitstream/10400.5/18246/1/2018_MGD_Cristiano%20Valente.pdf