



**INSTITUTO POLITÉCNICO DE LEIRIA**

# CICLODIAS

---

RELATÓRIO DE PROJETO EM SISTEMAS DE  
INFORMAÇÃO

GABRIEL SILVA N° 2201133  
IURI CARRASQUEIRO N° 2201127

PROGRAMAÇÃO DE SISTEMAS DE  
INFORMAÇÃO

02/2022

Relatório de Projeto em Sistemas de Informação para cumprimento dos requisitos necessários à realização da prova de apresentação de projeto do Curso Técnico Superior Profissional (TeSP) de **Programação de Sistemas de Informação** realizado sob a orientação de **Marco Vicente**

## DECLARAÇÃO

Declaro que este Relatório se encontra em condições de ser apreciada (o) pelo júri a designar.

O estudante 1,

---

Torres Vedras, .... de ..... de .....

Relatório de Projeto em Sistemas de Informação para cumprimento dos requisitos necessários à realização da prova de apresentação de projeto do Curso Técnico Superior Profissional (TeSP) de **Programação de Sistemas de Informação** realizado sob a orientação de **Marco Vicente**

## DECLARAÇÃO

Declaro que este Relatório se encontra em condições de ser apreciada (o) pelo júri a designar.

O estudante 2,

---

Torres Vedras, .... de ..... de .....

## **Agradecimentos**

Queremos agradecer ao Ricardo Dias pela ideia original e nome do projeto, Fábio Gomes pela criação do logo, Diogo Martins e Rafael Bento por testarem a aplicação.

# Resumo

[Ciclodias]

[Gabriel Silva e Iuri Carrasqueiro]

O Ciclodias é um projeto que tem como objetivo principal permitir a monitorização de percursos de treino com foco nos utilizadores de ciclismo, gravando a rota do percurso percorrido, utilizando a API MapBox, e alguns dados de treino que poderão ser relevantes para o utilizador, utilizando a localização GPS.

O projeto é dividido em duas partes. Na primeira temos a aplicação Android, onde o utilizador realiza as suas sessões de treino, pode visualizar, também, as anteriores em detalhe, fazer a gestão do seu perfil e criar publicações no feed de notícias.

Na segunda parte, temos o Website. Aqui o utilizador pode gerir o seu perfil e visualizar os seus treinos com mais detalhe, com a inclusão de um gráfico com a variação da velocidade instantânea. Além disso, o Website também inclui todo o sistema de feed de notícias, ou seja, o utilizador pode publicar as suas sessões de treino, para outros utilizadores visualizarem e interagirem, e pode ainda reagir e comentar publicações feitas por outros utilizadores.

Para os administradores temos a *backend* do Website, sendo a página principal um *dashboard* com os dados mais relevantes do projeto, por exemplo, o número de sessões de treino realizadas por todos os utilizadores da aplicação. Este inclui outro separador, onde o administrador pode gerir todos os utilizadores da aplicação e alterar informações ou cargos destes.

PALAVRAS-CHAVE: Ciclodias, ciclismo, Mapbox

# Índices

## Índice

1.	Introdução .....	2
2.	Metodologia .....	3
3.	Arquitetura do Sistema.....	4
4.	Gestão do Projeto .....	5
4.1.	Jira.....	9
5.	Análise.....	11
5.1.	API .....	15
6.	Desenho.....	18
6.1.	Modelo de Dados.....	22
7.	Implementação .....	23
8.	Testes.....	25
9.	Conclusão e trabalho futuro .....	26
10.	Bibliografia.....	27

## Índice de figuras

Figura 1 - Número e percentagem de commits no repositório do Android.....	5
Figura 2 – Número e percentagem de commits no repositório do Website/API.....	5
Figura 3 – Issues Sprint 1.....	9
Figura 4 – Burndown Chart Sprint 1.....	9
Figura 5 – Issues Sprint 2.....	9
Figura 6 – Burndown Chart Sprint 2.....	10
Figura 7 – Issues Sprint 3.....	10
Figura 8 – Burndown Chart Sprint 3.....	10
Figura 9 - Mockup de Registo do utilizador.....	18
Figura 10 – Mockup de Login do utilizador .....	18
Figura 11 – Mockup da página principal da app .....	19
Figura 12 – Mockup do perfil de utilizador .....	19
Figura 13 – Mockup da página para iniciar o treino .....	20
Figura 14 – Mockup da página de treino .....	20
Figura 15 - Mockup da página do mapa durante o treino .....	21
Figura 16 - Mockup da página de pausa .....	21
Figura 17 - Mockup da página de resumo do treino.....	21
Figura 18 - Mockup da página de definições .....	21
Figura 19 – Modelo de dados .....	22





[Esta é sempre a página 1.]

# **1. Introdução**

No âmbito da unidade curricular de Projeto em Sistemas de Informação, este documento serve para contextualizar o projeto, explicar a metodologia escolhida e a arquitetura do sistema, tanto do website, como da aplicação Android. Ainda explicamos como foi feita a gestão do projeto, ou seja, como foram divididas as tarefas entre os membros do grupo, e apresentamos os desenhos iniciais do projeto.

De um modo mais extensivo, damos a conhecer a análise do projeto, onde mostramos todas as funcionalidades implementadas, também os requisitos funcionais gerais e específicos da aplicação Android e do Website, e damos a conhecer a implementação do mesmo.

Por fim, indicamos os testes realizados no projeto, tanto realizados por uma Framework específica, como gerais.

## 2. Metodologia

Para o desenvolvimento do Projeto em Sistemas de Informação, foi adotada a metodologia ágil, pela fácil divisão de tarefas e a adaptabilidade em adicionar novas funcionalidades.

O projeto foi dividido em duas partes, uma que inclui o Website e a API, ou seja, todo o código envolvendo a Yii Framework, e uma que inclui o programa Android, por outras palavras, todo o código envolvendo a programação Android. Cada um destes contém o seu próprio repositório Github e respetivos *branches*.

A comunicação e organização de tarefas entre os membros do grupo no início do projeto foi feita pela ferramenta Jira, onde elaboramos Sprints com as tarefas por realizar naquele momento. Mas, devido ao grupo consistir de apenas 2 elementos, o uso desta ferramenta tornou-se irrelevante e não demonstrou nenhuma utilidade para o mesmo, portanto, começou-se a organizar diariamente as tarefas, durante as aulas presencialmente e em casa por videochamada.

### 3. Arquitetura do Sistema

Para a realização do projeto foram utilizadas as seguintes tecnologias:

Website:

- *Framework* Yii2 para a manipulação e apresentação de dados no Website;
- *Framework* Yii2-adminlte3 para a interface da *Backend* do Website;
- *Framework* Codeception para a realização de testes unitários, funcionais e de aceitação;
- Biblioteca Mapbox em Javascript para apresentação da rota de treino dos percursos realizados pelos utilizadores da aplicação;
- Biblioteca ApexCharts para a apresentação do gráfico da velocidade instantânea dos treinos do utilizador;

Android:

- API Mapbox em Java para mostrar a localização do utilizador num mapa e o desenho do percurso feito pelo utilizador no mesmo;
- Biblioteca Volley para a comunicação com a API;
- Biblioteca EclipsePahoMqttJava para apresentação notificações de interações nas publicações do utilizador no *feed* de Notícias e de alterações na *Leaderboard* do Website;

## 4. Gestão do Projeto

A gestão do projeto dividiu-se nas duas vertentes explicadas anteriormente, ou seja, na aplicação Android e no Website/API. Ambos os membros do grupo trabalharam nas duas vertentes, mas com o Gabriel a ter mais foco na aplicação Android e o Iuri a ter mais foco no Website e API.

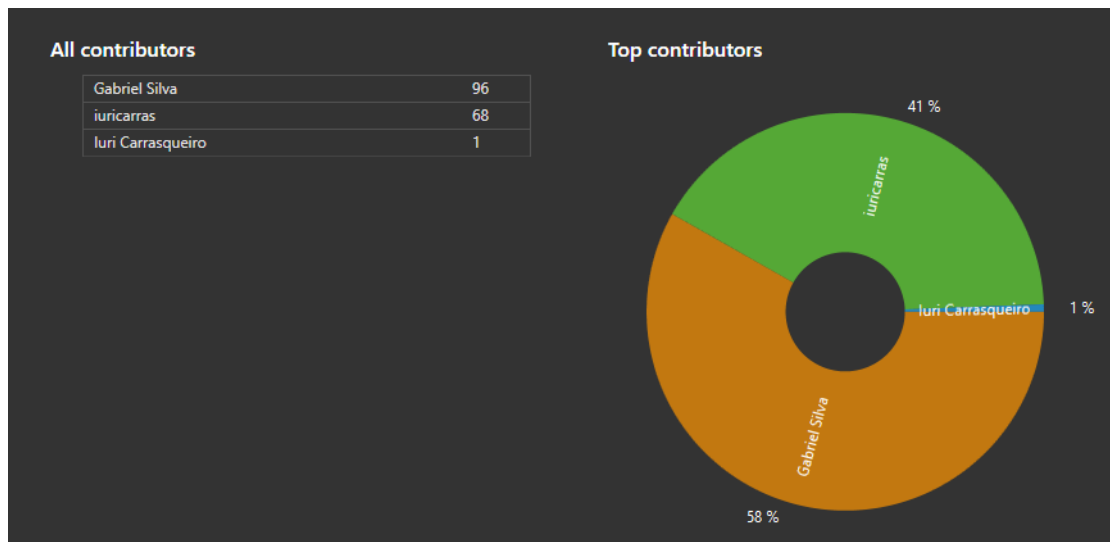


Figura 1 - Número e percentagem de commits no repositório do Android

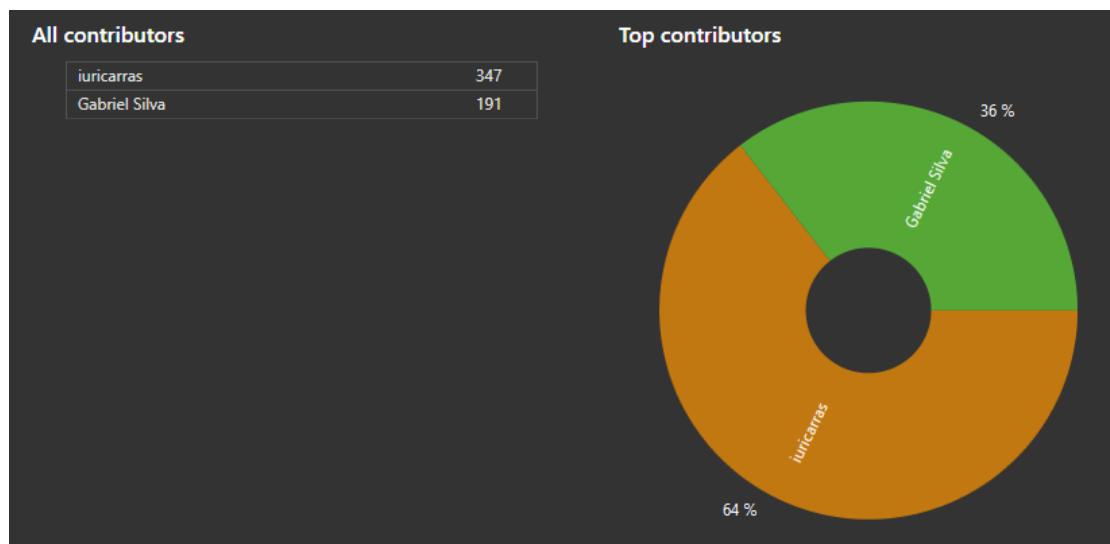


Figura 2 – Número e percentagem de commits no repositório do Website/API

Em termos das tarefas principais, divisão e calendarização das mesmas, fizemos a seguinte enumeração:

**Android:**

- T1** – Pesquisa de ideias para o projeto, e criação do repositório;
  - T2** – Realização dos *Mockups* para Android;
  - T3** – Criação dos *Layouts* do projeto;
  - T4** – Implementação da API do mapa;
  - T5** – Código para login e registo do utilizador;
  - T6** – Código para visualização e edição do perfil;
  - T7** - Código para a sessão de treino (Inicio, Em Progresso, Em Progresso (Mapa), Pausa, Resultados);
  - T8** – Código para visualização do mapa em um fragmento;
  - T9** – Código da MainPage para a visualização dos percursos mais recentes realizados pelo utilizador;
  - T10** – Comunicação com a API (Login e Registo);
  - T11** – Desenvolvimento de todas funções para comunicação com a API (Criar Treinos, Editar Perfil, etc....);
  - T12** – Fase de testes;
  - T13** – Correção de erros;
  - T14** – Implementação de funcionalidades baseadas em *messaging* no projeto para as Notificações;
  - T15** - Escrita do relatório do projeto e correção de erros;
- 
- T1** – 1 semana – ambos os membros;
  - T2** – 2 dias – ambos os membros;
  - T3** – 1 semana – Gabriel Silva;

**T4** – 2 semanas – Iuri Carrasqueiro;  
**T5** – 1 semana – Gabriel Silva;  
**T6** – 1 semana – Gabriel Silva;  
**T7** – 2 semanas – Iuri Carrasqueiro;  
**T8** – 1 semana – Iuri Carrasqueiro;  
**T9** – 1 semana – Gabriel Silva;  
**T10** – 1 semanas – ambos os membros;  
**T11** – 3 semanas - ambos os membros;  
**T12** – 1 semanas - ambos os membros;  
**T13** – 3 semanas - ambos os membros;  
**T14** – 1 semana – Iuri Carrasqueiro;  
**T15** – 1 semana – ambos os membros;

**Website:**

**T1** – Pesquisa de ideias para o projeto e criação do repositório;  
**T2** - Realização da página principal da *Frontend*;  
**T3** - Desenvolvimento do *dashboard* da *Backend*;  
**T4** - Criação da página de perfil do utilizador na *Frontend*;  
**T5** - Criação do histórico das sessões de treino e detalhes do mesmo;  
**T6** - Desenvolvimento do sistema de *feed* de notícias;  
**T7** - Realização dos testes unitários, funcionais e de aceitação;  
**T8** - Escrita do relatório do projeto e correção de erros;

**T1** – 1 semana – ambos os membros;  
**T2** – 1 semana – Iuri Carrasqueiro;  
**T3** – 2 semanas – Gabriel Silva;  
**T4** – 1 semana – Iuri Carrasqueiro;

**T5** – 2 semanas – Iuri Carrasqueiro;

**T6** – 6 semanas – Iuri Carrasqueiro;

**T7** – 2 semanas – Gabriel Silva;

**T8** – 1 semana – ambos os membros;

#### **API:**

**T1** – Pesquisa de ideias para o projeto;

**T2** - Criação dos métodos do utilizador e das sessões de treino;

**T3** - Criação dos métodos do *feed* de notícias;

**T4** - Implementação de funcionalidades baseadas em *messaging*

**T5** - Escrita do relatório do projeto e correção de erros;

**T1** – 1 semana – ambos os membros;

**T2** – 2 semana – Gabriel Silva;

**T3** – 2 semanas – Iuri Carrasqueiro;

**T4** – 1 semana – Iuri Carrasqueiro;

**T5** – 1 semanas – ambos os membros;



## 4.1. JIRA

Como indicado anteriormente, a comunicação e organização de tarefas foram, inicialmente, feitas utilizando a ferramenta Jira, onde se planeou e se executou 3 Sprints, descritos nas seguintes figuras:

### Sprint 1:

Completed issues							<a href="#">View in issue navigator</a>
Key	Summary	Issue type	Epic	Status	Assignee	Story points	
CP-1	Criação Repositorio Github	Story		DONE	65%	1	
CP-2	Desenvolvimento do relatório da proposta de projeto	Story		DONE	65%	2	
CP-3	Definição das ideias iniciais e futuras para o projeto	Story		DONE		4	
CP-4	Criação de User Stories	Story		DONE		2	
CP-5	Criação de Mockups	Story		DONE		4	

Figura 3 – Issues Sprint 1



Figura 4 – Burndown Chart Sprint 1

### Sprint 2:

Incomplete issues							<a href="#">View in issue navigator</a>
Key	Summary	Issue type	Epic	Status	Assignee	Story points	
CP-7	Desenvolvimento páginas Web - front-office (Web)	Story		DONE		4	
CP-11	Desenvolvimento páginas Web - back-office (Web)	Story		DONE		4	
CP-10	Relatório de especificação dos requisitos funcionais (Web)	Story		DONE		3	

Completed issues							<a href="#">View in issue navigator</a>
Key	Summary	Issue type	Epic	Status	Assignee	Story points	
CP-6	Desenvolvimento das Activities (Android)	Story		DONE	65%	4	
CP-13	Desenvolvimento DB	Story		DONE		2	

Figura 5 – Issues Sprint 2



Figura 6 – Burndown Chart Sprint 2

### Sprint 3:

Completed issues [View in issue navigator](#)

Key	Summary	Issue type	Epic	Status	Assignee	Story points
CP-8	Ligações entre Activities (Android)	Story		DONE		2
CP-12	Instalar e configurar sistema RBAC (Web)	Story		DONE		3
CP-9	Ligações entre páginas (Web)	Story		DONE		2
CP-11	Desenvolvimento páginas Web - back-office (Web)	Story		DONE		4
CP-7	Desenvolvimento páginas Web - front-office (Web)	Story		DONE		4
CP-10	Relatório de especificação dos requisitos funcionais (Web)	Story		DONE		3
CP-14	Relatório AMSI	Story		DONE		3

Figura 7 – Issues Sprint 3

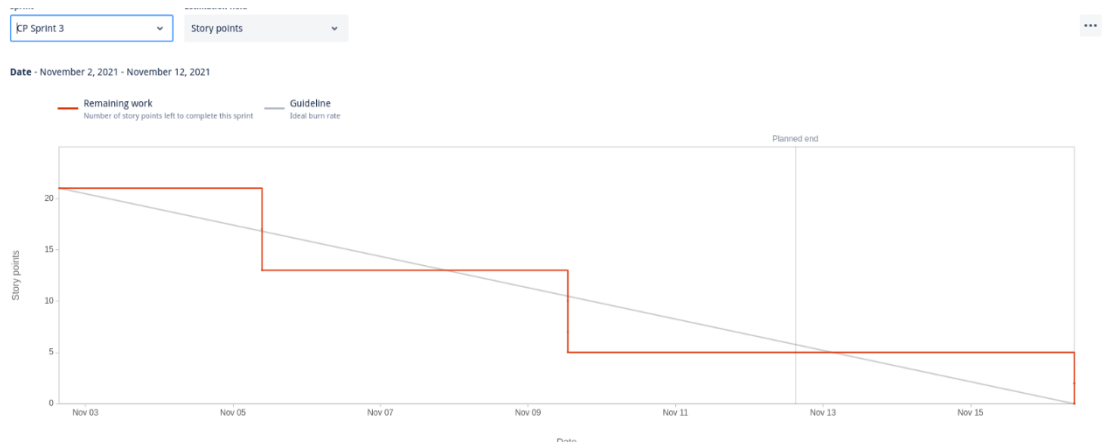


Figura 8 – Burndown Chart Sprint 3

## 5. Análise

O projeto Ciclodias permite o monitoramento de exercício físico do utilizador, guardando informações dos treinos realizados, assim como a sua rota percorrida, permitindo a criação de publicações para todos os utilizadores poderem visualizar e interagir com essas.

Os objetivos principais deste projeto são:

- Permitir o utilizador criar, entrar e gerir a sua conta o mais eficientemente possível;
- Permitir o utilizador visualizar todos os seus treinos detalhadamente;
- Permitir ao utilizador realizar um treino tanto em modo online como modo offline, fazendo a sincronização dos treinos assim que houver Internet;
- Permitir a aplicação ser executada em segundo plano;
- Permitir a gestão das sessões de treino do utilizador;
- Permitir a criação de publicações no *feed* de notícias, a partir de um treino do utilizador;
- Permitir a visualização e interação de qualquer utilizador com sessão iniciada nas publicações presentes no *feed* de notícias;
- Receber notificações na aplicação Android, providenciadas pelo Website;

#### Requisitos do projeto Geral:

- Permitir a criação de uma conta no projeto;
- Permitir iniciar e terminar sessão o mais eficientemente possível;
- Permitir ao utilizador editar o primeiro e último nome assim como a data de nascimento do seu perfil;
- Permitir ao utilizador apagar o seu perfil. Quando se realiza esta operação todos os dados da conta são apagados, assim como os treinos realizados pelo utilizador, publicações feitas pelo mesmo, e quaisquer interações feitas pelo mesmo em publicações de outros utilizadores;
- Apresentar ao utilizador todos os treinos que este realizou com os dados mais relevantes;
- Permitir ao utilizador visualizar os detalhes pormenorizados de um treino;
- Permitir ao utilizador apagar um treino a partir dos detalhes do mesmo. Neste caso, é verificado se existe alguma publicação criada para esse treino. Caso exista, todas as interações feitas para a mesma são apagadas para depois poder-se apagar a própria publicação e o treino, sem quaisquer erros e sem deixar dados desnecessários na API. Caso não exista nenhuma publicação, o treino é simplesmente apagado;
- Permitir ao utilizador criar uma publicação no *feed* de notícias a partir dos detalhes de um treino;

### Requisitos da aplicação Android:

- Carregar da API para as SharedPreferences os dados do utilizador que fez login, incluindo o seu *token* para pedidos da API, permitindo assim fazer automaticamente login na aplicação sem que se tenha de cada vez que se saia da aplicação inserir os dados novamente;
- Carregar da API todos os treinos realizados pelo utilizador que se iniciou sessão para a Base de dados local, para permitir a visualização dos mesmos em modo offline;
- Terminar a sessão do utilizador na aplicação, com isso, eliminar todos os dados na base de dados local e SharedPreferences, deixando a aplicação sem dados inutilizados;
- Permitir a obtenção da localização GPS do utilizador para realizar sessões de treino na aplicação. É pedido ao utilizador estas permissões na primeira vez que este entra na *activity* para iniciar um treino, ou para visualizar os detalhes de um. Caso o utilizador não o permita, este é redirecionado para a página principal, isto acontece até que este permita a obtenção da localização. Depois de permitido, estas permissões não são mais pedidas ao utilizador;
- Permitir ao utilizador realizar treinos em modo offline, guardando-os na Base de dados local, fazendo-se a sincronização com a API assim que houver Internet e quando se navegar para a página principal da aplicação;
- Permitir ao utilizador iniciar uma sessão de treino;
- Permitir a aplicação ser executada em segundo plano durante o treino;
- Apresentar ao utilizador no menu principal durante o treino, os dados mais relevantes deste, de modo a permitir a fácil visualização dos mesmos;
- Permitir a visualização do mapa durante o treino;
- Permitir colocar o treino em pausa e retomar quando o utilizador pretender;
- Permitir ao utilizador terminar o treino;
- Permitir ao utilizador rever a sua sessão de treino, com a rota desenhada no mapa e permitir este escolher um nome para a sessão. caso a pretenda guardar. Este também pode sair sem guardar a sessão;
- Guardar na API as alterações feitas aos treinos e utilizador;
- Receber notificações providenciadas pelo Website;

- Utilizar caixas de confirmação em todas as operações que possam ter algum impacto na aplicação, para que o utilizador não realize operações indesejadas;

#### Requisitos do Website *Frontend*:

- Permitir a visualização do menu principal com a *Leaderboard* do website, a qualquer utilizador independentemente que tenha login feito ou não;
- Bloquear o acesso a todas as páginas exceto a página principal, login e registo a utilizadores sem login feito;
- Bloquear o acesso a administradores de fazer login.
- Bloquear o acesso a utilizadores a páginas para moderadores;
- Permitir a visualização de publicações no *feed* de notícias a qualquer utilizador com login feito;
- Permitir a interação dos utilizadores nas publicações de qualquer utilizador (reagir à publicação ou escrever comentários);
- Permitir apagar uma publicação que foi criada pelo mesmo utilizador a partir do *feed* de notícias;
- Permitir ao utilizador visualizar apenas as suas publicações, para as poder gerir facilmente;
- Permitir ao moderador aceder a uma página para gerir todos os comentários das publicações;

#### Requisitos do Website *Backend*:

- Permitir o acesso a esta parte do Website apenas a administradores, outros utilizadores não têm acesso;
- Visualização de um *dashboard* com informações gerais do projeto (Ex: Número de sessões de treino);
- Permitir a visualização e fazer a gestão de todos os utilizadores e alterar os seus cargos rapidamente;

## 5.1. API

Em termos de API, está a ser utilizado os seguintes endereços:

### 5.1.1. Ciclismo

- **Criar Treino** - POST <http://ciclodias.duckdns.org/admin/v1/ciclismo> - Método CRUD responsável por adicionar uma nova sessão de treino do utilizador à base de dados;  
Parâmetros:
  - String “nome\_percurso”;
  - int “duracao”;
  - int “distancia”;
  - float “velocidade\_media”;
  - float “velocidade\_maxima”;
  - JSONArray “velocidade\_grafico”;
  - String “rota”;
- **Ver Treinos** - GET <http://ciclodias.duckdns.org/admin/v1/ciclismo> - Método CRUD que devolve todos os treinos realizados pelo utilizador;
- **Alterar Treino** - PUT <http://ciclodias.duckdns.org/admin/v1/ciclismo/x> - Método CRUD que altera uma sessão de treino, a pedido do utilizador. Este método também contém verificação de se o treino pertence ao utilizador;  
Parâmetros:
  - String “nome\_percurso”;
- **Apagar Treino** - DELETE <http://ciclodias.duckdns.org/admin/v1/ciclismo/x> - Método CRUD que elimina um treino da base de dados, a pedido do utilizador. Também este contém verificação;

- **Sincronizar Treinos** - POST

<http://ciclodias.duckdns.org/admin/v1/ciclismo/sync> - Método Custom que recebe um array de sessões de treino, adiciona-as na base de dados e devolve todos os treinos realizados pelo utilizador;

Parâmetros:

- JSONArray de treinos (Object Ciclismo);

### 5.1.2. User

- **Registar Utilizador** - POST

<http://ciclodias.duckdns.org/admin/v1/registo/signup> - Método Custom que regista um novo utilizador na base de dados;

Parâmetros:

- String “username”;
- String “email”;
- String “password”;
- String “primeiro\_nome”;
- String “ultimo\_nome”;

- **Login Utilizador** - POST

<http://ciclodias.duckdns.org/admin/v1/login/login> - Método Custom que faz o login de um utilizador, devolvendo o token de acesso para futuros pedidos API e informação sobre o perfil;

Parâmetros:

- String “username”;
- String “password”;

- **Ver Perfil** - GET <http://ciclodias.duckdns.org/admin/v1/user/x> - Método CRUD que devolve informações sobre o perfil do um utilizador, a pedido do mesmo. Este método contém verificação de se o perfil pertence ao utilizador;



- **Alterar Perfil** - PUT <http://ciclodias.duckdns.org/admin/v1/user/x> - Método CRUD que altera as informações de perfil de um utilizador, a pedido do mesmo. Este método também contém verificação de se o perfil pertence ao utilizador;

Parâmetros:

- String “primeiro\_nome”;
- String “ultimo\_nome”;
- String “data\_nascimento”;

- **Apagar Perfil** - DELETE <http://ciclodias.duckdns.org/admin/v1/user/x> - Método CRUD que apaga um utilizador, juntamente com todos os seus treinos e publicações, a pedido do utilizador. Este método também contém verificação;

### 5.1.3. Publicação

- **Criar Publicação** - POST <http://ciclodias.duckdns.org/admin/v1/publicacao> - Método CRUD que adiciona uma nova publicação à base de dados, a partir de um treino. Este método contém verificação de se o treino que irá ser publicado é do utilizador;

Parâmetros:

- int “ciclismo\_id”;

## 6. Desenho



The mockup for the registration screen features a back arrow in the top left corner. Below it is a square icon with an 'X' inside. The title 'Registo' is centered. The form includes two input fields for 'Primeiro Nome' and 'Último Nome', followed by two 'Email' fields, a 'Password' field, and a 'Confirmar Password' field. A 'Registrar-se' button is at the bottom.

←



Registo

Primeiro Nome Último Nome

Email

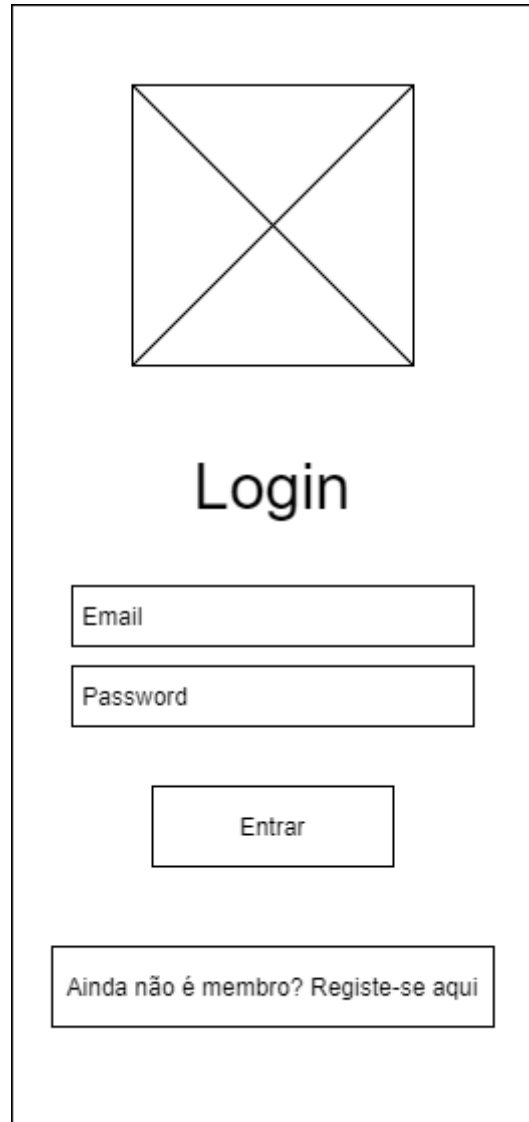
Email

Password

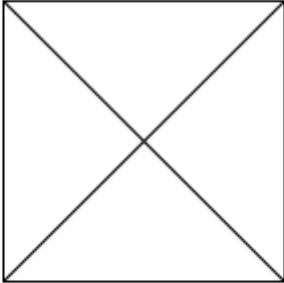
Confirmar Password

Registrar-se

Figura 9 - Mockup de Registo do utilizador



The mockup for the login screen features a square icon with an 'X' inside. The title 'Login' is centered. The form includes an 'Email' field and a 'Password' field. An 'Entrar' button is below the fields. At the bottom, there is a link that says 'Ainda não é membro? Registe-se aqui'.



Login

Email

Password

Entrar

Ainda não é membro? Registe-se aqui

Figura 10 – Mockup de Login do utilizador

Home


Atividades Recentes

Distância
Dia/ Mês

Duração
Vel Média

Distância
Dia/ Mês

Duração
Vel Média


Distância
Dia/ Mês


Duração
Vel Média








Figura 11 – Mockup da página principal da app

Perfil



Nome


Email


Nome de Utilizador


Idade

Guardar Alterações

Atividade Total do Utilizador

Distância
Vel Média

Tempo








Figura 12 – Mockup do perfil de utilizador

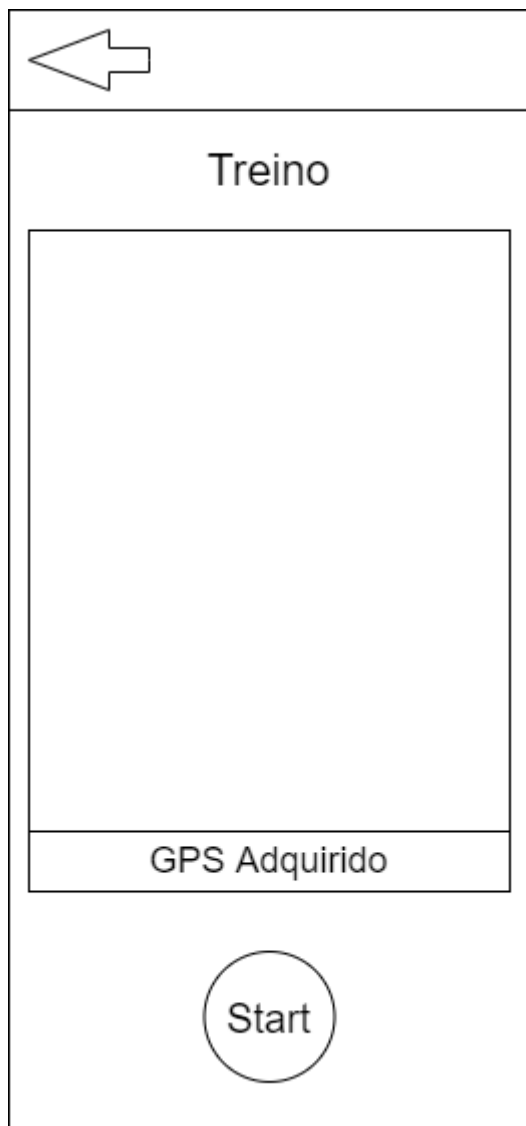


Figura 13 – Mockup da página para iniciar o treino

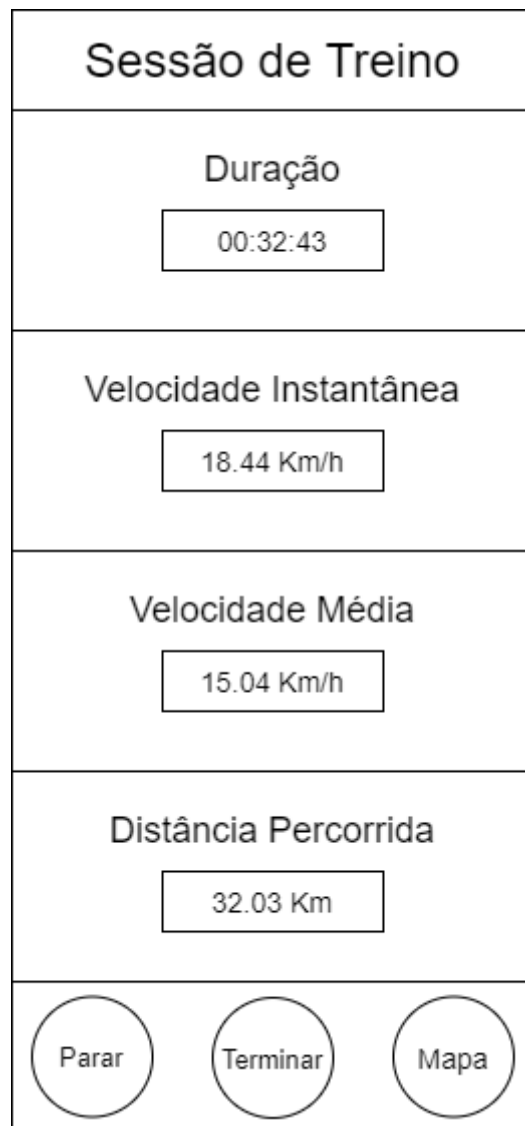


Figura 14 – Mockup da página de treino

# Resumo do Treino

00:02:30













V Med: 18.30 Km/h

1.02 Km

V Max: 32.30 Km/h

Guardar

Sair

Definições	
	
	
	
	
	
	
	
	
	
	
	
	

## 6.1. MODELO DE DADOS

A base de dados do projeto consiste em 11 tabelas, onde 5 são criadas pela Yii Framework. As outras 6, são criadas por código mySQL, desenvolvido pelo grupo. Todas estas tabelas servem para armazenar os dados do projeto.

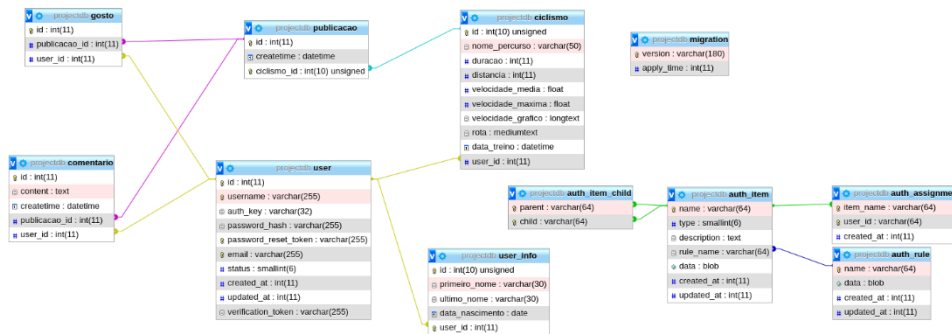


Figura 19 – Modelo de dados

## 7. Implementação

Em termos de implementação, o desenvolvimento do nosso projeto começou pela pesquisa de informação, o estudo de diversas API's externas, com as funcionalidades necessárias para o nosso projeto, e a criação dos repositórios Github.

Ao terminar a pesquisa de informação, começou-se a estudar a documentação da API externa escolhida, MapBox API, e iniciou-se a criação dos *mockups* das *activities* da aplicação Android. Ao terminar esta última tarefa, começou-se a criação das próprias *activities* no Android Studio.

Para o estudo da API externa, foi criado um projeto de testes, onde foi implementado todo o código que era estudado.

Além destas tarefas, desenvolveu-se a página principal do Website e os relatórios para as primeiras etapas do projeto das unidades curriculares de AMSI e PlatSI.

Após as primeiras etapas, desenvolveu-se o código da nossa API, com todos os métodos necessários a serem utilizados pela aplicação Android, e a implementação do código da API externa na mesma.

Para implementar esta última, desenvolveu-se as *activities* necessárias para a realização de um treino, a ligação entre estas e a classe da API externa, e o armazenamento dos dados dos treinos na base de dados local, exceto a informação da rota e do gráfico de velocidade.

Para mostrar os dados dos treinos realizados, implementou-se uma RecyclerView na página principal da aplicação Android, onde são apresentados os dados principais destes, e a *activity* de detalhes das sessões de treino, onde são exibidos todos os dados sobre esse.

Após isso, passou-se a desenvolver o código de comunicação com a nossa API na aplicação Android, mais precisamente, os métodos de login, registo e gestão do utilizador. Logo a seguir, os métodos de criação e listagem de sessões de treinos, onde, também, se desenvolveu a funcionalidade de criação de novas sessões de treinos sem haver acesso à Internet, com um sistema de sincronização com a nossa API no momento que houver novamente Internet.

A este ponto do desenvolvimento da aplicação Android, iniciou-se uma fase de testes, onde foi-se testando a criação de novos treinos, a criação da rota do mesmo e se os dados

estavam a ser armazenados corretamente. Também iniciou-se o desenvolvimento da *Backend* do Website.

Após esta fase de testes, implementou-se os métodos da nossa API de alteração e eliminação de uma sessão de treino, eliminação do próprio perfil na aplicação Android, e o armazenamento da rota do treino, tanto na base de dados local, como na nossa API.

Ao terminar as tarefas anteriores, passou-se a desenvolver o Website, onde se criou página de histórico de sessões de treino, implementou-se a API externa e a página de detalhes de cada treino. Enquanto isso, foi-se corrigindo alguns erros que apareciam na aplicação Android.

Depois destas páginas estarem criadas, implementou-se as páginas do *feed* de notícias e todas as funcionalidades do mesmo, como a possibilidade de reagir a uma publicação ou a possibilidade de comentar a mesma. Esta tarefa levou algum tempo a ser desenvolvida, devido a envolver bastantes funcionalidades diferentes. Além disto, começou-se a desenvolver os testes unitários, funcionais e de aceitação da *framework* Codeception na Yii *Framework*.

A este ponto, começou-se a armazenar os dados do gráfico da velocidade, tanto na base de dados local do Android, como na nossa API, e implementou-se o gráfico da velocidade nos detalhes de cada treino, utilizando esses dados.

Neste momento, começou-se a preparar o Website para a entrega na unidade curricular de PlatSI, comentando o código e realizando o relatório para a UC. Depois dessa entrega, criou-se alguns métodos na nossa API que ainda não tinham sido criados, como por exemplo, os métodos para a criação de novas publicações, e a implementação de funcionalidades baseadas em *messaging*, onde se desenvolveu um sistema de notificações.

Realizou-se o relatório para a unidade curricular de SISis e fez-se a devida entrega.

Por fim, implementou-se o sistema de notificações no Android e corrigiu-se os últimos erros da aplicação, com isso, preparando-a para a sua entrega na unidade curricular de AMSI. Realizou-se o relatório e fez-se a devida entrega.

Antes da entrega final na UC de Projeto em Sistemas de Informação corrigiu-se alguns erros que fomos encontrando, atualizou-se a aplicação com novas versões das bibliotecas e realizou-se o relatório e a apresentação, com isto fez-se a devida entrega do projeto.



## 8. Testes

Para o Website, os testes desenvolvidos no projeto consistiram em:

- Testes Unitários – 30 Testes;
- Testes Funcionais – 17 Testes;
- Testes Aceitação – 2 Testes;

Estes testes foram desenvolvidos utilizando a *framework* Codeception para a Yii Framework.

Nos testes unitários foram testados todos os métodos CRUD (Create, Read, Update, Delete) dos *models* utilizados.

Os testes funcionais têm o objetivo de testar todos os métodos utilizados na API, por exemplo, criar um utilizador.

Os testes de aceitação no nosso projeto são executados utilizando o Selenium e mostram o Website a funcionar em tempo real, onde é apresentado o processo de login, alterar um perfil e colocar/remover um gosto numa publicação do *feed* de notícias;

Estes testes são executados em poucos segundos e ajudaram-nos a verificar rapidamente se as funções principais do Website mantinham-se funcionais mesmo após serem feitas alterações que poderiam causar erros mais tarde.

Para a aplicação Android não foram desenvolvidos testes utilizando ferramentas, mas sim existiram vários testes feitos manualmente para verificar se existiam quaisquer erros para serem corrigidos. Devido á natureza do nosso projeto estes testes foram realizados ao ar livre e em movimento, consequência da utilização da localização GPS.

## 9. Conclusão e trabalho futuro

Para concluir, podemos dizer que gostámos bastante de trabalhar neste projeto. Em termos do Website, adquirimos bastantes conhecimentos sobre como trabalhar com uma *framework* avançada, como implementar um sistema RBAC e como criar testes unitários, funcionais e de aceitação. Contudo, a vasta complexidade da *framework* Yii2 dificultou a aprendizagem e a implementação de funcionalidades adicionais. Porém, também facilitou bastante o fluxo da programação ao longo destes meses.

Em termos da aplicação Android, implementámos as funcionalidades que nos propusemos inicialmente com a adição de alguns extras, assim como criar publicações após fazer um treino. Tornou-se um projeto bastante complexo, pela implementação de uma API externa, baseada totalmente na linguagem de programação Kotlin.

Aprendemos bastante com este projeto e, depois da apresentação, pretendemos continuar a desenvolvê-lo para o aperfeiçoarmos e adicionarmos mais funcionalidades de forma a pudermos o publicar um dia mais tarde.

Este foi um projeto que gostei bastante de trabalhar, já que era uma ideia que gostaria de desenvolver há bastante tempo. Foi um projeto bastante complexo, por envolver tecnologias que nunca tinha trabalhado, mas deu para adquirir novos conhecimentos em programação Android e em linguagens de programação Web. - Iuri Carrasqueiro

Gostei de trabalhar neste projeto. Este envolveu manipularmos uma boa variedade de tecnologias que futuramente irão nos ser úteis, tanto para currículo como para nós como programadores. - Gabriel Silva.

## 10. Bibliografia

<https://docs.mapbox.com/android/navigation/guides/>

<https://docs.mapbox.com/android/beta/maps/guides/>

<https://forum.yiiframework.com/t/how-to-create-a-pretty-url/79879/5>

<https://www.yiiframework.com/doc/guide/2.0/en/rest-authentication>

<https://stackoverflow.com/questions/27474059/yii-override-generic-rest-functions/27496376>

<https://stackoverflow.com/questions/27648504/yii2-override-generic-create-action-in-rest-activecontroller/50744982#50744982>

<https://github.com/yiisoft/yii2-app-advanced/blob/master/docs/guide/start-testing.md>

<https://stackoverflow.com/questions/43224842/how-to-add-multiple-icons-on-action-bar-in-android/43225435>

<https://stackoverflow.com/questions/28296708/get-clicked-item-and-its-position-in-recyclerview>