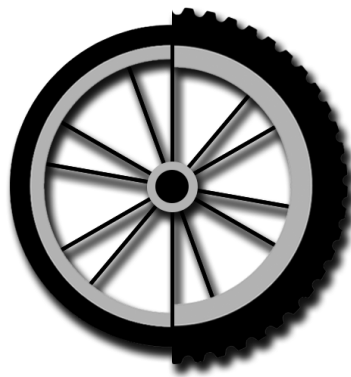


Apresentação do Projeto (Etapa 2)

Acesso Móvel a Sistemas de Informação

CicloDias



Letra do grupo: E

Elementos do grupo:

Nº: 2201127 Nome: Iuri Carrasqueiro

Nº: 2201133 Nome: Gabriel Silva

Torres Vedras, 24 de janeiro de 2022

Conta Github partilhada: https://github.com/GabrielMCS01/PSI_AMSI

Conta Jira partilhada:

<https://iuricarraspsi.atlassian.net/jira/software/projects/CP/boards/1>

Website Ciclodias: <http://ciclodias.duckdns.org/>

Índice

INTRODUÇÃO	3
1. CONTEXTUALIZAÇÃO DA APLICAÇÃO	4
2. REQUISITOS FINAIS IMPLEMENTADOS	5
2.1. MOTIVAÇÃO PARA O DESENVOLVIMENTO DO PROJETO	5
3. METODOLOGIA E PROCEDIMENTOS	6
3.1. API	8
4. PLANEAMENTO	11
4.1. TAREFAS PRINCIPAIS REALIZADAS	11
4.2. CALENDARIZAÇÃO E DISTRIBUIÇÃO DAS TAREFAS	12
5. TECNOLOGIAS UTILIZADAS E PESQUISADAS	13
6. DIFICULDADES E SOLUÇÕES	14
7. JUSTIFICAÇÃO DAS SOLUÇÕES IMPLEMENTADAS	15
CONCLUSÃO	16

Introdução

No âmbito da unidade curricular de Acesso Móvel a Sistemas de Informação do 1º Semestre do 2º ano do Curso TeSP de Programação de Sistemas de Informação do Instituto Politécnico de Leiria, foi criado o projeto “Ciclodias”, que consiste na elaboração de uma aplicação que monitoriza a atividade física do seu utilizador, apresentando informações essenciais do desporto praticado e realiza a gravação do percurso.

O projeto é comum entre várias disciplinas, sendo esta responsável pelo desenvolvimento da aplicação Android.

1. Contextualização da Aplicação

O projeto CicloDias é um sistema que permite a monitorização de atividade física do utilizador, mostrando-lhe informações essenciais, como a distância e velocidades, complementado com a gravação do percurso realizado utilizando a API Mapbox para gerar a rota do mesmo, utilizando a localização GPS para adquirir esses valores.

Com esses dados, o utilizador pode visualizar o seu histórico de atividade no menu principal, contendo as informações essenciais desses treinos, tanto na aplicação Android como no Website, clicando em cada treino pode-se o visualizar com todos os seus dados assim como o percurso percorrido no mapa.

O utilizador também tem a possibilidade de fazer a gestão do seu perfil, como editar ou apagar o mesmo.

2. Requisitos Finais Implementados

- Permitir o registo e login do utilizador de forma rápida e eficiente;
- Permitir terminar a sessão do utilizador;
- Permite fazer a gestão da conta do utilizador com o login feito;
- Permitir a obtenção da localização GPS do utilizador para realizar percursos, guardando a rota do mesmo;
- Permitir a realização de treinos em modo Offline, guardando-os em BD Local, fazendo a sincronização com a API assim que houver Internet e quando se navegar para a página principal;
- Apresentar dados essenciais durante a atividade física como Tempo, Distância, Velocidade Instantânea, Velocidade Média;
- Permitir a visualização do mapa durante o treino;
- Permitir pausar e retomar a sessão de treino;
- Permitir a aplicação ser executada em segundo plano durante o treino;
- Preencher a base de dados com as informações essenciais acerca da sessão de treino;
- Mostrar o histórico das sessões de treino realizadas pelo utilizador;
- Permitir fazer a gestão das sessões de treino do utilizador;
- Permitir a criação de publicações a partir dos detalhes dos treinos;
- Receber notificações providenciadas pelo Website;

2.1. Motivação para o Desenvolvimento do Projeto

Uma das razões principais para o desenvolvimento deste projeto foi o quão abrangente ele é, utilizámos uma API externa para os treinos, criámos uma mini rede social, o que permitiu ser uma boa fonte de aprendizagem.

Este projeto também nos permitiu fazer bastantes pausas, já que cada vez que se precisou de testar alguma funcionalidade da aplicação, tivemos que fazer caminhadas para o testar, uma vez que era necessária localização GPS.

3. Metodologia e Procedimentos

Começámos por fazer uma pesquisa de ideias em outras aplicações populares no mesmo ramo, e também fizemos os layouts iniciais da aplicação.

Implementámos o código base para termos acesso a todos os botões e caixas de texto no código em todas as *Activities* necessárias.

Pesquisámos algum tempo como implementar a API do mapa no Android num projeto à parte e quando tivemos a base do mapa a trabalhar implementámos no projeto principal.

Implementámos uma *NavBar* que permite o acesso às 3 funcionalidades principais da aplicação.

Implementámos a biblioteca Volley, assim que tínhamos feito uma parte da API para começarmos a implementarmos esses métodos.

Implementámos o Login e Registo na aplicação, assim como a visualização do perfil com os dados provenientes da API.

Logo a seguir implementámos o código para criarmos sessões de treino, mas sem enviarmos ainda a rota e o *array* de velocidade para a API.

Fizemos bastantes otimizações e correções no projeto a partir deste ponto, para que a aplicação se tornasse mais estável, exemplos disso são: A aplicação não fosse abaixo ao guardar os dados na API; Limpeza dos dados de sessões de treino anteriores.

Implementámos um *OptionsMenu* alternativo para acolher diferentes ícones com diferentes funcionalidades, conforme a necessidade.

Implementámos a *RecyclerView* com todos os treinos feitos pelo utilizador, esta é carregada quando o utilizador faz login na aplicação.

Mais tarde preparámos a aplicação para funcionar em modo offline, os dados dos treinos são guardados em BD Local e assim que houver internet e navegar-se para o menu principal os novos dados são enviados para a API.

Implementou-se mais na fase final os treinos a enviar a rota e o *array* de velocidades para a API.

Implementou-se funcionalidades baseadas em *messaging* na aplicação para enviar notificações ao utilizador quando alguém interage com as suas publicações ou quando alguém alcança um novo recorde de distância, duração ou velocidade média.

3.1. API

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

3.1.1. 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);

3.1.2. 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 de 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;

3.1.3. 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 ciclismo. Este método contém verificação de se o treino que irá ser publicado é do utilizador;

Parâmetros:

- int “ciclismo_id”;

4. Planeamento

4.1. Tarefas principais realizadas

- T1 – Pesquisa de ideias para o projeto, e criação de repositórios;
- 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, Pausa, Resultados);
- T8 – Código para visualização do mapa em um *fragment*;
- T9 – Código da MainPage para a visualização dos percursos mais recentes realizados pelo utilizador;
- T10 – Comunicação com a API (Login e Registo);
- T11 – Fase de testes;
- T12 – Desenvolvimento de todas funções para comunicação com a API (Criar Treinos, Editar Perfil, etc....);
- 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;

4.2. Calendarização e distribuição das tarefas

- 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 – 1 semanas - ambos os membros;
- T12 – 3 semanas - ambos os membros;
- T13 – 3 semanas - ambos os membros;
- T14 – 1 semana – Iuri Carrasqueiro;
- T15 – 1 semana – ambos os membros;

5. Tecnologias Utilizadas e Pesquisadas

Para a utilização do mapa decidimos utilizar a API Mapbox, por ser utilizada em muitas outras aplicações reconhecidas, além disso, esta tem muitas das funcionalidades necessárias para o nosso projeto funcionar.

Utilizamos a biblioteca Volley para comunicação com a API. Esta era uma biblioteca de utilização obrigatória e não é muito complexa de se implementar.

Por fim, para trabalharmos com o sistema de notificações, utilizamos a biblioteca Eclipse Paho Android Client (MQTT). Estas notificações aparecem para os utilizadores quando alguém interage com as suas publicações ou quando alguém alcança um novo recorde de distância, duração ou velocidade média.

6. Dificuldades e Soluções

A implementação do mapa inicialmente foi demorada, teve que se criar um projeto á parte para testar a aplicação e só quando tivemos a base do mapa a trabalhar é que o implementámos na aplicação final, depois desta implementação, tivemos alguns problemas para iniciar sessões de treino e guardar os pontos de localização para desenhar a rota no mapa.

A funcionalidade de modo *offline* deu-nos alguns problemas no início, devido á complexidade da funcionalidade. Teve que se alterar as funções responsáveis por criar novos treinos e da sincronização com a API. Além disso, implementou-se o armazenamento dos novos treinos que foram realizados enquanto em modo *offline*, para uma sincronização futura.

Manter a aplicação a trabalhar em segundo plano durante o treino também deu problemas, principalmente quando se tentava voltar á aplicação e esta não mostrava a *Activity* da sessão de treino. Para resolver, implementou-se uma variável que está ativa durante a execução do treino.

Como algumas funcionalidades da API Mapbox continuavam a estar ativas em segundo plano, mesmo o utilizador estar em *Activities* onde estas funcionalidades não eram necessárias, causava alguns problemas ao voltar para *Activities* onde estas eram necessárias, para isto, teve-se que dar *override* a algumas funções como `onBackPressed()`, `onResume()` e `onPause()` para resolver este problema.

De novo manter a aplicação a trabalhar em segundo plano para receber notificações, teve que se criar um serviço para estas funcionarem em segundo plano.

7. Justificação das Soluções Implementadas

Decidimos utilizar uma *RecyclerView* no menu principal devido a ser mais simples de trabalhar e de navegar entre os treinos feitos pelo utilizador, ao mesmo tempo sem comprometer com algum dado importante a este.

Refizemos a *OptionsMenu* do menu principal, dos detalhes do treino e do perfil do utilizador para esta se adequar às necessidades destas *Activitys*.

Mantive-mos a mesma paleta de cores em todos os menus de forma a condizer com o logo da aplicação.

Conclusão

Para concluir, sabemos que o projeto está bom e que se enquadra em todos os requisitos, implementámos as funcionalidades propostas 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.

Por resolver temos o sistema de notificações, que só funciona quando a aplicação está minimizada ou aberta, ou seja, quando se fecha totalmente a aplicação, o utilizador deixa de receber notificações.

Aprendemos bastante com este projeto e, depois das apresentações, pretendemos continuar a desenvolver este projeto para o aperfeiçoarmos e adicionarmos mais funcionalidades de forma a publicarmos 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. – Iuri Carrasqueiro

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

Fontes

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

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