

**Licenciatura em Engenharia de Telecomunicações e
informática**

Desenvolvimento de Software e Sistemas Móveis

“FavoriteSoccer”

Eduardo Martins-1200892 / Pedro Soares-1200909

Docentes: Paulo Baltarejo Sousa e Carlos Filipe Freitas

Conteúdo

Introdução	4
Contextualização	4
Ide, linguagem e ferramentas	5
Análise	6
Modelo de domínio	6
Requisitos não funcionais	6
Requisitos funcionais	7
Use Case Diagram:	7
SSD Diagrams:	8
Design	10
Arquitetura Física	10
Flux Architecture	11
Organização do código	11
	11
Sequence Diagrams	12
	12
Mock Ups	13
Listar equipas de uma equipa:	13
	13
Remove liga dos favoritos:	14
Adicionar uma liga aos favoritos:	14
Implementação	15
Código do LeaguesListItem	15
Código do LeaguesList:	16
Código do Action:	16
Código da Reducer:	17
Testes	18
Adicionar uma liga aos favoritos:	18
Remover uma liga dos favoritos:	18
Listar as equipas de uma liga:	19
Conclusão	19
Objetivos	19
Visão Geral	19
Bibliografia	20

Figura 1 IntelliJ IDE logo.....	5
Figura 2 Visual Paradigm Logo	5
Figura 3 React Native Logo	5
Figura 4 Domain Model.....	6
Figura 5 Use Case Diagram.....	7
Figura 6 SSD-UC01.....	8
Figura 7 SSD-UC02.....	8
Figura 8 SSD-UC03.....	9
Figura 9 SSD-UC04.....	9
Figura 10 SSD-UC05.....	10
Figura 11 Usage of deployment diagram	10
Figura 12 Flux Architecture	11
Figura 13 Usage of package diagram	11
Figura 14 SD-UC01.....	12
Figura 15 SD-US05	13
Figura 16 SD-UC04.....	12
Figura 17 Listar equipas de uma determinada liga	13
Figura 18 Remover uma liga dos favoritos.....	14
Figura 19 Adicionar uma liga aos favoritos	14
Figura 20 Adicionar liga aos favoritos	18
Figura 21 Remover liga dos favoritos.....	18
Figura 22 Listar equipas da respetiva liga	19

Introdução

Contextualização

Este projeto tem como objetivo a criação de uma aplicação que permite ao utilizador aceder a uma lista de todas as principais ligas de futebol mundial.

Funcionalidades do Programa:

- Ter acesso à informação relativa as principais ligas de futebol, tais como tabelas classificativas e respetivos pontos
- Possibilidade de adicionar como favoritas as ligas que o utilizador pretender, estando destacadas num novo layout.
- Possibilidade de remover como favorita alguma liga.

Ide, linguagem e ferramentas

Para este projeto foi utilizado o *IntelliJ IDEA* desenvolvido pela *JetBrains* que permite escrever em duas linguagens, Kotlin e Java, sendo que a adotada, neste caso, foi a linguagem Javascript.



Figura 1 IntelliJ IDE logo

Para além deste, foi utilizado o *Visual Paradigm* para a projeção dos diagramas.



Figura 2 Visual Paradigm Logo

Foi também utilizada uma ferramenta, React Native, uma biblioteca, criada pelo Facebook que permite desenvolver aplicações para android e IOS.

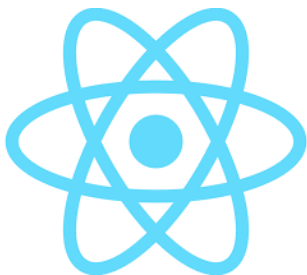


Figura 3 React Native Logo

Análise

Modelo de domínio

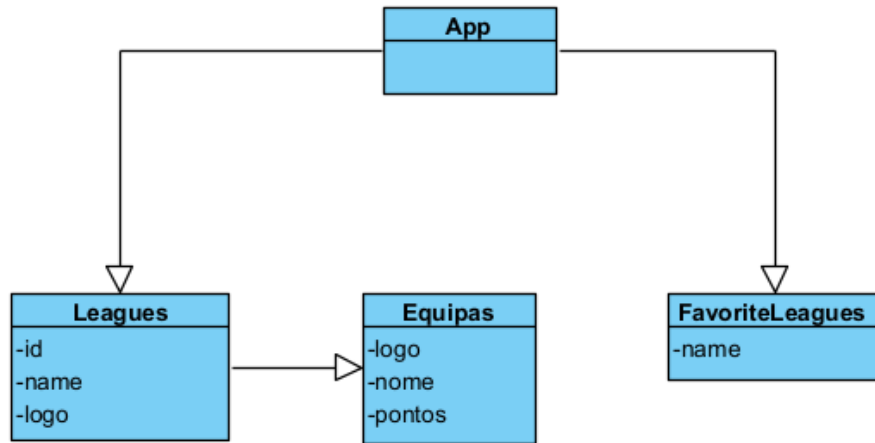


Figura 4 Domain Model

Requisitos não funcionais

- Usabilidade:
 - Interface Gráfica.
- Confiabilidade:
 - Não.
- Performance:
 - Não.
- Suportabilidade:
 - Testes;
 - Plataforma Andorid e IOS;
 - API.
- Restrições de Design:
 - Não.
- Restrições de implementação:
 - Linguagem JAVASCRIPT.
- Restrições de interface:
 - Interface gráfica do Android e IOS

Requisitos funcionais

Use Case Diagram:

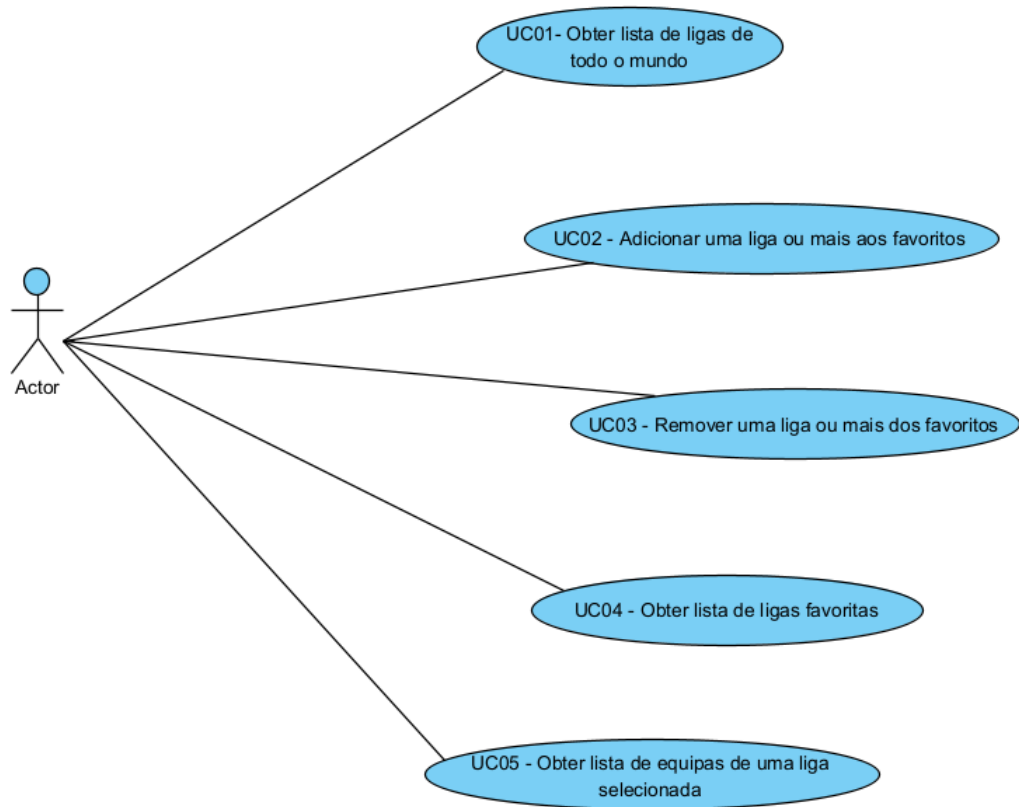


Figura 5 Use Case Diagram

SSD Diagrams:

UC01: Nesta UC obtemos a lista das ligas, disponíveis na API.

Descrição	Obter lista de ligas
Pasta Básica	<ul style="list-style-type: none">• O usuário escolhe o screen das Ligas.• O sistema retorna as ligas presentes na API.
Pasta alternativa	Nenhuma

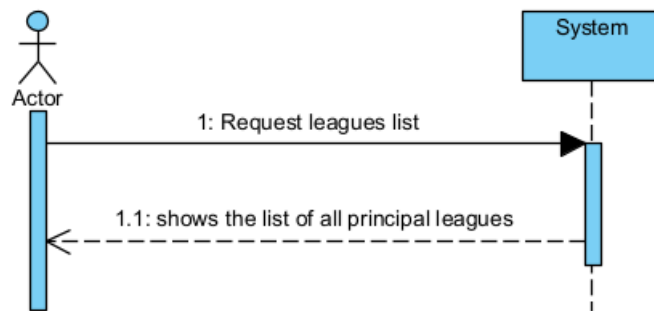


Figura 6 SSD-UC01

UC02: Nesta UC adicionamos uma Liga, clicando no Botão da respetiva.

Descrição	Adicionar liga aos favoritos
Pasta Básica	<ul style="list-style-type: none">• O usuário escolhe o screen das Ligas.• O usuário clica no botão de adicionar aos favoritos.• O sistema adiciona a liga aos favoritos.
Pasta alternativa	Nenhuma

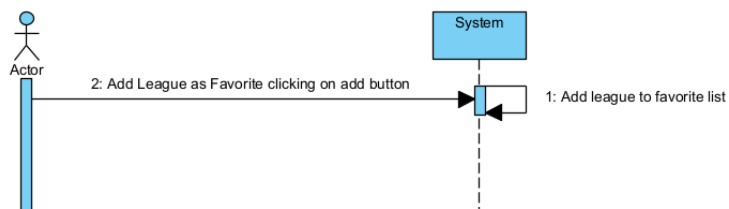


Figura 7 SSD-UC02

UC03: Nesta UC removemos uma Liga dos favoritos, previamente adicionada aos favoritos, usando o botão respectivo dela.

Descrição	Remover liga dos favoritos
Pasta Básica	<ul style="list-style-type: none"> O usuário escolhe o screen dos Favoritos. O usuário clica no botão de Remover. O sistema remove a Liga dos Favoritos.
Pasta alternativa	Nenhuma

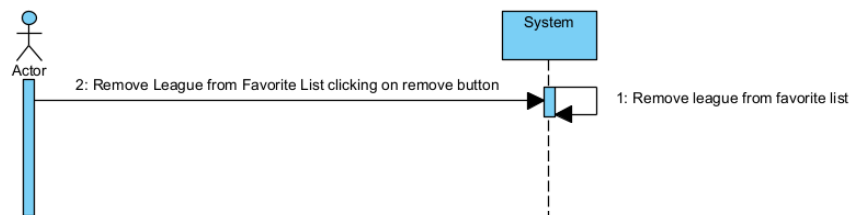


Figura 8 SSD-UC03

UC04: Nesta UC obtemos a lista das ligas adicionadas aos favoritos.

Descrição	Obter lista de equipas favoritas
Pasta Básica	<ul style="list-style-type: none"> O usuário escolhe o screen dos Favoritos O sistema retorna a lista de ligas favoritas
Pasta alternativa	Nenhuma

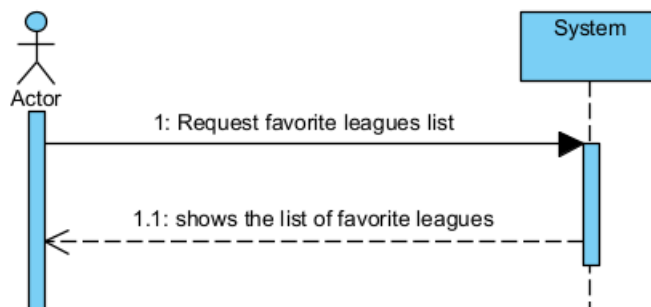


Figura 9 SSD-UC04

UC05: Nesta UC obtemos a lista de equipas, assim como a sua respetiva classificação, ao clicar no seu logo.

Descrição	Obter lista de equipas da liga
Pasta Básica	<ul style="list-style-type: none">• O usuário escolhe o screen das Ligas.• O usuário clica no logo da Liga.• O sistema retorna as equipas presentes na respetiva liga.
Pasta alternativa	Nenhuma

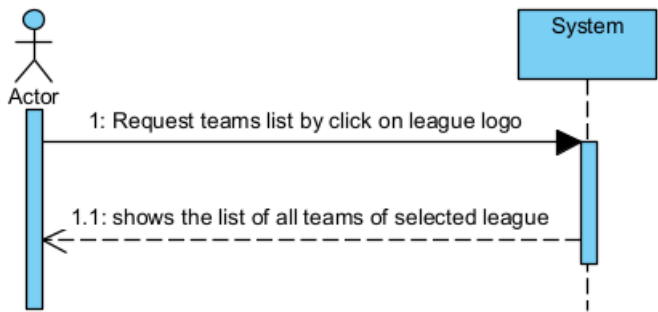


Figura 10 SSD-UC05

Design

Arquitetura Física

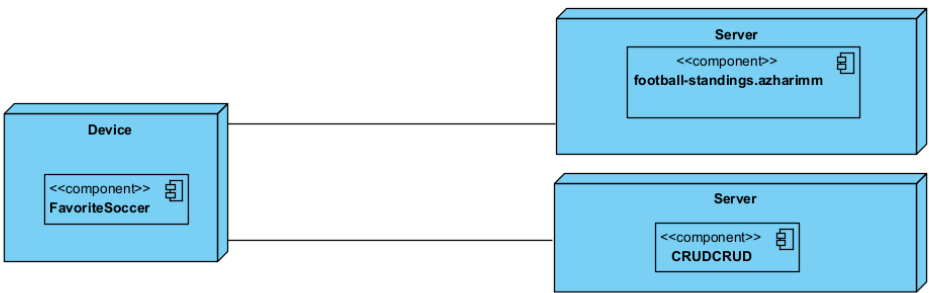


Figura 11 Usage of deployment diagram

Neste trabalho são utilizadas duas APIs, a “football-standings-azharimm” que é usada apenas para obter as ligas e as suas respetivas equipas e a “CRUDCRUD” que é usada para adicionarmos as ligas favoritas e removermos.

Flux Architecture

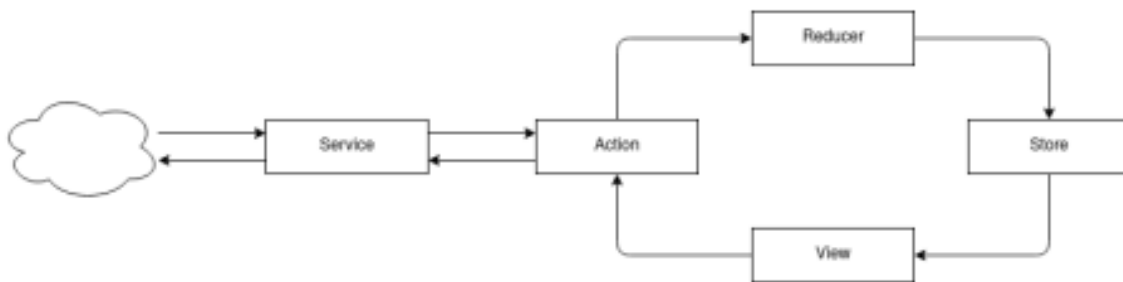


Figura 12 Flux Architecture

Organização do código

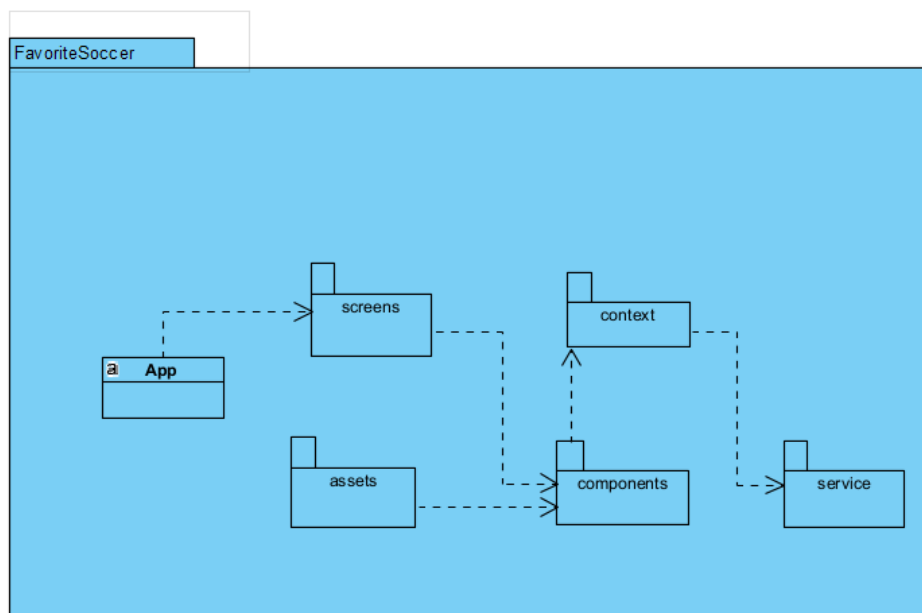


Figura 13 Usage of package diagram

Sequence Diagrams

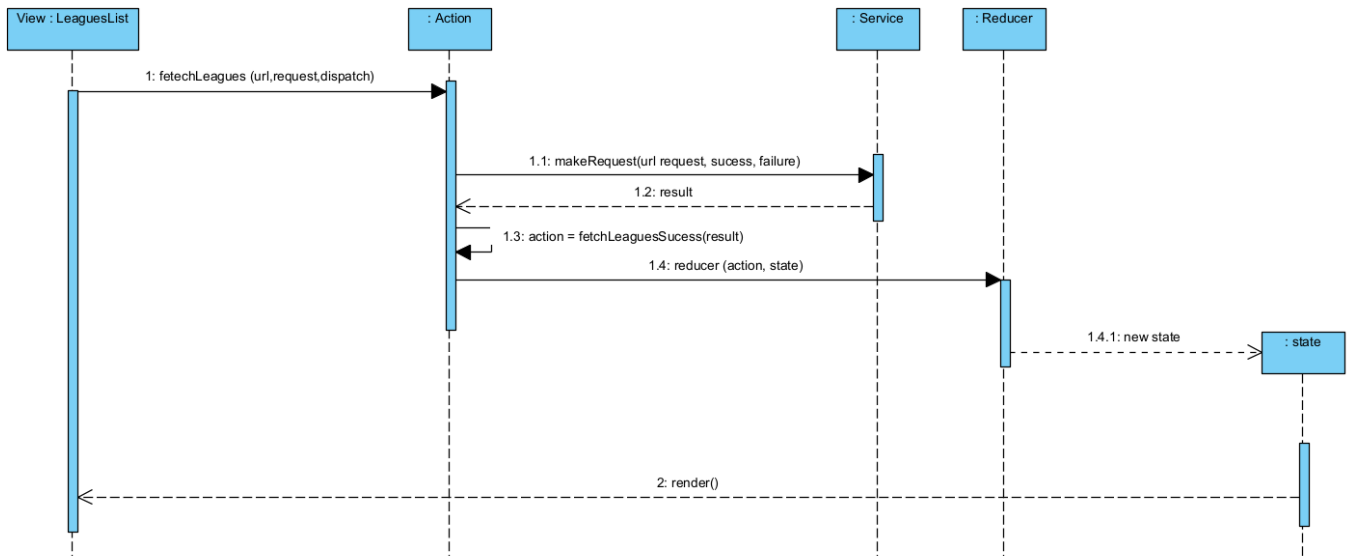


Figura 14 SD-UC01

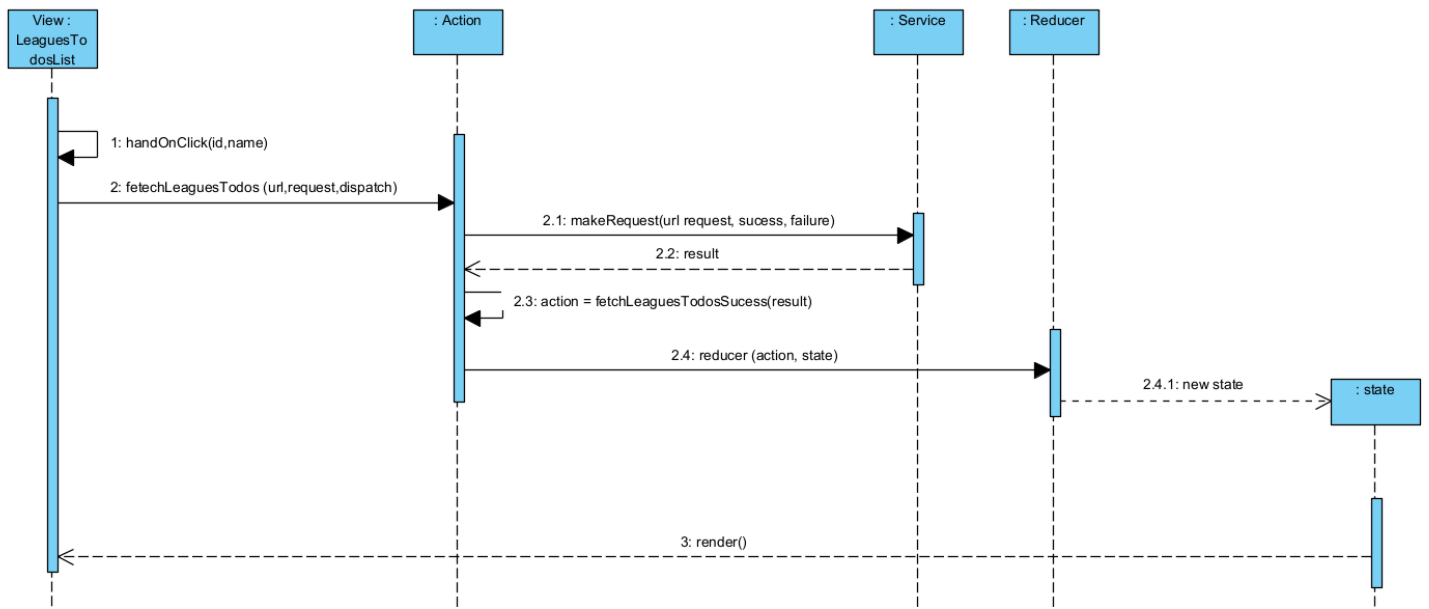


Figura 15 SD-UC04

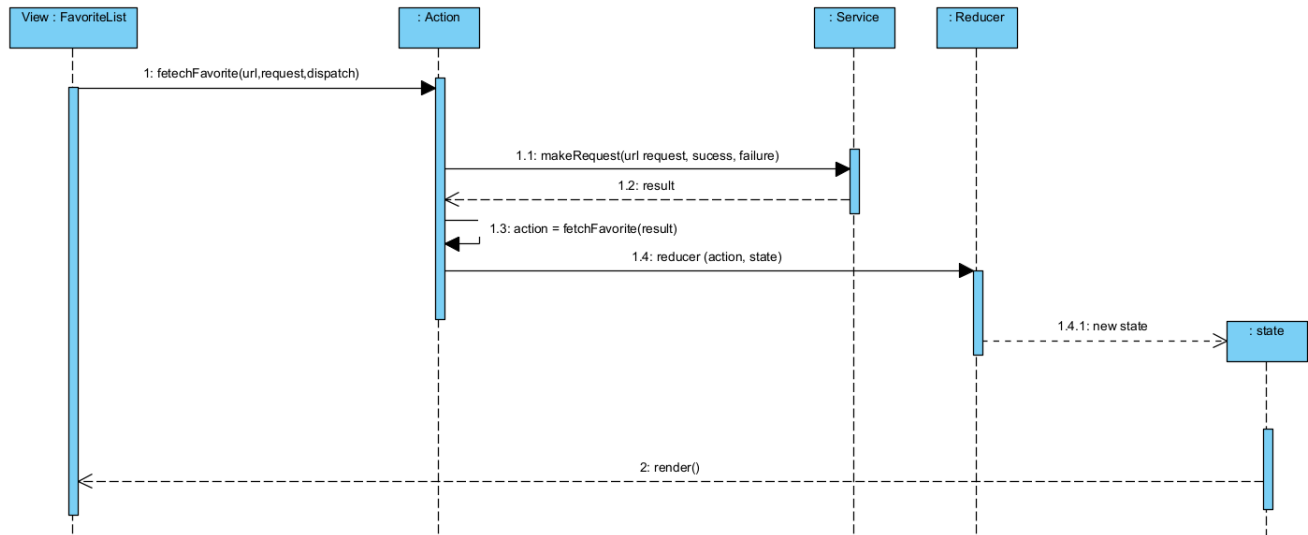


Figura 16 SD-US05

Mock Ups

Listar equipos de una equipo:

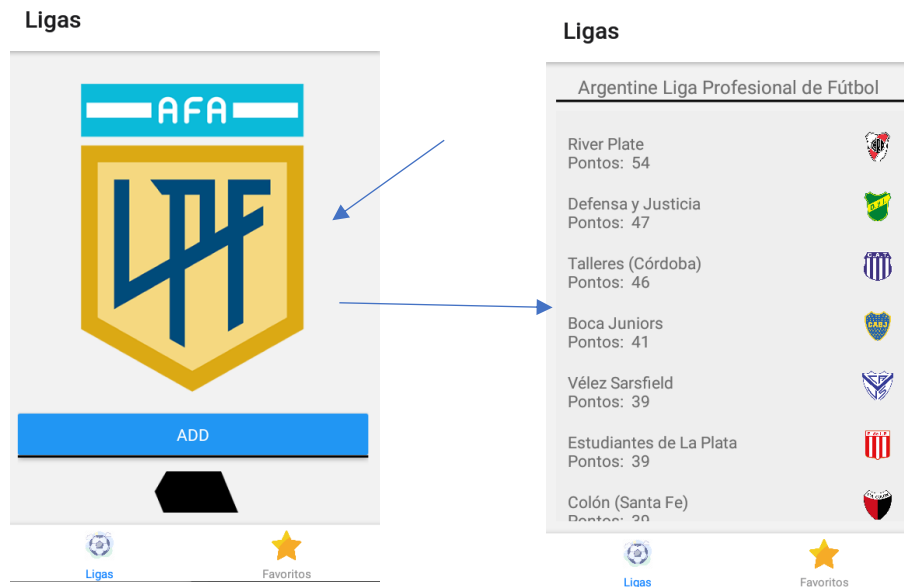


Figura 17 Listar equipos de una determinada liga

Remove liga dos favoritos:



Figura 18 Remover uma liga dos favoritos

Adicionar uma liga aos favoritos:

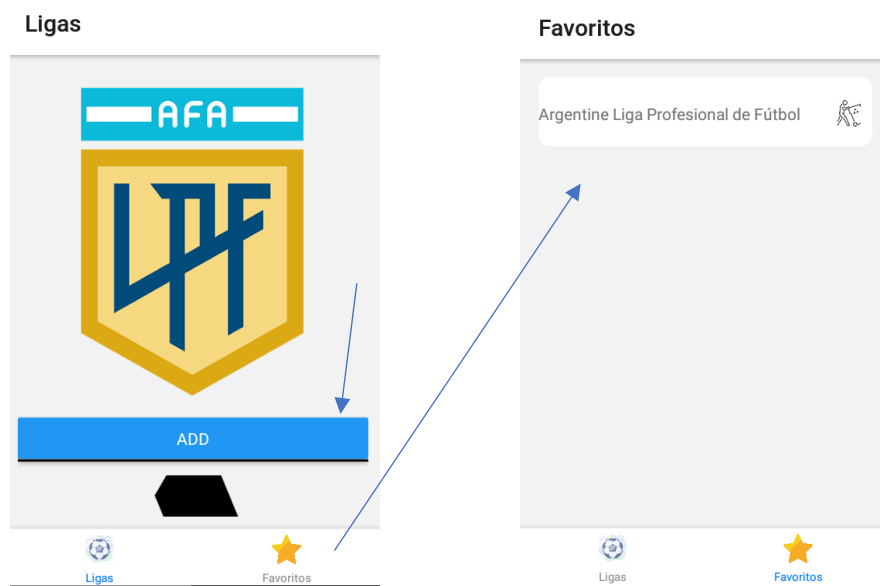


Figura 19 Adicionar uma liga aos favoritos

Implementação

Código do LeaguesListItem

```
1. const _Post = item => {
2.   fetch(
3.     'https://crudcrud.com/api/a847e5124aaf47e3af34b8577963b168/unicorns',
4.     {
5.       method: 'POST',
6.       headers: {
7.         Accept: 'application/json',
8.         'Content-type': 'application/json',
9.       },
10.      body: JSON.stringify({
11.        Name: item.name,
12.      }),
13.    },
14.  ).then(response => response.json());
15. }
```

Neste excerto de código é efetuado o “Post”, do nome da liga favorita, para a API.

Código do LeaguesList:

```
1. const {state, dispatch} = useContext(AppContext);
2. const {ligas} = state;
3. const {loading, error, data} = ligas;
4. useEffect(() => {
5.   dispatch(fetchLeaguesStarted());
6.   const url = `${URL_API}/leagues`;
7.   const request = {};
8.   fetchLeagues(url, request, dispatch);
9. }, []);
10.
```

É acionada uma mudança de estado, que no Action vai receber uma “type” que lhe vai dizer o que fazer, que no caso é atribuir ao state o initialState

Função que vai atribuir a array data, dentro do state, a informação proveniente da API

Assim, resumindo, neste excerto de código temos onde é iniciado a mudança de estado para os dados da API e onde é iniciada a ligação a API.

Código do Action:

```
1. export function fetchLeaguesSuccess(ligas) {
2.   return {
3.     type: FETCH_LEAGUES_SUCCESS,
4.     payload: {
5.       data: [...ligas],
6.     },
7.   };
8. }
9.
```

Contem as informações que devem ser mudadas no state

Código da Reducer:

```
1. function reducer(state, action) {  
2.   switch (action.type) {  
3.     case FETCH_LEAGUES_STARTED:  
4.       return {  
5.         ...state,  
6.         ligas: {  
7.           loading: true,  
8.           error: null,  
9.           data: [],  
10.        },  
11.      };  
12.     case FETCH_LEAGUES_SUCCESS:  
13.       return {  
14.         ...state,  
15.         ligas: {  
16.           loading: false,  
17.           error: null,  
18.           data: [...action.payload.data],  
19.        },  
20.      };  
21.     default:  
22.       return state;  
23.   }
```

Como foi dito anteriormente, aqui é criado ou alterado um state e atribuído o initialState

Aqui, é realizada a atribuição dos dados provenientes da API que vão para o payload, no Action.

Testes

Adicionar uma liga aos favoritos:

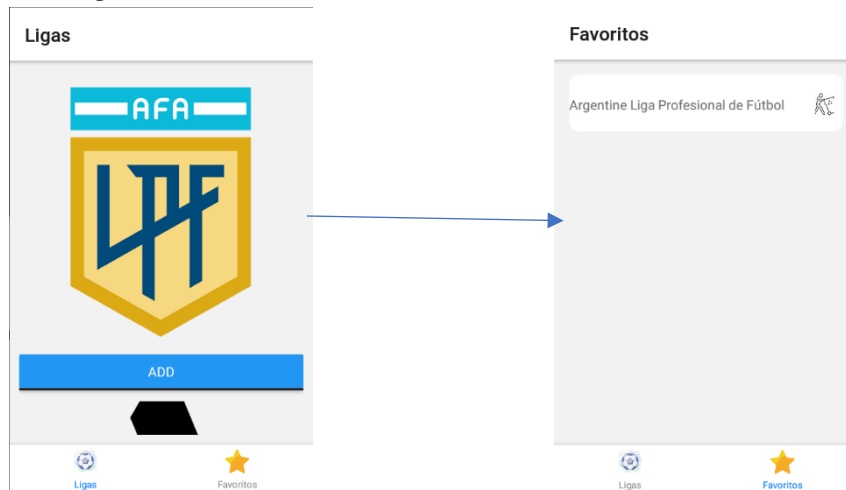


Figura 20 Adicionar liga aos favoritos

Remover uma liga dos favoritos:

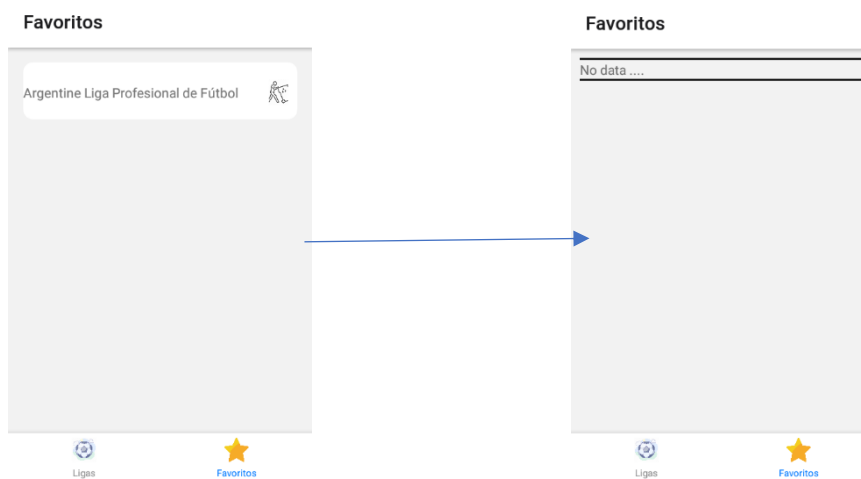


Figura 21 Remover liga dos favoritos

Listar as equipas de uma liga:



Figura 22 Listar equipas da respetiva liga

Conclusão

Objetivos

Este projeto teve como objetivo aplicar e cimentar todos os conhecimentos adquiridos nas aulas com a criação de uma aplicação programada em Java.

Visão Geral

De uma forma geral, o projeto foi bem sucedido tendo cumprido todos os objetivos e funcionalidades previstas pelo grupo, apesar de algumas dificuldades que nos desafiaram mas que nos permitiram melhorar as competências na programação, principalmente na linguagem Java.

Agradecimentos

Por fim, queremos agradecer aos docentes da disciplina Paulo Baltarejo Sousa e Carlos Filipe Freitas pelo apoio e disponibilidade prestada ao longo da realização de todo o projeto.

Bibliografia

- [1] – Plataforma Moodle da disciplina de Desenvolvimento de Software e Sistemas Móveis;
- [2] – Wikipedia “https://pt.wikipedia.org/wiki/Wikipédia:Página_principal”;
- [3] – AZHARIMM API “<https://github.com/azharimm/football-standings-api>”;
- [4] – CRUDCRUD “<https://crudcrud.com/>”;
- [4] – Stack Overflow “<https://pt.stackoverflow.com>”.