

# **Especificação de Requisitos de Software**

**por**

**reBOOK**

**Versão 1.7**

**Elaborado por Grupo 13**

**Laboratório de Desenvolvimento de Software**

**03/11/2020**

# Índice

<b>1. Introdução</b>	<b>1</b>
1.1 Propósito	1
1.2 Âmbito	1
<b>2. Descrição geral</b>	<b>2</b>
2.1 Perspetiva do Produto	2
2.2 Funcionalidades	2
2.3 Ambiente Operacional	2
<b>3. Requisitos específicos</b>	<b>3</b>
3.1 Requisitos funcionais	3
3.2 Requisitos não funcionais	9
<b>4. Mockups</b>	<b>10</b>
4.1 Login	10
4.2 Editar dados de conta	11
4.3 Adicionar livro	12
4.4 Editar livro	13
4.5 Ver lista de livros	14
4.6 Sistema de Match	15
4.7 Sistema de troca de mensagens	16
4.8 Lista de Favoritos	17
4.9 Sistema de procura	18
4.10 Dashboard da aplicação	19
<b>5. Use Cases</b>	<b>20</b>
5.1 Use Case 1 - Gestão de utilizadores	20
5.2 Use Case 2 - Interações	22
5.3 Use case 3 – Sistema de procura	24
5.4 Use Case 4 - Gestão de livros	25
5.5 Use Case 5 - Gestão de Favoritos	27
<b>6. Explicação Geral do funcionamento da aplicação</b>	<b>29</b>
6.1 Visão Geral	29
6.2 Estrutura da aplicação	29
6.3 Modelo de dados	29
6.4 Autenticação JWT (Json Web Token)	30
6.5 Ferramentas e Metodologias utilizadas	31
<b>7. Arquitetura do Projeto</b>	<b>34</b>
<b>8. Diagrama de Classes</b>	<b>35</b>
<b>9. Conclusão e limitações</b>	<b>36</b>

## Índice de figuras

Figure 1 - Login .....	10
Figura 2 - Mockup Login .....	10
Figura 3 - Mockup editar conta.....	11
Figure 4 - Mockup Adicionar livro .....	12
Figure 5 - Mockup editar livro.....	13
Figure 6 - Mockup ver lista de livros.....	14
Figure 7 - Mockup matchs .....	15
Figure 8 - Mockup Mensagens .....	16
Figure 9 - Mockup Lista de Favoritos.....	17
Figure 10 - Mockup Sistema de procura.....	18
Figure 11 - Mockup Dashboard .....	19
Figure 12 - Use Case gestão de Utilizadores .....	20
Figure 13 - Use Case Interações .....	22
Figure 14 - Use Case Sistema de Procura .....	24
Figure 15 - Use Case Gestão de livros.....	25
Figure 16 - Use Case gestão de favoritos.....	27
Figure 17 - Autenticação via JWT .....	30
Figure 18 - Autenticação via JWT .....	30
Figure 19 - JetBrains Rider (IDE).....	31
Figure 20 - ASP.NET Core.....	31
Figure 21 - GitLab ESTG .....	32
Figure 22 - Documentação Swagger.....	33
Figure 23 - Arquitetura do Projeto.....	34
Figure 24 - Diagrama de Classes .....	35

## Historico de revisões

Nome	Data	Razão da alteração	Versão
João Leocádio, Paulo Brito, Carlos Meireles, Manuel Garrido	26/10/2020	Adição da introdução	0.1
João Leocádio, Paulo Brito	27/10/2020	Adição de mockups	0.2
Manuel Garrido, Carlos Meireles	29/10/2020	Adição de Use Cases	0.3
João Leocádio, Paulo Brito, Carlos Meireles, Manuel Garrido	30/10/2020	Adição dos requisitos	0.4
João Leocádio, Paulo Brito	03/11/2020	Formatação e revisão da ortografia	1.0
João Leocádio	03/12/2020	Substituição do Use Case de utilizadores	1.1
Manuel Garrido	07/12/2020	Formatação e revisão da ortografia	1.2
João Leocádio	17/12/2020	Arquitetura do Projeto	1.3
João Leocádio	22/12/2020	Diagrama de Classes	1.4
João Leocádio	27/12/2020	Autenticação JWT	1.5
João Leocádio	07/01/2021	Ferramentas e Metodologias	1.6
João Leocádio	08/01/2021	Índice de imagens	1.7

# **1. Introdução**

## **1.1 Propósito**

O objetivo principal deste documento reside essencialmente em descrever ao detalhe todos os requisitos de software do projeto escolhido. Como tal, o documento explicará de forma eficiente e completa, todos os requisitos, restrições e interações com o utilizador através da interface, para que este valide e aprove o programa desenvolvido.

## **1.2 Âmbito**

A nossa ideia surgiu, enquanto pensavamos na importância de manter uma leitura vasta e regular na nossa rotina.

Deste modo, a principal finalidade do nosso software, é o de “dar uma nova vida” a livros usados e promover a partilha/troca desses objetos entre várias pessoas.

Assim, o projeto proposto consiste na elaboração de um software voltado para a vertente de livros. Através desta aplicação, será possível ao utilizador trocar, ver e seguir livros de vários outros utilizadores. Também será possível trocar mensagens através de um sistema de chat incorporado na aplicação interagindo com outros utilizadores e também gerir e editar a sua própria conta.

## 2. Descrição geral

### 2.1 Perspetiva do Produto

Para a (re)Book procederemos à construção de uma WEB API utilizando a framework DotNet e também procederemos ao desenvolvimento de uma aplicação para android utilizando a framework Xamarian com vista a consumir os serviços implementados na API.

Após criação de uma conta, o utilizador poderá adicionar livros à sua lista. Cada livro terá uma ou várias fotos, uma descrição, assim como uma avaliação do seu estado (novo, usado, muito usado).

Na funcionalidade principal da aplicação o utilizador poderá ver os livros que pertencem a utilizadores na sua área numa interface ao estilo do *“Tinder”*.

Na interface, aparece uma foto do livro assim como os seus dados, aqui o utilizador pode seleccionar gostar ou passar. Se gostar, o livro é colocado na sua lista de favoritos, caso o dono deste livro gostar de um livro do utilizador, iremos ter um match que desbloqueará a funcionalidade de chat entre estes dois utilizadores de maneira a poderem combinar uma troca. Nesta visão o utilizador pode também escolher não voltar a ver o livro na sua lista de pesquisa, reportar este livro aos administradores ou redefinir o seu range.

O utilizador também poderá aceder ao perfil do dono desse livro de maneira a ver a sua lista.

Também é possível efetuar uma pesquisa por todos os livros na área.

### 2.2 Funcionalidades

- Criar, editar, remover ou bloquear utilizadores;
- Adicionar/remover livros da lista de troca;
- Adicionar/remover livros da lista de favoritos;
- Sistema de match com outros utilizadores;
- Sistema de chat em caso de match;
- Sistema de pesquisa de livros num range definido;
- Sistema de pesquisa de utilizadores.

### 2.3 Ambiente Operacional

A aplicação será desenvolvida para plataformas android.

## 3. Requisitos específicos

### 3.1 Requisitos funcionais

#### 3.1.1 Requisito funcional <0001>

Categoria: Gestão de Utilizador

Nome: Autenticar Conta

Id: CU – RF – 01

Descrição: Autenticação de conta com nome de utilizador e palavra passe

Restrições: Para a correta autenticação de conta, o login tem que ser efetuado com sucesso

Prioridade: 1

Estado: Fixo

Verificações: Se a conta existir e dados inseridos corretamente dá-se a autenticação. Caso contrário, dá erro.

#### 3.1.2 Requisito funcional <0002>

Categoria: Gestão de Utilizador

Nome: Editar Conta

Id: CU – RF – 02

Descrição: Editar dados na conta do utilizador

Restrições: Para a correta edição de conta, o login tem que ser efetuado com sucesso

Prioridade: 1

Estado: Fixo

Verificações: Alterações na conta são guardadas

#### 3.1.3 Requisito funcional <0003>

Categoria: Gestão de Utilizador

Nome: Remover Conta

Id: CU – RF – 03

Descrição: Remover conta de utilizador da base de dados

Restrições: Para a correta eliminação da conta, o login tem que ser efetuado com sucesso

Prioridade: 1

Estado: Fixo

Verificações: Conta é eliminada e utilizador é remetido para a homepage

#### 3.1.4 Requisito funcional <0004>

Categoria: Gestão de Utilizador

Nome: Criar Conta

Id: CU – RF – 04

Descrição: Cria conta de utilizador e grava-a na base de dados

Restrições: Ao ser criado a conta, será necessário fornecer um ID e a password.

Prioridade: 1

Estado: Fixo

Verificações: Conta é adicionada à base de dados e utilizador remetido à página de perfil

### **3.1.5 Requisito funcional <0005>**

Categoria: Gestão de favoritos

Nome: Adicionar livro aos favoritos

Id: CF – RF – 05

Descrição: Permite adicionar um livro à lista de favoritos

Restrições: Livro não deve pertencer a utilizador nem constar da lista

Prioridade: 1

Estado: Fixo

Verificações: Livro é adicionado à lista de favoritos

### **3.1.6 Requisito funcional <0006>**

Categoria: Gestão de Livros

Nome: Ver informação de livro

Id: CF – RF – 06

Descrição: Permite ver mais informação detalhada de um livro em específico.

Restrições: Sucesso em condições normais, insucesso caso página não exista.

Prioridade: 1

Estado: Fixo

Verificações: Utilizador é remetido para a página de info do livro

### **3.1.7 Requisito funcional <0007>**

Categoria: Gestão de favoritos

Nome: Ver lista de favoritos

Id: CF – RF – 07

Descrição: Permite ver a lista de livros adicionados aos favoritos

Restrições:

Prioridade: 1

Estado: Fixo

Verificações: Utilizador é remetido para uma página com a lista dos livros favoritos

### **3.1.8 Requisito funcional <0008>**

Categoria: Gestão de favoritos

Nome: Remover livro da lista de favoritos

Id: CF – RF – 08

Descrição: Permite remover um livro da lista de favoritos.

Restrições: Livro deve constar da lista de favoritos.

Prioridade: 1

Estado: Fixo

Verificações: Livro é removido da lista de favoritos



### **3.1.9 Requisito funcional <0009>**

Categoria: Gestão de livros

Nome: Adicionar livro

Id: CL – RF – 09

Descrição: Permite adicionar um livro ao sistema.

Restrições: Livro deve conter nome, autor, descrição do estado e pelo menos uma foto.

Prioridade: 1

Estado: Fixo

Verificações: Livro é adicionado à lista de livros do utilizador

### **3.1.10 Requisito funcional <0010>**

Categoria: Gestão de livros

Nome: Remover livro

Id: CL – RF – 10

Descrição: Permite remover um livro da lista de livros do utilizador.

Restrições: Livro deve constar da lista de livros do utilizador.

Prioridade: 1

Estado: Fixo

Verificações: Livro é removido da lista de livros do utilizador.

### **3.1.11 Requisito funcional <0011>**

Categoria: Gestão de livros

Nome: Editar informação de livros

Id: CL – RF – 11

Descrição: Permite editar a informação de um livro na lista de livros do utilizador.

Restrições: Livro deve conter nome, autor, descrição do estado e pelo menos uma foto.

Prioridade: 1

Estado: Fixo

Verificações: Informação do livro é alterada

### **3.1.12 Requisito funcional <0012>**

Categoria: Gestão de livros

Nome: Ver lista de livros

Id: CL – RF – 12

Descrição: Permite ver os livros encontrados na lista de livros do utilizador.

Restrições:

Prioridade: 1

Estado: Fixo

Verificações: Utilizador é remetido para uma página com a sua lista de livros

### **3.1.13 Requisito funcional <0013>**

Categoria: Sistema de procura

Nome: Procura de livros

Id: CSP – RF – 13

Descrição: Permite a procura de livros.

Restrições :

Prioridade: 1

Estado: Fixo

Verificações: Utilizador é remetido para uma página com os resultados da procura

### **3.1.14 Requisito funcional <0014>**

Categoria: Sistema de procura

Nome: Procura de users

Id: CSP – RF – 14

Descrição: Permite procurar users.

Restrições :

Prioridade: 1

Estado: Fixo

Verificações: Utilizador é remetido para uma página com os resultados da procura

### **3.1.15 Requisito funcional <0015>**

Categoria: Interações

Nome: Não voltar a mostrar livro

Id: CI – RF – 15

Descrição: Impede certo livro de voltar a aparecer na pesquisa.

Restrições :

Prioridade: 1

Estado: Fixo

Verificações: Livro não volta a aparecer na lista de pesquisa

### **3.1.16 Requisito funcional <0016>**

Categoria: Interações

Nome: Passar/Gostar de livro

Id: CI – RF – 16

Descrição: Permite passar ou gostar de livro (que o adiciona à lista de favoritos) na visão para swipe.

Restrições:

Prioridade: 1

Estado: Fixo

Verificações: Aplicação mostra o próximo livro ao utilizador, caso exista.

### **3.1.17 Requisito funcional <0017>**

Categoria: Interações

Nome: Avaliar utilizador

Id: CI – RF – 17

Descrição: Permite avaliar um utilizador.

Restrições: Um Utilizador só pode avaliar outro caso tenham tido um match entre eles.

Prioridade: 1

Estado: Fixo

Verificações: Avaliação é adicionada à média de avaliação do utilizador em questão

### **3.1.18 Requisito funcional <0018>**

Categoria: Interações

Nome: Bloquear utilizador

Id: CI – RF – 18

Descrição: Permite bloquear um utilizador.

Restrições:

Prioridade: 1

Estado: Fixo

Verificações: Utilizador nem os seus livros voltam a aparecer para utilizador que o bloqueou.

### **3.1.19 Requisito funcional <0019>**

Categoria: Interações

Nome: Reportar utilizador

Id: CI – RF – 19

Descrição: Permite reportar um utilizador enviando uma mensagem aos administradores da aplicação.

Restrições:

Prioridade: 1

Estado: Fixo

Verificações:

### **3.1.20 Requisito funcional <0020>**

Categoria: Interações

Nome: Match

Id: CI – RF – 20

Descrição: Funcionalidade que avisa dois utilizadores quando ambos têm livros pertencentes ao outro na sua lista de favoritos

Restrições:

Prioridade: 1

Estado: Fixo

Verificações: Cada utilizador recebe uma notificação a informá-los do match.

### **3.1.21 Requisito funcional <0021>**

Categoria: Interações

Nome: Sistema de envio de mensagens

Id: CI – RF – 21

Descrição: Permite a interação com outro utilizador através de mensagens.

Restrições: Deve ter ocorrido um match entre dois utilizadores para eles poderem trocar mensagens

Prioridade: 1

Estado: Fixo

Verificações: Nomes de utilizador aparecem na inbox dos dois.

## **3.2 Requisitos não funcionais**

### **3.2.1 Requisito não funcional <0022>**

Categoria: Interoperabilidade

Nome: Base de dados

Id: CU - RNF – 01

Descrição: Os dados deverão ser armazenados em SQL Server

Estado: Fixo

### **3.2.2 Requisito não funcional <0023>**

Categoria: Usabilidade

Nome: Fácil acesso

Id: CL - RNF – 02

Descrição: A aplicação deverá ser intuitiva, de maneira ao utilizador não necessitar de qualquer treino ou tutorial para tirar proveito desta.

Estado: Fixo

### **3.2.3 Requisito não funcional <0024>**

Categoria: Segurança

Nome: Segurança de informação

Id: CL - RNF – 03

Descrição: O sistema deve preservar a integridade da informação de conta de cada utilizador

Estado: Fixo

## 4. Mockups

### 4.1 Login



*Figura 2 - Mockup Login*

Categoria: Gestão de Utilizadores

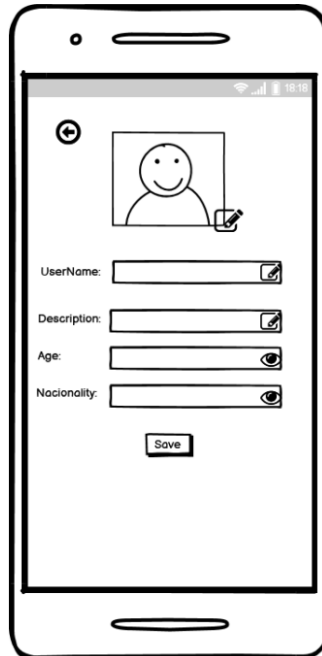
Nome: Login

Id: Mockup - Login

Descrição: Interface que permite a autenticação e o login na aplicação,

Estado: Fixo

## 4.2 Editar dados de conta



*Figura 3 - Mockup editar conta*

Categoria: Gestão de Utilizadores

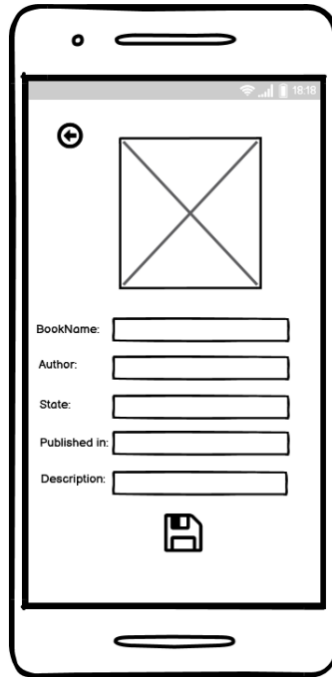
Nome: Editar dados de conta

Id: Mockup - Editar dados de conta

Descrição: Interface que permite ao utilizador editar os seus dados pessoais.

Estado: Fixo

### 4.3 Adicionar livro



*Figure 4 - Mockup Adicionar livro*

Categoria: Gestão de Livros

Nome: Adicionar livros

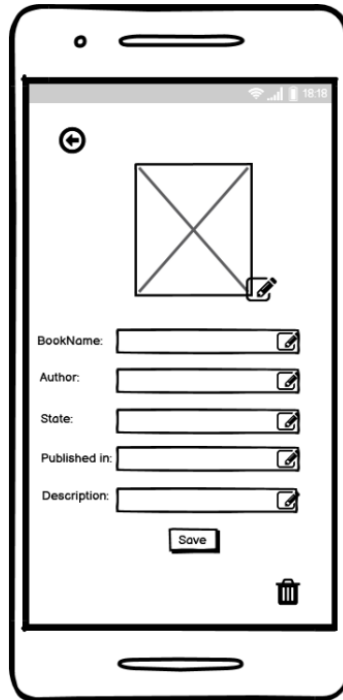
Id: Mockup – Adicionar livro

Descrição: Interface que permite ao utilizador adicionar um livro ao sistema.

Estado: Fixo



## 4.4 Editar livro



*Figure 5 - Mockup editar livro*

Categoria: Gestão de Livros

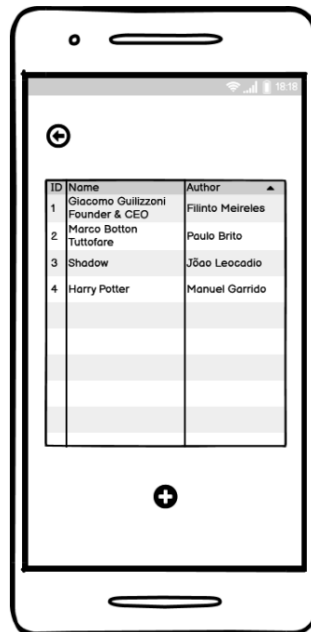
Nome: Editar livros

Id: Mockup – Editar livro

Descrição: Interface que permite ao utilizador editar um livro.

Estado: Fixo

## 4.5 Ver lista de livros



*Figure 6 - Mockup ver lista de livros*

Categoria: Gestão de Livros

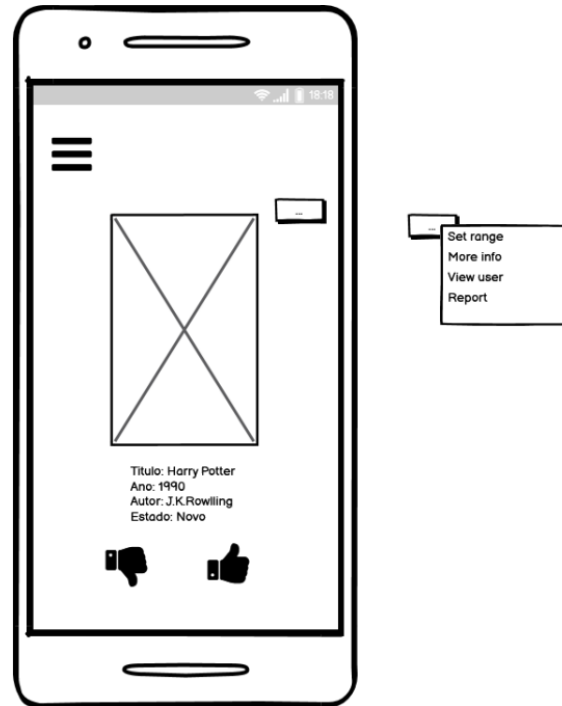
Nome: Listar livros

Id: Mockup – Listar livros

Descrição: Interface que permite ao utilizador ver a sua lista de livros.

Estado: Fixo

## 4.6 Sistema de Match



*Figure 7 - Mockup matchs*

Categoria: Interações

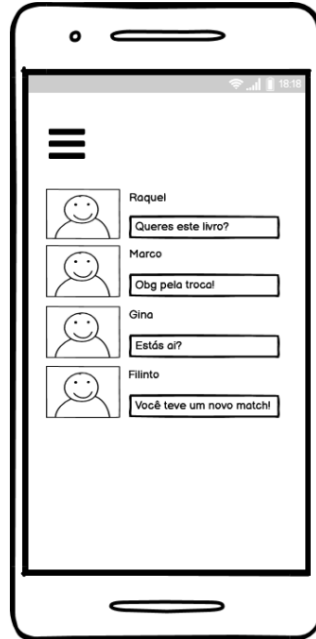
Nome: Match

Id: Mockup – Match

Descrição: Interface que permite ao utilizador ver livros e dar like ou dislike.

Estado: Fixo

## 4.7 Sistema de troca de mensagens



*Figure 8 - Mockup Mensagens*

Categoria: Interações

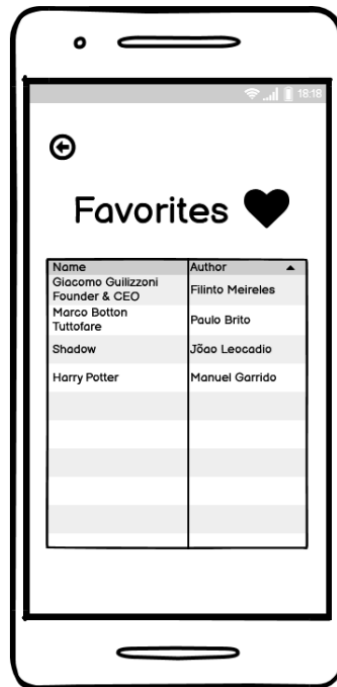
Nome: Chat

Id: Mockup – Chat

Descrição: Interface que permite ao utilizador ver e trocar mensagens com outros utilizadores.

Estado: Fixo

## 4.8 Lista de Favoritos



*Figure 9 - Mockup Lista de Favoritos*

Categoria: Gestão de favoritos

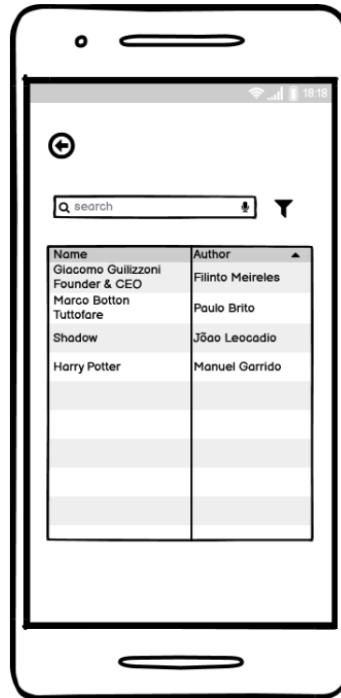
Nome: Ver lista de favoritos

Id: Mockup – Ver lista de favoritos

Descrição: Interface que permite listar todos livros na lista de favoritos

Estado: Fixo

## 4.9 Sistema de procura



*Figure 10 - Mockup Sistema de procura*

Categoria: Sistema de procura

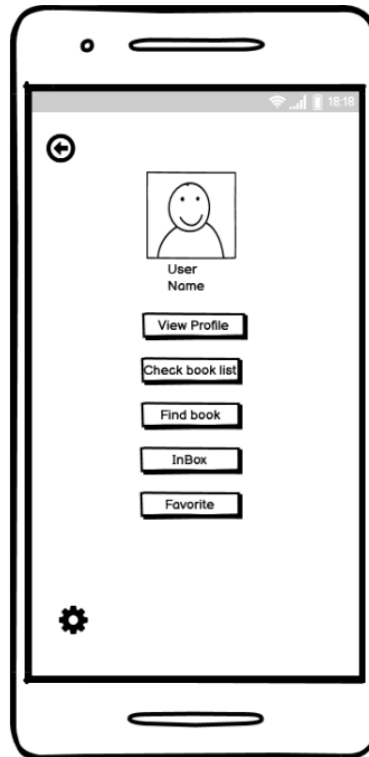
Nome: Pesquisa evoluída

Id: Mockup – Procurar livros através de filtros

Descrição: Interface que permite procurar livros através de filtros

Estado: Fixo

## 4.10 Dashboard da aplicação



*Figure 11 - Mockup Dashboard*

Categoria: Dashboard

Nome: Dashboard da aplicação

Id: Mockup – Menu da aplicação

Descrição: Dashboard

Estado: Fixo

## 5. Use Cases

### 5.1 Use Case 1 - Gestão de utilizadores

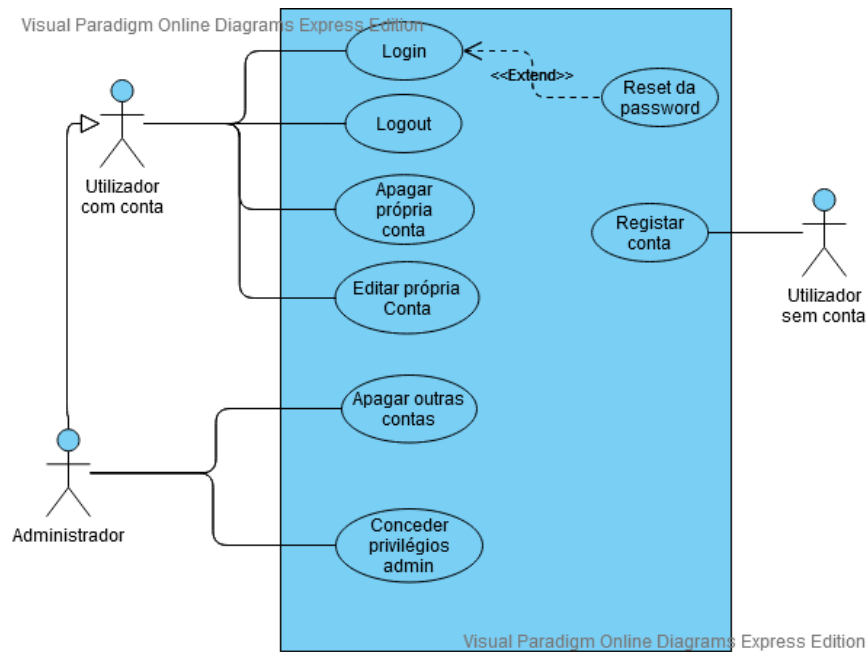


Figure 12 - Use Case gestão de Utilizadores

Gestão de utilizadores	Editar dados de conta
<b>Atores</b>	Utilizador
<b>Pré-Condições</b>	Login efetuado com sucesso
<b>Fluxo Básico</b>	1- O use case começa quando o Utilizador seleciona o botão de editar dados. 2- O Utilizador deverá preencher os campos em branco com a informação pedida relativa à edição de dados. 3- Posteriormente, o utilizador escolherá se pretende gravar (editar dados) ou então sair sem gravar e assim não editar a sua informação pessoal.
<b>Pós-Condições</b>	A Informação será atualizada no sistema.



Gestão de utilizadores	Remover conta
<b>Atores</b>	Utilizador
<b>Pré-Condições</b>	Login efetuado com sucesso
<b>Fluxo Básico</b>	1- O use case começa quando o Utilizador seleciona o botão de remover conta. 2- O Utilizador deverá preencher os campos em branco com a informação pedida relativa à remoção de conta.
<b>Pós-Condições</b>	A conta é apagada do sistema.

Gestão de utilizadores	Criar conta
<b>Atores</b>	Utilizador
<b>Fluxo Básico</b>	1- O use case começa quando o Utilizador seleciona o botão de criar conta. 2- O Utilizador deverá preencher os campos em branco com a informação pedida relativa à criação de conta. 3- O utilizador concluiu a criação de conta.
<b>Pós-Condições</b>	O novo utilizador será adicionado no sistema.

## 5.2 Use Case 2 - Interações

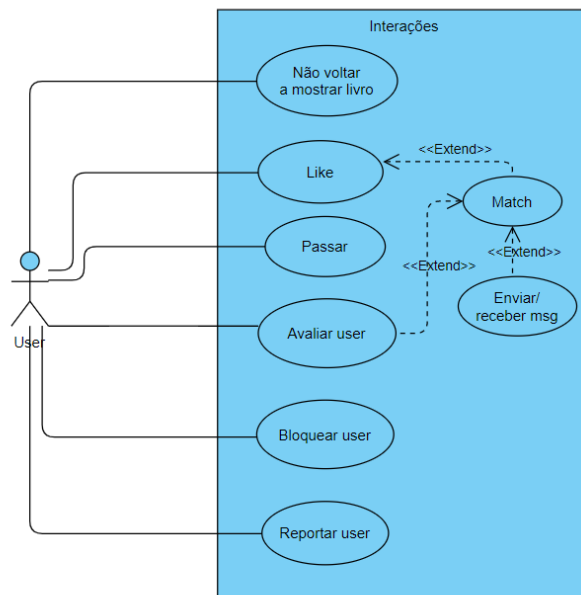


Figure 13 - Use Case Interações

Interações	Não voltar a mostrar livro
Atores	Utilizador
Fluxo Básico	1- O use case começa quando o Utilizador seleciona o botão de Não voltar a mostrar livro.
Pós-Condições	O utilizador bloqueou com sucesso o livro.

Interações	Passar
Atores	Utilizador
Fluxo Básico	1- O use case começa quando o Utilizador seleciona o botão de passar.
Pós-Condições	O utilizador passou com sucesso o livro.

Interações	Bloquear utilizador
Atores	Utilizador
Fluxo Básico	1- O use case começa quando o Utilizador seleciona o botão de bloquear utilizador
Pós-Condições	O utilizador bloqueou com sucesso o utilizador.

Interações		Reportar utilizador
<b>Atores</b>		Utilizador
<b>Fluxo Básico</b>		1- O use case começa quando o Utilizador seleciona o botão de reportar 2- O utilizador concluiu o reporte de utilizador.
<b>Pós-Condições</b>		O utilizador reportou com sucesso o utilizador.

### 5.3 Use case 3 – Sistema de procura

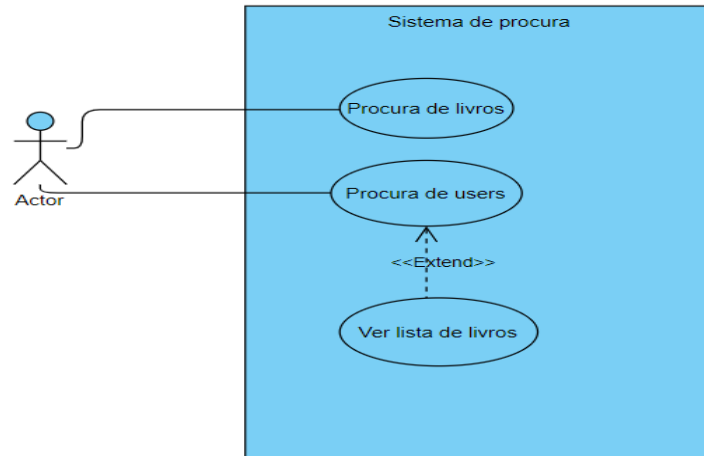


Figure 14 - Use Case Sistema de Procura

Sistema de procura	Procura de Livros
Atores	Utilizador;
Pré-Condições	O utilizador tem de efetuar login.
Fluxo Básico	1- O use case começa quando o utilizador seleciona o botão de procura de livros. 2- O utilizador procura o livro que desejar. 3- O utilizador conclui a procura do livro.

Sistema de procura	Procura de Utilizadores
Atores	Utilizador;
Pré-Condições	O utilizador tem de efetuar login.
Fluxo Básico	1- O use case começa quando o utilizador seleciona o botão de procura de utilizadores. 2- O utilizador procura o utilizador que desejar. 3- O utilizador conclui a procura do utilizador.

## 5.4 Use Case 4 - Gestão de livros

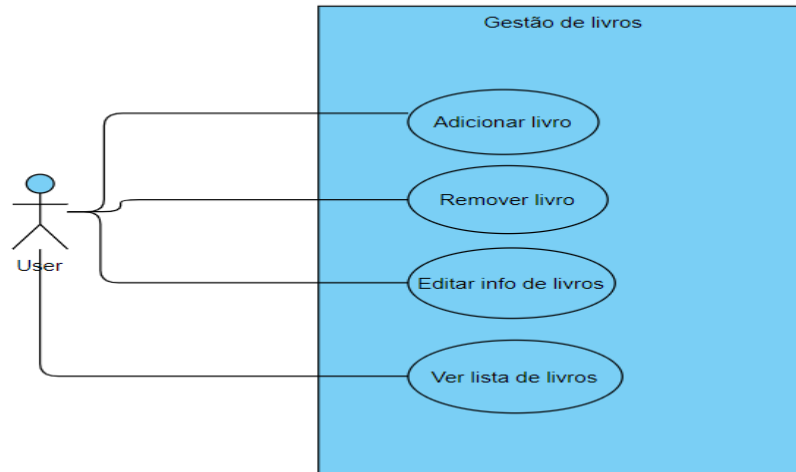


Figure 15 - Use Case Gestão de livros

Gestão de livros	Adicionar livros
Atores	Utilizador
Fluxo Básico	1- O use case começa quando o Utilizador seleciona o botão de adicionar livro 2- O utilizador preenche a informação do livro que pretende adicionar.
Pós-Condições	O livro é adicionado ao sistema.

Gestão de livros	Remover livros
Atores	Utilizador
Fluxo Básico	1- O use case começa quando o Utilizador seleciona o botão de Remover livro 2- O utilizador preenche a informação do livro que pretende remover.
Pós-Condições	O livro é removido do sistema.

Gestão de livros	Editar informação de livros
<b>Atores</b>	Utilizador
<b>Fluxo Básico</b>	1- O use case começa quando o Utilizador seleciona o botão de editar info de livros 2- O utilizador preenche a informação do livro que pretende editar 3- A informação é atualizada
<b>Pós-Condições</b>	A informação editada é atualizada no sistema.

Gestão de livros	Ver lista de livros
<b>Atores</b>	Utilizador
<b>Fluxo Básico</b>	1- O use case começa quando o Utilizador seleciona o botão de ver lista de livros. 2- A lista é apresentada

## 5.5 Use Case 5 - Gestão de Favoritos

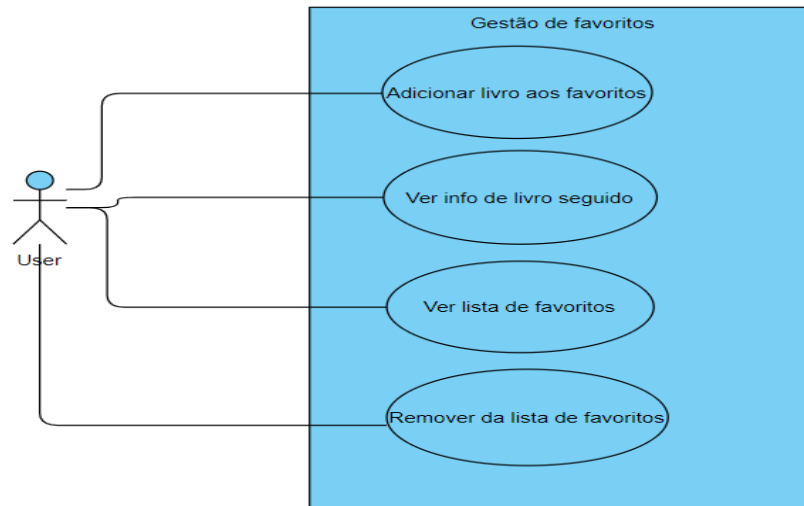


Figure 16 - Use Case gestão de favoritos

Gestão de favoritos	Adicionar livro aos favoritos
Atores	Utilizador
Fluxo Básico	1- O use case começa quando o utilizador seleciona o botão de adicionar livro aos favoritos. 2- O livro é adicionado aos favoritos.
Pós-Condições	O livro é adicionado à lista de favoritos do utilizador.

Gestão de favoritos	Ver informação de livro seguido
Atores	Utilizador
Fluxo Básico	1- O use case começa quando o Utilizador seleciona o botão de ver informação de livro seguido. 2- A informação pretendida é mostrada ao utilizador.
Pós-Condições	A informação é apresentada ao utilizador.

<b>Gestão de favoritos</b>	<b>Ver lista de favoritos</b>
<b>Atores</b>	Utilizador
<b>Fluxo Básico</b>	1- O use case começa quando o Utilizador seleciona o botão de ver lista de favoritos. 2- A informação pretendida é mostrada ao utilizador.
<b>Pós-Condições</b>	A informação é apresentada ao utilizador.

<b>Gestão de favoritos</b>	<b>Remover da lista de favoritos</b>
<b>Atores</b>	Utilizador
<b>Fluxo Básico</b>	1- O use case começa quando o Utilizador seleciona o botão de remover da lista de favoritos. 2- O utilizador escolhe o livro que pretende remover da lista. 3- O livro é removido da lista de favoritos.
<b>Pós-Condições</b>	O livro é removido da lista de favoritos.



## **6. Explicação Geral do funcionamento da aplicação**

### **6.1 Visão Geral**

A Rebook é uma aplicação focada na troca de livros usados. Após registo, um utilizador pode adicionar à sua lista os livros que tem para troca, cada livro deve ter nome, autor e uma avaliação do seu estado (1 para muito mau estado, 5 para novo), assim como uma ou mais fotos.

Por outro lado, um utilizador pode ver os livros de outros utilizadores no seu range através da Swipe View, se gostar de um livro este é adicionado à sua lista de favoritos, se o dono desse livro tiver um dos livros do utilizador na sua lista de favoritos ocorre um match abrindo a funcionalidade de mensagens entre utilizadores de maneira a que estes possam combinar uma troca de livros.

Se o utilizador não gostar do livro este não volta a aparecer na Swipe View, pode no entanto encontrá-lo recorrendo à funcionalidade de "Search Book".

Outra funcionalidade inerente à aplicação passa por poder avaliar outro utilizador. Isto cria um sistema de reputação através do qual podemos definir se um utilizador é digno de confiança e se é seguro trocar livros com ele.

Um utilizador pode também submeter um report a outro utilizador, assim como bloqueá-lo.

### **6.2 Estrutura da aplicação**

O back end da aplicação foi construído em Asp.Net core, usando como base de dados MySQL ao qual se liga através de EF Core. Através das funções presentes em Data regulamos o acesso à base de dados, e através dos controllers regulamos a comunicação com o front end.

O front end foi construído em Xamarin.Forms. De momento apenas está a ser considerado o deployment da aplicação em android, estando em aberta a possibilidade de deployment para iOS no futuro.

### **6.3 Modelo de dados.**

#### **As nossas entidades base são:**

- - Books: Para guardar livros
- - Users: Para guardar dados de utilizador
- - Reports: Para guardar reports
- - Messages: Para guardar mensagens.

#### **Temos entidades cujo propósito é relacionar estas entidades:**

- - BooksLiked: Associa utilizador a livros que gostou
- - UsersRated: Associa utilizador a outro que este avaliou
- - Matches: Associa dois utilizadores e verifica se há match

## 6.4 Autenticação JWT (Json Web Token)

A segurança é um dos aspetos mais importantes nas aplicações. Por essa razão, utilizou-se o método JWT (Json Web Token) que permite a autenticação entre duas partes por meio de um token que valida uma requisição web. Esse token, não é mais que um simples código em Base64 que permite armazenar objetos JSON juntamente com os dados que permitem a autenticação da requisição.

Na figura seguinte, é possível ver um cliente a enviar no corpo do request HTTP dados como endereço de e-mail e senha ao endpoint de autenticação da API.

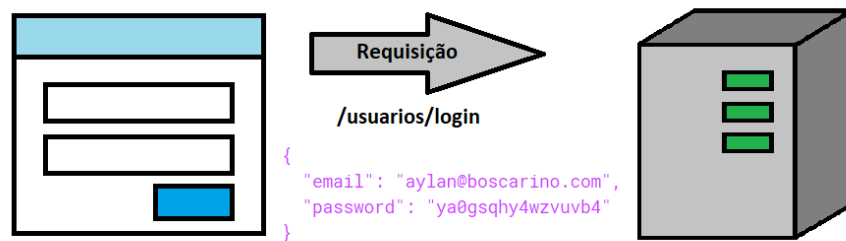


Figure 17 - Autenticação via JWT

Depois dos dados enviados pelo cliente serem devidamente autenticados pelo servidor, é criado um token JWT assinado com um segredo interno da API e enviará este token de volta ao cliente, como podemos ver na imagem seguinte.



Figure 18 - Autenticação via JWT

Depois de estar munido com o token autenticado, o cliente detém acesso aos endpoints da aplicação que antes não tinha. Para ocorrer esse acesso, é necessário informar esse token no header, e por convenção, após a palavra Bearer.

## 6.5 Ferramentas e Metodologias utilizadas

### 6.5.1 JetBrains Rider

O JetBrains Rider foi o ambiente de desenvolvimento (IDE) eleito para a elaboração do software. Na próxima figura, podemos observar o IDE e também um trecho de código produzido para este projeto.

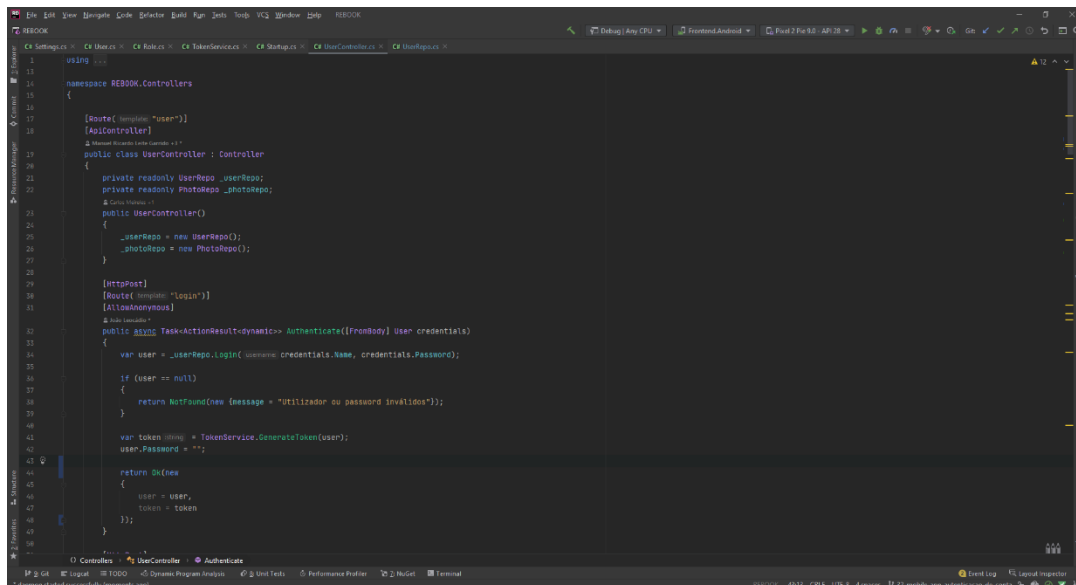


Figure 19 - JetBrains Rider (IDE)

### 6.5.2 Framework ASP.NET Core

ASP.NET Core é um framework de desenvolvimento web com vista ao desenvolvimento de aplicações web. Foi desenvolvido pela Microsoft, e por ser leve, rápido, modular e funcionar em conjunto com .NET Core tornou-se num framework popular entre a comunidade.

Por ser uma plataforma totalmente modular, possui também vários recursos que podem ser adicionados através de pacotes Nuget, o que permite à aplicação ser mais performática. Para além de tudo isto, o ASP.NET Core não necessita de um IDE específico, o que permite uma fácil portabilidade sem correr o risco de perder algum recurso ou funcionalidade.



Figure 20 - ASP.NET Core

### 6.5.3 Gitlab

O GitLab é um repositório de software baseado em git, com suporte a Wiki, gestor de tarefas e possui CI (Continuous Integration) e também CD (Continuous Delivery). Deste modo, o GitLab é utilizado para controlo de versões.

No GitLab está presente uma wiki com a documentação do projeto, e será utilizado como repositório local no computador de cada Dev (desenvolvedor), permitindo a submissão das alterações quando desejado e trabalho independentemente do repositório online. A figura , mostra o ambiente do Gitlab mais propriamente a Wiki do projeto.

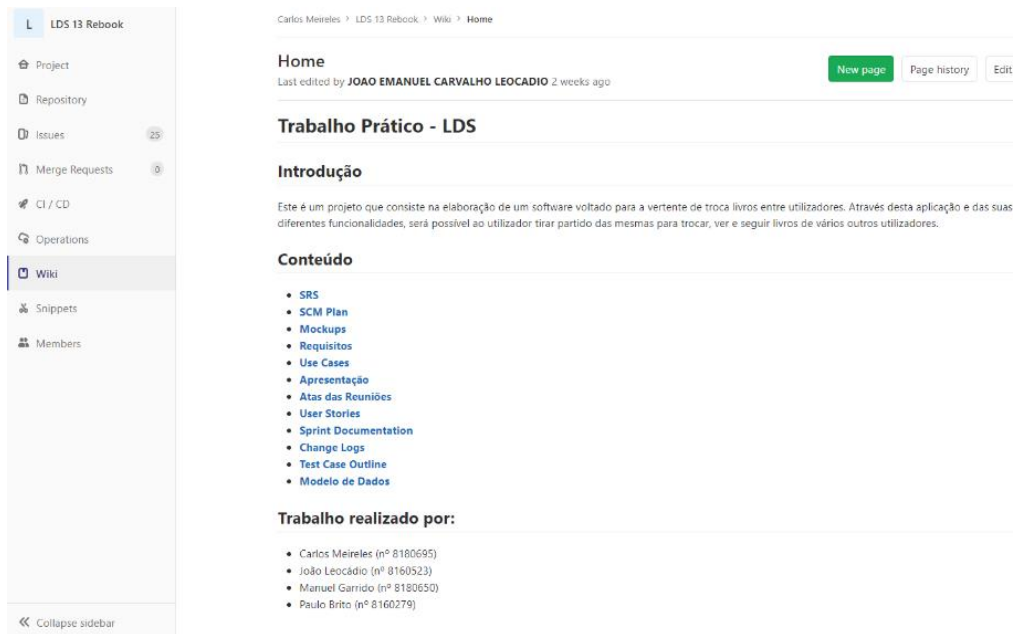


Figure 21 - GitLab ESTG

#### 6.5.4 Documentação Swagger

Ter uma boa documentação da REST API é primordial para os desenvolvedores e testers compreenderem claramente o comportamento dela, pois são estas as responsáveis por desempenhar um papel fundamental na comunicação entre as aplicações.

O conceito de REST aplica-se ao desenvolvimento de aplicações que comunicam através de um serviço ou servidor por via WEB. Deste modo, REST API usa o protocolo HTTP (*HyperText Transfer Protocol*) utilizando os métodos POST(inserir), DELETE(Apagar), PUT(Editar) dados no servidor ou serviços.

Em suma, o Swagger foi a framework utilizada para a documentação da API do projeto. Assim, é possível vislumbrar de forma mais fácil a descrição, consumos e visualização de serviços da nossa REST API, como se pode observar na próxima figura.

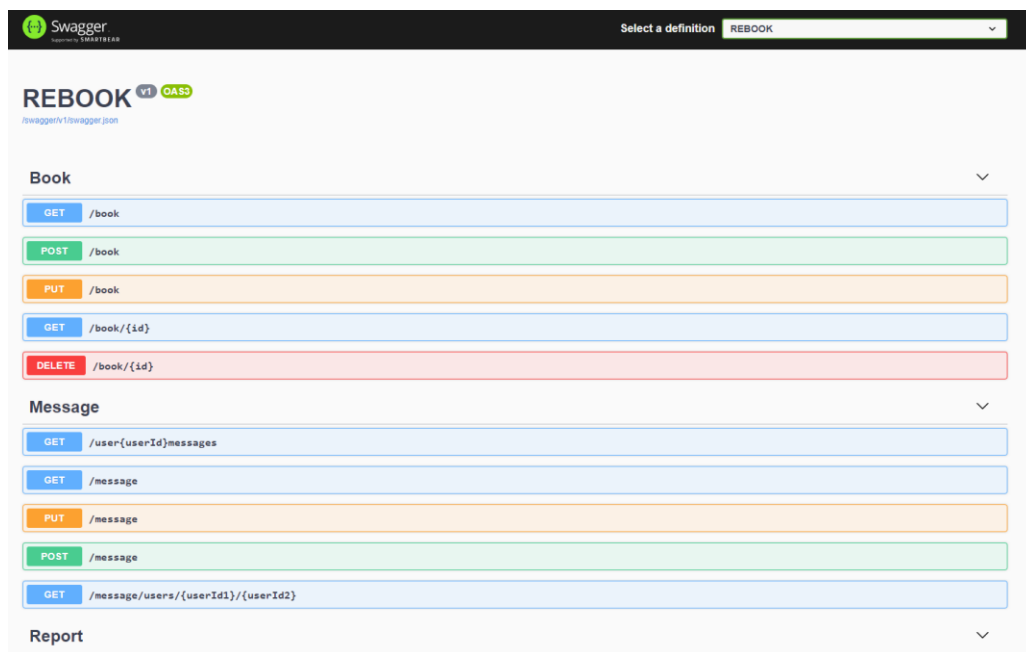


Figure 22 - Documentação Swagger

## 7. Arquitetura do Projeto

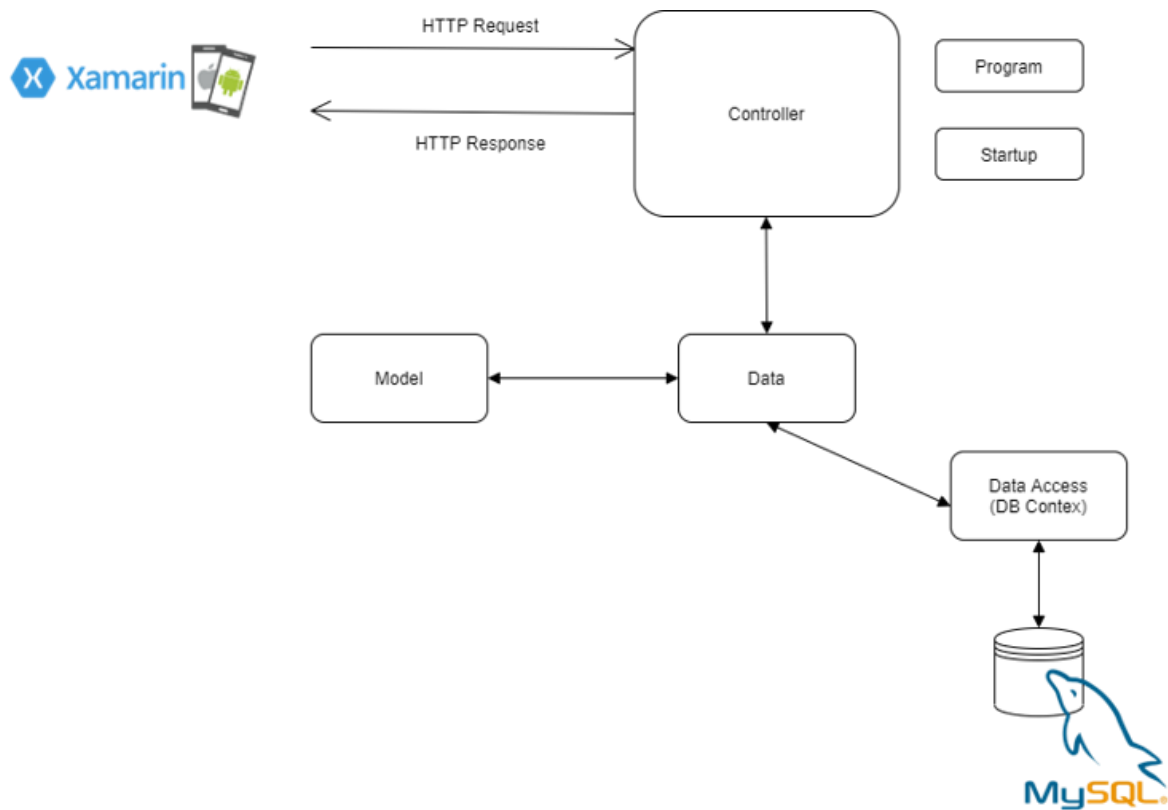


Figure 23 - Arquitetura do Projeto

Nesta imagem, é possível ver o desenho da arquitetura para a aplicação desenvolvida. No componente Data é onde se encontram as funções desenvolvidas para ler e escrever da base de dados. Os controllers regulam a comunicação com o frontend executando as funções que estão presentes no data dependendo dos HTTP requests. Por sua vez, no componente model é onde definimos as entidades associadas ao projeto. Por último, o frontend (Xamarin) consumirá todos os pedidos feitos ao servidor.

## 8. Diagrama de Classes

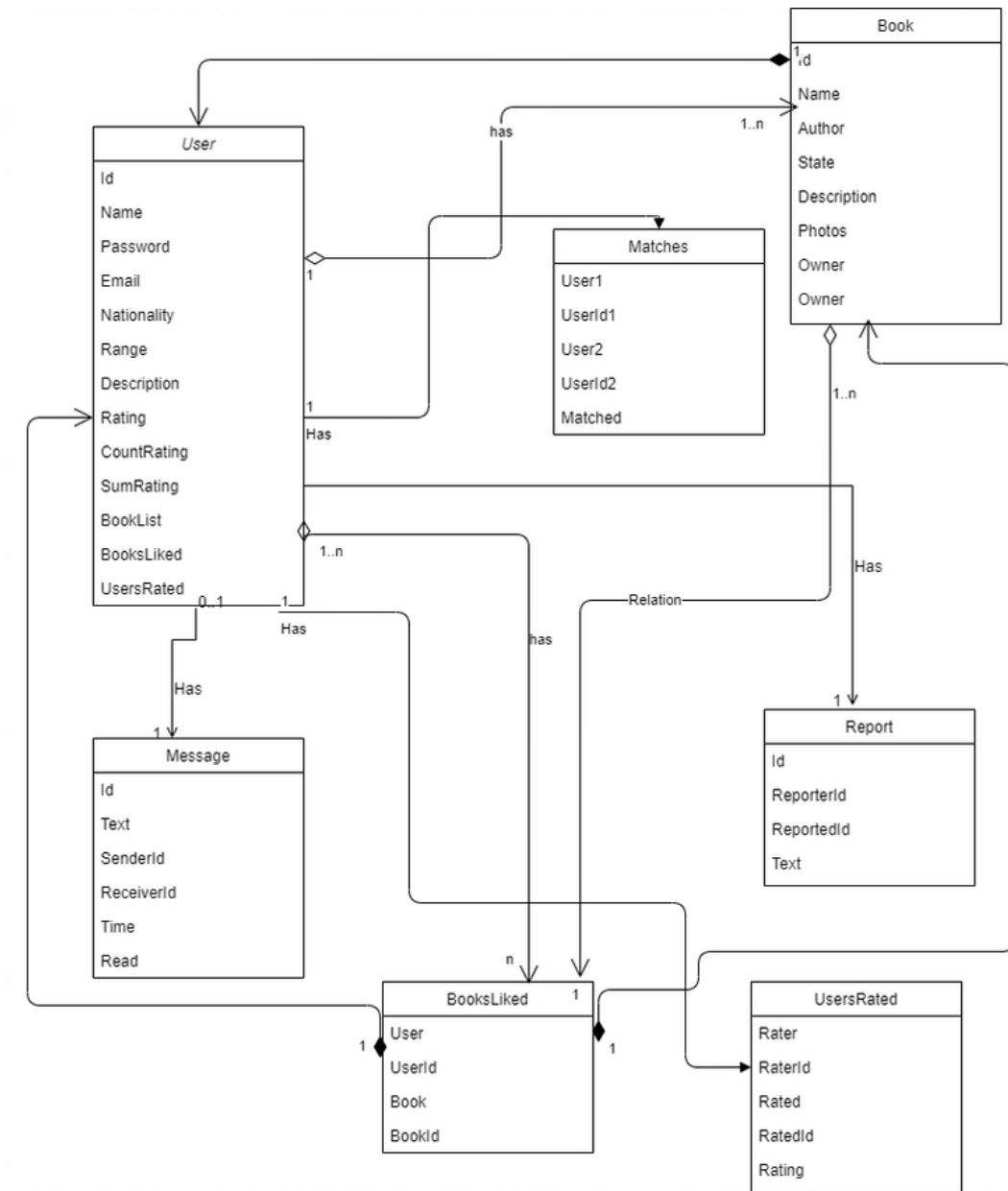


Figure 24 - Diagrama de Classes

## **9. Conclusão e limitações**

Este projeto, teve como principal objetivo explorar os conceitos lecionados ao longo do curso como metodologias de testes, metodologias de desenvolvimento ágeis, Continuous Integration (CI) e planos de Software Configuration Management (SCM).

Assim e resumindo, foi elaborado uma aplicação mobile que funcionará na plataforma Android e permitirá a troca de livros entre vários utilizadores, ou seja, o utilizador consegue ver vários livros e poderá dar “like” ou passar com o intuito de haver um match.

Deste modo e através do desenvolvimento desta aplicação, pode-se afirmar que expandimos as nossas respetivas competências de análise de um problema, identificação de requisitos, planear e implementar um plano de testes. Utilizar metodologias de desenvolvimento ágil como o SCRUM no desenvolvimento de software, usar ferramentas de suporte (ASP NET core, Swagger, Xunit, Gitlab) e perceber a importância de trabalhar em equipa assim como manter um ambiente saudável entre todos.

Neste contexto é de realçar todo o conhecimento transmitido pelos docentes da unidade curricular, que grande importância terá no futuro e que tão rico tornou o presente.

No que diz respeito às limitações sentidas ao longo de todo o percurso de desenvolvimento deste projeto, pode-se referir que foram algumas embora não impediram o bom desenvolvimento do mesmo.

A primeira limitação surgiu logo no início, que ficou marcado por algum receio, por desconhecimento dos métodos usados no trabalho, tais como, dotnet e ef core. Uma vez que nunca haviam sido utilizados outrora.

Na elaboração deste relatório, a principal limitação sentida prendeu-se com a gestão do tempo disponível para o mesmo.

Outras limitações ocorreram durante o decorrer do desenvolvimento do projeto, mas que rapidamente foram ultrapassadas graças ao espírito de equipa e companheirismo que é comum neste grupo de trabalho.