

CENTRALMUSIC

REPORT

Version 1.3

18/02/2021

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Samuel Cunha	18/02/2021	Samuel Cunha	18/02/2021	Introdução, âmbito e introdução das ferramentas
1.1	Samuel Cunha	18/02/2021	Samuel Cunha	18/02/2021	Finalização do tópico de ferramentas e âmbito
1.2	Samuel Cunha	18/02/2021	Samuel Cunha	18/02/2021	Introdução das arquiteturas e metodologias
1.3	Samuel Cunha	18/02/2021	Samuel Cunha Marcelo Carvalho	18/02/2021	Revisão do documento para a entrega 2

Índice

1	Introdução.....	5
2	Âmbito	5
3	Arquitetura	6
3.1	Padrão DAO na API	6
4	Ferramentas	8
4.1	Git / GitLab.....	8
4.2	GitLab Wiki / Documentação	9
4.3	.Net 5 e <i>Visual Studio</i>	9
4.3.1	Testes	10
4.4	Postman	10
4.5	Azure	11
4.6	Flutter SDK e Dart	11
4.7	Android Studio.....	11
5	Metodologia	12
5.1	SCRUM	12
5.1.1	Sprint #1	14
5.1.2	Sprint #2	14
5.1.3	Sprint #3	14
6	Conclusão.....	15

Índice de figuras

Figura 1 – Arquitetura de alto nível	7
Figura 2 -.gitingore	8
Figura 3 - Estrutura do projeto.....	9
Figura 4 - Exemplo de testes	10
Figura 5 - postman logo.....	10
Figura 6 - Azure Logo.....	11
Figura 7 - Flutter e Dart	11
Figura 8 - Android Studio	11
Figura 9 - Board Backlogs	12
Figura 10 - Board Development	13

1 Introdução

Esta secção contém a introdução deste documento. Este documento destina-se a descrever o uso e identificar, as ferramentas usadas para o desenvolvimento do projeto de Laboratório de Desenvolvimento de Software.

Na secção 2 será mencionado o âmbito em que este trabalho se enquadra. Na secção 3 é apresentada uma arquitetura de alto nível da aplicação. Na secção 4 serão identificadas as ferramentas usadas e como é que foram utilizadas no âmbito da metodologia SCRUM. Na secção 5 será explicado de que modo é que a metodologia *SCRUM* foi aplicada usando o *GitLab* da *ESTG*.

2 Âmbito

Este projeto desenvolvido no âmbito da disciplina de LDS, dedica-se ao desenvolvimento de uma aplicação (mobile), com o objetivo de troca e venda de instrumentos musicais. Este sistema será composto por uma aplicação para dispositivos móveis desenvolvida em Flutter. Para a disponibilização dos serviços de negócio, irá ter uma aplicação no *backend* uma *API* desenvolvida em ASP.net core.

Para mais informações sobre os requisitos do software, veio em conjunto com este documento:

- Software Requirements Specification

Junto desta documentação, vem também em anexo, diagramas representantes da arquitetura e cenários a que se aplicam as funcionalidades, isto é:

- Diagramas de Classes
- Diagramas de Componentes
- Diagramas de Use Case

Com este projeto prático, também se pretende que sejam demonstradas competências relativamente ao desenvolvimento ágil de software, com foco em práticas de garantia de qualidade de software, nomeadamente, testes, *Software Configuration Management (SCM)* e uso de padrões de segurança. Pretende-se que seja seguida uma abordagem baseada em *Continuous Integration*, procurando suportar e automatizar o mais possível o processo de construção de software, usando também metodologias ágeis, tal como já dito anteriormente, *SCRUM*.

3 Arquitetura

3.1 Padrão DAO na API

O padrão DAO (*Data Access Object*) é usado para separar dados de baixo nível que a API usa ao efetuar operações na base de dados, das operações de serviços de negócios de alto nível. Com este padrão podemos escolher que informação apresentar ao exterior e privar o exterior de ver informação indesejada.

Como consta na figura a seguir, os *controllers* da API recebem os pedidos da app do utilizador. Nestes pedidos pode vir informação correspondente aos modelos que são expostos, que depois são mapeados para entidades internas a que o utilizador não tem acesso.

A partir da informação destas entidades são feitas as operações na base de dados. As entidades e os modelos são os nossos *Data transfer objects* (DTOs).

Para enviar informação de volta ao utilizador, as entidades provenientes da BD são mapeadas para os modelos a serem expostos ao exterior, que por sua vez são enviados pelos *controllers* de volta à aplicação do utilizador.

Na figura seguinte é possível perceber melhor este padrão, através do diagrama de arquitetura:

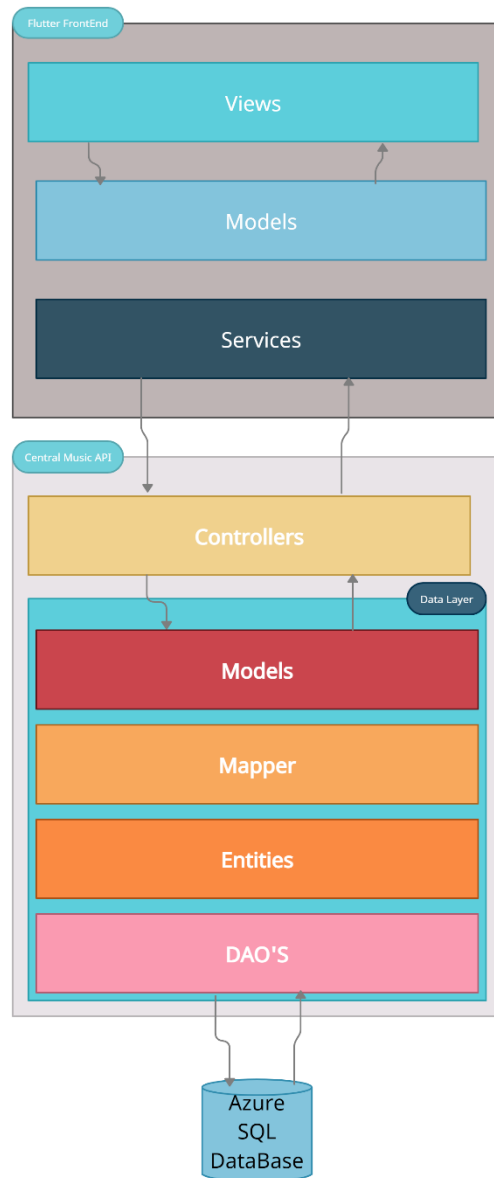


Figura 1 – Arquitetura de alto nível

4 Ferramentas

4.1 Git / GitLab

O projeto desenvolvido foi gerido usando o GitLab da ESTG: <https://gitlab.estg.ipp.pt/8160526/centralmusic>.

Quando o repositório foi criado, logo de seguida foi colocado um ficheiro “*.gitignore*”, de modo a que só fosse para o repositório o código fonte, os testes e outros ficheiros que não fossem gerados pelos *IDEs* usados pelos membros dos grupos. O ficheiro “*.gitignore*” foi criado usando a ferramenta: <http://www.gitignore.io/>, com as seguintes opções:

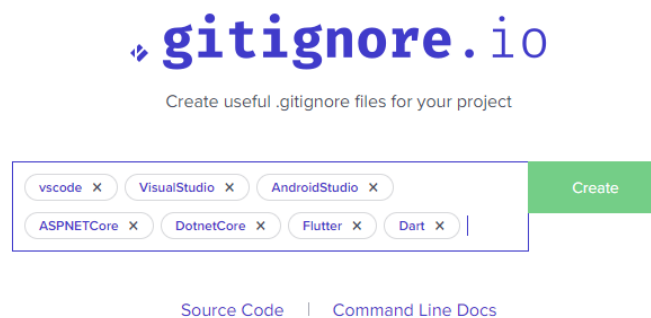


Figura 2 -.gitingore

Depois de criado o repositório com o projeto base, e com o “*.gitignore*”, este estava pronto a ser usado.

A gestão deste projeto foi feita usando, a técnica de *branching* de *GitFlow*. Existe uma *branch* “*master*” que só mantém o código estável, normalmente código proveniente de *releases* estáveis. Existe também uma *Branch* chamada “*Develop*”, que mantém o código em desenvolvimento. Quando foram feitas modificações ao projeto ou desenvolvimento de *features*, estas eram feitas em *branches* originadas a partir da *Develop*, e quando era dado o *merge*, as *branches* eram apagadas e os *commits* passados para a *target branch*.

4.2 GitLab Wiki / Documentação

A documentação foi colocada na *wiki* do *GitLab* recorrendo a ajuda a *snippets*. Os *snippets* serviram para anexar ficheiros, usar os links de download dos mesmos na *wiki*. Os ficheiros anexados foram: *SCM Plan*, *Test Design Specification*, *Test Case Outline*, *todos os mockups*, *todas as atas*, *os diagramas de classes e de arquitetura* e este relatório.

Wiki: <https://gitlab.estg.ipp.pt/8160526/centralmusic/wikis/home>

4.3 .Net 5 e Visual Studio

Este projeto foi desenvolvido utilizando *dotNet5* e o *Visual Studio*. Foi com estas ferramentas que também construímos e executamos os testes numa *API* de testes dentro da mesma solução como consta nas seguintes figuras:

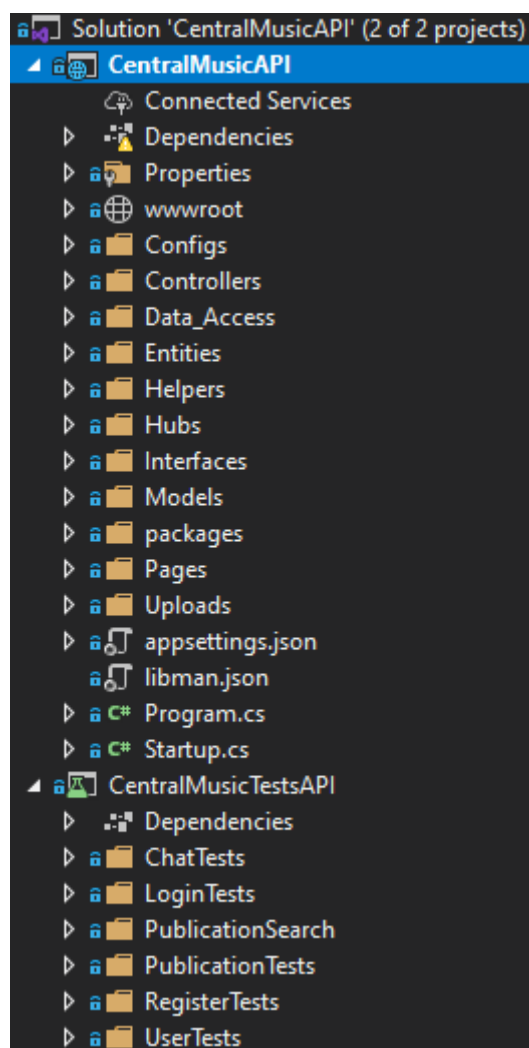
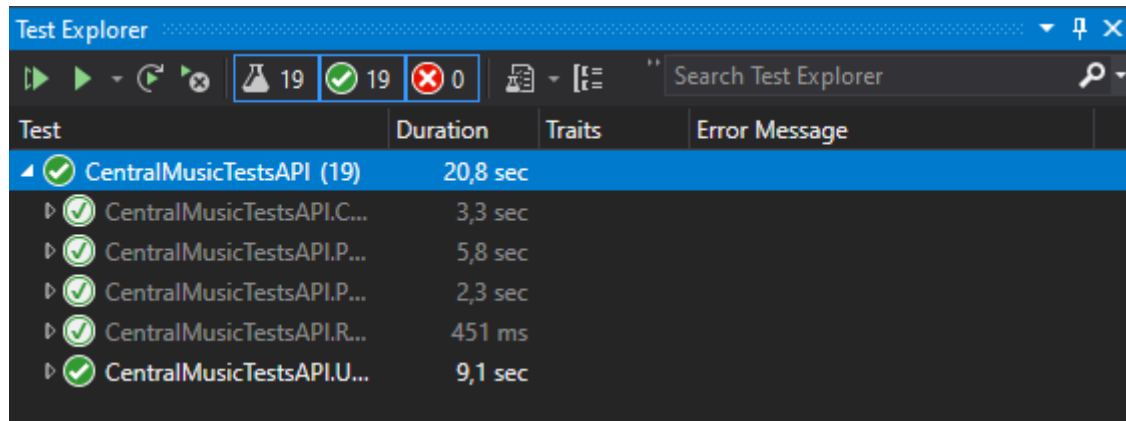


Figura 3 - Estrutura do projeto

4.3.1 Testes

Os testes foram realizados usando a ferramenta *xUnit* para *dotNet* e foi usado o visual studio para recolher informação de *code coverage* como consta nas figuras abaixo, para os testes da funcionalidade em tempo real de Chat foi utilizado uma página Html para simular a conversa entre dois utilizadores.



The screenshot shows the Test Explorer window with a blue header. Below the header, there are icons for test status: a green checkmark for 19 tests, a red X for 0 tests, and a green checkmark for 19 tests. A search bar labeled 'Search Test Explorer' is on the right. The main table lists tests with columns for Test, Duration, Traits, and Error Message.

Test	Duration	Traits	Error Message
CentralMusicTestsAPI (19)	20,8 sec		
CentralMusicTestsAPI.C...	3,3 sec		
CentralMusicTestsAPI.P...	5,8 sec		
CentralMusicTestsAPI.P...	2,3 sec		
CentralMusicTestsAPI.R...	451 ms		
CentralMusicTestsAPI.U...	9,1 sec		

Figura 4 - Exemplo de testes

Foi realizada documentação para todos os testes feitos:

- Test Design Specification
- Test Case Outline
- Teste de Gestão de imagens

Estes dois documentos foram entregues juntamente com este relatório.

Nota: A figura 3 não representa o estado final dos testes, pois ainda podem surgir mais funcionalidades.

4.4 Postman

O Postman foi a ferramenta usada para testar todas as rotas que foram feitas ao longo do trabalho, de modo a verificar o sucesso da implementação das mesmas.



<https://www.postman.com/>

Figura 5 - postman logo

4.5 Azure

A plataforma *Azure* foi a ferramenta utilizada para hospedar a base de dados do sistema, usando os créditos disponíveis no azure para estudantes. Para além da base de dados, foi usado o Azure DevOps para executar as pipelines relativas ao trabalho, e correr testes, visto que as pipelines do gitlab da ESTG estavam com problemas.



<https://azure.microsoft.com/pt-pt/>

Figura 6 - Azure Logo

4.6 Flutter SDK e Dart

Para o desenvolvimento da aplicação mobile, está a ser utilizada a framework flutter para que a aplicação pudesse ser desenvolvida tanto para android, como para iOS, usando a linguagem de programação da google, o Dart.



<https://flutter.dev/>

Figura 7 - Flutter e Dart

4.7 Android Studio

Como IDE para a implementação da plataforma mobile, foi utilizado o Android Studio, que nos permitiu também testar a aplicação com recurso a um emulador.



<https://developer.android.com/studio>

Figura 8 - Android Studio

5 Metodologia

5.1 SCRUM

A metodologia ágil usada no desenvolvimento deste produto é *SCRUM*. Ao seguir a metodologia *SCRUM* desenvolvemos o software de uma forma iterativa, usando *sprints*. No caso deste trabalho, os *sprints* foram mencionados no *Gitlab* nas *milestones*. Estes *sprints* foram iniciados depois de um *sprint planning*, uma reunião entre os membros do grupo que definiram um *backlog* priorizado para a equipa de desenvolvimento implementar no trabalho. *Sprint review* também foi feito de modo a perceber o que foi feito ao longo do tempo até a um determinado sprint, para perceber o que se teve de alterar / corrigir, como por exemplo *bugs* ou *features*, que poderiam ser úteis para englobar no planeamento do *sprint* seguinte.

No *gitlab* criamos 2 boards nas *issues* – (*Backlogs e Development*):

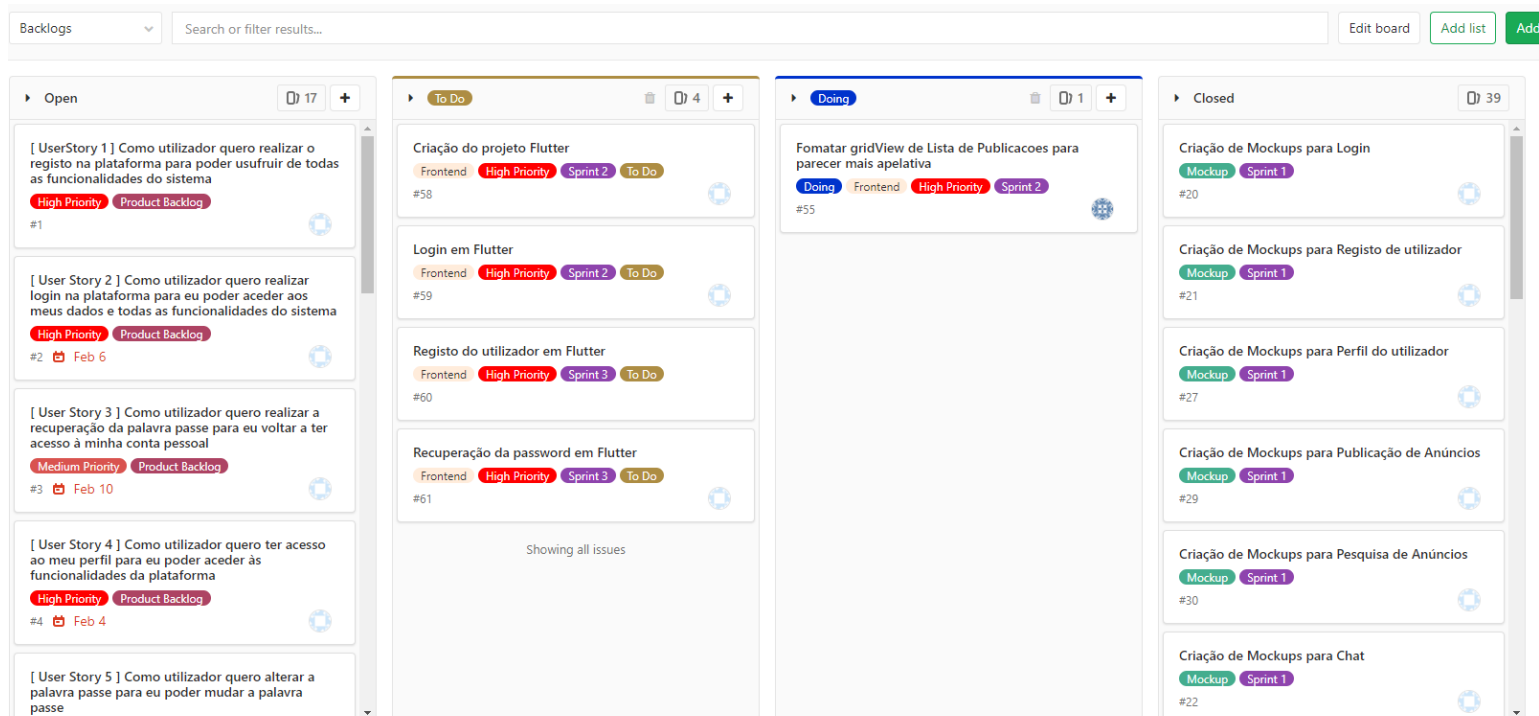


Figura 9 - Board Backlogs

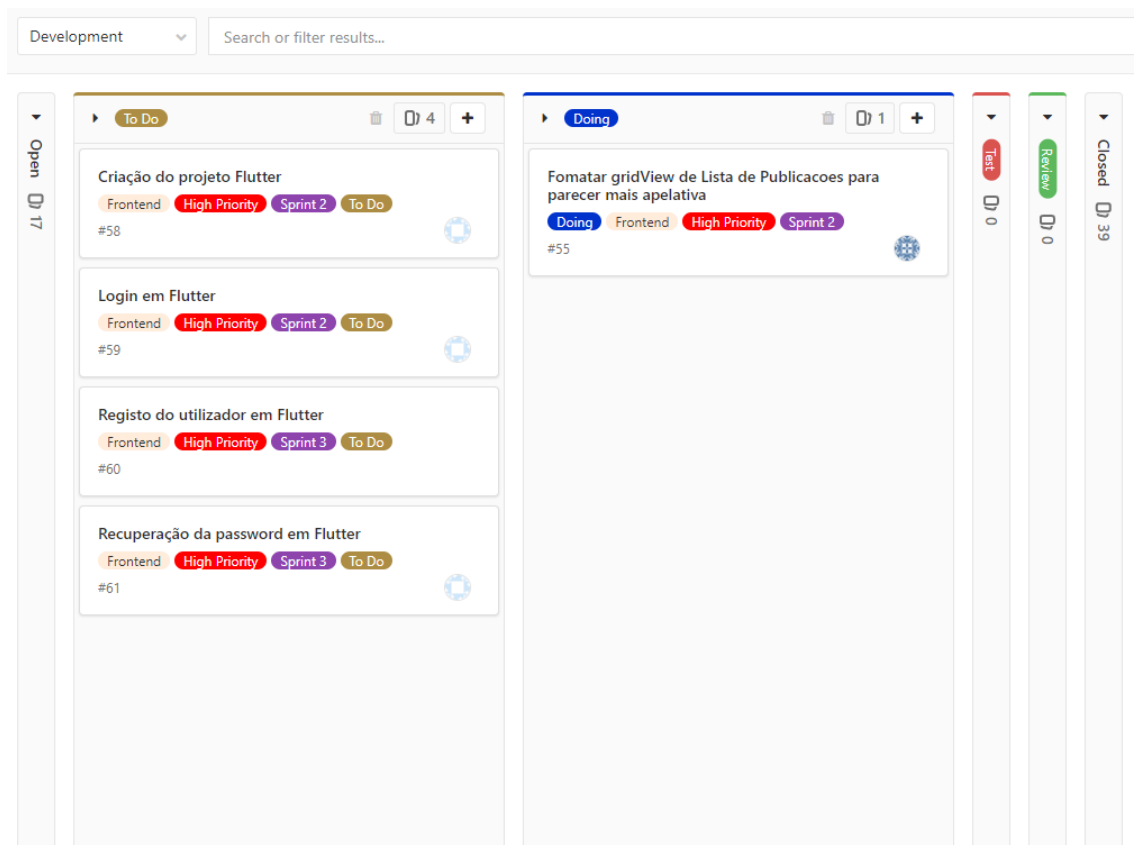


Figura 10 - Board Development

Sprints como milestones:

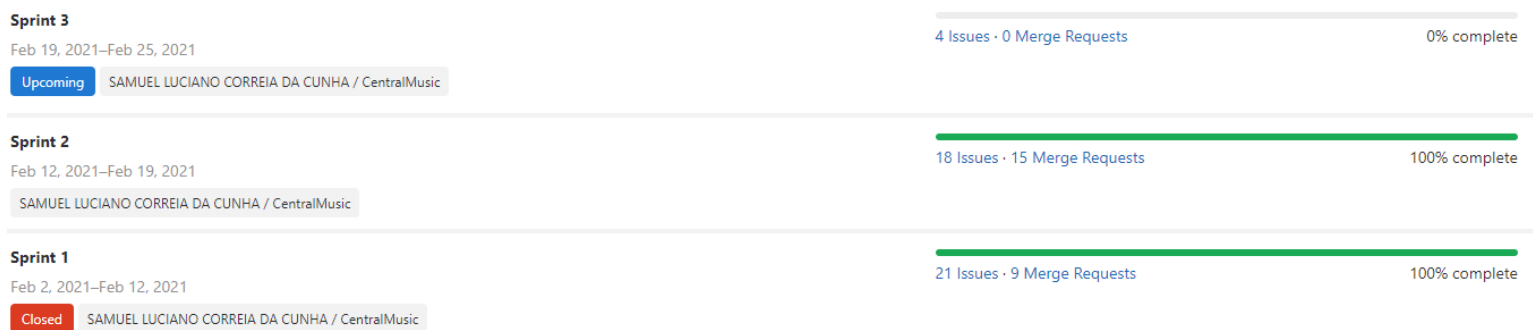


Figure 11 - Sprints

Nas subsecções seguintes serão apresentados os sumários das reuniões a cada sprint.

5.1.1 Sprint #1

Durante este sprint, temos como objetivo terminar o core da RestAPI que tem os seguintes requisitos:

- Desenvolvimento da documentação inicial e início do backend;
- Implementação do Sistema de Login (com autenticação e autorização);
- Implementação do Sistema de Registo de utilizadores;
- Implementação das rotas para o perfil do utilizador;
- Implementação das rotas para publicação de anúncio;
- Implementação das rotas para pesquisa de anúncios;
- Implementação do Sistema de chat.

5.1.2 Sprint #2

Durante este sprint, temos como objetivo:

- Melhorias à API e finalização;
- Testes unitários;
- Início da implementação em Flutter.

5.1.3 Sprint #3

Durante este sprint, temos como objetivo:

- Continuação da implementação da interface gráfica em Flutter para dispositivos móveis, utilizando a Rest API para estabelecer a comunicação com os dados;
- Reformulação e finalização da Rest API;
- Entrega final do projeto.

6 Conclusão

Posto isto, consideramos esta unidade curricular como uma das mais importantes do curso se não a mais importante, visto que foi uma excelente preparação para o que nos espera do estágio/projeto final. Encaramos o nosso projeto como extremamente importante e, para já, todo o trabalho está a correr dentro das expectativas desejadas.