Relatório Técnico – Especificações do Produto

Easy Farming

Disciplina:	IES - Introdução à Engenharia de Software
Data:	Aveiro, 4/01/2023
Estudantes:	81382: Catarina Marques 102383: Vânia Morais 103415: João Sousa
Resumo:	Sistema de aconselhamento *online* para agricultura que permite ao utilizador introduzir informações acerca dos seus cultivos e receber notificações para a respectiva gestão.

Índice:

1 Introdução

Este projecto surge no âmbito de Introdução à Engenharia de Software, abrangendo os conhecimentos adquiridos na disciplina, com passagem pelas várias fases, desde o conceito à sua implementação multicamada.

Para implementar a nossa ideia, recorremos a várias plataformas, como o *GitHub* para guardar o código fonte e arquivos e o *Jira* para a organização de tarefas e gestão de tempo. Usamos também o *Postman* para nos ajudar na gestão da base de dados e o *Docker* para a execução do nosso programa em *containers*.

Para uma melhor organização e distribuição de tarefas, cada membro do grupo tem o seu papel:

- Team Manager - Vânia Morais
- DevOps - João Sousa
- Architect - Catarina Marques

É de notar que, por sermos apenas três elementos neste grupo, todos tomaram o papel de Product Owner e de Developer.

2 Conceito do Produto

Definição do produto

Easy Farming pretende ajudar os utilizadores a impulsionar o sucesso das suas hortas, independentemente do seu nível de experiência agrícola. O sistema permite o reconhecimento e identificação da planta através de *upload* de uma fotografia, e dá algumas informações relevantes como nome, imagem, descrição, classe, família, género, reino, ordem, filo e se é uma planta tóxica para animais domésticos (cães e gatos).

Além disso, esta *web application* permite introduzir informações acerca de culturas e receber notificações para a respectiva gestão, nomeadamente avisos sobre rega consoante os níveis de humidade do ar, precipitação e temperatura atuais.

Oferece também, aos utilizadores premium, a oportunidade de ajudar os desenvolvedores da app a gerir API das plantas, podendo fazer alterações nos campos da descrição, classe, família, género, reino, ordem, filo e se a planta é inofensiva.

Por último, o sistema dispõe de um fórum com questões de outros utilizadores que podem servir de inspiração e para esclarecer dúvidas em comum.

Personas e motivações

Ana Moreira é uma jovem adulta de 30 anos, engenheira de gestão industrial, que finalmente conseguiu comprar a sua primeira casa com o seu marido. Tem um cão chamado Timmy, que tem muita energia.

Motivação: Ana quer ter um pequeno jardim como *hobbie*, mas não sabe como cuidar dele, nem que plantas podem ser nocivas para o seu animal de estimação.

Pedro Cardozo tem 57 anos e sempre foi agricultor. Tem uma grande plantação, com várias frutas e vegetais, mas às vezes gosta de variar as sementeiras. Ele sabe que diferentes plantas têm diferentes cuidados, datas de colheita e poda, mas com a idade, não se consegue lembrar de todos.

Motivação: Pedro quer complementar a sua experiência com a de outras pessoas que têm plantações similares à sua.

Cenários principais

Ana quer saber mais sobre uma planta - Enquanto passeava o Timmy no parque, a Ana encontrou uma flor que achou bonita para o seu jardim, mas que não conhecia. Tirou uma fotografia e fez *upload* na *app* para descobrir o nome da flor e as suas características. Na *app*, descobriu que era da família do aloe vera, e nociva para cães em caso de contacto.

Pedro semeia melões pela primeira vez - O Pedro tem um canteiro vazio e decidiu fazer um novo cultivo. Como tinha curiosidade de semear melões, foi à *app*, fez *login* e adicionou melões à sua lista de plantas. Como não sabia muito sobre este tipo de cultivo, viu na app as informações acerca do tipo de melões que escolheu.

Ana vai à loja comprar as plantas para o seu jardim - Depois do episódio no parque, a Ana decidiu fazer *Login* na *app* e guardar todas as plantas de que gostou para plantar no seu jardim. Quando chegou à loja, já sabia que plantas pedir.

Pedro torna-se utilizador premium - Por ter muitos cultivos e experiência, os desenvolvedores da app decidiram tornar o Pedro um utilizador premium, o que lhe permite ter acesso a todas as plantas existentes na base de dados. Ao entrar neste novo campo, o Pedro reparou que a planta batata tinha uma descrição incompleta e decidiu alterá-la. Esta alteração foi submetida para mais tarde ser revista pelos desenvolvedores.

Ana usa o fórum para tirar algumas dúvidas - Por ser uma jardineira de primeira viagem, a Ana tem algumas dúvidas de como tratar das suas plantas e por isso vai consultar o fórum para verificar se alguém tem dicas sobre as suas plantas.

3 Arquitetura

Principais requisitos e restrições

- O sistema baseia-se numa *web application* que permite ao utilizador introduzir o nome das suas culturas e receber informações sobre os cuidados a ter com as plantas. Como tal, é necessário que a plataforma permita o *login* dos utilizadores, assim como a introdução dos nomes das sementeiras.
- O sistema deve permitir a atualização de dados por utilizadores *premium*.
- O sistema deve permitir que o utilizador consulte as dúvidas e respostas de outros utilizadores no fórum.
- O sistema deve permitir o envio de notificações para o utilizador acerca da temperatura, probabilidade de precipitação e níveis de humidade do ar atuais.
- Além disso, deve conseguir reconhecer a espécie da planta através de fotografia e receber as suas informações mais relevantes.

Plano de arquitetura

Para o desenvolvimento da ***web app*** vão ser utilizados HTML, CSS, JavaScript de forma a criar uma app interativa para os utilizadores.

Em relação à **base de dados**, foi escolhido MySQL para armazenar e gerir os dados. Para a **geração de dados**, será utilizado um *script* em Python que estará conectado com o backend em que será conectado por meio de uma fila de mensagens gerida através do RabbitMQ.

Quanto ao **backend** utilizamos Spring Boot para simplificar a criação da aplicação e utilizamos uma REST API para simplificar o seu desenvolvimento.

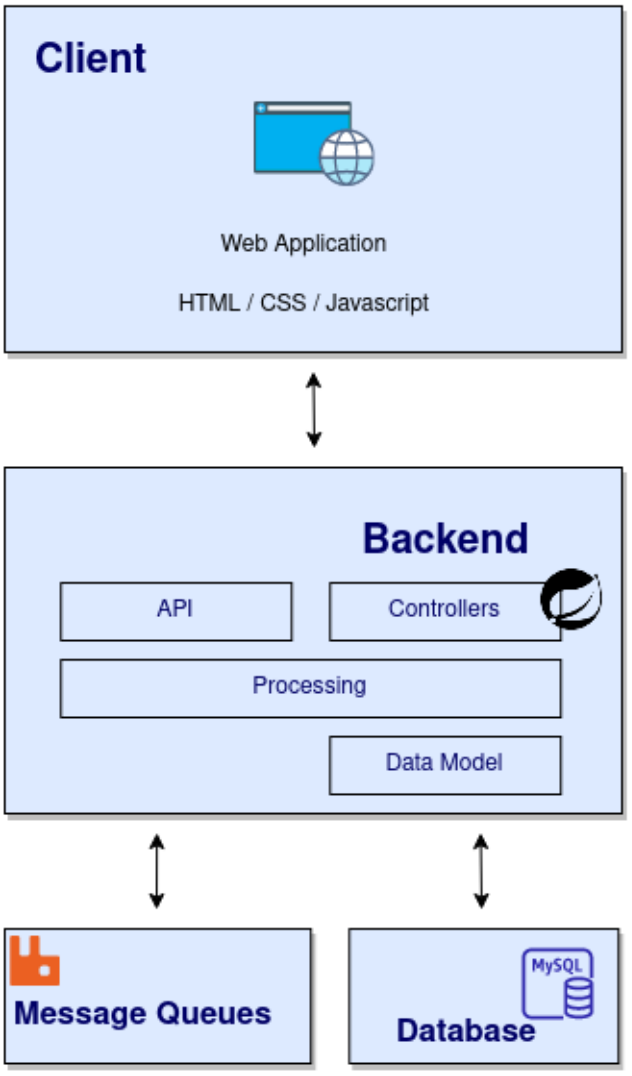

Figura 1- Diagrama de Arquitetura do Sistema

Interações entre módulos

1. Os dados são gerados pelo *script* em python e depois publicados numa *queue* do RabbitMQ.
2. Os dados no message broker são consumidos pela lógica implementada no Spring Boot, guardando a informação na base de dados.
3. Através da conexão da aplicação com o backend, é possível à aplicação receber os dados processados.
4. O utilizador vê os dados que pretende, sendo feito um pedido à API cada vez que o mesmo fizer upload a uma imagem.
5. A API retorna toda a informação sobre a imagem, que vai ser enviada para a base de dados e depois passada ao message broker
6. Depois o message broker conecta ao *script* em python, o que gera novos dados.

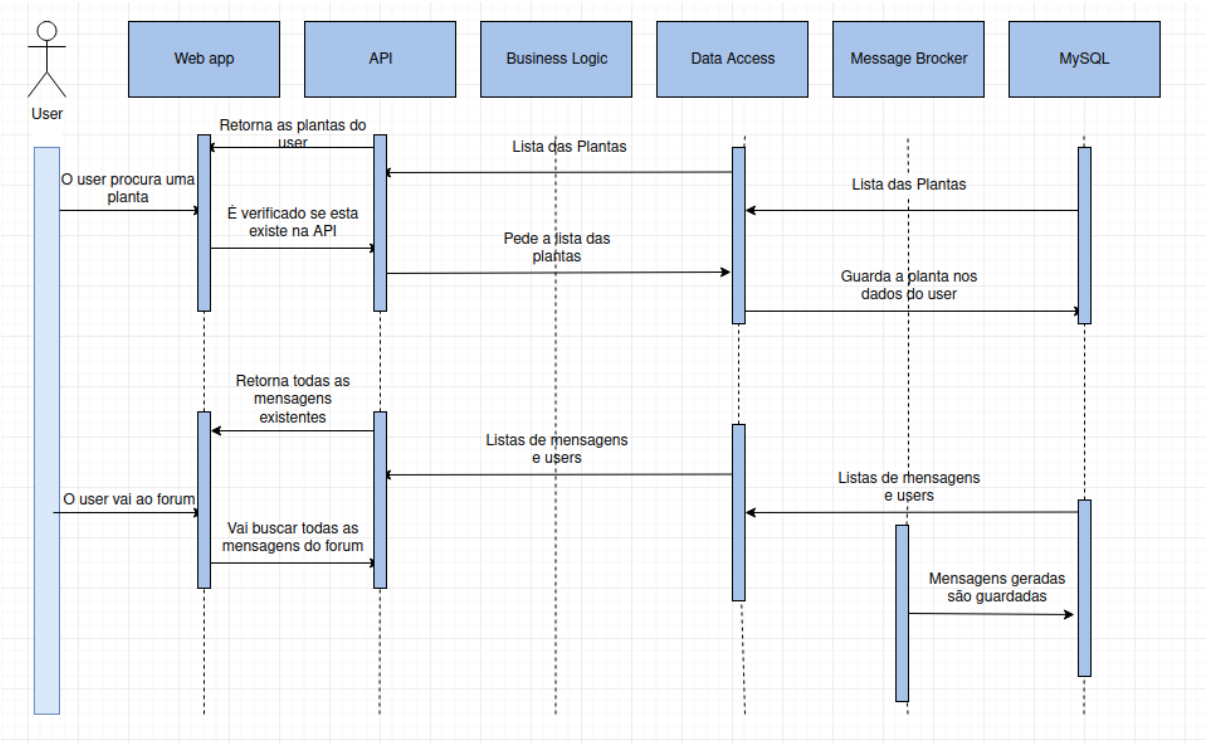

Figura 2- Diagrama de sequência com 2 exemplos

4 Perspetiva da informação

PlantRepository

- Plant: id, username, name, image, description, classs (escrito desta maneira por class ser uma palavra chave no Java), family, genus, kingdom, orderr (escrito desta forma porque order é uma palavra chave no MySQL), phylum, harmless

UserARepository

- UserA: id, username, name, image, description, classs (escrito desta maneira por class ser uma palavra chave no Java), family, genus, kingdom, orderr (escrito desta forma porque order é uma palavra chave no MySQL), phylum, harmless

UserPRepository

- UserP: id, username, name, image, description, class (escrito desta maneira por classs ser uma palavra chave no Java), family, genus, kingdom, orderr (escrito desta forma porque order é uma palavra chave no MySQL), phylum, harmless

ForumRepository

- Username, question, username1, answer1, username2, answer2, username3, answer3, username4, answer4, username5, answer5

MessagesRepository

- Id, weatherAlert

5 Ferramentas

Jira

Monitorização de tarefas e Gestão de projecto

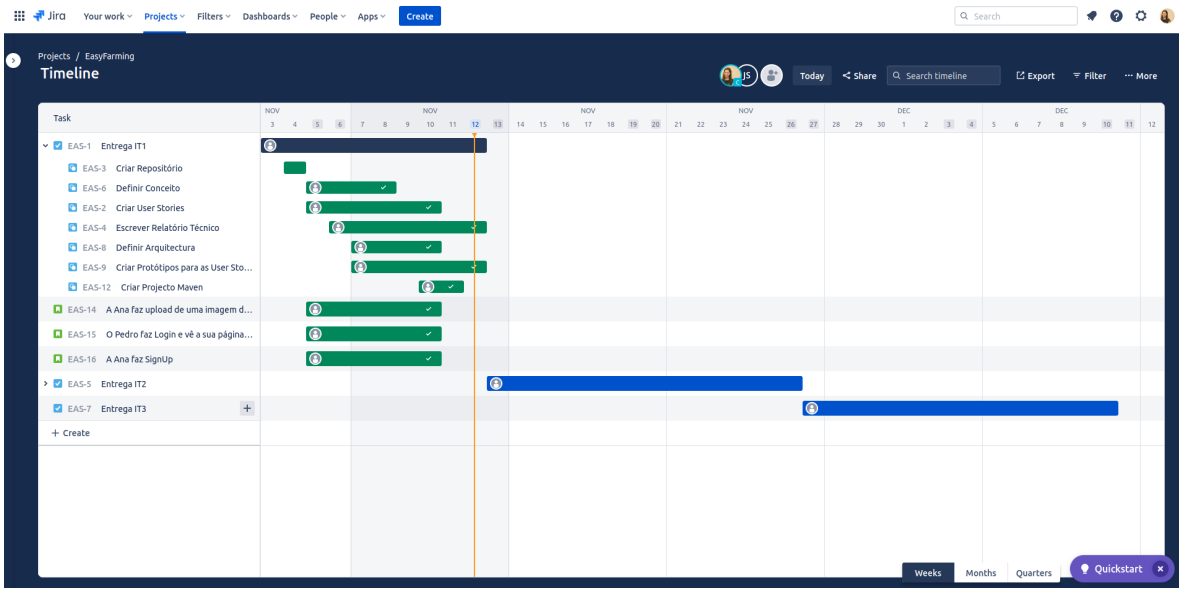

Figura 3. Timeline do Projecto

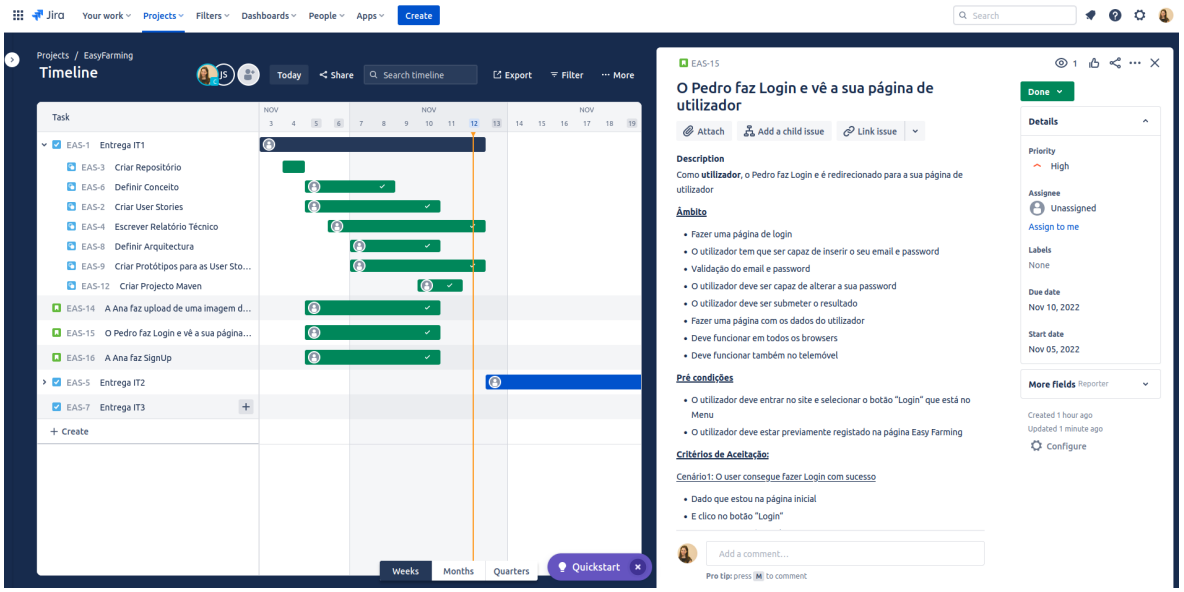

Figura 4. Exemplo de um dos *user stories* introduzidos no Jira

GitHub

Usamos o GitHub para guardar código fonte e arquivos

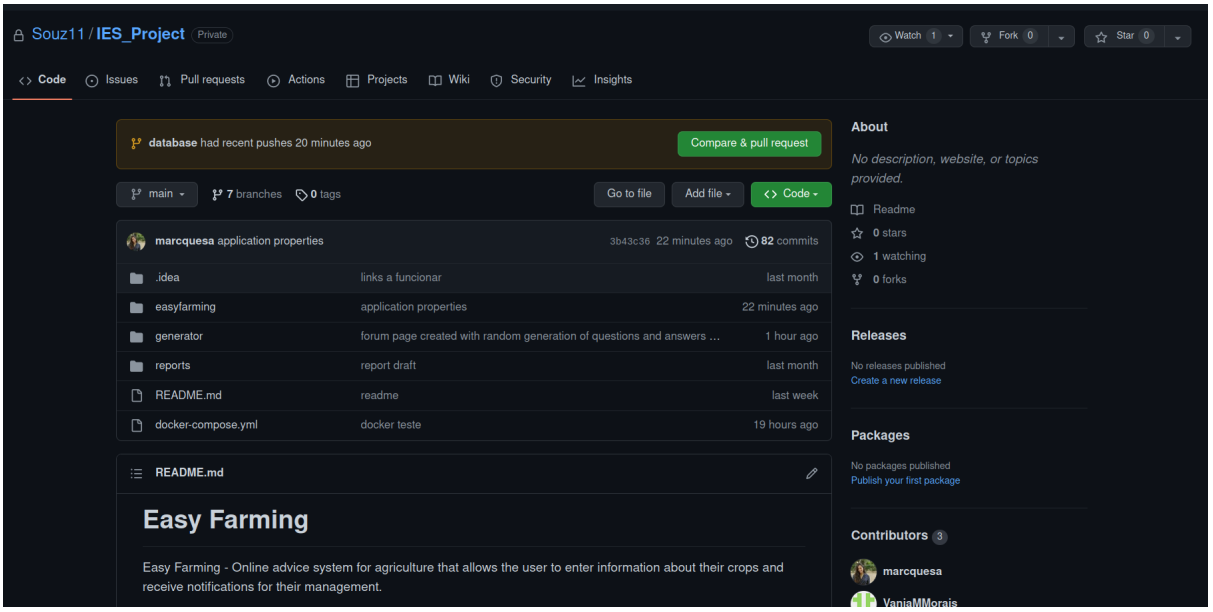

Figura 5. Repositório do trabalho no GitHub

6 Documentação da API

Plant Controller

Get - Receber todas as plantas - http://localhost:8090/api/v1/plants

Post - Guardar novas plantas - http://localhost:8090/api/v1/plants

Get - Saber se a planta existe- http://localhost:8090/api/v1/plantByName/{name}

Get - Receber os dados de uma planta em específico- http://localhost:8090/api/v1/plants/{name}

Put - Atualizar os dados de uma planta- http://localhost:8090/api/v1/updatePlant/{name}

Delete - Apagar todas as plantas- http://localhost:8090/api/v1/deleteAll

UserA Controller

Get - Receber todas as plantas - http://localhost:8090/api/v1/userAplants

Post - Guardar novas plantas - http://localhost:8090/api/v1/userAplants

Get - Saber se a planta existe- http://localhost:8090/api/v1/userAplantByName/{name}

Get - Receber os dados de uma planta em específico- http://localhost:8090/api/v1/userAplants/{name}

Delete - Apagar uma planta- http://localhost:8090/api/v1/deletePlantA/{name}

UserP Controller

Get - Receber todas as plantas - http://localhost:8090/api/v1/userPplants

Post - Guardar novas plantas - http://localhost:8090/api/v1/userPplants

Get - Saber se a planta existe- http://localhost:8090/api/v1/userPplantByName/{name}

Get - Receber os dados de uma planta em específico- http://localhost:8090/api/v1/userPplants/{name}

Delete - Apagar uma planta- http://localhost:8090/api/v1/deletePlantP/{name}

MessagesController

Get - Receber todas as mensagens - http://localhost:8090/messages/allmessages

Post - Guardar novas mensagens - http://localhost:8090/messages/allmessages

Delete - limpar a base de dados - http://localhost:8090/messages/deleteall

ForumController

Get - Receber todas as mensagens -
http://localhost:8090/forumMessages/allForumMessages

Post - Guardar novas mensagens -
http://localhost:8090/forumMessages/allForumMessages

Delete - limpar a base de dados - http://localhost:8090/forumMessages/deleteall

7 Referências e recursos

Enciclopédia de plantas: https://www.picturethisai.com/
Dúvidas sobre html e css: https://www.w3schools.com/
Inspiração e API para informação das plantas: https://plant.id/