

Relatório Técnico – Especificações do Produto

# Easy Farming

Disciplina: IES - Introdução à Engenharia de Software

Data: Aveiro, 12/12/2022

Estudantes: 81382: Catarina Marques  
102383: Vânia Morais  
103415: João Sousa

Resumo: Sistema de aconselhamento *online* para agricultura que permite ao utilizador introduzir informações acerca das suas culturas e receber notificações para a respectiva gestão.

## Índice:

### [1 Introdução](#)

### [2 Conceito do Produto](#)

[Personas e motivações](#)

[Cenários principais](#)

### [3 Arquitetura](#)

[Principais requisitos e restrições](#)

[Plano de arquitetura](#)

[Interações entre módulos](#)

### [4 Ferramentas](#)

[Jira](#)

### [5 Referências e recursos](#)

# 1 Introdução

Easy Farming pretende ajudar os utilizadores a impulsionar o sucesso das suas hortas, independentemente do seu nível de experiência agrícola. Esta *web application* permite introduzir informações acerca de culturas e receber notificações para a respectiva gestão, nomeadamente épocas indicadas para o seu cultivo, horários de rega, regularidade de aplicação de produtos fitofarmacêuticos indicados no tratamento de doenças, informações sobre poda e colheita. O sistema permite ainda o reconhecimento e identificação da planta através de *upload* de uma fotografia, bem como os cuidados a ter e outras informações relevantes.

Este projecto surge no âmbito de Introdução à Engenharia de Software, abrangendo os conhecimentos adquiridos na disciplina, implica a passagem pelas várias fases, desde o conceito à sua implementação multi-camada.

## 2 Conceito do Produto

### Personas e motivações



**Ana** 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** 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 estar a par de todas as novas pestes e doenças que podem comprometer a sua plantação, mas não sabe onde encontrar fontes que lhe dêem soluções viáveis.

## 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 das rosas, e nociva para cães em caso de longo 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* qual a melhor altura para semeá-los e os cuidados a ter com eles.

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

## 3 Arquitetura

### Principais requisitos e restrições

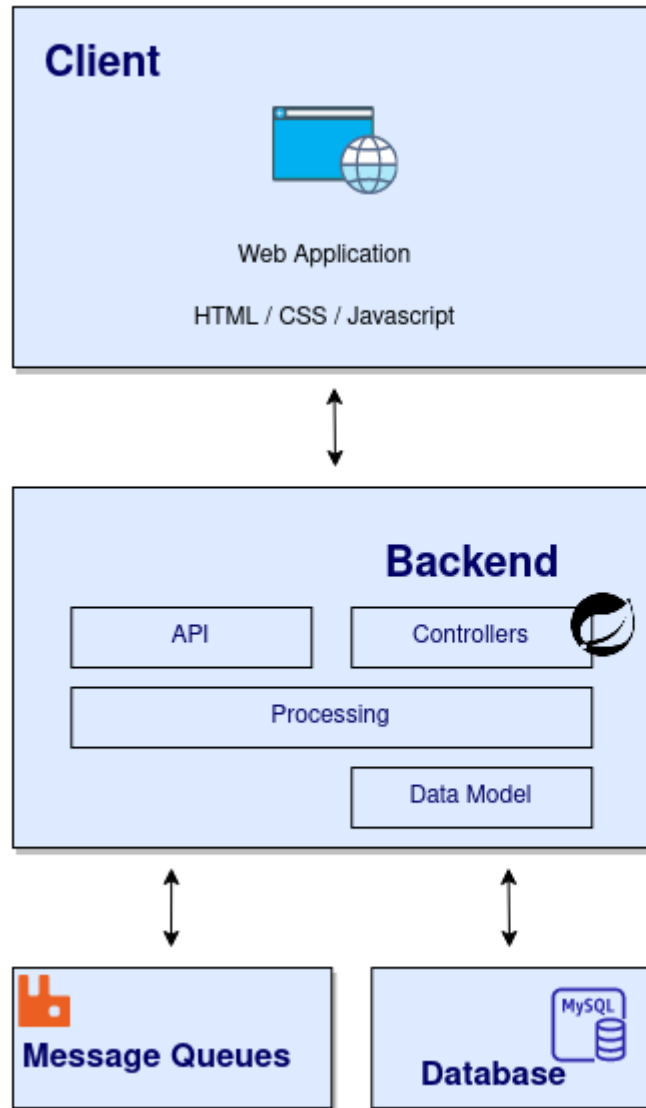
- O sistema baseia-se numa *web application* que permite ao utilizador introduzir o nome das suas culturas e receber informações sobre os cuidados a ter com as plantas. Como tal, é necessário que a plataforma permita o registo e o *login* dos utilizadores, assim como a introdução dos nomes das sementeiras.
- A plataforma deve estar atualizada com sugestões de culturas adequadas à época.
- O sistema deve permitir o envio de notificações para o utilizador acerca de horários de rega e poda.
- Além disso, deve conseguir reconhecer a espécie da planta através de fotografia e receber informações mais relevantes.

### Plano de arquitetura

Para o desenvolvimento da **web app** vão ser utilizados HTML, CSS e 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 gerenciada 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.



## Interações entre módulos

1. Os dados serão gerados pelo script em python e depois publicados num tópico do message broker.
2. Os dados no message broker vão ser consumidos pela lógica implementada no Spring Boot, guardando a informação na base de dados
3. Através da conexão da aplicação com o backend, será 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 irá gerar novos dados.

## 4 Ferramentas

### Jira

#### Monitorização de tarefas e Gestão de projecto

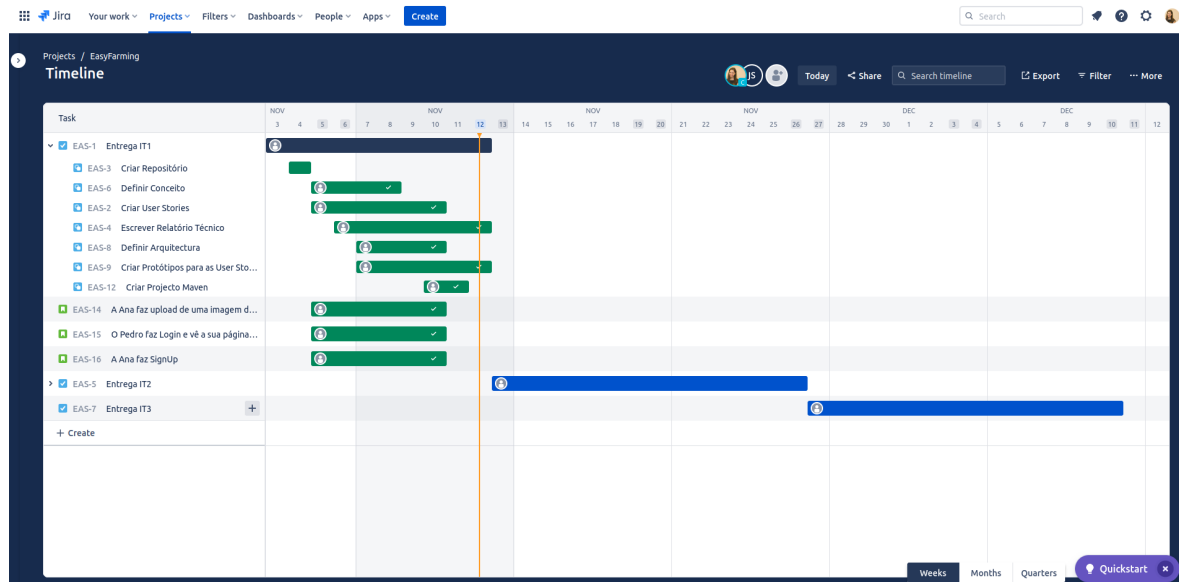


Figura 1. Timeline do Projecto

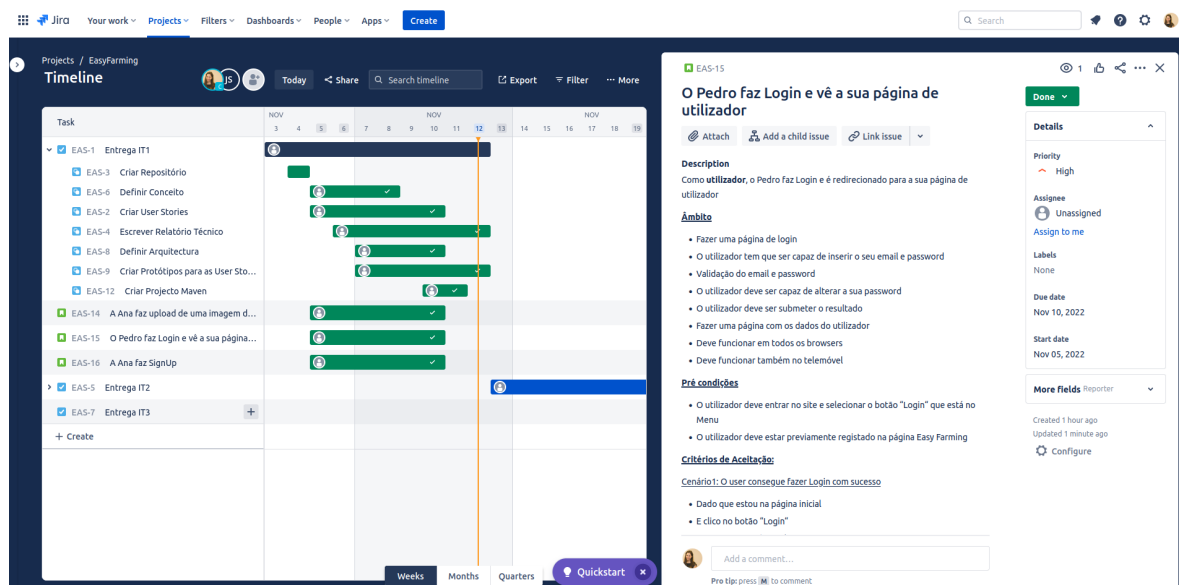


Figura 2. Exemplo de um dos user stories introduzidos no Jira

## 5 Referências e recursos

Enciclopédia de plantas: <https://www.picturethisai.com/>

Dúvidas sobre html e css: <https://www.w3schools.com/>