



ESTG

2023 SMART CONTRACT AND WEB DAPP FOR TRACEABILITY IN THE OLIVE OIL PRODUCTION CHAIN



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

SMART CONTRACT AND WEB DAPP FOR TRACEABILITY IN THE OLIVE OIL PRODUCTION CHAIN

Marcelo Filipe Antunes Fernandes

INSTITUTO POLITÉCNICO DE VIANA DO CASTELO

Smart Contract and Web DApp for Traceability in the Olive Oil Production chain

Marcelo Filipe Antunes Fernandes



Master degree in Informatics Engineering
Escola Superior de Tecnologia e Gestão

Advisor: Prof. Doctor Maria Estrela Ferreira Cruz

Advisor: Prof. Doctor António Miguel Rosado da Cruz

June 2, 2023

Smart Contract and Web DApp for Traceability in the Olive Oil Production chain

Marcelo Filipe Antunes Fernandes

Master degree in Informatics Engineering

June 2, 2023

Resumo

Hoje em dia, à medida que as pessoas se tornam mais conscientes sobre a saúde, a adopção de uma dieta saudável e equilibrada tornou-se cada vez mais importante para uma melhor qualidade de vida. Uma boa dieta não só ajuda a prevenir e combater doenças, como também contribui para o bem-estar geral. Consequentemente, os consumidores estão a prestar mais atenção aos alimentos que consomem, procurando produtos que ofereçam valor nutricional e sustentabilidade e não apenas o reconhecimento da marca.

Portugal é um dos maiores produtores de azeite do mundo, e o azeite é uma componente fundamental da dieta mediterrânica. O valor nutricional do azeite está directamente ligado a todas as fases da cadeia de valor do produto, desde o cultivo até ao consumo.

Para garantir que os consumidores tenham acesso a informação fiável e precisa sobre a qualidade do azeite que adquirem, é importante estabelecer a rastreabilidade na cadeia de produção. Ao fazê-lo, a confiança dos consumidores pode ser consolidada.

Este estudo centra-se na rastreabilidade na cadeia de produção do azeite, bem como nos indicadores de sustentabilidade social e ambiental e nos indicadores de qualidade. O objectivo é criar uma solução que utilize a tecnologia Blockchain e *Smart Contracts* para permitir que informações mais fiáveis e transparentes sobre a produção e qualidade do azeite sejam facilmente acessíveis aos consumidores.

Este caso de estudo é importante porque sublinha a necessidade de maior transparência na cadeia da valor da produção alimentar, bem como os benefícios de utilizar novas tecnologias como a Blockchain para melhorar a transparência e a rastreabilidade. Também demonstra a importância da sustentabilidade na produção de produtos alimentares, e como pode ter impacto tanto no ambiente como na saúde dos consumidores.

Abstract

Nowadays, as people become more health-conscious, adopting a healthy and balanced diet has become increasingly important for a better quality of life. A good diet not only helps to prevent and fight diseases, but also contributes to overall well-being. At the same time, consumers are concerned about climate change and the sustainability of the planet. Consequently, consumers are paying more attention to the food they consume, seeking products that offer nutritional value and sustainability rather than just brand recognition.

Portugal is one of the largest olive oil producers in the world, and olive oil is a fundamental component of the Mediterranean diet. The nutritional value of olive oil is directly linked to every stage of the product's value chain, from cultivation to consumption.

To ensure that consumers have access to reliable and accurate information about the quality of the olive oil they purchase, it is important to establish traceability in the production chain. By doing so, consumer confidence can be consolidated.

This study focuses on traceability in the olive oil production chain, as well as social and environmental sustainability indicators and quality indicators. The goal is to create a solution that utilizes Blockchain technology and *Smart Contracts* to enable more reliable and transparent information about olive oil production and quality to be easily accessible to consumers.

This case study is important because it highlights the need for greater transparency in the food production chain, as well as the benefits of utilizing new technologies like Blockchain to improve transparency and traceability. It also demonstrates the significance of sustainability in the production of food products, and how it can impact both the environment and consumer health.

Acknowledgments

Completing a master's thesis is a journey full of uncertainties and doubts, and it takes a lot of dedication and hard work to reach the finish line. I would like to express my deepest gratitude to all the people who supported me throughout this journey.

First and foremost, I would like to thank my thesis supervisors, Professor António Miguel Cruz and Professor Maria Estrela Cruz, for their invaluable guidance, patience, and encouragement. Without their support, this thesis would not have been possible.

I am also grateful to everyone who directly or indirectly contributed to this project, including research participants, colleagues, friends, and family members. I especially want to express my appreciation to my classmates from the master's degree, particularly Hugo and Filipe, for their valuable feedback and camaraderie.

To my family specially to my *dad* and my *mom*, I owe a debt of gratitude for their unwavering support and understanding, even when I was buried in work and unable to spend as much time with them as I would have liked.

I would like to express my heartfelt thanks to my business partner, Leandro, for always believing in my potential and providing me with the space and resources to pursue this project.

I am also indebted to Engineer Miguel Santos, who believed in this project as much as we did and agreed to let us use olive oil as the pilot test of this project.

Last but not least, I would like to thank my girlfriend, Ana, for her endless encouragement, motivation, and unwavering presence throughout this entire process. Her love and support kept me going, even when I felt like giving up.

Thank you all for being a part of this journey and for helping me achieve my goals.

Marcelo Antunes Fernandes

Glossary

AI Artificial Intelligence. 15, 20

ANACOM Autoridade Nacional de Comunicações. 1

API Application Programming Interface. 6–8, 10, 13, 29, 33–36, 40, 41, 49–51, 53

APP Application. 6, 8, 14, 34–36, 39–41, 50, 51

attr Attributes. 40, 44–49

BPMN Business Process Model an Notation. ix, 25, 26

CSS Cascading Style Sheets. 14

DApp Decentralized Application. 4, 6, 57

DB Database. 36

dev Development. 36

DSR Design Science Research. 3, 4

enum Enumerator. 48, 49

env environment. 36, 37

EVOO Extra Virgin Olive Oil. 2, 20, 21, 26, 27, 78

FR Functional Requirement. 28

h Hour. 66

HTML Hyper Text Markup Language. 14

HTTP Hypertext Transfer Protocol. 6–8, 35, 40, 51, 65

IoT Internet of Things. 15, 16, 68

JWT JSON Web Token. 8, 35

l Liters. 48, 59, 60, 62, 65, 66

- m** Minute. 60, 66
- MB** Megabyte. 29
- ms** milliseconds. 29, 59–63, 65, 66
- N/A** Not Applicable. 40, 44–49
- NFC** Near-Field Communication. 19
- NFR** Non Functional Requirement. 28, 29
- pBFT** Practical Byzantine Fault Tolerance. 18
- PDO** Protected Designation of Origin. 19
- PGI** Protected Geographical Indication. 19
- PoA** Proof-of-Authority. 18
- PoET** Proof-of-Elapsed Time. 18
- PoL** Proof-of-Luck. 18
- PoS** Proof-of-Stake. 17
- PoW** Proof-of-Work. 17
- prod** Production. 36, 37
- RFID** Radio-Frequency Identification. 19
- TSG** Traditional Specialty Guaranteed. 19
- UE** European Union. 22, 26, 27
- vm** Virtual machine. 10
- WSL** Windows Subsystem for Linux. 10

For my godchildren...

Marcelo Antunes Fernandes

Contents

1	Introduction	1
1.1	Contextualization and Motivation	2
1.2	Objectives	2
1.3	Research method	3
1.4	Document structure	5
2	Concepts, Tools, Languages and Frameworks	6
2.1	Concepts	6
2.1.1	Decentralized Application	6
2.1.2	HTTP	6
2.1.3	REST API	7
2.1.4	Multi-tenant	7
2.1.5	Unit Of Work Pattern	7
2.1.6	JWT	8
2.1.7	I/O	8
2.2	Tools	8
2.2.1	Azure DevOps	8
2.2.2	Hangfire	9
2.2.3	Quartz.NET	9
2.2.4	Swagger	10
2.2.5	WSL	10
2.2.6	Docker	10
2.2.7	Hyperledger fabric	11
2.2.8	Fablo and Fablo REST	12
2.3	Languages and frameworks	12
2.3.1	C# / ASP.NET Core	12
2.3.2	Microsoft SQL Server	13
2.3.3	Go	13
2.3.4	Ionic framework with Angular	14
3	State of the art	15
3.1	The Blockchain Technology	15
3.1.1	Types of Blockchain	16
3.1.2	Consensus protocols	17
3.2	Blockchain-based solutions for tracking food products	18
3.3	Blockchain and sustainability in the value chain	19
3.4	Blockchain in the traceability of the olive oil production chain	19
3.4.1	Devoleum	20

3.4.2	BRUSCHETTA	20
4	Problem analysis	22
4.1	Value chain of food products	22
4.2	The olive oil production process	23
4.3	The olive oil production business process	25
4.4	The olive oil value chain traceability	26
4.5	The olive oil sustainability indicators	26
4.6	Requirements	28
4.6.1	Functional requirements	28
4.6.2	Non Functional Requirements	29
4.7	Use case diagram	29
4.8	Database model	30
5	Design and implementation	34
5.1	Database	34
5.2	Architecture	35
5.3	Implementation	35
5.3.1	Project management	36
5.3.2	APP	39
5.3.3	API	49
5.3.4	Blockchain	50
5.3.5	From Application to Blockchain	51
5.3.6	Store information and queue to Blockchain	52
5.3.7	Olive Production Smart Contract	54
5.3.8	Traceability	56
6	Evaluation and results	58
6.1	Case study	58
6.2	Data collect	58
6.2.1	Harvest	59
6.2.2	Transport	60
6.2.3	Olive mill	61
6.3	Analysis	63
6.3.1	Results	63
6.3.2	Traceability	63
6.3.3	Blockchain execution times results	65
6.3.4	Quality and sustainability results	65
6.4	Discussion	66
7	Conslusions and future work	68
A	Pictures	76
B	Blockchain data for Madural variety production	79
C	Pitch deck	87

List of Figures

1.1	Design Science Research activities (extracted from [Cruz et al., 2014]).	4
2.1	Structure of a typical web-based DApp and centralized App (extracted from [Min and Cai, 2022]).	7
2.2	Unit of Work Pattern (extracted from [Dykstra, 2022]).	8
2.3	Azure DevOps Services (extracted from [azu, 2022b]).	9
2.4	Fablo (extracted from [Fablo, 2022a]).	13
2.5	Ionic Framework with Angular development.	14
3.1	Blockchain blocks example (extracted from [Kamilaris et al., 2019]).	16
3.2	Value chain of a food product.	20
3.3	BRUSCHETTA architecture (extracted from [Arena et al., 2019]).	21
4.1	Value chain of a food product.	23
4.2	Traditional and modern processes in olive oil production (extracted from [Safa et al., 2017]).	25
4.3	Olive oil production Business Process Model an Notation (BPMN).	26
4.4	Use case diagram.	30
4.5	Database model.	32
5.1	Architecture.	36
5.2	February to May Sprints.	37
5.3	June to September Sprints.	38
5.4	October Sprint and launching.	39
5.5	Application Structure.	40
5.6	Application login form.	41
5.7	Application Menu.	42
5.8	Application Tabs.	44
5.9	Application olive mills list and the olive mill details.	45
5.10	Application Olive varieties list and the olive variety details.	46
5.11	Olive oil production and the list of its activities.	47
5.12	Olive oil production activity with measurements.	49
5.13	API Structure.	50
5.14	Blockchain Structure.	51
5.15	From Application to Blockchain diagram.	52
5.16	Store information and queue to Blockchain.	53
5.17	Request Blockchain.	54
5.18	Olive Production Smart Contract.	55
5.19	Smart Contract Assets.	56

5.20	Relations between production, activities and measurements.	57
5.21	Example of traceability.	57
6.1	Madural olive oil production chain registered activities.	64
6.2	Blockchain execution times results.	65
A.1	Harvest pictures.	76
A.2	Transport pictures.	77
A.3	Olive mill pictures	78

List of Tables

4.1	Table of olive oil sustainability and quality indicators.	27
4.2	Functional requirements.	28
4.3	Non Functional Requirements.	29
4.4	Application Users.	30
4.5	Database entities.	33
5.1	APP Structure.	39
5.2	Login form.	40
5.3	Olive mill form.	44
5.4	Olive variety form.	46
5.5	Olive oil production form.	47
5.6	Olive oil production activity form.	48
5.7	Olive oil production activity measurement form.	49
5.8	API Structure.	50
5.9	Blockchain Structure.	50
6.1	Harvest execution times	59
6.2	Transport execution times	61
6.3	Olive mill execution times	62
6.4	Table of olive oil sustainability and quality indicators collected.	66

Chapter 1

Introduction

Portugal is one of the largest olive oil producers in the world. Olive oil is one of the basic ingredients used in Mediterranean cuisine and diet. Olive oil is an ingredient with great nutritional value and is very healthy, which is why it is used in the creation of many other food products, such as canned sardines, biscuits, cakes, etc. Currently, consumers are very attentive to the products they consume and try to be informed about all aspects of the product. Also, people try to lead a healthier life which involves having a healthy diet. This can be pointed out as one of the growth factors in olive oil consumption.

At the same time, it is undeniable, technology is increasingly part of people's daily lives. Statistics presented by [Autoridade Nacional de Comunicações \(ANACOM\)](#) show that, increasingly, the population has access to the Internet ("the number of [Internet] accesses reached 1.3 thousand in the 3rd quarter of 2021, representing growth of 4.7% over the same period last year") [[ANACOM, 2022](#)], which makes consumers more informed, as they have everything they are looking for just a click away and, therefore, increasingly demanding. Added to all this is a greater environmental concern, with planet Earth living in extreme situations due to global warming.

Given these challenges, it is imperative that companies keep up with the constant demands of the market, increasing innovation in their processes and offering products and services that increases the consumers value. This master thesis focuses on building a Blockchain-based technology to implement traceability of the olive oil value chain. The Blockchain technology provides transparency, traceability and security in transactions [[Rosado da Cruz and Cruz, 2020](#)]. This technology allows to collect real-time data and the creation of smart contracts in order to meet the needs of its users.

Depending on the context, the term sustainability has different interpretations. Sometimes the term sustainability is interpreted as an environmental view only, but currently, the term incorporates the environment, the economy and society, thus forming the three pillars of sustainability [[Matzembacher et al., 2018](#)]. Thus, and as already described here, environmental sustainability can be a factor to be taken into account in the decision about the purchase of a product by the final consumer, because individuals who are concerned with environmental sustainability, try not to participate in production processes that are harmful to the environment [[Matzembacher et al., 2018](#)].

1.1 Contextualization and Motivation

As mentioned in the previous section, nowadays people are increasingly adopting a healthier and more sustainable way of life. A healthy and balanced diet is therefore essential to achieve a better quality of life, as it helps prevent and fight diseases.

Thus, people are giving more and more importance to the food they consume, not looking for food based on the brand, but based on its attributes, such as nutritional value or sustainability indicators. These attributes are directly related to all stages of the value chain of a product, from its creation (cultivation) to its arrival to the final consumer.

For the final consumer to have access to this information, it is necessary to create ways to ensure that it is available and reliable, thus consolidating consumer confidence. Given the growing concerns of consumers about the products they choose to consume, and the needs of industry to meet these concerns, the problem of food product traceability arises.

Traceability involves recording information about where, who, when and how each step in the value chain is performed, from production to the final consumer. Thus, traceability is considered "part of logistics management and evaluation by means of identification to the registered" [Garaus and Treiblmaier, 2021]. The European Commission defines traceability as "the ability to trace any food, feed, food-producing animal or substance that will be used for consumption, through all stages of production, processing and distribution" [European Commission, 2007]. At the production level, it allows manufacturers to meet regulatory requirements by tracking the entire manufacturing, processing and consumption process. It also allows the reduction of distribution of products that are not appropriate or of unsatisfactory quality [Olsen and Borit, 2013]. At the consumer level, and given that nowadays there is an increased concern about the products purchased, as well as greater access to new technologies, which translates into an increasingly informed market, it gives them the possibility to know the origins of the products they consume, thus increasing their confidence in the product.

In the specific case of the **Extra Virgin Olive Oil (EVOO)** production it is essential to record the execution times of each step, as well as the environmental factors, such as temperature and humidity, since these directly influence the quality of the olive oil.

1.2 Objectives

The main objective of this project is to implement a platform allowing to collect and record traceability information about sustainability indicators and about who, when and where each step in olive oil value chain was executed, allowing to trace back and forward the olive oil products. The solution must provide a user-friendly interface so that the final consumer can easily access the collected information. The solution must also allow a business partner to query and integrate the information with information from its own value chain, thus facilitating the traceability of products that use olive oil as the basic ingredient.

For this, it is necessary to proceed through a set of steps, namely:

- Know and specify the olive oil production business process to know the main activities on which information will be collected.
- Identify the sustainability indicators that need to be measured and, therefore, store information about it.
- Study the blockchain platform that best suits the implementation of the problem.
- Design the traceability solution.
- Develop and implement the traceability solution by selecting the suitable tools.
- Select a real case scenario to test the implemented solution. Ideally, the solution should be ready to collect data about harvest in October 2022.
- Collect and evaluate the results.

1.3 Research method

There are several existing research methods, and each is applicable and appropriate given the area of study in question. The most appropriate approach depends on the topic and the research questions being analyzed [David E. Avison and Nielsen, 1999]. In this project, the methodology selected was **Design Science Research (DSR)** mainly because this work intends to create a practical and useful solution that allows to increase value for the institutions (eg agricultural cooperatives) or companies involved.

DSR is a fairly recent research method, which had its beginnings in engineering and other applied sciences [Venable, 2006]. This method highlights the development of artifacts and its main goal is to improve the performance of companies and organizations by solving specific problems. In this way, it allows, directly or indirectly, the increase of their profits [Cruz and Cruz, 2020]. **DSR** focuses on solving problems or improving the performance of solutions[Cruz and Cruz, 2020].

According to Hevner, research using the **DSR** method requires the creation of a novel artifact for solving a domain-specific problem [Hevner and Chatterjee, 2010]. The construction of the artifact is complete when it satisfies requirements and constraints of the problem it is intended to solve. The **DSR** method is a problem-solving process in which "knowledge and understanding of the design of a problem and its solution are acquired in the construction and application of an artifact"[Hevner et al., 2004].

The **DSR** method, on the one hand, solves problems that have never been solved before in a unique and innovative way and, on the other hand, solves problems that have been solved before, but in a more efficient and effective way [Hevner and Chatterjee, 2010]. Ken Peffers identifies common points concerning the stages of development of a research that uses the **DSR** method [Pfeffers et al., 2006]. Thus, he defines a process model to develop and evaluate a research, in Information Systems, that uses the **DSR** method. The process consists of six activities [Cruz et al., 2014]:

1. Identification of the problem and motivation;
2. Definition of the objectives for the solution;
3. Design and development;
4. Demonstration;
5. Evaluation;
6. Communication.

These activities can be executed in sequence, i.e. from 1 to 6, or they can have a different order. Some of the activities can even be performed several times [Cruz, 2011] as can see in 1.1.

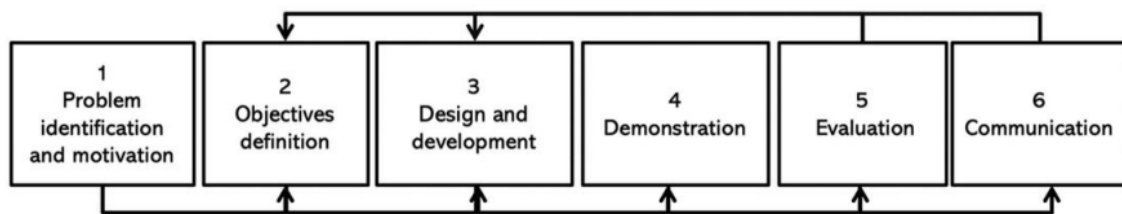


Figure 1.1: Design Science Research activities (extracted from [Cruz et al., 2014]).

For the development of this research, we are following the **DSR** methodology. **DSR** is a research method that is based on the construction of innovative and useful artifacts, that by solving specific problems, improve the performance of a company or organizations thus allowing to, directly or indirectly, increase their profits [Cruz and Cruz, 2020, Hevner and Chatterjee, 2010].

DSR may be defined as a process consisting of six main activities[Cruz and Cruz, 2020]. and these activities are explained next for the context of the investigation presented herein:

1. Problem identification and motivation - for end consumers, it is important to know what is happening in the olive oil value chain, so that they can be informed about what they are consuming. from the company's point of view, it is important to avoid counterfeits, maintain the olive oil's reputation and keep customers satisfied.
2. Definition of the objectives for the solution - it is intended to develop a Web **Decentralized Application (DApp)** for tracking traceability and the sustainability indicators in the olive oil production value chain, so that the final consumer may, through the batch code, know all the information related to all stages of the chain of value.
3. Design and implementation - A solution will be developed involving a smart contract and a web application.
4. Demonstration - at this stage, we will demonstrate that the artifacts are capable of responding to the research problems mentioned in this document, by putting it into practice in the case study of the *Azeite dos Santos Premium* by *Olive & All* company. *Olive & All* company has a vast olive grove, located in the north of Portugal and made up of centenary olive trees.

5. Evaluation - simulations/tests of the developed platform will be performed, including performance tests, usability and especially process streamlining will be evaluated to.
6. Communication - once all the approval tests have passed, the results will be presented and published in a journal in the area and/or at conferences.

1.4 Document structure

This master's thesis is structured as follows: The first, and current chapter, presents some introductory notes, the contextualization and motivation for this work. Then the main objectives are presented along with the methodological approach used in this master's thesis.

Chapter 2 presents some concepts and definitions that are needed throughout the research and presents the main tools used during this work development and implementation.

Chapter 3 presents the state of the art starting by a brief presentation of Blockchain Technology. Then, some works implementing traceability in food product using Blockchain are presented. At the end of this chapter, works that implement traceability of olive oil by using Blockchain technology are presented.

Chapter 4 presents the study about the problem as well as the requirements to implement the solution.

Chapter 5 presents the design and implementation of the proposed framework, mainly the requirements are identified and the main software models as is the case of use case model, domain model, etc. are presented.

Chapter 6 presents the validation of the created platform by putting it into practice in a real olive oil production. For that, the *Azeite dos Santos Premium* produced in the north of Portugal is used as a case study (section 6.1). The results are summarized in section 6.3 and a discussion about the obtained results are presented in section 6.4.

Finally, in chapter 7 some conclusions and future work are presented.

A list of references and an appendix are also included at the end of the document.

Chapter 2

Concepts, Tools, Languages and Frameworks

This section introduces and explains some concepts that are talked about throughout the report. It is also covered are some definitions that are important to the understanding of the project.

The tools, technologies, languages and frameworks used in this project are also explained, what are their and what are the advantages of using them in this implementation.

2.1 Concepts

2.1.1 Decentralized Application

DApp are programs that operate on a blockchain or a peer-to-peer network of computers, rather than relying on a single computer [Min and Cai, 2022]. They are not controlled by a single authority and offer increased privacy and flexibility. However, scaling can be a challenge, and traditional front-end frameworks may not be suited to **DApp** user interfaces, particularly with issues around identity and blockchain state usage. In most cases, a backend **Application Programming Interface (API)** is needed to manage all interactions [Min and Cai, 2022]. Popular platforms for creating **DApp** include Ethereum and Hyperledger Fabric.

A comparison between normal **Application (APP)** and **DApp** can be found on 2.1.

2.1.2 HTTP

Hypertext Transfer Protocol (HTTP) defines a set of request methods to indicate the desired action to be performed for a given resource. Although they can also be nouns, these request methods are sometimes referred to as HTTP verbs [Mozzila, 2022]. These **HTTP** verbs are directly connected to REST **API** and like says before the commonly used are GET, POST, PUT, PATCH, and DELETE.

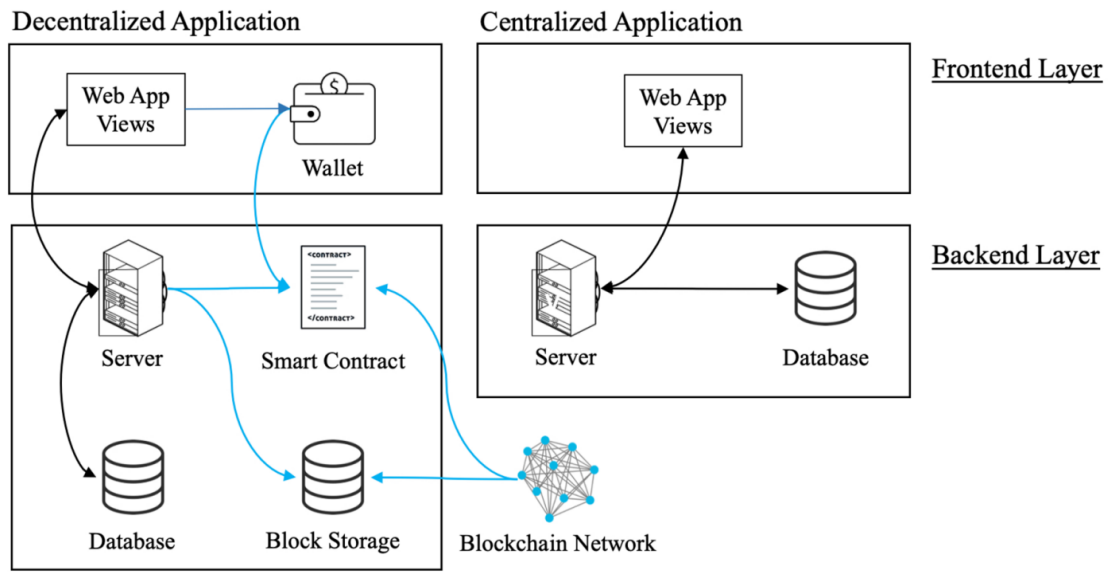


Figure 2.1: Structure of a typical web-based DApp and centralized App (extracted from [Min and Cai, 2022]).

2.1.3 REST API

Before describing REST API, we need to understand what an API is. API stands for application programmable interface [Lane, 2022], which is a method of communication that can be used between applications or computers to allow integration between themselves. An API call starts with one party requesting the API for a specific resource. The API then responds with a response containing the requested resource or execution status.

There exist different architectures of API, and one of them is REST [Akana, 2022]. REST stands for representational state transfer and utilizes HTTP to handle the communication. When using REST, two parties must be involved: a client that initiates a request and a server that responds with the requested resource or status code. In the REST architecture, there is a set of instructions that are commonly used: GET, POST, PUT, PATCH, and DELETE [Microsoft, 2022b]. The named instructions manipulate or fetch data from a given resource.

2.1.4 Multi-tenant

A tenancy model determines how each tenant's data is mapped to storage [Microsoft, 2022a].

2.1.5 Unit Of Work Pattern

The repository and unit of work patterns are intended to create an abstraction layer between the data access layer and the business logic layer of an application. Implementing these patterns can help insulate your application from changes in the data store and can facilitate automated unit testing or test-driven development (TDD) [Dykstra, 2022].

The figure 2.2 shows one way to conceptualize the relationships between the controller and context classes compared to not using the repository or unit of work pattern at all.

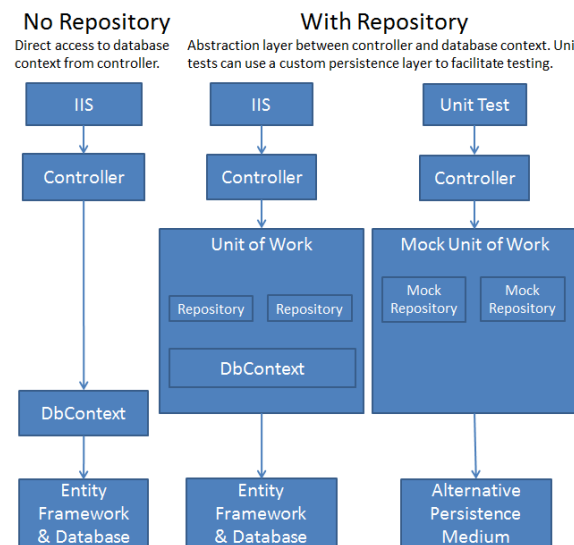


Figure 2.2: Unit of Work Pattern (extracted from [Dykstra, 2022]).

2.1.6 JWT

JSON Web Token (JWT), are an industry standard for representing statements between two stakeholders [JWT.io, 2022b]. This information can be verified and trusted because it is digitally signed. These can be signed using a secret word, or a public/private key pair [JWT.io, 2022a].

This authentication mechanism was used between the APP and the API and between and API and the Blockchain.

2.1.7 I/O

I/O, short for Input/Output. Referring to the flow of data transmitted over a **HTTP** request.

2.2 Tools

2.2.1 Azure DevOps

Azure DevOps supports a collaborative culture and set of processes that bring together developers, project managers, and contributors to develop software. It allows organizations to create and improve products at a faster pace than they can with traditional software development approaches [azu, 2022a].

Azure DevOps provides integrated features that you can access through your web browser or IDE client. You can use all the services included with Azure DevOps, or choose just what you need to complement your existing workflows [azu, 2022a].

- **Azure boards** - Delivers a suite of Agile tools to support planning and tracking work, code defects, and issues using Kanban and Scrum methods.
- **Azure Repos** - Provides Git repositories or Team Foundation Version Control (TFVC) for source control of your code.
- **Azure Pipelines** - Provides build and release services to support continuous integration and delivery of your applications.
- **Azure Test Plans** - Provides several tools to test your apps, including manual/exploratory testing and continuous testing.
- **Azure Artifacts** - Allows teams to share packages such as Maven, npm, NuGet, and more from public and private sources and integrate package sharing into your pipelines.

All these services were used during the project.



Figure 2.3: Azure DevOps Services (extracted from [azu, 2022b]).

2.2.2 Hangfire

Hangfire provides a unified programming model to handle background tasks in a reliable way and run them on shared hosting, dedicated hosting or in cloud. You can start with a simple setup and grow computational power for background jobs [hangfire.io, 2022].

Used as a background job to register the data in the Blockchain.

2.2.3 Quartz.NET

Quartz.NET is a job scheduling library inspired by Java's Quartz library. It is an open-source initiative that aims to provide a number of features and functionalities for scheduling and executing tasks in .NET applications [qua, 2022].

The first key concept of Quartz.NET is the concept of Jobs, which represents a task to be scheduled and executed by it, and is represented by the IJob interface. This interface has only

one method, called `execute`, which contains the implementation of the routine to be executed [qua, 2022].

Job scheduler to run tasks automatically and periodically.

2.2.4 Swagger

Swagger (OpenAPI) is a language-agnostic specification for describing REST API. It allows both computers and humans to understand the capabilities of a REST API without direct access to the source code. Its main goals are to:

- Minimize the amount of work needed to connect decoupled services.
- Reduce the amount of time needed to accurately document a service.

Swagger UI

Swagger UI offers a web-based UI that provides information about the service, using the generated OpenAPI specification.

2.2.5 WSL

The **Windows Subsystem for Linux (WSL)** lets developers run a GNU/Linux environment – including most command-line tools, utilities, and applications – directly on Windows, unmodified, without the overhead of a traditional virtual machine or dualboot setup [Microsoft, 2022c].

WSL 2

WSL 2 is a new version of the Windows Subsystem for Linux architecture that powers the Windows Subsystem for Linux to run ELF64 Linux binaries on Windows. Its primary goals are to increase file system performance, as well as adding full system call compatibility [Microsoft, 2022c].

This new architecture changes how these Linux binaries interact with Windows and your computer's hardware, but still provides the same user experience as in **WSL 1** (the current widely available version) [Microsoft, 2022c].

Individual Linux distributions can be run with either the **WSL 1** or **WSL 2** architecture. Each distribution can be upgraded or downgraded at any time and you can run **WSL 1** and **WSL 2** distributions side by side. **WSL 2** uses an entirely new architecture that benefits from running a real Linux kernel [Microsoft, 2022c].

2.2.6 Docker

Docker is a software development platform for virtualization with multiple Operating systems running on the same host. It helps to separate infrastructure and applications in order to deliver software quickly. Unlike Hypervisors, which are used for creating **Virtual machine (vm)**, virtualization in Docker is performed on system-level, also called Docker containers [Taylor, 2022].

2.2.6.1 IBM Blockchain platform

IBM Blockchain Platform (IBP) is an IBM Cloud™ M product built on Fabric. It automates fabric deployment and provides tools to manage your network, taking you from the initial development of an enterprise network to the deployment of applications [Morris et al., 2018].

IBM Blockchain Platform helps you create a Blockchain network with a few clicks and provides an easy-to-use interface to create channels and deploy chaincode. When you're ready to expand your network, IBP can also easily invite new members, create channels in your network, customize governance policies, manage network participant credentials, and more [Morris et al., 2018].

IBM Blockchain Platform has multiple service levels for different stages of development. The Starter plan, currently in beta, is ideal for those developing and testing their business network. The IBP Starter Program offers a fully configured fabric network with two member organizations. Each organization consists of a peer, a CA, and an ordering service shared by the network. It provides sample applications to get you started and quickly experience trading assets on the live network [Morris et al., 2018].

Used to test Smart Contracts in the development environment.

2.2.7 Hyperledger fabric

In 2015, the Linux Foundation founded the Hyperledger project, which aims to unite different industries, promoting Blockchain technology, while encouraging a collaborative approach towards the development of Blockchain technologies through a community process, with intellectual property rights that encourage open development and the adoption of standards over time [Linux foundation, 2022].

Hyperledger Fabric is a Blockchain project, differing from the others in that it is private and permissioned, with members registering through a Membership Service Provider (MSP). The data resulting from this ledger can be stored in several formats, consensus mechanisms can be exchanged inside and outside, and it supports different MSPs [Linux foundation, 2022].

To maintain privacy for those who wish to do so, Hyperledger Fabric allows channels to be created so that defined processes are not publicly shared.

Hyperledger Fabric smart contracts are written as a chaincode and are invoked by an application external to the Blockchain when that application needs to interact with the ledger. In most cases, the chaincode interacts only with the database component of the ledger, the global state, and not with the transaction log [Linux foundation, 2022].

A chaincode can be implemented in several programming languages. Currently, Go, Node.js and Java are supported.

Transactions must be recorded in the ledger in the order they occur, even though they may be between different sets of participants in the network. For this to occur, the order of transactions must be established and a method to reject incorrect transactions that have been entered into the ledger in error (or maliciously) must be implemented [Linux foundation, 2022].

The Hyperledger Fabric is designed to allow newcomers to the network to choose a consensus mechanism that best represents the existing relationships between participants [Linux foundation, 2022].

2.2.7.1 Consensus Algorithms in Hyperledger Fabric

Hyperledger Fabric supports various consensus algorithms that can be chosen based on the desired characteristics of the network. Two commonly used consensus algorithms in Hyperledger Fabric are the Kafka Consensus and the Raft Consensus [Linux foundation, 2022].

- **Kafka Consensus** - The Kafka Consensus algorithm in Hyperledger Fabric is a Byzantine Fault Tolerant (BFT) consensus protocol. It is based on a distributed log that provides fault tolerance and allows for a high degree of scalability. Kafka Consensus ensures agreement among participating nodes even in the presence of malicious or faulty nodes, making it suitable for enterprise-grade blockchain networks [Linux foundation, 2022].
- **Raft Consensus** - The Raft Consensus algorithm in Hyperledger Fabric is a leader-based consensus protocol designed for ease of understanding and implementation. It simplifies the consensus process by electing a leader who coordinates the agreement among the participating nodes. Raft Consensus provides fault tolerance and supports dynamic reconfiguration of the network, allowing nodes to join or leave the network seamlessly [Linux foundation, 2022].

2.2.8 Fablo and Fablo REST

Fablo is a simple tool to generate the Hyperledger Fabric Blockchain network and run it on Docker. It supports RAFT and solo consensus protocols, multiple organizations and channels, chaincode installation and upgrade [Fablo, 2022a].

Fablo REST is a simple REST API interface for Hyperledger Fabric Blockchain network. Supported features [Fablo, 2022b]:

- Enroll, reenroll, register and list identities;
- Discover the network for a given channel;
- Query and invoke chaincode.

Fablo REST should work with any available Hyperledger Fabric network, however it is also integrated with Fablo, a simple tool to generate the Hyperledger Fabric Blockchain network and run it on Docker [Fablo, 2022b].

2.3 Languages and frameworks

2.3.1 C# / ASP.NET Core

C# is considered a compiled language [csh, 2022] by some but still has similarities to the intermediate approach. When C# is compiled, the code is transformed into an intermediate language



Figure 2.4: Fablo (extracted from [Fablo, 2022a]).

(IL) and stored until the code is needed, similar to what Node.js and Python do. In Windows, the IL files have the file extension “.dll” or “.exe”. When the code is to be executed, the intermediate language is compiled using just-in-time compiling. It seems like the words “compiled language” have many interpretations. Still, considering how C# is compiled and its similarities to Node.js compiling, it appears that C# uses an intermediate approach rather than a pure compiling method.

ASP.NET Core [Daniel Roth and Luttin, 2022] is an open-source web framework managed by Microsoft that uses C# as a programming language. The original ASP.NET framework can only be hosted on Windows, however, the Core edition that we will use in this study can also be hosted on Linux machines. ASP.NET Core is an asynchronous framework [Rousos, 2022] in that way that it uses a thread pool to distribute the workload to handle concurrent requests.

This framework was used to develop the project’s API.

2.3.2 Microsoft SQL Server

Microsoft SQL Server is a relational database management system (RDBMS) that supports a wide variety of transaction processing, business intelligence and analytics applications in corporate IT environments. Microsoft SQL Server is one of the three market-leading database technologies, along with Oracle Database and IBM’s DB2 [Adam Hughes, 2022].

Like other RDBMS software, Microsoft SQL Server is built on top of SQL, a standardized programming language that database administrators (DBAs) and other IT professionals use to manage databases and query the data they contain. SQL Server is tied to Transact-SQL (T-SQL), an implementation of SQL from Microsoft that adds a set of proprietary programming extensions to the standard language [Adam Hughes, 2022].

This database was used to store offchain data.

2.3.3 Go

Go, also known as Golang, is an open-source, compiled, and statically typed programming language designed by Google. It is built to be simple, high-performing, readable, and efficient.

This was the programming language chosen to program the Smart Contracts.

2.3.4 Ionic framework with Angular

Ionic framework is an open-source UI toolkit for building performant, high-quality mobile apps, desktop apps, and progressive web apps using web technologies such as **Hyper Text Markup Language (HTML)**, **Cascading Style Sheets (CSS)**, and JavaScript. It allows developers to build once and run everywhere. It was created by Max Lynch, Ben Sperry, and Adam Bradley of Drifty Co. in 2013. The first beta version of the Ionic framework was released in March 2014.

Ionic provides a set of tools for building native iOS and Android applications, and mobile-ready Progressive Web Apps, using familiar web libraries, frameworks, and languages.

The figure 2.5 shows the development with Ionic and Angular. **HTML**, **CSS**, and JavaScript is used to build the application and the capacitor is used to use native components and deploy the application to different environments.

This was the framework used to develop the APP.

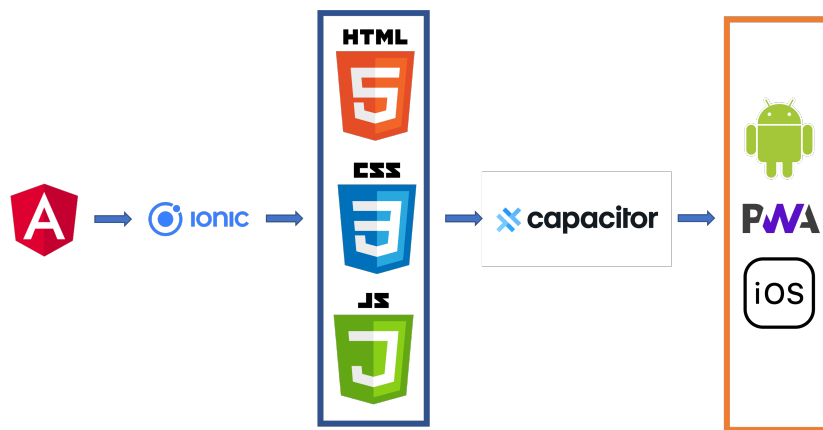


Figure 2.5: Ionic Framework with Angular development.

Chapter 3

State of the art

In this chapter we start by making a presentation of blockchain technology. Then we present some approaches that implement the traceability of food products using blockchain. At the end, we present existing frameworks that allow tracking olive oil products.

3.1 The Blockchain Technology

When talking about traceability solutions, especially for food industry, Blockchain technology is pointed out as one of the best suited for implementing the traceability of products [Cruz et al., 2019], thus driving the improvement of food safety.

Since the introduction of Bitcoin in 2008 [Nakamoto and Bitcoin, 2008], the science behind Blockchain has been applied to different business scenarios and has been very successful, providing proof of its functionality. This technology allows various organizations and other entities to work together, as this technology promotes its transparency and fault tolerance [Buttafoco, 2019].

Blockchain technology provides not only transparency, but traceability and security for transactions, real-time data and smart contracts in order to meet the needs of its users [Nandi et al., 2021]. This technology also has the possibility to integrate with other areas, such as Big Data, Artificial Intelligence (AI), Internet of Things (IoT), Cloud Computing, among others [Tsai et al., 2021].

Blockchain is currently used as a decentralized database for traceability between operators in value chains. Characteristics such as transparency, durability, and process integration in value chains are achieved by storing transactions in chronological order, which create a permanent and immutable record. Each tangible, traceable item must be connected to its intangible digital twin, whose information will be recorded on the platform.

The IoT allows a simple way to do just that. According to [Tao et al., 2018], IoT technology can help aggregate data at any stage of the product lifecycle with devices that can ensure perfect traceability and reveal its complete history. By combining it with Blockchain technology, this information becomes immutable, private and transparent when it comes to data sharing as well as tokenization of assets [Pradana et al., 2020], providing token ownership that acts as a digital

watermark, corresponding to physical ownership [Jacobovitz, 2016]. Whenever a transaction happens for a specific product, its cycle data in the process can be captured by the use of IoT devices and managed properly with the use of Blockchain technology.

The figure 3.1 illustrates an example of a blockchain with n blocks, where each successive block contains the hash of the previous block, a timestamp, the transaction information, the nonce number for the mining process, and other details necessary for the protocol to work.

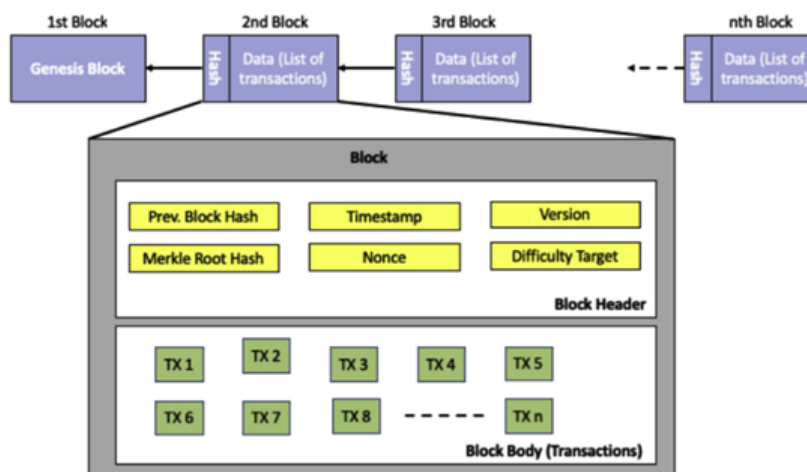


Figure 3.1: Blockchain blocks example (extracted from [Kamilaris et al., 2019]).

3.1.1 Types of Blockchain

As mentioned in the previous section since of the introduction of Blockchain technology and the successful creation of the first blockchain in 2008, Bitcoin, a new type of blockchain, has emerged specifically to alleviate some of the existing problems. The idea is to increase the possibility of using the same technology for other purposes.

Ethereum (founded in 2013) is a decentralized platform based on blockchain technology with a new feature called smart contracts. These smart contracts are digital programs (written in specific language and compiled in bytecode), run by a Turing-complete virtual machine, the Ethereum Virtual Machine (EVM). These contracts can be used to verify that a potential transaction meets certain conditions [Amani et al., 2018], namely H. If they meet some logical conditions.

In 2015 came Hyperledger, a new platform developed by the Linux Foundation, IBM, SAP, and Intel. Several open source Blockchains and related tools. Hyperledger Fabric is an open source blockchain platform Designed for cross-company and on-premise environments. Other platforms have since emerged, such as Corda, an open-source blockchain platform created by R3. Developed for the financial industry [Valenta and Sandner, 2017]. The programming language used in a smart contract depends on the blockchain it is targeting. In Ethereum, Solidity can be used, Hyperledger can use Go and Java programming languages, and Corda can use Smart Contracts can be written in Kotlin or Java [Valenta and Sandner, 2017].

Regarding permission types, Blockchains can be classified as public, private or consortium [Agrawal et al., 2021, Zheng et al., 2018]:

- **Public Blockchains** – this type of blockchain is open to anyone who wants to participate anonymously and anonymously Verify the transaction. Since then, these networks have the most secure of all blockchain types Nodes are random validating readers who wish to become validators. Let's take the largest Today's blockchain networks such as Bitcoin and Ethereum.
- **Private Blockchain** - is a private network owned by an entity that has full control over who can participate In the network and the rules governing its operation. This type of application is used in case Where data needs to be kept private and in full control of it and who can access it.
- **Consortium chain** - can be accessed by known people allowed by network rules, and also defines validators. Such a blockchain is important when multiple trusted entities require Working together must have equal voting rights in the chain.

Consensus participation has two modes of operation: **Permissionless** and **Permissioned** [Valenta and Sandner, 2017].

- **Permissionless** - anyone can join and interact with the blockchain. Depends on the consensus protocol. The Ethereum platform is one of many examples of such blockchains.
- **Permitted** - Different participants have different permissions, so only a few correctly identified participants can create new blocks in the blockchain. Hyperledger and Corda blockchains are examples of this.

3.1.2 Consensus protocols

The modes of operation are associated and related to the consensus protocol used. The consensus protocol defines how the consensus among participants in order to add a new block, is reached. It is used to know if, when and by whom, a block can be added to the blockchain. By abiding to a consensus protocol, each user must verify the authenticity of each block before agreeing to include it in the blockchain. There are several types of consensus protocols [Zheng et al., 2018, Bamakan et al., 2020]:

- **Proof-of-Work (PoW)** - based on the competition between miners to solve complex mathematical problems that give the fastest miners the possibility to validate the transaction and earn a reward. This type of protocol requires a huge computational power and demands high energy consumption to work.
- **Proof-of-Stake (PoS)** - gives privilege to miners with more tokens and more time on the network, which are chosen randomly to validate transactions. This type of protocol is safer

as several miners are selected to validate a transaction and if the validation is done poorly, they lose a percentage of their tokens.

- **Proof-of-Elapsed Time (PoET)** - based on a random counter at each node in the network that determines who has the right to create the next node and wins the reward. This type of protocol does not require a lot of processing power, being based on randomness, and any node is equally candidate to validate the next transaction.
- **Proof-of-Authority (PoA)** - based on the appointment of trusted validators by the network in which they are obliged to show their identification to the network and assume as collateral their own civil reputation. This type of blockchain is more energy efficient and faster due to the fact that it is not necessary to process any type of algorithm to validate a new block as these are entirely entrusted to the validators. For these reasons this type of protocol is only used in private networks that determine who are the trusted people.
- **Proof-of-Luck (PoL)** - validator's selection is made through a random value between 0 and 1, where the node with the highest number is the winner and is rewarded with the ability to validate the next transaction [Bada et al., 2021].
- **Practical Byzantine Fault Tolerance (pBFT)** - The idea behind this consensus method is the resolution of the Byzantine General problem. It works relying on the Bizantine Fault Tolerance algorithm, saving the network from failing because of malicious nodes, connection losses, or delay from peers. In this consensus protocol, all nodes must participate in the voting process, and the consensus is reached when more than two-thirds of the nodes are in favor to add the next block to the chain [Bamakan et al., 2020].

There are other consensus protocols, such as Proof-of-Weight, Proof-of-burn, Proof-of-Capacity, Proof-of-Importance, Proof-of-Activity, etc., but the ones listed above are the most used [Andrey and Petr, 2019]. The consensus protocol used is one of the factors that influence the energy consumed by the blockchain. According to [Bamakan et al., 2020], the used consensus protocol not only influences the energy consumed by the blockchain, but also influences the performance and the scalability of the blockchain.

3.2 Blockchain-based solutions for tracking food products

Blockchain technology fits all needs to implement product traceability. Therefore, this technology has been used in many traceability solutions, especially for tracking food products [Alves. et al., 2021, Rosado da Cruz and Cruz, 2020]. Some of this solutions are presented next.

In [Cruz and Rosado da Cruz, 2020], the authors presented a Blockchain-based solution to trace fish and fishery products from harvest (sea, rivers or lakes) or aquaculture to the final consumer. The authors used Ethereum blockchain to implement traceability by storing information about each activity involved in the fish and fishery value chains [Cruz et al., 2019].

In [Alves. et al., 2021] the authors presented a Blockchain-based solution to implement traceability of Protected Designation of Origin (PDO), Protected Geographical Indication (PGI) and Traditional Specialty Guaranteed (TSG) products from their origin to the final consumer. The proposed platform helps assuring the source of the products, avoiding forgeries and providing transparency by using Hyperledger Fabric blockchain[Alves. et al., 2021].

In [Miatton and Amado, 2020] the authors presented a platform blockchain-based to trace the coffee bean produced in Colombia. The platform uses the Hyperledger Fabric blockchain to store the data about each activity in the coffee value chain.

In [Biswas et al., 2017] the authors present a blockchain-based platform to trace the supply chain of the wine.

3.3 Blockchain and sustainability in the value chain

The information and security offered by blockchain can help production companies to perform less environmentally harmful practices in agriculture, livestock farming or fishing [Cruz et al., 2019]. The new sustainable business models that are expanding in contemporary society increasingly feature digitalization. This is due to the implementation of Industry 4.0 [Tsai et al., 2021].

Blockchain can contribute to the development and coordination of new sustainable business models through sharing, optimization, virtualization and, most importantly, exchange, thus constituting a driving force for achieving sustainability goals [Tsai et al., 2021]. This technology can lead to waste reduction and the promotion of an increasingly traceable and safe product offering, increasing social responsibility, which should be considered as a key element to achieve sustainability goals. It can also contribute significantly, since it provides a complex interconnected system capable of updating servers simultaneously and irreversibly. Thus, resorting to the use of Blockchain technology can contribute to increasing the performance of companies and also more effectively achieve sustainable goals [European Commission, 2007].

3.4 Blockchain in the traceability of the olive oil production chain

When talking about traceability solutions, especially for food industry, Blockchain technology is pointed out as one of the technologies that are best suited for implementing traceability of a product [Cruz et al., 2019], thus driving the improvement of food safety.

We can see in figure 3.2 an example of traceability with the use of Blockchain technology. We have the entire physical process related to the digital process. Regarding the digital process, it consists of several digital technologies (e.g. QR Codes, Radio-Frequency Identification (RFID), Near-Field Communication (NFC), etc.) that serve to streamline the process of storing information that is accepted by all participating parties. The information that is recorded (called blocks) in each transaction is validated by the business partners of the food supply network, forming a consensus among all participants. After each block is validated, it is added to the chain of transactions, thus becoming a transparent, reliable, and digital process.

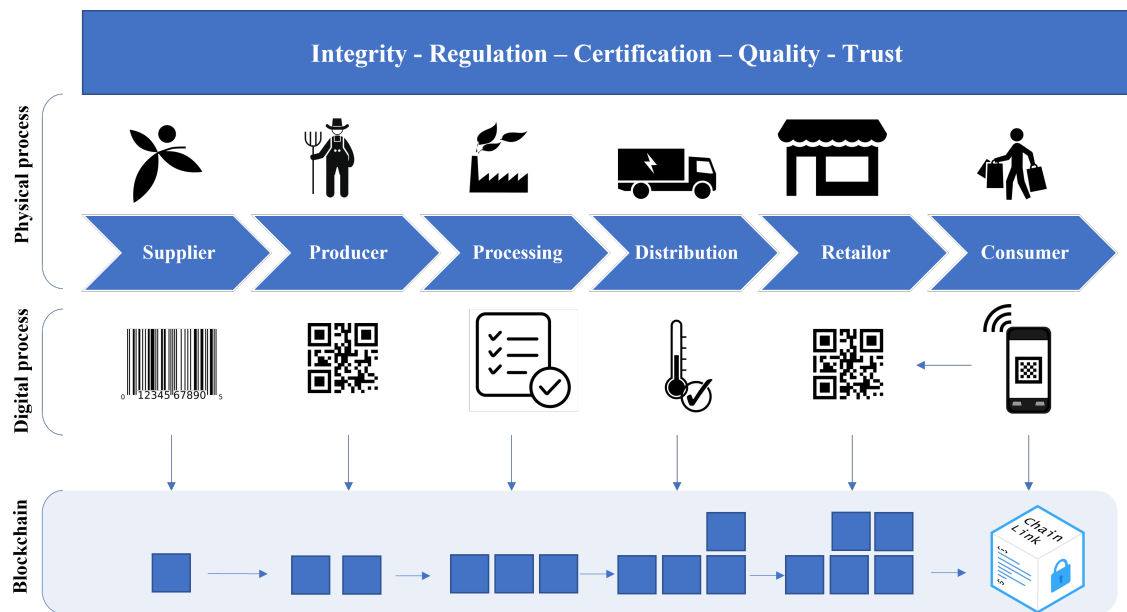


Figure 3.2: Value chain of a food product.

There are already several platforms that resort to the use of Blockchain technology to trace food, such as wine, coffee and fish [Biswas et al., 2017, Pradana et al., 2020, Cruz and Rosado da Cruz, 2020, Rosado da Cruz and Cruz, 2020].

In the case of Olive Oil traceability, the platforms considered most relevant are the following:

3.4.1 Devoleum

Devoleum is an open source web application, which transforms physical or digital data in the food value chain into true and authentic stories. It is a decentralized solution using Ethereum Blockchain, AI and IPFS technologies [Mercuri et al., 2021].

Devoleum records all stages of the production process through smart contracts, which allows the reduction of transaction costs, since the transparency of the elements contained in the smart contract and combined with the impossibility of data manipulation by third parties and, consequently, eliminates intermediaries from the food value chain [Mercuri et al., 2021].

The goal of this platform is to offer a transparent and accessible service to consumers and producers, having as an added value the immutability of the Blockchain, thus proving its authenticity of the impossibility of manipulating the stored data [Mercuri et al., 2021].

3.4.2 BRUSCHETTA

BRUSCHETTA is an Blockchain-based application for traceability and certification of **EVOO**.

The goal of BRUSCHETTA is to monitor all production processes in order to obtain critical information to estimate the quality of the final product [Arena et al., 2019].

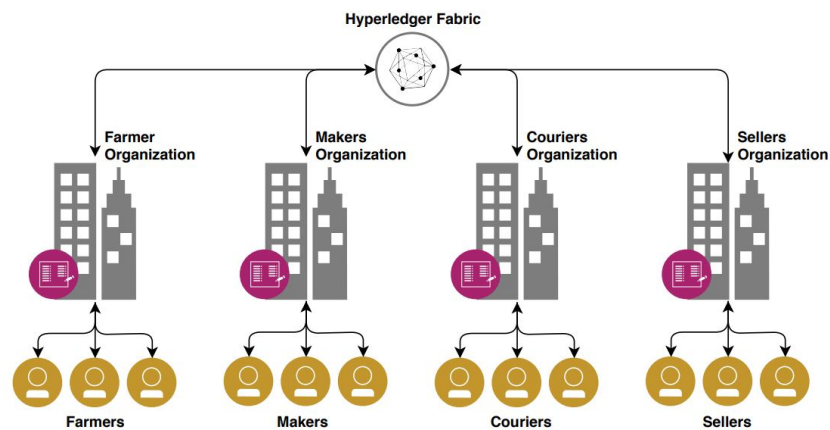


Figure 3.3: BRUSCHETTA architecture (extracted from [Arena et al., 2019]).

As can be seen in 3.3, BRUSCHETTA uses Hyperledger Fabric as a Blockchain technology. BRUSCHETTA divides the value chain of **EVOO** production into six distinct but interconnected processes [Arena et al., 2019]:

1. Olive cultivation;
2. Olive harvest;
3. Transportation;
4. Olive transformation;
5. Packaging of olive oil;
6. Transporting the olive oil.

So that the entire history of each process can be reconstructed when needed, BRUSCHETTA designs a different profile for each process, where specifically each profile represents one or more transactions that are generated by nodes and are included in the Hyperledger Fabric Blockchain [Arena et al., 2019].

Chapter 4

Problem analysis

All over the world, there are different standards, for different destinations, that regulate the information required for traceability [Kamilaris et al., 2019]. Some entities choose to register this information beyond the established one, in order to guarantee higher security levels. However, the opposite is also true. For example, a recent study revealed that in Brazil, out of 4577 products, only about 1% offer traceability information [Matzembacher et al., 2018].

In Portugal, as in other **European Union (UE)** countries, **UE** legislation must be followed to ensure the "ability to trace and follow a food, feed, food-producing animal, or expected to be incorporated into a food or feed, at all stages of production and distribution" [European Commission, 2007].

4.1 Value chain of food products

A value chain of a food product is based on and distributed over several stages where each stage may have a different actor [Caro et al., 2018, Cruz et al., 2019] as can see it in figure 4.1. Thus, the main stages are as follows:

- **Production** - this stage represents all agricultural activities and practices by the producer.
- **Processing** - this stage represents the transformation, in whole or in part, of a primary product into one or more secondary products. Along the value chain of a product, there may be several transformations.
- **Packaging** - the stage of packaging in which the individual product can be uniquely identified by the production lot code with information such as the day of production.
- **Transportation** - once the product is packaged, it needs to be distributed. The delivery date depends from product to product, and there may be a storage stage here.

After distribution, the product is delivered to the retailer who will be responsible for selling the product to the **final consumer**. This will be the final actor in the chain, who buys the product.

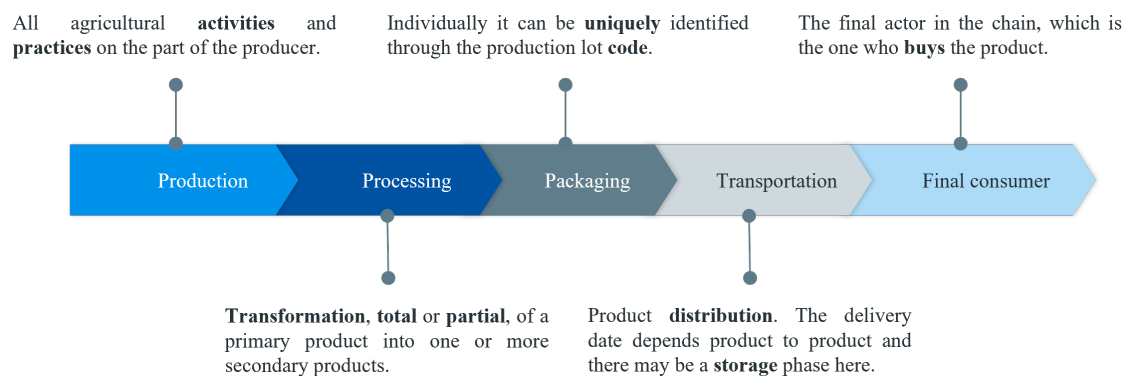


Figure 4.1: Value chain of a food product.

As previously mentioned, nowadays the market has consumers who want to know the products they consume, and therefore they want to be aware of this whole process, all the stages and which actors are involved, as well as the required quality standards, production methods, etc. There are already several platforms that support the traceability of food products, from production to the final consumer, such as [Biswas et al., 2017, Caro et al., 2018, Oliveira et al., 2021].

4.2 The olive oil production process

The European Union is the world's largest producer of olive oil, and it also leads in terms of consumption and exports. With olive groves covering approximately 4 million hectares of land, mainly in Mediterranean countries within the EU, traditional, intensive, and super-intensive olive farming methods are combined to produce around 67% of the world's olive oil [European Commission, 2020].

Italy and Spain are the EU countries with the highest consumption of olive oil, each consuming approximately 500,000 tonnes per year. Greece, on the other hand, boasts the highest per capita consumption of olive oil in the EU, with approximately 12 kg per person per year. Portugal is also one of the largest producers of olive oil within the EU, accounting for around 5% of its total production [European Commission, 2020].

Overall, the EU consumes approximately 53% of the world's olive oil production. Thanks to its long history of olive cultivation, the EU has established itself as the dominant force in the global olive oil market [European Commission, 2020].

Olive oil is a type of edible oil that is made from the fruit of the olive tree (*Olea europaea*). The olive tree is native to the Mediterranean region and has been cultivated for thousands of years for its oil, as well as for its fruit and wood. The process for making olive oil involves several steps, including harvesting, sorting, washing, grinding, malaxation, separation, and filtering. In this article, we will take a closer look at each of these steps and discuss the scientific principles and techniques involved in the olive oil production process [Safa et al., 2017].

- **Harvesting:** Olives are typically harvested by hand using rakes or by shaking the trees. They can also be harvested mechanically using machines. The timing of the harvest is important, as it can affect the quality and flavor of the oil. Olives are generally harvested when they are ripe, but not overripe, as overripe olives can produce oil that is rancid or off-flavored. In addition, olives that are harvested too early may not have fully developed their oil content, resulting in lower yields [Safa et al., 2017].
- **Sorting:** After the olives are harvested, they are sorted to remove any that are overripe, underripe, or damaged. This is an important step in the production process, as it ensures that only high-quality olives are used to make the oil. The sorting process can be done manually or mechanically, using machines such as vibrating sorters or air sorters [Safa et al., 2017].
- **Washing:** The olives are washed to remove any dirt, leaves, or other debris that may be present. This is typically done using water and may involve the use of brushes or other mechanical means to loosen any dirt or contaminants.
- **Grinding:** The olives are ground into a paste using large stones or steel blades. The grinding process breaks down the olive tissue, releasing the oil from the cells. The paste is then mixed or "malaxated" to allow the oil droplets to come together and form a larger mass. This process can be done using machines or by hand [Safa et al., 2017].
- **Malaxation:** During the malaxation process, the paste is mixed or agitated to allow the oil droplets to come together and form a larger mass. This can be done using a machine called a malaxer, which uses a combination of heat and mechanical action to mix the paste. The length of the malaxation process can vary, but it typically lasts for 30 minutes to an hour [Safa et al., 2017].
- **Separation:** The oil is separated from the water and solids in the paste using a centrifuge or other mechanical means. The paste is placed in a container, such as a drum or tank, and spun at high speeds, causing the heavier water and solids to settle to the bottom, while the lighter oil rises to the top [Safa et al., 2017].
- **Filtering:** The oil is filtered to remove any remaining solids or impurities. This may be done using a variety of methods, such as passing the oil through a filter cloth or using a machine called a decanter to separate the oil from the solids [Safa et al., 2017].
- **Bottling:** The oil is bottled and labeled for sale. The oil is typically bottled in dark glass or metal containers, which helps to protect it from light and oxygen, both of which can contribute to the breakdown of the oil [Safa et al., 2017].

The figure 4.2 shows the difference between traditional an modern process (used in this project);

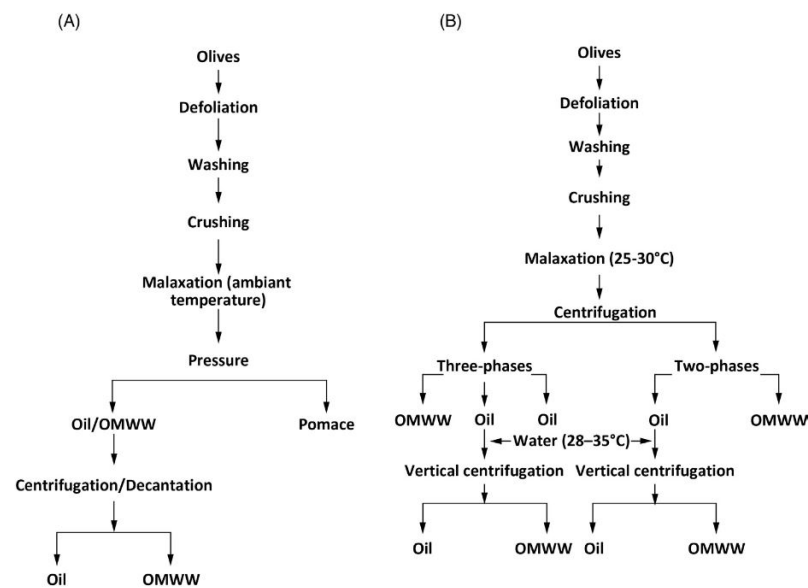


Figure 4.2: Traditional and modern processes in olive oil production (extracted from [Safa et al., 2017]).

(A) Traditional process. (B) Modern process.

4.3 The olive oil production business process

The olive oil manufacturing business process model is shown in Figure 4.3, by using BPMN language. The process is internal to an olive oil production company and represents the main activities from olive cultivation to bottling.

As we may see in figure 4.3, the process starts in the olive grove where the olive trees are treated throughout the year so that, at the right time, the olives are harvest (second activity in figure 4.3). After that, the olives are transported to the olive oil mill where they are weighed, washed and their quality evaluated. When the quality is not good enough, the olives are discarded. After that the olives are crushed, then are blended, centrifuged and put in vats for some time (usually more than a month). After that the olive oil is bottle up. Each bottle is properly labeled.

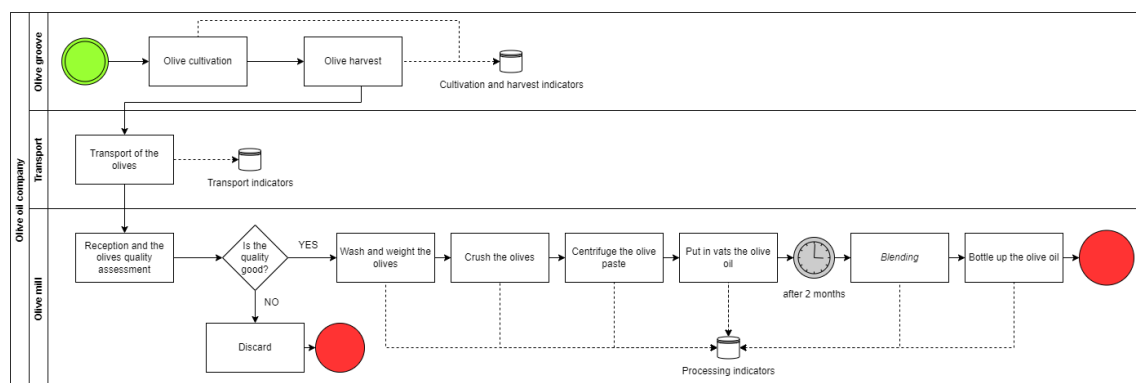


Figure 4.3: Olive oil production BPMN.

4.4 The olive oil value chain traceability

The production of Olive Oil involves several steps such as retailing, washing, grinding, beating and the separation of EVOO [Buttafoco, 2019]. In order to maintain the natural organoleptic characteristics, according to the European Commission Regulation n 1513/2001, olives should be processed immediately after harvest to minimize oxidation and preserve low acidity [Safa et al., 2017]. All operations at the mill as well as humidity and temperature conditions (for example) can influence the quality of the Olive Oil. Traceability also prevents incorrect labeling regarding the geographical origin and cultivation of the olive [Violino et al., 2019].

In the case of Olive Oil, traceability assumes a great relevance because it allows to guarantee the origin of the products and raw materials, more specifically, the identification of the different varieties of olives, protecting the consumers against frauds [Muzzalupo et al., 2015].

Like the traceability of other food products, the traceability of olive oil is very important to ensure the quality and integrity of the product, increasing consumer confidence. To this end, there is the Commission's Implementing Regulation (UE) No 29/2012 on marketing standards for olive oil. This regulation states the need to establish, at the European Union level, a mandatory system of designation of origin, limited to EVOO and Virgin Olive Oil, which complies with precise conditions [Baer, 2015]. Traceability prevents incorrect labeling regarding the geographical origin and cultivation of the olive, thus ensuring correct information for consumers [Violino et al., 2019]. EVOO costs 4 to 5 times more than other vegetable oils. This oil has greater and better nutritional and organoleptic properties, but this is accompanied by higher production costs. Thus, the higher cost is necessary to guarantee the best quality. The implementation of traceability therefore represents a cost to producers [Violino et al., 2019].

4.5 The olive oil sustainability indicators

Traceability of quality attributes, in the olive oil sector, is of great importance, as it makes possible to guarantee the origin of products and raw materials, including the identification of the different varieties of olives, thus protecting consumers against fraud [Muzzalupo et al., 2015]. Traceability

prevents incorrect labeling in relation to the geographical origin and olives quality attributes, thus ensuring correct information for consumers [Tao et al., 2018].

Like traceability of other food products, the traceability of olive oil is important to guarantee the quality and integrity of the product, increasing consumer confidence. The European Commission's Implementing Regulation (UE) No. 29/2012, concerning the marketing standards for olive oil, affirms the need to establish, at the European Union level, a mandatory regime of designation of origin, limited to **EVOO** and Virgin Olive Oil, which obeys precise conditions [Baer, 2015].

EVOO costs 4 to 5 times more than other vegetable oils. This oil has greater and better nutritional and organoleptic properties, but this is accompanied by higher production costs. Thus, the higher cost is a side-effect of ensuring the best quality. Implementing traceability features, also represents a cost [Violino et al., 2019].

Table 4.1: Table of olive oil sustainability and quality indicators.

		Sustainability			Quality		
		Water	Fuel	Energy	Humidity	Temperature	Execution times
Olive grove	Olive Cultivation	✓		✓			✓
	Olive Harvest		✓		✓	✓	✓
Transportation	Transport		✓		✓	✓	✓
Olive oil mill	Washing	✓		✓	✓	✓	✓
	Crushing			✓	✓	✓	✓
	Centrifugation			✓	✓	✓	✓
	Bottling			✓	✓	✓	✓

4.6 Requirements

This section of the project documentation provides a detailed overview of all **Functional Requirement (FR)** and **Non Functional Requirement (NFR)** that must be met for the project's success. By describing each **FR** and **NFR**, we can have a clear understanding of what is expected of them to meet the project's objectives. The **FR** outline the specific features and functions that the system must perform to meet the user's needs. On the other hand, the **NFR** describe the system's performance criteria, including factors like reliability, usability, scalability, and security, that must be met to ensure the system's overall success. By including both **FR** and **NFR** in this section, we can ensure that they have a comprehensive understanding of what needs to be achieved, allowing them to work efficiently and effectively towards meeting the project's goals.

4.6.1 Functional requirements

A **FR** is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements in Software Engineering are also called Functional Specification [Martin, 2022].

All **FR** for the project are listed and described in detail in the reference table 4.2. The table includes a comprehensive breakdown of each requirement, consisting of its unique code, a clear description of what it entails, and its priority level.

Table 4.2: Functional requirements.

Code	Description	Priority
FR01	Register and manage tenants to them use the application.	High
FR02	Register and manage olive varieties to create olive oil productions of them.	High
FR03	Register and manage olive groves to create olive oil productions of them.	High
FR04	Register and manage olive mills to relate it in olive oil productions activities.	High
FR05	register and manage productions to be traceable.	High
FR06	Register and manage activities in production to be traceable.	High
FR07	Register and manage measurements in activities to be traceable.	High
FR08	Take pictures related to productions.	Medium
FR09	Take pictures related to production activities.	Medium
FR10	Take pictures related to production activity measurements.	Medium
FR11	Download QR Code related to production to sanc it and check the production.	Medium
FR12	Download QR Code related to production activity and check the traceability.	Medium
FR13	Download QR Code related to activity measurement to scan it and check the measurement.	Medium
FR14	Scan a QR Code and check the production traceability.	Medium
FR15	Read all application logs to check if application has errors or warnings.	Low
FR16	Read all blockchain logs to check if blockchain has errors.	Low

Continued on next page.

Table 4.2 – continued from previous page.

Code	Description	Priority
FR17	Certify the production.	High

4.6.2 Non Functional Requirements

NFR specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system [Martin, 2022].

All **NFR** for the project are detailed in the reference table 4.3. The table comprises a comprehensive list of each requirement, consisting of its unique code and a clear description of what it entails.

Table 4.3: Non Functional Requirements.

Code	Description
NFR01	All code needs to be written in English.
NFR02	The architecture and code needs to be clean (<i>clean architecture</i>).
NFR03	Every API request needs to respond less than 100 milliseconds (ms) .
NFR04	The API needs to be scalable.
NFR05	All external components used in project needs to be in their latest versions.
NFR06	All external components used in project must be open source.
NFR07	All communications with external entities needs to be queued.
NFR08	All API exceptions needs to be logged.
NFR09	All execution times in blockchain needs to be logged.
NFR10	All pictures needs to be resized to have less than 1 Megabyte (MB) .

4.7 Use case diagram

Use Case Diagram captures the system's functionality and requirements by using actors and use cases. Use Cases model the services, tasks, function that a system needs to perform. Use cases represent high-level functionalities and how a user will handle the system. Use-cases are the core concepts of Unified Modelling language modeling [Walker, 2022].

The figure 4.4 provides a visual representation of the use case diagram, showcasing the actions that each actor can perform. For further details on each actor and their corresponding actions, please refer to Table 4.4 which provides a concise description of each actor and their associated tasks.

It is important to note that the use cases related to logs from *Application Owner* actor have the sole purpose of verifying that the entire system is operational.

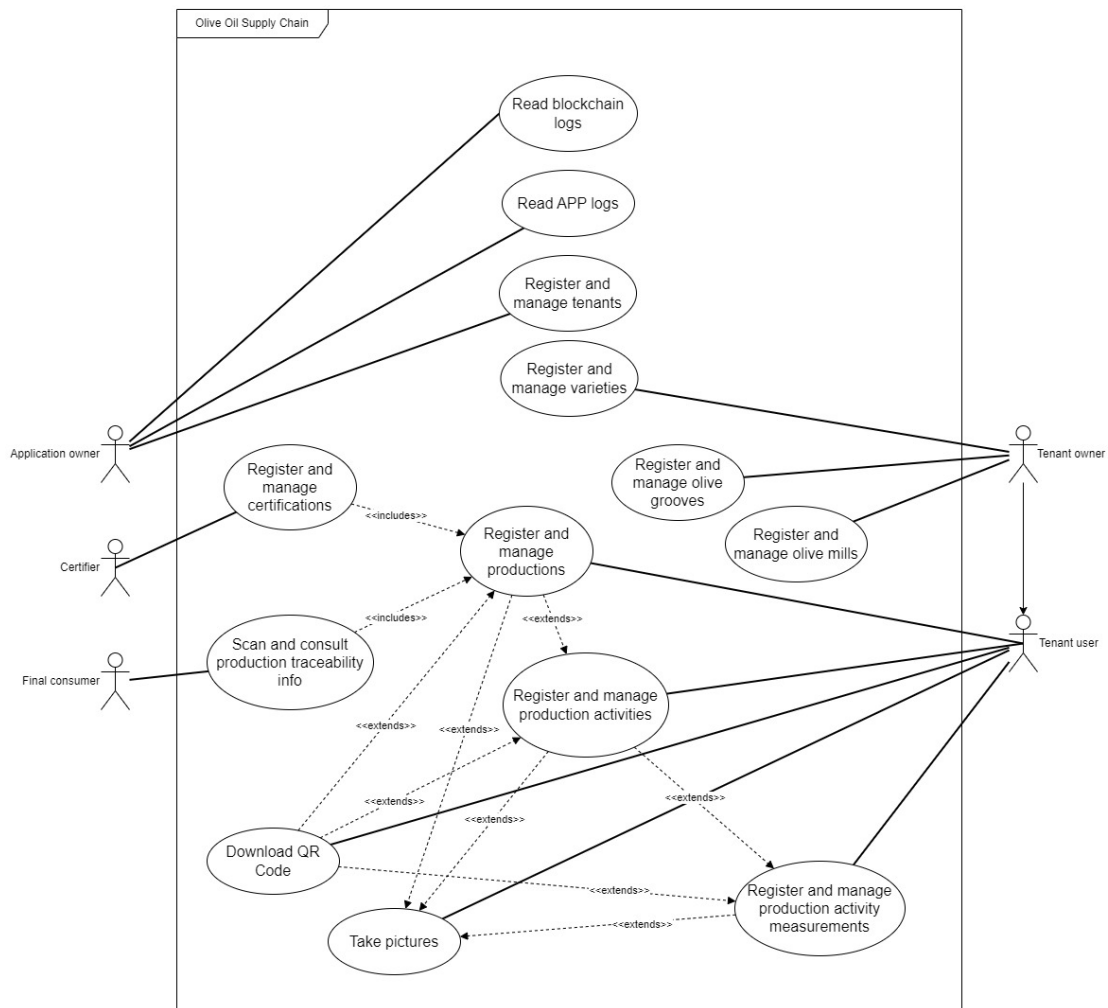


Figure 4.4: Use case diagram.

Table 4.4: Application Users.

User type	Description
Application owner	This user has access to all features in the application.
Tenant owner	This user has access to manage their olive oil company.
Tenant user	This user has access register production activities, measurements, photos and QRcodes.
Certifier	This user has access to certify productions.
Final consumer	This user has access to consult productions.

4.8 Database model

The database model for the problem is illustrated in Figure 4.5, while Table 4.5 provides a comprehensive list of database entities and their respective descriptions. The domain model was created

based on the following statements:

- An olive grove can have one or more varieties, and each variety can be associated with one or more olive groves;
- An olive grove can have one or more productions, and each production is associated with only one olive grove;
- A production can have one or more production activities, and each production activity is associated with only one production;
- Each production activity must have a type and indicate what is generated by that activity;
- A production activity can have one or more input production activities;
- Each production activity can be associated with one olive mill, and each olive mill can be associated with one or more production activities;
- A production activity can have one or more production activity measurements, and each measurement belongs to only one production activity;
- Each production activity measurement must have a measurement type;
- Each entity is associated with a tenant;
- A premium can be associated with one or more productions, and a production can be associated with one or more premiums;
- A production can have one or more certifications;
- A premium production can have one or more certifications;
- A log can have one log type, and each log type can be associated with one or more logs;
- A production, production activity, or production activity measurement can have one or more photos (the relationship is artificial for the sake of generality).

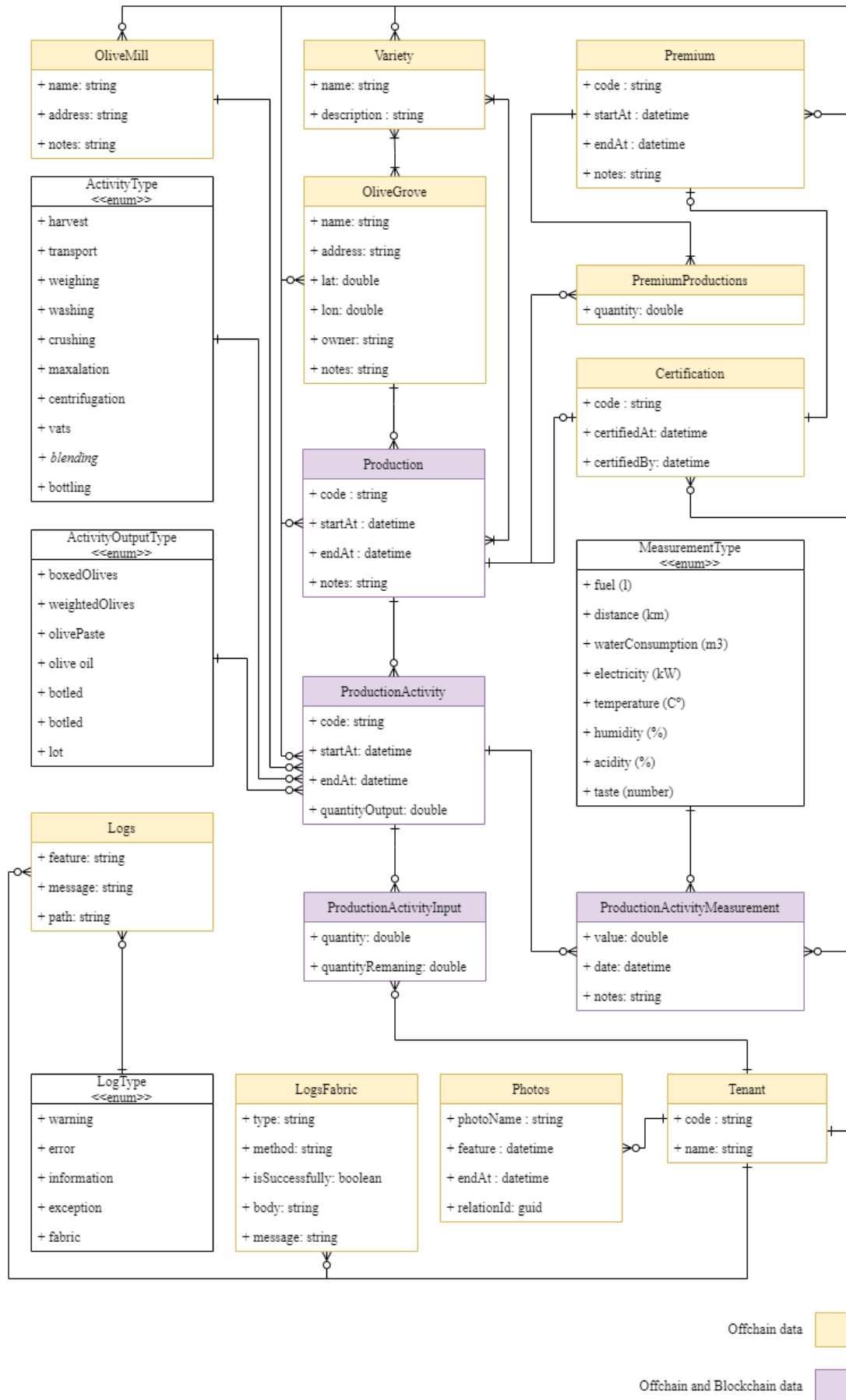


Figure 4.5: Database model.

Table 4.5: Database entities.

Entity	Description
Certifications	This entity represents where certifications are stored.
Logs	This entity represents where API logs (errors, informations and exceptions) are stored.
Logs Fabric	This entity represents where Blockchain logs (status, errors, informations and exceptions) are stored.
OliveGroves	This entity represents where olive groves are stored.
OliveGroveVarieties	This entity represents where the varieties in each olive grove are stored. Each olive grove can have one or more varieties.
OliveMills	This entity represents where olive mills are stored.
Photos	This entity represents where all the the photos (relation id, feature and photo path) are stored.
Premiums	This entity represents where premium productions are stored.
PremiumProductions	this entity represents the list of the productions in each Premium.
Productions	This entity represents where productions are stored.
ProductionActivities	This entity represents where activities in each production are stored.
ProductionActivityProductionActivityInputs	This entity represents where activities origin in one activity are stored.
ProductionActivityMeasurements	This entity represents where measurements in each production activity are stored.
Tenants	This entity represents where tenants are stored.
Varieties	This entity represents where olive varieties are stored.

Chapter 5

Design and implementation

The purpose of this chapter is to provide a detailed explanation of the design and implementation of the project. It covers the use case diagram, domain model, database structure, project architecture, and implementation process.

The chapter begins with the use case diagram, which outlines the system requirements and ensures that the **APP** meets the project needs. The domain model helps to identify the system's key entities and their relationships. The database structure is designed based on the entities identified in the domain model.

The chapter also then specifies the project architecture, including the overall structure of the **APP**, **API**, and blockchain. The architecture was designed to ensure that the different components work together seamlessly and that data flows smoothly between them.

Finally goes on to describe the implementation process, which covers project management, **APP** structure, key screens, **API** and blockchain structure, and data flow between the **APP**, **API**, and Blockchain. The **APP** structure was designed to ensure that the different components work together seamlessly, and the key screens were developed to provide a user-friendly interface. Finally, the **API** and Blockchain structures were designed to ensure efficient data transfer and security.

5.1 Database

According to figure 4.5, and following the code-first approach, we ensured that the database was optimized to address the problem.

Code first approach focuses on designing the application's data model first, and then using that design to create the database schema. By doing this, we can ensure that the database design is in line with the application's requirements and can effectively store and retrieve the data required by the system.

In addition, the inheritance feature was used to enable all entities/tables to share common properties/field These shared properties/fields were identified as follows:

- **Id:** Guid

- **CreatedAt** : DateTime
- **CreatedBy** : string
- **UpdatedAt** : DateTime
- **UpdatedBy** : string
- **RowVersion** : byte[]
- **IsActive** : boolean
- **IsSystem** : boolean
- **IsDeleted** : boolean
- **TenantId** : Guid

5.2 Architecture

The figure 5.1 illustrates the architecture of the project. The data from the application to the database (offchain data) and the Blockchain. This communication occurs via **HTTP** requests and a REST **API**, which allows the application to send and receive data from the database and blockchain.

It is important to note that a valid token (**JWT**) is required for both the connection between the **APP** and the **API**, as well as the connection between the **API** and the Blockchain. This ensures that only authorized users can access and modify the data on the database and Blockchain.

When a record is added to the database (offchain data), it is also inserted into the Blockchain (Fablo network) through a background task using Hangfire. This allows the insertion of data into the blockchain to happen asynchronously, without affecting the performance of the application.

For this proof of concept, we choose the method with a background task because we were unsure if the Blockchain would work as expected, and it was important to save all the data.

In order to maintain the integrity of the data, a background task (Quartz.NET) is run periodically and automatically to verify that the offchain data matches the immutability of the data on the Blockchain. This helps to ensure that the data on the database and blockchain remains consistent and accurate.

5.3 Implementation

This section shows the entire implementation of the project. Starting in project planning, the **APP** structure and design, the **API** and Blockchain structure and the workflow between **APP** to the Blockchain.

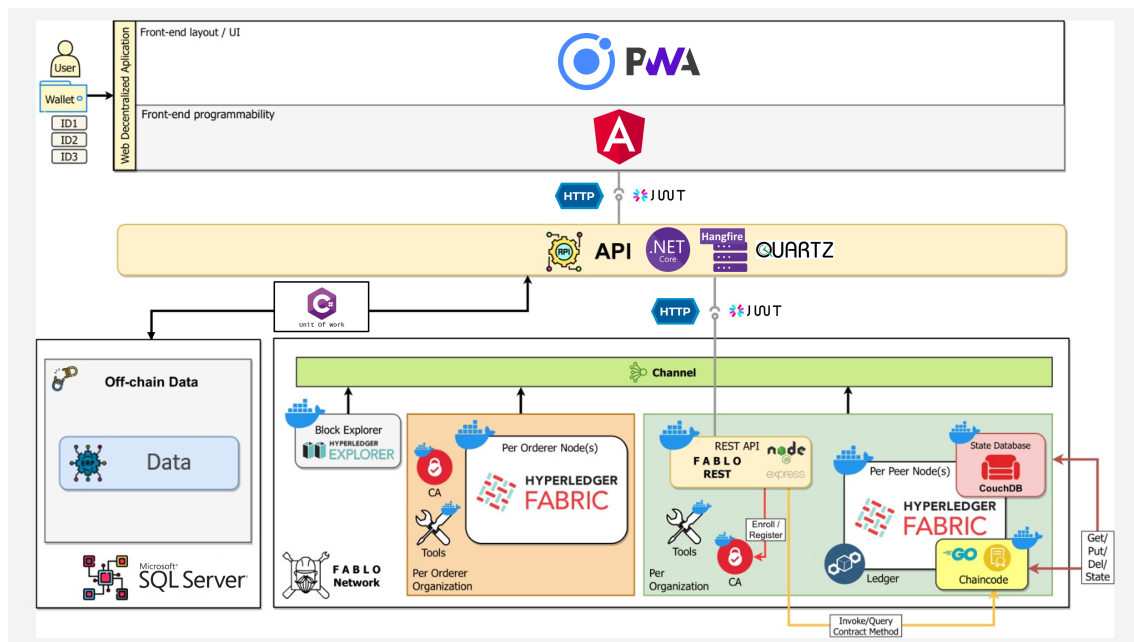


Figure 5.1: Architecture.

5.3.1 Project management

To all project management has been used Azure DevOps described in 2.2.1, and the working methodology chosen was Scrum with 1 month sprints.

In the beginning of the each sprint all packages and plugins was been updated to their latest versions, after that all has been tested and fix all that this update broke. For lucky, no situations were found in every update.

The figures 5.2, 5.3 and 5.4 shows the entire project planning as well as what is done in each sprint.

5.3.1.1 February to May Sprints

February 2022 was the first sprint and it was the Azure DevOps configuration, as well as the the choose of the **APP**, **API** and Blockchain technologies. It also defined the environments (**Development (dev)**) and **Production (prod)**) as well as how to deploy.

- To **dev environment (env)** it was *localhost* hosted in IIS server.
- To **prod env** it was *https://valuechain.wolfmartindustries.pt/* hosted in IIS server.

In the **March** sprint it was the creation of the each project and push to the online repositories. For this it was used the Azure Repos described in 2.2.1.

In the **April** sprint it was the UnitOfWork setting up to connect to the **Database (DB)** and the multi-tenancy setup. It was also setup the authentication and authorization and choose the roles of the users. In **APP** it was developed the connection to the **API**, as well as the authentication and authorization methods. Multi-language was also setup in this sprint.

In the end of the sprint all has been deployed and tested in **prod env**.

In the **May** sprint we started to do all that is involved in productions. So, was done, olive mills, olive varieties and olive groves management.

Like in the **April** sprint in end of the **May** sprint all has been deployed and tested in **prod env**.

The figure 5.3 shows all the October sprint backlog items.

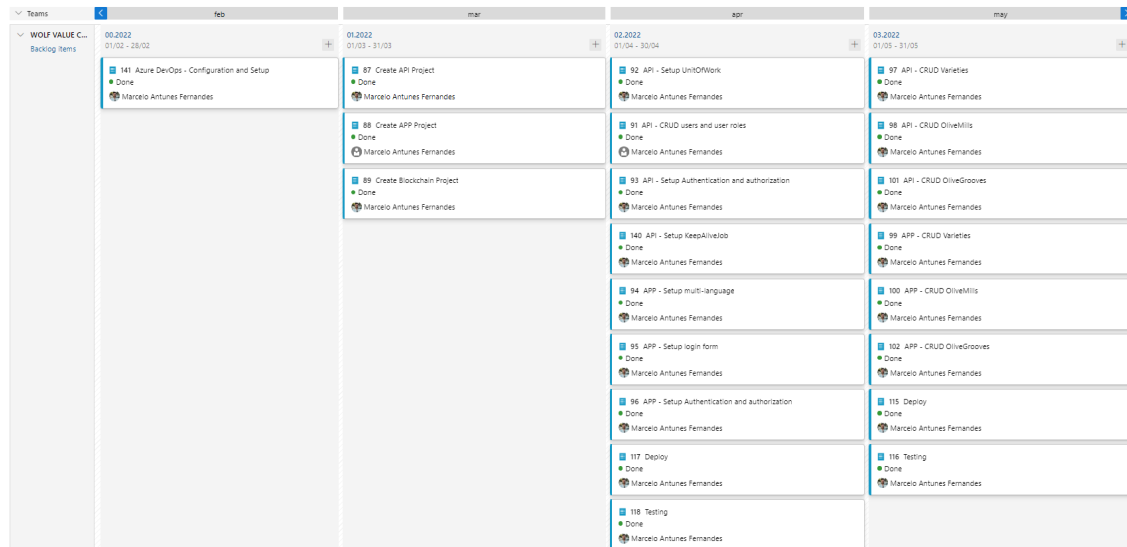


Figure 5.2: February to May Sprints.

5.3.1.2 June to September Sprints

In the **June, July and August** sprints the focus was the olive oil production as well as the activities and the activity measurements. It also created a generic component to take and read all the pictures as well as created a generic component to generate QR Codes. This component was created in a generic way to use them in all production stages.

Like in the previous sprints in end of each sprint all has been deployed and tested in **prod env**.

Here, we notice a little problem, with the size of the pictures, and we need to refactor some code to resize pictures before send it to the server, but without losing quality.

In the **September** sprint it was the Blockchain setup. It was the definition of the assets, and the write of the smart contracts. It was also the first blockchain deploy and the fake real test.

The figure 5.3 shows all the October sprint backlog items.

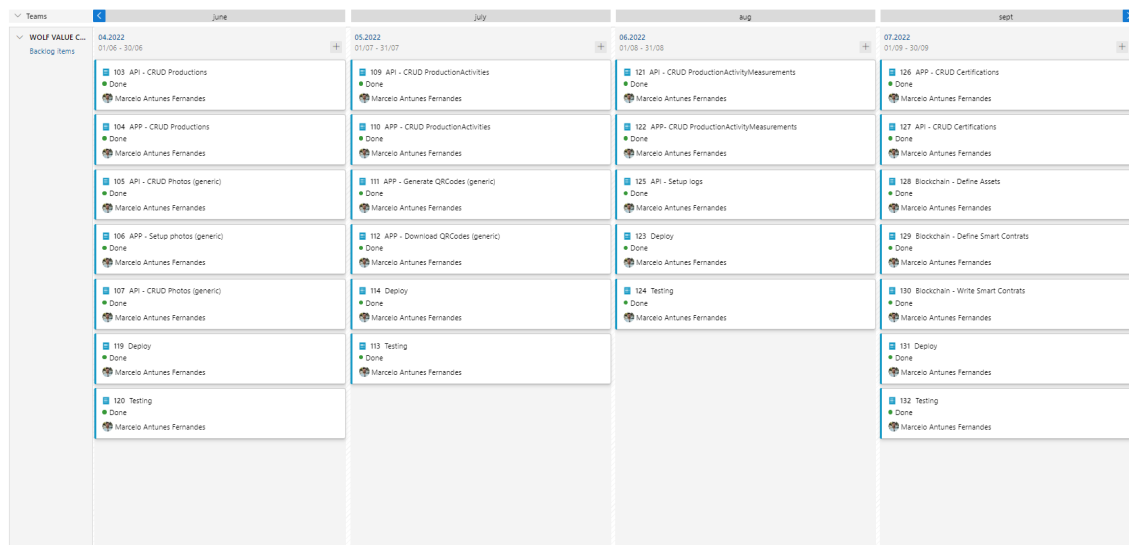


Figure 5.3: June to September Sprints.

5.3.1.3 October Sprint and launching

The **October** sprint, was the most expected sprint. In this sprint was created the connection to the Blockchain and all the log's setup. It also tested everything, but as mentioned in 5.2 we don't know, if in the day everything will be work. The figure 5.4 shows all the October sprint backlog items.

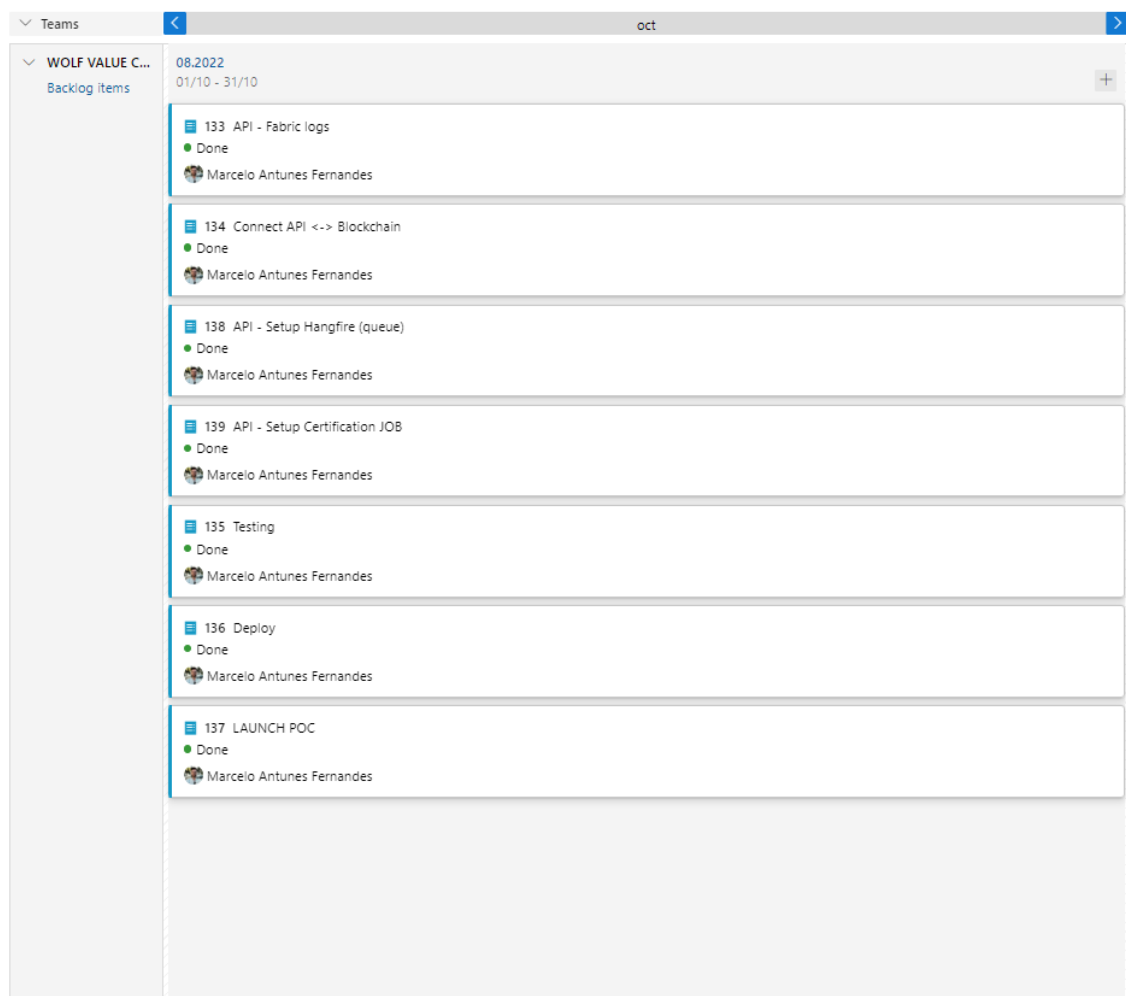


Figure 5.4: October Sprint and launching.

5.3.2 APP

For this project, has been used Ionic framework with Angular described in 2.3.4 and the following explains the structure as well as the major screens.

5.3.2.1 APP Structure

The figure 5.5 shows the APP structure and the table 5.1 explains the structure.

Table 5.1: APP Structure.

Folder	Description
components	This folder contains the generic components like photos, qrcode, lists used in the APP.
guards	This folder contains the guards that validates if user is <i>logged in</i> . It also validate the roles of user <i>logged in</i> .

Continued on next page.

Table 5.1 – continued from previous page.

Folder	Description
pages	This folder contains all the pages (public and secure) used in the APP.
services	This folder contains all HTTP requests to the API.
utils	This folder contains generic functions used in the APP.
assets	This folder contains the assets used in the APP.
environments	This folder contains the APP environments like <i>development</i> and <i>production</i> .
theme	This folder the APP stylesheets.

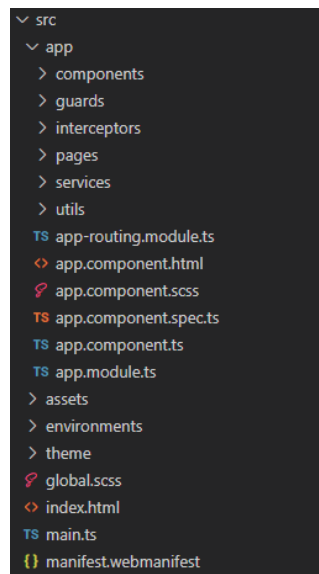


Figure 5.5: Application Structure.

5.3.2.2 APP Login

The figure 5.6 shows the login screen and the table 5.2 shows the elements on the form.

Table 5.2: Login form.

Field Id	Field Type	Additional Attributes (attr)	Action
email	email	required maxlength = 50	Not Applicable (N/A)
password	password	required maxlength = 20	N/A
signin	submit	N/A	POST: api/authentication/signin
password-recovery	N/A	N/A	GOTO: /password-recovery

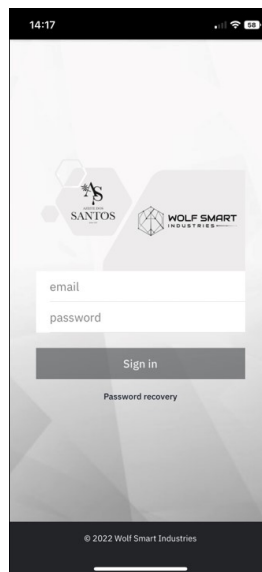


Figure 5.6: Application login form.

5.3.2.3 APP Menu

The figure 5.7 shows the menu with the following options:

- The **olive mills** menu option navigates to the list of olive mills.
GOTO: /olive-mills-list
- The **varieties** menu option navigates to the list of olive varieties.
GOTO: /varieties-list
- The **logs** menu option navigates to the **API** logs.
GOTO: /logs
- The **fabric logs** menu option navigates to blockchain logs.
GOTO: /logs-fabric

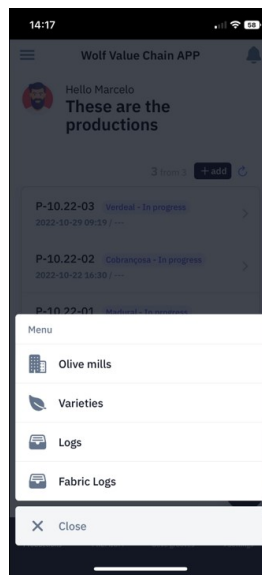


Figure 5.7: Application Menu.

5.3.2.4 Tabs

The figure 5.8 shows the tabs on the application.

5.3.2.5 Productions

The **Productions** tab shows the list of olive oil productions.

This list contains the following properties per each list item:

- Code.
- Start At Date.
- End At Date.
- Variety name.
- Status (In progress or Done).

Display: /tab-productions

Item list GOTO: /olive-grove-details/:id

Add new one GOTO: /olive-grove-details

GET: api/productions

5.3.2.6 PREMIUM

The **PREMIUM** tab shows the list of *premium* productions.

This list contains the following properties per each list item:

- Code.

- Start At Date.
- End At Date.
- Status (In progress or Done).

Display: /tab-premiums

Item list GOTO: /premium-details/:id

Add new one GOTO: /premium-details

GET: api/premiums

5.3.2.7 Olive groves

The **Olive groves** tab shows the list of the olive groves.

This list contains the following properties per each list item:

- Name;
- Address;
- Owner.

Display: /tab-olive-groves

Item list GOTO: /olive-grove-details/:id

Add new one GOTO: /olive-grove-details

GET: api/olivegroves

5.3.2.8 Settings

The **Settings** tab shows the application settings and has the following options:

- The **Accounting settings** option navigates to the account settings of the current user logged in.
GOTO /user-details/:id
- The **Users** option navigates to the list of the users*.
GOTO /users-list
- The **Language** option open the popup to choose language with the following options:
 - English
 - Portuguese
- The **Change password** option navigates to user user change password.
GOTO /change-password
- The **Notifications** option toggle to enable or disable push notifications for current user logged in**.

* Only admin access.

** Future work.

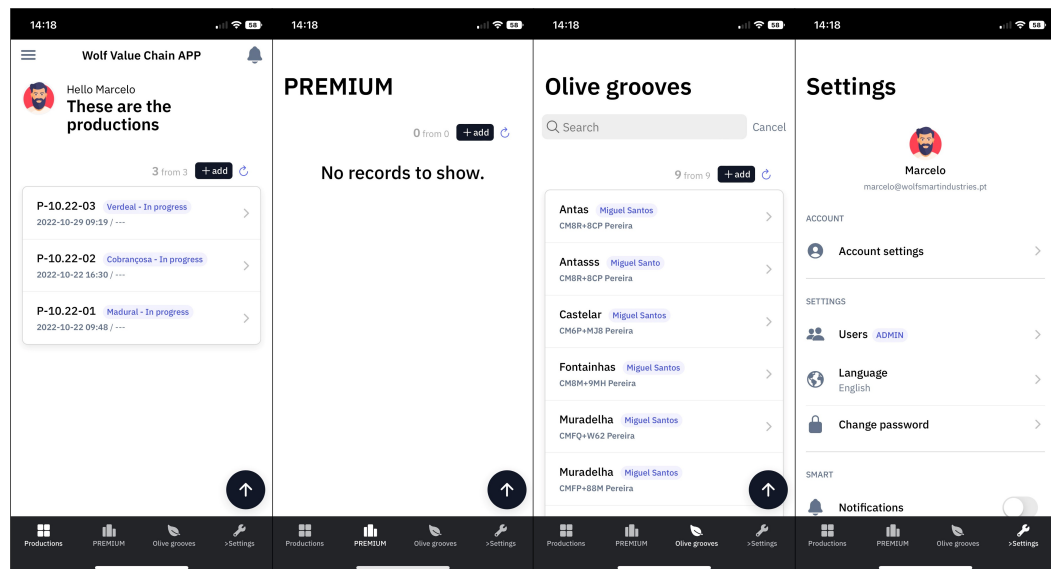


Figure 5.8: Application Tabs.

5.3.2.9 Olive mills

The figure 5.9 shows the olive mills list and olive mill details.

This list contains the following properties per each list item:

- Name.
- Address.

Display: /olive-mills-list

Item list GOTO: /olive-mill-details/:id

Add new one GOTO: /olive-mill-details

GET: api/olivemills

The table 5.3 contains the elements on the olive mill form:

Table 5.3: Olive mill form.

Field Id	Field Type	Additional attr	Action
code	text	required maxlength = 10	N/A
description	text	required maxlength = 10	N/A

Continued on next page.

Table 5.3 – continued from previous page.

Field Id	Field Type	Additional attr	Action
notes	text-area	N/A	N/A
save	submit	N/A	POST: api/olivemills/create or update
delete	button	N/A	POST: api/olivemills/delete

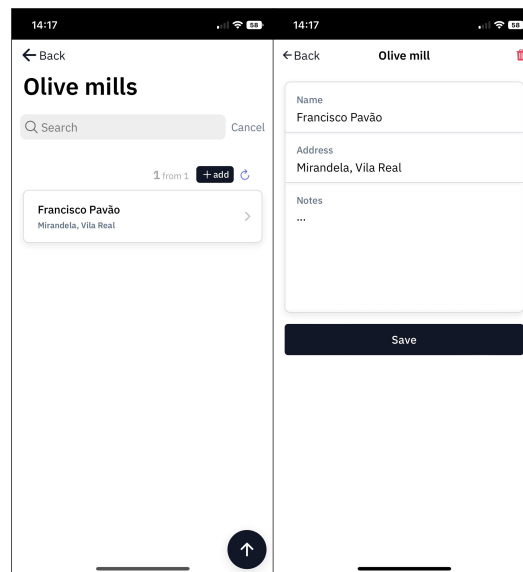


Figure 5.9: Application olive mills list and the olive mill details.

5.3.2.10 Olive varieties

The figure 5.9 shows the olive varieties list and the olive variety details.

This list contains the following properties per each list item:

- Name.
- Description.

Display: /olive-varieties-list

Item list GOTO: /olive-variety-details/:id

Add new one GOTO: /olive-variety-details

GET: api/olivevarieties

The table 5.4 contains the elements on the olive variety form:

Table 5.4: Olive variety form.

Field Id	Field Type	Additional attr	Action
code	text	required maxlength = 10	N/A
description	textarea	required	N/A
save	submit	N/A	POST: api/olivevarieties/create or update
delete	button	N/A	POST: api/olivevarieties/delete

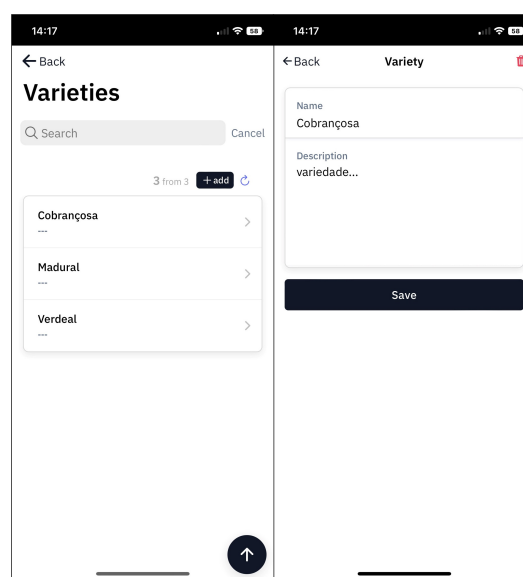


Figure 5.10: Application Olive varieties list and the olive variety details.

5.3.2.11 Olive oil productions

The figure 5.11 shows the production and the list of the activities in production.

The table 5.5 contains the elements on the production form and the production activities list contains the following properties per each list item:

- Code.
- Start at.
- End at.
- State (In progress or done).

Table 5.5: Olive oil production form.

Field Id	Field Type	Additional attr	Action
code	text	required maxlength = 10	N/A
startat	datetime-local	required	N/A
endat	datetime-local	N/A	N/A
varietyid	select	required	GET: api/varieties
notes	textarea	N/A	N/A
save	submit	N/A	POST: api/productions/create or update
delete	button	N/A	POST: api/productions/delete
certify	button	N/A	POST: api/productions/certify

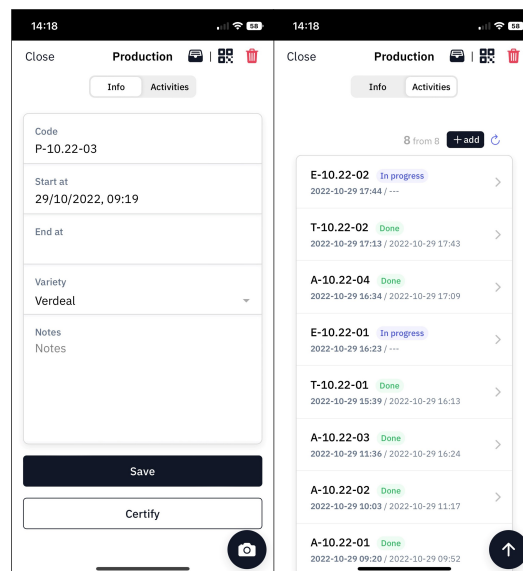


Figure 5.11: Olive oil production and the list of its activities.

5.3.2.12 Olive oil production activity

The figure 5.11 shows the production activity, the origins of the activity and the list of the measurements in production activity.

The table 5.6 contains the elements on the production activity form and the 5.7 contains the elements on the production activity measurement form. The production activity measurements list contains the following properties per each list item:

- Measurement type;
- Date;

- Measurement value.

Table 5.6: Olive oil production activity form.

Field Id	Field Type	Additional attr	Action
activityType	select	required	from Enumerator (enum) 0 - Harvest 1 - Transport 2 - Entrance 3 - Crushing 4 - Maxalation 5 - Centrifugation 6 - Vats 7 - Bottling
startat	datetime-local	required	N/A
endat	datetime-local	N/A	N/A
outputType	select	required	from enum 0 - Olive box 1 - Olive weighted 2 - Olive paste 3 - Olive oil 4 - Botles
quantity	number	required	N/A
unitType	select	required	from enum 0 - Unity 1 - Kg 2 - Liters (l)
oliveGroveId	select	required	GET: api/olivegroves
notes	textarea	N/A	N/A
save	submit	N/A	POST: api/productions/activities/create or update
delete	button	N/A	POST: api/productions/activities/delete

Table 5.7: Olive oil production activity measurement form.

Field Id	Field Type	Additional attr	Action
measurementType	select	required	from enum 0 - Fuel (l) 1 - Distance (km) 2 - Water consumption (l) 3 - Electricity (kW) 4 - Temperature (°C) 5 - Humidity (%) 6 - Acidity (%) 7 - Taste (1 - 5)
value	text	required	N/A
date	datetime-local	required	N/A
notes	textarea	N/A	N/A
save	submit	N/A	POST: api/productions/activities/create or update
delete	button	N/A	POST: api/productions/activities/delete

The figure displays four sequential screenshots of a mobile application interface for managing olive oil production activities and measurements.

- Screen 1 (Activity Form):** Shows the 'Activity' form with tabs for 'Info', 'Inputs', and 'Measurements'. Fields include Code (E-10.22-02), Activity type (Entrance), Start at (29/10/2022, 17:44), End at, Output type (Azeitonas pesadas), Quantity (0), Unit type (Kg), Olive groove (Select), and Olive mill (Francisco Pavão).
- Screen 2 (Inputs):** Shows the 'Inputs' tab with a list of activities. The first activity is T-10.22-02 with Quantity remaining 2 and Quantity 2. A red 'Delete' button is visible at the bottom.
- Screen 3 (Measurements):** Shows the 'Measurements' tab with a list of measurements. The first measurement is Temperature (°C) with value 16 and date 2022-10-29 17:53 / 16. A '+add' button is visible at the top.
- Screen 4 (Measurement Form):** Shows the 'Measurement' form with fields for Measurement type (Temperature (°C)), Value (16), Date (29/10/2022, 17:53), and Notes (Temperatura ambiente). A 'Save' button is at the bottom.

Figure 5.12: Olive oil production activity with measurements.

5.3.3 API

The structure of the **API** is depicted in Figure 5.13 and the following list provides more information about each component of the structure:

Table 5.8: API Structure.

Folder	Description
Controllers	This folder represents all entry points in the API .
Data	This folder represents database context, entities, migrations, unit of work and generic repository.
Extensions	This folder represents the new methods added in the some classes.
files	This folder contains saved files like images.
IO	This folder represents all inputs and outputs in the API .
Jobs	This folder represents all the jobs in the API .
Middlewares	This folder represents all middlewares like user validation in the API .
Services	This folder represents all the <i>business</i> logic as well as the database access.

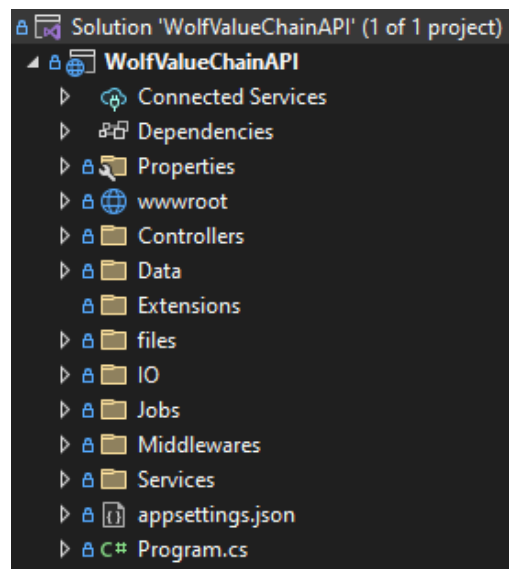


Figure 5.13: API Structure.

5.3.4 Blockchain

The figure 5.14 shows the **APP** structure and the table 5.9 explains the structure.

Table 5.9: Blockchain Structure.

File / Folder	Description
assets	This folder contains all the assets presents in Blockchain.
chaincodes	This folders contains all the chaincodes complied.
fablo-target	This folder represents the Blockchain network.

Continued on next page.

Table 5.9 – continued from previous page.

File / Folder	Description
utils	This folder contains all the generic functions used in Blockchain.
olive-production-contract.go	This is the smart contract.
olive-production-contract_test.go	This is all the smart contract tests.

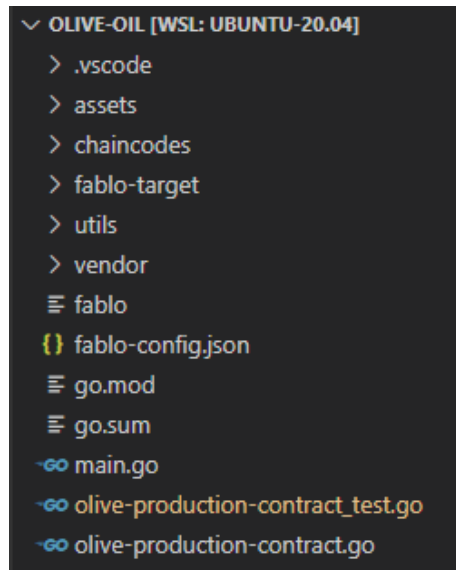


Figure 5.14: Blockchain Structure.

5.3.5 From Application to Blockchain

The diagram 5.15 represents, as an example, the flow of data from APP to the Blockchain during the creation of a new production activity. At APP level, and based on the form described in 5.6, a user inserts data and by a HTTP request with a valid token, goes to the API.

At the API level, the information is stored in the database using UnitOfWork and is also stored in the Blockchain with a background task created using Hangfire.

Before goes to the Blockchain, the *Start At* time is registered and, after response, is also registered the *End At* time.

The communication between API and Blockchain is by HTTP request with a valid token.

After stored in database with UnitOfWork the user is notified and the APP refresh the form.

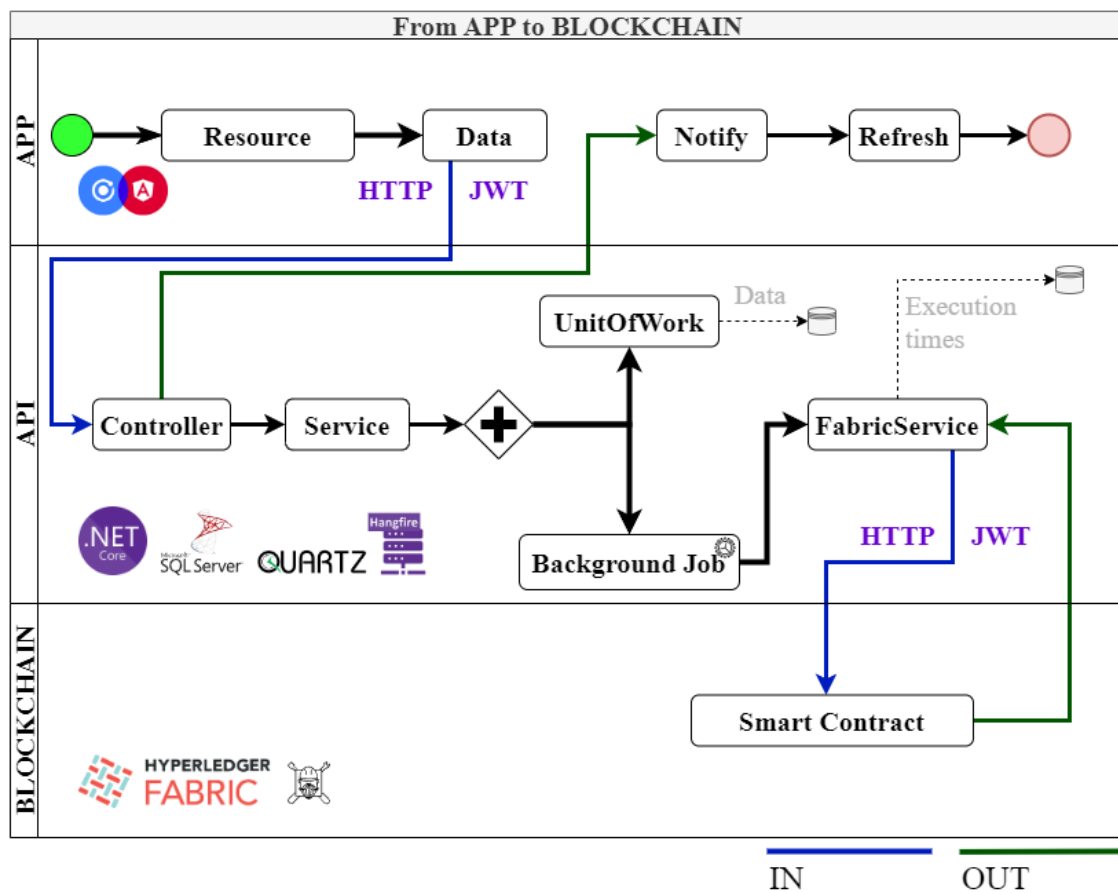


Figure 5.15: From Application to Blockchain diagram.

5.3.6 Store information and queue to Blockchain

The figure 5.16 shows the using of the UnitOfWork to store information in database and the background task to request the Blockchain.


```
2 references
public async Task<ActivityDto> CreateActivityAsync(ActivityModel model)
{
    var entity = await _unitOfWork.
        ProductionActivitiesRepository.
        CreateAsync(new ProductionActivity(model));

    var activity = await GetActivityByIdAsync(entity.Id);

    //save to chain
    BackgroundJob.Enqueue(() => _fabricService.UpdateProductionAssetAsync(activity.ProductionId));

    return activity;
}
```

Figure 5.16: Store information and queue to Blockchain.

The figure 5.17 shows the communication between the API and the Blockchain and the following points describes the what the function does.

1. Calls the GetTokenAsync method to obtain a token for authentication.
2. If a token is obtained and it is not null or empty, the function creates a new instance of the HttpClient class and sets the base address for the API using a value from a configuration file.
3. Sets the HTTP headers for the client to accept JSON and to include the authentication token in the request.
4. Creates a JSON payload using the method and args arguments passed to the function and serializes it using the JsonConvert.SerializeObject method.
5. Sends an HTTP POST request to the Fabric API with the specified endpoint and the JSON payload.
6. Reads the response message from the request and logs the result using the _logsService class.

Prior to the execution of the function, a Fabric log is created with the label 'START', and once the function has completed, another Fabric log is generated with the label 'END' with all execution times recorded.

```

2 references
private async Task<T> RequestFabric<T>(string type, string[] args, string method)
{
    TokenDto? token = await GetTokenAsync();
    if (token is not null && string.IsNullOrEmpty(token.Token) is false)
    {
        using (var client = new HttpClient())
        {
            client.BaseAddress = new Uri(_configuration["FABRIC:APIURL"]);
            client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
            client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", token.Token);

            var data = new { method, args };
            var serializerSettings = new JsonSerializerSettings();
            serializerSettings.ContractResolver = new CamelCasePropertyNamesContractResolver();
            var json = JsonConvert.SerializeObject(data, serializerSettings);
            var httpContent = new StringContent(json, Encoding.UTF8, "application/json");

            var request = await client.PostAsync($"{type}/{_configuration["FABRIC:CHANNEL"]}/{_configuration["FABRIC:CHAINCODE"]}", httpContent);

            var message = await request.Content.ReadAsStringAsync();
            await _logsService.CreateFabricLogAsync(type, method, request.IsSuccessStatusCode, json, message);

            return await request.Content.ReadFromJsonAsync<T>();
        }
    }
    return default(T);
}

```

Figure 5.17: Request Blockchain.

5.3.7 Olive Production Smart Contract

The figure 5.18 shows the Olive Production Smart Contract. This smart contract has the following functions:

- **ProductionAssetExists** - returns true when asset with given ID exists in world state.
- **ReadProductionAsset** - returns true when asset with given ID exists in world state.
- **CreateProductionAsset** - creates a new instance of Production in world state.
- **UpdateProductionAsset** - retrieves an instance of Production from the world state and updates its value.
- **DeleteProductionAsset** - deletes an instance of Production from the world state.
- **GetProductionAssetHistory** - returns the chain of custody for an Production since issuance.

```

//this contract represents contract for managing CRUD for OliveProductionContract.
> type OliveProductionContract struct {...
}

// ProductionAssetExists returns true when asset with given ID exists in world state.
> func (c *OliveProductionContract) ProductionAssetExists(ctx contractapi.TransactionContextInterface, id string) (bool, error) {...
}

// ReadProductionAsset retrieves an instance of Production from the world state.
> func (c *OliveProductionContract) ReadProductionAsset(ctx contractapi.TransactionContextInterface, id string) (*asset.Production, error) {...
}

// CreateProductionAsset creates a new instance of Production.
> func (c *OliveProductionContract) CreateProductionAsset(ctx contractapi.TransactionContextInterface, id string, production asset.Production) error {...
}

// UpdateProductionAsset retrieves an instance of Production from the world state and updates its value.
> func (c *OliveProductionContract) UpdateProductionAsset(ctx contractapi.TransactionContextInterface, id string, production asset.Production) error {...
}

// DeleteProductionAsset deletes an instance of Production from the world state.
> func (c *OliveProductionContract) DeleteProductionAsset(ctx contractapi.TransactionContextInterface, id string) error {...
}

// GetProductionAssetHistory returns the chain of custody for an Production since issuance.
> func (c *OliveProductionContract) GetProductionAssetHistory(ctx contractapi.TransactionContextInterface, id string) ([]utils.HistoryQueryResult, error) {...
}

```

Figure 5.18: Olive Production Smart Contract.

The figure 5.19 shows the smart contract assets.

Assets are represented in Hyperledger Fabric as a collection of key-value pairs, with state changes recorded as transactions on a Channel ledger. Assets can be represented in binary and/or JSON form [Hyperledger, 2022].

- **Production** - this asset represents production by variety.
- **ProductionActivity** - this asset represents the productions activities.
- **ProductionActivityMeasurement** - creates a new instance of Production in world state.
- **ProductionActivityInput** - this asset represents the productions activities inputs.

```
//this asset represents production by variety
type Production struct {
  base.BaseAsset
  Code      string      `json:"code"`
  StartAt   string      `json:"startAt"`
  EndAt     string      `json:"endAt,omitempty" metadata:"endAt,optional"`
  VarietyId string      `json:"varietyId"`
  Variety   string      `json:"variety"`
  Notes     string      `json:"notes,omitempty" metadata:"notes,optional"`
  Activities []*ProductionActivity `json:"activities,omitempty" metadata:"activities,optional"`
}

//this asset represents the productions activities
type ProductionActivity struct {
  base.BaseAsset
  Code            string      `json:"code"`
  ProductionId    string      `json:"productionId"`
  StartAt         string      `json:"startAt"`
  EndAt           string      `json:"endAt,omitempty" metadata:"endAt,optional"`
  ActivityType    string      `json:"activityType"`
  ActivityOutputType string    `json:"activityOutputType"`
  QuantityOutput  string      `json:"quantityOutput"`
  UnitType        string      `json:"unitType"`
  Notes           string      `json:"notes,omitempty" metadata:"notes,optional"`
  OliveGrooveId   string      `json:"oliveGrooveId,omitempty" metadata:"oliveGrooveId,optional"`
  OliveGrooveName string      `json:"oliveGrooveName,omitempty" metadata:"oliveGrooveName,optional"`
  OliveGrooveAddress string    `json:"oliveGrooveAddress,omitempty" metadata:"oliveGrooveAddress,optional"`
  OliveGrooveOwner string    `json:"oliveGrooveOwner,omitempty" metadata:"oliveGrooveOwner,optional"`
  OliveMillId     string      `json:"oliveMillId,omitempty" metadata:"oliveMillId,optional"`
  OliveMillName   string      `json:"oliveMillName,omitempty" metadata:"oliveMillName,optional"`
  Measurements    []*ProductionActivityMeasurement `json:"measurements,omitempty" metadata:"measurements,optional"`
  Inputs          []*ProductionActivityInput      `json:"inputs,omitempty" metadata:"inputs,optional"`
}

//this asset represents the productions activities measurements
type ProductionActivityMeasurement struct {
  base.BaseAsset
  ProductionActivityId string `json:"productionActivityId"`
  MeasurementType      string `json:"measurementType"`
  Date                 string `json:"date"`
  Value                string `json:"value"`
  Notes                string `json:"notes,omitempty" metadata:"notes,optional"`
}

//this asset represents the productions activities inputs
type ProductionActivityInput struct {
  base.BaseAsset
  Quantity          string `json:"quantity"`
  QuantityRemaning  string `json:"quantityRemaning"`
  ProductionActivityId string `json:"productionActivityId"`
  ProductionActivityInputId string `json:"productionActivityInputId"`
  ProductionActivityInputCode string `json:"productionActivityInputCode"`
}
```

Figure 5.19: Smart Contract Assets.

5.3.8 Traceability

Considering the problem and data model presented in 4.8, it is observed that each production activity contains its associated production and the input activities that resulted in this production activity. The model enables the establishment of a clear and precise relationship between the production activities and their respective inputs, providing an accurate representation of the production process.

Moreover, the model establishes a direct relationship between the production activities and the measurements, with each measurement being linked to a specific production activity. This linkage enables the easy tracking of the production activity's origin, as well as the accurate determination of the associated measurements.

By having the origin of each activity readily available, navigating between activities becomes seamless, allowing for the identification of the production process's starting and ending points. Furthermore, the model enables the tracing of the origin of each activity through measurements, thereby ensuring complete traceability.

The figure 5.20 provides a simplified representation of the relationship between production, production activities, input activities, and measurements. This figure serves as an essential tool for managing and tracking the production process, ensuring quality control and accountability.

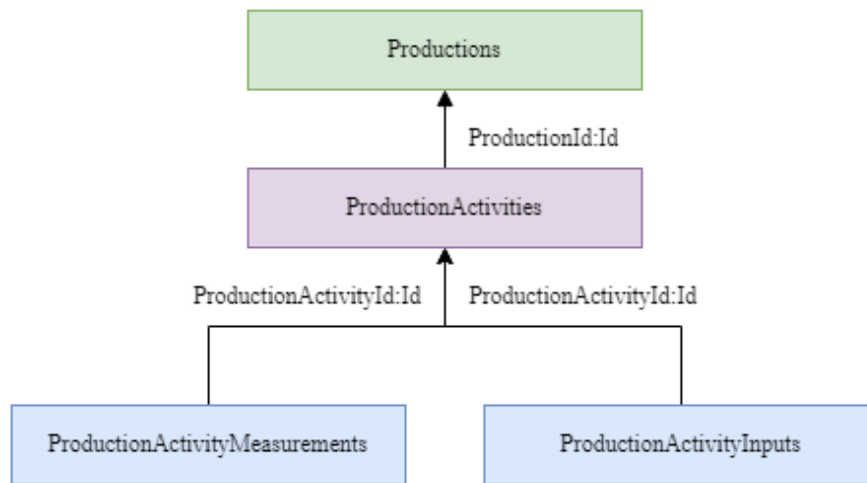


Figure 5.20: Relations between production, activities and measurements.

The figure 5.21 depicts an example of the traceability of a production process that consists of a series of interconnected activities, each associated with a measurement. The figure illustrates how the data model described above can be used to manage and track the production process, ensuring quality control and accountability.

As shown in the figure, each production activity is linked to its associated production and input activities, providing a clear representation of the production process. Furthermore, each measurement is directly related to a specific production activity, allowing for the accurate tracking of the activity's origin and associated measurements.

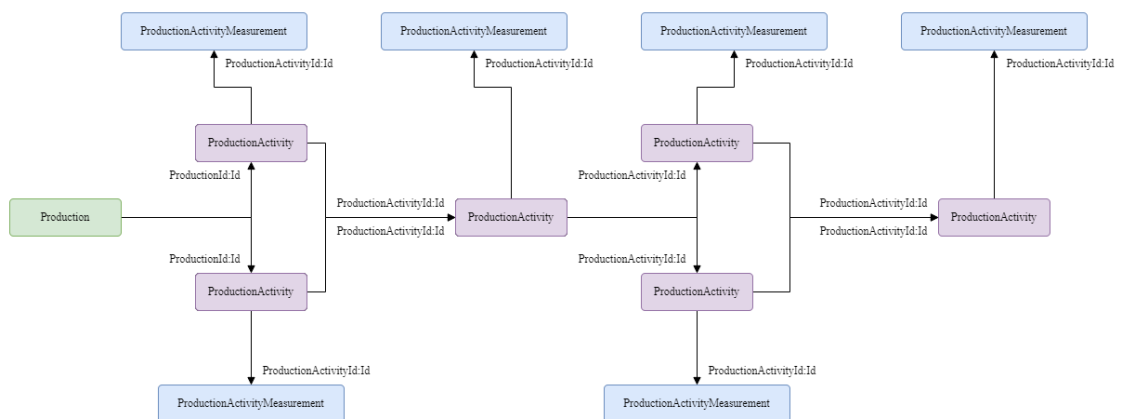


Figure 5.21: Example of traceability.

As described in section 5.3.5, traceability can be achieved through both relational databases and blockchain technology. When using a relational database, it is implemented as a typical application. However, when using blockchain, it becomes a **DApp**.

Chapter 6

Evaluation and results

6.1 Case study

As described in section 1 this project aims to provide a solution to the problem of tracking the production of Olive Oil from Quinta Castelar in Mirandela, Vila Real, Portugal, specifically the Olive Oil Premium dos Santos. The olive grove from which this oil is produced has a rich history, with some of the trees having been planted as early as 1922.

The production process for the Olive Oil Premium dos Santos is meticulously recorded in the Blockchain through the use of IoT devices such as sensors and QR Codes. This ensures that the entire process is streamlined and that important information such as execution times and environmental factors such as temperature and humidity are captured. These factors are crucial to maintaining the quality of the Olive Oil.

6.2 Data collect

For this Proof of Concept (POC), three productions were conducted on three different days over two weekends: October 22nd and 23rd, and October 28th. The productions were done using three different varieties of olives: *Verdeal*, *Madural*, and *Cobrançosa*.

With a focus on the *Madural* variety, which was produced on the first day, the data collection was divided into three main groups: **Harvest**, **Transport** and **Olive mill**. The data recorded in the blockchain included the start and end times of each step, as well as the effective time, which were all recorded in the **Blockchain**. Additionally, it was recorded the location of the harvest, the weather conditions and the temperature during the harvest, and the number of olives picked. The transport data recorded included the route, the transport company, the duration of the trip and the temperature during transport, and in the olive mill data recorded included the type of equipment used, the duration of the mill process and the temperature during the mill process. All this data was used to ensure the traceability and quality of the Olive Oil.

6.2.1 Harvest

We define the beginning of the harvest as the moment when the olive shaker starts shaking the first olive tree and the end of the harvest as the moment when the shaker reaches its maximum capacity and unloads the olives into pallets, which are the appropriate containers for transporting the olives.

The first harvest of the *Madural* variety was conducted on October 22nd, with a start time of 9:49 a.m. in the olive grove named *Muradilha 1* as recorded in the data table. This harvest had the code A-10.22-01. The weather at the time was rainy, which made the harvest difficult. Due to the rainy weather, the frequency of shaking had to be lowered to prevent damage to the olive trees, resulting in a lower yield. More than 20 olive trees were harvested in this session, generating a yield of 0.5 pallets of olives.

The second harvest with the code A-10.22-02 began at 10:16 a.m., immediately following the first harvest, in the same olive grove *Muradilha 1*. Despite the weather conditions, the yield was 1.25 pallets of olives.

The third harvest, with the code A-10.22-03, began at 10:56 a.m. in another olive grove, *Muradilha 2*, with weather conditions that were still rainy but less intense. The yield generated from this harvest was 0.75 pallets of olives.

The fourth and final harvest of the day, with the code A-10.22-04, began at 1:58 p.m. in the olive grove *Castelar* and was completed at 2:40 p.m. The yield was 1.5 pallets of olives, and the pallets were loaded onto a trailer for transport to the olive mill.

Throughout the process, images were taken (appendix A) to document the harvest, the location of the olive groves, and other relevant information.

For this four harvests the weather conditions was 15° C and the tractor consumed 40l of agricultural diesel.

6.2.1.1 Execution times

The table 6.1 shows the execution times in the harvest activities. It was record the start at time when **Blockchain** has requested and the end at when request ends.

Table 6.1: Harvest execution times

Start At	End At	ms
2022-10-22 09:49:12.7014282	2022-10-22 09:49:15.9464598	3000
2022-10-22 09:50:19.9817462	2022-10-22 09:50:22.9052965	2000
2022-10-22 09:52:38.2059929	2022-10-22 09:52:41.0761783	2871
2022-10-22 09:53:13.2467891	2022-10-22 09:53:16.0772132	2831
2022-10-22 09:58:50.1958363	2022-10-22 09:58:53.2361106	3041
2022-10-22 10:12:42.4085855	2022-10-22 10:12:45.2785100	2870
2022-10-22 10:13:39.0839939	2022-10-22 10:13:41.9344009	2851
2022-10-22 10:16:26.2734356	2022-10-22 10:16:29.3219856	3048
Continued on next page		

Table 6.1 – continued from previous page

Start At	End At	ms
2022-10-22 10:17:22.2196141	2022-10-22 10:17:25.2622751	3043
2022-10-22 10:20:09.5074348	2022-10-22 10:20:12.3523784	2845
2022-10-22 10:37:59.4187981	2022-10-22 10:38:02.4566008	3038
2022-10-22 10:51:49.5198521	2022-10-22 10:51:52.5547644	3035
2022-10-22 10:56:55.4818440	2022-10-22 10:56:58.5960896	3115
2022-10-22 10:59:30.1300744	2022-10-22 10:59:33.0073976	2877
2022-10-22 11:00:20.2693418	2022-10-22 11:00:23.1280760	2859
2022-10-22 11:02:16.4997186	2022-10-22 11:02:19.5822829	3083
2022-10-22 11:05:01.9228662	2022-10-22 11:05:04.7699400	2847
2022-10-22 11:09:03.1831703	2022-10-22 11:09:06.2780272	3095
2022-10-22 11:31:42.1325798	2022-10-22 11:31:45.1481303	3016
2022-10-22 13:59:10.9226030	2022-10-22 13:59:13.9586331	3036
2022-10-22 14:00:35.9038949	2022-10-22 14:00:38.7693986	2866
2022-10-22 14:23:20.8881866	2022-10-22 14:23:23.9261426	3038

6.2.2 Transport

We considered a transport, whenever the olive pallets were moved from the olive groves to the olive mill (place where the production of oil is made). In the production of this variety, it was only necessary to use a transport with the code T-10.22-01, because as we saw above in the harvest detail, the harvest generated 4 pallets of olives, these pallets were loaded onto a trailer, which in turn was transported by a tractor. This transport started at 2:50 pm, and took 35Minute (m) to reach its destination, the olive mill, covering a total distance of 18km. The transport ended when it arrived at the mill. All this data was recorded during the process.

Throughout the process, like the harvest process, images were taken (appendix A) to document the transport, the location of the olive grove and the olive mill, and other relevant information.

For this transport process the weather conditions was 15° C and the tractor consumed 5l of agricultural diesel.

6.2.2.1 Execution times

The table 6.2 shows the execution times in the transport activity. It was record the start at time when **Blockchain** has requested and the end at when request ends.

Table 6.2: Transport execution times

Start At	End At	ms
2022-10-22 14:40:43.9559234	2022-10-22 14:40:47.0013285	3046
2022-10-22 14:47:59.7136786	2022-10-22 14:48:02.7541147	3041
2022-10-22 14:49:39.9509605	2022-10-22 14:49:42.7996081	2849
2022-10-22 14:49:54.4747723	2022-10-22 14:49:57.3264162	2852
2022-10-22 14:50:11.0629980	2022-10-22 14:50:13.8958158	2833
2022-10-22 14:50:30.2751440	2022-10-22 14:50:33.1228364	2847
2022-10-22 14:50:49.7983021	2022-10-22 14:50:52.6586747	2860
2022-10-22 14:51:29.4982451	2022-10-22 14:51:32.4072792	2909
2022-10-22 14:55:35.8106125	2022-10-22 14:55:38.8813279	3071
2022-10-22 15:35:28.6783468	2022-10-22 15:35:31.8978430	3219

6.2.3 Olive mill

In the olive mill we consider several processes, of which:

- **Delivery** - a delivery is considered when the olive mill collects pallets from transportation and places the olives through a cleaning and washing process, resulting in a weight of olives.
- **Grinding** - a grinding process is considered when heavy and clean olives from the delivery process are fed through conveyor belts, slowly grinding the olive to form a paste.
- **Thermo Mixer** - a thermo mixer process is considered when the paste from the grinding process enters large capacity cylinders for kneading, resulting in a more homogeneous paste compared to the grinding paste.
- **Centrifugation** - a centrifugation process is considered when the paste from the thermo mixer enters a process of separating liquid olive oil from all other elements, resulting in only the oil.
- **Vats** - a vat process is considered when the olive oil leaves the centrifugation process and is stored in olive oil vats.

These processes are described, taking into account the production of the *Madural* variety.

The delivery process for this production began at 3:42 PM with the code E-10.22-01. As soon as the pallets were unloaded at the mill, the cleaning process began, which involved separating the olives from leaves and branches, followed by washing with water. The olives were then weighed, resulting in a weight of 1162.7 kg of clean olives. The delivery process was completed at 4:34 PM. The entire process is automated.

The grinding process began at 4:45 PM with the code M-10.22-01, just after the delivery. The olive was provided in a linear way for grinding, and as soon as the process generated olive paste,

the thermo mixer process began with the code B-10.22-01, resulting in a more homogeneous paste. These two processes ran in parallel for over an hour, resulting in a weight of 1162.7kg.

The centrifugation process began at 3:57 PM with the code C-10.22-01, following the thermo mixer process. The process separated all the elements that were not part of the olive oil's constitution, resulting in 132 kg of liquid olive oil. The process ended at 7:18 PM, as the olive paste went through two different types of centrifugation, the first to separate solid elements of the paste, such as remains of stones, and the second to separate water from the olive oil itself. The result was pure extra virgin olive oil. The profitability of the harvest was calculated based on the weight of the oil.

After the process, Engineer Francisco Pavão and other members of the mill did a tasting test to evaluate the olive oil and give it notes before storing it in vats. These notes were also registered on the blockchain.

The vats process, with the code Cubas-10.22-01, started at 7:40 PM. The appropriate vat was chosen based on the amount of olive oil generated, and the filling process began, resulting in a vat with 132 kg of olive oil, which took 4 minutes.

Throughout the process, images were taken (appendix A) to document the olive mill and other relevant information.

For olive mill process the weather conditions was 20°C (delivery) and 21°C (grinding, thermo mixer, centrifugation and vats), and the olive mill consumed 1000l of water.

Important note: The water used was the same water used throughout the day, which means that in all delivery processes the water consumed is the same on the same day.

6.2.3.1 Execution times

The table 6.3 shows the execution times in the transport activity. It was record the start at time when **Blockchain** has requested and the end at when request ends.

Table 6.3: Olive mill execution times

Start At	End At	ms
2022-10-22 15:43:32.4327696	2022-10-22 15:43:35.4781895	3046
2022-10-22 15:45:18.7258120	2022-10-22 15:45:21.5649900	2839
2022-10-22 15:46:36.9198164	2022-10-22 15:46:39.7716340	2852
2022-10-22 16:08:43.1038301	2022-10-22 16:08:46.1385273	3035
2022-10-22 16:12:48.9548758	2022-10-22 16:12:51.8426663	2888
2022-10-22 16:34:16.8746479	2022-10-22 16:34:19.9129767	3038
2022-10-22 16:40:51.1268200	2022-10-22 16:40:54.1984695	3072
2022-10-22 16:41:26.0719457	2022-10-22 16:41:28.9183996	2847
2022-10-22 16:42:10.3272697	2022-10-22 16:42:13.2313818	2904
2022-10-22 16:48:47.2812131	2022-10-22 16:48:50.4404732	3159
2022-10-22 16:49:30.4390512	2022-10-22 16:49:33.4108404	2971
2022-10-22 16:50:13.4392143	2022-10-22 16:50:16.4634387	3024
Continued on next page		

Table 6.3 – continued from previous page

Start At	End At	ms
2022-10-22 17:08:18.5614803	2022-10-22 17:08:21.6615434	3100
2022-10-22 17:09:48.2449307	2022-10-22 17:09:51.1002952	2856
2022-10-22 17:10:49.9586938	2022-10-22 17:10:52.8162307	2858
2022-10-22 17:48:04.2928359	2022-10-22 17:48:07.4338022	3141
2022-10-22 17:48:47.4049130	2022-10-22 17:48:50.2772108	2873
2022-10-22 17:58:24.9992185	2022-10-22 17:58:28.2092558	3210
2022-10-22 18:02:48.0312657	2022-10-22 18:02:51.0773177	3046
2022-10-22 18:09:02.5041201	2022-10-22 18:09:05.4024153	2898
2022-10-22 18:09:37.7773995	2022-10-22 18:09:40.8038844	3026
2022-10-22 18:10:09.2456652	2022-10-22 18:10:12.0808916	2835
2022-10-22 18:16:52.6227025	2022-10-22 18:16:55.8010077	3179
2022-10-22 18:17:20.4334113	2022-10-22 18:17:23.4334263	3000
2022-10-22 18:44:06.4096046	2022-10-22 18:44:09.5491790	3140
2022-10-22 19:01:03.2465433	2022-10-22 19:01:06.5332139	3287
2022-10-22 19:18:49.8948394	2022-10-22 19:18:53.0871869	3193
2022-10-22 19:19:56.2159711	2022-10-22 19:19:59.2990124	3084
2022-10-22 19:21:38.5636370	2022-10-22 19:21:41.5932004	3030
2022-10-22 19:41:36.2858334	2022-10-22 19:41:39.4194028	3134
2022-10-22 19:44:59.1059948	2022-10-22 19:45:02.0998538	2994
2022-10-22 19:45:11.5492311	2022-10-22 19:45:14.5954798	3046
2022-10-22 19:48:06.9110308	2022-10-22 19:48:10.0835014	3172
2022-10-22 21:52:07.4727398	2022-10-22 21:52:10.6442562	3172
2022-10-22 22:03:23.6082299	2022-10-22 22:03:26.8136690	3205

6.3 Analysis

6.3.1 Results

During the production of the *Madural* variety, a total of 4 harvests were conducted, resulting in 4 units of olive pallets. These 4 pallets were transported to the olive oil mill, which upon arrival, weighed a total of 1162.7kg.

The olives were then processed, resulting in 1162.7kg of olive paste, without any wastage or losses during the process. Finally, the olive paste underwent centrifugation, resulting in the production of 132kg of high-quality olive oil.

6.3.2 Traceability

Figure 6.1 provides a graphical representation of the production process for *Madural* variety olive oil. This diagram enables a comprehensive understanding of the production process, from the initial activities of production to the vats activity.

In addition to the diagram, the listings 6.1 and 6.2 showcase the first and second blocks in JSON format on the Blockchain. These listings provide a detailed overview of the production process, ensuring full transparency and traceability.

The final block, which contains the details of the *Madural* production, can be found in the appendix B.

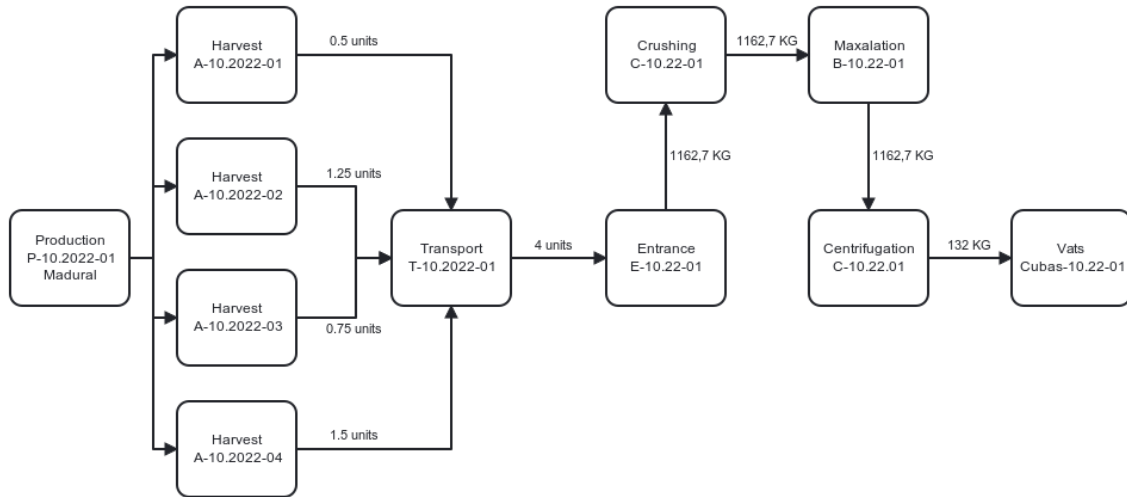


Figure 6.1: Madural olive oil production chain registered activities.

Listing 6.1: First blockchain block.

```

1 {
2   "code": "P-10.22-01",
3   "startAt": "22/10/2022 09:48:00",
4   "endAt": "",
5   "varietyId": "071c0e9b-72dd-48f4-2264-08daaf76b9ff",
6   "variety": "Madural",
7   "notes": "Tempo chuvoso\r\nMas com abertura durante a manha.",
8   "activities": []
9   "id": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
10  "createdAt": "22/10/2022 09:49:12",
11  "createdBy": "leandro@wolfsmartindustries.pt",
12  "updatedAt": "",
13  "updatedBy": "",
14  "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
15  "rowVersion": "00000824195"
16 }

```

Listing 6.2: Second blockchain block.

```

1 {
2   "code": "P-10.22-01",
3   "startAt": "22/10/2022 09:48:00",
4   "endAt": "",
5   "varietyId": "071c0e9b-72dd-48f4-2264-08daaf76b9ff",
6   "variety": "Madural",
7   "notes": "Tempo chuvoso\r\nMas com abertura durante a manha.",
8   "activities": [
9     {
10      "code": "A-10.22-01",
11      "productionId": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
12      "startAt": "22/10/2022 09:49:00",
13      "endAt": "22/10/2022 10:16:00",
14      "activityType": "Harvest",
15      "activityOutputType": "BoxedOlives",
16      "quantityOutput": "0,5",
17      "unitType": "Unity",

```

```

18     "notes": "Tempo de chuva, o que dificultou a apanha.\r\nDevido a isso a frequencia do varejo tem de ser mais baixa
19         para nao danificar os Ramos da Oliveira e como resultado nao sai tanta azeitona\r\nMedia de 20 Oliveiras",
20     "oliveGroveId": "3700968c-7457-448b-c94d-08dab409e770",
21     "olivegroveName": "Muradilha 1",
22     "olivegroveAddress": "CMFP+88M Pereira",
23     "olivegroveOwner": "Miguel Santos",
24     "oliveMillId": "",
25     "oliveMillAddress": "",
26     "measurements": [],
27     "inputs": [],
28     "id": "ba9fbe8b-3447-4bab-d6b8-08dab40a72fe",
29     "createdAt": "22/10/2022 09:50:19",
30     "createdBy": "leandro@wolfsmartindustries.pt",
31     "updatedAt": "22/10/2022 10:16:26",
32     "updatedBy": "leandro@wolfsmartindustries.pt",
33     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
34     "rowVersion": "00000821036"
35 },
36 "id": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
37 "createdAt": "22/10/2022 09:49:12",
38 "createdBy": "leandro@wolfsmartindustries.pt",
39 "updatedAt": "",
40 "updatedBy": "",
41 "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
42 "rowVersion": "00000824195"
43 ]

```

6.3.3 Blockchain execution times results

As explained in section 5.2, we opted to implement a background task and record all blockchain execution duration's, which played a critical role in the success of this project.

Throughout the *Madural* production process, 67 HTTP requests were issued, and the average response time was 3000ms. After analyzing the data, we determined that the execution times remained consistent regardless of the workload.

Figure 6.2 illustrates the execution times for the harvest, transport, and olive mill processes.

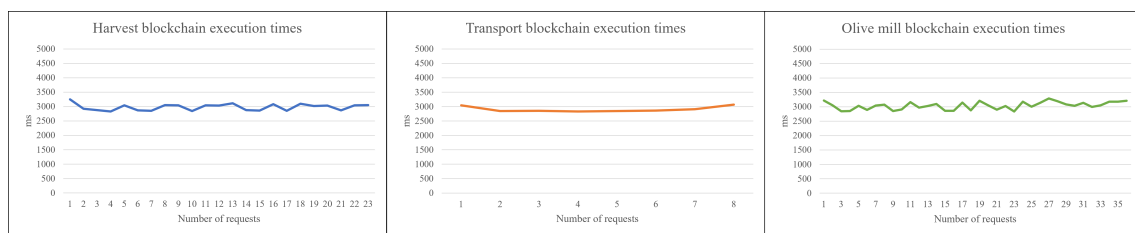


Figure 6.2: Blockchain execution times results.

6.3.4 Quality and sustainability results

As mentioned in section 6.2, sustainability and quality indicators were also recorded for this project. Unfortunately, due to technical limitations, energy and humidity indicators could not be collected. However, we were able to collect data on the consumption of 45l of agricultural diesel and 1000l of water during the production process. The temperature during the harvest and transport was 15°C, while in the olive mill it was 20°C/21°C, and in the Thermo mixer it was 26°C.

Table 6.4 displays all the indicators that were collected during the production of the *Madural* variety. It should also be noted that the Thermo mixer and centrifugation processes were conducted simultaneously.

Table 6.4: Table of olive oil sustainability and quality indicators collected.

		Sustainability			Quality		
		Water	Fuel	Energy	Humidity	Temperature	Execution times
Olive grove	Olive Harvest	—	40 l	—	—	15°C	4Hour (h)24m
Transportation	Transport	—	5 l	—	—	15°C	35m
Olive oil mill	Delivery	1000 l	—	—	—	20°C	52m
	Grinding	—	—	—	—	21°C	1h16m
	Thermo mixer	—	—	—	—	26°C	1h21m
	Centrifugation	—	—	—	—	21°C	1h21m
	Vats	—	—	—	—	21°C	4m

6.4 Discussion

The research project initially had uncertain expectations related to what results could be achieved. It was also unclear whether blockchain technology could be implemented effectively in the context of the project.

To address these uncertainties, the approach taken was to register all data in both, the traditional off-chain database (SQL server) and in the blockchain through a background task. This ensured that the performance of the application was not impacted while also logging the execution times that were crucial for analyzing the results. The primary goal was to have all the data recorded accurately and efficiently.

Through this approach, we realized that it is ideal to record all traceability in the traditional off-chain database and to register only on the blockchain once the production is completed. This registration should only be carried out by a certified professional. It was noted that the execution time was significantly longer when registering data on the blockchain, taking approximately 3000ms per registration.

The combination of a traditional off-chain database and the blockchain was effective as it was faster to register and query data in a traditional database. The blockchain should only function as a notary, ensuring data immutability. This was done through a background job that runs automatically every day, verifying that the off-chain data in the database is consistent with the immutability of the blockchain.

The analysis of sustainability indicators showed that it is difficult to determine whether the production was sustainable or not. However, it can be inferred that using 45l of agricultural diesel and an old tractor could be harmful to the environment.

Unfortunately, it was not possible to obtain humidity indicators, but temperature and execution times at each stage were recorded, which according to the literature review, are indicative of the best quality for olive oil.

Unfortunately, it was not possible to track the olive oil bottling as well as all quality and sustainability indicators due to uncontrollable external factors.

In conclusion, this study demonstrated that it is possible to leverage the best of both worlds – the speed of a traditional relational database and the immutability of the blockchain – by carefully selecting when and what data to record in the blockchain, and ensuring the accuracy and consistency of data between the off-chain database and the blockchain.

In addition to the results obtained, it is important to consider ways in which the execution times of our tests could be improved. One approach would be to optimize the management of assets by implementing a caching system for the token, thereby avoiding repeated calls to the blockchain. By retrieving the token only once and storing it in a cache, the number of requests made to the blockchain can be reduced, resulting in a significant improvement in execution times.

Another way to optimize asset management is to reduce the amount of unnecessary data transmitted and processed. Specifically, we can minimize the assets by including only the strictly necessary information for executing operations. By carefully analyzing the set of information needed for each operation and removing any excess data, we can reduce the amount of data transmitted and processed, further improving the system's efficiency.

In addition to these optimizations, it is worth considering the use of a caching system for reading data from the blockchain. Because the blockchain is immutable, once data is stored, it cannot be changed or deleted. Therefore, a cache system can be used to store frequently accessed data in memory, reducing the number of reads from the blockchain and thereby improving the system's overall performance.

Taken together, these optimizations would improve the efficiency of our asset management system, reducing execution times and improving the overall performance. Additionally, our tests showed that the system is highly scalable, able to perform well even under heavy loads. With further optimizations and the incorporation of caching systems, we believe our system has the potential to become a leading solution for managing blockchain assets.

Chapter 7

Conslusions and future work

As a conclusion we may say that the objectives of this research work have been accomplished and even exceeded. Thus, a blockchain-based platform to store and trace sustainability indicators in an olive oil value chain was design, created and tested in a real scenario.

A research paper [Fernandes et al., 2022] was also published in: 2022 17th Iberian Conference on Information Systems and Technologies (CISTI) and another one with demonstration and results was submitted to 18th Iberian Conference on Information Systems and Technologies (CISTI).

Blockchain was the selected technology to store information because it allows transparency and immutability of stored information. Thus, since olive oil can be used as an ingredient for the creation of other food products, it allows the information to be used to integrate with information from possible business partners and thus encourage full traceability of more elaborate products.

The use of blockchain technology in a real scenario allowed us to draw our own conclusions about this technology, especially about the cost and performance that are aspects usually pointed out as weak points of this technology. In fact, nowadays, people are increasingly adopting a healthier and more sustainable lifestyle. A healthy and balanced diet is essential for achieving a better quality of life. People are placing more and more importance on the food they consume. Consumers are no longer seeking food based on brand, but rather based on attributes such as nutritional value or sustainability index.

The lack of traceability and authenticity in the olive oil market is a serious problem that affects the quality and trust of consumers. Food safety for olive oil is an important issue that can affect the health of consumers and the reputation of the manufacturer. Unsustainable olive oil production can cause environmental and social damage, and affect the image of the manufacturer in the eyes of conscious consumers. The lack of transparency in the olive oil value chain can make it difficult for consumers to trust and differentiate the product.

Using Blockchain & IoT technologies to ensure traceability and authenticity in olive oil production can help prevent fraud and adulteration, ensuring food safety. Tracking growing and production conditions to ensure sustainability indicators in the olive oil production value chain can also help. Providing transparency of the entire olive oil production value chain to end consumers can help build their trust.

By doing this, we can achieve food safety, ensure product integrity, and increase consumer confidence.

This dissertation paved the way for exciting future work in the area and gave rise to a pitch deck (appendix C) and an investment application to Portugal Ventures. The ultimate goal is to create a solution for traceability and certification in the food industry. With the growing concerns around food safety and the increasing demand for transparency in supply chains, it is more important than ever to ensure that consumers have access to reliable information about the products they are consuming.

The proposed solution will leverage cutting-edge technologies such as blockchain and IoT to create a secure and transparent system for tracking food products from farm to table. By implementing this system, we hope to reduce food waste, improve food safety, and increase consumer trust in the industry. With the support of Portugal Ventures, we believe that we can bring this innovative solution to market and make a real impact in the food industry.

In conclusion, this dissertation is just the beginning of a larger mission to revolutionize the way we think about food traceability and certification. With a strong foundation of research and a clear plan for implementation, we are confident that we can create a solution that will benefit consumers, producers, and retailers alike. We are excited about the opportunity to work with Portugal Ventures or another *business angle* and look forward to taking the next steps towards making this vision a reality.

Wolf Value Chain - From tradition to innovation.

Bibliography

- [csh, 2022] (2022). How is c# compiled? [Online; accessed 06 nov 2022].
- [qua, 2022] (2022). Quartz.net: Job scheduling. [Online; accessed 06 nov 2022].
- [azu, 2022a] (2022a). What is azure devops? [Online; accessed 06 nov 2022].
- [azu, 2022b] (2022b). What is devops? [Online; accessed 06 nov 2022].
- [Adam Hughes, 2022] Adam Hughes, C. S. (2022). Microsoft sql server. [Online; accessed 06 nov 2022].
- [Agrawal et al., 2021] Agrawal, T. K., Kumar, V., Pal, R., Wang, L., and Chen, Y. (2021). Blockchain-based framework for supply chain traceability: A case example of textile and clothing industry. *Computers & Industrial Engineering*, 154:107130.
- [Akana, 2022] Akana (2022). What is a rest api? [Online; accessed 06 nov 2022].
- [Alves. et al., 2021] Alves., L., Carvalhido., T., Cruz., E., and Rosado da Cruz., A. (2021). Using blockchain to trace pdo/pgi/tsg products. In *Proceedings of the 23rd International Conference on Enterprise Information Systems - Volume 2: ICEIS*, pages 368–376. INSTICC, SciTePress.
- [Amani et al., 2018] Amani, S., Bégel, M., Bortin, M., and Staples, M. (2018). Towards verifying ethereum smart contract bytecode in isabelle/hol. In *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2018*, page 66–77, New York, NY, USA. Association for Computing Machinery.
- [ANACOM, 2022] ANACOM (2022). Acesso à internet via satélite - disponibilidade, mensalidades e atributos. Technical report. ANACOM, Autoridade Nacional de Comunicações. <https://www.anacom.pt/render.jsp?contentId=1714138>. [Online; accessed 06 nov 2022].
- [Andrey and Petr, 2019] Andrey, A. and Petr, C. (2019). Review of existing consensus algorithms blockchain. In *2019 International Conference "Quality Management, Transport and Information Security, Information Technologies" (IT&QM&IS)*, pages 124–127.
- [Arena et al., 2019] Arena, A., Bianchini, A., Perazzo, P., Vallati, C., and Dini, G. (2019). Bruschetta: An iot blockchain-based framework for certifying extra virgin olive oil supply chain. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 173–179.

- [Bada et al., 2021] Bada, A. O., Damianou, A., Angelopoulos, C. M., and Katos, V. (2021). Towards a green blockchain: A review of consensus mechanisms and their energy consumption. In *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 503–511.
- [Baer, 2015] Baer, I. M. P. C. (2015). Rastreabilidade de azeites virgens provenientes das cultivares cordovil de serpa e galega vulgar na região do alentejo. Technical report. <http://hdl.handle.net/10400.1/8752>.
- [Bamakan et al., 2020] Bamakan, S. M. H., Motavali, A., and Babaei Bondarti, A. (2020). A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Systems with Applications*, 154:113385.
- [Biswas et al., 2017] Biswas, K., Muthukkumarasamy, V., and Tan, W. L. (2017). Blockchain Based Wine Supply Chain Traceability System. *Proceedings of the 2017 Future Technologies Conference (FTC)*, pages 56–62.
- [Buttafoco, 2019] Buttafoco, O. A. (2019). Emerging solutions for the improvement of food traceability in the eu: Examining the use of blockchain technology for tracing extra virgin olive oil (evoo).
- [Caro et al., 2018] Caro, M. P., Ali, M. S., Vecchio, M., and Giaffreda, R. (2018). Blockchain-based traceability in agri-food supply chain management: A practical implementation. In *2018 IoT Vertical and Topical Summit on Agriculture - Tuscany (IOT Tuscany)*, pages 1–4.
- [Cruz and Rosado da Cruz, 2020] Cruz, E. and Rosado da Cruz, A. M. (2020). Using blockchain to implement traceability on fishery value chain. pages 501–508.
- [Cruz, 2011] Cruz, E. F. (2011). Design science research em sistemas de informação. *no. June*, 2011.
- [Cruz and Cruz, 2020] Cruz, E. F. and Cruz, A. M. R. d. (2020). Design science research for is/it projects: Focus on digital transformation. In *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.
- [Cruz et al., 2014] Cruz, E. F., Machado, R. J., and Santos, M. Y. (2014). Derivation of data-driven software models from business process representations. In *2014 9th International Conference on the Quality of Information and Communications Technology*, pages 276–281.
- [Cruz et al., 2019] Cruz, E. F., Rosado da Cruz, A. M., and Gomes, R. (2019). Analysis of a traceability and quality monitoring platform for the fishery and aquaculture value chain. In *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.
- [Daniel Roth and Luttin, 2022] Daniel Roth, R. A. and Luttin, S. (2022). Overview to asp.net core. [Online; accessed 06 nov 2022].

- [David E. Avison and Nielsen, 1999] David E. Avison, Francis Lau, M. D. M. and Nielsen, P. A. (1999). Action research. *Communications of the ACM*, Vol. 42 No. 1:94 – 97.
- [Dykstra, 2022] Dykstra, T. (2022). The repository and unit of work patterns. [Online; accessed 06 nov 2022].
- [European Comission, 2020] European Comission (2020). Panorâmica da produção e do comércio de azeite na ue. Technical report. Panorâmica da produção e do comércio de azeite na UE. [Online; accessed 06 nov 2022].
- [European Commission, 2007] European Commission (2007). Factsheet: Food traceability. Technical report. [Online; accessed 06 nov 2022].
- [Fablo, 2022a] Fablo (2022a). Fablo. [Online; accessed 06 nov 2022].
- [Fablo, 2022b] Fablo (2022b). Fablo rest. [Online; accessed 06 nov 2022].
- [Fernandes et al., 2022] Fernandes, M. A., Cruz, E. F., and Rosado Da Cruz, A. M. (2022). Smart contract and web dapp for traceability in the olive oil production chain. In *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.
- [Garaus and Treiblmaier, 2021] Garaus, M. and Treiblmaier, H. (2021). The influence of blockchain-based food traceability on retailer choice: The mediating role of trust. *Food Control*, 129:108082.
- [hangfire.io, 2022] hangfire.io (2022). Hangfire overview. [Online; accessed 06 nov 2022].
- [Hevner and Chatterjee, 2010] Hevner, A. and Chatterjee, S. (2010). *Design Science Research in Information Systems*, pages 9–22. Springer US, Boston, MA.
- [Hevner et al., 2004] Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.
- [Hyperledger, 2022] Hyperledger (2022). Hyperledger fabric model. Technical report.
- [Jacobovitz, 2016] Jacobovitz, O. (2016). Blockchain for identity management. *The Lynne and William Frankel Center for Computer Science Department of Computer Science. Ben-Gurion University, Beer Sheva*, 1:9.
- [JWT.io, 2022a] JWT.io (2022a). Introduction to json web tokens. [Online; accessed 06 nov 2022].
- [JWT.io, 2022b] JWT.io (2022b). Jwt.io. [Online; accessed 06 nov 2022].
- [Kamilaris et al., 2019] Kamilaris, A., Fonts, A., and Prenafeta-Boldú, F. X. (2019). The rise of blockchain technology in agriculture and food supply chains. *Trends in Food Science & Technology*, 91:640–652.

- [Lane, 2022] Lane, K. (2022). Intro to apis: What is an api? [Online; accessed 06 nov 2022].
- [Linux foundation, 2022] Linux foundation (2022). What is hyperledger fabric? [Online; accessed 06 nov 2022].
- [Martin, 2022] Martin, M. (2022). Functional vs non functional requirements. [Online; accessed 06 nov 2022].
- [Matzembacher et al., 2018] Matzembacher, D. E., do Carmo Stangherlin, I., Slongo, L. A., and Cataldi, R. (2018). An integration of traceability elements and their impact in consumer's trust. *Food Control*, 92:420–429.
- [Mercuri et al., 2021] Mercuri, F., della Corte, G., and Ricci, F. (2021). Blockchain technology and sustainable business models: A case study of devoleum. *Sustainability*, 13(10).
- [Miatton and Amado, 2020] Miatton, F. and Amado, L. (2020). Fairness, transparency and traceability in the coffee value chain through blockchain innovation. In *2020 International Conference on Technology and Entrepreneurship - Virtual (ICTE-V)*, pages 1–6.
- [Microsoft, 2022a] Microsoft (2022a). Multi-tenant saas database tenancy patterns. [Online; accessed 06 nov 2022].
- [Microsoft, 2022b] Microsoft (2022b). Restful web api design. [Online; accessed 06 nov 2022].
- [Microsoft, 2022c] Microsoft (2022c). What is the windows subsystem for linux? [Online; accessed 06 nov 2022].
- [Min and Cai, 2022] Min, T. and Cai, W. (2022). Portrait of decentralized application users: an overview based on large-scale ethereum data. *CCF Transactions on Pervasive Computing and Interaction*, 4.
- [Morris et al., 2018] Morris, V., Adivi, R., Asara, R., Cousens, M., Gupta, N., Lincoln, N., Mosakowski, B., Sun, H. W., et al. (2018). *Developing a Blockchain Business Network with Hyperledger Composer using the IBM Blockchain Platform Starter Plan*. IBM Redbooks.
- [Mozzila, 2022] Mozzila (2022). Http request methods. [Online; accessed 06 nov 2022].
- [Muzzalupo et al., 2015] Muzzalupo, I., Pisani, F., Greco, F., and Chiappetta, A. (2015). Direct dna amplification from virgin olive oil for traceability and authenticity. *European Food Research and Technology*, 241.
- [Nakamoto and Bitcoin, 2008] Nakamoto, S. and Bitcoin, A. (2008). A peer-to-peer electronic cash system. *Bitcoin*.—URL: <https://bitcoin.org/bitcoin.pdf>, 4:2.
- [Nandi et al., 2021] Nandi, S., Sarkis, J., Hervani, A. A., and Helms, M. M. (2021). Redesigning supply chains using blockchain-enabled circular economy and covid-19 experiences. *Sustainable Production and Consumption*, 27:10–22.

- [Oliveira et al., 2021] Oliveira, J., Lima, J. E., da Silva, D., Kuprych, V., Faria, P. M., Teixeira, C., Ferreira Cruz, E., and Rosado da Cruz, A. M. (2021). Traceability system for quality monitoring in the fishery and aquaculture value chain. *Journal of Agriculture and Food Research*, 5:100169.
- [Olsen and Borit, 2013] Olsen, P. and Borit, M. (2013). How to define traceability. *Trends in Food Science & Technology*, 29:142–150.
- [Pfeffers et al., 2006] Pfeffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., and Bragge, J. (2006). The design science research process: A model for producing and presenting information systems research. In *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006)*, Claremont, CA, USA, pages 83–106.
- [Pradana et al., 2020] Pradana, I. G. M. T., Djatna, T., and Hermadi, I. (2020). Blockchain modeling for traceability information system in supply chain of coffee agroindustry. In *2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 217–224.
- [Rosado da Cruz and Cruz, 2020] Rosado da Cruz, A. M. and Cruz, E. (2020). Blockchain-based traceability platforms as a tool for sustainability. pages 330–337.
- [Rousos, 2022] Rousos, M. (2022). Asp.net core performance best practices. <https://learn.microsoft.com/en-us/aspnet/core/performance/performance-best-practices?view=aspnetcore-6.0> [Online; accessed 06 nov 2022].
- [Safa et al., 2017] Safa, S., El Abbassi, A., Kiai, H., Hafidi, A., Sayadi, S., and Galanakis, C. (2017). *Olive oil production sector: Environmental effects and sustainability challenges*, pages 1–28.
- [Tao et al., 2018] Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., and Sui, F. (2018). Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*, 94.
- [Taylor, 2022] Taylor, D. (2022). Docker tutorial for beginners: Basics, architecture, containers. [Online; accessed 06 nov 2022].
- [Tsai et al., 2021] Tsai, F. M., Bui, T.-D., Tseng, M.-L., Ali, M. H., Lim, M. K., and Chiu, A. S. (2021). Sustainable supply chain management trends in world regions: A data-driven analysis. *Resources, Conservation and Recycling*, 167:105421.
- [Valenta and Sandner, 2017] Valenta, M. and Sandner, P. (2017). Comparison of ethereum, hyperledger fabric and corda. *Frankfurt School Blockchain Center*, 8:1–8.
- [Venable, 2006] Venable, J. (2006). The role of theory and theorising in design science research. *First International Conference on Design Science Research in Information Systems and Technology*.

- [Violino et al., 2019] Violino, S., Pallottino, F., Sperandio, G., Figorilli, S., Antonucci, F., Ioannoni, V., Fappiano, D., and Costa, C. (2019). Are the innovative electronic labels for extra virgin olive oil sustainable, traceable, and accepted by consumers? *Foods*, 8(11).
- [Walker, 2022] Walker, A. (2022). Uml use case diagram: Tutorial with example. [Online; accessed 06 nov 2022].
- [Zheng et al., 2018] Zheng, Z., Xie, S., Dai, H.-N., Chen, X., and Wang, H. (2018). Blockchain challenges and opportunities: A survey. *International journal of web and grid services*, 14(4):352–375.

Appendix A

Pictures

This appendix has some pictures captured during *Madural* production described in chapter 6.

A.0.0.1 Harvest pictures

The figure A.1 illustrates the olive tree shaking, the moment loading of the pallets and the pallets with olives.



Figure A.1: Harvest pictures.

A.0.0.2 Transport pictures

The figure A.2 illustrates the tractor with the pallets containing olives and the route between the olive grove and the olive mill.

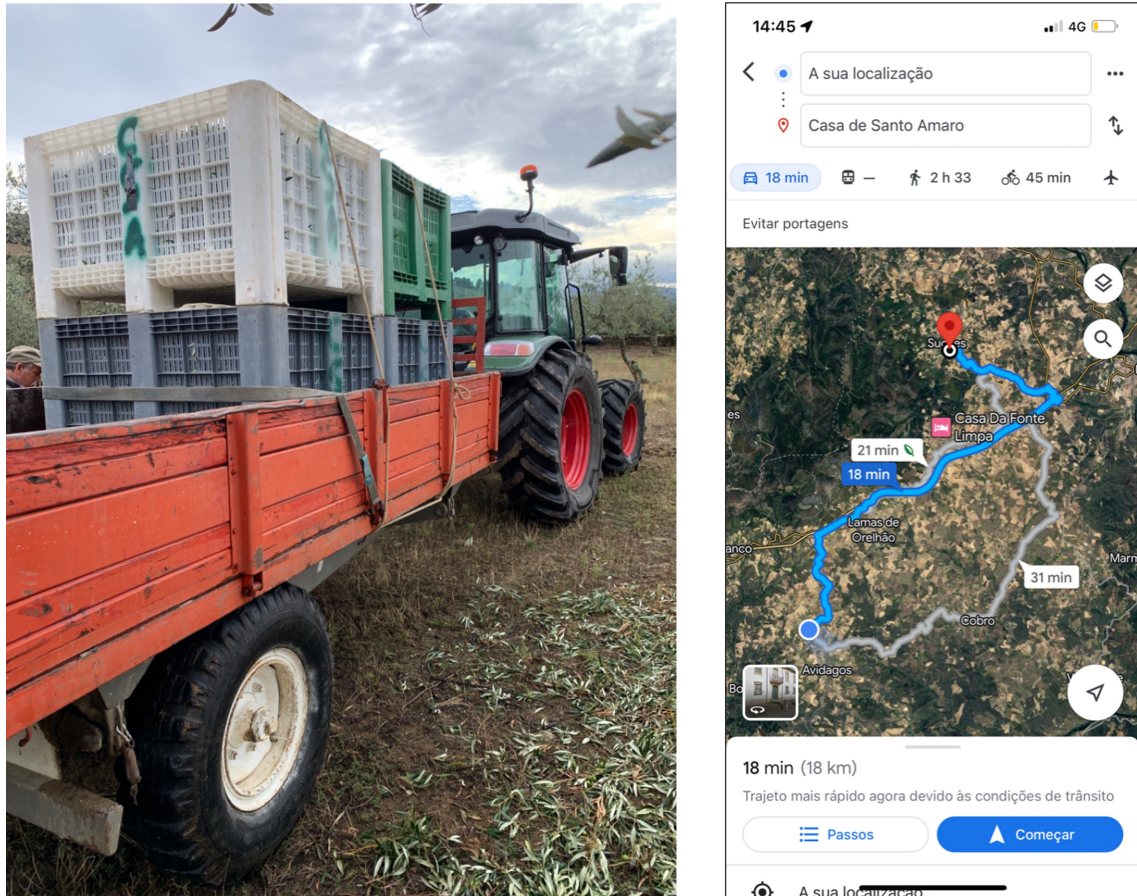


Figure A.2: Transport pictures.

A.0.0.3 Olive mill pictures

The figure A.3 illustrates the some steps like delivery, griding, thermo mixer and centrifugation in the olive mill.

And also illustrates the moment that we finally taste the **EVOO**.



Figure A.3: Olive mill pictures

Appendix B

Blockchain data for Madural variety production

This appendix contains the full JSON object for *Madural* production described in 6.

```
1 {
2   "code": "P-10.22-01",
3   "startAt": "22/10/2022 09:48:00",
4   "endAt": "",
5   "varietyId": "071c0e9b-72dd-48f4-2264-08daaf76b9ff",
6   "variety": "Madural",
7   "notes": "Tempo chuvoso\r\nMas com abertura durante a manha.",
8   "activities": [
9     {
10       "code": "A-10.22-01",
11       "productionId": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
12       "startAt": "22/10/2022 09:49:00",
13       "endAt": "22/10/2022 10:16:00",
14       "activityType": "Harvest",
15       "activityOutputType": "BoxedOlives",
16       "quantityOutput": "0,5",
17       "unitType": "Unity",
18       "notes": "Tempo de chuva, o que dificultou a apanha.\r\nDevido a isso a frequencia do varejo tem de ser mais baixa
19         para nao danificar os Ramos da Oliveira e como resultado nao sai tanta azeitona\r\nMedia de 20 Oliveiras",
20       "oliveGrooveId": "3700968c-7457-448b-c94d-08dab409e770",
21       "olivegrooveName": "Muradelha 1",
22       "olivegrooveAddress": "CMFP+88M Pereira",
23       "olivegrooveOwner": "Miguel Santos",
24       "oliveMillId": "",
25       "oliveMillAddress": "",
26       "measurements": [
27         {
28           "productionActivityId": "ba9fbe8b-3447-4bab-d6b8-08dab40a72fe",
29           "measurementType": "Temperature",
30           "date": "22/10/2022 09:50:00",
31           "value": "15",
32           "notes": "",
33           "id": "e5d588b8-a4ff-4269-b70a-08dab40ac569",
34           "createdAt": "22/10/2022 09:52:38",
35           "createdBy": "leandro@wolfsmartindustries.pt",
36           "updatedAt": "",
37           "updatedBy": "",
38           "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
39           "rowVersion": "000008209255"
40         }
41       ],
42       "inputs": [],
43       "id": "ba9fbe8b-3447-4bab-d6b8-08dab40a72fe",
44       "createdAt": "22/10/2022 09:50:19",
45       "createdBy": "leandro@wolfsmartindustries.pt",
46       "updatedAt": "22/10/2022 10:16:26",
47       "updatedBy": "leandro@wolfsmartindustries.pt",
```

```

48     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
49     "rowVersion": "00000821036"
50 },
51 {
52     "code": "A-10.22-02",
53     "productionId": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
54     "startAt": "22/10/2022 10:16:00",
55     "endAt": "22/10/2022 10:49:00",
56     "activityType": "Harvest",
57     "activityOutputType": "BoxedOlives",
58     "quantityOutput": "1,25",
59     "unitType": "Unity",
60     "notes": "continuacao da apanha 1\r\nInfo tractor da apanha gasta em\r\nmedia em 8h de trabalho 60/70 litros gasoleo
        agricula.\r\nApartir da 30 Oliveira a Maquina ja acusava capacidade cheia, mas continuamos um pouco mais",
61     "oliveGrooveId": "3700968c-7457-448b-c94d-08dab409e770",
62     "olivegrooveName": "Muradilha 1",
63     "olivegrooveAddress": "CMFP+88M Pereira",
64     "olivegrooveOwner": "Miguel Santos",
65     "oliveMillId": "",
66     "oliveMillAddress": "",
67     "measurements": [
68         {
69             "productionActivityId": "9c639f1e-b22e-4c67-6e16-08dab40e39a0",
70             "measurementType": "Temperature",
71             "date": "22/10/2022 10:19:00",
72             "value": "15",
73             "notes": "",
74             "id": "28a8c58d-7eaf-428e-756e-08dab40e9db0",
75             "createdAt": "22/10/2022 10:20:09",
76             "createdBy": "leandro@wolfsmartindustries.pt",
77             "updatedAt": "",
78             "updatedBy": "",
79             "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
80             "rowVersion": "00000824124"
81         }
82     ],
83     "inputs": [],
84     "id": "9c639f1e-b22e-4c67-6e16-08dab40e39a0",
85     "createdAt": "22/10/2022 10:17:21",
86     "createdBy": "leandro@wolfsmartindustries.pt",
87     "updatedAt": "22/10/2022 11:09:03",
88     "updatedBy": "leandro@wolfsmartindustries.pt",
89     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
90     "rowVersion": "00000824112"
91 },
92 {
93     "code": "A-10.22-03",
94     "productionId": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
95     "startAt": "22/10/2022 10:56:00",
96     "endAt": "22/10/2022 11:31:00",
97     "activityType": "Harvest",
98     "activityOutputType": "BoxedOlives",
99     "quantityOutput": "0,75",
100    "unitType": "Unity",
101    "notes": "continuacao da apanha 1 e 2",
102    "oliveGrooveId": "255f6f1b-547f-47df-e70f-08dab4140986",
103    "olivegrooveName": "Muradilha 2",
104    "olivegrooveAddress": "CMCP+HF6 Pereira",
105    "olivegrooveOwner": "Miguel Santos",
106    "oliveMillId": "",
107    "oliveMillAddress": "",
108    "measurements": [
109        {
110            "productionActivityId": "19632a97-1b85-43a1-6e17-08dab40e39a0",
111            "measurementType": "Temperature",
112            "date": "22/10/2022 11:00:00",
113            "value": "16",
114            "notes": "",
115            "id": "d5725de0-119d-4090-756f-08dab40e9db0",
116            "createdAt": "22/10/2022 11:00:20",
117            "createdBy": "leandro@wolfsmartindustries.pt",
118            "updatedAt": "",
119            "updatedBy": "",
120            "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
121            "rowVersion": "00000824189"
122        }
123    ],

```

```

124     "inputs": [],
125     "id": "19632a97-1b85-43a1-6e17-08dab40e39a0",
126     "createdAt": "22/10/2022 10:56:55",
127     "createdBy": "leandro@wolfsmartindustries.pt",
128     "updatedAt": "22/10/2022 11:31:42",
129     "updatedBy": "leandro@wolfsmartindustries.pt",
130     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
131     "rowVersion": "000008241123"
132   },
133   {
134     "code": "A-10.22-04",
135     "productionId": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
136     "startAt": "22/10/2022 13:58:00",
137     "endAt": "22/10/2022 14:40:00",
138     "activityType": "Harvest",
139     "activityOutputType": "BoxedOlives",
140     "quantityOutput": "1,5",
141     "unitType": "Unity",
142     "notes": "Estamos a apanhar com chuva\r\nApanha, apenas para completar as 4 caixas de madural.\r\nObjectivo fazer 1
143     caixa e meia",
144     "oliveGrooveId": "545b58f8-0618-42b1-e710-08dab4140986",
145     "olivegrooveName": "Castelar",
146     "olivegrooveAddress": "CM6P+MJ8 Pereira",
147     "olivegrooveOwner": "Miguel Santos",
148     "oliveMillId": "",
149     "oliveMillAddress": "",
150     "measurements": [
151       {
152         "productionActivityId": "d44b3056-e09d-4613-6e18-08dab40e39a0",
153         "measurementType": "Temperature",
154         "date": "22/10/2022 14:00:00",
155         "value": "17",
156         "notes": "",
157         "id": "03b3c882-720b-471f-7570-08dab40e9db0",
158         "createdAt": "22/10/2022 14:00:35",
159         "createdBy": "leandro@wolfsmartindustries.pt",
160         "updatedAt": "",
161         "updatedBy": "",
162         "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
163         "rowVersion": "000008241147"
164       }
165     ],
166     "inputs": [],
167     "id": "d44b3056-e09d-4613-6e18-08dab40e39a0",
168     "createdAt": "22/10/2022 13:59:10",
169     "createdBy": "leandro@wolfsmartindustries.pt",
170     "updatedAt": "22/10/2022 14:40:43",
171     "updatedBy": "leandro@wolfsmartindustries.pt",
172     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
173     "rowVersion": "000008241198"
174   },
175   {
176     "code": "T-10.22-01",
177     "productionId": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
178     "startAt": "22/10/2022 14:50:00",
179     "endAt": "22/10/2022 15:25:00",
180     "activityType": "Transport",
181     "activityOutputType": "BoxedOlives",
182     "quantityOutput": "4",
183     "unitType": "Unity",
184     "notes": "o transporte e feito no tractor o que faz demorar muito tempo a viagem",
185     "oliveGrooveId": "545b58f8-0618-42b1-e710-08dab4140986",
186     "olivegrooveName": "Castelar",
187     "olivegrooveAddress": "CM6P+MJ8 Pereira",
188     "olivegrooveOwner": "Miguel Santos",
189     "oliveMillId": "b9086453-d171-46db-a4ff-08dab368115a",
190     "oliveMillAddress": "Francisco Pavao",
191     "measurements": [
192       {
193         "productionActivityId": "a8d7cd0d-fb0e-4301-6e19-08dab40e39a0",
194         "measurementType": "Distance",
195         "date": "22/10/2022 14:51:00",
196         "value": "18",
197         "notes": "",
198         "id": "1b0a1fe2-24a4-4440-7571-08dab40e9db0",
199         "createdAt": "22/10/2022 14:51:29",
200         "createdBy": "leandro@wolfsmartindustries.pt",

```

```

200     "updatedAt": "",
201     "updatedBy": "",
202     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
203     "rowVersion": "000008241239"
204   }
205 ],
206 "inputs": [
207   {
208     "quantity": "0,5",
209     "quantityRemaning": "0,5",
210     "productionActivityId": "a8d7cd0d-fb0e-4301-6e19-08dab40e39a0",
211     "productionActivityInputId": "ba9fbe8b-3447-4bab-d6b8-08dab40a72fe",
212     "productionActivityInputCode": "A-10.22-01",
213     "id": "ae8ac181-00db-47d2-4f01-08dab4344cab",
214     "createdAt": "22/10/2022 14:49:54",
215     "createdBy": "leandro@wolfsmartindustries.pt",
216     "updatedAt": "",
217     "updatedBy": "",
218     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
219     "rowVersion": "000008241223"
220   },
221   {
222     "quantity": "1,25",
223     "quantityRemaning": "1,25",
224     "productionActivityId": "a8d7cd0d-fb0e-4301-6e19-08dab40e39a0",
225     "productionActivityInputId": "9c639f1e-b22e-4c67-6e16-08dab40e39a0",
226     "productionActivityInputCode": "A-10.22-02",
227     "id": "5c2ef339-3fcc-466c-4f02-08dab4344cab",
228     "createdAt": "22/10/2022 14:50:11",
229     "createdBy": "leandro@wolfsmartindustries.pt",
230     "updatedAt": "",
231     "updatedBy": "",
232     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
233     "rowVersion": "000008241227"
234   },
235   {
236     "quantity": "0,75",
237     "quantityRemaning": "0,75",
238     "productionActivityId": "a8d7cd0d-fb0e-4301-6e19-08dab40e39a0",
239     "productionActivityInputId": "19632a97-1b85-43a1-6e17-08dab40e39a0",
240     "productionActivityInputCode": "A-10.22-03",
241     "id": "e81ed09d-852a-4a03-4f03-08dab4344cab",
242     "createdAt": "22/10/2022 14:50:30",
243     "createdBy": "leandro@wolfsmartindustries.pt",
244     "updatedAt": "",
245     "updatedBy": "",
246     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
247     "rowVersion": "000008241231"
248   },
249   {
250     "quantity": "1,5",
251     "quantityRemaning": "1,5",
252     "productionActivityId": "a8d7cd0d-fb0e-4301-6e19-08dab40e39a0",
253     "productionActivityInputId": "d44b3056-e09d-4613-6e18-08dab40e39a0",
254     "productionActivityInputCode": "A-10.22-04",
255     "id": "da5abb1b-51f1-46cb-4f04-08dab4344cab",
256     "createdAt": "22/10/2022 14:50:49",
257     "createdBy": "leandro@wolfsmartindustries.pt",
258     "updatedAt": "",
259     "updatedBy": "",
260     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
261     "rowVersion": "000008241235"
262   }
263 ],
264 "id": "a8d7cd0d-fb0e-4301-6e19-08dab40e39a0",
265 "createdAt": "22/10/2022 14:47:59",
266 "createdBy": "leandro@wolfsmartindustries.pt",
267 "updatedAt": "22/10/2022 15:35:28",
268 "updatedBy": "leandro@wolfsmartindustries.pt",
269 "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
270 "rowVersion": "0000091681"
271 },
272 {
273   "code": "E-10.22-01",
274   "productionId": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
275   "startAt": "22/10/2022 15:42:00",
276   "endAt": "22/10/2022 16:34:00",

```



```

277     "activityType": "Entrance",
278     "activityOutputType": "WeightedOlives",
279     "quantityOutput": "1162,7",
280     "unitType": "Kg",
281     "notes": "",
282     "oliveGrooveId": "",
283     "olivegrooveName": "",
284     "olivegrooveAddress": "",
285     "olivegrooveOwner": "",
286     "oliveMillId": "b9086453-d171-46db-a4ff-08dab368115a",
287     "oliveMillAddress": "Francisco Pavao",
288     "measurements": [
289         {
290             "productionActivityId": "091fce2d-1b2f-48f2-a223-08dab43bcac1",
291             "measurementType": "Temperature",
292             "date": "22/10/2022 15:46:00",
293             "value": "20",
294             "notes": "null",
295             "id": "ab468e88-043f-4609-e185-08dab43c38b2",
296             "createdAt": "22/10/2022 15:46:36",
297             "createdBy": "leandro@wolfsmartindustries.pt",
298             "updatedAt": "22/10/2022 16:08:43",
299             "updatedBy": "leandro@wolfsmartindustries.pt",
300             "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
301             "rowVersion": "00000916106"
302         },
303         {
304             "productionActivityId": "091fce2d-1b2f-48f2-a223-08dab43bcac1",
305             "measurementType": "WaterConsumption",
306             "date": "22/10/2022 16:12:00",
307             "value": "1000",
308             "notes": "Esta agua e mudada ao dia.\r\nPor isso nao e usada apenas na lavagem desta entrada",
309             "id": "7b6536c6-cb35-4c70-e186-08dab43c38b2",
310             "createdAt": "22/10/2022 16:12:48",
311             "createdBy": "leandro@wolfsmartindustries.pt",
312             "updatedAt": "",
313             "updatedBy": "",
314             "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
315             "rowVersion": "00000916112"
316         }
317     ],
318     "inputs": [
319         {
320             "quantity": "4",
321             "quantityRemaning": "4",
322             "productionActivityId": "091fce2d-1b2f-48f2-a223-08dab43bcac1",
323             "productionActivityInputId": "a8d7cd0d-fb0e-4301-6e19-08dab40e39a0",
324             "productionActivityInputCode": "T-10.22-01",
325             "id": "739993c9-4f00-490d-17a4-08dab43c0a17",
326             "createdAt": "22/10/2022 15:45:18",
327             "createdBy": "leandro@wolfsmartindustries.pt",
328             "updatedAt": "",
329             "updatedBy": "",
330             "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
331             "rowVersion": "0000091694"
332         }
333     ],
334     "id": "091fce2d-1b2f-48f2-a223-08dab43bcac1",
335     "createdAt": "22/10/2022 15:43:32",
336     "createdBy": "leandro@wolfsmartindustries.pt",
337     "updatedAt": "22/10/2022 16:34:16",
338     "updatedBy": "leandro@wolfsmartindustries.pt",
339     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
340     "rowVersion": "00000916120"
341 },
342 {
343     "code": "M-10.22-01",
344     "productionId": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
345     "startAt": "22/10/2022 16:45:00",
346     "endAt": "22/10/2022 18:01:00",
347     "activityType": "Crushing",
348     "activityOutputType": "OlivePaste",
349     "quantityOutput": "1162,7",
350     "unitType": "Kg",
351     "notes": "Moagem(Moenda) e Maxalacao(Termo Batedora) Sao um ciclo continuo.\r\nAssim que comeca uma, a outra arranca logo a seguir.\r\nDepois acaba a moagem e tambem acaba a macalacao.",
352     "oliveGrooveId": "",

```

```

353     "olivegrooveName": "",
354     "olivegrooveAddress": "",
355     "olivegrooveOwner": "",
356     "oliveMillId": "b9086453-d171-46db-a4ff-08dab368115a",
357     "oliveMillAddress": "Francisco Pavao",
358     "measurements": [
359         {
360             "productionActivityId": "0a08a525-b41a-441b-a224-08dab43bcac1",
361             "measurementType": "Temperature",
362             "date": "22/10/2022 16:41:00",
363             "value": "21",
364             "notes": "O dispositivo nao esta em bom funcionamento.",
365             "id": "14d8ce67-19a5-4e40-e187-08dab43c38b2",
366             "createdAt": "22/10/2022 16:42:10",
367             "createdBy": "leandro@wolfsmartindustries.pt",
368             "updatedAt": "",
369             "updatedBy": "",
370             "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
371             "rowVersion": "00000916133"
372         }
373     ],
374     "inputs": [
375         {
376             "quantity": "1162,7",
377             "quantityRemaning": "1162,7",
378             "productionActivityId": "0a08a525-b41a-441b-a224-08dab43bcac1",
379             "productionActivityInputId": "091fce2d-1b2f-48f2-a223-08dab43bcac1",
380             "productionActivityInputCode": "E-10.22-01",
381             "id": "6bac322b-2da6-480d-17a7-08dab43c0a17",
382             "createdAt": "22/10/2022 18:17:20",
383             "createdBy": "leandro@wolfsmartindustries.pt",
384             "updatedAt": "",
385             "updatedBy": "",
386             "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
387             "rowVersion": "00000916211"
388         }
389     ],
390     "id": "0a08a525-b41a-441b-a224-08dab43bcac1",
391     "createdAt": "22/10/2022 16:40:51",
392     "createdBy": "leandro@wolfsmartindustries.pt",
393     "updatedAt": "22/10/2022 22:03:23",
394     "updatedBy": "marcelo@wolfsmartindustries.pt",
395     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
396     "rowVersion": "0000091729"
397 },
398 {
399     "code": "B-10.22-01",
400     "productionId": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
401     "startAt": "22/10/2022 16:48:00",
402     "endAt": "22/10/2022 18:01:00",
403     "activityType": "Centrifugation",
404     "activityOutputType": "OlivePaste",
405     "quantityOutput": "1162,7",
406     "unitType": "Kg",
407     "notes": "",
408     "oliveGrooveId": "",
409     "olivegrooveName": "",
410     "olivegrooveAddress": "",
411     "olivegrooveOwner": "",
412     "oliveMillId": "b9086453-d171-46db-a4ff-08dab368115a",
413     "oliveMillAddress": "Francisco Pavao",
414     "measurements": [
415         {
416             "productionActivityId": "65135a1a-9722-4f1e-a225-08dab43bcac1",
417             "measurementType": "Temperature",
418             "date": "22/10/2022 16:49:00",
419             "value": "21",
420             "notes": "Temperatura ambiente",
421             "id": "e3fffc80d-090a-424d-e188-08dab43c38b2",
422             "createdAt": "22/10/2022 16:50:13",
423             "createdBy": "leandro@wolfsmartindustries.pt",
424             "updatedAt": "",
425             "updatedBy": "",
426             "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
427             "rowVersion": "00000916146"
428         }
429     ],

```



```

430     "inputs": [
431         {
432             "quantity": "1162,7",
433             "quantityRemaning": "1162,7",
434             "productionActivityId": "65135a1a-9722-4f1e-a225-08dab43bcac1",
435             "productionActivityInputId": "0a08a525-b41a-441b-a224-08dab43bcac1",
436             "productionActivityInputCode": "M-10.22-01",
437             "id": "a6090b47-34a3-4e6d-17a6-08dab43c0a17",
438             "createdAt": "22/10/2022 16:49:30",
439             "createdBy": "leandro@wolfsmartindustries.pt",
440             "updatedAt": "",
441             "updatedBy": "",
442             "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
443             "rowVersion": "00000916142"
444         }
445     ],
446     "id": "65135a1a-9722-4f1e-a225-08dab43bcac1",
447     "createdAt": "22/10/2022 16:48:47",
448     "createdBy": "leandro@wolfsmartindustries.pt",
449     "updatedAt": "22/10/2022 21:52:07",
450     "updatedBy": "marcelo@wolfsmartindustries.pt",
451     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
452     "rowVersion": "0000091723"
453 },
454 {
455     "code": "C-10.22-01",
456     "productionId": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
457     "startAt": "22/10/2022 17:57:00",
458     "endAt": "22/10/2022 19:18:00",
459     "activityType": "Vats",
460     "activityOutputType": "OliveOil",
461     "quantityOutput": "132",
462     "unitType": "Kg",
463     "notes": "11% de rendimento em kg\r\n144 em litros, densidade do azeite 0,906",
464     "oliveGrooveId": "",
465     "olivegrooveName": "",
466     "olivegrooveAddress": "",
467     "olivegrooveOwner": "",
468     "oliveMillId": "b9086453-d171-46db-a4ff-08dab368115a",
469     "oliveMillAddress": "Francisco Pavao",
470     "measurements": [
471         {
472             "productionActivityId": "f6196109-b7c7-43fc-a229-08dab43bcac1",
473             "measurementType": "Temperature",
474             "date": "22/10/2022 18:02:00",
475             "value": "21",
476             "notes": "",
477             "id": "403e2a4c-d26b-4038-e189-08dab43c38b2",
478             "createdAt": "22/10/2022 18:02:48",
479             "createdBy": "leandro@wolfsmartindustries.pt",
480             "updatedAt": "",
481             "updatedBy": "",
482             "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
483             "rowVersion": "00000916187"
484         },
485         {
486             "productionActivityId": "f6196109-b7c7-43fc-a229-08dab43bcac1",
487             "measurementType": "Taste",
488             "date": "22/10/2022 18:58:00",
489             "value": "4",
490             "notes": "Tracos de maca verde, ligeiramente amarga e picante.\r\n",
491             "id": "25fac5d3-c72d-4ace-e18a-08dab43c38b2",
492             "createdAt": "22/10/2022 19:01:03",
493             "createdBy": "leandro@wolfsmartindustries.pt",
494             "updatedAt": "",
495             "updatedBy": "",
496             "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
497             "rowVersion": "00000916225"
498         }
499     ],
500     "inputs": [],
501     "id": "f6196109-b7c7-43fc-a229-08dab43bcac1",
502     "createdAt": "22/10/2022 17:58:24",
503     "createdBy": "leandro@wolfsmartindustries.pt",
504     "updatedAt": "22/10/2022 19:48:06",
505     "updatedBy": "leandro@wolfsmartindustries.pt",
506     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",

```

```

507     "rowVersion": "000009179"
508   },
509   {
510     "code": "Cubas-10.22-01",
511     "productionId": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
512     "startAt": "22/10/2022 19:40:00",
513     "endAt": "22/10/2022 19:44:00",
514     "activityType": "Blending",
515     "activityOutputType": "OliveOil",
516     "quantityOutput": "132",
517     "unitType": "Kg",
518     "notes": "",
519     "oliveGrooveId": "",
520     "olivegrooveName": "",
521     "olivegrooveAddress": "",
522     "olivegrooveOwner": "",
523     "oliveMillId": "b9086453-d171-46db-a4ff-08dab368115a",
524     "oliveMillAddress": "Francisco Pavao",
525     "measurements": [],
526     "inputs": [
527       {
528         "quantity": "132",
529         "quantityRemaining": "132",
530         "productionActivityId": "22cb4e56-a6a6-4ae0-a22b-08dab43bcac1",
531         "productionActivityInputId": "f6196109-b7c7-43fc-a229-08dab43bcac1",
532         "productionActivityInputCode": "C-10.22-01",
533         "id": "4cd63a90-df3d-4130-17a8-08dab43c0a17",
534         "createdAt": "22/10/2022 19:45:11",
535         "createdBy": "leandro@wolfsmartindustries.pt",
536         "updatedAt": "",
537         "updatedBy": "",
538         "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
539         "rowVersion": "000009175"
540       }
541     ],
542     "id": "22cb4e56-a6a6-4ae0-a22b-08dab43bcac1",
543     "createdAt": "22/10/2022 19:41:36",
544     "createdBy": "leandro@wolfsmartindustries.pt",
545     "updatedAt": "22/10/2022 19:44:59",
546     "updatedBy": "leandro@wolfsmartindustries.pt",
547     "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
548     "rowVersion": "000009171"
549   }
550 ],
551 "id": "02fbd269-2381-4f5f-d2e1-08dab40a4ad6",
552 "createdAt": "22/10/2022 09:49:12",
553 "createdBy": "leandro@wolfsmartindustries.pt",
554 "updatedAt": "22/10/2022 11:02:16",
555 "updatedBy": "leandro@wolfsmartindustries.pt",
556 "tenantId": "e6fe784c-e9a6-49d3-e28a-08dab200dc83",
557 "rowVersion": "00000824195"
558 }

```

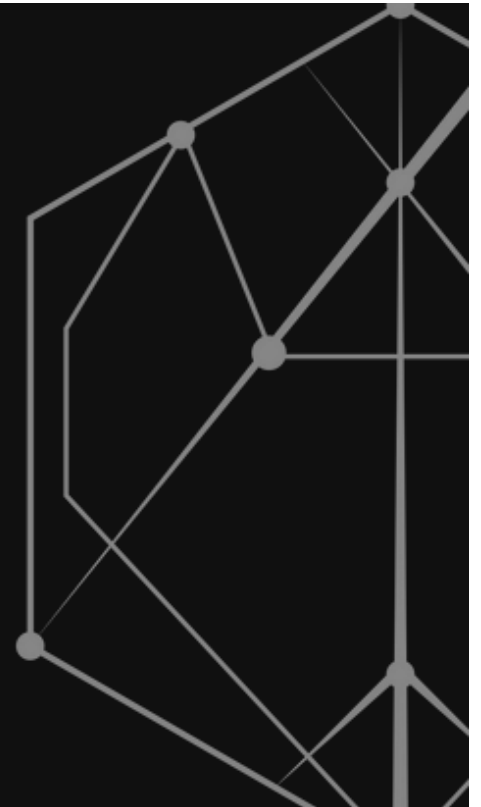
Appendix C

Pitch deck

Pitch deck that was sent for an investment project application from Portugal Ventures.

Wolf Value Chain

Da **tradição** à **inovação**.



"People want to know, quite rightly, where ingredients they give to their baby have come from."

Chris Tyas
Global Head of Supply Chain, Nestlé

Wolf Value Chain

Introdução.

- Hoje em dia, as pessoas estão, cada vez mais, a adotar um modo de vida mais **saudável e sustentável**.
- Uma alimentação saudável e equilibrada é essencial para se conseguir uma melhor **qualidade de vida**.
- As pessoas estão a dar cada vez mais **importância** aos alimentos que consomem.
- Os consumidores já não procuram alimentos com base na marca, mas sim com base em atributos como o **valor nutricional** ou o **índice de sustentabilidade**.

Wolf Value Chain

Problema.

- A falta de **rastreabilidade** e **autenticidade** no mercado de azeite é um problema sério que afeta a qualidade e a confiança dos consumidores.
- A **segurança** alimentar do azeite é um problema importante que pode afetar a saúde dos consumidores e a **reputação** do fabricante.
- A produção de azeite não **sustentável** pode causar danos ambientais e sociais, e afetar a imagem do fabricante perante os consumidores conscientes.
- A falta de **transparência** na cadeia de valor do azeite pode dificultar a confiança dos consumidores e diferenciação do produto.



Wolf Value Chain

Solução.

- Utilizar tecnologias Blockchain & IoT para garantir a **rastreabilidade** e **autenticidade** na produção do azeite de modo a evitar fraudes e adulterações garantindo assim a **segurança** alimentar.
- **Rastrear** condições de cultivo e de produção para garantir indicadores de **sustentabilidade** na cadeia de valor da produção do azeite.
- Fornecer **transparência** de toda a cadeia de valor da produção do azeite aos consumidores finais para garantir a sua **confiança**.



- Integridade - Regulamentação - Certificação - Qualidade -

Wolf Value Chain



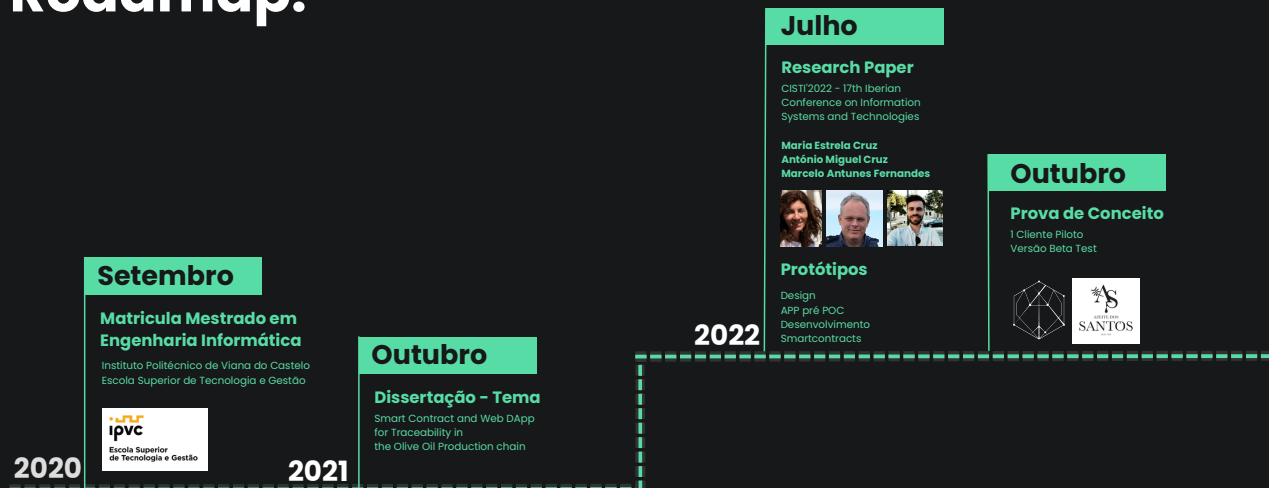
Aumentar a **confiança** dos consumidores.

Garantir a **integridade** dos produtos.

Segurança alimentar.

Wolf Value Chain

Roadmap.



Julho

Research Paper

CISTI'2022 - 17th Iberian Conference on Information Systems and Technologies

Maria Estrela Cruz
António Miguel Cruz
Marcelo Antunes Fernandes



Protótipos

Design
APP pré POC
Desenvolvimento
Smartcontracts

Outubro

Prova de Conceito

1 Cliente Piloto
Versão Beta Test



2022

Wolf Value Chain

Roadmap.



Nova ronda de investimento



Julho

Marketing

Sensores IOT

Novas Funcionalidades

Recrutamento

Hardware Engineer
Comercial

Dezembro

Lançamento Marketplace

Lançamento da MVP

100 Produções Azeite
50 Clientes Piloto na produção de vinho

Portugal Ventures



2023

2024



Wolf Value Chain

Estratégia de marketing.



Apoiar pesquisas e desenvolvimento e **promover** a educação e consciencialização sobre a produção de azeite.

Parcerias com associações e organizações do setor para **melhorar** a imagem e a reputação da empresa.

Realização de eventos próprios para **promover** a produção de azeite e a empresa.

Participação em eventos e conferências do setor para **aumentar** a **visibilidade** da empresa e criar parcerias com outros profissionais e especialistas do setor, como feiras comerciais, exposições e conferências.

Redes sociais e marketing online para **atrair** e manter clientes, com conteúdo atraente e realizar campanhas publicitárias para aumentar a consciencialização sobre o azeite.

Wolf Value Chain

Market Size.

Em média, 3 milhões de toneladas de azeite são produzidas em todo o mundo todos os anos. Cerca de 2 milhões de toneladas desta produção acontece na UE.

MUNDO **3 Mt.p.ano**

EUROPA **2 Mt.p.ano**



66%



15%



13%



5%

LATICÍNIOS

VINUCULTURA

PECUÁRIA

CEREAIS

A UE é também o maior consumidor de azeite, com cerca de 1,5 milhões de toneladas de azeite (50% da produção mundial) consumido na UE todos os anos, e o maior exportador de azeite, exportando cerca de 570 000 toneladas por ano.

Wolf Value Chain

Outros mercados.

Existem outros mercados, em que a Wolf Value Chain também se enquadra na perfeição.



A União Europeia é a maior produtora de vinho a nível mundial. Entre 2016 e 2020, a produção média anual foi de 165 milhões de hectolitros.

Em 2020, representava 45 % da superfície vitícola mundial, 64 % da produção e 48 % do consumo.

Outros mercados.

LATICÍNIOS

PECUÁRIA

CEREAIS

Wolf Value Chain

Noticias.

O mundo escondido da fraude alimentar:
Sabe mesmo o que come?

Por que fraudes no azeite são tão comuns
— e como escolher o mais saudável no
supermercado

ASAE apreende 73.000 Litros de Azeite
falsificado



A reputação do azeite nacional está neste momento a passar por uma contrariedade, mas de forma involuntária e além-fronteiras. No Brasil, uma auditoria ao setor revelou que quase todas as marcas envasilhadas naquele país revelaram a presença de outros óleos, e muitas destas marcas são vendidas como azeite português. “O produto não sai de cá defraudado”, assegura Mariana Matos. “São empresas brasileiras que fazem o milagre dos pães: importam dez e daí produzem mil. As autoridades brasileiras já têm fechado fábricas por causa disso. Mas é uma realidade que causa danos à imagem do nosso azeite.”

<https://visao.sapo.pt/atualidade/sociedade/2018-04-28-o-mundo-escondido-da-fraude-alimentar-sabe-mesmo-o-que-come/>

Wolf Value Chain

Análise de concorrentes.

A Wolf Value Chain distingue-se por fazer a rastreabilidade a todos os níveis, incluindo a produção, bem como indicadores de sustentabilidade.

	 Wolf Value Chain	 IBM Food Trust	 Carrefour Chain
Serviços	• Rastreabilidade • Certificação • Consultoria na produção • Marketplace	• Rastrear transação de produtos alimentares	• Rastrear transação de produtos alimentares
Modelo de negócios	• Produção • Certificação • SaaS • Marketplace	• Pay as you go	• Pay as you go
Mercado	• Europa	• América	• Europa • América
Vantagem competitiva	• Rastreabilidade a todos os níveis • Indicadores de sustentabilidade	• Brand	• Brand

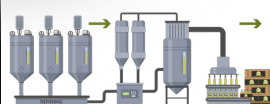
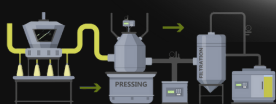
Modelo de negócio.

Certificação de produtos
(Certificação)

Software de gestão
(SaaS)

Instalação de hardware
(Serviços)

Marketplace com produtos certificados
(Marketplace)



Wolf Value Chain

Análise SWOT.



FORÇAS

- Empresa altamente inovadora.
- Projeto único.
- Colaboradores experientes.
- Qualidade.
- Capacidade de analisar novas oportunidades e/ou necessidades do mercado.



FRAQUEZAS

- Falta de recursos.
- Conceito novo.
- Empresa nova.



OPORTUNIDADES

- Possibilidade de expansão para mercados internacionais.
- Escalabilidade do projeto.
- Possibilidade de ser uma referência a nível internacional.
- Possibilidade de criar o standard de produções.



AMEAÇAS

- Fatores externos não controláveis.
- Novos concorrentes.
- Instabilidade na economia.
- Não adoção do produto.

Wolf Value Chain



Leandro Mota

Project manager

3+ experience years project management @Bosch
5+ experience years hardware engineer
PMP® certified
Project management teacher since 2022 @IPVC

<https://www.linkedin.com/in/leandro-mota/>



Marcelo Antunes Fernandes

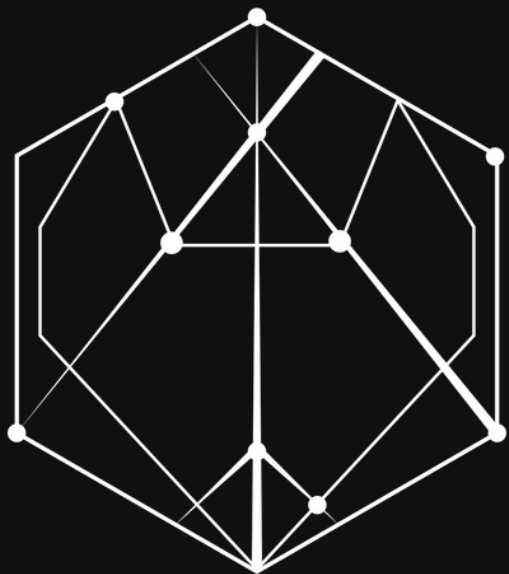
Head of development

9+ experience years software development
Agile software development with Scrum certified
Microsoft® certified
Software engineering teacher since 2021 @IPVC

<https://www.linkedin.com/in/marcelofilipeantunes/>

Olá!
#werarewolfgang.

Wolf Value Chain



Join us!

Morada

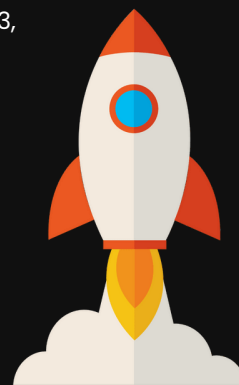
EN101 Avenida Barros & Soares nº423,
Braga, Portugal

Contacto

+351 969 692 675
+351 963 857 456

Email

geral@wolfsmartindustries.pt



Wolf Value Chain