

Desenvolvimento de uma solução integrada de rastreabilidade para a cadeia de valor agroalimentar

João Pedro Lourenço - 42495

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Informática no âmbito da dupla diplomação com a Universidade Tecnológica Federal do Paraná.

Trabalho orientado por:

Prof. Dr. Paulo Alexandre Vara Alves

Prof. Dr. Diego Roberto Antunes

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

2020-2021

Desenvolvimento de uma solução integrada de rastreabilidade para a cadeia de valor agroalimentar

João Pedro Lourenço - 42495

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Informática no âmbito da dupla diplomação com a Universidade Tecnológica Federal do Paraná.

Trabalho orientado por:

Prof. Dr. Paulo Alexandre Vara Alves

Prof. Dr. Diego Roberto Antunes

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

2020-2021

Dedicatória

Dedico este trabalho primeiramente aos meu pais Máximo Machado Lourenço e Marilu Aperecida Rodrigues da Costa que sempre apoiaram meus sonhos e incentivaram meus estudos. Também aos meus irmãos Luiz Pozitel Neto e Gabriela Lourenço pela amizade e atenção dedicadas sempre que precisei. E a todos que de alguma forma me inspiraram a me tornar uma pessoa melhor.

Agradecimentos

Agradeço primeiramente a minha família por todo o apoio, incentivo, paciência e confiança.

Agradeço ao professor Dr. Paulo Alves que me introduziu em um projeto de desenvolvimento de software de um sistema de rastreabilidade, auxiliando na compreensão do problema e acompanhando todo o processo de desenvolvimento, guiando e contribuindo para todas as minhas decisões. Agradeço também a todos os professores que contribuíram com a minha formação acadêmica e profissional durante a minha vida e aos meus amigos e colegas que tornaram a minha jornada mais fácil e agradável. Agradeço à Universidade Tecnológica Federal do Paraná e ao Instituto Politécnico de Bragança por fornecer toda a estrutura e profissionais que contribuíram com o meu crescimento como cidadão e profissional.

Resumo

A expansão do acesso a informação trouxe diversas mudanças a sociedade, uma delas refere-se ao fato de os consumidores estarem cada vez mais interessados em ter conhecimento sobre os processos da cadeia produtiva dos alimentos, que consomem diariamente. Essa mudança trouxe diversos desafios aos governos e empresas do setor alimentício, sendo um deles o de identificar meios de rastrear a cadeia produtiva desde a origem até ao consumidor, visando melhorar a segurança alimentar de um país. Neste contexto uma das abordagens mais comuns são uso e implementação de sistemas de rastreabilidades digitais, cujo o objetivo é inspeccionar o processo de produção, informando o consumidor de forma detalhada sobre o mesmo, dando-lhe o poder de decidir entre consumir ou não um produto, direccionando o mercado para atitudes mais sustentáveis. O presente trabalho visa apresentar um sistema de rastreabilidade, desde o levantamento de requisitos e etapas de desenvolvimento, até a disponibilização de uma solução integrada. A solução proposta integra todas as diversas fases da cadeia, usando conceitos da tecnologia blockchain, útil tanto para os agentes internos da cadeia, quanto para os consumidores. Dessa forma visou-se trazer uma solução para o problema descrito, com uso de tecnologias em ascensão, que tem por objetivo contribuir com a cadeia de valor agroalimentar.

Palavras-chave: Sistemas de rastreabilidade, Cadeia de valor agroalimentar, Blockchain, Aplicações web.

Abstract

The expansion of access to information brought several changes to society, one of which is the fact that consumers are interested in knowing about the processes of the food production chain, which they consume daily. This change introduced challenges to the government and companies in the food sector, one of them is to identify ways to trace a production chain from the origin to the consumer, to improve the food security of a country. In this context, one of the most common approaches is the use and development of digital traceability systems, whose objective is to inspect the production process, informing the consumer in detail about it, giving him the power to decide whether or not to consume a product, directing the market towards more sustainable attitudes. The present work aims to present a traceability system since requirements and all development stages, to the deploy stage. The proposed solution integrates all the different phases of the chain, using blockchain technology concepts, useful both for internal agents in the chain and for consumers. Thus, the aim was to bring a solution to the problem described, using emerging technologies, which aims to contribute to an agri-food value chain.

Keywords: Traceability Systems, Food Supply Chain, Blockchain, Web application.

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	3
1.3	Estrutura do Documento	4
2	Estado da Arte	5
2.1	Rastreabilidade de alimentos	5
2.2	Sistema de rastreabilidade com blockchain	8
2.3	Código de barras	12
3	Levantamento e análise de requisitos	17
3.1	Problema	17
3.2	Abordagem proposta	20
3.2.1	Enquadramento da proposta	20
3.2.1.1	Perguntas gerais	21
3.2.1.2	Perguntas específicas	22
3.2.2	Perfis de utilizadores	24
3.2.2.1	Produtor	25
3.2.2.2	Transportador	25
3.2.2.3	Transformador	25
3.2.2.4	Armazenista	26
3.2.2.5	Retalhista	26

3.2.2.6	Gestor de qualidade	26
3.2.2.7	Consumidor	26
3.2.3	Atores	27
3.2.4	Requisitos Funcionais	30
3.2.5	Requisitos Não Funcionais	34
3.2.5.1	Usabilidade	34
3.2.5.2	Segurança	35
3.2.5.3	Performance	36
3.2.5.4	Interoperabilidade	36
3.2.5.5	Compatibilidade	37
3.2.5.6	Padrões	37
3.2.6	Casos de uso	37
3.2.7	Diagramas de atividades	39
3.2.8	Diagrama de classes	43
3.2.9	Diagramas de sequência	43
3.2.10	Modelo entidade relacionamento	46
3.2.11	Mockups	47
3.2.12	Arquitetura proposta	51
3.2.13	Componentes da Arquitetura	53
4	Desenvolvimento da plataforma de rastreabilidade	57
4.1	Tecnologias	57
4.1.1	React	58
4.1.2	MongoDB	59
4.1.3	Flask	59
4.1.4	Nginx	60
4.1.5	Docker	60
4.2	Camada de interface	61
4.2.1	Estrutura de Arquivos	61

4.2.2	Página Inicial	63
4.2.3	Página de Autenticação	64
4.2.4	Páginas para registrar a rastreabilidade	65
4.2.5	Página para visualizar a rastreabilidade	66
4.2.6	Página de gestão de produções	68
4.2.7	Página de gerenciamento de sublote	71
4.2.8	Controlo de estado	72
4.2.9	Controlo de rotas	73
4.3	Camada de Negócio	74
4.3.1	Módulo de Autenticação	74
4.3.1.1	Iniciar sessão	74
4.3.1.2	Controle de acesso de uma empresa	75
4.3.1.3	Recuperar palavra-passe	75
4.3.1.4	Verificar token	76
4.3.1.5	Registrar funcionários	76
4.3.2	Módulo de Rastreabilidade	76
4.3.3	Blockchain	80
4.4	Implementação da arquitetura	86
4.5	Disponibilização da solução	87
5	Conclusões e Trabalhos Futuros	91
A	User Stories	A1

Lista de Tabelas

3.1	Lista de perfis de utilizadores	24
3.2	Lista de atores da plataforma Biotrace.	29
3.3	User Stories (Não registado/ Consumidor)	30
3.4	User Stories (Utilizadores registados)	31
3.5	User Stories (Administrador)	31
3.6	User Stories (Produtor Gestor)	32
3.7	User Stories (Transportador Gestor)	33
3.8	User Stories (Retalhista Gestor)	34
A.1	User Stories (Não registado/ Consumidor)	A1
A.2	User Stories (Utilizadores registados)	A2
A.3	User Stories (Administrador)	A2
A.4	User Stories (Produtor Gestor)	A3
A.5	User Stories (Produtor Supervisor)	A4
A.6	User Stories (Produtor Funcionário)	A4
A.7	User Stories (Transportador Gestor)	A5
A.8	User Stories (Transportador Supervisor)	A6
A.9	User Stories (Transportador Funcionário)	A6
A.10	User Stories (Transformador Gestor)	A7
A.11	User Stories (Transformador Supervisor)	A8
A.12	User Stories (Transformador Funcionário)	A9
A.13	User Stories (Armazenista Gestor)	A9

A.14 User Stories (Armazenista Supervisor)	A10
A.15 User Stories (Armazenista Funcionário)	A10
A.16 User Stories (Retalhista Gestor)	A11
A.17 User Stories (Retalhista Supervisor)	A12
A.18 User Stories (Retalhista Funcionário)	A12

Lista de Figuras

2.1	Exemplo do sistema de rastreabilidade (adaptado de Franco [12]).	8
2.2	Blocos conectados através de uma chave criptográfica (adaptado de Horiuchi [24]).	10
2.3	Exemplo de um código de barras GS1-128.	14
3.1	Modelo do Ciclo de Vida de Desenvolvimento de Software.	18
3.2	Diagrama de casos de uso	38
3.3	Diagrama de atividades da leitura de QR Code.	40
3.4	Diagrama de atividades da rastreabilidade de um produto.	41
3.5	Diagrama de atividades para apresentar informações de um produto ao cliente.	42
3.6	Diagrama de classes.	43
3.7	Diagrama de sequência do login.	44
3.8	Diagrama de sequência para consultar produções.	45
3.9	Modelo entidade relacionamento inicial.	46
3.10	Modelo entidade relacionamento final.	47
3.11	Página Home Mockup.	48
3.12	Página de autenticação Mockup.	49
3.13	Página de gestão de produção Mockup.	50
3.14	Mockups do fluxo do consumidor.	51
3.15	Arquitetura geral Biotrace.	52
3.16	Componentes funcionais da arquitetura.	53

4.1	Estrutura de arquivos da camada visual (React).	62
4.2	Página inicial.	63
4.3	Páginas do fluxo do consumidor.	64
4.4	Página de autenticação.	65
4.5	Páginas para efetuar leitura do QR Code.	66
4.6	Páginas da rastreabilidade de um produto.	67
4.7	página de gestão de produções.	68
4.8	Modal para visualizar a rastreabilidade.	69
4.9	PDF de um lote cadastrado na plataforma.	70
4.10	Página para gestão dos sublotes.	72
4.11	Fluxo de informação.	73
4.12	Fluxo de informação.	74
4.13	Código para determinar a ação.	78
4.14	Item do dicionário que determina se pode registrar a rastreabilidade.	79
4.15	Decorator para salvar um ledger.	83
4.16	Decorator para editar um ledger.	84
4.17	Decorator para consultar um ledger.	85
4.18	Arquitetura da plataforma.	86
4.19	Dockerfile React.	88
4.20	Dockerfile Flask.	88
4.21	Docker Compose.	89

Capítulo 1

Introdução

Este capítulo tem o objetivo de apresentar a estrutura do trabalho e o problema levantado na pesquisa, realizando um breve enquadramento teórico em volta do cenário atual e finalizando com os objectivos propostos.

1.1 Enquadramento

Nas últimas décadas, tem-se notado que cada vez mais o consumidor demonstra preocupado com os alimentos que consomem. Simultaneamente a essa preocupação surge o interesse em saber de onde vem o alimento, como ele é obtido, como ele é armazenado, as condições de transporte e principalmente se ele é seguro para ser consumido [1].

Esse crescimento de interesse por questões relacionadas à segurança alimentar tem pressionados os produtores e transportadores a disponibilizarem alimentos seguros. Para que isso ocorra, é necessário que o Estado institua políticas públicas capazes de controlar toda a Cadeia de valor agroalimentar (CVAA), garantindo, nesse processo, a adoção das boas práticas de gestão de qualidade do International Standard for Quality Management Systems (ISO), o que implica níveis elevados de segurança dos alimentos [2].

Como resposta ao dever do Estado de fiscalizar a produção de alimentos, surge a noção de rastreabilidade de alimentos, com o intuito de assegurar um elevado nível de proteção e qualidade alimentar, bem como com o objetivo de detectar alimentos que podem expor

o consumidor à nocividade, a rastreabilidade permite identificar o elemento causador e imediatamente o problema pode ser sanado [2].

A proposta da rastreabilidade, portanto, implica em uma política de segurança capaz de permitir a avaliação e o controle dos riscos trazidos por alimentos. Assim sendo, tem-se um sistema de controle que tem como escopo a supervisão e concretização da legislação, sendo esse um sistema que administra toda a CVAA [2].

Cabe, ao Estado, instituir a rastreabilidade dos alimentos, concebendo esta como um procedimento que permite seguir e localizar os produtos, contemplando os aspectos desde a sua matéria-prima até a produção, comercialização, registro, identificação e transmissão da informação ao consumidor [3]. Nesse sentido, a possibilidade de identificação de elementos nocivos à saúde humana, à qualidade de vida e ao meio ambiente pode ocorrer em qualquer etapa da CVAA e, por esse motivo, a rastreabilidade torna-se um elemento essencial à garantia da qualidade alimentar [4].

O maior objetivo de um sistema de rastreabilidade é o de criar um novo hábito no mercado interno e criar uma cultura de participação atuante da sociedade para que esta não consuma produtos que não estejam de acordo com a legislação e que não são seguros [1]. Cabe, ao Estado, elevar o nível da transferência, tornando público os relatórios de inspeção, proporcionando, aos consumidores, a escolha de não consumir alimentos que, de alguma maneira, agredam a saúde [3]. Tal atitude consciente e crítica implicará na disponibilização de alimentos seguros no mercado. Para tanto, devem obter informações sobre os alimentos, escolhendo apenas aqueles que adotem boas práticas de gestão de qualidade [2].

A procura por produtos seguros e de qualidade tem impulsionado a evolução do setor agroalimentar e das empresas, procurando soluções que contribuam para a gestão de informação e de qualidade. No entanto, estas soluções e a informação nelas contidas apresentam-se desconectadas ao longo da CVAA, pouco acessíveis a outras fases da cadeia e, devido ao facto de serem realizadas/ registadas manualmente, resultam numa consulta difícil pelos vários agentes económicos ao longo da CVAA e também por parte do consumidor final.

As informações estarem conectadas ao longo da CVAA é essencial para o sucesso de um sistema de rastreabilidade, porém se elas não estiverem seguras a rastreabilidade é comprometida. Os alimentos além de serem essenciais a vida humana, são percebidos como uma mercadoria comercial comum, com fins lucrativos. Neste contexto, o número de atividades ilegais, promovidas por produtores e fornecedores, cresceu. Essas atividades tem o intuito de obter margens de lucros cada vez maiores, contudo ameaçam a segurança alimentar dos países [5]. Evidencia-se, assim, a importância de criar meios que protejam a integridade dos dados.

Neste contexto, pretende-se endereçar os desafios identificados e promover os benefícios da rastreabilidade de produtos ao longo da CVAA, através da criação de uma solução de rastreabilidade digital, integrada e interoperável que conecte as distintas fases da CVAA e garanta a integridade do dados.

1.2 Objetivos

O presente trabalho faz parte do desenvolvimento de uma plataforma de rastreabilidade, tendo o seu foco em desenvolvimento de software, mais especificamente no desenvolvimento de uma plataforma completa com interfaces gráficas e com um servidor capaz de armazenar e gerenciar toda a cadeia produtiva disponibilizada através de um site responsivo.

Neste cenário, este trabalho propõe realizar as seguintes atividades no âmbito de desenvolvimento da plataforma:

- Desenvolver uma plataforma web que suporte a rastreabilidade da cadeia agroalimentar
- Desenvolver um serviço para gerar e recolher dados nas diversas fases da CVAA
- Desenvolver funcionalidade para gerar códigos de barras nas normas GS1
- Aplicar conceitos da tecnologia blockchain para garantir a integridade dos dados

Os resultados obtidos no processo são um site que pode ser utilizado tanto pelo consumidor de um produto quanto por um funcionário que trabalha na cadeia produtiva de algum produto. Esse site deverá auxiliar os gestores no controle da cadeia, e os consumidores com o objectivo de uma maior transparência. Trazendo um solução flexível para qualquer produto e qualquer cadeia, diminuindo a complexidade do processo de rastreabilidade, e tornando sua existência possível e de fácil acesso.

1.3 Estrutura do Documento

O documento é dividido em 5 capítulos, focando-se nas principais fases do ciclo de desenvolvimento. No capítulo 2 é feito um estudo teórico em volta dos principais temas envolvidos no problema, trazendo um contexto geral e abordando as principais tecnologias utilizadas no processo. No capítulo 3 são abordadas todas as partes que antecedem o desenvolvimento de um software, como o levantamento de requisitos, modelagem do sistema e criação dos mockups do site. No capítulo 4 aborda todo o processo de desenvolvimento da plataforma, desde da definição das tecnologias até a disponibilização da plataforma para uso. Finalmente, o último capítulo apresenta as conclusões e direções de trabalho futuro.

Capítulo 2

Estado da Arte

Neste capítulo é apresentado toda a contextualização teórica acerca dos principais temas relacionados a pesquisa. Através de uma revisão de literatura, são apresentados trabalhos relacionados a rastreabilidade de alimentos, bem como a tecnologia de blockchain aplicada nos sistemas de rastreabilidade, código de barras e normas globais.

2.1 Rastreabilidade de alimentos

Quando se trata de segurança dos alimentos, o aumento crescente de preocupação tem implicado em alterações no processo de produção [6]. Cita-se, como exemplo, a proibição do uso de antibióticos na produção de frango em vigor desde 2012 na União Europeia. [7]. Os consumidores cada vez mais se preocupam com a segurança dos alimentos consumidos [2], [7]. Todavia, além da preocupação significativa com a segurança dos alimentos, com foco no consumidor final, há, também, a preocupação com a introdução de uma cultura de segurança de alimentos nas organizações para que, dessa forma, os colaboradores percebam a importância na cadeia de valor alimentar [8].

Dentro das organizações deve haver a percepção de todos os colaboradores, o que implica a boa comunicação, compromisso ao usar recursos e conscientização quanto aos riscos relacionados à segurança alimentar [9]. Uma vez que os consumidores atuais demandam

produtos mais seguros, o mercado tem buscado ferramentas que aumentam o uso intensivo de dados [10]. Assim, há uma mudança de perspectiva em virtude da necessidade de maior segurança e precisão, o que implica a necessidade da rastreabilidade. Esta é uma ferramenta cada vez mais utilizada para trazer segurança na cadeia de valor alimentar, detectando problema de conservação dos alimentos e o cumprimento da legislação em vigor [11].

A rastreabilidade pode ser definida como a possibilidade de adquirir todas as informações relacionadas a um produto, desde o início da sua cadeia produtiva até o consumidor final, a partir de registos identificados [7], [12]. O objetivo é o de controlar na íntegra o lote de um produto em uma cadeia de valor. A identificação e o registro do lote em cada etapa do processo são aspectos obrigatórios para garantir a procedência de todas as informações envolvidas em uma cadeia de valor [12].

A rastreabilidade, então, permite traçar o caminho da história, uso e localização de uma mercadoria individual por meio da impressão de números de identificação [13]. Esses números de identificação são aplicados em itens individuais ou em lotes, bem como variam entre códigos, datas ou combinações. A rastreabilidade, portanto, previne a entrada de alimentos sem qualidade ou segurança no mercado, assim como simplifica a localização de problemas, pois consegue identificar os locais/etapas que estão fora da normalidade [10].

A rastreabilidade aponta, portanto, os limites aceitáveis voltados à implantação de medidas para o controle, assim como reduz-se o volume de devolução dos produtos (recall) e estabelece responsabilidades [13]. Porém para que a rastreabilidade se consolide no processo produtivo, é necessário que os colaboradores percebam a importância, pois segundo [10] ainda há certa insegurança e os custos para adesão são altos em função da tecnologia necessária, e com isso as empresas hesitam em aderir à rastreabilidade.

A rastreabilidade é uma tendência, que está a caminhar para se tornar uma necessidade do mercado, e quando bem implementada, serve como um sistema de gestão para a empresa proprietária [10]. Neste sentido diversas abordagens de rastreabilidades foram desenvolvidas para auxiliar a rastreabilidade de alimentos e promover seus benefícios [9].

Entre as abordagens, países da União Europeia desenvolveram um Sistema de alerta

rápido para alimentos e rações. As informações são disponibilizadas de forma estruturada por meio de uma plataforma online, que possibilita os entes da cadeia a tomarem decisões mais rápidas [14].

Este estudo [15] indica, que em 2013, o status atual era que muitos produtores de alimentos tinham bons sistemas de rastreabilidade, mas a troca (especialmente a troca eletrônica) de informações entre os elos da cadeia de abastecimento eram muito demorada ou difícil devido à diversidade e natureza proprietária dos respectivos sistemas. Visando esse problema, a UE também desenvolveu o TraceFood, um framework para fornecer um padrão internacional, sem propriedade, para troca eletrônica de dados relacionados à rastreabilidade de alimentos.

Em [16], os atores propõem um sistema de rastreabilidade para a gestão de qualidade na CVAA, utilizando código de barras bidimensional (QR Code) e tecnologia de identificação por radiofrequência. No entanto essa solução não é eficiente, pois não conectam as informações em um sistema, tornando obrigatório todos os participantes da CVAA armazenar as informações necessárias relacionadas ao produto alimentar, para que, quando solicitadas, as informações possam ser fornecidas [9].

Neste estudo [12], podemos ver um sistema de rastreabilidade mais simples com apenas duas fases, representada na figura 2.1, a primeira fase envolve o produtor, que precisa registrar as informações que deseja disponibilizar ao consumidor final, bem como gerar os códigos de barras bidimensional (QRCode) de acordo com os lotes cadastrados. A segunda fase concentra-se no consumidor final, que precisa efectuar a leitura do QRCode acoplado ao rótulo do produto e visualizando as informações disponibilizadas pelo produtor.

O estudo [17] explora uma integração entre uma plataforma móvel ligada a internet e um sistema de informação geográfica. Com base nessa integração, foram criadas funções que permitem mapear e gerenciar dados, manter registros de produção e dar suporte para consulta e tomada de decisão.

Um estudo demonstrou que é possível verificar informações sobre a existência de certificação para alimentos livres de agrotóxicos, data de validade e outros, a partir de um sistema de rastreabilidade electrónica. Além disso, o consumidor também pode verificar

a autenticidade do produto, a localização e os métodos de cultivo [9].

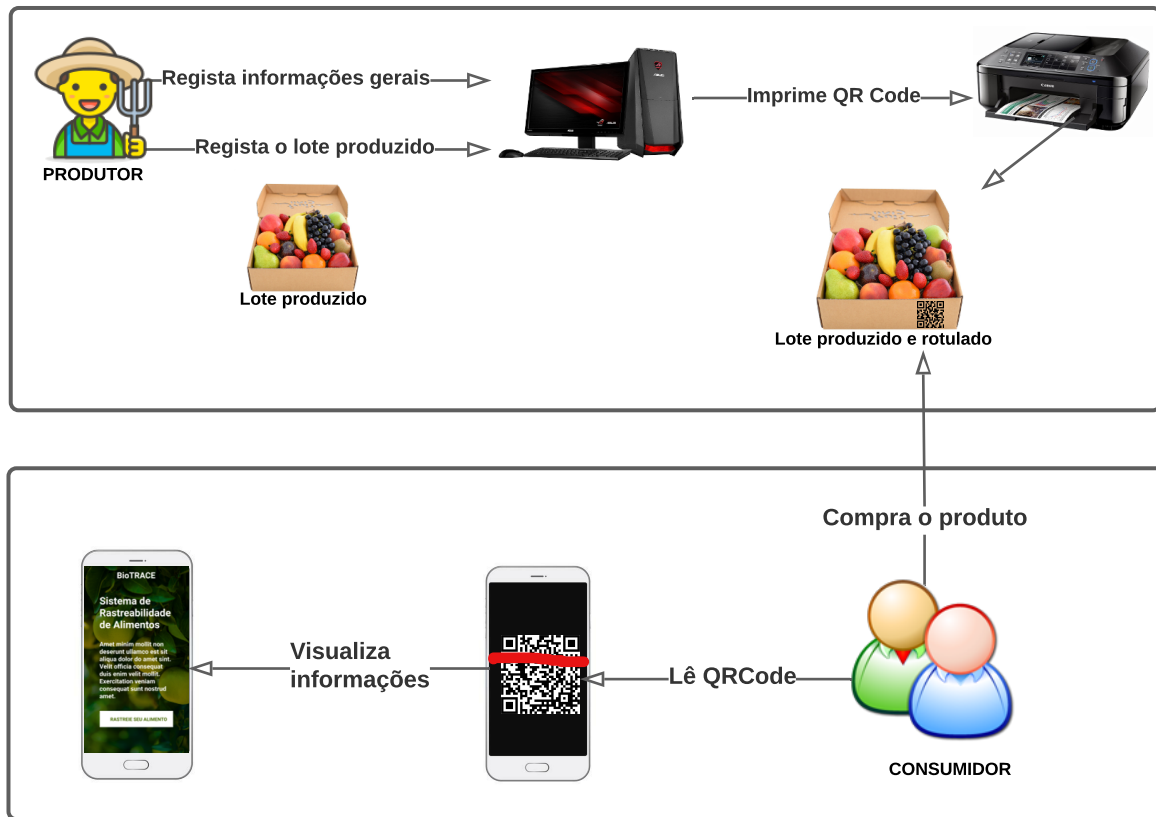


Figura 2.1: Exemplo do sistema de rastreabilidade (adaptado de Franco [12]).

Neste cenário, é possível observar o grande interesse em resolver problemas voltados a segurança alimentar, e é fundamental promover ainda mais ações políticas e governamentais que incetivem as empresas a adotarem a rastreabilidade de alimentos [9].

2.2 Sistema de rastreabilidade com blockchain

Os consumidores têm se tornado cada vez mais exigentes quanto à transparência de toda a cadeia dos alimentos consumidos diariamente [18]. Uma pesquisa realizada nos Estados Unidos com 1.500 pessoas identificou que 70% dos consumidores apontaram que desejam

dos varejistas mais transparência [19]. O Walmart, maior varejista norte-americano, exige, desde 2015, de uma empresa da Guatemala (Las Palmas) produtora de óleo de palma, um certificado de sustentabilidade para que continue a fornecer matéria-prima à empresa estadunidense [20].

Nesse contexto, as empresas têm adotado tecnologias que se mostram eficazes à redução de gaps de desinformação com o foco voltado à segurança alimentar [21]. Uma das tecnologias empregadas por tais empresas é a blockchain, utilizada, sobretudo, por players do mercado financeiro e criptomoedas [22].

O blockchain trata-se de uma tecnologia de banco de dados distribuída e, assim, é construída a partir de uma lista que não pode ser adulterada. Contempla as transações rotuladas por um tempo que são encapsuladas e criptografadas em blocos de informações encadeados [23]. Popularmente, é empregada junto a um conjunto que contempla outros níveis de segurança, como em criptomoedas (como é o caso do Bitcoin) [24]. A sua estrutura permite que diferentes organismos realizem transações entre si em uma rede de computadores na qual ninguém é validado por uma instituição central em razão da combinação entre uma rede peer-to-peer [24].

A tecnologia de Blockchain representa uma nova forma de desenvolver aplicações distribuídas e imutáveis, garantindo a segurança da informação. A IBM oferece, em seu portfólio, o blockchain e, assim, criou uma solução para o rastreamento de cadeias de suprimento de alimentos, sendo que é utilizada por grandes varejistas, como Walmart, Dole e Nestlé [25]. Ademais, diversas estratégias de planejamento têm sido empregadas para viabilizar a tecnologia.

No Blockchain os conjuntos de informações são mantidos em blocos, e para cada alteração validada é criado outro bloco com as novas informações, esse novo bloco é conectado, através de uma chave criptográfica, com o bloco antigo, formando assim uma cadeia interligadas de blocos, que encontra-se representada na figura 2.2.

Essa cadeia contém o histórico de todas alterações, e é replicada a outros computadores que compõem o sistema. Assim, uma adulteração em um dado registro que contém a informação de origem equivalente a um alimento implicaria na edição das informações em

todos os demais blocos da cadeia, armazenados, por sua vez, em computadores diferentes, garantindo, assim, a imutabilidade das informações inseridas [19].

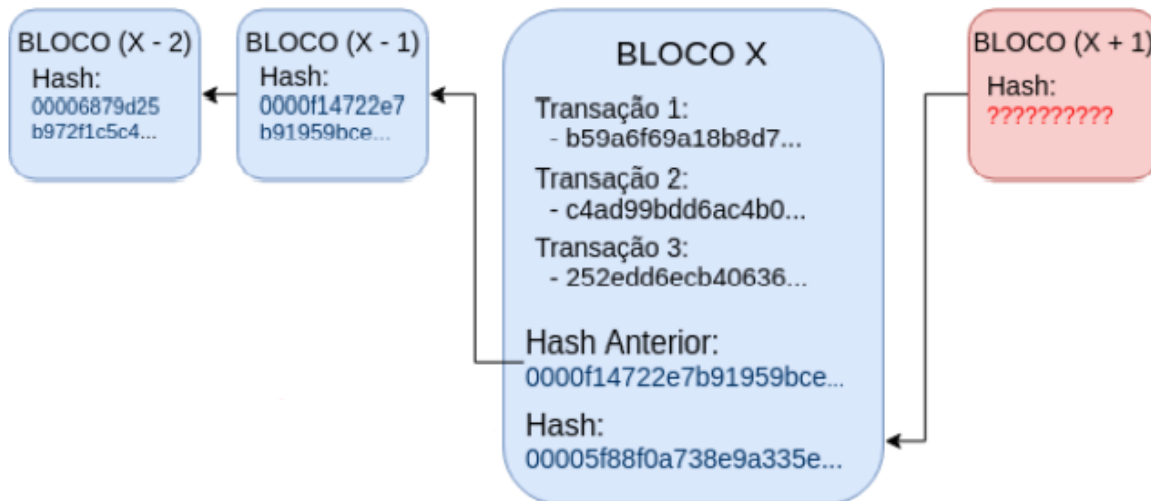


Figura 2.2: Blocos conectados através de uma chave criptográfica (adaptado de Horiuchi [24]).

Sendo assim, os blockchains têm atraído a atenção de diversas indústrias, incluindo a alimentícia (aplicado aos sistemas de rastreabilidade). O motivo de tal interesse está atrelado ao emprego de uma estrutura baseada em uma blockchain, de modo que os aplicativos que anteriormente eram executados apenas por meio de uma entidade central passam a operar de forma descentralizada, o que permite a obtenção da mesma funcionalidade com uma melhor confiabilidade [26].

Assim, ao atuar de maneira descentralizada, todas as partes envolvidas podem ter uma maior visibilidade e transparência em relação aos processos e transações que ocorrem em rede [27]. Além disso, impede-se as possíveis falhas no aplicativo relacionadas ao ponto central, uma vez que esta pode ser operado por qualquer um dos participantes. O blockchain, portanto, permite a rede realizar transações confiáveis e seguras, a partir das quais os participantes não precisem de nenhum nível de confiança entre si [24].

Diante desse cenário, o uso contínuo da blockchain em diversos setores, como é o caso do alimentício, implica o uso de mecanismos específicos, como voltados ao consenso, visto

que fomentam uma maior credibilidade por detrás de todas as interações feitas em rede, tornando as transações, ao mesmo tempo, mais seguras e, sobretudo, transparentes [18].

Os envolvidos em uma cadeia de alimentos – produtores, transportadores, armazenistas, transformadores, retalhistas e outros – têm como objetivo a demonstração, ao consumidor, da superioridade de seus métodos e produtos [28]. O blockchain facilita a demonstração, pois unifica todas as partes e, assim, fornece uma maior integração de dados entre os participantes [26]. Esta tecnologia facilita a estrutura de dados, uma vez que permite a descrição dos atributos provenientes do produto.

Os principais usos de blockchain em uma cadeia de alimentos estão relacionados a algumas características elementares. A primeira delas é a transparência, cujo objetivo é o de facilitar a troca de informação, criar cópias digitais de toda a informação e validar a qualidade do produto em torno desta cadeia de alimentos [29]. O objetivo é cumprido à medida em que cada participante pode compartilhar seus dados sobre as condições atuais do produto, ajudando, então, cada participante a identificar as informações sobre a vida útil do mesmo.

O segundo uso está associada à segurança. O blockchain é utilizado para a criação de blocos únicos e criptografados e, dessa forma, tratando-se de uma estrutura pública, bem como descentralizada, uma das vantagens é o ganho de uma notória segurança adicional [30]. Assim sendo, quando os registros são efetuados nos blocos interligados, não é possível modificar um de seus blocos de maneira localizada e individual [21]. Dessa forma, para que a tarefa possa ser bem executada, seria necessária a adesão a um poder computacional capaz de quebrar a criptografia, isto é, modificar o conteúdo dos blocos e a devida validação para que, então, a cadeia de blocos continuasse válida [24]. Nesse sentido, cada vez mais, esta tecnologia tem sido empregada nos mais diversos setores, pois garante-se transparência e segurança [23].

Com base nesses conceitos, muitos sistemas de rastreabilidade foram desenvolvidos com a tecnologia Blockchain. Essa tecnologia está se tornando um excelente meio de evitar fraudes, e contribuir com a segurança alimentar de um país.

Em [30], é proposto um sistema de rastreabilidade baseado em Blockchain para a

CVAA. O sistema AgriBlockIoT, como é nomeado, permite a integração de IoT com Blockchain e é totalmente descentralizado, garantindo transparência e imutabilidade dos registros.

Existem diversas plataformas que fornecem serviços com a tecnologia blockchain como a IBM Blockchain, Amazon Quantum Ledger Database (QLDB), Everledger, TagOne e outros. As CVAA que utilizam a tecnologia blockchain podem aumentar a velocidade, reduzir erros e aumentar a transparência [31]. Desse modo, a tecnologia Blockchain tem sido uma solução viável para as empresas.

2.3 Código de barras

O cliente final, hoje em dia, como foi dito anteriormente, tem se tornado cada vez mais exigente quanto à informação sobre o produto que deseja adquirir, visto que está mais consciente quanto ao fato de que a sua boa saúde está relacionada com a necessidade de uma alimentação adequada e segura [32]. A mudança é um desafio para os envolvidos da CVAA que começaram a ter uma maior preocupação relativa à qualidade da informação, pois esta é um recurso significativo para o cliente [33], bem como para qualquer uma das entidades que integram a CVAA. Há a necessidade de estabelecer relações de interdependência entre todas as entidades da CVAA para que exista uma passagem do fluxo de informação ao longo da CVAA, isto é, para que chegue até o cliente final [32]. Neste sentido, tem-se adotado tecnologias que garantem a passagem do fluxo de informação, como o código de barras.

O código de barras é compreendido como uma espécie de bilhete que aponta para uma identidade única e inequívoca acerca dos produtos, o que facilita o fluxo de informação entre as entidades da CVAA e permite a interoperabilidade ao longo desta cadeia [33]. Com o intuito de otimizar os fluxos e uma armazenagem eficiente, torna-se importante analisar a atividade de etiquetagem, uma vez que acrescenta valor à CVAA [34]. A etiquetagem é processo a partir do qual se identifica um item e, assim, as etiquetas são inseridas nos produtos e possibilitam o correto transporte dentro e fora do armazém, tanto

no nível da armazenagem quanto na expedição ou transporte. A etiquetagem, ao longo de toda a cadeia de abastecimento, tem como objetivo a garantia da rastreabilidade dos produtos [34].

Para isso, o fluxo físico dos bens deve estar ligado ao fluxo eletrônico da informação. Quando um produto é vendido e sai do estoque, essa informação deve ser atualizada de maneira constante [32]. Surge, nesse contexto, a Global Standard One (GS1), uma organização internacional, sem fins lucrativos, que tem o intuito de auxiliar as empresas – pequenas, médias ou grandes – a implementar normas e/ou standards globais com o objetivo de reduzir as ineficiências ao longo das cadeias de valor, propondo melhorias que permitam a integração dos fluxos de informação e material [33].

A GS1, portanto, defende a incorporação de um conjunto de normas globais, reconhecidas, dessa forma, internacionalmente. É uma forma de gerir de maneira eficiente as cadeias de valor, o que fomenta benefícios voltados à eficiência na gestão de estoques por meio da criação de um fluxo de informação, de modo que do produtor ao retalhista pode-se obter melhorias no controle e visibilidade da cadeia de valor [32]. Atualmente, os standards propostos pela GS1 são empregados em setores como a saúde, transportes e logística, aeronáutica, defesa, construção e no ramo alimentício. O processo de etiquetagem é iniciado com a identificação do item após a produção, o que demanda a atribuição de um código único e inequívoco, sendo este reconhecido, posteriormente, em toda a cadeia, de modo que pode ser captado em qualquer ponto da cadeia [32]. Na figura 2.3 encontra-se representado um código de barras com as normas GS1.

Nesse sentido, a organização dos fluxos de informação ao longo das cadeias e redes depende da aceitabilidade de padrões de comunicação organizacionais em todo o mundo, como é o caso da GS1. Os códigos de barras, portanto, contemplam a tecnologia de etiquetagem mais empregada a nível mundial [35]. Esta tecnologia pode ser adaptada de maneira específica a qualquer setor. São etiquetas lineares construídas a partir de barras verticais, pretas e brancas [35]. Essas permitem a captura da informação por meio da aproximação de um leitor scanner. As barras são lidas por meio de um computador, porém, acompanham, sempre, uma sequência de números, sendo esses visíveis a olho nu.

Em relação à simbologia dos códigos de barras, cada dígito é convertido em números binários (bites 0 e 1), sendo essas reconhecidas por computadores. As barras claras correspondem ao bite 0 e as escuras, por sua vez, ao bite 1, sendo essas representações gráficas de tais combinações [32].



Figura 2.3: Exemplo de um código de barras GS1-128.

Menciona-se que esta tecnologia é amplamente empregada no âmbito do controle de processos, movimentação e armazenamento [36]. Diante desse cenário, cabe listar algumas vantagens e desvantagens associadas a esta tecnologia. Dentre as vantagens associadas aos códigos de barras, há a garantia da rastreabilidade, retorno do investimento e redução de custos [36]. Facilita-se, portanto, toda a gestão, fomentando-se a eficiência nas operações, porém o processo não seria possível sem a existência de standards de abrangência global, pois esses possibilitam e facilitam a comunicação e a troca de informação e, como mencionado, os consumidores finais, cada vez mais, têm exigido maior clareza, precisão e transparência quanto às informações sobre os produtos consumidos [33].

Uma das desvantagens em relação ao código de barras, é que ele é decodificado pelos scanners ou leitores ópticos através da reflexão e absorção da luz. Por isso em condições ambientais difíceis, algumas vezes, dificulta a captura da informação contida nos códigos de barras [35].

Em [16], como foi dito anteriormente, os atores propõem um sistema de rastreabilidade com a tecnologia de identificação por radiofrequência. A tecnologia de identificação por radiofrequência, por exemplo, não sofre interferência em ambientes com pouca ou nenhuma luz, já que são identificados por radiofrequência e não por leitores ópticos, porém essa tecnologia possui um elevado custo, e por isso não tem sido adotada com tanta

frequência, como o código de barras [14].

Capítulo 3

Levantamento e análise de requisitos

Neste capítulo é apresentado uma descrição detalhada do problema, caracterizando a metodologia utilizada durante o desenvolvimento do trabalho. Por fim, é apresentado a abordagem proposta em conjunto com os diagramas Unified Modeling Language (UML), com o objetivo de indicar explicitamente as tarefas a desempenhar pelo sistema, e os agentes que interagem com o mesmo.

3.1 Problema

Como foi explicado anteriormente, uma das principais dificuldades da CVAA, hoje em dia, é a incapacidade de vincular registos entre as fases ao longo da cadeia, devido ao facto de serem realizadas/ registadas manualmente. Essa dificuldade penaliza uma atuação rápida por exemplo em situações de fraude. Outro desafio identificado foi implementar uma forma para que evitem alterações indesejadas desses registos, garantindo que não ocorra fraudes alimentares que possam prejudicar a segurança alimentar do país. O desenvolvimento de uma solução capaz de resolver os desafios identificados, exigiram conhecimento em diversas áreas, desde Blockchain e gestão de informação até o desenvolvimento de um produto digital em formato de sistema web que englobasse todas as fases da CVAA garantindo a vinculação dos registos e a segurança dos dados registados.

Assim o presente trabalho tem por objetivo solucionar a integração das fases da CVAA

e garantir a integridade dos dados registados, propondo uma solução web compatível com celular e desktop.

Para alcançar o objetivo proposto foi utilizado um modelo do ciclo de vida de desenvolvimento de software, para que o desenvolvimento do sistema fosse feito de forma organizada e gradual. O modelo em questão é o modelo incremental, que como o nome propõe, divide o desenvolvimento em incrementos que produzem o sistema de forma a evoluir gradualmente até a sua versão final. O modelo é composto por diversas fases que seguem uma ordem cronológica, sendo descritas na figura 3.1 [37].

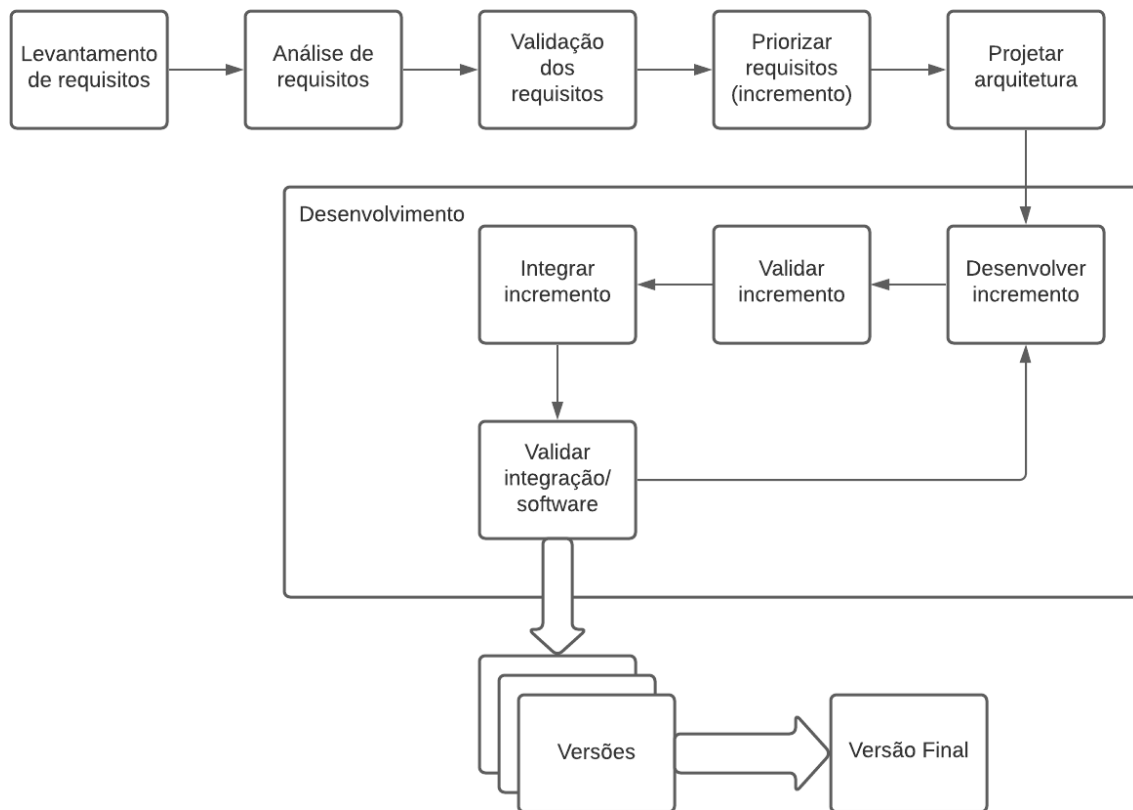


Figura 3.1: Modelo do Ciclo de Vida de Desenvolvimento de Software.

Levantamento de requisitos é a primeira etapa no ciclo de desenvolvimento de software, onde são definidas as funcionalidades e o escopo do projeto. As dificuldades dessa etapa são conseguir enxergar todos os cenários possíveis e interpretar os cenários da forma

correta, de forma que evite erros durante a criação das funcionalidades [37].

Na análise dos requisitos é feito um estudo dos dados levantados na etapa anterior, de onde são construídos modelos com a finalidade de representar o sistema de software a ser desenvolvido. Para a construção dos modelos do sistema foi utilizado a UML uma linguagem de modelação que permite apresentar o sistema de forma padronizada.

A validação dos requisitos, é a etapa, tal como o nome indica, onde é feita a validação quanto à consistência, precisão, contextualização e viabilidade dos requisitos levantados. Este processo envolve a revisão dos requisitos e a validação dos diagramas UML.

Feita a validação dos requisitos, começa a etapa para priorizar os requisitos, esta etapa ajuda a direccionar os nossos recursos da forma correta, garantindo que as funcionalidades essenciais do software sejam implementadas.

Projectar a arquitectura de um software é uma das etapas mais difíceis do ciclo de vida do desenvolvimento de um software. Questões arquitecturais englobam organização e estrutura geral de controle, protocolos de comunicação, sincronização, alocação de funcionalidade a componentes e selecção de alternativas de projecto. Todos esses factores compreendem o projecto no nível arquitectural e estão directamente relacionados com a organização do sistema e, portanto, afectam os atributos de qualidade como desempenho, portabilidade, confiabilidade, disponibilidade, entre outros [37].

Após a arquitetura do software ser projetada, começa o processo de desenvolvimento das funcionalidades, este processo pode ser repetido diversas vezes, gerando sempre uma versão funcional.

Para a etapa de desenvolvimento foi utilizado uma metodologia ágil, denominada Scrum, na expectativa de uma melhoria na gestão e no planeamento das funcionalidades. Apesar desta metodologia estar orientada para a gestão de equipas, foi utilizado alguns dos princípios e conceitos mais importantes que estão contidos no Scrum. Um dos conceitos utilizados, foi o conceito de product backlog, para ter um desenvolvimento iterativo organizados em sprints curtos, focados na entrega de uma versão funcional no final de cada sprint [37]. Também foi feita reuniões semanais com o professor para a recolha de feedbacks, com o objetivo de encontrar oportunidades de melhorias e pontos de mudanças

necessárias para melhorar o enquadramento da solução.

Para o acompanhamento das funcionalidades foi utilizado o quadro kanban, uma ferramenta que apoia o Scrum garantindo a transparência e inspeção. O kanban é, normalmente, preenchido por post-its, onde em cada um deles possui uma funcionalidade a ser desenvolvido. Cada projeto pode e deve adaptar o seu quadro conforme as necessidades, no entanto é geralmente constituído pelas colunas, Product Backlog, “A fazer”, “Em execução”, “Em teste” e “Concluídas”. Os posts-its navegam entre as colunas, de forma a indicar o estado atual da funcionalidade, até chegar à última, indicando que a funcionalidade foi desenvolvida [37], [38].

3.2 Abordagem proposta

Esta secção vai percorrer todas as etapas que antecedem o desenvolvimento, ou seja, levantamento de requisitos, análise dos requisitos, validação dos requisitos, priorização dos requisitos e a definição da arquitectura do software.

Para realizar todas essas etapas com mais qualidade, foi necessário realizar um enquadramento da proposta para conseguirmos perceber algumas questões sobre o software que iríamos desenvolver. Depois foi necessário identificar os perfis de utilizadores e o seu papel dentro da plataforma, seguidamente de uma visão geral dos requisitos funcionais e dos requisitos não funcionais e finalizando com os modelos UML e diagramas, de forma a especificar a arquitectura do software e a interação dos atores com o mesmo.

3.2.1 Enquadramento da proposta

De modo a tentar perceber melhor a respostas ao problema em questão foi feita duas checklists com as algumas questões relacionadas ao problema, uma com questões gerais e outra com questões específicas. As perguntas tinha como objectivo determinar o que seria feito e de que modo seria feito. Dando uma estrutura inicial para a etapa de levantamento de requisitos.

3.2.1.1 Perguntas gerais

As perguntas gerais definidas tem objetivo de perceber questões sobre quem vai utilizar o software e como, no total foram determinadas 3 perguntas que são:

- Quem vão ser os stakeholders do software?
- Quais os perfis e necessidades dos stakeholders?
- Como o software vai ser disponibilizado?

A seguir é apresentado a resposta dada a cada pergunta levantada. Respostas que direccionaram todas as etapas que antecedem o desenvolvimento.

Quem vão ser os stakeholders do software? O software será utilizado pelos funcionários das empresas que participam de alguma forma da CVAA. As empresas que utilizarão o software, terão que ter uma das cinco funções identificadas, sendo elas, produzir, transportar, transformar, armazenar ou vender. As empresas ficam responsáveis por registar todos os seus funcionários e determinar os níveis de acesso de cada um. Além das empresas, o software também será utilizado pelo cliente final que poderá consultar a rastreabilidade, sem precisar de realizar um cadastro prévio.

Quais os perfis e necessidades dos stakeholders? Existem 5 tipos empresas determinadas na CVAA e cada tipo de empresa possui suas necessidades, além de que cada funcionário da empresa pode ter um nível diferente de acesso. Para este projeto foi determinado 3 níveis de acessos, sendo eles, o funcionário que pode somente visualizar, entregar e receber algum lote, o supervisor que pode fazer tudo que o funcionário faz e gerenciar informações do lote e o gestor que pode fazer tudo que o supervisor faz, mas no domínio da empresa.

Como a plataforma vai ser disponibilizada? Para que a plataforma seja acessível e viável, ela será disponibilizada através de um site responsivo, podendo ser utilizada tanto

no celular quanto no computador. O uso no celular será mais voltado para ler os código de barras bidimensionais e o uso no computador será mais direccionado para gestão das informações, porém ambas funcionalidades poderão ser executadas em qualquer ambiente.

3.2.1.2 Perguntas específicas

As perguntas específicas definidas tem objetivo de perceber questões acerca da rastreabilidade e da solução proposta, no total foram determinadas 4 perguntas que são:

- Quais dados são essenciais estarem presentes quando um consumidor consultar a rastreabilidade de um produto?
- Quais dados são importantes estarem presentes quando um consumidor consultar a rastreabilidade de um produto?
- Como será feito para garantir a conexão entre as diversas fases da rastreabilidade?
- Como o sistema vai impedir alterações indesejadas?

A seguir é apresentada a resposta dada a cada pergunta específica levantada.

Quais dados são essenciais estarem presentes quando um consumidor consultar a rastreabilidade de um produto? Para uma rastreabilidade adequada de alimentos, baseado nas informações do site da GS1 [39], foi determinado que o sistema deveria apresentar dados como:

- Código do lote
- Data de produção
- Data de embalagem
- Data de validade
- Quem é o produtor

Quais dados são importantes estarem presentes quando um consumidor consultar a rastreabilidade de um produto? Para uma rastreabilidade adequada de alimentos, baseado nas informações do site da GS1 [39], foi determinado que no sistema seria importante apresentar dados como:

- Descrição da produção
- Tamanho da embalagem
- Descrição da transformação
- Certificados
- Local da produção
- Quem é o transportador?
- Quem é o armazenador?
- Quem é o transformador?
- Quem é o retalhista?
- Qual foi o horário de entrega?
- Qual a temperatura que o produto foi transportado?

Como será feito para garantir a conexão entre as diversas fases da rastreabilidade? Para garantir a conexão entre as diversas fases, para cada lote registrado, será gerado um código de barras bidimensional (QRCode), que deverá ser lido, através da plataforma, quando o lote for entregue e recebido. Essa conexão entre os fluxos manuais e digitais é essencial para o sistema conseguir rastrear toda a trajetória do lote e identificar os agentes que participaram da cadeia e permiti-los gerenciar dados do lote em questão.

Como o sistema vai impedir alterações indesejadas? Para impedir alterações nas informações do lote será desenvolvido um histórico de modificações verificável através de criptografia, baseado nos conceitos de blockchain privada.

3.2.2 Perfis de utilizadores

Baseado nas respostas das questões anteriores foi determinado diversos perfis de utilizadores, a seguir, na tabela 3.1, é apresentado a lista com todos os perfis de utilizadores identificados, com o seu nível de acesso ao sistema e uma breve descrição da sua função.

Perfil	Nível de acesso	Descrição
Consumidor	Utilizador não registado	Consultar toda a cadeia de valor agroalimentar.
Produtor	Utilizador registado	Criar produções e lotes, adicionar informações relevantes na produção e no lote e enviar o lote.
Transportador	Utilizador registado	Receber e enviar o lote. Sincronizar dados de sensores.
Transformador	Utilizador registado	Receber, adicionar informações sobre a transformação e enviar o lote. Enviar e criar sublotes a partir de lotes recebidos.
Armazenista	Utilizador registado	Receber e enviar o lote. Sincronizar dados dos sensores.
Retalhista	Utilizador registado	Receber lote e adicionar feedback sobre o lote recebido.
Gestor de qualidade	Utilizador registado	Acompanhar toda a cadeia de valor agroalimentar

Tabela 3.1: Lista de perfis de utilizadores

Para uma descrição mais detalhada, nas próximas secções também é apresentado uma visão geral de cada perfil, descrevendo exatamente o seu papel no sistema. Indicando o que cada perfil deve fazer para que o sistema Biotrace cumpra os seus objetivos.

3.2.2.1 Produtor

O Produtor tem o papel de criar produções, dando início a várias cadeias de valor agroalimentar, além disso ele deve documentar a produção da melhor forma para o consumidor. Algumas das informações importantes a serem documentadas são: local da produção, data de início da produção, data final da produção, tipo de produto, qualidade do produto, etc. Assim que a produção é finalizada, o produtor pode dividi-la em lotes e encaminha-las ao transportador. Para encaminhar uma produção é necessário ler o código QR de cada lote, para que fiquem registadas as informações como a data e a hora que o produto foi despachado pelo produtor.

3.2.2.2 Transportador

O Transportador tem o papel de transporte dos produtos entre o produtor e o transformador, armazenista ou retalhista, para isso ele precisa ler o código QR de cada lote transportado, tanto na entrada quanto na saída do lote, assim a plataforma consegue registar o momento em que o lote ficou em transito. Além disso o Transportador pode ter sensores de temperatura e humidade, que registam a cada minuto a temperatura e humidade local, esses dados podem ser sincronizados com os lotes que estavam em transito no momento.

3.2.2.3 Transformador

O transformador tem o papel de documentar toda a transformação do produto. Assim que o transformador receber o lote, ele precisa ler o código QR do lote para a plataforma registar seu recebimento, feito isso ele pode documentar toda a transformação que foi feita no lote, registando dados como: quantidade recebida, quantidade transformada, data da transformação, entre outros. Logo que o lote estiver pronto, o transformador precisa ler o código QR do lote novamente para que a plataforma registre seu envio e dê sequência na rastreabilidade do lote.

3.2.2.4 Armazenista

O armazenista tem o papel de armazenar os produtos para posterior distribuição, para isso ele precisa ler o código QR de cada lote armazenado, tanto na entrada quanto na saída do lote, assim a plataforma consegue registrar o momento em que o lote ficou armazenado. Além disso o armazenista pode ter sensores de temperatura e humidade, que registam a cada minuto a temperatura e humidade local, esses dados podem ser sincronizados com os lotes que estavam armazenados no momento.

3.2.2.5 Retalhista

O Retalhista tem o papel de vender os produtos, sendo o penúltimo elemento da cadeia de valor, antecedendo somente o consumidor final. Os retalhistas devem realizar as atividades necessárias para a venda do produto diretamente aos consumidores finais, que os utilizam para fins estritamente pessoais. Contudo é de extrema importância que assim que o retalhista receber o lote do produto, ler o código QR para a plataforma e disponibilizar a rastreabilidade do lote para o consumidor final.

3.2.2.6 Gestor de qualidade

O Gestor de qualidade tem o papel de acompanhar as etapas de produção e o resultado final dos produtos da empresa em que atua, visando, assim, garantir que estes estão de acordo com o que é previsto pelas normas e diretrizes gerais, tal como as da própria empresa. Após a tabela é descrito, com mais detalhes, o papel de cada agente no sistema.

3.2.2.7 Consumidor

O Consumidor, após a compra de qualquer produto Biotrace, tem a opção de consultar toda a cadeia de valor alimentar, através de um código QR. Podendo visualizar muitas informações sobre o produto em questão, como quem foi o produtor, quem foi o transportador, quando foi produzido, onde foi produzido, em qual temperatura o produto foi transportado entre outras. Essas informações acrescentam muito valor ao produto, pois

o consumidor se sente mais seguro e confiante em relação ao produto que comprou.

3.2.3 Atores

Os atores, de forma simplista, são os utilizadores da plataforma. Eles podem ser tanto humanos quanto uma entidade externa que interage com o sistema para executar uma tarefa importante. Na Tabela 3.2 é apresentado os atores da plataforma Biotrace que foram identificados a partir dos perfis de utilizadores.

Ator	Descrição
Não registado/ Consumidor	Este ator representa todos os utilizadores que não foram registados ou identificados no sistema e o consumidor que não precisa ser registado para visualizar a rastreabilidade
Utilizador registado e autenticado	Este ator representa todos os utilizadores registado e autenticado pelo sistema. Dentro deste de grupo esta incluído, o administrador, os produtores, os transportadores, os transformadores, os armazenistas, os retalhistas, os gestores de qualidade e os consumidores registados.
Administrador	Este ator representa o administrador da plataforma Biotrace, sendo responsável por administrar, gerir e monitorizar a plataforma Biotrace.
Produtor	Este ator representa todos os utilizadores que estão registados em uma empresa de produção.
Produtor Gestor	Este ator representa todos os utilizadores que estão registados em uma empresa de produção como gestor.
Produtor Supervisor	Este ator representa todos os utilizadores que estão registados em uma empresa de produção como supervisor.
Continua na próxima página	

Tabela 3.2 – Continuação da página anterior

Ator	Descrição
Produtor Funcionário	Este ator representa todos os utilizadores que estão registados em uma empresa de produção como funcionário.
Transportador	Este ator representa todos os utilizadores que estão registados em uma empresa de transporte.
Transportador Gestor	Este ator representa todos os utilizadores que estão registados em uma empresa de transporte como gestor.
Transportador Supervisor	Este ator representa todos os utilizadores que estão registados em uma empresa de transporte como supervisor.
Transportador Funcionário	Este ator representa todos os utilizadores que estão registados em uma empresa de transporte como funcionário.
Transformador	Este ator representa todos os utilizadores que estão registados em uma empresa de transformação.
Transformador Gestor	Este ator representa todos os utilizadores que estão registados em uma empresa de transformação como gestor.
Transformador Supervisor	Este ator representa todos os utilizadores que estão registados em uma empresa de transformação como supervisor.
Transformador Funcionário	Este ator representa todos os utilizadores que estão registados em uma empresa de transformação como funcionário.
Armazenista	Este ator representa todos os utilizadores que estão registados em uma empresa de armazenamento.
Continua na próxima página	

Tabela 3.2 – Continuação da página anterior

Ator	Descrição
Armazenista Gestor	Este ator representa todos os utilizadores que estão registados em uma empresa de armazenamento como gestor.
Armazenista Supervisor	Este ator representa todos os utilizadores que estão registados em uma empresa de armazenamento como supervisor.
Armazenista Funcionário	Este ator representa todos os utilizadores que estão registados em uma empresa de armazenamento como funcionário.
Retalhista	Este ator representa todos os utilizadores que estão registados em uma empresa de retalho.
Retalhista Gestor	Este ator representa todos os utilizadores que estão registados em uma empresa de retalho como gestor.
Retalhista Supervisor	Este ator representa todos os utilizadores que estão registados em uma empresa de retalho como supervisor.
Retalhista Funcionário	Este ator representa todos os utilizadores que estão registados em uma empresa de retalho como funcionário.
Gestor de qualidade	Este ator representa todos os utilizadores que estão registados em uma empresa de produção como gestor de qualidade.
Sensor	Este ator representa todos os sensores que estão registados em uma empresa de transporte ou de armazenamento.

Tabela 3.2: Lista de atores da plataforma Biotrace.

3.2.4 Requisitos Funcionais

Os requisitos funcionais do sistema representam o que o software faz em termos de tarefas e serviços, basicamente, são funcionalidades constituídas a partir do levantamento de requisitos, e precisam conter pelo menos uma descrição e origem [40]. Para este trabalho foi decidido que a organização dos requisitos funcionais seria feita através de User Stories, um artefato de desenvolvimento utilizado em sistemas geridos com uma metodologia ágil. Em resumo, User Stories são descrições simples de uma funcionalidade a partir do ponto de vista do utilizador [37].

A seguir são apresentadas as principais User Stories, com o objetivo de especificar o que alguns atores pretendem fazer na plataforma Biotrace. Para cada ator foi feito uma tabela com suas stories, cada story possui um identificador, nome, prioridade e descrição se encontra no anexo A. Na tabela 3.3 são apresentadas as user stories referentes aos atores não registados na plataforma.

ID	Nome	Prioridade(1-5)	Descrição
USUNR01	Solicitar acesso para uma nova empresa	1	Pretendo solicitar acesso à plataforma para uma nova empresa
USUNR02	Ler QR Code de um alimento	1	Pretendo ler QR Code de um alimento e poder ver as suas informações de rastreabilidade

Tabela 3.3: User Stories (Não registado/ Consumidor)

Na tabela 3.4 são apresentadas as user stories referente aos atores registado na plataforma.

ID	Nome	Prioridade(1-5)	Descrição
USUR01	Iniciar sessão	1	Pretendo iniciar sessão na plataforma
USUR02	Encerrar sessão	1	Pretendo encerrar sessão na plataforma
USUR03	Recuperar palavra-passe	4	Pretendo recuperar a minha palavra-passe
USUR04	Atualizar perfil	3	Pretendo atualizar meu perfil

Tabela 3.4: User Stories (Utilizadores registados)

Na tabela 3.5 são apresentadas as user stories referente ao administrador da plataforma Biotrace.

ID	Nome	Prioridade(1-5)	Descrição
USADMB01	Gerir empresas	2	Pretendo adicionar, visualizar, editar e remover empresas
USADMB02	Gerir produtos	2	Pretendo adicionar, visualizar, editar e remover produtos
USADMB03	Gerir utilizadores	2	Pretendo adicionar, visualizar,editar e remover utilizadores

Tabela 3.5: User Stories (Administrador)

Na tabela 3.6 são apresentadas as user stories referente ao produtor gestor da plataforma Biotrace.

ID	Nome	Prioridade(1-5)	Descrição
USPG01	Gerir produções	1	Pretendo adicionar, visualizar, remover e editar produções
USPG02	Gerir lotes	1	Pretendo adicionar, visualizar, remover e editar lotes
USPG03	Visualizar sub lotes	3	Pretendo visualizar os sub lotes
USPG04	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USPG05	Ler QR Code de um lote	2	Pretendo ler um QR Code e registar que enviei o lote
USPG06	Gerir supervisores	4	Pretendo adicionar, visualizar, remover e editar supervisores
USPG07	Gerir funcionários	4	Pretendo adicionar, visualizar, remover e editar funcionários

Tabela 3.6: User Stories (Produtor Gestor)

Na tabela 3.7 são apresentadas as user stories referente ao transportador gestor da plataforma Biotrace.

ID	Nome	Prioridade(1-5)	Descrição
USTG01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa participou
Continua na próxima página			

Tabela 3.7 – Continuação da página anterior

ID	Nome	Prioridade(1-5)	Descrição
USTG02	Visualizar sub lotes	3	Pretendo visualizar os sub lotes que minha empresa participou
USTG03	Sincronizar dados dos sensores	3	Pretendo sincronizar os dados dos sensores
USTG04	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USTG05	Ler QR Code de um lote	2	Pretendo ler um QR Code
USTG06	Gerir supervisores	4	Pretendo adicionar, visualizar, remover e editar supervisores
USTG07	Gerir funcionários	4	Pretendo adicionar, visualizar, remover e editar funcionários
USTG08	Gerir sensores	3	Pretendo adicionar, visualizar, remover e editar sensores

Tabela 3.7: User Stories (Transportador Gestor)

Na tabela 3.8 são apresentadas as user stories referente ao retalhista gestor da plataforma Biotrace.

ID	Nome	Prioridade(1-5)	Descrição
USRG01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa recebeu
USRG02	Visualizar sub lotes	3	Pretendo visualizar os sub lotes que minha empresa recebeu
USRG03	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USRG04	Ler QR Code de um lote	2	Pretendo ler um QR Code
USRG05	Gerir supervisores	4	Pretendo adicionar, visualizar, remover e editar supervisores
USRG06	Gerir funcionários	4	Pretendo adicionar, visualizar, remover e editar funcionários

Tabela 3.8: User Stories (Retalhista Gestor)

3.2.5 Requisitos Não Funcionais

Os requisitos não funcionais são requisitos que estabelecem como o sistema se comportará em determinadas situações [40]. A seguir são apresentados os requisitos funcionais divididos em 6 categorias, sendo elas usabilidade, segurança, performance, interoperabilidade, compatibilidade e padrões.

3.2.5.1 Usabilidade

Os requisitos de usabilidade são voltados para garantir que o sistema seja fácil de aprender e fácil de usar [40]. Para a plataforma contemplar todos os requisitos de usabilidade

levantados, a plataforma deve cumprir com os seguintes requisitos:

- Possuir um design responsivo simples e efetivo;
- Apresentar feedbacks de erros amigáveis.

3.2.5.2 Segurança

Os requisitos de segurança correspondem às definições sobre as regras de segurança para criação de utilizadores, os procedimentos exigidos para a utilização de senhas, a necessidade de criptografia e restantes questões relacionadas para garantir a proteção dos dados [40]. Para alcançar os objectivos de segurança da plataforma foram levantados os seguintes requisitos:

- A plataforma Biotrace deve possuir um Administrador que pode aceitar ou rejeitar acesso de empresas e utilizadores da plataforma;
- Somente Gestores da empresa poderá registar novos utilizadores;
- A senha de todas as contas deve ser gerada aleatoriamente e enviadas para o email registado, sendo assim solicitada a troca da mesma;
- Toda a senha deve ter no mínimo 8 dígitos, uma letra maiúscula, uma minúscula, um número e um caracter especial;
- Para a recuperação da senha deve ser enviado um código por token que permite a alteração da senha;
- A senha deve ser salva com a criptografia bcrypt;
- O MongoDB deve ser configurado com autenticação;
- Todas as portas expostas devem estar com o protocolo Hyper Text Transfer Protocol Secure (HTTPS);
- Os dados da rastreabilidade devem ser imutáveis;

3.2.5.3 Performance

Os requisitos de performance são voltados para garantir que o sistema funcione sem lentidão, sem problemas por falta de espaço em disco ou com outras ocorrências que impactem na qualidade de uso do sistema [40], para alcançar os objectivos de performance da plataforma foram levantados os seguintes requisitos:

- A plataforma Biotrace deve responder a qualquer requisição no intervalo máximo de 3 segundos;
- A plataforma Biotrace deve deletar dados antigos após um determinado tempo;
- A plataforma Biotrace deve ser configurada com nginx para suportar escalabilidade horizontal futuramente;
- A plataforma Biotrace deve rodar pelo menos em uma VM com 4 núcleos e 8 GB de ram.

3.2.5.4 Interoperabilidade

Os requisitos de interoperabilidade são voltados para garantir que dois ou mais sistemas se comuniquem de forma transparente [40], para alcançar os objectivos de interoperabilidade da plataforma foram levantados os seguintes requisitos:

- O frontend deve ser capaz de utilizar qualquer funcionalidade do backend. A integração deve ser pelo protocolo HTTP.
- O backend deve ser capaz de salvar, editar, remover e ler dados da base de dados.
- Para o acesso a dados da rede de sensores será utilizado um Broker, que utilizará o protocolo MQTT, com capacidade de guardar diretamente os dados dos sensores na base de dados, após uma qualificação prévia dos sensores com validação da entidade e do lote.

3.2.5.5 Compatibilidade

Os requisitos de compatibilidade são voltados para garantir as compatibilidades necessárias para a execução do sistema [40], o único requisito de compatibilidade levantado foi que a plataforma Biotrace deverá ser capaz de ser utilizada em qualquer navegador web que suporte PWA.

3.2.5.6 Padrões

Os requisitos de padrões são voltados para garantir que o desenvolvimento e a definição do sistema sejam padronizados [40], para alcançar os objectivos de padrões da plataforma foram levantados os seguintes requisitos:

- Toda a modelagem deve ser feita utilizando diagramas UML.
- Todo o projeto deverá ser feito utilizando o padrão de software Model-View-Controller (MVC), uma arquitetura separada em três camadas, a camada de interface, a camada de negócio e a camada de dados. Cada camada conterá apenas os algoritmos relacionados à sua responsabilidade.
- A camada de interface deverá contemplar lógicas de controle de estado, validação de campos e acionamento de comandos.
- A camada de negócio deverá contemplar as regras de negócio associadas aos requisitos funcionais. Todas as regras de negócio deverão ter testes funcionais.
- A camada de dados deverá abrigar as lógicas de manipulação de dados.

3.2.6 Casos de uso

Os casos de uso fazem parte do contexto da UML, sendo utilizados para demonstrar as diferentes maneiras que os atores podem interagir com o sistema. Portanto, a sua aplicação é importante para a análise comportamental do sistema [37].

A figura 3.2 apresenta o caso de uso geral do sistema, o diagrama apresentado foi dividido em módulos, em que cada módulo são identificado os atores que podem interagir com o mesmo e a forma como podem interagir.

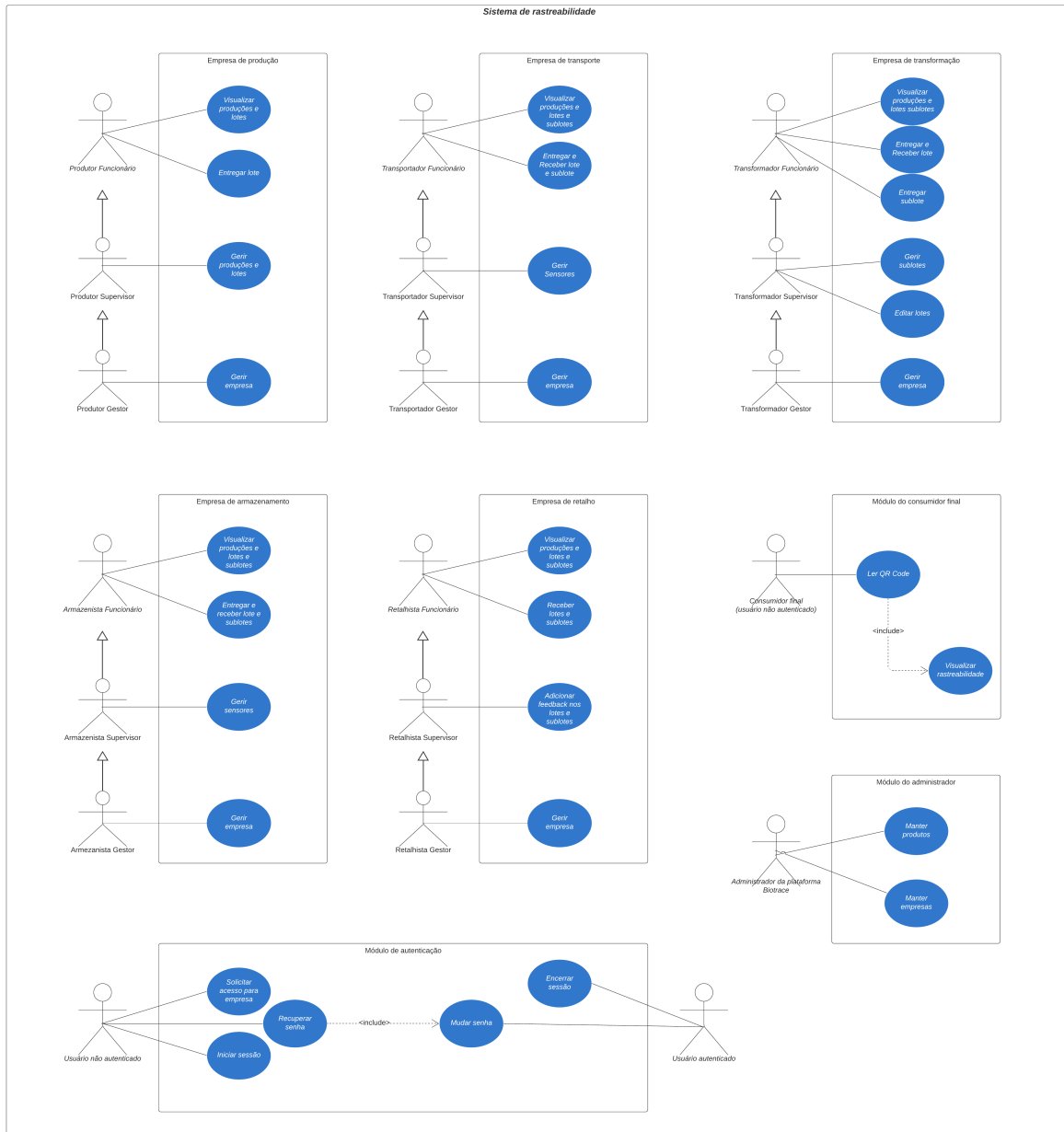


Figura 3.2: Diagrama de casos de uso

3.2.7 Diagramas de atividades

Nesta subsecção serão apresentados os principais diagramas de atividades do sistema.

O diagrama de atividades descreve o fluxo de atividades e ações de um sistema em um determinado cenário, sendo geralmente utilizado para conseguir visualizar o comportamento do sistema, em uma perspectiva orientada a objetos [37], [41].

A figura 3.3 apresenta o diagrama de atividades de quando ocorre a leitura de um QR Code, de um produto Biotrace, por um agente da CVAA, visando identificar o fluxo da atividade e as ações executadas pelo sistema no processo. Este diagrama apresenta o fluxo de quando um agente da CVAA efetua a leitura de um QR Code para receber ou entregar um lote. Se trata de um fluxo complexo e com varias verificações com o objetivo de evitar erros na rastreabilidade de um produto. Por exemplo, se um lote ainda não foi enviado, o sistema não pode permitir que algum ator receba ele, portanto o sistema deve enviar um alerta avisando que não é possível receber aquele lote até que ele seja entregue.

Para que a informação da rastreabilidade chegue ao consumidor final esse fluxo precisa ocorrer pelo menos 4 vezes, podendo ocorrer mais vezes, caso a rastreabilidade seja mais complexa e tenha mais agentes. Por exemplo, se a rastreabilidade incluir todos os tipos de empresas identificadas, esse fluxo vai ocorrer pelo menos 12 vezes, como é apresentado na figura 3.4. Por dentro de cada atividade deste fluxo ocorre o fluxo que foi apresentado anteriormente.

Na figura 3.5, é apresentado como funciona o fluxo de atividades quando o consumidor final solicita verificar a rastreabilidade de um produto. As atividades chave desse diagrama são quando o módulo ledger, que abriga todas as regras de negócio do histórico de alterações criptograficamente verificável, verifica os dados do produto, isso garante que, caso tenha alguma informação modificada de forma indesejada, a rastreabilidade do produto não seja apresentada para o consumidor.

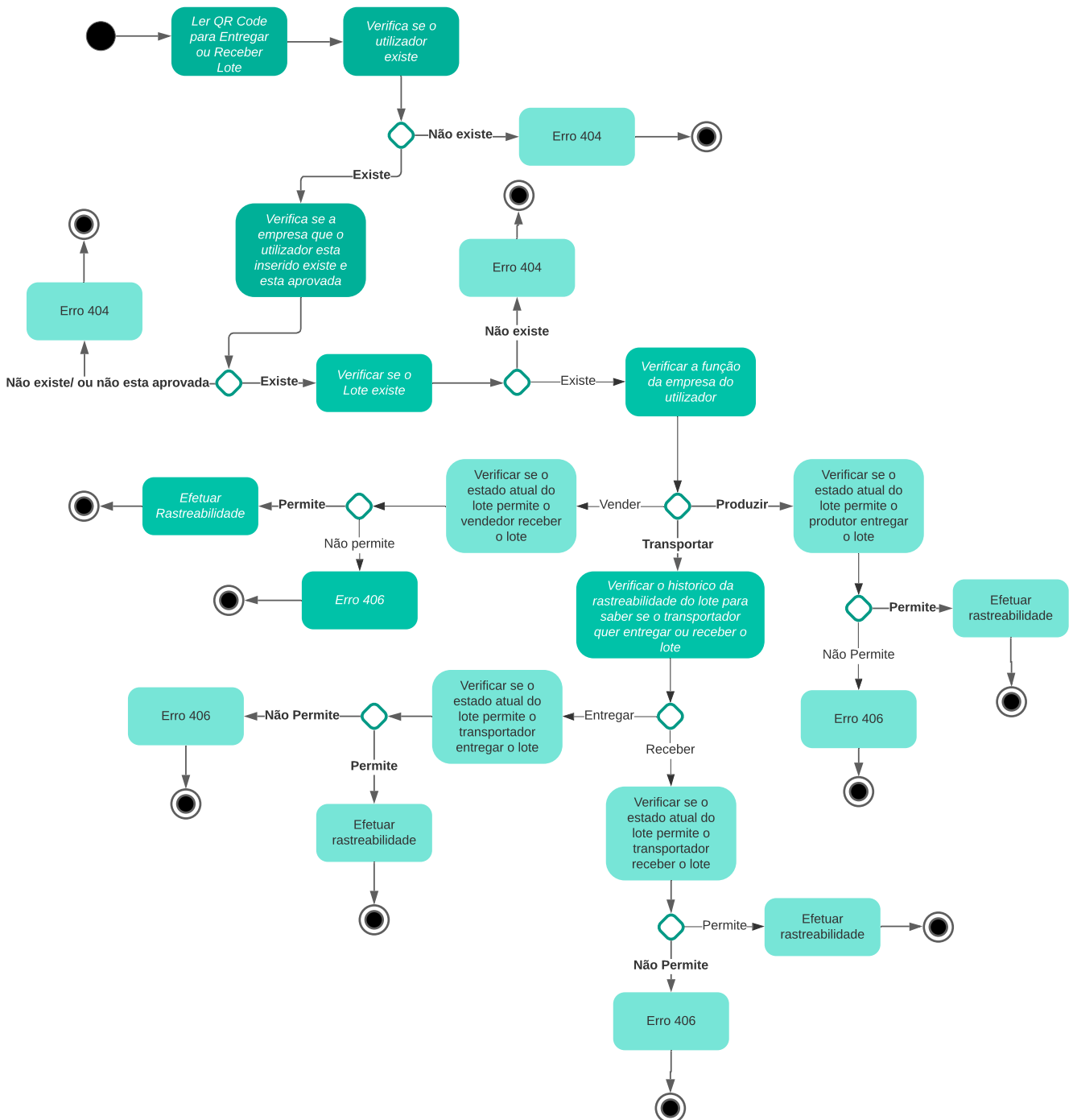


Figura 3.3: Diagrama de atividades da leitura de QR Code.

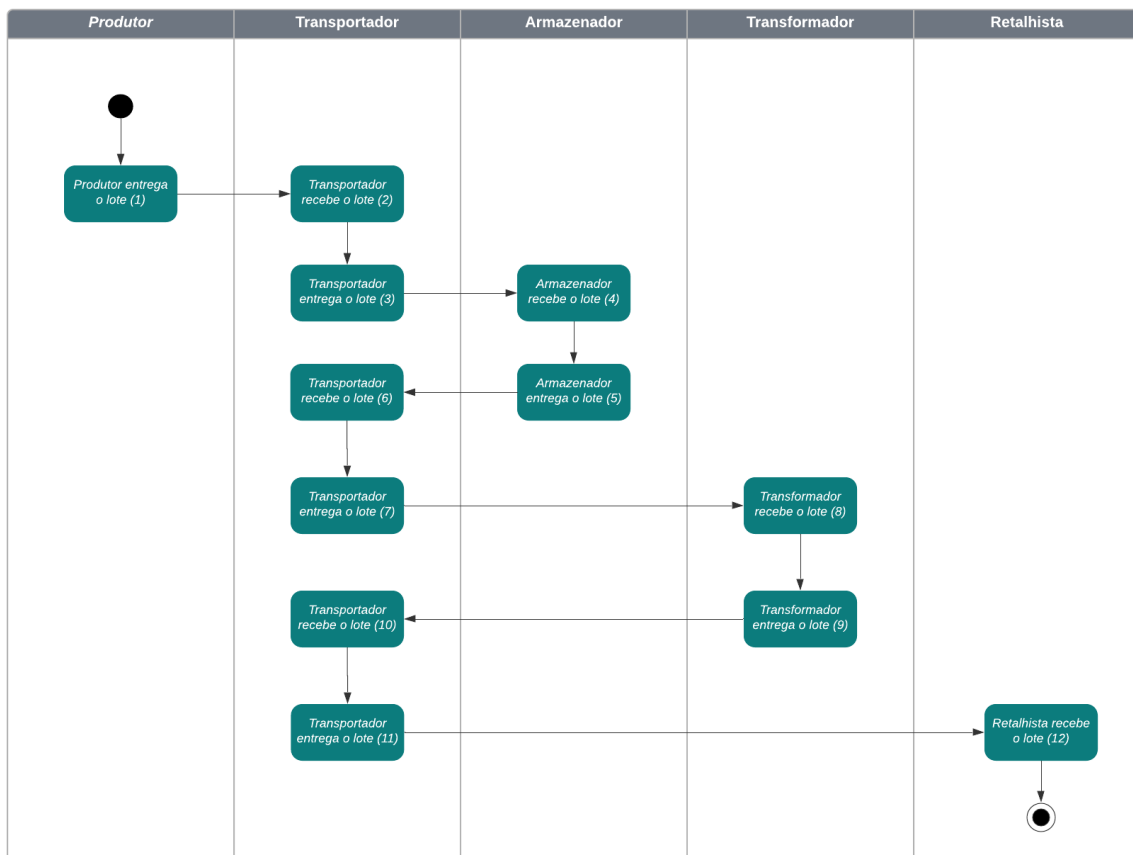


Figura 3.4: Diagrama de atividades da rastreabilidade de um produto.

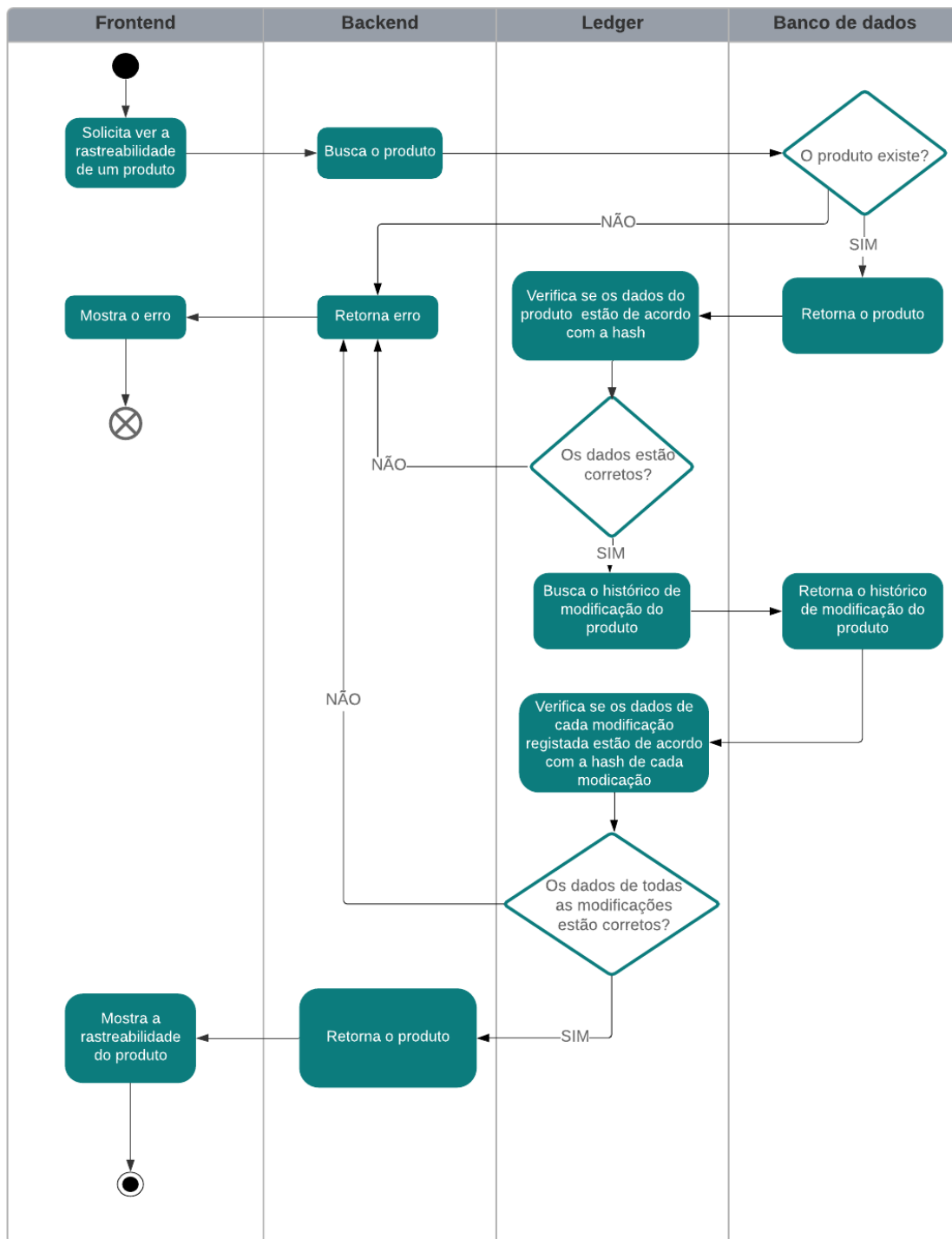


Figura 3.5: Diagrama de atividades para apresentar informações de um produto ao cliente.

3.2.8 Diagrama de classes

O diagrama de classes tem o objetivo de descrever a estrutura do sistema [37], nele é apresentado todas as classes do sistema e seus atributos. Os atributos foram determinados a partir dos requisitos levantados anteriormente. Na figura 3.6, é apresentado o diagrama de classe geral do sistema, como pode ser constatado a estrutura do sistema é composta por um total de 12 classes com diversos atributos e diversas conexões entre elas.

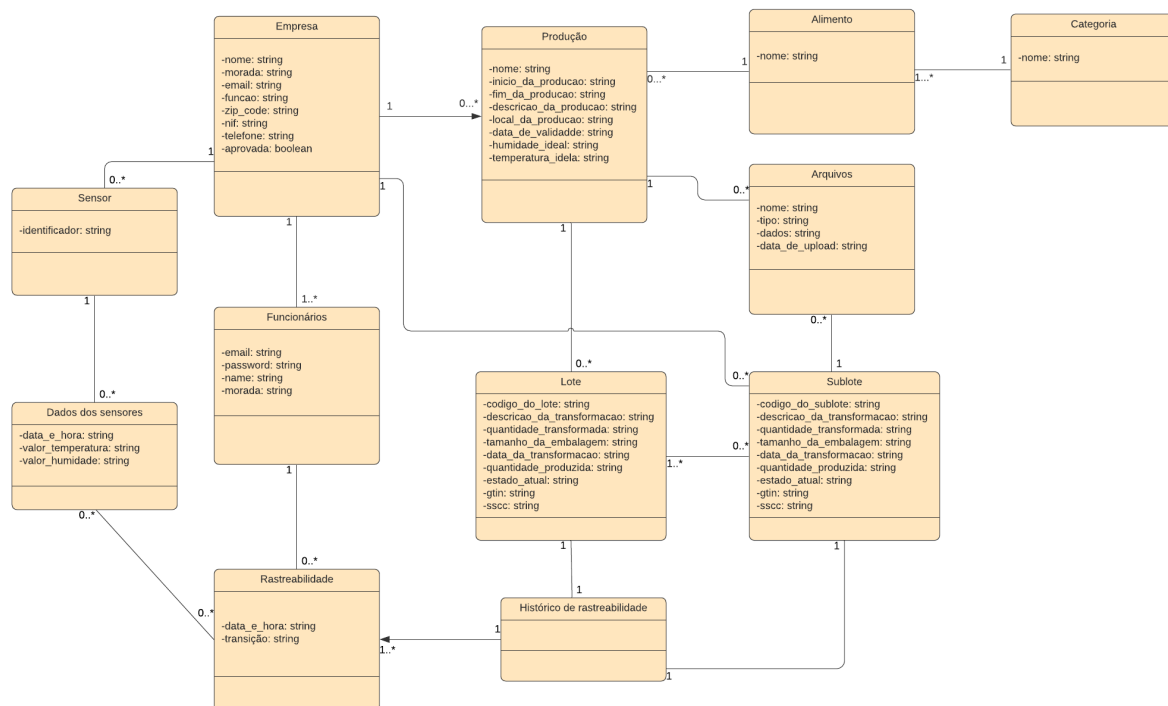


Figura 3.6: Diagrama de classes.

3.2.9 Diagramas de sequência

O diagrama de sequência, como o nome propõe, tem o objectivo de representar a sequência de um processo, ele dá ênfase à ordenação temporal em que as mensagens são trocadas entre os objetos de um sistema [37]. Na figura 3.7, podemos ver o processo de login de um usuário no sistema.

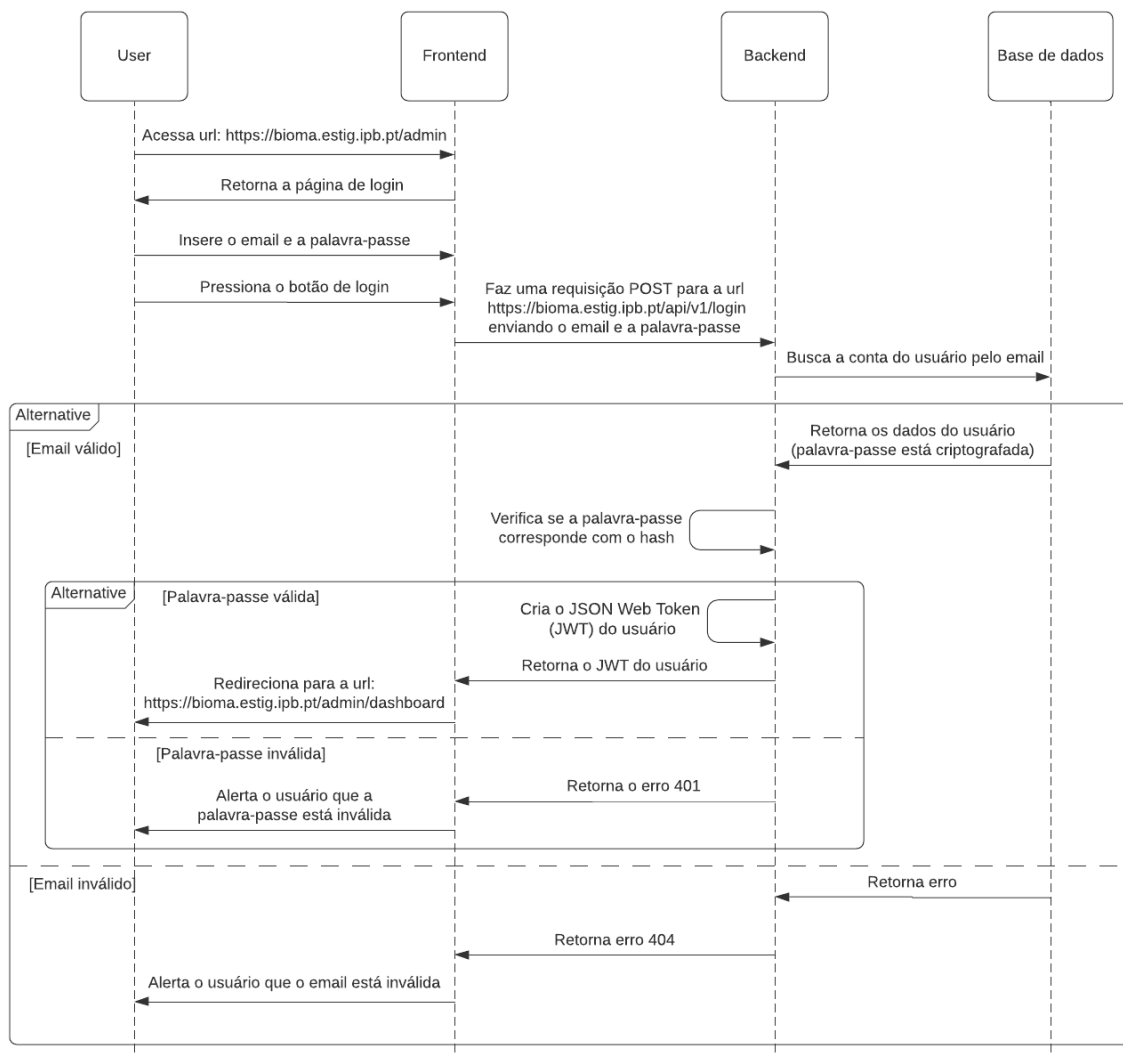


Figura 3.7: Diagrama de sequência do login.

Como pode ser visto no final, do cenário de sucesso, o usuário consegue obter seu JSON Web Token (JWT) e é redireccionado para o dashboard da sua empresa. Esse token é muito importante, pois ele funcionará como um crachá que será usado para fazer o controle de acesso das funcionalidades do sistema. Todas as requisições que o usuário fizer, no sistema, daqui para frente, precisará ser enviado este token junto, só assim o sistema conseguirá reconhecer o usuário que está fazendo a requisição e avaliar se o

mesmo tem as devidas permissões para realizar aquela requisição.

Na figura 3.8, podemos ver o diagrama de sequência de quando um usuário navega para a página que apresenta as produções. Para o sistema identificar quais produções devem ser apresentadas ao usuário, o frontend envia o token JWT que informa quem é o usuário que está fazendo a requisição para o backend, e então assim, após o backend validar o token, baseado nas informações do token, o backend retorna as produções que o usuário em questão pode visualizar.

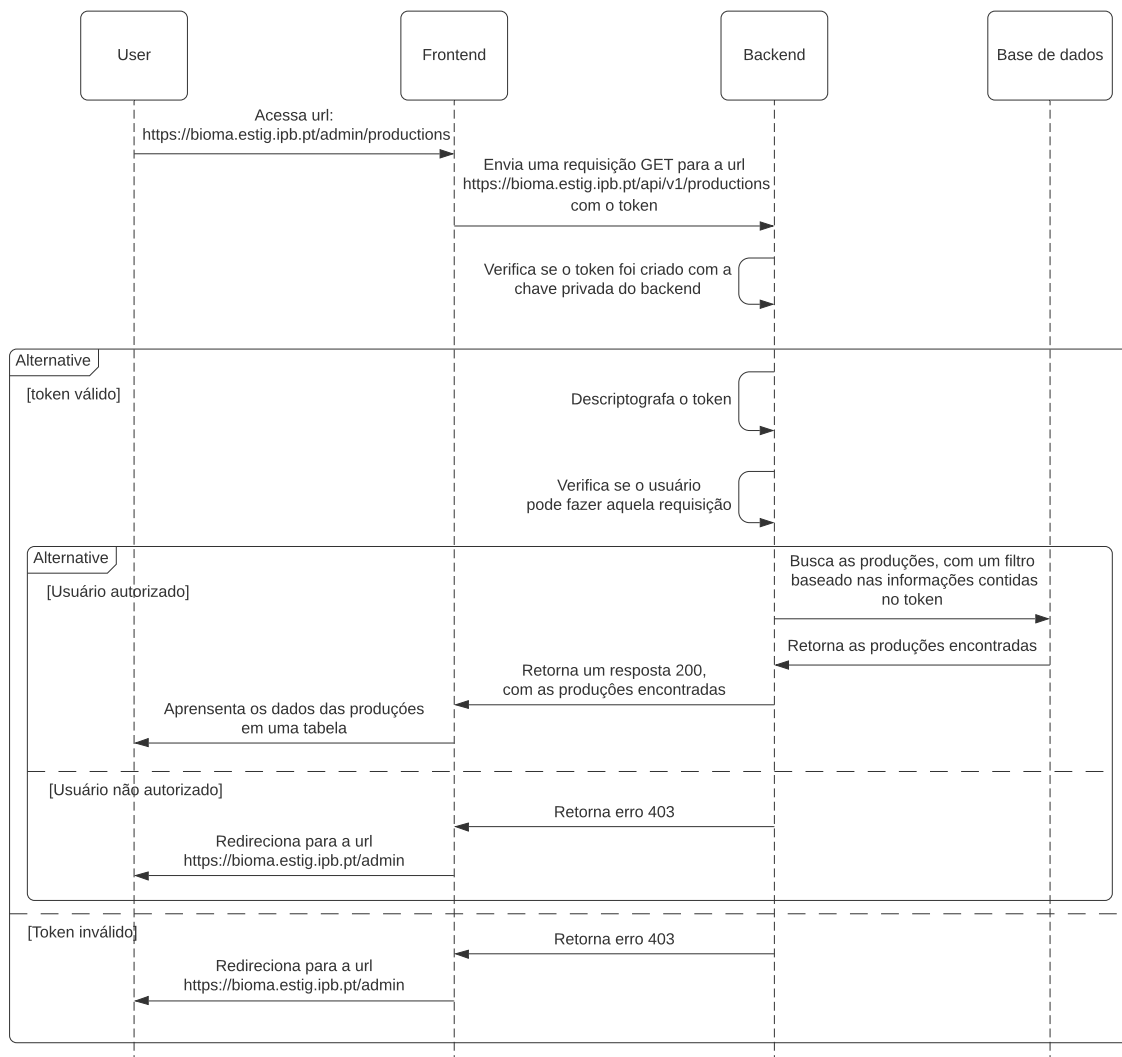


Figura 3.8: Diagrama de sequência para consultar produções.

3.2.10 Modelo entidade relacionamento

Na figura 3.9 é apresentado o modelo entidade relacionamento proposto para o sistema. Esse modelo exemplifica como os dados serão armazenados. Este modelo foi proposto inicialmente e ainda não possuía as coleções e os dados necessários para o funcionamento do histórico de alterações criptograficamente verificável.

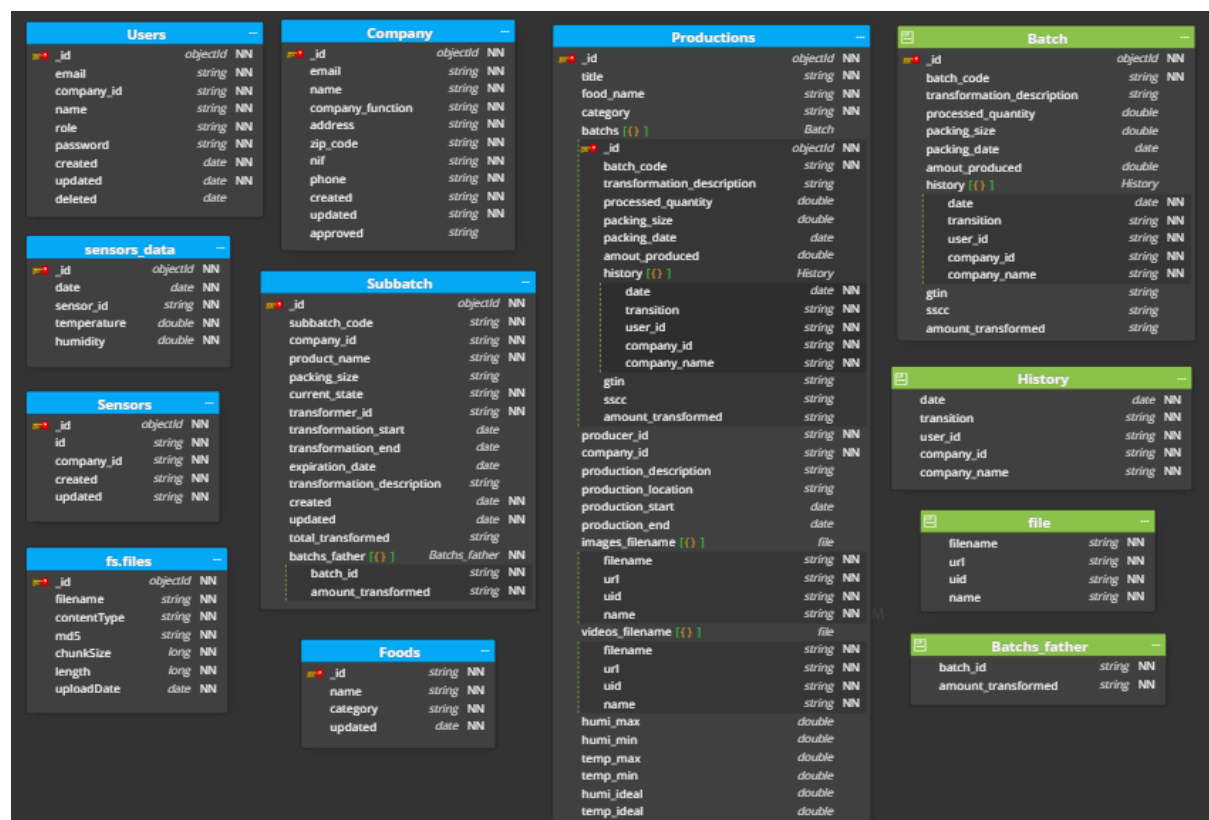


Figura 3.9: Modelo entidade relacionamento inicial.

Na figura 3.10 podemos observar o modelo entidade relacionamento com as coleções e os dados necessários para o funcionamento do histórico de alterações criptograficamente verificável.

Como podemos ver, o histórico de alterações criptograficamente verificável adiciona diversas informações necessárias, nas coleções existente, para o seu funcionamento, além de criar mais 4 coleções que vão registrar o histórico de alterações das principais coleções

do sistema.

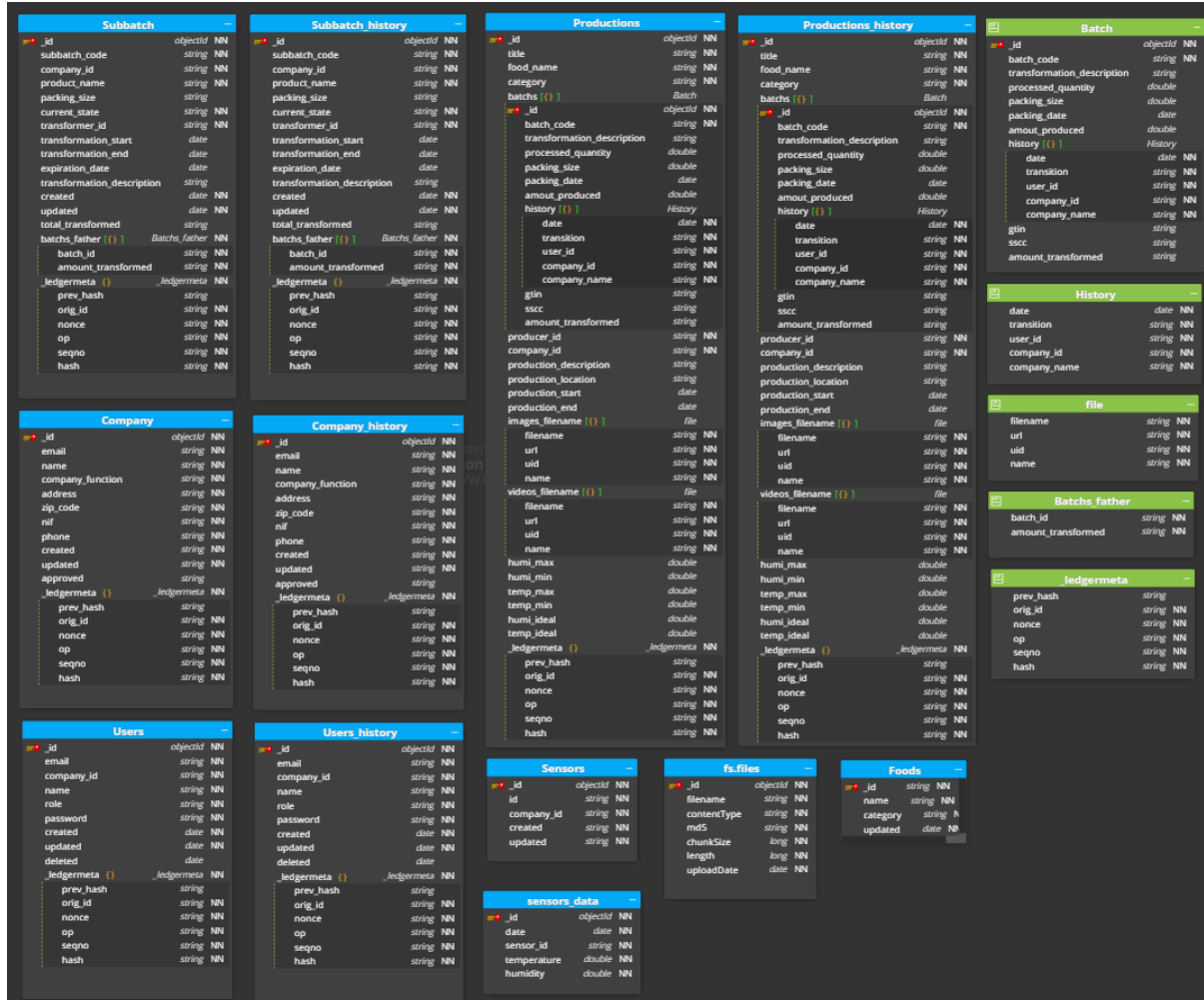


Figura 3.10: Modelo entidade relacionamento final.

3.2.11 Mockups

Nesta subseção vai ser apresentado as principais mockups da plataforma, salientando-se que as mockups são apenas uma sugestão do design final da plataforma, não sendo completamente fiel ao resultado final.

A seguir na figura 3.11 é apresentado o mockup representativo da página Home da

plataforma, essa é a página inicial da plataforma. A página tem o objectivo de fazer o marketing da plataforma portanto possui uma breve explicação do propósito da plataforma e algumas vantagens da rastreabilidade, além disso possui um botão que redirecciona os consumidores para a página onde eles podem rastrear o seu produto e verificarem todas as suas informações.



Figura 3.11: Página Home Mockup.

A seguir, na figura 3.12, é apresentado a mockup da página inicial para os agentes internos da cadeia de valor entrar na plataforma, ela dá acesso ao módulo que mantém toda a rastreabilidade. Nela também pode ser feita a solicitação de acesso da plataforma.



The mockup shows a login interface for 'BioTRACE'. It features a title 'BioTRACE' in bold, followed by the instruction 'Entre na sua conta!'. Below this are two input fields: 'Email' and 'Palavra-passe'. To the left of the 'Palavra-passe' field is a checkbox labeled 'Lembrar-me'. To the right is a link 'Esqueceu a palavra-passe?'. A blue button labeled 'INICIAR SESSÃO' is positioned below the inputs. At the bottom, there is a link 'Não tem conta? Solicite sua conta!'.

BioTRACE

Entre na sua conta!

Email

Palavra-passe

☐ Lembrar-me [Esqueceu a palavra-passe?](#)

INICIAR SESSÃO

[Não tem conta? Solicite sua conta!](#)

Figura 3.12: Página de autenticação Mockup.

Na figura 3.13 está representado o mockup da página de gestão de produções, nela o produtor poderá visualizar, criar, editar e deletar suas produções.

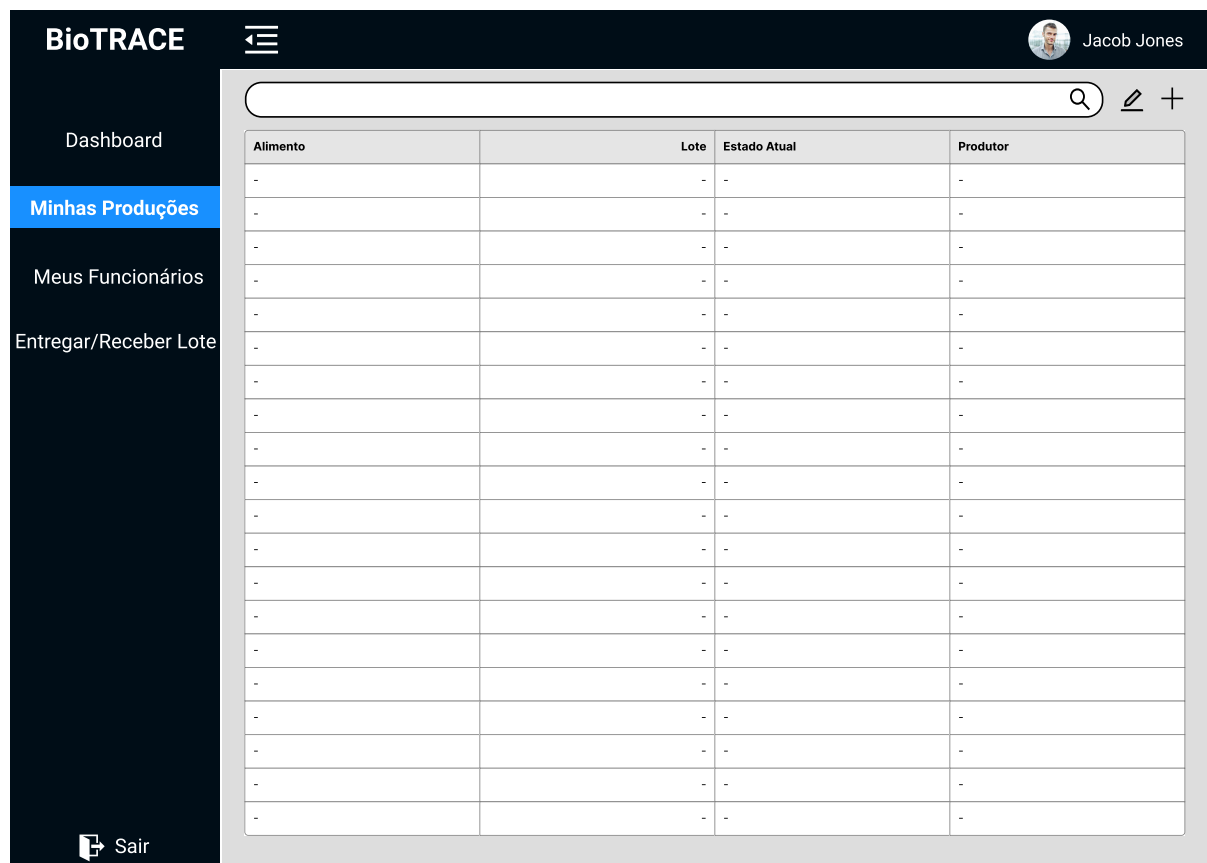


Figura 3.13: Página de gestão de produção Mockup.

Na figura 3.14 está representado as mockups responsáveis pelo fluxo do consumidor, ele acessa a página inicial, clica no botão para ver rastreabilidade, é redirecionado para tela da câmera, assim que efetuar a leitura de um código QR, a plataforma redirecionará o consumidor para a tela responsável por mostrar a informação da rastreabilidade referente ao lote do produto.

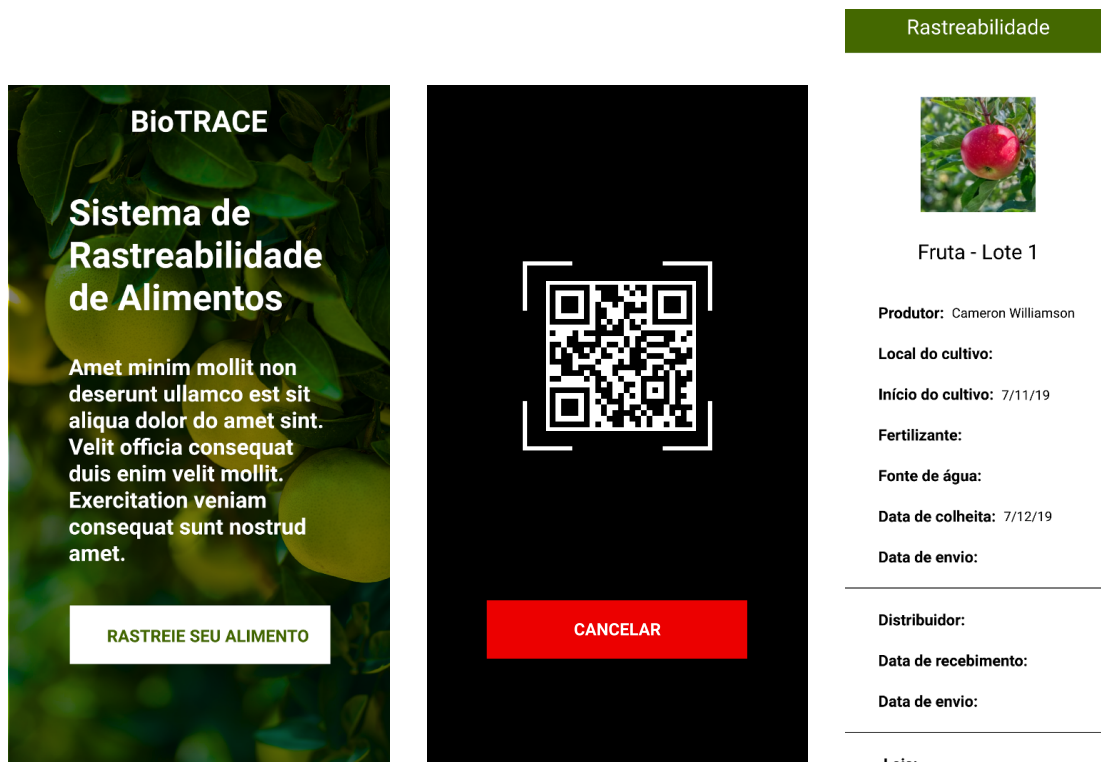


Figura 3.14: Mockups do fluxo do consumidor.

3.2.12 Arquitetura proposta

A arquitetura geral proposta para a plataforma Biotrace é composta por três camadas: camada visual, de negócios e de dados, sendo representado na figura 3.15.

A camada de visual será composta por uma aplicação Progressive Web Application (PWA) que tem por objetivo, para o consumidor, fazer o scan do código QR de um determinado produto e visualizar a informação de rastreabilidade desse produto. Para os agentes da cadeia a aplicação tem por objectivo a gestão de toda a cadeia de valor agroalimentar, onde podem ser geridos produtos, produções, lotes, códigos QR, entre outras funcionalidades de registo do ciclo de vida dos produtos e da cadeia de valor

alimentar.

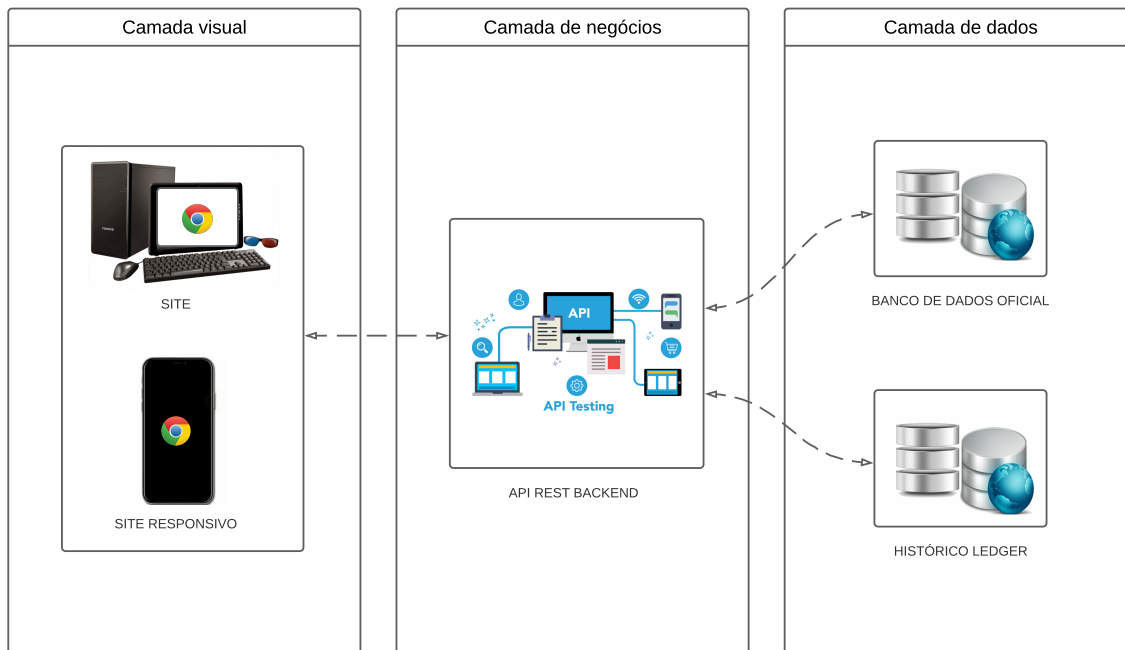


Figura 3.15: Arquitetura geral Biotrace.

A API da camada de negócios será uma Representational State Transfer (REST) API que permite a gestão da base de dados e a interação com a Blockchain para registro das operações e dados da sensorização, garantido a imutabilidade da informação. A REST API disponibiliza todos os serviços que permitem o funcionamento da aplicação site.

A camada de dados é constituída por uma base de dados não relacional que permite a gestão de todos os dados necessários ao funcionamento da plataforma, desde perfil de utilizadores, dados sobre os fluxos da cadeia de valor, estado dos produtos, entre outros. A bases de dados terá os mecanismos de segurança recomendados para as aplicações web e de imutabilidade dos dados.

Para garantir a rastreabilidade e a imutabilidade dos dados, será implementado um histórico de alterações criptograficamente verificável aplicando conceitos da tecnologia Blockchain, que irá garantir uma informação mais segura para o consumidor.

3.2.13 Componentes da Arquitetura

A interação entre os atores e os componentes funcionais da arquitetura é representado na Figura 3.16, podendo-se verificar que a plataforma na perspectiva dos atores principais, agentes da cadeia de valor, é apenas um site responsivo, não sendo visível todo o sistema por trás envolvido.

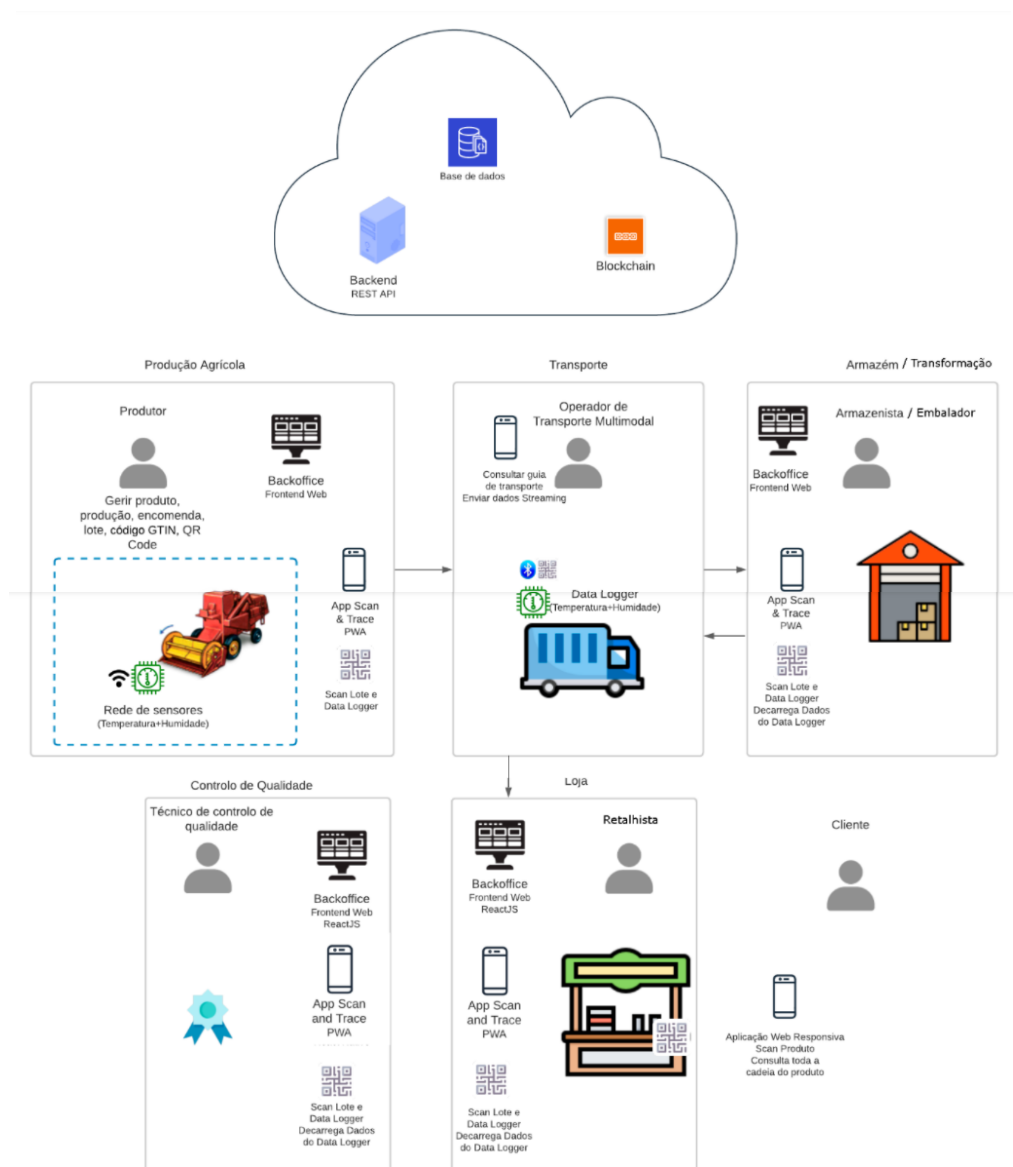


Figura 3.16: Componentes funcionais da arquitetura.

O produtor poderá utilizar o site da plataforma para registrar as suas produções e gerar os códigos QR dos lotes dos produtos. Pretende-se que sejam usados dois tipos de código, o código de barras baseado na norma GS1 que permite a interoperabilidade e a padronização entre os sistemas de rastreabilidade, e o código QR que irá ser afixado na embalagem do lote, que permite abrir uma URL com toda a informação de rastreabilidade do produto e também registrar a entrada e a saída do lote de cada etapa da cadeia de valor. O código QR poderá ser gerado por qualquer agente da CVAA.

O transportador poderá registrar a receção e a entrega dos lotes, assim como fazer a sincronização dos dados de temperatura e humidade do data logger, dados que normalmente fazem parte do ticket de frios. Além disso na cadeia é necessário sempre ter um transportador entre as diversas fases, sendo um ator alto valor para a cadeia.

O transformador tem o papel de processar e/ou embalar alimentos que podem ter origem em vários produtores para criar um sub lote a enviar para o retalhista. Durante o processo de transformação de alimentos devem ser gerados novos códigos segundo a norma GS1 e novos códigos QR, de forma a identificar os produtos que compõe a matéria prima, assim como manter a rastreabilidade dos novos produtos até chegarem ao consumidor. Caso o processo de transformação seja somente de embalagem, o transformador pode registrar na plataforma informações relevantes sobre o processo de embalagem e assim disponibilizar mais informações para o consumidor.

O armazenista poderá existir na cadeia de valor como entidade independente ou poderá ser feito esse papel pelo produtor, transformador ou retalhista. Os perfis de membros ativos da cadeia de valor têm acesso às mesmas funcionalidades de gestão de lotes e de dados, sendo transparente para a rastreabilidade se algum dos atores identificados não participa no processo de um determinado lote.

O retalhista tem um papel importante na cadeia de valor ao garantir que o produto é entregue em condições ao cliente e que foi seguida toda a regulamentação aplicável. Poderá consultar toda a informação de rastreabilidade do produto para verificar se as condições exigidas para aquele produto foram cumpridas. Caso o produtor ou transformador não tenha emitido o código QR para o consumidor deverá fazê-lo nesta etapa de modo a que

o produto possa ser rastreado pelo consumidor.

O consumidor poderá através do Scan do código QR disponível na embalagem, abrir uma página de rastreabilidade do produto com todas as informações do produto, tais como o produtor, variedade de produto, localização geográfica, selos de garantia de qualidade, transportador, transformador, retalhista, assim como os dados de temperatura e humidade durante o transporte, validade, entre outros parâmetros.

Capítulo 4

Desenvolvimento da plataforma de rastreabilidade

Neste capítulo é demonstrado todo o fluxo de desenvolvimento do sistema, identificando cada etapa, desde a seleção das ferramentas até o desenvolvimento da plataforma Biotrace proposta como solução para os problemas identificados.

4.1 Tecnologias

Nessa seção são apresentadas as principais tecnologias que foram utilizadas para o desenvolvimento de todas as aplicações que compõe a plataforma Biotrace.

A escolha das tecnologias foram feitas no início do desenvolvimento, no início do processo foi feito um levantamento das tecnologias mais utilizadas do mercado e que possuem mais suporte da comunidade, após o levantamento foi feita a escolha das tecnologias mais apropriadas ao projeto, levando em consideração os requisitos funcionais e não funcionais, os diagramas UML, a modelagem dos dados e os mockups.

4.1.1 React

A tecnologia utilizada para o desenvolvimento das interfaces foi o React, uma biblioteca JavaScript, capaz de criar uma aplicação que carrega uma única página HTML e todos os códigos JavaScript e CSS necessários para a aplicação ser executada. Dessa forma a aplicação não necessita realizar outras requisições para o servidor quando ocorre uma interação, o que significa que a aplicação não é recarregada. [42].

React foi criado por Jordan Walke e atualmente é mantido pelo Facebook. Tornou-se uma das bibliotecas Javascript mais famosas para o desenvolvimento de aplicações web e é utilizada pelo Facebook no ambiente de produção [43]. Para conseguir realizar alterações nas páginas HTML, precisamos manipular os objetos da interface, e isso é feito através do Document Object Model (DOM), que contém todos os objetos da interface e permiti manipulá-los com uma linguagem de script.

No React foi introduzido um conceito novo que é fundamental para o sucesso da biblioteca, o conceito de DOM virtual. O DOM virtual pode ser renderizado tanto no lado do cliente como no lado do servidor. Quando um componente da interface é atualizado, o React primeiro atualiza o DOM virtual, depois verifica com o DOM real o que está diferente, portanto foi modificado, e então atualiza, no DOM real, apenas os componentes que foram modificados no DOM virtual. Fazendo assim a menor quantidade de manipulação no DOM real, conseguindo ser mais leve que a biblioteca JavaScript Angular [43].

Para definir a estrutura de renderização de cada interface, o React utiliza o JSX que se trata de uma sintaxe semelhante a XML/HTML. A sintaxe é utilizada por pré-processadores como o Babel que converte o JSX em um código que o Browser conseguirá entender. Em resumo o JSX permite escrever código HTML em arquivos JavaScript.

Em React, podem ser criadas componentes reutilizáveis, ou seja, uma parte da interface pode ser usada em várias partes de uma aplicação. Neste sentido, no ecossistema do React existem diversas bibliotecas de componentes prontos para serem utilizados. Neste projeto foi utilizado a biblioteca Ant Design, uma das bibliotecas de componentes mais utilizadas atualmente. Para o controlo do estado da plataforma foi utilizada a biblioteca Redux

e para gerir os efeitos colaterais foi utilizada a biblioteca `redux-saga`. Para realizar as requisições para a API foi utilizado `axios` e para a estilização da interface foi utilizado o `styled-components`.

4.1.2 MongoDB

Para a base de dados foi decidido utilizar uma base de dados não relacional, devido a ser mais flexível e escalável que as bases de dados relacionais. Essa decisão foi tomada pelo motivo de precisar guardar dados de sensores.

Em uma base de dados não relacional, podem-se adicionar novos dados sem ter que se pré-definir uma estrutura, podendo ser guardados dados não estruturados, semiestruturados e estruturados.

Dentro das opções de bases de dados não relacionais foi decidido utilizar o MongoDB por diversos motivos como ser gratuito, ser de código aberto, oferecer uma arquitetura eficiente e escalável.

4.1.3 Flask

Para o desenvolvimento de uma API, com a arquitetura REST foi utilizado o Flask, um microframework escrito na linguagem Python, uma linguagem simples, com uma comunidade muito vasta e com diversas bibliotecas de algoritmos de inteligência artificial [44].

Primeiro foi decidido que a tecnologia escolhida seria do ecossistema do Python, por causa das diversas bibliotecas de algoritmos de inteligência artificial que podem vir a contribuir com a plataforma Biotrace em trabalho futuros.

Dentro do ecossistema do Python existem diversas frameworks, portanto para escolher o framework foi levado em consideração a tecnologia para o banco de dados. Como foi decido utilizar uma base de dados não relacional, o framework Flask apresentou-se como uma boa escolha, devido a ser uma microframework muito flexível que funciona perfeitamente com bases de dados não relacionais, como o MongoDB.

4.1.4 Nginx

Com o objectivo de melhorar a performance da aplicação web foi utilizado o Nginx. Um software de código aberto, confiável com recursos de alto desempenho, que pode ser útil para diferentes fins, como proxy reverso [45].

No Nginx foi configurado como o site da aplicação seria disponibilizado, devido ao fato da aplicação precisar ter acesso ao hardware do usuário, o site da aplicação foi disponibilizado via HTTPS. O HTTPS fornece comunicação segura entre usuários e o site, e também a autenticação do site acessado [45].

Manter vários certificados HTTPS é dispendioso, visto que um certificado HTTPS tem um custo financeiro e operacional [45]. Portanto para evitar de usar vários certificados foi configurado, no Nginx, um proxy reverso que redireciona as requisições via HTTPS para o flask que não possui certificado HTTPS, garantindo uma comunicação segura sem a necessidade de manter outro certificado HTTPS.

4.1.5 Docker

A containerização de aplicações é um método que combina a aplicação com suas dependências e bibliotecas, e executa em espaço de usuário isolado chamado contêiner. Um contêiner é um ambiente de computação totalmente isolado e portátil, ele possui tudo que precisa para executar uma aplicação incluindo seus binários, bibliotecas, dependências e arquivos de configuração [46].

Com a containerização, você tem menos despesas gerais durante a inicialização e não precisa configurar sistemas operacionais separados para cada aplicação, pois todos compartilham um núcleo do sistema operacional. Devido a essa alta eficiência, foi utilizado o Docker, uma plataforma capaz de efetuar a containerização das aplicações implementadas garantindo que as aplicações funcionam em todos os ambientes [46].

4.2 Camada de interface

A camada visual da aplicação é onde se concentra o uso da tecnologia React, é a camada que os utilizadores tem acesso. A construção dessa camada foi feita baseada nos mockups feitos na etapa anterior. Nesta seção é apresentado, a estrutura do arquivo, as principais páginas desenvolvidas da aplicação e algumas questões sobre controle de fluxo e de estado.

4.2.1 Estrutura de Arquivos

Uma boa estrutura de arquivos é muito importante para uma boa organização de projecto, pois organiza os arquivos em pastas de acordo com os seus contextos, gerando uma clareza maior ao desenvolvedor. Na figura 4.1 é apresentado a estrutura de arquivos da camada visual da aplicação.

Para que fique mais claro a estrutura, a seguir é apresentado um breve resumo do propósito das principais pastas da aplicação, localizadas na pasta "src":

- Components: esta pasta contém todos os componentes reutilizáveis da aplicação.
- Helpers: esta pasta possui os componentes auxiliares da aplicação.
- Hooks: esta pasta contém o arquivo de controle de token do projeto..
- Pages: os arquivos presentes na pasta pages são utilizados para a construção das interfaces da aplicação..
- Services: é onde se concentra os arquivos que fazem a integração com outros serviços.
- Store: é onde se concentra os arquivos de controle de estado e controle de efeitos.
- Styles: é onde se concentra os arquivos de estilização das páginas.

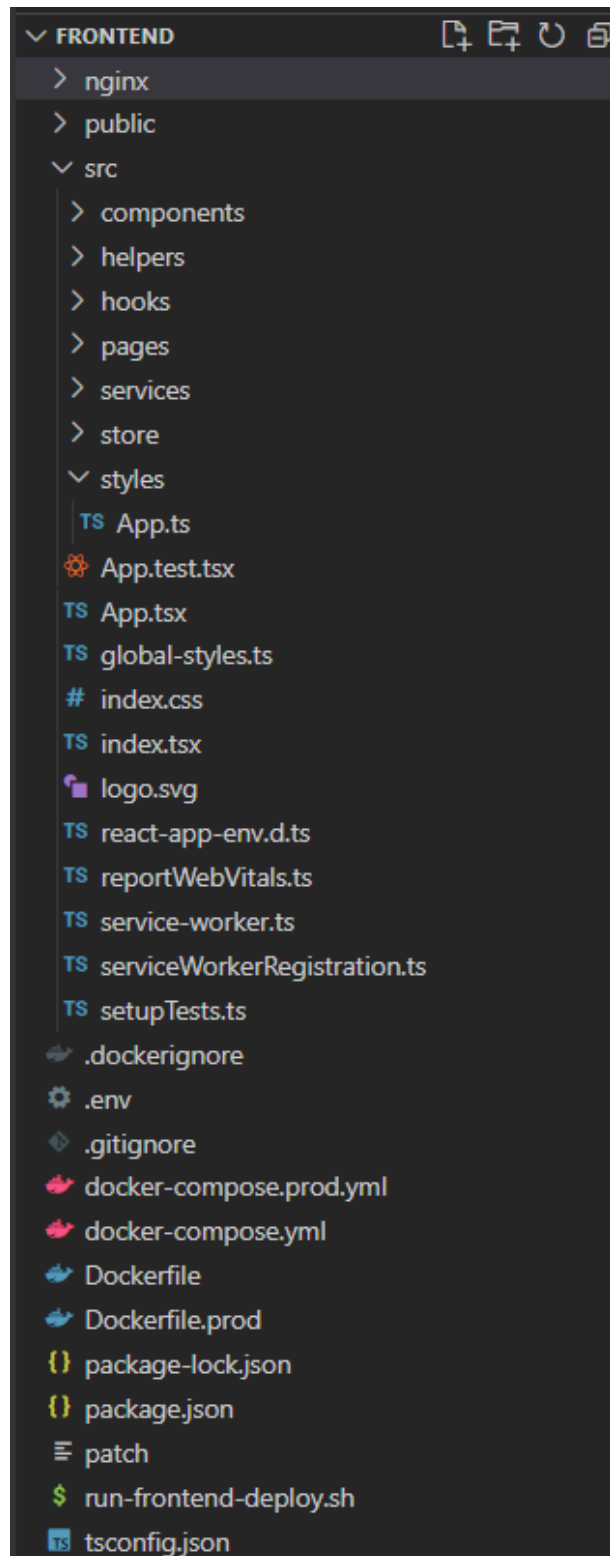


Figura 4.1: Estrutura de arquivos da camada visual (React).

4.2.2 Página Inicial

A página inicial do sistema, que está representada na figura 4.2, foi definida para o marketing digital, com o objetivo de conscientizar e informar os usuários sobre a importância da rastreabilidade de alimentos. É apresentada uma descrição da proposta do sistema e seus benefícios, bem como um botão para o consumidor final visualizar a rastreabilidade de algum alimento, esse botão redireciona o consumidor final para a tela onde a aplicação solicitará acesso da câmera ao hardware do usuário. Nessa página o consumidor poderá ler o código QR de algum produto Biotrace, assim que a leitura for efetuada e confirmada, o usuário é redirecionado para a página responsável por disponibilizar as informações da rastreabilidade do produto Biotrace em questão.

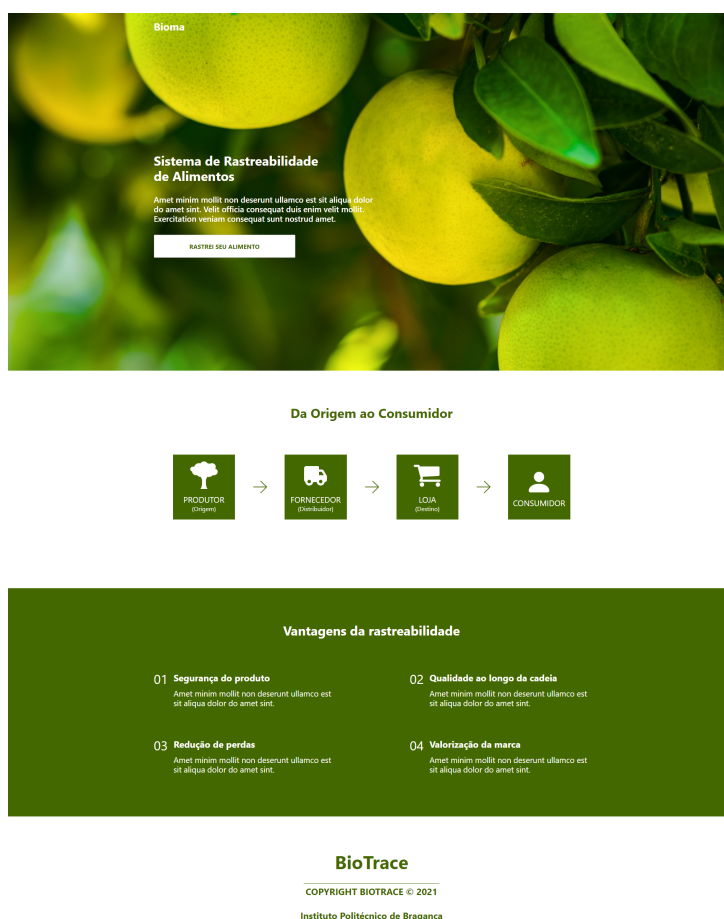


Figura 4.2: Página inicial.

Na figura 4.3 está representado as páginas que compõe o fluxo do consumidor no celular.

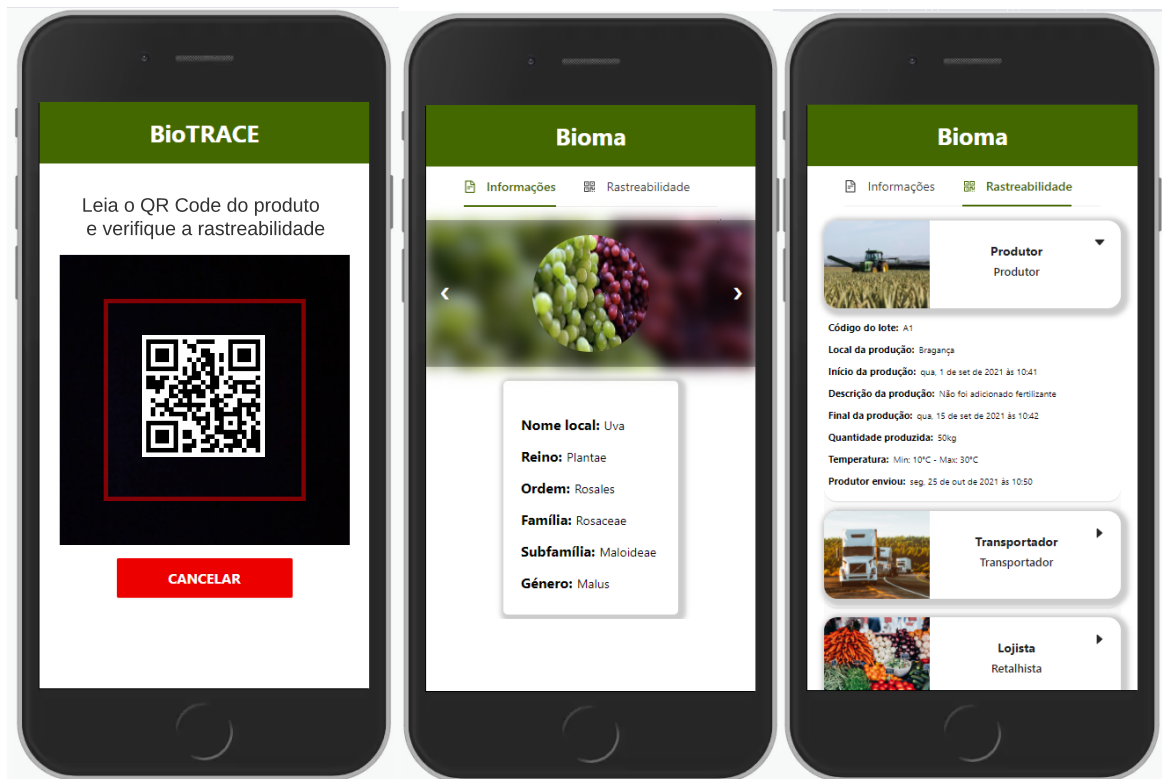


Figura 4.3: Páginas do fluxo do consumidor.

4.2.3 Página de Autenticação

A página de autenticação do sistema, que está representada na figura 4.4, é responsável por disponibilizar as funcionalidades de autenticação, sendo elas, iniciar sessão, recuperar senha e solicitar acesso. É por essa página que todos os agentes da CVAA entram no sistema, para isso basta o sistema validar os dados inseridos no campo email e palavra-passe e redireccionar o usuário para a página Dashboard.



* Email

* Palavra-passe



[Esqueceu a palavra-passe?](#)

Entrar

[Solicitar Acesso](#)

Figura 4.4: Página de autenticação.

4.2.4 Páginas para registrar a rastreabilidade

A página para registrar a rastreabilidade é onde os agentes da CVAA podem ler o código QR de algum lote ou sub-lote, para informar a plataforma que recebeu ou entregou algum lote ou sub-lote. Na figura 4.5, são apresentados as páginas responsáveis pelo fluxo para efetuar a leitura.

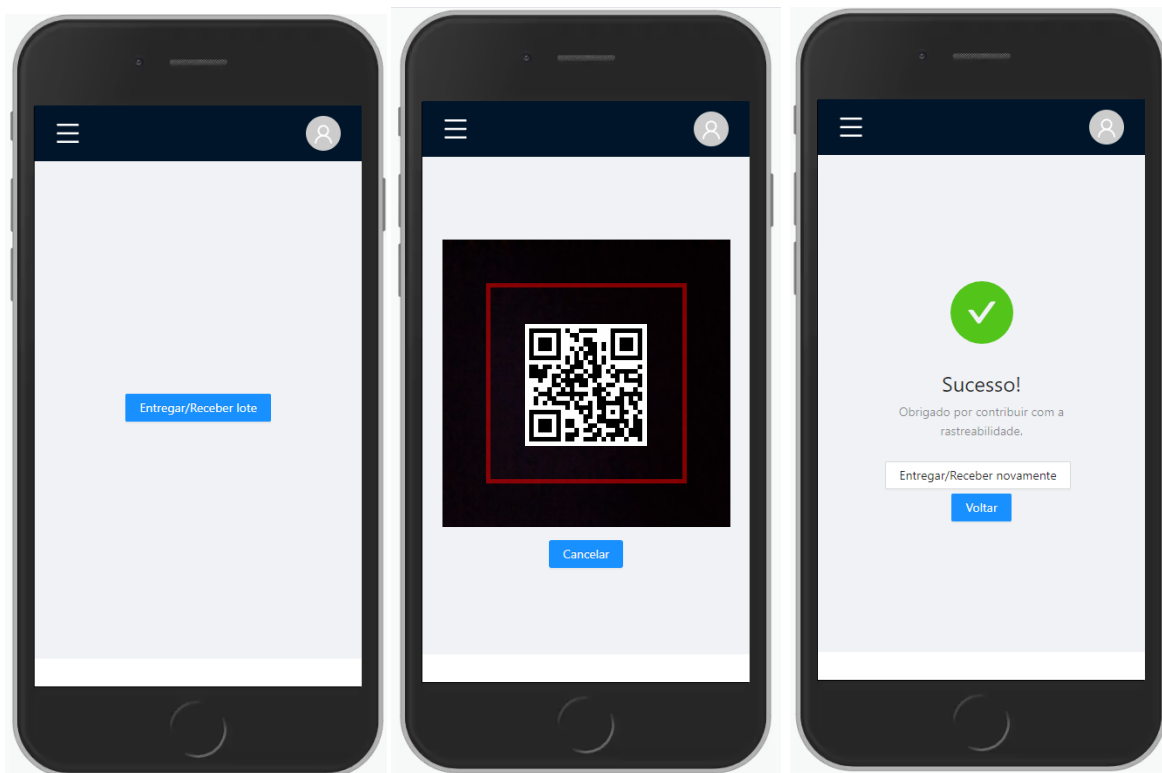


Figura 4.5: Páginas para efetuar leitura do QR Code.

Como pode ser visto é um processo simples para o utilizador, porém para o sistema, se trata de um processo complexo. Assim conseguimos reduzir a quantidade de erros manuais.

4.2.5 Página para visualizar a rastreabilidade

A página para visualizar a rastreabilidade é onde o consumidor poderá consultar todas as informações, que foram disponibilizadas, do produto que comprou, para aceder a esta página basta abrir a url do código QR ou ir na página inicial e efetuar a leitura do código QR que também redireciona. Na figura 4.6 é apresentada a página responsável por apresentar as informações referentes à rastreabilidade de um produto.

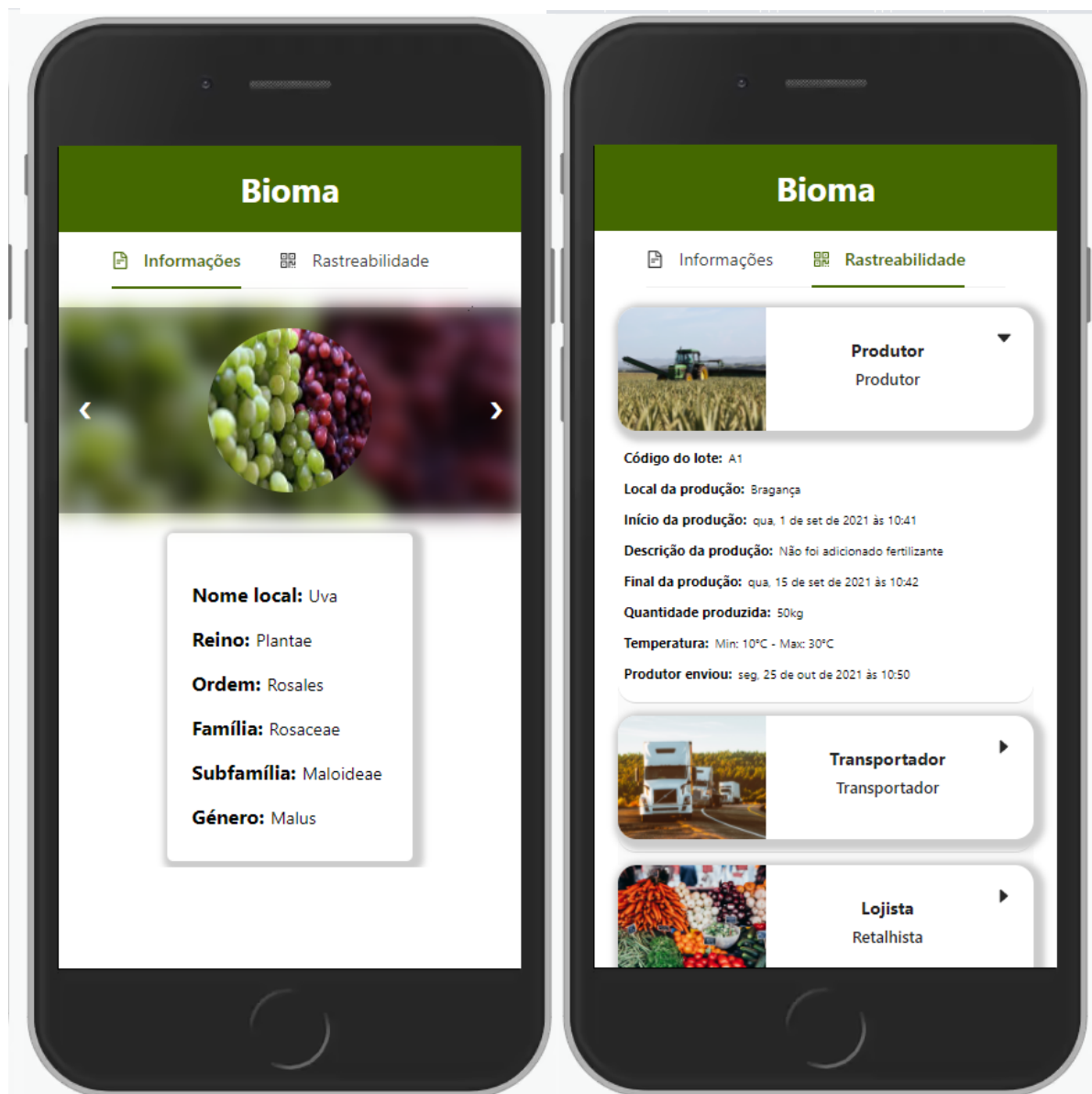


Figura 4.6: Páginas da rastreabilidade de um produto.

Para que a url do código QR redirecione para uma página existente é necessário que o lote, referente aquele código, já tenha chegado ao ultimo estágio da cadeia, o retalhista. Portanto, assim que o lote chega ao retalhista a url passa a ser válida, podendo ser acessada

por qualquer pessoa.

4.2.6 Página de gestão de produções

A página de gestão de produções possui diversas funcionalidades, que variam de acordo com o papel do utilizador. Na figura 4.7 é apresentada a página a partir da visão de um produtor gestor, portanto a página contém todas as funcionalidades necessárias para manipular as informações de uma produção e de um lote, como, visualizar, criar, editar e deletar.



bioma

soluções para gestão agrícola

Dashboard

Minhas Produções

Entrega/Receber lote

Meus Funcionários

Pesquisar por  Digite o título



Temperatura

Gerir Informação

Título

Categoria

Alimento

GTIN

SSCC

Início da produção

Local da produção

Fim da produção

Descrição da produção

Data de validade

Min

Max

Produção de maçã

Fruta

Maçã

2 de outubro de 2021

Porto

20 de outubro de 2021

Não foi adicionado fertilizante

11 de novembro de 2021

8°C

32°C



Produção de carne

Carne

Pato

1 de abril de 2021

Porto

23 de setembro de 2021

12 de dezembro de 2021

-12°C

12°C



Produção de uvas

Fruta

Uva

1 de setembro de 2021

Bragança

15 de setembro de 2021

Não foi adicionado fertilizante

1 de outubro de 2021

10°C

30°C



☐

Código do lote

☐

Quantidade produzida

☐

Descrição da transformação

☐

Quantidade transformada

☐

Tamanho da embalagem

☐

Data de embalamento

☐

Estado atual

☐

Status

☐

Visualizar Rastreabilidade

☐

Gerir Informação

A2

60kg

Foi embalada a vacuo

58kg

750gr

16 de setembro de 2021

Armazenista recebeu

Em progresso

Visualizar



A1

50kg

Lojista recebeu

Finalizado

Visualizar



A3

Transformador recebeu

Em progresso

Visualizar



1

10 / page

Figura 4.7: página de gestão de produções.

Como pode ser visto as informações são apresentadas em forma de tabela, cada linha representa uma produção, e cada coluna uma informação referente à produção. As linhas podem ser expandidas, com um clique, mostrando todos os lotes daquela produção, em formato de tabela, no mesmo modelo, sendo apresentadas em linhas os lotes e as colunas apresentam as informações de cada lote.

Além de colunas para informações, como pode ser visto, há também colunas com botões que accionam funcionalidades ou abrem uma modal com outras informações. Os

principais botões são o lixo e o olho, esses botões, que são apresentados em todas as linhas, possibilitam deletar e editar a produção ou lote em questão, respectivamente.

Outro botão importante é o para visualizar rastreabilidade, que só esta presente nas linhas da tabela de lote, é responsável por abrir uma modal com todas as informações referentes a leitura do código QR daquele lote. Na figura 4.8, é possível visualizar como essas informações são apresentadas ao utilizador no celular.

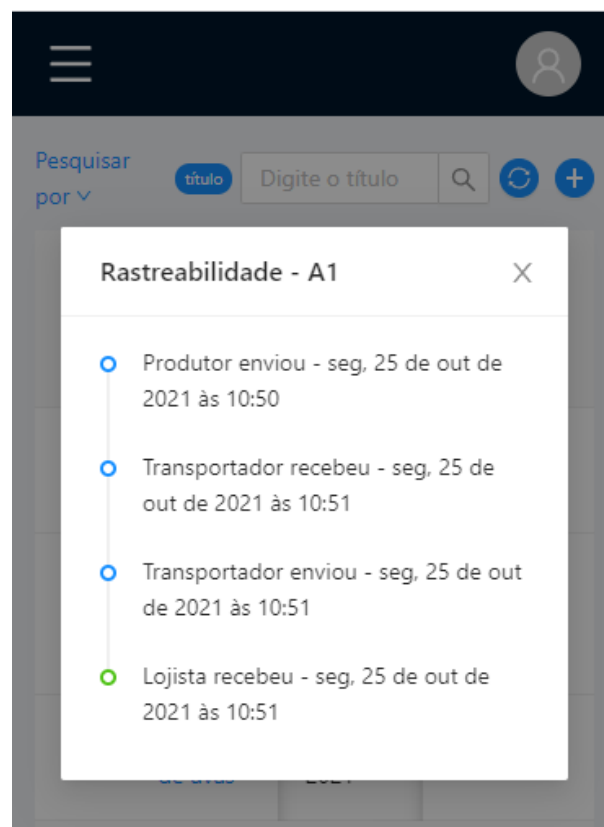


Figura 4.8: Modal para visualizar a rastreabilidade.

Como pode ser notado, a primeira coluna da tabela há um checkbox, onde possibilita os produtores selecionarem os lotes que desejam visualizar o código QR. Com os lotes selecionados, nessa mesma página abre uma lista com os códigos QRs, e um botão para fazer download de um PDF com os códigos QRs, os códigos de barras e algumas outras informações dos lotes selecionados. Na figura 4.9 é apresentado um PDF gerado a partir da plataforma.

Produção de uvas

Local: Bragança

Data de produção: 15 de setembro de 2021

Peso: 60 kg

Data de embalagem: 16 de setembro de 2021

Data de validade: 1 de outubro de 2021

Lote: A1



Figura 4.9: PDF de um lote cadastrado na plataforma.

Para cada lote selecionado é gerado uma página no PDF que apresenta o código QR e o código GS1 do lote, possibilitando ao produtor imprimir o PDF e etiquetar os respectivos

lotes.

O código de barras GS1 do lote é gerado a partir das informações fornecidas, principalmente, pelo produtor e pelo transformador. Portanto dependendo das necessidades da rastreabilidade os códigos de barras podem variar de tamanho, alguns podem ter mais informações outros podem ter menos, dependendo dos dados que foram registrados em cada lote.

O código QR é responsável por registrar a rastreabilidade na plataforma e levar o consumidor para a página onde ele possa visualizar as informações do lote. O seu valor é uma url que redireciona para a plataforma na página de rastreabilidade do lote, portanto caso o código seja lido em qualquer software o consumidor vai ser redirecionado para a plataforma. Já para registrar a rastreabilidade o agente precisa ler através da plataforma.

4.2.7 Página de gerenciamento de sublote

A página responsável para gerenciar sublotes, como a página de produção, também possui diversas funcionalidades que variam de acordo com o utilizador. Na figura 4.10 é apresentado a página a partir da visão do transformador. Pois o transformador é único da plataforma que pode transformar lotes em sublotes. As informações do sublote são também apresentadas em tabela no mesmo modelo, linha para sublotes e coluna para as informações e funcionalidades.

Para criar um sublote o transformador clica no botão "Criar sublote", insere as informações do sublote e confirma. As principais informações são, quais lotes e quantos quilos de cada lote foi utilizado, qual o código do sublote e quando ocorreu a transformação.

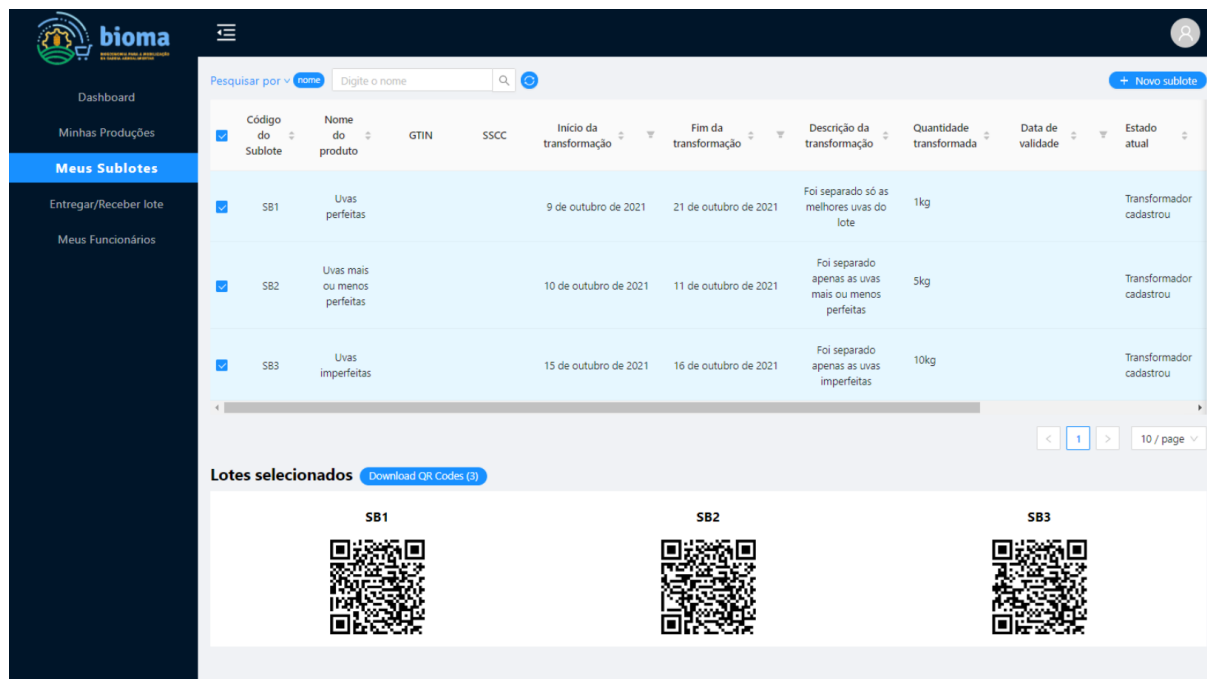


Figura 4.10: Página para gestão dos sublotes.

4.2.8 Controlo de estado

Para o controlo do estado da plataforma foi utilizada a biblioteca Redux, uma biblioteca que possibilita controlar o estado de toda a aplicação, descrevendo como este estado será atualizado em resposta aos objetos de ação, os chamados actions. Além do redux foi utilizado também a biblioteca redux-saga, que gerencia todos os efeitos colaterais das actions.

Para a biblioteca fazer o controlo de estado de forma adequada é necessário dividir os objetos da aplicação em módulos baseado no contexto de cada objeto. Cada módulo da aplicação possui as suas Actions, o seu Redux Store e o seu Saga, na figura 4.11 é apresentado um diagrama que demonstra o funcionamento do mesmo.

O produtor quando deseja criar uma produção, ao clicar no botão para criar, ele dispara uma action que vai chamar o Saga do módulo de produções que vai buscar os dados necessários no Redux Store e enviar para a API, solicitando que o mesmo crie uma

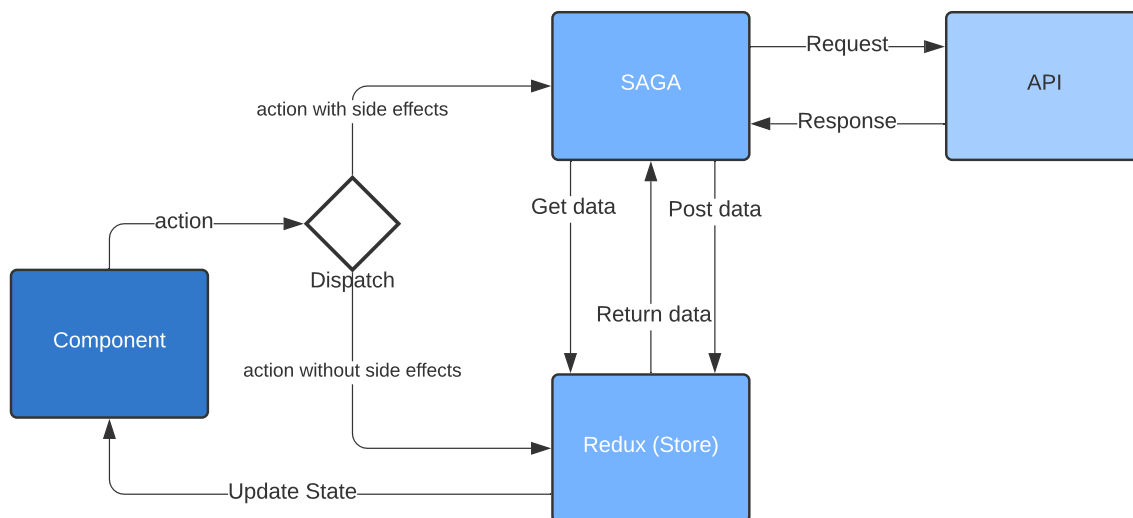


Figura 4.11: Fluxo de informação.

produção. Em caso de sucesso, a produção é criada e aparece uma mensagem de sucesso para o produtor.

4.2.9 Controlo de rotas

Para fazer o controlo de rotas na plataforma foi necessário desenvolver dois componentes que englobam as páginas e são responsáveis por verificar o token e redirecionar o usuário caso o token seja válido ou inválido, dependendo da função do componente. Por exemplo se uma página é privada e o usuário não está com a sessão ativa ele deve ser redirecionado para a página de login, já se o usuário esta com a sessão activada e tentar acessar a página de login, ele deve ser redirecionado para o dashboard da plataforma.

Portanto foi feito um componente que controla as páginas privadas da plataforma e um que controla a página de login. O componente da página privada é dinâmico e pode receber em forma de parâmetro todos os tipos de usuários que podem acessar essa página e todos os que não podem.

4.3 Camada de Negócio

A camada de negócio é onde se concentra o uso da tecnologia Flask, que foi utilizada para construir uma API que dá estrutura e apoio às ações do usuário. Na figura 4.12 é apresentado o fluxo de informação desde da camada visual até chegar no banco de dados, focando-se na camada de negócio onde está a API feita em Flask.

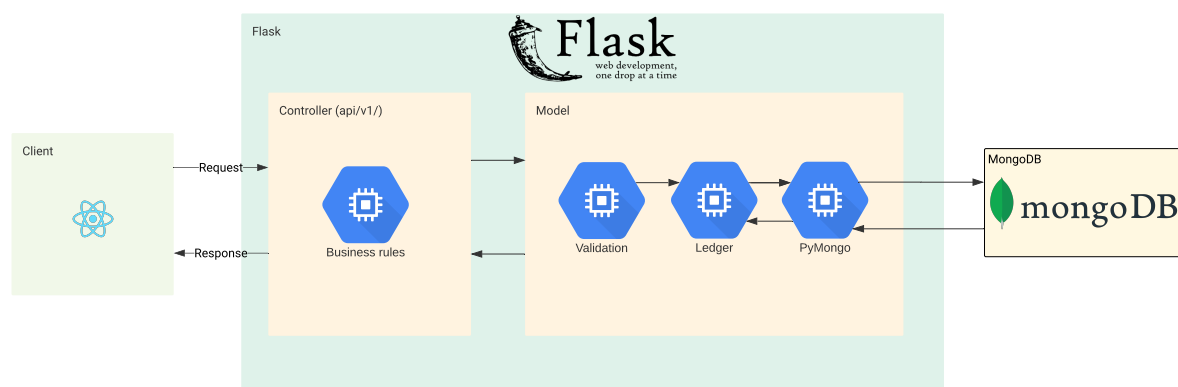


Figura 4.12: Fluxo de informação.

4.3.1 Módulo de Autenticação

O módulo de autenticação do sistema é composto por diversas funcionalidades, dentre elas estão, iniciar sessão, solicitar acesso de uma empresa, liberar acesso de uma empresa, recuperar palavra-passe, verificar token e registar funcionários.

4.3.1.1 Iniciar sessão

Para essa funcionalidade foi definido um endpoint na API, com o método POST e com a rota `"/login"`. Esse endpoint recebe um email e uma palavra-passe, e retorna um JWT em caso de sucesso. O JWT, em questão, possui informações do usuário, da empresa e da sua função na mesma. Com base nessas informações o React consegue saber quais páginas e funcionalidades o usuário pode ou não acessar.

Para garantir a autenticidade do token, o próprio é assinado com uma palavra-secreta, sendo assim só tokens assinados com essa palavra-secreta são válidos pelo sistema.

4.3.1.2 Controle de acesso de uma empresa

O controle completo de acesso de uma empresa é definido por 2 endpoints, um de solicitar acesso e outro de aprovar ou recusar acesso.

Para solicitar acesso de uma empresa foi definido um endpoint na API, com o método POST e com a rota `"/companies"`, esse endpoint recebe os dados da empresa e registra no banco de dados.

Com os dados da empresa registrados no banco de dados, o administrador da plataforma Biotrace pode consultar e aceitar o acesso da empresa, através do endpoint `"/companies/companyId/approve"`, para esse endpoint retornar sucesso é necessário enviar um JWT válido com os dados do administrador da plataforma e um `companyId` de uma empresa que exista e não foi aprovada ainda.

4.3.1.3 Recuperar palavra-passe

Para essa funcionalidade do sistema, foi definido 3 endpoints, que devem ser chamados em uma ordem específica pelo React, todos com o método POST. O primeiro é o `"/forgot-password"`, nesse endpoint o usuário envia seu email, declarando que esqueceu sua palavra-passe, enviando o sistema um código para o email. Com o código, o usuário digita no campo e então o sistema verifica se o código está certo através do endpoint `"/verify-code"`, caso o código esteja correto a aplicação permite a alteração da palavra-passe do usuário.

Para que a alteração ocorra de fato, é necessário uma chamada para o último endpoint, o `"/change-password"`, nesse endpoint é necessário enviar a nova palavra-passe e o código que foi enviado no email do usuário. Assim a API garante que o usuário só vai conseguir recuperar e mudar a palavra-passe caso tenha acesso ao email da conta.

Com esse fluxo completo de três endpoints o usuário consegue mudar palavra-passe e continuar acessando a plataforma.

4.3.1.4 Verificar token

Essa funcionalidade é uma das mais importante da plataforma, ela é responsável por validar o token. Para essa funcionalidade foi feita um decorator que é chamado antes de todos os endpoints, com exceção dos endpoint de autenticação que não precisam de token,

Esse decorator funciona da seguinte maneira, caso o decorator seja chamado sem nenhum parâmetro ele apenas vai validar o token. Caso seja passado com o perfil do utilizador como parâmetros, a funcionalidade além de validar o token vai barrar todos que não estão presentes nos parâmetros.

Então essa funcionalidade garante que as outras funcionalidades sejam acessíveis apenas quando o usuário possuir um token válido e tiver na lista de parâmetros quando for o caso. Portanto essa funcionalidade é de grande importância para o sistema, pois controla o acesso de toda a API.

4.3.1.5 Registrar funcionários

Para essa funcionalidade foi definido um endpoint com o método POST na rota `"/companies/employees"`, nesse endpoint além de ser necessário enviar o nome, email e função do funcionário também é preciso enviar um JWT de um gestor de empresa, é com esse token que o sistema vai saber qual é a empresa daquele funcionário e se deve ou não registrar o funcionário em questão. Entre receber os dados e retornar foi feito com sucesso certo o sistema gera uma palavra-passe aleatória que é enviada por email ao funcionário.

4.3.2 Módulo de Rastreabilidade

O módulo de rastreabilidade é principal módulo do sistema, é nele que se encontra as principais funcionalidades do sistema, desde da leitura do código QR para o consumidor visualizar a rastreabilidade até a leitura do código QR para registrar a rastreabilidade pelos agentes da CVAA.

O módulo de rastreabilidade engloba todo o processo de um lote, desde sua criação

até chegar ao retalho e ser vendido ao consumidor. Para criar um lote na plataforma é necessário antes um produtor criar uma produção. Essa produção deve ser subdividida em lotes, cada lote vai possuir suas próprias informações e um código QR único. Para que isso seja minimamente possível foi definido três endpoints, um para criar uma produção, outro para editar uma produção e outro para editar um lote.

O primeiro endpoint é o `"/productions"`, com o método POST, nele é possível criar uma produção, porém são necessários alguns requisitos para que a plataforma crie a produção. Deste modo é preciso enviar o JWT de um produtor gestor ou supervisor, e algumas informações obrigatórias da produção como, qual o produto, onde foi produzido, quando foi produzido entre outras, além dessas informações podem ser enviadas outras informações que são opcionais.

O segundo endpoint é o `"/productions/[production_id]"`, com o método PUT, esse endpoint serve para editar as informações de uma produção, porém além disso ele cria os lotes de uma produção, contanto para isso é necessário enviar um JWT do produtor gestor ou supervisor da empresa que criou a produção, e os códigos referentes a cada lotes.

O terceiro endpoint é o `"/productions/batches/[batch_id]"`, com o método PUT. Nele é possível editar e adicionar informações sobre o lote, como GTIN, SSCC, quantidade produzida, quantidade transformada, entre outras. Esse endpoint pode ser chamado por diversos agentes da cadeia, porém cada agente tem um grupo específico de informações que podem adicionar ou editar sobre o lote, e o agente para poder editar precisa ter recebido o lote através da plataforma, que será explicado posteriormente.

Com os lotes criados, o produtor pode fazer download dos códigos QRs e etiquetar seus lotes. Após a etiquetagem, o produtor precisa enviar os lotes para um ou mais transportadores, para isso é necessário que ele faça a leitura do código QR de todos os lotes, através da plataforma, para informar a plataforma que enviou os lotes, como foi dito anteriormente esse processo é um processo simples para o utilizador porém complexo ao sistema, com o objetivo de evitar erros manuais.

O utilizador precisa simplesmente efetuar a leitura do código QR, e a plataforma através dos dados do lote e de quem está efetuando a leitura determina qual é a ação

pretendida, e avalia se é possível efetuar aquela ação no cenário presente.

Para isso foi necessário criar um meio para controlar a rastreabilidade e o estado do lote. O endpoint responsável por isso é o `"/productions/batches/[batch_id]read-qrcode"`, com o método POST. O endpoint executa diversas rotinas durante o seu fluxo de funcionamento, as principais são, para identificar qual ação está sendo efetuada, e se é possível executar essa ação. Na figura 4.13, podemos ver o código responsável por descobrir a ação.

```
392 def get_action(history, role, company_id):
393     if "PRODUCER" in role:
394         return "Produtor enviou"
395     if "SHOPKEEPER" in role:
396         return "Lojista recebeu"
397
398     listfilter = list(filter(lambda element: element["company_id"] == company_id and "tentou" not in element["transition"], history))
399     count = len(listfilter)
400
401     if "TRANSPORTER" in role:
402         if count % 2 == 0:
403             return "Transportador recebeu"
404         else:
405             return "Transportador enviou"
406
407     if "TRANSFORMER" in role:
408         if count % 2 == 0:
409             if len(history) == 0:
410                 return "Transformador enviou"
411             else:
412                 return "Transformador recebeu"
413         else:
414             return "Transformador enviou"
415
416     if "STORER" in role:
417         if count % 2 == 0:
418             return "Armazenista recebeu"
419         else:
420             return "Armazenista enviou"
```

Figura 4.13: Código para determinar a ação.

Com uma ação determinada o sistema avalia se pode efetuar aquela ação naquele estado em que o lote se encontra, para isso foi necessário mapear todos os estados possíveis com todas as ações e suas transições. Para isso foi feito um dicionário em python, que baseado no estado atual do lote e a ação, o sistema efetua ou retorna erro. Na figura 4.14 podemos ver o estado "Transportador enviou", e dependendo da ação pode dar erro ou sucesso. Em caso de erro o sistema registra que deu erro e avisa que não foi possível efetuar a leitura do lote, e em caso de sucesso o sistema registra a rastreabilidade, muda o estado atual do lote e informa que ocorreu tudo certo.

```
133  ✓      "Transportador enviou": {  
134          "Produtor enviou": erro,  
135          "Transportador recebeu": sucesso,  
136          "Transportador enviou": erro,  
137          "Transformador recebeu": sucesso,  
138          "Transformador enviou": erro,  
139          "Armazenista recebeu": sucesso,  
140          "Armazenista enviou": erro,  
141          "Lojista recebeu": sucesso,  
142      },
```

Figura 4.14: Item do dicionário que determina se pode registrar a rastreabilidade.

Neste cenário a plataforma já sabe que os lotes foram enviados pelo produtor, porém a plataforma não sabe quem os recebeu, por isso, assim que os transportadores receberem os lotes, precisam ler os códigos QRs informando para a plataforma quem, quando e onde recebeu aquele lote.

Durante esse processo, o produtor deve preencher as informações da produção e dos lotes, informações que são essenciais para a rastreabilidade, como, onde ocorreu a produção, quando ocorreu, como foi o processo, qual a temperatura ideal para transporte, e também além disso ele pode adicionar videos, imagens ou certificados dos produtos para disponibilizar para o consumidor.

Quando os lotes chegam ao seu destino os transportadores devem ler os códigos e quem recebeu também, assim a plataforma consegue identificar quando o lote foi entregue e quando foi recebido. Este processo é obrigatório em todas as fases da CVAA.

O lote antes de chegar ao retalhista, pode ir para um transformador que pode adicionar mais informações no lote ou até mesmo transformar vários lotes em sublotos de outro produto, para isso foi necessário criar endpoints que fizesse toda a gestão dos sublotos.

Os sublotes são como os lotes, a única diferença é que os sublotes possuem lotes base, os lotes bases são apresentados na rastreabilidade dos sublotes e também possuem sua própria rastreabilidade. Lotes bases são todos os lotes transformados em um sublote, o lote base deixa de poder avançar na cadeia, mas ainda assim o consumidor pode ver a sua rastreabilidade acessando pelo QR Code do sublote criado.

Quando um lote ou sublote chega ao retalhista a url, que apresenta as informações do lote ou sublote, presente no QR Code, passa a ser válida e pode ser acessada por qualquer pessoa. Entende-se que quando o lote chega no retalhista, ele finalizou a cadeia, e portanto suas informações devem ser públicas, para que o cliente possa decidir se deve consumir ou não aquele produto.

4.3.3 Blockchain

Para a implementação de uma Blockchain podem ser adotadas abordagens de Blockchain públicas, privadas ou híbridas. As Blockchain públicas são acessíveis publicamente através da Internet, mantendo os dados, software e o desenvolvimento abertos ao público. Exemplos de blockchains públicas são Bitcoin, Ethereum e Dash. Por outro lado, as Blockchain privadas ao contrário das públicas, dependem de uma unidade central que controla todas as ações dentro da Blockchain, sendo considerado um sistema centralizado. Exemplos de Blockchain privadas são Hyperledger e Corda [21].

O modelo híbrido efetua uma fusão do conceito de Blockchain pública e privada, sendo a participação na rede privada, ou seja, o acesso aos recursos da rede é controlado por uma ou mais entidades, mas está acessível ao público explorar tudo o que acontece na Blockchain, bloco por bloco.

O uso da tecnologia de Blockchain apresenta como principal desvantagem a exigência de uma grande capacidade de processamento ou de uma rede capaz de lidar com grandes volumes de dados, devido a envolver procedimentos criptográficos. Se o sistema tem muitos acessos ou com muitos nós, dependendo de como a cadeia de blocos é estruturada, a capacidade de resposta pode ser longa e não ser adequada aos requisitos do projeto.

Na plataforma Biotrace foi explorada a abordagem de Blockchain privada, uma vez que a CVAA do país pode ter um órgão central que controla todas as ações dentro da blockchain. Além disso, nesta abordagem não é necessário um grande processamento de dados, como em uma Blockchain pública, pois o sistema não precisa de validação e consenso entre os nós.

Deste modo, além do estudo da integração de uma Blockchain descentralizada, foi estudada uma solução centralizada baseada em Ledger Database, que segue o princípio de segurança e imutabilidade dos recursos de uma Blockchain, mas com uma forma de operação mais simples e centralizado. Na prática uma base de dados Ledger é uma base de dados NoSQL que funciona em modo de imutabilidade, ou seja, é garantido de forma transparente que os dados não são alterados.

A plataforma Biotrace tem o objetivo de registrar todas as mudanças completas ao longo do tempo e garantir que essas mudanças não foram adulteradas. O primeiro requer o armazenamento de todas as alterações em um registo ao longo do tempo, e o segundo geralmente é implementado com uma combinação de auditoria de banco de dados e controles de segurança, mas não leva a uma verificação fácil.

Deste modo, foi feita uma biblioteca com diversas funcionalidades como prova de conceito de uma solução de bases de dados Ledger, como a BigChainDB, de modo a tentar garantir, da melhor forma, os requisitos de rastreabilidade e de imutabilidade dos dados e processos da cadeia de valor alimentar.

A garantia de imutabilidade de uma base de dados em modo Ledger é conseguido quando um dado é gravado, em que todo o conjunto de dados do registo é gravado numa hash. Ou seja, todos os dados de um documento são emparelhados e é criada uma hash única. Além da hash do registo atual, é adicionada uma hash do documento anterior criando uma sequência com um princípio de funcionamento idêntico à Blockchain. Se algum dado for alterado a hash deixa de ser válida, funcionando segundo um princípio de imutabilidade e de verificação dessa imutabilidade.

Na plataforma Biotrace todas as informações são armazenadas no modo somente acréscimo. A suposição é que, uma vez armazenadas por um motivo, as informações não seriam

alteradas posteriormente por uma entidade não autorizada. A verificação criptográfica do histórico de alterações ajuda a determinar a violação de informações por exemplo, após a leitura do documento.

De forma resumida a biblioteca implementada funciona como um histórico de alterações criptograficamente verificável de documentos armazenados no MongoDB. Ela fornece funções para armazenar documentos em uma coleção junto com a prova de violação criptográfica, atualmente, um SHA256 sobre o documento e seu histórico, juntamente com os metadados do documento criados pela biblioteca. Quaisquer atualizações em um documento existente resultam automaticamente em uma nova cópia do documento que está sendo inserido. As versões atual e anterior de um documento são armazenadas em uma coleção separada. O nome desta coleção separada é "collectionname_history". Qualquer inserção ou atualização nesta coleção de histórico é gerenciada pela biblioteca.

Como este foi o ultimo modulo implementado do sistema, para que a integração, com o restante já feito, não tivesse nenhum problema, foi utilizado um padrão de projeto, conhecido como decorator. Esse padrão permite adicionar um comportamento a um objeto já existente do sistema, ou seja, agrega dinamicamente funcionalidades adicionais a um objeto sem comprometer seu comportamento antigo.

Na figura 4.15 é apresentado o decorator responsável por modificar uma funcionalidade já existente do sistema, a funcionalidade de armazenar um documento no MongoDB, pois agora é necessário armazenar o documento junto com a prova de violação criptográfica, e também armazenar o documento em uma outra coleção separada que contém histórico de todas as alterações feitas.

Na figura 4.16 é apresentado o decorator responsável por modificar a funcionalidade de editar um documento no MongoDB, pois é necessário agora registrar todas as alterações, bem como atualizar a prova de violação criptográfica.

Na figura 4.17 é apresentado o decorator responsável por modificar as funcionalidades de buscar um documento no MongoDB, pois agora é necessário que o sistema verifique a prova de violação criptográfica antes de retornar qualquer informação, evitando assim disponibilizar dados fraudulentos. Essa função verifica o histórico completo de alterações,

verificando a prova de violação criptográfica de cada documento armazenado, ou seja, cada bloco da Blockchain. Só quando todas as provas de violações forem verificadas e validadas, o sistema apresenta os dados, qualquer tipo de violação, o sistema não disponibilizará os dados ao usuário.

```
67 def insert_one_ledger():
68     def wrapper(fn):
69         @wraps(fn)
70         def decorator(*args, **kwargs):
71             history_col_name = kwargs.get("collection_name") + "_history"
72             history_col = args[0].db[history_col_name]
73
74             random_bytes = urandom(32)
75             doc_id = str(ObjectId())
76
77             random_bytes = urandom(32)
78             nonce = b64encode(random_bytes).decode("utf-8")
79
80             if "_id" not in kwargs.get("element").keys():
81                 kwargs.get("element")["_id"] = doc_id
82
83             kwargs.get("element")["_ledgermeta"] = {}
84             kwargs.get("element")["_ledgermeta"]["prev_hash"] = ""
85             kwargs.get("element")["_ledgermeta"]["orig_id"] = kwargs.get("element")["_id"]
86             kwargs.get("element")["_ledgermeta"]["nonce"] = nonce
87             kwargs.get("element")["_ledgermeta"]["op"] = "INSERTONE"
88             kwargs.get("element")["_ledgermeta"]["seqno"] = 1
89             del kwargs.get("element")["_id"]
90
91             json_str = json.dumps(kwargs.get("element"), sort_keys=True, default=str).encode("utf-8")
92             digestsha256 = hashlib.sha256(json_str)
93
94
95             kwargs.get("element")["_ledgermeta"]["hash"] = digestsha256.hexdigest()
96             kwargs.get("element")["_id"] = kwargs.get("element")["_ledgermeta"]["orig_id"]
97
98             with args[0].client.start_session() as s:
99                 s.start_transaction()
100                 response = fn(*args, **kwargs)
101
102                 del kwargs.get("element")["_id"]
103                 history_col.insert_one(kwargs.get("element"))
104                 s.commit_transaction()
105
106             return response
107         return decorator
108     return wrapper
```

Figura 4.15: Decorator para salvar um ledger.

```

111 def update_one_ledger():
112     def wrapper(fn):
113         @wraps(fn)
114         def decorator(*args, **kwargs):
115
116             history_col_name = kwargs.get("collection_name") + "_history"
117             history_col = args[0].db[history_col_name]
118             col = args[0].db[kwargs.get("collection_name")]
119
120             random_bytes = urandom(32)
121             nonce = b64encode(random_bytes).decode("utf-8")
122
123             query = {"_ledgermeta.orig_id": kwargs.get("id")}
124
125             kwargs.get("element")["_ledgermeta.nonce"] = nonce
126             kwargs.get("element")["_ledgermeta.op"] = "UPDATEONE"
127             kwargs.get("element")["$inc"] = {"_ledgermeta.seqno": 1}
128
129             with args[0].client.start_session() as s:
130                 s.start_transaction()      jp-lourenco, a month ago • save
131
132                 response_fn = fn(*args, **kwargs)
133
134                 item = col.find_one(query)
135
136                 item["_ledgermeta"]["prev_hash"] = item["_ledgermeta"]["hash"]
137
138                 del item["_id"]
139                 del item["_ledgermeta"]["hash"]
140                 updated = item["updated"]
141                 del item["updated"]
142                 created = item["created"]
143                 del item["created"]
144
145                 json_str = json.dumps(item, sort_keys=True, default=str).encode("utf-8")
146                 digestsha256 = hashlib.sha256(json_str)
147                 item["_ledgermeta"]["hash"] = digestsha256.hexdigest()
148                 item["updated"] = updated
149                 item["created"] = created
150                 response = history_col.insert_one(item)
151
152                 query_history = {"_ledgermeta.orig_id": response.inserted_id}
153
154                 new_element = {"$set": {}}
155                 new_element["$set"]["_ledgermeta.prev_hash"] = item["_ledgermeta"]["prev_hash"]
156                 new_element["$set"]["_ledgermeta.hash"] = item["_ledgermeta"]["hash"]
157                 history_col.update_one(query_history, new_element)
158                 col.update_one(query, new_element)
159                 s.commit_transaction()
160
161             return response_fn
162         return decorator
163     return wrapper

```

Figura 4.16: Decorator para editar um ledger.


```

14 def verify_ledger():
15     def wrapper(fn):
16         @wraps(fn)
17         def decorator(*args, **kwargs):
18
19             items_found = False
20
21             history_col_name = kwargs.get("collection_name") + "_history"
22             history_col = args[0].db[history_col_name]
23
24             if kwargs.get("criteria"):
25                 items = history_col.find(kwargs.get("criteria")).sort("_ledgermeta.seqno", 1)
26             elif kwargs.get("id"):
27                 items = history_col.find({"_id":kwargs.get("id")}).sort("_ledgermeta.seqno", 1)
28             else:
29                 items = history_col.find({}).sort("_ledgermeta.seqno", 1)
30
31             if not items:
32                 return None
33
34             response = fn(*args, **kwargs)
35
36             if verify(items) == True and verify(response) == True:
37                 return response
38             else:
39                 raise InvalidUsage("Alguma coisa deu errado! Tente novamente mais tarde!", status_code=500)
40
41         return decorator
42     return wrapper
43
44 def verify(items):
45     if type(items) is dict:
46         items = [copy.deepcopy(items)]
47     else:
48         items = copy.deepcopy(items)
49     for item in items:
50         if "_ledgermeta" in item:
51             items_found = True
52             orig_hash = item["_ledgermeta"]["hash"]
53             del item["_id"]
54             del item["_ledgermeta"]["hash"]
55             del item["updated"]
56             del item["created"]
57
58             json_str = json.dumps(item, sort_keys=True, default=str).encode("utf-8")
59             computed_hash = hashlib.sha256(json_str).hexdigest()
60             if ( orig_hash != computed_hash):
61                 return False
62         else:
63             continue
64     return True

```

Figura 4.17: Decorator para consultar um ledger.

4.4 Implementação da arquitetura

Na figura 4.18 é apresentada a arquitetura implementada para que a plataforma cumpra com os seus requisitos, todos os serviços foram containerizados com Docker provendo assim o isolamento entre os serviços e consequentemente maior segurança.

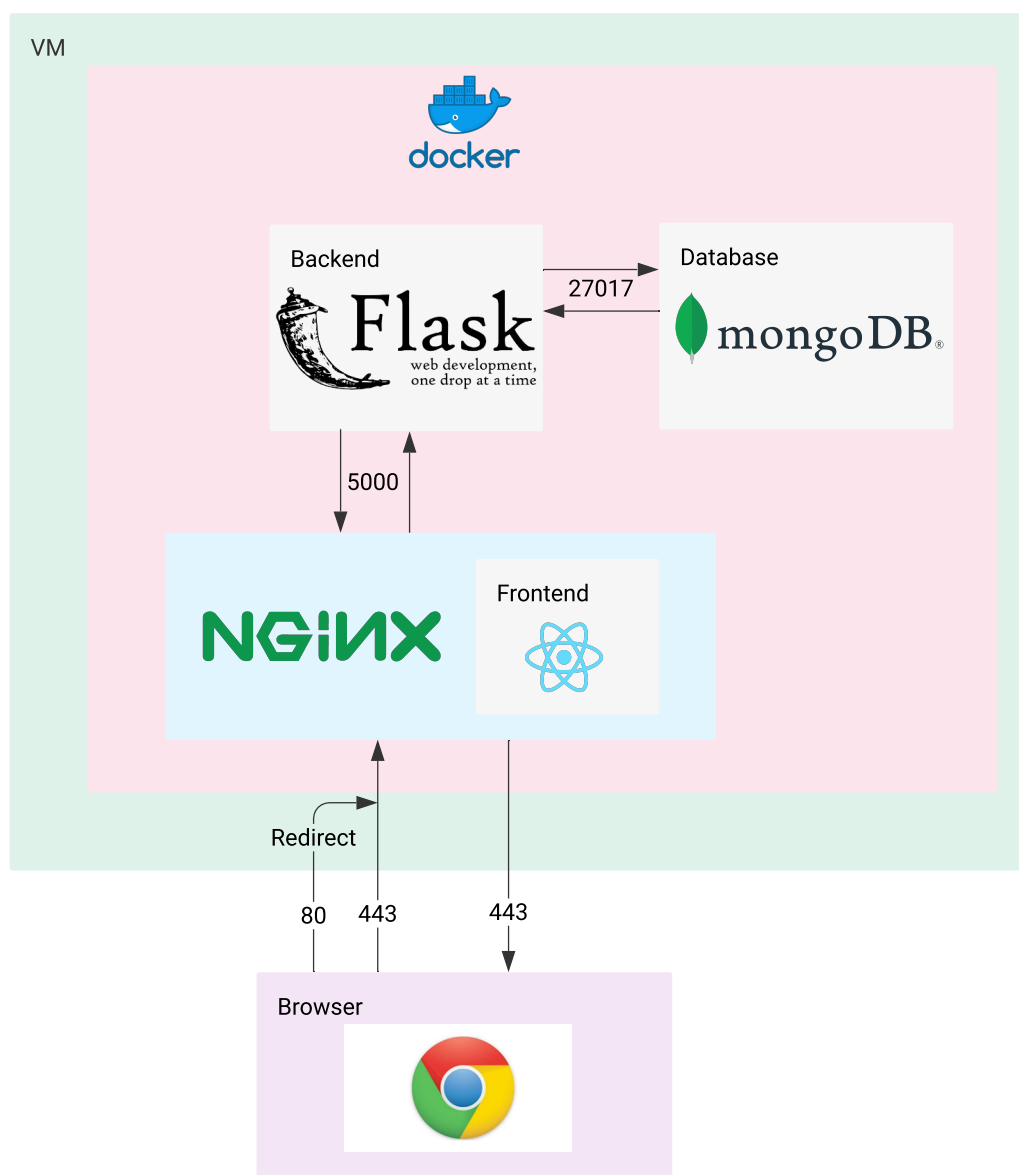


Figura 4.18: Arquitetura da plataforma.

Para cumprir com os requisitos da plataforma foram containerizados e integrados quatro serviços, sendo eles:

- Nginx, serviço capaz de gerenciar as requisições que funciona como a entrada da arquitetura;
- React, serviço que fornece toda a interface gráfica da plataforma e faz comunicação com a API;
- Flask, serviço que contém toda as regras de negócio e faz comunicação com o banco de dados;
- MongoDB, serviço que armazena todas as informações da plataforma.

4.5 Disponibilização da solução

Para disponibilizar a solução foi necessário integrar todas as partes da plataforma, camada visual, camada de negócios e camada de dados, e para isso foi utilizado o Docker e o Nginx.

O Docker foi utilizado para possibilitar que todas as partes da plataforma fossem containerizada, conseguindo assim, garantir que todas as partes funcionem em todos os ambientes e compartilhem o mesmo sistema operacional. Para isso foi necessário fazer alguns arquivos de configuração dizendo todos detalhes para criar um contêiner com nosso programa. Na figura 4.19 é apresentado o arquivo de configuração da camada visual com nginx. Nesse arquivo é indicado onde esta a aplicação, quais são as dependências, qual comando para instalá-las, em qual porta a aplicação vai ser disponibilizada, e outros.

Na figura 4.20 é apresentado o arquivo de configuração do flask, neste arquivo é apontado tudo para criar um container da API, desde da versão do python até às dependências que são utilizadas no projeto.

Para orquestrar todos os containers foi utilizada uma ferramenta da plataforma Docker, o Docker Compose. Na figura 4.21 é apresentado a configuração da infraestrutura da plataforma no Docker Compose.

```

1  # React build
2  FROM node:13.12.0-alpine as build
3  WORKDIR /frontend
4  ENV PATH /frontend/node_modules/.bin:$PATH
5  COPY package.json ./
6  COPY package-lock.json ./
7  RUN npm ci --silent
8  RUN npm install react-scripts@4.0.3 -g --silent
9  COPY . ./
10 RUN npm run build
11
12 # Nginx build
13 FROM nginx:stable-alpine
14 COPY --from=build /frontend/build /usr/share/nginx/html
15 COPY nginx/nginx.conf /etc/nginx/nginx.conf
16 EXPOSE 80 443
17 CMD ["nginx", "-g", "daemon off;"]

```

Figura 4.19: Dockerfile React.

```

1  FROM python:3.9.0-slim-buster
2  RUN mkdir -p /usr/src/app
3  WORKDIR /usr/src/app
4  ENV PYTHONDONTWRITEBYTECODE 1
5  ENV PYTHONUNBUFFERED 1
6  COPY ./requirements.txt /usr/src/app
7  RUN pip install --no-cache-dir -r requirements.txt
8  COPY ./src /usr/src/app
9  COPY ./main.py /usr/src/app
10 COPY ./setup.py /usr/src/app
11 CMD ["python", "main.py"]

```

Figura 4.20: Dockerfile Flask.

```
1  version: '3.7'
2
3  services:
4
5      biotrace-frontend-prod:
6          container_name: biotrace-frontend-prod
7          build:
8              context: .
9              dockerfile: Dockerfile.prod
10         ports:
11             - '80:80'
12             - '443:443'
13         volumes:
14             - /etc/data/cert/bioma.estig.ipb.pt:/etc/nginx/certs
15
16         backend:
17             restart: always
18             container_name: backend
19             build:
20                 context: ../Backend
21                 dockerfile: Dockerfile
22             expose:
23                 - 5000
24             volumes:
25                 - ../Backend:/usr/src/app
26
27         db:
28             image: mongo:latest
29             hostname: biotrace
30             expose:
31                 - 27017
32             command: [--auth]
33             restart: always
```

Figura 4.21: Docker Compose.

No Nginx foi configurado o React com um certificado SSL, permitindo utilizar o protocolo HTTPS para estabelecer uma comunicação segura com o site. Além do React, também foi configurado um proxy-reverse que redireciona uma rota específica do React, no caso `/api/v1`, para o Flask, conseguindo assim reaproveitar o certificado SSL do React no Flask. O site implementado está disponível no domínio <https://bioma.estig.ipb.pt/>.

Capítulo 5

Conclusões e Trabalhos Futuros

A proposta desse capítulo é sintetizar e proporcionar uma visão geral do desenvolvimento do trabalho, de modo a analisar se a solução proposta engloba todos os requisitos levantados no início do trabalho.

A preocupação da população em geral com os alimentos consumidos está em constante crescimento, e um meio de ajudar e suprir a necessidade dos consumidores é a utilização de aplicações de rastreabilidade, sendo uma realidade que já está inserida no contexto social, porém diversas soluções ainda se encontram desconectadas ao longo da cadeia e muitas vezes não disponibilizam as informações de forma clara ao usuário.

Conectar as fases da cadeia fornece diversos benefícios à cadeia. Aos gestores traz um maior controle da sua produção e aos consumidores traz uma maior transparência em relação ao produto que foi adquirido e também gera uma relação de confiança com a marca. Além disso, um sistema centralizado se torna um meio viável para uma boa integração da cadeia de valor e garante uma boa rastreabilidade.

O objetivo proposto de unir as diversas fases pôde ser alcançado com um sistema centralizado com padrões bem definidos e amplamente utilizados, como o código de barras bidimensional, que foi utilizado para registrar todas as conexões entre as diversas fases que o lote percorre dentro da plataforma, e código de barras tradicional no padrão GS1, que permite a interoperabilidade e a padronização entre os sistemas de rastreabilidade.

Para realizar e alcançar o desenvolvimento de uma plataforma de rastreabilidade que

pudesse integrar todas as fases e beneficiar toda a cadeia produtiva, foi necessário desenvolver e configurar diversos serviços, para isso foi necessário analisar várias abordagens, definir e desenvolver uma arquitetura com foco na escalabilidade, segurança e flexibilidade.

Para garantir a integridade dos dados foi desenvolvido um histórico de alterações verificável através de criptografia no MongoDB, aplicando conceitos da tecnologia blockchain. Antes de disponibilizar os dados da rastreabilidade para o consumidor, foi feita uma rotina que valida todas as alterações feitas no lote e apresenta as informações para o consumidor apenas se as alterações foram feitas através da API Biotrace. Isso garante que a informação que o consumidor verá sempre será íntegra.

Neste sentido o presente trabalho desenvolveu uma arquitetura simples e efetiva usando tecnologias atuais do mercado. As interfaces foram desenvolvidas com React. A API foi feita com Flask. O banco de dados utilizado, foi o MongoDB. Todos os serviços foram containerizados com Docker, e foram disponibilizados com Nginx com certificado SSL nas rotas.

A solução desenvolvida pode ser acessada por qualquer celular ou computador, que tenha acesso a internet, e poderá ser útil tanto para os agentes internos da cadeia, quanto para os consumidores, por apresentar e disponibilizar informações referentes à história de um produto e permitir os gestores a controlar a sua cadeia de forma mais produtiva.

Trabalho futuros podem abordar conceitos de blockchain descentralizada, replicando os históricos de alterações criptograficamente verificável para diversas máquinas distintas. Neste caso é importante o sistema aplicar conceitos de consenso para ter uma forma de controlar as alterações na rede. Além disso, por se tratar de uma arquitetura containerizada é possível utilizar a tecnologia kubernetes para replicar containers entre diferentes nós e garantir alta disponibilidade. Em relação a gestão dos dados seria interessante ter uma base de dados relacional com dados estruturados para conseguir gerar relatórios de análise e de inspeção.

De modo a melhorar o desempenho da plataforma no celular fica proposto como trabalho futuro a implementação de uma aplicação móvel nativa. Também fica proposto

o desenvolvimento de funcionalidades, que melhorem a cadeia produtiva em qualquer aspecto, como obter feedback do consumidor e adicionar sensores de localização.

Bibliografia

- [1] C. E. C. de Aguiar Freire e C. de Lara Shecaira, «A importância da rastreabilidade dos alimentos de origem animal frente aos surtos alimentares,» *Pubvet*, vol. 14, n.º 11, 2020. DOI: 10.31533/pubvet.v14n11a682.1-8. URL: <https://www.pubvet.com.br/artigo/7248/a-importacircncia-da-rastreabilidade-dos-alimentos-de-origem-animal-frente-aos-surtos-alimentares-revisatildeo>.
- [2] J. C. de Andrade, «Percepção do consumidor frente aos riscos associados aos alimentos, sua segurança e rastreabilidade,» *Brazilian Journal of Food Technology*, vol. 16, n.º 3, 2013. DOI: 10.1590/s1981-67232013005000023. URL: <https://www.scielo.br/j/bjft/a/G6NXXPM79sRRX78HKm6xb8t/?format=pdf&lang=pt>.
- [3] R. T. M. Machado, «Sinais de qualidade e rastreabilidade de alimentos: uma visão sistêmica,» *Organizações Rurais e Agroindustriais*, vol. 7, n.º 2, 2005. URL: <https://ageconsearch.umn.edu/record/43985/>.
- [4] J. Rauta, L. J. Paetzold e C. A. Winck, «Rastreabilidade na cadeia produtiva do leite como vantagem competitiva,» *Revista em Agronegocio e Meio Ambiente*, vol. 10, n.º 2, 2017. DOI: 10.17765/2176-9168.2017v10n2p459-474. URL: <https://www.scopus.com/authid/detail.uri?authorId=57194604114>.
- [5] M. Ribeiro, A. Ramos, V. Ferreira, J. Cunha e C. Fante, «Tecnologias de rastreabilidade, segurança e controle de resíduos de agrotóxicos na cadeia produtiva de alimentos de origem vegetal: um estudo de revisão,» *Research, Society and Development*, vol. 9, e5291210780, dez. de 2020. DOI: 10.33448/rsd-v9i12.10780.

- [6] X. Weng e S. Neethirajan, «Ensuring food safety: Quality monitoring using microfluidics,» *Trends in food science and technology*, vol. 65, n.º 2, pp. 10–22, 2017. DOI: <https://doi.org/10.1016/j.tifs.2017.04.015>.
- [7] D. B. F. Furlan, «Viabilidade de coleta de dados para implantação de sistema de rastreabilidade na cadeia produtiva de ovos,» tese de mestrado, 2019. URL: <https://repositorio.unesp.br/handle/11449/181243>.
- [8] E. D. Boeck, A. V. Mortier, L. Jacxsens, L. Dequidt e P. Vlerick, «Towards an extended food safety culture model: Studying the moderating role of burnout and jobstress the mediating role of food safety knowledge and motivation in the relation between food safety climate and food safety behavior,» *Trends in Food Science and Technology*, vol. 62, pp. 202–214, 2017. DOI: 10.1016/j.tifs.2017.01.004.
- [9] M. C. Ribeiro, A. M. Ramos, V. A. Ferreira, J. R. da Cunha e C. A. Fante, «Tecnologias de rastreabilidade segurança e controle de resíduos de agrotóxicos na cadeia produtiva de alimentos de origem vegetal: um estudo de revisão,» *Research Society and Development*, vol. 9, n.º 12, 2020. DOI: 10.33448/rsd-v9i12.10780.
- [10] J. Rauta, L. J. Paetzold e C. A. Winck, «A Rastreabilidade na cadeia produtiva do leite como vantagem competitiva,» *Revista em Agronegócio e meio Ambiente*, vol. 10, n.º 2, pp. 459–474, 2017. DOI: 10.17765/2176-9168.2017v10n2p459-474.
- [11] J. Kovacart, H. den Bakker, L. M. Carroll e M. Wiedmann, «Precision food safety: a systems approach to food safety facilitated by genomics tools,» *TrAC Trends in Analytical Chemistry*, vol. 96, pp. 52–61, 2017. DOI: <https://doi.org/10.1016/j.trac.2017.06.001>.
- [12] J. R. Franco, L. P. Petroski, A. M. Guimarães, M. S. M. G. Vaz e R. Falate, «Desenvolvimento de sistema para rastreabilidade de alimentos orgânicos aplicando diretrizes IHC,» *Anais SULCOMP*, vol. 8, pp. 1–9, 2017, ISSN: 2359-2656.
- [13] L. CHAPAVAL e F. S. F. ALVES, *Rastreabilidade na produção de leite de cabra: diferencial para um agronegócio sustentável*, Online; accessed 04 April 2021, 2008.

URL: <http://www.infoteca.cnptia.embrapa.br/infoteca/handle/doc/533940>.

- [14] U. Europeia, *The Rapid Alert System for Food and Feed. 2018 Annual Report*, Online; accessed 06 April 2021, 2018. URL: <http://www.foodlaw.reading.ac.uk/pdf/2019-Commission-RASFF-Annual-Report-2018.pdf>.
- [15] P. O. Jostein Storøy Maitri Thakur, «The TraceFood Framework – Principles and guidelines for implementing traceability in food value chains,» vol. 115, pp. 41–48, 2013. DOI: <https://doi.org/10.1016/j.jfoodeng.2012.09.018>.
- [16] J.-P. Qian, X.-T. Yang, X.-M. Wu, L. Zhao, B.-L. Fan e B. Xing, «A traceability system incorporating 2D barcode and RFID technology for wheat flour mills,» *Computers and Electronics in Agriculture*, vol. 89, pp. 76–85, 2012. DOI: <https://doi.org/10.1016/j.compag.2012.08.004>.
- [17] M. Li, J.-P. Qian, X.-T. Yang, C.-H. Sun e Z.-T. Ji, «A PDA-based record-keeping and decision-support system for traceability in cucumber production,» *Computers and Electronics in Agriculture*, vol. 70, pp. 69–77, 2010. DOI: <https://doi.org/10.1016/j.compag.2009.09.009>.
- [18] Y. Wang, J. H. Han e P. Beynon-Davies, «Understanding blockchain technology for future supply chains: a systematic literature review and research agenda,» *Supply Chain Management: An International Journal*, vol. 24, dez. de 2018. DOI: [10.1108/SCM-03-2018-0148](https://doi.org/10.1108/SCM-03-2018-0148).
- [19] F. A. G., O. R. de S., Q. C. M. e L. C. P., «Potential of Blockchain Application for Sustainable Food Chain Traceability: a prospective study,» *Cadernos De Prospecção*, vol. 14, n.º 3, 2021. DOI: <https://doi.org/10.9771/cp.v14i3.38358>.
- [20] A. ABADI M.; BALL, *Palm oil plantations are under pressure to prove they're sustainable: but the process can have devastating effects on local communities and the plantations themselves*. Online: accessed 07 April 2021, 2020. URL: <https://>

www.businessinsider.com/palm-oil-plantationsustainable-environment-guatemala-2020-4.

- [21] K. Demestichas, N. Peppes, T. Alexakis e E. Adamopoulou, «Blockchain in Agriculture Traceability Systems: A Review,» *Applied Sciences*, vol. 10, p. 4113, jun. de 2020. DOI: 10.3390/app10124113.
- [22] A. Lessak, R. Dias e I. Frey, «BLOCKCHAIN: PROSPECÇÃO TECNOLÓGICA EM BASES DE PATENTES,» *Cadernos de Prospecção*, vol. 11, p. 876, set. de 2018. DOI: 10.9771/cp.v11i3.27006.
- [23] R. Mendonça, O. Gomes, P. Pereira, A. Borges e J. Miranda Nacif, «Utilização de Blockchain na Rastreabilidade da Cadeia Produtiva do Leite,» dez. de 2020, pp. 55–60. DOI: 10.5753/wblockchain.2020.12433.
- [24] F. S. HORIUCHI, *Rastreabilidade de um modelo de cadeia produtiva agrícola generalizado em uma rede blockchain*. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação), Universidade de Londrina, Londrina, PR, Brazil., 2018.
- [25] R. ANZALONE, *IBM Blockchain Is Growing In The Food Industry During Covid-19*. Online: accessed 07 April 2021, 2020. URL: https://www.forbes.com/sites/robertanzalone/2020/06/04/ibm-blockchaintechnology-is-growing-in-the-food-industry-during-covid-19/?_lrsc=b61ca797-e00b-4255-b515-.
- [26] S. Saberi, M. Kouhizadeh, J. Sarkis e L. Shen, «RASTREABILIDADE DE UM MODELO DE CADEIA PRODUTIVA AGRÍCOLA GENERALIZADO EM UMA REDE BLOCKCHAIN,» *International Journal of Production Research*, vol. 57, n.º 7, pp. 2117–2135, 2018. DOI: <https://doi.org/10.1080/00207543.2018.1533261>.
- [27] H. L. P. SILVA, *Método de rastreabilidade de produtos agrícolas com a utilização de blockchain*. Dissertação (Mestrado em Computação Aplicada), Universidade Estadual de Ponta Grossa, Ponta Grossa, PR, Brazil., 2020.

- [28] B. Smith, «Developing sustainable food supply chains,» *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, vol. 363, pp. 849–61, mar. de 2008. DOI: 10.1098/rstb.2007.2187.
- [29] J. Galvez, J. Mejuto e J. Simal-Gandara, «Future challenges on the use of block-chain for food traceability analysis,» *TrAC Trends in Analytical Chemistry*, vol. 107, ago. de 2018. DOI: 10.1016/j.trac.2018.08.011.
- [30] M. R. Pincheira Caro, M. S. Ali, M. Vecchio e R. Giaffreda, «Blockchain-based traceability in Agri-Food supply chain management: A practical implementation,» mai. de 2018, pp. 1–4. DOI: 10.1109/IOT-TUSCANY.2018.8373021.
- [31] *Best Blockchain Supply Chain Software*, Online; accessed 06 April 2021, 2021. URL: <https://sourceforge.net/software/blockchain-supply-chain/>.
- [32] C. A. R. GOUVEIA, «A utilização dos códigos de barras ao longo da cadeia de abastecimento - estudo de caso da empresa Luís Simões.,» tese de mestrado, 2014. URL: <https://www.repository.utl.pt/handle/10400.5/7626>.
- [33] J. F. P. LOURENÇO, *Um modelo de rastreabilidade usando standards GS1: caso de estudo aplicado aos produtos horto frutícolas em Portugal*. Dissertação (Mestrado em Engenharia e Gestão Industrial) - Universidade Nova de Lisboa, Lisboa, Portugal., 2015.
- [34] J. C. CARVALHO, *Logística e Gestão da Cadeia de Abastecimento*. 2^a ed. Edições Sílabo, 2010, ISBN: 9789895610853.
- [35] G. White, G. Gardiner, G. Prabhakar e A. Abd Razak, «A Comparison of Bar-coding and RFID Technologies in Practice,» *Journal of Information, Information Technology and Organizations*, vol. 2, jan. de 2007. DOI: 10.28945/142.
- [36] C. V. lattes Giordano, *Um estudo sobre o impacto de tecnologias emergentes: o caso das etiquetas de rádio frequência na gestão da cadeia de suprimentos*. Dissertação (Mestrado em Administração) - Pontifícia Universidade Católica de São Paulo, São Paulo, Brazil., 2007.

- [37] R. Wazlawick, *Engenharia de software: Conceitos e práticas*. Elsevier Editora Ltda., 2019, ISBN: 9788535292732. URL: <https://books.google.com.br/books?id=d1qnDwAAQBAJ>.
- [38] S. G. B. Viana, «Implementação da metodologia ágil Scrum numa empresa do setor da construção,» tese de mestrado, 2017. URL: <http://hdl.handle.net/1822/49541>.
- [39] *GS1 Portugal*, Online; accessed 09 April 2021, 2021. URL: <https://gs1pt.org/>.
- [40] R. Wazlawick, *Análise e Projeto de Sistemas de Informação Orientados*. Elsevier Brasil, 2011, ISBN: 9788535211177. URL: <https://books.google.pt/books?id=Cf6tE2zISf0C>.
- [41] N. SulaimanSharifah, S. Syed e A. Ahmad, «Logical Approach: Consistency Rules between Activity Diagram and Class Diagram,» *International Journal on Advanced Science, Engineering and Information Technology*, vol. 9, n.º 2, 2019, ISSN: 2088-5334. DOI: 10.18517/ijaseit.9.1.7581.
- [42] *Glossário de Termos React*, Online; accessed 04 April 2021, 2021. URL: <https://pt-br.reactjs.org/docs/glossary.html>.
- [43] A. Kumar e R. K. Singh, «COMPARATIVE ANALYSIS OF ANGULARJS AND REACTJS,» *International Journal of Latest Trends in Engineering and Technology*, vol. 9, n.º 4, 2016. DOI: 10.21172/1.74.030.
- [44] E. Susanti e E. Mailoa, «RESTful API Implementation in Making a Master Data Planogram Using the Flask Framework,» *Journal of Information Technology and Computer Science*, vol. 5, n.º 3, 2020. DOI: 10.25126/jitecs.202053189.
- [45] Z. Lei e H. Zhou, «Cost-Effective Server-side Re-deployment for Web-based Online Laboratories Using NGINX Reverse Proxy,» *IFAC-PapersOnLine*, vol. 53, n.º 2, 2020. DOI: 10.1016/j.ifacol.2020.12.1748.

- [46] A. M. Potdar, N. D G, S. Kengond e M. M. Mulla, «Performance Evaluation of Docker Container and Virtual Machine,» *Procedia Computer Science*, vol. 171, pp. 1419–1428, 2020, Third International Conference on Computing and Network Communications (CoCoNet’19), ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2020.04.152>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050920311315>.

Apêndice A

User Stories

Neste capítulo apresenta todos os User Stories feitos na fase de levantamento de requisitos.

ID	Nome	Prioridade(1-5)	Descrição
USUNR01	Solicitar acesso para uma nova empresa	1	Pretendo solicitar acesso à plataforma para uma nova empresa
USUNR02	Ler QR Code de um alimento	1	Pretendo ler QR Code de um alimento e poder ver as suas informações de rastreabilidade

Tabela A.1: User Stories (Não registrado/ Consumidor)

ID	Nome	Prioridade(1-5)	Descrição
USUR01	Iniciar sessão	1	Pretendo iniciar sessão na plataforma
USUR02	Encerrar sessão	1	Pretendo encerrar sessão na plataforma
Continua na próxima página			

Tabela A.2 – Continuação da página anterior

ID	Nome	Prioridade(1-5)	Descrição
USUR03	Recuperar palavra-passe	4	Pretendo recuperar a minha palavra-passe
USUR04	Atualizar perfil	3	Pretendo atualizar meu perfil

Tabela A.2: User Stories (Utilizadores registados)

ID	Nome	Prioridade(1-5)	Descrição
USADMB01	Gerir empresas	2	Pretendo adicionar, visualizar, editar e remover empresas
USADMB02	Gerir produtos	2	Pretendo adicionar, visualizar, editar e remover produtos
USADMB03	Gerir utilizadores	2	Pretendo adicionar, visualizar,editar e remover utilizadores

Tabela A.3: User Stories (Administrador)

ID	Nome	Prioridade(1-5)	Descrição
USPG01	Gerir produções	1	Pretendo adicionar, visualizar, remover e editar produções
Continua na próxima página			

Tabela A.4 – Continuação da página anterior

ID	Nome	Prioridade(1-5)	Descrição
USPG02	Gerir lotes	1	Pretendo adicionar, visualizar, remover e editar lotes
USPG03	Visualizar sub lotes	3	Pretendo visualizar os sub lotes
USPG04	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USPG05	Ler QR Code de um lote	2	Pretendo ler um QR Code e registar que enviei o lote
USPG06	Gerir supervisores	4	Pretendo adicionar, visualizar, remover e editar supervisores
USPG07	Gerir funcionários	4	Pretendo adicionar, visualizar, remover e editar funcionários

Tabela A.4: User Stories (Produtor Gestor)

ID	Nome	Prioridade(1-5)	Descrição
USPS01	Gerir produções	1	Pretendo adicionar, visualizar, remover e editar produções
USPS02	Gerir lotes	1	Pretendo adicionar, visualizar, remover e editar lotes
USPS03	Visualizar sub lotes	3	Pretendo visualizar os sub lotes
Continua na próxima página			

Tabela A.5 – Continuação da página anterior

ID	Nome	Prioridade(1-5)	Descrição
USPS04	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USPS05	Ler QR Code de um lote	2	Pretendo ler um QR Code e registrar que envie o lote

Tabela A.5: User Stories (Produtor Supervisor)

ID	Nome	Prioridade(1-5)	Descrição
USPF01	Visualizar produções	1	Pretendo visualizar produções
USPF02	Visualizar lotes	1	Pretendo visualizar lotes
USPF03	Visualizar sub lotes	3	Pretendo visualizar os sub lotes
USPF04	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USPF05	Ler QR Code de um lote	2	Pretendo ler um QR Code e registrar que envie o lote

Tabela A.6: User Stories (Produtor Funcionário)

ID	Nome	Prioridade(1-5)	Descrição
USTG01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa participou
Continua na próxima página			

Tabela A.7 – Continuação da página anterior

ID	Nome	Prioridade(1-5)	Descrição
USTG02	Visualizar sub lotes	3	Pretendo visualizar os sub lotes que minha empresa participou
USTG03	Sincronizar dados dos sensores	3	Pretendo sincronizar os dados dos sensores
USTG04	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USTG05	Ler QR Code de um lote	2	Pretendo ler um QR Code
USTG06	Gerir supervisores	4	Pretendo adicionar, visualizar, remover e editar supervisores
USTG07	Gerir funcionários	4	Pretendo adicionar, visualizar, remover e editar funcionários
USTG08	Gerir sensores	3	Pretendo adicionar, visualizar, remover e editar sensores

Tabela A.7: User Stories (Transportador Gestor)

ID	Nome	Prioridade(1-5)	Descrição
USTS01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa participou
Continua na próxima página			

Tabela A.8 – Continuação da página anterior

ID	Nome	Prioridade(1-5)	Descrição
USTS02	Visualizar sub lotes	3	Pretendo visualizar os sub lotes que minha empresa participou
USTS03	Sincronizar dados dos sensores	3	Pretendo sincronizar os dados dos sensores
USTS04	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USTS05	Ler QR Code de um lote	2	Pretendo ler um QR Code

Tabela A.8: User Stories (Transportador Supervisor)

ID	Nome	Prioridade(1-5)	Descrição
USTF01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa participou
USTF02	Visualizar sub lotes	3	Pretendo visualizar os sub lotes que minha empresa participou
USTF03	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USTF04	Ler QR Code de um lote	2	Pretendo ler um QR Code

Tabela A.9: User Stories (Transportador Funcionário)

ID	Nome	Prioridade(1-5)	Descrição
USTFG01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa participou
USTFG02	Editar lotes	3	Pretendo editar os lotes que minha empresa participou
USTFG03	Gerir sub lotes	3	Pretendo adicionar, visualizar, editar e remover os sub lotes que minha empresa participou
USTFG04	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USTFG05	Ler QR Code de um lote	2	Pretendo ler um QR Code
USTFG06	Gerir funcionários	4	Pretendo adicionar, visualizar, remover e editar funcionários
USTFG07	Gerir sensores	3	Pretendo adicionar, visualizar, remover e editar sensores

Tabela A.10: User Stories (Transformador Gestor)

ID	Nome	Prioridade(1-5)	Descrição
USTFG01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa participou
Continua na próxima página			

Tabela A.11 – Continuação da página anterior

ID	Nome	Prioridade(1-5)	Descrição
USTFG02	Editar lotes	3	Pretendo editar os lotes que minha empresa participou
USTFG03	Gerir sub lotes	3	Pretendo adicionar, visualizar, editar e remover os sub lotes que minha empresa participou
USTFG04	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USTFG05	Ler QR Code de um lote	2	Pretendo ler um QR Code

Tabela A.11: User Stories (Transformador Supervisor)

ID	Nome	Prioridade(1-5)	Descrição
USTFF01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa participou
USTFF02	Visualizar lotes	3	Pretendo visualizar os lotes que minha empresa participou
USTFF03	Visualizar sub lotes	3	Pretendo visualizar os sub lotes que minha empresa participou
USTFF04	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USTFF05	Ler QR Code de um lote	2	Pretendo ler um QR Code
Continua na próxima página			

Tabela A.12 – Continuação da página anterior

ID	Nome	Prioridade(1-5)	Descrição
-----------	-------------	------------------------	------------------

Tabela A.12: User Stories (Transformador Funcionário)

ID	Nome	Prioridade(1-5)	Descrição
USAG01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa participou
USAG02	Visualizar sub lotes	3	Pretendo visualizar os sub lotes que minha empresa participou
USAG03	Sincronizar dados dos sensores	3	Pretendo sincronizar os dados dos sensores
USAG04	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USAG05	Ler QR Code de um lote	2	Pretendo ler um QR Code
USAG06	Gerir supervisores	4	Pretendo adicionar, visualizar, remover e editar supervisores
USAG07	Gerir funcionários	4	Pretendo adicionar, visualizar, remover e editar funcionários
USAG08	Gerir sensores	3	Pretendo adicionar, visualizar, remover e editar sensores

Tabela A.13: User Stories (Armazenista Gestor)

ID	Nome	Prioridade(1-5)	Descrição
USAS01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa participou
USAS02	Visualizar sub lotes	3	Pretendo visualizar os sub lotes que minha empresa participou
USAS03	Sincronizar dados dos sensores	3	Pretendo sincronizar os dados dos sensores
USAS04	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USAS05	Ler QR Code de um lote	2	Pretendo ler um QR Code

Tabela A.14: User Stories (Armazenista Supervisor)

ID	Nome	Prioridade(1-5)	Descrição
USAF01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa participou
USAF02	Visualizar sub lotes	3	Pretendo visualizar os sub lotes que minha empresa participou
USAF03	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USAF04	Ler QR Code de um lote	2	Pretendo ler um QR Code

Tabela A.15: User Stories (Armazenista Funcionário)

ID	Nome	Prioridade(1-5)	Descrição
USRG01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa recebeu
USRG02	Visualizar sub lotes	3	Pretendo visualizar os sub lotes que minha empresa recebeu
USRG03	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USRG04	Ler QR Code de um lote	2	Pretendo ler um QR Code
USRG05	Gerir supervisores	4	Pretendo adicionar, visualizar, remover e editar supervisores
USRG06	Gerir funcionários	4	Pretendo adicionar, visualizar, remover e editar funcionários

Tabela A.16: User Stories (Retalhista Gestor)

ID	Nome	Prioridade(1-5)	Descrição
USRS01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa recebeu
USRS02	Visualizar sub lotes	3	Pretendo visualizar os sub lotes que minha empresa recebeu
Continua na próxima página			

Tabela A.17 – Continuação da página anterior

ID	Nome	Prioridade(1-5)	Descrição
USRS03	Download de QR Code e Código de barras	2	Pretendo fazer download de QR Code
USRS04	Ler QR Code de um lote	2	Pretendo ler um QR Code

Tabela A.17: User Stories (Retalhista Supervisor)

ID	Nome	Prioridade(1-5)	Descrição
USRF01	Visualizar produções	1	Pretendo visualizar produções com os lotes que a minha empresa recebeu
USRF02	Visualizar sub lotes	3	Pretendo visualizar os sub lotes que minha empresa recebeu
USRF03	Ler QR Code de um lote	2	Pretendo ler um QR Code

Tabela A.18: User Stories (Retalhista Funcionário)