

# Documento da Arquitetura da API Embrapa's viticulture

Versão 1.0

Data: 25/10/2024

## HISTÓRICO DE REVISÕES

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
25/10/2024	1.0	Criação do documento de da API Embrapa's viticulture	Carlos Bezerra (CHRB)
25/10/2024	1.1	Revisão do Documento	Jackson Castro (JC)

# SUMÁRIO

SUMÁRIO	2
<b>1. Introdução</b>	<b>3</b>
1.1 Finalidade	3
1.2 Escopo	3
1.3 Definições, Acrônimos e Abreviações	3
<b>2. Representação da Arquitetura</b>	<b>4</b>
2.1 Framework FastAPI	4
2.2 Representational State Transfer (REST)	4
2.3 JWT	5
2.4 Arquitetura de Solução	6
2.4.1 Nível 1 - Contexto	6
2.4.2 Nível 2 - Container	6
2.4.3 Nível 3 - Componentes	7
2.4.4 Nível 4 - Código	8
<b>3. Metas e Restrições de Arquitetura</b>	<b>8</b>
<b>4. Visualização por Caso de Uso</b>	<b>9</b>
4.1 Casos de Uso	9
• Login por usuário	9
• Consultar Produção	9
• Consultar Processamento	9
• Consultar Comercialização	9
• Consultar Importação	9
• Consultar Exportação	9
4.2 Arquitetura do Casos de Uso	9
6.2 Visualização de Dependências	10
<b>7. Tamanho e Desempenho</b>	<b>11</b>
<b>8. Qualidade</b>	<b>11</b>

# 1. Introdução

---

## 1.1 Finalidade

---

O documento em questão tem como objetivo apresentar a arquitetura da API Embrapa's viticulture (API de consulta pública nos dados do site vitivnicultura da Empraba) descrevendo todos os fundamentos aplicados em sua construção e como se dá a interação dos componentes de sua estrutura. Serão apresentadas funcionalidades centrais que demonstram a organização da arquitetura.

## 1.2 Escopo

---

O desenvolvimento do projeto tem como referência arquitetural toda a base apresentada neste documento, desde sua organização até a estruturação, expondo a tecnologia utilizada para consultar os dados de produção, processamento, comercialização, importação e exportação do site vitivinicultura. A autenticação e autorização de sistemas também serão tratadas. Esse documento tem como foco um guia para evitar a inexatidão e garantir sua coerência.

## 1.3 Definições, Acrônimos e Abreviações

---

- REST - Transferência de Estado Representacional
- API - Interface de programação de aplicações.
- JWT- JSON Web Token.

## 2. Representação da Arquitetura

---

### 2.1 Framework FastAPI

---

FastAPI é um framework Python moderno, projetado para simplicidade, velocidade e eficiência. A combinação de diversas funcionalidades modernas do Python como anotações de tipo e suporte à concorrência, facilitando o desenvolvimento de APIs.

O código fonte do framework foi inteiramente desenvolvido utilizando o recurso de type hints do Python, isso possibilita que as IDEs e Editores de Código Fonte possuam um melhor intellisense, fazendo assim com que se gaste menos tempo debugando o código.

### 2.2 Representational State Transfer (REST)

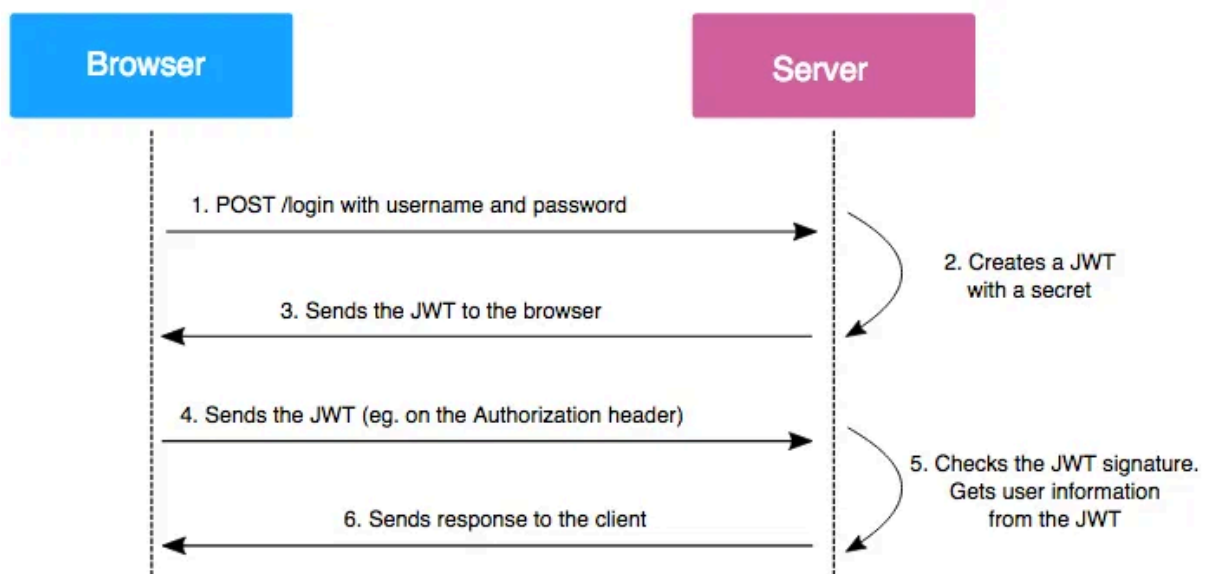
---

Para expor funcionalidades via web services utilizaremos o padrão REST que é um estilo de arquitetura que define um conjunto de restrições e propriedades baseados em HTTP. Web Services que obedecem ao estilo arquitetural REST, ou web services RESTful, fornecem interoperabilidade entre sistemas de computadores na Internet. Os web services compatíveis com REST permitem que os sistemas solicitantes acessem e manipulem representações textuais de recursos da Web usando um conjunto uniforme e predefinido de operações sem estado.

## 2.3 JWT

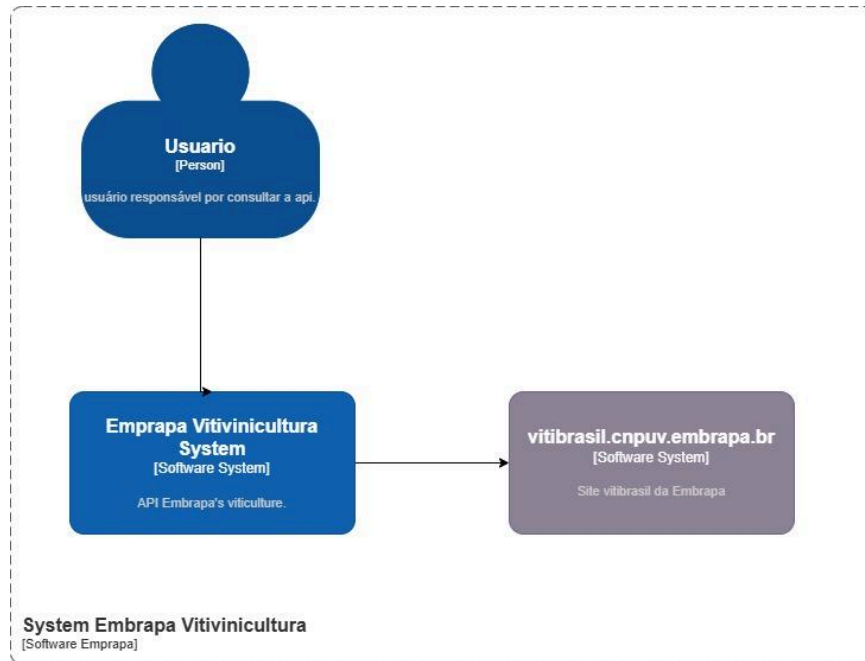
O JSON Web Token é utilizado em dois principais cenários, quando queremos realizar um processo de autorização em nossa aplicação ou quando queremos realizar troca de informações, abaixo detalhamos um pouco mais sobre o uso do JWT em cada um dos cenários citados. Autorização: Este é o cenário mais comum para o uso do JWT. Depois que o usuário estiver autenticado, cada requisição subsequente incluirá o JWT, permitindo que o usuário acesse rotas, serviços e recursos permitidos com esse token.

Troca de Informações: JSON Web Tokens são uma boa maneira de transmitir informações com segurança entre as partes. Como os JWTs podem ser assinados, por exemplo, usando pares de chaves pública e privada podemos garantir que os remetentes são quem dizem ser.

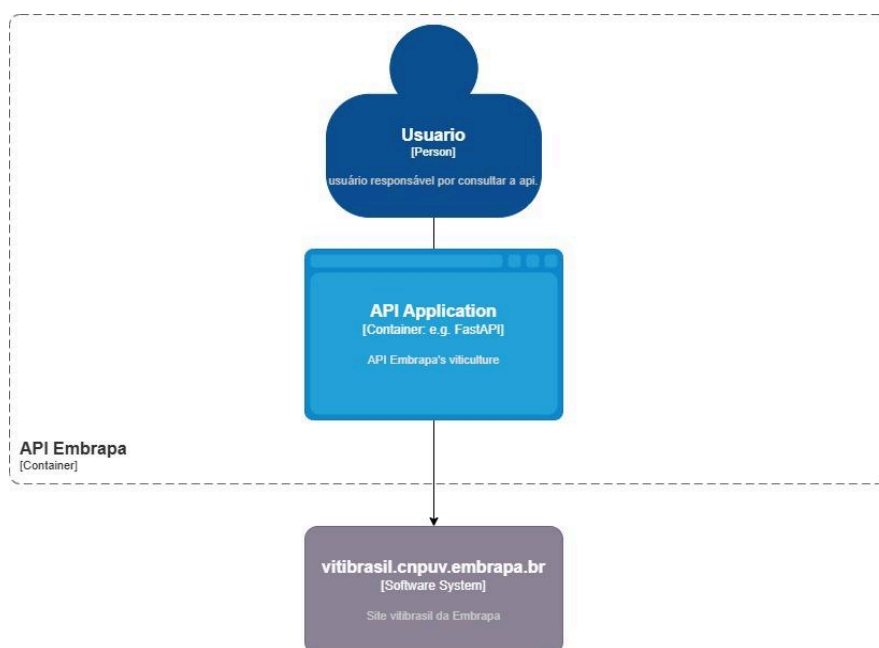


## 2.4 Arquitetura de Solução

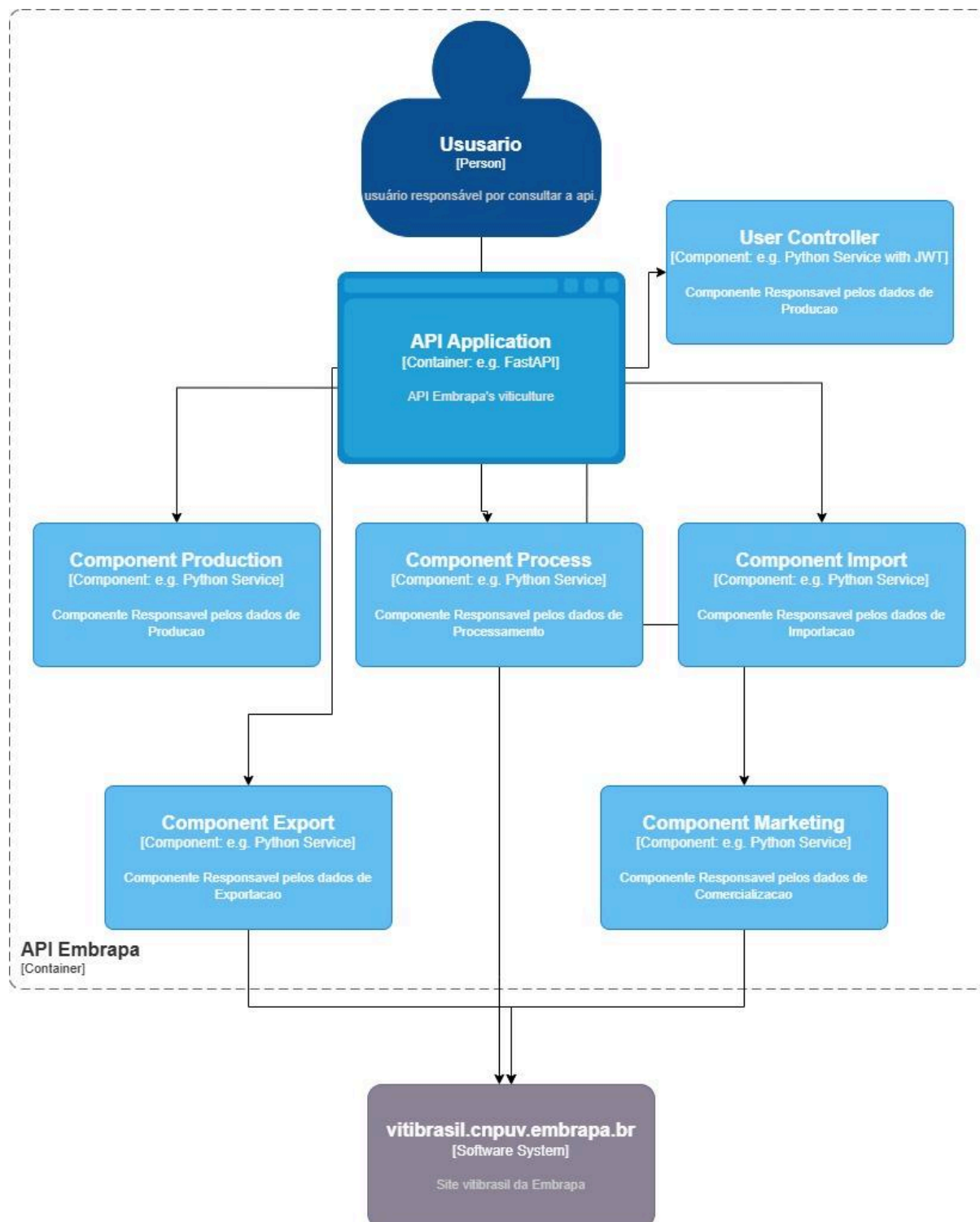
### 2.4.1 Nível 1 - Contexto



### 2.4.2 Nível 2 - Container

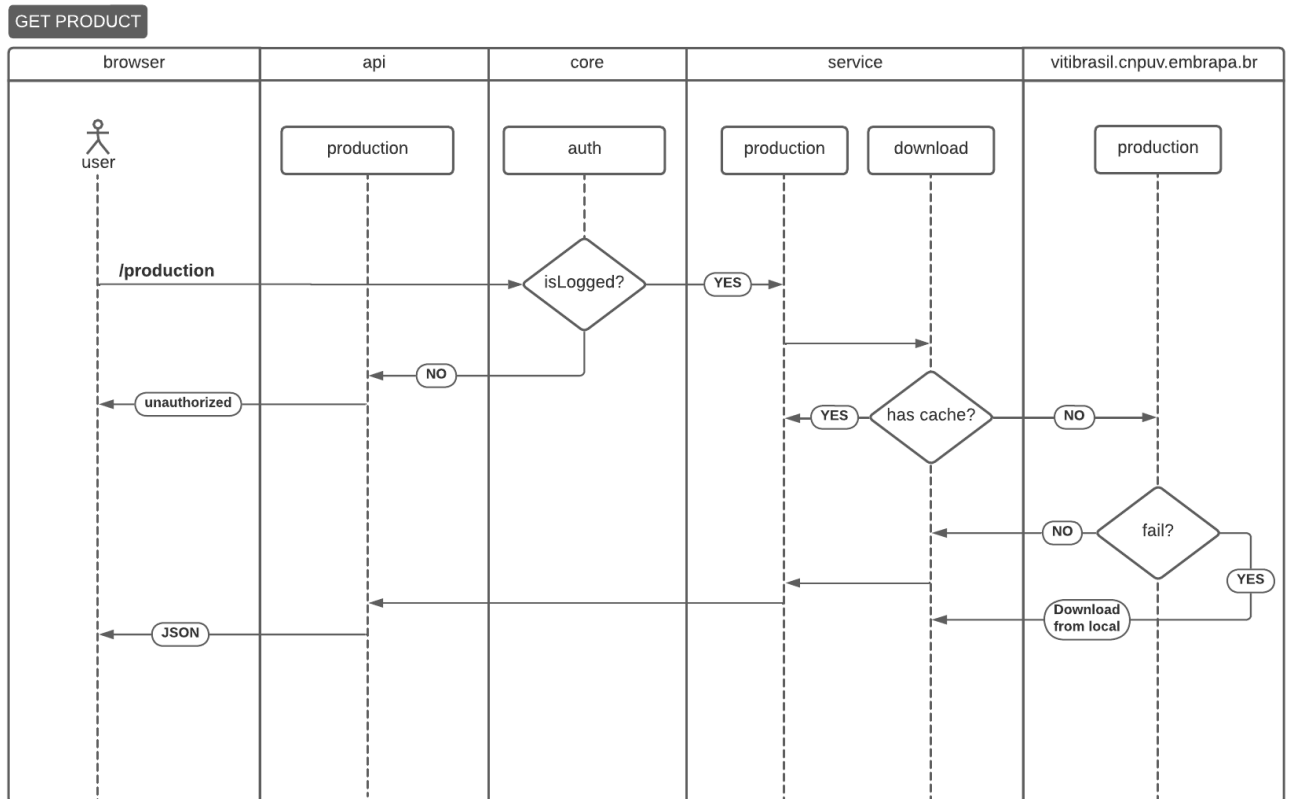


### 2.4.3 Nível 3 - Componentes





## 2.4.4 Nível 4 - Código



## 3. Metas e Restrições de Arquitetura

- Deve permitir o acesso por meio de cadastro de um usuário e autenticação.
- Deve a partir de informações como usuário, senha e o sistema validar o acesso e retornar um Token.
- Os sistemas devem estar preparados para a disponibilidade do site vitivinicultura da Embrapa.

## 4. Visualização por Caso de Uso

---

A Visão de Caso de Uso é uma entrada importante para a seleção do conjunto de cenários com atores e casos que são o foco de uma iteração. Ele descreve o conjunto de cenários que representam alguma funcionalidade central significativa. Nesse cenário temos o Ator Usuário interagindo com o Login de Usuário e Fazendo as Consultas na plataforma Vitivinicultura da Embrapa.

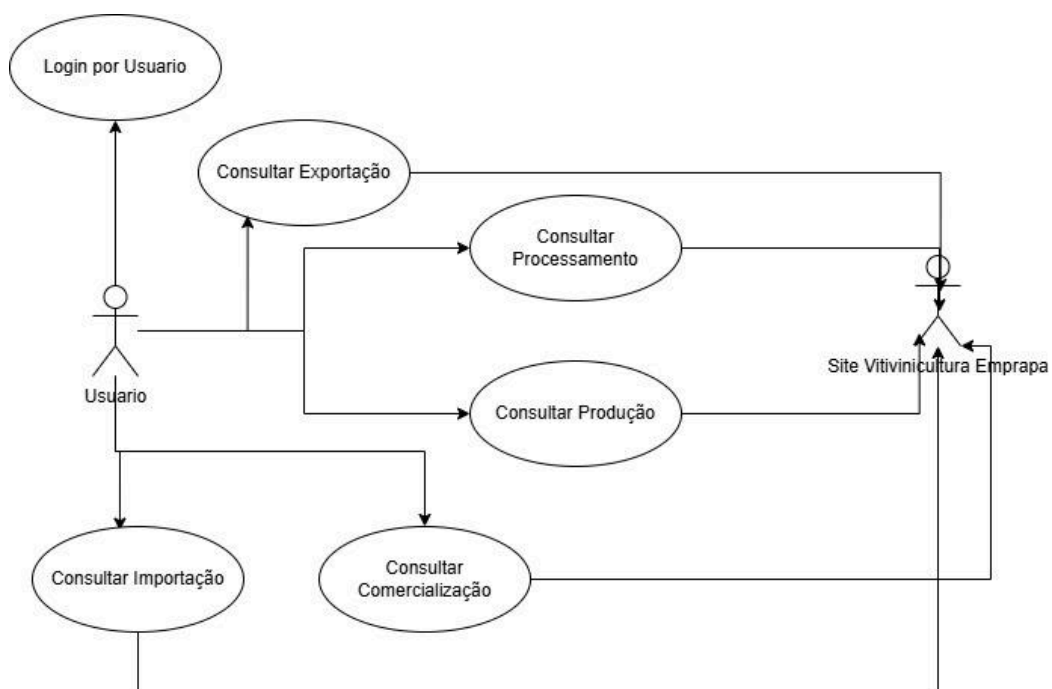
### 4.1 Casos de Uso

---

- Login por usuário
- Consultar Produção
- Consultar Processamento
- Consultar Comercialização
- Consultar Importação
- Consultar Exportação

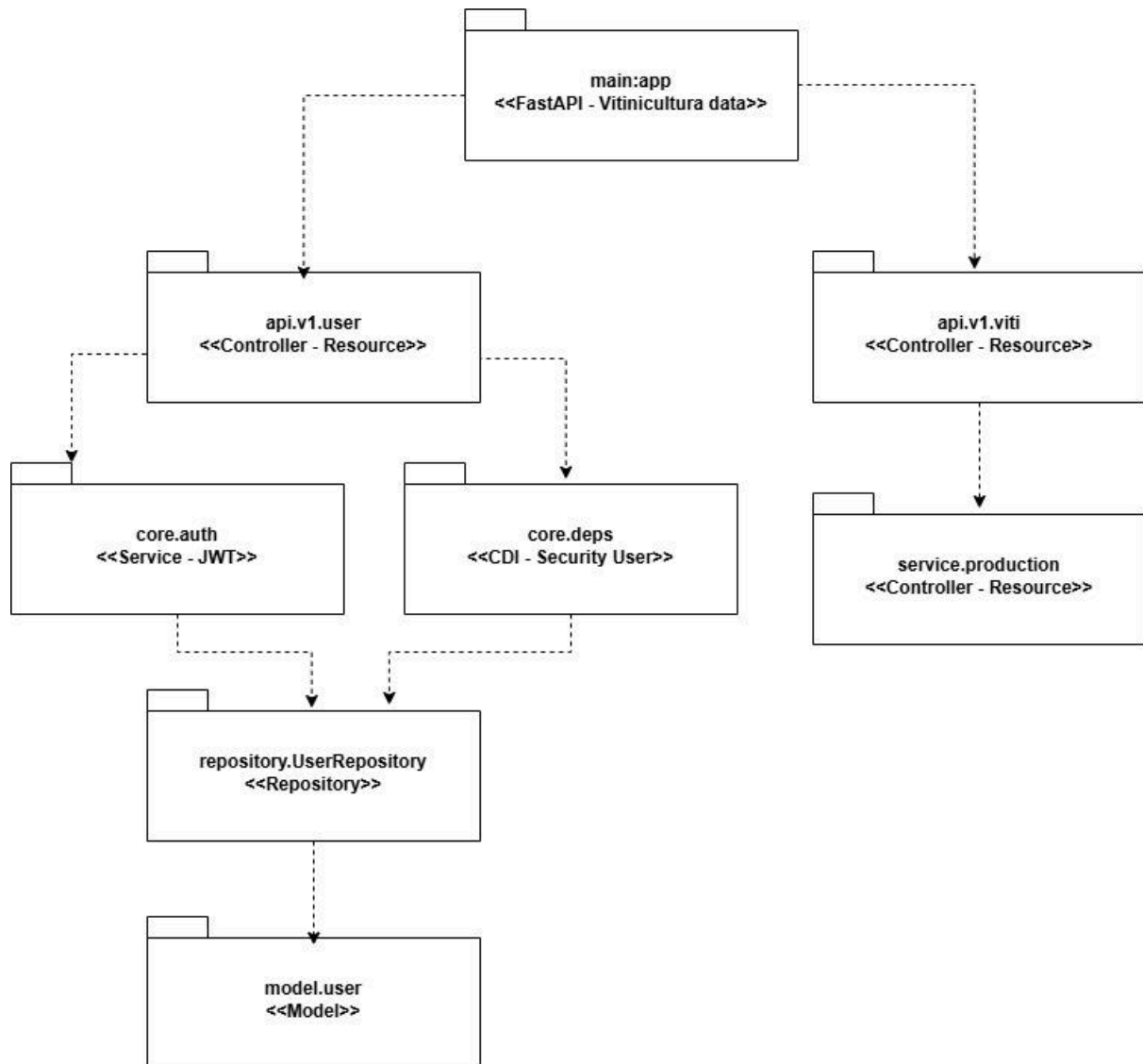
### 4.2 Arquitetura do Casos de Uso

---



## 6.2 Visualização de Dependências

---



## 7. Tamanho e Desempenho

---

Está prevista uma grande quantidade de acessos ao sistema e um grande processamento de volume de dados por parte do nosso software. Um dos pontos de Desempenho a ser identificado foi a disponibilidade.

## 8. Qualidade

---

A arquitetura utilizada no sistema tem efeito direto em sua consistência, pois propõe uma maneira de organização das camadas da aplicação que as tornam mais independentes e torna o código com uma maior facilidade de compreensão e manutenção, seguindo os padrões utilizados na arquitetura REST e melhores práticas, teremos um sistema mais confiável. Além desta ser uma arquitetura com uma grande comunidade, é o framework adequado para as necessidades do projeto, tendo em vista o foco no desenvolvimento do Back-end para um sistema de consulta de dados na plataforma vitivinicultura da Embrapa. O software garantirá a segurança dos dados informados pelo usuário, de modo que esses dados sejam guardados de forma segura e confiável, por meio de padrões de segurança que proporcionará ao usuário sua devida privacidade.