

UNIVERSIDADE FEDERAL DO PIAUÍ  
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS  
CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**FERRAMENTA COMPUTACIONAL GESTORA DE BIOPROCESSOS  
INDUSTRIAIS DE PRODUÇÃO E BENEFICIAMENTO DO MEL ORGÂNICO**

JONNISON LIMA FERREIRA

**PICOS-PI**

**2015**

JONNISON LIMA FERREIRA

**FERRAMENTA COMPUTACIONAL GESTORA DE BIOPROCESSOS  
INDUSTRIAIS DE PRODUÇÃO E BENEFICIAMENTO DO MEL ORGÂNICO**

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação, Campus Senador Helvídio Nunes de Barros, da Universidade Federal do Piauí como parte dos requisitos para obtenção do Grau de Bacharelado.

Orientador: Prof. MSc. Alcilene Dalília de Sousa.

**PICOS-PI**

**2015**

**FICHA CATALOGRÁFICA**  
**Serviço de Processamento Técnico**  
**Catálogo na Fonte**

**F383f** Ferreira, Jonnison Lima.  
Ferramenta computacional gestora de  
bioprocessos industriais de produção e beneficiamento do mel  
orgânico / Jonnison Lima Ferreira. - 2015.  
CD-ROM : il. ; 4 ¾ pol. (51 f.)

Monografia(Bacharelado em Sistemas de Informação) –  
Universidade Federal do Piauí. Picos-PI, 2015.  
Orientador(A): Prof. M.<sup>a</sup> Patricia Medyna L. de L. Drumond

1. Sistema Web. 2. Python e Django. 3. Controle de  
Produção Orgânica de Mel. I. Título.

**CDD 004.21**

FERRAMENTA COMPUTACIONAL GESTORA DE BIOPROCESSOS  
INDUSTRIAIS DE PRODUÇÃO E BENEFICIAMENTO DO MEL ORGÂNICO

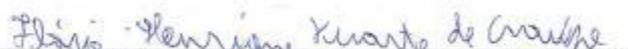
JONNISON DE LIMA FERREIRA

Monografia aprovada como exigência parcial para obtenção do  
grau de Bacharel em Sistemas de Informação.

Data de Aprovação

Picos – PI. 26 de junho de 20  

  
Prof.<sup>a</sup>. Me. Alcilene Dalíia de Sousa  
Orientadora

  
Prof. Me. Flávio Henrique Duarte de Araújo  
Membro

  
Prof. Esp. Leonardo Pereira de Sousa  
Membro

Dedico este trabalho a minha família, a minha mãe(Iraci), e meus irmãos (Joyciara e Jonnys) por sempre me apoiarem. Dedico também todos os meus amigos por toda ajuda e incentivo durante essa caminhada.

## **AGRADECIMENTOS**

É difícil citar todos os nomes que foram importantes para esse conquista e todos aqueles que sempre torceram por mim durante todos esses anos. Primeiramente quero agradecer a Deus, por sempre colocar coisas boas no meu caminho. Depois quero agradecer a toda a minha família, em especial minha mãe, pelos ensinamentos, conselhos e pelo esforço em sempre proporcionar a seus filhos tudo de melhor.

Agradeço a todos os meus professores, desde os professores da época do infantil até aqueles que sempre me incentivaram e acreditaram em mim na época de vestibular. Agradeço também a todos os professores do curso de Sistemas de Informação, desde aqueles em que eu já assisti aula em alguma disciplina, até aqueles que sempre tiveram tirando dúvidas, agradeço em especial também meus professores IFPI, que me apresentaram o mundo da tecnologia da informação, em especial ao Aislan, que hoje é meu companheiro de trabalho. Agradeço em especial a minha professora orientadora Alcilene, que sempre esteve disponível pra tirar minhas dúvidas.

Agradeço também a todos os profissionais com quem trabalhei, por estarem sempre ajudando, tirando dúvidas, passando experiência e aprendendo junto, em especial meus amigos da prefeitura de Picos, colegas de trabalho da UFPI, os meus grandes amigos do Instituto Multicom e da Body Desenvolvimento.

Em nossas vidas, aparecem pessoas que irão está do seu lado por pouco tempo e aquelas que irão permanecer o resto da vida ao seu lado. Pessoas que sempre são importantes na vida da gente são os amigos, e graças a Deus sempre tive amizades verdadeiras, agradeço a todos por estarem sempre presente na minha vida, nos bons momentos e nos ruins, sempre ajudando, incentivando, se divertindo, tendo paciência nos momentos difíceis. Deixo um agradecimento especial ao Nonato, Woshington e Celso, que sempre ajudaram, incentivaram, cobraram e tem tanto mérito nessa conquista quanto eu. A todos vocês deixo meus sinceros agradecimentos, muito obrigado!

"Time has come for everyone, to think what we have done. Open your eyes and see, it's not a dream.(O tempo veio para todos, para pensar no que fizemos Abra seus olhos e veja, isso não é um sonho)"

Sonata Artica - Abandoned,  
Pleased, Brainwashed, Exploited

"Nada, absolutamente nada resiste ao trabalho."

Euclides de Jesus Zerbini.

## RESUMO

O uso de modelos tecnológicos na produção de alimentos orgânicos ainda é pouco utilizado, em contramão a isso, a utilização de sistemas de informação tem aumentado de forma exponencial. A cultura agropecuária precisa cada vez mais adotar esses modelos afim de obter um melhor aproveitamento de recursos, além de servir como instrumento de controle dos processos e garantidor dos padrões destes tipos de alimentos. Nesse sentido o uso de modelos tecnológicos aplicados ao processo de produção de mel orgânico é benéfico no sentido de aumentar o controle, velocidade e segurança dos processos envolvidos na sua produção, além de produzir informações mais precisas, e até previsões sobre a quantidade e qualidade da produção. O objetivo desse trabalho foi desenvolver uma ferramenta computacional com integração à *internet* e mapeamento de georreferenciamento, capaz de avaliar, gerenciar e fornecer apoio tecnológico aos processos da cultura do mel em padrões orgânicos. Para o seu desenvolvimento foram utilizadas: a metodologia de desenvolvimento *Scrum*, a modelagem UML, a linguagem de programação *Python*, o *Framework Django* e o *Framework de front-end Bootstrap*. Por fim, foram mostradas as principais funcionalidades para verificar sua conformidade com as necessidades do problema.

**Palavras-chave:** Sistema *Web*, *Python* e *Django*, Controle de Produção Orgânica de Mel.

## ABSTRACT

The use of technologic models in the production of organic food is still little, on the other hand, the use of information systems is increasing. The agricultural culture is in an increasing need of the adoption of these models in order to obtain improvements in the use of resources as well as serving as an instrument for the control of the processes and to insure the pattern of this kind of food. In this sense, the use of technologic models in the process of producing organic honey is beneficent as it improves the control, speed and the safety of the processes involved in its production, as well as producing more precise information, and even forecasting the amount and quality of the production. The goal of this work is to develop a computational tool integrated to the internet and georeferencing mapping, able to evaluate, manage and supply technologic support to the processes of honey culture in organic patterns. For its development the following tools were used: Scrum development methodology, UML modeling, the programming language Python, the Framework Django and the Front-end Framework Bootstrap. At last, the main functionalities were shown to verify their conformity with the needs of the issue.

**Keywords:** Web system, Python and Django, Control of Organic Honey Production

## LISTA DE FIGURAS

Figura 1 - Exemplo código HTML .....	22
Figura 2 - Código CSS .....	23
Figura 3 - Ciclo de Desenvolvimento <i>Scrum</i> .....	26
Figura 4 - Casos de uso autenticação .....	37
Figura 5 - Diagrama de classes autenticação .....	39
Figura 6 - Casos de uso entrada .....	40
Figura 7 - Diagrama de classes entrada .....	41
Figura 8 - Casos de uso beneficiamento .....	42
Figura 9 - Diagrama de classes beneficiamento .....	43
Figura 10 - Menu do usuário .....	44
Figura 11 - Menu da aplicação .....	44
Figura 12 - Lista de usuários cadastrados .....	45
Figura 13 - Tela de cadastro de usuário .....	46
Figura 14 - Tela de cadastro de cooperados .....	46
Figura 15 - Tela de cadastro de entrada .....	47
Figura 16 - Tela de cadastro de itens de entrada .....	47
Figura 17 - Tela de cadastro de beneficiamento .....	48

## LISTA DE TABELAS

Tabela 1 - <i>Product BackLog</i> .....	36
---	----

## LISTA DE SIGLAS E ABREVIATURAS

ARO	<i>Army Research Office</i>
BSD	<i>Berkeley Software Distribution</i>
CSS	<i>Cascading Style Sheets</i>
DRY	<i>Don't Repeat yourself</i>
GPS	Sistema de Posicionamento Global
GPL	<i>General Public License</i>
HTML	<i>HyperText Markup Language</i>
JSON	<i>JavaScript Object Notation</i>
MVC	<i>Model-View-Controller</i>
MTV	<i>Model-Template-View</i>
NSF	<i>National Science Foundation</i>
ORM	<i>Object-relational mapper</i>
PSF	<i>Python Software Foundation</i>
SP	<i>Story points</i>
SQL	<i>Structured Query Language</i>
URL	<i>Uniform Resource Locator</i>
UML	<i>Unified Modeling Language</i>
VN	Valor de negócio
XML	<i>Extensible Markup Language</i>

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>13</b>
<b>1.1</b>	<b>Objetivo</b> .....	<b>14</b>
<b>1.2</b>	<b>Organização do Documento</b> .....	<b>15</b>
<b>2</b>	<b>ESTADO DA ARTE</b> .....	<b>16</b>
<b>2.1</b>	<b>Georreferenciamento</b> .....	<b>16</b>
<b>2.2</b>	<b>Lado Servidor</b> .....	<b>16</b>
2.2.1	<i>Python</i> .....	17
2.2.2	<i>Django</i> .....	18
2.2.3	<i>Banco de dados PostgreSQL</i> .....	20
<b>2.3</b>	<b>Lado Cliente</b> .....	<b>21</b>
2.3.1	<i>HTML</i> .....	21
2.3.2	<i>Cascading Style Sheets - CSS</i> .....	23
2.3.3	<i>JavaScript e JQuery</i> .....	24
<b>2.4</b>	<b>Engenharia de Software</b> .....	<b>25</b>
2.4.1	<i>Scrum</i> .....	25
2.4.2	<i>Linguagem de Modelagem Unificada (Unified Modeling Language – UML)</i> ...27	
<b>3</b>	<b>PROBLEMÁTICA</b> .....	<b>29</b>
<b>3.1</b>	<b>A Casa Ápis</b> .....	<b>29</b>
<b>3.2</b>	<b>Entendendo o Problema</b> .....	<b>30</b>
<b>3.3</b>	<b>O Sistema Web</b> .....	<b>31</b>
<b>4</b>	<b>IMPLEMENTAÇÃO/RESULTADOS/CENÁRIO</b> .....	<b>33</b>
<b>4.1</b>	<b>Descrição da Implementação</b> .....	<b>33</b>
<b>4.2</b>	<b>Resultados</b> .....	<b>44</b>
<b>5</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b> .....	<b>49</b>
	<b>REFERENCIAS</b> .....	<b>50</b>

## 1 INTRODUÇÃO

No decorrer da evolução desenfreada e desordenada do mercado percebe-se que a relação existente entre a industrialização, a urbanização, a evolução do mercado e a produção de alimentos culminou em problemas de natureza estruturais frente a uma gama de artifícios tecnológicos que não param de se aperfeiçoar.

Os paradigmas a que se referem todas estas relações se estendem em todo o mundo e gera questionamentos acerca da manutenção da cultura agrícola, no que se refere ao abastecimento alimentício, e o papel urbano de industrializar e comercializar. É pertinente perceber que a atuação tecnológica se faz necessário também na produção orgânica de alimentos, independente da sua transformação, pois serve como instrumento de controle dos processos e garantidor dos padrões estes tipos de alimentos.

O uso de modelos tecnológicos na produção de alimentos orgânicos ainda é pouco utilizado, esta deficiência se deve ao fato de uma visão pragmática que limita as fases de elaboração de modelos e não refletem a necessidade de monitoramento de todas as etapas, que deveria descrever detalhadamente os fenômenos existentes no objeto de estudo por meio de uma abordagem multidisciplinar. A ausência de informações qualitativas e quantitativas disponíveis, geralmente em decorrência da falta de publicação de dados que facilitem a elaboração, manutenção e atualização de um banco de dados detalhados relativos às questões mais variadas relacionadas ao ambiente agropecuário, comprovam ainda mais tal deficiência.

Uma análise completa dos processos que envolvem a produção do mel orgânico, envolvendo a mensuração das variáveis que os influenciam de uma forma mais pontual é, na maioria das vezes, difícil de ser conduzida por conta da ausência de aporte tecnológico. Uma maneira de contornar esse problema é recorrer à prospecção de tecnologias existentes em sistemas semelhantes, representados por uma proximidade do objetivo do estudo desta proposta, facilitando a descrição, controle e padronização dos processos envolvidos. Prospecção tecnológica corresponde a um meio sistemático de mapear desenvolvimentos científicos e tecnológicos futuros capazes de influenciar de forma significativa uma indústria, a economia ou a sociedade como um todo e tomar por base seus objetivos de: criar planos de contingências para aproveitar oportunidades ou enfrentar ameaças, e; construir padrões tecnológicos desejáveis para determinados segmentos.

Mediante essa justificativa, torna-se possível a intenção de criar modelos tecnológicos de monitoramento que identifiquem os problemas de forma mais eficiente e, posteriormente faça sua adequação à linguagem computacional, objetivando o acompanhamento deste sistema pela gestão dos processos a partir das especificações orgânicas.

No que se refere ao uso de tecnologia para os objetivos desta proposta, deve-se ressaltar a utilização de instrumentos de Georreferenciamento como atividade de apoio da linguagem computacional a ser gerada. E sua utilização corresponde ao processo inicial de mapeamento da cadeia produtiva tendo em vista a mensuração detalhada da quantidade de caixas/colmeias, sua localização, sua gestão e sua produtividade.

Vale ressaltar que o sistema desenvolvido utiliza das informações geradas pelo Georreferenciamento, assim como outras cadastradas no sistema, com usabilidade, seguindo padrões de interação com o usuário. Por ser um sistema *web*, essa informação pode ser acessada de qualquer lugar por um usuário com acesso a internet, seja um produtor, uma cooperativa, um funcionário da unidade de extração ou um cliente que pode acessar o sistema e verificar os locais exatos de onde partiu o mel.

O projeto foi então desenvolvido de acordo com as necessidades apresentadas pela Casa Ápis, localizada na Rua Santo Antônio, número 296, bairro Pantanal da cidade de Picos-PI, é uma central de cooperativas de produção de mel (orgânico e tradicional), atuante na macrorregião da cidade de Picos, com grande participação no mercado internacional com sua produção, principalmente no setor de mel orgânico. A central possui diversas cooperativas vinculadas, cooperativas essas, onde os apicultores estão associados.

## **1.1 Objetivo**

O objetivo desse trabalho foi desenvolver uma ferramenta computacional com integração à internet e mapeamento de georreferenciamento, capaz de avaliar, gerenciar e fornecer apoio tecnológico aos processos da cultura do mel em padrões orgânicos, produzidos por meio da agricultura familiar e beneficiados em cooperativas solidárias, desde seus aspectos naturais até sua certificação orgânica para a

negociação no mercado interno e externo, bem como, determinar a efetividade na utilização dos recursos e o controle da qualidade dos bioprocessos industriais envolvidos.

## **1.2 Organização do Documento**

O trabalho desenvolvido será apresentado nos próximos capítulos organizado da seguinte maneira:

- Capítulo 2 – Estado da Arte: Esse capítulo relata as tecnologias utilizadas durante o desenvolvimento do projeto, descrevendo os principais conceitos e alguns exemplos simples.
- Capítulo 3 – Problemática: Nesse capítulo é descrito a problema que o projeto solucionou, demonstrando de maneira clara as necessidades da Casa Ápis.
- Capítulo 4 – Sistema Desenvolvido e Resultados: Nesse capítulo é apresentado o desenvolvimento do projeto, mostrando como o mesmo ocorreu, e os resultados obtidos.
- Capítulo 5 – Conclusões e Trabalhos Futuros: Esse capítulo é responsável por abordar quais os conhecimentos adquiridos e os problemas enfrentados no decorrer do desenvolvimento. Também são apresentadas ideias e dicas para trabalhos futuros, afim de melhorar e dar sequência ao que foi desenvolvido nesse projeto.

## 2 ESTADO DA ARTE

Durante o desenvolvimento de um projeto, é necessário a utilização de alguns recursos importantes para que seja possível obter êxito. Quando se trata do desenvolvimento de uma aplicação *web*, não é diferente, existem diversas tecnologias que podem ser utilizadas para se atingir o objetivo principal. As principais tecnologias utilizadas durante esse projeto são abordadas neste capítulo.

### 2.1 Georreferenciamento

O Georreferenciamento corresponde, segundo Augusto (2006) a obtenção das coordenadas (pertencentes ao sistema no qual se pretende georreferenciar) de pontos da imagem ou do mapa, conhecidos como pontos de controle. Os pontos de controle são locais que oferecem uma feição física perfeitamente identificável. A obtenção das coordenadas dos pontos de controle pode ser realizada em campo (a partir de levantamentos topográficos, GPS (Sistema de Posicionamento Global), ou ainda por meio de mesas digitalizadoras, ou outras imagens ou mapas (em papel ou digitais) georreferenciados.

O georreferenciamento será utilizado no processo inicial de mapeamento da cadeia produtiva tendo em vista a mensuração detalhada da quantidade de caixas/colmeias, sua localização, sua gestão, sua produtividade e além de toda a movimentação do mel desde a colmeia ao fim do seu processo de beneficiamento.

### 2.2 Lado Servidor

O funcionamento dos sistemas *web* acontece da seguinte maneira: o usuário (*client-side*) faz uma solicitação de uma página *web* para um servidor (*server-side*), onde esse servidor recebe a requisição do usuário e retorna alguma resposta para o navegador. Quando essa resposta chega do lado do cliente, o navegador interpreta todo o código devolvido pelo servidor, e exibe para o usuário (NIEDERAUER, 2011).

Baseado em tal afirmação, conclui-se que, todos os dados do sistema são armazenados pelo servidor e repassados para o cliente na forma de *hypertexto*, sendo o mesmo responsável pelo gerenciamento dos dados.

### 2.2.1 Python

*Python* é uma linguagem de altíssimo nível (em inglês, *Very High Level Language*) orientada a objeto, de tipagem dinâmica e forte, interpretada e interativa. A linguagem foi criada em 1990 por Guido van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda (CWI) e tinha originalmente foco em usuários como físicos e engenheiros. O Python foi concebido a partir de outra linguagem existente na época, chamada ABC (BORGES, 2010).

O nome da linguagem origina-se do nome da série humorística britânica *Monty Python's Flying Circus*, do grupo humorístico britânico *Monty Python*. Guido queria criar uma linguagem de altíssimo nível, que agregasse características importantes de diversas linguagens e programação. Além disso, queria que essa linguagem de programação mantivesse uma sintaxe simples e clara (Santana,2010). Python é um software de código aberto (com licença compatível com a *General Public License* (GPL), porém menos restritiva, permitindo que o *Python* seja inclusive incorporado em produtos proprietários). A especificação da linguagem é mantida pela *Python Software Foundation* (PSF) (BORGES,2010). A linguagem *Python* apresenta uma série de vantagens e recursos interessantes que foram inspirados e aproveitados de outras linguagens de sucesso, tornando-a assim uma linguagem bastante flexível.

Reis (2010) lista alguns recursos apresentados pela linguagem:

- Os conceitos fundamentais da linguagem são simples de entender.
- A sintaxe da linguagem é clara e fácil de aprender; o código produzido é normalmente curto e legível.
- Os tipos pré-definidos incluídos em *Python* são poderosos, e ainda assim simples de usar.
- A linguagem possui um interpretador de comandos interativo que permite aprender e testar rapidamente trechos de código.
- *Python* é expressivo, com abstrações de alto nível. Na grande maioria dos casos, um programa em *Python* será muito mais curto que seu correspondente escrito em outra linguagem. Isto também faz com o ciclo de desenvolvimento seja rápido e apresente potencial de defeitos reduzido – menos código, menos oportunidade para errar.

- Existe suporte para uma diversidade grande de bibliotecas externas. Ou seja, pode-se fazer em *Python* qualquer tipo de programa, mesmo que utilize gráficos, funções matemáticas complexas, ou uma determinada base de dados SQL (*Structured Query Language*).
- É possível escrever extensões a *Python* em C e C++ quando é necessário desempenho máximo, ou quando for desejável fazer interface com alguma ferramenta que possua biblioteca apenas nestas linguagens.
- *Python* permite que o programa execute inalterado em múltiplas plataformas; em outras palavras, a sua aplicação feita para *Linux* normalmente funcionará sem problemas em *Windows* e em outros sistemas onde existir um interpretador *Python*.
- *Python* é pouco punitivo: em geral, ‘tudo pode’ e há poucas restrições arbitrárias. Esta propriedade acaba por tornar prazeroso o aprendizado e uso da linguagem.
- *Python* é livre: além do interpretador ser distribuído como software livre (e portanto, gratuitamente), pode ser usado para criar qualquer tipo de software — proprietário ou livre. O projeto e implementação da linguagem é discutido aberta e diariamente em uma lista de correio eletrônico, e qualquer um é bem-vindo para propor alterações por meio de um processo simples e pouco burocrático.

Atualmente, *Python* ocupa o 8º lugar no ranking das linguagens de programação mais populares do mundo, segundo o Índice Tiobe (TIOBE, 2014). A implementação oficial do *Python* é mantida pela *PSF* e escrita em C, e por isso, é também conhecida como *CPython*. Existem também implementações de *Python* para *.NET* (*IronPython*), *JVM* (*Jython*) e em *Python* (*PyPy*)(BORGES,2010).

### 2.2.2 Django

Segundo Santana e Galesi(2010), o *Django* é um *framework* de desenvolvimento web, escrito em *Python*, que foi criado pelo grupo editorial “*The World Company*” para criação da versão *web* dos seus jornais. Posteriormente, em 2005, foi liberado sob a licença *BSD* (*Berkeley Software Distribution*), tornando um software de código aberto. O *Django* utiliza o conceito de *DRY – Don’t Repeat yourself*

(Não se repita), que trabalha com a ideia de usar convenções em substituição às configurações.

O *Django* é um *superframework*, pois é formado de vários *frameworks*(componentes) menores com funções mais específicas, Santana e Galesi (2010) listam os principais componentes:

- **ORM – Object-relational mapper (Mapeador objeto-relacional):** permite que você programe utilizando objetos sem se preocupar com a persistência desses dados no seu banco de dados relacional. Essa camada de software abstrai toda a comunicação com seu banco de dados, permitindo que você manipule seus objetos sem o uso de *SQL* (é possível acessar diretamente o banco de dados em uma aplicação *Django*, mas esse tipo de uso só se torna interessante na otimização de determinadas consultas).
- **Template System:** fornece uma linguagem para criação de templates (*HTML - HyperText Markup Language, XML - Extensible Markup Language, JSON - JavaScript Object Notation* etc.) usados na geração de páginas dinâmicas.
- **Sistema de administração:** o *Django* é um dos poucos frameworks web que disponibiliza uma interface, gerada quase automaticamente, que permite a administração dos objetos de negócio da sua aplicação.
- **URL (Uniform Resource Locator) dispatcher:** cuida do processamento das URLs do sistema executando funções especificadas pelo desenvolvedor e possibilitando o uso de URLs amigáveis ao usuário.
- **Internacionalização:** facilita a internacionalização de seu sistema, permitindo que ele funcione corretamente com diversos idiomas.
- **Formulários:** geração automática de formulários e facilitação na manipulação dos dados enviados por meio deles.
- **Segurança:** gerenciamento de autenticação de usuários e controle de permissões.
- **Outros componentes:** serialização de dados, sistema de testes automatizado, serviço de mensagens (*e-mails* e mensagens entre

usuários do sistema), sistema de cache de objetos, geração de geração de *feeds* (RSS/Atom), paginação de resultados.

O Django utiliza o paradigma MVC (*Model-View-Controller*) para desenvolvimento, na documentação oficial do Django é utilizada a sigla MTV (*Model-Template-View*), que representa a mesma coisa dita com palavras diferentes. Santana e Galesi (2010) definem que isso significa que a sua aplicação tem três partes principais:

- **Model:** contém a representação dos dados do projeto. Normalmente é a parte que conversa com o banco de dados e que persiste seus dados. No caso Django, o sistema ORM frequentemente é usado na criação de nossas classes de *Model*.
- **View:** cuida da visualização do conteúdo no formato escolhido. Frequentemente, há várias *Views* para um mesmo tipo de *Model*. No caso do Django, o sistema de *Templates* é quem cumpre esse papel.
- **Controller:** cuida do acesso e da manipulação dos dados do *Model*, bem como de outras partes que não envolvem o *Model* diretamente. No Django, esse papel é feito por dois componentes: o *URL dispatcher* e as *Views* (por isso MTV).

Por fim, utilizando-se do framework Django obtém-se agilidade, organização nos projetos, pois tudo tem seu lugar definido, distribuídos entre as três camadas com suas responsabilidades que permite um trabalho bem “centrado” e modularizado.

### 2.2.3 Banco de dados PostgreSQL

O *PostgreSQL* é um sistema de gerenciamento de banco de dados relacional, de código aberto que utiliza licença *BSD*. Segundo Milani (2008) o *PostgreSQL* teve origem em um projeto chamado de *POSTGRES* na Universidade *Berkeley*, na Califórnia (EUA), em 1986. Uma equipe orientada pelo professor Michael Stonebraker foi designada para criar o modelo e as regras de um novo sistema de armazenamento de dados, com o apoio de diversos órgãos, entre eles o Army Research Office (ARO) e o National Science Foundation (NSF).

O *PostgreSQL* não possui limitação de tamanho máximo para seus bancos de dados, sendo limitado apenas pelo hardware disponível pelo computador onde está

sendo armazenadas suas informações. Segundo Milani (2009) sua limitação dá-se em nível de tabela, com um limite máximo de 32TB por tabela. Além do mais, é possível ter registros (linhas) com até 1.6TB, campos com até 1GB, tabelas com até 1.600 campos e índices ilimitados para aceleração da busca dos resultados. A estabilidade do *PostgreSQL* também é um de seus recursos mais interessantes, pois foi projetado para executar no método 24/7 (24 horas por dia, sete dias na semana). Em outras palavras, executar indefinidamente. Assim, este banco de dados tem sido bastante utilizado também na internet, seja em sites, portais, lojas virtuais ou soluções em informática.

O software desenvolvido utiliza o *PostgreSQL* como sistema de gerenciamento de sua base de dados em todos os seus módulos.

## 2.3 Lado Cliente

O lado cliente de uma aplicação *web*, é responsável por realizar a requisição e se comunicar com o servidor, da mesma forma que receber a resposta da requisição e apresentar ao usuário da aplicação. Dessa forma conclui-se que é o lado cliente que realiza a comunicação direta com o usuário.

### 2.3.1 HTML

Samy (2008) explica que HTML é a sigla em inglês para *HyperText Markup Language*, que, em português, significa linguagem para marcação de hipertexto. Dessa maneira, fica entendido que essa linguagem é responsável por inserir conteúdo em documentos para *web*. De acordo com Jonathan Stark e Brian Jepson (2012), o HTML é uma das três principais tecnologias utilizadas para o desenvolvimento de aplicações *web*, juntamente com o CSS (*Cascading Style Sheets*) e *JavaScript*, que serão abordados no decorrer desse trabalho.

Segundo Eis e Ferreira (2012), a informação é de fundamental importância dentro de um aplicativo *web*, e deve ser acessada a qualquer hora, de qualquer lugar e por qualquer dispositivo ou meio de acesso, ou seja, a informação é tudo que o usuário consome. Sob tal afirmação, fica entendido que o HTML é o principal responsável pela manipulação da informação, dando-lhe significado. Os meios

tecnológicos utilizados para acessarem uma aplicação *web* não conseguem entender visualmente o que é um título, um parágrafo, uma imagem ou qualquer outro tipo de informação, e essa é a importância da marcação HTML, definir para os meios tecnológicos como devem tratar cada coisa.

O HTML é o principal responsável por definir o conteúdo de páginas na internet, onde essas páginas são compostas por elementos hierarquicamente organizados, sendo conhecidos como tags, (EIS e FERREIRA, 2012). Essa abordagem do autor deixa claro que os elementos das páginas *web* são representados por tags HTML e que cada uma possui significado diferente para cada tipo de informação que será contida na página ou aplicação. Lembrando que essas tags são marcadas pelos sinais de “menor que” (<) e “maior que” (>), identificando assim a abertura e fechamento das tags. Alguns exemplos são listados a seguir:

- <html> </html> - Tag que identifica uma página HTML, onde todo o seu conteúdo deve ficar dentro dessa tag;
- <head> </head> - Irá conter algumas informações de cabeçalho da página, como título, descrição, linguagem da página, entre outros;
- <title> </title> - Define o título da página *web*;
- <p> </p> - Especifica um parágrafo;
- <table> </table> - Define uma tabela;
- <img> - Marca essa informação como sendo uma imagem.

A Figura 1 é um exemplo simples de uma página com código HTML, mostrando algumas tags HTML e suas funcionalidades de marcação:

```
<html>
<head>
  <title>Título da página web</title>
</head>
<body>
  <h1>Minha primeira página HTML</h1>
  <p>Essa página trás meu primeiro código html</p>
</body>
</html>
```

Figura 1 - Exemplo código HTML

Fonte: O autor (2015)

### 2.3.2 Cascading Style Sheets - CSS

CSS ou Folhas de Estilos em Cascata é uma linguagem de folhas de estilos utilizada para definir a apresentação visual de um documento HTML, (STARK e JEPSON, 2012). De acordo com Samy (2011), a CSS é responsável por definir todas as funções de apresentação de um documento HTML e deixar uma página *web* mais bonita.

Eis e Ferreira (2012) afirmam que, o CSS tem como principal objetivo, formatar o conteúdo HTML de forma que seja visualmente agradável em qualquer meio de acesso, ou seja, para que a informação apareça de maneira adequada, independente do meio de acesso.

O CSS permite alterar propriedades como: cor de fundo de um elemento HTML, cor e tamanho da letra, posicionamento dos elementos, efeitos visuais, entre outros. Um exemplo de um código CSS é apresentado na Figura 2, onde a *tag* <p> irá receber uma cor de fundo azul, cor da letra vermelho e tamanho da letra de 12 pixels.

```
p{  
    color: red;  
    background: blue;  
    font-size: 12px;  
}
```

**Figura 2 - Código CSS**

**Fonte: O autor (2015)**

Fica entendido que a CSS é a linguagem responsável por realizar a formatação de documentos HTML e deixar as informações das páginas *web* mais bem apresentadas para os usuários finais, deixando as aplicações da internet com uma apresentação mais agradável.

### 2.3.3 *JavaScript e JQuery*

Balduíno (2012) diz que, *JavaScript* é uma linguagem de *script* que possibilita aos desenvolvedores adicionar um nível de interatividade e funções as aplicações *web*. O autor afirma ainda, que o *JavaScript* é o principal responsável pelo comportamento das páginas HTML.

Com o *JavaScript* é possível definir se os elementos HTML serão arrastados, dimensionados, rotacionados, reformatados, etc (EIS e FERREIRA, 2012). Com essa linguagem, ainda é possível manipular as características dos elementos pelo CSS, ou seja, o *JavaScript* controla os valores definidos pelo CSS e manipula suas propriedades.

Stark e Jepson (2012) afirmam que, *JavaScript* é uma linguagem de *script* que pode ser adicionada a uma página HTML para torná-la mais interativa e conveniente para o usuário. Ou seja, com um código *JavaScript*, é possível inspecionar valores digitados em um formulário para garantir que sejam valores válidos ou até mesmo mostrar ou esconder elementos de uma página, dependendo de onde o usuário clicar. Essa linguagem de *script* pode até contatar o servidor *web* para executar alterações no banco de dados sem atualizar a página *web* atual.

Existem bibliotecas para várias linguagens de programação, onde as mesmas possuem as mesmas características da linguagem de programação pura, mas realizam funções de forma simples e direta, facilitando durante o desenvolvimento e utilização das linguagens. O *JavaScript* possui diversas bibliotecas que otimizam o desenvolvimento e uso da linguagem, sendo a biblioteca *JQuery* é a mais conhecida delas.

*JQuery* é uma biblioteca *JavaScript* que simplifica a manipulação de documentos HTML, eventos, animações e interações para desenvolvimento rápido de aplicações *web*. Devido à sua simplicidade e flexibilidade, o *JQuery* acabou se tornando a biblioteca *JavaScript* mais utilizada, (BALDUINO, 2012).

Segundo Samy (2008), *JQuery* é uma biblioteca *JavaScript* e disponibilizada como *software* livre e aberto, ou seja, essa biblioteca pode ser utilizada gratuitamente em projetos pessoais ou até mesmo em projetos comerciais. O autor afirma ainda, que em relação a biblioteca, a mesma é compatível com todos os navegadores *web*. Sendo uma de suas principais vantagens o enorme apoio e envolvimento de uma gigantesca comunidade de desenvolvedores.

## 2.4 Engenharia de Software

Segundo Pressman (2011), todo software está sujeito a mudanças, sendo que essas mudanças podem ser realizadas pela mesma equipe que desenvolveu o software ou não. Durante as fases de desenvolvimento e manutenção de *software*, é necessário a utilização de técnicas e padrões que melhorem o *software*. O autor afirma ainda, que isso é papel da Engenharia de *Software*, onde segundo o autor, Engenharia de *Software* é a aplicação de técnicas e métodos no desenvolvimento, na operação e na manutenção do *software*. A seguir, são apresentados algumas das principais fases da Engenharia de *Software*.

### 2.4.1 Scrum

De acordo com Sabbagh (2013), *Scrum* é um *framework* Ágil, simples e leve, utilizado para a gestão do desenvolvimento de produtos complexos imersos em ambientes complexos. O autor afirma ainda, que a metodologia de desenvolvimento adotada pelo *Scrum* faz com que os desenvolvedores ou a equipe de desenvolvimento entregue partes do projeto com frequência, reduzindo assim, os riscos do projeto. O *Scrum* é ágil porque sua utilização segue princípios e valores do Manifesto para Desenvolvimento Ágil de *Software*, o qual, seus valores são:

“Indivíduos e interações mais que processos e ferramentas; *Software* em funcionamento mais que documentação abrangente; Colaboração com o cliente mais que negociação de contratos; Responder a mudanças mais que seguir um plano” (BECK *et al.*, 2001).

Segundo Sabbagh (2013) o uso do *Scrum* traz benefícios que incluem:

- Entregas frequentes de retorno ao investimento dos clientes;
- Redução dos riscos do projeto;
- Maior qualidade no produto gerado;
- Mudanças utilizadas como vantagem competitiva;
- Visibilidade do progresso do projeto;
- Redução do desperdício;
- Aumento de produtividade.

O *Scrum* faz com que os possíveis erros ou falhas de desenvolvimento possam ser corrigidos o mais rápido possível, antes mesmo do projeto está concluído ou em fases finais. Fica claro ainda, que a metodologia utilizada pelo *Scrum* só pode ser atendida quando existe um elo de ligação e comunicação entre a equipe de desenvolvimento e os usuários futuros do projeto e/ou as partes envolvidas.

Segundo Douglas (2009), os participantes do projeto são organizados de acordo com três papéis: O *Scrum Master* é um novo papel de gerente introduzido pelo *Scrum*, sendo sua principal função, é ser facilitador do processo e ajudar as pessoas a resolver os problemas, enquanto dissemina e aplica o processo dentro da organização; O time *Scrum* deve ser multidisciplinar e auto organizado, devendo ser pequeno, com no máximo 10 a 15 participantes. O conjunto do conhecimento dos seus participantes deve englobar todas as características necessárias para a implementação do projeto a que se propõe; O *Product Owner* é o cliente de um time *Scrum*, ele é responsável pelo retorno do investimento de um produto. Entre suas tarefas estão: definir todas as funcionalidades do sistema; priorizar as funcionalidades a serem incluídas em cada *Sprint*, de acordo com o valor de negócio; aceitar ou rejeitar entregas do time.

Baseado nas afirmações dos autores, o ciclo de desenvolvimento no *Scrum* mostrado na Figura 3, se inicia na elaboração do *Product BackLog*, que é, uma lista de funcionalidades a serem desenvolvidas no sistema, organizado na forma de histórias de usuário, que nada mais é, do que, uma descrição concisa de uma necessidade do usuário do produto (ou seja, de um "requisito") sob o ponto de vista desse usuário.

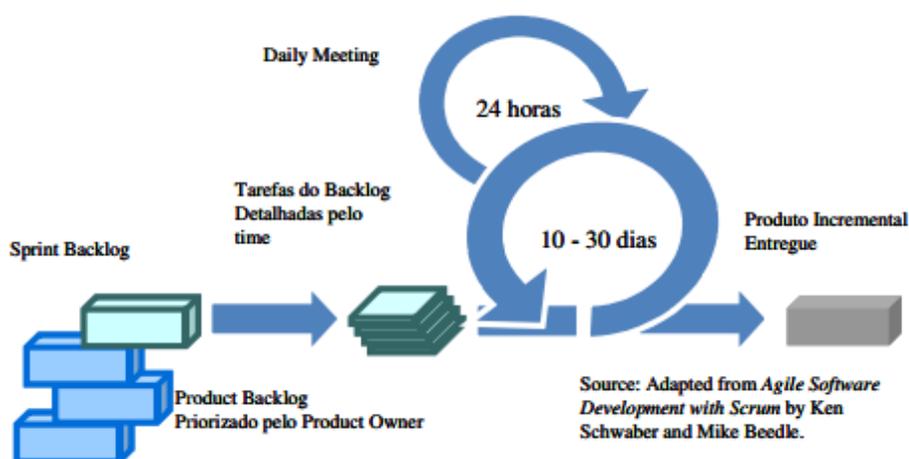


Figura 3 - Ciclo de Desenvolvimento *Scrum*

Fonte: VARASCHIM (2009, P.03)

De acordo com a Figura 3, o desenvolvimento é dividido em *Sprints*, que duram de duas a quatro semanas, e ao final da sprint é mostrado ao *Product Owner*, e o projeto é revisado, durante a *sprint* são realizadas reuniões diárias onde é verificado o andamento do projeto. Um conjunto de *sprints* que formam uma versão utilizada do produto é chamado de *release*.

#### 2.4.2 Linguagem de Modelagem Unificada (Unified Modeling Language – UML)

Durante o desenvolvimento de um projeto, é aconselhável e de fundamental importância, realizar a análise e coleta de requisitos do projeto. Segundo Pádua (2003), o valor de um produto vem de suas características, e os requisitos são as características que definem os critérios de aceitação de um produto. Ou seja, o processo de análise de requisitos é um levantamento das funcionalidades que o sistema deve atender e as tarefas que o mesmo irá realizar, entendendo e colocando no papel, o que uma aplicação destina-se a fazer depois de construída.

Durante a análise de requisitos, é de fundamental importância a construção de gráficos e diagramas que representem as funcionalidades da aplicação. Isso é feito através da Linguagem de Modelagem Unificada (*Unified Modeling Language – UML*) que segundo Braude (2005) é uma notação gráfica para expressar projetos orientados a objetos. Ela faz uso de diagramas e gráficos para representar as funcionalidades do sistema. Existem diversos diagramas que podem fazer representação do sistema, alguns exemplos são: diagrama de classe; diagrama de caso de uso; diagrama de sequência, entre outros.

Segundo Guedes (2011), UML é uma linguagem visual utilizada para modelar *softwares* baseados no paradigma de orientação a objetos, ou seja, é uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicação. O autor acrescenta ainda, que essa linguagem tornou-se, nos últimos anos, a linguagem padrão de modelagem adotada internacionalmente pela indústria de engenharia de *software*.

Fica entendido que a UML não é uma linguagem de programação, e sim uma linguagem de modelagem, uma notação, cujo objetivo é auxiliar os engenheiros de *software* e desenvolvedores a definirem as características do sistema. Por mais simples que seja, todo e qualquer sistema deve ser modelado antes de se iniciar sua

implementação, pois todo sistema de informação costuma mudar, o que acaba facilitando a manutenção do sistema, caso exista uma modelagem da aplicação, (GUEDES, 2011).

Todas ferramentas apresentadas, serviram como base para o entendimento de toda a problemática do sistema, podendo assim, conhecer, entender e solucionar os diversos problemas identificados durante o desenvolvimento do projeto.

### **3 PROBLEMÁTICA**

Esse capítulo mostra as especificações do problema solucionado pela ferramenta de gestão de bioprocessos industriais de produção e beneficiamento do mel orgânico proposto nesse trabalho. Será mostrado o funcionamento atual da empresa, especificando os problemas enfrentados e que foram solucionados com a ferramenta desenvolvida.

#### **3.1 A Casa Ápis**

O uso de modelos tecnológicos na produção de alimentos orgânicos ainda é pouco utilizado, esta deficiência se deve ao fato de uma visão pragmática que limita as fases de elaboração de modelos e não refletem a necessidade de monitoramento de todas as etapas, que deveria descrever detalhadamente os fenômenos existentes no objeto de estudo por meio de uma abordagem multidisciplinar. Para Araújo (2009), a ausência de informações qualitativas e quantitativas disponíveis, geralmente em decorrência da falta de publicação de dados que facultem a elaboração, manutenção e atualização de um banco de dados detalhados relativos às questões mais variadas relacionadas ao ambiente agropecuário, comprovam ainda mais tal deficiência.

A Casa Ápis, localizada na Rua Santo Antônio, número 296, bairro Pantanal da cidade de Picos-PI, é uma central de cooperativas de produção de mel (orgânico e tradicional), atuante na macrorregião da cidade de Picos, com grande participação no mercado internacional com sua produção, principalmente no setor de mel orgânico. A central possui diversas cooperativas vinculadas, cooperativas essas, onde os apicultores estão associados.

A empresa necessita de um sistema que automatize o monitoramento da produção e beneficiamento de mel orgânico, que era feito de maneira totalmente manual, por meio de diferentes planilhas eletrônicas, o que causava grande inconsistência nos dados e um maior desperdício de tempo no desenvolvimento dessas tarefas.

O desenvolvimento desse projeto aconteceu de maneira colaborativa entre os membros e funcionários da Casa Ápis, que também são usuários do sistema. Como as funcionalidades desenvolvidas foram especificadas e acompanhadas pelos usuários finais, o sistema atende as reais necessidades da instituição, trazendo diversos benefícios para a mesma.

### 3.2 Entendendo o Problema

A extração do mel, a ser considerado completamente orgânico, apresenta uma série de dificuldades, sobretudo, porque quebra uma série de paradigmas existentes nos procedimentos realizados pelos apicultores, criadores de abelhas, que geralmente trabalham de forma empírica, sem seguir padrões de técnicas e processos, sendo necessário reeducar este produtor para a assimilação de processos mais higiênicos, seguros e padronizados que referenciam as características orgânicas.

O processo de produção de mel orgânico começa na coleta do mel nos apiários, que são os locais onde o apicultor matem suas colmeias produtoras de mel. Após a coleta, o mel é levado a uma unidade de extração, que pertence à cooperativa que o apicultor está associado, onde são retiradas as impurezas e armazenado em baldes ou tambores separados por cor, em uma análise superficial, assim, formam-se lotes para envio a central. O valor do mel varia de acordo com sua cor e a época do ano. A cooperativa é responsável por gerenciar as unidades de extração e fazer com que se cumpram as exigências para que o mel seja considerado orgânico, e então, seja enviado a central (Casa Ápis).

Após a chegada do mel na Casa Ápis, o mesmo é conferido de acordo com a nota de envio da cooperativa, e novamente separado por cor. Dessa vez, por uma análise mais detalhada, realizada com o auxílio de equipamentos específicos para sua análise, obtendo assim sua cor de maneira precisa. O valor do mel repassado ao apicultor depende da cotação do mel no dia que ocorre a entrada na central de cooperativas, após o recebimento o mesmo é armazenado no estoque de matéria prima, onde continua separado por cor e apicultor.

O beneficiamento do mel se dá por uma série de processos e etapas que agregam valor ao produto final, o primeiro passo para o beneficiamento é a seleção da matéria prima, visando encontrar cor ideal. A primeira etapa é realizada na mesa coletora/descristalizadora, onde é homogeneizado e descristalizado se necessário, e filtrado. Ao término da homogeneização, o mel é encaminhado para o desumidificador, caso a umidade esteja acima de ideal, sendo a mesma monitorada até atingir o valor ideal. Após o processo de desumidificação, o mel é bombeado para os decantadores e em seguida envasado.

Todo o estoque de matéria prima fica armazenado até que ocorra uma solicitação de compra por clientes da Casa Ápis. Assim, o mel só passa pelo processo de beneficiamento quando já possui comprador definido, afim de se obter o mel mais homogêneo possível, utilizando os processos citados no parágrafo anterior. Após o beneficiamento, o lote de mel é estocado e enviado posteriormente ao cliente.

Todas as etapas, desde a extração ao beneficiamento, os processos não ocorrem no mesmo local, ou seja, o mel passa por diversos locais e procedimentos para que se atinja o produto final. Assim, é necessário informar ao cliente final, todos os locais por onde o produto passou – apiários, unidades de extração, central – para que o mesmo obtenha detalhes da qualidade do mel.

### **3.3 O Sistema Web**

Diante das necessidades encontradas na Casa Ápis, foi necessário analisar e estudar os processos internos, afim de identificar os maiores problemas e encontrar uma solução que pudesse resolvê-los. Durante essa etapa foi realizada diversas entrevistas e visitas com as pessoas ligadas a cada processo da empresa para identificação dos requisitos e elaboração do modelo computacional que atendesse as reais necessidades dos usuários do sistema.

Para que se iniciasse o desenvolvimento do projeto, foi necessário levantar alguns questionamentos a respeito da rotina na empresa, como por exemplo, quais e quantos são os usuários do sistema? O que cada usuário pode fazer? Quais dados seriam armazenados? Que relatórios poderiam ser obtidos? Como seria realizado o acompanhamento do produto, identificando por quais locais o mesmo passou e como o cliente receberia essa informação? Diante desses e outros levantamentos, foi possível identificar quais funcionalidades seriam implementadas no sistema *web*.

Segundo Filho (2003), uma boa engenharia de requisitos é um passo essencial para o desenvolvimento de um bom produto de *software*, em qualquer caso. De acordo com tal afirmação, durante o levantamento e a análise de requisitos, diversas perguntas devem ser levantadas pelos desenvolvedores, até mesmo aquelas que não estejam no entendimento do cliente. Ou seja, o cliente quer uma funcionalidade no sistema, mas ele não sabe os riscos e nem a maneira que esses riscos devem ser tratados. Tudo isso deve ser analisado por parte dos desenvolvedores da aplicação.

Na maioria das vezes, o que geralmente acontece é, o cliente ou usuário não saber identificar quais são suas verdadeiras necessidades para a aplicação.

Diante da análise e levantamento de requisitos, percebeu-se que devido as diferentes localizações geográficas em que o produto iria passar e os seus diversos usuários – apicultor, funcionário da produção, responsável pela análise, cliente –, um sistema *web* seria a melhor solução para os problemas apresentados, pois assim, seria possível todos os usuários nas suas diferentes localizações acompanharem todas as informações geradas pelo sistema.

## 4 IMPLEMENTAÇÃO/RESULTADOS/CENÁRIO

O cliente (item 3.1) solicitou um sistema que pudesse ser utilizado no gerenciamento da produção do mel orgânico, onde o operador de produção receba o mel vindo das unidades de extração e realize o controle da entrada da matéria prima no estoque, sendo o setor administrativo responsável por gerenciar o estoque e a produção do mel beneficiado. Levando em conta que os apiários e as unidades de extração devem obedecer as normas e exigências das certificadoras, para que o mel seja considerado orgânico dentro dos padrões estabelecidos. Sendo assim, o consultor deve cadastrar esses dados e preencher os *checklists* exigidos.

### 4.1 Descrição da Implementação

Durante o processo de desenvolvimento e implementação do sistema, foi necessário conhecer e entender os procedimentos utilizados por todas as partes e usuários, desde os apiários e unidades de extração aos processos realizados na própria central de cooperativas – estoque de matéria prima, análise, beneficiamento – para que se chegasse no mínimo, a um protótipo que atendesse as funcionalidades exigidas, tornando assim, um sistema usável.

O sistema proposto foi desenvolvido utilizando a metodologia *Scrum* (sub tópico 2.4.1), a qual realiza análise em forma de histórias de usuário, onde cada história está associada a uma necessidade do usuário. O primeiro passo foi definir o *Product BackLog*, com as histórias de usuário definidas pelo *Product Owner*, um exemplo resumido do *Product BackLog* está na tabela 1.

ID	TEMA	DESCRIÇÃO	CRITÉRIOS DE ACEITAÇÃO	VALOR DE NEGÓCIO (VN)	ESTIMATIVA (STORY POINTS - SP)	ROI (VN/ SP)	SPRINT
1	Cadastro	Como consultor preciso cadastrar as cooperativas para identificar os produtores de mel.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	1000	11	90,9	1

ID	TEMA	DESCRIÇÃO	CRITÉRIOS DE ACEITAÇÃO	VALOR DE NEGÓCIO (VN)	ESTIMATIVA (STORY POINTS - SP)	ROI (VN/SP)	SPRINT
2	Cadastro	Como consultor preciso cadastrar os cooperado para identificar a produção do mesmo.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	3000	20	150	1
3	Cadastro	Como consultor preciso cadastrar as UE (casas de mel) para identificar onde o mel é envasado.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	800	5	160	1
4	Cadastro	Como consultor preciso cadastrar os apiários para identificar o local de produção e mapear a produtividade.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	500	3	167	1
5	Cadastro	Como consultor preciso cadastrar as análises das entradas para classificar o tipo de meu armazenado.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	1200	50	24	2
6	Cadastro	Como administrativo preciso cadastrar os funcionários para manter registro dos mesmos.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	1200	100	12	2
7	Cadastro	Como administrativo preciso cadastrar a cotação para ser referenciada na entrada.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	3000	20	150	2

ID	TEMA	DESCRIÇÃO	CRITÉRIOS DE ACEITAÇÃO	VALOR DE NEGÓCIO (VN)	ESTIMATIVA (STORY POINTS - SP)	ROI (VN/SP)	SPRINT
8	Cadastro	Como administrativo preciso associar a cotação a entrada.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	500	11	45,5	2
9	Cadastro	Como administrativo preciso cadastrar o cliente do sistema para identificar a empresa responsável.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	900	8	113	2
10	Cadastro	Como administrativo preciso cadastrar o produto para identificar os itens comercializados pelo cliente.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	1000	11	90,9	3
11	Cadastro	Como administrativo preciso cadastrar a matéria-prima para identificar os insumos necessários para a produção do produto	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	3000	8	375	3
12	Cadastro	Como operador preciso cadastrar os recipientes para identificar onde é armazenado a matéria-prima.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	1200	3	400	3
13	Cadastro	Como operador preciso cadastrar as entradas com seus respectivos	Verificar o preenchimento dos campos obrigatórios; Exibir	500	3	167	3

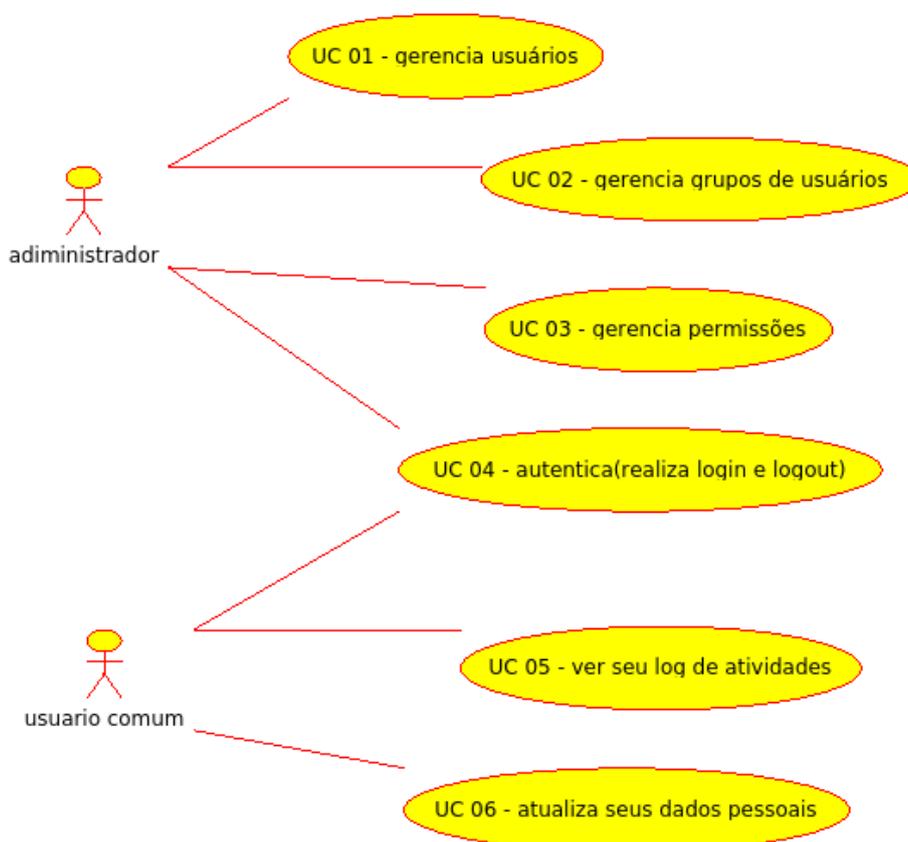
ID	TEMA	DESCRIÇÃO	CRITÉRIOS DE ACEITAÇÃO	VALOR DE NEGÓCIO (VN)	ESTIMATIVA (STORY POINTS - SP)	ROI (VN/ SP)	SPRINT
		itens para identificar o carregamento de matéria-prima	mensagem de sucesso.				
14	CheckList	Como Especialista preciso cadastrar os itens com suas respectivas opções á serem avaliados para identificar o nível tecnológico.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	800	3	267	3
15	CheckList	Como consultor preciso avaliar o nível tecnológico de acordo com as opções especificadas para identificar o nível tecnológico.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso; Retornar o nível tecnológico;	800	3	267	4
16	CheckList	Como consultor gostaria de verificar o histórico de avaliações de nível tecnológico para realizar o monitoramento;	Exibir o histórico;	1000	11	90,9	4
17	Beneficiamento	Como operador preciso cadastrar o beneficiamento para realizar a transformação de matéria-prima em produto final.	Verificar o preenchimento dos campos obrigatórios; Exibir mensagem de sucesso.	3000	5	600	4

Tabela 1 - Product BackLog

O *Product BackLog* possui oito atributos, especificados a seguir:

1. *ID*: é responsável por identificar de forma única as histórias do *Product BackLog*.
2. Tema: corresponde ao módulo do sistema ao qual a história pertence.
3. Descrição: Contém a história de usuário
4. Critérios de aceitação: possui os requisitos para que funcionalidade implementada seja considerada válida.
5. Valor de Negócio: Define a importância da história para o produto final, quanto maior o valor, mais importante é a história.
6. Estimativa (*Story Points*): Contém o custo do ponto de vista do desenvolvimento para se implementar a história.
7. ROI (VN/SP): Retorno sobre o investimento, é a relação entre o valor de negócio e a estimativa de custo
8. *Sprint*: Informa em que *Sprint* a funcionalidade foi implementada.

Com base na histórias de usuário descritas na Tabela 1, o sistema foi dividido em 3 subsistemas, o primeiro subsistema foi feito a partir do aplicativo de autenticação do *Django*, foi desenvolvido então o diagrama de casos de uso e o diagrama de classes. A Figura 4 mostra o diagrama de casos de uso.

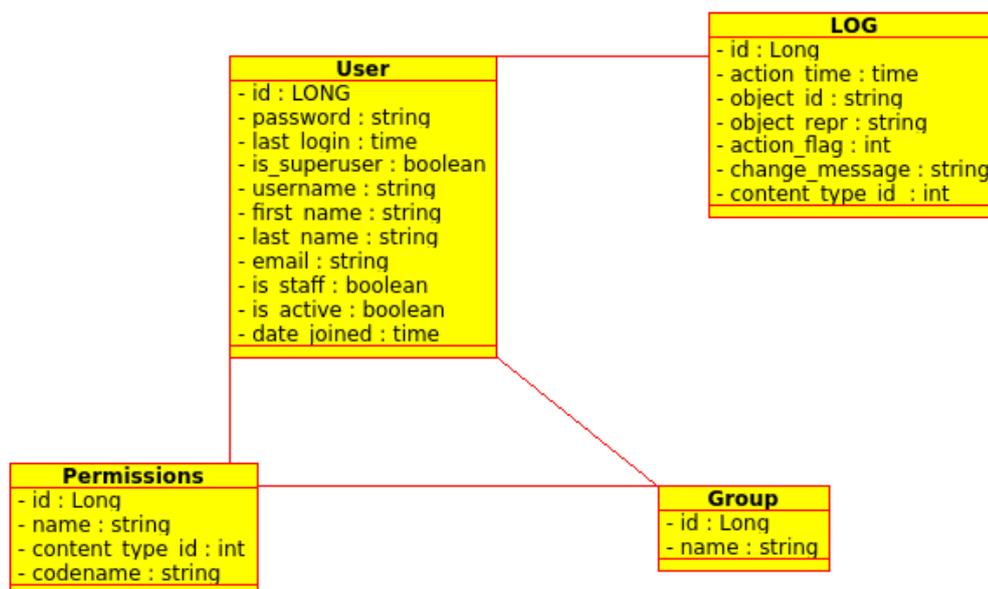


**Figura 4 - Casos de uso autenticação**

Fonte: O autor (2015)

O diagrama da Figura 4, mostra os principais casos de uso do subsistema de autenticação, onde há dois tipos de usuário, o administrador que irá fazer o controle de permissões dos usuários e grupos de usuário, e o usuário comum que usará o sistema para de autenticar e cadastrar seus dados pessoais. O casos de uso são detalhados a seguir:

- **UC01 – Gerencia usuários**
  - **Ator:** Administrador
  - **Descrição:** O usuário administrador pode cadastrar, excluir, editar e ativar ou desativar os usuários do sistema
- **UC02 – Gerencia grupos de usuários**
  - **Ator:** Administrador
  - **Descrição:** O usuário administrador pode cadastrar, excluir, editar os grupos de usuário do sistema, controla também os usuários que fazem parte de cada grupo
- **UC03 – Gerencia permissões**
  - **Ator:** Administrador
  - **Descrição:** O usuário administrador concede ou retira permissões de usuários e/ou grupos de usuários, definindo que parte do sistema cada um pode utilizar
- **UC04 – Autentica (Realiza login e logout)**
  - **Ator:** Administrador e usuário comum
  - **Descrição:** Todos os usuários devem se autenticar, para que o sistema os identifique defina suas permissões acesso
- **UC05 – Ver seu *log* de atividades**
  - **Ator:** Usuário comum
  - **Descrição:** Cada usuários pode ver *log* das atividades que realizou no sistema
- **UC06 – Atualiza seus dados pessoais**
  - **Ator:** usuário comum
  - **Descrição:** O usuário pode alterar seus dados pessoais, nome, email, sobrenome

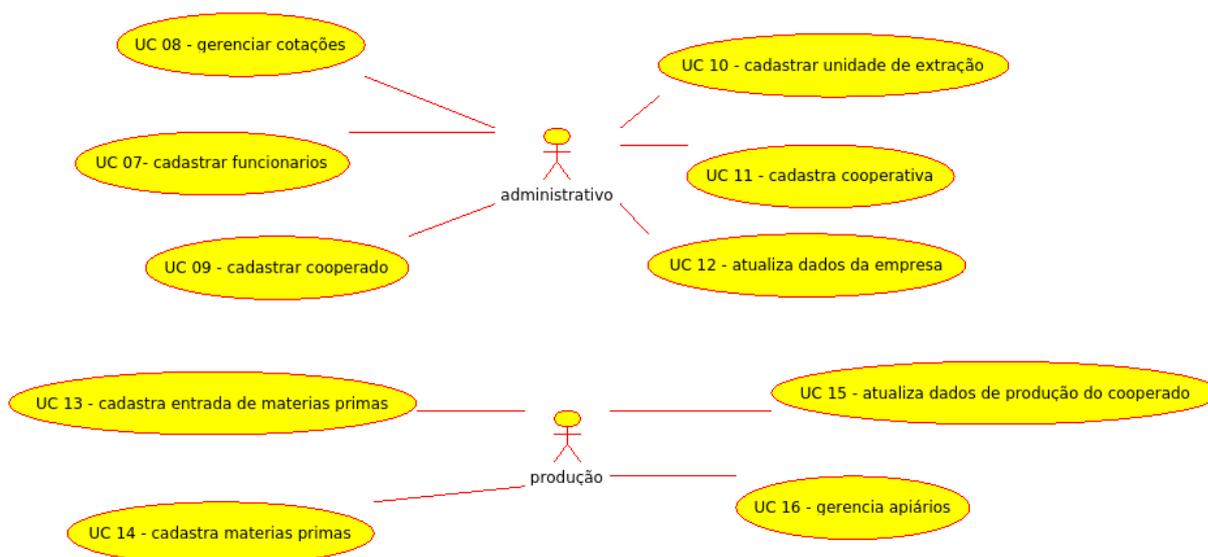


**Figura 5 - Diagrama de classes autenticação**

**Fonte: O autor (2015)**

A Figura 5 mostra o diagrama de classes do subsistema de autenticação. Onde a classe “User” representa os usuários do sistema com seus atributos, essa classe se relaciona com a classe “Permissions”, que representa as permissões do sistema, cada usuário pode ter uma ou várias permissões e uma mesma permissão pode ser dada a mais de um usuário. A classe “User” se relaciona também com a classe “Group” que representam um conjunto de permissões, formando assim grupos de usuários, para isso a classe “Group” também se relaciona com a classe “Permissions”. A classe “Log” representa o *log* de operações realizadas pelos usuários, por isso a classe também se relaciona com “User”.

O segundo subsistema desenvolvido foi o de entradas, que é responsável por cadastrar todos os dados de cooperativas, unidades de extração, cooperados, apiários e as entradas de mel na central de cooperativas. A Figura 6 mostra o diagrama de casos de uso desse subsistema.



**Figura 6 - Casos de uso entrada**

Fonte: O autor(2015)

O diagrama da Figura 6, mostra os principais casos de uso do sistema de entrada, que possui dois tipos de usuário: O administrativo, que define os usuários do setor administrativo da empresa, que controla as notas de entrada, cotação atual do mel, os dados pessoais de todos os membros da empresa; O segundo é o usuário da produção, responsável por receber a matéria prima e armazená-la no local adequado. O casos de uso são detalhados a seguir:

- **UC07 – Cadastrar funcionários**
  - **Ator:** Administrativo
  - **Descrição:** O sistema deve controlar o cadastro de funcionários que trabalham na empresa.
- **UC08 – Gerenciar cotações**
  - **Ator:** Administrativo
  - **Descrição:** O usuário administrativo é responsável por cadastrar as cotações de mel a serem usadas no repasse ao cooperado.
- **UC09 – Cadastrar cooperado**
  - **Ator:** Administrativo
  - **Descrição:** O sistema deve controlar o cadastro dos cooperado associados a empresa.
- **UC10 – Cadastrar unidade de extração**
  - **Ator:** Administrativo
  - **Descrição:** O sistema deve conter o cadastro das unidades de extração de cada cooperativa vinculada a central
- **UC11 – Cadastra cooperativa**

- **Ator:** Administrativo
- **Descrição:** O sistema deve conter o cadastro das cooperativas vinculas a central
- **UC12 – Atualiza dados da empresa**
  - **Ator:** Administrativo
  - **Descrição:** O setor administrativo pode atualizar os dados da instituição, como endereço, nome.
- **UC13 – Cadastra entradas de matéria prima**
  - **Ator:** Produção
  - **Descrição:** O sistema possui um módulo de cadastro das entradas de matérias primas na empresa
- **UC14 – Cadastra matérias primas**
  - **Ator:** Produção
  - **Descrição:** O sistema possui um módulo de cadastro das matérias primas utilizadas pela empresa
- **UC15 – Atualiza dados de produção do cooperado**
  - **Ator:** Produção
  - **Descrição:** O sistema deve fornecer área para atualização dos dados de produção do cooperado, como por exemplo os dados relativos a certificação orgânica.
- **UC16 – Gerencia apiários**
  - **Ator:** Produção
  - **Descrição:** O funcionário da produção é responsável por cadastrar a localização e demais dados dos apiários de cada cooperado.

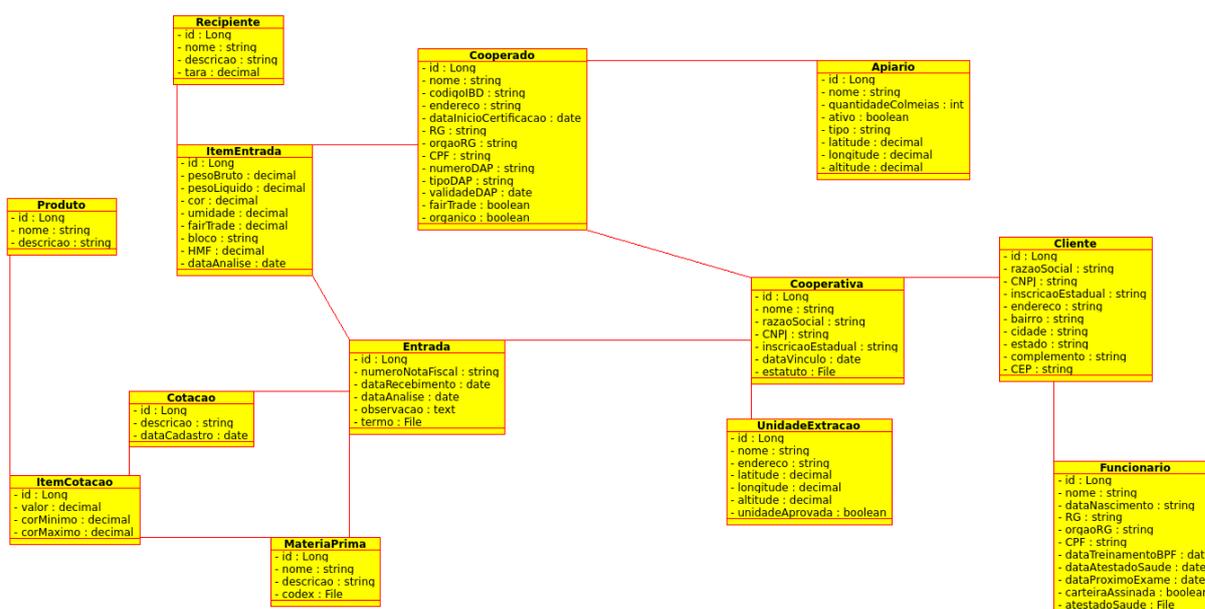


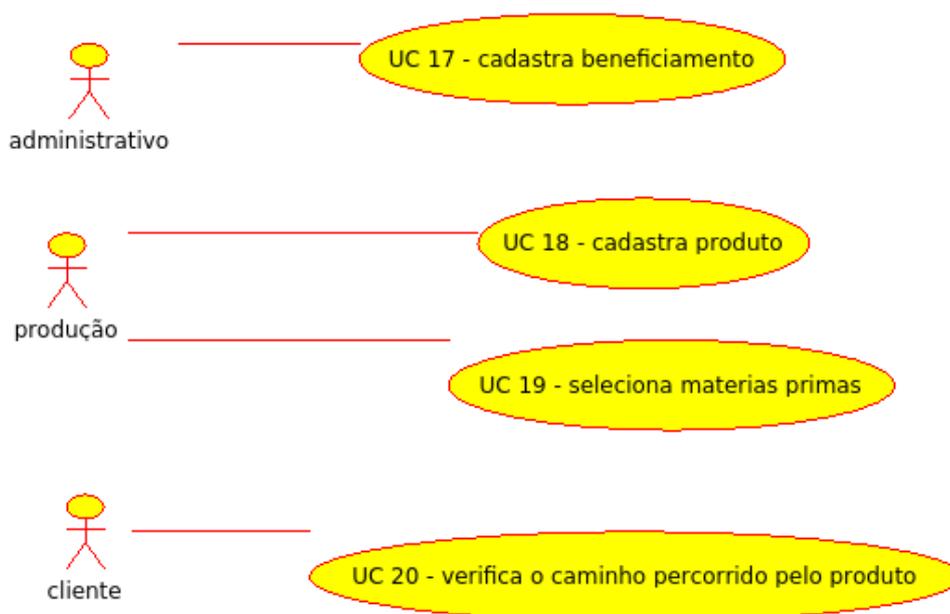
Figura 7 - Diagrama de classes entrada

Fonte: O autor(2015)

A Figura 7 mostra o diagrama de classes do subsistema de autenticação. Onde a classe “Entrada” é a responsável por gerenciar os dados relativos a chegada de matéria na central, representada no diagrama pela classe “MateriaPrima”. Cada entrada tem sua cotação, portanto, a classe “Cotacao” está relacionada com entrada, cada matéria prima possui um valor de cotação, dependendo de sua qualidade e produto gerador por ela, para isso, existe a classe “ItemCotacao” que contém esses valores. A entrada é composta por itens, que é a produção de cada tipo de matéria prima do cooperado, para essa informação temos a classe “ItemEntrada”, a classe “Cooperado” representa os dados os cooperados.

A classe “Cliente” representa os dados da empresa, essa classe está relacionada com a classe “Funcionario”, que representa os funcionários da empresa. A classe “Cliente se relaciona ainda com a classe “Cooperativa”, que representa as cooperativas vinculadas a empresa. Cada cooperativa possui unidades de extração, representadas no diagrama pela classe “UnidadeExtracao”. Os cooperados estão vinculados a uma cooperativa, por isso ocorre o relacionamento entre as classes “Cooperativa” e “Cooperado”.

O terceiro subsistema desenvolvido foi o de beneficiamento, que é responsável por cadastrar os dados da fase de beneficiamento da matéria prima e geração do produto beneficiado. A Figura 8 mostra o diagrama de casos de uso desse subsistema.



**Figura 8 - Casos de uso beneficiamento**

Fonte: O autor(2015)

O diagrama da Figura 8, mostra os principais casos de uso do sistema de beneficiamento, que possui três tipos de usuário: O administrativo, que solicita o beneficiamento; O segundo é o usuário da produção, responsável por selecionar a matéria prima e cadastrar o produto gerado; o terceiro é cliente que pode verificar o caminho percorrido desde a matéria prima ao beneficiamento. O casos de uso são detalhados a seguir:

- **UC17 – Cadastra beneficiamento**
  - **Ator:** Administrativo
  - **Descrição:** O setor Administrativo cadastra o beneficiamento para que a produção possa começar o processo.
- **UC18 – Cadastra Produto**
  - **Ator:** Produção
  - **Descrição:** O setor de produção cadastra os dados do produto beneficiado.
- **UC19 – Seleciona matérias primas**
  - **Ator:** Produção
  - **Descrição:** O setor de produção seleciona as matérias primas e cadastra as mesmas no processo de beneficiamento
- **UC20 – Verifica o caminho percorrido pelo produto**
  - **Ator:** Cliente
  - **Descrição:** O cliente deve poder verificar o caminho percorrido pelo produto desde o apiário ao beneficiamento

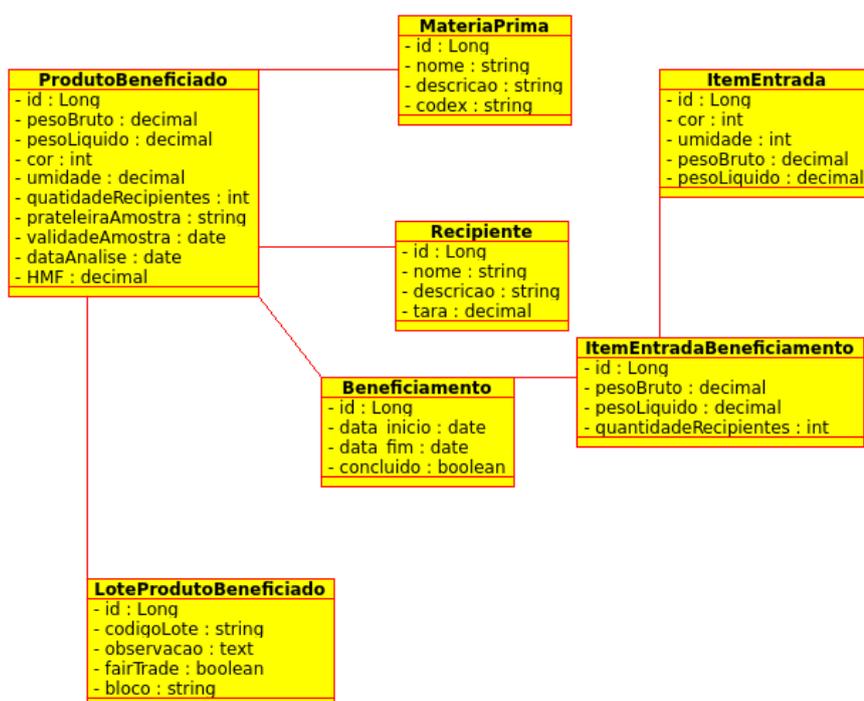


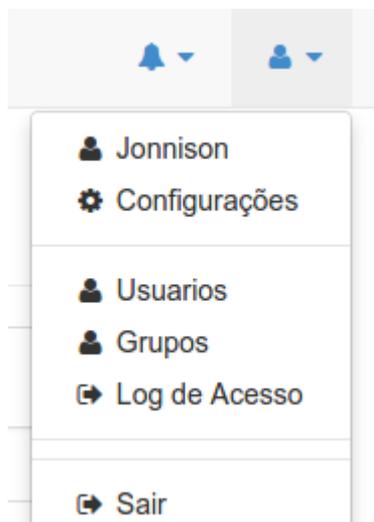
Figura 9 - Diagrama de classes beneficiamento

Fonte: O autor(2015)

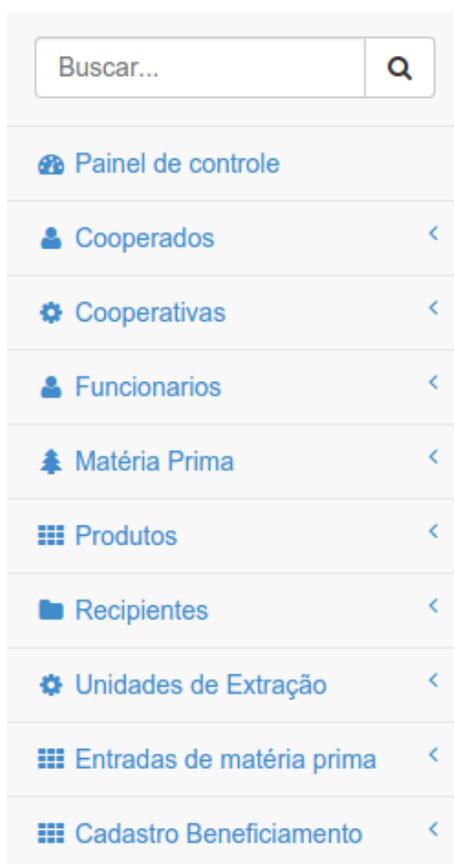
A Figura 9 mostra o diagrama de classes do subsistema de beneficiamento. Onde a classe “Beneficiamento” que representa o processo de beneficiamento de matéria prima, se relaciona com a classe “ProdutoBeneficiado” que representa o produto gerado pelo processo de beneficiamento, a classe “ProdutoBeneficiado” se relaciona com “LoteProdutoBeneficiado”, que representa os dados do lote produzido. O beneficiamento possui itens, representados no sistema pela classe “ItemBeneficiamento”, o item do beneficiamento é composto por partes do item de entrada descrito no subsistema de entrada.

## 4.2 Resultados

Com base nos requisitos analisados na subseção 4.1 foram desenvolvidos os três subsistemas com suas funcionalidades específicas. No decorrer deste item serão apresentas as principais telas do sistema, explicando suas principais funcionalidades. As Figuras 10 e 11 ilustram os menus com todas as funcionalidades disponíveis no sistema.



**Figura 10 - Menu do usuário**  
Fonte: O autor(2015)



**Figura 11 - Menu da aplicação**  
Fonte: O autor(2015)

A figura 11 ilustra o menu de acesso a todas as funcionalidades do sistema desenvolvido, permitindo assim que o usuário realize as atividades especificadas na subseção 4.1, atendendo dessa maneira os requisitos da empresa. A Figura 10 mostra o menu de acesso as configurações de usuário, onde o mesmo pode configurar seus dados pessoais, ver seu *log* de acesso, gerenciamento de usuários e grupos de usuários.

O subsistema de autenticação é responsável por fornecer o controle de acesso as funcionalidades, definindo que usuários podem acessar determinada funcionalidade. Esse módulo possui controle de acesso baseado em permissões que podem ser definidas em grupo ou para um usuário específico.

Selecione usuário para modificar

Ação: ----- Fazer 0 de 2 selecionados Adicionar usuário

<input type="checkbox"/>	Usuário	Endereço de email	Primeiro nome	Último nome	Membro da equipe
<input type="checkbox"/>	jonnison	jonnison@institutomulticom.com	Jonnison	Ferreira	✔
<input type="checkbox"/>	teste				✘

2 usuários

**Figura 12 - Lista de usuários cadastrados**

**Fonte: O autor(2015)**

A Figura 12 mostra os usuários do sistema, com suas principais informações e algumas ações simples como opção de excluir ou desativar usuários, além do acesso a opção de cadastro de novos usuários.

## Modificar usuário

Informações pessoais | **Permissões** | Datas importantes

### Permissões

**Ativo**  
Indica que o usuário será tratado como ativo. Ao invés de excluir contas de usuário, desmarque isso.

**Membro da equipe**  
Indica que usuário consegue acessar este site de administração.

**Status de superusuário**  
Indica que este usuário tem todas as permissões sem atribuí-las explicitamente.

**Permissões do usuário:**  
admin | entrada de log | Can add log entry  
admin | entrada de log | Can change log entry  
admin | entrada de log | Can delete log entry  
auth | grupo | Can add group

Permissões específicas para este usuário. Mantenha o "Control", ou "Command" no Mac, pressionado para selecionar mais de uma opção.

**Grupos:** Admin, Produção

Os grupos que este usuário pertence. Um usuário terá todas as permissões concedidas a cada um de seus grupos. Mantenha o "Control", ou "Command" no Mac, pressionado para selecionar mais de uma opção.

Apagar | Salvar e adicionar outro(a) | Salvar e continuar editando | **Salvar**

**Figura 13 - Tela de cadastro de usuário**

Fonte: O autor(2015)

A Figura 13 mostra a tela de cadastro de usuário, onde é possível cadastrar seus dados pessoais, e as suas permissões, assim como selecionar grupos de permissões a serem aplicadas ao usuário atual, o cadastro de grupos de usuário pode ser acessado pelo com mostrado na figura 10.

O subsistema de entrada é composto pelo cadastro de entrada a cooperativa e todos os demais cadastros que o completam. A Figura 14 mostra parte da tela de cadastro de cooperado, onde é possível cadastrar os membros da cooperativas, os demais dados necessários podem ser acessados pelo menu da aplicação ilustrado na Figura 11.

### Adicionar Cooperado

Codigo do IBD:

Bairro:

Complemento:

RG:

Observacao:

Nome:

Cidade:

CEP:

Orgao do RG:

Endereco:

Estado:

Data de inicio da certificacao:

CPF:

Possui contrato com empresa de comercializacao:

Data de vinculo:

Numero DAP:

Membro da associacao Cooperativa:

Modalidade da apicultura:

Situacao:

TipoDAP:

**Figura 14 - Tela de cadastro de cooperados**

Fonte: O autor (2015)

As Figuras 15 e 16 mostram a principal tela do subsistema, o cadastro de entrada, onde é cadastrado os dados referentes a entrada e os dados referente as cada item de entrada, como cooperado, umidade, peso, cor, data da análise e demais itens. Os cadastros auxiliares ao cadastro de entrada podem ser acessados através do menu ilustrado na Figura 11.

## Adicionar Entrada

<b>Numero da nota fiscal:</b> <input type="text" value="teste"/>	<b>Responsavel:</b> teste <input type="text" value="Escreva algum texto para buscar e autocomple"/>	<b>Data do recebimento:</b> <input type="text" value="21/12/2014"/>
<b>Termo:</b> <input type="button" value="Escolher arquivo"/> Nenhum arqu...elecionado	<b>MateriaPrima:</b> <input type="text" value="mel"/>	<b>Data da analise:</b> <input type="text" value="21/12/2014"/>
<b>Observação:</b> <div style="border: 1px solid #ccc; height: 100px;"></div>	<input type="button" value="Salvar"/>	<b>Cooperativa:</b> <input type="text" value="teste"/>

**Figura 15 - Tela de cadastro de entrada**

Fonte: O autor (2015)

## Itens Entrada

ID	Cooperado	Peso Bruto	Cor	Umidade	Opções
1	teste	20,0000000000	20	20,0000000000	<a href="#">Editar</a>

<b>Cooperado:</b> <input type="text" value="-----"/>	<b>Peso Bruto:</b> <input type="text"/>	<b>Peso líquido:</b> <input type="text"/>
<b>Cor:</b> <input type="text"/>	<b>Umidade:</b> <input type="text"/>	<b>Termo:</b> <input type="button" value="Escolher arquivo"/> Nenhum arqu...elecionado
<b>FairTrade:</b> <input type="checkbox"/>	<b>Bloco:</b> <input type="text"/>	<b>Conferido:</b> <input type="checkbox"/>
<b>Observações:</b> <div style="border: 1px solid #ccc; height: 100px;"></div>	<b>Quantidade de recipientes:</b> <input type="text"/>	<b>Recipiente:</b> <input type="text" value="-----"/>
	<b>DataAnalise:</b> <input type="text"/>	<b>HMF:</b> <input type="text"/>
	<b>Prateleira da amostra:</b> <input type="text"/>	<b>Data de validade da amostra:</b> <input type="text"/>

**Figura 16 - Tela de cadastro de itens de entrada**

Fonte: o autor(2015)

O subsistema de beneficiamento é composto pelo cadastro de beneficiamento e os demais cadastros de que o auxiliam, todos podem ser acessados pelo menu da aplicação da Figura 11. A tela de cadastro de beneficiamento é ilustrado na Figura 17, onde é possível cadastrar os dados referentes ao beneficiamento e os itens que o compõem.

## Cadastro Beneficiamento

Data de inicio: 
 Data do fim: 
 MateriaPrima:

Concluido:

## Itens Entrada Beneficiamento

ID	Item Entrada	Peso Bruto	Peso Líquido	Quantidade de Recipientes	Opções
2	1	10,000	9,000	5	<a href="#">Remover</a>
3	1	10,000	9,000	5	<a href="#">Remover</a>

ItemEntrada: 
 Peso bruto: 
 Peso líquido:

Quantidade de recipientes:

**Figura 17 - Tela de cadastro de beneficiamento**

**Fonte: O autor (2015)**

Todas as funcionalidades apresentadas, atualmente, estão em fase de implantação, onde os usuário recebem treinamento e começam a entender e utilizar o sistema.

## 5 CONCLUSÃO E TRABALHOS FUTUROS

Antes do desenvolvimento do projeto, a Casa Ápis não encontrou nenhuma alternativa de sistema no mercado que atendesse a sua real necessidade. Sendo a maior dificuldade o gerenciamento e acompanhamento das tarefas rotineiras da empresa. Uma solução seria um sistema que atendesse todos os problemas específicos da instituição.

Para solucionar os problemas especificados no decorrer do projeto, foi desenvolvido um sistema *web* para gerenciamento da produção de mel orgânico, permitindo assim que os usuários acessem o sistema de qualquer lugar com acesso à *internet*.

Durante o projeto foi especificado todo o seu processo de desenvolvimento do sistema, começando pelo estudo do problema a ser solucionado, ou seja, o estudo da produção de mel. Em seguida, foi descrito o processo de análise de requisitos e implementação do sistema, por último foram mostradas as principais telas do sistema. Atualmente, o projeto encontra-se na fase final de teste de implantação, onde os usuários passam pelo treinamento e utilizam o sistema no dia a dia, por isso os resultados são a descrição das funcionalidades desenvolvidas e testadas com base nos requisitos.

Em relação aos trabalhos futuros, têm-se como objetivos, o desenvolvimento de um aplicativo *mobile* para que se possa colher os dados que ficam em locais isolados, sem acesso a internet, como os apiários, por exemplo. Também é possível o uso de técnicas de mineração de dados e *BigData* para que possa obter previsões quantitativas e qualitativas da produção. Existe a necessidade do desenvolvimento de um módulo de controle financeiro.

## REFERENCIAS

AUGUSTO, E. A. A. Georreferenciamento de Imóveis Rurais. São Paulo, IRIB, 2006.

BALDUINO, P. Dominando JavaScript com JQuery. Casa do Código, São Paulo, Brasil, 2012.

BRAUDE, E. Projeto de Software. Da programação à arquitetura: uma abordagem baseada em Java. Bookman, Porto Alegre, Brasil, 2005.

BECK, K.; BEEDLE, M.; BENNEKUM, A. V.; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; GRENNING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R. C.; MELLOR, S.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. Manifesto para Desenvolvimento Ágil de Software, 2001, disponível em <<http://agilemanifesto.org/iso/ptbr/>>, acessado em: 25 de maio de 2015

BORGES, L. E. Python para desenvolvedores. Rio de Janeiro: 2ª edição, 2010, disponível em <<http://ark4n.wordpress.com/python/>>, Acesso em 17 de julho de 2014.

EIS, D.; FERREIRA, E.. HTML5 e CSS3: com farinha e pimenta. Tableless, São Paulo, Brasil, 2012.

FILHO, W. de P. P.. Engenharia de Software: Fundamentos, Métodos e Padrões. LTC, ro, Brasil, 2003.

GUEDES, G. T. A. UML 2: Uma Abordagem Prática. Novatec, São Paulo, Brasil, 2011.

MILANI, A. PostgreSQL - Guia do Programador - São Paulo: Novatec,. 2008.

NIEDERAUER, J. Desenvolvendo web sites com PHP. Novatec, São Paulo, Brasil, 2011.

PICOLI, P. O. PROJETO PRO-MEL. Texto publicado na revista Mensagem Doce, edição nº 56, maio de 2000.

QUINTELLA, C. M. Editorial. Cadernos de Prospecção. v.1, p.3 - 3, 2008.

QUINTELLA, C. M., CASTRO, M. P. Glicerina Bruta (GB) oriunda da produção de biodiesel: transformando este subproduto em co-produto com alto valor agregado e baixo custo preparativo, uma oportunidade de negócio. Bahia Análise & Dados, v.18, p.635 - 646, 2009.

QUINTELLA, C. M., COSTA NETO, P. R., CRUZ, R. S., Almeida Neto, J. A., Miyazaki, S. F., CASTRO, M. P. Prospecção Tecnológica do Biodiesel no Estado da Bahia: Panorama Atual e Perspectivas na Geração e Apropriação de Conhecimento. Bahia Análise & Dados. , v.18, p.581 - 591, 2009.

QUINTELLA, C. M., TEIXEIRA, L. S. G., KORN, M. G. A., COSTA NETO, P. R., TORRES, E. A., CASTRO, M. P., JESUS, C. A. C. Cadeia do biodiesel da bancada à indústria: uma visão geral com prospecção de tarefas e oportunidades para P&D&I. Química Nova. , v.32, p.793 - 808, 2009.

REIS, C. R. Python na Prática: Um curso objetivo de programação em Python, abril 2010. Disponível em: < <http://www.async.com.br/projects/pnp/pnp.pdf> >. Acesso em: 17 julho 2014.

SAMARA, B. S.; MORSCH, M. A. Comportamento do consumidor: conceitos e casos. São Paulo: Prentice Hall, 2005.

SANTANA, O. e GALESI, T. Python e Django. São Paulo: Novatec, 2010.

SILVA, M. S. Criando sites com HTML: sites de alta qualidade com HTML e CSS. Novatec, São Paulo, Brasil, 2008.

SILVA, M. S. CSS3: Desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. Novatec, São Paulo, Brasil, 2011.

SILVA, M. S. JQuery: A Biblioteca do Programador JavaScript. Novatec, São Paulo, Brasil, 2008.

SILVA, M. S. Scrum: Gestão Ágil para Projetos de Sucesso. Casa do Código, São Paulo, Brasil, 2013.

SOUSA, A. R. R. de. Central de Cooperativas de Cajucultores do Estado do Piauí COCAJUPI. COCAJUPI Picos. 19 de março de 2009. Disponível em < <http://www.cocajupi.net/quemsomos.php>> Acesso em 26 de setembro de 2009.

STARK, J.; JEPSON, B. Construindo Aplicativos Android com HTML, CSS e JavaScript. Novatec, São Paulo, Brasil, 2012.

TIOBE. Software: Tiobe Index. TIOBE SOFTWARE - The coding standards company, julho 2014. Disponível em: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> >. Acesso em: 17 julho 2014.

VARASCHIM, J. D. Implantando o SCRUM em um Ambiente de Desenvolvimento de Produtos para Internet. Pontifícia Universidade Católica Do Rio De Janeiro. Rio de Janeiro, 2009.