

**Gabriel Sena San Martin, Pedro Vinicius de Oliveira Pequeno,  
Victor Yan Pereira e Lima, Wyller Douglas Martins de Sousa**

# **Chicken Tracker**

**Brasil**

**Julho de 2018**

**Gabriel Sena San Martin, Pedro Vinicius de Oliveira Pequeno,  
Victor Yan Pereira e Lima, Wyller Douglas Martins de Sousa**

## **Chicken Tracker**

Instituto Federal de Educação, Ciência e Tecnologia do Amazonas – IFAM  
Ensino Médio-Técnico em Informática

Orientador: Emmerson Santa Rita

Brasil  
Julho de 2018

## Resumo

A avicultura está crescendo cada vez mais na exportação brasileira, participando ativamente do Produto Interno Bruto (PIB) do Brasil. Assim, o agrobusiness de aves e ovos está tendo um maior número de produção e de campos destinados a essa produção, com isso a necessidade de organização dos avicultores aumenta de forma significativa. O Chicken Tracker propõe uma solução online. Um website que trata da organização interna do tratamento dos produtos pertencentes ao ramo da avicultura, controlando os custos envolvendo fornecedores e recursos. Além disso, um espaço para cadastrar os galpões e incubadoras de ovos, com especificações dos recursos necessários para manter mensalmente esse espaço. O proprietário terá a possibilidade de olhar seu negócio através de gráficos que mostram se o custo destinado à produção e o lucro estão coerentes com seu objetivo, assim como outros gráficos que mostram se a quantidade produzida está sendo totalmente aproveitada.

**Palavras-chaves:** Avicultura; website; organização; gráficos; agrobusiness.

## **Abstract**

Poultry farming is growing more and more in Brazilian exports, actively participating in Brazil's Gross Domestic Product (GDP). Thus, the agribusiness of poultry and eggs is having a greater number of production and fields destined to this production, with this increases the need of greater organization of the poultry farmers. Chicken Tracker proposes an online solution. A website that deals with the internal organization of the treatment of the products belonging to the poultry industry, controlling the costs involving suppliers and resources. In addition, a space to register the egg hatchery and incubators, with specifications and resources necessary to maintain this space monthly. The owner will be able to look at his business through charts that show if the cost for production and profit are consistent with his goal, as well as other graphs that showed if the quantity produced is being fully taken advantage of.

**Key-words:** Poultry farming; Website; organization; charts;

## Lista de ilustrações

Figura 1 – MVC (fonte tutorialspoint) . . . . .	12
Figura 2 – Servlet e JSP (fonte IBM) . . . . .	14
Figura 3 – Diagrama de caso de uso (fonte própria, feito no Astah) . . . . .	19
Figura 4 – Modelo Entidade-Relacionamento 1 (fonte própria, feito no Astah). .	30
Figura 5 – Modelo Entidade-Relacionamento 2 (fonte própria, feito no Astah). .	31
Figura 6 – Modelo Relacional de tabelas (fonte própria, feito no MySQL work- bench). . . . .	32
Figura 7 – Modelo MVC do projeto (fonte própria, feito no Canvas). . . . .	37
Figura 8 – Diagrama de classes (fonte própria, feito no Astah). . . . .	38
Figura 9 – Diagrama de pacotes (fonte própria, feito no Astah). . . . .	39
Figura 10 – Diagrama web - 1 (fonte própria, feito no Astah). . . . .	40
Figura 11 – Diagrama web - 2 (fonte própria, feito no Astah). . . . .	41
Figura 12 – Diagrama web - 3 (fonte própria, feito no Astah). . . . .	41
Figura 13 – Diagrama web - 4 (fonte própria, feito no Astah). . . . .	42
Figura 14 – Diagrama de sequencia Cadastrar Usuário (fonte própria, feito no Astah). . . . .	43
Figura 15 – Diagrama de sequencia Fazer Login (fonte própria, feito no Astah). .	43
Figura 16 – Diagrama de sequencia Manter Perfil (fonte própria, feito no Astah). .	44
Figura 17 – Diagrama de sequencia Manter Negocio (fonte própria, feito no Astah). .	45
Figura 18 – Diagrama de sequencia Manter Estabelecimento (fonte própria, feito no Astah). . . . .	46
Figura 19 – Diagrama de sequencia Manter Produto (fonte própria, feito no Astah). .	47
Figura 20 – Diagrama de sequencia Manter Fornecedores (fonte própria, feito no Astah). . . . .	48
Figura 21 – Diagrama de sequencia Manter Fornecimento (fonte própria, feito no Astah). . . . .	49
Figura 22 – Diagrama de sequencia Manter Funcionários (fonte própria, feito no Astah). . . . .	50
Figura 23 – Diagrama de sequencia Manter Vacina (fonte própria, feito no Astah). .	51
Figura 24 – Diagrama de sequencia Pesquisar Negócio (fonte própria, feito no Astah). . . . .	51

## **Lista de abreviaturas e siglas**

IDE	Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
JSP	JavaServer Pages
MER	Modelo Entidade-Relacionamento
MR	Modelo Relacional
MVC	Model-View-Controller (Modelo-visão-controlador)
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language (Linguagem de Consulta Estruturada)
CASE	Computer-Aided Software Engineering (Engenharia de Software Assistida por Computador)
UML	Unified Modeling Language (Linguagem de Modelagem Unificada)

## Sumário

<b>Lista de ilustrações</b>	<b>4</b>
<b>1 Introdução</b>	<b>8</b>
<b>2 Descrição do problema</b>	<b>8</b>
<b>3 Justificativa</b>	<b>8</b>
<b>4 Objetivos</b>	<b>9</b>
4.1 Objetivo geral	9
4.2 Objetivos específicos	9
<b>5 Trabalhos relacionados</b>	<b>9</b>
5.1 AGROFRANGO	9
<b>6 Fundamentação Teórica</b>	<b>9</b>
<b>7 Materiais e Métodos</b>	<b>11</b>
7.1 Modelo iterativo e incremental	11
7.2 Scrum	11
7.3 MVC	12
7.4 Singleton	12
7.5 Projeto Maven	13
7.6 HTML5	13
7.7 CSS3	13
7.8 JavaScript	13
7.9 Java EE	14
7.10 SQL	14
7.11 Hibernate	15
7.12 Bootstrap	15
7.13 JQuery	15
7.14 Charts.js	15
7.15 NetBeans	16
7.16 TomCat	16
7.17 MySQL	16
7.18 Astah	17
7.19 brModelo	17
<b>8 Metodologia</b>	<b>17</b>
<b>9 Modelagem do Sistema</b>	<b>17</b>
9.1 Requisitos	17
9.1.1 Requisitos funcionais	18
9.1.2 Requisitos não funcionais	18
9.2 Diagrama de caso de uso	18

9.3	Modelo Entidade-relacionamento . . . . .	29
9.4	Modelo relacional de tabelas . . . . .	31
9.5	Dicionário de dados . . . . .	32
9.6	Diagrama de classes . . . . .	36
9.7	Diagrama de pacotes . . . . .	39
9.8	Diagramas web . . . . .	40
9.8.1	Diagrama Web - 1 . . . . .	40
9.8.2	Diagrama web - 2 . . . . .	41
9.8.3	Diagrama web - 3 . . . . .	41
9.8.4	Diagrama web - 4 . . . . .	42
9.9	Diagrama de Sequência . . . . .	42
9.9.1	Diagrama de sequência Cadastrar Usuário . . . . .	42
9.9.2	Diagrama de sequência Fazer Login . . . . .	43
9.9.3	Diagrama de sequência Manter Perfil . . . . .	44
9.9.4	Diagrama de sequência Manter Negócio . . . . .	44
9.9.5	Diagrama de sequência Manter Estabelecimentos . . . . .	45
9.9.6	Diagrama de sequência Manter Produto . . . . .	46
9.9.7	Diagrama de sequência Manter Fornecedores . . . . .	47
9.9.8	Diagrama de sequência Manter Fornecimento . . . . .	48
9.9.9	Diagrama de sequência Manter Funcionário . . . . .	49
9.9.10	Diagrama de sequência Manter Vacina . . . . .	50
9.9.11	Diagrama de sequência Pesquisar Negócio . . . . .	51
<b>10</b>	<b>Projeto de telas . . . . .</b>	<b>52</b>
<b>11</b>	<b>Cronograma . . . . .</b>	<b>53</b>
<b>12</b>	<b>Informações adicionais . . . . .</b>	<b>54</b>
<b>13</b>	<b>Resultados esperados . . . . .</b>	<b>54</b>
<b>14</b>	<b>Conclusão - considerações finais . . . . .</b>	<b>54</b>
	 <b>Referências . . . . .</b>	 <b>55</b>



## **1 Introdução**

Desde a antiguidade, a carne de frango e seus ovos são bastante consumidos e são considerados uns dos alimentos mais proteicos. No Brasil, a região sudeste foi a primeira a ter presença de produtores familiares que trabalhavam com o tratamento de aves para sua subsistência. Com o passar do tempo, a carne foi mais consumida e mais produtores se dedicaram ao ramo, integrando outros tipos de produção e industrialização na prática de avicultura. A partir da década de 70, começaram a exportação da carne de aves no Brasil e, no começo do século XXI, o país é considerado um dos maiores produtores de carne de aves, chegando a ser o segundo maior produtor até o ano de 2016, atrás somente dos EUA, de acordo com o Departamento de Agricultura do Estados Unidos (USDA).

Com o mercado interno e externo solicitando milhões de quilos da carne de aves, os avicultores utilizam cada vez mais a tecnologia para melhorar a qualidade e produção da carne e ovos de aves. A Associação Brasileira de Proteína Animal (ABPA) descreveu no seu site: “Nas granjas brasileiras, a excelência tecnológica em genética, manejo e ambiência garantiram saltos produtivos que colocaram o país como terceiro maior produtor mundial de carne de frango, com mais de 12 milhões de toneladas anuais de carne de frango”.

Porém, no Brasil, não existe um sistema completo para suprir as necessidades de um avicultor. Um sistema que o produtor possa controlar seus funcionários, produtos, galinheiros, fornecedores, entre outros. Esse sistema deve ser acessível em qualquer plataforma.

## **2 Descrição do problema**

Para um avicultor ter um grande sucesso, a organização do espaço de trabalho é muito importante. Fazer o controle da quantidade e do pagamento de funcionários, estado de máquinas, a qualidade do ambiente onde ficam as aves de postura e de corte, assim como verificar seus custos com estoque e seu lucro com a produção, planejando, dessa forma, as suas decisões futuras.

As informações costumam ser assinadas em papéis, ou seja, necessita de espaço para armazenamento e tempo para procurar informações.

## **3 Justificativa**

Para produção de aves de postura e de corte é necessário ter um grande ambiente para tratamento da produção, contando com uma ambientação e tecnologia para aumento da produção, assim como diminuição das perdas por doenças e má planejamento. Com isso, avicultores precisam organizar seu espaço de trabalho, de maneira a controlar seu espaço interno e seus gastos, assim como verificar se duas decisões e planejamentos estão sendo lucrativas e qualidade para seus clientes.

Com um website, proporcionando a visualização em qualquer dispositivo conectado a internet, o produtor terá total controle do seu trabalho, contando com um controle de estoque e fornecedores, da mesma maneira que controlará gastos com funcionário, equipamentos, recursos (alimentação para aves, medicamentos, luz, água), fornecedores desses recursos.

O avicultor ainda poderá aproveitar o recurso de vendas de produtos em atacado online e pessoas jurídicas cadastrando-se no website e solicitando ao avicultor para ser um fornecedor, com isso o produtor terá controle de seus ganhos mensalmente e quando terá que produzir para manter seus contratos com as empresas, do mesmo modo que terá domínio sobre seu estoque quando terá que gastar para manter com o necessário para produção.

Nesse modo, o website será importante para pequenos, médios e grandes avicultores para sua organização administrativa e ter seus resultado em gráficos para estudar suas futuras ações.

## **4 Objetivos**

### **4.1 Objetivo geral**

Criar uma plataforma web que possibilita ao avicultor controlar seu negócio.

### **4.2 Objetivos específicos**

- Manter produtos e fornecimentos;
- Manter funcionários e gastos;
- Manter estabelecimentos;
- Possibilitar visualização de gráficos sobre o desempenho mensal/anual;

## **5 Trabalhos relacionados**

### **5.1 AGROFRANGO**

O software AgroFrango foi desenvolvido para atender avicultores de grande, médio e pequeno porte. Estruturado com objetivo de oferecer ao avicultor uma ferramenta simples, intuitiva e que pudesse englobar toda a gestão da criação das aves em um único sistema, permitindo assim o controle de informações zootécnicas e de custos de forma otimizada e segura. (AGROFRANGO, 2018)

## **6 Fundamentação Teórica**

O Brasil, em 2016, chegou a ser o segundo maior produtor de carne de aves de acordo com a USDA. O pesquisador (QUEVEDO, 2003), no site Avicultura Industrial,

descreveu: “O Brasil é hoje o segundo maior produtor e exportador de carne de frango. Produziu em 2002, 7,6 milhões de toneladas e exportou 1,3 milhão. O consumo interno per capita de frango chegou a 34 quilos no ano passado. Já a produção de ovos, ultrapassou 21 bilhões de unidades e o consumo per capita atingiu 125 unidades.”

Na pesquisa feita por (FRANCO, 2017), ela descreve o que esperar do mercado nesses tempos:

“[...] é esperada a expansão da produção de carne de frango no Brasil, apesar dos altos custos e contração no consumo interno. O USDA estima aceleração nas importações mundiais. Estão entre os 10 principais importadores de carne de frango: Japão, México, Arábia Saudita, União Europeia, Iraque, África do Sul, China, Hong Kong, Emirados Árabes e Filipinas. Todavia, o órgão presume que Japão e Arábia Saudita devam reduzir as importações, o que pode impactar as exportações brasileiras. Por outro lado, a China, que tem o Brasil como principal fornecedor, deve elevar em mais de 50% as importações.”

Observando e analisando essas pesquisas, muitos agricultores estarão voltados para esse ramo e o Brasil será um dos maiores produtores do mundo por um bom tempo. Nesse contexto, o Brasil terá que disponibilizar recursos para os avicultores conseguirem manter o mercado interno e externo. A revista (AVISTE, 2011) comentou sobre o assunto:

“Muitas pessoas nem percebem, ou não sabem, que a avicultura tem contribuição fundamental para a melhoria da alimentação dos brasileiros. A carne de frango e o ovo têm sido itens cada vez mais presentes na mesa da população do Brasil, registrando, ano a ano, aumentos significativos no consumo. Outro aspecto social destacável no setor é o uso de mão-de-obra intensiva, o que, quase com certeza, torna a avicultura a maior geradora de empregos de toda a agropecuária. Ou será que existe outra atividade na agricultura ou na produção animal que ocupe, direta e indiretamente, um contingente estimado em 4,5 milhões de pessoas?”

O Brasil não é um dos maiores produtores apenas por sorte, ele também é um dos maiores investidores nesse ramo. O governo propõe leis para atrair atenção de pessoas para esse ramo, do mesmo modo que investe na tecnologia para o aumento da produtividade e do lucro, “O Brasil é hoje o segundo maior produtor e exportador de carne de frango. Produziu em 2002, 7,6 milhões de toneladas e exportou 1,3 milhão. O consumo interno per capita de frango chegou a 34 quilos no ano passado. Já a produção de ovos, ultrapassou 21 bilhões de unidades e o consumo per capita atingiu 125 unidades.”, diz (QUEVEDO, 2003) na sua pesquisa.

A implantação de um website nesse mundo de negócios qualifica o avicultor para ter seu negócio mais visto positivamente, como se fosse uma estratégia de mar-

keting, da mesma forma, um website demonstra confiabilidade para outras empresas ou avicultores. A (SITEPT, 2018) comenta sobre a ideia de ter um website para seu negócio: “Ter um site é muito mais barato, flexível e duradouro do que as formas tradicionais de publicidade. Não deixe que o custo inicial o assuste. O website da sua empresa terá toda a informação que quiser que tenha, não tem limitação de espaço e pode ser organizada da forma que melhor entender. Ao contrário de um panfleto, por exemplo, este cartão-de-visita interativo terá toda a informação sempre atualizada, e o conteúdo pode ser alterado de forma a melhor satisfazer a sua estratégia de negócio. Além disso, estará permanentemente acessível, podendo acumular a informação ou substituir a desatualizada por outra mais recente.”

## **7 Materiais e Métodos**

Neste capítulo, será demonstrado as tecnologias, as ferramentas e os padrões de arquitetura e de projeto utilizados no sistema.

### **7.1 Modelo iterativo e incremental**

O modelo incremental é baseado na entrega do produto em partes. Cada parte do produto entregue é uma parte funcional do sistema, assim sendo possível verificar e utilizar o sistema em pequenas partes, mesmo que o projeto ainda esteja em desenvolvimento.

O modelo iterativo permite a criação de um modelo de solução para cada parte funcional por vez e testá-lo. Após o teste, será possível verificar falhas ou perceber que houve um mal entendimento sobre a funcionalidade, assim, após essas descobertas, planejar e entregar um novo modelo de solução a partir do antigo. E continua dessa forma até a parte funcional ser totalmente o que se esperava pelo usuário.

### **7.2 Scrum**

O Scrum é uma metodologia ágil para planejamento e gestão de projetos de software. Nele, é utilizado um ciclo de vida chamado *sprints*, onde é mostrado uma parte funcional do projeto em cada um deles, aplicando assim o modelo incremental.

A metodologia Scrum é formada pelo Scrum Master que controla as atividades da equipe de programadores. Ele é responsável pela liderança e pelo andamento do projeto.

O time de desenvolvimento ou equipe de programadores é um conjunto de pessoas que planejam, programam e testam a ferramenta. E existe um Product Owner que define o projeto e suas funcionalidades, ou seja, quem pede o software.

Existe uma lista que é conhecida como *Product Backlog* que armazena as funcionalidades a serem desenvolvidas. No início de cada sprint, faz-se um Sprint Planning

Meeting, ou seja, uma reunião de planejamento na qual o Product Owner prioriza os itens do Product Backlog e a equipe seleciona as atividades que ela será capaz de implementar durante o sprint que se inicia. As tarefas alocadas em um sprint são transferidas do Product Backlog para o Sprint Backlog.

Todo dia, a equipe faz uma breve reunião chamada de Daily Scrum. O objetivo é entender e conhecer sobre o que foi feito no dia anterior, identificar falhas ou erros e priorizar o trabalho do dia que se inicia.

Ao final de um sprint, a equipe apresenta as funcionalidades implementadas em uma *Sprint Review Meeting*. Finalmente, faz-se uma *Sprint Retrospective* e a equipe parte para o planejamento do próximo sprint. E assim o ciclo se repete.

### 7.3 MVC

O padrão de arquitetura MVC prioriza separar o projeto em três camadas, que são: Modelo, Controlador e Visão.

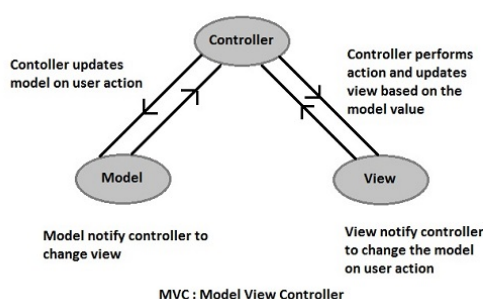
O modelo representa cada entidade do banco de dados como uma classe, ou seja, são as classes que permite controlar os dados do sistema.

A visão é a superfície do sistema, isto é, a "parte bonita". Esse modelo representa a interface e a interação que o usuário tem com o sistema.

O controlador é justamente o intermediário entre essas camadas. Neste modelo é implantado as regras de negócio e ligar a interface com parte mais interna do sistema, as classes modelos e a de conexão com o banco de dados.

A imagem abaixo mostra justamente como ocorre a comunicação entre os modelos:

Figura 1 – MVC (fonte tutorialspoint)



### 7.4 Singleton

O Singleton é um padrão de projeto que tem como objetivo o gerenciamento de memória, onde cada classe tem apenas uma única instância.

Na orientação a objetos, as classes são modelos de objetos. Para utilizarmos

um objeto, precisamos instanciá-lo, ou seja, reservar um espaço de memória para armazenarmos o valor do objeto e seus atributos. Porém, é possível uma classe ter vários instâncias.

No padrão Singleton, só existe um objeto para cada classe, assim deixando o sistema leve e com menos necessidade de memória.

## 7.5 Projeto Maven

Todo projeto necessita de um modelo de organização de arquivos e bibliotecas para compilação. Porém, dependendo do tamanho do projeto, existirão vários arquivos e bibliotecas para serem baixadas (algumas, configuradas).

O projeto Maven visa justamente facilitar esse processo. Ele conta com o recurso de *archetype*, onde existem vários modelos de organização de arquivos, dependendo do tipo de projeto.

Outro recurso do Maven muito importante é justamente o controle de dependências do projeto, ou seja, as bibliotecas. Ele consegue criar um repositório único local para todas as dependências que poderá ser utilizada em vários projetos. Esse controle ainda permite fazer o download automaticamente e colocá-las no local certo.

## 7.6 HTML5

A linguagem de marcação HTML tem como objetivo estruturar e apresentar o conteúdo na web. É uma linguagem baseada em tags, onde eles dão um valor semântico para cada estrutura de uma página web.

Na última versão do HTML, a linguagem ganhou novas tags e ganhou um novo sentido e objetivo, agora ela deve se preocupar apenas e exclusivamente a semântica, ou seja, o significado de um texto, de um título, de uma imagem para aquela página e aquele site.

## 7.7 CSS3

Com o HTML5 e seu novo objetivo, o CSS3 ganhou novos recursos e trabalha diretamente com o HTML. A função do CSS3 é fazer o estilo da página.

Todas as configurações de fonte, cor, plano de fundo, posição de imagem, entre outras coisas é dever do CSS3 planejar e realizar.

## 7.8 JavaScript

Com a divisão de camadas da web, o HTML5 ficou com a informação, da mesma forma que o CSS3 ficou com a formatação. O JavaScript ficou justamente com o comportamento.

Os scripts são executados ao lado do cliente e interage com o usuário sem a

necessidade deste script passar pelo servidor, assim controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

## 7.9 Java EE

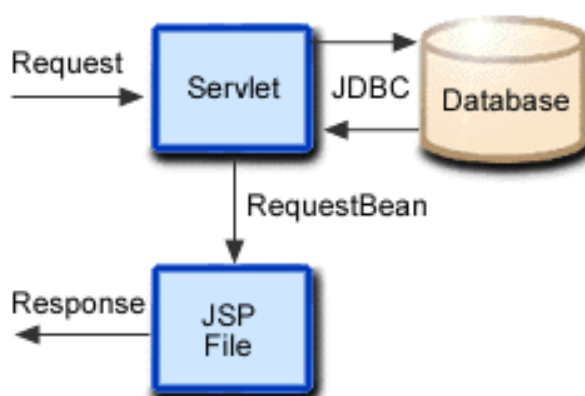
Diferentemente do JavaScript, o Java EE trabalha ao lado do servidor e não do cliente. A tecnologia Java EE trabalha com dois importantes conceitos: Servlets e JSP's.

Servlet significa pequeno servidor. Ele é uma classe-controle que trabalha com as requisições (requests) e respostas (responses) ao usuários. A classe que representa um servlet herda os métodos abstratos de `HttpServlet`, sendo eles os métodos `doGet`, `doPost`, `doPut` e `doDelete` para se responder às requisições feitas pelo usuário.

Porém, a geração de páginas web dinâmicas com servlets dá um pouco de trabalho, já que a resposta é uma página HTML e precisa ser montada "à mão". Por isso a tecnologia de JSP é tão importante para uma página dinâmica, já que trabalha junto com o HTML e o servidor converte para o usuário como se fosse tudo tags HTML.

O JSP é uma mistura da linguagem Java e HTML, assim podendo realizar operações em tempo de execução sem chamar um servlet. Isso é interessante, já que as tarefas mais simples ficam por conta do JSP e a conexão com o banco de dados, por parte do servlet. Isso pode ser representado pela figura a seguir:

Figura 2 – Servlet e JSP (fonte IBM)



## 7.10 SQL

A linguagem SQL é um padrão internacional de linguagem para banco de dados. A padronização dessa linguagem foi muito importante, já existem vários SGBD's com padrões diferentes, isso permite que o engenheiro de software troque de SGBD's, porém com bastante facilidade, já que os dois seguem a mesma base de linguagem.

## 7.11 Hibernate

Quando se faz o mapeamento do MER para a orientação de dados, há um problema. O MER se baseia no modelo relacional, enquanto as linguagens OO, em objetos e classes.

Além disso, escrever código SQL custa tempo de desenvolvimento e manutenção, mais quando há mistura de código Java e SQL, por exemplo, usando estruturas que conectam com o banco de dados, como o JDBC.

O Hibernate é um framework mapeamento objeto-relacional escrito em Java. Por ser escrito na mesma linguagem, integra perfeitamente em qualquer aplicação Java. Além disso, realiza o mapeamento dos objetos para o banco de dados automaticamente, assim como operações essenciais: adicionar, excluir, alterar e consultar.

## 7.12 Bootstrap

O CSS3 cuida da formatação das páginas web. Porém, requer conhecimento e tempo para desenhar um página totalmente, componente por componente. Para facilitar o trabalho de um programador front-end e agilizar o processo de desenvolvimento de interfaces, existem vários templates em HTML, CSS e JavaScript prontos para serem implementados em projetos web.

O Bootstrap é, justamente, um framework web com código-fonte aberto para desenvolvimento front-end de aplicações web usando HTML, CSS e JavaScript. Com ele, é possível criar sites totalmente responsivos de uma forma simples e veloz.

## 7.13 JQuery

jQuery foi desenvolvido por John Resig e lançado em dezembro de 2006. É uma biblioteca de código aberto para funções de JavaScript para interagir com o HTML. Ela facilita escrever scripts em JavaScript para Client-Side.

A sintaxe do jQuery foi desenvolvida para tornar mais simples a navegação do documento HTML, a seleção de elementos DOM, criar animações, manipular eventos, desenvolver aplicações AJAX e criação de plugins sobre ela. Tais facilidades permitem aos desenvolvedores criarem camadas de abstração para interações de baixo nível de modo simplificado em aplicações web dinâmicas de grande complexidade.

## 7.14 Charts.js

Charts.js é uma biblioteca em JavaScript para geração do mais variados gráficos. Ele permite, por exemplo, misturar o gráfico de linhas e barras para clara distinção visual entre conjuntos de dados.

A biblioteca permite plota conjuntos de dados complexos e esparsos no tempo



de data, escalas logarítmicas ou mesmo totalmente personalizadas com facilidade, com transições impressionantes ao alterar dados, atualizar cores e adicionar conjuntos de dados.

## 7.15 NetBeans

O NetBeans IDE é um ambiente de desenvolvimento integrado gratuito e de código aberto para desenvolvedores de software nas linguagens Java, JavaScript, HTML5, CSS3, entre outros. O IDE é executado em muitas plataformas, como Windows, Linux, Solaris e MacOS.

Ele gerencia vários tipos de recursos, como banco de dados e servidores. Além de ser personalizável, possui um autocompletar avançado, ferramenta de automatização da construção de programas e TOMCAT, servidor de aplicações Java para web, suporte a Database (banco de dados), Data view e Connection wizard que são os módulos embutidos na IDE.

## 7.16 TomCat

O Tomcat é um servidor web Java, mais especificamente, um container de servlets. O Tomcat implementa as tecnologias Java Servlet e JSP.

Ele tem a capacidade de atuar também como servidor web, ou pode funcionar integrado a um servidor web dedicado como o Apache ou o IIS. Como servidor web, ele provê um servidor web HTTP puramente em Java.

O servidor inclui ferramentas para configuração e gerenciamento, o que também pode ser feito editando-se manualmente arquivos de configuração formatados em XML.

## 7.17 MySQL

MySQL é um SGBD que utiliza a linguagem SQL para execução das queries. Ele possui portabilidade, compatibilidade, pouco exigente quanto a recursos de novos hardware.

O MySQL é o que permite realizar as operações de salvar, deletar, alterar e consultar qualquer tupla de um banco de dados. Ele é o intermediário entre o programador e o armazenamento de dados.

A Oracle disponibiliza ainda uma ferramenta que integra desenvolvimento, administração, design de banco de dados, criação e manutenção de SQL em um único ambiente de desenvolvimento integrado para o MySQL.

## 7.18 Astah

O Astah é uma ferramenta CASE para criação de diagramas da UML. A ferramenta tem portabilidade para diagramas de classes, diagramas de caso de uso, diagramas de sequência, além de outros.

Além de salvar e abrir, o Astah cria a possibilidade de exportar os diagramas para imagem nos formatos PNG e JPEG.

A ferramenta permite, além de salvar e abrir o diagrama, exportar para imagem nos formatos PNG e JPEG.

## 7.19 brModelo

O brModelo é uma ferramenta gráfica case *freeware* para criação do MER. A ferramenta permite, ainda, exporta para imagem o diagrama MER.

# 8 Metodologia

Para o desenvolvimento do projeto Chicken Tracker foi designado uma equipe com 4 componentes que trabalharam com a metodologia ágil Scrum para a organização e a verificação do progresso do sistema. Toda terça e quinta há uma reunião para percebermos o progresso da equipe e verificar as próximas atividades. As tarefas recebem uma pontuação de acordo com a dificuldade, assim podemos prever o tempo designado para tal tarefa. Todas as atividades, realizadas ou não, estão descritas em uma planilha do Google Planilhas.

O projeto terá duas partes para desenvolvimento, uma visual (front-end) e uma de realização de operações e conexão com banco de dados (back-end). Pelo padrão MVC, o projeto está dividido para manter a separação dos modelos, controles e visões.

Em relação à linguagem de programação utilizada, por se tratar de um sistema web, serão utilizadas o HTML (tratamento da parte conceitual e semântica do sistema), CSS (tratamento da parte visual do projeto), JavaScript (para manipular ações serem realizadas ao decorrer das ações dos usuários), Java Web (Servlets e JSP para tratamento de requisições) e o SQL com o Hibernate( criação e manipulação do banco de dados).

As ferramentas utilizadas serão o NetBeans (IDE para o Java Web), SGBD MySQL e o Astah Community (criação dos diagramas necessários).

# 9 Modelagem do Sistema

## 9.1 Requisitos

Nesse capítulo fala-se sobre o desenvolvimento do projeto e como foi modelado. Os requisitos funcionais e não-funcionais definidos serão mostrados e os diagramas

necessários para apresenta visões de diferentes níveis do software.

### 9.1.1 Requisitos funcionais

- Manter negócio;
- Manter funcionários;
- Manter perfil de usuário;
- Cadastrar usuário no sistema;
- Fazer login;
- Manter produto;
- Manter estabelecimentos;
- Manter fornecedores;
- Manter fornecimento;
- Manter vacinas;
- Demonstrar graficamente o desempenho do negócio/estabelecimento diante o lucro em relação aos gastos e de produção em relação as vendas;

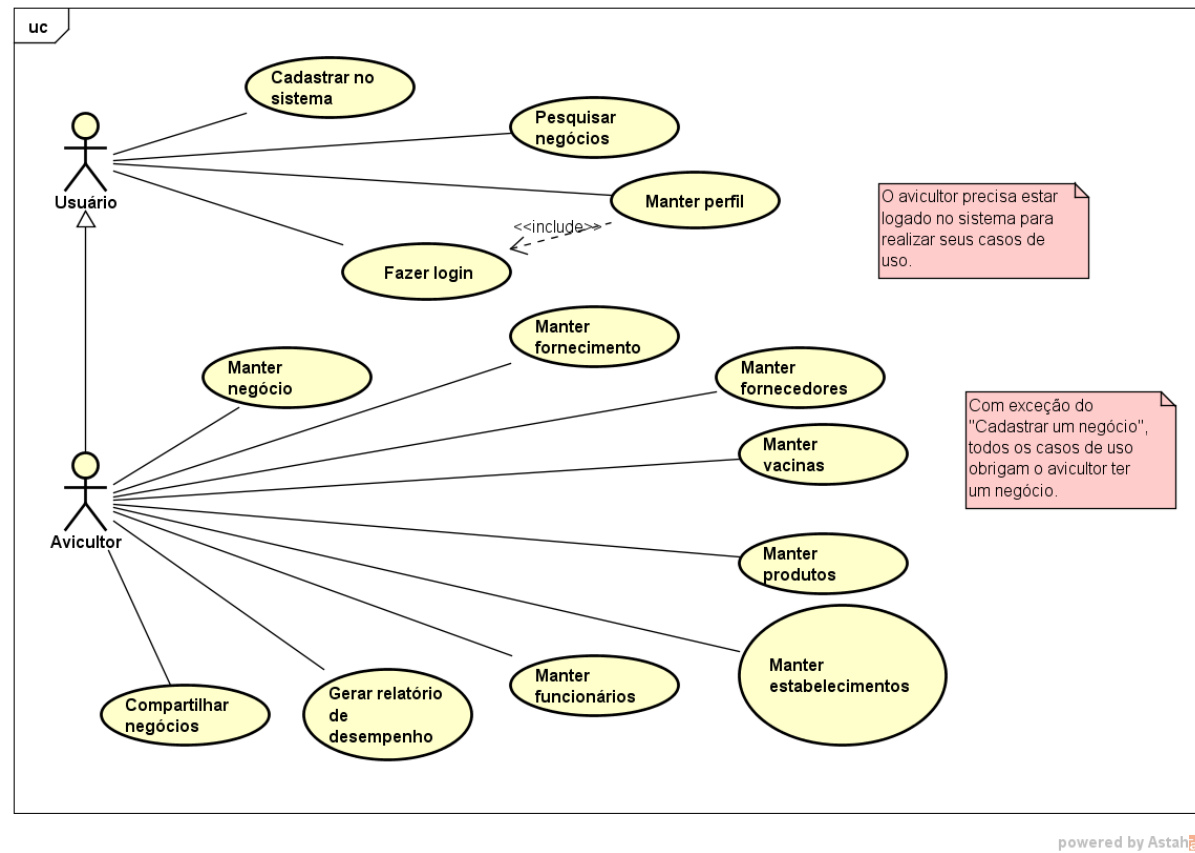
### 9.1.2 Requisitos não funcionais

- No primeiro momento, usar uma criptografia para não mostrar a senha no banco de dados;
- A interface deverá ser montada de forma que fique responsivo e de fácil entendimento.

## 9.2 Diagrama de caso de uso

O diagrama de caso de uso descreve todas as funcionalidades que o sistema possui, assim como os usuários que irão utilizá-las. Também, é possível identificar as obrigações que o usuário deverá ter ou fazer para realizar determinadas funções.

Figura 3 – Diagrama de caso de uso (fonte própria, feito no Astah)



Abaixo estão listadas todos os casos de uso em forma de tabelas, de tal forma que é possível ter uma outra visão cada funcionalidade. Nessa documentação de cada caso de uso é possível entender a sua funcionalidade e como ela funcionará passo a passo.

Tabela 1 – Documentação do caso de uso cadastra no sistema

Nome do caso de uso	Cadastrar no sistema
Caso de Uso Geral	-
Ator principal	Usuário
Ator secundário	-
Resumo	Este caso de uso descreve as etapas necessárias para que o usuário se cadastre no sistema
Pré-condições	-
Pós-condições	-
Fluxo principal	

Ações do Ator	Ações do Sistema
I. Selecionar a opção de cadastrar-se no sistema.	
	II. Carregar o arquivo de cadastro de usuário;
III. Inserir os dados necessários.	
	IV. Validar os campos preenchidos.
	V. Verificar se o usuário já existe.
	VI. Inserir o usuário no banco de dados.
	VII. Realizar o Login do usuário cadastrado
Fluxo alternativo	
Ações do ator	Ações do sistema
	I. Caso o usuário já tenha cadastro, carregar a tela de login.
Fluxo de exceção	
Ações do ator	Ações do sistema
	I. Caso o usuário tenha inserido os dados errados, informar erro e impedir a continuação do processo.

Tabela 2 – Documentação do caso de uso Fazer login

Nome do caso de uso	Fazer login
Caso de Uso Geral	-
Ator principal	Usuário
Ator secundário	-
Resumo	Este caso de uso descreve as etapas necessárias para que o usuário faça o login no sistema
Pré-condições	-
Pós-condições	-

Fluxo principal	
Ações do Ator	Ações do Sistema
I. Selecionar a opção de login no sistema.	
	II. Carregar o arquivo de login.
III. Inserir os dados necessários.	
	IV. Validar os campos preenchidos.
	V. Verificar se os campos estão iguais no banco de dados.
	VI. Realizar o Login do usuário.
Fluxo de exceção	
Ações do ator	Ações do sistema
	I. Caso o usuário tenha inserido os dados errados, informar erro e impedir a continuação do processo.

Tabela 3 – Documentação do caso de uso Manter perfil

Nome do caso de uso	Manter perfil
Caso de Uso Geral	-
Ator principal	Usuário
Ator secundário	-
Resumo	Este caso de uso descreve as etapas necessárias para que o usuário altere ou exclua seu perfil.
Pré-condições	- Estar logado no sistema
Pós-condições	-
Fluxo principal	
Ações do Ator	Ações do Sistema
I. Selecionar a opção de alterar ou excluir o perfil.	

	II. Carregar a tela de acordo com a opção desejada.
III. Inserir os dados necessários.	
	IV. Validar os campos preenchidos.
	VI. Realizar a operação desejada.
Fluxo de exceção	
Ações do ator	Ações do sistema
	I. Caso o usuário tenha inserido os dados errados, informar erro e impedir a continuação do processo.

Tabela 4 – Documentação do caso de uso Pesquisar negócios

Nome do caso de uso	Pesquisar negócios
Caso de Uso Geral	-
Ator principal	Usuário
Ator secundário	-
Resumo	Este caso de uso descreve as etapas necessárias para que o usuário pesquise negócios cadastrados
Pré-condições	-
Pós-condições	-
Fluxo principal	
Ações do Ator	Ações do Sistema
I. Selecionar a opção de pesquisar negócios.	
	II. Carregar a tela de pesquisar negócios.
III. Inserir o nome do negócio	

	IV. Carregar a tabela com os registros de negócios com o nome ou que comecem com o nome digitado pelo usuário.
V. Clicar sobre o negócio desejado.	
	VI. Carregar a tela de detalhes do negócio desejado.
Fluxo de exceção	
Ações do ator	Ações do sistema
	I. Caso não tenha nenhum negócio cadastrado ou haja falha com a conexão com o banco de dados, informar erro e impedir a continuação do processo.

Tabela 5 – Documentação do caso de uso Manter negócio

Nome do caso de uso	Manter negócio
Caso de Uso Geral	-
Ator principal	Avicultor
Ator secundário	-
Resumo	Este caso de uso descreve as etapas necessárias para que o usuário cadastre altere ou exclua um negócio
Pré-condições	Estar logado no sistema
Pós-condições	-
Fluxo principal	
Ações do Ator	Ações do Sistema
I. Selecionar a opção de cadastrar, alterar ou excluir um negócio.	
	II. Carregar a tela da opção desejada.
III. Inserir os dados necessários.	



	IV. Validar os dados inseridos.
	V. Realizar o processo desejado
Fluxo de exceção	
Ações do ator	Ações do sistema
	I. Caso o usuário tenha inserido dados inválidos ou haja falha com o banco de dados, informar erro e impedir a continuação do processo.

Tabela 6 – Documentação do caso de uso Manter fornecimento

Nome do caso de uso	Manter fornecimento
Caso de Uso Geral	-
Ator principal	Avicultor
Ator secundário	-
Resumo	Este caso de uso descreve as etapas necessárias para que o usuário cadastre altere ou exclua um fornecimento
Pré-condições	Estar logado no sistema e ter pelo menos um negócio cadastrado.
Pós-condições	-
Fluxo principal	
Ações do Ator	Ações do Sistema
I. Selecionar a opção de cadastrar, alterar ou excluir um fornecimento.	
	II. Carregar a tela da opção desejada.
III. Inserir os dados necessários.	
	IV. Validar os dados inseridos.
	V. Realizar o processo desejado
Fluxo de exceção	
Ações do ator	Ações do sistema

	I. Caso o usuário tenha inserido dados inválidos ou haja falha com o banco de dados, informar erro e impedir a continuação do processo.
--	---

Tabela 7 – Documentação do caso de uso Manter fornecedores

Nome do caso de uso	Manter fornecedores
Caso de Uso Geral	-
Ator principal	Avicultor
Ator secundário	-
Resumo	Este caso de uso descreve as etapas necessárias para que o usuário cadastre altere ou exclua um fornecedor
Pré-condições	Estar logado no sistema e ter pelo menos um negócio e um estabelecimento cadastrado.
Pós-condições	-
Fluxo principal	
Ações do Ator	Ações do Sistema
I. Selecionar a opção de cadastrar, alterar ou excluir um fornecedor.	
	II. Carregar a tela da opção desejada.
III. Inserir os dados necessários.	
	IV. Validar os dados inseridos.
	V. Realizar o processo desejado
Fluxo de exceção	
Ações do ator	Ações do sistema
	I. Caso o usuário tenha inserido dados inválidos ou haja falha com o banco de dados, informar erro e impedir a continuação do processo.

Tabela 8 – Documentação do caso de uso Manter vacinas

Nome do caso de uso	Manter vacinas
Caso de Uso Geral	-
Ator principal	Avicultor
Ator secundário	-
Resumo	Este caso de uso descreve as etapas necessárias para que o usuário cadastre, altere ou exclua uma vacina
Pré-condições	Estar logado no sistema e ter pelo menos um negócio e um estabelecimento cadastrado.
Pós-condições	-
Fluxo principal	
Ações do Ator	Ações do Sistema
I. Selecionar a opção de cadastrar, alterar ou excluir uma vacina.	
	II. Carregar a tela da opção desejada.
III. Inserir os dados necessários.	
	IV. Validar os dados inseridos.
	V. Realizar o processo desejado
Fluxo de exceção	
Ações do ator	Ações do sistema
	I. Caso o usuário tenha inserido dados inválidos ou haja falha com o banco de dados, informar erro e impedir a continuação do processo.

Tabela 9 – Documentação do caso de uso Manter produtos

Nome do caso de uso	Manter produtos
Caso de Uso Geral	-
Ator principal	Avicultor

Ator secundário	-
Resumo	Este caso de uso descreve as etapas necessárias para que o usuário cadastre altere ou exclua um produto
Pré-condições	Estar logado no sistema e ter pelo menos um negócio e um estabelecimento cadastrado.
Pós-condições	-
Fluxo principal	
Ações do Ator	Ações do Sistema
I. Selecionar a opção de cadastrar, alterar ou excluir um produto.	
	II. Carregar a tela da opção desejada.
III. Inserir os dados necessários.	
	IV. Validar os dados inseridos.
	V. Realizar o processo desejado
Fluxo de exceção	
Ações do ator	Ações do sistema
	I. Caso o usuário tenha inserido dados inválidos ou haja falha com o banco de dados, informar erro e impedir a continuação do processo.

Tabela 10 – Documentação do caso de uso Manter estabelecimento

Nome do caso de uso	Manter estabelecimento
Caso de Uso Geral	-
Ator principal	Avicultor
Ator secundário	-
Resumo	Este caso de uso descreve as etapas necessárias para que o usuário cadastre altere ou exclua um estabelecimento

Pré-condições	Estar logado no sistema e ter pelo menos um negócio cadastrado.
Pós-condições	-
Fluxo principal	
Ações do Ator	Ações do Sistema
I. Selecionar a opção de cadastrar, alterar ou excluir um estabelecimento.	
	II. Carregar a tela da opção desejada.
III. Inserir os dados necessários.	
	IV. Validar os dados inseridos.
	V. Realizar o processo desejado
Fluxo de exceção	
Ações do ator	Ações do sistema
	I. Caso o usuário tenha inserido dados inválidos ou haja falha com o banco de dados, informar erro e impedir a continuação do processo.

Tabela 11 – Documentação do caso de uso Gerar relatórios de desempenho

Nome do caso de uso	Manter Gerar relatórios de desempenho
Caso de Uso Geral	-
Ator principal	Avicultor
Ator secundário	-
Resumo	Este caso de uso descreve as etapas necessárias para que o avicultor visualize seus desempenho graficamente
Pré-condições	Estar logado no sistema e ter pelo menos um negócio e um estabelecimento cadastrado.
Pós-condições	-
Fluxo principal	

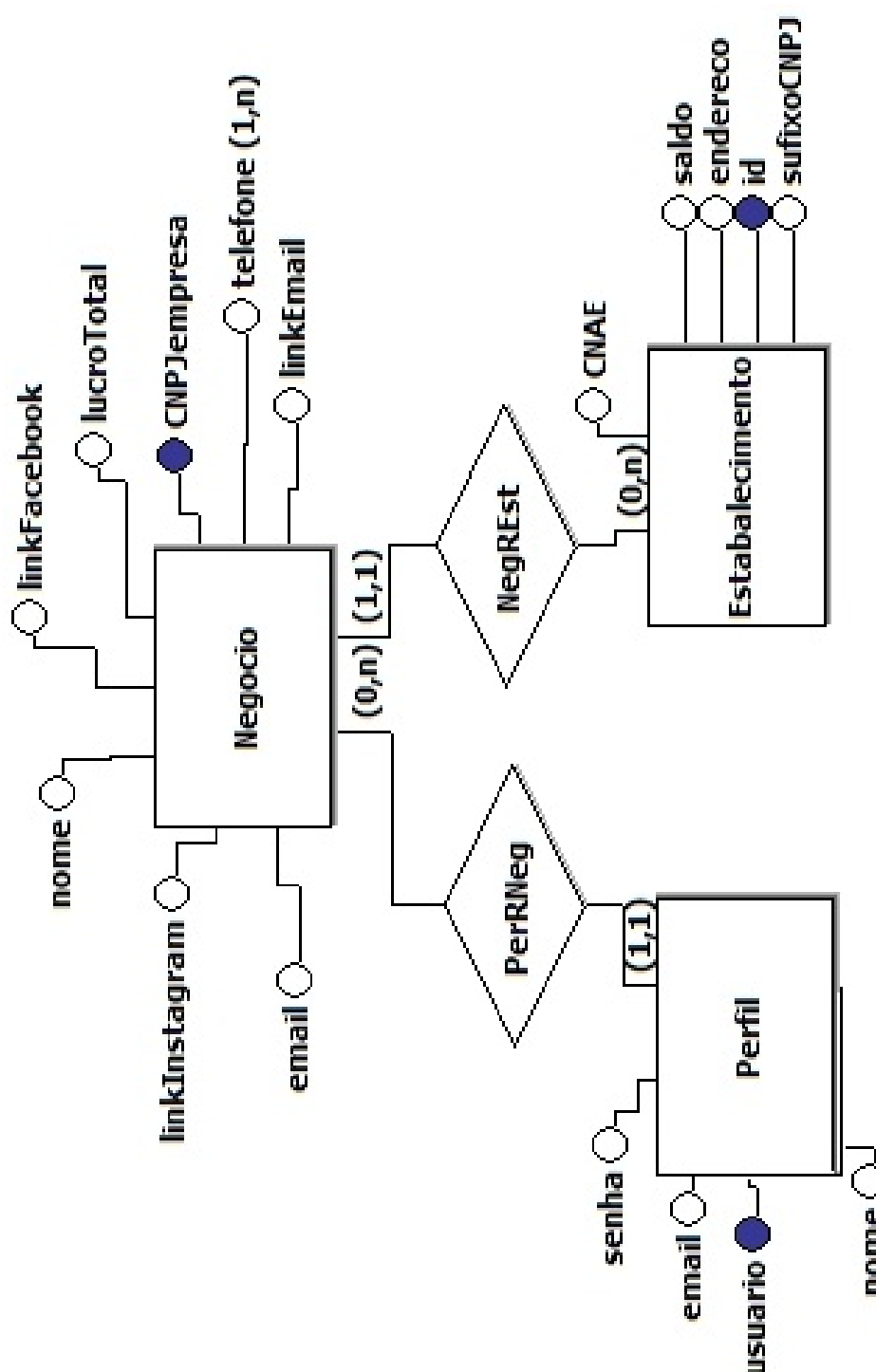
Ações do Ator	Ações do Sistema
I. Selecionar a opção de visualizar gráficos	
	II. Carregar a tela com os gráficos mensais e anuais.
Fluxo de exceção	
Ações do ator	Ações do sistema
	I. Caso haja falha com o banco de dados, informar erro e impedir a continuação do processo.

### 9.3 Modelo Entidade-relacionamento

O Modelo Entidade-Relacionamento (MER) é uma visão conceitual do projeto de banco de dados. Nele é demonstrados as entidades, os relacionamentos e restrições que ocorreram no sistema trabalhado.

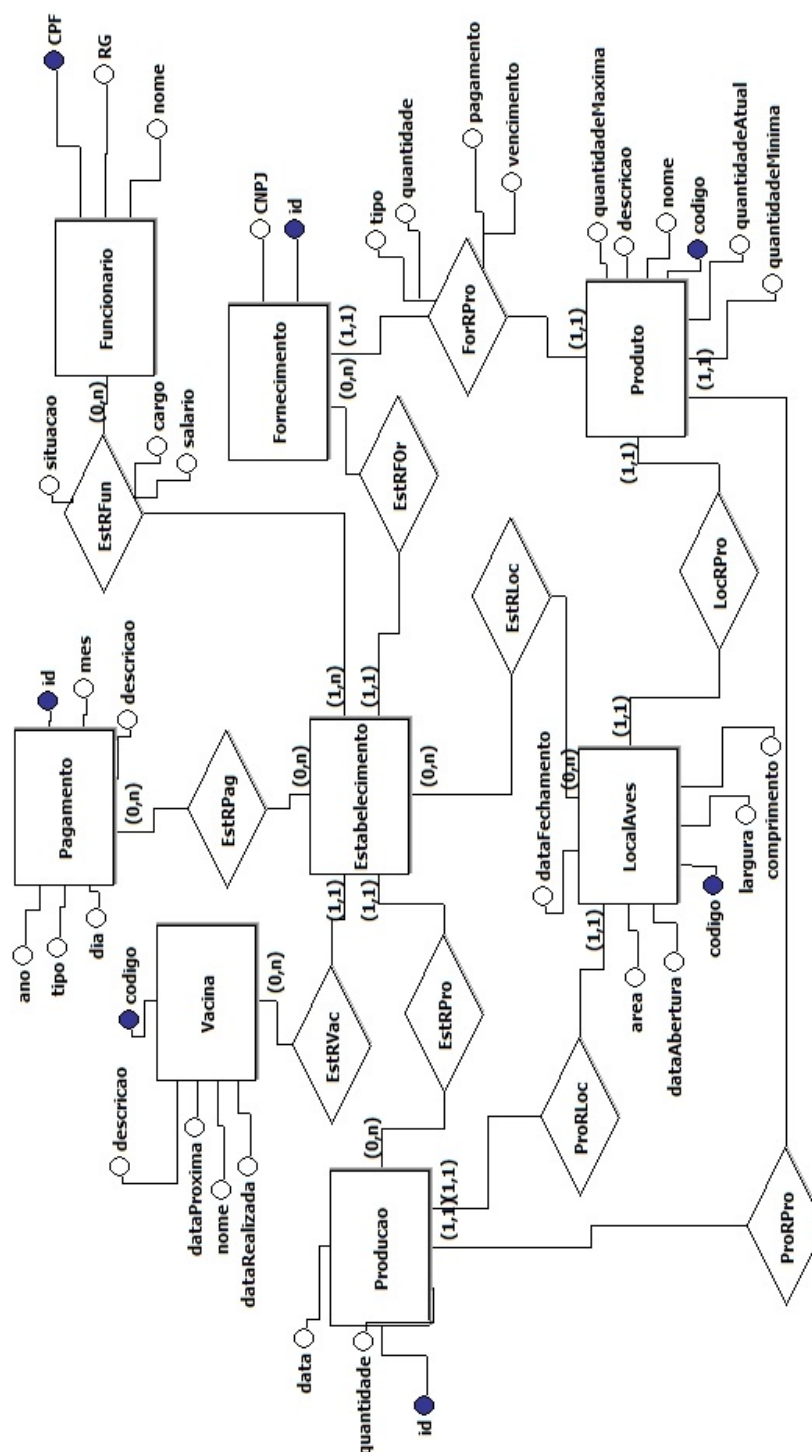
No primeiro MER está definido os relacionamentos entre avicultor, negócio e estabelecimentos. Podemos perceber que um avicultor pode ter vários negócios ou estabelecimentos. Isso ocorre porque um negócio é um conjunto de estabelecimentos, assim é possível de um sócio cuidar do negócio ao todo, ou apenas de um estabelecimento. A partir do momento que os negócios e estabelecimentos são criados, eles podem ser compartilhados, ou seja, um negócio ou um estabelecimento podem ser controlados por mais de um avicultor.

Figura 4 – Modelo Entidade-Relacionamento 1 (fonte própria, feito no Astah).



No segundo diagrama do MER é mostrado como ocorre a interação entre o estabelecimento e as entidades ligadas a ele. Isso ocorre porque o negócio é um conjunto de estabelecimentos, porém produtos, vacinas, funcionários e as outras entidades são exclusivamente de um estabelecimento.

Figura 5 – Modelo Entidade-Relacionamento 2 (fonte própria, feito no Astah).

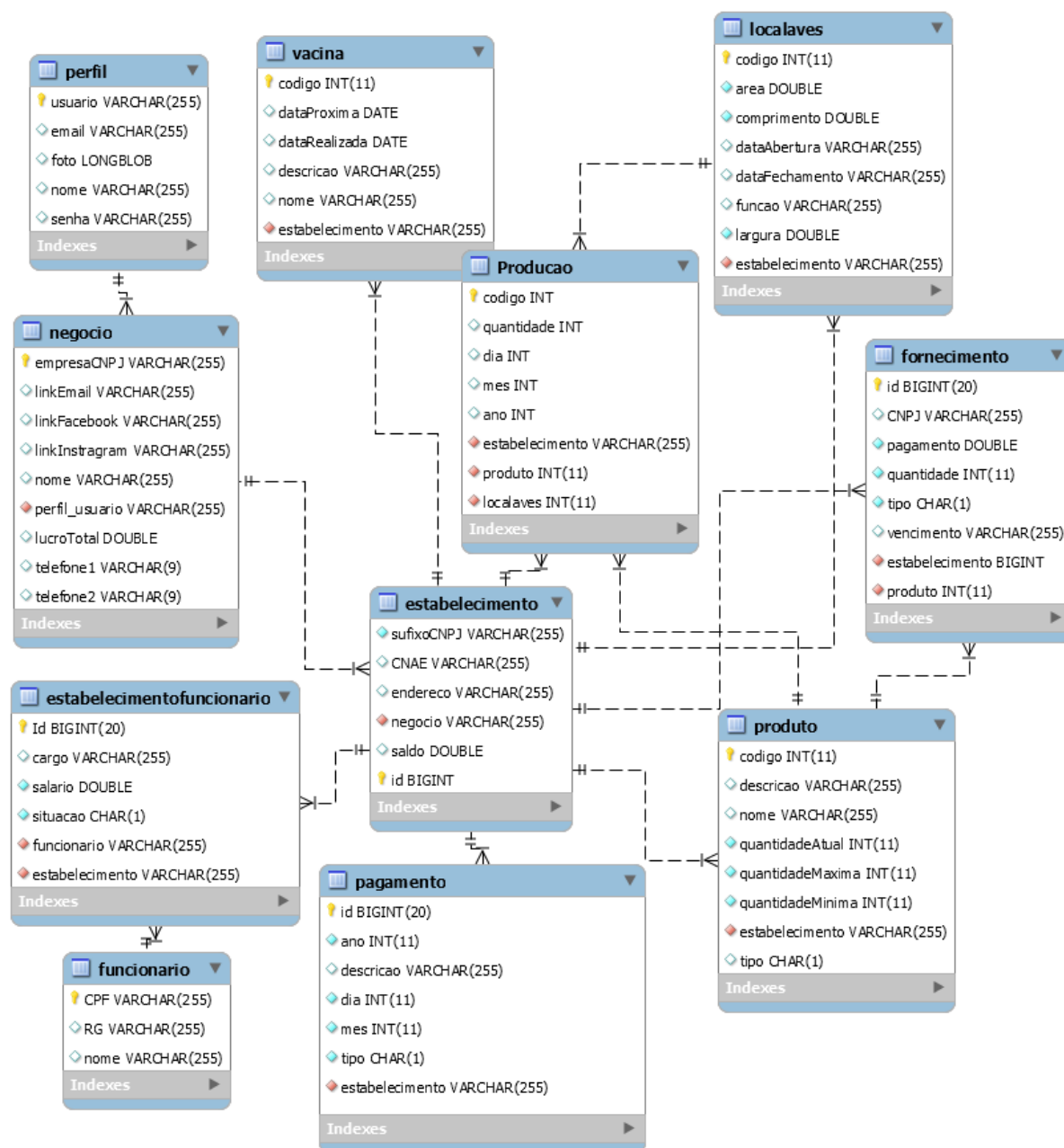


## 9.4 Modelo relacional de tabelas

O MR de tabelas representa o mapeamento do MER para um modelo lógico do projeto de um banco de dados. A imagem abaixo demonstra o MR de tabelas do projeto.



Figura 6 – Modelo Relacional de tabelas (fonte própria, feito no MySQL workbench).



## 9.5 Dicionário de dados

O dicionário de dados é uma lista de tabelas que define como os dados estão estruturados no SGBD. As tabelas abaixo, demonstram do DD do Chicken Tracker.

Tabela 12 – Tabela perfil

Nome	Tipo	Tamanho	Nulo	Descrição
usuario(PK)	VARCHAR	255	Não	Identifica o login do usuário
email	VARCHAR	255	Não	O e-mail do usuário, ele recebe avisos por esse e-mail
nome	VARCHAR	255	Não	Identifica o nome do usuário
foto	VARCHAR	255	Sim	Caminho da foto do usuário
senha	VARCHAR	255	Não	Senha para o usuário entrar no sistema

Tabela 13 – Tabela negocio

Nome	Tipo	Tamanho	Nulo	Descrição
empresaCNPJ	VARCHAR	255	Não	Representa a primeira parte do CNPJ da empresa
email	VARCHAR	255	Sim	E-mail para contato do negócio
linkFacebook	VARCHAR	255	Sim	Define o facebook para contato do negócio
userInstagram	VARCHAR	255	Sim	Define o instagram para contato do negócio
telefone1	VARCHAR	9	Não	Telefone 1 do negócio
telefone2	VARCHAR	9	Sim	Telefone 2 do negócio
nome	VARCHAR	255	Não	Representa o nome do negócio
perfil_id(FK)	VARCHAR	255	Não	Representa o usuário avicultor dono do negócio

Tabela 14 – Tabela estabelecimento

Nome	Tipo	Tamanho	Nulo	Descrição
id (PK)	BIGINT	20	Não	Identifica o estabelecimento
sufixoCNPJ	VARCHAR	255	Não	Representa a outra parte do CNPJ para identificar o estabelecimento
CNAE	VARCHAR	255	Não	Define o ramo de trabalho do estabelecimento
endereco	VARCHAR	255	Não	Define o endereço completo do estabelecimento
negocio(FK)	VARCHAR	255	Não	Representa a qual negócio o estabelecimento pertence

Tabela 15 – Tabela estabelecimento\_funcionario

Nome	Tipo	Tamanho	Nulo	Descrição
id(PK)	BIGINT	20	Não	É o identificador da tabela
cargo	VARCHAR	255	Não	O cargo do funcionário em tal estabelecimento
salário	DOUBLE	64	Não	O salário do funcionário em tal estabelecimento
situacao	CHAR	1	Não	A situação do funcionário em tal estabelecimento
funcionario(FK)	VARCHAR	255	Não	Define o funcionário do relacionamento
estbmnto(FK)	VARCHAR	255	Não	Define o estabelecimento do relacionamento

Tabela 16 – Tabela funcionario

Nome	Tipo	Tamanho	Nulo	Descrição
CPF (PK)	VARCHAR	255	Não	Representa o CPF do funcionário
RG	VARCHAR	255	Não	Representa o RG do funcionário
Nome	VARCHAR	255	Não	Representa o nome do funcionário

Tabela 17 – Tabela fornecimento

Nome	Tipo	Tamanho	Nulo	Descrição
id(PK)	BIGINT	20	Não	Identificador da tabela
CNPJ	VARCHAR	255	Não	CNPJ da empresa parceira
pagamento	DOUBLE	64	Não	O valor da mensalidade
quantidade	INT	11	Não	Quantidade do produto a ser fornecido
vencimento	VARCHAR	255	Não	Data final do contrato
tipo	CHAR	1	Não	Representa se é fornecimento ou fornecedor
estbmnto(FK)	VARCHAR	255	Não	Representa o estabelecimento participante do contrato
produto (FK)	INT	11	Não	Identifica o produto a ser fornecido

Tabela 18 – Tabela pagamento

Nome	Tipo	Tamanho	Nulo	Descrição
id (PK)	BIGINT	20	Não	Identificador da tabela
valor	DOUBLE	64	Não	Identifica o valor do pagamento
ano	INT	11	Não	Identifica o ano do pagamento
mês	BIGINT	11	Não	Identifica o mês do pagamento
dia	INT	11	Não	Identifica o dia do pagamento
tipo	CHAR	1	Não	Identifica se é gasto ou despesa
descricao	VARCHAR	255	Não	Descreve o motivo do pagamento
estbmnto (FK)	VARCHAR	255	Não	Representa o estabelecimento

Tabela 19 – Tabela produto

Nome	Tipo	Tamanho	Nulo	Descrição
codigo (PK)	INT	11	Não	Código do produto
nome	VARCHAR	255	Não	Nome do produto
descricao	VARCHAR	255	Não	Descrição do produto
tipo	CHAR	1	Não	Define se o produto é para produção ou estoque
quantidadeAtual	INT	11	Não	Define a quantidade atual no estoque
quantidadeMin	INT	11	Não	Define a quantidade mínima que deve ter no estoque
quantidadeMax	INT	11	Não	Define a quantidade máxima que o estoque suporta
estbmnto (FK)	VARCHAR	255	Não	O estabelecimento dono

Tabela 20 – Tabela localaves

Nome	Tipo	Tamanho	Nulo	Descrição
codigo (PK)	INT	11	Não	Código do aviário
area	DOUBLE	64	Sim	Representa a área ocupada pelo aviário
comprimento	DOUBLE	64	Sim	Representa o comprimento do aviário
largura	DOUBLE	64	Sim	Representa a largura do aviário
dataAbertura	VARCHAR	255	Não	Data que o aviário foi criado
dataFecha	VARCHAR	255	Não	Data que o aviário foi fechado
quantidade	INT	11	Não	Representa a quantidade do produto que será produzido
produto (FK)	INT	11	Não	Representa o produto que será produzido
estbmnto (FK)	VARCHAR	255	Não	Representa o estabelecimento

Tabela 21 – Tabela Producao

Nome	Tipo	Tamanho	Nulo	Descrição
id (PK)	BIGINT	20	Não	Identifica a produção
quantidade	INT	11	Não	Representa a quantidade do produto produzido
dia	INT	11	Não	Dia da produção
mes	INT	11	Não	Mês da produção
ano	INT	11	Não	Ano da produção
produto (FK)	INT	11	Não	Representa o produto que será produzido
localaves (FK)	INT	11	Não	Representa o aviário
estbmnto (FK)	VARCHAR	255	Não	Representa o estabelecimento

Tabela 22 – Tabela vacina

Nome	Tipo	Tamanho	Nulo	Descrição
codigo	INT	11	Não	Codigo da vacina
dataProxima	date	24	Não	Data da próxima vacinação
dataRealizada	date	24	Não	Data da última vacinação
descricao	VARCHAR	255	Sim	Descrição da vacina
nome	VARCHAR	255	Não	Nome da vacina
estbmnto (FK)	VARCHAR	255	Não	Representa o estabelecimento

## 9.6 Diagrama de classes

A partir do MER é possível criar uma visão mais lógica do sistema, ou seja, como iremos controlar, em termos de programação, os dados, em outras palavras, como vamos representar as entidades e os relacionamentos do MER em forma orientada a objetos.

Este diagrama de classe além de mostrar o mapeamento do MER para a linguagem Java, mostra também a utilização do padrão de projeto Singleton, onde o atributo "getInstance" armazena a instância única de um objeto.

É importante dizer que apenas as classes de modelo estão apresentadas, ainda existem outras classes, como os servlets e os controllers, essas duas classes têm a estrutura para ter contado com o banco de dados.

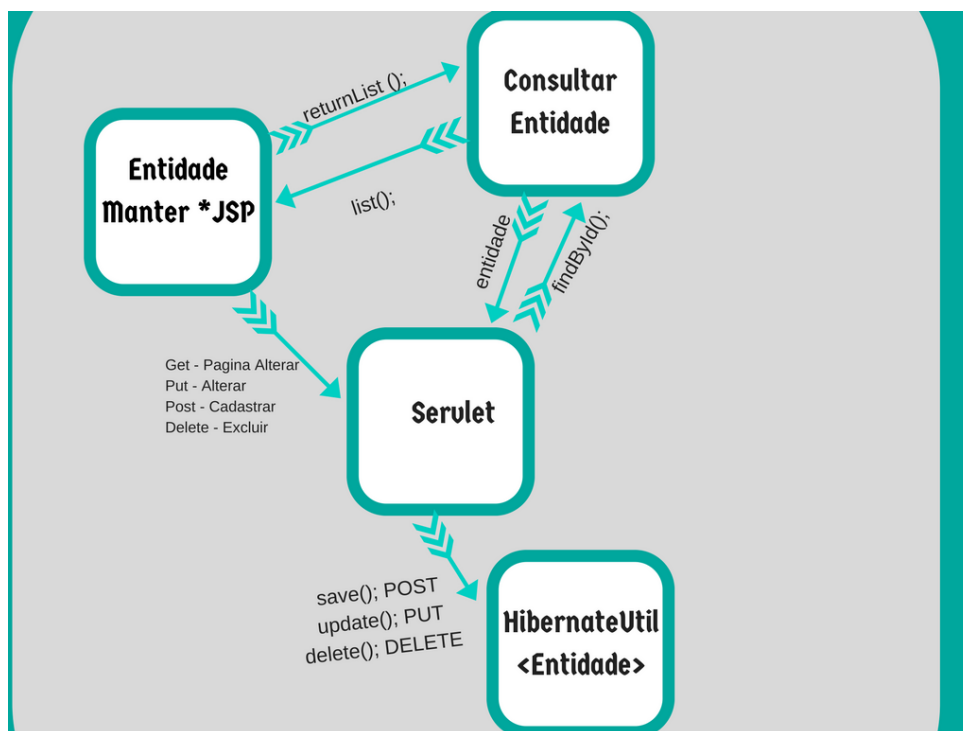
No próximo tópico será demonstrado a mapeação das páginas web em JSP, porém é interessante explicar como funciona a ligação entre as classes e páginas JSP.

Existem 4 tipos de requisições do usuário: GET(redirecionamento de página), POST(cadastro), PUT(alteração) e DELETE(exclusão). Cada requisição dessa é tratada em um servlet específico, de acordo com a entidade trabalhada no momento.

Existe, ainda, uma classe controle, essa trata de qualquer outro tipo de conexão com o banco de dados, ou seja, consultas específicas com o banco de dados, como lista uma entidade ou procurar um usuário específico.

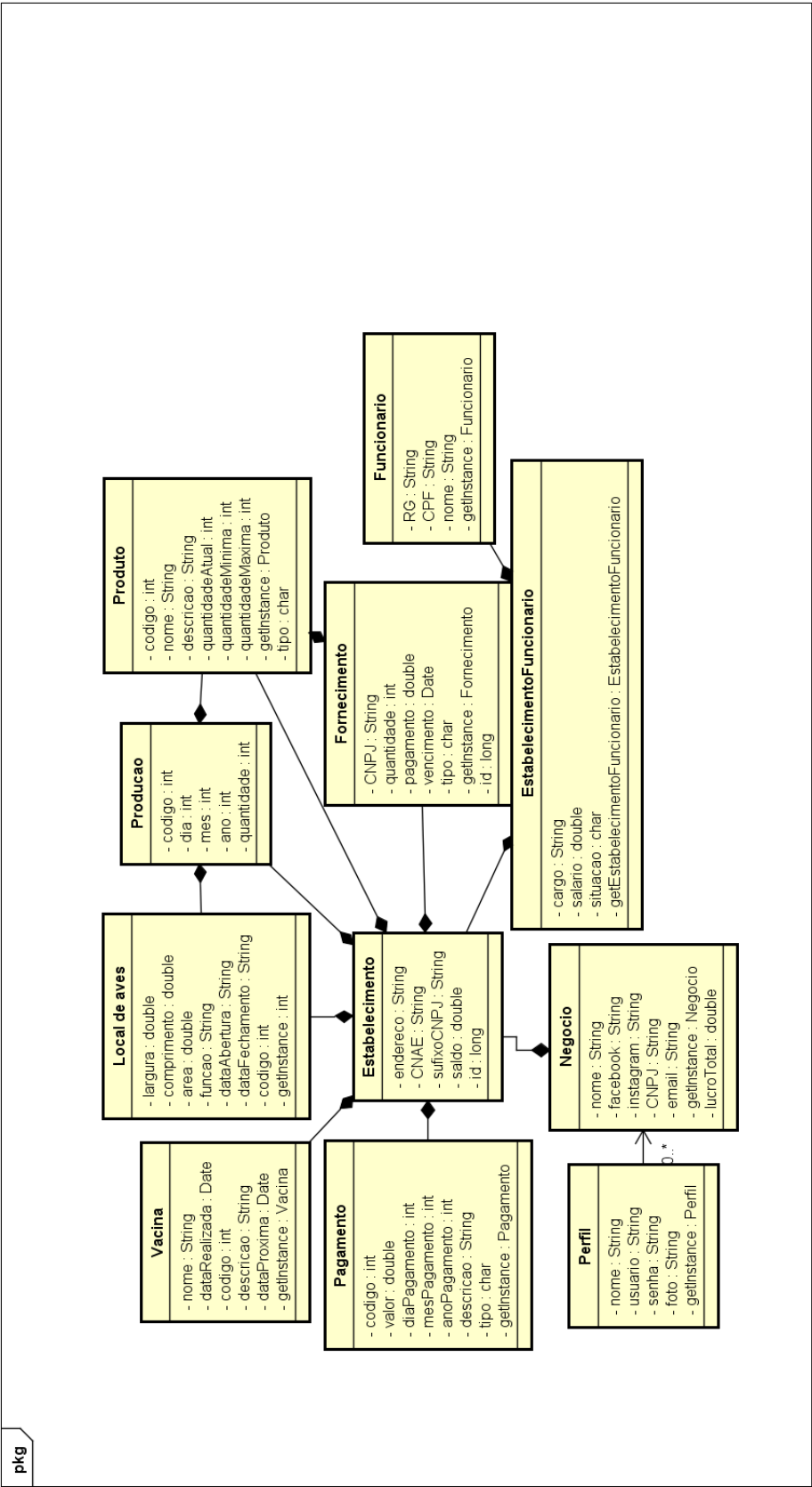
A imagem abaixo mostra melhor a relação entre as páginas de requisições e os servlet, seguindo o padrão MVC:

Figura 7 – Modelo MVC do projeto (fonte própria, feito no Canvas).



Na imagem abaixo agora, demonstra as classes modelos do projeto:

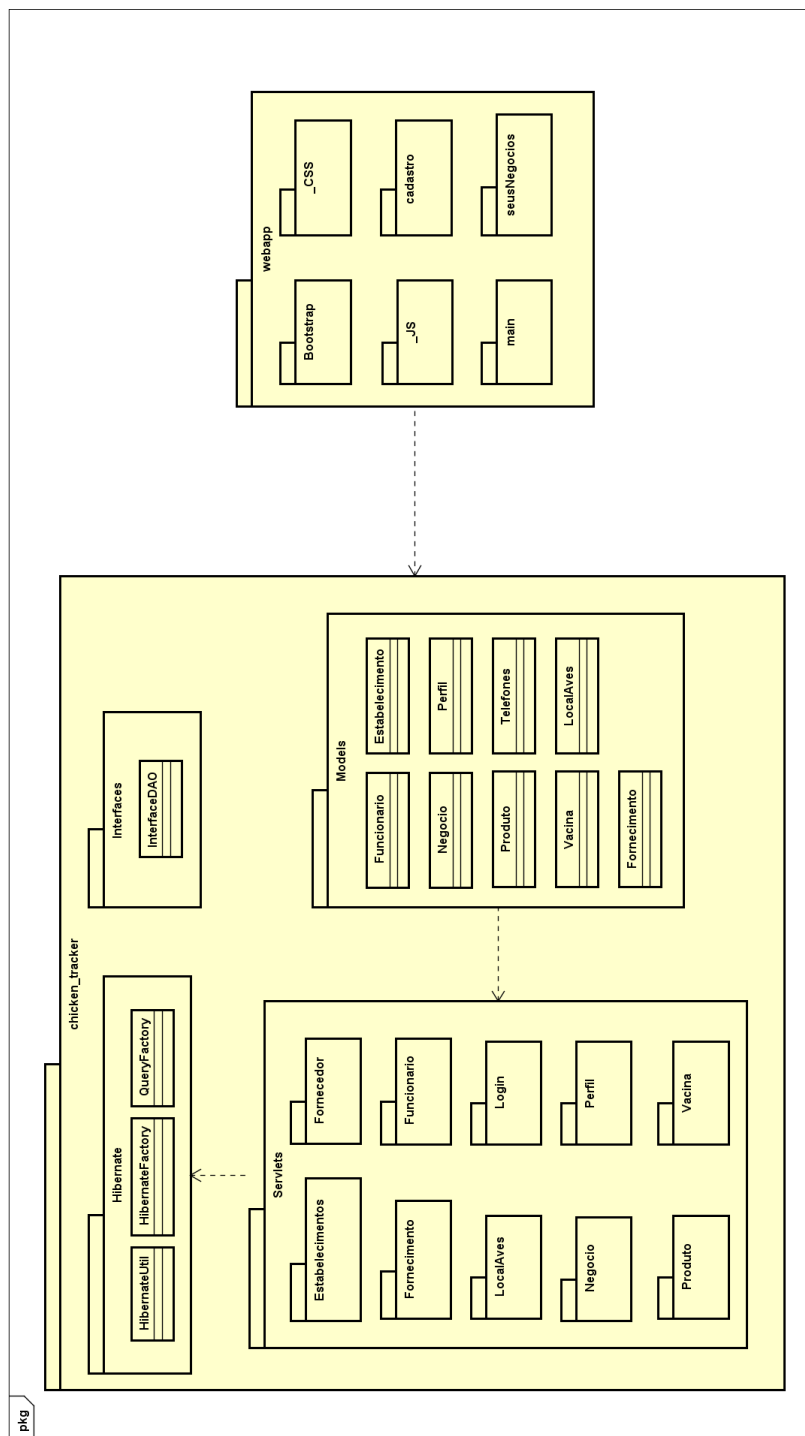
Figura 8 – Diagrama de classes (fonte própria, feito no Astah).



## 9.7 Diagrama de pacotes

O diagrama de pacotes define como os módulos estão divididos e organizados. Cada pacote é uma pasta no diretório contendo uma parte funcional do sistema. Nesse diagrama, é possível verificar os acoplamentos entre as partes do software.

Figura 9 – Diagrama de pacotes (fonte própria, feito no Astah).





## 9.8 Diagramas web

Outra forma de representarmos a Visão lógica, neste caso da interface, é pelo diagrama web. Este diagrama por meio de esterótipos que são extensões do diagrama de classes representa cada tipo de interface. É importante enfatizar que cada classe é uma interface no sistema.

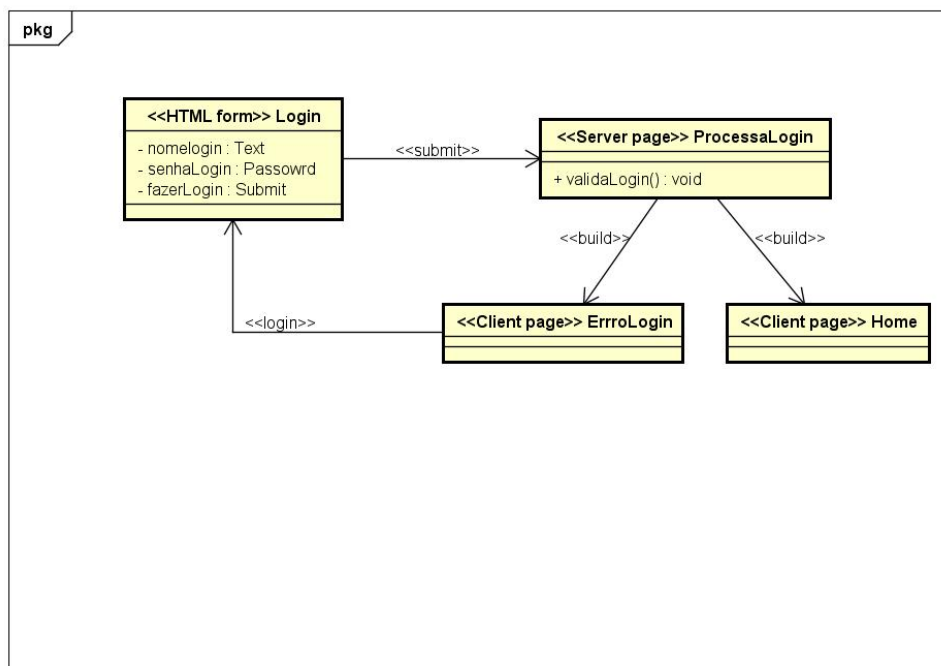
Existem três estereótipos de classes: Client Page, voltado para o que o usuário tem acesso quando requisitado; Server Page, controla as funções das interfaces; Form, possuem atributos de entrada.

Cada Client Page foi baseado em classes do diagrama de classe. O Server page representa o controle de cada uma das operações requisitadas pelo usuário.

As imagens abaixo estão representando a descrição.

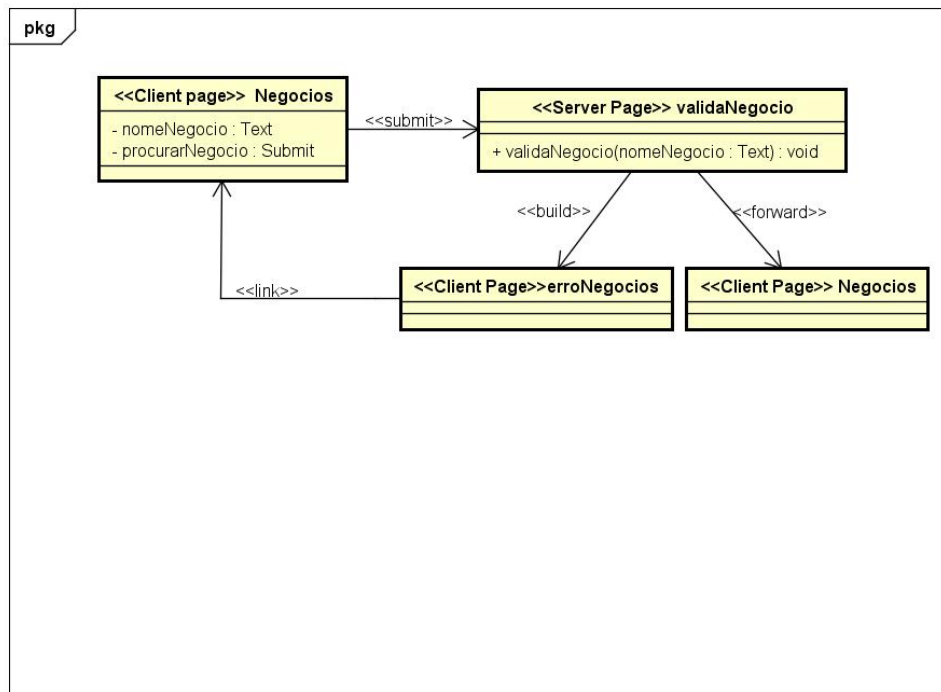
### 9.8.1 Diagrama Web - 1

Figura 10 – Diagrama web - 1 (fonte própria, feito no Astah).



### 9.8.2 Diagrama web - 2

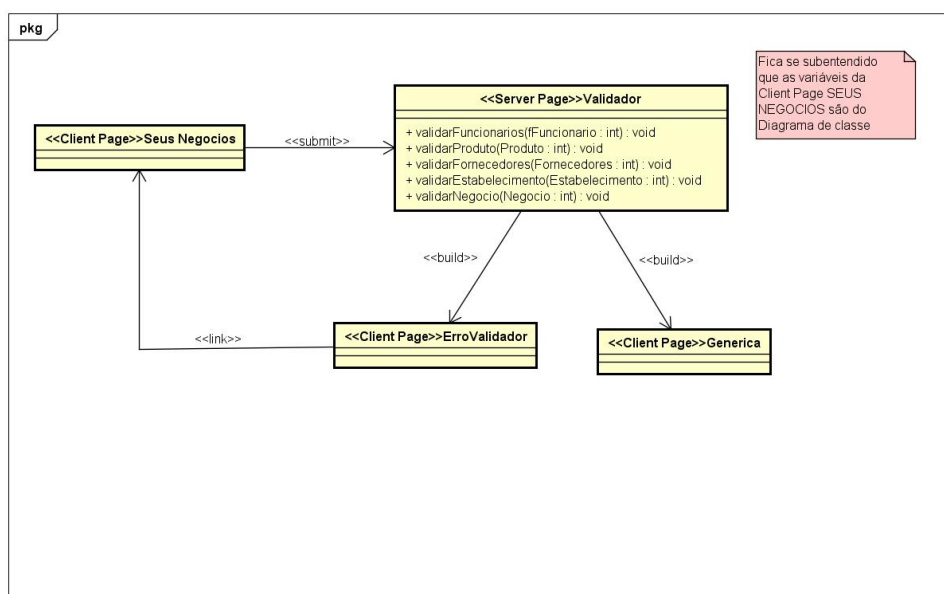
Figura 11 – Diagrama web - 2 (fonte própria, feito no Astah).



powered by Astah

### 9.8.3 Diagrama web - 3

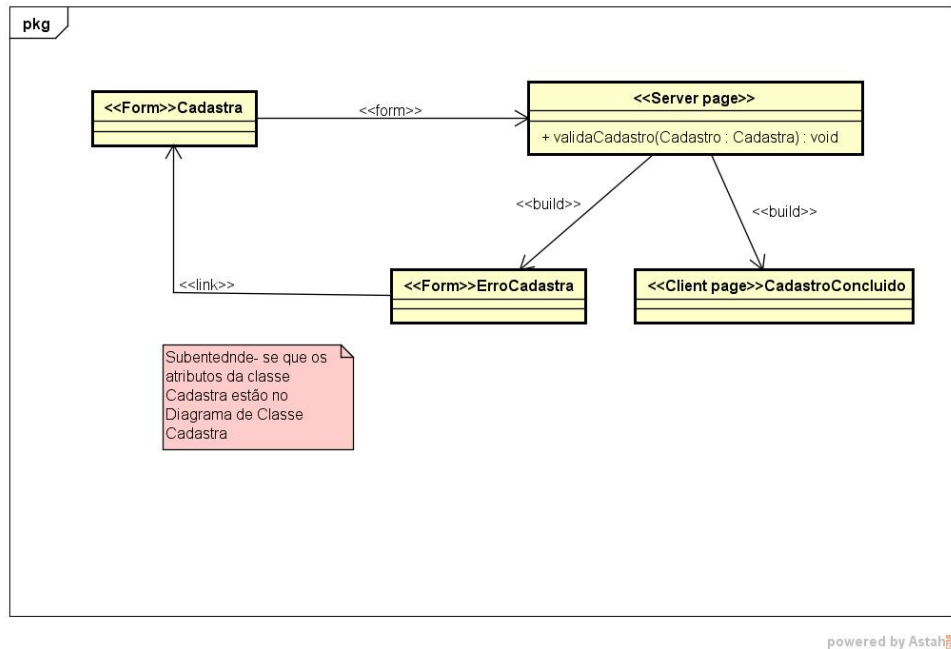
Figura 12 – Diagrama web - 3 (fonte própria, feito no Astah).



powered by Astah

### 9.8.4 Diagrama web - 4

Figura 13 – Diagrama web - 4 (fonte própria, feito no Astah).



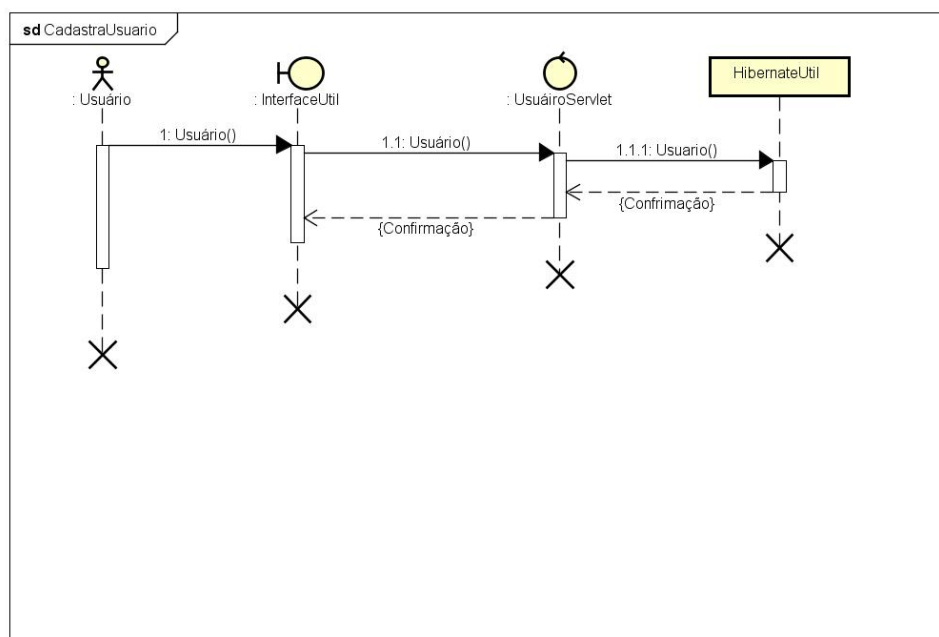
## 9.9 Diagrama de Sequência

O Diagrama de Sequência é um diagrama comportamental que procura determinar a sequência de eventos que ocorrem em um determinado processo, identificando quais mensagens devem ser disparadas entre os elementos envolvidos e em que ordem. Este diagrama determina a ordem em que os eventos ocorrem, as mensagens que são enviadas, os métodos que são chamados e como os objetos interagem dentro de um determinado processo é o objetivo principal desse diagrama.

### 9.9.1 Diagrama de sequência Cadastrar Usuário

O diagrama de sequência cadastrar usuário mostra interação do Usuário, que já foi cadastrado no sistema, com a interfaceUtil. Para efetuar o cadastro é necessário informar os dados do usuário a ser cadastrado, a interface passa a controlador e por fim a instância da classe Hibernate que conectará com o banco de dados. Para melhor compreensão olhe a figura abaixo:

Figura 14 – Diagrama de sequencia Cadastrar Usuário (fonte própria, feito no Astah).

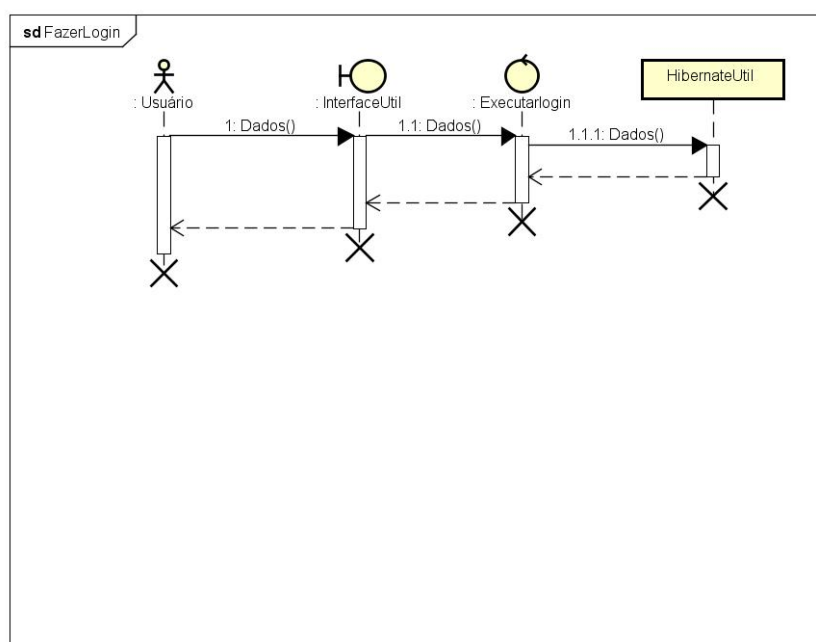


powered by Astah

### 9.9.2 Diagrama de sequência Fazer Login

No diagrama de sequência fazer login, o Usuário informa os seus referentes dados para efetuar o seu login e ter acesso a suas informações no Site, olha a figura abaixo:

Figura 15 – Diagrama de sequencia Fazer Login (fonte própria, feito no Astah).

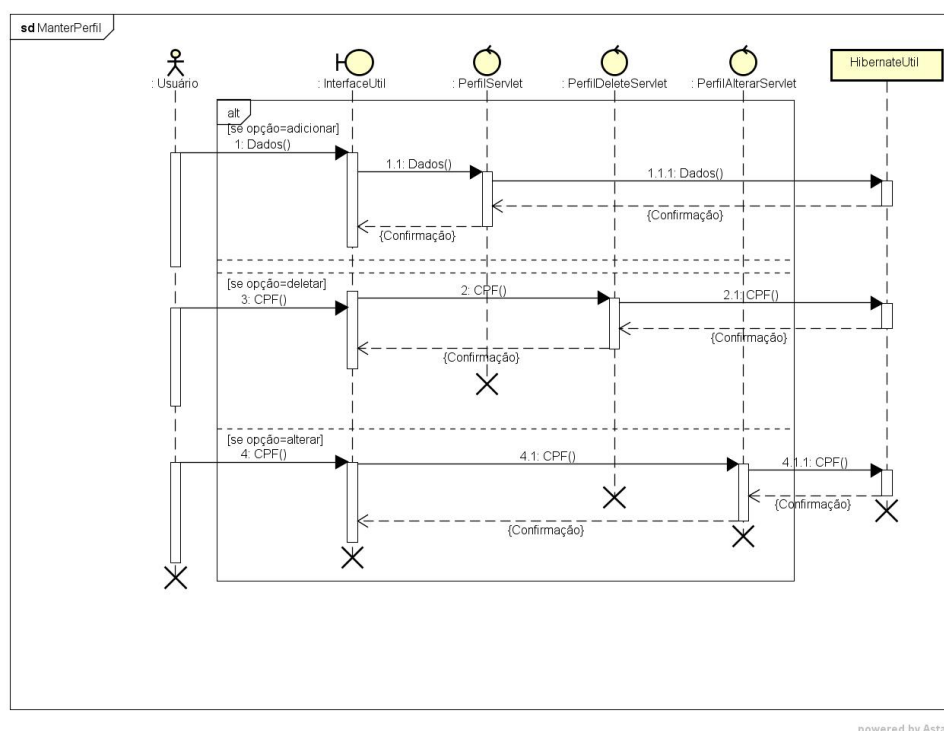


powered by Astah

### 9.9.3 Diagrama de sequência Manter Perfil

O diagrama de sequência manter perfil, representa o que o usuário que já efetuou o seu login no site tem acesso. Ele pode adicionar, alterar, excluir. Para adicionar o Usuário a interfaceUtil e ela vai fornecer seus dados a o controlador PerfilServlet, para alterar o seus dados a interfaceUtil vai informar seus CPF a o controlador PerfilAlterarServlet, e para excluir o controlador PerfilExcluirServlet, olhe figura abaixo:

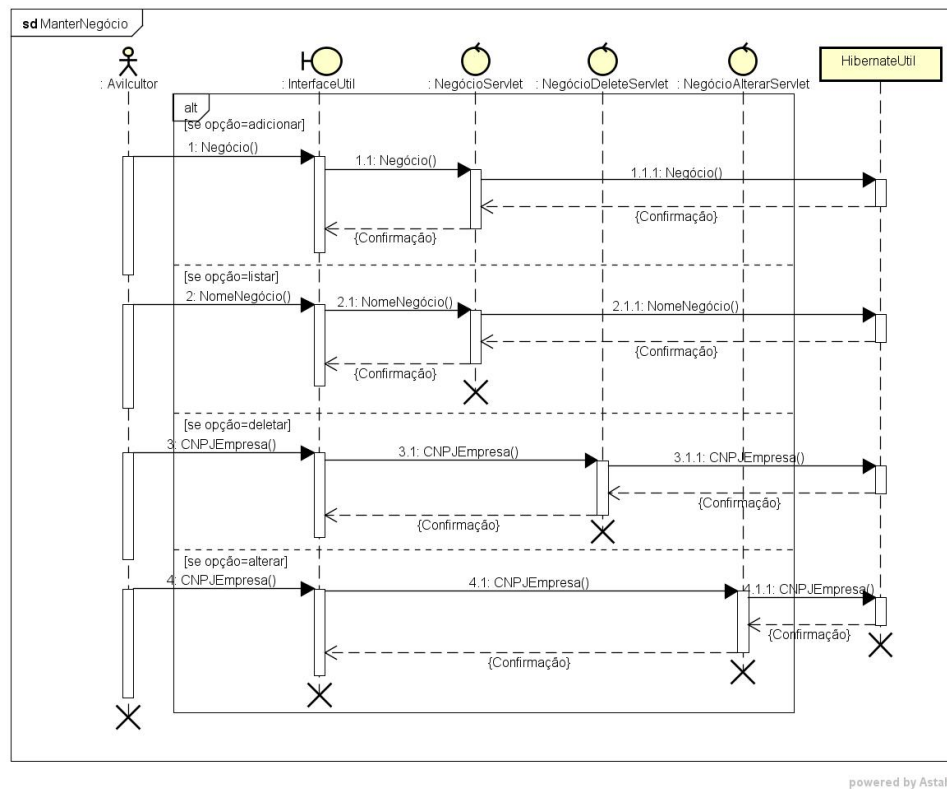
Figura 16 – Diagrama de sequencia Manter Perfil (fonte própria, feito no Astah).



### 9.9.4 Diagrama de sequência Manter Negócio

No diagrama abaixo, o ator Avicultor poderá adicionar, listar, deletar e alterar o Negócio. Para adicionar o ator fornece a interceUtil os dados do Negócio e ela fornecerá ao controlador NegócioServlet os dados e por fim Conectará a Classe que faz conexão com o bando de dados. Para listar o ator fornecerá a interfaceUtil o nome do negocio ela enviará a nome do negocio ao controlador NegócioServlet e por fim ao banco de dados. Para alterar e excluir o usuário vai inserir os dados a interface conectar aos controlador NegócioAlterarServlet e NegócioExcluirServlet respectivamente.

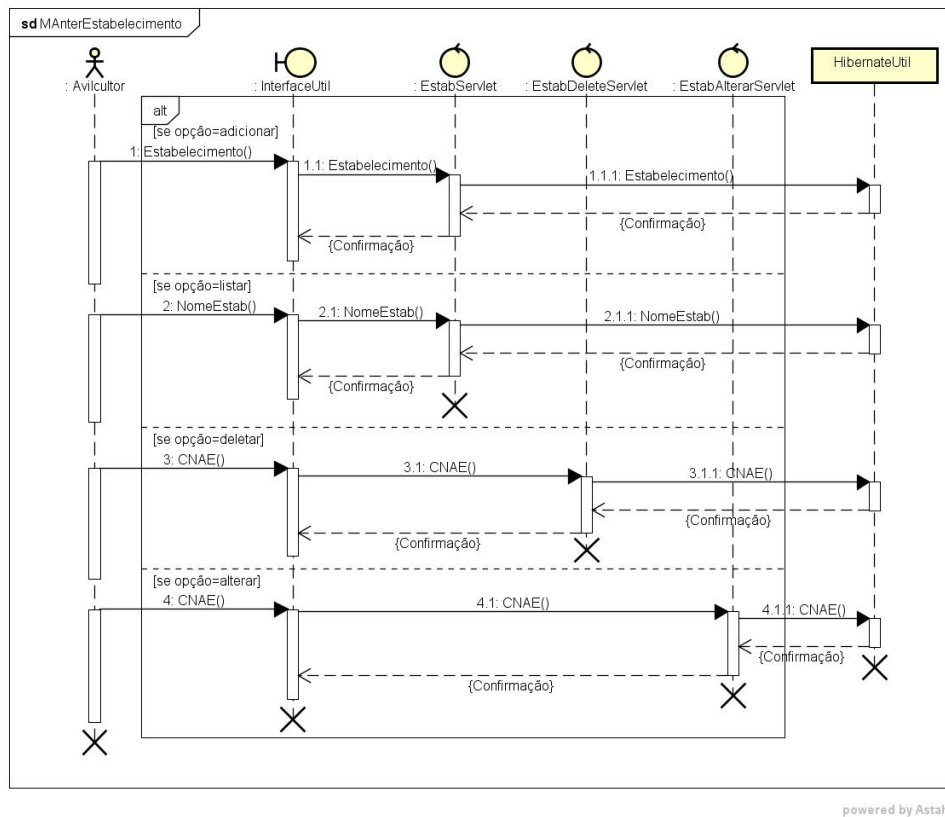
Figura 17 – Diagrama de sequencia Manter Negocio (fonte própria, feito no Astah).



### 9.9.5 Diagrama de sequência Manter Estabelecimentos

No diagrama abaixo, o ator Avicultor poderá adicionar, listar, deletar e alterar o Estabelecimento. Para adicionar o ator fornece a interfaceUtil os dados do estabelecimento e ela fornecerá ao controlador EstabelecimentoServlet os dados e por fim Conectará a Classe Hibernate que faz conexão com o bando de dados. Para listar o ator fornecerá a interfaceUtil o nome do fornecimento ela enviará ao controlador EstabelecimentoServlet e por fim ao banco de dados. Para alterar e excluir o usuário vai inserir os dados a interface conectará aos controlador EstabelecimentoAlterarServlet e EstabelecimentoExcluirServlet respectivamente.

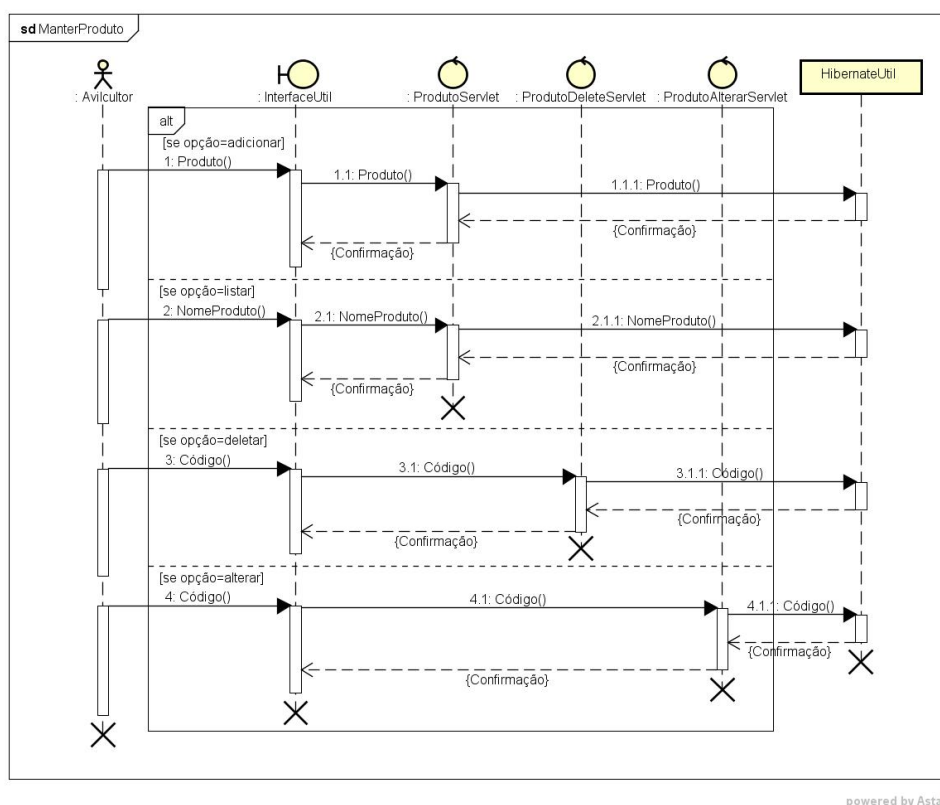
Figura 18 – Diagrama de sequencia Manter Estabelecimento (fonte própria, feito no Astah).



### 9.9.6 Diagrama de sequência Manter Produto

No diagrama abaixo, o ator Avicultor poderá adicionar, listar, deletar e alterar o Produto. Para adicionar o ator fornece a interfaceUtil os dados do Produto e ela fornecerá ao controlador ProdutoServlet os dados e por fim Conectará a Classe Hibernate que faz conexão com o bando de dados. Para listar o ator fornecerá a interfaceUtil o nome do fornecimento ela enviará ao controlador ProdutoServlet e por fim ao banco de dados. Para alterar e excluir o usuário vai inserir os dados a interface conectará aos controlador ProdutoAlterarServlet e ProdutoExcluirServlet respectivamente.

Figura 19 – Diagrama de sequencia Manter Produto (fonte própria, feito no Astah).

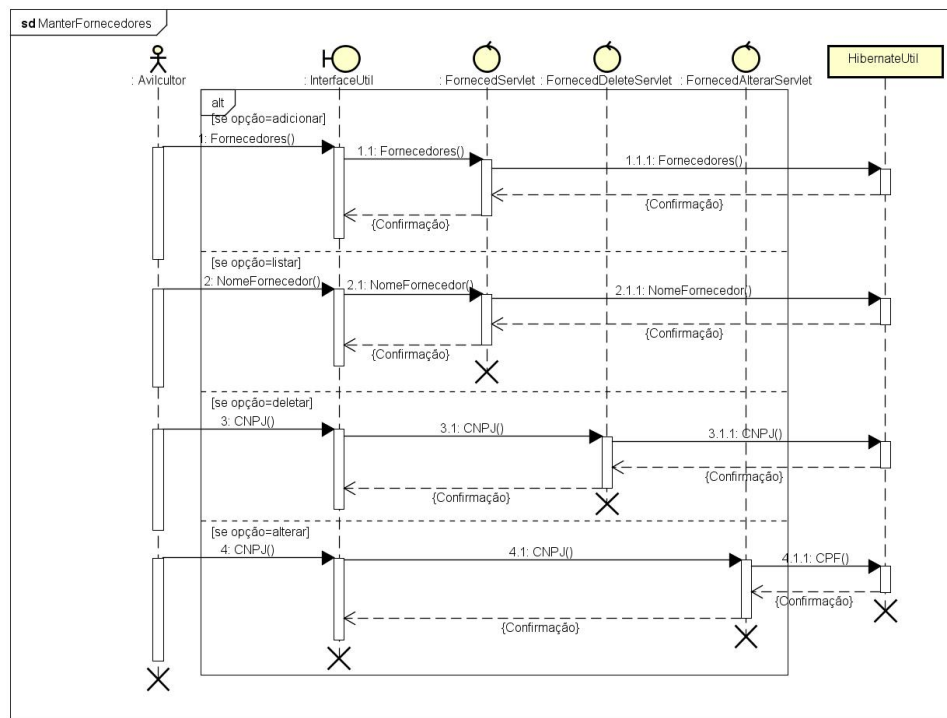


### 9.9.7 Diagrama de sequência Manter Fornecedores

No diagrama abaixo, o ator Avicultor poderá adicionar, listar, deletar e alterar o Fornecedores. Para adicionar o ator fornece a interfaceUtil os dados do Fornecedores e ela fornecerá ao controlador FornecedoresServlet os dados e por fim Conectará a Classe Hibernate que faz conexão com o bando de dados. Para listar o ator fornecerá a interfaceUtil o nome do Fornecedores ela enviará ao controlador FornecedoresServlet e por fim ao banco de dados. Para alterar e excluir o usuário vai inserir os dados a interface conectará aos controlador FornecedoresAlterarServlet e FornecedoresExcluirServlet respectivamente.



Figura 20 – Diagrama de sequencia Manter Fornecedores (fonte própria, feito no Astah).

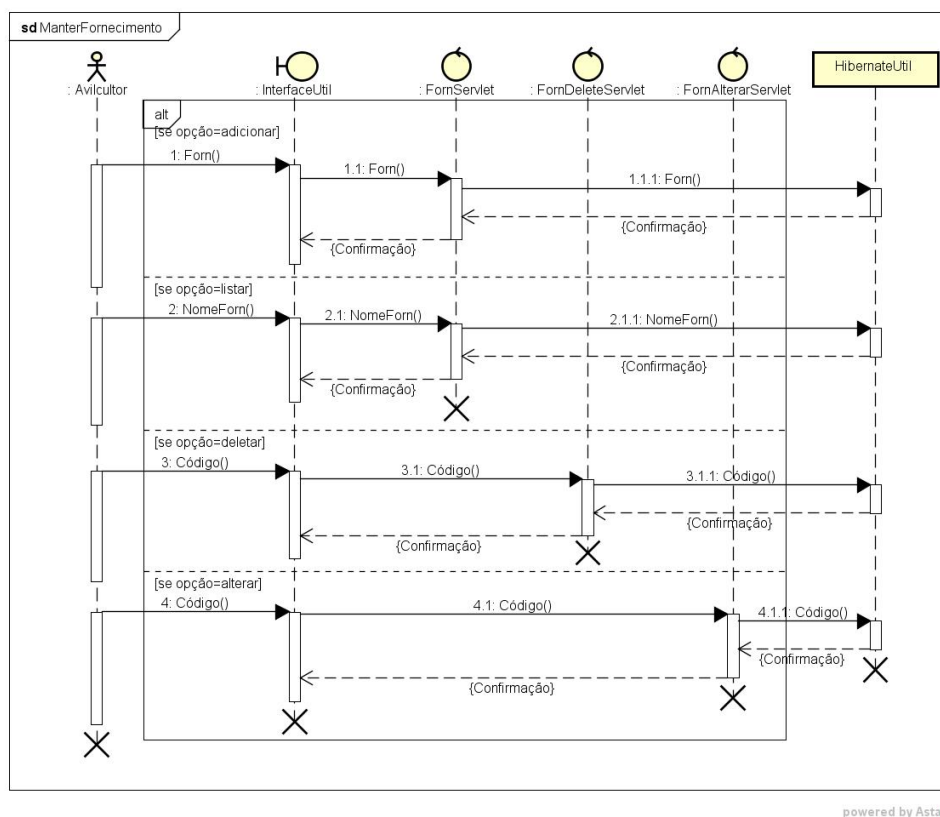


powered by Astah

### 9.9.8 Diagrama de sequência Manter Fornecimento

No diagrama abaixo, o ator Avicultor poderá adicionar, listar, deletar e alterar o Fornecimento. Para adicionar o ator fornece a interfaceUtil os dados do Fornecimento e ela fornecerá ao controlador FornecimentoServlet os dados e por fim Conectará a Classe Hibernate que faz conexão com o bando de dados. Para listar o ator fornecerá a interfaceUtil o nome do Fornecimento ela enviará ao controlador FornecimentoServlet e por fim ao banco de dados. Para alterar e excluir o usuário vai inserir os dados a interface conectará aos controlador FornecimentoAlterarServlet e FornecimentoExcluirServlet respectivamente.

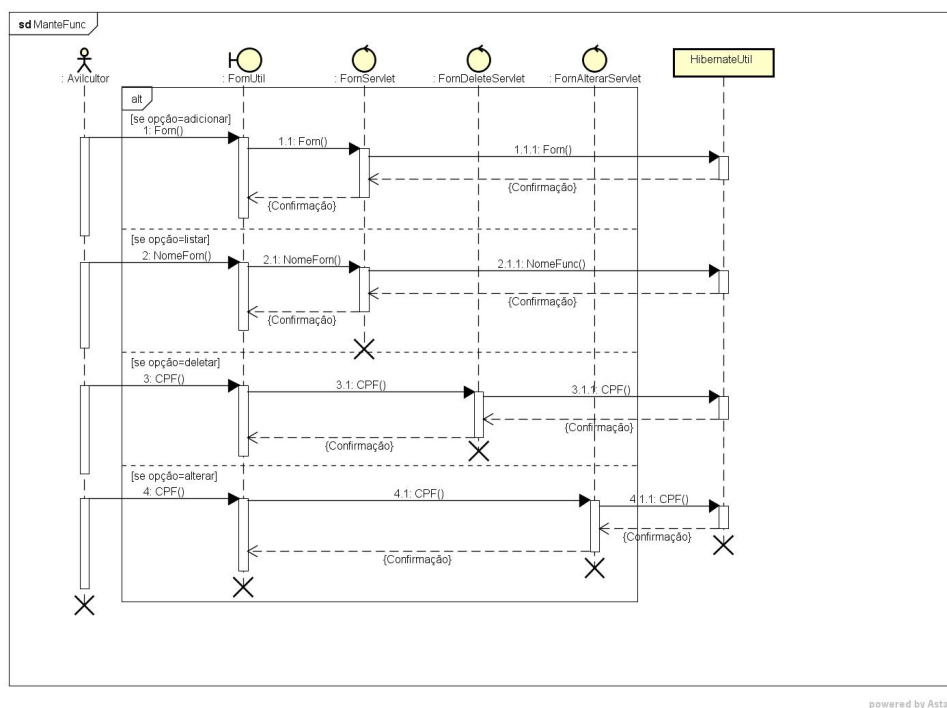
Figura 21 – Diagrama de sequencia Manter Fornecimento (fonte própria, feito no Astah).



### 9.9.9 Diagrama de sequência Manter Funcionário

No diagrama abaixo, o ator Avicultor poderá adicionar, listar, deletar e alterar o Funcionários. Para adicionar o ator fornece a interfaceUtil os dados do Funcionários e ela fornecerá ao controlador FuncionáriosServlet os dados e por fim Conectará a Classe Hibernate que faz conexão com o bando de dados. Para listar o ator fornecerá a interfaceUtil o nome do Fornecimento ela enviará ao controlador FuncionáriosServlet e por fim ao banco de dados. Para alterar e excluir o usuário vai inserir os dados a interface conectará aos controlador FuncionáriosAlterarServlet e FuncionáriosExcluirServlet respectivamente.

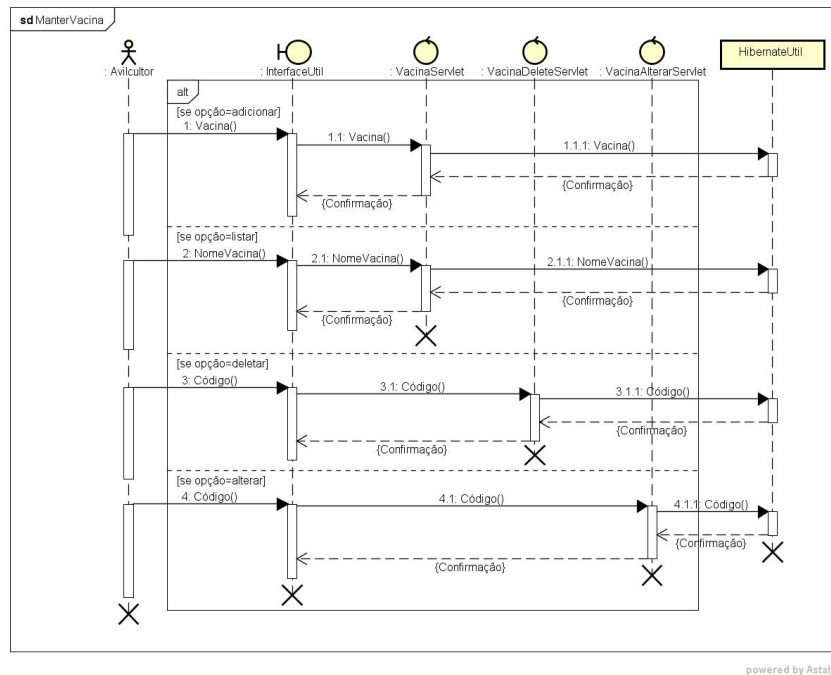
Figura 22 – Diagrama de sequencia Manter Funcionários (fonte própria, feito no Astah).



### 9.9.10 Diagrama de sequência Manter Vacina

No diagrama abaixo, o ator Avicultor poderá adicionar, listar, deletar e alterar o Vacina. Para adicionar o ator fornece a interface Util os dados do Vacina e ela fornecerá ao controlador VacinasServlet os dados e por fim Conectará a Classe Hibernate que faz conexão com o bando de dados. Para listar, o ator fornecerá a interface Util o nome da Vacina e ela enviará ao controlador VacinaServlet e por fim ao banco de dados. Para alterar e excluir o usuário vai inserir os dados a interface conectará aos controlador VacinaAlterarServlet e VacinaExcluirServlet respectivamente.

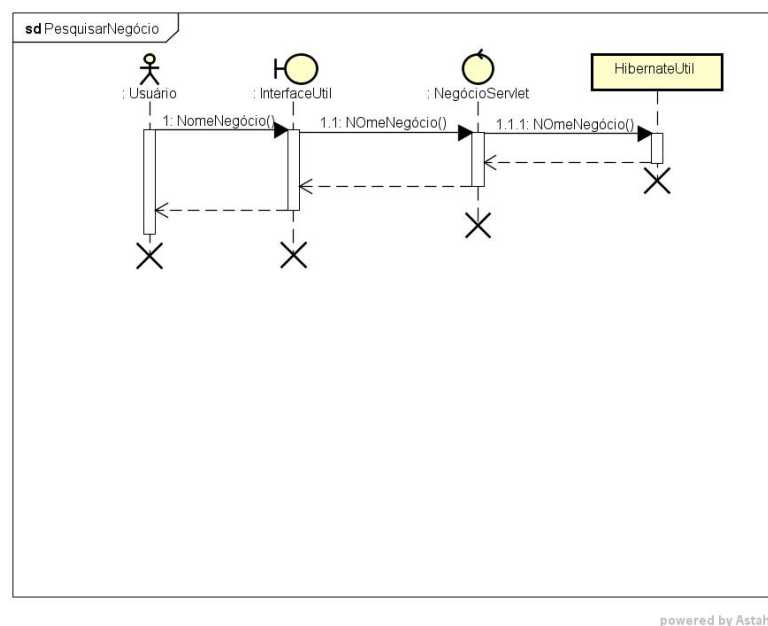
Figura 23 – Diagrama de sequencia Manter Vacina (fonte própria, feito no Astah).



### 9.9.11 Diagrama de sequência Pesquisar Negócio

No digrama abaixo, o Avicultor poderá pesquisar outros negócios informando a interfaceUtil o nome e ela fornecerá a NegocioServlet e por fim ao bando de dados, que retornará à interface retornará uma lista:

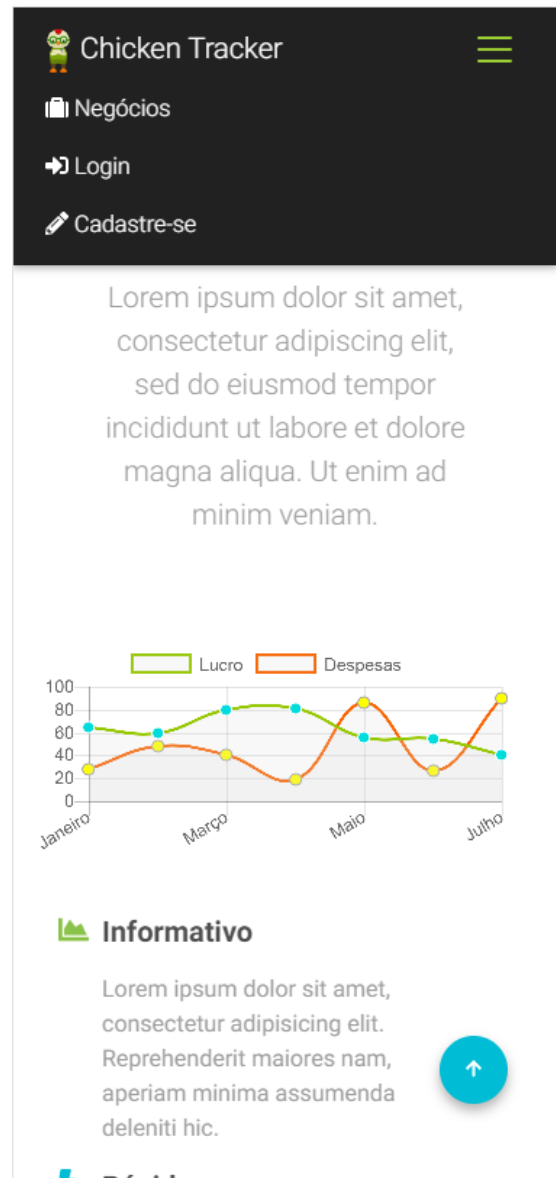
Figura 24 – Diagrama de sequencia Pesquisar Negócio (fonte própria, feito no Astah).



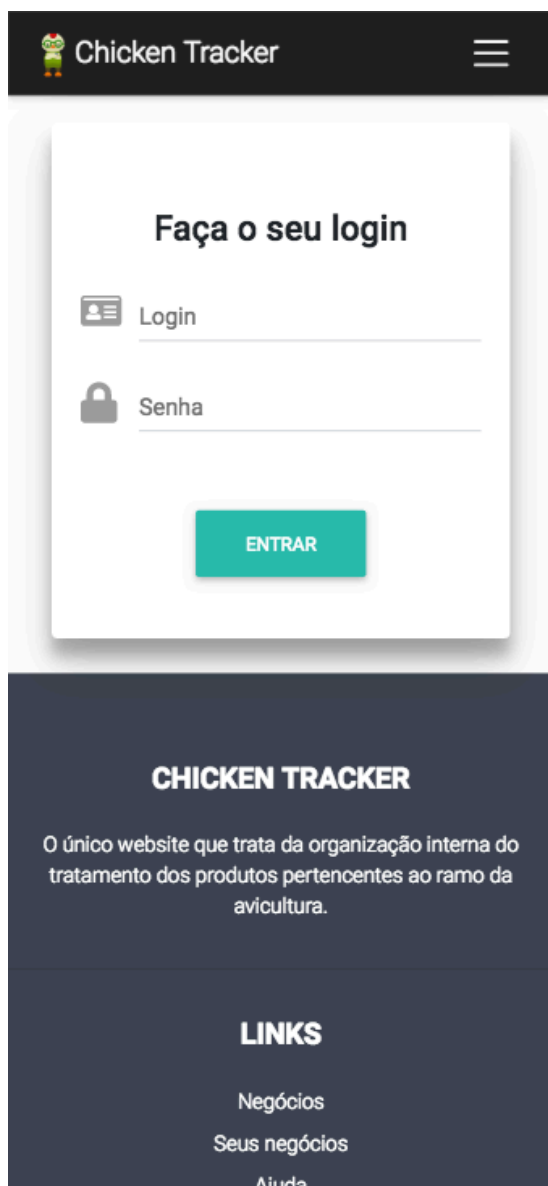
## 10 Projeto de telas



(a) Página inicial do Chicken Tracker (fonte própria).



(b) Página inicial mostrando menu drop-down (fonte própria).



**Faça o seu login**

Login

Senha

**ENTRAR**

**CHICKEN TRACKER**

O único website que trata da organização interna do tratamento dos produtos pertencentes ao ramo da avicultura.

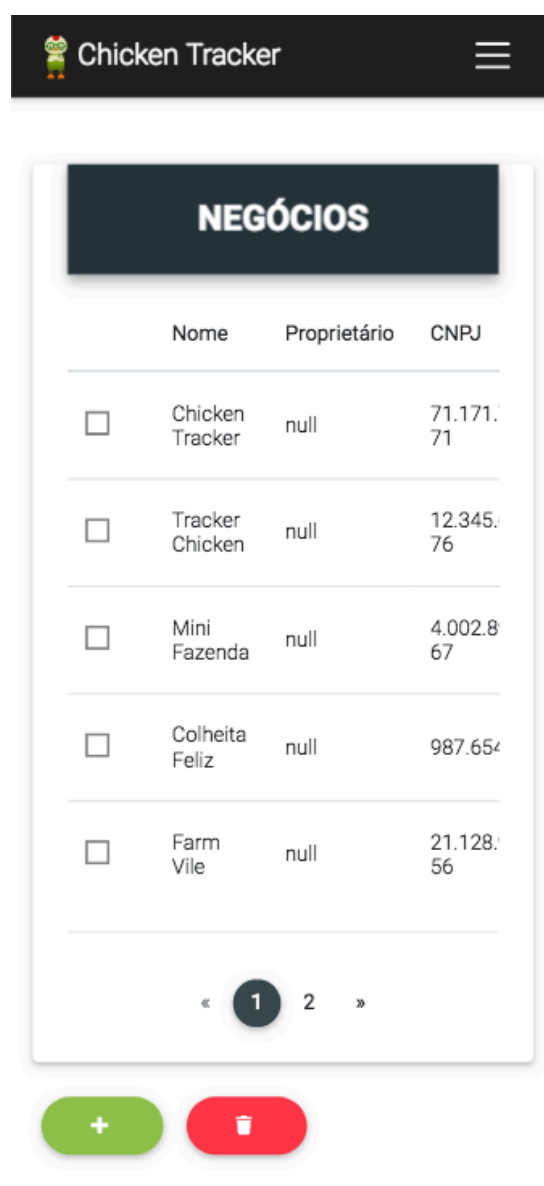
**LINKS**

Negócios

Seus negócios

Ajuda

(c) Página que mostra formulário login do site (fonte própria).



	Nome	Proprietário	CNPJ
<input type="checkbox"/>	Chicken Tracker	null	71.171.71
<input type="checkbox"/>	Tracker Chicken	null	12.345.76
<input type="checkbox"/>	Mini Fazenda	null	4.002.867
<input type="checkbox"/>	Colheita Feliz	null	987.654
<input type="checkbox"/>	Farm Vile	null	21.128.56

« 1 2 »

**+** **🗑**

(d) Página que mostra a lista de negócios de um usuário logado (fonte própria).

## 11 Cronograma

Atividades e/ou etapas conforme mostrado na tabela 1:

1. Modelagem do Diagrama de caso de uso e definição do layout da aplicação;
2. Criação dos diagramas de classe e sequência;
3. Programação dos casos de uso Criar Entidade, Selecionar entidade e Criar Relacionamento;
4. Programação dos casos de uso Criar generalização e Criar agregação;
5. Programação do caso de uso Compilar;

6. Teste com as turmas de segundo ano do ensino médio do curso de informática do IFAM CMC em 2018;
7. Correção das possíveis falhas;
8. Finalização da programação e documentação;

Tabela 23 – Cronograma de atividades

Atividades	2017					2018						
	8	9	10	11	12	1	2	3	4	5	6	7
1	x											
2	x											
3		x	x									
4				x								
5					x	x						
6								x		x		
7							x		x		x	
8											x	

## 12 Informações adicionais

O mercado de aves do Brasil está crescendo cada dia mais, por isso a ideia de um website proporcionará uma contribuição para os avicultores que contarão com um website para melhorar sua organização e sua aparência para demais empresas e pessoas que vão atrás de negócios de avicultores, um dos principais da agropecuária. Os quatro componentes designados para esse projeto são alunos formandos do ensino técnico médio em informática de nível integrado pelo Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM). São alunos com notas qualificadas para projetos dentro do IFAM e possuem um bom desempenho em sala de aula. Esses já desenvolveram outros projetos, porém este será o maior desafio deles por não conhecerem adequadamente as linguagens que serão utilizadas e o assunto tratado.

## 13 Resultados esperados

O projeto irá proporcionar uma melhor organização dos negócios de avicultores, aumentando a produtividade e qualidade dos produtos, crescendo a produção de aves de corte e postura, mantendo a produção necessária para o mercado consumidor, da mesma maneira que aumentará o número de exportações. É esperado também a diminuição de estragos e desperdício, já que os avicultores e donos dos negócios terão ideia da quantidade necessária para produzir e manter seus consumidores.

## 14 Conclusão - considerações finais

O objetivo desde trabalho foi mostrar a área que o projeto estará inserido e a importância dele para o estado brasileiro. Nele, é possível verificar toda a metodologia

adotada e a modelagem do sistema. Para equipe desenvolvedora, o desafio esperado foi exagerado, tornando-se mais viável ao começar o desenvolvimento, já que existem vários fóruns e as próprias documentações das linguagens e frameworks que auxiliam o projeto. Os alunos ganharam bastante experiência com a utilização de novas linguagens e novas IDE's, o que contribui para o desenvolvimento acadêmico e profissional de cada um.

## Referências

- 1 AGROFRANGO - Criação de Frango de corte. 2018. <<http://www.agromanager.com.br/solucoes/avicultura/agrofrango>>. Citado na página 9.
- 2 QUEVEDO, A. *A história da avicultura brasileira*. 2003. Disponível em: <<https://www.aviculturaindustrial.com.br/imprensa/a-historia-da-avicultura-brasileira/20030520-151203-0539>>. Citado 2 vezes nas páginas 9 e 10.
- 3 FRANCO, A. S. M. A avicultura no brasil. *Análise Conjuntural - IPARDES*, 2017. Citado na página 10.
- 4 A importância econômica e social da avicultura. *Produção animal - Avicultura*, n. 54, p. 4, 2011. Disponível em: <[http://www.avisite.com.br/revista/pdfs/revista\\_edicao54.pdf](http://www.avisite.com.br/revista/pdfs/revista_edicao54.pdf)>. Citado na página 10.
- 5 PT, S. *Porque é importante uma empresa ou negócio ter um site*. 2018. Disponível em: <<https://www.site.pt/blog/porque-importante-empresa-negocio-site/>>. Citado na página 11.