



**GABRIEL PIOVESAN MELCHIORI PERUZZA**

**DESENVOLVIMENTO DO SISTEMA WEB SENSORESPONSE:  
UM SOFTWARE PARA PREDIÇÃO DE RESPOSTAS SENSORIAIS  
POR MEIO DE PARÂMETROS FÍSICO-QUÍMICOS**

**LAVRAS – MG**

**2022**

**GABRIEL PIOVESAN MELCHIORI PERUZZA**

**DESENVOLVIMENTO DO SISTEMA WEB SENSORESPONSE:  
UM SOFTWARE PARA PREDIÇÃO DE RESPOSTAS SENSORIAIS POR MEIO DE  
PARÂMETROS FÍSICO-QUÍMICOS**

Relatório técnico apresentado à Universidade Federal de Lavras, como parte das exigências para obtenção do título de Bacharel em Ciência da Computação.

Profa. Dra. Ana Paula Piovesan Melchiori  
Orientadora

**LAVRAS – MG**  
**2022**

**GABRIEL PIOVESAN MELCHIORI PERUZZA**

**DESENVOLVIMENTO DO SISTEMA WEB SENSORESPONSE: UM SOFTWARE  
PARA PREDIÇÃO DE RESPOSTAS SENSORIAIS POR MEIO DE PARÂMETROS  
FÍSICO-QUÍMICOS**

Relatório técnico apresentado à Universidade Federal de Lavras, como parte das exigências para obtenção do título de Bacharel em Ciência da Computação.

APROVADA em 28 de Abril de 2022.

Ana Paula Piovesan Melchiori UFLA  
Ana Carla Marques Pinheiro UFLA  
Joaquim Quinteiro Uchôa UFLA  
Juliana Galvani Greggi UFLA

Profa. Dra. Ana Paula Piovesan Melchiori  
Orientadora

**LAVRAS – MG  
2022**

## RESUMO

O sistema SensoResponse consiste em uma plataforma Web, capaz de informar aos produtores de frutas os parâmetros de aceitação de seus produtos, por meio da análise dos parâmetros físico-químicos das amostras fornecidas ao sistema. A proposta é uma evolução do aplicativo SensoResponse para a plataforma Web, pensada pela Profa. Ana Carla Marques Pinheiro em conjunto com a Profa. Dra. Ana Paula Piovesan Melchiori, com o objetivo de tornar o sistema mais acessível e rentável. Além disso, a aplicação é hospedada pelo próprio sistema de servidores da Universidade Federal de Lavras (UFLA), o que garante maior controle de acesso e segurança do sistema.

**Palavras-chave:** Desenvolvimento Web, Análise sensorial, Frutas, Parâmetros físico-químicos.

## LISTA DE FIGURAS

Figura 2.1 – Logotipo Windows 10 . . . . .	11
Figura 2.2 – Logotipo Linguagem Java . . . . .	12
Figura 2.3 – Logotipo Play Framework . . . . .	13
Figura 2.4 – Logotipo Node.js® . . . . .	13
Figura 2.5 – Logotipo Bower . . . . .	14
Figura 2.6 – Logotipo Grunt . . . . .	15
Figura 2.7 – Logotipo Visual Studio Code . . . . .	16
Figura 2.8 – Logotipo IntelliJ Idea . . . . .	16
Figura 2.9 – Logotipo PostgreSQL . . . . .	17
Figura 2.10 – Logotipo PgAdmin4 . . . . .	17
Figura 2.11 – Logotipo NGINX . . . . .	17
Figura 2.12 – Representação do modelo MVC . . . . .	19
Figura 3.1 – Tela home . . . . .	23
Figura 3.2 – Tela de cadastro de usuário . . . . .	23
Figura 3.3 – Tela de login de usuário . . . . .	24
Figura 3.4 – Tela de perfil do usuário . . . . .	24
Figura 3.5 – Tela de inserção de amostras . . . . .	25
Figura 3.6 – Tela de listagem de amostras . . . . .	25
Figura 3.7 – Verificação de formulário do sistema . . . . .	26
Figura 3.8 – Modal indicando erro de cadastro de usuário . . . . .	26
Figura 4.1 – Modelo representacional da metodologia Scrum . . . . .	28
Figura 1 – Representação do Espaço de cor CIE L*C*h° . . . . .	32

## LISTA DE TABELAS

Tabela 2.1 – Requisitos Funcionais do sistema . . . . .	20
Tabela 2.2 – Requisitos Não Funcionais do sistema . . . . .	20

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>6</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>8</b>
<b>2.1</b>	<b>Análise Sensorial</b>	<b>8</b>
<b>2.1.1</b>	<b>Relevância do processo de análise sensorial</b>	<b>8</b>
<b>2.1.2</b>	<b>Processo de análise utilizado no sistema</b>	<b>8</b>
<b>2.1.2.1</b>	<b>Predição da resposta sensorial</b>	<b>9</b>
<b>2.1.2.2</b>	<b>Parâmetros de saída da plataforma Web</b>	<b>10</b>
<b>2.2</b>	<b>Tecnologias Utilizadas</b>	<b>11</b>
<b>2.2.1</b>	<b>Sistemas Operacionais</b>	<b>11</b>
<b>2.2.2</b>	<b>Linguagens de Programação</b>	<b>11</b>
<b>2.2.3</b>	<b>Play Framework</b>	<b>12</b>
<b>2.2.4</b>	<b>Node.js</b>	<b>13</b>
<b>2.2.5</b>	<b>Bower</b>	<b>14</b>
<b>2.2.6</b>	<b>Grunt</b>	<b>14</b>
<b>2.2.7</b>	<b>Ambientes de Desenvolvimento Integrado (IDEs)</b>	<b>15</b>
<b>2.2.8</b>	<b>Banco de Dados</b>	<b>16</b>
<b>2.2.9</b>	<b>NGINX</b>	<b>17</b>
<b>2.3</b>	<b>Modelo MVC</b>	<b>18</b>
<b>2.4</b>	<b>Atividades Desenvolvidas</b>	<b>19</b>
<b>2.4.1</b>	<b>Análise de Requisitos</b>	<b>19</b>
<b>2.4.1.1</b>	<b>Requisitos Funcionais</b>	<b>20</b>
<b>2.4.1.2</b>	<b>Requisitos Não Funcionais</b>	<b>20</b>
<b>2.4.2</b>	<b>Configuração de Ambiente</b>	<b>21</b>
<b>2.4.3</b>	<b>Desenvolvimento da Aplicação</b>	<b>21</b>
<b>2.4.4</b>	<b>Configuração do Servidor</b>	<b>22</b>
<b>3</b>	<b>Resultados Alcançados</b>	<b>23</b>
<b>4</b>	<b>METODOLOGIA</b>	<b>27</b>
<b>4.1</b>	<b>Metodologia Scrum</b>	<b>27</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>29</b>
	<b>REFERÊNCIAS</b>	<b>30</b>
	<b>APENDICE A – Espaço de cor CIE L*C*h°</b>	<b>32</b>

## 1 INTRODUÇÃO

No Brasil, os frutos são comercializados sem qualquer controle de sua qualidade intrínseca, ou seja, sem a análise de seus aspectos físico-químicos e sensoriais. Essa falta de controle leva à comercialização de frutas com base, principalmente, na oferta e demanda de produtos no mercado, fazendo com que o consumidor adquira produtos de baixa qualidade, o que, conseqüentemente, causa insatisfação e frustração do cliente.

O uso de modelos preditivos (modelos matemáticos) pode ser eficientemente desenvolvido e aplicado para prever respostas sensoriais a partir de dados físicos e físico-químicos. O estabelecimento de padrões mínimos de qualidade de diferentes tipos de frutas e o desenvolvimento de modelos preditivos de respostas sensoriais, baseados em análises físicas e físico-químicas simples e rápidas, é capaz de promover o aprimoramento do controle de qualidade de frutas. O processo de análise sensorial das frutas é explicado mais detalhadamente na seção 2.1.

Foi criado um projeto para desenvolvimento de um sistema Web, intitulado “Desenvolvimento de modelos preditivos de aceitabilidade e padrões mínimos de qualidade de diferentes tipos de frutas”, coordenado pela Profa. Ana Carla Marques Pinheiro, pertencente ao Departamento de Ciência dos Alimentos (DCA) da UFLA, enviado e financiado pela FAPEMIG (APQ-02283-15), desenvolvido voluntariamente por Gabriel Piovesan Melchiori Peruzza, estudante de Ciência da Computação pela UFLA, sob orientação da Profa. Dra. Ana Paula Piovesan Melchiori, pertencente ao Departamento de Computação Aplicada (DAC).

O sistema Web, nomeado como SensoResponse, está adaptado apenas para ser acessado via computador, exibindo uma aplicação desconfigurada ao ser acessado por meio de um dispositivo móvel. Sua primeira versão disponibilizada consiste numa aplicação simples, que deve ser aprimorada para melhorar a experiência do usuário. Por falta de voluntários ao projeto, o sistema foi desenvolvido apenas por uma única pessoa, sem a disponibilização de protótipos ou designs referentes ao visual esperado. As tecnologias utilizadas no desenvolvimento de todo o sistema pode ser encontrado na seção 2.2, as quais são explicadas detalhadamente. Além disso, o link de acesso ao sistema pode ser encontrado na referência a seguir (PERUZZA, 2021).

A aplicação SensoResponse busca ser uma importante ferramenta de consulta, tanto para os produtores quanto para os consumidores, além de proporcionar aos centros de distribuição a capacidade de classificar e separar as frutas com base em seus aspectos intrínsecos como, por exemplo, sua acidez. Esses dados servem como subsídio para os modelos preditivos utilizados pelo sistema, os quais fornecem uma predição da resposta sensorial da fruta.



A predição, posteriormente, é utilizada como uma forma de prever a qualidade da amostra analisada. Os consumidores terão informações mais claras acerca do produto que estão adquirindo, podendo avaliar o custo e benefício no momento da aquisição, garantindo sua satisfação, enquanto os produtores poderão garantir melhores preços e maior qualidade para os produtos.

## **2 REFERENCIAL TEÓRICO**

Este capítulo destina-se a proporcionar uma visão mais ampla sobre o processo de análise sensorial das frutas, além de discorrer sobre as tecnologias utilizadas no processo de desenvolvimento da aplicação, seu processo de estruturação e disponibilização ao usuário.

### **2.1 Análise Sensorial**

A análise sensorial consiste na medição, análise e interpretação das reações percebidas pelos cinco sentidos (visão, olfato, gosto, tato e audição) com relação às características dos alimentos. Desse modo, pode-se avaliar a qualidade dos alimentos por meio de diversos fatores como seus parâmetros físicos e sensoriais (TEIXEIRA, 2009).

#### **2.1.1 Relevância do processo de análise sensorial**

As características sensoriais dos alimentos são relevantes, pois são fruto da interação entre o produto e o consumidor. Com esses parâmetros, pode-se fazer uma avaliação verossímil das características dos alimentos, sendo a avaliação sensorial um fator primordial na determinação da vida útil de diversos produtos alimentícios (FONSECA, 2014).

O aumento da inclusão de frutas no cardápio pela população brasileira faz com que se torne importante o entendimento do comportamento do consumidor brasileiro, especialmente seus critérios utilizados para escolha desse tipo de produto, os quais são a base para a definição de estratégias destinadas ao mercado (CORRÊA, 2014).

Com a avaliação sensorial, torna-se viável a precificação de frutas com base em suas notas recebidas na aceitação sensorial, além de proporcionar produtos de melhor qualidade e uma maior satisfação do consumidor (CORRÊA, 2014).

#### **2.1.2 Processo de análise utilizado no sistema**

O processo de análise utilizado pelo sistema consiste na utilização de alguns modelos matemáticos, baseados na combinação de constantes e características base de cada espécie de fruta com as características fornecidas pelo usuário. Esses parâmetros fornecidos pelo usuário são os parâmetros físico-químicos obtidos no estudo das características de sua amostra, sendo posteriormente utilizados para realizar a predição da resposta sensorial da fruta.

Podemos separar os parâmetros utilizados e fornecidos pelo sistema em duas classes diferentes: os dados físico-químicos, os quais são os dados de entrada para o sistema, e as respostas sensoriais calculadas a partir de modelos multivariados, que consistem nos dados de saída do sistema.

### 2.1.2.1 Predição da resposta sensorial

As características base de cada espécie de fruta são fornecidas por um administrador do sistema, enquanto as características de amostra são fornecidas pelos usuários. Esses parâmetros consistem em:

- **Potencial Hidrogênico:** O potencial hidrogênico (pH) indica a acidez ou basicidade da fruta.
- **Acidez Titulável:** A acidez titulável representa a quantidade de ácido presente na fruta, sendo seu valor inversamente proporcional ao do pH. Ou seja, quanto maior o pH, menor a acidez titulável (e vice-versa).
- **Sólidos Solúveis:** Consiste no total de sólidos dissolvidos em água, sendo eles açúcares, sais, proteínas, entre outros.
- **L\*:** A coordenada L\* faz parte do sistema de classificação da coloração da fruta, representando o quão clara ou escura ela é, ou seja, varia do preto ao branco.
- **a\*:** A coordenada a\* faz parte do sistema de classificação da coloração da fruta, representando a variação do verde ao vermelho.
- **b\*:** A coordenada b\* faz parte do sistema de classificação da coloração da fruta, representando a variação do azul ao amarelo.
- **c\*:** Representa a saturação, ou seja, a concentração da cor na fruta, obtido a partir de um cálculo que utiliza como parâmetros a\* e b\*.
- **h°** Medição do ângulo entre a\* e b\*, utilizado para verificar a coloração da fruta com mais facilidade.
- **Firmeza:** Firmeza da fruta, obtida por meio de teste de punção.
- **Altura:** Altura da fruta.

- **Diâmetro:** Diâmetro da fruta.
- **Rendimento:** Quantidade de suco fornecida pela fruta.

Para compreender melhor a análise de coloração das frutas, visite o Apêndice A. Todos os parâmetros utilizados na análise sensorial e suas respectivas descrições são baseados na tese de predição da aceitação sensorial de (RIBEIRO, 2018). Além disso, há parâmetros que não são utilizados pelos modelos matemáticos de predição do sistema, sendo eles:

- **Nome da fruta:** Utilizado apenas como forma de identificação da fruta no sistema, garantindo que não hajam duplicatas.
- **Nome da amostra:** Utilizado apenas como forma de identificação da amostra no sistema, facilitando ao usuário pesquisar e encontrar a amostra desejada.

#### 2.1.2.2 Parâmetros de saída da plataforma Web

O usuário, após cadastrar uma amostra de uma fruta no sistema, é capaz de solicitar as predições de sua resposta sensorial. Os parâmetros calculados e fornecidos pelo sistema são:

- **Aceitação:** Impressão global da fruta, ou seja, o quão agradável ela é aos sentidos. Valores baixos indicam baixa aceitação, enquanto valores altos indicam alta aceitação.
- **Expectativa:** Expectativa com relação à qualidade da fruta. Valores baixos indicam baixa expectativa, enquanto valores altos indicam alta expectativa.
- **Doçura Ideal:** Doçura da fruta. Valores baixos indicam que a fruta é menos doce do que o ideal, enquanto valores altos indicam que a fruta é mais doce do que o ideal. A doçura ideal representa o valor médio.
- **Acidez Ideal:** Acidez da fruta. Valores baixos indicam que a fruta é menos ácida do que o ideal, enquanto valores altos indicam que a fruta é mais ácida do que o ideal. A acidez ideal representa o valor médio.
- **Suculência Ideal:** Suculência da fruta. Valores baixos indicam que a fruta é menos suculenta do que o ideal, enquanto valores altos indicam que a fruta é mais suculenta do que o ideal. A suculência ideal representa o valor médio.

## 2.2 Tecnologias Utilizadas

As tecnologias foram escolhidas com base no conhecimento prévio do voluntário, pelo fato da necessidade urgente de desenvolvimento do sistema. Devido a isso, não foram cogitadas outras tecnologias que poderiam ser mais atuais ou mais apropriadas para o desenvolvimento, pois o processo de aprendizado de novas tecnologias levaria a um possível atraso.

### 2.2.1 Sistemas Operacionais

O sistema Windows 10 foi utilizado para desenvolvimento da aplicação, devido ao costume do voluntário de utilizá-lo diariamente. Porém, para configuração de servidor dentro do domínio da universidade, foi utilizado o sistema Linux Ubuntu 18.04, por conta de seus protocolos de segurança e facilidade de configuração de acesso remoto.

Figura 2.1 – Logotipo Windows 10



Fonte: Olhar Digital, 2020.

### 2.2.2 Linguagens de Programação

As linguagens de programação utilizadas podem ser separadas nas categorias de frontend, ou seja, a parte visível ao usuário, o backend, que consiste na parte não visível ao usuário e o banco de dados, responsável por armazenar os dados necessários para o funcionamento correto do sistema. Essa separação busca facilitar o entendimento das tecnologias e relacioná-las com mais precisão.

No frontend, foram utilizadas as linguagens HTML, CSS e JavaScript. Toda a parte de organização e disposição dos elementos das telas presentes no sistema é feita em HTML, enquanto para sua estilização é utilizado o CSS. Já o JavaScript é utilizado em todo o processo

de criação de variáveis e possíveis cálculos, fornecendo esses parâmetros ao HTML. Além disso, o JavaScript é responsável pela comunicação entre frontend e backend.

Para o backend, foi utilizada a linguagem de programação Java (versão 8), versão escolhida por ser compatível com o framework utilizado para o backend, descrito na seção 2.2.3. Além disso, a linguagem Java proporciona uma obtenção de dados do banco de dados por meio de objetos, os quais são predefinidos pelo desenvolvedor.

Já a linguagem utilizada no banco de dados é a SQL, linguagem padrão utilizada pelos bancos de dados relacionais. Mais informações sobre o banco de dados e suas tecnologias estão descritas na seção 2.2.8.

Figura 2.2 – Logotipo Linguagem Java



Fonte: Ravel Tecnologia, 2022.

### 2.2.3 Play Framework

É uma Interface de Programação de Aplicações (API), capaz de se comunicar com outras aplicações por meio de requisições. As APIs simplificam a conexão entre as diversas estruturas de um projeto, facilitando o processo de verificação e consulta de um servidor (O..., 2017).

O framework possui uma ferramenta de recompilação automática, que permite ao programador modificar os arquivos referentes ao backend sem que haja necessidade de recompilar o programa manualmente (Play..., 2021), facilitando o processo de desenvolvimento da aplicação.

Sua estruturação no modelo MVC, explicado na seção 2.3), e a configuração de rotas para requisições do modelo de arquitetura REST permite uma comunicação bem limpa, devido à configuração de rotas específicas para diferentes requisições. Além disso, a utilização da Java Persistence API (JPA) faz com que haja um mapeamento correto dos objetos buscados no banco de dados, retornando um objeto baseado em uma *Model*.

Outra ferramenta facilitadora providenciada pelo Play é a configuração de *evolutions*, que são utilizadas para configurar o banco de dados utilizado pelo sistema. Uma explicação mais detalhada de seu funcionamento pode ser encontrada na seção 2.2.8.

Figura 2.3 – Logotipo Play Framework



Fonte: Lightbend, 2021.

#### 2.2.4 Node.js

É um ambiente de execução JavaScript, utilizado em aplicações Web. Suas funções nativas raramente usam dados de entrada ou fornecem dados de saída, o que torna o sistema altamente escalável. Além disso, proporciona um sistema de requisições http com retorno de resposta (Node..., 2022), ideal para a execução de requisições integradas com o backend.

Além disso, algumas bibliotecas existentes nesse ambiente fornecem ferramentas que agilizam o processo de desenvolvimento de uma aplicação, ajudando o desenvolvedor a executar tarefas mais facilmente. Por exemplo, no sistema são utilizados as bibliotecas Bower (seção 2.2.5) e Grunt (seção 2.2.6), utilizadas para configuração do projeto.

Figura 2.4 – Logotipo Node.js®



Fonte: Opus Software, 2021.

### 2.2.5 Bower

É um gerenciador de pacotes pertencente ao Node.js (seção 2.2.4). Com a utilização do bower, a manutenção de versões de pacotes utilizados pelo sistema e sua instalação são facilitados, sendo manipulados por meio de comandos de instalação e configuração. Todos os pacotes são definidos dentro do arquivo "bower.json", sendo facilmente modificável e escalável (Bower..., 2022). Foi utilizado para configurar todos os pacotes e bibliotecas necessárias no decorrer do desenvolvimento do sistema.

Figura 2.5 – Logotipo Bower



Fonte: GitHub Incorporation, 2022.

### 2.2.6 Grunt

É um automatizador de tarefas, que permite ao desenvolvedor gerar *scripts* para realizar ações de maneira automática e programada (Grunt..., 2022). Foi utilizado para configurar a execução de tarefas repetitivas utilizadas pelo sistema, como o processo de compilação e organização dos arquivos para sua execução e disponibilização.

Uma das tarefas mais importantes executadas pelo Grunt é a reunião de todos os arquivos necessários à execução do sistema, tanto pertencentes ao backend quanto ao frontend, dentro de uma pasta específica, nomeada de "public". Com essa separação, a máquina utilizada como servidor recebe todos os arquivos concentrados em um mesmo diretório, permitindo que o sistema seja executado integralmente com apenas um comando.



Figura 2.6 – Logotipo Grunt



Fonte: felipefialho.com, 2022.

### 2.2.7 Ambientes de Desenvolvimento Integrado (IDEs)

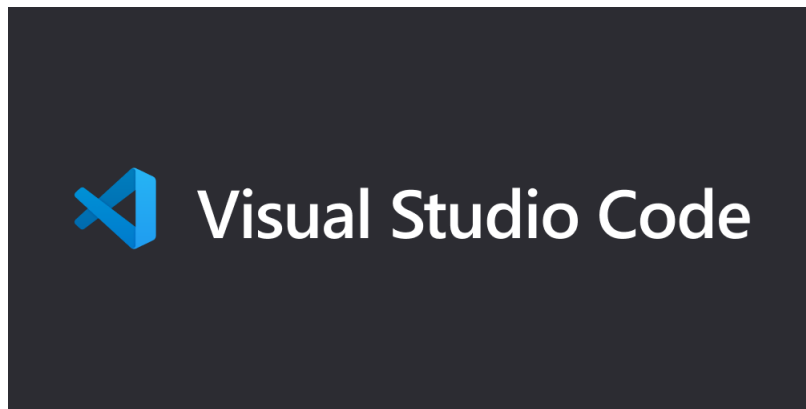
Foram utilizados dois ambientes de desenvolvimento durante o processo de desenvolvimento do software, devido à necessidade de algumas funcionalidades específicas e da separação entre frontend e backend.

Para a programação frontend foi utilizada a IDE Visual Studio Code, sendo escolhida com base na sua compatibilidade com as linguagens JavaScript, CSS e HTML, facilitando a programação e a prevenção de erros.

Já o backend foi desenvolvido utilizando a IDE IntelliJ IDEA, que possui compatibilidade com a linguagem Java e SQL. Além disso, possui a funcionalidade de inicializar o Play Framework, executando o servidor no próprio ambiente.

Ambas as IDEs possuem busca avançada de arquivos e seus conteúdos, além de indicar quais mudanças foram feitas no projeto. Com essas ferramentas, o desenvolvedor é capaz de corrigir erros e acessar os arquivos com mais facilidade.

Figura 2.7 – Logotipo Visual Studio Code



Fonte: Turma de Arquitetura de Software UFCG, 2020.

Figura 2.8 – Logotipo IntelliJ Idea



Fonte: Twitter Incorporation, 2022.

### 2.2.8 Banco de Dados

O banco de dados utilizado pela aplicação é o PostgreSQL, um banco de dados relacional (PostgreSQL..., 2022). A construção de sua instância referente à aplicação é feita por meio de *Evolutions*, providenciada pelo Play Framework (seção 2.2.3).

As *evolutions* consistem em arquivos de configuração de banco de dados, que são executados pelo próprio Play, populando o banco com as configurações fornecidas. Vários arquivos desse tipo podem ser configurados ao longo do desenvolvimento do sistema, a fim de atualizar o banco de dados sem a necessidade de alterar os arquivos antigos. Desse modo, é mantido um histórico da evolução do banco de dados.

Para facilitar a manipulação e visualização de dados existentes no banco, foi utilizada a Interface Gráfica do Usuário (GUI) providenciada pelo PostgreSQL, chamada PgAdmin (versão 4). A utilização de uma GUI facilita o acesso ao banco e a visualização de seus dados, agilizando o processo de desenvolvimento.

Figura 2.9 – Logotipo PostgreSQL



Fonte: ComparaSoftware SPA, 2021.

Figura 2.10 – Logotipo PgAdmin4



Fonte: Coffops, 2022.

### 2.2.9 NGINX

É uma ferramenta de proxy de servidor, gerenciando requisições externas e redirecionando-as para o servidor da aplicação (NGINX..., 2022). Foi utilizado para configurar o acesso remoto ao sistema, por meio do redirecionamento do endereço Web para a porta onde está sendo executado o programa, fazendo com que se possa acessar o sistema por meio de um navegador sem que haja acesso ao computador da universidade.

Figura 2.11 – Logotipo NGINX



Fonte: Google LLC, 2022.

## 2.3 Modelo MVC

MVC é um padrão arquitetural, cuja principal característica é facilitar o processo de organização do controle e desenvolvimento de um projeto. Em contrapartida, a análise e modelagem de um sistema nesse padrão tende a ser mais demorada (LEMOS et al., 2013). Seu nome provém das iniciais de suas subdivisões, as quais são explicadas a seguir.

Model é onde ocorre as operações básicas de um banco de dados, ou seja, as operações de *create*, *read*, *update* e *delete* (CRUD) (LEMOS et al., 2013). Sendo assim, a parte responsável pela comunicação com os dados armazenados pertence à Model.

View é a parte visual do sistema, ou seja, todos os arquivos referentes ao HTML do sistema (LEMOS et al., 2013). Podemos incluir também os arquivos de estilização do sistema, ou seja, os arquivos CSS, já que eles fazem parte da estilização do HTML. Sendo assim, a parte responsável pela maneira como o usuário visualiza o sistema pertence à View.

Controller é onde ocorre toda a lógica de processamento dos dados de entrada no sistema, além de administrar o fluxo da aplicação (LEMOS et al., 2013). Sendo assim, a parte responsável pelos cálculos e manipulações de dados são pertencentes à Controller.

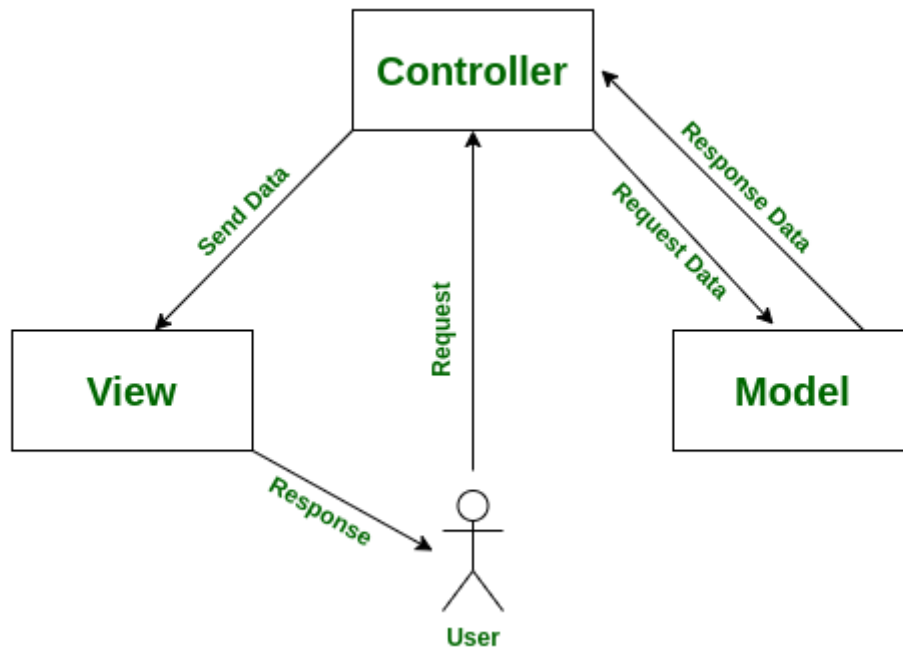
Com essas subdivisões reunidas, as quais se comunicam entre si, temos um sistema organizado no padrão de arquitetura MVC. A View recebe seus dados de exibição a partir da Controller que, por sua vez, obtém os dados armazenados por conta das operações realizadas pela Model.

No sistema desenvolvido, a Model é localizada no backend, sendo suas operações de consulta ao banco de dados realizada pelo Play Framework (seção 2.2.3). A View é localizada no frontend, onde se encontram todos os arquivos HTML e CSS. Já a Controller é dividida em duas partes, uma referente ao frontend e uma ao backend.

A Controller do frontend é responsável por manipular os dados de entrada fornecidos pelo usuário, a fim de enviá-los ao backend preparados e organizados para serem manipulados. A Controller do backend é responsável por manipular os dados fornecidos pelas operações da Model, a fim de enviá-los ao frontend conforme possíveis restrições.

No caso do sistema, ambas as Controllers ainda são responsáveis por se comunicarem entre si por meio da utilização de rotas. Desse modo, quando o usuário faz alguma requisição ao sistema que necessite de comunicação com o backend, há uma comunicação interna entre as Controllers que permite com que rotas do frontend possam se comunicar com as rotas do backend.

Figura 2.12 – Representação do modelo MVC



Fonte: Acervo Lima, 2022.

## 2.4 Atividades Desenvolvidas

Diversas atividades foram realizadas durante o desenvolvimento da aplicação, desde o processo de análise de requisitos e configuração de máquina até o desenvolvimento e disponibilização da aplicação.

### 2.4.1 Análise de Requisitos

Para o planejamento e análise dos requisitos do projeto, foi feita uma reunião com a solicitante do sistema e minha orientadora, além de diversas comunicações por mensagem durante o desenvolvimento da aplicação. Com as informações obtidas tanto pela reunião quanto pela troca de mensagens, é possível definir todos os requisitos necessários ao sistema.

Foi executada uma análise de todas as informações retiradas no momento da reunião, com o objetivo de definir quais eram os requisitos do sistema. Desse modo, foram obtidos os seus requisitos funcionais e não funcionais.

### 2.4.1.1 Requisitos Funcionais

Os requisitos funcionais descrevem o comportamento do sistema, ou seja, indicam o que o software deve fazer (MACHADO, 2016). São os requisitos mais importantes do sistema, visto que devem ser atendidos para que o sistema cumpra o papel que lhe foi designado. Para o projeto desenvolvido, definiram-se os requisitos funcionais da tabela a seguir.

Tabela 2.1 – Requisitos Funcionais do sistema

Requisito	Descrição
Gerenciar Usuários	Responsável pela inclusão, alteração, leitura e exclusão dos dados de usuário. Também é responsável pela autenticação de usuário no sistema.
Gerenciar Frutas	Responsável pela inclusão, alteração, leitura e exclusão das frutas do sistema por administradores.
Gerenciar Amostras	Responsável pela inclusão, alteração, leitura e exclusão das amostras do sistema por qualquer usuário, desde que a amostra pertença a ele.
Gerenciar Análise Sensorial	Responsável pelos cálculos relacionados à análise sensorial das amostras do usuário e sua disponibilização.

Fonte: Backlog do sistema Web SensoResponse.

### 2.4.1.2 Requisitos Não Funcionais

Os requisitos não funcionais não estão ligados diretamente às funções do sistema. Geralmente, descrevem ações necessárias para manutenção da segurança e disponibilidade da aplicação, garantindo que tudo esteja bem configurado (MACHADO, 2016). No sistema desenvolvido, foram descritos os requisitos não funcionais da tabela a seguir.

Tabela 2.2 – Requisitos Não Funcionais do sistema

Requisito	Descrição
Hospedagem da aplicação	Como forma de garantir a segurança e a disponibilidade da aplicação, sua hospedagem deve ser efetuada em máquina interna do servidor da UFLA.
Minimização de erros	Modais de confirmação de ação devem ser disponibilizadas em algumas ações, a fim de evitar erros.
Feedback ao usuário	Erros de preenchimento dos formulários e respostas do sistema devem ser informados ao usuário.

Fonte: Backlog do sistema Web SensoResponse.

## 2.4.2 Configuração de Ambiente

Para iniciar o processo de desenvolvimento do sistema, é necessário configurar a máquina para suportar as tecnologias escolhidas na configuração do projeto, além de preparar todo o ambiente de desenvolvimento.

Primeiramente, foram instaladas e configuradas todas as tecnologias utilizadas em suas respectivas versões. Desse modo, podemos prosseguir com as integrações necessárias, configurando as tecnologias para se comunicarem entre si.

Para o frontend, foi criado um projeto, o qual possui seus arquivos de configuração e de instalação de dependências, na IDE Visual Studio Code (seção 2.2.7). Foram executados comandos utilizando o Node.js (seção 2.2.4), Bower (seção 2.2.5) e Grunt (seção 2.2.6), a fim de instalar as dependências necessárias.

Para o backend, foi criado um projeto na IDE IntelliJ Idea (seção 2.2.7), utilizando do Play Framework (seção 2.2.3) para sua configuração. Por meio da configuração de um *script* de execução do servidor, o Play executa a aplicação, utilizando o computador local como forma de acesso, a partir de uma rota específica. Além disso, conecta-se ao banco de dados, para que o backend possa executar suas operações básicas.

Para o banco de dados, foram executadas as evolutions do Play Framework, criando o banco com as configurações desejadas. Para visualização das persistências de dados ao banco, foi configurado o PgAdmin com o acesso ao banco correspondente à aplicação (seção 2.2.8), a fim de facilitar a verificação dos dados armazenados.

## 2.4.3 Desenvolvimento da Aplicação

Consiste no desenvolvimento de toda a lógica referente ao sistema, ou seja, todos os cálculos, manipulações de objetos na Controller e suas persistências ao banco de dados na Model, além da disponibilização dos dados na View (seção 2.3).

Para manter o fluxo da aplicação no frontend, todas as requisições foram configuradas utilizando dados de sua Controller, sendo passados ao backend pela utilização de uma Service, que apenas interliga as rotas do frontend com as do backend.

As rotas do backend, ao serem requisitadas pelo frontend, são direcionadas para funções dentro de sua Controller, que é utilizada para filtrar os dados do banco de dados. Na busca de dados paginados, por exemplo, a Controller do backend busca apenas os dados necessários

por cada página. Após a lógica necessária executada, os dados são retornados ao frontend pela mesma rota chamada anteriormente.

A organização do projeto foi estabelecida utilizando-se uma divisão entre duas partes, uma responsável pelo controle de acesso ao sistema e uma responsável pelos cálculos e dados disponibilizados ao usuário. Além disso, testes foram realizados em cada componente integrante do sistema, com o objetivo de encontrar erros e exceções e corrigi-los antes da disponibilização da aplicação ao usuário.

#### **2.4.4 Configuração do Servidor**

Para disponibilização da aplicação ao usuário, utilizando-se o servidor fornecido pela UFLA, é necessária a configuração de redirecionamento de rotas e de servidor, instalando as dependências necessárias. O método utilizado para acessar a máquina utilizada como servidor consiste na utilização da Rede Privada Virtual (VPN) da universidade, que possui parâmetros de segurança e de transferência de dados.

Por meio de acesso SSL (Secure Sockets Layer), foi acessada a máquina remotamente para instalação e configuração das tecnologias utilizadas para execução do sistema. Ao acessar a máquina do sistema, apenas tem-se acesso por meio de prompt de comando, o que dificulta todo o processo de instalação das tecnologias. Após efetuadas as instalações, foi enviado a pasta referente ao projeto já pronta para execução, via FTP (File Transfer Protocol).

Foi utilizado o NGINX (seção 2.2.9) para configuração de redirecionamento de rotas, permitindo que o sistema possa ser acessado via Web. Ao digitar o endereço do site, o usuário consegue acessar o servidor local da máquina (PERUZZA, 2021). Além disso, foi criado um Serviço Linux, de forma que o servidor pudesse rodar em segundo plano. Para isso, foi criado um programa que executa o servidor, adicionando-o aos serviços do Linux.

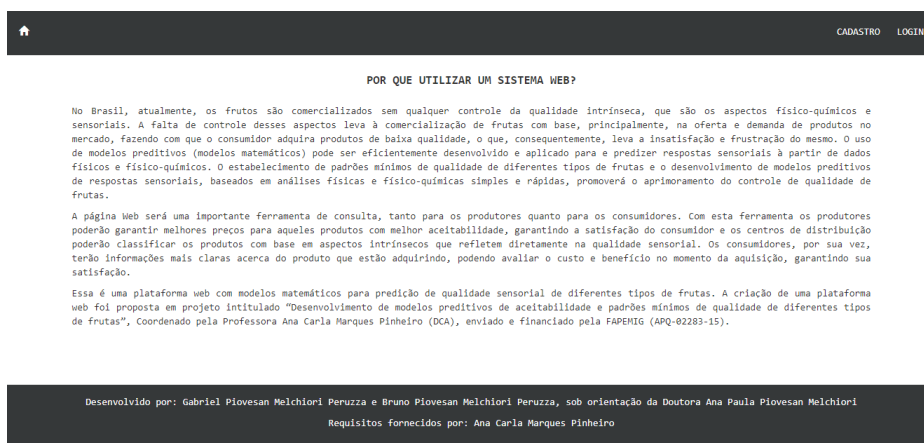
Caso o servidor falhe, o próprio sistema reinicializa o serviço, retomando a execução da aplicação. Por exemplo, caso haja uma queda de energia e o computador reinicialize, o serviço será rodado automaticamente no momento de inicialização do sistema, conforme configurado no Serviço Linux criado.



### 3 RESULTADOS ALCANÇADOS

Como produto de todo o desenvolvimento do projeto, chegamos ao final da versão 1 do sistema Web SensoResponse. A seguir, seguem capturas de tela do programa em execução, sendo acessado pelo site onde fica hospedado o projeto (PERUZZA, 2021). As capturas referem-se às telas acessíveis por um usuário comum do sistema, ou seja, sem privilégios de administrador.

Figura 3.1 – Tela home



Fonte: Imagem do autor, 2022

A tela home é exibida ao acessar o site, a qual contém uma breve descrição do que é o sistema e sua importância.

Figura 3.2 – Tela de cadastro de usuário

Fonte: Imagem do autor, 2022

A tela de cadastro de usuário é responsável por inserir novos usuários no sistema.

Figura 3.3 – Tela de login de usuário

Desenvolvido por: Gabriel Piovesan Melchiori Peruzza e Bruno Piovesan Melchiori Peruzza, sob orientação da Doutora Ana Paula Piovesan Melchiori  
Requisitos fornecidos por: Ana Carla Marques Pinheiro

Fonte: Imagem do autor, 2022

A tela de login de usuário é responsável por redirecionar o usuário para dentro da aplicação, caso suas credenciais estejam corretas.

Figura 3.4 – Tela de perfil do usuário

Fonte: Imagem do autor, 2022

A tela de perfil do usuário exibe as informações inseridas no cadastro do usuário. A imagem foi editada para preservar a privacidade dos dados.

Figura 3.5 – Tela de inserção de amostras

ADICIONAR AMOSTRA

Nome: \_\_\_\_\_

Fruta:   
Não há frutas adicionadas no sistema

Potencial Hidrogênico: \_\_\_\_\_

Acidez Titulável: \_\_\_\_\_

Sólidos Solúveis: \_\_\_\_\_

L\*: \_\_\_\_\_

a\*: \_\_\_\_\_

Fonte: Imagem do autor, 2022

A tela de inserção de amostras permite ao usuário inserir as amostras que deseja analisar no sistema.

Figura 3.6 – Tela de listagem de amostras

LISTAGEM DE AMOSTRAS

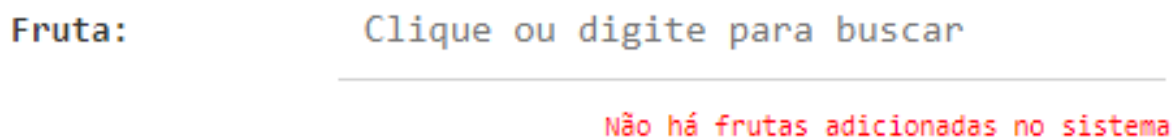
Não há frutas adicionadas no sistema

Desenvolvido por: Gabriel Piovesan Melchiori Peruzza e Bruno Piovesan Melchiori Peruzza, sob orientação da Doutora Ana Paula Piovesan Melchiori  
Requisitos fornecidos por: Ana Carla Marques Pinheiro

Fonte: Imagem do autor, 2022

A tela de listagem de amostras permite ao usuário visualizar as suas amostras inseridas, além de exibir seus dados e permitir efetuar a análise.

Figura 3.7 – Verificação de formulário do sistema



Fonte: Imagem do autor, 2022

Como o sistema não está sendo utilizado, ainda não há frutas cadastradas por um administrador, impedindo o usuário de realizar a inserção de uma fruta. Além disso, os formulários indicam quando algum campo está incorreto, como é o caso da situação citada anteriormente.

Figura 3.8 – Modal indicando erro de cadastro de usuário



Fonte: Imagem do autor, 2022

Outro ponto importante é a exibição de modais ao usuário, indicando respostas do servidor mediante algumas ações no sistema.

## 4 METODOLOGIA

Este capítulo destina-se a explicar a metodologia utilizada no desenvolvimento da explicação, descrever suas vantagens e como foi utilizada no sistema. A escolha de uma metodologia consistente pode ser um eficiente método para agilizar e organizar os métodos de análise e programação dos componentes integrantes do projeto.

### 4.1 Metodologia Scrum

É uma metodologia ágil, que tem como objetivo fornecer um modelo organizacional para o projeto, a fim de tornar as informações mais claras e o processo de desenvolvimento mais dinâmico (SOARES, 2004). Como a equipe de desenvolvimento foi constituída de apenas uma pessoa, alguns dos métodos utilizados na metodologia Scrum foram ignorados, pois alguns de seus processos só se fazem necessários quando há diversas pessoas envolvidas no projeto.

O Scrum é dividido em três fases principais, sendo elas o pré-planejamento, o desenvolvimento e o pós-planejamento (SOARES, 2004). Com essa divisão, facilita-se todo o processo de organização da aplicação, visto que todos os requisitos necessários ao projeto são meticulosamente descritos dentro de cada uma de suas fases.

A fase de pré-planejamento foi efetuada a partir do recolhimento dos requisitos com o requisitante do sistema, podendo assim planejar quais as ferramentas necessárias ao desenvolvimento e a definição das atividades a serem executadas.

O desenvolvimento foi planejado baseado nas tecnologias que o desenvolvedor domina, sendo desenvolvido de maneira flexível, permitindo que alterações futuras sejam implementadas sem a necessidade de reescrever alguma parte do código. Caso haja a necessidade de adição de pacotes ou de Controllers conforme evolução da aplicação, o sistema estará pronto para recebê-las, sem que haja interferência no que foi desenvolvido até então.

Após terminado o desenvolvimento, passamos para a fase de pós-planejamento. Antes de disponibilizar o sistema ao usuário, foram feitos testes unitários no sistema, a fim de averiguar se todos os requisitos necessários foram cumpridos. Com todo o projeto funcionando corretamente, efetuou-se a documentação do sistema, permitindo que futuros desenvolvedores consigam configurar seu ambiente de programação sem dificuldades.

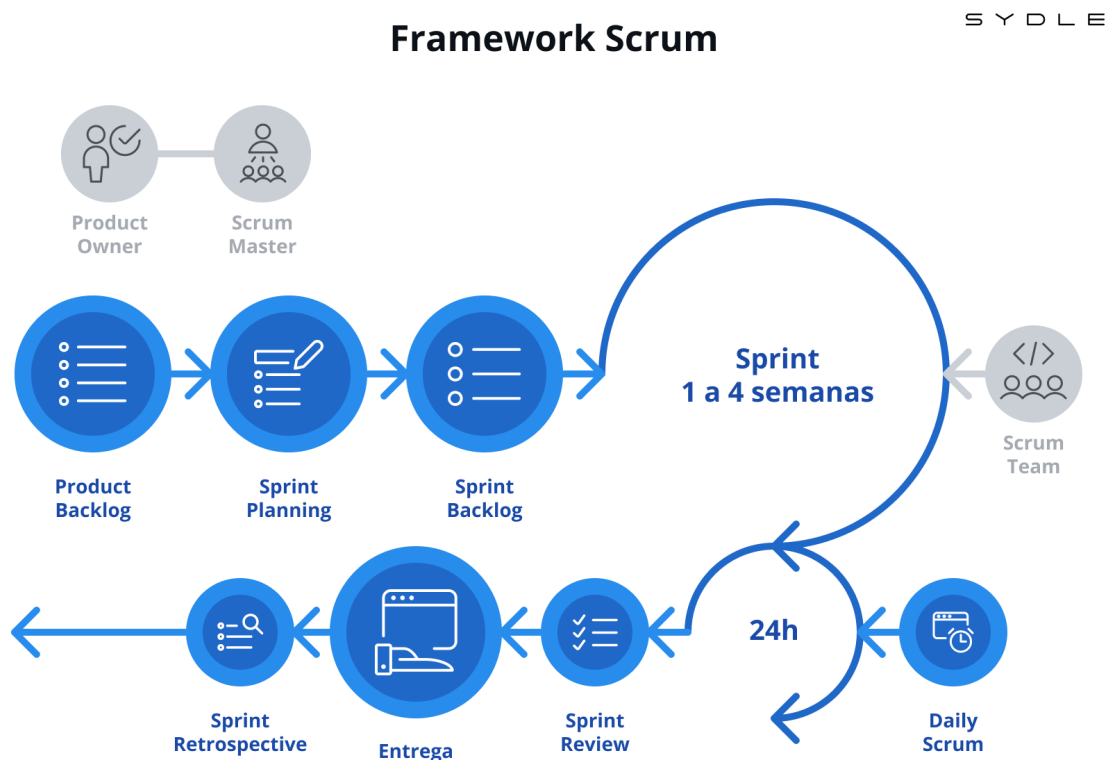
Além de todo o planejamento que foi executado, é importante que haja *feedback* dos usuários sobre a experiência ao utilizar o sistema. Como ainda não houve a oportunidade de

utilização do sistema pela requisitante, esse processo foi postergado, devendo ser efetuado antes do desenvolvimento e planejamento de possíveis melhorias.

Vale lembrar que essa metodologia utiliza de três principais cargos. O Product Owner é responsável por tomar todas as decisões do projeto, enquanto o Scrum Master gerencia o Time de Desenvolvimento, o qual é responsável pela execução do processo de desenvolvimento (SOARES, 2004). Todas essas funções foram executadas simultaneamente pelo voluntário, devido à falta de outros voluntários ao projeto.

Outra parte essencial do Scrum é a organização por Sprints, que são períodos de desenvolvimento limitados por um tempo determinado, a fim de entregar o produto desejado no tempo estipulado. No caso do projeto em questão, não foram utilizadas Sprints, visto que não havia um prazo específico para entrega.

Figura 4.1 – Modelo representacional da metodologia Scrum



Fonte: SYDLE, 2022.

## 5 CONCLUSÃO

O sistema Web SensoResponse tem como seu principal objetivo a disponibilização de uma aplicação acessível por qualquer navegador, visando facilitar seu acesso pelos usuários, sem que haja a necessidade de alguma instalação específica. Além disso, o projeto pode ser facilmente modificável, devido aos padrões de organização e programação utilizados no decorrer de seu planejamento e desenvolvimento.

Por se tratar da primeira versão desenvolvida do sistema e da escassez de recursos humanos, o sistema não foi baseado em um design, sendo sua aparência bem simples. Para que o sistema evolua, é necessário que hajam pesquisas de satisfação do usuário, obtendo-se as informações necessárias para o planejamento e implementação de possíveis mudanças e correções, caso seja necessário.

Como o sistema ainda não foi utilizado, não há como aferir o grau de facilidade de uso e de satisfação do usuário. Porém, com os requisitos do sistema definidos, é possível definir se o sistema cumpre ou não o seu papel. Sendo assim, após todos os testes realizados na aplicação, verificou-se que todos os requisitos cogitados foram atendidos, satisfazendo o planejamento previsto para o projeto.

## REFERÊNCIAS

- Bower home page. **Bower**, 2022. Acessado em: 01/04/2022. Disponível em: <<https://www.bower.io>>.
- Compreendendo o Espaço de Cor CIE L\*C\*h. **Konica Minolta Sensing Americas Incorporation**, 2020. Acessado em: 01/04/2022. Disponível em: <<https://sensing.konicaminolta.us/br/blog/compreendendo-o-espaco-de-cor-cie-lch/>>.
- CORRÊA, S. C. Predição da aceitação sensorial de frutas por meio de parâmetros físicos e físico-químicos utilizando modelo multivariado. **Universidade Federal de Lavras**, p. 60, 2014. Disponível em: <<http://repositorio.ufla.br/jspui/handle/1/2128>>.
- FONSECA, A. V. V. da. Perfil sensorial, aceitação e caracterização em compostos bioativos de néctares mistos de frutas tropicais. **Universidade Federal do Ceará - Biblioteca de Ciências e Tecnologia**, 2014. Disponível em: <<https://ppgcta.ufc.br/wp-content/uploads/2018/09/10-tesevalqudriavasconcelos.pdf>>.
- Grunt: The JavaScript Task Runner. **OpenJS Foundation**, 2022. Acessado em: 01/04/2022. Disponível em: <<https://gruntjs.com>>.
- LEMONS, M. F. de et al. APLICABILIDADE DA ARQUITETURA MVC EM UMA APLICAÇÃO WEB (Webapps). **Universidade José do Rosário Vellano - UNIFENAS**, 2013. Disponível em: <<https://revistas.unifenas.br/index.php/RE3C/article/view/54>>.
- MACHADO, F. N. R. **Análise e Gestão de Requisitos de Software – Onde nascem os sistemas**. São Paulo: Editora Érica, 2016. Disponível em: <<https://books.google.com.br/books?hl=pt-BR&lr=&id=MYdiDwAAQBAJ&oi=fnd&pg=PT5&dq=requisitos+de+um+software&ots=DkZgiBg-P6&sig=wZ9kOX9V7SYxgibYISwKTtPVgMY#v=onepage&q&f=false>>.
- NGINX home page. **F5 Networks Incorporation**, 2022. Acessado em: 01/04/2022. Disponível em: <<https://nginx.org/en/>>.
- Node.js about section. **NodeJS**, 2022. Acessado em: 01/04/2022. Disponível em: <<https://nodejs.org/en/about/>>.
- O que é uma API?. **RedHat Incorporation**, 2017. Acessado em: 01/04/2022. Disponível em: <<https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>>.
- PERUZZA, G. P. M. **Análise sensorial de alimentos: sistema Web SensoResponse**. 2021. Acessado em: 01/04/2022. Disponível em: <<http://sensorial.dca.ufla.br>>.
- Play Framework overview. **Lightbend**, 2021. Acessado em: 01/04/2022. Disponível em: <<https://www.playframework.com/documentation/1.5.x/overview>>.
- PostgreSQL about section. **The PostgreSQL Global Development Group**, 2022. Acessado em: 01/04/2022. Disponível em: <<https://www.postgresql.org/about/>>.
- RIBEIRO, M. N. Predição da aceitação sensorial e padrões de qualidade para diferentes tipos de frutas por parâmetros físicos e físico-químicos. **Universidade Federal de Lavras**, 2018. Relatório Técnico enviado para prestação de contas junto à FAPEMIG (Processo APQ-02283-15 da Demanda Universal).



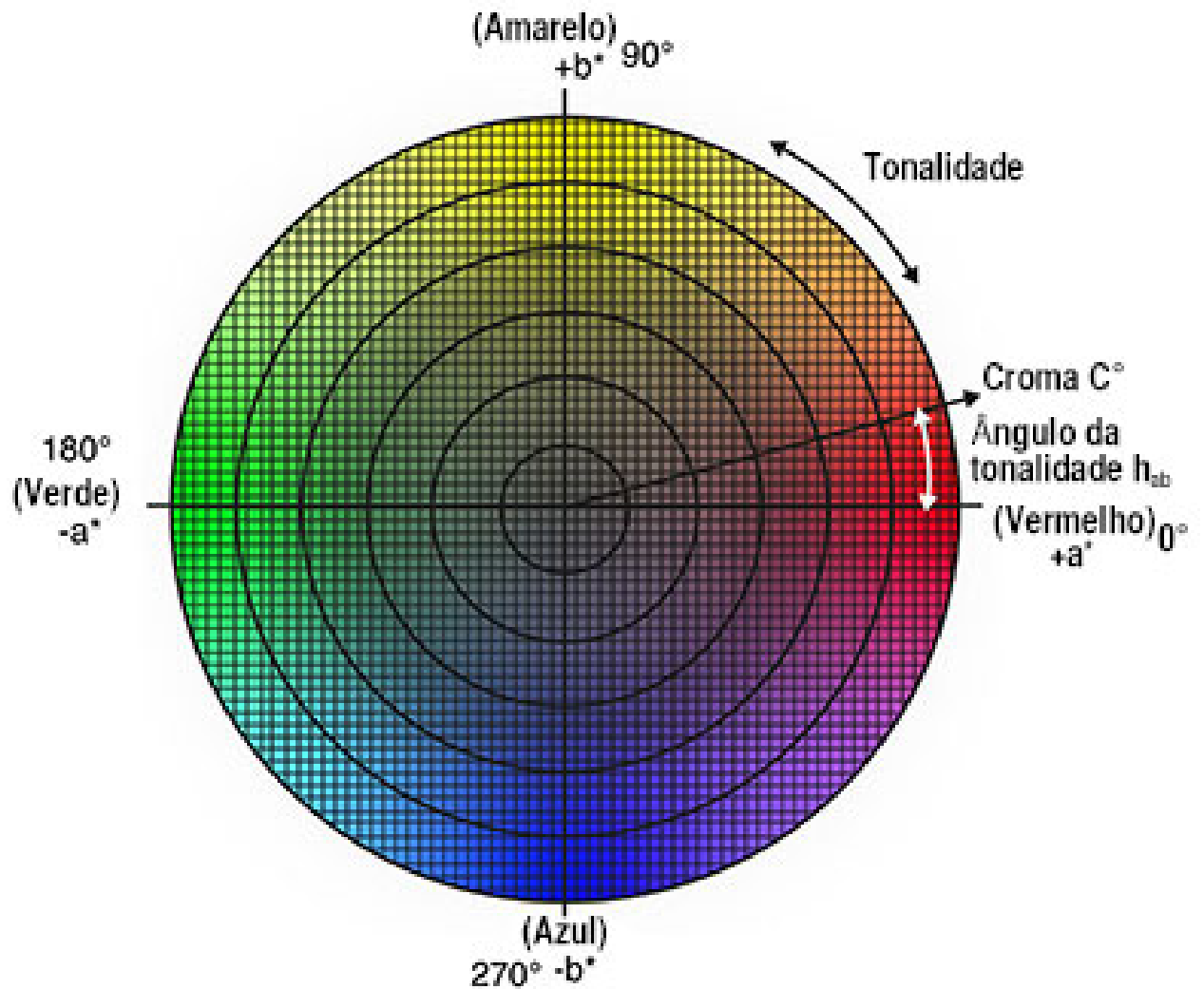
SOARES, M. dos S. Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software. **Universidade Presidente Antônio Carlos - UNIPAC**, 2004. Disponível em: <<http://periodicosibepes.org.br/index.php/reinfo/article/view/146/38>>.

TEIXEIRA, L. V. Análise sensorial na indústria de alimentos. **Revista do Instituto de Laticínios Cândido Tostes**, v. 64, n. 366, 2009. Disponível em: <<https://revistadoilct.com.br/rilct/article/view/70>>.

### APÊNDICE A – Espaço de cor CIE L\*C\*h°

É o espaço de cor utilizado no processo de análise sensorial das características relacionadas à coloração da fruta, onde L\* indica luminosidade, C\* a saturação e h° o ângulo de tonalidade (Compreendendo... , 2020).

Figura 1 – Representação do Espaço de cor CIE L\*C\*h°



Fonte: Konica Minolta Sensing Americas Incorporation, 2020.