



**Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"**

**AFONSO HENRIQUE DOS SANTOS SILVA**

**DESENVOLVIMENTO DE APLICAÇÕES HÍBRIDAS UTILIZANDO AS  
TECNOLOGIAS ANGULAR 2 E IONIC 2**

**Assis/SP  
2017**



Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"

**AFONSO HENRIQUE DOS SANTOS SILVA**

**DESENVOLVIMENTO DE APLICAÇÕES HÍBRIDAS UTILIZANDO AS  
TECNOLOGIAS ANGULAR 2 E IONIC 2**

Trabalho de Conclusão de curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Análise e Desenvolvimento de Sistemas.

**Orientando:** Afonso Henrique dos Santos Silva

**Orientador:** Dr. Almir Rogério Camolesi

**Assis/SP  
2017**

FICHA CATALOGRÁFICA

S586d SILVA, Afonso Henrique dos Santos  
Desenvolvimento de aplicações híbridas utilizando as tecnologias  
angular 2 e ionic 2 / Afonso Henrique dos Santos Silva. – Assis,2017.

49p.

Trabalho de conclusão do curso (Análise e Desenvolvimento de  
Sistemas). – Fundação Educacional do Município de Assis-FEMA

Orientador: Dr. Almir Rogério Camolesi

1.Angular 2 2.Ionic 2 3.Javascript

CDD 005.133

# DESENVOLVIMENTO DE APLICAÇÕES HÍBRIDAS UTILIZANDO AS TECNOLOGIAS ANGULAR 2 E IONIC 2

**AFONSO HENRIQUE DOS SANTOS SILVA**

Trabalho de conclusão de curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

**Orientador:** Dr. Almir Rogério Camolesi.

**Avaliador:** Ma. Diomara Martins Reigato Barros.

## DEDICATÓRIA

Dedico esse trabalho à minha família, e em especial aos meus pais, pois eles me deram toda a estrutura, sabedoria de vida e todo o apoio para a realização deste trabalho e por fim, a todos os amigos e colegas que apoiam com energias positivas.

## **AGRADECIMENTOS**

Primeiramente agradeço a Deus por todas as bênçãos e energias positivas que recebi. Bênçãos que foram muito mais do que pedi ou esperava receber. Deus que sempre está em nossos corações nos dando forças, alegrias, bondade em vida, e continua proporcionando paz, saúde, amor e coragem. A Deus toda glória, honra e louvor!

Ao Drº Prof. Almir Rogério Camolesi, meu orientador, pela sua fundamental contribuição no desenvolvimento do trabalho desde o início; pela paciência; e pelos ensinamentos repassados, não somente acadêmicos, mas ensinamentos de vida. Sem dúvidas é um grande professor e uma pessoa iluminada por Deus.

A minha família que sempre me apoiou e me proporcionou bons conselhos que me ajudaram a realizar este trabalho.

Aos meus amigos e colegas que ajudaram com energias positivas, e companheirismo nesta caminhada para realização deste trabalho.

Comece fazendo o que é necessário, depois o que é possível, e de repente você estará fazendo o impossível.

São Francisco de Assis

## RESUMO

A tecnologia presente nos dispositivos móveis a cada dia vem com o objetivo de mudar de vez a forma que as pessoas interagem entre si e com ao redor do mundo. Isso tudo graças as infinitas possibilidades que esta tecnologia presente possa fazer, devido a sua usabilidade, praticidade e fluidez com relação as atividades do cotidiano tornando indispensável a utilização dos aplicativos para facilitar a vida das pessoas. Plataformas dos dispositivos móveis como iOS, Android e Windows Phone trazem para os desenvolvedores um novo caminho para atender todas as tecnologias como o intuito de fomentar o mercado das tecnologias móveis como por exemplo *smartphones*, *Tablets*, consoles portáteis e entre outros.

O presente trabalho tem como proposta apresentar análise e estudo de caso sobre o desenvolvimento de um aplicativo móvel multiplataforma, utilizando os frameworks *Angular 2* e *Ionic 2* que se baseiam em tecnologias web como: HTML, CSS, *JavaScript* e *TypeScript*. Para o estudo de caso será apresentado o aplicativo para Gestão de Validade de data de produtos alimentícios na qual tem como objetivo de evitar possíveis desperdícios de alimentos em residências.

**Palavras-chave:** *Angular 2; Ionic 2; TypeScript; JavaScript*

## **ABSTRACT**

The technology present in everyday mobile devices comes with the goal of changing the way people interact with each other and around the world. This is all thanks to the infinite possibilities that this present technology can do, due to its usability, practicality and fluidity in relation to the daily activities, making it indispensable to use the applications to make life easier for people. Platforms of mobile devices such as iOS, Android and Windows Phone bring developers a new way to meet all technologies in order to foster the market of mobile technologies such as smartphones, tablets, portable consoles and others.

The present work has as proposal to present analysis and case study on the Development of a cross-platform mobile application, using the frameworks Angular 2 and Ionic 2 that are based on web technologies like: HTML, CSS, JavaScript and TypeScript. For the case study will be presented the application for Validity Management of food products in which it aims to avoid possible waste of food in homes.

**Keywords:** *Angular 2; Ionic 2; TypeScript; JavaScript*

## LISTA DE ILUSTRAÇÕES

Figura 1 Tecnologias utilizadas no projeto.....	23
Figura 2 Estrutura de uma árvore DOM.....	25
Figura 3 Exemplo de Injeção de dependência.....	26
Figura 4 - Event-Loop do Node.js.....	28
Figura 5 Repositório online do Node.js.....	29
Figura 6 - Mapa mental.....	35
Figura 7 - Caso de Uso.....	37
Figura 8 - Diagrama de Atividade de consultar produtos.....	38
Figura 9 - Diagrama de Atividade de cadastrar usuários.....	38
Figura 10 - Diagrama de Sequência – Cadastrar Usuário.....	39
Figura 11 - Diagrama de Sequência – Consultar Produto.....	39
Figura 12- Diagrama de Entidade e relacionamento.....	40
Figura 13 Diagrama de Classe.....	41
Figura 14 - Estrutura Analítica do Projeto.....	42
Figura 15 - Cronograma.....	<b>Erro! Indicador não definido.</b>
Figura 16 - Arquitetura da solução.....	44
Figura 17 Utilização da API StrongLoopback.....	45
Figura 18 VisualStudio Code.....	46
Figura 19 Interface da aplicação.....	47

## LISTA DE ABREVIATURAS E SIGLAS

**API** - Application Programming Interface

**CSS** - Cascading Style Sheets

**DOM** – Document Object Model

**HTML** - Hypertext Markup Language

**IDE** - Integrated Development Environment

**JS** - JavaScript

**JSON** - JavaScript Object Notation

**SOAP** - Simple Object Access Protocol

**UML** - Unified Modeling Language

**WEB** – World Wide Web

**W3C** - World Wide Web Consortium

**XML** - Extensible Markup Language

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>16</b>
1.1. OBJETIVOS.....	17
1.2. PÚBLICO-ALVO .....	18
1.3. JUSTIFICATIVA .....	18
1.4. ESTRUTURA DO TRABALHO.....	19
<b>2. DESENVOLVIMENTO DO PROJETO .....</b>	<b>20</b>
2.1. UML .....	20
2.1.1.Diagrama de Caso de Uso.....	21
2.1.2.Diagrama de Classes.....	21
2.1.3.Diagrama de Atividade .....	21
2.1.4.Diagrama de Sequência .....	22
2.1.5.Diagrama de Entidade e Relacionamento.....	22
2.2. DESENVOLVIMENTO HIBRIDO.....	23
2.2.1.Angular 2 .....	24
2.2.2.JavaScript.....	26
2.2.3.TypeScript .....	27
2.2.4.Node.js.....	27
2.2.5.Ionic 2 .....	30
2.2.6.Apache Cordova .....	31
2.2.7.Bootstrap.....	32
2.2.8.Visual Studio Code .....	33
2.2.9.Webservice.....	33
<b>3. ANÁLISE, ESPECIFICAÇÃO DO SISTEMA E PROJETO DO ESTUDO DE CASO .....</b>	<b>35</b>
3.1. ANÁLISE E ESPECIFICAÇÃO DO SISTEMA.....	35
3.1.1.Mapa Mental .....	35
3.1.2.Lista de Eventos .....	36
3.1.3.Caso de Uso .....	37
3.1.4.Diagrama de Atividade .....	38
3.1.5.Diagrama de Sequência .....	39
3.1.6.Diagrama de Entidade e Relacionamento.....	40

<b>3.1.7. Diagrama de Classe</b> .....	<b>41</b>
<b>3.2 ESTRUTURA DO PROJETO</b> .....	<b>42</b>
<b>3.2.1. Estrutura Analítica do Projeto (EAP)</b> .....	<b>42</b>
<b>3.2.2. Cronograma</b> .....	Erro! Indicador não definido.
<b>3.2.3. Especificação de custos</b> .....	<b>43</b>
<b>4. IMPLEMENTAÇÃO DA SOLUÇÃO</b> .....	<b>44</b>
4.1. BASES DE DADOS .....	44
4.2. WEB SERVICE .....	45
4.3. APLICAÇÃO HÍBRIDA .....	46
<b>5. CONCLUSÃO</b> .....	<b>48</b>
5.1. TRABALHOS FUTUROS .....	48
<b>6. REFERÊNCIAS</b> .....	<b>49</b>

# 1. INTRODUÇÃO

A tecnologia vem avançando a cada dia, trazendo várias possibilidades, e cada vez mais os dispositivos estão tendenciados a ser portáteis ou móveis. O crescente avanço das tecnologias dos dispositivos móveis, incluindo Smartphones e Tablets<sup>1</sup>, proporcionaram para as pessoas um importante recurso indispensável para realizarem suas tarefas do cotidiano, graças aos seus vastos recursos e funções.

A tecnologia móvel não só está presente na vida dos usuários comuns, mas também no domínio do mercado de negócios, disponibilizando para as empresas uma acessibilidade maior e uma rapidez na forma de trabalhar.

Através da tecnologia em nuvem<sup>2</sup>, os dispositivos móveis ganharam força no que se diz respeito aos benefícios da computação em nuvem, garantindo acessibilidade rápida, custo, velocidade, produtividade, confiabilidade e desempenho no acesso a informações de qualquer lugar que estejam. Essas vantagens foram primordiais para caírem no gosto dos desenvolvedores de software que investem suas fichas em aplicativos web.

Existem dois tipos de aplicações móveis, os Nativos que inclui um processo de desenvolvimento para uma plataforma específica, utilizando todo o potencial e funcional da plataforma para a qual ele foi desenvolvido. Como exemplo de um aplicativo nativo podemos citar o sistema operacional iOS<sup>3</sup>, pois precisaria utilizar um Mac, usaria o XCode<sup>4</sup> como IDE e desenvolveria em Swift<sup>5</sup> ou Objective C. Entretanto um dispositivo que tenha como sistema operacional o Android<sup>6</sup>, poderíamos utilizar de uma forma mais flexível o computador, mas provavelmente utilizaríamos Java<sup>7</sup> como linguagem de programação e o Android Studio<sup>8</sup> como IDE.

Os aplicativos Híbridos<sup>9</sup> são parcialmente nativos e como os nativos, eles devem ser baixados através de um aplicativo de loja (como Google Play do Android , App Store da Apple e Windows Store do Windows Phone), ficam armazenados na tela principal do dispositivo e podem aproveitar todos os recursos do dispositivo (GPS, câmera,

---

<sup>1</sup> Disponível em < [http://bdm.unb.br/bitstream/10483/9405/1/2014\\_GustavoLeuzingerCoutinho.pdf](http://bdm.unb.br/bitstream/10483/9405/1/2014_GustavoLeuzingerCoutinho.pdf) >

<sup>2</sup> Disponível em < <https://azure.microsoft.com/pt-br/overview/what-is-cloud-computing/> >

<sup>3</sup> Disponível em < <https://www.apple.com/br/ios/ios-10/> >

<sup>4</sup> Disponível em < <https://developer.apple.com/xcode/> >

<sup>5</sup> Disponível em < <https://developer.apple.com/swift/> >

<sup>6</sup> Disponível em < [https://www.android.com/intl/pt-BR\\_br/](https://www.android.com/intl/pt-BR_br/) >

<sup>7</sup> Disponível em < [https://www.java.com/pt\\_BR/download/faq/whatis\\_java.xml](https://www.java.com/pt_BR/download/faq/whatis_java.xml) >

<sup>8</sup> Disponível em <http://www.androidpro.com.br/android-studio-configurando-ambiente/>

<sup>9</sup> Disponível em < <http://www.luisaambros.com/blog/diferenca-entre-aplicativos-nativos-hibridos-e-mobile-web-apps/> >

cronometro, gestos e música). Como web apps, eles podem ser baseados em HTML5 e exibidos através de um navegador embutido no aplicativo, tendo parte ou conteúdo total carregado da web.

Os aplicativos híbridos são populares porque permite desenvolvimento multiplataforma, utilizando o mesmo HTML para diferentes sistemas operacionais como através de tecnologias como Cordova, angular 2 e Ionic 2 e permitem, inclusive compilar para o formato nativo, reduzindo custos de produção.

As principais vantagens das aplicações móveis são:

- Maior eficiência;
- O acesso as principais funcionalidades do dispositivos e hardwares;
- Componentes projetados para a plataforma;
- Armazenamento local de dados;
- Maior desempenho.

## 1.1. OBJETIVOS

O objetivo principal deste trabalho é a análise e o estudo sobre as tecnologias de desenvolvimento de aplicações híbridas<sup>10</sup>, utilizando os *frameworks*, Ionic2 e Angular2.

Portanto, esse trabalho tem como intenção aplicar os conhecimentos adquiridos e análise para o desenvolvimento de um aplicativo híbrido de Gestão de validade de data de produtos alimentícios. O aplicativo tem como finalidade facilitar ao consumidor a visualização de produtos que estão em prazo de vencimento, proporcionando para o consumidor economia, praticidade e um consumo mais consciente em relação ao desperdício de alimentos. Além dos frameworks, serão apresentadas as linguagens de Script Web, tais como HTML5, CSS3, Java Script e TypeScript, utilizadas como bases tecnológicas para o desenvolvimento do aplicativo móvel híbrido.

---

<sup>10</sup> Disponível em < <http://www.devmedia.com.br/aplicacoes-moveis-nativas-ou-web/30392>>

O aplicativo fará a integração com outro sistema, por meio de serviços Web (Web service), que independentemente do sistema operacional, irá ser visualizado pelo usuário.

## 1.2. PÚBLICO-ALVO

O aplicativo híbrido foi desenvolvido para os consumidores que desejam controlar vencimentos dos produtos alimentícios em suas residências, proporcionando praticidade, economia e um consumo mais consciente.

Os conhecimentos apresentados poderão ajudar e contribuir com futuras pesquisas, para aqueles que desejam aprender sobre o desenvolvimento híbrido.

## 1.3. JUSTIFICATIVA

A justificativa para a abordagem do estudo de caso do desenvolvimento do aplicativo Híbrido, se baseia na ideia de utilizar no desenvolvimento as tecnologias recentes no caso os frameworks, Angular 2 e Ionic 2. Tais tecnologias que estão em seu estado inicial de evolução tecnológica. Além disso, para o desenvolvimento do aplicativo foi feita uma pesquisa sobre a integração de várias tecnologias como exemplo das linguagens Scripts<sup>11</sup> (Java Script, HTML, CSS, TypeScript) e outras tecnologias apresentadas ao decorrer do trabalho que terá como fator principal a comunicação entre si para o seu devido funcionamento.

O aplicativo que foi desenvolvido tem como foco consumidores que acabam não obtendo controle na validade dos alimentos que estão na despensa ou na geladeira, evitando assim, possíveis desperdícios que acabam afetando no bolso dos consumidores. Então, tendo em vista este problema, houve a necessidade de criação

---

<sup>11</sup> Disponível em < <https://www.tecmundo.com.br/programacao/1185-o-que-e-script-.htm>>

de um aplicativo que proporcione um controle desses vencimentos, que irá facilitar a identificação de possíveis alimentos que estejam com curto prazo de vencimento.

#### 1.4. ESTRUTURA DO TRABALHO

Este trabalho está dividido em capítulos. O primeiro capítulo apresenta introdução, objetivo, público alvo e justificativa para o desenvolvimento deste trabalho.

O segundo capítulo apresenta as tecnologias e ferramentas utilizadas para o desenvolvimento do trabalho.

O terceiro capítulo apresenta duas seções 3.1 destinada a análise e a especificação do sistema, o levantamento de requisitos, mapa mental, lista de eventos, e a modelagem através da linguagem de modelagem UML utilizada para realizar tal tarefa. Na 3.2 seção destina-se a estrutura de desenvolvimento do sistema, onde são descritas a estrutura analítica do projeto, especificação de custos, e cronograma do trabalho.

No quarto capítulo apresenta a implementação do sistema, onde são apresentadas o projeto em camadas e a interface do sistema.

No quinto e último capítulo apresenta a conclusão e trabalhos futuros.

## 2. DESENVOLVIMENTO DO PROJETO

No desenvolvimento do aplicativo híbrido foi utilizado o editor de desenvolvimento Visual Studio Code proporcionando a utilização dos frameworks Angular 2 e Ionic 2. A programação foi desenvolvida com a linguagem TypeScript baseada em Java Script que é uma das linguagens mais utilizadas para o desenvolvimento de aplicativos Web Mobile. O layout foi desenvolvido com base em HTML5 e CSS3. Para a integração do aplicativo com o sistema de controle principal foi criada uma aplicação Node.js Web Service<sup>12</sup>(Serviços de Web) e para a análise e a diagramação do aplicativo a tecnologia UML.

### 2.1. UML

UML (Unified Modeling Language) é uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos, cujo o objetivo é auxiliar na definição das características do sistema, tais como seus requisitos, seu comportamento, estrutura lógica, a dinâmica de seus processos até mesmo suas necessidades físicas em relação ao equipamento sobre a qual o sistema deve ser implantado (GUEDES,2011, p.19).

A seguir a definição dos diagramas UML, que são: Caso de Uso, Atividade, Sequência, Classe e Modelo de Diagrama Entidade Relacionamento.

---

<sup>12</sup> Disponível em < <http://www.opensoft.pt/web-service/>>

### **2.1.1. Diagrama de Caso de Uso**

O Diagrama de Caso de Uso é utilizado normalmente nas fases de análise e levantamento de requisitos do sistema, embora venha ser consultado durante todo o processo de modelagem e possa servir de base para outros diagramas. Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema irá se comportar. Procura identificar os atores (usuários, outros sistemas ou até mesmo algum hardware especial) que utilizarão de alguma forma o software, bem como os serviços, ou seja as funcionalidades que o sistema disponibilizará aos atores, conhecidos como caso de uso. (GUEDES,2011, p.30).

### **2.1.2. Diagrama de Classes**

O Diagrama de Classes, define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si. É considerado um dos mais importantes da UML e serve de apoio para a maioria dos demais diagramas. (GUEDES,2011, p.31).

### **2.1.3. Diagrama de Atividade**

O Diagrama de Atividade descreve os passos a serem percorridos para a conclusão de uma atividade específica, podendo esta ser representada por um método com certo grau de complexidade, um algoritmo, ou mesmo por um processo completo. Concentra-se na representação do fluxo de controle de uma atividade. (GUEDES,2011, p.36).

#### **2.1.4. Diagrama de Sequência**

O Diagrama de Sequência é um diagrama comportamental que se preocupa com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo. Costuma identificar o evento gerador do processo modelado, bem como o autor responsável por esse evento, e determina como o processo deve se desenrolar a ser concluído por meio da chamada de métodos disparados por mensagens enviadas entre os objetos. (GUEDES,2011, p.36).

#### **2.1.5. Diagrama de Entidade e Relacionamento**

Diagrama entidade relacionamento (DER)<sup>13</sup> é um modelo diagramático, que descreve o modelo de dados de um sistema com alto nível de abstração, que resulta do processo de modelagem executado pelo projetista de dados que conhece o MER e é a principal representação gráfica do Modelo de Entidades e Relacionamentos.

O modelo de entidade e relacionamento (MER) é usado para representar o modelo conceitual do negócio e de alto nível, ou seja, é um conjunto de conceitos e elementos de modelagem que o projetista do banco de dados precisa conhecer.

---

<sup>13</sup> Disponível em < <http://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>>

## 2.2. DESENVOLVIMENTO HÍBRIDO

Nesta seção serão apresentadas as tecnologias que foram utilizados para desenvolvimento híbrido como os frameworks Angular 2, Ionic 2 as linguagens Scripts que são JavaScript, TypeScript, CSS, junto com a plataforma de desenvolvimento mobile Cordova, a utilização do editor de código Visual Studio Code e para o Web Service o Node.js. A figura 1, a seguir apresenta os logos que identificam visualmente as tecnologias utilizadas neste trabalho.



Figura 1 Tecnologias utilizadas no projeto

### 2.2.1. Angular 2

Angular 2 é um framework<sup>14</sup> JavaScript que permite desenvolver aplicações web e móvel, desenvolvido pela Google em 2014 e seu principal objetivo de facilitar o desenvolvimento de aplicativos que podem ser acessados por um navegador web, sob o padrão Modelo-Visão-Visão-Modelo para facilitar tanto no desenvolvimento quanto no teste dos aplicativos.

O Angular2 utiliza o conceito SPA<sup>15</sup> que é um modelo de desenvolvimento de aplicações Web e mobile, utilizada em empresas como Microsoft, Google e Facebook .Nesse tipo de aplicação, apenas uma parte dela muda e a outra permanece fixa, fazendo parte de um *template* que é uma camada de visão de um sistema desenvolvido sob o conceito de programação orientada a objetos. Isso traz melhoras para a experiência do usuário e tráfego de dados. As vantagens de utilizar Angular 2<sup>16</sup> na aplicação são:

- É Open-Source;
- Arquiteturas da aplicação em camadas bem definidas;
- Permite a criação modular e de componentes reutilizáveis;
- Facilita a etapa de testes de forma automatizada;
- É mantido pela Google;
- Pode ser utilizando no desenvolvimento junto com o TypeScript.

Angular 2 utiliza o CLI angular que é uma ferramenta de interface de linha de comando que pode criar um projeto, adiciona arquivos e executa uma variedade de tarefas de desenvolvimento contínuas, como teste, empacotamento e implementação.

---

<sup>14</sup> Disponível em < <http://blog.alura.com.br/angular-2-0-final-e-lancado/>>

<sup>15</sup> Disponível em < <http://andremenegassi.com.br/blog/index.php/spa-single-page-application-e-restrestful-o-que-e/>>

<sup>16</sup> Disponível em < <http://www.andrefelizardo.com.br/blog/o-que-e-angularjs/>> .

### 2.2.1.1. Web Components

O Angular 2 utiliza *Web Components* que normalmente se refere a uma coleção de quatro especificações relacionadas ao W3C<sup>17</sup> que são :

- *Custom Elements* : São ativadas pela extensão de HTML por meio de tags personalizadas;
- *HTML Imports* : Possibilita o empacotamento de vários recursos como HTML, CSS, JS e entre outros;
- *Template Element*: Permite a inclusão de HTML inerte em um documento;
- *Shadow DOM* : Permite o encapsulamento de DOM e CSS.

### 2.2.1.2. Document Object Model

O Document Object Model (DOM) é criado a partir de uma página HTML, é uma representação do código-fonte HTML ou XML organizado em uma estrutura de árvore chamada de árvore DOM. A figura 2, a seguir apresenta a estrutura de uma árvore em DOM.

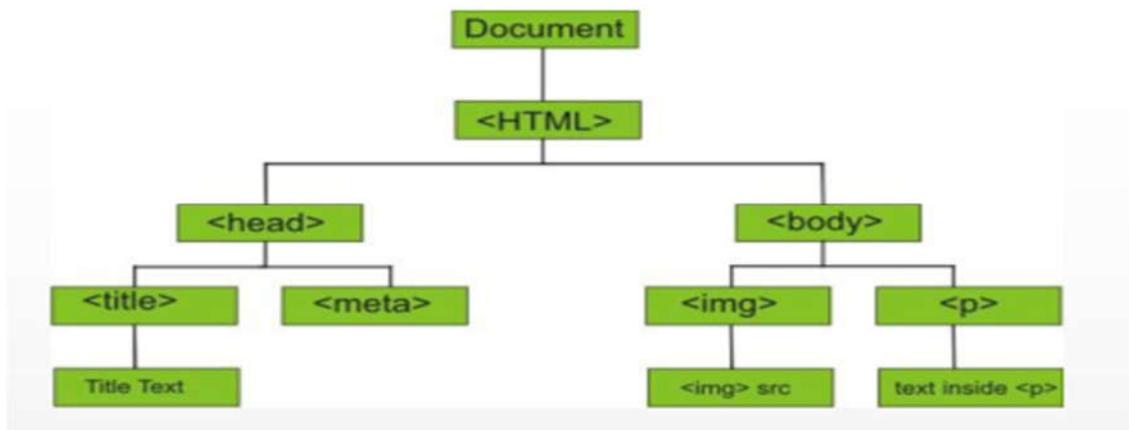


Figura 2 Estrutura de uma árvore DOM

<sup>17</sup> Disponível em < <http://www.w3c.br/Home/WebHome>>

### 2.2.1.3. Injeção de dependência

A injeção de dependência é um padrão de design de software que proporciona a redução de Componentes, mudando a maneira de como as dependências dos componentes são tratadas. Na figura 3 mostra a utilização de injeção de dependência no serviço da classe ProdutoService.

```
7  @Injectable()      // Injeção de dependência
8  export class ProdutoService {
9
10     private path = 'produtos';
11
12     constructor(private http: Http, private httpUtil: HttpUtilService) {
13     }
14
15     listarTodos(): Observable<Produto[]> {
16
17         return this.http.get(this.httpUtil.url(this.path), this.httpUtil.headers())
18             .map(this.httpUtil.extrairDados)
19             .catch(this.httpUtil.processarErros);
20     }
21 }
```

Figura 3 Exemplo de Injeção de dependência

## 2.2.2. JavaScript

O JavaScript<sup>18</sup> é uma linguagem de programação desenvolvida pela Netscape em parceria com a Sun Microsystems, criada com a objetivo de proporcionar interatividade e funcionalidade as páginas da Web.

A linguagem JavaScript proporcionou por dar origem a especificação técnica ECMAScript que é um padrão oficial reconhecido pela indústria mais conhecida pelo nome de JavaScript, e da versão desenvolvida pela Microsoft ter denominado o nome de JScript, na realidade, se denominam como implementações que respeitam a norma padrão ECMAScript que agregaram novas funcionalidades, porém, respeitando as especificações oficiais.

<sup>18</sup> Disponível em <<http://br.ccm.net/faq/2680-javascript-introducao-a-linguagem-javascript>>

O código escrito em JavaScript destina-se a ser executado pela web browser quando a página HTML que o contém é visualizada. O JavaScript é uma parte importante da página, e faz com que o navegador web seja capaz de tomar decisões com o comportamento do conteúdo é apresentado e como pode ser manipulado.

O JavaScript<sup>19</sup> tem como característica executar programas localmente, interagindo mais ao lado do cliente. Assim o JavaScript tem como objetivo fornecer as páginas web a possibilidade de programação, processamento e transformação de dados recebidos ou enviados, interagindo com a exibição e a marcação do conteúdo da linguagem HTML e com o estilo desse conteúdo proporcionado pelo CSS.

No navegador os Scripts de código escritos em JavaScript são executados permitindo por exemplo, atualizar parte do conteúdo de uma página web sem carregá-la totalmente após preencher um formulário, através de técnicas de programação como o AJAX<sup>20</sup>. Portanto o JavaScript permite o desenvolvimento de uma infinidade de sites completos e totalmente funcionais para diversas finalidades.

### 2.2.3. TypeScript

TypeScript<sup>21</sup> é uma linguagem para desenvolvimento JavaScript, criada pela Microsoft com o objetivo de escrever o código utilizando uma estrutura fortemente tipada além de ter o código compilado para o JavaScript. O TypeScript possibilita escrever o código em módulos, classes e interfaces para auxiliar na criação de componentes robustos. Com a linguagem TypeScript podemos escrever interfaces que nos permitem adquirir informações de componentes e bibliotecas JavaScript existentes. Por ser compilado para o JavaScript o TypeScript roda em qualquer navegador, no Node.js, ou em qualquer outro ambiente JavaScript compatível com ES3. Além disso, possibilita a checagem estática de tipos, na navegação baseada em símbolos, auto completar e refatoração automática de código.

---

<sup>19</sup> Disponível em <https://canaltech.com.br/materia/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/>

<sup>20</sup> Disponível em < <http://codigofonte.uol.com.br/artigos/o-que-e-o-ajax-e-como-ele-funciona>>

<sup>21</sup> Disponível em < <http://www.diullei.com/TypeScript-ptBR/>>

## 2.2.4. Node.js

O Node.js é uma plataforma altamente escalável e de baixo nível que permite criar servidores web para a execução remota de aplicações JavaScript. Com o Node.js o desenvolvedor irá programar diretamente com diversos protocolos de rede e internet, ou utilizar bibliotecas que acessam diversos recursos do sistema operacional.

Node.js é orientado a eventos e uma das características principais é a utilização do *Event-Loop* que é responsável por escutar e emitir eventos, ou seja, se colocarmos em execução praticamente é um loop infinito que a cada iteração, é verificado na fila de eventos se um determinado evento for disparado e assim que for disparado o Event-Loop executa e envia para fila de executados, quando um evento está sendo executado podemos implementar qualquer lógica dentro do evento, proporcionado pelo mecanismo *Callback* que é uma função utilizada em JavaScript.



**OS EVENTOS SÃO PROCESSADOS UM POR VEZ**

Figura 4 - Event-Loop do Node.js

O Node.js soluciona o problema do aplicativo web para suportar mais usuários, ou seja, resolve esta questão mudando a maneira como a conexão é tratada no servidor, possibilitando que o mesmo o execute, podendo então atender um grande número de

conexões simultâneas. Serviços importantes como PayPal, Netflix, Walmart, Uber e LinkedIn utilizam o Node.js para seu funcionamento.

#### 2.2.4.1. NPM

O NPM (Node Package Manager) ou Gerenciador de pacotes do Node é um repositório online para publicação de projetos de código aberto para o Node.js e um utilitário de linha de comando que interage com este repositório online, que por sua vez auxilia na instalação de pacotes, visualização de versão e gerenciamento de dependências. A figura 5 mostra o repositório online do Node.js.

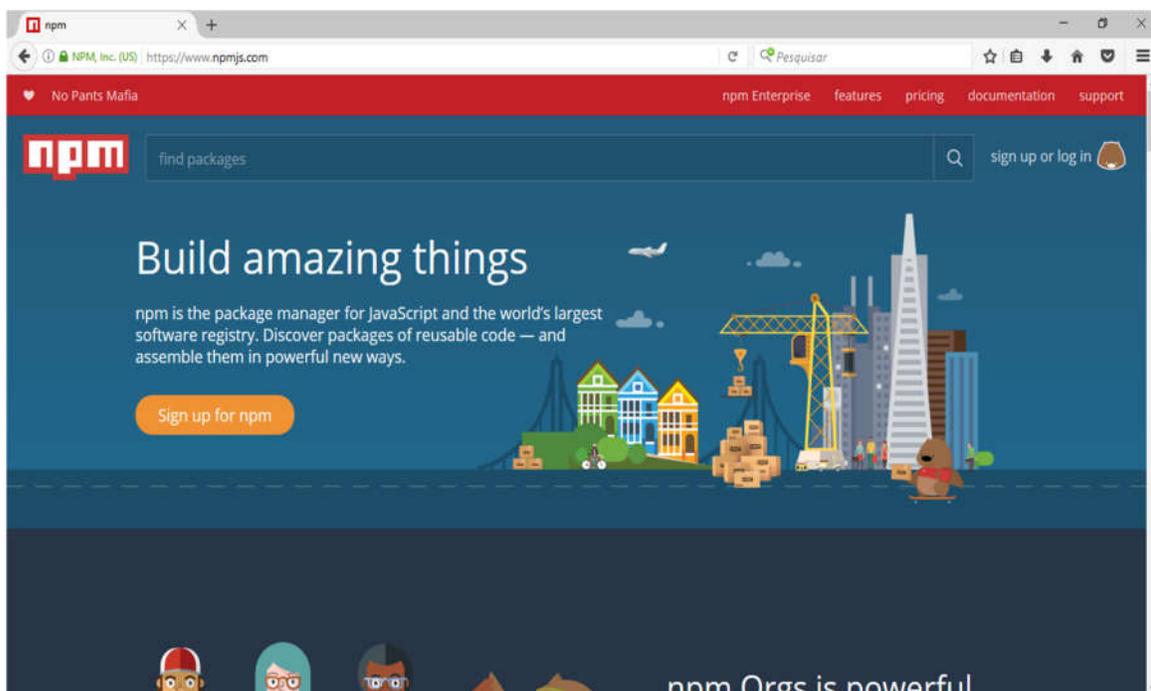


Figura 5 Repositório online do Node.js

## 2.2.5. Ionic 2

O Ionic 2<sup>22</sup> é um framework de código aberto que permite aos desenvolvedores a criação de aplicativos móveis de alta qualidade e desempenho, utilizando tecnologias de linguagens Web como HTML, CSS e JavaScript. Seu objetivo é focado principalmente a aparência de um aplicativo. Entretanto o Ionic consegue trabalhar em conjunto com essas tecnologias, a fim de simplificar uma grande parte do seu processo de desenvolvimento de aplicativos web.

Ionic atualmente exige a utilização do Angular, a fim de trabalhar em seu máximo potencial. No futuro o Ionic planeja se expandir, a fim de suportar uma ampla variedade de frameworks JavaScript (GOIS,2017).O framework Ionic 2 tem várias vantagens e características que são:

- As mudanças significativas entre o lançamento Angular e Angular 2 são aplicáveis ao Ionic 2;
- É desenvolvido em cima de Angular ou seja, o Ionic 2 foi baseado em Angular 2;
- Disponibilidade do Sass<sup>23</sup> para a personalização, rápida .

### 2.2.5.1. Componentes

Os componentes do Ionic são elementos de interface do usuário reutilizáveis que tem como objetivo servir como blocos de construção para o seu aplicativo para dispositivos móveis. Os componentes são compostos por HTML, CSS e JavaScript. Cada componente Ionic se adapta à plataforma em que seu aplicativo está sendo executado. Este processo é denominado como Continuidade de Plataforma e aprofundado em Theming (BOOTH,2017).

---

<sup>22</sup> Disponível em < <http://ionicframework.com/docs/v2/intro/concepts/>>

<sup>23</sup> Disponível em < <https://websocialdev.com/2013/11/introducao-ao-sass/>>

### 2.2.5.2. Theming

Os Temas (Theming) são conjuntos de estilos que são aplicados a um aplicativo. O Ionic utiliza o tema claro por padrão. A continuidade de plataforma do Ionic permite que os componentes tenham estilos específicos de plataforma ou seja os estilos do aplicativo serão alterados com base na plataforma em que for executado como por exemplo o iOS, Android, WindowsPhone (BOOTH,2017).

### 2.2.5.3. Ionic CLI

O Ionic CLI é a interface de linha de comando, uma ferramenta que fornece uma série de comandos úteis que facilitam para desenvolvedores Ionic.

## 2.2.6. Apache Cordova

O Apache Cordova é uma plataforma de desenvolvimento móvel com APIs, que tem como objetivo de permitir que o desenvolvedor acesse as funções nativas do dispositivo, como a câmera, GPS ou o acelerômetro.

No desenvolvimento com o Apache Cordova é executado com as tecnologias de programação como HTML, CSS, e JavaScript e tem a possibilidade de utilizar para o desenvolvimento os frameworks mobile como Bootstrap, JQuery mobile, Sencha Touch, Ionic e Lungo (WARGO, 2014).

O objetivo principal ao desenvolver com o Apache Cordova<sup>24</sup>, é o desenvolvimento de um aplicativo híbrido, onde o seu código pode ser compilado para diversas plataformas junto com todas as APIs para o acesso de funções do dispositivo que são instaladas na

---

<sup>24</sup> Disponível em < <https://cordova.apache.org/>>

própria aplicação. Alguns exemplos de plataformas que o aplicativo híbrido pode compilar:

- iOS;
- Android;
- Windows Phone;
- Firefox OS;
- Tizen;
- BlackBerry.

### 2.2.7. Bootstrap

Bootstrap<sup>25</sup> é um framework que tem como objetivo de simplificar o desenvolvimento de Web Site com a tecnologia móvel responsiva sem necessidade de utilizar o CSS para o desenvolvimento.

O Bootstrap possui uma diversidade de componentes em JavaScript que auxiliam aos Designers a implementação como por exemplo *Menu Dropdown, Modal, Carousel, Slide Show, Tooltip* entre outros.

Algumas características do bootstrap:

- Guia de aplicação;
- Totalmente gratuito;
- Possui uma interface amigável e moderna;
- Sistema responsivo;
- Integração com qualquer linguagem de programação;
- Possui uma vasta diversidade de *themas*.

---

<sup>25</sup> Disponível em < <http://getbootstrap.com.br/>>

### 2.2.8. Visual Studio Code

O Visual Studio Code é um editor de código-fonte com as principais ferramentas de desenvolvimento, produzida pela Microsoft com o principal objetivo em editar, criar e depurar códigos com mais facilidade. Possui suporte integrado para a conclusão do código IntelliSense<sup>26</sup> que é o termo geral para vários recursos como Listar Membros, informações do parâmetro, informação rápida e completar palavra. Esses recursos auxiliam a aprendizagem do código e manter o acompanhamento dos parâmetros que está sendo digitado e a adição de chamadas a métodos e propriedade pressionando apenas algumas teclas.

Visual Studio Code proporciona suporte incorporado e enriquecido para o desenvolvimento do Node.js com JavaScript, e TypeScript, também inclui ferramentas para tecnologias da Web com HTML, CSS, LESS, SASS e JSON, ótimo para criar aplicativos web e nuvem em uma vasta plataforma.

### 2.2.9. WebService

WebService é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes, ou seja, seu principal objetivo é a comunicação de aplicações através da internet. Esta aplicação é realizada com o foco em facilitar a EAI<sup>27</sup> que significa a integração das aplicações de uma empresa que abrange o conceito de interoperabilidade que é a capacidade de um sistema de se comunicar de forma transparente com o outro sistema.

Com a tecnologia WebService é possível determinar que novas aplicações possam interagir com aplicações que já existem e que sistemas desenvolvidos em plataformas diferentes possam ser compatíveis. Essa tecnologia abrange componentes que permitem aplicações enviar e receber dados em formato XML. Cada aplicação pode ter a sua própria linguagem que é traduzida para uma linguagem universal, o formato XML.

---

<sup>26</sup> Disponível em < <https://msdn.microsoft.com/pt-BR/library/hcw1s69b.aspx> > .

<sup>27</sup> Enterprise Application integration

Um dos fundamentos básicos para a construção de uma web service são os padrões XML e SOAP. No protocolo HTTP é realizado normalmente o transporte de dados e os dados são transferidos no formato XML, encapsulados pelo protocolo SOAP.

As tecnologias Web Services para as empresas podem trazer agilidade para os processos e eficiência na comunicação entre cadeias de produção ou de logística. As comunicações entre sistemas passam a ser dinâmicas e principalmente segura, pois não há intervenção humana.

### 3. ANÁLISE, ESPECIFICAÇÃO DO SISTEMA E PROJETO DO ESTUDO DE CASO

Para o desenvolvimento do sistema proposto, foi utilizado a tecnologia UML (Linguagem de Modelagem Unificada) para a análise e validação dos requisitos, onde toda a parte gráfica foi criada com base na ferramenta Astah<sup>28</sup>. Para o desenvolvimento DER (Diagrama Entidade Relacionamento) foi utilizado a ferramenta DBDesigner<sup>29</sup>. Este capítulo foi dividido em duas seções: A 3.1- análise e Especificação do sistema onde foi abordado o mapa mental, lista de eventos, e a modelagem através da linguagem de modelagem UML, e a 3.2-Estrutura do projeto foi apresentado a estrutura analítica do projeto, cronograma, e especificação de custos.

#### 3.1. ANÁLISE E ESPECIFICAÇÃO DO SISTEMA

##### 3.1.1. Mapa Mental

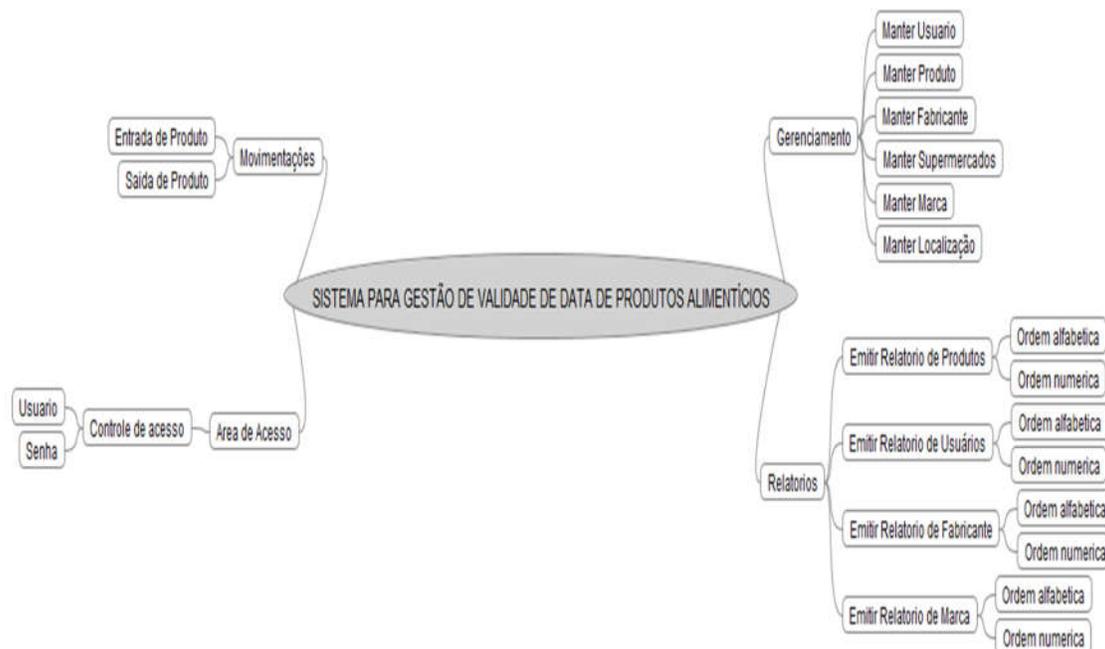


Figura 6 - Mapa mental

<sup>28</sup> Disponível em <http://www.astah.net>

<sup>29</sup> Disponível em < <http://www.devmedia.com.br/dbdesigner-modelagem-e-implementacao-de-banco-de-dados/30897>>

### **3.1.2. Lista de Eventos**

- 1.** Efetuar Cadastro;
- 2.** Efetuar Acesso;
- 3.** Manter Cadastro;
- 4.** Manter Produto;
- 5.** Efetuar Entrada do Produto;
- 6.** Efetuar Saída do Produto;
- 7.** Manter Fabricante;
- 8.** Manter Supermercado;
- 9.** Manter Localização;
- 10.** Manter Marca;
- 11.** Emitir Relatório de Produtos;
- 12.** Emitir Relatório de Fabricante;
- 13.** Emitir Relatório de Marca ;
- 14.** Emitir Relatório de Entrada;
- 15.** Emitir Relatório de Saída;
- 16.** Emitir Relatório de Localização;
- 17.** Emitir Relatório de Supermercado.

### 3.1.3. Caso de Uso



powered by Astah

Figura 7 - Caso de Uso

### 3.1.4. Diagrama de Atividade

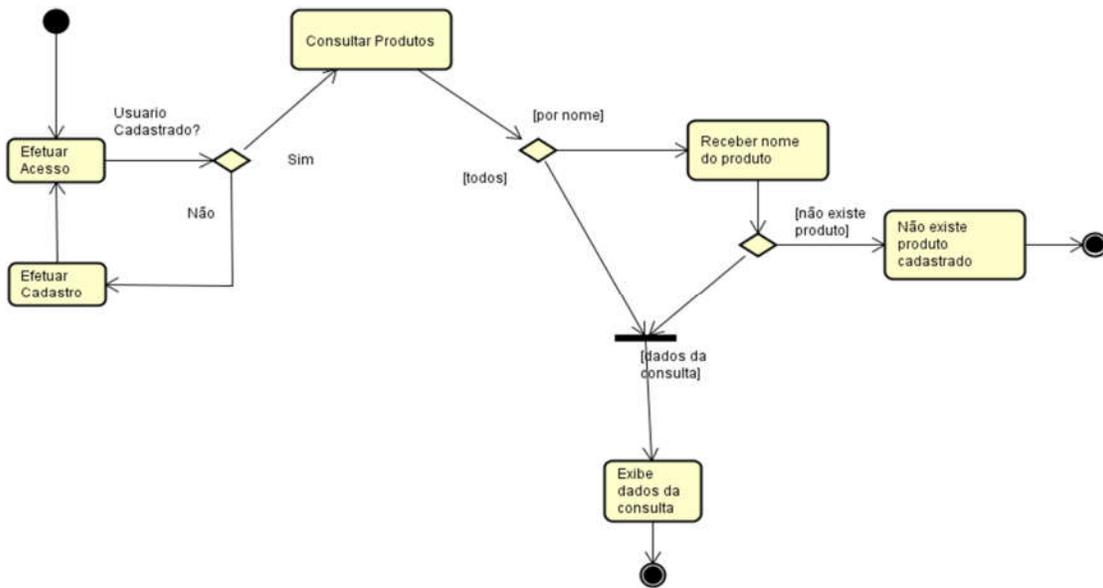
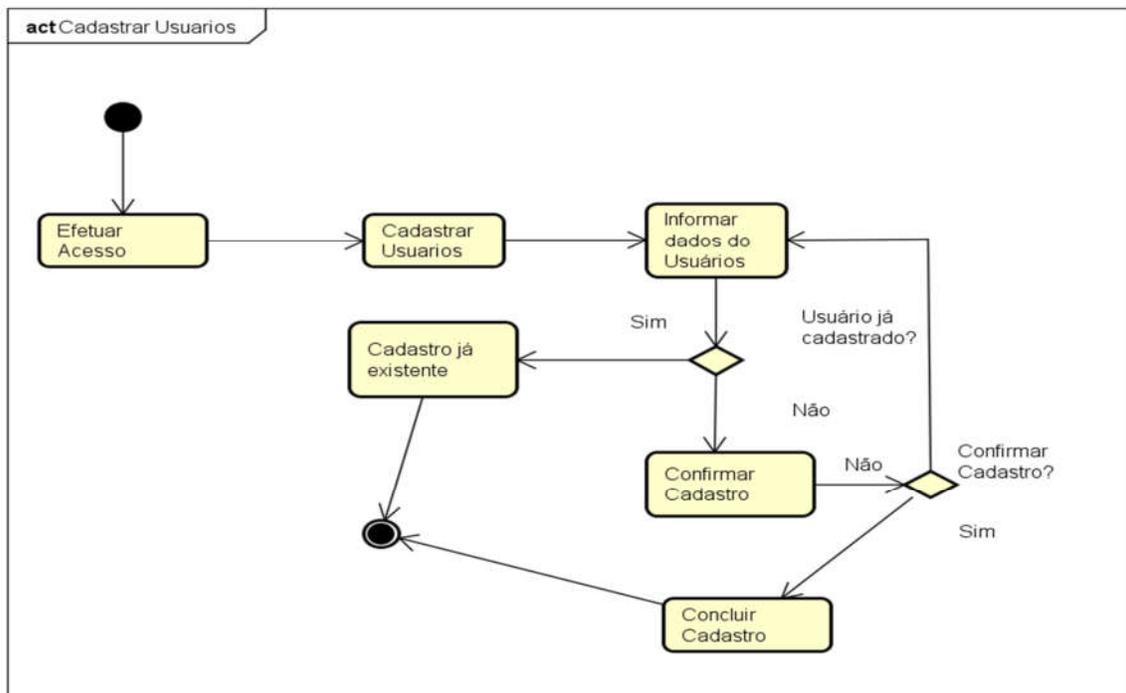


Figura 8 - Diagrama de Atividade de consultar produtos



powered by Astah

Figura 9 - Diagrama de Atividade de cadastrar usuários

### 3.1.5. Diagrama de Sequência

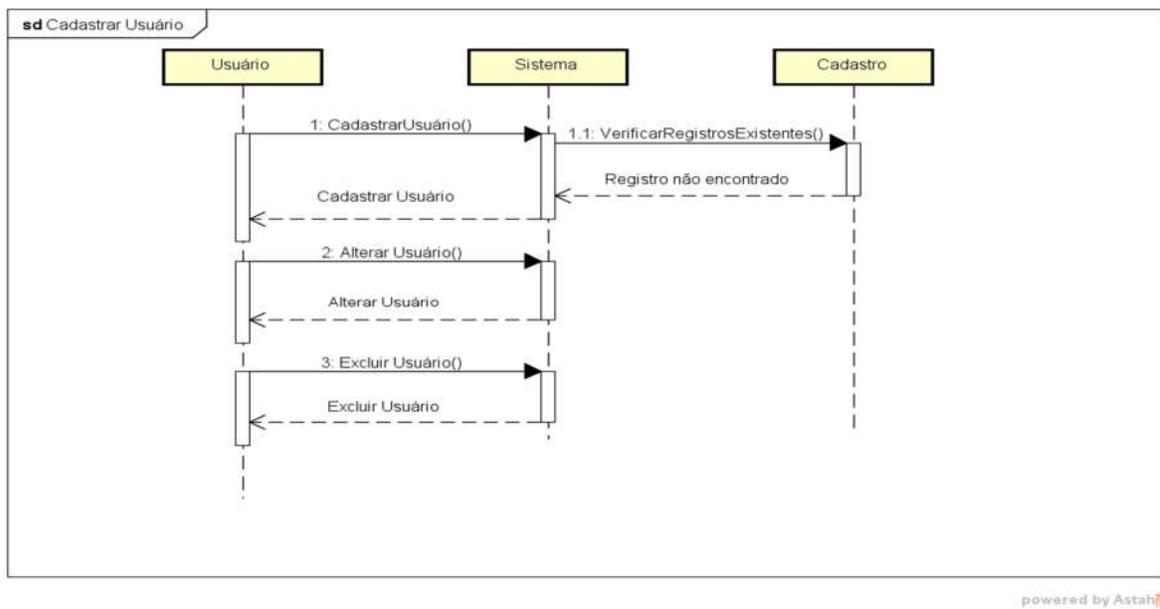


Figura 10 - Diagrama de Sequência – Cadastrar Usuário

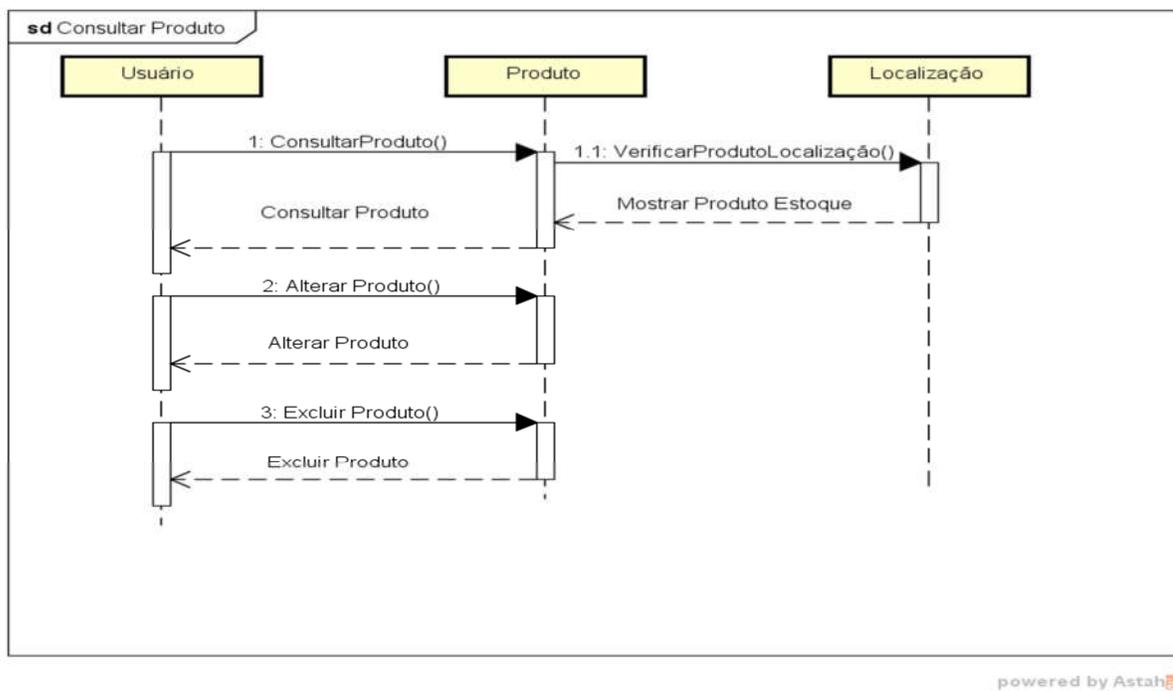


Figura 11 - Diagrama de Sequência – Consultar Produto

### 3.1.6. Diagrama de Entidade e Relacionamento

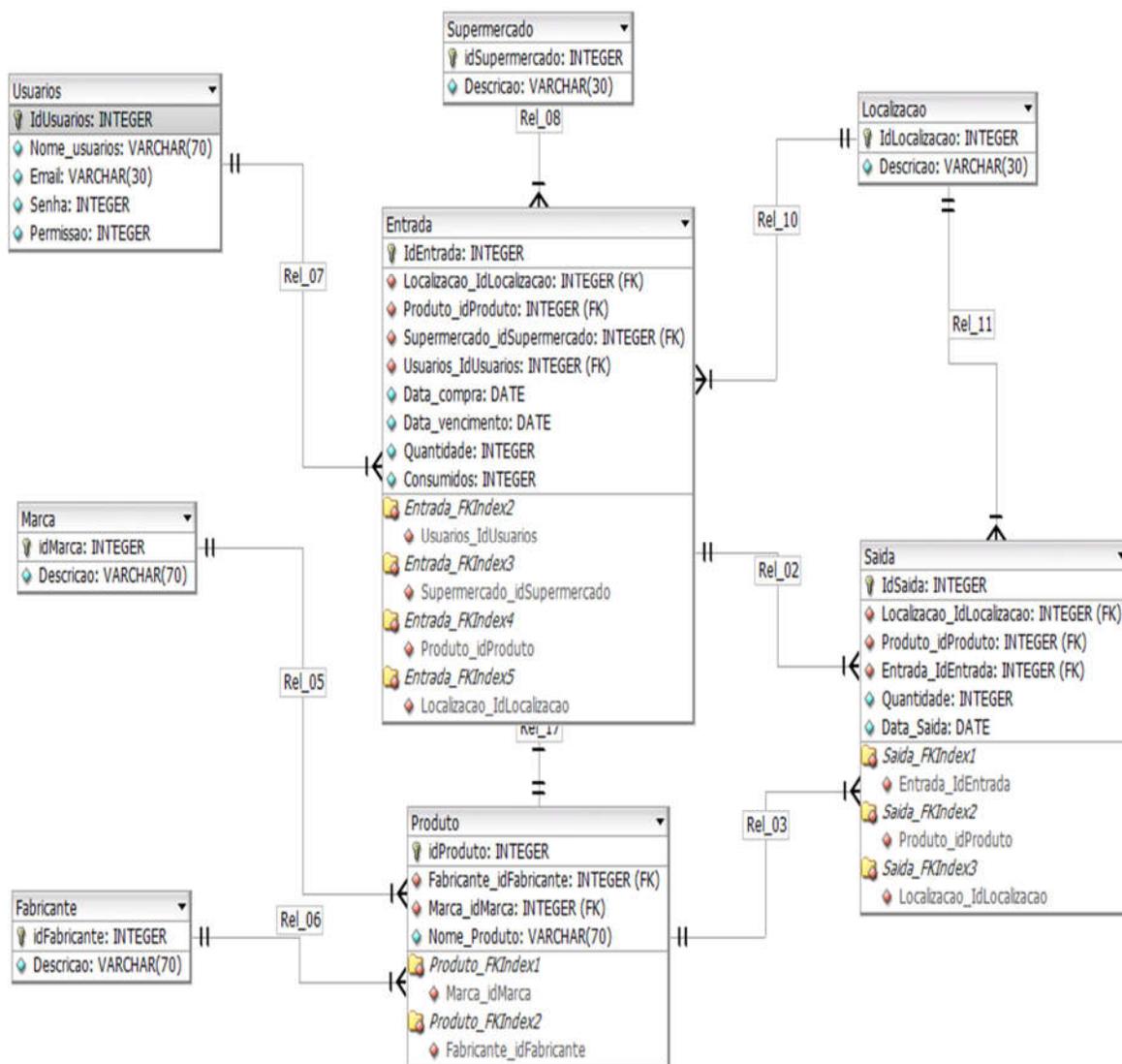
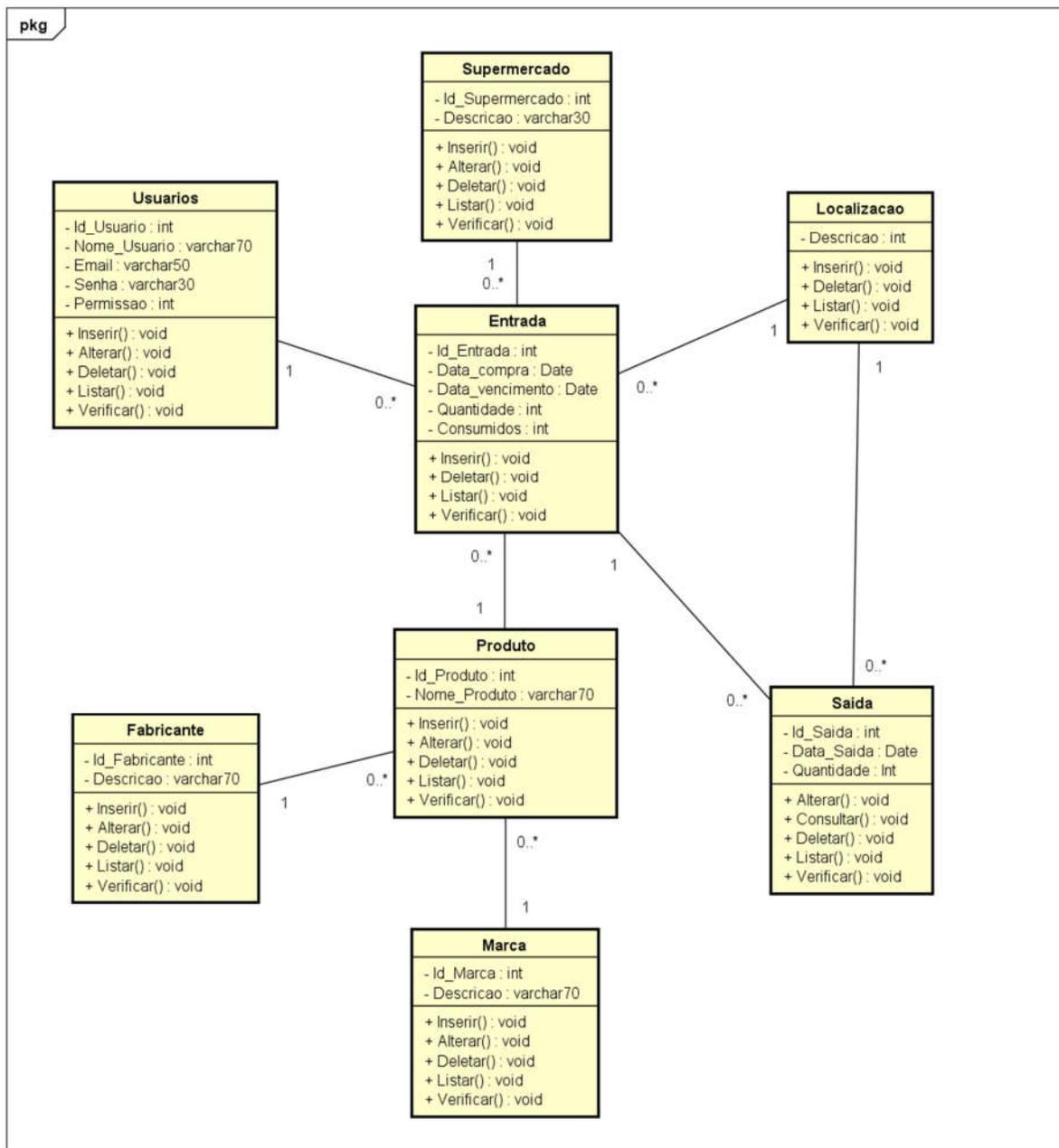


Figura 12- Diagrama de Entidade e relacionamento

## 3.1.7. Diagrama de Classe

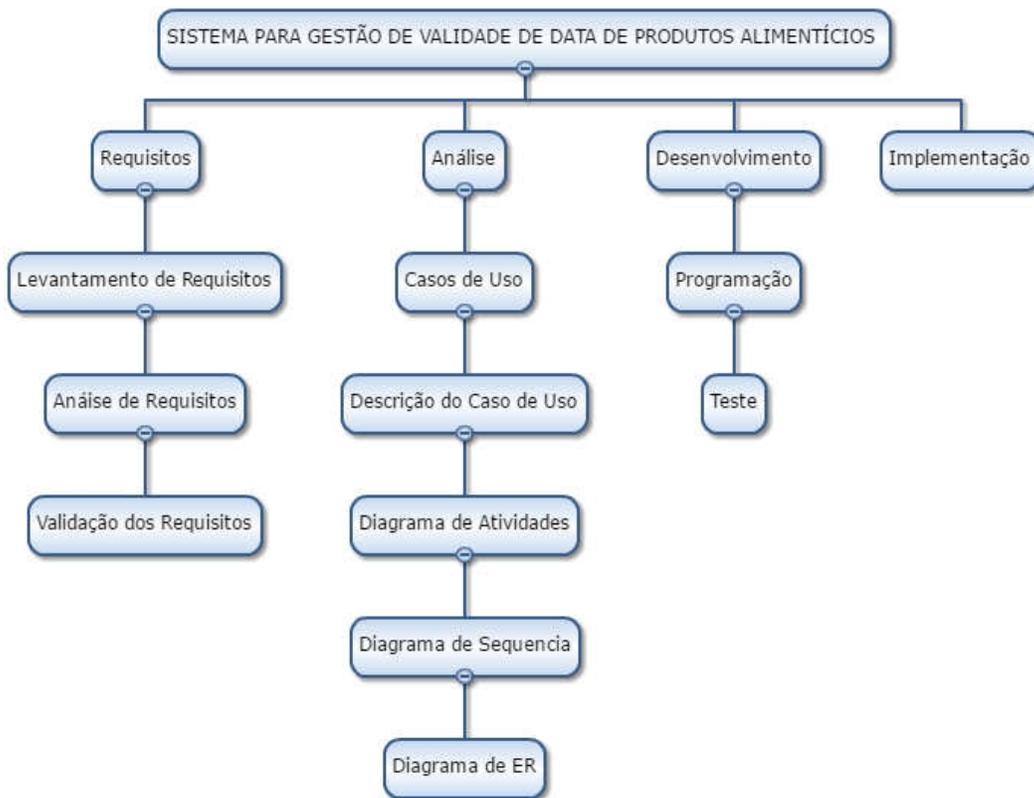


powered by Astah

Figura 13 Diagrama de Classe

## 3.2 ESTRUTURA DO PROJETO

### 3.2.1. Estrutura Analítica do Projeto (EAP)



www.wbstool.com

Figura 14 - Estrutura Analítica do Projeto

### 3.2.2. Especificação de custos

Programador:

Custo diário; R\$ 115,00.

Total de dias; 120.

Custo Total;  $(120 * 115,00) = R\$ 13.800,00$ .

Notebook:

Custo; R\$ 1.800,00.

Depreciação mensal;  $R\$ 1.800/12$  (meses) = R\$ 150,00 por mês.

Custo diário;  $R\$ 150,00 / 30$ (dias) = R\$ 5,00 por dia.

Custo de 120 dias;  $R\$ 5,00 * 120 = 600,00$ .

Orçamento Total:

Programador; R\$ 13.800,00.

Notebook; R\$ 600,00.

Total ;R\$ 14.400,00.

## 4. IMPLEMENTAÇÃO DA SOLUÇÃO

Para a implementação do aplicativo *GESTÃO DE VALIDADE DE DATA DE PRODUTOS ALIMENTÍCIOS* foi utilizada a seguinte arquitetura de software ilustrada na figura a seguir:

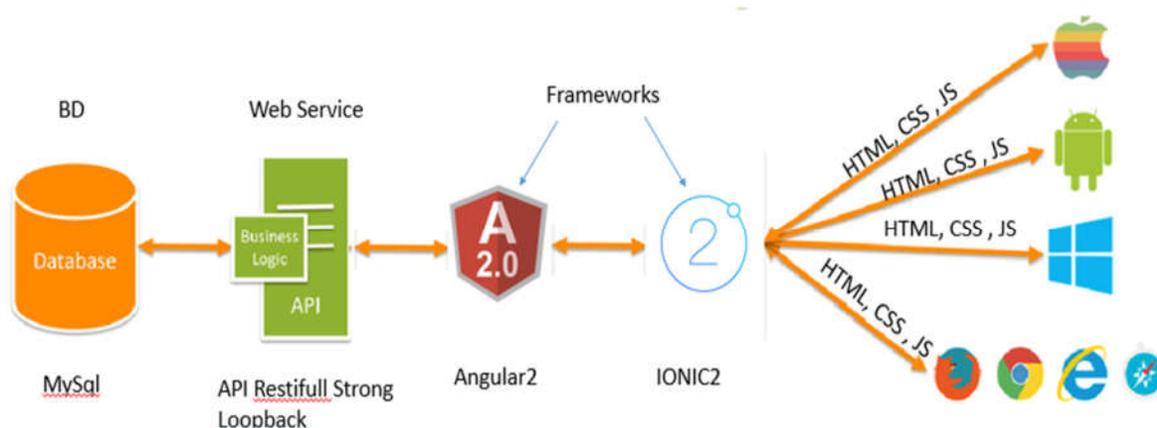


Figura 15 - Arquitetura da solução

Essa arquitetura contém todas as tecnologias envolvidas para o desenvolvimento da solução proposta, e para o estudo de caso sobre a integração de diferentes tecnologias com foco principal nos frameworks **Angular 2** e **Ionic 2**.

### 4.1. BASE DE DADOS

Para criar a base de dados foi utilizado o banco MySQL, um servidor e gerenciador de banco de dados (SGBD) relacional. Além de ser uma ferramenta gratuita criada na base da licença de software livre, proporciona qualidade, robustez, segurança e projetado inicialmente para trabalhar com aplicações de pequeno e médio porte, mas hoje atendendo às aplicações de grande porte e com mais vantagens do que seus concorrentes.

## 4.2. WEB SERVICE

Para criar um serviço de aplicação *restfull* foi utilizada o framework da API Node.js o *API Strong Loopback*<sup>30</sup> que é responsável por servir dados para a aplicação desenvolvida. A utilização da API favorece para que diversos tipos de aplicações consultem um mesmo servidor centralizado e focado apenas a trabalharem com os dados da aplicação. A API é quem vai enviar (POST) ou receber (GET) das informações, para o aplicativo de *Gestão de validade de data de produtos alimentícios*, através dos arquivos tipo JSON<sup>31</sup> que irá estruturar as informações de uma forma compacta do que os arquivos do tipo XML<sup>32</sup>. Na figura 17 mostra um exemplo de uma API StrongLoopback em execução.

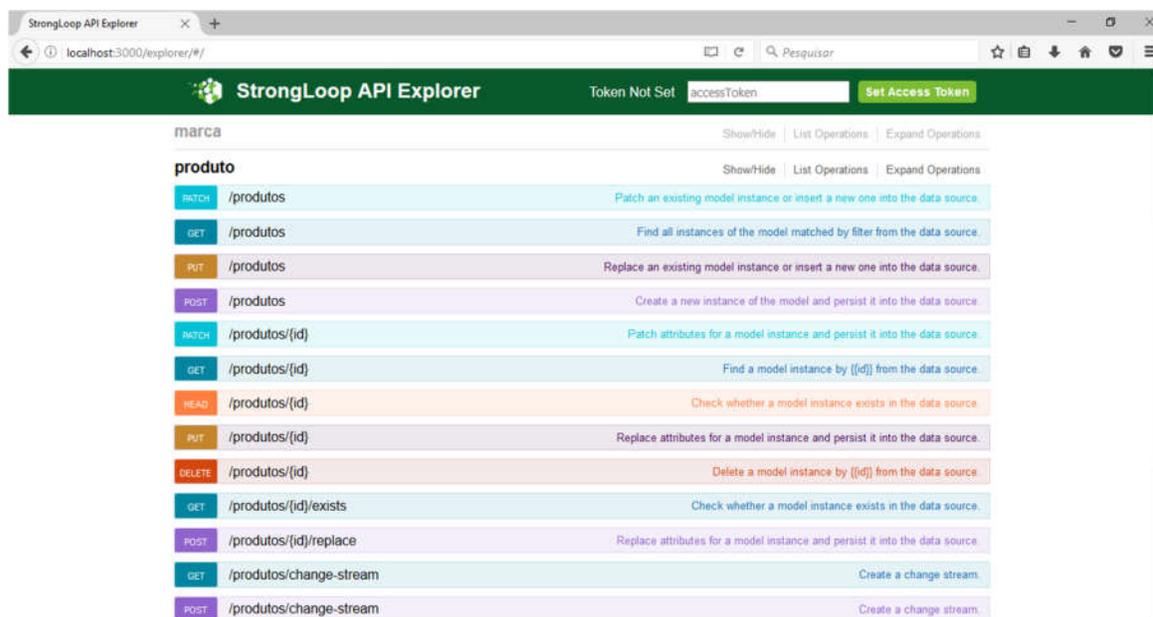


Figura 16 Utilização da API StrongLoopback

<sup>30</sup> Disponível em : < <https://loopback.io/> >

<sup>31</sup> Disponível em: <http://www.devmedia.com.br/introducao-ao-formato-json/25275>.

<sup>32</sup> Disponível em : <https://www.tecmundo.com.br/programacao/1762-o-que-e-xml-.htm>.

### 4.3. APLICAÇÃO HÍBRIDA

Para o desenvolvimento da aplicação híbrida foi utilizado a ferramenta Visual Studio Code, onde foi utilizado o framework Ionic 2 que por sua vez tem como base o Angular 2. Toda a aplicação foi desenvolvida com as linguagens Scripts como HTML, JS, TypeScript e CSS. Como padrão de desenvolvimento foi utilizado o padrão de camadas conhecido como MVVM (Model View ViewModel) ou Modelo Visão VisãoModelo.

Na camada Model são representadas as regras de negócio, que pode ser desde uma simples entidade de usuário até uma entidade complexa de ações comerciais.

A View é representada pela interface gráfica que tem como objetivo a apresentação do modelo de dados ao usuário, já na ViewModel contém a lógica de interface do usuário, como por exemplo os comandos, os eventos, e uma referência para o modelo.

Na figura 18 apresenta a estrutura desenvolvida para uma melhor organização da aplicação web, dividida em módulos.

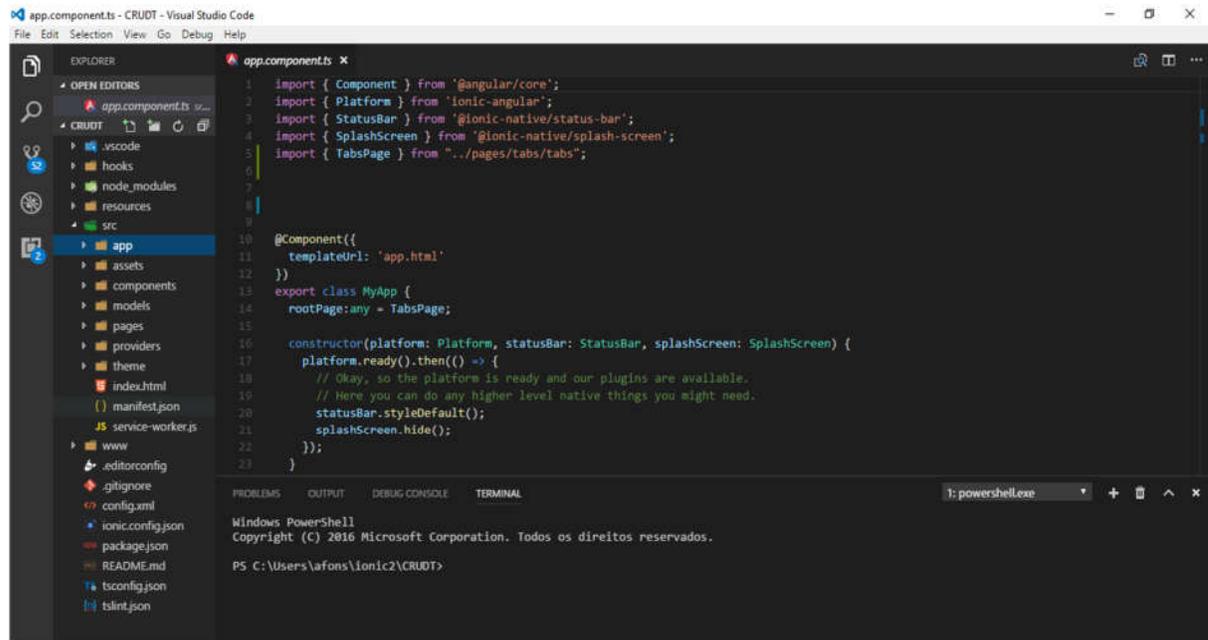
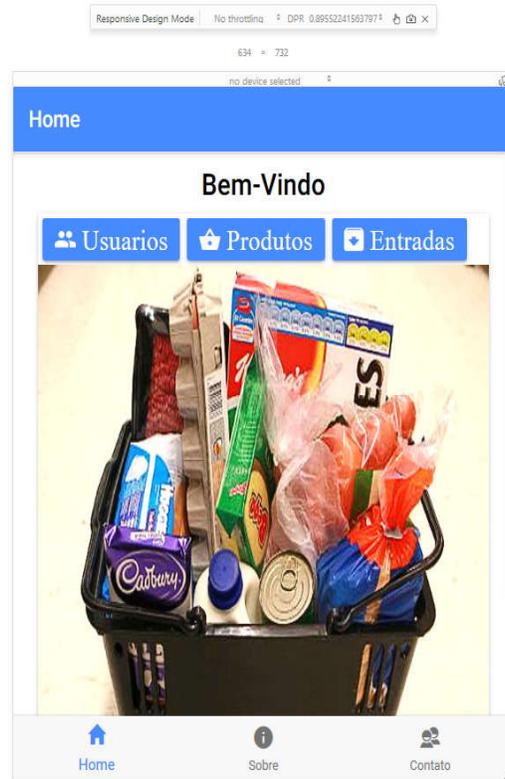


Figura 17 VisualStudio Code

Na figura 19 apresenta a interface da aplicação desenvolvida.



**Figura 18 Interface da aplicação**

## 5. CONCLUSÃO

Com a análise das ferramentas e tecnologias apresentadas no presente trabalho, pode-se concluir que o desenvolvimento de uma aplicação híbrida apesar das facilidades que os frameworks Angular 2 e Ionic 2 oferecem, foi mais difícil do que era esperado, uma vez que os materiais de estudo dessas tecnologias são limitados, pois os frameworks são relativamente novos no mercado e a grande maioria do material encontrado foi em língua estrangeira (inglês).

As utilizações desses frameworks de desenvolvimento trouxeram a necessidade de conhecimento da arquitetura de desenvolvimento web e das linguagens Scripts como HTML, CSS, e JavaScript junto com a linguagem TypeScript que também foi essencial para o desenvolvimento do sistema de forma efetiva.

O desenvolvimento da análise e especificação do sistema, com o uso dos diagramas da UML na modelagem, permitiu ter uma visão clara do sistema, facilitando a identificação das funcionalidades necessárias para um melhor gerenciamento.

### 5.1. TRABALHOS FUTUROS

Será considerado para trabalhos futuros no sistema para gestão de data de produtos alimentícios, a implementação da entrada de produtos na despensa ou na geladeira por meio da captação do código de barras do produto, através da câmera do smartphone, que vai permitir ao usuário mais rapidez e praticidade para cadastrar seus produtos alimentícios.

## 6. REFERÊNCIAS

BOOTH, Joseph D. **Angular 2 Succinctly**. 1.Ed. Syncfusion, 2017.

COUTINHO, Gustavo Leuzinger. **A Era dos Smartphones: Um estudo Exploratório sobre o uso dos Smartphones no Brasil**. 2014. 67p. Monografia; Universidade de Brasília; Faculdade de Comunicação, Comunicação social, habitação em Publicidade e propaganda. Brasília-DF, 2014.

FENTON, Steve. **TypeScript Succinctly**. 1. Ed. Syncfusion, 2013.

GOIS, Adrian. **Ionic Framework : Construa aplicativos para todas as plataformas mobile**. 1 Ed. Casa do código, 2017.

GUEDES, Gilleanes T.A **UML 2: Uma abordagem prática**. 2 Ed. São Paulo: Novatec, 2011.

JUNIOR, Jauri da Cruz. **Solução multiplataforma para smartphone utilizando os frameworks Sencha Touch e PhoneGap integrado à tecnologia web servisse java**. 2014. 72p. Monografia; Fundação Educacional do Município de Assis ; Assis-SP, 2014.

PEREIRA, Caio Ribeiro **Aplicações web real-time com Node.js**. Casa do código, 2013.

PEREIRA, Caio Ribeiro **Construindo APIs Rest com Node.js**. Casa do código, 2013.

PHAN, Hoc. **Ionic 2 Cookbook**. 2. Ed. Parckt Publishing, 2016.

SOMMERVILLE, Ian. **Engenharia de Software**. 6. Ed. São Paulo: Person, 2003.

WARGO, John M. **Apache Cordova API Cookbook**. 1. Ed. Addison-Wesley, 2014.