UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

CAMPUS CORNÉLIO PROCÓPIO

BACHARELADO EM ENGENHARIA DE SOFTWARE

MAGNO CARVALHO DOS SANTOS

**DESENVOLVIMENTO DE SISTEMA DE GERENCIAMENTO DE PUBLICIDADE COM GEOLOCALIZAÇÃO**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2018

MAGNO CARVALHO DOS SANTOS

**DESENVOLVIMENTO DE SISTEMA DE GERENCIAMENTO DE PUBLICIDADE COM GEOLOCALIZAÇÃO**

Proposta de Conclusão de Curso de graduação, apresentado à disciplina Trabalho de Conclusão de Curso 1, do curso de Engenharia de Software da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Bacharel.

Orientador: Eduardo Cotrin Teixeira

CORNÉLIO PROCÓPIO

2018

**DEDICATÓRIA**

Dedico este trabalho à minha família,

em especial meus pais, pelo amor,

incentivo e apoio incondicional.

**AGRADECIMENTOS**

Se diante dessa longa caminhada foi possível escrever esta proposta, agradeço primeiramente a minha família, mesmo estando a milhares de quilômetros de distância sempre me apoiaram incondicionalmente, minhas escolhas e sonhos, por ter me dado força e determinação para vencer todos os obstáculos que surgiram ao longo desta graduação.

Contudo, sinto-me realizado por ter alcançado essa primeira etapa do Trabalho de Conclusão de curso do bacharelado em Engenharia de Software, nesta tão almejada universidade pública. Essa graduação contribuiu muito para minha vida pessoal e profissional ao longo de quatro anos cursados.

Agradeço também a Universidade Tecnológica Federal do Paraná: corpo docente, técnico administrativo, funcionários, colegas de sala por terem me agregado conhecimentos e experiência de vida ao longo desse curso. Tudo isso será de muita importância para minha carreira profissional.

Agradeço a todos os meus amigos que estudamos juntos no decorrer desta graduação.

Agradeço em especial ao Prof. Eduardo Cotrin Teixeira por ter aceitado ser meu orientador e ter acreditado em minha capacidade de realizar este trabalho. Confesso que todas as suas orientações foram de grande importância para realizar este trabalho de conclusão de curso. Além disso, pode contando com sua disponibilidade, paciência e conhecimento transmitido durante essa primeira parte do trabalho de conclusão de curso, tudo isso foi imprescindível para o desenvolvimento deste projeto.

Abraço a todos!

**EPÍGRAFE**

“Nós só podemos ver um pouco do futuro,

mas o suficiente para perceber que há

muito a fazer.”

Alan Turing

**RESUMO**

SANTOS, M. C. dos. **Desenvolvimento de Sistema de Gerenciamento de Publicidade com Geolocalização**. Trabalho de Conclusão de Curso (Graduação) - Engenharia de Software. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

Existem diversas plataformas atualmente destinadas à publicidade para o setor de varejo. Seja em redes sociais ou em plataformas de publicidade, um dos grandes problemas encontrados pelas pequenas e médias empresas do varejo é encontrar mídias de publicidade que sejam eficientes e com boa relação entre custo e benefício. O sistema proposto visa criar uma alternativa de publicidade que permita à essas empresas publicar suas promoções, permitindo que seus clientes na região possam receber as promoções via aplicativo ou site *web* utilizando o GPS (*Global Positioning System*) para direcionar as publicações conforme a posição do usuário. A aplicação será desenvolvida utilizando as mais recentes tecnologias de desenvolvimento multiplataforma, entre elas Angular 6, Node.js, Cordova e MongoDB.

**Palavras-chave:** MongoDB, NodeJS, Angular, Publicidade, Geolocalização.

**ABSTRACT**

SANTOS, M. C. dos. *Development of Geolocation Advertising Management System.* Trabalho de conclusão de curso (Graduation)- Engenharia de Computação - UTFPR - Campus Cornélio Procópio, 2018.

There are several platforms currently targeting advertising for the retail industry. Whether it's social networks or advertising platforms, one of the biggest problems encountered by small and medium-sized retailers is finding cost-effective and cost-effective advertising media. The proposed system aims to create an advertising alternative that allows these companies to publish their promotions, allowing their customers in the region to receive the promotions via the application or web site using the GPS (Global Positioning System) to direct the publications according to the user's position. The application will be developed using the latest multiplatform development technologies, among them Angular 6, Node.js, Cordova and MongoDB.

**Keywords**: MongoDB, NodeJS, Angular, Publicity, Geolocation.

|  |  |
| --- | --- |
|  |  |
|  |  |

**SUMÁRIO**

[1 INTRODUÇÃO 10](#_Toc3166885)

[1.1 Contexto 10](#_Toc3166886)

[1.2 Objetivos gerais 11](#_Toc3166887)

[1.3 Objetivos específicos 11](#_Toc3166888)

[1.4 Problemas e premissas 11](#_Toc3166889)

[1.5 Organização do trabalho 12](#_Toc3166890)

[2 FUNDAMENTAÇÃO TEÓRICA 13](#_Toc3166891)

[2.1. Desenvolvimento híbrido de software 13](#_Toc3166892)

[2.2 Tecnologias utilizadas no desenvolvimento híbrido 14](#_Toc3166893)

[2.2.1 HTML e CSS 15](#_Toc3166894)

[2.2.2 Cordova 15](#_Toc3166895)

[2.2.3 SASS 16](#_Toc3166896)

[2.2.4 JavaScript 16](#_Toc3166897)

[2.2.5 Typescript 16](#_Toc3166898)

[2.2.6 Angular 17](#_Toc3166899)

[2.2.7 Angular Material 2 17](#_Toc3166900)

[2.3 Banco de dados 18](#_Toc3166901)

[2.3.1 MongoDB 18](#_Toc3166902)

[2.4 Backend 19](#_Toc3166903)

[2.4.1 Node.js 20](#_Toc3166904)

[2.4.2 Express.js 20](#_Toc3166905)

[2.5 Modelo de desenvolvimento de software 21](#_Toc3166906)

[3 PROPOSTA 22](#_Toc3166907)

[3.1 Levantamento de requisitos 22](#_Toc3166908)

[3.2 Diagramas UML 24](#_Toc3166909)

[3.2.1 Casos de uso 25](#_Toc3166910)

[3.3 Protótipos 27](#_Toc3166911)

[3.3.1 Protótipos Aplicativo 27](#_Toc3166912)

[3.3.2 Protótipos empresa 31](#_Toc3166913)

[4 CRONOGRAMA DE DESENVOLVIMENTO 34](#_Toc3166914)

[5 CONSIDERAÇÕES FINAIS 35](#_Toc3166915)

[6 REFERÊNCIAS 36](#_Toc3166916)

# 1 INTRODUÇÃO

## 1.1 Contexto

## 1.2 Objetivos gerais

O objetivo deste trabalho é aplicar conceitos da Engenharia de Software no desenvolvimento de um aplicativo *mobile* multiplataforma capaz de obter dados da localização dos dispositivos do usuário, a fim de calcular quais promoções publicadas na plataforma estão próximas do local do usuário, e assim pesquisar por produtos em promoções em sua região, dando às empresas uma plataforma *online* onde ela possa postar as promoções de seu estabelecimento e acompanhar a interação com os clientes.

## 1.3 Objetivos específicos

Os objetivos específicos a serem alcançados nesta proposta de desenvolvimento são:

* Desenvolvimento de aplicação multiplataforma, onde o usuário possa consultar pela sua localização, as promoções divulgadas mais próximas.
* Desenvolvimento de uma aplicação *WEB* para divulgação das promoções e publicidades pelas empresas.
* (API) *backend* para concentração e manipulação dos dados de ambas as visões.

## 1.4 Problemas e premissas

No ano de 2018, com o acesso à internet em dispositivos móveis cada vez maior entre o público, temos um cenário atrativo para a publicidade (MEDIA, 2018). Antes da popularização da informática, as empresas em geral usavam meios tradicionais de comunicação como as redes de televisão, rádio, jornais e revistas para veiculação de publicidade para o grande público. Entretanto com o avanço da tecnologia, todo o comportamento do consumo mudou. A publicidade tradicional como meio de informação, atração, persuasão e recordação talvez já não funcione há muito tempo, sendo a hora dos profissionais de publicidade irem na direção de outras soluções e mídias, revendo as técnicas de produção publicitária e reinventando a publicidade (Figueredo, 2017).

A proposta de software pretende atender exatamente essa demanda, fazendo com que as empresas estejam cada vez mais presentes no meio virtual. Por outro lado, o atual momento da economia brasileira se mostra propício para o incremento da publicidade, pois cerca de 72% da população declara que mudou sua rotina financeira, buscando alternativas e menores preços (SPC Brasil e CNDL, 2018).

## 1.5 Organização do trabalho

A estrutura do presente trabalho divide-se da seguinte maneira:

O capítulo 1 contém a Introdução, na qual é apresentada a motivação e justificativa do trabalho além de uma visão geral do problema.

O capítulo 2 trata da Fundamentação Teórica que envolve o tema do trabalho, as tecnologias e metodologias utilizadas.

O capítulo 3 trata dos métodos de programação e conhecimentos utilizados pelo sistema, apresentando os requisitos funcionais e não funcionais do software, diagramas UML e protótipos.

O capítulo 4 trata do cronograma previsto para a conclusão do trabalho de conclusão de curso.

O capítulo 5, Conclusão, apresenta a dificuldades encontradas, juntamente com os resultados já obtidos.

# 2 [FUNDAMENTAÇÃO TEÓRICA](https://docs.google.com/document/d/1GehFPsJ4qtaoMjaEgtfR6j_yZ1448rvd2cyMZ9dN3MM/edit#heading=h.1t3h5sf)

A ideia do software proposto aqui surgiu em uma competição de empreendedorismo em Londrina (*Startup* *Weekend* *2016*), onde a equipe formada por alunos da UTFPR do campus de Cornélio Procópio foi a grande campeã[[1]](#footnote-2).

Para implementar esse projeto serão aplicadas diversas tecnologias e conceitos, começando pelo desenvolvimento de software hibrido, utilizando TypeScript para codificação geral e banco de dados NoSQL MongoDB.

Este capítulo apresenta informações resultantes do levantamento bibliográfico realizado para o desenvolvimento desta proposta. São apresentados alguns conceitos e informações utilizados para uma melhor visão do projeto.

## 2.1. Desenvolvimento híbrido de software

O desenvolvimento híbrido é a criação de um aplicativo que possa ser executado em qualquer sistema operacional, ou pelo menos em mais de um deles. Com isso o desenvolvedor diminui o tempo de programação, que seria aumentado para cada sistema em que o aplicativo pudesse ser executado (XANTHOPOULOS; XINOGALOS, 2013).

De acordo com Madureira (2017), o aplicativo nativo é programado na linguagem de cada sistema operacional, como Java no Android e Objective-C no iOS, cada plataforma apresentando suas próprias ferramentas e elementos de interface. Mediante essas características, o *app* nativo é desenvolvido para utilização em uma plataforma específica, como iOS ou Android, sendo capaz de explorar todas as potencialidades da plataforma para a qual foi criado, conseguindo ter acesso a diversos recursos dos aparelhos como GPS, câmera, calendário, lista de contatos, entre outros.

É importante salientar que nem sempre os aplicativos nativos precisam da internet para seu funcionamento. Madureira (2017) afirma ainda que os desenvolvedores respeitam a um padrão de *design* que são oferecidos para cada sistema operacional, como um guia. A Figura 1 a seguir mostra a diferença entre o desenvolvimento nativo e híbrido.

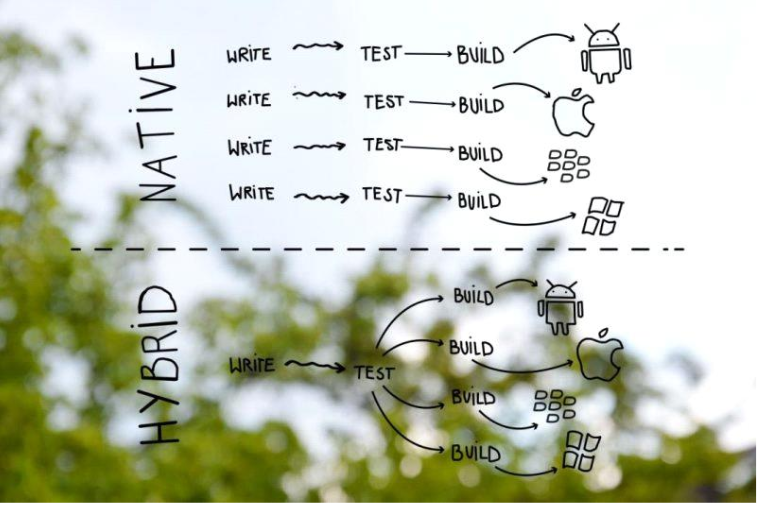


Figura 1 - Diferenças entre o desenvolvimento híbrido e nativo

**Fonte: RADEJ (2017).**

Analisando a Figura 1, observa-se que uma das características que mais diferenciam o aplicativo nativo do aplicativo híbrido é o processo de desenvolvimento. Para desenvolver uma aplicação nativa, como mencionado na definição e ilustrado na Figura, cada dispositivo tem seu próprio processo de desenvolvimento exclusivo (RADEJ, 2017).

Em contrapartida para desenvolver uma aplicação híbrida, o desenvolvimento é iniciado com uma única programação, essa programação é testada e compilada de forma única, e atribuída simultaneamente às plataformas. Caso o desenvolvedor queira acrescentar ou alterar algum detalhe ou requisito do aplicativo, apenas modifica-se todos os códigos fonte, de uma só vez, não precisando alterar todos um a um (VENTEU; PINTO, 2018).

## 2.2 Tecnologias utilizadas no desenvolvimento híbrido

As tecnologias mais utilizadas para esse desenvolvimento híbrido são HTML, CSS e JavaScript (MADUREIRA, 2017). Partindo desse conceito de junção dessas principais tecnologias e a utilização das linguagens de programação *web*, o resultado é o funcionamento do aplicativo em diferentes plataformas. Assim, a seguir são descritas algumas características dessas tecnologias que serão utilizadas no desenvolvimento desse projeto.

### 2.2.1 HTML e CSS

HTML é a sigla em inglês para *Hyper Text Markup Language*. Desenvolvida em 1991 por Sir Tim Berners-Lee, em conjunto com o surgimento do protocolo HTTP. A linguagem permite que durante o desenvolvimento de páginas para *web* seja possível mesclar HTML com outras linguagens como JAVA, Hypertext Preprocessor – PHP, dentre outras (SILVA, 2010).

Já CSS é a abreviação para o termo em inglês *Cascading Style Sheet*, traduzido para o português como *Folhas de Estilo em Cascata*. CSS é o elemento responsável pela estilização dos elementos como cores, bordas, fontes, tamanhos de letras e imagens entre outras customizações. HTML tem como finalidade ser uma linguagem exclusivamente de marcação e estruturação de conteúdo, não cabendo a ela a responsabilidade de fornecer a apresentação de elementos estilizados, como por exemplo, cores, tamanhos, e demais aspectos visuais. É de responsabilidade do CSS toda a apresentação e estilização dos componentes (SILVA, 2010).

### 2.2.2 Cordova

Apache Cordova é um *framework* de desenvolvimento móvel de código aberto. Ele permite que sejam utilizadas tecnologias *web* padrão, como HTML5, CSS3 e JavaScript, para desenvolvimento multiplataforma. Os aplicativos são executados em invólucros direcionados para cada plataforma, e contam com *plugins* para acessar os recursos de cada dispositivo, tais como GPS, sistema de arquivos e câmera (WARGO, 2015).

Cordova envolve um *app* HTML/JavaScript em um recipiente nativo que pode acessar as funções do dispositivo de várias plataformas. Estas funções são acessadas através de uma API (*Application Program Interface*) JavaScript unificada, permitindo que com um único código seja possível atingir quase todos os telefones e/ou *tablets* no mercado e publicar as aplicações em suas lojas de aplicativos (WARGO, 2015).

As plataformas suportadas pelo *framework* são: Android, iOS, BlackBerry10, Ubuntu, Windows Phone 8, Bada, FireOs e LG We OS. A principal vantagem do Cordova é o pacote de *plugin*s que ele oferece. Estes permitem que a aplicação acesse recursos nativos do dispositivo. Além daqueles oferecidos pelo Cordova, também existem *plugins* de terceiros compatíveis com a plataforma, ou ainda, a possibilidade de criar-se um *plugin* próprio para integrar a aplicação (WARGO, 2015).

### 2.2.3 SASS

O SASS é uma extensão ao CSS que permite agilizar o desenvolvimento de códigos CSS, utilizando uma sintaxe mais elegante e de fácil manutenção, permitindo também o uso de variáveis e funções. Tem ganhado popularidade por trazer novos recursos e facilidade no desenvolvimento (CATLIN; WEIZENBAUM; EPPSTEIN, 2006)

### 2.2.4 JavaScript

A linguagem ECMAScript, popularmente conhecida por JavaScript, foi desenvolvida por Brendan Eich e surgiu originalmente implementada como parte dos navegadores para carregar os *scripts* na máquina do cliente, sem que fosse necessária sua interação com o servidor (POWERS, 2010). Atualmente está na versão ECMAScript 7 (ES7), entretanto as suas novas funcionalidades não estão implementadas na maioria dos navegadores (LERNER, 2018).

### 2.2.5 Typescript

TypeScript é um superconjunto de JavaScript desenvolvido pela Microsoft que adiciona tipagem e alguns outros recursos à linguagem (FOLEY, 2012). Angular possui como linguagem de desenvolvimento o TypeScript, e também suporta JavaScript e Dart. TypeScript trata-se apenas de uma linguagem que visa agilizar o desenvolvimento das aplicações, sendo considerada pela comunidade a 4ª linguagem "mais amada", de acordo com uma pesquisa conduzida pelo site Stack Overflow em 2018 (STACK OVERFLOW, 2018).

Quando o código TypeScript é compilado gera-se o código JavaScript na versão ES6 para que o navegador possa ler e executar os *scripts*, esta versão gerada também pode ser configurada nos arquivos de configuração *ts* (LERNER, 2018).

### 2.2.6 Angular

Angular é um *framework* JavaScript que possui como principal contribuidor a Google, contando inclusive com uma comunidade dedicada ao seu desenvolvimento. Tem o propósito de alcançar a velocidade máxima possível na plataforma *web* atual e ir além via *Web* *Workers*, renderizando no lado do servidor ao invés do lado do cliente. Permite fácil reutilização de código seja para aplicação *web*, *web* móvel, nativa móvel ou *desktop* (GOOGLE, 2018a).

Angular é responsável pela renderização das telas no navegador do cliente e seu grande diferencial é a Orientação a Componentes, permitindo assim uma melhor organização e reutilização de código. Possui documentação completa em seu *site* oficial[[2]](#footnote-3)(GOOGLE, 2018a).

### 2.2.7 Angular Material 2

O Angular Material é um framework desenvolvido por uma comunidade de designers e programadores, que implementa as especificações do Material Design da Google, que tem por objetivo criar uma linguagem visual que sintetize os princípios de boas práticas de design, permitindo um sistema que proporcione uma experiência unificada entre as plataformas e tamanhos de dispositivos.

O Angular em si não nos obriga a utilizar determinado *framework* CSS, nem mesmo traz por padrões componentes prontos, é necessário que se crie os componentes que deseja utilizar. Tendo isto em mente o Angular Material 2 vem para dar uma impulsionada a mais no desenvolvimento pelo programador sem conhecimentos avançados de UI, pois os seus componentes são facilmente manipulados através de tags próprias com atributos autoexplicativos, com diversos outros componentes já prontos, tabelas, animações e temas (GOOGLE, 2018b).

## 2.3 Banco de dados

O Sistema Gerenciador de Banco de Dados escolhido para o desenvolvimento do software proposto foi o MongoDB, uma solução NoSQL (*Not Only SQL* - não apenas SQL), devido ao grande grau de maturidade que o mesmo já possui, bem como pela ampla adoção pela comunidade, além de sua eficiência quando aliado ao Node.js (SCHROEDER; SANTOS, 2014), além dos quesitos; flexibilidade com estruturas de dados intuitivas e flexíveis (CUNHA, 2011), escalabilidade, distribuindo os dados por clusters melhor que os SGBDs relacionais (CHODOROW, 2013), disponibilidade já que a maioria dos bancos de dados NoSQL oferecem eficientes arquiteturas de replicação de dados que proporciona aos NoSQLs maior disponibilidade e custo benefício (CUNHA, 2011).

### 2.3.1 MongoDB

O MongoDB é um banco de dados de código aberto gratuito de alta performance, sem esquemas e orientado à documentos, disponível no github[[3]](#footnote-4). Foi lançado em fevereiro de 2009 pela empresa 10gen, escrito na linguagem de programação C++, o que o torna portável para diferentes Sistemas Operacionais (CHODOROW, 2013).

O MongoDB é visto como um dos mais bem-sucedidos bancos NoSQL. Tal fato se deve à sua linguagem de consulta de alta performance, a qual baseia-se em documentos e na facilidade de importar os dados de um banco relacional.

Segundo Sadalage e Fowler (2013), os bancos de dados NoSQL baseiam-se nas necessidades da *web* no século XXI.

O MongoDB armazena dados dentro de documentos semelhantes a JSON (usando BSON — uma versão binária de JSON), que retém os dados usando pares de chave/valor. Um banco de dados orientado a documentos armazena, recupera e gerencia dados semiestruturados. O elemento dos dados é chamado documento. Os documentos são endereçados no banco de dados via uma chave única que representa o documento (MONGODB, 2018).

Por ser orientado à documentos JSON, muitas aplicações podem modelar informações de modo mais próximo ao natural, pois os dados podem ser aninhados em hierarquias complexas e ainda assim serem indexáveis e fáceis de buscar, igual ao que já é feito em JavaScript. Neste tipo de banco (*document-based* ou *document-oriented*), tem-se coleções de documentos, nas quais cada documento é auto-suficiente e contém todos os dados, ao invés do conceito de não repetição e de chaves estrangeiras presentes no modelo relacional (CUNHA, 2011).

A ideia é não utilizar junções (relacionamento entre tabelas) nas consultas, para não prejudicar a eficiência das consultas. A modelagem da base deve ser feita de forma que a cada query será feita uma única busca no banco, e que com apenas uma chave primária possa ser selecionado tudo que for necessário.

## 2.4 Backend

Uma aplicação *backend* em computação é o sistema responsável por armazenar e/ou processar e fornecer dados para que outras aplicações ou sistemas possam consumir esses dados de diversas formas. Geralmente os dados de todo o sistema são armazenados em bancos de dados e trabalhados por outros processos, para que possam gerar novas informações posteriormente (BACKENDLESS, 2018).

É também comum encontrar toda a estrutura de regras de negócio de um sistema sob a responsabilidade de uma aplicação *backend*, uma vez que esta é a que dispõe de mais recursos computacionais quando a comparamos com aplicações cliente, como páginas web sendo requisitadas de notebooks de uso pessoal, aplicativos móveis rodando em smartphones comuns e outros (PLURALSIGHT, 2015).

Em um contexto geral, o papel fundamental das aplicações de *backend* é a disponibilização de informação remota por meio de protocolos de comunicação, utilizando a internet, *bluetooth* e outros padrões de transporte de informação.

### **2.4.1 Node.js**

O Node.js é uma plataforma de desenvolvimento *web* que funciona com a linguagem JavaScript no lado do servidor, para criação de aplicações e páginas *web* de alta escalabilidade (NODEBR, 2016). Foi concebida por Ryan Dahl em 2009, e desde então vem ganhando muita popularidade entre os desenvolvedores *web*, sendo utilizado por grandes empresas e instituições como LinkedIn, Microsoft, GitHub e MySpace (HANDY, 2011).

A arquitetura do Node.js é composta em sua maior parte por componentes desenvolvidos em C++ e em JavaScript. A escolha da linguagem JavaScript foi devido à enorme quantidade de bibliotecas para I/O que a linguagem possui, permitindo assim criar as funções assíncronas para a plataforma. (NODEBR, 2016).

O Node.js foi criado de forma que suas funcionalidades pudessem ser estendidas através de módulos, para implementar diversos componentes *middleware* que facilitem o desenvolvimento de aplicações *web*. Estes módulos são geralmente instalados através de um gerenciador de pacotes conhecido como npm, que significa *Node Package Manager*, e serve para facilitar diversos aspectos relacionados à aplicação como a compilação, a instalação e a atualização e também funciona como um gerenciador de dependências (NODE.JS, 2018).

A escolha pelo Node.js se deve à sua arquitetura orientada a eventos, sua comunidade ativa e sua alta escalabilidade e performance.

### 2.4.2 Express.js

O Express.js é um *framework* que visa facilitar o desenvolvimento de aplicações w*eb* com o Node.js, e é um dos pacotes mais populares entre os desenvolvedores de acordo com as estatísticas de registro do NPM (NPM, 2018), com o Express.js é possível instanciar servidores *web* e receber requisições HTTP de maneira trivial, além de permitir criar um conjunto de diretórios com uma arquitetura padrão, e também organizar as rotas dos arquivos para as *views* da aplicação. Com uma infinidade de métodos utilitários HTTP e *middleware* à sua disposição, é possível criar uma API robusta, de maneira rápida e fácil (EXPRESS.JS, 2018).

## 2.5 Modelo de desenvolvimento de software

O processo de desenvolvimento seguirá a metodologia de desenvolvimento ágil Scrum Solo, que se caracteriza como um processo iterativo e incremental que une as boas práticas delineadas pelo *Personal Software Process* (PSP) e pelo Scrum. O Scrum Solo tem no geral uma estrutura similar ao Scrum tradicional proposto por Schwaber e Sutherland. Este processo fora utilizado por alunos do curso de Engenharia da Computação e Análise de Desenvolvimento de Sistemas da UTFPR nos anos de 2012 a 2014 com sucesso no desenvolvimento. Ex-alunos desenvolvedores de software relataram que o processo contempla as necessidades de gestão de projetos (PAGOTTO, 2016).

No desenvolvimento do projeto será usado o cronograma de processos definido no Scrum Solo. Será gerado o *Requeriment* do projeto definindo os *product backlogs*, levantamento e validação de requisitos junto ao público e clientes empresas, gerando um repositório online com todas as informações do projeto. Posteriormente, seguindo o Scrum Solo, serão definidas as *Sprints* de desenvolvimento, uma vez que todos os *product backlog* estarãodefinidos. Ao final de cada *Sprint* de uma semana será gerado um *Deployment* para disponibilizar uma versão com todas as modificações realizada em cada *Sprint.*

Durante o desenvolvimento será registrado tudo que for feito para um relatório desse trabalho, já que hoje ainda não há como mensurar a quantidade de *Sprints* a serem realizadas, pois o mesmo só será possível após o início do levantamento e validação de requisitos junto ao público e clientes empresas.

# 3 PROPOSTA

Neste capítulo são apresentados alguns conceitos e informações para dar uma melhor visão do projeto, abrangendo o levantamento de requisitos funcionais e não funcionais, diagramas UML (*Unified Modelling Language*) e protótipos iniciais.

## 3.1 Levantamento de requisitos

O levantamento de requisitos é o processo de criação, onde cliente, analistas, engenheiros, gerentes e basicamente todos envolvidos no desenvolvimento do software criam, em conjunto, um grupo de requisitos baseados nas tarefas que compõem o escopo do projeto que o cliente deseja (PRESSMAN, 2011).

Na engenharia de software, os requisitos devem ser claros, quantificáveis e livres de interpretação dúbia (PAULA FILHO; PÁDUA, 2016). Esses requisitos podem ser divididos em funcionais e não-funcionais (SOMMERVILLE, 2011). Os requisitos funcionais definem o que o software deverá fazer, descrevendo detalhes técnicos ou manipulação e armazenamento de dados. Já os requisitos não-funcionais são os requisitos que definem a qualidade e o desempenho do software (PRESSMAN; MAXIM, 2016).

Os requisitos estão divididos em dois módulos: o primeiro módulo é o do aplicativo, que será a parte do software onde o usuário pode acessar as promoções postadas, esse módulo será multiplataforma tendo versão web, Android e IOS. O outro módulo, da empresa, é onde as empresas publicam as promoções de seu estabelecimento.

**Requisitos Funcionais do Aplicativo:**

* RFA-001 – O Sistema deverá permitir ao usuário efetuar *login* utilizando perfil da rede social Facebook.
* RFA-002 – O Sistema deverá capturar a localização do usuário e enviá-la ao servidor.
* RFA-003 – O Sistema deverá atualizar a localização do usuário quando o mesmo se movimentar.
* RFA-004 – O Sistema deverá requisitar ao usuário a permissão de acessar sua localidade.
* RFA-005 – O Sistema deverá calcular a distância entre o usuário e a localização das empresas com promoções ativas próximas, pela localização do mesmo.
* RFA-006 – O Sistema deverá permitir que o usuário acesse informações sobre a empresas ativas com promoções.
* RFA-007 – O Sistema deverá filtrar a exibições de promoções pelo perfil do usuário.
* RFA-008 – O Sistema deverá permitir que o usuário salve suas promoções preferidas.
* RFA-009 – O Sistema deverá permitir que o usuário edite suas informações de perfil.
* RFA-010 – O Sistema deverá permitir que o usuário faça o “*CHECK-IN*” na promoção que ele vai interagir.
* RFA-011 – O Sistema deverá usar os parâmetros de busca do usuário para configuração ao fazer a busca de promoções na região.
* RFA-012 – O Sistema deverá usar como parâmetro inicial as 10 empresas localizadas na proximidade do cliente.
* RFA-013 – O Sistema deverá usar com padrão no seu *dashboard* carregamento dinâmico.
* RFA-014 – O Sistema deverá permitir que o usuário liste suas promoções marcadas como favoritas.

**Requisitos funcionais da plataforma empresa:**

* RFPE-001 – A plataforma empresa deverá autenticar uma solicitação de *login*.
* RFPE-002 – A plataforma empresa deverá informar o alcance de visualização que empresa esteja tendo.
* RFPE-003 – A plataforma empresa deverá permitir o cadastro de novas promoções pelo administrador da empresa.
* RFPE-004 – A plataforma empresa deverá permitir o cadastro e edição das informações da empresa.
* RFPE-005 – A plataforma empresa deve permitir que o administrador da empresa visualize a quantidade de ‘*CHECK-IN’* e favoritos de suas promoções cadastradas.
* RFPE-006 – A plataforma empresa deverá permitir a edição das promoções já cadastradas.

**Requisitos não-funcionais de ambiente:**

* RNFA-001 – O Sistema deverá ser escrito em linguagem TypeScript.
* RNFA-002 – O Sistema deverá executar sobre plataforma Android.
* RNFA-003 – O Sistema deverá ser construído utilizando a ferramenta de desenvolvimento Visual Studio Code.
* RNFA-004 – O Sistema deverá ter conectividade com a Internet para conexão com a API.
* RNFA-005 – O Sistema deverá ter acesso à localização do usuário enquanto utiliza a aplicação.

## 3.2 Diagramas UML

Diagramas UML (*Unified Modelling Language*), são tipos de linguagens em forma de diagramas que auxiliam os desenvolvedores a especificar, visualizar, construir e documentar modelos de softwares orientados por objetos (BOOCH; RUMBAUGH, 2005).

Empresas praticantes da metodologia ágil veem a modelagem com pouco valor para o cliente, e pouco útil quando feita anteriormente ao início do desenvolvimento efetivo do projeto (ABRAHAMSSON; BABAR; KRUCHTEN, 2010). Há também a percepção que programadores não gostam de fazer modelagem e documentação. Assim, se os desenvolvedores ágeis não perceberem a importância da modelagem, eles não vão usá-la, nem seus artefatos (SCHROEDER; SANTOS, 2014).

Parece consenso que o uso deste arcabouço com muita antecipação antes do início do desenvolvimento efetivo do software pode levar a decisões equivocadas. Também é prudente avaliar que a modelagem deve ser de acordo com o contexto: projetos grandes e complexos necessitam de um esforço de modelagem maior. Apesar da arquitetura evoluir interativamente, decisões de maior dificuldade de mudança e mais permanentes devem ser tomadas nas primeiras interações (ABRAHAMSSON; BABAR; KRUCHTEN, 2010).

Tendo o foco no desenvolvimento ágil, inicialmente o único diagrama UML criado para o software proposto nesse trabalho será o diagrama de caso de uso. Outros diagramas UML serão desenvolvidos conforme a demanda da exigência dos requisitos. Nessa fase do projeto, que exige apenas uma abstração inicial da proposta, foram desenvolvidos o caso de uso junto com os protótipos do projeto, aliados ao modelo de processo do Scrum Solo.

### 3.2.1 Casos de uso

Um caso de uso é uma descrição de um comportamento de um sistema, seja ele o sistema completo, ou apenas uma tarefa separada dentro do sistema como um todo, sendo que esta descrição é realizada relatando passo a passo a interação do usuário com o sistema, em um cenário ótimo (MELO, 2004).

A seguir estão descritos os casos de uso que contemplam a solução apresentada, sendo definidos os atores como usuário aplicativo e usuário empresa:

O usuário aplicativo interage com o módulo aplicativo conforme descrito nos requisitos funcionais no item 3.1 referentes a plataforma aplicativo. Ele pode acessar a lista de promoções cadastradas nas proximidades de sua localização via GPS, uma vez que esteja autenticado no sistema.

Já o usuário empresa interage com o módulo empresa descrito nos requisitos funcionais no item 3.1 referentes a plataforma empresa. Ele pode cadastrar as promoções de seu estabelecimento e verificar a interação das promoções já cadastradas com seus clientes, uma vez que esteja autenticado no sistema.

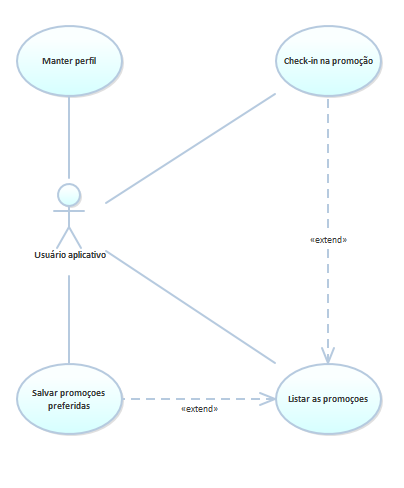


Figura 2 – Caso de uso usuário aplicativo

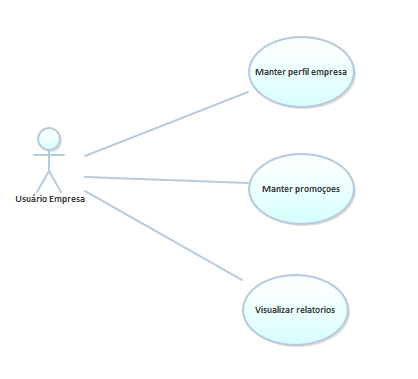


Figura 3 - Caso de uso usuário empresa

## 3.3 Protótipos

Um protótipo é um conceito resultante em uma estrutura física (hardware, papel, etc.) ou digital (imagem, vídeo 3D, HTML, etc.), que permite aos interessados uma experiência de escolha de aspectos fundamentais de um produto em potencial (GOMES, 2013),

Em virtude da crescente utilização de dispositivos móveis (*smatphones* e *tablets*) (PAULO, 2015), nessa proposta buscou-se encontrar um ambiente de programação onde fosse possível o desenvolvimento de ferramentas que independessem de plataforma, mas que fossem adequadas aos sistemas móveis atuais (também conhecido como “*mobile first*”), onde todo o desenvolvimento do projeto *web* é inicialmente projetado com foco nos dispositivos móveis e posteriormente para a visão *desktop* (MARCOTTE, 2017).

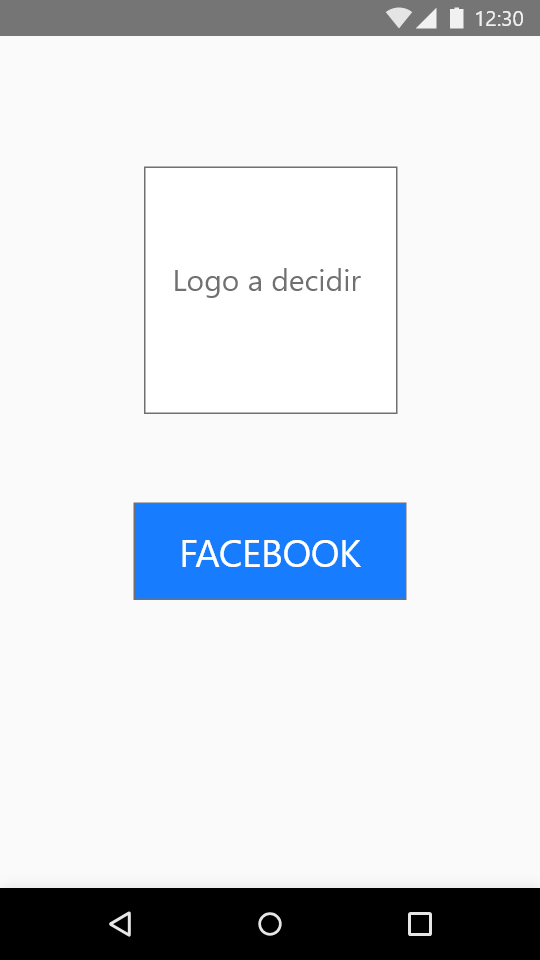
Como o projeto tem duas vertentes, uma para usuário aplicativo e outra usuário empresa uma vez que os requisitos e caso de uso foram definidos, a primeira versão dos protótipos fora criada. Para a criação foi utilizado o software Adobe XD para o *design* de experiência do usuário, o mesmo é desenvolvido e publicado pela Adobe Systems.

### 3.3.1 Protótipos Aplicativo

Nessa sessão teremos os principais protótipos da visão aplicativo.

#### **3.3.1.1 Tela de login**

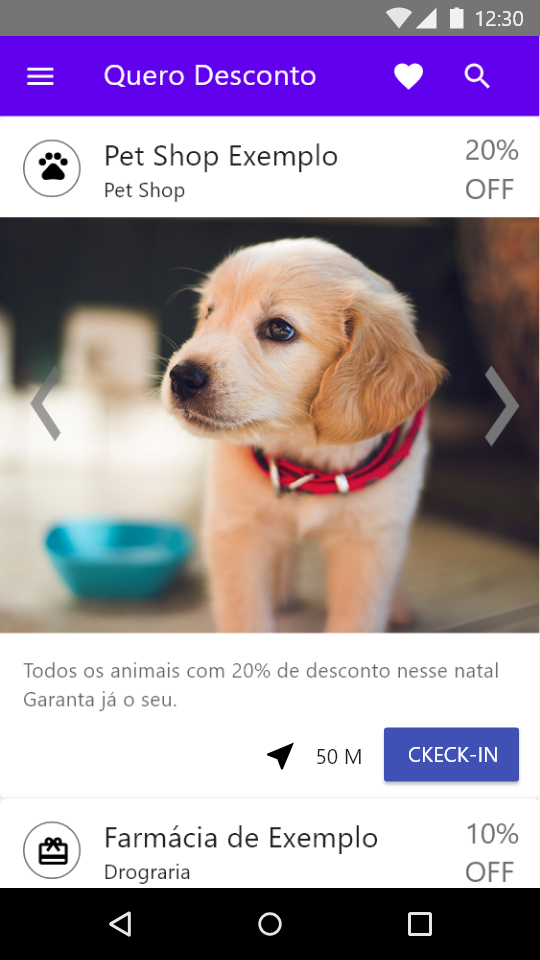
Nesta tela, observada na Figura 4, é mostrado o ícone do aplicativo, e o botão de *login* com o Facebook. A única opção no momento para entrar na aplicação é apertando o botão de *login*. Caso ocorra algum erro, uma mensagem é apresentada ao usuário do aplicativo. Esta tela não é mostrada caso o usuário já tenha realizado o *login* anteriormente.



**Figura 4 – Tela de login**

#### **3.3.1.2 Tela Principal**

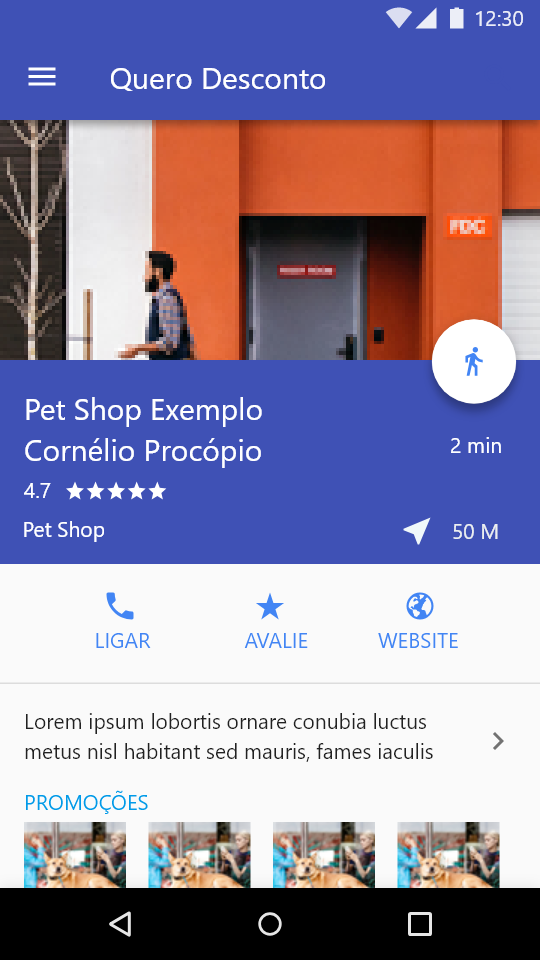
Nesta tela, observada na Figura 5, O menu principal é a tela base da aplicação, por meio dela é possível acessar as funcionalidades do aplicativo. A principal funcionalidade é de acessar e listar as promoções próximas do usuário do aplicativo, ordenadas pela proximidade ao usuário. Apertando no canto superior esquerdo podem-se observar as opções do menu, e apertando no canto superior direito, podem-se acessar os filtros das promoções a serem mostradas. Apertando na opção *check-in* da promoção o usuário tem acesso a informações mais detalhadas sobre a mesma, além de acesso ao desconto cadastrado. Clicando no nome da empresa o usuário tem acesso a informações detalhadas da empresa como observado na Figura 6.



**Figura 5 – Tela de Principal**

**3.3.1.3 Tela Perfil Empresa**

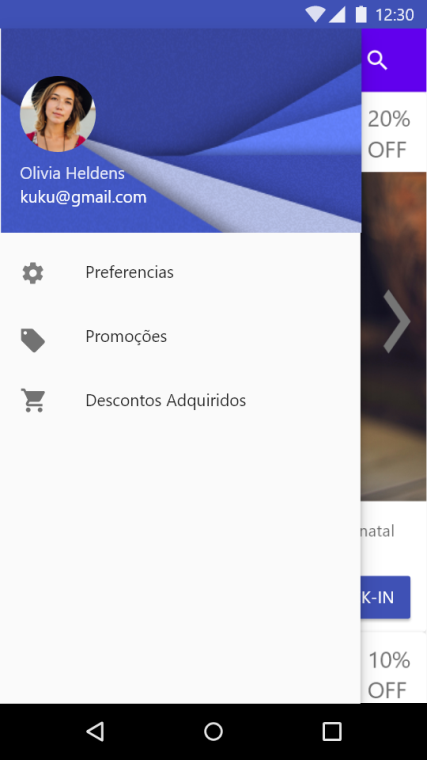
Nesta tela, observada na Figura 6, o usuário tem acesso as informações da tela da empresa, como distância do seu local calculado usando dados do GPS, telefone para contato da empresa, uma pequena avaliação, as ofertas e promoções da empresa, além do *link* do *website* ou rede social da empresa.



**Figura 6 – Tela de Perfil empresa**

#### **3.3.1.4 Tela Menu principal**

Nesta tela (Figura 7), o usuário tem acesso às informações do menu principal como foto de perfil e nome do usuário, além das opções de interagir com itens do menu.



**Figura 7 – Tela de menu principal**

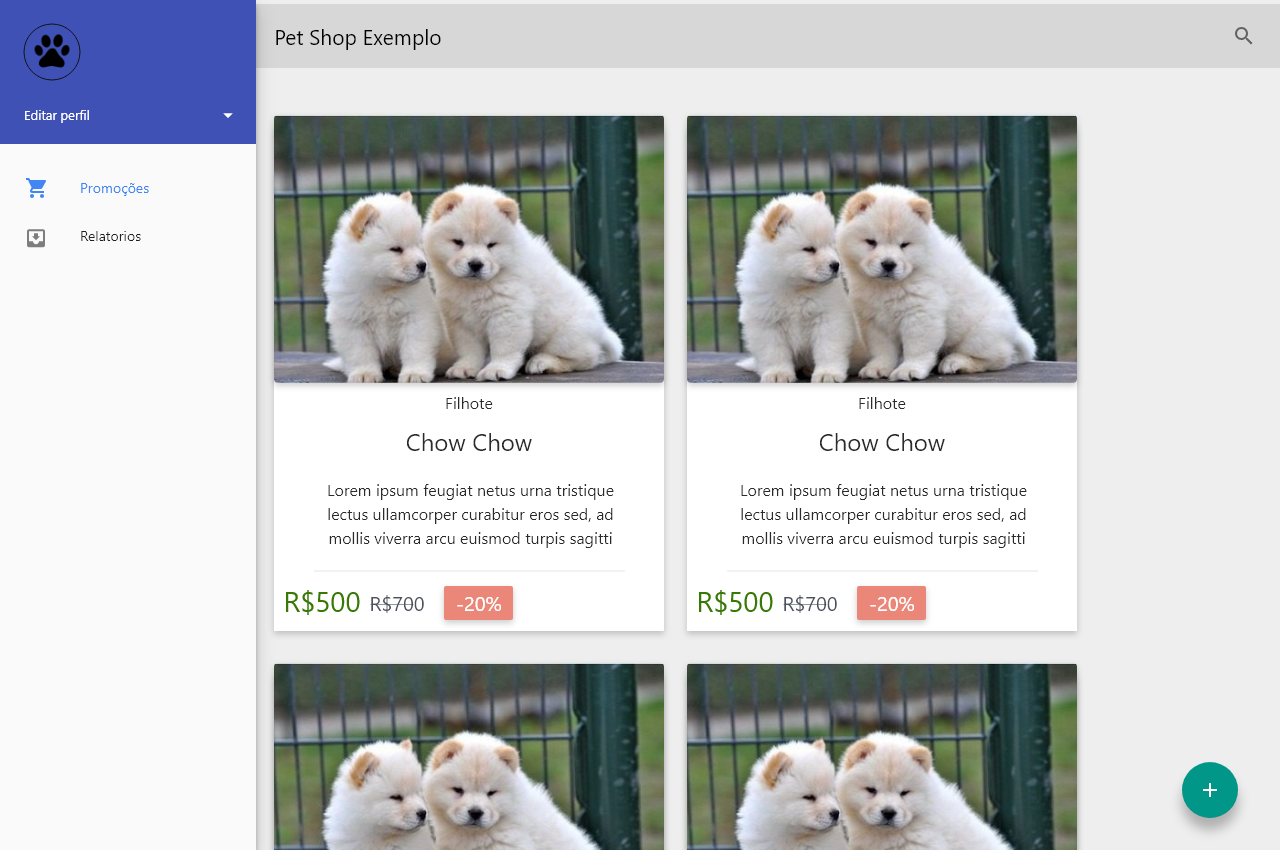
### 3.3.2 Protótipos empresa

Nessa sessão teremos os principais protótipos da visão empresa. Os protótipos foram criados diferentes da visão aplicativo utilizando um modelo *desktop* de 1280 x 850.

#### **3.3.2.1 Tela principal**

Nesta tela (Figura 8 - Tela principal da visão empresa), o usuário empresa tem acesso a lista de todas as promoções cadastradas. Na lateral esquerda há um menu com as opções de editar o perfil da empresa, promoções e relatório No canto inferior direito há um botão flutuante para realizar o cadastro de nova promoção, e no canto superior direito há um Ícone de lupa com um *input* para realizar busca nas promoções listadas.

A listagem utiliza o padrão de design dos Cards do Angular Material[[4]](#footnote-5), trazendo várias informações da promoção, como nome, preço e imagem.

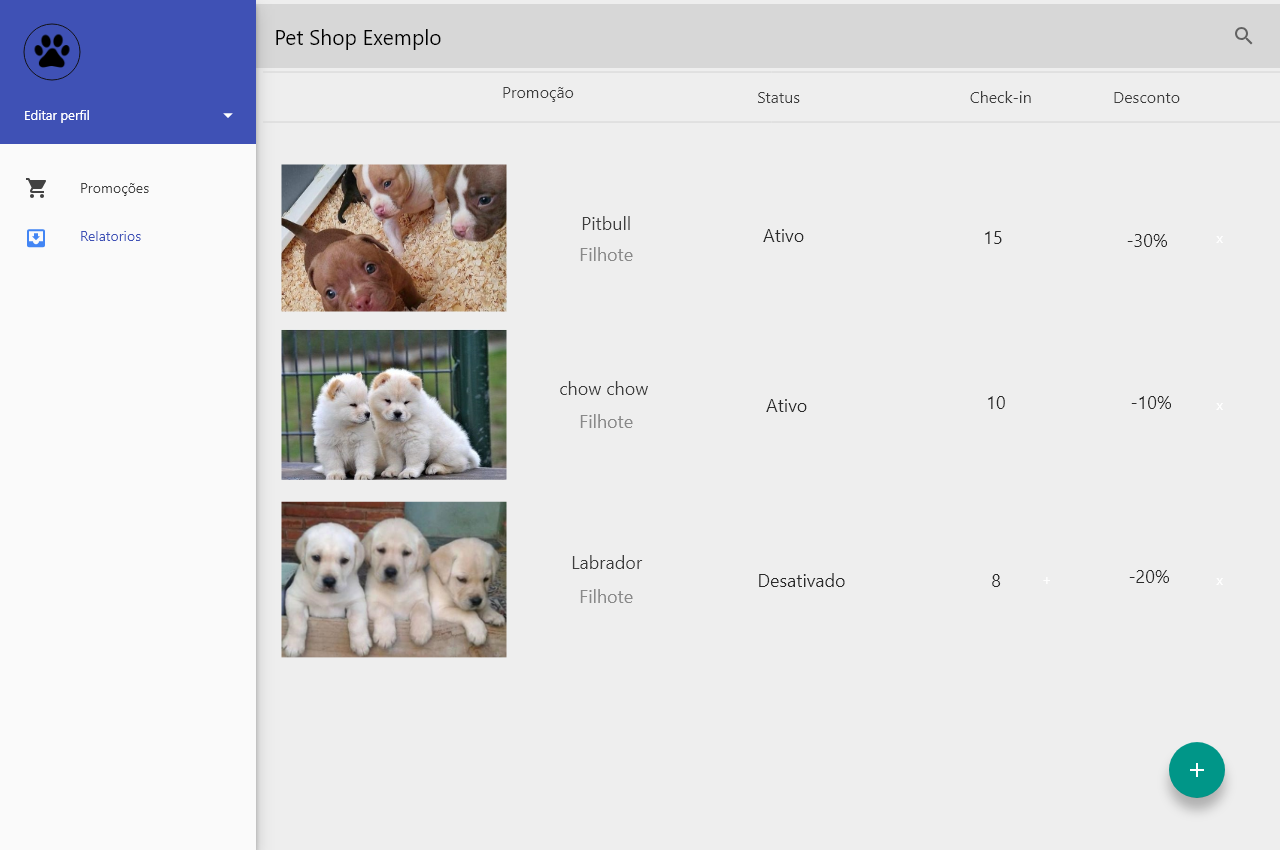


**Figura 8 – Tela Principal**

#### **3.3.2.2 Tela de Relatório**

Nesta tela, observada na Figura 9, o usuário empresa tem acesso ao relatório em tabela das interações dos clientes com suas promoções publicadas. No canto inferior direito há um botão flutuante de cadastro de nova promoção, e na esquerda há um menu lateral com as opções para editar o perfil da empresa, promoções e relatório.

A listagem utiliza o padrão de design dos *Table* do Angular Material[[5]](#footnote-6), trazendo várias informações da promoção, como nome, status, valor do desconto, imagem atribuída a promoção e quantidades de interações



**Figura 9 – Tela de Relatório**

# 4 CRONOGRAMA DE DESENVOLVIMENTO

Cronograma previsto para a conclusão do projeto, elaboração de todos os requisitos e do projeto proposto.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Atividades** | **Nov** | **Dez** | **Jan** | **Fev** | **Mar** | **Abr** | **Mai** | **Jun** |
| Correção e Conclusão da proposta | x |  |  |  |  |  |  |  |
| Planejamento do desenvolvimento  (*Requeriment*) |  | x | x |  |  |  |  |  |
| Desenvolvimento (*Sprints*) |  |  |  | x | x | x | x |  |
| Conclusão do TCC 2 |  |  |  |  |  |  |  | x |

5 CONSIDERAÇÕES FINAIS

Durante a elaboração da proposta, este projeto demonstrou-se bastante ambicioso devido o uso de tecnologias que são consideradas novas. Porém, foi possível realizar um estudo aprofundado das tecnologias que serão usadas no desenvolvimento. Para a realização do desenvolvimento do software, também será realizado um estudo para validar e levantar possíveis novos requisitos, tendo como foco principal a grande diversidade nos produtos que podem ser publicados na plataforma.

As dificuldades encontradas no desenvolvimento dessa proposta, foram de encontrar exemplos de desenvolvimento de proposta de software utilizando metodologia de desenvolvimento ágil Scrum solo.

Todo o desenvolvimento do aplicativo e o seu código fonte estará disponibilizado de forma aberta na plataforma Github.

# 6 REFERÊNCIAS

ABRAHAMSSON, Pekka; BABAR, Muhammad Ali; KRUCHTEN, Philippe**. Agility and architecture: Can they coexist?. IEEE Software**, v. 27, n. 2, 2010.

ANTOUN, Henrique. Web 2.0. Mauad Editora Ltda, 2014.

BACKENDLESS. **WHAT IS BACKEND AS A SERVICE?** 2018. Disponível em: <https://backendless.com/platform/backend-as-a-service/>. Acessado em: 15 out. 2018

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: Guia do Usuário. 2. ed.** São Paulo: Editora Campus, 2005.

BROINIZI, M. E. B. **Ordenação evolutiva de anúncios em publicidade computacional.** 2015. 102 f. Texto (Obtenção de Título de Doutor em Ciências) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2015.

CATLIN, H.; WEIZENBAUM, N.; EPPSTEIN, C. SASS**: Syntactically Awesome Style Sheets**, 2006. Disponível em: <https://sass-lang.com/>. Acesso em: 10 set. 2018.

CHODOROW, K. **MongoDB The Definitive Guide: Powerful and Scalable Data Storage**. O’Reilly Media, Inc, 2013.

CUNHA, T. M. de A. **Escalabilidade de Sistemas com Banco de Dados NoSQL: um Estudo de Caso Comparativo com MongoDB e MySQL**. 85 f. Salvador: Centro Universitário da Bahia – Estácio. 2011.

EXPRESS.JS. Fast, unopinionated, minimalist web framework for Node.js. Disponível em: <http://expressjs.com/>. Acesso em: 29 set. 2018.

FIGUEREDO, Cristiane. **Publicidade na era do consumidor digital: como o crescimento das mídias sociais vem interferindo no modo de fazer publicidade**. 2017.

FOLEY, Mary Jo. **Microsoft takes the wraps off TypeScript, a superset of JavaScript. ZDNet.** CBS Interactive. Disponível em:<https://www.zdnet.com/article/microsoft-takes-the-wraps-off-typescript-a-superset-of-javascript/>. Acesso em: 12 set. 2018.

GOMES, C. T. N. **Introdução a prototipação e apresentação do Axure RP 6.5**. DevMedia, Disponível em: <https://www.devmedia.com.br/introducao-a-prototipacao-e-apresentacao-do-axure-rp-6-5/27978>. Acessado em: 12 out. 2018.

GOODCHILD M. F. **Citizens as sensors: The world of volunteered geography**. GeoJournal , 69:211-221, Aug. 2007

GOOGLE. **Angular** 2018. Disponível em: <https://angular.io/>. Acesso em: 14 set. 2018.

GOOGLE. **Angular Material 2**. 2018. Disponível em:<https://material.angular.io/>. Acesso em: 14 set. 2018.

HANDY, Alex. **Node.js pushes JavaScript to the server-side** – SD Times. 2011. Disponível em: <https://sdtimes.com/javascript/node-js-pushes-javascript-to-the-server-side/>. Acesso em: 20 set. 2018.

LERNER, A. et al. **Ng-book 2 The Complete Book on AngularJS 2**. Disponível em: <https://www.ngbook.com/2/>. Acesso em: 11 set. 2018.

MADUREIRA, Daniel. **Aplicativo nativo, web App ou aplicativo híbrido?.** Use Mobile. Ouro Preto, 2017. Disponível em: <https://usemobile.com.br/aplicativo-nativo-web-hibrido/>. Acesso em: 29 set. 2018.

MARCOTTE, E. **Responsive web design: A book apart n° 4**. Editions Eyrolles, 2017

MEDIA, Vox. **Advertisers will spend $40 billion more on internet ads than on TV ads this year.** Disponível em:<https://www.recode.net/2018/3/26/17163852/online-internet-advertisers-outspend-tv-ads-advertisers-social-video-mobile-40-billion-2018>. Acesso em 10 set. 2018.

MEIRELLES, Fernando S. **29ª Pesquisa Anual do Uso de TI**, GV-EAESP - GVcia, 2018.

MELO, Ana Cristina**. Desenvolvendo Aplicações com UML 2**. Brasport, 2004.

MONGODB, Inc. **MongoDB**. Disponível em: <https://www.mongodb.com/>. Acesso em: 19 set. 2018.

NAVATHE, Shamkant B.; ELMASRI, Ramez. **Sistemas de banco de dados**. Sham, Addison. Ribeirão Preto SP, 2005.

NODE.JS,inc. **Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine**. Disponível em: <https://nodejs.org>. Acesso em: 20 set. 2018.

NODEBR. **O que é Node.js?** Disponível em: <http://nodebr.com/o-que-e-node-js/>. Acesso em: 20 set. 2018.

NPM, Inc. **Most Depended upon Packages. 2018**. Disponível em: <https://www.npmjs.com/browse/depended#product-navigation/>. Acesso em: 29 set. 2018.

PAGOTTO, Tiago et al. **Scrum solo: software process for individual development. In: Information Systems and Technologies (CISTI),** 2016 11th Iberian Conference on. IEEE, 2016. p. 1-6.

PAULA FILHO, W.; PÁDUA, D. E. **Engenharia de software; fundamentos, métodos e padrões, 2003.** Rio de Janeiro, Editora LTC, 2016.

PAULO, F. de S. **Brasil é terceiro país do mundo que fica mais tempo on-line no celular**. Disponível em: <http://www1.folha.uol.com.br/tec/2015/09/1679423-brasil-e-terceiro-pais-do-mundo-que-fica-mais-tempo-on-line-no-celular.shtml>. Acesso em: 26 ago. 2018.

PLURALSIGHT. **What’s the Difference Between the Front-End and BackEnd?** 2015. Disponível em: <https://www.pluralsight.com/blog/film-games/whats-difference-front-end-back-end> . Acessado em: 14 out. 2018

POWERS, S. **Aprendendo JavaScript**. São Paulo: Novatec, 2010.

PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software-8ª Edição. McGraw Hill Brasil, 2016.

PRESSMAN, S. Roger. **Engenharia de Software: Uma Abordagem Profissional. 7ª. Edição**. Rio de Janeiro–RJ. Editora McGraw-Hill, 2011.

RADEJ, Pawel. **4 major points why hybrid applications are profitable**. ECodile. 2017 Disponível em: <http://ecodile.com/2017/02/21/4-major-points-why-hybrid-applications-are-profitable>. Acesso em: 29 set. 2018.

SADALAGE, P; FOWLER, M. **NoSQL Essentials**. 1. ed. São Paulo: Novatec, 2013.

SCHROEDER, Ricardo; SANTOS, Fernando dos**. Arquitetura e Testes de Serviços Web de alto desempenho com NODE. JS E MONGODB**. Universidade do Estado de Santa Catarina, 2014.

SILVA, Maurício Samy. **HTML5 A linguagem que revolucionou a web**. São Paulo: Novatec Editora Ltda, 2010.

SOMMERVILLE, I. **Engenharia de Software. 9ª.** ed. São Paulo: Pearson Education, 2011.

SPC Brasil e CNDL. **72% dos brasileiros mudaram seus hábitos financeiros por causa da crise econômica.** Disponível em:<https://www.spcbrasil.org.br/pesquisas/pesquisa/4281>. Acesso em: 10 set. 2018.

STACK OVERFLOW. **Developer Survey 2018**. Stack Overflow. Disponível em: <https://insights.stackoverflow.com/survey/2018#technology-most-loved-dreaded-and-wanted-languages>. Acesso em: 13 set. 2018.

VENTEU, Kelly Cristina; PINTO, Giuliano Scombatti. **DESENVOLVIMENTO MÓVEL** **HÍBRIDO**. Revista Interface Tecnológica, v. 15, n. 1, p. 86-96, 2018.

WARGO, John M. **Apache Cordova 4 Programming**. Pearson Education, 2015.

XANTHOPOULOS, Spyros; XINOGALOS, Stelios. **A comparative analysis of cross-platform development approaches for mobile applications**. In: Proceedings of the 6th Balkan Conference in Informatics. ACM, 2013. p. 213-220.

1. http://blogdoaluno.utfpr.edu.br/?p=9529 [↑](#footnote-ref-2)
2. https://angular.io/guide/quickstart [↑](#footnote-ref-3)
3. Endereço Eletrônico: https://github.com/mongodb/mongo [↑](#footnote-ref-4)
4. https://material.angular.io/components/card/examples [↑](#footnote-ref-5)
5. https://material.angular.io/components/table/examples [↑](#footnote-ref-6)