

SISTEMA DE VITRINE VIRTUAL PARA SUPERMERCADO¹

Carlos Eduardo da Silva Dorneles

Graduando do 8º período do curso de Sistemas de informação do UNIPAM.

E-mail: carloses@unipam.edu.br

Juliana Lilis da Silva

Mestre em Ciência da Computação. Professora do curso de Sistemas de Informação do UNIPAM.

E-mail: juliana@unipam.edu.br

RESUMO: Este artigo apresenta o desenvolvimento de um sistema de vitrine virtual envolvendo um sistema web e uma aplicação mobile. O principal objetivo do sistema é divulgar os produtos de uma empresa do setor de supermercados, levando ao cliente maior comodidade e agilidade em suas compras e nas pesquisas de valores de produtos. Para uma maior interação com o cliente foi desenvolvido um aplicativo mobile e para a empresa foi desenvolvido um painel de controle para gerenciamento do conteúdo.

PALAVRAS-CHAVE: Vitrine virtual, aplicativo mobile, agenda de ofertas, lista de compras.

ABSTRACT: This article presents the development of a virtual showcase system involving a web system and a mobile application. The main objective of the system is to disclose the products of a company in the supermarket sector, giving the customer greater convenience and agility in their purchases and product value surveys. For more customer interaction a mobile application was developed and a control panel for content management was developed for the company.

KEYWORDS: Virtual showcase, mobile application, offers calendar, shopping list.

1 INTRODUÇÃO

O planeta terra está em constante mudança. Pode se notar que o clima e a sua geografia mudaram bastante se comparado com alguns anos atrás. Isso tudo se deve ao homem, que a cada dia explora mais e mais o planeta terra em busca de matéria-prima. Essa busca exaustiva é motivada por várias questões, tendo pontos positivos e negativos. Como ponto positivo pode-se verificar que, com a descoberta de novas matérias-primas, o homem consegue evoluir a ciência tanto quanto a tecnologia, tecnologia essa que tem pontos bastante positivos, os quais podem ser notados nos dias atuais.

¹ Trabalho apresentado na área temática IV – Sistemas de Informação: engenharia e desenvolvimento de software - do XV Congresso Mineiro de Empreendedorismo, realizado de 5 a 8 de novembro de 2018.

Com a evolução da tecnologia, os computadores, que antes eram do tamanho de casas, hoje já são tão compactos que até cabem em uma mochila escolar ou na palma da mão, no caso dos smartphones. Com a popularização dos celulares, os orelhões e até mesmo as máquinas fotográficas já estão bastante escassos, devido à drástica redução de sua utilização. As mensagens que gastavam dias para chegar ao destinatário hoje já são feitas de forma quase instantânea pela internet. Pode-se perceber que a tecnologia vem invadindo o meio em geral e as crianças, jovens, adultos e idosos estão se adaptando a ela cada vez mais rápido.

Com toda essa evolução muitas empresas começaram a se reestruturar para se adaptarem às novas tecnologias. As redes de supermercados, por exemplo, vêm investindo bastante no seu parque tecnológico a fim de se adaptarem aos novos modelos de negócios. Percebe-se também que nos últimos anos os chamados e-commerce e vitrines virtuais invadiram a internet entregando ao usuário mais comodidade e agilidade nas suas compras.

Com o intuito de disponibilizar uma ferramenta que possibilite aos supermercados uma comunicação mais efetiva com seus clientes e aproveitando desse novo modelo de negócios, foi realizado o desenvolvimento de um sistema de vitrine virtual para levar às pessoas a marca juntamente com a história da empresa, mostrando seus benefícios e vantagens a fim de conquistar mais clientes. Além disso, é disponibilizada uma plataforma onde os clientes terão maior agilidade durante suas compras, podendo os mesmos criarem lista de compras e ainda compartilhá-las com outras pessoas através das redes sociais.

Assim, foram implementados os seguintes objetivos específicos durante o desenvolvimento do sistema:

- Criar um aplicativo *mobile* e um sistema *web*, onde o cliente consegue visualizar as informações disponibilizadas pela empresa e gerenciar suas listas de compras.
- Criar um sistema *web* para painel de controle, onde o administrador da empresa pode gerenciar os dados que serão exibidos ao cliente.

Para o gerenciamento dos conteúdos que são exibidos no aplicativo *mobile*, o sistema tem um painel de controle, o qual contém os seguintes objetivos específicos:

- ✓ Criar gerenciador de pessoas: este é o local onde o administrador pode visualizar, cadastrar ou editar os dados de uma pessoa no sistema.
- ✓ Criar gerenciador de usuários: nesta função são visualizados os dados de usuários podendo cadastrar ou editar um determinado usuário. Estes são utilizados durante a autenticação e na restrição de acesso a determinadas páginas ou ações do sistema.
- ✓ Criar gerenciador de permissões de acesso: podem ser gerenciadas as permissões de usuários, podendo cadastrar, alterar ou excluir a permissão de um usuário de acessar determinadas funções do sistema.
- ✓ Criar gerenciador de setor e atributos: nesta área é realizado o cadastro, edição, exclusão e visualização de setores ou atributos, estes são vinculados aos produtos.
- ✓ Criar gerenciador de produtos: nesta área é realizado o cadastro, edição e visualização de um produto.

- ✓ Criar gerenciador de ofertas: nesta área é realizado o cadastro de agendas de ofertas, onde em uma data específica um ou mais produtos estarão com preço diferente do tradicional por determinado período. O usuário ainda pode visualizar ou editar uma agenda de oferta.
- ✓ Criar visualizador de listas de compras: nesta área a empresa pode visualizar as listas de compras de seus clientes, não podendo interferir nas mesmas. O aplicativo *mobile*, que é voltado aos clientes, teve os seguintes objetivos específicos:
 - ✓ Criar visualizador de produtos: o cliente pode visualizar os produtos, suas características e ainda adicioná-los a uma lista de compras.
 - ✓ Criar visualizador de ofertas: o cliente pode visualizar todos os produtos em oferta.
 - ✓ Criar gerenciador de listas de compras: o cliente pode criar uma ou mais listas de compras, onde o mesmo pode adicionar produtos a ela, manualmente ou pelo leitor de código de barras. As listas de compras podem ser visualizadas quando o cliente estiver online ou offline.

O sistema foi desenvolvido visando atender a empresas que não têm uma plataforma fixa de comunicação com seus clientes. Muitos supermercados utilizam redes sociais, banners, encartes, folders, entre outras formas de comunicação que nem sempre chegam a todos os clientes desejados e interessados. Essas formas são eficientes, podendo chegar até em pessoas que a empresa menos espera, mas para terem um alcance maior e um trabalho de marketing mais elaborado é necessário um local apropriado e de autoria própria para divulgação. Com uma plataforma única de comunicação, as empresas conseguirão reduzir custos, levar a informação aos clientes interessados e ainda terem uma melhor forma de divulgação de sua marca.

2 REFERENCIAL TEÓRICO

Nesta seção são abordados alguns conceitos de serviços como manipulação de imagens, leitura de código de barras e segurança. Estes conceitos foram utilizados no desenvolvimento do projeto.

2.1 SEGURANÇA

Nesta seção são abordados os métodos de segurança que foram implementados no sistema, como autenticação e permissão.

2.1.1 Autenticação

A autenticação é utilizada para verificar a identidade digital do usuário de um sistema no momento de sua solicitação de acesso. Hoje, encontram-se vários métodos de autenticação como autenticação de formulário, senha descartável, leitor biométrico, reconhecimento de voz, reconhecimento facial, entre outros, cada um com suas vantagens e desvantagens (FILHO, 2009).

A forma de autenticação utilizada neste projeto foi por autenticação de formulários, na qual as informações do usuário são armazenadas em uma fonte de dados externa, fazendo assim uma comparação das credenciais apresentadas pelo usuário com a outra já pré-definida no sistema.

Depois de autenticado, o sistema informa os dados do usuário de forma segura a fim de verificar e permitir o acesso do mesmo a funções restritas. Existem várias formas de se fazer isso e as mais conhecidas são os *cookies* e *tokens*. Neste projeto foram utilizados os *tokens*, sendo a biblioteca JWT (*Json Web Token*) a responsável por transmitir os dados da autenticação para o usuário de forma segura.

A autenticação baseada em *token* é apátrida. O servidor não mantém um registro de quais usuários iniciaram sessão ou quais JWTs foram emitidos. Em vez disso, cada solicitação ao servidor é acompanhada por um *token* que o servidor usa para verificar a autenticidade da solicitação. O *token* geralmente é enviado como um cabeçalho de autorização de adição na forma de *Bearer {JWT}*, mas pode ser adicionado no corpo de uma solicitação *POST* ou mesmo como um parâmetro de consulta. (KUKIC, 2016).

O JWT gera uma chave de segurança única para cada requisição de autenticação, também chamada de *token*, onde a mesma possui uma vida útil pré-estabelecida pela API. Com isso os dados dos usuários estão seguros, dificultando com que pessoas não autorizadas tenham acesso a eles.

2.1.2 Permissão

Permissão é um processo de verificação de quais os privilégios que o usuário tem dentro do sistema. Para isso, trabalha-se com Lista de Controle de Acesso ou ACL. A ACL é responsável por definir as permissões que um usuário tem a um determinado componente ou serviço do sistema. Para o sistema fornecer acesso a um recurso, ele deve conferir na ACL se o usuário tem privilégios de acesso ao recurso (LOOPBACK, 2018).

Esse recurso foi implementado neste projeto de forma bem objetiva e flexível para o usuário. Com isso, ele consegue definir as permissões de cada usuário do sistema, deixando-o mais seguro, visto que cada usuário terá acesso a somente o que lhe interessa.

2.2 LEITOR DE CÓDIGO DE BARRAS

Código de barras é uma representação gráfica de dados numéricos e alfanuméricos, onde sua leitura é feita a partir de um leitor de código de barras. Os dados capturados por essa leitura são compreendidos por uma função de computador que por sua vez converte-os para letras e números compreensíveis por humanos. A utilização do código de barras é muito comum na indústria que os utiliza para identificar de forma única cada tipo de produto que ela produz (NONUS, 2018).

Neste projeto o cliente pode cadastrar listas de compras e cada lista de compras pode conter vários produtos. Para que o cliente não tenha que ficar digitando os dados dos produtos ele tem a opção de utilizar o leitor de código de barras para

assim buscar os dados do produto. Quando o usuário aponta a câmera do smartphone para o código de barras, automaticamente o aplicativo irá ler o código, buscando no servidor os dados do produto e apresentando-os ao usuário.

Para o aplicativo decodificar o código lido foi utilizada uma biblioteca chamada Cordova Plugin Scanner, a qual trabalha junto com a biblioteca Cordova, que é responsável por acessar recursos nativos do dispositivo.

2.3 MANIPULAÇÃO DE IMAGENS

Há 40 anos, compartilhar uma imagem era totalmente diferente do que se vê nos dias atuais, era preciso utilizar mídias físicas como fitas magnéticas e disquetes, por exemplo. Hoje em dia tudo é mais fácil, quando o usuário troca a sua foto de perfil, publica uma imagem na sua rede social ou envia uma foto por Whatsapp o que ele está fazendo é compartilhando uma imagem, passando do seu dispositivo para um serviço da internet.

Para este projeto foi criado um serviço de upload e manipulação de imagens, sendo o mesmo desenvolvido em Node.js e fornecendo, assim, uma solução completa de gerenciamento de imagens, incluindo upload, armazenamento, otimização e entrega.

3 METODOLOGIA

O presente projeto foi desenvolvido tendo por base as técnicas da metodologia ágil, o Scrum. “O Scrum é um *framework* ágil, simples e leve, utilizado para a gestão do desenvolvimento de produtos complexos, imersos em ambientes complexos”. (SABBAGH, 2013, p. 17). Utilizando essa metodologia para desenvolvimento, o cliente pôde acompanhar de forma progressiva o andamento do projeto.

O Kanban, que também foi usado para gerenciar o desenvolvimento deste projeto, é o nome dado a um quadro cuja função é mostrar como está o andamento do projeto. Os itens do *product backlog* selecionados em uma *sprint* do Scrum eram quebrados em tarefas para que ficasse mais fácil transformá-los em um incremento para o produto. Nesse momento é que foram utilizados os *post-its*, as linhas e as raiais do quadro Kanban (MADUREIRA, 2012).

Durante o desenvolvimento do projeto foram utilizadas todas as ferramentas e linguagens especificadas a seguir:

- JavaScript é a linguagem base de todo o projeto, e foi utilizada durante e após o desenvolvimento da API e do aplicativo *mobile*.
- Node.js é uma plataforma que trabalha sobre o motor JavaScript do Google Chrome e foi utilizado durante e após o desenvolvimento da API.
- Adonis.js é um *framework* de desenvolvimento tanto para *back-end* quanto *front-end* trabalhando sobre o Node.js. Para esse projeto foi utilizado somente para o desenvolvimento da API.
- Angular é um *framework* de desenvolvimento *front-end* que é mantido pela Google. Este foi utilizado para construção do painel de controle e aplicativo *mobile*.

- Ionic é um *framework* CSS para desenvolvimento de aplicativos *mobile*. O mesmo trabalha sobre o angular e foi utilizado no desenvolvimento do aplicativo *mobile*.
- Cordova é uma ferramenta que trabalha junto ao Ionic, fornecendo funções para acesso a recursos nativos dos dispositivos.
- MySQL é uma ferramenta para gerenciamento de banco de dados e utiliza como linguagem o SQL. Este é o banco de dados utilizado no projeto.
- Visual Studio Code é o editor de texto que foi utilizado durante o desenvolvimento do sistema.
- Postman é uma ferramenta muito útil para realizar testes em API, podendo testar diversas rotas rapidamente, além de combinações de parâmetros e retornos.
- SQL Power Architect é uma ferramenta de modelagem de dados.
- MySql Workbench é uma ferramenta visual para gerenciamento de banco de dados.
- Trello é a plataforma de colaboração visual que oferece perspectiva de equipes em projetos.

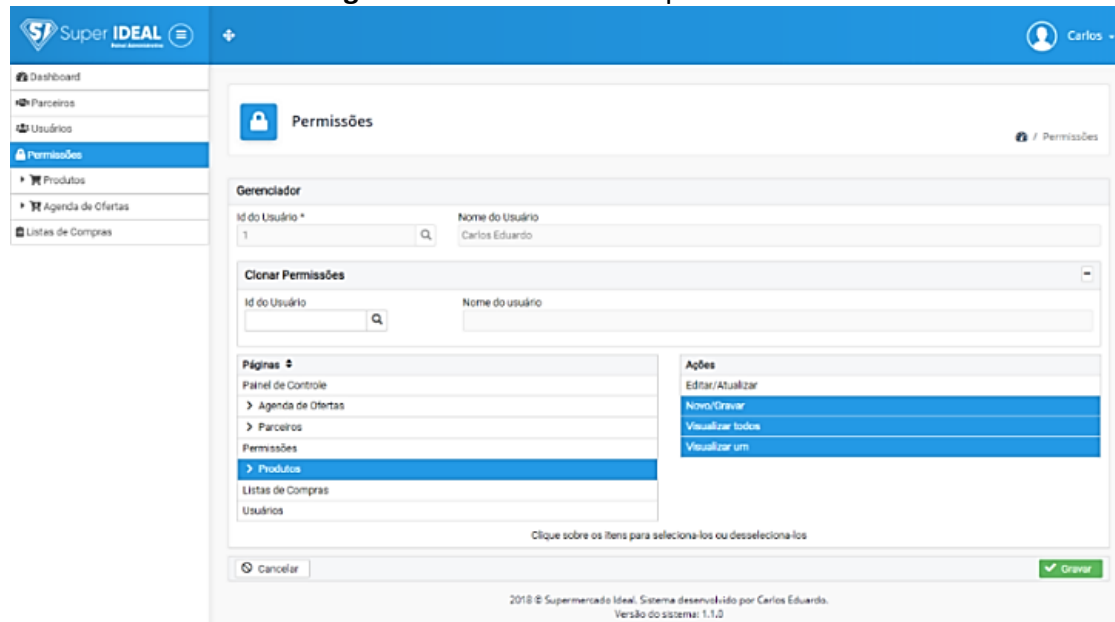
Para o desenvolvimento deste projeto, foi feito um levantamento dos problemas que um supermercado apresenta no momento de divulgar seus produtos na internet. Com isso, foram especificados os requisitos e traçados todos os passos para a criação do sistema, para que ao seu término fosse possível entregá-lo de forma a satisfazer às necessidades do cliente. O objetivo era que a cada passo do desenvolvimento a empresa pudesse passar *feedbacks* para possíveis melhorias e, assim, o mesmo ser analisado e realizadas as devidas correções.

4 DESENVOLVIMENTO E RESULTADOS

A fim de criar um sistema mais completo e robusto de vitrine virtual, desenvolveu-se um painel de controle para gerenciamento do conteúdo.

Na Figura 1 observa-se a área de controle de permissões, onde, para cada usuário do sistema, pode-se definir funções específicas em uma determinada página. Uma funcionalidade a se destacar é a possibilidade de duplicar as permissões de um usuário para outro usuário. Com isso o sistema torna-se mais flexível e seguro, pois as permissões são definidas por usuário.

Figura 1 – Gerenciador de permissões

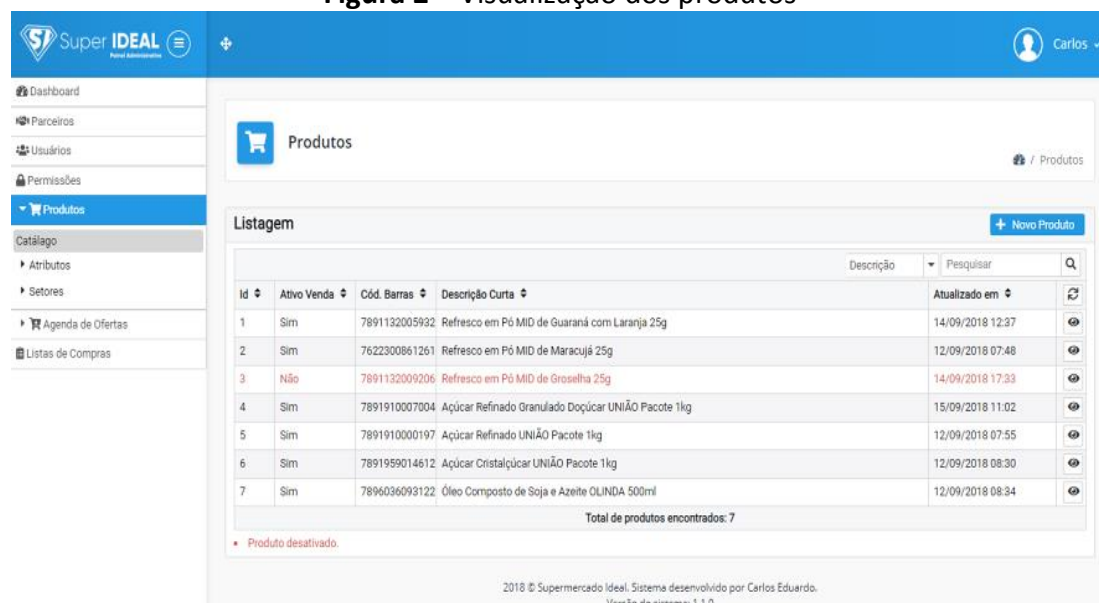


Fonte: Elaborado pelo autor

A Figura 2 representa a listagem de produtos cadastrados no sistema, na qual são apresentados alguns dados referentes ao produto com funções de pesquisa personalizada.

As páginas de usuários, parceiros, atributos e setores de produtos, tipos de atributos e setores, agenda de ofertas, tipos de agendas, motivos de oferta e listas de compras utilizam a mesma estrutura apresentada na Figura 2, mas com o conteúdo respectivo a página.

Figura 2 – Visualização dos produtos



Fonte: Elaborado pelo autor

A Figura 3 apresenta os detalhes de um determinado produto, neste espaço o usuário visualiza todos os dados referentes ao mesmo, podendo atualizá-lo se necessário. Essa ferramenta também é utilizada quando é necessário realizar um novo cadastro de produto.

As páginas de usuários, parceiros, atributos de produtos, setores de produtos, tipos de atributos, tipos de setores, agenda de ofertas, tipos de agendas, motivos de oferta e listas de compras utilizam a mesma estrutura apresentada na Figura 3, mas com os atributos respectivos a cada página.

Figura 3 – Visualização dos detalhes do produto

Super IDEAL

Dashboard

Parceiros

Usuários

Permissões

Produtos

Catálogo

Atributos

Setores

Agenda de Ofertas

Listas de Compras

Detalhes

Produtos / Detalhes

Produto #4

Cadastro

Imagens

Ativo

Cód. Barras *

Descrição Curta *

7891910007004

Açúcar Refinado Granulado Doçúcar UNIÃO Pacote 1kg

Descrição Longa

Ingredientes:

Origem vegetal: saccharose de cana de açúcar.

Tipo Embalagem *

Pacote

Preço *

R\$ 5,99

Setorização

Departamento *

Alimentos

Categoria *

Mercaria

Subcategoria

Açúcar refinado

Atributos

Grupo de Preço

Selecionar Grupo de Preço

Marca *

União

Cadastrado por Carlos Eduardo dia 12/09/2018 07:53

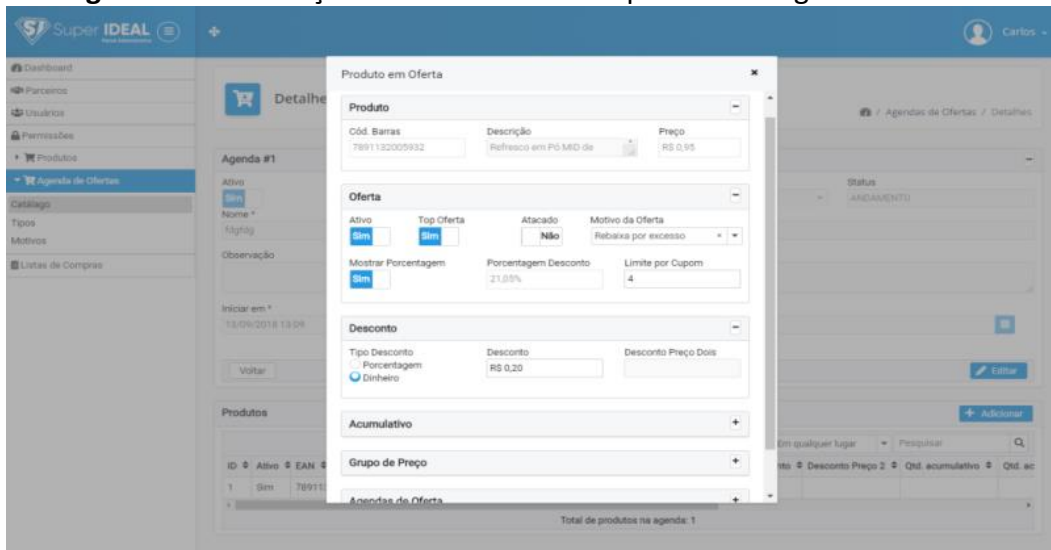
Atualizado por Carlos Eduardo dia 15/09/2018 11:02

Fonte: Elaborado pelo autor

A Figura 4 apresenta os dados de um produto em uma agenda de oferta, em que os atributos disponíveis são personalizados de acordo com o tipo de agenda.

Para tornar as agendas de ofertas mais flexíveis, o usuário pode criar tipos de agenda de oferta que possibilita definir quais atributos que um produto pode receber. Também é possível cadastrar os motivos de oferta para assim haver um controle futuro sobre os fatos que definiram que o produto fosse colocado em oferta. Outra funcionalidade interessante é a possibilidade de adicionar os produtos que pertencem ao mesmo grupo de preço na agenda de oferta, evitando assim duplicidade de dados e facilitando o cadastro do produto na agenda de oferta.

Figura 4 – Visualização dos detalhes de um produto da agenda de oferta



Fonte: Elaborado pelo autor

As figuras apresentadas anteriormente nesta seção foram desenvolvidas para que a empresa contratante pudesse gerenciar o conteúdo a ser apresentado no aplicativo mobile. As figuras apresentadas a seguir estão representando o aplicativo que o cliente da empresa contratante terá acesso.

A Figura 5 apresenta a interface principal do aplicativo, na qual são mostrados os produtos em oferta e um carrossel de imagens para divulgação. Na Figura 6 são mostradas as categorias dos produtos e na Figura 7 são mostrados os produtos de uma determinada categoria.

Figura 5 – Tela principal



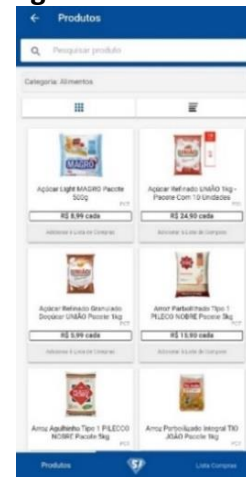
Fonte: Elaborado pelo autor

Figura 6 – Categorias



Fonte: Elaborado pelo autor

Figura 7 – Produtos

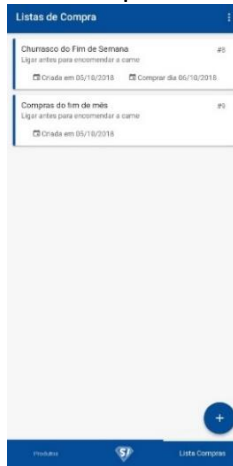


Fonte: Elaborado pelo autor

A Figura 8 apresenta as listas de compra de um usuário específico. Já a Figura 9 mostra os produtos de uma determinada lista de compras e nela o cliente consegue pesquisar um produto pelo código de barras, digitando-o ou utilizando o leitor de código de barras para poder adicioná-lo a lista. A Figura 10 apresenta os dados da

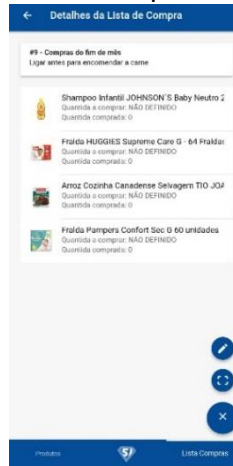
empresa como CNPJ, endereço comercial, e-mail, telefone e um mapa indicando a localização.

Figura 8 – Listas de compras



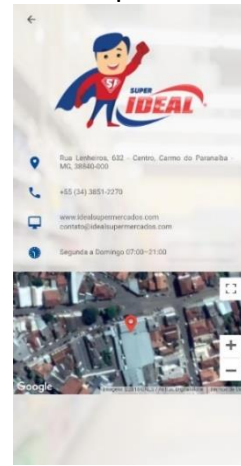
Fonte: Elaborado pelo autor

Figura 9 – Produtos da lista de compra



Fonte: Elaborado pelo autor

Figura 10 – Dados da empresa



Fonte: Elaborado pelo autor

Todas as funções que o aplicativo e o painel de controle disponibilizam para o usuário, dependem de um serviço. Este serviço recebe o nome de API e é responsável por receber as requisições e devolver algo para o usuário.

Para o aplicativo mobile ou painel de controle poder consumir os recursos que a API tem a oferecer, eles devem fazer requisições utilizando um método do protocolo HTTP. Os métodos utilizados nos recursos da API foram: GET, POST, PATCH e DELETE. Cada recurso tem um método HTTP e uma identificação única, os quais devem ser informados em cada requisição. A identificação dos recursos utilizados na API, seguiu o conceito de URI, que é um dos padrões utilizados na Web. Todo recurso devolve uma mensagem no formato JSON, que é um formato compacto e eficiente para troca de mensagens.

A Figura 11 exemplifica o conceito descrito acima com um trecho do código fonte da API do projeto. Na linha 212 pode-se perceber que o recurso utiliza o método GET do HTTP e isto significa ela apenas obtém informações existentes no servidor. A linha 213 é a identificação única do recurso e especifica qual o caminho deve ser utilizado para o mesmo ser consumo. Já na linha 214 é informado qual a função que deve ser chamada no *controller* do recurso assim que receber uma requisição.

A linha 215 indica quais os *middlewares* que devem ser chamados. Sempre que houver uma requisição de recurso, a primeira coisa a ser verificada são os *middlewares*. Se todos os *middlewares* derem um retorno verdadeiro a API busca e processa os dados, assim retornando uma mensagem. O *middleware Throttle* que está na linha 217 é responsável por limitar a quantidade de requisições que o recurso pode receber em um determinado tempo. Neste exemplo é especificado que a cada 60 segundos o recurso pode receber no máximo 30 requisições por sessão.

Na linha 218 o *middleware Auth* especifica que o recurso somente pode ser acessado por usuários que enviarem um *token* válido pelo *header*. Este *token* somente

é válido se tiver sido gerado utilizando o mesmo autenticador, neste caso é o *jwt_email*.

Na linha 219 é mostrado o *middleware Check*, que é responsável por verificar se o usuário tem permissão de acesso ao recurso. Este *middleware* recebe dois parâmetros, o primeiro representa o nome da página e o segundo representa o nome da ação. Neste caso, o primeiro parâmetro é o *shopping_list* e o segundo é o *view*.

Figura 11 – Recurso de listagem de listas de compras

```
212 Route.get(  
213   'shopping/lists/:idList/product',  
214   'Public/ShoppingLists/ShoppingListProductsController.index'  
215 ).middleware(  
216   [  
217     'throttle:30,60',  
218     'auth:jwt_email',  
219     'check:SHOPPING_LIST,VIEW'  
220   ]  
221 )
```

Fonte: Elaborado pelo autor

Todos os demais recursos utilizam a mesma estrutura, mas com os dados apropriados a cada um. Alguns recursos não precisam utilizar o *middleware Auth* e *Check*, pois eles são recursos públicos, como exemplo observa-se o recurso de autenticação do usuário.

Sempre que há algum tipo de erro na API é retornado um JSON com os dados do mesmo. Na Figura 12 é apresentada a classe que é responsável por receber o erro, fazer o tratamento do mesmo e retorná-lo ao usuário que fez a requisição. Na linha 12, um método assíncrono espera receber um dado do tipo exceção e na linha 18 é feito o tratamento do erro e logo em seguida retornado uma mensagem do tipo JSON com os dados do erro.

Figura 12 – Classe de exceção de erros

```
5 class ErrorException extends GE.LogicalException {  
6   /**  
7    * Handle this exception by itself  
8    * http://adonisjs.com/docs/4.0/exceptions#\_custom\_exceptions  
9    * throw new CustomException(message, status, code)  
10   */  
11   async handle (error, { request, response }) {  
12     switch (error.code) {  
13       case 'ER_DUP_ENTRY': error.message = 'Os dados informados estão duplicados'; break  
14       case 'ER_NO_REFERENCED_ROW_2': error.message = 'Alguns dados informados não existe na tabela'; break  
15       case 'E_MISSING_DATABASE_ROW': error.message = 'Não foi encontrado nenhum resultado com os dados informados'; break  
16     }  
17  
18     response.status(error.status).json({  
19       message: error.message,  
20       status: error.status,  
21       code: error.code ? error.code : null,  
22       name: error.name ? error.name : null  
23     })  
24   }  
25 }
```

Fonte: Elaborado pelo autor

Na Figura 13 está um exemplo de como é chamada a classe de exceção de erro. A exceção *ErrorException*, especificada na linha 60, é a classe mostrada na Figura 12, para a qual são informados alguns parâmetros, como a mensagem do erro, o status e o código.

Figura 13 – Chamada de exceção de erro

```

54     } catch (error) {
55         switch (error.name) {
56             case 'UserNotFoundException': error.message = 'E-mail ou senha inválidos'; break
57             case 'PasswordMismatchException': error.message = 'E-mail ou senha inválidos'; break
58         }
59
60         throw new ErrorException(error.message, error.status, error.code)
61     }

```

Fonte: Elaborado pelo autor

Na Figura 14 é apresentada a estrutura de uma mensagem de erro que o usuário recebe.

Figura 14 – Estrutura de uma mensagem de erro

```

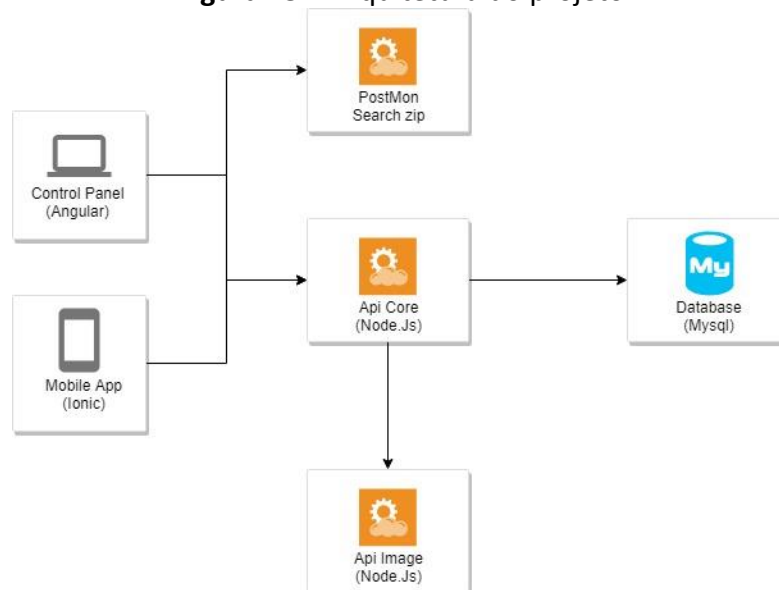
1  {
2    "message": "O token não foi informado ou não é válido",
3    "status": 401,
4    "code": "E_INVALID_JWT_TOKEN",
5    "name": "InvalidJwtToken"
6  }

```

Fonte: Elaborado pelo autor

Por fim, a Figura 15 representa a arquitetura do projeto. Nela está descrito o que foi desenvolvido e as tecnologias utilizadas.

Figura 15 – Arquitetura do projeto



Fonte: Elaborado pelo autor

Pode-se perceber na arquitetura do projeto Figura 15, que a API Core é o serviço principal, o qual é responsável por receber as requisições vindas do painel de controle e do aplicativo mobile, processá-las, fazendo consultas no banco de dados quando necessário, e devolver uma mensagem do tipo JSON ao usuário. Toda requisição do painel de controle ou do aplicativo mobile é feita diretamente na API Core. O serviço de imagem é responsável por receber as imagens, armazená-las e devolvê-las quando necessário. Este serviço é acessado pela API Core, que recebe as mensagens de retorno e, por fim, devolve-as ao usuário. O serviço *Postmon* é um serviço de terceiro, no qual o painel de controle faz uma requisição enviando um código postal e, assim, recebendo uma mensagem do tipo JSON com os dados relacionados ao código postal.

5 CONCLUSÃO

Com base no levantamento apresentado neste artigo, pode-se perceber que o sistema atende a todos os requisitos mínimos que se espera de uma vitrine virtual.

O sistema conta com vários recursos como cadastro e manipulação de dados pessoais, usuários, produtos e agendas de ofertas. O grande diferencial é a praticidade que o usuário tem para manipulá-los, pois o mesmo pode facilmente cadastrar novos atributos em cada recurso, assim, não deixando o usuário preso somente ao que o sistema disponibiliza. Outro grande diferencial pode ser encontrado na área de cadastro de agenda de ofertas, pois além dos tipos de ofertas que o sistema fornece, o usuário pode criar seu próprio tipo de oferta.

A parte de segurança da informação foi muito bem planejada e estruturada neste projeto. Se observado em outros sistemas, pode-se perceber que as permissões são distribuídas por tipos de usuários e quando implantado em grandes empresas isso pode não atender muito bem, pois mesmo que os usuários trabalhem no mesmo setor, um pode ter permissões que o outro não necessita ou não pode ter acesso. No sistema desenvolvido foi criada uma forma diferente de se atribuir permissões aos usuários, pois nele o administrador pode restringir para cada usuário somente o que ele realmente necessita ter acesso, assim não se utiliza tipos de usuários para tratar controle de acesso a funções do sistema.

Também foi criado um aplicativo mobile com suporte às plataformas Android e iOS, atendendo a uma grande parte do público mobile. Com uma interface simples e agradável, o usuário consegue facilmente navegar em suas páginas e identificar o que realmente necessita. O gerenciamento e manipulação de imagens, que foi criado especialmente para este sistema, diminui o consumo de dados do cliente ao visualizar os produtos, pois as imagens são apresentadas com uma dimensão menor, assim diminuindo o tamanho dos arquivos a serem baixados pelo usuário.

O próximo passo para aprimorar esse sistema é a criação de um website que tenha as mesmas funções do aplicativo mobile. Além disso, também serão implantadas funções de autenticação utilizando as redes sociais. Para a empresa serão implementadas funções para gerenciamento de e-mails direto do painel de controle e funções como cotação, entrada e saída de notas fiscais com integração na receita federal, controle de estoques e envio de e-mails em massa. Assim, em um futuro

próximo, o sistema deixará de ser apenas um sistema de vitrine virtual e passará a ser um sistema de e-commerce.

REFERÊNCIAS

FILHO SÓCRATES. **Segurança da Informação**: Autenticação. Agosto 2009. Disponível em <<http://waltercunha.com/blog/2009/08/19/seguranca-da-informacao-autenticacao>>. Acesso em: 13 Jun. 2018.

KUKIC, A. **Cookies vs Tokens**: The Definitive Guide. Maio 2016. Disponível em <<https://auth0.com/blog/cookies-vs-tokens-definitive-guide>>. Acesso em: 04 Mar. 2018.

LOOPBACK. **Authentication, authorization, and permissions**. Disponível em <<https://loopback.io/doc/en/lb2/Authentication-authorization-and-permissions.html>>. Acesso em: 15 Mar. 2018.

MADUREIRA, F. **O Quadro de Tarefas no Scrum**. Janeiro 2012. Disponível em <<http://blog.myscrumhalf.com/2012/01/o-quadro-de-tarefas-no-scrum>>. Acesso em: 10 Mar. 2018.

NONUS. **Leitor a laser de código de barras**. Disponível em <<http://www.nonus.com.br/leitor-laser-codigo-barras.php>>. Acesso em: 15 Mar. 2018.

SABBAGH, R. **Scrum**: Gestão ágil para projetos de sucesso, v.1, ed. Casa do código, 2013.