

Sistema Web Responsivo de Pedidos para Representantes Comerciais de uma Empresa de Fabricação e Comércio de Artefatos de Couro de Farroupilha

Ricardo Tenedini, Rogério Xavier de Azambuja

Instituto Federal do Rio Grande do Sul - *campus* Farroupilha - Av. São Vicente, 785 -
CEP 95174-274 - Bairro Cinquentenário - Farroupilha - RS - Brasil

ricardotenedini@gmail.com, rogerio.xavier@farroupilha.ifrs.edu.br

Abstract. *This course completion work students developing an ordering software for sales representatives. Seek to deal with a problem situation observed in a leather manufacturing and trading company in the municipality of Farroupilha: the dependence on a computer to fill out an order sheet, and access to e-mail to send it to the company. As a solution, a responsive Web system was developed, capable of being accessed by any device with internet access, allowing the preparation and forwarding of orders in a simplified way. It was used in the construction of Vue.js and NodeJS framework technologies, PHP and Javascript programming languages, HTML and CSS markup language, PostgreSQL database and free hosting (Heroku). As a responsive website, the system is capable of adapting to different sizes of displays (smartphone, tablet, desktop), providing convenience and practicality in managing customers and registering product orders for manufacturing.*

Resumo. *Este trabalho de conclusão de curso descreve o desenvolvimento de um software de pedidos para representantes comerciais. Procura tratar uma situação-problema observada em uma empresa de fabricação e comércio de utensílios de couro do município de Farroupilha: a dependência de um computador para o preenchimento de uma planilha de pedidos, e o acesso ao e-mail para enviá-la à empresa. Como solução, foi desenvolvido um sistema Web responsivo, capaz de ser acessado por qualquer dispositivo com acesso à internet, possibilitando a elaboração e encaminhamento de pedidos de forma simplificada. Foi utilizado na construção as tecnologias de Frameworks Vue.js e NodeJS, linguagens de programação PHP e Javascript, linguagem de marcação HTML e CSS, banco de dados PostgreSQL e hospedagem gratuita (Heroku). Como um site responsivo, o sistema é capaz de se adaptar a diversos tamanhos de displays (smartphone, tablet, desktop), proporcionando comodidade e praticidade no gerenciamento de clientes e registro de pedidos de produtos para a fabricação.*

1. Introdução

Atualmente o uso da tecnologia se faz cada vez mais presente no cotidiano das pessoas, o que tem levado a criação de aplicativos e sistemas de informação que facilitam diversas atividades da nossa rotina. Para os usuários de sistemas de informação, o conhecimento de acesso, utilização e compartilhamento de dados *on-line* vem crescendo e essas ações estão se tornando cada vez mais naturais no cotidiano. Além disso, essas ações estão sendo realizadas cada vez mais a partir de dispositivos móveis e não somente, por computadores *desktops* como eram feitas no passado recente (CAMDEN; MATTEWS, 2013 apud AMARAL; NERIS, 2015, p.89).

Um detalhe muito importante são as diferenças na usabilidade entre dispositivos móveis e *desktops*. As diferenças são tão grandes que precisamos de *design* diferente para atacar esses públicos. De acordo com (LOPES, 2013 apud FRANÇA 2015, p.77), há várias maneiras que isso pode ser feito: sites distintos para mobile e *desktop*; servidor otimizando as páginas; ou *design* responsivo para o software se ajustar à plataforma do usuário.

Através de sites responsivos, é possível fazer com que eles se adaptem a todos os tamanhos de *displays*, independente do dispositivo que esteja sendo utilizado (*desktop*, smartphones, tablets entre outros), não havendo a necessidade de criação de um mesmo site com foco para cada tipo de dispositivo (FRANÇA, 2015, p.77).

Os sistemas Web atuam de forma a facilitar os pequenos processos do cotidiano, ganhando enorme importância devido ao fato de agilizar parte da rotina de seus usuários, tornando mais práticas as atividades rotineiras, deixando de lado planilhas complexas que dependem especificamente de um computador para operá-las.

A Web é uma ferramenta dinâmica, que é capaz de acelerar exponencialmente negócios, firmar relacionamentos em todas as áreas. As empresas de diferentes magnitudes precisam ter uma visão ampliada e alavancar seu empreendimento pela Web, para evitar a perda de espaço competitivo ou até mesmo se tornar obsoleta nessa geração que evolui a cada mínimo espaço de tempo. (Costa, 2018)

Observando a necessidade de um sistema para o envio de pedidos por parte dos representantes comerciais de uma empresa de fabricação e comércio de artefatos de couro do município de Farroupilha, que dependem do computador para preencher e

enviar os pedidos, através de planilhas eletrônicas, o objetivo deste trabalho foi desenvolver um sistema Web responsivo, a fim de facilitar a criação do pedido através de qualquer dispositivo com acesso à internet e *browser*¹ atualizado, independente do sistema operacional.

Este artigo também trata a importância do desenvolvimento Web com *design* responsivo, com capacidade de ser adaptável à diferentes tamanhos de *displays* e sem abrir mão de itens de usabilidade para os usuários.

O restante do artigo está organizado da seguinte maneira: A Seção 2 especifica os objetivos; na Seção 3 o trabalho é justificado; a Seção 4 apresenta a análise e modelagem do sistema; a Seção 5 trata do desenvolvimento e tecnologias; a Seção 6 apresenta os resultados, com as funcionalidades do sistema e ilustração do uso; e por fim, a Seção 7 aborda as considerações finais.

2. Objetivos

2.1 Geral

Desenvolver um sistema Web responsivo para que os representantes comerciais de uma empresa específica possam criar e enviar on-line os seus pedidos de produtos para a fabricação.

2.2 Específicos

- Proporcionar aos representantes comerciais um sistema on-line amigável para a criação de pedidos e a manutenção do cadastro de clientes;
- Padronizar os documentos de pedidos de produtos que a empresa recebe;
- Proporcionar uma imersão tecnológica à empresa e seus representantes comerciais;

3. Justificativa

Segundo França (2015, p.75), tudo evolui, trazendo melhorias para os dispositivos que acessam a Internet. E, para garantir que um site seja usual e adaptável a 90% dos dispositivos, faz-se necessária a utilização do Web *design* responsivo.

Um site é responsivo quando automaticamente se encaixa no dispositivo do usuário (smartphone, desktop, tablet). O site responsivo muda a disposição e aparência

¹ Programa de computador que torna possível ler informações na Internet, acessar sites, páginas ou ter acesso aos recursos disponibilizados pela Web. É um programa que executa linguagens cliente-side (lado cliente) diretamente no computador do usuário.

de seu conteúdo baseado no tamanho do monitor do qual está sendo feito o acesso. (NISKA, 2014, apud AMARAL e NERIS, 2015, p. 89),

O desenvolvimento de um sistema Web envolve a implementação da (I) lógica, a qual está atrelada a função pela qual o sistema foi desenvolvido; (II) a interface, meio visual do qual o usuário interage com o sistema, e (III) estrutura navegacional, organização/disposição dos elementos visuais (botões, informações, etc.) do sistema. (DIAS et al, 2010, p. 43).

Conforme Kalbach (2009, p.26) “a fundação para todos os sites Web, HTML (e seu primo mais restrito XHTML) são linguagens de programação básicas para criar mecanismos de navegação e a estrutura de um site Web”. Todos os códigos que são enviados de um servidor para um navegador pessoal são representados pelas tecnologias *front-end*. Desta forma, o navegador que recebe estes códigos o interpreta e o desenha na tela. Essas tecnologias são constituídas pelo Cascading Style Sheets (CSS), JavaScript, Frames, Flash, XForms e o Ajax.

Wike (2005 apud REIS, 2007, p.6) destaque que:

[...]utilizar padrões para a Web é extremamente vantajoso, pois proporciona um maior controle sobre a página. Quando é dito que uma página é compatível com os padrões, significa que o documento consiste de HTML ou XHTML válido, utiliza CSS para layout, é bem-estruturado e semanticamente correto. Esses fatores podem garantir que o site seja acessado por qualquer dispositivo, seja ele móvel, tátil, desktop etc.

Na ciência da computação, o *front-end* e o *back-end* são termos generalizados que condizem com às etapas inicial e final de um processo. Para Morrison (2008, apud Amaral e Neris, 2015, p. 89), o “*front-end* é responsável por coletar a entrada em várias formas do usuário e processá-la para adequá-la a uma especificação em que o *back-end* possa utilizar”.

O *back-end* executa as regras de negócio da aplicação, ele é responsável por validar e consistir os dados vindos do front-end. *Front-end* é uma espécie de interface entre o usuário e o back-end. Ambos podem estar distribuídos entre um ou mais sistemas. (MORRISON, 2008, apud Amaral e Neris, 2015, p. 89)

Com a intenção de agilizar e padronizar a forma que os representantes comerciais de uma empresa de fabricação e comércio de artefatos de couro realizam os pedidos,

surgiu a necessidade de criar um sistema para atender tal demanda. Para de fato ser um sistema prático, optou-se por ser um sistema Web responsivo e desta forma, ele pode ser acessado de qualquer dispositivo, independente do sistema operacional utilizado (Windows, Linux, Android, IOS, entre outros).

No sistema desenvolvido, o representante comercial poderá gerenciar os seus clientes e criar o pedido (selecionando o cliente, os produtos e as demais informações cadastradas pela empresa). Após criar o pedido, ele poderá salvá-lo para que a empresa consiga visualizar on-line diretamente no sistema.

No módulo da empresa, poderá ser realizado o gerenciamento dos seus representantes comerciais, dos produtos que são fabricados e dos clientes. Esses são os cadastros básicos e obrigatórios para que os representantes consigam acessar o sistema para realizar os pedidos. Além dos cadastrados, em seu módulo do sistema, a empresa poderá gerenciar todos os pedidos realizados pelos representantes.

4. Análise e Modelagem

Para obter as definições necessárias à construção do sistema de pedidos, foram realizadas reuniões com o responsável da empresa, nas quais foi executado o levantamento e identificação de requisitos, e a definição do escopo do sistema. Estes requisitos foram classificados como funcionais, não funcionais, de sistemas e de negócio. Conforme Sommerville (2011), os requisitos: (I) “funcionais representam o modo no qual o sistema deve se comportar em determinada situação”; (II) “não funcionais representam as restrições aos serviços ou funções oferecidos pelo sistema”, (III) “de sistema são descrições mais detalhadas das funções, serviços e restrições operacionais do sistema de software”. E devido as características e necessidades, foi definido o desenvolvimento para plataforma Web.

Foi possível fazer o levantamento de alguns requisitos funcionais, não funcionais e de sistema, conforme apresentado nas tabelas a seguir.

Tabela 1: Relação de Requisitos Funcionais

Requisitos Funcionais		
Código	Nome	Detalhamento
Módulo Empresa		
RF1	Manter representantes	O usuário poderá gerenciar todos os representantes comerciais no acesso ao sistema.
RF2	Manter produtos	O usuário precisará cadastrar e dar manutenção dos produtos fabricados.
RF3	Manter clientes	O usuário poderá cadastrar e gerenciar toda a carteira de clientes.
RF4	Manter pedidos	O usuário poderá gerenciar os pedidos e alterar o <i>status</i> (aberto, confirmado, concluído, cancelado).
Módulo Representante		
RF5	Manter clientes	O usuário poderá cadastrar e gerenciar a carteira de clientes.
RF6	Criar pedidos	O usuário poderá criar pedidos para fabricação.
RF7	Manter Pedidos	O usuário poderá gerenciar os pedidos e alterar o <i>status</i> (aberto, confirmado, concluído, cancelado)

Fonte: Produção Própria

Tabela 2: Relação de Requisitos Não Funcionais

Requisitos Não Funcionais		
Código	Nome	Detalhamento
RNF1	Usabilidade	O sistema deverá ser de fácil utilização, aprendido e executado.
RNF2	Segurança	O sistema deverá ser seguro (nos acessos, armazenamento e comunicação).
RNF3	Portabilidade	O sistema deverá funcionar em diferentes dispositivos.

Fonte: Produção Própria

Tabela 3: Relação de Requisitos de Sistema

Requisitos de Sistemas		
Código	Nome	Detalhamento
RS1	Interface de login	O sistema precisará ter interface de login para validar o acesso do representante
RS2	Armazenamento	O sistema precisará salvar no banco de dados os dados dos clientes e produtos para que o representante consiga criar/fazer o pedido
RS3	Informar cliente no pedido	Não será possível enviar o pedido se o cliente não estiver selecionado
RS4	Informar produto e quantidade	Não será possível enviar o pedido se não tiver produto e quantidade adicionados ao pedido

Fonte: Produção Própria

Em termos gerais, o sistema de pedidos contempla em seu escopo: a empresa cadastra os representantes, os produtos e seus materiais e cores. O representante tem acesso ao cadastro e manutenção da sua carteira de clientes para qual fará a vendas dos produtos fabricados pela empresa. Para criar o pedido é necessário verificar se há o cliente cadastrado no sistema, caso não haja, o representante obrigatoriamente precisará fazer o cadastro. Em novo pedido, o representante comercial informa a data de entrega, o cliente, a transportadora e principalmente adicionar os produtos (com matéria-prima, cor, valor unitário, quantidade e observação). Ao concluir o pedido é possível revisá-lo para confirmar se está tudo certo para a produção, caso seja necessário, o pedido poderá ser editado tanto pelo representante, quanto pela empresa. O sistema manterá todas as informações salvas no banco de dados, no qual, em seu login, a empresa visualizará os pedidos em aberto para produção e também fará a manutenção de todos os clientes.

A seguir, na Figura 1, é apresentado o Diagrama de Casos de Uso para um melhor entendimento e apresentação do funcionamento do sistema e seus requisitos identificados.

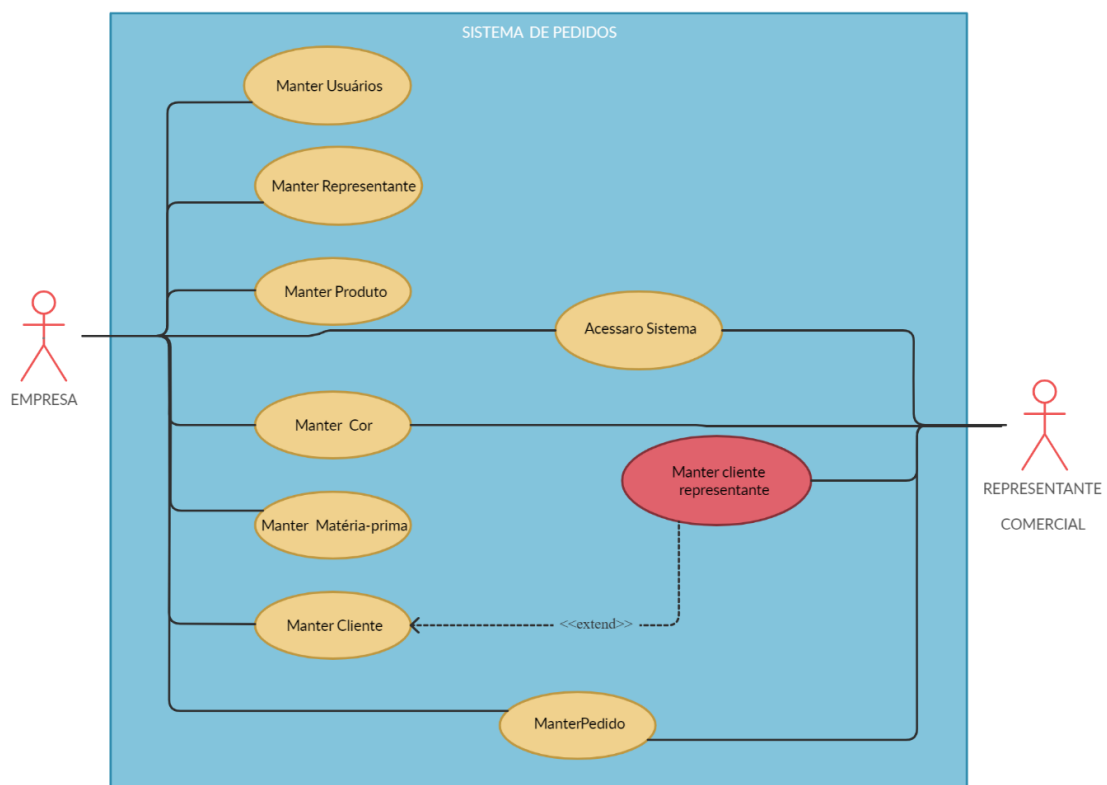


Figura 1: Diagrama de Casos de Uso do sistema de pedidos
Fonte: Produção Própria

No escopo do sistema foram identificados dois atores, Empresa e Representante Comercial. A seguir, é apresentada uma breve descrição dos casos de uso.

Manter Usuários Após efetuar o logon no sistema, a empresa pode gerenciar os usuários, informando o e-mail que servirá como login, a senha e o perfil (empresa ou representante).

Manter Representante: A empresa gerencia todos os representantes nesta interface, informando o nome completo e vinculando com a empresa e com o e-mail de login já criado anteriormente.

Manter Produto: Para poder criar os pedidos, é necessário ter produto para vender. Aqui são cadastrados pela empresa, todos os produtos do catálogo que serão oferecidos aos clientes, cadastrando o nome e referência de cada um.

Manter Cor: Dependendo o tipo do produto vendido, pode-se escolher uma infinidade de matérias-primas (couro, tecido, entre outros) com cores bem distintas. Aqui, não é possível apenas deixar algumas cores cadastradas, tanto a empresa, quanto o representante, poderão cadastrar novas cores.

Manter Clientes: A empresa sempre terá disponível a sua carteira de clientes, podendo gerenciá-los, cadastrar novos e visualizar todos os que os representantes comerciais adicionaram.

Manter Cliente do Representante: O representante comercial pode cadastrar e dar manutenção à carteira de clientes.

Manter Matéria-prima: Um mesmo produto (mochila, carteira, etc.) pode ser fabricado com diferentes matérias-primas, sendo assim, constantemente a empresa necessitará dar manutenção e disponibilizar aos seus representantes, os materiais (couros, tecidos, sintéticos) disponíveis para a fabricação dos produtos.

Manter Pedido: Esta é a parte principal do sistema, aqui o representante cadastra todos os dados que serão visualizados pela empresa para a fabricação e envio dos produtos aos clientes. É nesta interface que a empresa gerencia os pedidos e altera o *status* (aberto, confirmado, concluído, cancelado) conforme necessário.

Tendo a definição, classificação e representação dos requisitos validados pela empresa no diagrama de casos de uso, foi elaborado o modelo entidade e

relacionamento (ER) apresentado na Figura 3, tendo como base, a planilha (Figura 2) utilizada pelos representantes comerciais da empresa.


		CLIENTE:				PEDIDO - No.:				
		ENDEREÇO:								
		CIDADE:				ESTADO:		CEP:		
		CNPJ:				INSC./ESTADUAL:				
		CONDIÇÕES:				Fone/Fax:				
		Data/Entrega:				E-mail:				
Transportadora:				Marca:						
	QUANT	COURO	REF.	PT	PIN				PREÇO	TOTAL
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
TOTAL PEÇAS									TOTAL	
DATA:				COMPRADOR:						
OBSERVAÇÕES:										

Figura 2: Planilha utilizada pelos representantes
Fonte: Produção Própria

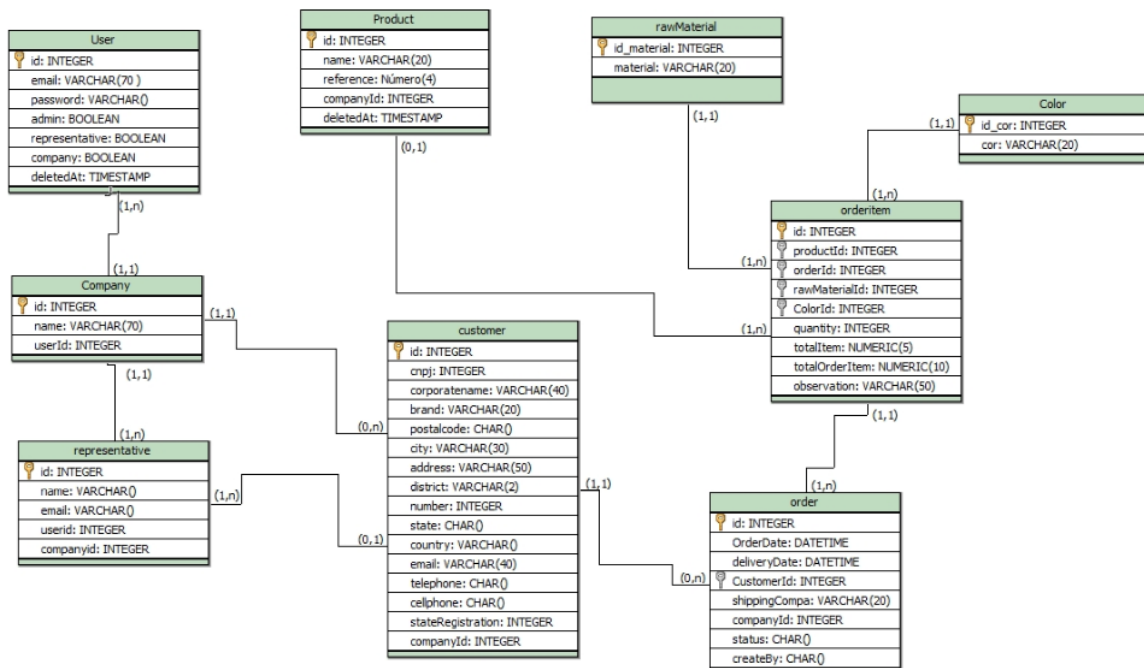


Figura 3: Entidade e relacionamento - Banco de dados
Fonte: Produção Própria

A partir da planilha utilizada pelos representantes e o modelo ER, foi construído um protótipo da interface de pedidos (Figura 4), além disso (não ilustradas), foram criadas as interfaces de manutenção de clientes para os representantes comerciais e a empresa, e também, gerenciamento de produtos e representantes somente para a empresa.

Ref	Qtd	Couro	Cor	Preço Uni	Total
112	30	Cobra	Marron	R\$ 8	R\$ 240
030	100	Sintético	Rosa	R\$ 6	R\$ 600
055	75	Couro	Preta	R\$ 5	R\$ 375
012	200	Couro	Marron	R\$ 9	R\$ 1800

Figura 4: Interface para realização de pedidos
Fonte: Produção Própria

5. Desenvolvimento

Conforme descrito na seção 2 (objetivos), a premissa para o Sistema de Pedidos deste trabalho, é que funcione em diferentes plataformas através da Web. Sendo assim, o desenvolvimento do sistema se baseia em um conceito/tecnologia mais recente, o PWA (*Progressive Web App*).

Para a desenvolvimento do projeto, foi utilizado o modelo cascata que, conforme Sommerville (2011, p. 19), “esse modelo considera as atividades fundamentais do processo de especificação, desenvolvimento, validação e evolução”, representadas por fases específicas, como: especificação de requisitos, projeto de software, implementação, teste e assim por diante.

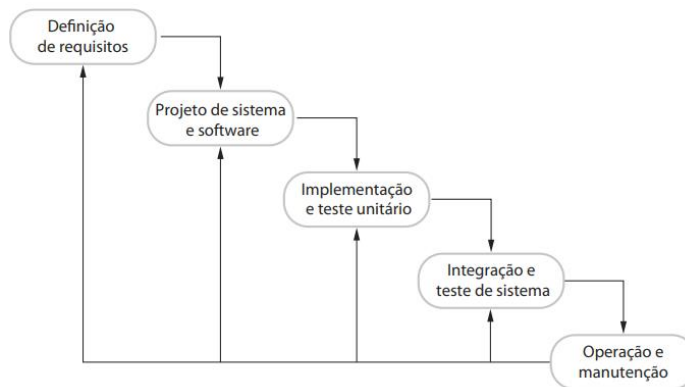


Figura 5: O Modelo em Cascata
Fonte: Somerville (2011, p. 20)

A seguir, uma breve descrição das ferramentas e tecnologias que compõem o desenvolvimento do sistema de pedidos será apresentada.

5.1 Ferramentas e tecnologias

O sistema de pedidos exige que o usuário esteja conectado à internet no momento do uso, isso é necessário para que haja comunicação com todos os serviços da nuvem. A arquitetura do sistema, representada pela Figura 6, mostra a estrutura fundamental do sistema, apresentando os elementos que compõem o sistema e o relacionamento entre eles.

Foram utilizadas diversas tecnologias para atender as necessidades específicas do desenvolvimento do sistema. Todas essas tecnologias, demonstradas na arquitetura, serão detalhadas nos tópicos subsequentes.

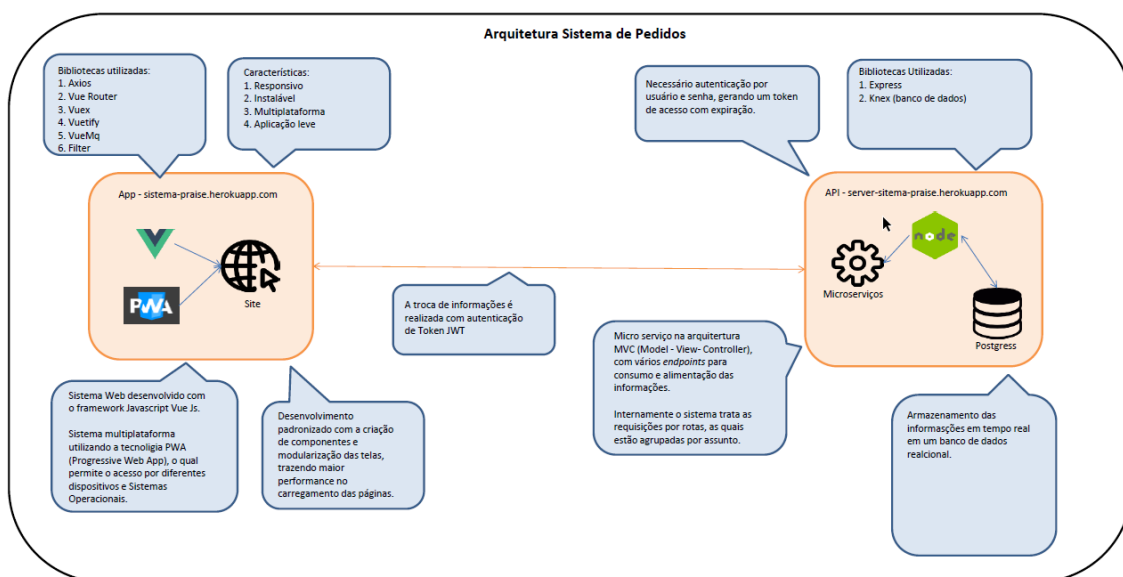


Figura 6: Arquitetura do sistema
Fonte: Produção Própria

5.1.1 Heroku

O Heroku (Heroku, 2020), uma plataforma em nuvem como um serviço (PaaS – Platform as a Service), do qual suporta várias linguagens de programação, ficou responsável por ser o servidor da aplicação, hospedando o *front-end*, o *back-end* e o banco de dados. Esta plataforma é capaz de prover a disponibilidade da aplicação em qualquer momento que for requisitado o acesso. Por se tratar de um serviço em nuvem, permite a escalabilidade de recursos sem afetar os acessos e requisições, independente do volume de acessos.

Todos os aplicativos Heroku são executados em uma coleção de contêineres leves do Linux chamados de Dynos. Um único dyno pode atender a milhares de solicitações por segundo, mas o desempenho depende muito da linguagem e da estrutura utilizada. Ambientes multiencadeados ou orientados a eventos como Java, Unicorn, EventMachine e Node.js podem lidar com muitas solicitações simultâneas.

5.1.2 Vue.js

No desenvolvimento do *front-end* do Sistema de Pedidos foi utilizado o *Vue.js*, um *Framework JavaScript Progressivo* para construção de interfaces de usuário. A sua biblioteca principal é focada exclusivamente na camada visual (*view layer*), sendo fácil adotar e integrar com outros projetos ou bibliotecas existentes. Baseado em componentes, dos quais são construídos utilizando HTML², esse *framework* utiliza o conceito *Virtual DOM* (Documento Virtual). Em páginas convencionais, a estrutura não sofre alteração entre as requisições após ser carregada, ou seja, ela permanece imutável, enquanto no conceito *Virtual DOM* o documento é mutável, permitindo que alguns que componentes sofram alterações sem a necessidade de carregar toda a página (VUEJS, 2020).

Algumas bibliotecas foram utilizadas no *Vue.js* para implementar funcionalidades específicas no sistema, as principais são:

² HTML significa HiperText Markup Language, em tradução direta, Linguagem de Marcação de Hipertexto. É a linguagem base da internet e é utilizada na construção de páginas *Web* e interpretada por browsers (navegadores).

- *Axios*: Utilizado para consumir e exibir dados de APIs³. Ele é um cliente HTTP baseado em Promises. O sistema desenvolvido utiliza requisições *Rest* e trata as respostas em *Json*.
- *Vue-router*: Roteador oficial do *Vue.js*. É ideal para criação de rotas simples.
- *Vuex*: Ele é padrão de gerenciamento de estado para aplicativos *Vue.js*. “Ele serve como um *store* centralizado para todos os componentes em uma aplicação, com regras garantindo que o estado só possa ser mutado de forma previsível” (VUEX, 2020).
- *Vuetify*: Biblioteca/Framework responsivo em *Vue* de UI (*User Interface*), ou em português interface do usuário, baseado no Material Design (VUETIFY, 2020). Ele tem suporte ao *Progressive Web App (PWA)*, também utilizado no sistema.
- *VueMq (MediaQuery)*: Puglin baseado na API *matchMedia*, detectando mudanças no tamanho da tela, ele reduz a complexidade em lidar com consultas de mídia e design responsivo.
- *Filter*: Criação de filtros personalizados, para aplicações de formatações de texto corriqueiras.

5.1.3 Node.js

No desenvolvimento do *back-end* foi utilizado o *Node.js*, um motor de execução para aplicações *JavaScript* projetado para construir aplicativos de rede escalonável. A cada conexão, o retorno de chamada é disparado, mas se não houver trabalho a ser feito, o *Node.js* ficará inativo (NODEJS, 2020). No projeto, o motor do *Node.js* serviu para a execução do *NPM (Node Package Manager)*, ou em português, Gerenciador de Pacotes do *Node*. Ele conta com uma interface de linhas de Comando (CLI) onde é possível, de forma prática, instalar, remover e atualizar pacotes, facilitando assim, a sua atualização futura.

5.1.4 Microsserviços

Derivada da SOA (do inglês *Service Oriented Architecture*), o microsserviço é uma arquitetura com a finalidade de projetar aplicativos de software como conjuntos de

³ API refere-se ao termo em inglês “Application Programming Interface” que significa em tradução para português “Interface de Programação de Aplicativos”. É um conjunto de normas que possibilita a comunicação entre plataformas através de uma série de padrões e protocolos.

serviços implantáveis independentemente. Nesse estilo de arquitetura “existem certas características comuns em torno da organização em torno da capacidade de negócios, implantação automatizada, inteligência nos terminais e controle descentralizado de linguagens e dados” (FOWLER e LEWIS, 2014).

Em suma, o estilo de arquitetura de microsserviço é uma abordagem para desenvolver um único aplicativo como um conjunto de pequenos serviços, cada um executando em seu próprio processo e se comunicando com mecanismos leves, geralmente uma API de recurso HTTP. Esses serviços são desenvolvidos em torno de recursos de negócios e podem ser implantados de forma independente por meio de máquinas de implantação totalmente automatizadas. Há um mínimo de gerenciamento centralizado desses serviços, que podem ser escritos em diferentes linguagens de programação e usar diferentes tecnologias de armazenamento de dados. (FOWLER e LEWIS, 2014)

5.1.5 Heroku Postgress

O Heroku Postgress (Heroku, 2020) foi a opção para o banco de dados SQL, por estar em *cloud* e ser gerenciado como serviço, fica disponível para integração e é ideal para aplicativos de escalonamento. Uma das vantagens é que não é necessário se preocupar com a instalação e manutenção de um servidor de banco de dados, por ser um serviço, a proteção contínua ou aplicação de *patch* de segurança mais recente fica a cargo da plataforma. O PostgreSQL é um gerenciado de banco de dados relacional e de código aberto mais popular do mundo (PostgreSQL, 2020).

Para este sistema, foi escolhido o plano gratuito *Hobby Tier* do Heroku Postgress, o qual permite no máximo 20 conexões simultâneas e 1 GB de armazenamento, tendo também limitação de performasse e não há suporte para “consultas caras”, que gastam uma quantidade significativa de seu tempo de execução lendo e gravando no disco. A vantagem é que, se em algum momento for necessário mais recurso a nível de banco de dados, pode-se aderir a planos melhores (*Standard Tier, Premium Tier, Private Tier* ou *Shield Tier*).

5.1.6 Visual Code

Foi utilizado o software Visual Studio Code como Ambiente de Desenvolvimento Integrado (IDE – *Integrated Development Environment*) para o desenvolvimento do sistema. O Visual Studio Code é um editor de código redefinido e otimizado para construir e depurar aplicativos modernos da web e nuvem, possui controle Git⁴ incorporado e é gratuito (VSCODE, 2020).

5.1.7 PWA - *Progressive Web App*

Conforme descrito no início deste capítulo, o conceito utilizado para o desenvolvimento do sistema para que de fato fosse responsivo e multiplataforma foi o PWA, pois além de funcionar como um sistema Web comum para *browsers desktops*, quando acesso através de *smartphone* ou *tablet*, ele se comporta como se fosse um aplicativo *mobile*.

Segundo SILVIA e TIOSSO (2020, p. 54), o conceito Progressive Web Application (PWA) foi criado com o objetivo de utilizar tecnologias voltadas para o desenvolvimento *web* e inseri-las no ambiente de desenvolvimento *mobile* fazendo do navegador o motor de execução, desta forma buscando independência do sistema operacional por parte da aplicação.

Lima (2019, apud SILVA e TIOSSO, 2020, p. 56) apresenta a PWA como:

[...]um conceito de desenvolvimento progressivo no qual se utilizam algumas linguagens bases: HTML, CSS e JS como principais, podendo fazer uso de alguns frameworks para o auxílio do desenvolvimento da aplicação. Em prática, é um site em modo *mobile*, mas com uma complexidade um pouco maior, visto que existem algumas características específicas para definir a PWA.

Abaixo estão listadas algumas das principais características supracitadas:

- *Service Worker*: especificado pela W3C, ele é um script que o navegador executa em segundo plano, independente da página web. Através dele é

⁴ Git é um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software. Com ele podemos criar todo histórico de alterações no código e facilmente voltar para qualquer ponto para saber como o código estava naquela data.

possível implementar recursos offline, como organizar o cache dos arquivos estáticos da página *web*, assim como sincronização de dados em segundo plano e recursos de notificações *push*.

- *Status 200*: O código *HTTP 200* é um *status* de retorno que indica que a requisição foi um sucesso.
- Utilização de protocolo *HTTPS*: Como é fundamental que qualquer aplicação necessite de segurança, torna-se obrigatório o uso do protocolo seguro *HTTPS*.
- Todo o tráfego *HTTP* deve ser redirecionado para *HTTPS*.
- Carregamento rápido em rede de internet móvel. O tempo de resposta de uma aplicação é fundamental para uma experiência positiva por parte do usuário.
- *Splash Screen* personalizado: é a interface de abertura, onde aparece uma inicialização com a logotipo da aplicação enquanto o sistema é carregado.
- Notificação para “instalar” o *App*: Com o objetivo de disponibilizar facilidade no acesso ao “aplicativo”, o usuário deve receber uma notificação se deseja instalar o aplicativo em seu dispositivo, assemelhando-se a um aplicativo nativo.

6. Resultados

Nesta seção serão apresentados os resultados do desenvolvimento, mostrando as funcionalidades do sistema e ilustrações do uso.

6.1 Login, interface inicial e perfis

Ao acessar o sistema pelo navegador, a primeira interface carregada será a de login. Na figura 7 estão apresentadas as interfaces de login do sistema acessadas de diferentes dispositivos. Na autenticação é feita a validação do usuário e senha cadastrados, onde obtendo sucesso, um token é atribuído. Em caso de entrada inválida é

apresentada a mensagem “E-mail/Senha inválidos!”. Qualquer troca de informações, independente da interface em que está, é realizada com autenticação criptografada de *token* JWT (JSON Web Token).

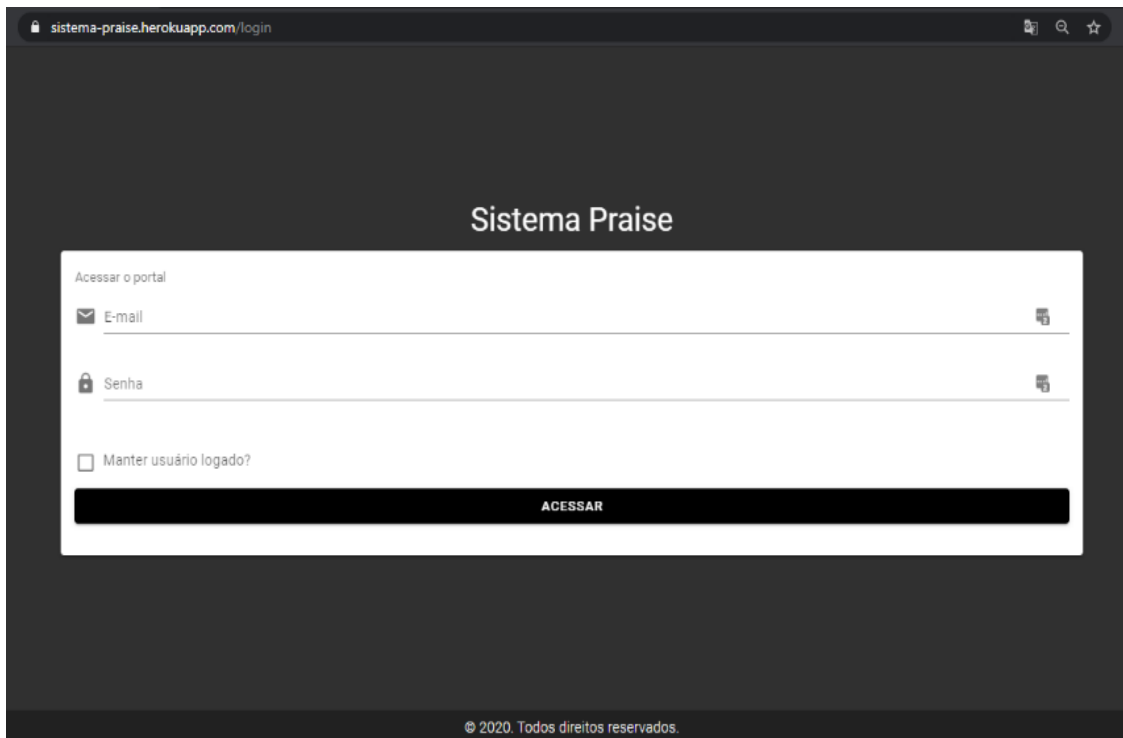


Figura 7: Interface de login no computador e mobile
Fonte: Produção Própria

Validado o acesso, o sistema será aberto com a interface inicial, a qual pode ser observada na figura 8. Na parte superior da interface mostra o usuário autenticado e a esquerda o menu com todas as opções liberadas de acordo com o perfil do usuário. A parte central ao abrir o sistema está vazia ainda, mas é um espaço para a empresa colocar um banner com alguma promoção especial ou aviso.

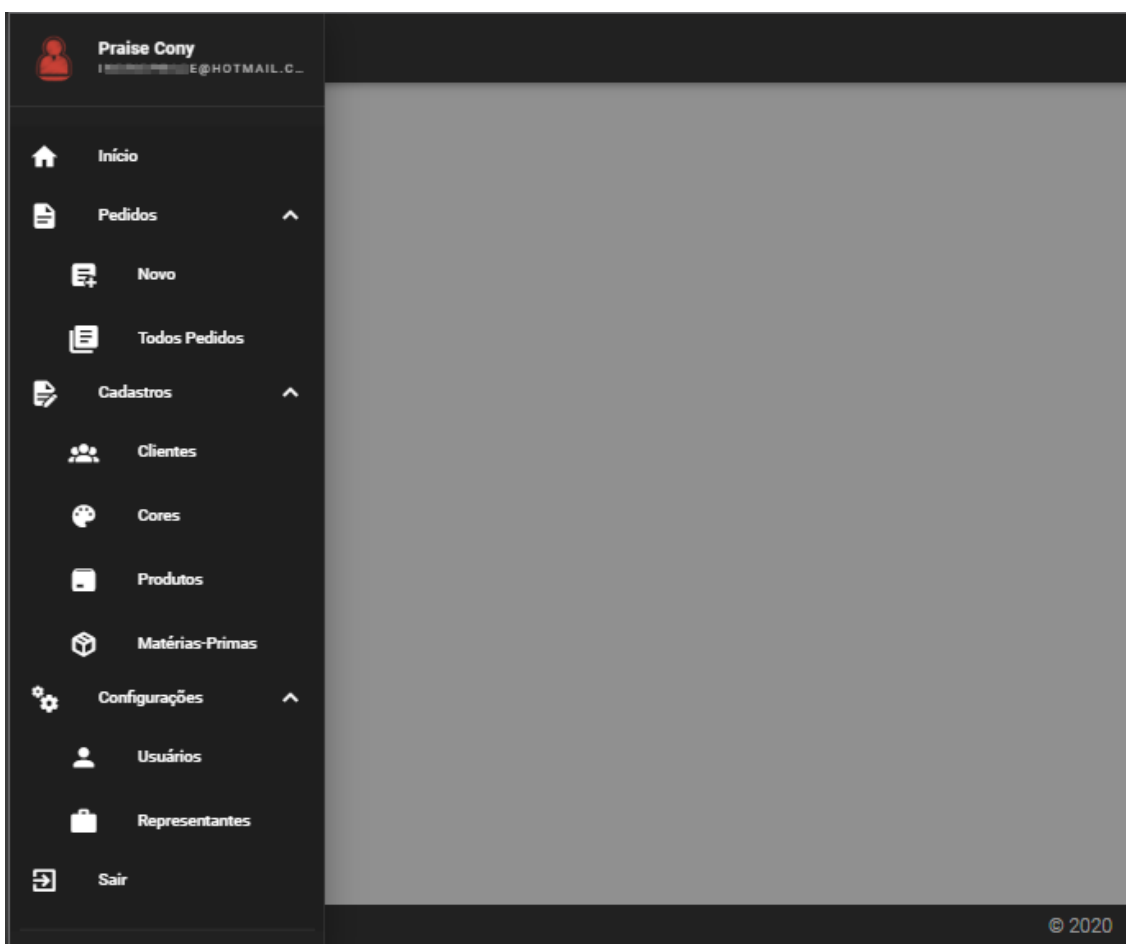


Figura 8:Interface inicial do sistema
Fonte: Produção Própria

Ao criar um usuário para o sistema, é possível vincular ele a um tipo de perfil, “Empresa” ou “Representante”. A tabela abaixo compara as funcionalidades dos acessos de cada perfil:

Tabela 4: Tabela de funcionalidades por perfil

Categoria	Funcionalidade	Perfil Empresa	Perfil Representante
Pedidos	Novo Pedido	SIM	SIM
	Visualizar Todos os pedidos	SIM	SIM
Cadastro	Cientes	SIM	SIM
	Cores	SIM	SIM
	Produtos	SIM	NÃO
	Matérias-Primas	SIM	NÃO
Configurações	Criar Usuários	SIM	NÃO
	Vincular Representantes	SIM	NÃO

Fonte: Produção Própria

As imagens a seguir mostram a diferença do menu de acordo com o perfil. Na primeira imagem é possível visualizar todas as funcionalidades do usuário com o perfil da empresa, enquanto na segunda, com o perfil representante, os acessos são mais limitados.

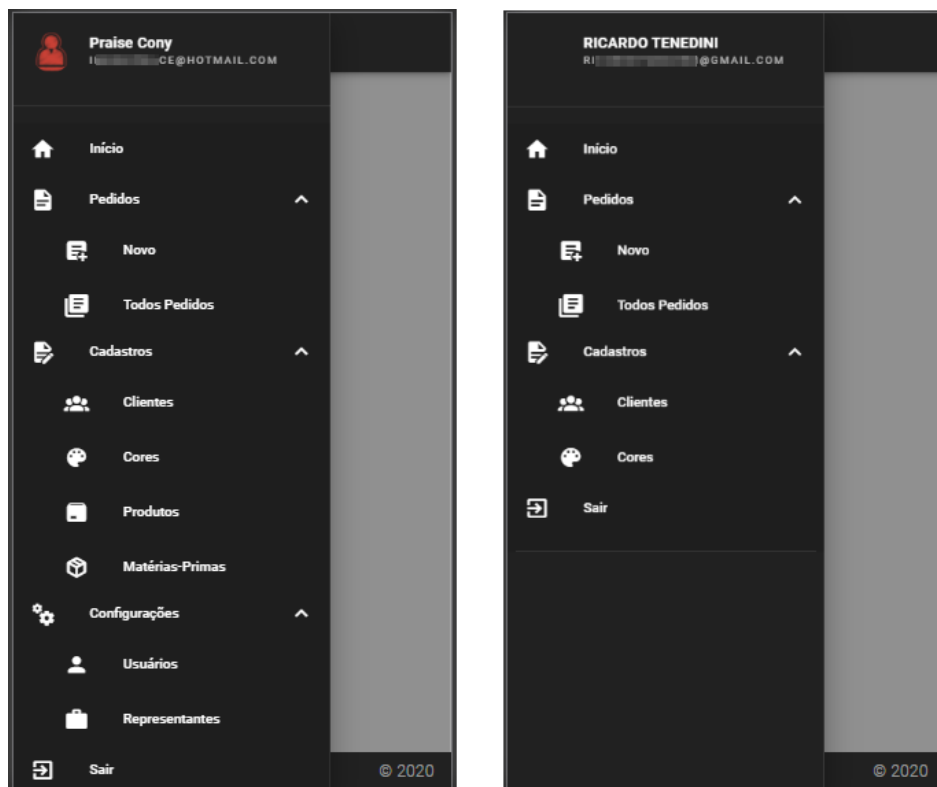


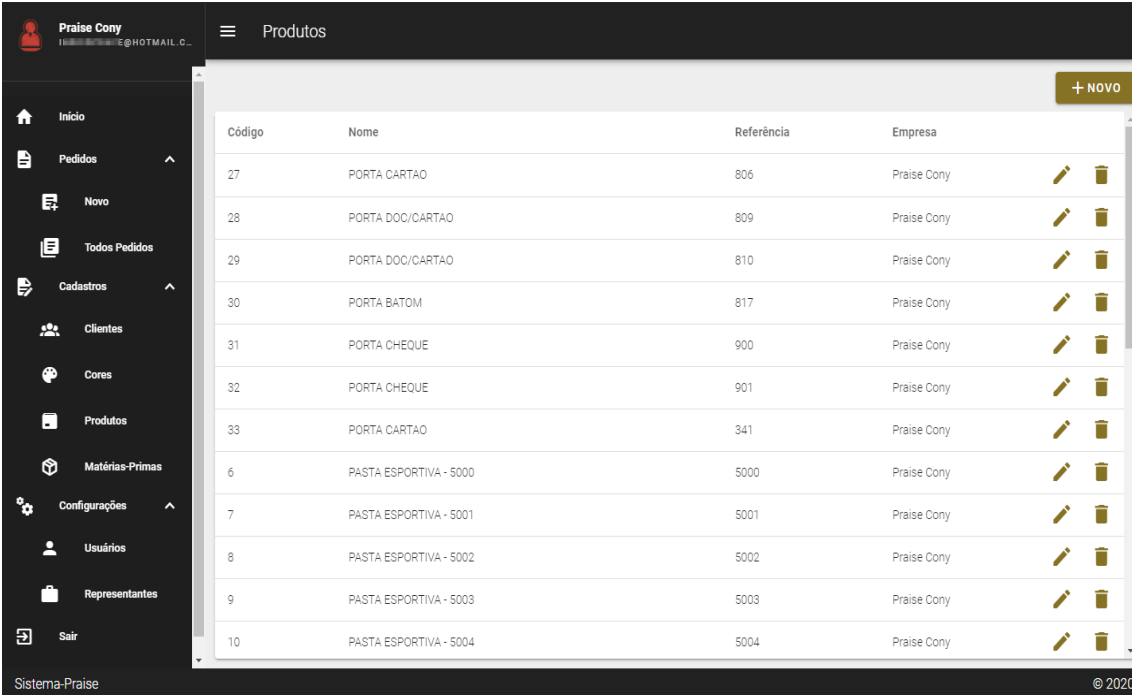
Figura 9: Diferença de menus de acordo com o perfil

Fonte: Produção Própria

6.2 Cadastros

Antes dos representantes comerciais poderem utilizar o sistema para fazerem os pedidos, é necessário que o sistema tenha disponível os produtos fabricados pela empresa, as matérias-primas, as principais cores e os clientes. Conforme apresentado na tabela de funcionalidade anteriormente, o cadastro de produtos e matérias-primas é exclusivo da empresa.

Ao clicar em “Produtos”, aparecerá no canto superior direito o botão “+ NOVO”, clicando nele serão solicitados o nome e referência do produto fabricado. Após o cadastro dos produtos, a lista completa deles irá aparecer (figura 10).



Código	Nome	Referência	Empresa		
27	PORTA CARTAO	806	Praise Cony		
28	PORTA DOC/CARTAO	809	Praise Cony		
29	PORTA DOC/CARTAO	810	Praise Cony		
30	PORTA BATOM	817	Praise Cony		
31	PORTA CHEQUE	900	Praise Cony		
32	PORTA CHEQUE	901	Praise Cony		
33	PORTA CARTAO	341	Praise Cony		
6	PASTA ESPORTIVA - 5000	5000	Praise Cony		
7	PASTA ESPORTIVA - 5001	5001	Praise Cony		
8	PASTA ESPORTIVA - 5002	5002	Praise Cony		
9	PASTA ESPORTIVA - 5003	5003	Praise Cony		
10	PASTA ESPORTIVA - 5004	5004	Praise Cony		

Figura 10: Cadastro de produtos
Fonte: Produção Própria

O layout de todos os cadastros segue o mesmo padrão, mostrando a tabela dos itens cadastrados e o botão “+ NOVO”. Para cadastrar a matéria-prima, será solicitado apenas o nome do material, a figura 11 mostra a lista de materiais cadastrados.

Código	Material	Empresa
2	Tinta Preta	Praise Cony
1	Couro	Praise Cony
3	Polino	Praise Cony
4	Freeway	Praise Cony
5	Veneza	Praise Cony
6	Véneto	Praise Cony
7	Vitória	Praise Cony
8	Floter	Praise Cony
9	Toscãna	Praise Cony
10	Verniz	Praise Cony
11	Jeans	Praise Cony
12	Debrid	Praise Cony

Figura 11: Cadastro de matéria-prima
Fonte: Produção Própria

O menu “Cores” possui uma particularidade, pelo fato de a matéria-prima normalmente ser o couro, quer dizer que ela pode ser tingida de qualquer cor pelos curtumes⁵, por este motivo, tanto a empresa quanto o representante comercial podem cadastrar novas cores dependendo da necessidade. As principais cores já são cadastradas (Figura 12) pois seguem um catálogo dos couros.

Código	Cor	Empresa
5	Cru	Sistema Praise - SP
6	Rosa	Sistema Praise - SP
7	Preta	Sistema Praise - SP
9	Cobra	Sistema Praise - SP
10	Jacaré	Sistema Praise - SP
11	Marron	Sistema Praise - SP
12	Whisky	Sistema Praise - SP
13	Tijolo	Sistema Praise - SP
14	Caramelo	Sistema Praise - SP
15	Bege	Sistema Praise - SP
8	Vinho	Sistema Praise - SP

Figura 12: Cadastro de cores
Fonte: Produção Própria

⁵ Curtume são operações de processamento do couro cru e, por extensão, ao local onde este processamento é feito. Tem por finalidade deixá-lo utilizável para a indústria e o atacado.

No menu “Clientes” serão cadastrados todos os clientes (pessoa jurídica) que os representantes comerciais atendem. Conforme descrito no capítulo 3 deste trabalho, tanto o representante quanto a empresa (fábrica) poderão cadastrar e dar manutenção aos clientes. Para o cadastro de cliente são necessárias as seguintes informações: CNPJ, nome (razão social), marca, CEP, endereço, número, bairro, inscrição estadual, telefone, celular e e-mail. E para facilitar o cadastro, ao informar o CEP, os campos correspondentes ao endereço são automaticamente cadastrados, sendo necessário apenas informar o número.

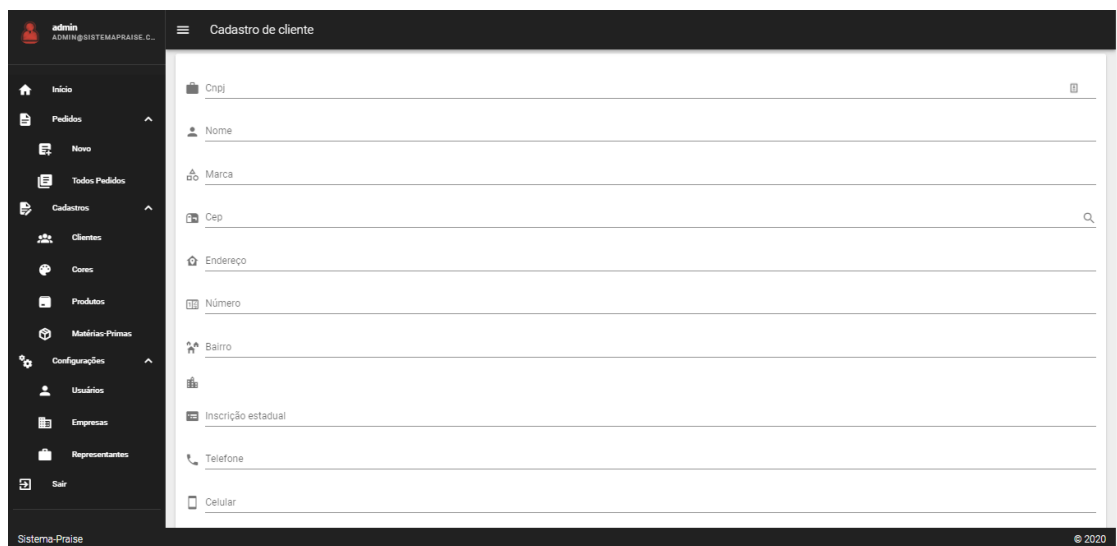


Figura 13: Cadastro de clientes
Fonte: Produção Própria

6.3 Cenários de uso

Os cenários de uso dos próximos itens são uma narrativa textual e ilustrativa de situações de interação do usuário com o sistema. Os cenários foram divididos em três demandas: (i) Consultar cliente; (ii) Realizar pedido; e (iii) Consultar pedido para fabricação.

6.3.1 Consultar cliente

Em um primeiro cenário de uso, o representante chega no cliente (Empresa Fictícia 1), uma loja de acessórios femininos (já cadastrada no sistema), e verifica se há a necessidade de reposição de estoque. Confirmando a necessidade e antes de realizar o pedido, ele irá confirmar se o cliente realmente está cadastrado no sistema e se os dados

estão corretos. Para isso, primeiramente ele acessa o sistema pelo smartphone e informa o e-mail e senha pessoal que foram cadastrados pela empresa.

Após a autenticação, o sistema abre a interface principal, a qual contém apenas os recursos essenciais de acordo com o seu perfil. Ao clicar no menu “Clientes”, o sistema carrega todos cadastrados e é possível confirmar que existe a Empresa fictícia 1 e que os dados estão corretos.

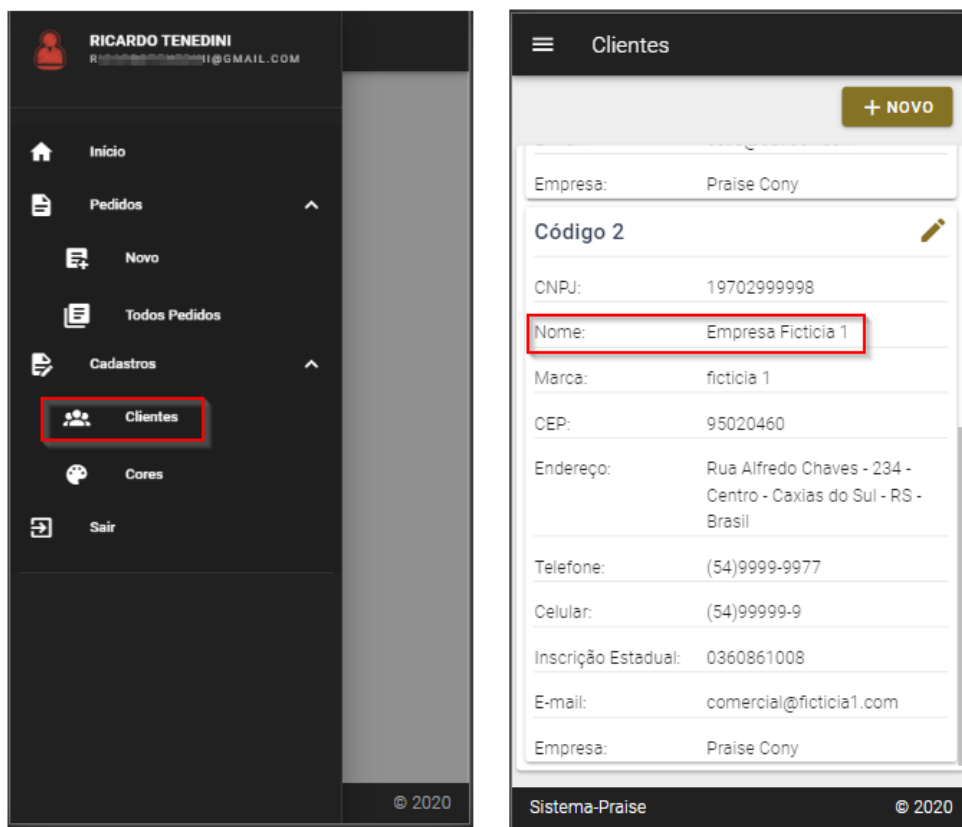


Figura 14: Passo a passo - Consultar Cliente
Fonte: Produção Própria

6.3.2 Realizar pedido

Dando seguimento cenário de uso I, foi confirmado que há necessidade de reposição de estoque e que o cadastro do cliente está atualizado. O representante vai na sessão Pedidos do menu e seleciona a opção “Novo”.

Ao acessar a opção “Novo”, um formulário inicial exigindo a data de entrega, o cliente e a transportadora são exibidos. O representante preenche com os dados solicitados e clica em editar produtos para então colocar os itens desejados pela loja. A inclusão dos itens ocorre forma simples, basta selecionar o produto, a matéria-prima e a cor, após é só informar a quantidade e o valor unitário para que o valor total seja

calculado para então o representante clicar em cadastrar. A cada item adicionado, a lista dos produtos é exibida.

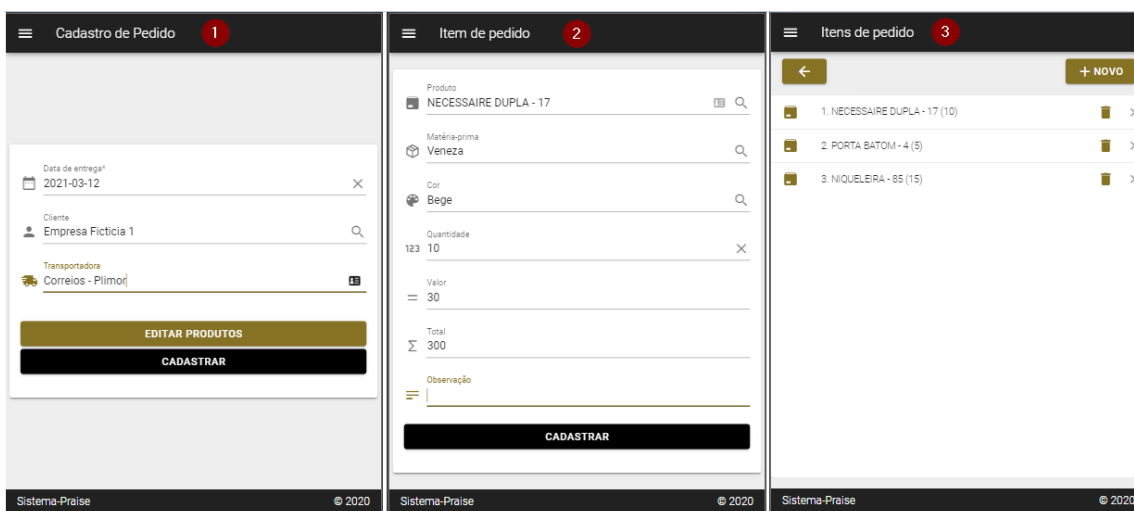


Figura 15: Passo a passo - realizar pedido
Fonte: Produção Própria

Confirmando os itens adicionados e suas quantidades juntamente com o cliente, o representante volta para a interface anterior do sistema e seleciona cadastrar para concluir o processo e automaticamente ser direcionado à interface com os pedidos. Nesta interface, o representante seleciona visualizar o pedido para confirmar novamente se todas as informações estão corretas.

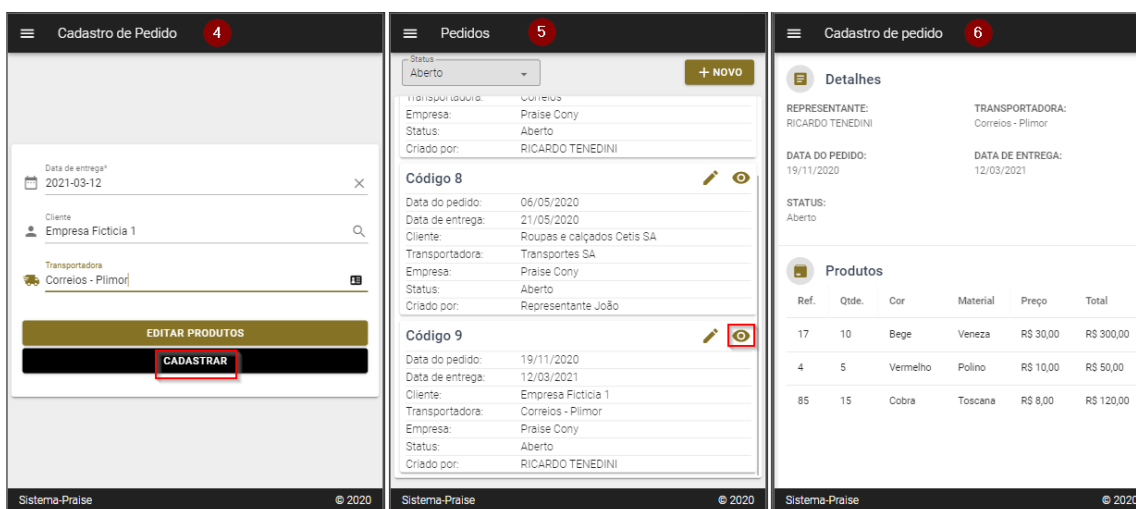


Figura 16: Passo a passo - realizar pedido
Fonte: Produção Própria

6.3.3 Consultar pedido para fabricação (empresa)

Um colaborador da empresa, responsável pela parte administrativa e comercial, acessa o sistema pelo computador para ver se há pedidos em aberto para encaminhar para a produção. Para isso primeiramente é feita a autenticação (figura 17) com as suas credenciais (perfil empresa). O processo é feito da mesma forma que o representante, neste caso, apenas é utilizado um dispositivo diferente.

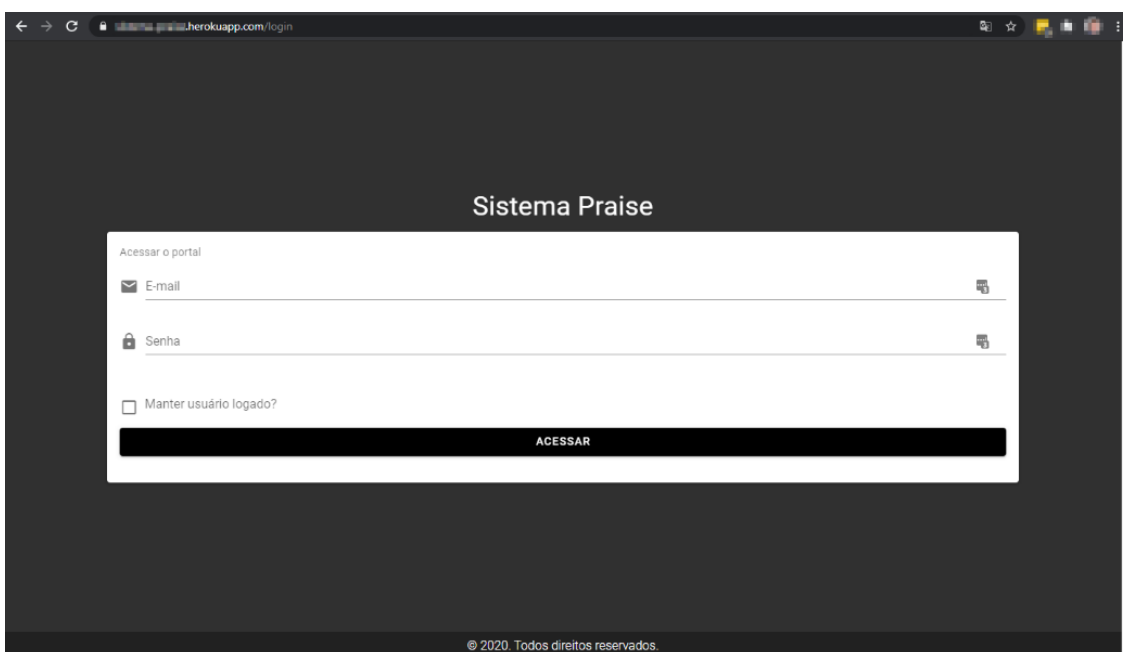


Figura 17: Interface de login pelo desktop
Fonte: Produção Própria

Após autenticar no sistema, para ver os pedidos em aberto, o colaborador vai no menu “Pedidos” e clica na opção “Todos Pedidos”. Neste momento todos os pedidos em aberto são carregados.

Alguns dados básicos são exibidos na interface, como: (i) como data do pedido, (ii) data de entrega, (iii) cliente; (iv) transportadora; (v) empresa, que sempre será a mesma; (vi) *status* e (vii) o nome do representante. Para ver os detalhes do pedido, com os itens a serem fabricados, o colaborador clica no ícone (um olho) de visualizar os detalhes.

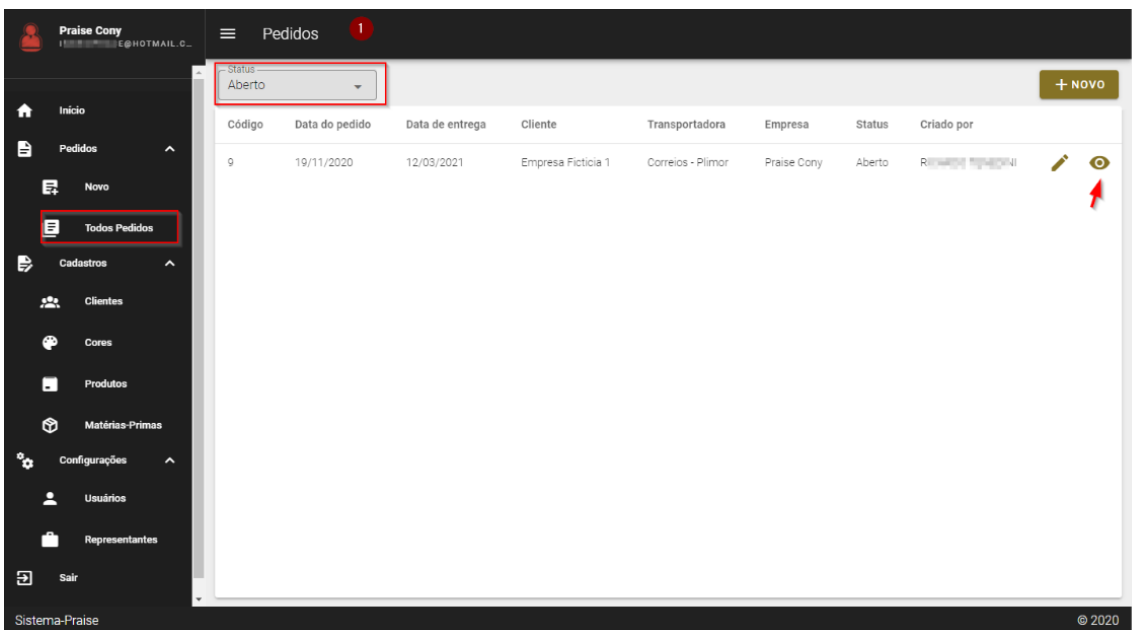


Figura 18: Passo a passo - consultar pedido
Fonte: Produção Própria

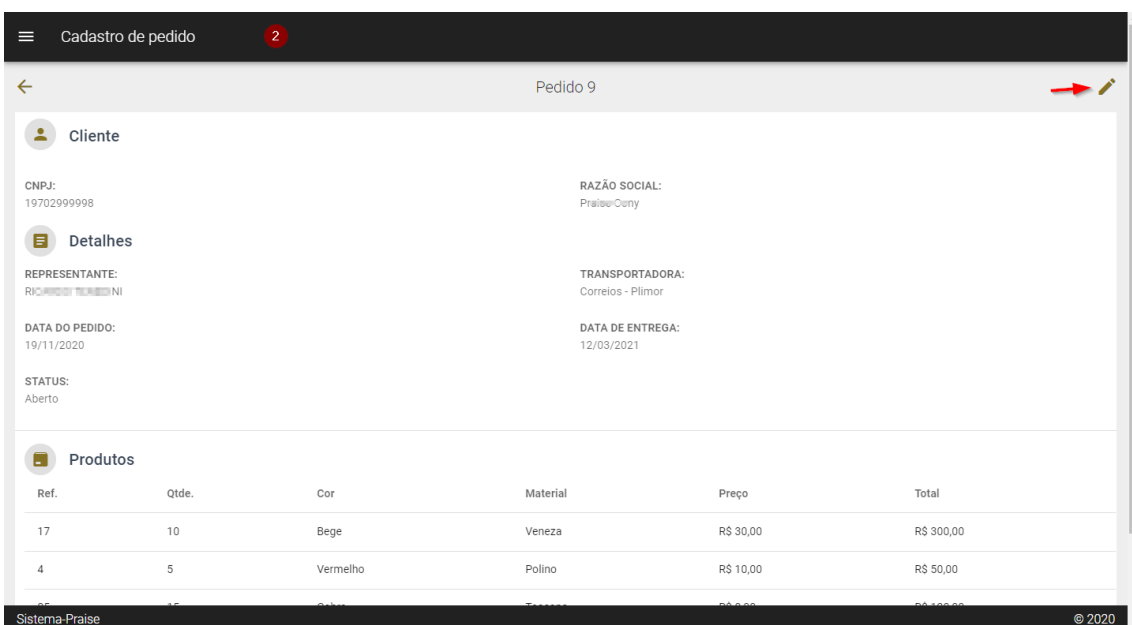


Figura 19: Passo a passo - consultar pedido
Fonte: Produção Própria

Após imprimir as informações para encaminhar para a produção, o colaborador clica em editar pedido (ícone de um lápis) e altera o *status* de “Aberto” para “Confirmado”, para o representante comercial saber que foi encaminhado para fabricação e posteriormente, quando é enviado para a transportadora, o *status* é alterado para “Concluído”.

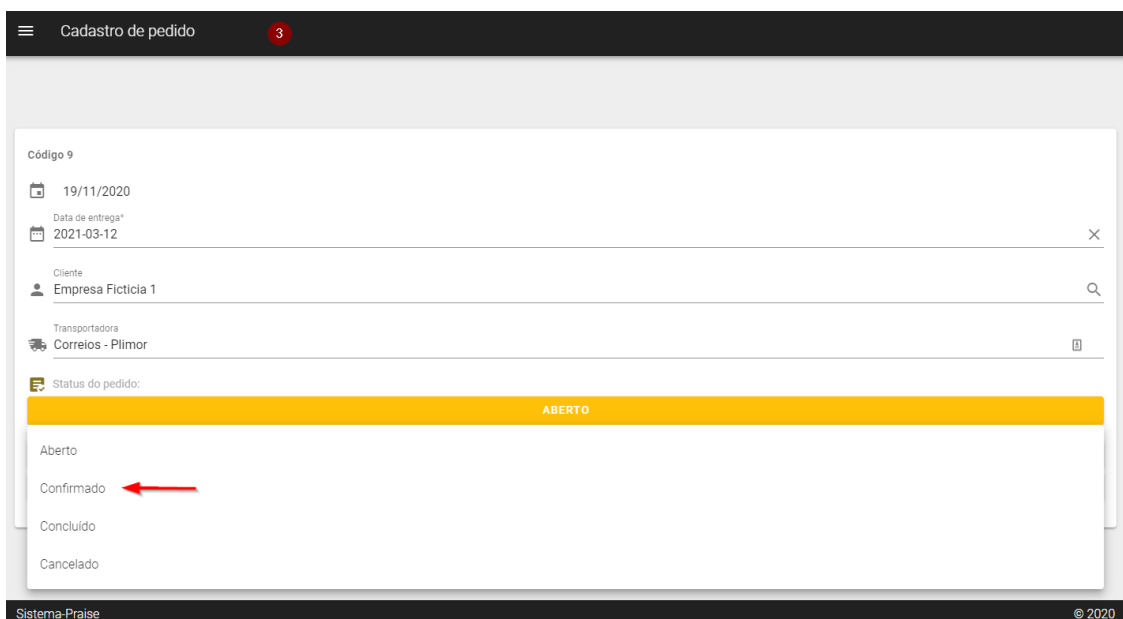


Figura 20: Passo a passo - Consultar pedido - Alterar status
Fonte: Produção Própria

7. Considerações finais

O presente trabalho teve como objetivo realizar estudos acerca da experiência do usuário em aplicações Web, e mais especificamente em aplicações Web progressivas, para o desenvolvimento de um sistema para gerenciamento de pedidos para representantes comerciais de uma empresa específica de venda e fabricação de artigos de couro. Esse sistema auxilia no gerenciamento de produtos, clientes, representantes e realização de pedidos, além disso, fornece uma forma mais fácil de utilização e um gerenciamento mais eficaz.

No texto foram apresentadas as principais tecnologias utilizadas e algumas interfaces do sistema e seus cenários de uso, exemplificando o padrão de interface adotado e a representação responsiva aos diferentes tamanhos de telas, demonstradas com imagens de acessos feitas por computador e *smartphone*.

O desenvolvimento do sistema se deu a partir da criação de dois “projetos”: Uma interface, identificada como *front-end*; e uma API, identificada como *back-end*. Enquanto no *front-end* são apresentados os elementos visuais, fundamentais para a interação do usuário com o sistema; no *back-end* são recebidas e processadas as requisições feitas através dessas interações. Todos os serviços, contanto com o banco de

dados, são consumidos e processados na nuvem na plataforma Heroku, uma *Cloud Application Platform*.

O projeto contemplou todas as expectativas funcionais exigidas pela empresa e os resultados obtidos na validação do sistema foram considerados satisfatórios. Os desenvolvimentos em Vue.js e Node.js, juntamente com a plataforma Heroku, permitiram criar um sistema com uma boa usabilidade e escalabilidade.

Durante o desenvolvimento surgiram algumas dificuldades, por nunca ter trabalhado com o Node.js, Vue.js e o Heroku, foram necessárias muitas pesquisas, tutoriais e cursos rápidos para então utilizar essas tecnologias, o que torna mais prazeroso ao ver o resultado. Um ponto muito positivo é que existem muitos materiais disponíveis, facilitando assim a implementação recursos e funcionalidades.

Para trabalhos futuros, sugere-se aprimorar o sistema adicionando novas funcionalidade, como por exemplo: (i) relatórios de vendas; (ii) cálculo de comissão dos representantes, (iii) emissão de nota fiscal; (iv) central de avisos. Há a necessidade também de hospedar o *front-end* em outro local, pois o Heroku é focado em *back-end*, e para manter no Heroku, o ideal seria usar a linguagem PHP, não Vue.js.

Referências

AMARAL, Rodrigo Augusto; NERIS, Vânia Paula de Almeida. **Análise comparativa entre frameworks de front-end para aplicações Web ricas visando reaproveitamento do back-end**. Revista T.I.S. janeiro-abril 2015. 88-96

COSTA, Mirlanda Sousa. **Sistemas Web e mobile: uma visão geral para negócios empresariais**. Revista Científica Multidisciplinar Núcleo do Conhecimento. Ano 03, Ed. 08, Vol. 09, pp. 82-99, Agosto de 2018. ISSN:2448-0959

DIAS, Ana Luiza; FORTES, Renata Pontin de Mattos; MASIERO, Paulo César e GOULARTE, Rudinei. **Uma Revisão Sistemática sobre a inserção de Acessibilidade nas fases de desenvolvimento da Engenharia de Software em sistemas Web**. Anais Estendidos, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação. São Paulo, 2010.

FOWLER, M. e LEWIS, J. (2014). **Microservices**. <<https://martinfowler.com/articles/microservices.html>>. Acessado em 03/11/2020.

FRANÇA, F. D. S. **WEB DESIGN RESPOSIVO: CAMINHOS PARA UM SITE ADAPTÁVEL**. INTERFACES CIENTÍFICAS - EXATAS E TECNOLÓGICAS, Junho 2015. 75-84.

HEROKU. **What**. Disponível em <<https://www.heroku.com/what>>. Acesso em 21 de outubro de 2020.

KALBACH, James. **Design de navegação Web: otimizando a experiência do usuário**. Porto Alegre: Bookman, 2009.

NODEJS. **About Node.js**. Disponível em <<https://nodejs.org/en/about/>>. Acessado em 21 de outubro de 2020.

REIS, Renata Tibiriçá. **Desenvolvimento Web com o Uso de Padrões: Tecnologias e Tendências**. Dissertação (Dissertação em Ciência da Computação) - Universidade Federal de Juiz de Fora. Juiz de Fora, 2007.

SILVA, J. K.; TIOSSO, F. **REVISÃO BIBLIOGRÁFICA SOBRE CONCEITO DE PROGRESSIVE WEB APPLICATIONS (PWA)**. Revista Interface Tecnológica, [S. l.], v. 17, n. 1, p. 53-64, 2020. DOI: 10.31510/infa.v17i1.713. Disponível em: <https://revista.fatectq.edu.br/index.php/interfacetecnologica/article/view/713>. Acesso em: 14 nov. 2020.

SUMMERVILLE, Ian. **Engenharia de Software**. 9ª Ed. São Paulo: Pearson Prentice Hall, 2011.

VUEJS. **Guide**. Disponível em <<https://vuejs.org/v2/guide/>>. Acesso em 21 de outubro de 2020.

VUEX. Disponível em < <https://vuex.vuejs.org/ptbr/>>. Acesso em 21 de outubro de 2020.

VUETIFY. Disponível em < <https://vuetifyjs.com/en/>>. Acesso em 21 de outubro de 2020.

VSCODE. **Docs: Getting Started**. Disponível em <<https://code.visualstudio.com/>>. Acesso em 01 de novembro de 2020.