



UNICEPLAC

Centro Universitário do Planalto Central Aparecido dos Santos

Curso de Sistemas de Informação

Trabalho de Conclusão de Curso

**Sistema para Gerenciamento e Controle Interno de
Micro e Pequenas Empresas: Aplicado em uma Empresa do Ramo
de Vidraçaria**

Brasília-DF
2020



UNICEPLAC

**DANIEL PONTES FREITAS SOARES
ITALO BORGES EVANGELISTA**

**Sistema para Gerenciamento e Controle Interno de
Micro e Pequenas Empresas: Aplicado em uma Empresa do Ramo
de Vidraçaria**

Artigo apresentado como requisito para conclusão do curso de Bacharelado em Sistemas de Informação pelo Centro Universitário do Planalto Central Aparecido dos Santos – Uniceplac.

Orientador: Dr. Sebastião Ivaldo Carneiro Portela

Brasília-DF
2020



UNICEPLAC

**DANIEL PONTES FREITAS SOARES
ITALO BORGES EVANGELISTA**

**Sistema para Gerenciamento e Controle Interno de
Micro e Pequenas Empresas: Aplicado em uma Empresa do Ramo de Vidraçaria**

Artigo apresentado como requisito para conclusão do curso de Bacharelado em Sistemas de Informação pelo Centro Universitário do Planalto Central Aparecido dos Santos – Uniceplac.

Gama, 15 de Dezembro de 2020.

Banca Examinadora

Prof. Sebastião Ivaldo Carneiro Portela
Orientador

Prof. Wilton Marinho Carneiro Souza
Examinador

Prof. Valmir de Lima Severiano
Examinador



UNICEPLAC

Sistema para Gerenciamento e Controle Interno de Micro e Pequenas Empresas: Aplicado em uma Empresa do Ramo de Vidraçaria

Daniel Pontes Freitas Soares¹

Italo Borges Evangelista²

Resumo:

Existem evidências que as micros e pequenas empresas têm dificuldades em possuir um sistema que ajude no controle e gerenciamento interno do negócio. Falta de recursos financeiros e o limitado domínio do ramo são fatores que dificultam a escolha e utilização de sistemas funcionais. Este estudo tem o propósito de suprir essas dificuldades com o desenvolvimento de um sistema aplicado inicialmente ao ramo de vidraçaria e que pode ser expandido para outros setores. O sistema foi criado usando metodologias ágeis e baseou-se na modularidade da gestão integrada (ERP). Deu-se preferência às ferramentas de desenvolvimento mais recentes como *GraphQL* e *Prisma*. Como resultado foi desenvolvido a autenticação do sistema, o módulo de cadastro de estoque e de clientes.

Palavras-chave: Micro e pequenas empresas. Metodologias Ágeis. Sistema de gerenciamento interno do negócio.

Abstract:

There is evidence that micro and small companies have difficulties to find a system that helps in the control and internal management of the business. Less financial resources and the limited domain of the business are factors that make choices difficult and use of functional systems. This study aims to supply these difficulties with the development of a system initially applied to the glass enterprise and which can be expanded to other sectors. The system was created using agile methodologies and based on the modularity of Enterprise Resource Planning (ERP). The most recent development technologies were preferred, such as GraphQL and Prisma. As a result, system authentication, the inventory and customer registration module were developed.

Keywords: Micro and small businesses. Agile Methodologies. Enterprise Resource Planning System.

¹Graduando do Curso de Sistemas de Informação, do Centro Universitário do Planalto Central Aparecido dos Santos – Uniceplac. E-mail: danielpfsoares@gmail.com.

² Graduando do Curso de Sistemas de Informação, do Centro Universitário do Planalto Central Aparecido dos Santos – Uniceplac. E-mail: italoborgesevangelista@gmail.com.



UNICEPLAC

1. INTRODUÇÃO

Existem vários processos que são aplicados para um bom gerenciamento interno, contudo, são poucos utilizados em função do elevado custo, desorganização, indisciplina e da falta de domínio no gerenciamento e funcionamento da empresa por parte dos empresários. Esses fatores acabam colocando em risco a dinâmica de funcionamento e crescimento da empresa.

De acordo com a análise do Sebrae (2010), todas as empresas necessitam de *software* de apoio à gestão dos processos de negócio para poder organizar e estruturar todas as informações existentes.

Para ter um sistema que atenda às necessidades do cliente, é necessário ter uma ferramenta que contribua no processo de gerenciamento interno para trazer resultados positivos. No caso específico de uma empresa do ramo de vidraçaria foi proposto a construção e validação no ambiente de desenvolvimento dos módulos de controle de estoque, controle de notas fiscais e relatórios, recursos humanos, controle de serviços e vendas e controle financeiro baseado nas características e metodologia de um sistema de gestão integrado (*ERP*).

Ao contrário das alternativas de sistemas disponíveis e padronizadas, a construção local de um sistema gera eficiência no atendimento das demandas específicas do ramos de atuação, garante expansividade de módulos de acordo com o crescimento da empresa têm custos reduzidos, suporte mais rápido e eficiente e acima de tudo, incentiva as pequenas empresas de *software* a ampliar a expertise local.

Nesse sentido, Souza (2000) afirma que “a alteração do pacote, quando feita mediante customização, pode levar a uma série de custos adicionais que se repetirão enquanto se utilizar o pacote”. A grande dificuldade das empresas que fazem o uso das tecnologias são as adaptações, pois, embora os sistemas consiga suprir as necessidades haverá uma série de alterações para adaptação e customização, fazendo com que as manutenções e os custos crescem gradativamente. Por isso, a criação de sistemas baseado nas necessidades específicas, costumam ser mais vantajosas.



UNICEPLAC

2. REVISÃO DE LITERATURA

2.1. Sistema de Gestão Integrado

Ao iniciar o processo de gerenciamento interno de uma empresa, o responsável por esse papel, utiliza ferramentas para facilitar o processo de controle interno e auxiliar as divisões de tarefas com seus respectivos setores. Neste caso, o sistema terá um auxílio de uma ferramenta chamada de Sistema de Gestão Integrado, do inglês, *Enterprise Resource Planning (ERP)*. Conforme Carvalho (2017):

Os sistemas *ERP* fornecem rastreamento e visibilidade global da informação de qualquer parte da empresa e de sua Cadeia de Suprimento, o que possibilita decisões inteligentes. Pode-se dizer que o *ERP* é um sistema integrado, que possibilita um fluxo de informações único, contínuo e consistente por toda a empresa, sob uma única base de dados. (CARVALHO, 2017, p. 3)

O motivo na qual o sistema será auxiliado pelo *ERP* é que as divisões de tarefas com seus respectivos setores são feitas através de módulos referentes a necessidade do cliente. Ao aplicar ao sistema, os módulos serão divididos em financeiro, controle de estoque, controle de notas fiscais e relatórios, recursos humanos, controle de serviços e vendas. A figura abaixo é uma representação da estrutura organizacional do *ERP* realizados por Saipos (2020):

Figura 1 – Sistema de Gestão Integrado



Fonte: IT REFRESH SISTEMAS, 2020.



UNICEPLAC

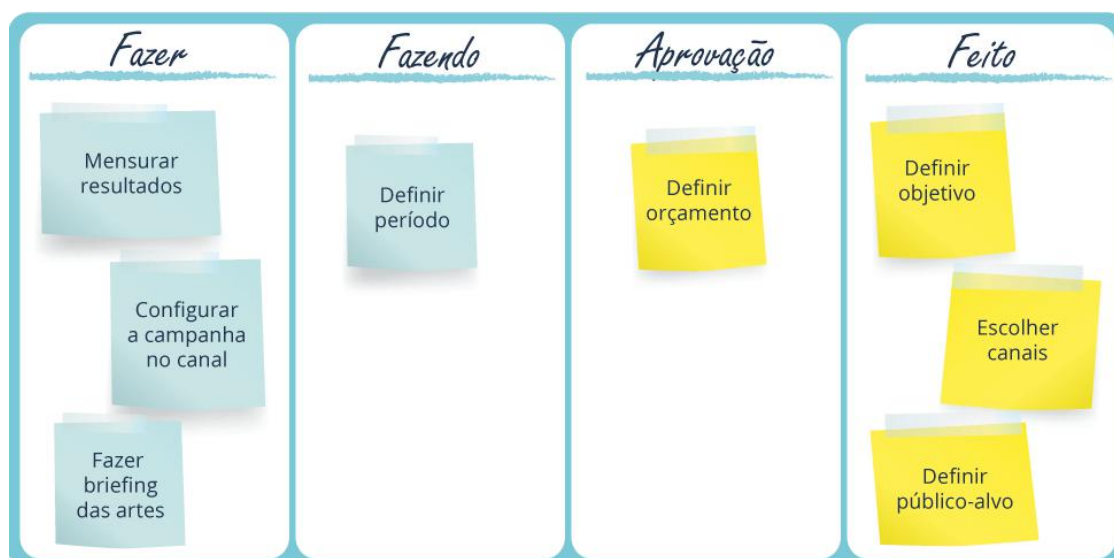
2.2. Estrutura para Construção de um Sistema e Tecnologias Envolvidas

Para a construção do sistema é necessário cumprir etapas do ciclo de vida de desenvolvimento de sistemas controlando e organizando todo o trabalho que já foi produzido e etapas futuras a partir de um estudo inicial. Desta forma, para construir o sistema de acordo com Sordi (2006):

O *software* da empresa deve ser: simples (para que todos os investidores e usuários possam usar o sistema com facilidade); flexível (para que possa acomodar em tempo as possíveis alterações requeridas pelo ambiente de negócio); e ser capaz de desvincular funcionalidades do negócio das tecnologias utilizadas para sua execução. (SORDI, 2006, p. 6).

Com o propósito de organizar e ordenar todo o serviço feito num projeto, o sistema precisa se basear nas metodologias de gerenciamento de projeto. A metodologia Scrum orienta a equipe, alinha demandas, possibilita reuniões rápidas para esclarecimento e discussões sobre o projeto, testa e revisa todo o serviço feito. A metodologia Kanban é utilizada no controle dos fluxos de produção e divisão de tarefas com graus de necessidade. A articulação dessas duas ferramentas, possibilita melhor gerenciamento, rapidez na entrega das tarefas e evita o retrabalho. A figura abaixo representa um quadro de tarefas organizadas no formato de um *Kanban* que ilustra as divisões que normalmente são usadas durante uma sprint baseado no *Scrum*. Pois essas metodologias, foram escolhidas e utilizadas no desenvolvimento do projeto.

Figura 2 – Scrum e Kanban



Fonte: ARTIA, 2019.



UNICEPLAC

2.2.1. Levantamento de Requisitos e Modularização

O levantamento de requisitos consiste em compreender e levantar todas as informações que dizem respeito às necessidades do cliente. É de grande importância saber documentar o que o cliente realmente precisa para que não ocorra perdas de informações e conseqüentemente, gerar funcionalidades que não atendam as demandas solicitadas. Nesse sentido, é importante diversificar as técnicas e ampliar os canais de comunicação entre o desenvolvedor e o cliente na etapa de levantamento de dados, como defende Farias (2017), “Para a etapa de levantamento de requisitos é necessária a aplicação de uma ou mais técnicas visando à captura de informações sobre o produto final desejado e, principalmente, das necessidades dos clientes/usuários”.

Na análise para o levantamento também é englobada a engenharia de requisitos que é um processo na qual reúne todas as atividades contribuintes para a produção dos documentos possibilitando possíveis manutenções de acordo com a construção do sistema. Vale lembrar que durante a construção de *software*, é possível haver mudanças do escopo do projeto. Como exemplo, de acordo também com Farias (2017), “por meio de entrevistas e apresentação de protótipos é possível instigar o futuro usuário a encontrar funcionalidades necessárias não previstas em uma análise prévia”. Sabe-se então que durante a construção, o *software* é um produto que contém mudanças constantes, tudo para suprir as necessidades do cliente sem fazer com que haja problemas na contratação do serviço.

No caso do sistema de vidraçaria, após levantar os requisitos, foi aplicado a modularização para separar em categorias todas as informações. A modularização é aplicada para dividir recursos e funcionalidades do sistema atribuindo responsabilidades aos atores (usuários) e tomou como base o *ERP*.

2.2.2. Arquitetura do *Software*

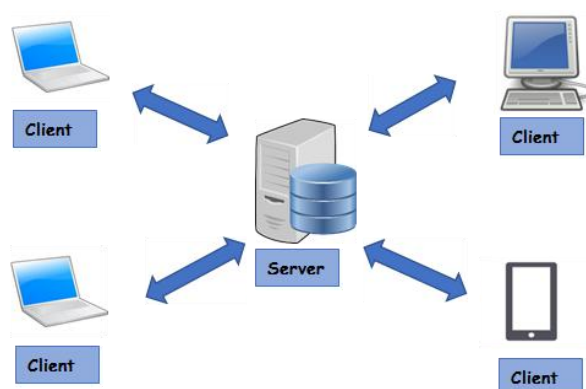
Ao passar pelas etapas de modularização e pelo entendimento do sistema, inicia-se a busca de tecnologias para a criação do *software*. Para facilitar o entendimento, a busca de tecnologias foi dividida em duas partes, a primeira sendo o *front-end* responsável pela interface do usuário e a segunda o *back-end*, responsável pela lógica, segurança e sigilo das informações.



UNICEPLAC

A natureza de sistemas para web pode ser separada em dois grupos; conforme a figura abaixo, os grupos divergem em tecnologias para cliente (também conhecido como *front-end*) e servidor (*back-end*).

Figura 3 – Cliente e Servidor



Fonte: OMNISCI, 2020.

2.2.2.1. *Front-end* (cliente)

Toda interface que interage com o usuário pode ser considerada como *front-end* de um sistema maior. Para o desenvolvimento do *front-end* será utilizado o *Nuxt.js*, que é um framework de código aberto e com muitos recursos recentes para se trabalhar e ganhar desempenho. Esse framework permite o desenvolvimento de uma *Single Page Application* (*SPA*). Aplicações *SPA* permitem que rode em uma única página, semelhante aos aplicativos desktop ou ao mobile, sem a necessidade de recarregamento de elementos. Além disso, outros recursos estarão disponíveis como, *Server Side Renderer* (*SSR*), *Progressive Web App* (*PWA*), *Middlewares*, *Layouts*, *Rotas*, *State Management Pattern* e muitos outros (NUXTJS, 2020).

2.2.2.2. *Back-end* (servidor)

O servidor é responsável por ter toda a regra de negócio e dados sensíveis que podem ou não exigir restrições de acesso ou dependerem de algum processo. O *back-end* será desenvolvido em *Node.js* utilizando tecnologias como *GraphQL* (para padronização da comunicação do servidor com o cliente. Seu intuito é ser uma linguagem de consulta adaptativa que retorne apenas o que foi pedido), *Prisma 2* (é um *Object Relational Mapper – ORM* –, camada responsável por comunicar com banco dados e reduzir acoplamento e



UNICEPLAC

repetição de código) e *Nexus* (para agilizar a criação de *CRUD's* – *Create, Read, Update and Delete* –, termo que referencia as operações básicas que um dado pode permutar. Ela se beneficia da tecnologia *GraphQL*, para gerar a configuração necessária da regra de negócio).

Todas as tecnologias citadas são gratuitas e contém uma redução de custos em manutenção. Com elas, será feito o desenvolvimento de uma Interface de Programação de Aplicação (*API*) que servirá as informações para o *front-end* e ficará disponível para que outros serviços possam utilizá-la (um aplicativo *mobile*, por exemplo).

2.2.2.3. Banco de Dados

A modelagem de dados é uma etapa de tratamento que depende do modelo de dados e do banco de dados.

O sistema é construído através da necessidade do cliente, as informações são inseridas e depois transformadas em dados. Essa etapa é importante e essencial, pois ela visa explorar todo o fluxo de dados que serão inseridos após a modelagem.

De acordo com Meira (2013), os "bancos de dados são usados em muitas aplicações, desde sistemas simples para controlar o estoque de material de uma loja a sistemas avançados como sistemas bancários e segurança pública". Pode-se dizer que o modelo organiza os dados em tabelas estruturadas de uma forma mais organizada, obedecendo as funcionalidades do sistema e controlando todo o fluxo de dados.

É necessário que a criação do banco seja dividida em etapas para que tenha uma boa organização. A primeira etapa é analisar os requisitos para a produção dos dados, na segunda etapa, organizar os dados em tabelas para que tenha uma boa estrutura e na terceira etapa, padronizar as tabelas de acordo com a necessidade do sistema. Ou seja, o banco de dados é uma estrutura regular que organiza informações com o intuito de agrupar informações para o mesmo destino.

Baseados nesses conceitos, será utilizado um Sistema de Gerenciamento de banco de dados (SGBD) numa estrutura relacional. Para isso, foi escolhido o *PostgreSQL* por ser uma solução de código aberto e livre, barato em termos de uso de recursos e bastante extensível.



UNICEPLAC

2.2.2.4. Infraestrutura

Durante todo o desenvolvimento do projeto será utilizada ferramenta de versionamento de código como *GIT*, *Docker* para geração de *containers* para criação do ambiente e, como editor de texto, *Visual Studio Code (VSCode)*.

3. PROCEDIMENTOS METODOLÓGICOS

Foi realizada uma entrevista com o cliente para entender as necessidades da empresa e como o sistema deve funcionar e o que deve ser priorizado. Durante essa entrevista, foram levantadas os requisitos funcionais para que o sistema supra todas as necessidades, funcionalidades e características da empresa. Conforme a Figura 4, foram levantados os seguintes requisitos.

Figura 4 - Levantamento de Requisitos

Usuário:	<ul style="list-style-type: none">- Clientes podem possuir desconto e serviços diferentes (marceneiro, por exemplo, não possuem instalação de vidros, ferragens e etc.)
Estoque:	<ul style="list-style-type: none">- Tem valor de revenda em corte de vidros, por exemplo;- Valores de serviço diferente por tipo de vidro (montagem);- Massa de vidro (instalação);- Kit de produtos;- Vidro de refugo - venda com valor padronizado;- Status - em produção, permitido etc.
Gerar Notas Fiscais e Relatórios:	<ul style="list-style-type: none">- Orçamento;- Pedido de corte;- Pedido de entrega para cliente;- Autorização - contrato para o cliente;- Pedido de entrega.
Recursos humanos:	<ul style="list-style-type: none">- Cadastro e controle de funcionário;- Controle de salário, vale, qualificações;- Banco de horas;- Métricas de serviços.
Controle de serviços e vendas:	<ul style="list-style-type: none">- Cadastro de serviços;- Controle de estado de serviços de mão de obra;- Planejamento, métricas e prazos de serviços.
Controle financeiro:	<ul style="list-style-type: none">- Fluxo de caixa (entrada e saída);- Controle, métricas e previsões de gastos;- Reposição de estoque.

Fonte: Dos autores, 2020.



UNICEPLAC

No processo de desenvolvimento do sistema foi utilizada a metodologia *Scrum* que guiou e orientou as ações. Possibilitou a equipe alinhar demandas, realizar reuniões para esclarecimentos e discussões sobre o projeto; testar e revisar as tarefas realizadas considerando a disponibilidade dos envolvidos – programadores e clientes. Após a definição das tarefas, uma estrutura de *Kanban* foi utilizada para ajudar a gerenciar o desenvolvimento.

Com o levantamento de requisitos, para colher os dados relacionados à necessidade do usuário e criação do *software*, ficou evidente que a maior necessidade da empresa de ter um sistema inovador, que pudesse fazer todo o cadastro de clientes e produtos, controle de estoque e controle orçamentário. Isso possibilitou a seleção de tecnologias, priorizando as mais recentes, dos últimos cinco anos.

As demandas da empresa foram modularizadas conforme o ERP, contudo, focalizou-se nos elementos mais urgentes e de curto prazo - o controle de estoque. Porém, referente aos outros módulos solicitados pelos usuários para a implementação no sistema, como planos futuros, será implementado os seguintes módulos com as suas respectivas funcionalidades:

3.1. Controle de Notas Fiscais e Relatórios

Terá como objetivo organizar todas as vendas e compras de produtos feitas respectivamente pelos clientes e funcionários.

3.2. Recursos Humanos

Terá uma função de controlar as folhas de pagamentos, folhas de pontos, rotinas de departamento pessoal, entre outros.

3.3. Controle de Serviços e Vendas

Será implementado para controlar as demandas existente para os funcionários e controlar todas as vendas feitas pela empresa.

3.4. Controle Financeiro

Controlará todos os gastos referente às compras e pagamentos respectivamente dos produtos e funcionários, e lucros gerados pela a empresa.



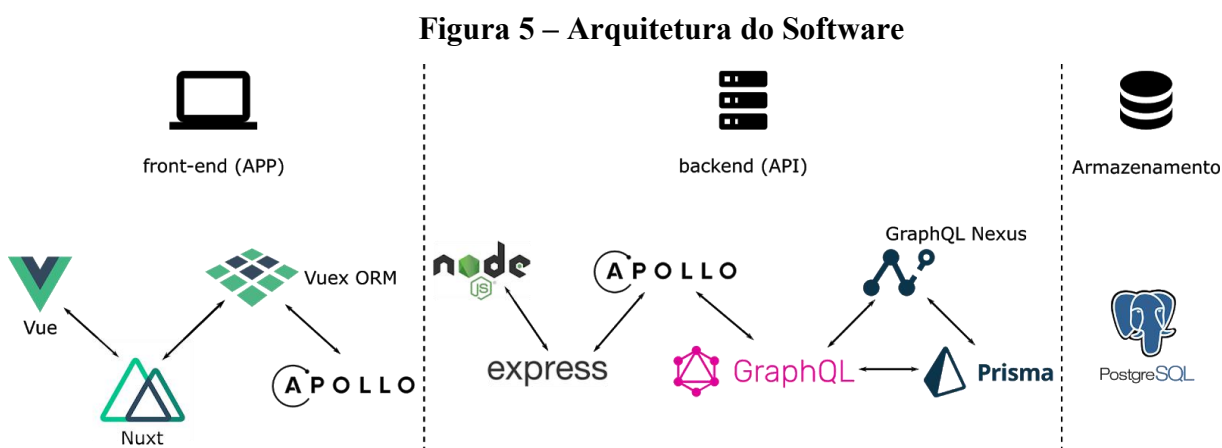
UNICEPLAC

A medida que o software era desenvolvido, em ambiente local, testes foram realizados para garantir a funcionalidade dos recursos e promover ajustes e correções, em conformidade com as tecnologias ágeis.

Para o desenvolvimento do sistema foi considerado duas dimensões, uma responsável pela regra de negócio do sistema (back-end) e outra responsável pela a parte visual (front-end). Para o front-end, utilizou-se o framework Nuxt (que usa o Vue para montagem da interface gráfica e recursos modernos de criação de websites), o Vuex ORM (controla informações para concepção da interface gráfica) e Apollo (permite a comunicação com o servidor de forma padronizada).

Para o back-end, por sua vez, utilizou-se Node.js (possibilita que o servidor use JavaScript como linguagem de programação), Express (servidor HTTP), Apollo, GraphQL (mantém a lógica de consulta de informações e permissões de acesso), Prisma (ORM para conexão com o banco de dados) e GraphQL Nexus (gera as configurações necessárias para a comunicação entre o cliente, GraphQL e Prisma).

Para o armazenamento, escolheu-se PostgreSQL (banco de dados relacional, seguro e gratuito). Os bancos relacionais não possuem um conceito mais moderno, mas são extremamente consolidados no mercado e são soluções de uso simples. Conforme a figura abaixo, o sistema consiste nas seguintes tecnologias.



Fonte: Dos autores, 2020.

O ambiente de desenvolvimento baseou-se na utilização do VSCode (ferramenta de edição de código), GIT (controle de versionamento de código) e Docker (controla os serviços de cliente, servidor e banco de dados por meio de máquinas virtuais).



UNICEPLAC

4. APRESENTAÇÃO E ANÁLISE DOS DADOS

O sistema desenvolvido possui uma interface limpa, intuitiva e moderna. Na versão atual apresenta recursos de autenticação, controle de estoque e cadastro de clientes, os quais terão suas funcionalidades analisadas, exploradas e discutidas nas seções a seguir.

4.1. Autenticação

Para acessar os recursos, é necessário passar por um processo de autenticação. A Figura 6, demonstra que é preciso ter um *email* e senha cadastrados para utilização. Uma pessoa não autorizada não poderá criar usuários e nem mesmo acessar as informações. Em um processo posterior, apenas usuários do sistema terão permissões para adicionar novos usuários. Para esse efeito, a segurança da informação foi aplicada para barrar usuários não permitidos e controlar todos os acessos de *login* por meio de criptografia de senha e *token* assinado pelo servidor para garantir a genuinidade do usuário.

4.2. Controle de Estoque

No recurso de estoque, pode-se cadastrar produtos e serviços. Os últimos 10 produtos cadastrados na aplicação já ficam visíveis para uma rápida busca. Na Figura 6, é representada a parte visual do *CRUD*, onde pode-se adicionar produtos ao estoque, editar informações gerais e excluí-los. Com a tecnologia *GraphQL Nexus*, agiliza a criação de *CRUD* dentro da *API*, se preocupando apenas com as regras de negócio.

No recurso de estoque, pode-se cadastrar produtos e serviços. Os últimos 10 produtos cadastrados na aplicação já ficam visíveis para uma rápida busca. Na figura 6, é representada a parte visual do *CRUD*, onde pode-se adicionar produtos ao estoque, editar informações gerais e excluí-los. Com a tecnologia *GraphQL Nexus*, agiliza a criação de *CRUD* dentro da *API*, se preocupando apenas com as regras de negócio.



UNICEPLAC

Figura 6 – Lista de Estoque

Categoria	Nome	Preço	Estoque	Ações
Vidro	Incolor 3mm	R\$ 1,23		
Ferragem	PU	R\$ 1,20		
Moldura	Lisa Branca	R\$ 1,60		
Vidro	Verde 6mm	R\$ 2,80		
Vidro	Fumê 8mm	R\$ 2,50		

Linhas por página: 10 1-5 de 5

Fonte: Dos autores, 2020.

Para uma rápida inserção de dados, as categorias de produtos já podem ser pesquisadas ou criadas em um único campo. Os campos para adicionar um novo produto podem ser vistos na Figura 7. Posteriormente, será possível cadastrar o estoque mínimo de um produto, receber aviso de estoque baixo, gerar métricas, realizar pedidos de reposição e cálculo de quantidade de estoque aproximado para matéria-prima.

Figura 7 – Cadastro de Novo Produto

Novo Produto

Categoria

Nome

Tipo de Unidade

R\$ 0,00

Peso (opcional)

Fonte: Dos autores, 2020.

A Figura 8, demonstra como são exibidas as informações de um produto. Cada produto pode ter diversas unidades que são ou não estocadas nos mesmos lugares, desde por terem dimensões diferentes ou, até mesmo, uma lógica já pré-existente de armazenagem. Por isso, é possível indicar os setores onde estão alocadas as unidades para uma fácil localização.



UNICEPLAC

Figura 8 – Informação do Produto

Incolor 3mm

Categoria: Vidro
Valor: R\$ 1,23
Peso: 0

Informações

Estoque Mínimo
+ Adicionar Estoque Mínimo

Estado Atual
Atualizado

Média de Estoque
56203 mm²

Unidades +

Setor	Quantidade	Altura(mm)	Largura(mm)
A1	2	100	312
A3	1	100	250

Fonte: Dos autores, 2020.

4.3. Cadastro de Cliente

Como pode ser visto na Figura 9, a aplicação possui um *CRUD* para o cadastro de informações dos clientes da organização. Essa funcionalidade permite que a empresa acesse os dados de endereço, por exemplo, de forma rápida e agilize outros processos que ocorrem fora do sistema.

Figura 9 – Cadastro Cliente

Clientes +

Nome	Email	CPF/CNPJ
Não há dados disponíveis		

Linhas por página: 5 - < >

Fonte: Dos autores, 2020.

5. CONSIDERAÇÕES FINAIS

O sistema foi desenvolvido para suprir pontos específicos de uma empresa do ramo de vidraçaria ao analisar a necessidade da empresa, constatando que o controle interno não era



UNICEPLAC

realizado por sistemas inovadores. O desenvolvimento foi dividido em graus de necessidades baseado no método *ERP* para organizar e dividir em módulos toda a estrutura do sistema.

Referente às metodologias ágeis, foi usado o *Scrum* para orientar a equipe durante o desenvolvimento, alinhar as demandas para que não ocorra acumulação e nem perda de tarefas, possíveis reuniões rápidas para esclarecimento e discussões sobre o projeto, testes e revisões de todos os serviços feitos. E a metodologia *Kanban*, para controlar os fluxos de produção e dividir tarefas com graus de necessidade.

Dadas as condições por falta de tempo e reuniões, somente uma parte do sistema foi desenvolvida. Pois levando em consideração suas necessidades específicas de forma a garantir precisão do funcionamento, o desenvolvimento foi baseado nas características e metodologia de um sistema de gestão integrado (*ERP*). Porém, apenas os módulos de controle de estoque e cadastro cliente foram realizados. É importante entender que a metodologia possui um grande escopo que é importante ser validado gradativamente pelo cliente para que atenda-o da melhor forma possível e por isso, foi decidido projetar apenas uma parte do *software*, seguindo as metodologias ágeis, para que fosse obtido uma versão estável e disponível à empresa. O sistema ainda está sofrendo alterações e em breve estará disponível ao cliente para que seja feita a validação do que foi desenvolvido.

Com o intuito de aproximar o sistema das reais necessidades do cliente, o sistema terá aplicações futuras como controle de notas fiscais e relatórios, recursos humanos, controle de serviços e vendas e controle financeiro que possibilitará o funcionamento completo do sistema.

REFERÊNCIAS

A QUERY language for your API. **GraphQL**. [S. 1.], 2020. Disponível em: <https://graphql.org>. Acesso em: 01 de nov. de 2020.

CARVALHO, Juliana Nadjara. **O impacto da implantação de sistemas ERP em funcionários da área fabril**, pág. 3. Santa Catarina, Florianópolis, 2017.

DOCUMENTATION Home: Explore the Apollo platform. **Apollo**. [S. 1.], 2020. Disponível em: <https://www.apollographql.com/docs>. Acesso em: 05 de nov. de 2020.

ERP – Sistema de gestão online. **IT REFRESH SISTEMAS**. [S. 1.], 2017. Disponível em: <https://www.itrefresh.com.br/portfolio/erp-online>. Acesso em: 01 de nov. de 2020.

ESPINHA, Roberto Gil. **Kanban: O que é e tudo sobre como gerenciar trabalho**. [S. 1.], 2019 Disponível em: <https://artia.com/kanban>. Acesso em: 01 de nov. de 2020.



UNICEPLAC

FARIA, Afro Netto Nunes. **O uso de técnicas de levantamento de requisitos para inovação no desenvolvimento de software**, pág. 3. México, 2017.

MEIRA, Regilan. **Banco de dados: Apostila de banco de dados**, pág.6. Bahia, Ilhéus, 2013.

MICROSOFT. **Editor de Consulta do SSMS (SQL Server Management Studio)**. [S. 1.], 2020. Disponível em: <https://docs.microsoft.com/pt-br/sql/ssms/fl-help/database-engine-query-editor-sql-server-management-studio?view=sql-server-ver15>. Acesso em: 05 de nov. de 2020.

NAKAGAWA, Elisa Yumi. **Aspectos teóricos e empíricos de teste de cobertura de software**, pág. 3. São Paulo, 2017.

NEXT-GENERATION ORM for Node.js and TypeScript. **Prisma**. [S. I.], 2020. Disponível em: <https://www.prisma.io>. Acesso em: 01 de nov. de 2020.

SOFTWARES de Gestão para ESC. **Sebrae**. [S. I.], 2010. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/sebraeaz/software-de-gestao-para-esc,00961db818cfc610VgnVCM1000004c00210aRCRD>. Acesso em: 30 de out. de 2020.

SORDI, José Osvaldo de. **Benefícios da arquitetura de software orientada a serviço para as empresas: Análise da experiência do abn amro Brasil**, pág. 6. São Paulo, Santos, 2006.

SOUZA, César Alexandre de. **Sistema Integrados de Gestão Empresarial: Estudos de Casos de Implementação de Sistemas ERP**, pág. 40. São Paulo, 2000.

THE INTUITIVE Vue Framework. **Nuxtjs**. [S. 1.], 2019. Disponível em: <https://nuxtjs.org>. Acesso em: 01 de nov. de 2020.

THE PROGRESSIVE JavaScript Framework. **Vuejs**. [S. 1.], 2014. Disponível em: <https://vuejs.org>. Acesso em: 01 de nov. de 2020.

WHAT is Client-Server? **OmniSci**. [S. I.], 2020. Disponível em: <https://www.omnisci.com/technical-glossary/client-server>. Acesso em: 10 de nov. de 2020.

Agradecimentos

Agradeço aos professores Sebastião Ivaldo Carneiro Portela por ter nos proporcionado tempo, paciência, informações e ideias durante a produção do trabalho, ao Leonardo Alves da Costa por ter nos dado a ideia de desenvolver o sistema e por ter nos apoiado, ao Wilton Marinho Carneiro Souza que nos indicou possíveis melhorias e implementações no sistema e no documento e ao Valmir por ter nos proporcionado dicas para a melhora da documentação.