

Desenvolvimento de um Sistema Web para gestão de compras de insumos de manutenção em contratos de prestação de serviços

Gelson Iezzi de Medeiros Garcia
Universidade Federal Rural do Semi-
árido - UFERSA
gelson.garcia@ufersa.edu.br

José Cláudio Medeiros de
Lima
Claretiano - Rede de
Educação
klawdyo@gmail.com

Paulo Gabriel Gadelha Queiroz
Departamento de Computação
- Universidade Federal Rural
do Semi-árido
pgabriel@ufersa.edu.br

RESUMO

Com o avanço e popularização das tecnologias voltadas ao gerenciamento de recursos institucionais, bem como a crescente demanda por eficiência e agilidade na prestação de serviços, desenvolver atividades baseadas em processos manuais torna-se um gargalo que compromete a produtividade e a confiabilidade dos resultados obtidos. Além disso, o execução do processo fica refém de profissionais que detém conhecimentos específicos sobre seu desenvolvimento. Pensando nesse contexto, objetivou-se desenvolver um sistema web para automatizar e unificar os processos de aquisição de insumos por meio de contratos de manutenção da Diretoria de Manutenção e Serviços Gerais da Universidade Federal Rural do Semi-Árido. Portanto, neste artigo apresenta-se o processo de desenvolvimento do sistema, com a descrição dos métodos, ferramentas e tecnologias utilizadas. Além disso, apresenta-se o sistema desenvolvido e sua validação, por meio da análise de aceitação dos usuários.

Palavras-chave

Sistema Web; Desenvolvimento de Software; Manutenção; Gestão; Contrato

Keywords

Web System; Software development; Maintenance; Management; Contract

1. INTRODUÇÃO

A manutenção nas estruturas físicas de prédios públicos é uma demanda contínua e, na maioria das vezes, os serviços devem ser realizados de maneira ágil para não comprometer a integridade do patrimônio público ou o funcionamento das atividades finalísticas do órgão ou entidade, conforme

disposto no art. 15 da Instrução normativa nº 5, de 26 de maio de 2017 [4]. Entretanto, para executar esses serviços em tempo hábil é necessário dispor de uma ampla variedade de insumos de manutenção.

A legislação brasileira, de acordo com o Art. 2º da Lei nº 8.666, de 21 de junho de 1993 que regulamenta o art. 37, inciso XXI, da Constituição Federal, institui normas para licitações e contratos da Administração Pública e dispõe que as obras, serviços, inclusive de publicidade, compras, alienações, concessões, permissões e locações da Administração Pública, quando contratadas com terceiros, serão necessariamente precedidas de licitação, ressalvadas as hipóteses previstas nessa Lei. Esses processos costumam ser burocráticos e morosos.

Caso as universidades façam a aquisição de todos os insumos necessários para realização de cada serviço de manutenção específico, por meio dos trâmites de licitação, certamente o tempo necessário para a execução dos serviços comprometeria o funcionamento das atividades finalísticas, além de ser impraticável do ponto de vista econômico. Diante dessas condições, o órgão público pode fazer a aquisição desses materiais por meio de contratos de prestação de serviços de manutenção previamente licitados. Para isso, é necessário seguir regras e orientações dos órgãos de controle.

As universidades públicas brasileiras, que na maioria dos casos, precisam manter o funcionamento regular de um número considerável de edificações, enfrentam o desafio de gerenciar esses recursos com o menor custo possível. Para isso, a utilização de tecnologias da informação é indispensável para elevar o nível do gerenciamento de contratos de serviços que visam atender a essas necessidades.

Ao considerar o cenário da Diretoria de Manutenção e Serviços Gerais (DMSG) da Universidade Federal Rural do Semiárido – UFERSA, onde são prestados serviços de manutenção de bens imóveis à comunidade acadêmica, observa-se que esses serviços são realizados por meio de contratos com empresas que fornecem equipamentos, mão de obra especializada e, quando necessário, insumos indispensáveis à realização de tais demandas. As gestões desses contratos são realizadas pelos servidores efetivos da universidade que, atualmente, o fazem utilizando o Sistema Integrado de Patrimônio, Administração e Contratos (SIPAC).

Para orientar os processos de contratação de serviços de manutenção, a administração pública dispõe do Sistema Nacional de Pesquisa de Custos e Índices da Construção Civil -

Sinapi que mantém uma tabela de referência de preços médios praticados nos comércios locais. A partir de 2013, através do Decreto 7983/2013, o Sinapi foi definido como balizador de custos para serviços contratados com recursos do Orçamento Geral da União – OGU.

De acordo com o documento elaborado pela Caixa Econômica Federal [10], insumo é o elemento básico da construção civil constituídos de materiais (cimento, blocos, telhas, tábuas, aço, etc.), equipamentos (betoneiras, caminhões, equipamentos de terraplanagem, etc.) e mão de obra (pedreiro, pintor, engenheiro, etc.). Neste trabalho, utiliza-se a expressão insumos de manutenção para referir-se apenas aos materiais.

Entretanto, no que se refere especificamente aos contratos supracitados, observa-se que existem processos não contemplados pelo SIPAC que obrigam os gestores a utilizarem planilhas eletrônicas com rotinas complexas para executar os processos. O que muitas vezes gera problemas de inconsistência e descentralização de informações, retrabalho e dificuldades no gerenciamento e no controle dos processos executados. Essa problemática impacta diretamente nos resultados obtidos, como por exemplo, atrasos na execução de ordens de serviço e entaves nos repasses de verbas à empresa contratada. Ao considerar o contexto e os problemas mencionados, objetivou-se desenvolver um sistema web para automatizar e unificar o fluxo de aquisição de insumos através de contratos de manutenção. Portanto, neste artigo apresenta-se o processo de desenvolvimento do referido sistema e os resultados obtidos com a sua utilização.

O restante deste artigo está organizado da seguinte forma: na seção 2, apresentam-se os sistemas relacionados; na seção 3, apresenta-se a explicação dos procedimentos e métodos abordados; na seção 4, detalha-se as atividades e ferramentas utilizadas; na seção 5, são apresentados o sistema e os resultados deste trabalho; e, por fim, na seção 6 são apresentadas as considerações finais deste trabalho.

2. TRABALHOS RELACIONADOS

Nesta seção, são detalhadas as informações sobre sistemas que possuem alguma semelhança com o sistema proposto neste trabalho, chamado Sistema de Gestão de Compras de Insumos de Manutenção (SIGCIM). O levantamento bibliográfico foi realizado, principalmente, nas seguintes bases de pesquisa: Google Scholar, Portal de Periódicos CAPES e Scientific Electronic Library Online. A pesquisa foi realizada utilizando as seguintes palavras-chave: sistema web, contrato, manutenção, infraestrutura, imóveis, insumos de manutenção, gestão de manutenção. Os resultados foram filtrados pela data de publicação, sendo considerados apenas os trabalhos publicados a partir do ano de 2011.

A partir da análise da bibliografia, não foi encontrado nenhum sistema que implemente as funcionalidades propostas por este trabalho. Entretanto, foram identificados dois sistemas que apoiam as atividades de gerenciamento de contratos da administração pública. O Sistema Integrado de Patrimônio, Administração e Contratos (SIPAC) e o Sistema Unificado de Administração Pública (SUAP).

2.1 Sistema Integrado de Patrimônio, Administração e Contratos

O SIPAC foi desenvolvido pela Superintendência de Informática da Universidade Federal do Rio Grande do Norte (SINFO-UFRN) e teve sua primeira versão lançada em 2006.

Segundo Sousa (2016), é utilizado em mais de 30 instituições da rede de cooperação da Universidade.

Possui o módulo de Infraestrutura que atualmente é utilizado pela DMSG. Entretanto nenhuma funcionalidade do sistema permite gerenciar solicitações de compras de insumos por meio de contratos de manutenção. Suas funcionalidades são limitadas a realizar solicitações de insumos ao almoxarifado da instituição. O sistema permite vincular uma requisição de material à uma ordem de serviço, o que tem o potencial de oferecer nível bastante razoável de informações sobre os insumos consumidos e a relação destes com os serviços de infraestrutura.

2.2 Sistema Unificado de Administração Pública

O SUAP é um sistema desenvolvido pela equipe de desenvolvimento da Diretoria de Gestão de TI (DIGTI) do Instituto Federal de Educação Tecnológica do Rio Grande do Norte. Os desenvolvedores se motivaram pelos problemas recorrentes em executar os trâmites da administração pública por meio de vários sistemas e tecnologias. Dessa forma, a equipe iniciou o desenvolvimento de um sistema para unificar todos os processos em uma única plataforma. O desenvolvimento do sistema iniciou-se em 2006 por uma equipe de técnicos e estagiários da Coordenação de Sistemas de Informação (COSINF) que é subordinada à DIGTI. De acordo com Casadei (2018) o SUAP é desenvolvido na linguagem de programação Python, sob o *framework* Django, utiliza o banco de dados PostgreSQL, além de serviços como NGINX e Gunicorn. Atualmente dispõe de mais de 40 módulos e é utilizado por mais de trinta outros órgãos públicos, dos quais podemos destacar a Escola Nacional de Educação Pública (ENAP), o Ministério da Cultura (MinC) e a Superintendência do Desenvolvimento do Centro-Oeste (Sudeco).

O módulo de Gestão de Contratos dispõe das seguintes funcionalidades: gestão de cronograma e fiscais; livro de ocorrências; medição de parcelas e geração de relatórios. Entretanto, no módulo de contratos, o sistema não dispõe de nenhuma funcionalidade que seja específica para os contratos de manutenção. Além disso, ao avaliar a listagem de módulos no Apêndice A de Casadei (2018), não foi identificado nenhum módulo de gestão de manutenção de infraestrutura.

3. PROCEDIMENTOS E MÉTODOS

Após a análise do cenário, decidiu-se utilizar princípios de métodos ágeis, que segundo Sommerville (2018) consistem em: envolver o cliente no processo de desenvolvimento, para que ele possa fornecer e priorizar novos requisitos e fazer avaliação das iterações do sistema; realizar entregas incrementais para que os clientes possam especificar os requisitos a serem incluídos em cada incremento; desenvolver com foco nas pessoas, reconhecendo suas habilidades e desenvolvendo uma maneira própria de trabalhar; aceitar as mudanças dos requisitos do sistema ao longo do processo de desenvolvimento. Dessa forma, deve-se investir energias em projetar um sistema capaz de: acomodar mudanças; manter a simplicidade do sistema e do processo de desenvolvimento; e trabalhar ativamente para eliminar complexidades.

As justificativas para a escolha dos princípios ágeis incluem o pequeno porte do sistema e o nível de empenho observado entre os servidores da diretoria, no sentido de contribuir com o desenvolvimento. Desse modo, a cada iteração, eram realizadas entrevistas com os interessados, desenhados

protótipos para subsidiar implementações de pequenas partes do sistema, e, em seguida, esses pequenos fragmentos eram testados para garantir o cumprimento dos requisitos propostos.

O método adotado iniciou-se com a avaliação dos sistemas similares existentes no mercado e das tecnologias e ferramentas disponíveis para o desenvolvimento da solução. Em seguida, foram realizados ciclos de atividades que incluem delimitar uma fração do sistema por meio de entrevistas com *stakeholders*. Após a identificação, a equipe imerge nas atividades do processo eleito, analisando planilhas e documentos previamente utilizados para produzir protótipos de baixa fidelidade. Para Benyon (2011), um protótipo pode ser uma implementação ou representação concreta, apesar de, não contemplar todas as peculiaridades do design de um sistema. Quando os artefatos apresentam a elucidação dos requisitos, a funcionalidade é desenvolvida e em seguida submetida aos usuários para que eles possam fazer a validação. Caso a implementação corresponda aos requisitos propostos, um novo ciclo é iniciado, caso contrário, os passos anteriores são revisados pela equipe a fim de mitigar os desentendimentos.

Para gerenciar o projeto de desenvolvimento foi utilizado o aplicativo Trello. Essa ferramenta foi utilizada para implementar a metodologia Scrum que, segundo Schawber (1997) objetiva fornecer um processo flexível para o projeto e desenvolvimento de sistemas. O processo é dividido em *sprints*, ciclos de desenvolvimento, em que a cada iteração uma fração do sistema é desenvolvida e testada. Observa-se que o Scrum objetiva resultados realmente úteis para os interessados [14].

3.1 Validação do sistema

Para realizar a validação do sistema, foi utilizado o modelo *Technology Acceptance Model* (TAM) desenvolvido por Davis (1989) para oferecer uma explicação sobre as variáveis determinantes da aceitação de uma tecnologia. Esse estudo é uma adaptação do modelo *Theory of Reasoned Action* (TRA), ou teoria da ação racionalizada, que restringe sua aplicação a usuários de computadores ou de sistemas de informação. O objetivo do modelo é explicar o comportamento dos usuários de sistemas de informação de forma a antecipar a aceitação ou a invalidação dos usuários, além de explicar as causas determinantes da aceitação dos sistemas, orientando os interessados no sentido das correções necessárias.

Esse estudo mostrou que os principais determinantes do modelo de aceitação de tecnologia são: Utilidade percebida (UP) e Facilidade de uso percebida (FUP).

Utilidade percebida refere-se ao grau que um entrevistado acredita que utilizar uma tecnologia ou sistema pode melhorar o seu desempenho em um trabalho. O segundo conceito é a facilidade de uso percebida que se refere às expectativas de um usuário de que a utilização de determinado sistema ou tecnologia o exija o mínimo de esforço físico e mental.

Ao considerar o modelo TAM, a pesquisa de aceitação do SIGCIM foi realizada entre os servidores da DMSG que desempenham funções relacionadas às requisições de insumos de manutenção da UFERSA. Trata-se de um grupo com 5 usuários. A seguir, na Tabela 1 apresenta-se o questionário aplicado aos usuários após a utilização do sistema no ambiente de testes. O questionário foi dividido em 3 seções, nas duas primeiras foi utilizada a escala de Likert em que

o usuário especifica o seu nível de concordância ou discordância em relação a uma proposição. Na primeira parte do questionário foram avaliados os construtos relativos à utilidade percebida (UP); na segunda, os construtos relativos à facilidade de uso percebida (FUP); e, na última seção foi solicitado que os usuários dissertassem sobre suas percepções, dúvidas e possíveis contribuições para o sistema.

4. DESENVOLVIMENTO

4.1 Requisitos

A partir dos artefatos levantados na execução do método, foi identificado o processo de aquisição de insumos para realização de serviços de manutenção. Esses processos seguem um fluxo bem definido na DMSG e, além disso, foi constatado que outras instituições de ensino utilizam processos similares para lidar com essa situação.

Para compreender e definir o fluxo das atividades, foram realizadas entrevistas com os servidores da DMSG, além do conhecimento prévio do autor deste trabalho que também é membro da equipe.

O fluxo do processo de aquisição de insumos de manutenção pode ser observado no fluxograma apresentado na Figura 1, que foi desenhado utilizando o padrão Business Process Model and Notation (BPMN). Cada raia apresenta as atividades desenvolvidas por um responsável.

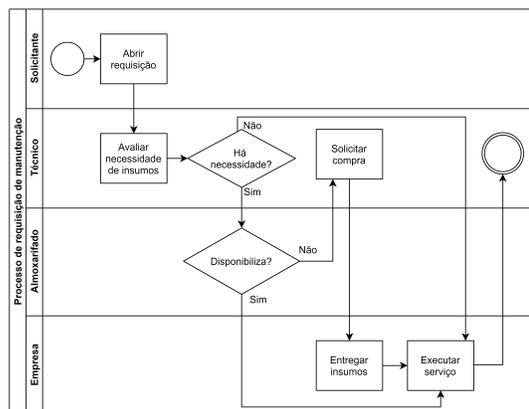


Figure 1: Fluxograma do processo de requisição de manutenção.

O ciclo inicia com a abertura de uma requisição. Em seguida um técnico especialista avalia a necessidade de insumos de manutenção para realização do serviço. Caso haja necessidade, os itens podem ser fornecidos pelo almoxarifado ou por meio de contrato de manutenção.

Caso o almoxarifado tenha disponibilidade dos itens necessários, o técnico solicita os itens ao almoxarifado, a requisição de manutenção é executada e o fluxo é finalizado. Em caso de indisponibilidade no estoque do almoxarifado, um outro fluxo é iniciado como pode ser observado na Figura 2. Neste caso, o técnico deverá realizar uma ordem de fornecimento dos materiais necessários para empresa contratada. Após a entrega dos itens, a requisição de manutenção é executada e o fluxo finalizado.

Table 1: Questionário de avaliação do sistema.

| Seção | Variável | Proposição |
|-------|-------------|--|
| 1 | UP1 | Aprender a usar o SIGCIM foi fácil. |
| 1 | UP2 | Utilizar o SIGCIM agrega valor ao meu trabalho. |
| 1 | UP3 | Utilizar o SIGCIM aumenta a minha produtividade. |
| 1 | UP4 | É fácil cadastrar novos insumos no SIGCIM. |
| 1 | UP5 | É fácil criar requisições de insumos de manutenção no SIGCIM. |
| 1 | UP6 | É fácil gerar a ordem de compra e enviá-la para a empresa contratada. |
| 2 | FUP1 | No SIGCIM eu sempre sei como realizar as atividades que eu preciso. |
| 2 | FUP2 | Acho complicado usar o SIGCIM. |
| 2 | FUP3 | O SIGCIM apresenta uma intenção compreensível e clara. |
| 2 | FUP4 | Os recursos de navegação do SIGCIM são fáceis de localizar. |
| 2 | FUP5 | O SIGCIM possui uma interface amigável. |
| 2 | FUP6 | É necessário muito esforço para utilizar o SIGCIM. |
| 2 | FUP7 | É necessário treinamento para iniciar o uso do SIGCIM. |
| 2 | FUP8 | Utilizar o SIGCIM é agradável. |
| 2 | FUP9 | Consigo utilizar o SIGCIM sem auxílio de um instrutor. |
| 3 | Importância | Você acha importante a diretoria de manutenção adotar o SIGCIM para as requisições de manutenção? Se sim, por quais motivos? |
| 3 | Dúvidas | Você teve alguma dúvida ao utilizar o sistema? Se sim, qual(is)? |
| 3 | Proposições | Você tem alguma proposição de melhoria? Se sim, disserte sobre ela (s). |

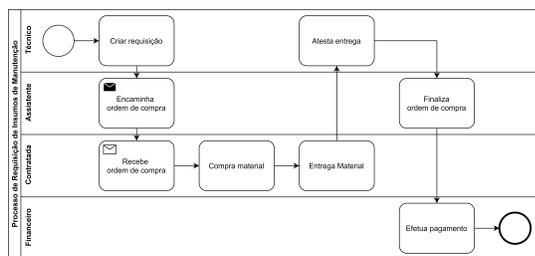


Figure 2: Fluxograma do processo de requisição de insumo de manutenção por meio de contrato.

Ao final de um ciclo mensal, uma lista dos itens adquiridos por meio do contrato de manutenção é anexada ao processo de faturamento para que os valores sejam repassados para a empresa contratada. Esse processo visa o princípio da eficiência e da economicidade, que segundo [1], zelam, respectivamente, por aperfeiçoar a alocação dos recursos públicos nas contratações e por alcançar a alternativa mais vantajosa do ponto de vista econômico.

Após compreender o fluxo do processo, foi escrito um documento de requisitos inicial, apresentado na Tabela 2. Cada requisito foi classificado como funcional (F) e não funcional (NF) em uma das três categorias a seguir: Essencial (E) - indispensável ao funcionamento do sistema; Importante (I) - torna o sistema mais completo, mas podem ser descartados conforme o cronograma; Secundário (S) - que não necessitam ser atendidos obrigatoriamente.

4.2 Protótipos

Foram desenvolvidos protótipos que consistiam em representações das telas do sistema. Esses protótipos serviram de apoio para as reuniões iniciais em cada etapa do desenvolvimento. As representações foram relevantes para a concepção dos problemas e possibilitaram a coleta de informações sobre as expectativas dos clientes ao interagir com algum componente do sistema como, por exemplo, o clique em um botão,

ou em algum *hyperlink*.

Em cada iteração, pelo menos, um protótipo era desenhado e em uma das iterações mais importantes foi desenhado o protótipo apresentado na Figura 3, que elucidou um dos requisitos centrais do sistema. O protótipo consiste na tela de criação de uma requisição de insumos de manutenção a partir de uma base de dados previamente importada.



Figure 3: Protótipo da tela de criação de requisição de insumos.

4.3 Tecnologias e ferramentas

Um dos fatores que mais impactou na escolha das tecnologias foi a popularidade. Pois, a partir de fóruns e comunidades de discussão ativos o suporte às resoluções de problemas e extinção de dúvidas são facilitados. Além disso, foram privilegiadas as tecnologias voltadas ao desenvolvimento ágil. Dessa forma, foi escolhida a linguagem de programação JavaScript que é uma linguagem interpretada, originalmente desenvolvida por Brendan Eich para tonar páginas web mais

Table 2: Requisitos elicitados.

| # | Requisito | Categoria | Classe | Implementado |
|----|---|-----------|--------|--------------|
| 1 | Níveis de acesso | E | NF | SIM |
| 2 | Armazenar dados de contratos | E | F | SIM |
| 3 | Armazenar dados de insumos | E | F | SIM |
| 4 | Armazenar informações de bases de dados | E | F | SIM |
| 5 | Importar insumos da base de dados SINAPI | E | F | SIM |
| 6 | Importar insumos de outras bases de dados | I | F | NAO |
| 7 | Armazenar dados de orçamentos | S | F | NAO |
| 8 | Realizar cotação de preços de um item | S | F | NAO |
| 9 | Criar requisições de insumos de manutenção | E | F | SIM |
| 10 | Registrar entrega de insumos de manutenção | E | F | SIM |
| 11 | Acompanhar ordens de compra de insumo | E | F | SIM |
| 12 | O sistema deverá ser desenvolvido totalmente com tecnologias livres | E | NF | SIM |
| 13 | Seguir as orientações da legislação e das normas vigentes | E | NF | SIM |

interativas [9]. Observa-se que a linguagem evoluiu, e atualmente, ocupa outros contextos, sendo utilizada também no *backend*. A seguir, são apresentadas as tecnologias utilizadas no sistema.

4.3.1 Node.js

Node.js, também chamada de Node, é uma plataforma para o desenvolvimento de aplicações Web que utiliza o JavaScript Engine V8 do Google [13]. Dessa forma, oferece uma maneira fácil, rápida e escalável de desenvolver aplicações Web. Uma grande vantagem do Node é que utiliza uma única linguagem tanto no lado do servidor, quanto no lado do cliente, facilitando o desenvolvimento de sistemas completos. Diferente de outros ambientes de desenvolvimento, Node opera apenas em *single-thread* utilizando uma tecnologia de operações de I/O não bloqueantes, de modo que dispensa as complexidades do gerenciamento de um sistema *multithreading*.

4.3.2 MongoDB

Para realizar a persistência de dados, foi utilizado o banco de dados MongoDB que é orientado a documentos e projetado para facilitar o desenvolvimento e a escalabilidade. Utiliza o modelo de estrutura de dados não relacional em que os dados são armazenados em coleções de arquivos que utilizam uma estrutura composta por pares de campo e valor possibilitando aninhar outros documentos e listas semelhante aos objetos JavaScript Object Notation (JSON) [15], como pode ser observado na Figura 4. Uma vantagem é que os documentos do MongoDB são harmônicos aos objetos JavaScript [12]. Podemos fazer uma relação entre bancos de dados relacionais e não relacionais em que as coleções correspondem às tabelas e os documentos correspondem às linhas.

```

{
  name: "sue",           ← field: value
  age: 26,              ← field: value
  status: "A",          ← field: value
  groups: [ "news", "sports" ] ← field: value
}

```

Figure 4: Formato dos documentos do MongoDB [12].

As características do MongoDB incluem alta performance e disponibilidade, uma rica linguagem de consultas, escalabilidade horizontal e suporte a múltiplos mecanismos de armazenamento.

4.3.3 Express

Express é um *framework* para a construção de servidores Web e interfaces de programação de aplicações (API) com Node que promete ser rápido, flexível e minimalista. Segundo Dickey (2015), é o *framework* mais usado para construção de aplicações Web com Node. Atualmente empresas como UBER, IBM, Fox Sports e outras multinacionais mantêm serviços Web construídos sobre o Express.

4.3.4 Bootstrap

O Bootstrap [3] é um conjunto de ferramentas de código aberto desenvolvida em HTML, CSS e JavaScript para desenvolvimento de aplicações responsivas. Atualmente, está na versão 4.4 e oferece uma extensiva quantidade de componentes pré-construídos e plug-ins poderosos construídos com jQuery que permitem a construção rápida de aplicações.

4.3.5 VueJS

O VueJS [17] é um *framework* progressivo de código aberto construído em JavaScript e foi incluído no projeto para facilitar a manipulação de objetos da árvore DOM e para agilizar o desenvolvimento da interface do usuário. A biblioteca principal do VueJS tem foco na camada visual, o que a torna de fácil integração com outras bibliotecas já utilizadas no projeto. O VueJS possibilita interligar os dados da aplicação aos objetos renderizados na tela do cliente, tornando-a reativa. Isso significa dizer que, ao alterar algum dado da aplicação automaticamente a informação é alterada em tempo real na página exibida ao cliente.

4.4 Arquitetura

Nesta seção será apresentada uma visão lógica da arquitetura do sistema que se baseou no modelo *Model View Control* (MVC) [11]. O sistema foi dividido em módulos para agrupar os requisitos, e dessa forma, facilitar o desenvolvimento de cada fração do sistema. Na Figura 5, apresenta-se a descrição da arquitetura lógica do sistema utilizando um diagrama de pacotes.

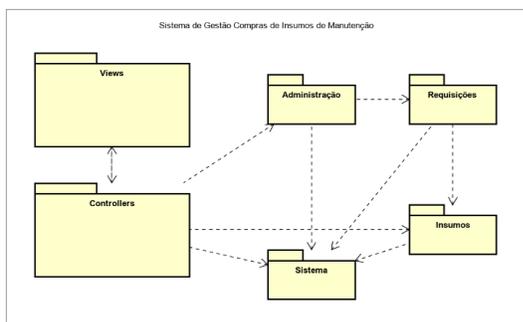


Figure 5: Arquitetura lógica do sistema.

O módulo de Insumos é responsável pela gestão de todas as informações e procedimentos relacionados aos insumos de manutenção. É composto pelas classes Insumo, Base de dados, Orçamento e Cotação. Esse módulo é responsável pela implementação dos requisitos relacionados a gestão de insumos e bases de dados conforme os requisitos de 3 a 8 listados na Tabela 2.

O módulo de Administração é responsável por implementar as funcionalidades de gestão de contratos, empresas, endereços de entrega e as informações dos usuários.

O módulo de Requisições é responsável pelas funcionalidades referentes ao gerenciamento de requisições e ordens de compra, bem como a entrega dos materiais. O módulo de Sistema é responsável por gerenciar as opções relativas ao sistema.

4.5 Resultados

4.5.1 Apresentação do Sistema

Com o objetivo de familiarizar o usuário ao sistema, a tela principal apresenta uma breve introdução sobre as funcionalidades. As informações aparecem separadas por módulos, conforme apresentado na Figura 6.



Figure 6: Tela principal do sistema.

A interface do sistema dispõe de um menu superior de navegação onde estão dispostas as funcionalidades separadas por categoria. Cada categoria, ao ser clicada, apresenta uma lista suspensa na qual é possível acessar as interfaces de suas funcionalidades. Do lado direito da barra de navegação é possível visualizar a matrícula do usuário ativo no sistema, conforme apresentado na Figura 7.



Figure 7: Menu de navegação.

As funcionalidades relativas ao gerenciamento de insumos de manutenção permitem realizar as quatro operações básicas de bancos de dados, popularmente conhecido por CRUD (acrônimo do inglês *Create, Read, Update and Delete*). Além disso, esse módulo foi projetado para possibilitar o cadastro de bases de dados que mantêm registros de preços praticados no mercado para posteriormente fazer a importação dos respectivos itens por meio de upload de planilhas eletrônicas fornecidas por sistemas de pesquisa de preços. A seguir, na Figura 8, apresenta-se a tela de listagem de insumos.



Figure 8: Tela de listagem de insumos de manutenção.

Na Figura 9, apresenta-se a tela de requisições. Nessa tela o usuário pode acompanhar as requisições que estão em processo de criação. A partir dessa tela, utilizando os ícones da coluna mais a direita da tabela de requisições, é possível, respectivamente: continuar trabalhando na requisição; encaminhar a ordem de compra para a empresa; e, excluir a requisição.

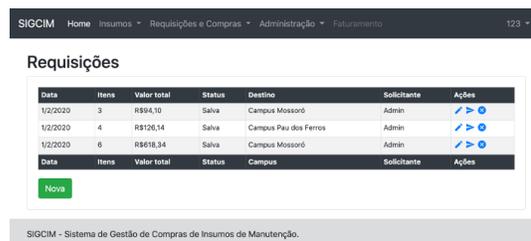


Figure 9: Tela de acompanhamento de requisições.

A tela de acompanhamento de ordens de compra, apresentada na Figura 10, permite que os usuários visualizem o estado de cada solicitação. Os usuários com perfil Técnico são habilitados a cadastrar a entrega dos insumos. As ordens de compra têm o estado alterado para entregue quando todos os seus itens forem atestados como recebidos. A partir de então, a ordem de compra está apta a ser encaminhada para faturamento.

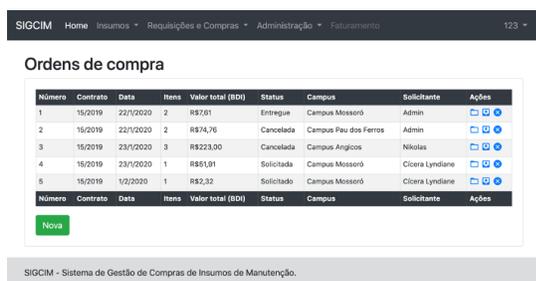


Figure 10: Tela de acompanhamento de ordens de compra.

Para ter acesso às funcionalidades, os usuários são classificados em perfis. Cada perfil permite que o usuário tenha acesso a um grupo de funcionalidades. O perfil denominado Contratada permite acesso somente a funcionalidades relativas ao acompanhamento de ordens de compras em que é possível acompanhar as ordens enviadas, além de, os itens entregues e pendentes. O perfil Técnico permite que o usuário acesse funcionalidades referentes ao cadastro e gerenciamento de insumos, cadastro de requisições e geração de ordens de compra conforme apresentado na Figura 11. Esse perfil também permite o cadastro da entrega dos itens solicitados. O perfil Assistente tem acesso ao gerenciamento de contratos e empresas, de bases de dados de insumos de manutenção e de usuários. Além dos perfis anteriores, existe o perfil Admin, que é projetado para ter acesso a todas opções e funcionalidades do sistema.

O sistema permite gerenciar aspectos administrativos dos processos, como, por exemplo: as informações de empresas contratadas, bem como dos contratos de manutenção; endereços de locais de entrega dos itens; informações sobre usuários; e outras opções.

4.5.2 Resultados da validação

O questionário de validação foi aplicado a três servidores, após a utilização do sistema em ambiente de testes. A seguir, apresenta-se a Tabela 3, em que é possível avaliar pela média das respostas que o sistema obteve uma boa avaliação em relação aos construtos de Utilidade Percebida e Facilidade de Uso Percebida.

Observa-se que a avaliação das perguntas FUP2 e FUP3 foram abaixo da média. Entretanto, os questionamentos referem-se, respectivamente, a quanto o usuário avalia ser complicado e a quantidade de esforço necessário para utilizar o sistema. Dessa forma, pode-se concluir que as respostas também foram positivas.

Table 3: Resultado do questionário de validação.

| Variável | Usuário 1 | Usuário 2 | Usuário 3 | Média |
|----------|-----------|-----------|-----------|-------|
| UP1 | 5 | 5 | 4 | 4,7 |
| UP2 | 5 | 5 | 5 | 5,0 |
| UP3 | 5 | 5 | 5 | 5,0 |
| UP4 | 4 | 5 | 5 | 4,7 |
| UP5 | 5 | 5 | 3 | 4,3 |
| UP6 | 5 | 5 | 5 | 5,0 |
| FUP1 | 5 | 5 | 3 | 4,3 |
| FUP2 | 1 | 5 | 1 | 2,3 |
| FUP3 | 5 | 5 | 4 | 4,7 |
| FUP4 | 4 | 5 | 5 | 4,7 |
| FUP5 | 4 | 5 | 3 | 4,0 |
| FUP6 | 1 | 2 | 2 | 1,7 |
| FUP7 | 2 | 5 | 5 | 4,0 |
| FUP8 | 5 | 5 | 5 | 5,0 |
| FUP9 | 5 | 5 | 4 | 4,7 |

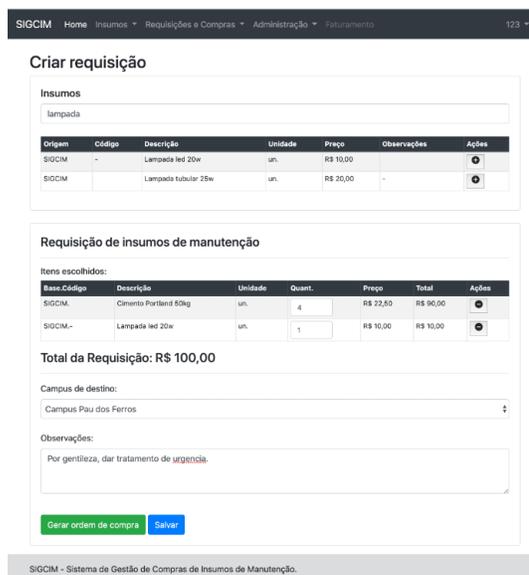


Figure 11: Tela de criação de requisição.

Em relação à seção 3 do questionário, em que os usuários foram convidados a dissertar sobre a importância, outras percepções e dúvidas na utilização do sistema, obtivemos as respostas dispostas na Tabela 4.

5. CONCLUSÕES

Este trabalho foi desenvolvido com o objetivo de melhorar o processo de requisição de insumos por meio de contratos de manutenção da DMSG. Dessa forma, foi desenvolvido um sistema web que oferece uma interface visual que contempla as fases iniciais do processo, tornando o trabalho da equipe mais produtivo e objetivo. O sistema implementado é composto por um servidor web que processa as requisições dos clientes e realiza a persistência dos dados em um sistema de gerenciamento de banco de dados não-relacional. A partir da implantação da solução, os problemas de ambiguidades de informações tendem a ser mitigados, visto que todas as informações do sistema são armazenadas em uma única estrutura, o que confere maior confiabilidade às informações

Table 4: Seção 3 do questionário de validação do sistema.

| Variável | Usuário 1 | Usuário 2 | Usuário 3 |
|-------------|---|--|---|
| Importância | Agilizar o processo com maior segurança e organização das requisições e ordens de compra. | Sim. Diminui o tráfego de planilhas entre e-mails e a possibilidade de erros | Sim. Para facilitar o tramite dos processos de compra. |
| Dúvidas | Não | Não | Sim. Dúvidas em relação a algumas funcionalidades que estão bloqueadas no menu: faturamento, orçamento e cotações. |
| Proposições | Não | Melhorar a visualização para dispositivos móveis. | Sim. Criação de quadros na tela principal para apresentar informações sobre as funcionalidades do sistema ao usuário. |

gerenciadas.

A partir da observação do questionário de validação é possível inferir que, apesar do número reduzido de participantes, o sistema obteve boa aceitação. As respostas subjetivas mostram que o sistema precisa de acertos e melhorias sobretudo no sentido de apresentar aos usuários mais detalhes sobre as suas funcionalidades.

Devido à complexidade das atividades e ao cronograma do trabalho, foi necessário definir um escopo para delimitar as funcionalidades desenvolvidas. Entretanto, para ampliar o suporte às atividades da diretoria é importante implementar outras funcionalidades e ampliar as que já foram implementadas. De modo que tais demandas podem ser alvo de trabalhos futuros, como, por exemplo, a geração de relatórios, a importação automática de bases de dados, geração de cotações a partir de orçamentos coletados no mercado local, integração com o sistema utilizado atualmente pela universidade, além de outras funcionalidades que podem ser demandadas.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] L. M. Araújo and M. I. A. Rodrigues. A relação entre os princípios da eficiência e da economicidade nos contratos administrativos. *Revista do Serviço Público*, 63(1):43–62, Jan. 2014.
- [2] D. Benyon. *Interação humano-computador*. Pearson Prentice Hall, São Paulo, 2011.
- [3] Bootstrap team. *Get started with Bootstrap, the world's most popular framework for building responsive, mobile-first sites, with jsDelivr and a template starter page.*, 2020.
- [4] Brasil. Instrução normativa nº 5, de 26 de maio de 2017. *Diário Oficial [da] República Federativa do Brasil*, 2017.
- [5] T. L. Casadei. Avaliação arquitetural do sistema suap: uma análise sistematizada sobre desempenho. Mestrado profissional em engenharia de software, Universidade Federal do Rio Grande do Norte, 2018.
- [6] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw. User acceptance of computer technology: A comparison of two theoretical models. *Management Science*, 35(8):982–1003, Aug. 1989.
- [7] A. G. de Sousa. Design de interfaces e arquitetura da informação em sistemas de gestão acadêmica: redesign do sipac-ufrrn. Mestrado profissional em design, Universidade Federal do Rio Grande do Norte, 2016.
- [8] J. Dickey. *Write modern web apps with the MEAN stack : Mongo, Express, AngularJS, and Node.js*. Peachpit Press, Berkeley, CA, 2015.
- [9] ECMA International. *Standard ECMA-262 - ECMAScript Language Specification*, 2011.
- [10] C. E. Federal. Sinapi: Metodologias e conceitos: Sistema nacional de pesquisa de custos e Índices da construção civil., 2019.
- [11] A. Leff and J. Rayfield. Web-application development using the model/view/controller design pattern. In *Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference*, pages 118–127, 2001.
- [12] MongoDB, Inc. *The MongoDB 4.2 Manual*, 2019.
- [13] OpenJS Foundation. *Node.js v12.22.1 Documentation*.
- [14] K. Schwaber. SCRUM development process. In *Business Object Design and Implementation*, pages 117–134. Springer London, 1997.
- [15] B. Smith. *Beginning JSON*. Apress, USA, 1st edition, 2015.
- [16] I. F. Sommerville. *Engenharia de Software*. Pearson Education do Brasil, São Paulo, 10 ed. edition, 2018.
- [17] Vue. Guia. <https://br.vuejs.org/v2/guide/>. [Documentação da Tecnologia].