INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO NORTE

THALES AZEVEDO SILVA

SISTEMA DE ADMINISTRAÇÃO DE RECURSOS PATRIMONIAIS DESTINADO AO SUPORTE DE COMPUTADORES E PERIFERÍCOS DA UGTSIC/SESAP

NATAL

2022

THALES AZEVEDO SILVA

SISTEMA DE ADMINISTRAÇÃO DE RECURSOS PATRIMONIAIS DESTINADO AO SUPORTE DE COMPUTADORES E PERIFERÍCOS DA UGTSIC/SESAP

Trabalho de Conclusão de Curso apresentado ao Curso Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientadora: M.ª Danielle Gomes de Freitas Medeiros.

NATAL

2022

Silva, Thales Azevedo.
S586s Sistema de administração de recursos patrimoniais destinado ao suporte de computadores e periféricos da UGTSIC/SESAP. – 2022.
45 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Natal, 2022.
Orientadora: M.ª Danielle Gomes de Freitas Medeiros.

1. Desenvolvimento de software. 2. Sistema de Administração de Recursos Patrimoniais – SARP. 3. Programação PHP. 4. Adianti Framework – Ferramenta de programação. I. Título.

CDU: 004.42

Catalogação na Publicação elaborada pelo Bibliotecário Joel de Albuquerque Melo Neto – CRB-15/320
Biblioteca Central Sebastião Fernandes (BCSF) – IFRN

THALES AZEVEDO SILVA

SISTEMA DE ADMINISTRAÇÃO DE RECURSOS PATRIMONIAIS DESTINADO AO SUPORTE DE COMPUTADORES E PERIFERÍCOS DA UGTSIC/SESAP

Trabalho de Conclusão de Curso apresentado ao Curso Tecnologia de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientadora: M.ª Danielle Gomes de Freitas Medeiros.

Trabalho de Conclusão de Curso aprovado em 25/08/2022 pela seguinte Banca Examinadora:

gov.br Documento assinado digitalmente
DANIELLE GOMES DE FREITAS MEDEIROS
Data: 27/06/2023 09:09:53-0300
Verifique em https://validar.iti.gov.br

M.a. Danielle Gomes de Freitas Medeiros – Orientadora

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

gov.br Documento assinado digitalmente
LEONARDO ATAIDE MINORA
Data: 14/07/2023 08:55:48-0300
Verifique em https://validar.iti.gov.br

M.e. Leonardo Ataide Minora

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

gov.br Documento assinado digitalmente
DANIEL WALMIR DOS SANTOS ALVES
Data: 27/06/2023 21:16:35-0300
Verifique em https://validar.iti.gov.br

Esp. Daniel Walmir dos Santos Alves

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

gov.br Documento assinado digitalmente
DEMOSTENES SANTOS DE SENA
Data: 27/06/2023 10:16:17-0300
Verifique em https://validar.iti.gov.br

Dr. Demostenes Santos de Sena

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

RESUMO

O Sistema de Administração de Recursos Patrimoniais (SARP) trata-se de uma aplicação web desenvolvida para o setor Unidade de Gestão de Tecnologia e Sistemas de Informação e Comunicação (UGTSIC) voltado a Secretária de Saúde Pública do Rio Grande do Norte (SESAP). Este propõe-se a substituir sua ferramenta antecessora Gestão de Patrimônio de Informática (GESPATI) que se tornou defasada, devido à falta de atualizações corretivas e novas implementações mediante necessidades da seção de manutenção, tornando-o obsoleto. Para tal, o SARP tem por intuito, permitir aos técnicos de manutenção, o diagnóstico dos computadores e ou periféricos mediante memorando solicitado pelo setor de origem, a fim de determinar a causa do problema e por consequência a abertura de ordem de serviço para iniciar os reparos ou alienação do equipamento por meio de descarte direto ou retorno a sua origem sem alterações. Cabe ainda ao sistema, efetuar o levantamento desses dispositivos, via relatórios, possibilitando o recebimento e a devolução de maneira segura e adequada. Este trabalho consiste entre as etapas de apresentação do setor de estágio e sua organização de trabalho, em seguida, o referencial teórico abrangendo sua documentação e as principais tecnologias utilizadas, concluindo com a etapa de desenvolvimento, especificando sua implementação, mediante funcionalidades essenciais e finalizando com o processo de implantação do sistema final.

Palavras-chave: sistema de administração de recursos patrimoniais; aplicação web; implementação; implantação.

ABSTRACT

The Administration System of Patrimonial Resources (ASPR) is a web application developed for the Technological Management Unit of Information Systems and Communication (TMUISC) from Public Health Secretary of Rio Grande do Norte (PHSR). Which proposes, the replacement of its ancestor tool, Patrimonial Management of information (PMI), considered outdated, for not having new corrective updates and new implementations according with the necessaries related to the computer maintenance department. The ASPR goal is, allowing the technician, do the diagnoses in the peripherals and its computers, relating it to a memo asked by the origin department, so it can be determining the probable causes and consequently open a service order, initiating repairs or declaring the equipment non-fixable, discarding directly, or sending it back to where it came from. It's also relevant to have filled the information related to those equipment's to receive and giving it back securely. The document, also consists, between steps, of the internship sector presentation and its organization, as well as the theorical referential exploring the documentation's work, and it's mainly approached technologies used, concluding by acknowledging the development process, specifying implementation, with its essential functionalities and wrapping it up with the implantation of the final product.

Keywords: administration system of patrimonial resources; web application; implementation; implantation.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE SIGLAS

SESAP	Secretaria de Saúde Pública do Rio Grande do Norte
UGTSIC	Unidade de Gestão Tecnologia de Sistemas de Informação e Comunicação
SABDP	Sistema de Acompanhamento Básico de Desenvolvimento de Projetos
GESPATI	Gestão de Patrimônio de Informática
SARP	Sistema de Administração de Recursos Patrimoniais

SUMÁRIO

1 INTRODUÇÃO

Este trabalho tem como objetivo relatar a experiência do autor durante o estágio curricular realizado na Secretaria de Saúde Pública do Rio Grande do Norte (SESAP), para o setor Unidade de Gestão de Tecnologia e Sistemas de Informação e Comunicação (UGTSIC) entre os períodos de 01 de março a 22 de novembro de 2021, atuando na seção de desenvolvimento com propósito de trabalhar em novos sistemas e dar suporte aos pré-existentes.

Serão descritos os detalhes do sistema principal, desenvolvido em parceria com o setor, constituído como o software a qual o autor mais se dedicou, a fim de relatar o processo de gerenciamento do projeto, suas especificações técnicas, visão geral abordando suas finalidades, validação por meio de reuniões, funcionalidades mais significativas e seu processo de implementação ao abordá-lo, mediante as necessidades do cliente e sua meta final.

O processo de desenvolvimento do Sistema de Administração de Recursos Patrimoniais (SARP) serve como substituto ao seu antecessor Gestão de Patrimônio de Informática (GESPATI) que gerenciava a movimentação de equipamentos destinados a seção de manutenção, porém com sua defasagem, devido à falta de atualizações corretivas e novas implementações mediante necessidades da seção de manutenção, este se tornou obsoleto.

Dessa forma, iniciou-se o processo de implementação a partir da linguagem de programação *PHP* e *Adianti framework* com objetivo de padronizar o novo sistema aos moldes do seu antecessor, aplicando novas utilidades mediante a interface dos templates disponibilizados pelo *Framework.*

Portanto, o SARP trata-se de uma aplicação web responsável por administrar a entrada e saída de equipamentos permitindo o seu cadastro, visualização de detalhes, criação e edição de memorandos descrevendo o diagnóstico e processos que deverão ser aplicados durante manutenção, ordens de serviço que são responsáveis por iniciar o processo de reparo dos dispositivos e descrição de todo o processo.

Durante o processo de análise, caso o diagnóstico não seja viável de reparo, o equipamento passa a alienação direta desses equipamentos, definidos como irrecuperáveis, mediante falta de peças ou curtos de placa mãe que por consequência são destinados a seu setor de origem ou descartados. Ressaltando que há a elaboração de relatórios desses aparelhos de acordo ao período dos serviços prestados, levantamento de quantidade, tipos, ajustes e detalhes específicos mediante período desejado.

1.1 OBJETIVO GERAL

Este documento tem por finalidade abordar o desenvolvimento do sistema SARP abrangendo sua análise, através da descrição de seus fluxos e diagramação, especificar as linguagens e frameworks utilizados durante o seu desenvolvimento abordando suas principais funcionalidades, bem como, seu código fonte.

1.2 OBJETIVOS ESPECÍFICOS

- Especificar o domínio e o processo de gerenciamento dos projetos da UGTSIC;
- Abordar a documentação e as tecnologias utilizadas para construção da aplicação;
- Apresentar o processo de implementação mediante código e suas funcionalidades;

1.3 ESTRUTURA DO TRABALHO

Na seção 2, o texto faz uma breve apresentação do setor público UGTSIC/SESAP, especificando a organização e as ferramentas de gestão dos sistemas e tarefas. A seção 3, explana o referencial teórico, detalhando as tecnologias utilizadas, assim como, a documentação presente no processo de análise. Em seguida, a seção 4 aborda a implementação do sistema ao apresentar suas funcionalidades e seu código fonte. Por fim, nas seções 5 e 6 são feitas as considerações finais e a declaração das referências bibliográficas.

2 UNIDADE DE GESTÃO DE TECNOLOGIA E SISTEMAS DE INFORMAÇÃO E COMUNICAÇÃO (UGTSIC)

A subcoordenadoria UGTSIC é responsável por organizar os processos tecnológicos, a segurança de dados e o fluxo de atividades ligadas aos computadores da secretaria de saúde em função de suas unidades espalhadas por todo o estado. Sua seção principal consiste pelo desenvolvimento, a fim de atender as demandas relacionadas aos sistemas de saúde, repassadas pelo ministério e a secretária de saúde que se comunicam com a SESAP.

O setor ainda se estendem a seção de redes, responsável pela segurança e infraestrutura da conexão local e as suas unidades correspondentes a SESAP, gerenciando seus servidores, sistemas de aplicações, prestação de apoio técnico em função da implantação de sistemas de informação e suporte nas áreas de redes a nível central e as demais unidades.

Por fim, a seção de manutenção acompanha de forma direta o processo de reparo de equipamentos, através da abertura de chamados aos dispositivos via garantia ou contrato, mediante diagnóstico, o equipamento é alienado ou reparado para então ser descartado ou retornado ao seu setor destinatário.

2.1 GERENCIAMENTO E CONTROLE DE PROJETOS UGTSIC

Durante o processo de desenvolvimento de software, seja este um sistema construído do zero ou um sistema que tenha necessidade de uma reformulação, é importante que haja um plano de ação que impacte positivamente na maneira em que a equipe se conduzirá e administrará as tarefas mediante os diversos sistemas existentes.

Na seção de desenvolvimento os programadores são atribuídos aos sistemas considerando que o desenvolvedor principal tenha maior conhecimento da tecnologia a qual será abordada e por consequência auxiliar o desenvolvedor júnior.

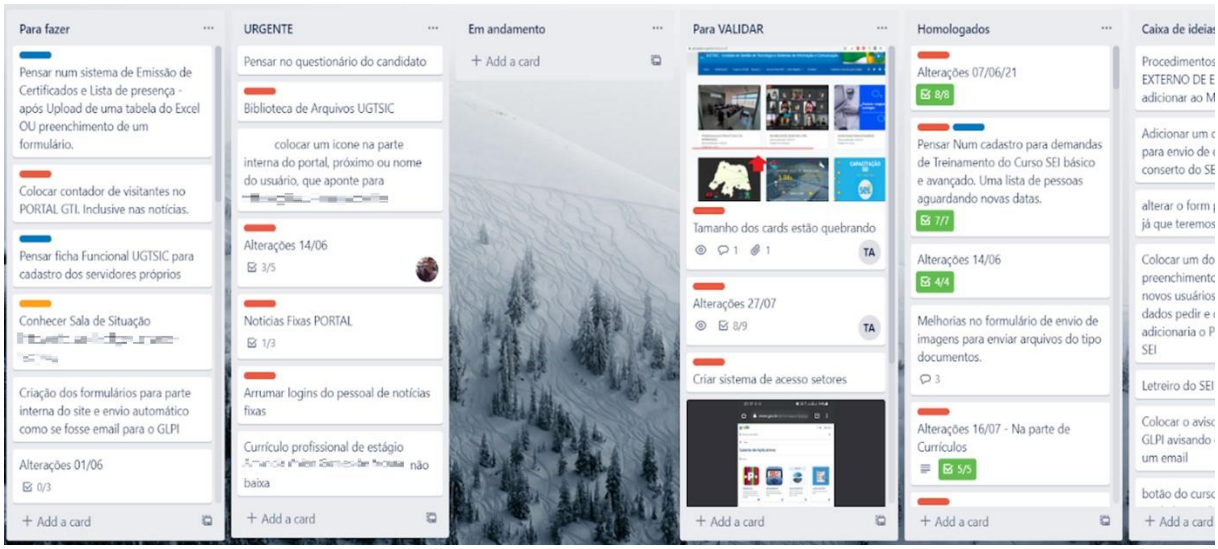
Contudo, para que o processo saia como definido e as entregas sejam estabelecidas, é necessário que presencialmente ou de maneira remota haja reuniões com os líderes dos setores correspondentes da implementação para uma melhor compreensão do problema e como poderão ser abordadas soluções ao software em foco.

2.1.1 Trello

Com a definição das tarefas a serem realizadas, estas são registradas e destinadas, a equipe mediante ferramenta de gerenciamento. Originalmente, esta seria uma etapa

administrada pelo Trello, ao criar um painel de controle dos projetos permitindo personalizar os fluxos de trabalho do grupo. Dessa maneira, era estimado as urgências das tarefas, os responsáveis, prazos, anotações e descrição de acordo a Figura 1.

Figura 1: Painel de Gerenciamento de Projetos no Trello

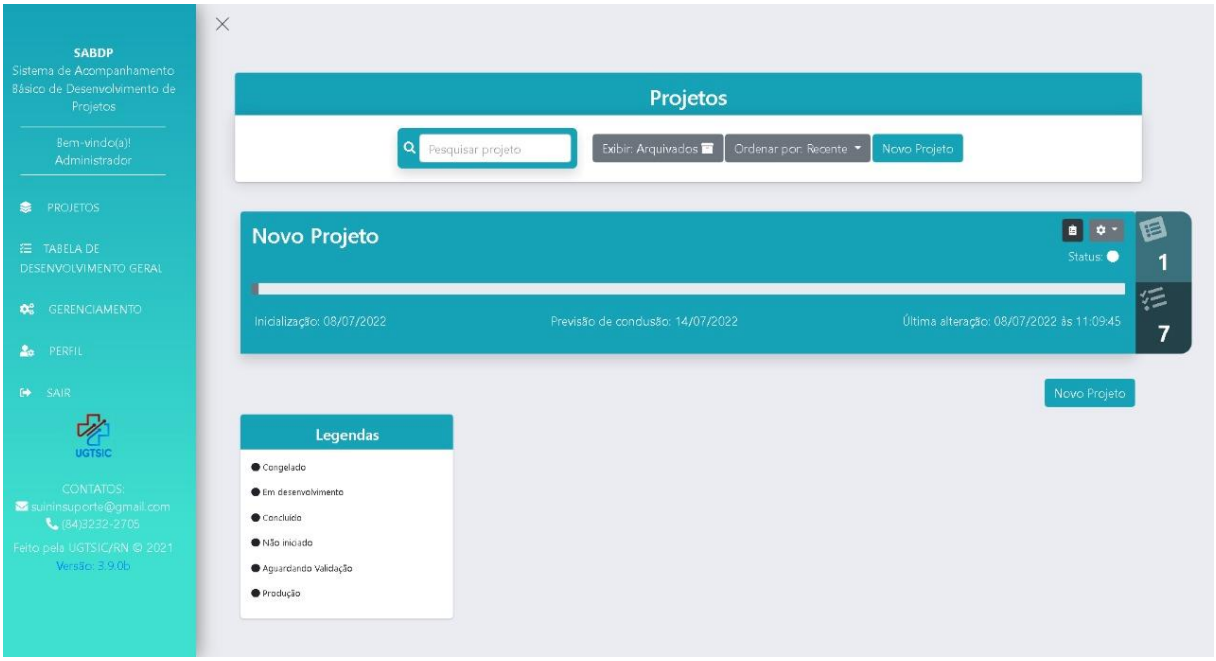


Fonte: UGTSIC/SESAP (2022)

2.1.2 Sistema de Acompanhamento Básico de Desenvolvimento de Projetos

Posteriormente, por questões de controle e segurança das informações de maneira a assegurar as informações descritas de forma, que tais dados, não fossem acessados externamente do setor UGTSIC foi introduzida uma nova ferramenta de gerenciamento o Sistema de Acompanhamento Básico de Desenvolvimento de Projetos (SABDP), representado pela Figura 2, cujas funcionalidades, se constituem de criar tarefas relacionadas aos projetos, tempo de execução, legendas que sinalizam o andamento de desenvolvimento, busca por projetos específicos, histórico por arquivamento, criação de novos projetos para gerenciamento, a fim de, proporcionar uma visão geral desses sistemas melhorando a visibilidade do escopo e o seu andamento.

Figura 2: Tela de gerenciamento SABDP



Fonte: UGTSIC/SESAP (2022)

No âmbito administrativo, o SABDP se fez muito necessário não só para os desenvolvedores, mas também para os responsáveis externos de maneira a tornar mais compreensível para os clientes as implementações que fossem efetuadas. Com isso, facilitando a criação de tarefas pelos gestores.

2.1.3 Git e Gitlab

"O versionamento consiste em estratégias para gerenciar as diferentes versões de um código, de um sistema ou de um modelo. É uma forma de administrar as mudanças que são feitas e de garantir mais segurança na transição de uma versão para outra." (SACRAMENTO, 2021, p. n.p)

Figura 3: Logo Git e Gitlab

Fonte: (LONG, 2022) e (STICKPNG, 2022)

Dessa maneira, é possível compreender que o código gerado, era posteriormente versionado pelo terminal em *GIT* a fim de criar *branches* que separassem as implementações efetuadas e que dessa maneira pudessem ser armazenadas no *GITLAB* com a ideia de separar essas *branches* entre uma *branch master* com a versão mais estável do sistema e consequentemente *branches* em base da principal para implementar alterações e posteriormente após processo de avaliação serem incluídas a *master*.

3 MÉTODOS E FERRAMENTAS UTILIZADOS

O processo de criação do SARP se divide entre as etapas de análise e desenvolvimento das quais foram de grande importância durante todos os passos de sua estruturação. Dessa forma, a seguir será abordado as principais linguagens como *HTML, CSS, PHP e Javascript* que constituem o código fonte do projeto, a documentação com o diagrama de classes e o fluxograma completo de funcionamento do sistema servindo como guia para o entendimento do funcionamento do sistema e por fim os frameworks Adianti e *BootStrap* utilizados.

3.1 LINGUAGENS

Nesta seção, serão descritas as linguagens utilizadas durante o processo de desenvolvimento. A primeira delas é o *HTML*, que se trata de uma linguagem de marcação, voltada ao esqueleto da página web que em conjunto das linguagens de programação *PHP* e *Javascript* definem o comportamento e a comunicação com o servidor mediante as requisições e por fim o *CSS* voltado para a estilização das páginas, auxiliando na criação de padrões visuais específicos e responsividade voltada ao *mobile-first*.

3.1.1 HTML

O *Hyper Text Markup Language* ou Linguagem de Marcação de Hipertexto permite ao desenvolvedor estruturar páginas *web* facilitando inserção de imagens, textos e links de acesso a páginas externas. Logo, a mesclagem entre o script *PHP* ao efetuar consultas no servidor de acordo a busca do usuário, retorna a este, o resultado interpretado pela linguagem de marcação que exibe então a página *web*.

> A Linguagem de Marcação de Hipertexto (*HTML*) é uma linguagem de computador que compõe a maior parte das páginas da internet e dos aplicativos online. Um hipertexto é um texto usado para fazer referência a outros textos, enquanto uma linguagem de marcação é composta por uma série de marcações que dizem para os servidores da web qual é o estilo e a estrutura de um documento. (L, 2022, p. n.p)

Na primeira linha do Quadro 1 é definido o documento *DOCTYPE* a fim de passar instruções ao navegador de que se trata de um código *HTML*. Na linha 2 Abre-se a tag *HTML* seguido por *lang* para definição do idioma que o código será escrito. Entre as linhas 5 a 8 é definido o *header* da página com informações de título e o tipo de codificação em

metadados, as linhas 10 a 13 definem o corpo da página a ser escrita e por fim na 14 o fechamento da tag *HTML* como conclusão da estrutura.

Quadro 1: Exemplo Código HTML

```
1 <!DOCTYPE html>
2
3 <html lang="pt-br">
4
5   <head>
6     <title>Título da página</title>
7     <meta charset="utf-8">
8   </head>
9
10  <body>
11    Aqui vai o código HTML que fará seu site aparecer.
12  </body>
13
14 </html>
```

Fonte: (TABLELESS, 2022)

3.1.2 CSS

Por mais que o *HTML* possa estruturar o esqueleto de uma página web este não será capaz de compor a estilização das páginas, a fim de definir suas cores, tamanho de imagens e principalmente sua responsividade durante sua utilização em dispositivos de telas menores.

Papel esse constituído pelo *Cascating Style Sheet* ou Folha de estilo em cascata de maneira que o *CSS* serve como um separador de conteúdo da representação visual do site. Logo, é possível alterar a cor do texto e do fundo, suas fontes e espaçamento entre parágrafos, bem como, o ajuste de layouts e imagens. (G, 2022)

No Quadro 2 a seguir as classes de nome *html e body* sinalizam as alterações que serão aplicadas no corpo da página do SARP. Entre as linhas 3 e 8 os parâmetros como a altura, margem, a fonte que o texto terá ao ser exibido, seu tamanho, linha de altura e cor do texto. Entre as linhas 11 a 16 são definidas margem e bordas de *hr*. E por fim, em *hr. slim* são definidas as margens do topo e abaixo.

Quadro 2: Exemplo de código CSS

```css
html, body
{
    height: 95.5%;
    margin: 0px;
    font-family: "Helvetica Neue",Helvetica,Arial,sans-serif;
    font-size: 14px;
    line-height: 1.42857143;
    color: #333;
}

hr
{
    margin: 20px 0;
    border: 0;
    border-bottom: 1px solid #e2e2e2;
}

hr.slim
{
    margin-top: 10px;
    margin-bottom: 10px;
}
```

Fonte: Autoria Própria (2022)

3.1.3 PHP

O *PHP* ou *Hypertext Preprocessor* é uma linguagem de *script open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web podendo ser embutida dentro do *HTML*. O que distingue o *PHP* de algo como o *Javascript* no lado do cliente é que o código é executado no servidor, gerando o *HTML* que é então enviado para o navegador. (THE PHP GROUP, 2022, p. n.p)

O Quadro 3 demonstra a função excluir memorando. Na primeira linha é declarada a função deleteMemorando passando o parâmetro de forma que entre as linhas 2 a 9 é aberta uma transação no banco de dados, para verificar, se há atualmente uma *key (id)* correspondente ao memorando solicitante de exclusão. Logo, o parâmetro passa a ser a *key(id)* em que se relaciona com o *id* de equipamento a qual o memorando será removido e a transação é concluída.

As linhas 10 a 13 representam a exceção em caso de falha na requisição, exibindo uma mensagem de erro. Ao final do processo representado pelas linhas 14 e 15, solicita-se o recarregamento da página EquipamentoPanelView, de acordo ao *id* de equipamento.

Quadro 3: Exemplo de código PHP

```php
public function deleteMemorando($param){
        try {
            TTransaction::open('banco');
            if(isset($param['key'])){
                $memorando = new Memorando($param['key']);
                $equipamento_id = $memorando->equipamento_id;
                $memorando->delete();
            }
            TTransaction::close();
        } catch (Exception $e)
        {
            new TMessage('error', $e->getMessage());
        }
        TApplication::loadPage("EquipamentoPanelView", 'onView',
  ["key"=>$param["equipamento_id"]]);
 }
```

Fonte: Autoria Própria (2022)

3.1.4 JavaScript

A linguagem *Javascript* que também pode ser abreviada para *JS* é uma linguagem considerada leve, interpretada e baseada em objetos de funções primeira classe, normalmente reconhecida, por linguagem de *script* para páginas *Web*, mas usada também em vários ambientes sem *browser*, estes sendo, *node.js*, *Apache CouchDB* e *Adobe Acrobat*. O *JavaScript* é uma linguagem baseada em protótipos, multi-paradigma e dinâmica de suporte a estilos de orientação a objetos, imperativos e declarativos. (MOZILLA CORPORATION'S, 2022)

O Quadro 4 que representa o *script* a seguir, foi retirado de parte do código utilizado em uma das principais funcionalidades do sistema SARP referente ao formulário de Ordem de Serviço. Nesse cenário, é criado a lógica do comportamento dos campos de texto, mediante situação do equipamento. Essas situações se dividem entre 1 e 5, denominados: Em manutenção, Consertado, Aguardando Alienação, Alienado e restituído.

No entanto, será detalhado o processo de alienação que se refere a dispositivos que não possuem mais conserto por perda total ou por peças que não podem ser providas pela seção de manutenção. Logo, as linhas 1 e 2 determinam o tipo de código a qual será escrito, sendo este, *Javascript*.

Em seguida, as linhas 4 e 5 abrem a função e declaram o campo nomeado por Situação_id para mudanças de acordo a situação. Por fim, as linhas 10 a 19 constroem a condição da situação de valor 4 em que determina a situação da OS como alienada. Dessa forma, o campo de texto classificado como Solucionado é definido como não obrigatório, enquanto os

campos Pendentes *e* Sem Solução são atribuídos como obrigatórios em função de seu preenchimento. Com isso, o *script* é encerrado e a função é concluída.

Quadro 4: Exemple de código Javascript

```
$script = new TElement('script');
$script->type = 'text/javascript';
$javascript = "
$(document).ready(function(){
    $('input[name=situacao_id]').trigger('change');
  });

  (...)

      if($('input[name=situacao_id]:checked').val()==4){
          $('textarea[name=txt_solucionado]').prop('required', false);
          $('textarea[name=txt_pendente]').attr('required', true);
          $('textarea[name=txt_semsolucao]').attr('required', true);
      }
    });
 ";
 $script->add($javascript);
 parent::add($script);
}
```

Fonte: Autoria Própria (2022)

3.2 DOCUMENTAÇÃO

Essa seção, apresentará a etapa de análise definida nos estágios iniciais do processo de criação do sistema, descrevendo o fluxograma de funcionamento, de maneira que sua estrutura completa, estará presente no APÊNDICE A – FLUXOGRAMA SARP, demonstrando o processo desde o recebimento do dispositivo, até a devolução ou descarte deste.

Em seguida, será exibido o diagrama de classes, cujo papel é de permitir a compreensão dos relacionamentos entre os *models* de acordo ao comportamento de cada etapa e suas respectivas funcionalidades mediante ordem definida no fluxograma. Para a criação desses documentos, foram passadas orientações e a ferramenta *draw.io*, de acordo também, a reuniões com os servidores da seção de manutenção, a fim de, compreender a rotina de trabalho, regra de negócio e observações em questão de melhorias as funcionalidades da antiga aplicação e sugestão de novas.

Segundo (ECONOMIZAQUI, 2022) “O Diagrama de Domínio é uma representação visual de classes conceituais (ideias, coisas ou objetos) do mundo real que são significativas no domínio do problema. Para ilustrá-lo é utilizada a notação UML para diagrama de classes.”

3.2.1 Fluxograma

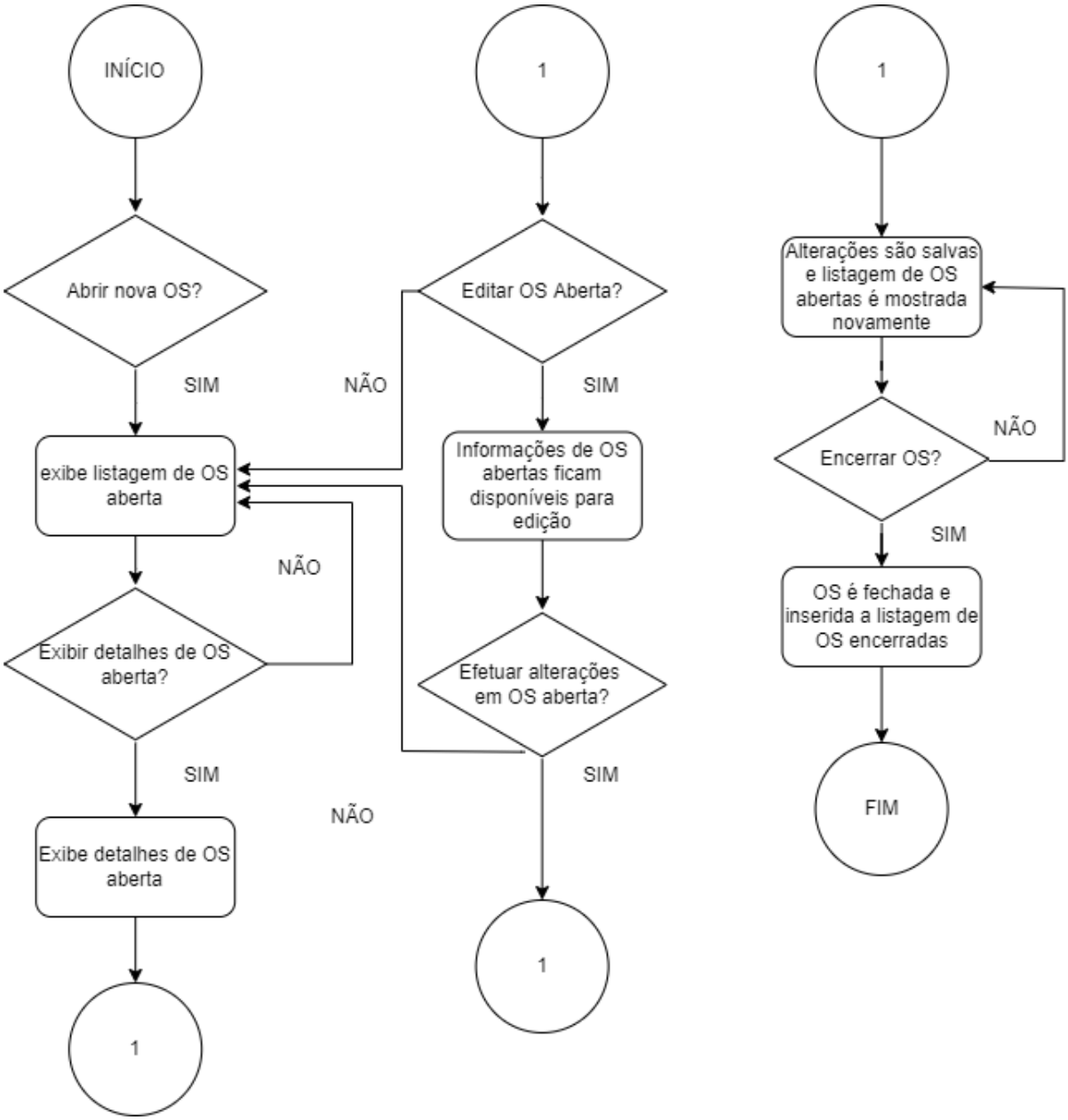
A Figura 4 a seguir, demonstra parte do fluxograma, na etapa de abrir uma nova Ordem de Serviço, correspondente aos serviços a serem prestados, após a análise registrada como Memorando, onde são especificadas as correções a serem executadas pela seção de manutenção classificando a ordem como aberta durante a execução dos procedimentos e de fechadas após sua conclusão.

Para tal, o processo se inicia ao perguntar ao usuário se ele deseja Abrir Nova OS e ao dizer que sim, este é destinado a Listagem de OS Aberta, onde é mais uma vez perguntado ao usuário se este deseja Exibir Detalhes de OS Aberta, ou senão, retornando a listagem de ordens de serviço.

Prosseguindo, pelo processo de Exibir Detalhes de OS Aberta, entre as opções prestadas, há a opção de editar a ordem de serviço selecionada. As informações em questão, passam a ser agora, editáveis e ao efetuar a edição, essas mudanças persistiram na base de dados.

Do contrário, caso nenhuma das opções anteriores tenham sido escolhidas o usuário retorna a listagem. Por fim, é perguntado se deverá Encerrar OS, de maneira que, ao solicitar, a ordem será fechada e removida da listagem de ordens de serviço abertas, passando para listagem de fechadas. Para compreender mais sobre o que são ordens de serviço abertas ou fechadas, mais detalhes, estarão presentes na seção 4.1.5 Ordem de Serviço Aberta e Fechada.

Figura 4: Fluxograma de Ordem de Serviço



Fonte: Autoria Própria (2021)

3.2.2 Diagrama de Classe de Domínio.

O diagrama de classe de domínio, representado por uma notação próxima a IDEF1X, refere-se ao processo de relacionamento entre as tabelas do banco de dados, a partir das entidades, relacionamentos e seus atributos. Logo, para a construção do diagrama representado pela Figura 5 referente ao comportamento dos *models* do sistema SARP são descritos os tipos de relacionamentos entre si, e a representação dos atributos de cada modelo demonstrado em sua estrutura, informando seu comportamento e utilizando o padrão de diagramação UML tradicional.

A tabela de Equipamento, relaciona-se com o Tipo, de maneira que estes se relacionam de 1 para 1, N (um e um ou muitos). Logo, o equipamento presente na tabela, pode ter apenas um tipo, enquanto a tabela com Tipo pode ter um ou mais equipamentos relacionados. Assim como, o relacionamento de Equipamento e Unidade. Em que o equipamento se refere apenas a uma unidade, mas a unidade pode se referir a um ou mais equipamentos.

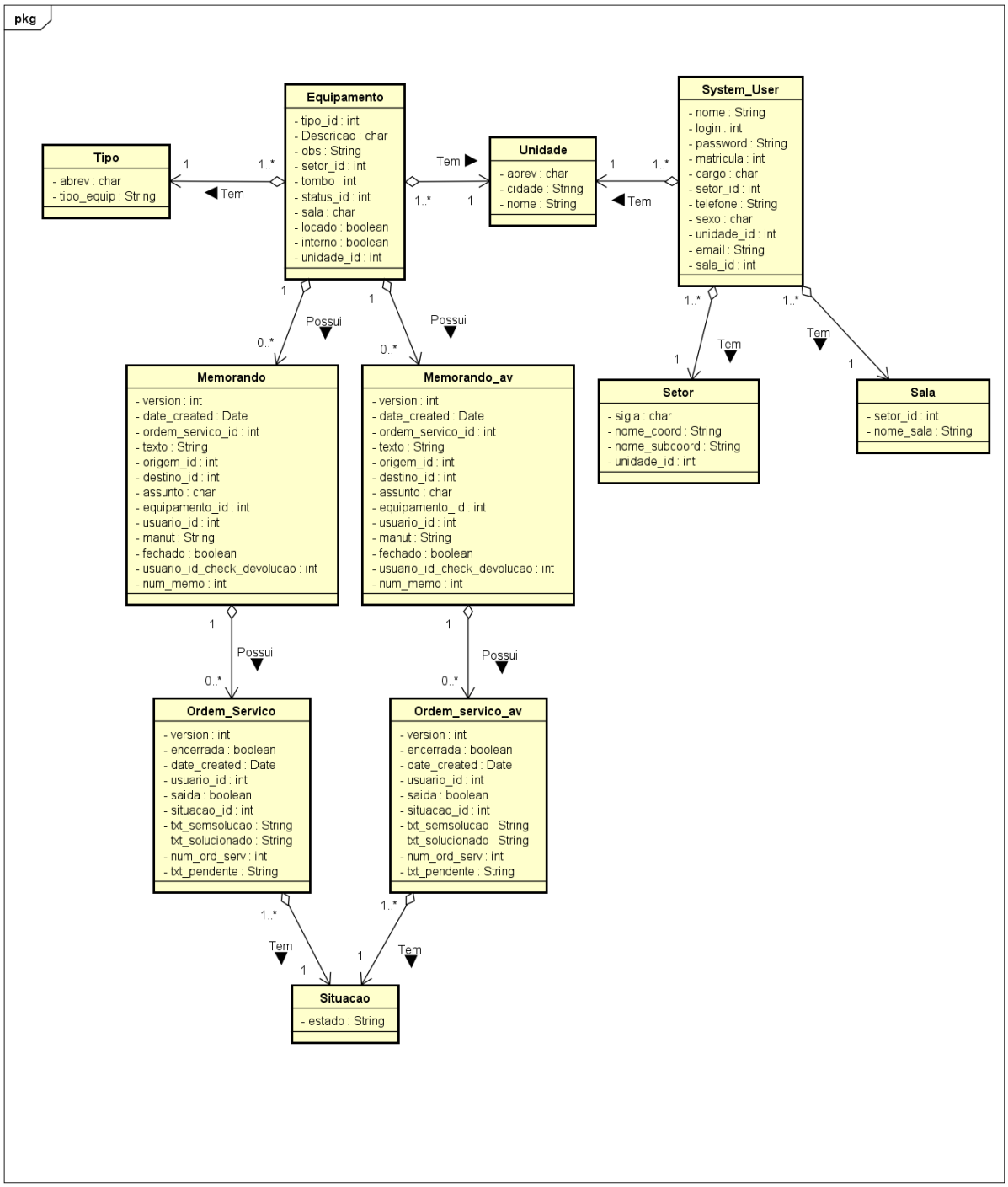
Passando para o relacionamento de Equipamento e Memorando 1 para 0, N (um para nenhum ou muitos), equipamento terá zero ou mais memorandos, enquanto a tabela referente a memorandos, poderá ter um equipamento. De mesma forma, será o comportamento de Equipamento e Memorando_av.

O relacionamento de Memorando e Ordem de Serviço, é representado por 1, 0, N (um para nenhum ou muitos) em que o memorando possuíra zero ou mais ordens de serviço relacionados. No entanto, a ordem de serviço poderá estar relacionada apenas a um memorando, de mesma forma, se comportando Memorando e Ordem_Servico_av.

Entre Ordem de Serviço e Situação o relacionamento, será de 1 para 1, N (um para um ou muitos) em que ordem de serviço poderá ter apenas uma situação relacionada e situação poderá ser relacionada a um ou mais ordens de serviço. O mesmo quesito está para ordem_servico_av e situação. No mais, temos a tabela de Unidade que irá se relacionar com *system_user de* 1 para 1, N (um para um ou muitos). Permitindo que uma unidade possa estar relacionada a um ou mais de um usuário, mas usuário, pode estar apenas relacionado a uma unidade.

System_user também se relaciona com as tabelas referentes a Sala e Setor em que seu relacionamento será de 1 para 1, N (um para um ou muitos). Sendo assim, o usuário poderá ter uma sala e um setor correspondente. Mas, as salas e setores poderão ter um ou mais usuários registrados.

Figura 5: Diagrama de Classe de Domínio



Fonte: Autoria Própria (2021)

3.3 FRAMEWORKS

Nesta seção, serão explorados os frameworks utilizados durante a estruturação do sistema. O principal deles, *Adianti Framework* conta com *templates* pré-prontos de escolha do desenvolvedor a fim de facilitar o processo de funções mais genéricas permitindo alterações e

implementação de novas funcionalidades especificas em conjunto da linguagem *PHP*. O *BootStrap* trata-se de um *framework* que também engloba *templates* e outras *features* de estilização auxiliando na responsividade e criação de conteúdo visual para as páginas reduzindo a utilização de *CSS* puro.

3.3.1 Adianti

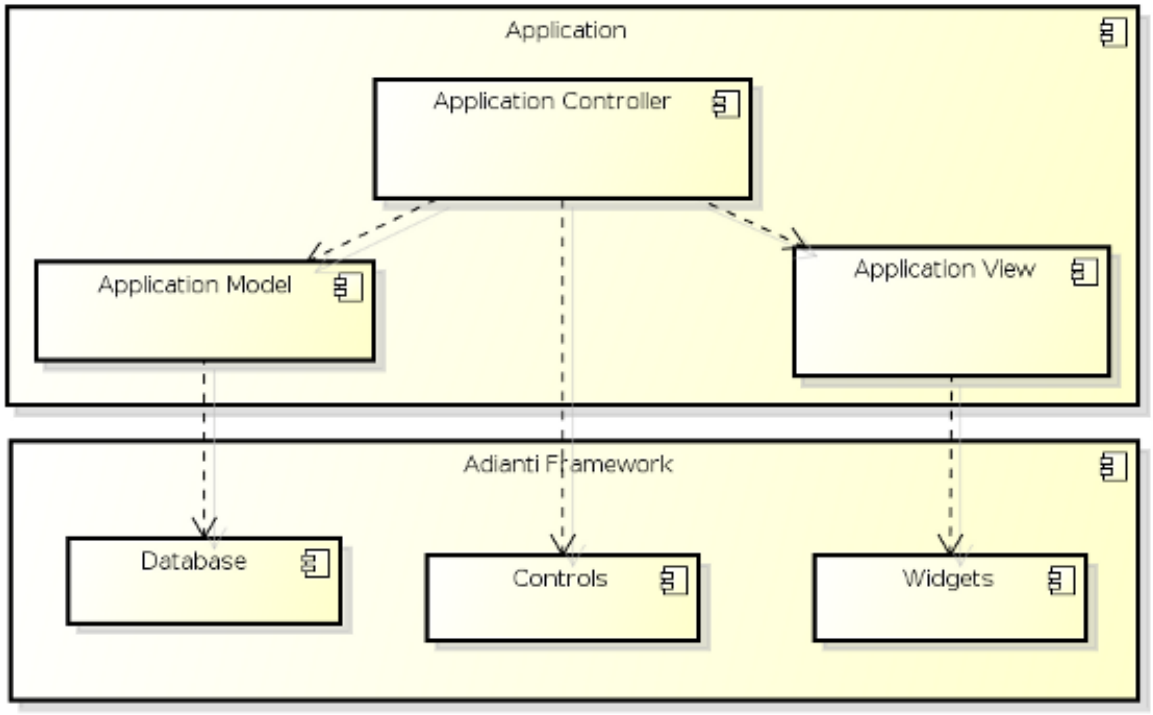
O *Adianti Framework* é uma arquitetura *open-source* para criação de sistemas em *PHP* focando no desenvolvimento de aplicações de negócios. Dessa forma, para acelerar a criação de sistemas, ele oferece uma série de componentes para a construção de formulários e *datagrids* e recursos para relatórios, gráficos, etiquetas, calendários e outros. Além de proporcionar uma gama de *templates* prontos e funcionais a fim de proporcionar controle de permissões, *logs* e interações entre usuários facilitando nos detalhes da implementação.

Uma aplicação criada com *Adianti Framework* será dividida em classes de diversos papéis, entre eles, o *model* que consiste em manipular os dados e desempenhar regras de negócios (*app/model*). A *view* define a interface visual na fronteira entre o sistema e o usuário, sendo representada por *HTML (app/resources)*. O controle responsável por receber ações vindas de uma classe *view e* tomar ações. E por fim, o *service* que presta serviços para a aplicação, como por exemplo, processar uma regra de negócio complexa, ou prestar um serviço para outra aplicação (*Ex: REST Service*). (DALL'OGLIO, 2019)

A Figura 6, procura demonstrar como está organizada a arquitetura do framework e representa a descrição feita a seguir:

> Na parte superior está o conjunto de classes criadas pelo desenvolvedor. Estas classes por sua vez dependem de componentes disponibilizados pelo framework. Assim, classes de modelo (Model) dependerão dos componentes Database, as classes de controle dependerão dos componentes de controle e as classes de visualização (View) dependerão dos widgets. (ADIANTI SOLUTIONS, 2022)

Figura 6: Arquitetura do Adianti Framework



Fonte: Adianti Solutions (2022)

3.3.2 Bootstrap

"O *BootStrap* é um *framework front-end* que fornece estruturas de *CSS* auxiliando a criação de sites e aplicações responsivas de forma rápida e simples." (LIMA, 2021, p. n.p) Logo, permitindo a utilização de classes específicas que permitem estilizar aspectos do site de forma mais limpa e eficiente a partir de modelos pré-existentes que possam ser customizados pelo desenvolvedor mediante necessidade do projeto. Tais classes e modelos, permitem a estilização de botões, *cards*, *navbars*, rodapés, estruturas responsivas e organização geral de páginas se utilizando de padrões *CSS* e *JS*.

O Quadro 5 demonstra algumas das classes *Bootstrap* para a versão voltada a construção de templates com Adianti Framework. A linha 1 representa a utilização da classe *BootstrapFormBuilder* passando por parâmetro *Form_sale_view* em que estilizará o formulário a ser criado de acordo ao modelo escolhido.

Na linha 2 é definido o título do formulário, na linha 3 é definidos as colunas da estrutura do *card* e por fim na linha 4 é definido o botão de ação de criação de um Novo Memorando e em que direciona para a página de formulário de memorando. Para detalhes visuais do botão é incluído também a classe *fas:plus fa-plus green* que estiliza a cor do botão para verde e inclui seu ícone.

Quadro 5: Exemplo de código BootStrap

```
$this->form = new BootstrapFormBuilder('form_Sale_View');
$this->form->setFormTitle('Equipamento');
$this->form->setColumnClasses(2, ['col-sm-3', 'col-sm-9']);
$this->form->addHeaderActionLink("Novo memorando", new
TAction(['MemorandoForm','create'], ['key' => $param['key']]), 'fas:plus fa-plus
green');
```

Fonte: Autoria Própria (2022)

3.4 BANCO DE DADOS

A base de dados do SARP constitui-se por informações sensíveis relacionadas aos servidores, equipamentos, setores ligados a SESAP e as unidades hospitalares que solicitam o reparo dos computadores. Logo, é importante que haja na estrutura do banco, tabelas que se relacionem entre si, para estruturar as hierarquias de acesso de acordo ao perfil do usuário. Além, das tabelas de memorandos, ordens de serviços determinando o registro de sua abertura ou fechamento para o controle de entrada e saída e dos profissionais envolvidos no processo de reparo, assim como, o registro de equipamentos alienados.

Essas informações são relevantes tanto para fatores de segurança e identificar autores de mudanças em suas atividades dentro do sistema, bem como, gerar relatórios e levantamentos gráficos dessas informações a fim de uma compreensão mais visual desses dados. Por tanto, para auxiliar a aplicação nessa tarefa foi utilizado o sistema de gerenciamento de banco de dados *phpMyAdmin.*

> *phpMyAdmin* é uma ferramenta de software livre escrita em *PHP* que se destina a lidar com a administração de um servidor de banco de dados *MySQL* ou *MariaDB*. Você pode usar *o phpMyAdmin* para executar a maioria das tarefas de administração, incluindo a criação de um banco de dados, a execução de consultas e a adição de contas de usuários. (THE PHPMYADMIN DEVEL TEAM REVISION, 2021, p. n.p)

4 DESENVOLVIMENTO

O SARP tem por foco a gestão sobre o controle de equipamentos da repartição interna e das unidades externas que entram e saem do setor de manutenção da (UGTSIC/SESAP). Tendo em vista, facilitar e aperfeiçoar os processos já existentes que tratam de receber, diagnosticar, armazenar e solucionar problemas de acordo com suas demandas correspondentes.

A necessidade da criação do SARP se deu em função do aperfeiçoamento dos processos ligados ao setor de manutenção da UGTSIC/SESAP que era originalmente administrado pelo sistema GESPATI de mesmo fim. No entanto, partes de suas funcionalidades se tornaram obsoletas, não intuitivas e mesmo funcionalidades que não existiam, mas que com o passar do tempo se fizeram necessárias mediante a evolução das demandas e necessidades do setor.

O sistema foi inicialmente recriado com suas funcionalidades básicas em uma nova arquitetura baseada em *PHP* e *Adianti Framework* a fim de que a interface fosse mais intuitiva e atrativa aos usuários. Outro fator importante, foi o de facilitar o processo de manutenção do sistema por desenvolvedores integrantes da equipe local, melhorando a comunicação entre o setor de manutenção e o de desenvolvimento ao que foi acordado para incluir ao sistema com plano de aplicar novas funcionalidades.

A aplicação está segmentada de acordo a Tabela 1 em quatro papéis, dentre eles, perfis de administrador, usuário, manutenção e suporte, permitindo funcionalidades que serão visíveis para esses usuários e seus respectivos grupos.

Tabela 1: Hierarquia de acesso SARP

Nível de Acesso	Funcionalidades de Acesso
Usuário	• Visualizar equipamentos do seu setor; • Criar memorando para manutenção.
Suporte	• Visualizar todos os equipamentos das unidades; • Cadastrar e editar equipamentos; • Cadastrar e editar usuários.
Manutenção	• Visualizar memorandos enviados ao setor;

	• Abrir Ordem de Serviço (OS); • Encerrar OS; • Efetuar saídas para setores; • Alienar e editar equipamentos; • Elaborar relatórios.
Administração	• Permissão para todas as funcionalidades anteriores; • Cadastrar salas, setores, unidades e gerenciar todo o sistema.

Fonte: Autoria Própria (2022)

4.1 IMPLEMENTAÇÃO

Para o plano de desenvolvimento de uma ferramenta web que seja capaz de suprir as necessidades do sistema anterior GESPATI, está sendo utilizado o Adianti Framework que auxilia na estruturação do *back-end* e que por sua vez renderiza o *Front-End*. Em função do banco de dados está sendo utilizado o *phpMyAdmin* para gerência das tabelas e seus respectivos relacionamentos.

A estrutura do código se divide similarmente ao padrão *Model-View-Controller* (*MVC*) em que as pastas se dividem separando e organizando de acordo a necessidade. Para os modelos estes são armazenados dentro da pasta *model*, *control* armazena os casos de usos do sistema e *config* faz a ponte entre a aplicação e o banco de dados.

4.1.1 Classe de Modelo

No Quadro 6, é demonstrado a classe de Ordem de Serviço para visualizar o comportamento em *models***.** Nela são declaradas suas constantes entre as linhas 3 a 5, que se comunicam com o banco, requisição de chaves estrangeiras nas linhas 7 a 8 e as funções seguintes abaixo declaram as variáveis da tabela e relacionam as externas a partir do *setters* e *getters*.

Quadro 6: Classe de Ordem de Serviço

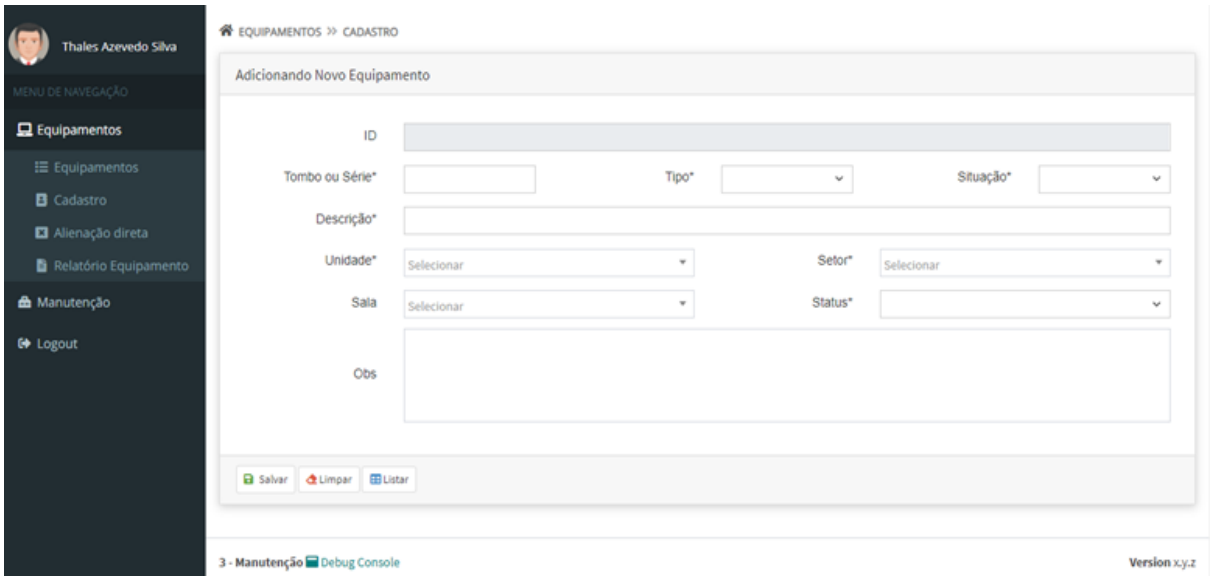
```php
class Ordem_Servico extends TRecord
{
    const TABLENAME = 'ordem_servico';
    const PRIMARYKEY= 'id';
    const IDPOLICY =  'max';

    private $usuario;
    private $situacao;

    public function __construct($id = NULL)
    {
        parent::__construct($id);
        parent::addAttribute('version');
        parent::addAttribute('encerrada');
        parent::addAttribute('date_created');
        parent::addAttribute('usuario_id');
        parent::addAttribute('saida');
        parent::addAttribute('situacao_id');
        parent::addAttribute('txt_semsolucao');
        parent::addAttribute('txt_solucionado');
        parent::addAttribute('num_ord_serv');
        parent::addAttribute('txt_pendente');

    }

    (...)

    public function set_situacao(Situacao $object){
        $this->situacao = $object;
        $this->situacao_id = $object->id;
    }

    public function get_situacao()
    {
        if (empty($this->situacao))
            $this->situacao = new Situacao($this->situacao_id);
        return $this->situacao;
    }
}
```

Fonte: Autoria Própria (2022)

4.1.2 Cadastro de Equipamento

Como demonstra a Figura 7, o fluxo se inicia pelo cadastramento de um novo equipamento, ao preencher inicialmente, o campo Tombo/Série que se refere ao código de identificação do equipamento, seguindo pelo Tipo que implica na especificação da máquina ou periférico que passará por Ajustes, além dos dados, de Situação e Descrição que fazem um resumo do problema e por fim informações de origem como Unidade, Setor, Sala, Status do equipamento e um espaço para possíveis observações.

Figura 7: Tela de cadastro com informações referentes ao equipamento.



Fonte: Autoria Própria (2021)

Na imagem representada pelo Quadro 7 é possível visualizar, os campos do formulário das linhas 1 a 7. Os classificados com *TEntry*, dos quais, especificam apenas os nomes dos dados que serão preenchidos manualmente pelo usuário, serão posteriormente gravados na tabela de equipamentos.

Os definidos por *TCombo* fazem uma requisição por Chave Estrangeira da tabela referente, trazendo uma listagem com os dados de seleção a escolha do usuário. As tabelas externas que acessam os dados para criar uma seleção dessas informações buscam as listagens de Unidades, Setor, Status e Tipo de Equipamento.

Entre as linhas 11 a 15 é definido o comportamento do campo Situação visto anteriormente na linha 4. O campo em questão trata-se de um *select* interno a tabela de equipamento e para que este possa trazer a listagem de situações são definidos os itens para

classificar como locado em caso de solicitação externa ou avulso quando tratar-se de setores internos a SESAP.

Ao clicar em Salvar, os dados preenchidos pelo usuário serão persistidos na base de dados e o usuário será redirecionado a listagem desses equipamentos e periféricos. As informações poderão ser alteradas ou excluídas a partir da mesma tela representada pela Figura *7* de acordo ao seu nível de acesso.

Quadro 7: Código de Formulário de Equipamento

```php
$id       = new TEntry('id');
$tombo     = new TEntry('tombo');
$tipo = new TDBCombo('tipo_id', 'banco', 'Tipo', 'id', 'tipo_equip');
$situacao  = new TCombo('locado');
$tombo     = new TEntry('tombo');
$descricao        = new TEntry('descricao');
$unidade_id = new TDBCombo('unidade_id', 'banco', 'Unidade', 'id',
'abrev');

(...)

$items = ['1' => 'Locado', '0' => 'Próprio'];
$situacao->addItems($items);

$items2 = ['SESAP' => 'SESAP', 'AVULSO' => 'AVULSO'];
$interno->addItems($items2);
```

Fonte: Autoria Própria (2022)

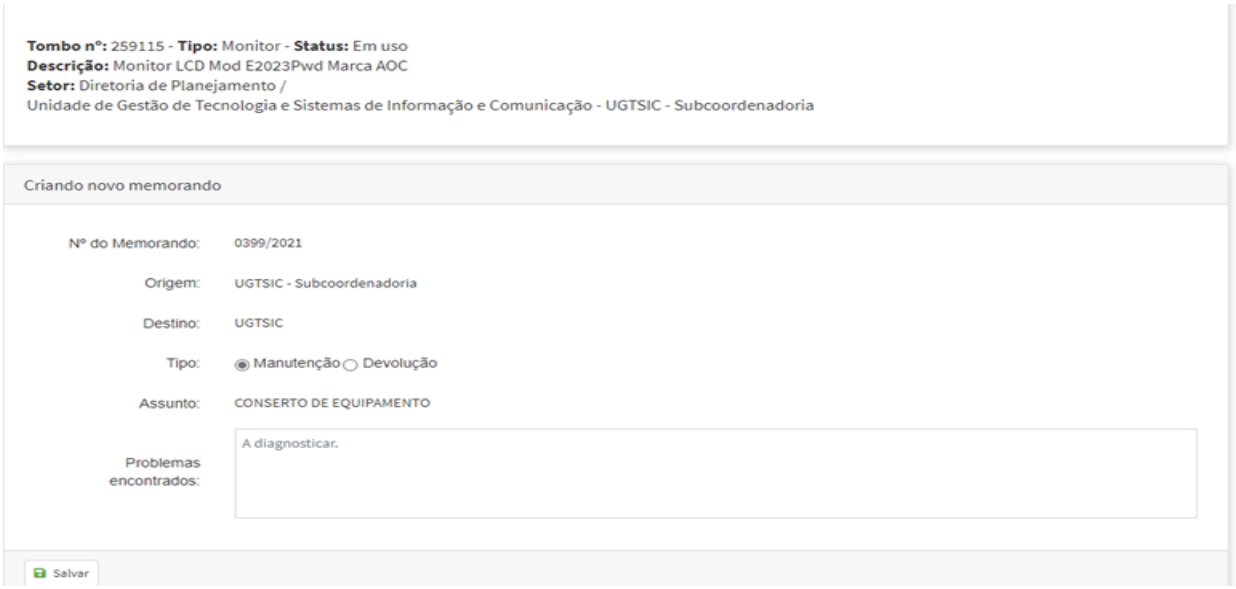
4.1.3 Cadastro de Memorando

A Figura 8 segue o processo ao trazer os principais dados de equipamento na listagem de dados superior, para a etapa de cadastro de um Novo Memorando, a fim de identificar e relacionar ao equipamento que a solicitação será aberta. Logo abaixo, é possível visualizar os campos automaticamente gerados como o N° de Memorando, Origem e Destino para identificar o local de reenvio e o assunto.

Os campos a serem preenchidos são o de Tipo de Serviço a ser efetuado denominado por Manutenção ou Devolução, assim como, o campo de Problemas Encontrados onde deverá ser especificado a razão da abertura do memorando. Ao cria-lo, o usuário será redirecionado a

Listagem de Memorandos em que este poderá Visualizar, Buscar por memorandos específicos e Abrir os detalhes onde será disponível gerar uma versão em PDF. Os detalhes envolvendo o processo de criação de relatórios estão disponíveis na seção 4.1.6 Relatórios.

Figura 8: Tela de novo memorando com as informações do equipamento.



Fonte: Autoria Própria (2021)

O Quadro 8 a seguir, demonstra o código em *Javascript* utilizado durante a etapa de criação de um novo memorando e como é administrado seu comportamento. A linha 1 específica como *script* o Novo Elemento. Nas linhas 2 e 3 são informados o tipo de *script* e por consequência o espaço onde este será escrito. A linha 4 abre a função para o *input([name=manut])*.

Entre 5 e 9 a condição é setada como 1 em caso de Manutenção, de maneira que o valor define o campo assunto do memorando como Conserto de Equipamento, em seguida, o campo de texto denominado Problemas Encontrados é preenchido com a mensagem A diagnosticar.

De mesma forma, o *else* representado pelas linhas 10 a 17, especificam a condição oposta em caso de o valor ser 2 representando o Memorando de Devolução. Logo, os passos são iguais aos anterior definindo o assunto como Devolução de Equipamento e o campo de texto apenas modificando o texto A diagnosticar por: Estamos Devolvendo o Equipamento Acima Especificado.

Quadro 8: Código script formulário de memorando.

```
$script = new TElement('script');
$script->type = 'text/javascript';
$javascript="
$('input[name=manut]').change(function(){
   if($(this).val()==1){
       $('#assunto').html('CONSERTO DE EQUIPAMENTO');
       $('textarea[name=texto]').attr('placeholder', 'A diagnosticar.');
       $('#".$label_problemas->getId()."').html('Problemas encontrados:');
       $('input[name=assunto]').val('CONSERTO DE EQUIPAMENTO');
    }else{
        $('#assunto').html('DEVOLUÇÃO DE EQUIPAMENTO');
        $('textarea[name=texto]').attr('placeholder', 'Estamos devolvendo o equipamento
  acima especificado.');
        $('#".$label_problemas->getId()."').html('Texto:');
        $('input[name=assunto]').val('DEVOLUÇÃO DE EQUIPAMENTO');
    }
  });
";
```

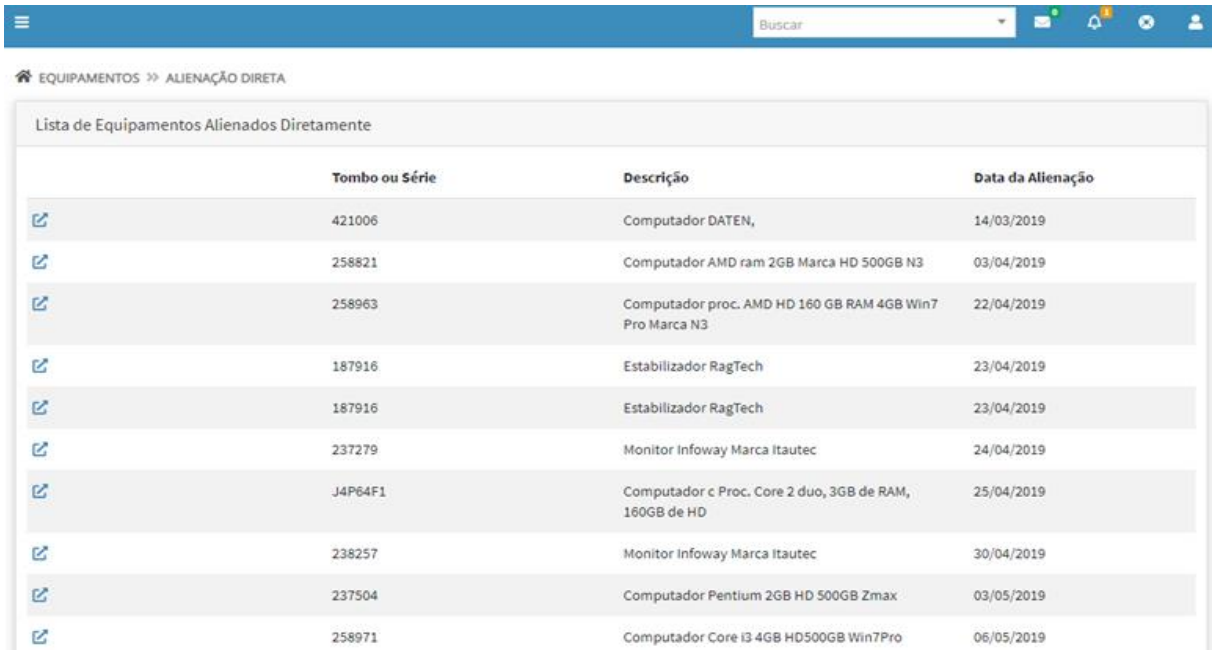
Fonte: Autoria Própria (2022)

4.1.4 Alienação direta

A Figura 9 está mostrando uma listagem de equipamentos considerados alienados. Este termo se refere aos dispositivos que a seção de manutenção não pôde consertar mediante falta de peças específicas ou por curto de placa mãe. Logo, durante o processo de análise o técnico responsável, ao entrar na lista de equipamentos, este determinará se este será criado um memorando e por consequência a ordem de serviço correspondente ou se haverá a alienação direta desse equipamento após verificação de reparo.

Os equipamentos com essa classificação são devolvidos aos setores responsáveis ou a depender do caso, descartados diretamente. A estrutura da lista, é bem similar as demais listas presentes no sistema, exceto, pelas opções de Alterar e Excluir o dispositivo alienado. Sendo assim, são especificados a opção de Detalhes referenciada pelo símbolo de “seta” que especifica as informações gerais do equipamento e a razão de seu descarte, o identificador Tombo/Série, sua Descrição Técnica e Data de Alienação.

Figura 9: Listagem de equipamentos alienados.



Fonte: Autoria Própria (2021)

Contextualizando o cenário do Quadro 9, este representa as funções de solicitações referentes a alienação direta dos equipamentos após diagnóstico e ação ser requisitada. A linha 1 abre a função DialogoAlienacao, a seguir a linha 2 define o parâmetro como *key* que correspondente ao equipamento. As linhas 4 e 5 criam a 1° ação de Alienação e a 2° ação de Visualização e por fim, as linhas, 7 a 9 relacionam o parâmetro *key* as ações e definem o campo *question* que certificando-se da decisão do usuário, dando-lhe duas opções, uma para prosseguir com o processo de alienação e a outra para visualização dos detalhes de equipamento.

Considerando que após o *modal* ser exibido e mesmo assim o usuário decidiu por alienar o dispositivo, as linhas 13 a 23 especificam a função AlienarEquipamento que abre uma transação na tabela de equipamento, solicitando permissão para aplicar alterações em seus status para alienado representado pelo número 4. Em seguida, são salvas as alterações feitas e a transação concluí com êxito. Em caso de erro, é estabelecido um *exception*, exibindo uma mensagem referente ao problema.

Quadro 9: Função diálogo e Alienar Equipamento.

```php
    public function dialagoAlienacao($param){

        $this->onView(["key"=>$param['key']]);
        $action1 = new TAction(array($this, 'alienarEquipamento'));
        $action2 = new TAction(array($this, 'onView'));

        $action1->setParameter('key', $param['key']);
        $action2->setParameter('key', $param['key']);
        new TQuestion('Deseja realmente alienar esse equipamento?', $action1, $action2);
    }

    //Função para mudar estado de equipamento para alienado
    public static function alienarEquipamento($param){
        try {
            TTransaction::open('permission');
            $equipamento = new Equipamento ($param['key']);
            $equipamento->status_id = 4;
            $equipamento->save();
            TTransaction::close();
        } catch (Exception $e) {
            new TMessage('error', $e->getMessage());
        }
    }
```

Fonte: Autoria Própria (2022)

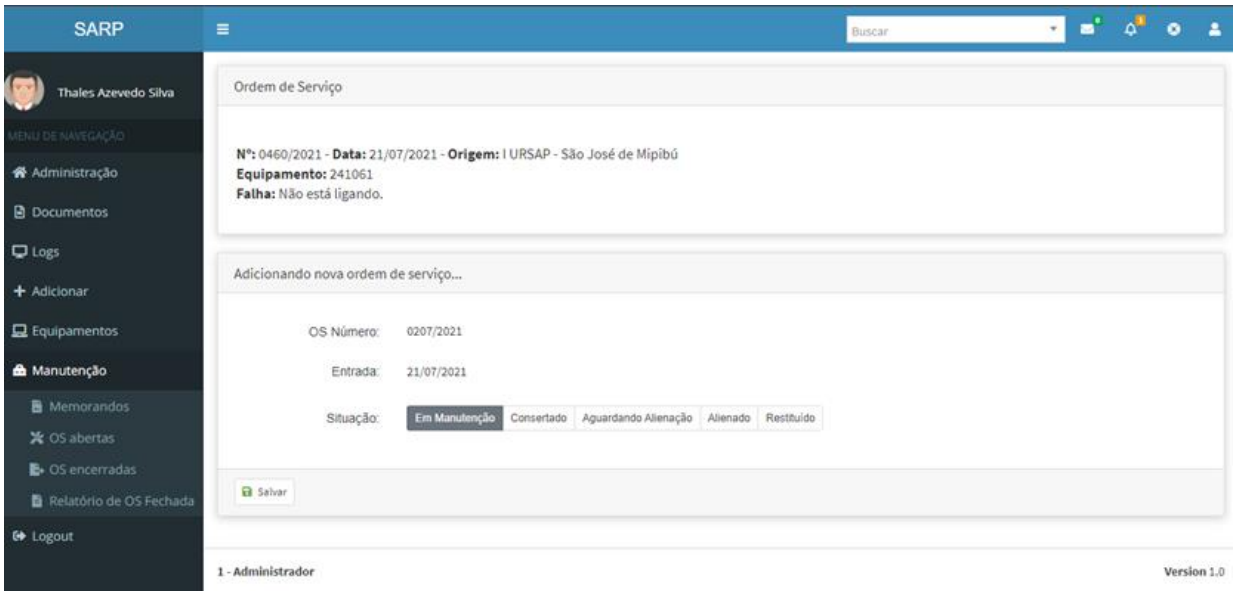
4.1.5 Ordem de Serviço Aberta e Fechada

Após criação do memorando especificando as informações de diagnóstico do equipamento, o passo a seguir, será de criar uma ordem de serviço que procederá nos ajustes referentes ao problema informado pelo memorando. Ao clicar em Abrir Nova OS, o usuário será direcionado ao formulário, representado pela Figura 10 onde irão conter informações relacionadas ao equipamento e ao memorando vinculado, sendo essas, o N° de Memorando, Data, Origem do dispositivo, Tombo e diagnóstico (Falha).

Em seguida, as informações preenchidas automaticamente, consistem no N° da Ordem de Serviço e data de Entrada do equipamento para execução de reparos. Cabe ao técnico responsável, definir o campo Situação manualmente, mediante etapa de ajuste que o dispositivo se encontra.

Após cadastro, o sistema direcionará o usuário a lista de Ordens Abertas, onde poderá ser feito a Busca por campos específicos, Alterações nas informações, assim como, a Visualização de detalhes e o seu Fechamento mediante encerramento dos reparos necessários. Sendo encerrada, a OS é removida da listagem de ordens abertas e direcionada a lista de fechadas em que permite a busca pelos campos de listagem e visualização dos detalhes da OS finalizada.

Figura 10: Tela de adição e edição ordem de serviço.



Fonte: Autoria Própria (2021)

A Figura 10 mostra as opções para o campo de situação entre elas temos Em Manutenção, Consertado, Aguardando Alienação, Alienado e Restituído. O Quadro 10 especifica a condição que implica no comportamento de ordens abertas e fechadas.

Considerando que a opção por padrão seja Em Manutenção, a ordem de serviço, não será fechada. Logo, na linha 1 a situação deverá iniciar como diferente de 1 para que a partir das linhas 3 e 4 seja definido que a situação sendo igual a 5 para casos de equipamento Restituído e campo de texto referente a solução esteja em branco, OS é considerada aberta, representada como 0.

Porém, de acordo as linhas 5 a 9 as demais situações de Conserto, Aguardando Alienação *e* Alienado irão definir ao salvar a ordem de serviço como fechada e representada por 1. Por fim, é definido a Data a qual esta foi encerrada e incluída a listagem de ordens encerradas.

Quadro 10: Código formulário de abertura de OS.

```
1  if ($param["situacao_id"] != 1) {
2
3      if ($param["situacao_id"] == 5 && empty($param["txt_semsolucao"])) {
4          $dados->encerrada = 0;
5      } else {
6          $dados->encerrada = 1;
7          $dados->saida = (new DateTime())->format('Y-m-d H:i:s');
8      }
9  }
```

Fonte: Autoria Própria (2022)

4.1.6 Relatórios

A etapa de relatórios, consiste em não apenas uma tela, mas este recurso, está presente em toda a estrutura da aplicação mediante as principais funcionalidades, demonstradas e descritas no decorrer deste trabalho. Durante todo o processo, desde a criação do memorando até a execução do reparo através da abertura de uma nova ordem de serviço, é de extrema importância que tais informações estejam impressas, a fim de que, haja cópias das solicitações com o setor cliente e com a seção de manutenção para que o processo envolvendo reparo e retorno a sua origem, de maneira formal e segura.

Estes equipamentos ou periféricos necessitantes de ajustes, deverão conter, não somente um levantamento virtual desses dados, mas em papel, considerando o contexto dos setores públicos que em parte ainda atuando através de documentações impressas. No cenário de um equipamento que tenha sido analisado e aberto um memorando, este ao exibir os detalhes do dispositivo, poderá gerar um documento em PDF. O presente representado pela Figura 11, terá em seu conteúdo, informações pertinentes como: N° Memorando, Data, Estado, Origem, Destino, Assunto e Contato.

Em detalhes de OS aberta, ao gerar o referido documento, o vigente trará o Tombo, Setor de origem, Memorando vinculado e a descrição do equipamento como informações principais. As ordens de serviços encerradas, também irão conter esses dados, porém com a adição dos dados envolvendo Saída, Defeitos e informações Do Conserto. Como sugestão dos gestores da seção de administração, foi implementado também, um relatório que permite trazer o levantamento de todas as OS fechadas referidas aos equipamentos e os reparos efetuados, mediante período de início e fim, trazendo o Total mediante pesquisa, Tipo do equipamento, Descrição e Situação.

Por fim, os relatórios ainda permitem a pesquisa por equipamentos que tenham sido alienados ou mesmo restituídos, ao informar o período desejado para que possa ser assim, gerada a listagem com as mesmas informações descritas anteriormente durante a relação de ordens de serviço fechadas.

Figura 11: Exemplo de relatório de memorando

Fonte: Autoria Própria (2021)

4.2 IMPLANTAÇÃO

Após o processo de desenvolvimento em que as funcionalidades foram aplicadas, é necessário partir para a etapa de implantação. Esta se constitui, pelo recebimento do banco e código homologado via Git pelo terminal, envolvendo a seção de Redes de Computadores, responsável por estruturar o sistema ao servidor da SESAP e consequentemente liberando as implementações, inicialmente em uma rota local para verificar se o comportamento está estável e posteriormente liberado para todo o setor.

Em seguida, considerando que a aplicação tenha sido implantada corretamente, a seção de manutenção, constituída como cliente alvo, deverá migrar suas atividades para a nova ferramenta. Dessa maneira, possíveis instabilidades, erros e problemas envolvendo comunicação com o banco, são relatadas a seção de desenvolvimento para o time vinculado a etapa de manutenções e possíveis ajustes.

Parte desses ajustes, cobriram a fase de relatórios, por conta de campos preenchidos manualmente que estavam sendo impressos fora do padrão determinado pelos documentos de memorando e ordens de serviços. Logo, tais correções foram aplicadas e destinadas, novamente a seção de redes para que pudessem ser atualizadas ao servidor as estruturas alteradas com

correções ou novas pequenas funcionalidades que auxiliassem as atuais. Por fim, as correções eram aplicadas, testadas localmente, testadas no servidor e liberadas a seção correspondente.

5 CONSIDERAÇÕES FINAIS

O SARP foi concluído com seu fim de atender a seções correspondentes ao setor da UGTSIC/SESAP, em especial a seção de manutenção e aos servidores responsáveis. Seu objetivo final teve por desígnio disponibilizar uma plataforma mais completa e concisa mediante as demandas necessárias durante o processo de recebimento de equipamentos, reparos e sua posterior devolução aos setores de origem.

Durante todo o processo de desenvolvimento, os conhecimentos adquiridos durante o decorrer da formação acadêmica, auxiliaram nas tomadas de decisões em um ambiente de trabalho que não possuía processos de software pré-definidos. Portanto, sprints, reuniões e padrões de documentação foram gradualmente aplicados, espelhando os modelos desenvolvidos em ambiente acadêmico.

O SARP por sua vez, também trouxe consigo, um grupo de funcionalidades renovadas pelo Adianti Framework, a fim de, permitir que tarefas base, fossem cumpridas com maior precisão e que ao decorrer de seus testes e utilização pelo cliente final, pudessem auxiliar no processo de lapidação ou introdução de funcionalidades que antes não puderam ser aplicadas no software GESPATI.

Concluindo-se assim, o SARP tem como papel o registro de equipamentos, análise, abertura de memorandos em função do registro desses diagnósticos, do contrário, a alienação direta desses dispositivos. Mas, que ao prosseguir com as etapas seguintes, abrirá uma ordem de serviço com intento de aplicar as correções sugeridas e concluir os serviços prestados ao encerrá-la. Por fim, possibilitando aos seus usuários o levantamento de dados, a partir do registro através de relatórios e cópias dos serviços em cada etapa, proporcionando segurança e controle entre a origem e o destino do equipamento.

REFERÊNCIAS

ADIANTI SOLUTIONS. **Arquitetura do Framework**. Adianti Solutions, 2022. Disponível em: <https://adianti.com.br/framework-architecture>. Acesso em: 8 Julho 2022.

DALL'OGLIO, Pablo. **Adianti Framework para PHP**. 1°. ed.

ECONOMIZAQUI. **Diagrama de Domínio**. EconomizAqui, 2022. Disponível em: <https://economizaqui.github.io/EconomizAqui/Requisitos/Modelagem/Tradicional/Diagrama-de-Dom%C3%ADnio/>. Acesso em: 1 Setembro 2022.

G, Ariane. **O que é CSS? Guia Básico para Iniciantes.** Hostinger Tutoriais, 14 Fevereiro 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css/>. Acesso em: 1 set. 2022.

JOEL. **MER e DER: Modelagem de Bancos de Dados**. Devmedia, 2014. Disponível em: <https://www.devmedia.com.br/mer-e-der-modelagem-de-bancos-de-dados/14332>. Acesso em: 16 Agosto 2022.

L, Andrei. **O Que é HTML? Guia Básico Para Iniciantes**. Hostinger Tutoriais, 25 Janeiro 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos>. Acesso em: 1 Setembro 2022.

LIMA, Guilherme. **Bootstrap - O que é, como e quando usar?** Alura, 21 Julho 2021. Disponível em: <https://www.alura.com.br/artigos/bootstrap>. Acesso em: 16 Agosto 2022.

LONG, Jason. **Git --everything-is-local**. Git, 2022. Disponível em: <https://git-scm.com/downloads/logos>. Acesso em: 15 ago. 2022.

MOZILLA CORPORATION'S. **JavaScript**. mdn web docs, 2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 29 Junho 2022.

SACRAMENTO, Gabriel. **VERSIONAMENTO DE CÓDIGO E DE SOFTWARE: ENTENDA CADA PROCESSO**. Tera, 1 Novembro 2021. Disponível em: <https://blog.somostera.com/data-science/versionamento#:~:text=O%20versionamento%20consiste%20em%20estrat%C3%A9gias,de%20uma%20vers%C3%A3o%20para%20outra.>. Acesso em: 16 Agosto 2022.

STICKPNG. **Logotipo GitLab**. stick PNG, 2022. Disponível em: <https://www.stickpng.com/pt-br/img/icones-logos-emojis/empresas-tecnicas/logotipo-completo-do-gitlab>. Acesso em: 15 ago. 2022.

TABLELESS. **Estrutura básica Iniciando o código básico de HTML**. tableless, 2022. Disponível em: <https://tableless.github.io/iniciantes/manual/html/estruturabasica.html>. Acesso em: 15 ago. 2022.

THE PHP GROUP. **O que é o PHP?** php, 2022. Disponível em: <https://www.php.net/manual/pt_BR/intro-whatis.php>. Acesso em: 29 Junho 2022.

THE PHPMYADMIN DEVEL TEAM REVISION. **Introdução**. phpMyAdmin, 2021. Disponível em: <https://docs.phpmyadmin.net/pt_BR/latest/intro.html>. Acesso em: 1 Julho 2022.

APÊNDICE A – FLUXOGRAMA SARP