

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

**Faculdade de Tecnologia Baixada Santista
Rubens Lara**

**Curso Superior de Tecnologia em
Sistemas para Internet**

**Gabriella Santos Barbosa
Jamilly Késia de C. Melo Silva**

SERVISYS
Sistema de gerenciamento de ordem de serviço

Santos, SP
2023

Gabriella Santos Barbosa
Jamilly Késia de C. Melo Silva

SERVISYS
Sistema de gerenciamento de ordem de serviço

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia Rubens Lara, como exigência para a obtenção do Título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Me. Claudio Nunes

Santos, SP
2023

RESUMO

O sistema ServiSys foi concebido com o propósito de facilitar a gestão de solicitações de manutenção e serviços, submetidas por Coordenadores de Cursos, à equipe do CPD (Centro de Processamento de Dados). Protótipos foram desenvolvidos para atender a ambos os públicos, solicitantes e administradores. A emissão manual de ordens de serviço frequentemente resultava na perda destes documentos, ocasionava demoras no processo de encaminhamento e aprovação das solicitações, ou ainda, enfrentava limitações no que tange ao armazenamento das respectivas folhas das ordens de serviço emitidas. Com a implementação do sistema, tais solicitações ocorreriam de forma online, possibilitando que o coordenador do CPD as recebesse de maneira imediata quando fossem submetidas pelo professor coordenador de curso. Além disso, o sistema proporcionaria um registro completo do histórico das solicitações, eliminando assim o risco de perda destas. Para obtermos acesso aos resultados, foram realizados testes do sistema já implementado na unidade de ensino, bem como monitoramento contínuo da usabilidade e do acesso por parte dos coordenadores de curso. Além disso, foram conduzidas avaliações para analisar como o sistema estava sendo recebido. Com a capacidade de atender tanto às necessidades dos Coordenadores quanto dos administradores de TI, o sistema visa a otimização dos processos relacionados à manutenção de equipamentos, resultando em maior agilidade e transparência nas operações de solicitações das ordens.

Palavras-chaves: Sistema. Manutenção. Coordenadores. Implementação.

ABSTRACT

The ServiSys system was designed with the purpose of facilitating the management of maintenance and service requests submitted by Course Coordinators to the CPD (Data Processing Center) team. Prototypes were developed to cater to both requesters and administrators. The manual issuance of service orders often resulted in the loss of these documents, causing delays in the processing and approval of requests, or facing limitations in terms of the storage of the respective service order sheets issued. With the implementation of the system, such requests would occur online, allowing the CPD coordinator to receive them immediately when submitted by the course coordinator. Furthermore, the system would provide a complete record of request history, thus eliminating the risk of their loss. To gain access to the results, tests were conducted on the system already implemented in the educational unit, as well as continuous monitoring of usability and access by course coordinators. In addition, evaluations were conducted to analyze how the system was being received. With the ability to meet the needs of both Coordinators and IT administrators, the system aims to optimize processes related to equipment maintenance, resulting in greater agility and transparency in the request operations for service orders.

Keywords: System. Maintenance. Coordinators. Implementation.

LISTA DE ABREVIATURAS E SIGLAS

CPD – CENTRO DE PROCESSAMENTO DE DADOS.....	9
T.I. – TECNOLOGIA DA INFORMAÇÃO.....	10
HTML – HIPERTEXT MARKUP LANGUAGE	10
SCSS – SASSY CASCADING STYLE SHEETS	10
ASP.NET – ACTIVE SERVER PAGES .NET.....	10
API – APPLICATION PROGRAMMING INTERFACE	10
UML – UNIFIED MODELING LANGUAGE.....	15
RF – REQUISITOS FUNCIONAIS	16
MER – MODELO ENTIDADE-RELACIONAMENTO.....	21
JWT – JSON WEB TOKEN	23

LISTA DE ILUSTRAÇÕES

Ilustração 1 – Painel de visão geral ServiceNow.....	11
Ilustração 2 – Painel de Atividades de Equipe	12
Ilustração 3 - Painel de Abertura de Chamado.....	12
Ilustração 4 – Painel de Preenchimento de Relatórios Realizados	13
Ilustração 5 – Painel de Gestão de Tarefas	14
Ilustração 6 – Diagrama de Classe	15
Ilustração 7 – Diagrama de caso de uso	18
Ilustração 8 – Modelo Entidade-Relacionamento	21
Ilustração 9 – Código do serviço OrdemService	22
Ilustração 10 – Código do Método GetOrdemServico	23
Ilustração 11 – Código método de criação de ordem de serviço.....	24
Ilustração 12 - Métrica geral do desempenho do teste.....	27
Ilustração 13 - Gráfico 1 – Teste de Usabilidade.....	29
Ilustração 14 - Gráfico 2 – Teste de Usabilidade.....	29
Ilustração 15 - Gráfico 3 – Teste de Usabilidade.....	30
Ilustração 16 – Gráfico 4 – Teste de Usabilidade.....	30
Ilustração 17 – Gráfico 5 – Teste de Usabilidade.....	31
Ilustração 18 – Código do método de atualização de ordem de serviço	34
Ilustração 19 - Código do método de seleção de todas as ordens de serviço	374
Ilustração 20 - Código do método de filtragem de ordem de serviço por usuário..	3835
Ilustração 21 - Código método de filtragem de ordem de serviço por ID.....	385
Ilustração 22 – Página de Login	346
Ilustração 23 – Página de solicitação de ordem de serviço.....	346
Ilustração 24 – Página de aprovações pendentes.....	347
Ilustração 25 – Página de aprovações do usuário.....	347
Ilustração 26 – Página de ordens de serviço do CPD	348
Ilustração 27 – Página de alteração de status da ordem de serviço	348
Ilustração 28 – Página de cadastro de usuário	349

LISTA DE TABELAS

TABELA 01 – TABELA DE REQUISITOS FUNCIONAIS	16
TABELA 02 – TABELA DE REQUISITOS NÃO FUNCIONAIS	17
TABELA 03 – TABELA DO CENÁRIO DE SOLICITANTE	28
TABELA 04 – TABELA DO CENÁRIO DE COORDENADOR DE T.I.	28
TABELA 05 – TABELA DO CENÁRIO DE TÉCNICO DE T.I.	28
TABELA 06 – RESPOSTAS DOS PARTICIPANTES DO CARD SORTING	28

SUMÁRIO

1 INTRODUÇÃO	9
1.1 OBJETIVO.....	10
1.1.1 OBJETIVO GERAL.....	10
1.1.2 OBJETIVOS ESPECÍFICOS.....	10
1.2 ESTADO DA ARTE.....	11
1.2.1 <i>SERVICENOW</i>	11
1.2.3 PIPEFY.....	13
2 DESENVOLVIMENTO.....	15
2.1 ANÁLISE DO SISTEMA.....	15
2.1.1 ANÁLISE DE REQUISITOS.....	16
2.1.2 DIAGRAMA DE CASO DE USO	17
2.1.3 FLUXO DE EVENTOS.....	18
2.2 BANCO DE DADOS	21
2.3 CAMADA DE NEGÓCIO.....	21
2.4 CAMADA DE APRESENTAÇÃO	24
3 RESULTADO.....	26
3.1 TESTES	26
3.1.1 TESTE DE CARD SORTING	26
3.1.2 TESTE DE USABILIDADE.....	27
3.2 CONCLUSÃO.....	31
REFERÊNCIAS.....	33
APÊNDICE A – CAMADA DE NEGÓCIO	34
APÊNDICE B – CAMADA DE APRESENTAÇÃO	36
APÊNDICE C – RESPOSTAS DO TESTE DE CARD SORTING	40

1 INTRODUÇÃO

Em um cenário de ensino técnico, onde uma diversidade de cursos é ministrada, especialmente aqueles que demandam o uso de laboratórios de informática, apresenta-se a necessidade recorrente da manutenção de máquinas e equipamentos, proporcionando assim a equipe manutenções denominada como Centro de Processamento de Dados (CPD) a responsabilidade de executar serviços essenciais.

As recorrentes dificuldades encontradas no registro de solicitações de manutenção se dão mediante a utilização de formulários em formato físico, suscetíveis a extravios e perdas, além do que a acumulação de documentos em papel propicia a possibilidade de confusões na gestão dessas ordens de serviço, o que representam inconvenientes no tratamento das demandas.

A automatização, ao longo das últimas décadas, tem desempenhado um papel fundamental na otimização de processos em diversas áreas, incluindo a tecnologia. De acordo com Gates (1995), “A primeira regra de qualquer tecnologia utilizada nos negócios é que a automação aplicada a uma operação eficiente aumentará a eficiência...”. Diante desse contexto, torna-se iminente reconhecer a necessidade de automatizar o processo de gerenciamento das ordens de serviço.

Ainda, Segundo MIKELSTEN, Daniel, “O benefício da automação inclui economia de mão-de-obra, economia de energia elétrica, economia de material e melhorias na qualidade, exatidão e precisão.” O que reforça que, tal iniciativa se aplicada não somente mitigaria os desafios de abordagem manual, como também traria uma série de benefícios, tais como a redução do risco de perdas de informações, histórico de solicitações e o próprio gerenciamento de solicitações. Em suma, a automação representaria um avanço substancial na eficiência operacional do CPD, contribuindo para uma gestão mais eficaz e transparente das operações de manutenção de equipamentos.

1.1 OBJETIVO

O objetivo desta seção é fornecer uma visão detalhada dos objetivos gerais e específicos do projeto, delineando as métricas utilizadas, referenciais de estado da arte relevantes e ferramentas para a construção do sistema.

1.1.1 OBJETIVO GERAL

O objetivo deste trabalho é automatizar integralmente o processo de recebimento, processamento e armazenamento de documentos referentes a solicitações de serviços de manutenção, amplamente conhecidas como ordens de serviço.

1.1.2 OBJETIVOS ESPECÍFICOS

Automatizar o processo de recebimento e processamento de ordens de serviço, levando em consideração os diferentes perfis de usuários envolvidos: os funcionários com cargos de supervisão, incluindo coordenadores de curso e outros cargos de responsabilidade equivalente ou superior, responsáveis por iniciar a solicitação de manutenção; o coordenador de Tecnologia da Informação (T.I.), encarregado de aprovar ou rejeitar cada solicitação; e o CPD, responsável pela execução da manutenção. A aplicação será um sistema web desenvolvido utilizando *HiperText Markup Language (HTML)*, *Sassy Cascading Style Sheets (SCSS)*, *Active Server Pages .NET Core (ASP.NET Core)* e o *framework Angular*, que consumirá uma *Application Programming Interface (API)* para obter informações sobre os computadores e modelos de cada máquina, garantindo que cada ordem de serviço contenha o modelo específico e padronizado.

Implementar um sistema de armazenamento automatizado para as ordens de serviço, utilizando uma área dedicada no banco de dados do sistema. Este banco será relacional, o que permitirá que as ordens sejam organizadas e arquivadas de forma sistemática, facilitando o acesso e a consulta quando necessário.

1.2 ESTADO DA ARTE

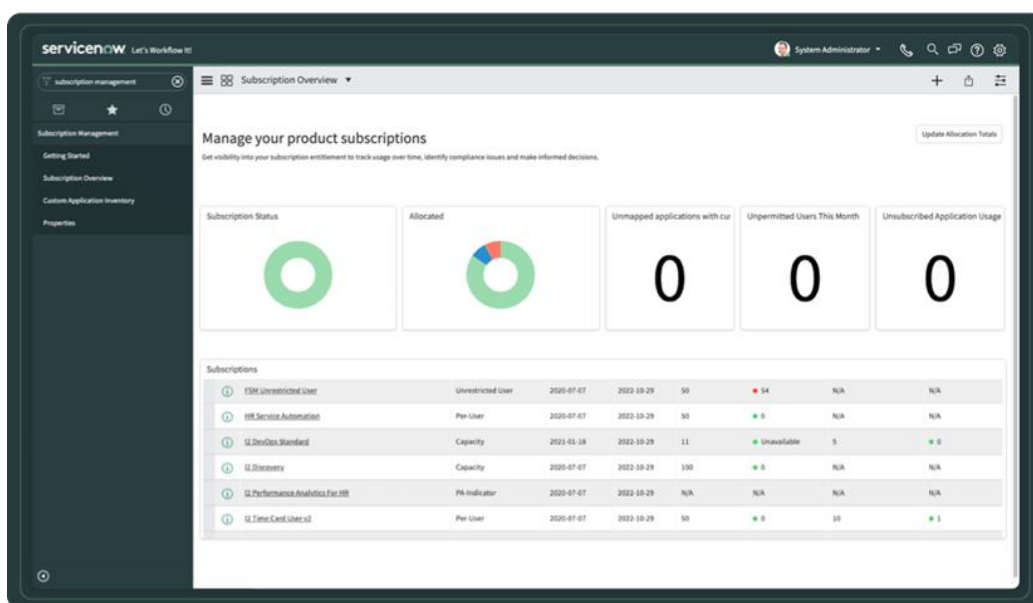
Nesta seção, é apresentada uma revisão do estado da arte em relação às aplicações de ordens de serviço e sistemas relacionados já existentes que serviram como base para o desenvolvimento do sistema ServiSys.

1.2.1 ServiceNow

O *ServiceNow* é uma plataforma de gerenciamento de serviços empresariais utilizada para automatizar e otimizar processos relacionados à gestão de serviços e tarefas dentro de organizações.

Suas principais características incluem a capacidade de adaptar seu *layout* e funcionalidades de acordo com as necessidades únicas de cada organização, a geração de relatórios personalizados e a capacidade de rastrear o status de todas as solicitações de serviço, permitindo uma visão completa do progresso de cada ordem de serviço (como na Ilustração 1), entre outras funcionalidades. Isso torna a plataforma flexível e personalizável, permitindo que as empresas a utilizem para uma variedade de finalidades, incluindo a gestão de ordens de serviço.

Ilustração 1 – Painel de visão geral ServiceNow



Fonte: ServiceNow, 2023

1.2.2 Produttivo

O Produttivo é um software de manutenção voltado para empresas que, por meio da automação, oferece a capacidade de gerar relatórios, orçamentos e checklists. Além disso, a plataforma permite o gerenciamento da agenda dos técnicos e a monitoração de indicadores essenciais.

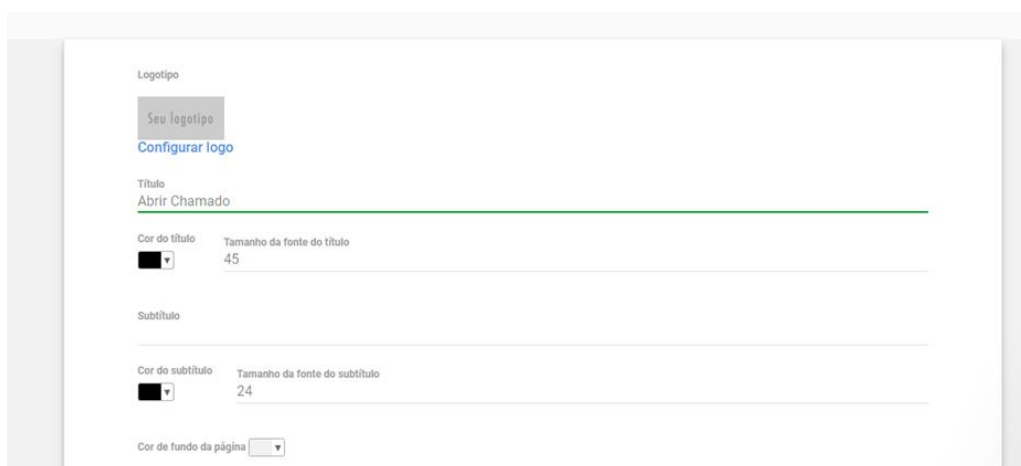
Este sistema abrange diversas funcionalidades, incluindo a abertura de chamados, conforme a Ilustração 2, preenchimento de checklists e formulários, assinatura eletrônica e disponibiliza indicadores e gráficos que acompanham o desempenho do trabalho.

Ilustração 2 – Painel de Atividades de Equipe



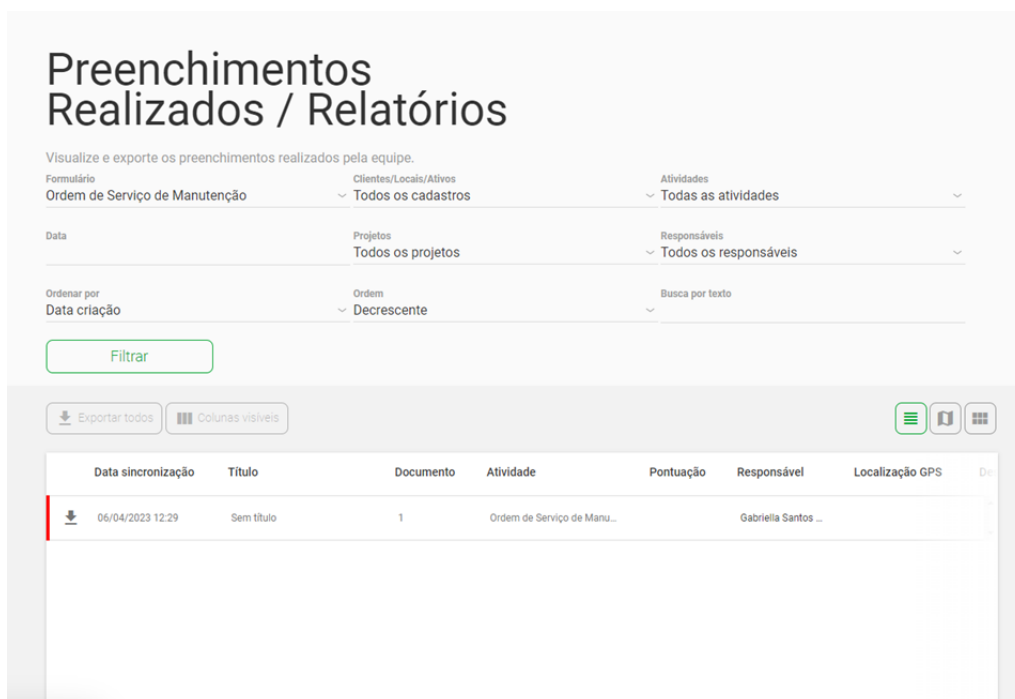
Fonte: Produttivo, 2023

Ilustração 3 - Painel de Abertura de Chamado



Fonte: Produttivo, 2023

Ilustração 4 – Painel de Preenchimento de Relatórios Realizados



Fonte: Produtivo, 2023

1.2.3 Pipefy

O Pipefy é uma plataforma multifuncional que oferecem funcionalidades para diversas áreas empresariais, abrangendo recursos para Recursos Humanos, requisições de pagamento em compras, processos de contas a pagar e receber na área financeira, bem como gerenciamento de serviços de T.I. Além disso, seu *layout* é personalizável, permitindo a criação de soluções para outras áreas que necessitem da plataforma.

Na área de Tecnologia da Informação (T.I.), o Pipefy tem como funcionalidade automatizar a triagem de chamados, integração com canais de comunicação e recebimento de e-mails, como Gmail, Outlook e Teams. Além disso, ele oferece uma organização de tarefas por meio de métodos Kanban, catálogos de serviços, relatórios, dashboards e outras ferramentas que auxiliam na visibilidade dos processos (ilustração 5).

Ilustração 5 – Painel de Gestão de Tarefas

The image shows a Kanban board in Pipefy for the project "[TI] Gestão de Incidentes". The board is organized into five columns representing different stages of incident management:

- Revisão (1)**: Contains one card with ID "INC0001", category "Aplicação", and description "Exemplo".
- Ajustes (0)**: Empty column.
- Confirmação (0)**: Empty column.
- Em Andamento (1)**: Contains one card with ID "INC0002", category "Aplicação", and description "Exemplo". It has a red "Alta" (High) priority label.
- Solucionados (1)**: Contains one card with ID "INC0003", category "Aplicação", description "Exemplo", and a "1min" completion time. It has a green checkmark icon.

A tooltip in the bottom left corner says: "Comece a adicionar informações no seu pipe. Aproveite para personalizar a forma como essas informações chegam aqui. Dispersar". Below it is a blue button labeled "+ Create new card".

Fonte: Pipefy, 2023

2 DESENVOLVIMENTO

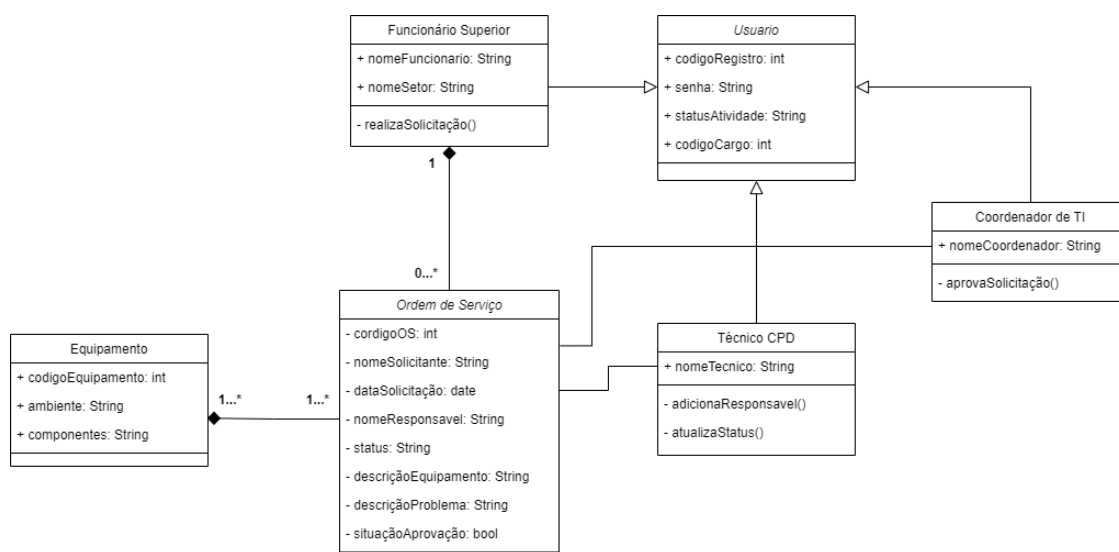
Abaixo, serão exploradas as etapas do desenvolvimento do sistema proposto, divididas em análise do sistema, estrutura do banco de dados, ferramentas empregadas para o desenvolvimento do *front-end* e principais rotinas implementadas no *back-end*.

2.1 ANÁLISE DO SISTEMA

Entre os vários mecanismos de análise disponíveis para o desenvolvimento de sistemas, optou-se pela *Unified Modeling Language (UML)*. UML é uma linguagem visual para modelar softwares de propósito geral, que se consolidou como a linguagem-padrão de modelagem na indústria de Engenharia de Software, aplicável a diversos domínios de aplicação. Na ilustração 6, está uma representação gráfica do sistema *ServiSys* em um digrama de classe utilizando os conceitos do UML.

A UML – Unified Modeling Language ou Linguagem de Modelagem Unificada – é uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos. É uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicação. Essa linguagem tornou-se, nos últimos anos, a linguagem-padrão de modelagem adotada internacionalmente pela indústria de engenharia de software. (GUEDES, 2009, p.19)

Ilustração 6 – Diagrama de Classe



Fonte: Autores, 2024

2.1.1 ANÁLISE DE REQUISITOS

Nesta fase ocorre a captura das intenções e as necessidades dos usuários a ser desenvolvido através dos requisitos funcionais (RF), como mostrado na tabela 1.

Um conjunto de requisitos pode ser definido como uma condição ou capacidade necessária que o software deve possuir para que o usuário possa resolver um problema ou atingir um objetivo ou para atender as necessidades ou restrições da organização ou dos outros componentes do sistema. (MACHADO, 2018)

Tabela 1 – Tabela de requisitos funcionais

ID	REQUISITOS FUNCIONAIS	DESCRIÇÃO
RF01	Solicitação de serviços	Permite que funcionários com autoridade de supervisão ou superior solicitem serviços de manutenção para computadores e equipamentos danificados ou com problemas.
RF02	Aprovação da ordem de serviço	Permite que o coordenador de T.I. filtre qual solicitação de ordem de serviço deve ser executada ou não.
RF03	Acompanhamento de manutenção	Permite que os funcionários com autoridade de supervisão ou superior acompanhem o progresso das solicitações de serviços, de acordo com o status da manutenção: se já foi concluída, está em andamento ou ainda será realizada, por exemplo.
RF04	Priorização de seleção	Permitir que os administradores de TI priorizem a solicitação de acordo com a gravidade do problema e a disponibilidade de recursos.

Fonte: Autores, 2023

Para a construção do sistema, é fundamental a definição de requisitos não funcionais. Segundo Pressman e Maxim (2016, p.141), um requisito não funcional pode ser descrito como um atributo de qualidade, de desempenho, de segurança ou como restrição geral em um sistema. Abaixo, na tabela 2, estão requisitos não funcionais identificados no sistema *ServiSys*.

Tabela 2 – Tabela de requisitos não funcionais

ID	REQUISITOS NÃO FUNCIONAIS	DESCRIÇÃO
RNF01	Usabilidade	O sistema deve ser fácil de usar para os professores, coordenadores de TI e assistentes administrativos.
RNF02	Desempenho	O sistema deve ser rápido e responsivo, permitindo que as solicitações sejam atendidas rapidamente.
RNF03	Segurança	O sistema deve garantir a segurança das informações dos usuários e dados confidenciais.
RNF04	Disponibilidade	O sistema deve estar sempre disponível para uso, com tempo mínimo de inatividade planejado para manutenção.
RNF05	Escalabilidade	O sistema deve ser escalável para lidar com um grande volume de solicitações e manutenções.

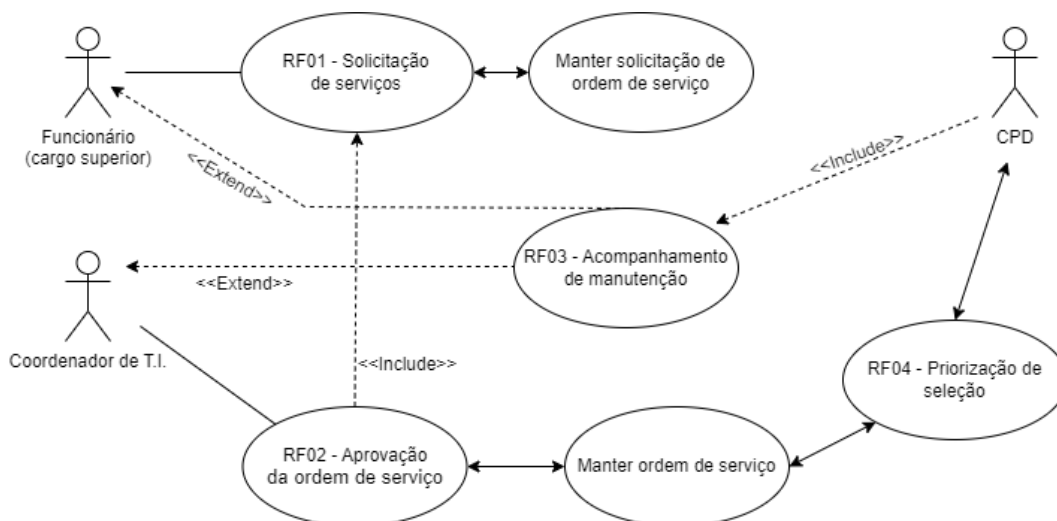
Fonte: Autores, 2023

2.1.2 DIAGRAMA DE CASO DE USO

Com os requisitos funcionais e não funcionais estabelecidos, o próximo passo é a criação do diagrama de caso de uso. Ele tem como finalidade representar de forma gráfica como os usuários interagem com um sistema e quais funcionalidades o sistema oferece. No diagrama na ilustração 7, os atores, como usuários e sistemas

externos, são representados por ícones, enquanto as diferentes ações que o sistema pode realizar são mostradas como elipses. As linhas conectando os atores aos casos de uso ilustram como eles se relacionam, ajudando a compreender quem faz o quê no sistema.

Ilustração 7 – Diagrama de caso de uso



Fonte: Autores, 2023

2.1.3 FLUXO DE EVENTOS

O fluxo de eventos tem por intenção caracterizar o início e o término do caso de uso, estabelecer a interação dos atores no processo em descrição, gerando assim um fluxo principal e um fluxo alternativo do comportamento de cada caso de uso do sistema. A descrição dos fluxos de eventos desse sistema pode ser consultada no Apêndice B.

Caso de Uso 1: Solicitação de Serviços

Ator: Funcionário (cargo superior)

Fluxo Básico:

1. Funcionário acessa o sistema e realiza login
2. O sistema oferece interface para solicitação de ordens de serviço
3. O funcionário preenche formulário com informações necessárias e envia a solicitação
4. O sistema confere que todas as informações obrigatórias estão preenchidas

5. O sistema confirma o envio da solicitação emitindo mensagem de sucesso.

Fluxo alternativo:

5. O sistema detecta a falta de informação de nome da máquina e ambiente
 - 5.1. O usuário preenche as duas informações e realiza envio
 - 5.2. Retorna ao fluxo básico.

Caso de Uso 2: Aprovação da ordem de serviço

Ator: Coordenador de T.I.

Fluxo Básico:

1. Coordenador de T.I acessa o sistema e realiza login
2. O sistema oferece interface para aprovação de ordens de serviço
3. O coordenador seleciona uma ordem de serviço para analisar
4. O coordenador verifica quantidade de ordens de serviço em andamento
5. Coordenador verifica que há material disponível e que é possível realizar pedido a tempo
6. O coordenador aprova solicitação de ordem de serviço
7. O sistema emite mensagem de sucesso

Fluxo alternativo:

5. O coordenador verifica que não é possível realizar pedido de material
 - 5.1. O coordenador não aprova o pedido
 - 5.2. Retorna ao passo 3.

Caso de Uso 3: Acompanhamento de manutenção

Ator: Coordenador de TI

Fluxo Básico:

1. Coordenador de TI acessa o sistema e realiza login
2. Sistema oferece tela de acompanhamento de manutenção
3. Coordenador seleciona ordem em andamento para visualizar
4. Sistema emite informações atuais e status da ordem de serviço
5. O coordenador de TI verifica que a manutenção está de acordo com as especificações da solicitação e atende aos padrões de qualidade exigidos.
6. Coordenador encerra manutenção
7. Sistema emite mensagem de sucesso e encaminha manutenção para aba de OS concluídas.

Fluxo Alternativo:

5. Coordenador verifica que manutenção está em andamento, esperando chegada de peças essenciais

- 5.1. Coordenador mantém ordem de serviço em andamento
- 5.2. Retorna ao passo 3.

Caso de Uso 3: Acompanhamento de manutenção

Ator: Funcionário (cargo superior)

Fluxo Básico:

1. Funcionário acessa o sistema e realiza login
2. Sistema oferece tela de solicitações realizadas por esse funcionário
3. Funcionário seleciona uma solicitação para visualizar
4. Sistema emite informações prévias e status da ordem de serviço
5. Funcionário vê status da ordem.

Fluxo Alternativo:

4. Sistema emite mensagem de ordem cancelada.
 - 5.1. Funcionário realiza outro pedido
 - 5.2. Retorna ao fluxo básico.

Caso de Uso 3: Acompanhamento de manutenção

Ator: CPD

Fluxo Básico:

1. Técnico do CPD acessa o sistema e realiza login
2. Sistema oferece tela solicitações em andamento
3. Técnico do CPD seleciona uma solicitação para visualizar
4. Sistema emite informações prévias e status da ordem de serviço
5. Funcionário vê status da ordem.
6. Funcionário atualiza andamento e especificações de sua ordem
7. Sistema emite mensagem de ordem atualizada com sucesso.

Fluxo Alternativo:

3. Técnico do CPD seleciona ordem de outro técnico para visualizar
 - 3.1. Sistema emite informações prévias e status da ordem de serviço
 - 3.2. Técnico do CPD não consegue atualizar ordem
 - 3.3. Retorna ao passo 2.

Caso de Uso 4: Priorização de Seleção

Ator: CPD

Fluxo Básico:

1. CPD acessa o sistema e realiza login

2. O sistema oferece interface para visualização de ordens de serviço
3. Técnico do CPD visualiza ordens de serviço
4. Técnico do CPD escolhe ordem de serviço em aberto informada com urgência pelo coordenador
5. O sistema emite mensagem de sucesso para ordem com urgência escolhida.

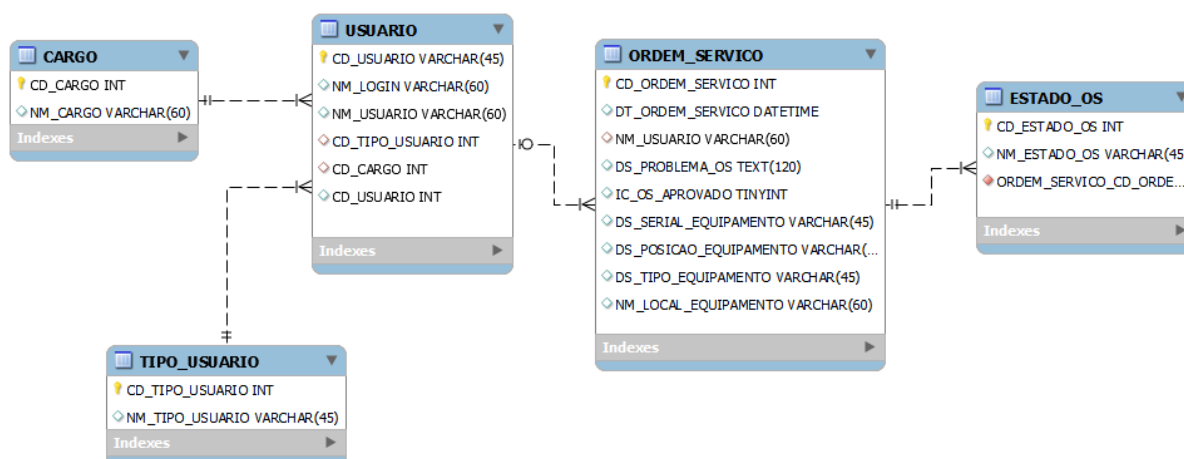
Fluxo alternativo:

4. Técnico do CPD não encontra ordem com urgência disponível para escolha
 - 5.1. Técnico do CPD escolhe ordem de serviço com urgência inferior
 - 5.2. Retorna ao passo 3.

2.2 BANCO DE DADOS

O objetivo é exemplificar a estrutura dos dados por meio de tabelas e apresentar o Modelo Entidade-Relacionamento (MER) correspondente. Pretendemos demonstrar a modelagem dos dados utilizando tabelas e esquematizar o MER para ilustrar as relações entre as entidades, conforme a ilustração 8.

Ilustração 8 – Modelo Entidade-Relacionamento



Fonte: Autores, 2023

2.3 CAMADA DE NEGÓCIO

Para o *front-end*, foi utilizado o *framework* Angular na versão 16.2, juntamente com *Typescript*, HTML e *Bootstrap* na versão 11. Visto que, a ação de foco do sistema gira em torno da criação e ciclo de vida de diversas ordens de serviço como um todo,

processo esse que rotaciona entre os 3 tipos de usuários diferentes cadastrados no sistema, sendo eles: O funcionário (professor coordenador), O coordenador de T.I. e o CPD, tendo o Administrador apenas para cadastro de um usuário no sistema.

Nesse sentido, atribuiu-se que uma funcionalidade relevante para o projeto é a integração com a API, através de um service que utiliza os métodos HTTP para requisição dos dados da ordem do funcionário.

Ilustração 9 – Código do serviço OrdemService

```
export class OrdemService {
  baseUrl = environment.apiUrl + 'api/OrdemService';

  constructor(private http: HttpClient) {}

  public GetOrdemService(): Observable<OrdemService[]>{
    return this.http
      .get<OrdemService[]>(this.baseUrl).pipe(take(1));
  }

  public GetOrdemServiceByUsuario(idUsuario: number): Observable<OrdemService[]> {
    return this.http
      .get<OrdemService[]>(`${this.baseUrl}/usuario/${idUsuario}`).pipe(take(1));
  }

  public GetOrdemServiceById(id: number): Observable<OrdemService>{
    return this.http
      .get<OrdemService>(`${this.baseUrl}/${id}`).pipe(take(1));
  }

  public PostOrdemService(ordens: OrdemService): Observable<OrdemService>{
    return this.http
      .post<OrdemService>(this.baseUrl, ordens).pipe(take(1));
  }

  public PatchOrdemService(id: number, changes: Partial<OrdemService>): Observable<any> {
    return this.http
      .patch(`${this.baseUrl}/Aprovado/${id}`, changes).pipe(take(1));
  }
}
```

Fonte: Autores, 2024

Assim como é demonstrado na ilustração 9, foi utilizado um serviço atribuído como *OrdemService* que é empregado para realizar a requisição de ordens de serviço através de três métodos HTTP, sendo eles o método GET para obter informações da ordem, POST para publicar uma nova ordem e PATCH para atualizar as ordens de serviço.

Ilustração 10 – Código do Método GetOrdemServico

```
public GetOrdemServico(): void {  
    this.ordemService.GetOrdemServico().subscribe({  
        next: (ordens: OrdemServico[]) => {  
            this.ordens = ordens;  
            this.ordensFiltradas = this.ordens;  
        },  
        error: (error: any) => console.log(error)  
    });  
}
```

Fonte: Autores, 2024

Através da ilustração 10, observa-se que o método *GetOrdemServico()* faz uma requisição para buscar todas as ordens através do serviço *ordemService*, se inscreve no Observable (*.subscribe*) que é retornado para receber os dados, e armazena a lista de ordens de serviço recebida em duas propriedades do componente (*ordens* e *ordensFiltradas*). Além do que se ocorrer um erro durante a requisição, o mesmo é registrado no console.

A arquitetura voltada ao *back-end* foi desenvolvida utilizando uma *RESTful API*, com *endpoints* específicos para que a camada de *front-end* possa se comunicar com o sistema e acessar as informações armazenadas no banco de dados. O ASP.NET Core na versão 8.0 foi a principal tecnologia utilizada no *back-end*, juntamente com um serviço de banco de dados *SQL Server*.

Para a construção da API, foi adotado uma abordagem com várias camadas de organização e serviços, incluindo *Controllers* e camadas de autenticação baseadas em *Json Web Token* (JWT). Uma das camadas essenciais é a de *Service*, responsável pelo serviço de transferência de ordens de serviço, que é a funcionalidade principal do sistema. Nessa camada, estão armazenados os métodos que gerenciam as informações, utilizando os principais *endpoints* de GET, PUT, POST e DELETE para manipular uma ordem de serviço. Cada *endpoint* é configurado e restrito de acordo

com a responsabilidade do usuário. Portanto, dependendo do cargo do usuário, ele terá acesso apenas aos *endpoints* específicos relacionados às suas funções.

Ilustração 11 – Código método de criação de ordem de serviço

```
2 referências
public async Task<OrdemServicoDto> AddOrdensServico(int userId, OrdemServicoDto model)
{
    try
    {
        var ordemServico = _mapper.Map<OrdemServico>(model);
        ordemServico.UserId = userId;

        _gerealPersist.Add<OrdemServico>(ordemServico);
        if (await _gerealPersist.SaveChangesAsync())
        {
            var ordemRetorno = await _ordemServicoPersist.GetOrdemServicoByIdAsync(userId, ordemServico.Id, false);
            return _mapper.Map<OrdemServicoDto>(ordemRetorno);
        }

        return null;
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}
```

Fonte: Autores, 2024

Na ilustração 11, a ordem de serviço é criada, recebendo como parâmetro de entrada para o método as informações do ID do usuário e dados essenciais para a criação da ordem. Ao criar a ordem, ela é automaticamente associada ao usuário com o ID correspondente fornecido como parâmetro. Esse método, conforme demonstrado aqui e utilizado nos próximos métodos abaixo, faz uso do *AutoMapper*. O *AutoMapper* é um *microframework* empregado para mapear quais informações são transferidas entre objetos, simplificando assim a relação de objetos na injeção de dependência. Os outros métodos citados estão apresentados no Apêndice B.

2.4 CAMADA DE APRESENTAÇÃO

Neste tópico, serão demonstradas as páginas principais do sistema para cada nível de usuário, incluindo o funcionário, o Coordenador de T.I., CPD e o Administrador. Para cada tipo de usuário, foram determinadas as restrições e funcionalidades específicas com base em suas posições e necessidades.

Todas as seções só poderão ser acessadas pelos usuários adequados após o login, passando pela página de *login* padrão (ilustração 22). Na página principal destinada ao funcionário, o objetivo principal ao utilizar o sistema é fazer a solicitação

de ordem de serviço. Ao acessar o atalho no menu lateral (presente em todas as páginas para fácil acesso), o funcionário poderá descrever a ocorrência para a qual a ordem de serviço será criada e, em seguida, solicitar a análise do Coordenador de T.I., como mostra na ilustração 23.

Para o Coordenador de T.I., a funcionalidade principal no sistema é revisar e aprovar ou rejeitar ordens de serviço. Portanto, no sistema, o Coordenador precisará apenas verificar as ordens pendentes e tomar a ação apropriada para aprová-las, como pode ser visto na ilustração 24. Além disso, o usuário terá acesso a um histórico das ordens que foram aprovadas ou rejeitadas, permitindo uma visão clara do status das solicitações, conforme a ilustração 25.

Para o CPD, as principais funcionalidades disponíveis podem incluir a visualização e possibilidade de alterar o status de ordens de serviço aprovadas conforme o fluxo de trabalho e o progresso da demanda. O CPD é responsável por atualizar e definir o status de cada ordem de serviço, permitindo que o funcionário que solicitou possa acompanhar conscientemente o progresso da demanda, sabendo se está concluída, em andamento ou aguardando a compra de algum material. As ilustrações 26 e 27 mostram as páginas que exibem as ordens de serviço atribuídas ao responsável no CPD e a opção de mudar o *status* conforme necessário.

Para o Administrador, sua única funcionalidade é cadastrar o usuário no sistema (ilustração 28). Por ser um sistema restrito, deixar a possibilidade de cadastro externo resultaria em pessoas que não estão no ciclo de negócio cadastradas.

Todas as ilustrações desse capítulo estão disponíveis no Apêndice B.

3 RESULTADO

Este capítulo tem como objetivo ilustrar e apresentar o resultado da aplicação na prática, por meio da realização de testes e da monitoria de uma pessoa especializada na área. A realização dos testes e dos feedbacks tem como principal objetivo identificar possíveis pontos de melhoria na aplicação e verificar se o que foi idealizado nos capítulos anteriores foi executado na prática.

3.1 TESTES

Para avaliar a usabilidade do projeto, foi realizado dois tipos de testes principais: *Card Sorting* e teste de usabilidade. O *Card Sorting* ajudou a compreender como os usuários organizam e categorizam informações. Já o teste de usabilidade envolveu a criação de cenários e tarefas específicas para que os participantes interagissem com o sistema. Esses testes foram fundamentais para garantir que o design final atendesse às necessidades e expectativas dos usuários.

3.1.1 TESTE DE CARD SORTING

Segundo Donna Spencer (2010, p. 67), “*Card sorting* é uma maneira prática de obter informações sobre como as pessoas pensam sobre conteúdo e categorias”. A metodologia de *Card Sorting* foi escolhida para entender o que o usuário espera ao idealizar um sistema de ordens de serviço, utilizando a dinâmica de arrastar cartões para suas respectivas categorias. Para a execução, utilizamos a *plataforma Optimal Workshop*.

Os participantes selecionados tinham uma compreensão em manutenção de máquinas e eram familiarizados com o uso de eletrônicos de computadores o que permitiu que tivessem a perspectiva do solicitante da ordem de serviço e compreendessem o fluxo de negócio. No total, oito pessoas participaram e, além de realizar o *Card Sorting*, responderam a dois questionários: um antes e outro depois da atividade. O questionário pós-atividade foi elaborado para entender o que os usuários sentiram falta ou não compreenderam no processo; as respostas estão na tabela, presente no Apêndice C.

Ilustração 12 – Métrica geral do desempenho do teste

	Menu fixo - Pági...	Menu fixo - Coor...	Menu Fixo - C...	Página de lo...	Página de solicit...	Página de orden...	Página de aprov...	Página de aprov...	Página de orden...	unsorted
Botão do menu "nova ordem"	75%				25%					
Botão do menu "suas ordens"	50%	13%	13%			25%				
Botão do menu "suas aprovações"		88%					13%			
Botão do menu "aprovações pen...		63%	13%					25%		
Botão do menu "ordens de serviço"	13%		50%		13%	13%		13%		
Email				100%						
Senha				100%						
Logotipo				88%						
Formulário de solicitação de ordem	13%				88%					
Prévia das ordens de serviço env...			25%			63%			13%	
Lista de todas as ordens enviadas			13%			38%		13%	38%	
Lista de todas as ordens aprovad...		38%					50%		13%	
Prévia das ordens aprovadas ou ...			13%			25%	38%	13%	13%	
Tabela "Ordens para serem atuali...		25%					38%	13%	25%	
Tabela "Ordens de serviço esper...		25%						63%	13%	

Fonte: Autores, 2024

Os usuários em uma média, mostraram um entendimento de layout compreensível com o que estava esperado (ilustração 12), alguns cards ficaram com diferenças grandes por conta do entendimento de que aquele card poderia fazer parte da tela escolhida por conta do nome semelhante das páginas, o que pode ter feito o participante se confundir ou não compreender onde exatamente faria mais sentido deixar o card. A maioria das páginas contem rotulagem parecida, por isso a compreensão da confusão

3.1.2 TESTE DE USABILIDADE

Para obter resultados concretos sobre o uso da aplicação e o fluxo de navegação, o teste de usabilidade foi o escolhido. Segundo BARNUM (2020), quando o teste de usabilidade faz parte do design e desenvolvimento, o conhecimento sobre a experiência dos usuários apoia todos os aspectos desses processos. Isso aumenta as chances de criar um produto final que atenda às expectativas e necessidades do público.

Foi apresentado três cenários com tarefas para os participantes executarem em nosso sistema, analisando se enfrentariam dificuldades ao interagir e navegar. Os participantes seguiram os seguintes cenários:

Tabela 3 – Tarefas do cenário de Solicitante

Cenário 1: Solicitante
Tarefa 1: Logar
Tarefa 2: Criar uma ordem
Tarefa 3: Visualizar uma ordem criada
Tarefa 4: Sair do sistema

Fonte: Autores, 2024

Tabela 4 – Tarefas do cenário de Coordenador de T.I.

Cenário 2: Coordenador de T.I.
Tarefa 1: Logar
Tarefa 2: Visualizar uma ordem
Tarefa 3: Aprovar uma ordem
Tarefa 4: Rejeitar uma ordem
Tarefa 5: Sair do sistema

Fonte: Autores, 2024

Tabela 5 – Tarefas do cenário de Técnico de T.I.

Cenário 3: Técnico de T.I.
Tarefa 1: Logar
Tarefa 2: Visualizar uma ordem
Tarefa 3: Mudar o status de serviço dessa ordem
Tarefa 4: Salvar a ordem atualizada
Tarefa 5: Sair do sistema

Fonte: Autores, 2024

Ao todo foram, 6 participantes concluíram as tarefas pelos três cenários em média de 3 minutos e meio. Após a finalização do teste com as telas, os participantes foram encaminhados para um formulário para que eles pudessem avaliar como foi a experiência e o que poderia ser melhorado no sistema.

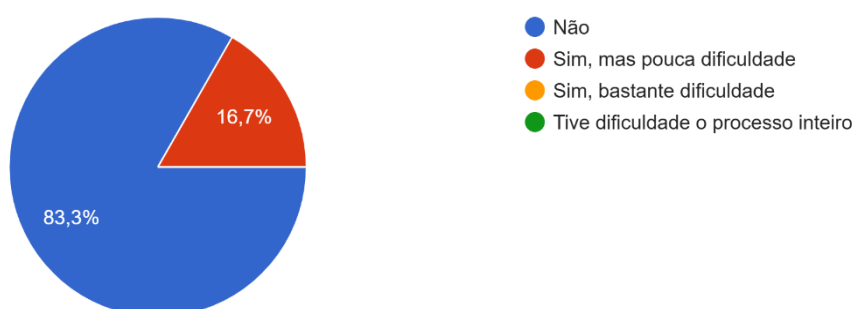
Na primeira pergunta, apresentada na ilustração x, foi direcionada ao cenário de Solicitante. Os usuários apresentaram pouca dificuldade ao utilizar o sistema,

tendo 83,3% das respostas relataram não ter dificuldade, e apenas 16,7% apresentaram um pouco de dificuldade.

Ilustração 13 – Gráfico 1 – Teste de Usabilidade

Cenário 1: Solicitante Alguma etapa do processo de criação de ordem pareceu confusa ou desnecessária? Houve dificuldade para criar uma ordem?

6 respostas



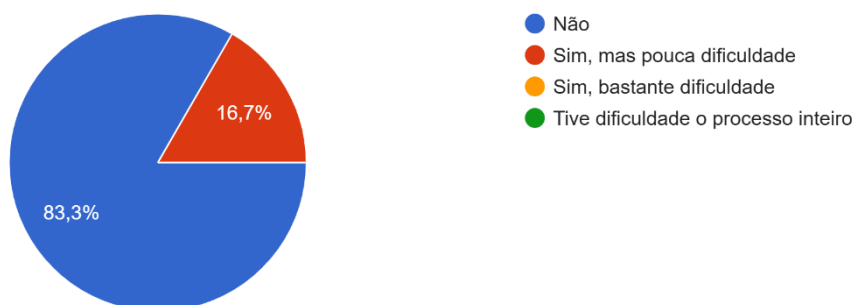
Fonte: Autores, 2024

A segunda pergunta, apresentada na ilustração x, foi direcionada ao cenário do Coordenador de T.I. Os participantes também relataram não ter dificuldades em realizar as tarefas propostas, com 83,3% das respostas indicando ausência de dificuldade e apenas 16,7% indicando alguma dificuldade em executar as tarefas.

Ilustração 14 – Gráfico 2 – Teste de Usabilidade

Cenário 2: Coordenador de T.I. Você encontrou dificuldade em aprovar ou recusar uma ordem?

6 respostas



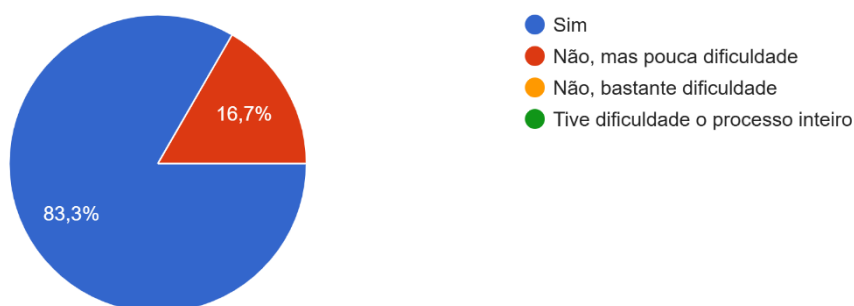
Fonte: Autores, 2024

A terceira pergunta, apresentada na ilustração x, também envolvia funcionalidades do cenário do Coordenador de T.I. e 83,3% dos participantes relataram facilidade em executar a tarefa proposta, enquanto 16,7% encontraram alguma dificuldade.

Ilustração 15 – Gráfico 3 – Teste de Usabilidade

Cenário 2: Coordenador de T.I. Foi fácil localizar e revisar as ordens pendentes?

6 respostas



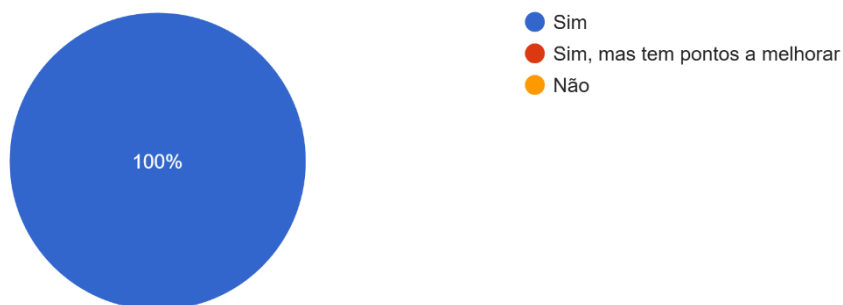
Fonte: Autores, 2024

A quarta pergunta, apresentada na ilustração x, foi direcionada ao cenário do Técnico de T.I., e 100% dos participantes relataram que o processo da tarefa foi intuitivo.

Ilustração 16 - Gráfico 4 – Teste de Usabilidade

Cenário 3: Técnico de T.I. O processo de atualização do status das ordens foi intuitivo?

6 respostas



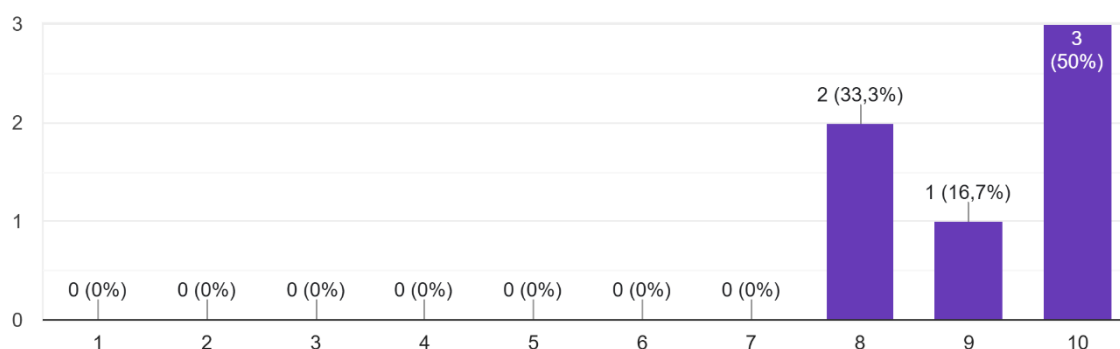
Fonte: Autores, 2024

Em uma escala de 1 a 10, os usuários classificaram a usabilidade e a experiência de uso do sistema com notas de 8 (33,3%), 9 (16,7%) e 10 (50%) (ilustração x).

Ilustração 17 – Gráfico 5 – Teste de usabilidade

Sistema geral Como você classificaria a sua experiência geral ao usar o sistema para executar ordens (em uma escala de 1 a 10)?

6 respostas



Fonte: Autores, 2024

3.2 CONCLUSÃO

O desenvolvimento e a prototipação do projeto foram realizados sob a orientação de um profissional da área de T.I., que atua como Coordenador do curso de Tecnologia em uma escola técnica. Esse profissional contribuiu não apenas na criação da ideia, mas também forneceu orientações sobre a metodologia e as regras de negócio relacionadas às ordens de serviço. Durante a fase inicial, ele esteve diretamente envolvido na construção do esquema do banco de dados e no delineamento do processo de solicitação, aprovação e execução das ordens de serviço na escola.

Após testes e orientações, concluímos que o que foi proposto atendeu às expectativas e aos resultados esperados. O principal objetivo do projeto era automatizar e facilitar o processo de requisição de ordens de serviço entre entidades, tornando-o compreensível até mesmo para usuários sem conhecimento prévio da regra de negócio.

Os resultados dos testes mostraram uma excelente prototipação de design e usabilidade, resultando em feedback positivo e sem necessidade de grandes

melhorias no processo do sistema. No entanto, com base nas orientações do coordenador, identificamos que a aplicação necessita de algumas melhorias funcionais para atender de forma mais completa às necessidades identificadas para uso em um contexto real. As funcionalidades iniciais implementadas representam um passo importante e promissor para futuros aprimoramentos.

REFERÊNCIAS

BARNUM, Carol M. **Usability Testing Essentials: Ready, Set...Test!**. 2. Ed. Morgan Kaufmann, 2020.

GATES, Bill. **Comunicação pessoal**. 1995.

GUEDES, Gilleanes Thorwald Araujo. **UML 2 - Uma Abordagem Prática**. 1. ed. São Paulo: Novatec Editora, 2009.

MACHADO, Felipe Nery Rodrigues. **Análise e Gestão de Requisitos de Software – Onde nascem os sistemas**. Editora Érica, 2018.

MIKELSTEN, Daniel. **Automação e Tecnologias Emergentes**. Volume 3 de Inteligência Artificial: A Quarta Revolução Industrial. E-book: Cambridge Stanford Books. 155p.

Pipefy. **Soluções**. Tecnologia da Informação. Disponível em: <https://www.pipefy.com/pt-br/itsm/>. Acesso em: 28 set. 2023.

PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de Software – Uma abordagem profissional**. 8 edição, Mc Graw Hill Editora, 2016.

Produttivo. **Funcionalidades**. Disponível em: <https://www.produttivo.com.br/funcionalidades/>. Acesso em: 27 set. 2023.

ServiceNow. **ServiceNow Platform**. Disponível em: <https://www.servicenow.com/now-platform.html>. Acesso em: 28 set. 2023.

SPENCER, Donna. **A Practical Guide to Information Architecture**. Five Simple Steps, 2010.

APÊNDICE A – CAMADA DE NEGÓCIO

Ilustração 18 – Código do método de atualização de ordem de serviço

```
2 referências
public async Task<OrdemServicoDto> UpdateOrdemServico(int userId, int osId, OrdemServicoDto model)
{
    try
    {
        var ordemServico = await _ordemServicoPersist.GetOrdemServicoByIdAsync(userId, osId, false);
        if (ordemServico == null) return null;

        model.Id = ordemServico.Id;
        model.UserId = userId;

        _mapper.Map(model, ordemServico);

        _gerealPersist.Update<OrdemServico>(ordemServico);
        if(await _gerealPersist.SaveChangesAsync())
        {
            var ordemRetorno = await _ordemServicoPersist.GetOrdemServicoByIdAsync(userId, ordemServico.Id, false);
            return _mapper.Map<OrdemServicoDto>(ordemRetorno);
        }

        return null;
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}
```

Fonte: Autores, 2024

Ilustração 1912 - Código do método de seleção de todas as ordens de serviço

```
2 referências
public async Task<OrdemServicoDto[]> GetAllOrdensServicoAsync(int userId, bool includeOrdemEquipamento = false)
{
    try
    {
        var ordemServico = await _ordemServicoPersist.GetAllOrdensServicoAsync(userId, includeOrdemEquipamento);
        if (ordemServico == null) return null;

        var result = _mapper.Map<OrdemServicoDto[]>(ordemServico);

        return result;
    }
    catch (Exception ex) { throw new Exception(ex.Message); }
}
```

Fonte: Autores, 2024

Ilustração 20 - Código do método de filtragem de ordem de serviço por usuário

```
2 referências
public async Task<OrdemServicoDto[]> GetAllOrdensServicoByUsuarioAsync(int userId, bool includeOrdemEquipamento)
{
    try
    {
        var ordemServico = await _ordemServicoPersist.GetAllOrdensServicoByUsuarioAsync(userId, includeOrdemEquipamento);
        if (ordemServico == null) return null;

        var result = _mapper.Map<OrdemServicoDto[]>(ordemServico);

        return result;
    }
    catch (Exception ex) { throw new Exception(ex.Message); }
}
```

Fonte: Autores, 2024

Ilustração 21 - Código método de filtragem de ordem de serviço por ID

```
2 referências
public async Task<OrdemServicoDto> GetOrdemServicoByIdAsync(int userId, int OrdemServicoId, bool includeOrdemEquipamento)
{
    try
    {
        var ordemServico = await _ordemServicoPersist.GetOrdemServicoByIdAsync(userId, OrdemServicoId, includeOrdemEquipamento);
        if (ordemServico == null) return null;

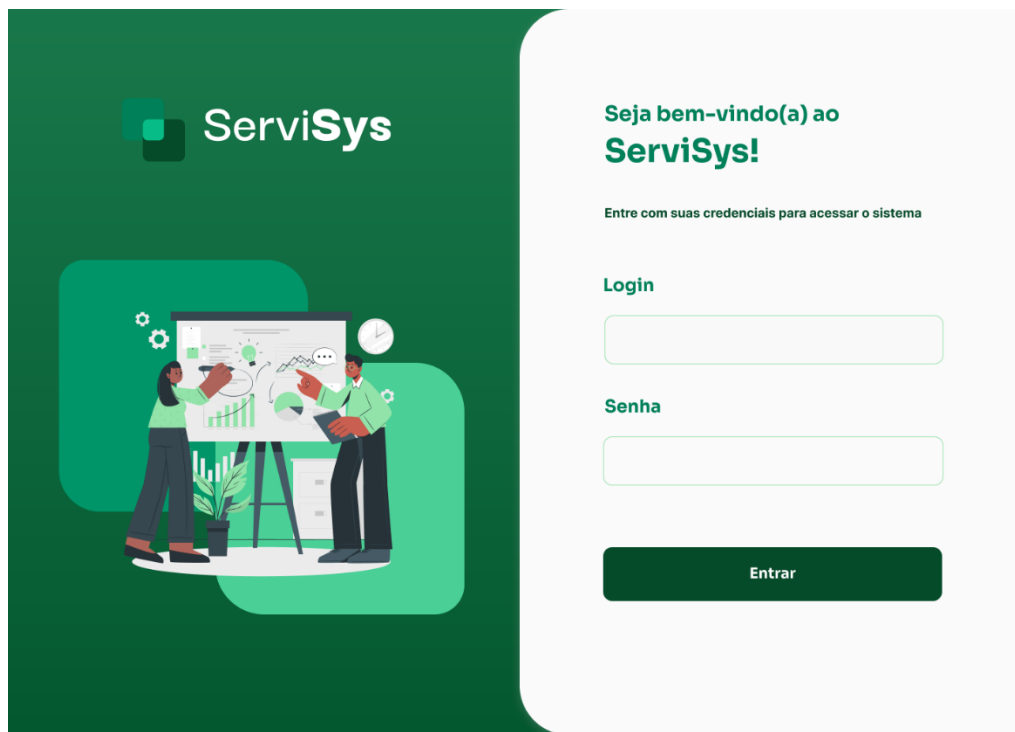
        var result = _mapper.Map<OrdemServicoDto>(ordemServico);

        return result;
    }
    catch (Exception ex) { throw new Exception(ex.Message); };
}
```

Fonte: Autores, 2024

APÊNDICE B – CAMADA DE APRESENTAÇÃO

Ilustração 22 – Página de Login



ServiSys

Seja bem-vindo(a) ao **ServiSys!**

Entre com suas credenciais para acessar o sistema

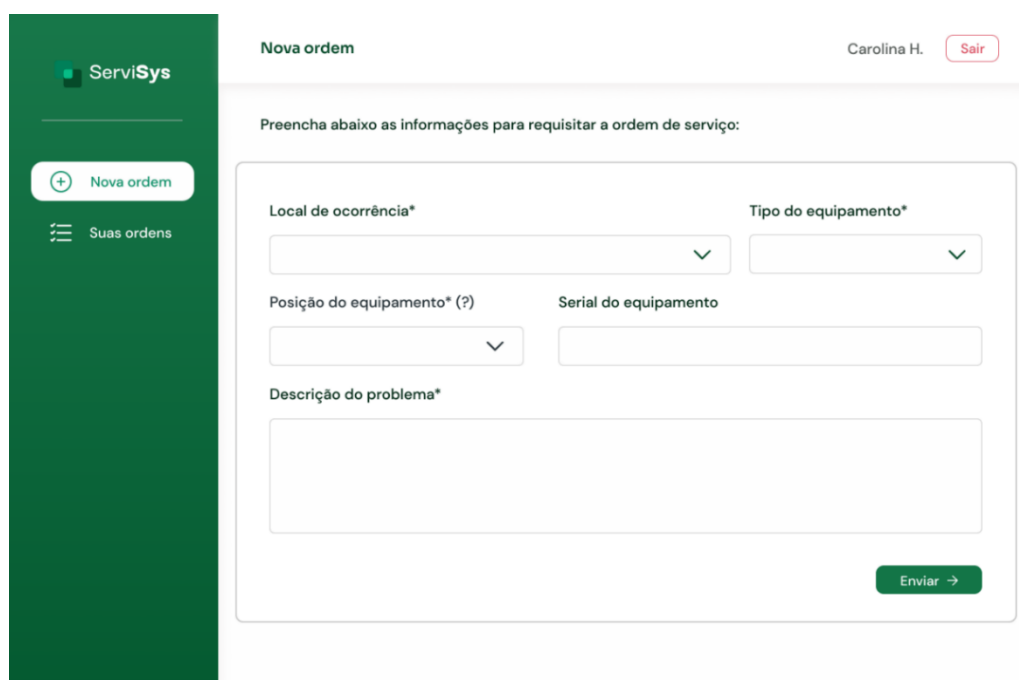
Login

Senha

Entrar

Fonte: Autores, 2024

Ilustração 23 – Página de solicitação de ordem de serviço



ServiSys

Nova ordem Carolina H. [Sair](#)

Preencha abaixo as informações para requisitar a ordem de serviço:

Local de ocorrência* Tipo do equipamento*

Posição do equipamento* (?) Serial do equipamento

Descrição do problema*

Enviar →

Fonte: Autores, 2024

Ilustração 24 – Página de aprovações pendentes

ServiSys ADM

Aprovações pendentes

Frederico A. [Sair](#)

Nessa página você encontra suas aprovações de ordens de serviço pendentes:

Ordem de serviço	Solicitante	Ações
OS4567SA	Juliana Nascimento	Ver ordem , Aprovar , Recusar
OS4567SA	Juliana Nascimento	Ver ordem , Aprovar , Recusar
OS4567SA	Patricia Santos	Ver ordem , Aprovar , Recusar
OS4567SA	Ayrton Nascimento	Ver ordem , Aprovar , Recusar

Fonte: Autores, 2024

Ilustração 25 – Página de aprovações do usuário

ServiSys ADM

Suas aprovações

Frederico A. [Sair](#)

Nessa página você encontra suas aprovações de ordens de serviço

ID da ordem	Status da ordem	Data da atribuição	Ações
OS4568SA	Aprovada ●	14-09-2023	Ver
OS4567SA	Aprovada ●	11-07-2023	Ver
OS4566SA	Recusada ●	27-06-2023	Ver
OS4565SA	Aprovada ●	11-03-2023	Ver

Fonte: Autores, 2024

Ilustração 2613 – Página de ordens de serviço do CPD

Ordens de serviço Thiago Lima Sair

Nessa página você encontra suas requisições de ordens de serviço

Todas as ordens Concluídas Em andamento

ID da ordem	Status da ordem	Data da atribuição	
OS4568SA	Em andamento	14-09-2023	Ver
OS4567SA	Concluída	11-07-2023	Ver
OS4566SA	Concluída	27-06-2023	Ver
OS4565SA	Compra	11-03-2023	Ver

Fonte: Autores, 2024

Ilustração 27 – Página de alteração de status da ordem de serviço

Ordem de serviço #OS4565SA

Local de ocorrência:
Laboratório de informática O7

Data da atribuição:
14-09-2023

Tipo do equipamento:
Computador

Status:
Em análise

Posição do equipamento:
Mesa O4

Serial do equipamento:
-

Descrição do problema:
O computador não está ligando e ouve bips

Salvar

Fonte: Autores, 2024

Ilustração 28 – Página de cadastro de usuário

ServiSys
Administrador

Novo usuário

Administrador [Sair](#)

Preencha abaixo as informações para requisitar a ordem de serviço:

Primeiro nome do usuário

Ultimo nome do usuário

E-mail do usuário

Nível do usuário Cargo do usuário

Senha

[Cadastrar](#) ✓

Fonte: Autores, 2024

APÊNDICE C – RESPOSTAS DO TESTE DE CARD SORTING**Tabela 6** – Respostas dos participantes do *Card Sorting*

Pergunta: Há alguma funcionalidade específica que você sente falta no sistema atual?	
Participante	Resposta do questionário
1	Não
2	Não, o sistema cumpre a sua função e parece ser bem completo.
3	Não.
4	tudo perfeito e icônico
5	Não
6	não
7	Creio que a "lista de todas as ordens aprovadas ou rejeitadas" deveria ser posta em uma página específica após clicar em um botão de menu fixo como "Suas aprovações" do Coordenador de T.I.
8	A Lista de todas as ordens aprovadas ou rejeitadas deveria estar em uma página específica ligado ao botão suas aprovações do menu fixo do Coordenador do TI.

Fonte: Autores, 2024