**Contextualização da Empresa Contratante**

ConnectWay Tecnologias e Soluções é uma empresa de pequeno a médio porte, localizada em São Paulo, especializada na prestação de serviços de infraestrutura de TI para empresas dos mais diversos segmentos, como clínicas, escritórios e comércios locais.

Apesar de atender empresas externamente, a ConnectWay também possui uma equipe interna de suporte técnico externa exclusivamente para colaboradores e parceiros. Hoje, são cerca de 150 dispositivos ativos (entre computadores, notebooks e equipamentos de rede), sendo atendidos por 4 técnicos internos que lidam com 70 aque lidam com 70 a 90 chamados semanais, entre obrigações simples e falhas mais graves.

**Situação Atual e Desafios Enfrentados**

Atualmente, a abertura de chamadas técnicas na ConnectWay ocorre de forma manual, via ligações telefônicas ou e-mails, diretamente com o setor de TI. Esse modelo gerou os seguintes problemas:

**Falta de Registro Centralizado**

* 70% dos chamados são abertos por e-mail, com exceções inconsistentes.
* 30% são feitos por telefone, com risco de perda total do histórico.
* Técnicos relataram dificuldade em encontrar o histórico de atendimento de um mesmo problema.
* Exemplo: Um computador com falha no acesso à rede foi atendido 3 vezes por semana, pois os técnicos não sabiam do histórico anterior.

**Priorização Ineficiente dos Chamados**

* Não há critérios definidos para níveis de criticidade.
* Um chamado simples (ex: redefinir senha) pode ser atendido antes de uma falha de rede.
* Os técnicos utilizam “ordem de chegada” como categorias, o que afeta os serviços essenciais.

**Falta de SLA e Acompanhamento**

* Não há definição de prazos máximos para atendimento.
* 40% dos usuários reclamam de demora superior a 48h para resolver problemas simples.
* Os usuários não recebem notificações sobre andamento ou resolução de chamada.
* Exemplo: Um problema com a impressora de um setor levou 5 dias para ser resolvido, pois não havia acompanhamento.

**Sobrecarga da Equipe Técnica**

* Os 4 técnicos internos atendem em média 22 chamadas por semana, cada um.
* Cerca de 50% dos chamados são dúvidas repetitivas, como:
* “Como instalar uma impressora?”
* “Erro de login no sistema X”
* Isso consome tempo valioso, que poderia ser usado em chamadas mais críticas.

**Ausência de Relatórios Gerenciais**

* Não há controle automático de:
* Quantidade de chamados abertos/fechados por técnico.
* Tipos de problemas mais frequentes.
* Tempo médio de resolução.
* Isso dificulta a tomada de decisões estratégicas, como contratação ou treinamento.

**Comunicação Fragmentada**

* Uma chamada pode ser aberta por e-mail, mas atualizada por telefone.
* Isso causa desorganização e falhas de informação.
* Os técnicos precisam procurar detalhes em conversas de e-mail, ligações e anotações manuais.

**Insatisfação dos Usuários Internos**

* Em pesquisa interna realizada pela própria ConnectWay (fictícia), os resultados apresentados:
* 62% dos usuários acham o atendimento técnico lento.
* 48% já abriram o mesmo chamado mais de uma vez por falta de retorno.
* 31% resolveram o problema por conta própria. ou com ajuda de terceiros.

**Impactos Práticos na Empresa**

* Perda de produtividade dos funcionários devido a falhas recorrentes e lentidão no suporte.
* Retrabalho constante, aumentando os custos operacionais.
* Dificuldade em escalar a operação de forma eficiente.
* Queda na satisfação interna dos colaboradores.

Gráfico dos tipos de chamadas internas mais frequentes:



**Empresa Desenvolvedora Contratada**

**Soluções NexaSoft**

**CNPJ:**

00.123.456/0001-78

**Endereço:**

Rua da Inovação, nº 245 – Sala 801 Bairro: Barra Funda

Cidade: São Paulo – SP CEP: 04567-000

**Contato:**

* Telefone: (11) 3123-4567
* WhatsApp: (11) 91234-5678
* E-mail:

**Descrição da empresa contratada**

Fundada em 2021, a NexaSoft Solutions é uma empresa de tecnologia especializada no desenvolvimento de sistemas personalizados para pequenas e médias empresas. Seu principal diferencial está na oferta de soluções integradas e eficientes, que abrangem plataformas desktop, web e mobile, sempre com foco em desempenho, escalabilidade e usabilidade. Composta por uma equipe enxuta de cinco profissionais altamente qualificados, a NexaSoft promove um atendimento próximo, ágil e altamente personalizado, garantindo flexibilidade e alto grau de adaptação às necessidades específicas de cada cliente.

**Funcionalidade da Inteligência Artificial (Estudo em Andamento)**

* Categorização automática de chamadas com base no conteúdo da descrição.
* Sugestão de soluções fundamentadas em histórico e base de conhecimento.
* Possibilidade de um assistente virtual de atendimento inicial, que filtra e prioriza chamadas.

**Proposta de Projeto:**

Sistema Digital Integrado de Gestão de Chamados com Inteligência Artificial.

**Objetivo Geral**

Desenvolver um Sistema Digital Integrado de Gestão de Chamados com componentes Desktop, Web e Mobile, utilizando tecnologias modernas e seguras, focado na qualidade do software (usabilidade, acessibilidade, desempenho, segurança e manutenibilidade), e que atende à Lei Geral de Proteção de Dados (LGPD).

**Descrição do Projeto**

**Arquitetura Modular**

* Interface Desktop (WPF ou Windows Forms) – gerenciamento interno dos chamados, acessível a técnicos e administradores.
* Interface Web (ASP.NET + C#) – plataforma para colaboradores acompanharem seus chamados, com interface responsiva.
* Aplicativo Mobile (Android) – canal acessível para abertura e consulta de chamadas pelos usuários.
* Banco de Dados (SQL Server) – armazenamento estruturado, seguro e centralizado.
* Camada de IA – integração futura com OpenAI para sugestão automática de soluções.

|  |  |  |  |
| --- | --- | --- | --- |
| **Critério de Qualidade** | **Descrição** | **Ações Concretas no Projeto** | **Ferramentas/Técnicas Utilizadas** |
| **Usabilidade** | Facilidade com que o usuário pode aprender e operar o sistema | - Layout intuitivo com botões claros e organizados- Feedback visual (ex: carregando, salvando, erro)- Linguagem simples e direta | - UX/UI com base em padrões de design- Prototipação (Figma ou Draw.io)- Testes com usuários reais |
| **Acessibilidade** | Garantia de que o sistema possa ser utilizado por pessoas com diferentes habilidades | - Alto contraste entre texto e fundo- Suporte para leitores de tela- Navegação via teclado | - Padrões WCAG 2.1- Ferramentas de verificação de acessibilidade (ex: Axe DevTools) |
| **Desempenho** | Agilidade no carregamento e resposta do sistema | - Otimização de consultas no banco- Reutilização de componentes- Cache de dados locais no mobile | - SQL Server Profiler- ASP.NET Caching- Testes de estresse (JMeter, k6) |
| **Segurança** | Proteção dos dados e do acesso ao sistema | - Autenticação e autorização por níveis de acesso- Criptografia de senhas e dados sensíveis- Conformidade com a LGPD | - ASP.NET Identity- Hash de senha (bcrypt)- Políticas de segurança (firewall, acesso mínimo) |
| **Testabilidade** | Facilidade de testar e validar o sistema | - Código modular e reutilizável- Testes automatizados por módulo- Relatórios de cobertura de testes | - NUnit, xUnit (para C#)- Testes manuais com checklist- Firebase Test Lab (Android) |
| **Manutenibilidade** | Facilidade de corrigir erros e fazer melhorias futuras | - Comentários no código- Documentação técnica do sistema e API- Uso de padrões (MVC) | - Swagger para documentação de API- Repositório Git bem estruturado |
| **Portabilidade** | Capacidade de funcionar em diferentes ambientes e dispositivos | - Versões para desktop, web e mobile- Interface responsiva- Compatibilidade com versões recentes do Windows e Android | - Xamarin / Android SDK- ASP.NET com design responsivo- Testes em múltiplos navegadores e dispositivos |

**Conformidade com a LGPD**

**Dados Pessoais Coletados:**

* Nome, e-mail, telefone, CPF, registros de chamadas.

**Ações para Conformidade:**

* Solicitação de consentimento explícito ao usuário no primeiro uso.
* Registro do motivo da coleta dos dados (finalidade).
* Restrição de acesso apenas para usuários autorizados (mínimo necessário).
* Implementação de critérios para anonimização ou exclusão de dados ao fim do ciclo de vida.
* Canal para o titular solicitar acesso, retificação ou exclusão de dados.
* Criação de um relatório de impacto à proteção de dados (RIPD).

**Resultados Esperados:**

* Redução de pelo menos 40% no tempo médio de resolução de chamados.
* Diminuição de falhas de comunicação e retrabalho.
* Aumento da satisfação dos usuários internos.
* Melhoria no controle gerencial e tomada de decisões baseadas em dados.
* Conformidade com a LGPD e mais confiança no relacionado de dados pessoais.

**Requisitos Funcionais (RF):**

|  |  |
| --- | --- |
| **Código** | **Descrição** |
| **RF01** | Permitir o cadastro de usuários (colaboradores, técnicos e administradores). |
| **RF02** | Permitir login com autenticação por nível de acesso. |
| **RF03** | Permitir que usuários abram chamados com descrição, categoria e urgência. |
| **RF04** | Permitir que técnicos visualizem, filtrem e atualizem o status dos chamados. |
| **RF05** | Permitir que usuários acompanhem o andamento de seus chamados. |
| **RF06** | Permitir que administradores cadastrem e editem categorias, usuários e configurações. |
| **RF07** | Gerar relatórios de atendimentos por período, tipo e status. |
| **RF08** | Enviar notificações (push ou e-mail) sobre atualizações dos chamados. |
| **RF09** | Registrar o histórico de ações realizadas em cada chamado. |
| **RF10** | Permitir o fechamento e avaliação do atendimento ao final do chamado. |
| **RF11** | Armazenar e consultar a base de conhecimento com soluções anteriores. |
| **RF12** | Permitir que o aplicativo mobile abra e acompanhe chamados remotamente. |
| **RF13** | (Futuro) Sugerir automaticamente uma categoria com base na descrição do chamado (IA). |
| **RF14** | (Futuro) Sugerir possíveis soluções com base em chamados anteriores (IA). |
| **RF15** | (Futuro) Permitir integração com serviços externos de IA (OpenAI, Watson, Azure AI). |

**NÃO FUNCIONAIS (RNF):**

|  |  |  |
| --- | --- | --- |
| **Código** | **Categoria** | **Descrição** |
| **RNF01** | Desempenho | O sistema deve responder a qualquer solicitação em até 3 segundos. |
| **RNF02** | Desempenho | Deve suportar até 200 chamados simultâneos sem perda de desempenho. |
| **RNF03** | Confiabilidade | Deve estar disponível para acesso 99% do tempo útil da semana. |
| **RNF04** | Usabilidade | A interface deve ser clara, intuitiva e com linguagem acessível. |
| **RNF05** | Acessibilidade | Deve permitir navegação por teclado e suporte a leitores de tela. |
| **RNF06** | Acessibilidade | Deve seguir contraste mínimo de 4,5:1 conforme padrões WCAG. |
| **RNF07** | Segurança | Dados pessoais devem ser criptografados e protegidos contra acesso não autorizado. |
| **RNF08** | Segurança | Autenticação e controle de acesso por níveis de permissão. |
| **RNF09** | LGPD | O sistema deve permitir consulta, edição e exclusão de dados pessoais conforme LGPD. |
| **RNF10** | Segurança | Todos os acessos e alterações devem ser registrados em log. |
| **RNF11** | Manutenibilidade | O sistema deve seguir arquitetura modular (MVC) e boa documentação. |
| **RNF12** | Escalabilidade | Deve permitir adição de novos módulos (como IA) sem reestruturar o sistema. |
| **RNF13** | Qualidade | O código-fonte deve ser versionado (Git) e seguir boas práticas de documentação. |

**Soluções para Dificuldades no Desenvolvimento:**

|  |  |  |
| --- | --- | --- |
| **Dificuldade** | **Descrição** | **Solução Proposta** |
| **Mudança de requisitos durante o projeto** | O cliente pode solicitar alterações mesmo após a aprovação inicial. | Adotar uma metodologia ágil (Scrum ou Kanban) com sprints curtos e validações contínuas. Criar um documento de escopo por etapa e formalizar mudanças. |
| **Equipe pequena (5 pessoas)** | Recursos humanos limitados para lidar com várias plataformas. | Dividir tarefas por especialidade (frontend, backend, mobile). Utilizar ferramentas de produtividade (Git, Trello, Azure DevOps). Fazer parcerias em partes específicas, se necessário. |
| **Dificuldade técnica em múltiplas tecnologias** | Nem todos os membros dominam WPF, Android, ASP.NET, SQL Server. | Promover treinamentos internos, criar tutoriais e repositórios compartilhados. Utilizar modelos prontos e boas práticas. |
| **Integração entre desktop, web e mobile** | Compartilhamento de dados entre plataformas pode gerar falhas. | Usar APIs RESTful padronizadas e bem documentadas. Garantir sincronização por banco de dados centralizado. Aplicar testes de integração contínuos. |
| **Falta de testes com usuários reais** | Funcionalidades podem ser entregues sem validação prática. | Criar versões MVP (mínimo produto viável) e realizar testes com funcionários reais da ConnectWay. Aplicar formulários de feedback. |
| **Baixa usabilidade e acessibilidade** | O sistema pode ser difícil de usar para pessoas com deficiência ou pouco conhecimento técnico. | Aplicar princípios de design universal. Usar cores com bom contraste, texto claro e navegação simples. Testar com leitores de tela e teclado. |
| **Demora no feedback do cliente** | Atrasos por não saber se a funcionalidade está aprovada ou precisa de ajustes. | Definir responsáveis de validação do lado da empresa cliente. Estabelecer reuniões quinzenais com demonstrações do progresso (dailies ou revisões). |
| **Conformidade com LGPD** | Risco de vazamento ou uso indevido de dados pessoais. | Implementar controle de consentimento, criptografia de dados pessoais, política de privacidade. Permitir acesso, edição e exclusão dos dados do usuário. |
| **Dificuldade na futura implementação de IA** | IA exige dados, testes e integração com serviços externos. | Planejar a IA para uma segunda fase. Iniciar coleta e classificação dos chamados para alimentar modelos futuros. Estudar APIs como OpenAI ou Azure AI. |
| **Falta de backup e segurança no banco** | Perda de dados ou tempo de inatividade crítico. | Configurar backups automáticos no SQL Server. Criar rotina de restauração testada. Aplicar camadas de segurança como firewall, criptografia e autenticação forte. |

**Meu Caso de Teste com a IA:**

Para validar essa funcionalidade, realizei testes simulados com exemplos reais de chamados. Abaixo alguns dos testes que utilizei:

|  |  |  |
| --- | --- | --- |
| **Descrição do Chamado (Input)** | **Categoria Esperada** | **Solução Esperada (Sugerida pela IA)** |
| "Não consigo imprimir na impressora da recepção" | Impressão/Dispositivos | Verificar conexão USB ou rede e reiniciar spooler de impressão |
| "Minha senha de login do sistema não funciona" | Acesso/Senhas | Realizar redefinição de senha via sistema interno |
| "A internet está oscilando muito no escritório 2" | Rede/Conectividade | Verificar cabo de rede, reiniciar switch local e checar link com provedor |

**Como a IA foi testada:**

* Para validação da funcionalidade de inteligência artificial, foram conduzidos testes com uma base composta por registros reais e simulados de chamados técnicos. Cada chamado incluía a descrição do problema, sua categoria e a solução aplicada anteriormente.
* Foi implementado um protótipo de categorização utilizando palavras-chave para sugestão automática de categorias e respectivas soluções. Os resultados sugeridos pela IA foram comparados com a categorização humana realizada por técnicos experientes.
* O processo contemplou cerca de 20 chamados distintos, e a avaliação final, conduzida por um técnico responsável, classificou as sugestões como corretas, parcialmente corretas ou incorretas. A taxa de acerto obtida foi de aproximadamente 75%, considerada bastante satisfatória para uma versão inicial.
* Todos os testes foram realizados com dados anonimizados, em conformidade com a LGPD, e as sugestões da IA foram submetidas à validação humana antes de qualquer ação. O processo foi documentado com foco na transparência e melhoria contínua.

**Cuidados com Validação e Conformidade com a LGPD**

* Durante os testes da funcionalidade de inteligência artificial, foram tomados os devidos cuidados para garantir o cumprimento da Lei Geral de Proteção de Dados (LGPD). Nenhum dado pessoal foi utilizado, sendo todos os testes realizados com informações técnicas ou previamente anonimizadas.
* A funcionalidade de IA foi concebida para atuar como uma ferramenta de apoio, sem autonomia decisória. Todas as sugestões geradas passam obrigatoriamente por validação técnica humana antes de qualquer ação prática.
* A implementação dessa funcionalidade está planejada para uma fase futura do projeto, considerando que a efetividade da IA depende de uma base consistente de dados históricos e de testes contínuos para aprimoramento seguro.
* Todo o processo de testes foi devidamente documentado, incluindo falhas, acertos e oportunidades de melhoria, garantindo total transparência e alinhamento com os princípios de qualidade e responsabilidade no desenvolvimento de soluções inteligentes.



Como controlar versões e mudanças

Controle de Versão com Git

**Ferramentas recomendadas :**

* **Git :** Sistema de controle de versão distribuída.
* **GitHub, GitLab, Bitbucket ou SourceForge :** Para repositórios online, além de funcionalidades extras como integração contínua.

**Boas práticas aplicadas**

Durante o desenvolvimento do projeto, foram aplicadas as seguintes boas práticas de controle de versão:

* Criação de Ramificações (Branches): Cada nova funcionalidade ou correção foi desenvolvida em uma branch específica, com nomes descritivos como feature/autenticacao-oauth ou bugfix/corrigir-layout.
* Commits Frequentes e Informativos: As alterações foram registradas por meio de commits pequenos, com mensagens claras e objetivas, seguindo convenções de nomenclatura como fix: ou feat:.
* Versionamento Semântico: O sistema adotou versionamento semântico, utilizando tags como v2.0.0 para indicar novas versões, melhorias ou correções significativas.
* Pull Requests (PRs): Antes da integração com a branch principal, todas as alterações foram submetidas à revisão por meio de Pull Requests, garantindo validação colaborativa.
* Backups Automatizados: O repositório foi configurado com rotinas de backup diário em plataformas como GitHub ou GitLab, assegurando a integridade dos dados e prevenção contra perdas.

Essas práticas contribuíram para a organização, rastreabilidade e qualidade do código ao longo de todo o ciclo de desenvolvimento.

**Gerenciamento de Mudanças Ferramentas recomendadas :**

* **Trello, Jira, Monday.com ou ClickUp :** Para gerenciar tarefas, sprints e mudanças de forma colaborativa.
* **Documentos de controle :** Como CHANGELOG.md, ou até mesmo planilhas compartilhadas para organizar as alterações.

**Procedimentos de Gerenciamento de Mudanças:**

* Solicitante da Mudança: Toda solicitação de alteração é devidamente registrada, identificando claramente o responsável pela demanda, seja um usuário final, membro da equipe técnica ou o próprio cliente.
* Descrição da Mudança: As modificações propostas são descritas de forma objetiva, incluindo o que será alterado, a justificativa da solicitação e os impactos potenciais no funcionamento do sistema.
* Dados da Solicitação: As informações relativas à mudança são organizadas e documentadas para garantir rastreabilidade, controle e histórico das alterações.
* Responsável pela Execução: A responsabilidade pela implementação da mudança é atribuída conforme a especialidade da equipe envolvida (frontend, backend, mobile, etc.), assegurando eficiência e domínio técnico.
* Versão Atual e Versão de Destino: Cada alteração é associada à versão vigente do sistema e à nova versão que será gerada após a implementação, promovendo clareza no versionamento.
* Status da Mudança: O acompanhamento do progresso é feito com base em um fluxo de status padronizado, como Aguardando Aprovação, Em Execução, Concluída ou Testada, facilitando a comunicação e o monitoramento entre as partes envolvidas**.**

Esse processo de controle contribuiu significativamente para a organização do projeto, mitigando retrabalho e assegurando a rastreabilidade de todas as modificações implementadas ao longo do ciclo de desenvolvimento.