

Desenvolvimento de Aplicativo Web para Controle de Equipamentos de TI da Rede Farmácia Nacional: Tecnologias e Metodologias

Lucas Samuel Dias¹, Júnio Moreira¹

¹Campus Patrocínio – Instituto Federal do Triângulo Mineiro (IFTM)
Patrocínio – MG – Brasil

lucas.samuel@estudante@iftm.edu.br, juniomoreira@iftm.edu.br

Abstract. *This article presents the development of the ‘RFN IT Equipment Control’ project. Due to the increase in equipment being sent to the IT Department of Rede Farmácia Nacional for maintenance, due to its diverse and growing number of branches, the student, who is also an IT Assistant at the company, realized the opportunity and need to apply the knowledge he had acquired to contribute to more effective and efficient management. Thus, collaborating not only to assist in his own work, but also to indirectly and directly assist other employees in the organization. In this sense, specific business rules were applied to the software to remedy the ineffectiveness and inefficiency of IT equipment control, which had been recurring for years. And, through this, with the use of various technologies, it will be shown, methodologically, how the system was developed and implemented to generate the expected results: undergoing several transformations until it was established in its current version.*

Resumo. *Este artigo apresenta o desenvolvimento do projeto ‘Controle de Equipamentos de TI da RFN’. Devido ao aumento de equipamentos sendo direcionados ao Departamento de TI da Rede Farmácia Nacional para manutenção, por suas diversas e crescentes filiais, percebeu-se pelo discente, que é também Auxiliar de TI da empresa, a oportunidade e necessidade de aplicar os conhecimentos conquistados para contribuir a uma gestão mais eficaz e eficiente. Colaborando, assim, para o auxílio não somente de seu próprio labor, mas também para os dos demais colaboradores, indireta e diretamente, da organização. Nesse sentido, no software foram aplicadas regras de negócios específicas que visam sanar a ineficácia e ineficiência do controle de equipamentos de TI — recorrentes há anos. E, mediante a isso, com o uso de diversas tecnologias, será mostrado, metodologicamente, como o sistema foi desenvolvido e implementado para gerar os resultados esperados: passando por várias transformações até se estabelecer na versão atual em que se situa.*

1. Introdução

Atualmente, a organização Rede Farmácia Nacional, fundada em Patrocínio - MG, em meados da década de 1980, detém um total de cinquenta e quatro filiais, as quais estão espalhadas em todo o Triângulo Mineiro; presente até mesmo em cidades mais distantes de Minas Gerais, como Unaí [Nacional 2025]. No cenário presente, desde o início desta década de 2020, é notável seu crescimento acelerado, e um dos motivos pelos quais isso

está sendo possível é em virtude do avanço dos produtos e projetos da tecnologia da informação.

Nesse sentido, o escopo do produto referente a um sistema de gestão de equipamentos é considerado algo genérico, assim como diante de qualquer outro de diferente segmento. No entanto, quando extraímos a partir dele uma ideia na qual visa aperfeiçoar operações estratégicas para a empresa, adentramos no escopo do projeto, que, por sua definição, é único e finito [PROJECT MANAGEMENT INSTITUTE 2021], como este referido projeto 'Controle de Equipamentos de TI da RFN'. Sistema esse designado ao Departamento de TI da Rede Nacional para sanar os problemas de ineficácia e ineficiência observada pelo discente, e também colaborador da empresa, desde sua admissão na respectiva organização farmacêutica como Auxiliar de TI.

Conforme fora obtendo conhecimento e desenvolvendo habilidades em diversas tecnologias, vendo, assim, a possibilidade de empregá-las em seu próprio labor diário, tal controle tornou-se mais eficaz e eficiente. Para corroborar com essa afirmativa, segue-se um cenário do qual não há mais: o funcionário de TI não conseguir verificar quais foram os envios realizados do mês — nos quais contêm informações relativas ao próprio equipamento, lojas (filiais) de origem e destino, data de envio, remetente —, como o de um microcomputador que chegara de uma filial específica para manutenção e reenvio.

Além disso, ao longo do período em que o sistema foi idealizado e aperfeiçoado, constatou-se que o crescimento acelerado da empresa tornava cada vez mais inviável o uso de planilhas e controles manuais. A ausência de rastreabilidade adequada, a dificuldade de visualizar informações históricas e a impossibilidade de gerar relatórios confiáveis impactavam diretamente a tomada de decisão no setor de TI. Assim, tornou-se evidente a necessidade de um sistema próprio, capaz de refletir com precisão os processos internos e atender às demandas específicas da organização.

Diante desse cenário, o desenvolvimento do software não se limitou apenas à implementação técnica, mas envolveu também uma compreensão aprofundada das rotinas operacionais e das regras de negócio da empresa. O projeto. Portanto, surgiu como uma solução estratégica, integrando conhecimentos adquiridos ao longo do curso tecnólogo de Análise e Desenvolvimento de Sistemas (ADS) a um problema real vivenciado no ambiente de trabalho. Essa integração permitiu não apenas otimizar atividades críticas, mas também promover maior confiabilidade, padronização, eficácia e eficiência no controle de equipamentos de TI da Rede Farmácia Nacional.

Inicialmente, ainda que planilhas eletrônicas tivessem sido implementadas pelo discente, manipular dados tornou-se uma tarefa demasiadamente ineficaz e ineficiente nesse ambiente: propício a erros constantes e empecilhos para escalar outras funcionalidades facilitadoras. Portanto, um sistema robusto e específico, que proporciona percepções poderosas para os controladores com suas funcionalidades e gatilhos presentes, os quais automatizam o processo, tanto dentro quanto fora do sistema, fora desenvolvido.

2. Fundamentação Teórica

2.1. Conceitos sobre Sistemas de Controle Patrimonial

A gestão de processos dentro de uma organização, torna-se essencial para a resolução de problemas existentes e, por conseguinte, de futuros — os quais podem e devem ser

evitados para maximizar os lucros [Object Management Group (OMG) 2013]. Todavia, muitas empresas, especialmente as pequenas com um crescimento acelerado, ignoram, seja por comodidade ou ausência de conhecimento, os impactos negativos da falta desse gerenciamento. E, as que não ignoram, adequam-se, em muitas ocasiões, a sistemas genéricos, que poderiam ser mais úteis se fossem desenvolvidos para se adequarem ao escopo intrínseco de determinada empresa — como é o caso de sistemas de controle patrimonial.

Neste projeto, com a utilização de um sistema específico, voltado às regras de negócio do Departamento de TI da Rede Farmácia Nacional — identificadas cotidianamente pelo discente em seu próprio labor —, pôde-se desenvolver um *software* que vislumbra suas reais necessidades. Indo além de simplesmente cadastrar, editar, excluir e consultar informações de equipamentos, para culminar, durante o processo, em tomadas de decisões premeditadas acerca de diversos estados possíveis de serem analisados. Nesse sentido, mesmo que outra organização possuísse certa similaridade quanto ao escopo e requisitos em relação a essa, a aplicação haveria modificações, pois cada empresa é singular. Logo, todo e qualquer projeto é único.

As nuances notadas em cada etapa do controle de equipamentos de TI agregaram em funcionalidades únicas para a aplicação, as quais diferenciam-na de qualquer outra do segmento por sua singularidade.

Além disso, sistemas de controle patrimonial desempenham um papel fundamental na manutenção da integridade dos ativos corporativos, uma vez que permitem rastrear todo o ciclo de vida dos equipamentos — desde sua aquisição, uso, movimentações internas e externas, até seu eventual descarte. Esse acompanhamento contínuo reduz riscos de inconsistências, perdas ou redundâncias de informação, além de favorecer auditorias. Quando esse monitoramento é adaptado às particularidades de cada organização, como no caso específico da Rede Farmácia Nacional, o sistema deixa de ser apenas uma ferramenta de registro e passa a atuar como um mecanismo estratégico de suporte à gestão, ampliando a capacidade de tomada de decisão baseada em dados confiáveis.

2.2. Arquitetura *Model-View-Controller* (MVC)

A arquitetura *Model-View-Controller* incorporada no Spring Boot visa dividir as funções dos códigos, tornando-os reutilizáveis, organizados e manuteníveis [SPRING 2024]. Dissonante a isso, como na primeira versão do software, comentado na subseção a seguir, não se utilizou integralmente o MVC, essas três características tornaram-se complexadas de serem concretizadas nela.

Model: Os atributos e as entidades em si são configurados para se refletirem e comunicarem diretamente no banco de dados. Funções são elaboradas e organizadas para serem utilizadas nos parâmetros de *repository* e *service* para leitura, salvamento, atualização e deleção de informações.

Controller: Via HTTP, os métodos padrão de transferência para páginas HTML utilizados foram: GET e POST. Ambos são configurados para realizar as operações CRUD (*Create*, *Read*, *Update* e *Delete*). O GET é responsável por *read* e *delete*, enquanto POST por *create* e *update*.

View: Os *templates* das entidades estruturados em *index*, *create* e *edit* comunicam-

se dinamicamente com o *controller*, que, por sua vez, comunica-se bilateralmente com o *model*. Essa ponte faz com que as informações sejam visualmente apresentadas ao usuário.

2.3. Conceito de Experiência do Usuário e Interface do Usuário (UX/UI)

A experiência do usuário e interface do usuário diante do sistema foram evoluindo de acordo com a percepção de inviabilidades presentes no primeiro protótipo de testes, conforme é demonstrado brevemente na interface da Figura 1, a partir do seu uso diário e longínquo. Perante a isso, notou-se que a eficácia e eficiência de vários processos poderiam ser melhor obtidos mediante uma remodelação que precisasse de menos cliques redundantes para realizar uma determinada ação, tornando-a mais fluida e natural, tanto funcional e quanto visualmente, como é visto parcialmente na Figura 1.



Figura 1. Tela de menu de computador da antiga versão (PHP)

Um exemplo cabível seria a necessidade de se passar, após clicar no botão de atualizar computador, o seu número de série, depois realizar uma busca para encontrá-lo clicando em um botão, e, por conseguinte, selecionar editar, atualizando o respectivo atributo, e salvar. Veja, o processo necessitou de no mínimo cinco cliques, além de serem contra intuitivos da maneira que estavam dispostos. Já na segunda versão, bastava-se pesquisar pelo atributo desejado na listagem e, conseqüentemente, encontrando-o dinamicamente, selecionar editar, modificando algum atributo, e salvar o respectivo computador. Agora, no total, precisou-se de quatro, mas mais que isso: o processo fora intuitivo.

As principais tecnologias de *frontend* utilizadas na primeira interface mantiveram-se as mesmas, mas já na segunda versão, a inserção do Bootstrap5 e de JavaScript, com bibliotecas e configurações pré-prontas para tabelas, filtragem de atributos e de outros elementos, auxiliaram o desenvolvimento a ser menos trabalhoso para se atingir os aprimoramentos desejados.

Logo, percebe-se que a UI e UX estão estritamente correlacionadas, uma vez que, quando se altera um, o outro, conseqüentemente, também é alterado, mesmo que minimamente. Desse maneira, contribuíram para a evolução mútua entre si neste *software*.

Enquanto localmente rodava-se com MySQL, optou-se pelo MariaDB no servidor — configurado com SSL, para adquirir maior segurança —, justamente pela sua performance superior em servidores com menor capacidade de processamento.

Além disso, há gatilhos que automatizam processos do sistema. Inicialmente, quando ele era centralizado no banco, também configurou-se *procedures* e *views*, que, embora não possuam mais ação direta na aplicação web, podem ser utilizadas para consultas no próprio SGBG por um Administrador de Banco de Dados (DBA), por exemplo, a fim de facilitar possíveis análises acentuadas. Tendo isso em vista, o MySQL Workbench, agilizou nas operações essenciais justamente por possuir interface gráfica, além de ser intuitiva.

Ademais, quanto à questão de segurança de dados, uma rotina de *backup* completo e automático foi definida para ser realizada todo dia, em um determinado horário em que o sistema não está em uso laboral. Em relação ao armazenamento, semanalmente os últimos backups são apagados a fim de não consumi-lo por inteiro, uma vez que é limitado. Corroborando a esse quesito, o arquivo gerado é compactado no formato da extensão .GZ.

3. Tecnologias Utilizadas

Em virtude do desenvolvimento do *software* ter se iniciado conforme o discente fora progredindo suas habilidades e conhecimentos sobre várias ferramentas, à princípio, a ideia originou-se a partir dos três estágios oriundos de banco de dados: conceitual, lógico e físico [ELMASRI and NAVATHE 2018]. Antes de ser implementado na terceira fase, com MySQL, foi testado em planilhas eletrônicas para entender seu real funcionamento quanto às regras de negócios do projeto.

Posteriormente, desenvolveu-se uma interface web utilizando HTML5, CSS3 e PHP — cogitou-se, à princípio, desenvolvê-lo em Java Swing, mas, devido à alta complexidade, e por ser ultrapassado, a ideia foi descartada. Tendo isso em vista, iniciou-se uma versão com Java Web pura, sem *frameworks*, contudo, por causa dos mesmos poréns, essa versão também fora desconsiderada.

Após certo período de uso, notou-se falta de padronização do *backend* em PHP, prejudicando, assim, a manutenibilidade dos códigos e visualização de melhorias para torná-lo mais eficiente. Nesse sentido, concomitantemente, percebeu-se um pseudo-MVC empregado, no qual havia diversos arquivos de configuração em demasia, os quais poderiam ser melhor organizados seguindo, fidedignamente, a estrutura Model-View-Controller (MVC). Além disso, também notou-se que a experiência e interface do usuário (UX/UI) não estavam condizente com os melhores protocolos existentes.

Diante disso, o *software* fora reformulado para cumprir tais quesitos. Contemplaram-se oportunidades e conveniência de reformulá-lo mediante o uso de outras ferramentas e linguagens, e aprimorar o uso das tecnologias já presentes. Dessa forma, a versão atual do projeto incorpora estas: Spring Boot (Java), JavaScript, HTML5, CSS3, Bootstrap5 e Thymeleaf.

Quanto à hospedagem, os serviços rodavam localmente na máquina empresarial do discente. No entanto, por razões de escalabilidade de acessos e segurança de dados, atualmente estão alocados em uma máquina virtual (VM) da Oracle Cloud, com sistema

operacional Linux Oracle. Logo, o servidor web e o de banco de dados estabelecidos foram, respectivamente: Nginx e MariaDB.

3.1. Tecnologias de *Frontend*

O HTML (*HyperText Markup Language*) é uma linguagem de marcação desenvolvida para aplicações web que visa estruturar os principais elementos de uma página web via *tags* [DEITEL and DEITEL 2020]. Alguns deles são: campos de texto, botões, tabelas, caixas de separação etc. Junto à ferramenta *template* Thymeleaf, há maior dinamismo das páginas, integrando-as facilmente ao *backend* em Java [THYMELEAF 2024]. Em conjunto com CSS (*Cascading Style Sheets*), torna a apresentação da estrutura personalizável, possibilitando a interface do usuário ser mais agradável e persuasiva [WORLD WIDE WEB CONSORTIUM 2025].

Para facilitar *designs* responsivos, como é o caso deste sistema, utilizado em dispositivos com tamanho de telas variadas, configurações pré-prontas do CSS puro vêm embutidas, bem como funcionalidades que seriam complexas de serem feitas estritamente com JavaScript, no *framework* Bootstrap.

O JavaScript é uma linguagem versátil, utilizada tanto no *frontend* quanto no *backend*, a partir da qual se desenvolveu diversos *frameworks* modernos e famigerados, como React.js e Node.js [DEITEL and DEITEL 2020].

Nessa perspectiva, o uso de bibliotecas do JavaScript, como DataTables e Select2, são responsáveis por tornar a configuração, respectivamente, de tabelas e listas suspensas, fácil e ágil. E visualmente, para o usuário, esteticamente minimalistas, agradáveis e intuitivas, seguindo os conceitos de UI/UX. Outra funcionalidade inserida, proporcionada pela biblioteca QuaggaJS do JS, refere-se à possibilidade de decodificar códigos de barras via câmera do dispositivo utilizado, seja qual for — desde que possua.

3.2. Tecnologias de *Backend*

O Spring Boot é um *framework* baseado em Java desenvolvido para simplificar as configurações referentes ao *backend* (SPRING, 2024). Tendo isso em visto, neste projeto, compõem-no as seguintes principais dependências:

- **Spring Web:** Define a aplicação como um projeto web, habilitando o Spring MVC e o servidor Tomcat embutido;
- **Spring JPA:** Habilita o uso da JPA (*Java Persistence API*), com *Hibernate* por padrão, para mapeamento objeto-relacional e acesso a dados;
- **Spring Security:** Implementa recursos de segurança, como autenticação, autorização e proteção básica de *endpoints*;
- **Spring Validation:** Adiciona suporte para validação e especificação de dados com *Bean Validation*;
- **Spring E-mail:** Fornece a funcionalidade de envio de e-mails;
- **Spring Thymeleaf:** É o motor de *template* que captura os dados do *backend* (Spring MVC) e os insere em arquivos HTML.

O JavaScript utilizado no *backend* fora útil para manipular, mediante à importação de biblioteca específica, arquivos Excel (.XLSX) por meio de tabelas escondidas e dispostas no diretório 'Planilhas'. Nelas há informação de data, atributo com o qual a delimitação de início e fim é obtida para salvar arquivo ou redirecioná-lo ao *endpoint* de e-mail para envio ao destinatário.

3.3. Apache Maven

Apache Maven responsabiliza-se por automatizar e gerenciar a compilação, empacotamento, documentação, testes e a distribuição de projetos Java, fornecendo um modelo de projeto uniforme, um ciclo de vida de *build* padronizado e um sistema robusto de gerenciamento de dependências para o projeto Spring Boot [SPRING 2024].

O Maven atinge essas responsabilidades por meio de seu arquivo central *pom.xml* (*Project Object Model*), no qual o desenvolvedor declara as dependências — bibliotecas, extensões, ferramentas para compilação, testes etc. —, e as metainformações do projeto. Ele segue o princípio de Convenção sobre Configuração, impondo uma estrutura de diretórios padrão e um ciclo de vida de *build* fixo que, quando executado, via *mvn package*, garante que o código seja compilado, os testes sejam executados, e um JAR executável final seja produzido, pronto para *deploy* no ambiente de produção. Essa padronização e automatização são cruciais para a agilidade no desenvolvimento com Spring Boot.

3.4. Máquina Virtual: Oracle Cloud

A máquina virtual gratuita da Oracle Cloud, rodando uma distribuição Linux da Oracle, sem interface gráfica, encaixou-se perfeitamente para se ter um ambiente com a melhor performance possível, ainda que fosse limitada à 1 GB de RAM e 50 GB de armazenamento. Houve, nesse sentido, configurações extras, como acréscimo de memória virtual (*swap*) de 4 GB para garantir eficiência para a VM e, consequentemente, fluidez para a aplicação web.

É possível acessá-la remotamente de qualquer lugar, desde que possua sua chave SSH. Ela, por sua vez, é única. E o processo para recuperá-la, em caso de perda, é extremamente penoso, podendo até mesmo ser necessário realizar uma configuração completamente do zero da instância para rodar o sistema [ORACLE 2025].

O arquivo .JAR gerado pelo Maven foi configurado no SO, mediante instalação do *driver* Java, para rodar como um serviço (*systemd*). Caso, por algum motivo, seja derrubado, reiniciará automaticamente, conforme configurado.

Em relação a configuração de segurança de redes, configurou-se um *firewall*, visando mitigar acessos maliciosos, via internet, que visam atacar o sistema, os quais prejudicam, dessa forma, a confiabilidade da aplicação. O serviço em questão escolhido, por ser leve frente a outros de mesma designação, é o *iptables*.

Nesse viés, os recursos do *iptables* foram instalados e configurados para permitir o acesso interno e externo somente de portas essenciais: localhost (127.0.0.1), SSH (22), HTTPS (443), restringindo, assim, o acesso às demais. No mais, objetivou-se limitar o acesso do MariaDB (3306) fora da aplicação web apenas a dois endereços de IP, os quais correspondem à rede interna da organização e a domiciliar do Auxiliar de TI, a fim de auxiliar a manipulação de dados, por meio do MySQL Workbench, sem a necessidade de estar dentro do terminal do banco na VM.

3.5. Servidor Web: Nginx

O servidor Nginx foi o adotado dentro da VM da Oracle Cloud em vez do Apache por ser, atualmente, mais moderno e performático, necessitando de menos processamento de máquina [KUNDA et al. 2017].

O certificado SSL/TLS e configurações de HTTPS e DNS, seguindo as boas práticas de segurança de Redes de Computadores, foram instaurados. Quanto ao último, usou-se o provedor gratuito 'duckdns.org' para personalizar o domínio, atrelando-o ao nome do respectivo sistema: Controle de Equipamentos de TI da RFN.

Por quesitos de ética e segurança, tanto o IP público do servidor/VM quanto o DNS (nome do *site*) não serão disponibilizados neste artigo.

3.6. Dashboard: Power BI

O Power BI, desenvolvido pela Microsoft, é uma plataforma por meio da qual há a possibilidade de se utilizar de uma fonte de dados — como um banco de dados — para extrair, transformar e carregar (ETL) em *dashboards* capazes de facilitar percepções de um negócio específico [MICROSOFT 2025].

Nesse sentido, sua implementação foi o toque final do qual o sistema necessitava. Junto a isso, informações valiosas em níveis micro e macro, principalmente referente ao segundo, puderam ser obtidas de forma mais fácil e ágil. E, mediante sua linguagem integrada (DAX), formulou-se cálculos e medidas para auxiliar em tal obtenção.

4. Metodologia e Prototipação

A metodologia utilizada para reconhecer o escopo e os requisitos do negócio do departamento da empresa, bem como a forma com a qual desenvolvê-los para o desenvolvimento de um sistema robusto, deu-se logo no início do curso tecnólogo de Análise e Desenvolvimento de Sistemas (ADS). Disciplinas como Engenharia de Software, Banco de Dados, Desenvolvimento Web, Experiência de Usuário, Laboratório de Redes, Segurança da Informação, Gerenciamento de Processos de Negócios, Gestão de Projetos, Extensão e Empreendedorismo agregaram, por meio do exercício de habilidades essenciais, no conhecimento necessário para desenvolver este projeto.

Ainda que não seja abordado detalhadamente: em relação à documentação e diagramas oriundos da Engenharia de Software, todos esses artefatos foram realizados. A seguir, há uma citação de quais foram utilizados para auxiliar no desenvolvimento.

Documentos de Requisitos: Obtém-se discernimento acerca do escopo do produto e de seus requisitos funcionais e não funcionais.

Modelos estruturais: Diagrama de Classes, Diagrama de Objetos, Diagrama de Componentes, Diagrama de Pacotes, Diagrama de Estrutura Composta, Diagrama de Instalação e Diagrama de Perfil de Extensão.

Modelos comportamentais: Diagrama de Casos de Usos, Diagrama de Atividades, Diagrama de Transição de Estados, Diagrama de Sequência, Diagrama de Comunicação e Diagrama de Tempo.

Embora não fosse a melhor forma, a prototipagem foi realizada quando se notou que o primeiro modelo, apresentado na amostragem da Figura 1, prejudicava a experiência

e interface do usuário. Desse modo, para auxiliá-lo, utilizou-se da ferramenta Figma para modelar uma prototipação de baixa e alta fidelidade, para nortear o desenvolvimento da segunda e definitiva versão.

A seguir, na Figura 1, há a representação, via protótipo de alta fidelidade, de como deveriam ser as telas de listagem das entidades no sistema final.

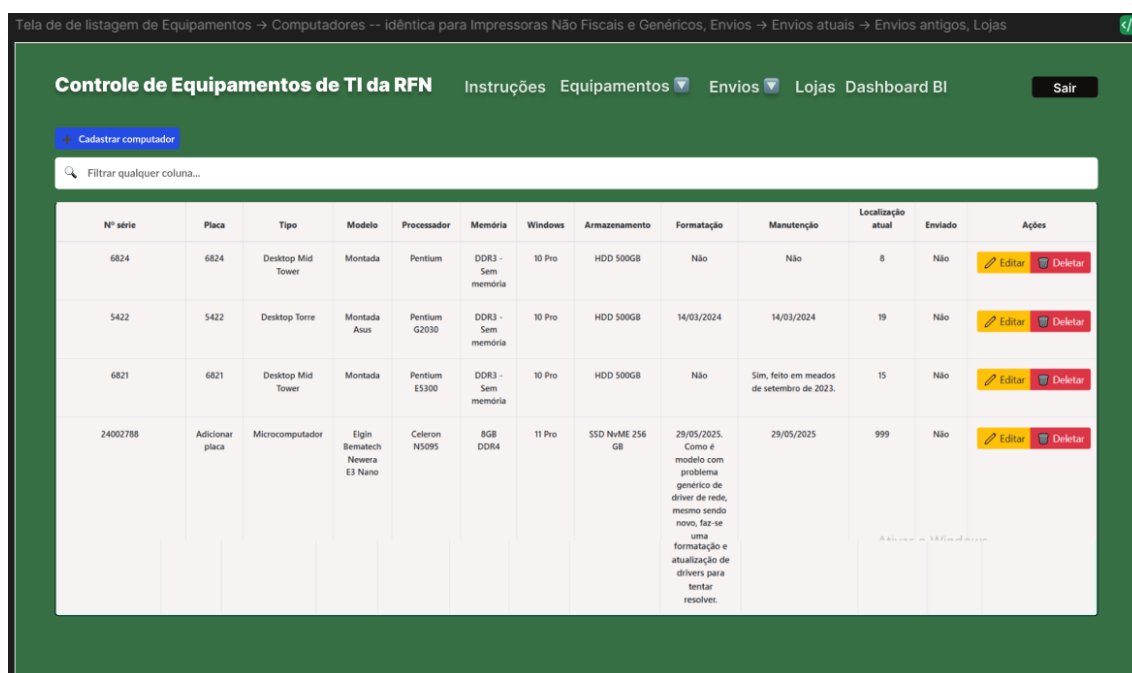


Figura 3. Protótipo de alta fidelidade da entidade de computador desenvolvido no Figma

Os conceitos da disciplina Experiência de Usuário possuíram profunda importância na resolução de problemas relacionados à UI/UX. Eles foram sentidos pelo discente, que, para comprovar sua visão, solicitava opiniões de outrem: colegas de trabalho da TI ou colegas do curso de ADS, e eles sugestionavam mostrando vias para melhoria — especialmente em relação às tonalidades das cores, em razão de seu daltonismo.

O ambiente de edição de código-fonte utilizado para desenvolver foi o Visual Studio Code. A fim de facilitar o desenvolvimento, utilizou-se o versionamento de códigos por meio do Git (repositório local) e GitHub (repositório remoto), gerenciados pelo GitHub Desktop.

4.1. Diagrama BPMN (*Business Process Model and Notation*)

O diagrama BPMN, elaborado na plataforma HEFLO, representa de forma detalhada o processo de controle de equipamentos de TI da empresa, permitindo compreender claramente o fluxo entre todas as entidades envolvidas — Auxiliar de TI, Gerente de TI, Contabilidade (Patrimônio) e Lojas (filiais). A Figura 4 apresenta essa dinâmica organizada por raias, evidenciando os responsáveis e as interações que compõem cada etapa do processo.

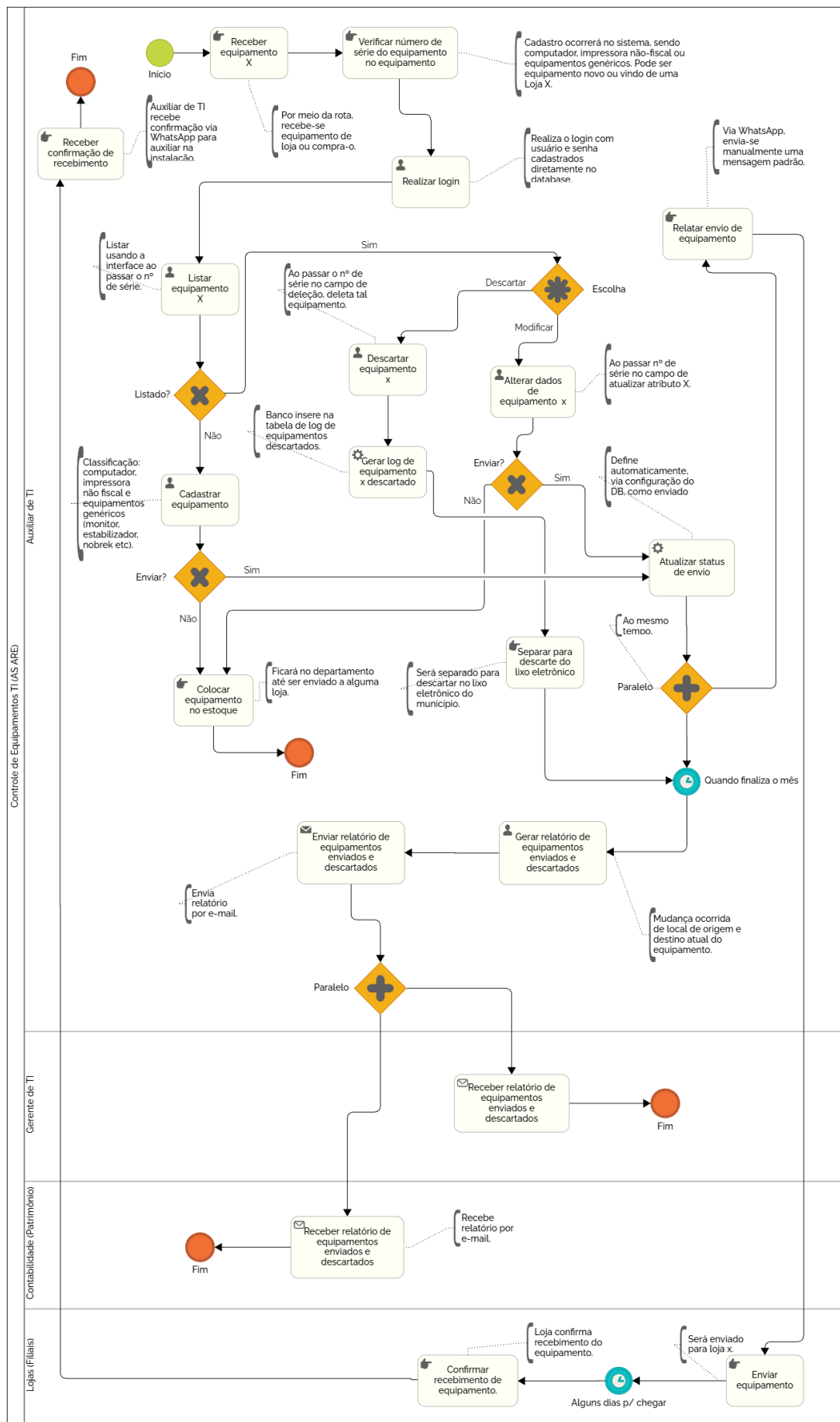


Figura 4. Diagrama BPMN do processo de controle de equipamentos da RFN

Vale ressaltar que o estado atual de como os processos ocorrem (*AS IS*) já foi modelado na perspectiva de como deveriam funcionar (*TO BE*), refletindo o cenário ideal de operação após a padronização e automação proporcionadas pelo sistema.

O envolvimento com outros setores fora da TI, como é o caso de Contabilidade (Patrimônio) e Lojas (Filiais), denota a completude de todo o projeto. E, embora seja perceptível a riqueza de nuances de vários processos que o envolvem, funcionalidades de serviços do sistema não foram detalhadas, em toda sua integridade, ao ponto de demonstrar tamanha eficácia e eficiência obtidas.

Essa evolução direta permitiu identificar gargalos, redundâncias e oportunidades de melhoria que não eram previamente evidenciadas. Além disso, o diagrama reforça a interdependência entre setores e destaca como a aplicação proposta integra e otimiza etapas críticas do fluxo organizacional. Tendo todas as indagações devidamente respondidas, entende-se que o projeto é plausível de ser executado a fim de maximizar eficácia e eficiência do que lhe é apresentado como desafio: controlar os equipamentos de TI da Rede Farmácia Nacional.

Ademais, algumas dessas etapas foram devidamente formalizadas somente após a implementação do sistema, como é o caso de envio semiautomático de relatórios mensais de equipamentos enviados e descartados via e-mail para Patrimônio e Gerente de TI, que será detalhado posteriormente.

4.2. Project Model Canvas (PMC)

O Project Model Canvas (PMC), desenvolvido pelo brasileiro José Finocchio Júnior [FINOCCHIO JÚNIOR 2013] a partir do Business Model Canvas de Alexander Osterwalder [OSTERWALDER and PIGNEUR 2010], adapta a proposta original. Conforme ilustrado no diagrama da Figura 5.

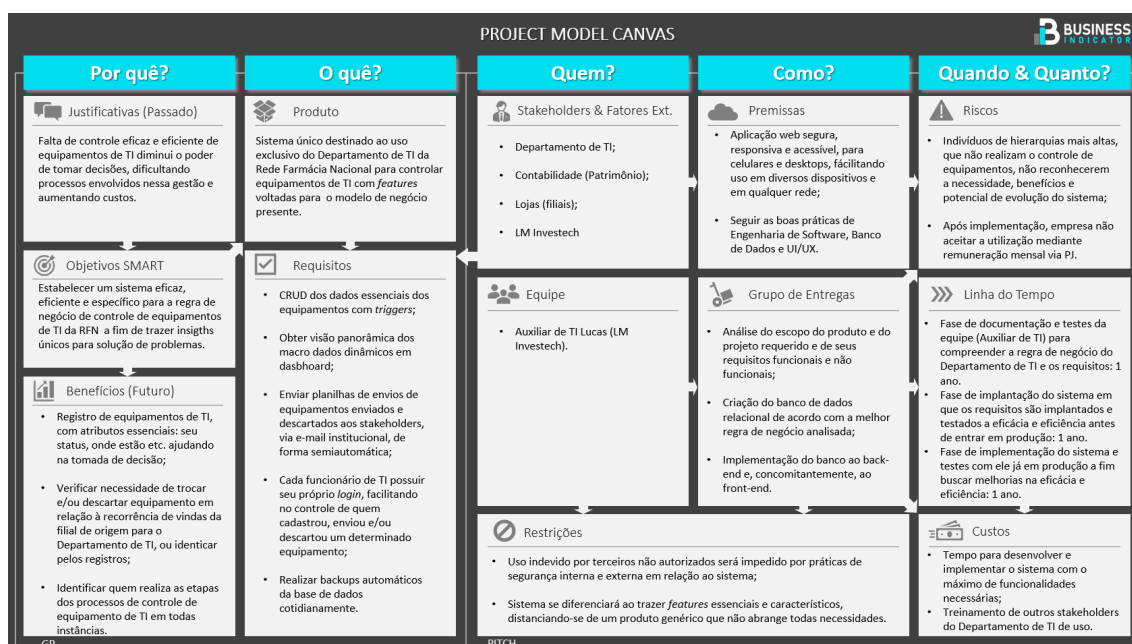


Figura 5. Project Model Canvas do projeto Controle de Equipamentos de TI da RFN

O PMC organiza o projeto em blocos temáticos que devem ser preenchidos com respostas objetivas às questões que os compõem. Essa estrutura, originalmente voltada à categorização e avaliação de viabilidade de modelos de negócio, é adaptada no contexto de projetos para permitir a organização, compreensão e análise integrada de seus principais elementos, facilitando a visualização do propósito, das entregas e das estratégias necessárias para sua execução.

O preenchimento completo desses blocos possibilita avaliar a viabilidade do projeto, sua coerência interna e sua capacidade de gerar os resultados desejados dentro da organização, especialmente no que se refere à eficácia e eficiência dos processos envolvidos [PROJECT MANAGEMENT INSTITUTE 2021]. Essa visão estruturada também auxilia na identificação de riscos, oportunidades e recursos necessários antes da execução.

Por fim, ignorar a oportunidade de planejar, organizar e aprimorar processos por meio de um projeto estruturado pode levar a prejuízos inevitáveis. Quando a empresa deixa de empregar metodologias como o PMC de forma estratégica, tende a sofrer, no futuro, com retrabalhos, ineficiências e custos decorrentes da ausência de planejamento adequado.

5. Desenvolvimento do Sistema

A apresentação geral do *software* será demonstrada nas subseções a seguir. Foca-se nas principais funcionalidades desenvolvidas e aprimoradas, tendo como base os conceitos de UX/UI: menos cliques para executar uma determinada ação, padronização de cores e tonalidades pastéis, simetria de elementos, menos informações sendo apresentadas de uma só vez para o usuário para não deixá-lo confuso, entre outras características fundamentais.

5.1. Tela de *Login*

A tela de *login* denota dois campos para o controlador inserir seu usuário e senha, conforme é visto pela Figura 6. Tais dados, sendo efetivados pelo Spring Security, possibilitam o acesso ao sistema de acordo com os privilégios atribuídos para cada tipo de usuário: Administrador ou Controlador.

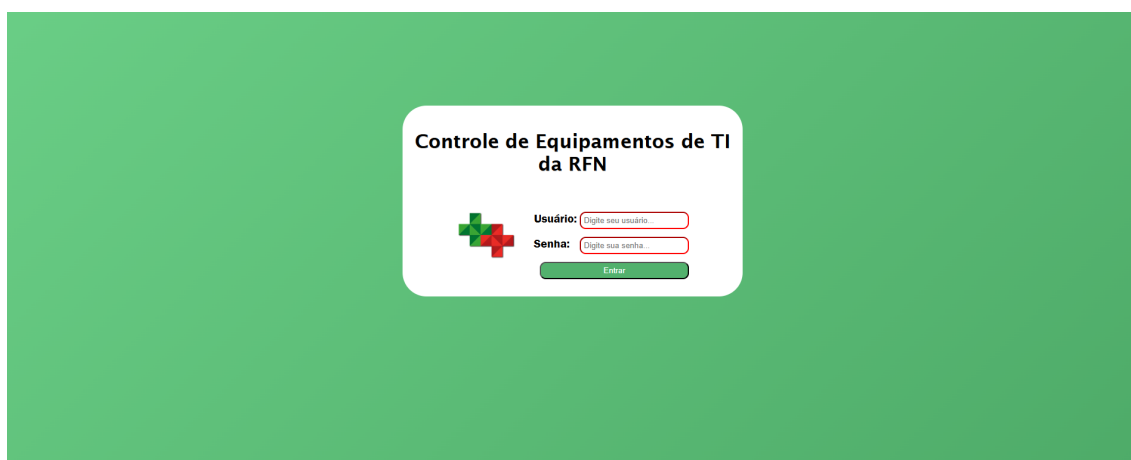


Figura 6. Tela de *login* do Controle de Equipamentos de TI da RFN

O motivo pelo qual não há tela de cadastro é justamente visar a segurança, uma vez que o sistema está acessível fora da rede interna — embora haja outros critérios de

segurança estabelecidos — para facilitar o acesso externo sem uma rede privada virtual (VPN). Logo, a criação de usuário e atribuição de seu papel ocorre pelo administrador. Após ele realizar o *login*, adquire acesso ao diretório */register* para efetuá-las.

5.2. Tela Principal: Dashboard Power BI

O Dashboard Power BI implementado, sendo a tela principal após o *login*, demonstra as informações armazenadas no servidor de banco de dados MariaDB. Está configurado para atualizar os dados, periodicamente, de hora em hora, desde que haja um *gateway* de dados locais para realizar a ponte de conexão segura com o SGBD. Embora o *gateway* de dados de redes virtuais pudesse ser o mais indicado, o processo envolve custos não planejados.

A visão macro em gráficos e *cards* dinâmicos integrados entre si instiga percepções que a visão micro, disposta em tabelas, detém dificuldades em realizar panoramicamente, como: contagem e filtragem de equipamentos em reserva, enviados e descartados por tipo, loja, cidade e mês, conforme é visto na Figura 7 a seguir.

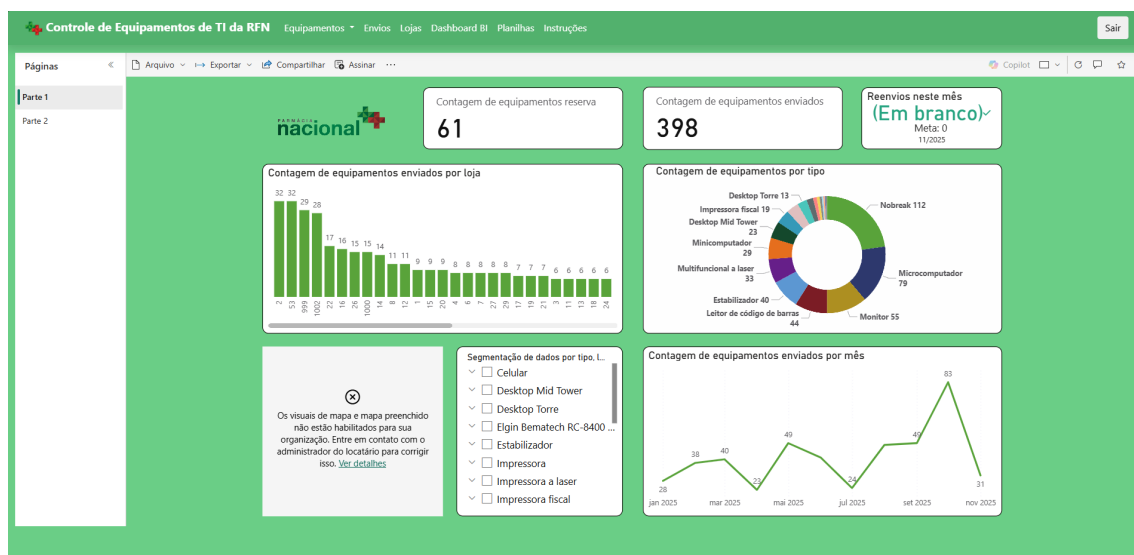


Figura 7. Tela da Parte 1 do *Dashboard* BI com visão panorâmica de macro dados

A configuração visual foi planejada para oferecer uma experiência clara, agradável e intuitiva ao usuário, permitindo uma análise eficiente dos dados. Além disso, o *dashboard* pode ser expandido com novas seções e perspectivas, ampliando ainda mais o suporte à tomada de decisões no modelo de negócio.

5.3. Tela de Listagem: Computador

A seção de listagem da entidade computador é semelhante às das outras duas entidades irmãs: impressoras não fiscais e genéricas. Os dados, após cadastrados, conforme é denotado na próxima subseção, são dispostos na forma de tabela. Nela pode-se configurar a quantidade de informações que se deseja ver em uma única página: 10, 25, 50 ou 100. Além de filtrá-las no campo de pesquisa por todos os atributos, os quais podem ser separados por espaço para discernir a busca em colunas distintas, conforme é visto na Figura 7[8] a seguir.

Controle de Equipamentos de TI da RFN Equipamentos * Envios Lojas Dashboard BI Planilhas Instruções Sair

Inserir computador

Exibir 10 resultados por página Pesquisar Buscar registros

Nº série	Placa	Tipo	Modelo	Processador	Memória	Windows	Armazenamento	Manutenção de software	Manutenção de hardware	Localização atual	Enviado	Ações
24003049	9106	Microcomputador	Microcomputador	i5 8257U	16GB DDR4	11 Pro	SSD NVMe 256GB	25/11/2025: formatação p/ correção de erros devido à estática; configuração padrão; otimização de SSD; usuário Suporte; BitDefender.	21/11/2025: retirado os parafusos da carcaça que geravam estática, retificando a máquina. Está funcionando com monitor com cabo de energia USB somente fora dele, devido ao efeito da estática.	1000	Não	Editar Descartar
23001104	10572	Microcomputador	Elgin Rematch Newera E3 Nano	Celeron N5095	8GB DDR4	11 Pro	SSD Sata 240GB	Nenhuma, pois é nova.	Nenhuma, pois é nova.	2	Sim	Editar Descartar
23004887	10573	Microcomputador	Elgin Rematch Newera E3 Nano	Celeron N5095	8GB DDR4	11 Pro	SSD Sata 120GB	Nenhuma, pois é nova.	Nenhuma, pois é nova.	2	Sim	Editar Descartar
7158G22	1123	Minicomputador	Dell OptiPlex 3020	Pentium G3240	8GB DDR3	11 Pro	SSD 120GB	Nenhuma realizada.	11/14/2025: assoprado positivo; passado limpa contato em todos circuitos; trocado a fonte queimada de modelo parecido, mas não coube 100%.	1	Sim	Editar Descartar
8KQBY02	10507	Notebook	Dell Inspiron 3421	i3 3217U	8GB DDR3	11 Pro	SSD Sata 240GB	4/11/2025: formatado; usuário Suporte; BitDefender.	4/11/2025: limpeza com álcool isopropílico e limpa contato. Teclado integrado não funciona.	34	Sim	Editar Descartar
2467	2467	Minicomputador	Dell Vostro 260s	Pentium G630	4GB DDR3	11 Pro	SSD Sata 120GB	31/10/2025: formatado; configurado para melhorar desempenho; Suporte; BitDefender.	31/10/2025: limpa com limpa contato todos os circuitos e trocado pasta térmica da CPU.	34	Não	Editar Descartar
N0MEVAL014448E51C9501	45	Notebook	Acer Aspire E1-572-G, BR200	i3 4030U	8GB DDR4	11 Pro	SSD Sata 120GB	03/11/2025: formatado; otimizado; configurado para conexão.	03/11/2025: limpa com limpa contato nas entradas e álcool isopropílico na carcaça.	999	Sim	Editar Descartar
20104297	9179	Microcomputador	Elgin Rematch RC 8400 ZICN	Celeron N5095	8GB DDR4	11 Pro	SSD Sata 240GB	29/10/2025: formatado.	29/10/2025: passado limpa contato na placa e demais componentes.	47	Não	Editar Descartar
9GVLP22	10	Notebook	Dell Inspiron 15 7000	i7 8550U	8GB DDR4	11 Pro	SSD NVme 120GB + SSD Sata 1TB	26/10/2025: formatado; usuário Suporte.	26/10/2025: limpeza externa com álcool isopropílico e limpa contato.	999	Sim	Editar Descartar
4010310	10123	Microcomputador	Elgin Rematch	Celeron	8GB DDR4	11 Pro	SSD 120GB	27/10/2025: configuração padrão para backup; usuário Suporte.	27/10/2025: desconectado o fan, que, com passar do	63	Sim	Editar Descartar

Mostrando de 1 até 10 de 138 registros

Anterior 1 2 3 4 5 ... 14 Próximo

Figura 8. Tela de listagem de cadastros de computador

Contudo, nota-se uma presença maior de atributos daquele comparado a esses, justamente por deter de maiores informações a serem extraídas para serem guardadas, modificadas, consultadas e analisadas. Ressalta-se que a entidade de lojas e envios possuem a mesma dinâmica de listagem, mas com as devidas alterações em relação aos seus próprios atributos dipostos.

5.4. Tela de Cadastro: Computador

O primeiro parágrafo da subseção anterior, em relação às semelhanças de configurações entre as entidades irmãs na listagem, também se encaixa diante do cadastro.

A página de editar da entidade computador e de ambas outras entidades assemelha-se a de cadastro, com exceção do atributo não oculto denominado 'Envio', no qual usuário seleciona 'Não', na lista suspensa, para quando, caso o equipamento tenha sido enviado e retornado, for permanecer em reserva por tempo indeterminado. Tendo isso em vista, o equipamento, no momento em que é inserido na entidade envio, aciona, por padrão, um gatilho que altera automaticamente o valor de 'Envio' para 'Sim', não necessitando de inserí-lo manualmente.

Seguiu-se as boas práticas de validação de dados, sendo estes campos necessários e validados na camada *frontend* e *backend*: número de série, placa, modelo, tipo e última localização. E a experiência do usuário facilitada para quando o usuário for inserir o número de série: ter a opção de digitar manualmente, bipar usando o leitor ou escaner por meio de câmera do dispositivo (*desktop* ou *smartphone*).

Para a maioria dos demais atributos, o navegador sugere, via *tag* datalist, a partir da digitação, valores a serem inseridos que já foram inseridos na coluna correspondente, os quais podem ser autocompletados e editados. Agiliza-se, assim, a inserção de dados.

A seguir, a Figura 9, retrata-se visualmente algumas dessas questões abordadas.

Controle de Equipamentos de TI da RFN

EquipamentosEnviosLojasDashboard BIPlanilhasInstruções

Sair

Cadastro de computador:

Número de série:

Digite, bipe ou escaneie o número de série do computador

Placa patrimonial:

Digite, bipe ou escaneie a placa de patrimônio do computador

Tipo:

Digite o tipo do computador

Modelo:

Digite o modelo do computador

Processador:

Digite o processador do computador

Memória:

Digite a memória RAM do computador

Windows:

Digite o Windows do computador

Armazenamento:

Digite o armazenamento do computador

Manutenção de software:

Digite a data e as informações sobre manutenção de software

Manutenção de hardware:

Digite a data e as informações da manutenção de hardware

Última localização:

0

Cadastrar computador

Retornar para lista de computadores

Figura 9. Tela de campos de cadastro de computador

Vale mencionar que a entidade de lojas e envios, embora possuam atributos distintos, detêm do mesmo princípio apresentado de cadastramento. Diante do primeiro, há inserção do número da loja, CNPJ, gerente e número de telefone. Em relação ao segundo, o campo de destino inserido altera o de localização na entidade mãe equipamento via gatilho, e são passados também número de série, motivo e data de envio.

5.5. Tela de Listagem de Descartes: Equipamentos

Esta subseção corresponde aos equipamentos descartados pelo usuário. Como se observa, não há um botão para inserção manual, pois, ao serem descartados na listagem de alguma das três entidades, o equipamento é automaticamente registrado nesta tabela a seguir na Figura 10 por meio de um gatilho. Esse mecanismo garante precisão na rastreabilidade e elimina a necessidade de duplicar ações operacionais por parte do controlador.

Controle de Equipamentos de TI da RFN

EquipamentosEnviosLojasDashboard BIPlanilhasInstruções

Sair

Exibir 25 resultados por página

Pesquisar

Buscar registros

Nº série	Placa	Tipo	Modelo	Motivo	Última localização	Data de descarte	Descartador	Ações
15907019797	Sem placa	Estabilizador	SMS Revolution Speedy uRS 155 115	Velho ou estragado	4	2025-11-26	root@localhost	Editar
FX2SRH1	7898	Minicomputador	Dell OptiPlex 330	Velho: retirado nenhuma peça.	21	2025-11-24	root@localhost	Editar
16511177467	Sem placa	Estabilizador	SMS Revolution Speedy uSP155 115	Velho ou estragado	16	2025-11-18	root@localhost	Editar
273950306120	Sem placa	Nobreak	SMS Station II uST6008I	Estragado: queimou a placa interna, não compensando conserto. Retirado cabos, ventoinha e painel frontal.	1	2025-11-12	root@localhost	Editar
US3885E7N1316129	98	Multifuncional a laser	Brother DCP-2540DW	Estragado: não valia a pena consertar. Reaproveitado peças para arrumar a impressora de mesmo modelo da loja 42.	23	2025-10-30	Lucas S.	Editar
101000800000301864	3469	Impressora fiscal	Bematech MP-4000 TH	Estragado: já havia passado por manutenção na RedSet, trocando o cabeçote, mas a impressão continuou falhada. E, após cair do armário e quebrar, tornou-se inutilizada.	22	2025-10-22	Lucas S.	Editar
1819NFP1	7661	Minicomputador	Dell Vostro 230	Velho: não se adequa a ter 8GB de RAM, limitando-se a apenas 4GB no total. Foi reaproveitado fonte não genérica, RAMs, cooler e processador.	2	2025-10-22	Lucas S.	Editar
4073	Sem placa	Leitor de código de barras	Bematech BR-400	Estragado: mau contato no cabo.	999	2025-10-22	root@localhost	Editar
6824	6824	Desktop Mid Tower	Montada	Velho ou estragado. Reaproveitado placa mãe, processador e memória.	8	2025-10-16	root@localhost	Editar
B0147973	Sem placa	Leitor de código de barras	Bematech BR310	Velho ou estragado	999	2025-09-10	root@localhost	Editar
238DWL00503	Sem placa	Leitor de código de barras	Bematech BR-310 USB	Velho ou estragado	999	2025-09-10	root@localhost	Editar

Mostrando de 1 até 25 de 50 registros

Anterior12Próximo

Figura 10. Tela de listagem de descartes de equipamentos

Alguns atributos são preenchidos de forma automática, como motivo, data de descarte e descartador. Em relação ao motivo — que, por padrão, recebe a string 'Velho e/ou estragado' — recomenda-se que o usuário revise e especifique a causa real do descarte, garantindo maior precisão histórica e facilitando análises posteriores.

Percebe-se também que não há opção para exclusão de registros, sendo essa ação restrita ao administrador do sistema. Tal limitação ocorre porque a remoção de um descarte pode impactar dados críticos vinculados a outras entidades, o que poderia comprometer a consistência e integridade das informações armazenadas.

Além disso, essa abordagem centralizada do processo de descarte reforça a segurança operacional do sistema, pois impede manipulações indevidas e assegura que cada alteração esteja associada a um responsável identificado. Ao padronizar o fluxo e automatizar etapas sensíveis, o módulo de descartes contribui significativamente para a confiabilidade do controle patrimonial e para a conformidade com práticas de gestão de ativos corporativos.

5.6. Tela de Planilhas: Equipamentos Enviados e Descartados

As duas planilhas geradas em um único arquivo .XLSX referem-se aos de envios e descartes de equipamentos. É possível determinar o intervalo de início e fim dos dados para salvar localmente ou enviar semiautomaticamente um e-mail para um ou mais destinatários, desde que sejam separados por ponto e vírgula, conforme é visto na Figura 11 a seguir.

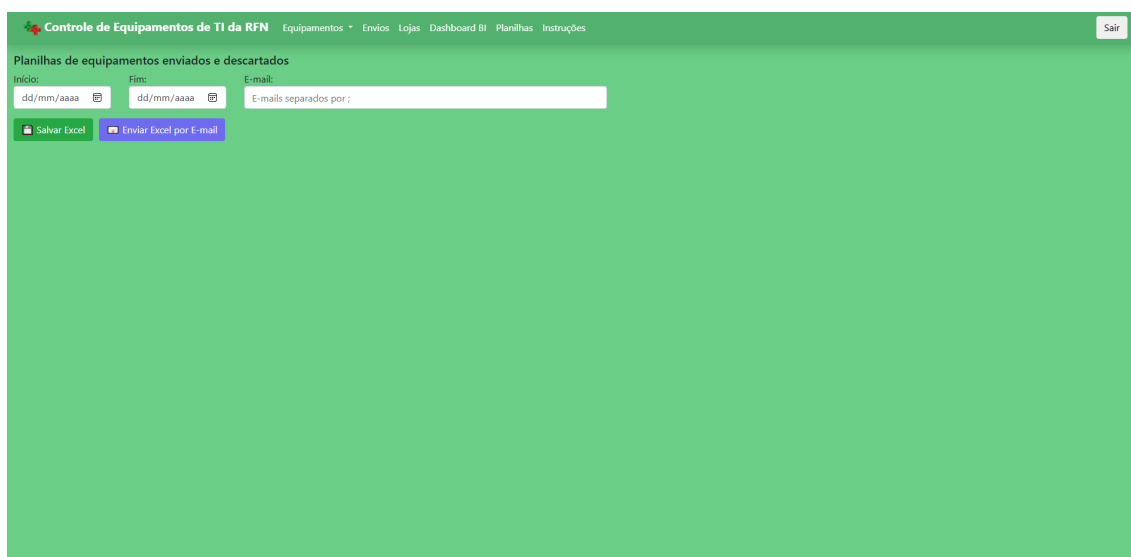


Figura 11. Tela de planilhas a serem geradas de equipamentos enviados e descartados

O respectivo e-mail do remetente está configurado, visando segurança, no *applicattion-properties* do Spring Boot. E fora predeterminada um envio mensal para os indivíduos responsáveis: Gerente de TI e demais funcionários de TI, bem como para o setor patrimonial da Contabilidade.

6. Conclusões

O *software* teve como premissa solucionar problemas agravados pela falta de gerenciamento de processos de negócios dentro do departamento de TI da empresa farmacêutica Nacional, na qual a matriz é sediada em Patrocínio - MG. Mediante a visão de reconhecê-los, padronizá-los e automatizá-los de maneira estratégica, o escopo do produto necessário para isso, e, por consequência, do projeto, tornaram o sistema passível de ser desenvolvido. E isso ocorreu devido à entrada do discente na organização na função de Auxiliar de TI, pois, caso estivesse fora desse ambiente interno, mesmo que ainda pudesse realizar entrevistas para se obter o real entendimento do modelo de negócio e dos requisitos, dificilmente obteria os resultados com tanta eficácia e eficiência.

Concomitante a isso, conforme seus conhecimentos e habilidades foram sendo agregados durante, dentro e fora do curso tecnólogo de ADS, os fragmentos que contemplam este projeto foram sendo trabalhados e remodelados para se encaixarem nas regras de negócios específicas, respeitando, axiomáticamente, os princípios e boas práticas da Engenharia de Software, de Banco de Dados, do MVC, da UX/UI etc. E, mesmo que esteja praticamente concluído, operações que visem aprimorá-lo e readequá-lo, de acordo com as iminentes necessidades, poderão ser trabalhadas.

Em razão do esforço contínuo no projeto por um longínquo tempo, com enfrentamento de diversas complexidades e adversidades, o discente está apto a fornecer o direito ao uso, em sua integridade, de maneira mensalmente remunerada, a partir do momento em que não estiver atuando na organização como CLT, mas como PJ. Pretende-se, nesse sentido, ofertá-lo mediante contrato feito pela **LM Investech — Soluções em TI e Inovação**, empresa da qual o discente é sócio.

Caso não haja êxito na proposta, o *software* será integralmente indisponibilizado, já que está patentado pela LM Investech. Nesse sentido, há a possibilidade de remodelá-lo para se especificar a regras de negócios similares de outras empresas.

Referências

- DEITEL, P. J. and DEITEL, H. M. (2020). *Internet e World Wide Web: Como Programar*. Pearson Education do Brasil, São Paulo.
- ELMASRI, R. and NAVATHE, S. B. (2018). *Sistemas de Banco de Dados*. Pearson Education do Brasil, São Paulo, 7 edition.
- FINOCCHIO JÚNIOR, J. (2013). *Project Model Canvas: como transformar ideias em projetos*. Elsevier, Rio de Janeiro.
- KUNDA, D., CHIHANA, S., and SINYINDA, M. (2017). Web server performance of apache and nginx: A systematic literature review. *Computer Engineering and Intelligent Systems*, 8(2):43–52.
- MARIADB FOUNDATION (2025). About mariadb foundation.
- MICROSOFT (2025). O que é o power bi?
- Nacional (2025). Quem somos: Farmácia nacional.
- Object Management Group (OMG) (2013). Business process model and notation (bpmn) version 2.0.2.
- ORACLE (2025). Documentação da oracle cloud infrastructure (oci).
- OSTERWALDER, A. and PIGNEUR, Y. (2010). *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Wiley, Hoboken, New Jersey.
- PROJECT MANAGEMENT INSTITUTE (2021). *Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK®)*. Project Management Institute, Inc., Newtown Square, PA, 7 edition.
- SPRING (2024). Spring boot reference documentation.
- THYMELEAF (2024). Tutorial: Using thymeleaf.
- WORLD WIDE WEB CONSORTIUM (2025). Cascading style sheets (css) - documentation.