UNIVERSIDADE SANTO AMARO

ENGENHARIA DE SOFTWARE

Ramon Ferreira da Silva

DESENVOLVIMENTO E VERIFICAÇÃO CONTROLADA DO HEFESTO PARA GESTÃO DE MANUTENÇÃO INDUSTRIAL

São Paulo

2026

Ramon Ferreira da Silva

DESENVOLVIMENTO E VERIFICAÇÃO CONTROLADA DO HEFESTO PARA GESTÃO DE MANUTENÇÃO INDUSTRIAL

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia de Software da Universidade Santo Amaro, como requisito parcial para obtenção do título de Bacharel em Engenharia de Software.

Professora da disciplina: Amanda da Silva Ferreira.

São Paulo

2026

RESUMO

Este trabalho apresenta o desenvolvimento e a verificação controlada do HEFESTO, uma aplicação web para registro e acompanhamento de manutenção industrial. O objetivo consiste em produzir uma versão utilizável e examinar a relação entre requisitos, implementação e comportamento observado. A pesquisa aplicada utiliza análise documental da trajetória de 2023 a 2026, inspeção da implementação 0.5.0, revisão local R27 de 30 de outubro de 2026 e análise dos registros de execução produzidos em 30 de outubro de 2026. O procedimento distingue testes automatizados, inspeção de código, capturas de interface e simulação com dados fictícios. A solução emprega HTML, CSS e JavaScript no navegador, serviços HTTP em Node.js e persistência em SQLite. O registro da suíte informa 203 testes aprovados e nenhuma falha. Um cenário HTTP complementar apresenta 16 verificações aprovadas, abrangendo permissões, atendimento corretivo, custo consolidado e persistência após reinício. A análise situa a aplicação web na evolução do HEFESTO, que inclui proposta de negócio, trabalhos acadêmicos, programas em terminal e protótipos visuais. Permanecem limitações de equivalência com requisitos anteriores e de avaliação em condições reais. Não foram demonstradas integração física com sensores, previsão de falhas, redução de custos ou avaliação formal de usabilidade. Conclui-se que a versão sustenta o fluxo examinado em ambiente controlado e oferece uma base técnica para avaliações posteriores com usuários e em operação industrial.

Palavras-chave: manutenção industrial; aplicação web; engenharia de software; testes automatizados; rastreabilidade.

ABSTRACT

This study presents the development and controlled verification of HEFESTO, a web application for recording and monitoring industrial maintenance. Its objective is to deliver a usable version and examine the relationship between requirements, implementation and observed behavior. The applied research combines documentary analysis of the project history from 2023 to 2026, inspection of software version 0.5.0, local revision R27 dated October 30, 2026, and analysis of execution records produced on October 30, 2026. The method distinguishes automated tests, code inspection, interface screenshots and simulation using fictitious data. The application uses HTML, CSS and JavaScript in the browser, Node.js HTTP services and SQLite persistence. The test suite record reports 203 passing tests and no failures. An additional HTTP scenario contains 16 successful checkpoints covering permissions, corrective maintenance, consolidated cost and persistence after restart. The findings place the web application within the evolution of HEFESTO, including its business proposal, academic work, terminal programs and visual prototypes. Limitations remain regarding equivalence to earlier requirements and evaluation under real operating conditions. Physical sensor integration, failure prediction, cost reduction and formal usability evaluation have not been demonstrated. The examined version supports the tested workflow in a controlled environment and provides a technical basis for subsequent user and industrial evaluation.

Keywords: industrial maintenance; web application; software engineering; automated testing; traceability.

LISTA DE ABREVIATURAS E SIGLAS

API – Interface de programação de aplicações

CLI – Interface de linha de comando

CSRF – Cross-site request forgery

CSS – Cascading Style Sheets

GPS – Global Positioning System

HTML – HyperText Markup Language

HTTP – Hypertext Transfer Protocol

HTTPS – HTTP sobre conexão protegida por TLS

IA – Inteligência artificial

JSON – JavaScript Object Notation

MFA – Autenticação multifator

PWA – Progressive Web App

RF – Requisito funcional

RNF – Requisito não funcional

SHA-256 – Função de hash de 256 bits

TCC – Trabalho de Conclusão de Curso

TLS – Transport Layer Security

UI/UX – Interface do usuário e experiência do usuário

SUMÁRIO

1 INTRODUÇÃO.....6
2 OBJETIVOS.....8
2.1 Objetivo geral.....8
2.2 Objetivos específicos.....8
3 METODOLOGIA.....9
3.1 Seleção das fontes e formação do corpus.....11
3.2 Identidade da versão e integridade.....11
3.3 Procedimento do cenário complementar.....12
3.4 Estratégia de interpretação e ameaças à validade.....13
4 DESENVOLVIMENTO.....14
4.1 Gestão de manutenção e qualidade do registro.....14
4.2 Requisitos e rastreabilidade da evolução.....15
4.3 Construção do artefato e limites da avaliação.....15
4.4 Interação e acessibilidade.....16
4.5 Persistência e segurança de aplicações web.....17
4.6 Implementação da versão de referência.....18
4.7 Organização do estado e controle de concorrência.....21
4.8 Perfis e limites de compartilhamento.....22
4.9 Inventário e ciclo do atendimento.....23
4.10 Almoxarifado e informação financeira.....24
4.11 Interface e distribuição.....25
5 RESULTADOS E DISCUSSÃO.....27
5.1 Leitura conjunta dos resultados.....31
5.2 Histórico do projeto e alcance das mudanças.....32
5.3 Limitações e implicações para o uso.....34
5.4 Proposta de continuidade verificável.....35
6 CONCLUSÃO.....36
REFERÊNCIAS.....37
APÊNDICE A RASTREABILIDADE DOS REQUISITOS.....40
APÊNDICE B IDENTIFICAÇÃO E REPRODUÇÃO.....43
APÊNDICE C PROTOCOLO PROPOSTO DE AVALIAÇÃO.....44

1 INTRODUÇÃO

A gestão de manutenção reúne informações sobre equipamentos, ocorrências, responsáveis e intervenções. Registrar uma ocorrência isoladamente não assegura que sua autorização, execução e conclusão possam ser acompanhadas de maneira consistente. No domínio tratado pelo HEFESTO, o problema de engenharia consiste em relacionar essas etapas em um sistema utilizável, preservando o vínculo com a máquina e limitando as ações e os dados disponíveis a cada perfil. O polo moveleiro de Linhares, Espírito Santo, constitui o contexto pretendido de aplicação. Este estudo, contudo, não utiliza observações de uma fábrica como resultados experimentais.

A literatura técnica distingue manutenção corretiva, preventiva e preditiva como abordagens com diferentes bases para decidir a intervenção (Pacific Northwest National Laboratory, [s.d.]). Essa distinção importa para o software: cadastrar uma manutenção como preditiva não demonstra capacidade de prever falhas. O presente recorte concentra-se na gestão dos registros e do atendimento. Sensores, aquisição física de grandezas e modelos de prognóstico permanecem como possibilidades de expansão, sem resultados de campo atribuídos à versão estudada.

A trajetória do HEFESTO antecede os projetos acadêmicos de 2025. O formulário HEFESTO Works, preparado no contexto do Espírito Hub, declara 13 de maio de 2023 como data de fundação do projeto e apresenta uma proposta de software para gestão de manutenção. Os materiais de apresentação de 2024 enfatizam marcenarias e pequenas e médias fábricas de móveis. Esses registros documentam a concepção e o contexto de negócio, sem demonstrar a execução de todas as capacidades anunciadas (Hefesto Works, 2023; Hefesto, 2024).

O trabalho Hefesto Simplificado, de 2024, descreve um programa em Python com interface CMD. Em 2025, o estudo de Laboratório de Programação documenta a evolução com SQLite, e o código preservado identifica HEFESTO CMD v0.1.0-beta. Os projetos de Sistemas e de UI/UX organizam requisitos e analisam a interação no terminal, propondo interfaces gráficas (Silva, 2024, 2025a, 2025b, 2025c, 2025d). A contribuição examinada em 2026 é a evolução para uma

implementação web e sua verificação controlada. Os resultados atuais permanecem vinculados à versão 0.5.0/R27-textos-local (Silva, 2026c).

A pergunta de pesquisa é: como implementar e verificar, em cenários controlados, uma aplicação web que vincule equipamentos, papéis de acesso e etapas do atendimento de manutenção, preservando a consistência do histórico e dos custos? A contribuição pretendida é um artefato utilizável acompanhado de evidências rastreáveis. A relação entre construção e avaliação do artefato aproxima o estudo da perspectiva de pesquisa em design science descrita por Hevner et al. (2004), sem caracterizar um experimento sobre desempenho humano ou organizacional.

O recorte temporal encerra-se nos registros de 30 de outubro de 2026. A unidade de análise é a versão 0.5.0 e os cenários documentados, não toda a operação futura do produto. A justificativa técnica está na possibilidade de examinar, em conjunto, regras de negócio, autorização e persistência. O trabalho não estima redução de paradas ou retorno financeiro, pois não dispõe de linha de base, acompanhamento longitudinal ou comparação entre condições de operação.

2 OBJETIVOS

2.1 Objetivo geral

Desenvolver uma versão utilizável do HEFESTO para gestão de manutenção industrial e verificar, em cenários controlados, o atendimento de comportamentos selecionados de acesso, registro, acompanhamento e persistência.

2.2 Objetivos específicos

Os objetivos específicos são: relacionar os requisitos históricos à implementação atual, registrando alterações de escopo; descrever a arquitetura e os mecanismos que sustentam os fluxos; examinar os registros de testes automatizados e do cenário HTTP complementar; confrontar os resultados com os critérios de aceitação do cenário; e delimitar os comportamentos comprovados e as avaliações necessárias para o uso em condições reais.

A avaliação concentra-se nas propriedades explicitadas no cenário de referência. O objetivo de produzir um aplicativo utilizável é preservado, enquanto aceitação humana, integração com sensores e ganhos operacionais são tratados como etapas posteriores. Essa delimitação evita prometer uma medida que não foi coletada.

3 METODOLOGIA

A pesquisa é aplicada e orientada ao desenvolvimento de software, com análise documental e verificação técnica em ambiente controlado. O percurso reúne análise dos artefatos anteriores, evolução da aplicação, inspeção da versão de referência e análise dos registros de execução. Não se realizou pesquisa de campo, experimento com participantes ou avaliação formal de usabilidade. O desenvolvimento de uma versão utilizável integra o escopo definido para esta pesquisa.

O corpus reúne o formulário HEFESTO Works de 2023, materiais de apresentação de 2024, o trabalho de programação de 2024 e os três trabalhos de 2025, além dos fontes Python preservados, do manuscrito preliminar de 25 de outubro e do pacote técnico de 30 de outubro de 2026. O handoff de 23 de outubro orienta a localização do acervo histórico. Datas declaradas nos documentos, datas de modificação e datas de cópia são tratadas separadamente. A identificação histórica não transfere para as versões antigas os resultados dos ensaios atuais (Hefesto Works, 2023; Hefesto, 2024; Silva, 2024, 2025a, 2025b, 2025c, 2025d, 2026a, 2026b).

O pacote de referência identifica hefesto-interface 0.5.0, revisão R27-textos-local, concluída em 30 de outubro de 2026 e não publicada. Foram examinados arquivos da aplicação, catálogo demonstrativo, logs, relatórios JSON e capturas. O acervo anterior conserva código e matrizes da R25; o registro de publicação R26 está separado. Versão do software, revisão local e publicação são identificações distintas (Silva, 2026b, 2026c).

Os resultados de execução apresentados neste trabalho provêm dos registros preservados no pacote técnico, produzidos com Node.js v24.19.0 em Windows em 30 de outubro. A preparação dos manuscritos compreendeu inspeção desses registros e conferência da integridade do pacote; não corresponde a uma segunda execução independente da suíte. O SWEBOK distingue testes, análise estática e avaliação em diferentes níveis, distinção utilizada para evitar que a presença de código seja tomada como prova de comportamento em todos os contextos (Washizaki, 2026).

O cenário complementar usa servidor HTTP local, banco SQLite temporário, empresas e pessoas fictícias, transporte de e-mail interceptado e coordenadas simuladas. A preparação do ensaio desativa MFA apenas nesse servidor. São verificadas restrições de perfil e empresa, bloqueios de transição, encerramento sem duplicação e recuperação após reinício. As capturas da interface real foram coletadas separadamente do roteiro HTTP; não representam avaliação com participantes.

Para interpretar o estágio de cada capacidade, adotaram-se cinco categorias: especificada, quando descrita como requisito; desenhada ou prototipada, quando representada em interface ou fluxo; implementada, quando identificada no código executável; testada, quando vinculada a procedimento, resultado e ambiente registrados; e planejada, quando depende de execução futura. Uma funcionalidade pode acumular categorias. “Testada” deve sempre indicar o tipo e o alcance do teste, sem significar homologação integral.

O Quadro 1 explicita a classificação aplicada às capacidades do projeto.

Quadro 1 – Estágios do projeto e interpretação da evidência

Estágio	Objeto	Interpretação
Especificado	Requisitos RF01–RF25 e RNF01–RNF05	Formulação histórica preservada
Desenhado ou prototipado	Design documentado em 2024 e estudo de UI/UX de 2025	Não comprova execução da aplicação
Implementado	Serviços HTTP, domínio e interface 0.5.0	Inspeção permite identificar mecanismos
Testado	Suíte e cenário HTTP de 30/10	Conclusão limitada aos casos registrados
Planejado	Avaliação humana, piloto e sensores	Procedimento proposto, ainda sem resultado

Fonte: elaboração do autor com base no acervo histórico e técnico do HEFESTO (Hefesto, 2024; Silva, 2025a, 2026b).

Os critérios definidos pelo roteiro incluem: impedir alteração de inventário pelo operador; isolar chamados por empresa; bloquear início sem aprovação; retirar valores da resposta ao operador; exigir conferência antes do encerramento; produzir um único histórico com custo de 29.000 centavos; e recuperar os registros após reinício. A revisão R27 acrescenta a verificação de textos em UTF-8 no serviço e na

persistência. A análise descritiva confronta logs, arquivos e imagens, sem inferência estatística.

3.1 Seleção das fontes e formação do corpus

A fundamentação foi selecionada por pertinência ao problema: gestão de manutenção, construção e avaliação de artefatos, engenharia de requisitos, testes, interação, persistência e segurança. Foram priorizadas publicações dos próprios autores, entidades técnicas e documentação oficial. A consulta não constitui revisão sistemática da literatura e não emprega busca exaustiva, protocolo de inclusão com dupla revisão ou síntese quantitativa de estudos.

O SWEBOK consultado corresponde à versão 4.0a, disponibilizada em agosto de 2026. Foram lidas seções sobre requisitos, rastreabilidade e testes. No caso de Hevner et al., das duas normas ISO e do SSDF, a consulta utilizou os resumos públicos identificados nas referências. Não se atribuem a esses acessos detalhes de métodos ou cláusulas não consultadas. Documentação web sem data explícita é identificada como sem data e acompanhada da data de acesso.

Os documentos do autor foram empregados como fontes primárias do histórico. A pesquisa não converte a nota de um projeto anterior em evidência de qualidade técnica atual. O corpus permite identificar o conteúdo dos requisitos e a trajetória dos artefatos, mas não permite reconstruir toda decisão de implementação ou todo teste realizado fora dos arquivos preservados. O pacote de 30 de outubro define a fronteira de observação adotada.

3.2 Identidade da versão e integridade

A identificação reúne a versão 0.5.0, a revisão R27-textos-local e a data dos registros de 30 de outubro de 2026. A R27 conserva os cabeçalhos e a posição da logo da R26 e não foi publicada. O registro de publicação R26, às 11:55 BRT, permanece separado. O número de revisão não é uma versão principal do software; a data de redação também não altera a data de execução dos testes.

Os 447 arquivos listados no manifesto do pacote R27 foram conferidos por tamanho e SHA-256, sem divergências. O acervo anterior, com 384 arquivos, e os documentos acadêmicos R3 foram preservados por hash no handoff. A integridade

estabelece correspondência com o pacote recebido, mas não demonstra, sozinha, autoria, independência ou inviolabilidade anterior dos registros.

Os arquivos de evidência foram preservados como fonte e não foram substituídos por resultados de uma nova execução. Isso permite distinguir a análise atual dos eventos relatados. Uma reprodução futura deve utilizar pasta de saída separada, registrar ambiente e manter o retrato original. O Apêndice B descreve os elementos necessários para realizar essa reprodução sem utilizar dados privados.

3.3 Procedimento do cenário complementar

O roteiro prepara uma administração isolada, cadastra uma empresa por HTTP e confirma o desafio de e-mail por transporte interceptado. Em seguida, cadastra a máquina CL01, cria convites e ativa os perfis utilizados. A preparação automatizada não representa onboarding observado com usuários, nem comprova que as mensagens chegaram a caixas postais reais. Ela estabelece as precondições do ensaio funcional.

Após a abertura do chamado, o roteiro verifica a visibilidade adequada ao gestor e a ausência do chamado no contexto de outra empresa. O orçamento precisa de aprovação antes da execução. A resposta ao operador omite valores financeiros. O deslocamento usa coordenadas sintéticas e, ao iniciar o trabalho, a posição compartilhada é removida. Nenhuma dessas etapas utiliza deslocamento físico de mantenedor.

O fechamento é tentado antes e depois da conferência. O teste exige bloqueio na primeira condição e histórico único na segunda. Uma nova tentativa de encerrar não pode duplicar o lançamento. Por fim, o servidor é reiniciado e ocorre nova autenticação antes da leitura dos registros. O critério examina persistência dos dados, sem exigir preservação da mesma sessão de navegador após o reinício.

A R27 corrige dados textuais do antigo roteiro de captura mediante a fixture tests/fixtures/tcc-scenario.mjs, além de textos da interface e de exportação. A auditoria reporta 189 arquivos, 17.906 entradas inventariadas e nenhuma ocorrência remanescente de codificação inválida no escopo examinado. Isso não equivale a cobertura linguística de todos os estados possíveis. Registros de tentativas

anteriores foram preservados; a expectativa de reais versus centavos corrigida na R25 não é apresentada como falha atual da R27 (Silva, 2026b, 2026c).

3.4 Estratégia de interpretação e ameaças à validade

A consistência dos resultados foi examinada por convergência entre a asserção do roteiro, o estado persistido relatado e a apresentação visual. Essa comparação não equivale a revisão independente, pois código, testes e registros pertencem ao mesmo projeto. Uma falha compartilhada entre regra e expectativa pode permanecer invisível. A revisão externa de cenários é, portanto, uma ampliação necessária da confiança.

A validade de construção é limitada pela distância entre os indicadores disponíveis e as propriedades desejadas. Contagem de testes não mede usabilidade; duração de suíte não mede desempenho interativo; igualdade de arquivos públicos não mede identidade do backend; e uma resposta HTTPS não mede todo o controle de acesso. Cada indicador foi vinculado apenas à propriedade que efetivamente observa.

A validade externa é limitada pelo uso de base temporária, dados fictícios e ambientes selecionados. Não há amostra de fábricas, turnos, profissionais ou dispositivos. A validade temporal também é delimitada: publicações posteriores podem modificar o produto. As conclusões não são generalizadas para todo o ciclo de vida do HEFESTO ou para a população de empresas industriais.

4 DESENVOLVIMENTO

4.1 Gestão de manutenção e qualidade do registro

No HEFESTO, a unidade que organiza o trabalho operacional é o atendimento associado a uma máquina. Uma mesma ocorrência pode gerar decisões distintas: autorização de gasto, designação de responsável, consumo de material e conferência da execução. O desenho do sistema precisa permitir reconstruir essas decisões, sem exigir que todos os perfis recebam os mesmos dados. A questão central não é apenas armazenar um texto descritivo, mas preservar relações e estados que possam ser consultados posteriormente.

A distinção entre intervenções corretivas, preventivas e preditivas oferece vocabulário para o domínio. Na classificação discutida pelo Pacific Northwest National Laboratory ([s.d.]), ações corretivas respondem a falhas, preventivas seguem programação e preditivas dependem da condição acompanhada. No software, os rótulos servem à organização dos registros. A comprovação de um método preditivo exigiria fontes de dados e avaliação que ultrapassam esse uso classificatório.

A presença de um histórico pode apoiar investigações futuras, mas não garante a qualidade dos dados que o alimentam. Descrições incompletas, seleção incorreta de máquina e ausência de conferência técnica podem produzir registros formalmente válidos e operacionalmente pouco úteis. Por isso, o projeto diferencia levantamento e conferência e conserva responsabilidades ao longo do chamado. A utilidade dessas distinções deve ser avaliada com pessoas que conheçam os equipamentos e o processo de trabalho.

O contexto moveleiro orienta nomes, exemplos e hipóteses de aplicação. Não se atribuem à Cimol ou a outra fábrica os dados da empresa fictícia utilizada no ensaio. A possibilidade de um piloto constitui continuidade, não amostra observada. Essa delimitação protege a interpretação dos custos exibidos: o painel soma intervenções cadastradas, sem estimar retorno de investimento ou estabelecer uma linha de base de falhas.

4.2 Requisitos e rastreabilidade da evolução

Um requisito histórico precisa conservar sua formulação para que mudanças possam ser discutidas. O SWEBOK relaciona rastreabilidade à consistência entre requisitos e produtos de trabalho e à análise de mudanças (Washizaki, 2026). Neste projeto, a aplicação dessa ideia é documental: cada RF ou RNF é ligado a implementação, evidência e limitação. Não se interpreta a existência dessa ligação como certificação automática de atendimento.

A mudança de terminal para navegador altera tanto a tecnologia quanto o contrato de uso. O antigo instalador Windows respondia a uma necessidade de distribuição local; o acesso web responde a outra forma de disponibilização. A substituição pode ser desejável, mas precisa aparecer como mudança de escopo. A mesma regra vale para excluir ou arquivar máquinas: preservar histórico pode justificar o arquivamento, embora ele não corresponda à exclusão física prevista anteriormente.

O significado de “administrador” também se modifica. No ambiente atual, a administração da plataforma possui responsabilidades distintas das do gestor da empresa. A revisão evita transportar permissões antigas para um papel apenas porque os nomes parecem equivalentes. A aceitação futura deve registrar quem pode alterar cada recurso, em qual empresa e em qual estado, considerando ainda vínculos com equipamentos e designações específicas.

A matriz não calcula uma porcentagem única de requisitos atendidos. Há itens compostos, mudanças contratuais e evidências de forças diferentes. Transformá-los em uma soma binária ocultaria, por exemplo, a diferença entre cadastro básico verificado e upload de fotografia real ainda não exercitado no cenário complementar. A análise preserva essa granularidade e indica a evidência adicional necessária para cada conclusão.

4.3 Construção do artefato e limites da avaliação

Hevner et al. (2004) associam a perspectiva de design science à construção e aplicação de artefatos para compreender problemas e soluções. O trabalho utiliza essa orientação em sentido delimitado: a contribuição está no software e na análise de seu comportamento documentado. Não se afirma a execução integral de um

protocolo externo de design science nem a descoberta de uma teoria causal sobre manutenção industrial.

O resumo público da ISO/IEC 25010 descreve um modelo de qualidade de produto que pode apoiar especificação, medição e avaliação (International Organization for Standardization, 2023). Aqui, a norma é referência para distinguir dimensões de qualidade; não foi aplicado um instrumento completo de conformidade. A correção observada em um fluxo, a organização do código e a experiência de uso permanecem questões diferentes.

A estratégia de avaliação reúne evidências com alcances complementares. A análise do código identifica como uma regra foi construída; o teste automatizado registra o comportamento diante de entradas escolhidas; a captura mostra uma condição visual; e uma avaliação com participantes observaria a realização de tarefas por pessoas. A ausência desta última não impede descrever os resultados técnicos, mas impede concluir que a interface é intuitiva ou adequada a toda rotina de trabalho.

4.4 Interação e acessibilidade

As heurísticas de Nielsen (1994) oferecem princípios gerais, como visibilidade de estado, consistência e prevenção de erros. O estudo de UI/UX de 2025 utilizou a análise de fluxos como antecedente da proposta gráfica (Silva, 2025a). Na evolução atual, a separação entre chamado aguardando aprovação e chamado encerrado oferece um exemplo concreto de estados que devem ser compreendidos. A presença do rótulo não comprova que o usuário o interpreta corretamente.

A ISO 9241-11 trata usabilidade como resultado do uso (International Organization for Standardization, 2018). Por isso, o trabalho distingue intenção de design e resultado observado. A leitura da página pública da norma fundamenta essa distinção conceitual; não substitui o texto integral nem autoriza declarar que a aplicação foi avaliada segundo todos os seus procedimentos ou conceitos.

As WCAG 2.2 apresentam critérios verificáveis de acessibilidade aplicáveis a diferentes dispositivos (World Wide Web Consortium, 2024). As capturas existentes não constituem auditoria desses critérios. Uma avaliação posterior deverá incluir navegação por teclado, foco, contraste, alternativas textuais, mensagens de erro e

ampliação, com registros por critério. Esses itens são uma proposta de continuidade e não resultados positivos presumidos.

4.5 Persistência e segurança de aplicações web

A documentação de HTTP do Node.js descreve a API utilizada para criar servidores e lidar com mensagens HTTP (Node.js, 2026a). No HEFESTO, o código do servidor combina roteamento, respostas e cabeçalhos com chamadas aos serviços do domínio. A camada HTTP é apenas uma parte da arquitetura; a verificação do estado e da autorização precisa ocorrer junto às operações que alteram os dados.

A classe DatabaseSync executa operações de forma síncrona (Node.js, 2026b). Em conjunto com transações, essa implementação permite examinar a unidade de persistência do cenário local. A documentação do SQLite explica a atomicidade de transações e suas condições de operação (SQLite, [s.d.]). Não se transfere essa propriedade geral para uma garantia irrestrita da implantação: configuração de armazenamento, backup e recuperação precisam de ensaios próprios.

Autenticação e autorização resolvem problemas diferentes. A primeira estabelece uma identidade na sessão; a segunda decide quais ações e recursos podem ser acessados. A OWASP recomenda privilégio mínimo e validação de permissões em cada requisição (OWASP Foundation, [s.d.]a). Esse princípio orienta a análise dos bloqueios do cenário, incluindo a tentativa de um operador cadastrar equipamento e o isolamento do chamado em relação a outra empresa.

A RFC 7914 especifica scrypt como função de derivação baseada em senha com custo de memória (Percival; Josefsson, 2016). A orientação de armazenamento da OWASP apresenta parâmetros para seu uso (OWASP Foundation, [s.d.]c). O estudo registra os parâmetros do código e a existência dos testes, sem transformar essa comparação pontual em aprovação do sistema de contas completo.

A prevenção de requisições forjadas exige avaliar mais de um mecanismo. A orientação da OWASP discute tokens e limites de atributos como SameSite (OWASP Foundation, [s.d.]b). O código do HEFESTO combina token de sessão, verificação de origem e proteção CSRF. A configuração efetiva do backend remoto

não foi auditada integralmente neste corpus; a conclusão continua limitada ao código e aos ensaios preservados.

O SSDF organiza práticas de desenvolvimento seguro que podem ser incorporadas ao ciclo de vida (Souppaya; Scarfone; Dodson, 2022). Sua utilização conceitual neste estudo reforça que segurança requer continuidade de revisão e tratamento de vulnerabilidades. O teste aprovado é um registro localizado no tempo, e não uma declaração de ausência permanente de risco.

4.6 Implementação da versão de referência

A aplicação utiliza HTML, CSS e JavaScript no navegador. A interface e o domínio são organizados em módulos, com rotinas próprias de construção e empacotamento. O servidor emprega node:http e serviços de contas, autorização e operações do domínio. A persistência utiliza node:sqlite e DatabaseSync. A documentação dessa classe esclarece que suas operações são síncronas (Node.js, 2026b). Essa escolha simplifica o acesso local ao banco, mas torna necessária uma avaliação específica de concorrência antes de extrapolar capacidade de atendimento.

O modelo de persistência atual combina tabelas relacionais e documentos JSON de estado armazenados em colunas, com revisões para detectar conflitos. Não corresponde ao esquema de oito tabelas do protótipo Python anterior. A implementação inspecionada utiliza transações para agrupar operações; a noção de atomicidade em SQLite refere-se à aplicação conjunta das alterações ou à ausência delas (SQLite, [s.d.]). O teste de reinício examina uma condição particular de persistência, sem substituir ensaios de falha de energia, corrupção ou restauração de backup.

O Quadro 2 organiza as responsabilidades que compõem o caminho entre navegador e persistência.

Quadro 2 – Camadas e responsabilidades da arquitetura

Camada	Elementos	Responsabilidade
Navegador	HTML, CSS e módulos JavaScript	Entrada e apresentação das operações
API HTTP	accounts-server.mjs	Sessão, requisições e respostas
Serviços	accounts-service.mjs e suite-service.mjs	Permissões e regras de operação
Persistência	SQLite e documentos JSON	Estado, revisões e auditoria

Fonte: elaboração do autor com base no acervo técnico do HEFESTO (Silva, 2026b).

O inventário relaciona a máquina à empresa e diferencia levantamento inicial de conferência técnica. A máquina fictícia CL01 foi cadastrada em levantamento, sem componentes conferidos. Assim, sua ausência em um indicador restrito a equipamentos conferidos não representa perda do registro. A aplicação oferece identificação, fabricante, modelo, localização, imagens, componentes e importação CSV; o cenário adicional verificou o cadastro básico, enquanto os demais campos dependem de seus testes específicos e de aceite posterior.

O atendimento conecta abertura, atribuição, aceite, orçamento, aprovação, execução, conferência e encerramento. As ações disponíveis dependem do perfil e do estado do chamado. O cenário selecionado exercitou um atendimento corretivo com preço fechado e separou a conferência pelo operador do encerramento pelo gestor. A dupla tentativa de fechamento foi verificada para evitar um segundo histórico. Esse comportamento interessa à rastreabilidade porque o registro final reúne máquina, responsável e custo sem tratar cada clique como nova intervenção.

A autorização é aplicada no servidor e inclui o escopo da empresa. A projeção das respostas por perfil restringe os dados enviados ao navegador, em vez de depender apenas da ocultação de elementos visuais. Essa decisão é compatível com a recomendação da OWASP de verificar permissões a cada requisição (OWASP Foundation, [s.d.]a). O ensaio demonstra casos particulares de bloqueio e isolamento; não permite afirmar que todas as combinações de recursos, papéis e estados foram auditadas.

As senhas utilizam scrypt com sal aleatório e comparação de tempo constante. O código inspecionado configura N = 32.768, r = 8 e p = 3, combinação listada entre as opções da orientação de armazenamento de senhas da OWASP

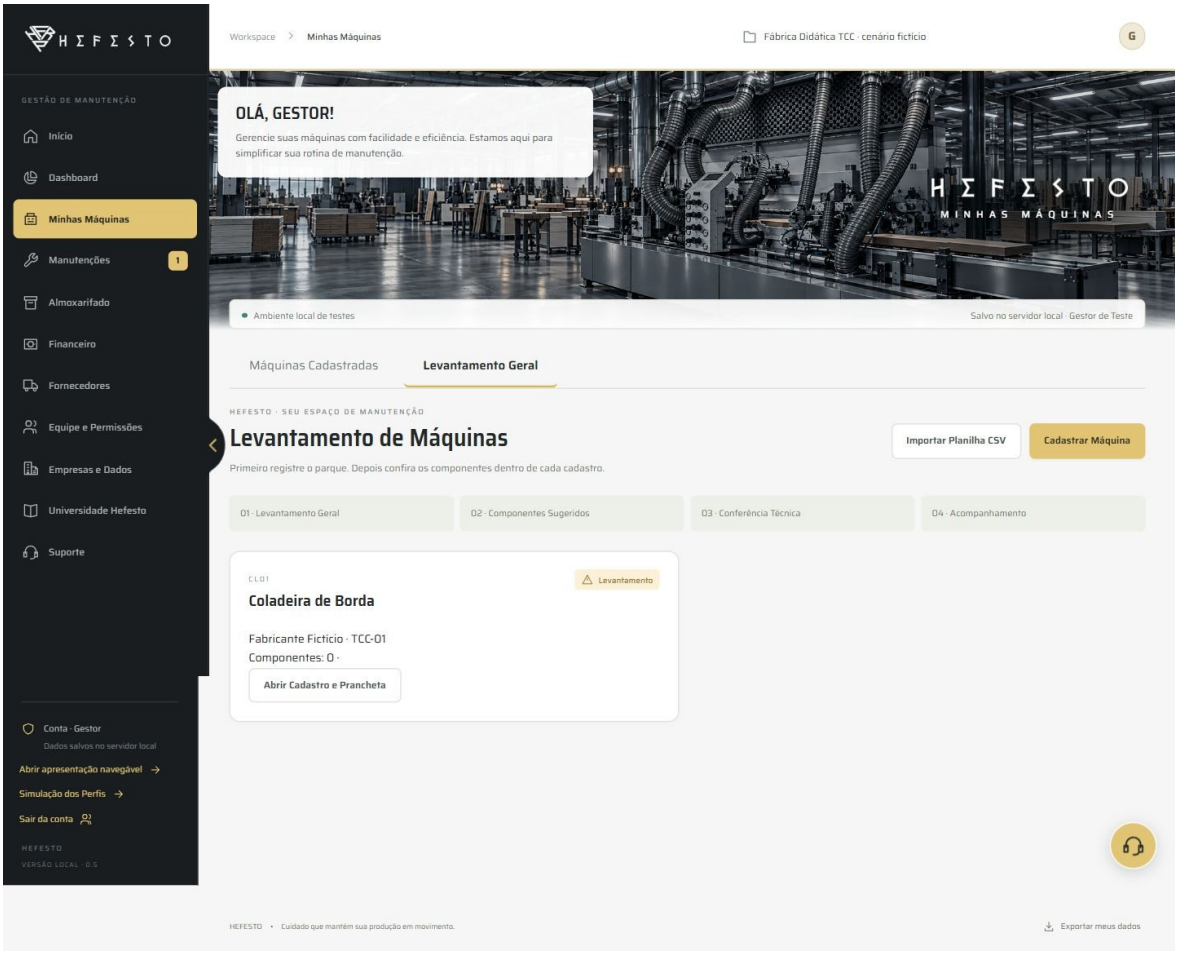
(OWASP Foundation, [s.d.]c). Há mecanismos de sessão, proteção contra requisições indevidas, limitação de tentativas e MFA administrativo. A existência desses controles e de seus testes não equivale a auditoria independente de segurança ou certificação da aplicação.

A identidade visual utiliza Saira no corpo, Saira Semi Condensed em títulos, base grafite e acentos dourados. A proposta de interação prioriza identificação de estado, termos do domínio e feedback, aspectos relacionados às heurísticas de Nielsen (1994). As imagens demonstram a aparência da implementação em contextos específicos; não sustentam, isoladamente, afirmações de facilidade de uso. O resultado de usabilidade depende do uso e do contexto, como esclarece o resumo público da ISO 9241-11 (International Organization for Standardization, 2018).

Quatro contextos precisam permanecer separados: demonstração pública com alterações temporárias; interface de arquivo local com estado em memória; empresa autenticada com autorização e persistência no servidor; e ambiente privado de testes com fábrica sintética. O acesso de exemplo à interface local não constitui autenticação real. A demonstração permite explorar o produto, mas o resultado de persistência deste trabalho pertence ao servidor HTTP temporário. A distribuição web também altera o requisito histórico de instalador Windows, sem atendê-lo literalmente.

A Figura 1 registra a máquina fictícia CL01 na etapa de levantamento, com o cadastro executado no servidor local do ensaio.

Figura 1 – Máquina fictícia cadastrada e ainda não conferida



Fonte: acervo do projeto, R27 local, 30 out. 2026, 12:56:46 BRT (Silva, 2026c). Dados fictícios; não representa operação industrial.

4.7 Organização do estado e controle de concorrência

O código separa a entrada HTTP dos serviços de conta e das operações do domínio, embora parte do roteamento e das verificações permaneça concentrada em arquivos extensos. A modularização facilita localizar responsabilidades, mas não constitui uma medida formal de manutenibilidade. O exame não levantou complexidade ciclomática, acoplamento ou custo de mudança, nem comparou alternativas arquiteturais em experimento.

A empresa mantém um documento de workspace em JSON e um número de revisão. Antes de aceitar uma atualização, o serviço confere a revisão recebida. A ideia implementada é evitar que uma janela sobrescreva silenciosamente alterações feitas em outra. O comportamento é distinto de editar cada entidade diretamente em uma tabela independente; a aplicação precisa validar o estado agregado antes de persistir sua nova versão.

O modelo agregado reúne vantagens e limites. A leitura de um estado completo facilita a projeção por perfil e a aplicação de regras que cruzam máquinas e planos. Ao mesmo tempo, crescimento do documento e frequência de escrita podem exigir investigação de custo e concorrência. O presente estudo não mediu esse ponto de mudança nem recomenda uma migração com base apenas na preferência por outra tecnologia.

A transação observada utiliza BEGIN IMMEDIATE, COMMIT e ROLLBACK. As alterações são acompanhadas de registros de auditoria. A presença desses comandos permite descrever a intenção e a implementação de agrupamento transacional; a evidência de execução disponível cobre o caminho do ensaio. Testes de interrupção em múltiplas etapas e restauração do banco constituem uma avaliação adicional.

4.8 Perfis e limites de compartilhamento

A administração da plataforma, o gestor, o mantenedor, o operador, RH e o almoxarife possuem responsabilidades diferentes. O profissional externo recebe acesso conforme designação. O cenário complementar cobre principalmente gestor, operador e mantenedor, além da tentativa de acesso por outra empresa. Os demais perfis aparecem no código e na suíte, sem que isso represente um atendimento humano completo realizado por todos eles.

A restrição financeira do operador foi exercitada no nível da resposta do serviço. Esse detalhe é relevante: uma interface que apenas escondesse o preço ainda poderia entregar os dados na resposta. O resultado observado é mais específico, pois o roteiro verifica a projeção disponível ao perfil. A conclusão permanece local ao contrato testado e precisa ser ampliada a exportações, anexos e outras rotas em uma auditoria abrangente.

O compartilhamento de localização depende do estado do atendimento. A implementação admite coordenadas e um link externo de mapa em contextos não simulados. O ensaio comprova apenas a passagem de coordenadas fictícias e sua remoção ao iniciar o trabalho. Não se demonstraram mapa cartográfico incorporado, cálculo de trajeto rodoviário, tempo estimado de chegada ou operação em segundo plano.

O cadastro e a confirmação de e-mail também não comprovam habilitação profissional. O sistema registra perfis e vínculos, mas este trabalho não verificou diplomas, competências técnicas ou identidade de mantenedores. A autorização para realizar uma ação no software é uma política interna da aplicação e não deve ser confundida com autorização técnica para intervir em equipamento industrial.

A Figura 2 apresenta um recorte da captura real do operador em viewport de 390 pixels, destacando os textos corrigidos e a localização simulada. Não houve participação de operador real nem medição de trajeto físico.

Figura 2 – Detalhe do chamado e da localização simulada no perfil operador



Fonte: acervo do projeto, R27 local, 30 out. 2026, 12:57:38 BRT (Silva, 2026c). Dados fictícios; não representa operação industrial.

4.9 Inventário e ciclo do atendimento

O inventário distingue dados iniciais e conferência. A máquina CL01 foi intencionalmente deixada em levantamento e não recebeu componentes no cenário. Essa condição torna visível uma fronteira do fluxo: cadastrar um nome e modelo não

confirma que o equipamento foi examinado. Uma avaliação humana deverá verificar se a linguagem usada evita interpretar rascunho como cadastro tecnicamente validado.

O arquivamento preserva o histórico e impede novas operações incompatíveis com a situação do equipamento, conforme os casos automatizados documentados. A escolha atende à necessidade de conservar registros, mas altera o requisito que previa exclusão. A discussão com o responsável pelo uso deve definir quem pode arquivar, em quais condições e como tratar chamados pendentes.

No chamado, orçamento e execução são etapas diferentes. O preço fechado do ensaio não foi derivado de tempo medido; foi informado como dado fictício do cenário. As duas horas incluídas no orçamento e os 30 minutos lançados no apontamento não fornecem indicador de eficiência. A conclusão de serviço depende da sequência de estados e conferências, não de uma alegação de reparo físico realizado.

O vínculo com o histórico permite consolidar custo por equipamento e período. As evidências de financeiro e histórico representam o mesmo atendimento, evitando contá-lo como dois eventos distintos. A repetição da tentativa de encerramento oferece um critério observável de não duplicação. Não foi mensurada a qualidade de relatórios em uma base de meses de operação.

4.10 Almoxarifado e informação financeira

O almoxarifado inclui materiais, reservas, entrega, consumo, devolução e ferramentas vinculadas a chamados. A suíte contém regras relacionadas a saldo e dupla baixa. O fluxo HTTP adicional de 30 de outubro não efetuou uma contagem física de estoque. O consumo interno mostrado como zero na captura não contradiz o valor de peças cobrado no orçamento fictício: são origens de informação diferentes.

A consistência financeira exige explicitar unidades e origem dos valores. O domínio armazena centavos e a interface apresenta reais. A correção do roteiro que comparava 290 com 29.000 ilustra o risco de interpretar a representação técnica sem o contrato de unidade. O trabalho conserva a composição do valor para permitir que o leitor confira a soma sem recorrer à aparência do painel.

O módulo financeiro agrega custos de manutenção e não representa contabilidade, faturamento ou contas a pagar. Da mesma forma, as categorias comerciais Essencial, Gestão e Inteligente organizam a proposta e a demonstração, mas não comprovam homologação de faturamento ou controle integral de direitos de assinatura. Essas capacidades ficam fora da conclusão técnica do ensaio.

4.11 Interface e distribuição

A aplicação utiliza fontes locais e artes do próprio produto, além de adaptações de layout para diferentes larguras. O empacotamento constrói os recursos usados nas entradas web. Há manifesto e service worker para instalação em navegadores compatíveis, sem que isso constitua um aplicativo nativo publicado em loja. A existência de cache também não comprova sincronização offline de operações autenticadas.

A demonstração pública informa o uso de dados fictícios e alterações temporárias. A entrada local index.html declara armazenamento em memória. Por isso, o modo de demonstração não pode ser utilizado para comprovar recuperação de dados do servidor. O ambiente autenticado, por sua vez, depende de configuração de banco, origem pública, bootstrap e transporte de e-mail.

A interface demonstrativa organiza a oferta em Essencial (Simples), com máquinas, chamados, agenda e checklists; Gestão (Intermediário), que acrescenta estoque, materiais, orçamentos e custos; e Inteligente (Avançado), com alertas e curvas de leituras simuladas. Esses conjuntos ilustram modularidade e diferem dos perfis de acesso. Não comprovam ativação comercial, sensores físicos ou previsão de falhas; preços e contratação permanecem em definição (Silva, 2026c).

Os perfis Gestor, Mantenedor e Operador estabelecem responsabilidades dentro do fluxo. A seleção de módulos define o conjunto de recursos apresentado na demonstração, enquanto a autorização do servidor limita as ações e os dados de cada usuário na empresa. O cenário HTTP verifica casos de restrição por perfil, mas não representa um ensaio exaustivo de todas as combinações entre modos, módulos e permissões.

O registro anterior da R26 identifica arquivos públicos consultados, sem estabelecer identidade do backend remoto. A R27 permanece local. A configuração

de produção, os segredos e o banco remoto não foram verificados nesta preparação documental. A análise mantém separados o artefato distribuído, a execução local e a configuração operacional.

O assistente contextual oferece regras, respostas e atalhos locais. A Universidade HEFESTO organiza orientações de uso. Esses elementos podem apoiar a navegação, porém não demonstram IA generativa, diagnóstico autônomo, curso reconhecido ou certificação. Sua eficácia como ajuda ao usuário deve ser observada por tarefas e dificuldades documentadas, e não pela presença de um ícone de atendimento.

5 RESULTADOS E DISCUSSÃO

O log tests-full.txt registra 203 testes aprovados, zero falhas, zero cancelamentos e zero testes ignorados. A duração informada é 20.257,3562 ms, aproximadamente 20,257 segundos. Esse tempo descreve a suíte, não a latência do site. Os testes abrangem regras de domínio, contratos HTTP e aspectos de interface simulada. A contagem não representa participantes, requisitos distintos ou percentual de cobertura do código (Silva, 2026c).

O relatório fluxo-controlado-r27.json registra 16 verificações aprovadas. O cenário termina com um chamado encerrado, um histórico e custo de 29.000 centavos. Após reinício e nova autenticação, o roteiro recupera máquina, chamado e histórico, preservando os textos acentuados. A evidência sustenta a integração entre serviço e persistência nas condições do ensaio, sem demonstrar comportamento para todas as configurações e entradas.

O Quadro 3 separa a unidade observada em cada procedimento e a limitação correspondente.

Quadro 3 – Síntese das evidências de execução

Procedimento	Resultado	Registro	Limite
Construção	Aprovada	build-final.txt	Sem cobertura integral de comportamento
Suíte automatizada	203 aprovados; 0 falhas	tests-full.txt	Sem cobertura percentual medida
Fluxo HTTP	16 verificações aprovadas	fluxo-controlado-r27.json	Dados fictícios; e-mail interceptado
Persistência	Registros e acentos recuperados	Verificação final do fluxo	Não equivale a restauração de backup
Capturas	13 imagens reais R27	Imagens e metadados	Sem avaliação com participantes
Publicação anterior	11 arquivos R26 iguais	Registro separado R26	R27 não publicada; backend não comparado

Fonte: elaboração do autor com base no acervo técnico do HEFESTO (Silva, 2026c).

O custo resulta de R$ 200,00 de serviço, R$ 40,00 de deslocamento e R$ 50,00 de peças, totalizando R$ 290,00. O apontamento de 30 minutos é fictício.

Esses valores verificam registro e consolidação de dados, sem representar preço praticado, economia obtida ou tempo de reparo industrial. A unidade monetária persistida é o centavo; o histórico é único no cenário executado.

A Tabela 1 permite conferir a composição do custo do atendimento fictício. Os valores são entradas do ensaio, não dados de uma intervenção industrial.

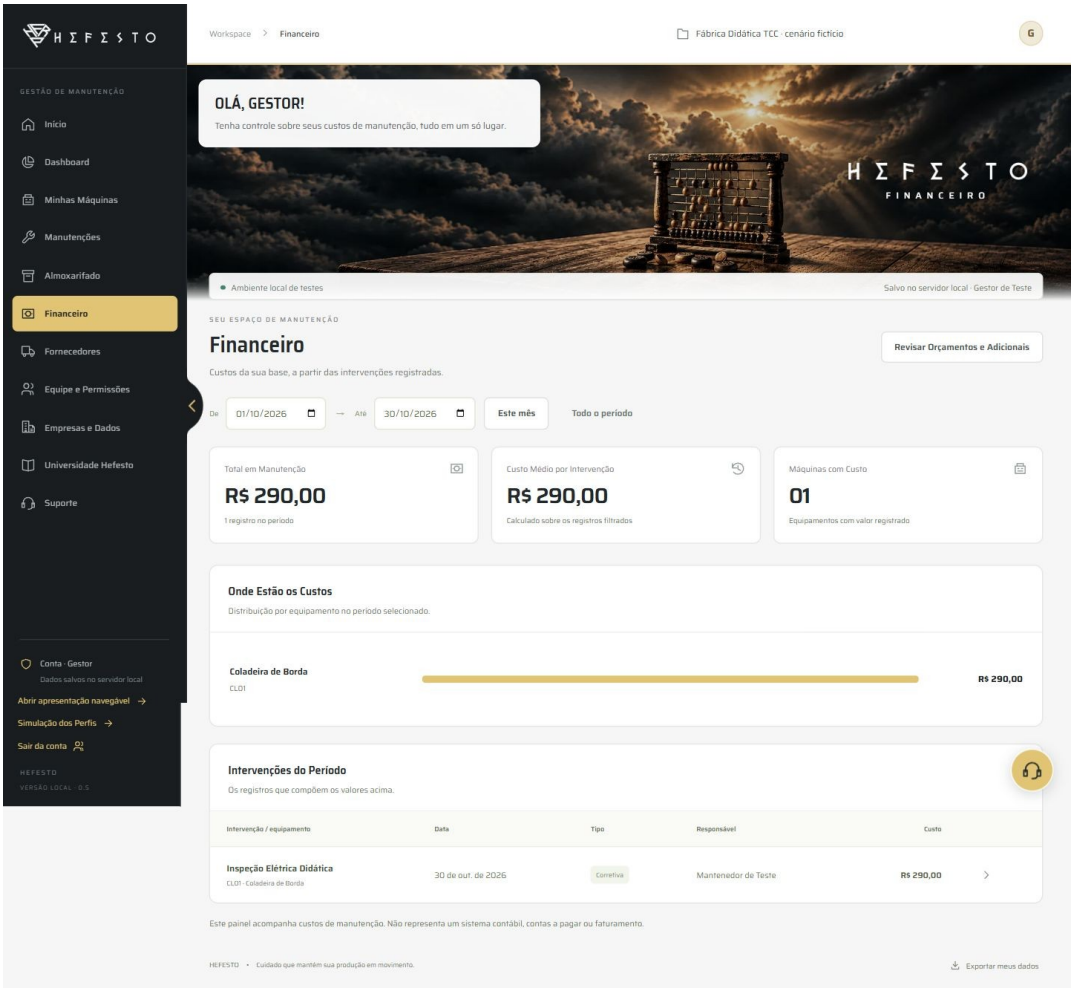
Tabela 1 – Composição do custo fictício do chamado

Componente	Valor em reais	Valor em centavos
Serviço	200,00	20.000
Deslocamento	40,00	4.000
Peças	50,00	5.000
Total	290,00	29.000

Fonte: elaboração do autor com base no acervo técnico do HEFESTO (Silva, 2026b).

A Figura 3 apresenta o mesmo total no módulo financeiro, vinculado a uma intervenção registrada.

Figura 3 – Custo consolidado no ambiente local autenticado



Fonte: acervo do projeto, R27 local, 30 out. 2026, 12:58:08 BRT (Silva, 2026c). Dados fictícios; não representa operação industrial.

A R27 reúne 13 capturas reais finais, realizadas em 30 de outubro de 2026, entre 12:56 e 13:07 BRT, com resolução, horário, texto renderizado e hash registrados. A coleta inclui telas desktop e viewport mobile de 390 pixels. A renderização documentada não cobre todos os dispositivos ou estados, nem substitui a avaliação humana de tarefas. Os dados são fictícios, e os cabeçalhos preservam a composição da R26 (Silva, 2026c).

O registro anterior de publicação R26, de 30 de outubro, informa igualdade de conteúdo para 11 arquivos públicos e redirecionamento de uma rota protegida. A comparação não abrange o backend privado, o banco ou a configuração remota. Esse registro não comprova publicação da R27, examinada localmente e identificada como não publicada no pacote técnico.

A rastreabilidade também revela atendimento parcial ou alteração contratual de requisitos. A relação de uma conta com várias empresas, a paginação e a busca

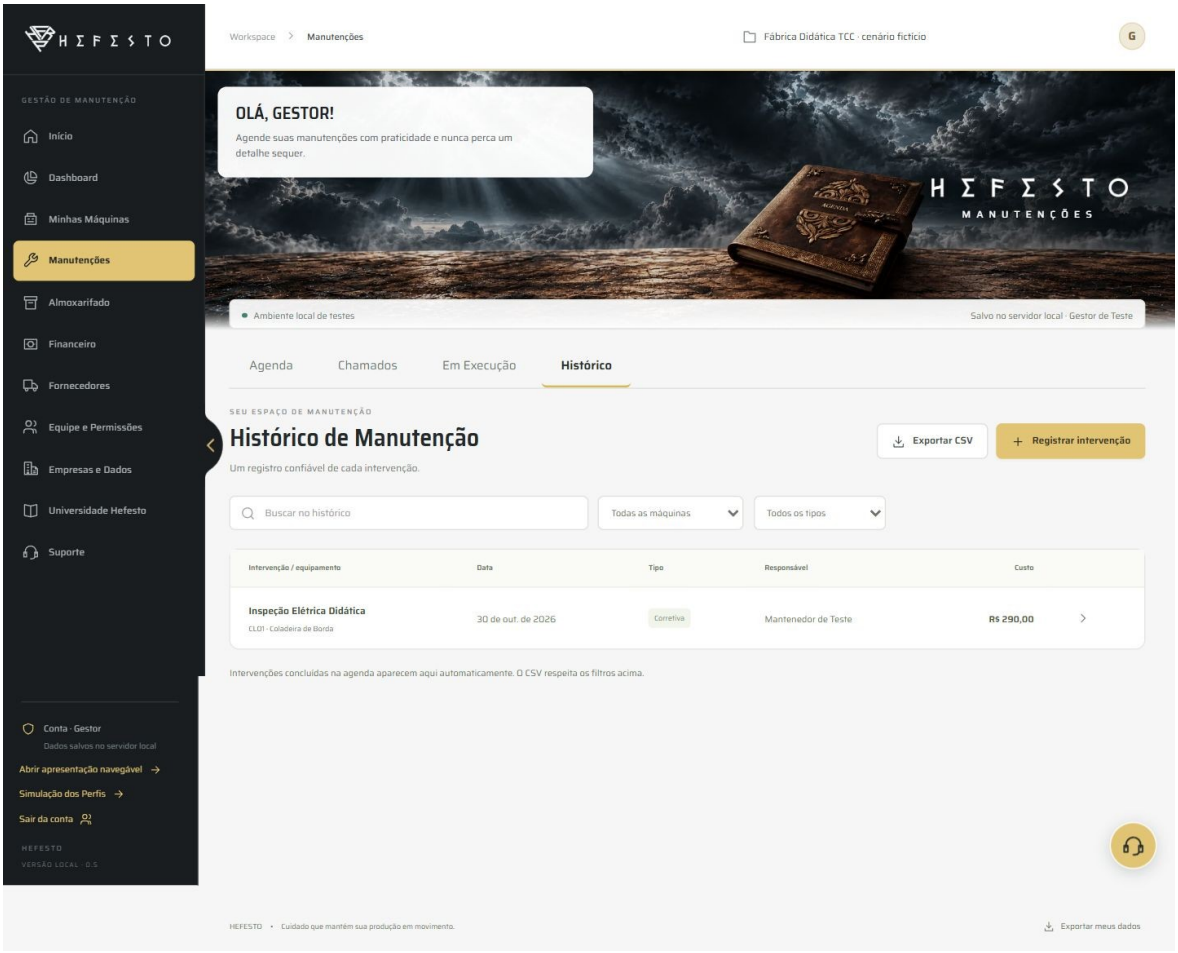
por fabricante não estão integralmente comprovadas. Arquivamento substitui exclusão física de máquina, e os papéis atuais distinguem administração da plataforma e gestão da empresa. Preservar essas diferenças evita transformar a evolução do produto em uma declaração artificial de atendimento total dos requisitos de 2025.

Os resultados sustentam uma aplicação funcional em fase de testes no recorte avaliado. A contribuição concreta é a associação entre o atendimento, as permissões e o histórico persistido, acompanhada de registros reproduzíveis. O alcance da conclusão é restrito por seleção de cenários, ausência de avaliadores independentes, uso de dados sintéticos e inexistência de observação longitudinal. A aprovação dos testes disponíveis reduz incertezas específicas; não elimina defeitos não exercitados.

Permanecem pendentes a avaliação humana de tarefas, a usabilidade, a câmera física, o deslocamento com GPS real, a restauração de backup e a operação industrial. Os indicadores sintéticos e estimativas de energia não representam medição calibrada. O assistente atual utiliza regras e respostas locais; não foi demonstrada IA generativa ou previsão autônoma de falhas. A continuidade requer procedimentos de coleta e critérios definidos antes do estudo, mantendo resultados futuros separados deste retrato.

A Figura 4 mostra o histórico do atendimento. O teste de não duplicação e a recuperação após reinício estão documentados no roteiro, não apenas nesta imagem.

Figura 4 – Histórico do atendimento fictício registrado



Fonte: acervo do projeto, R27 local, 30 out. 2026, 12:58:10 BRT (Silva, 2026c). Dados fictícios; não representa operação industrial.

5.1 Leitura conjunta dos resultados

O conjunto de evidências permite acompanhar uma cadeia concreta: o serviço aceita um cadastro autorizado, recusa uma ação incompatível, registra o atendimento e mantém o resultado após reinício. A força dessa cadeia está na relação entre estados, e não apenas na quantidade de telas ou testes. Ela mostra que o artefato ultrapassa uma representação visual, embora ainda não demonstre a totalidade de seu uso previsto.

A suíte e o cenário adicional têm unidades diferentes. A primeira registra 203 testes de granularidade variada; o segundo reúne 16 verificações de um procedimento encadeado, incluindo preparação administrativa. Somar as contagens ocultaria sobreposição e diferença de propósito. Os resultados são apresentados separadamente, sem inferir cobertura total.

A contagem de capturas também não mede cobertura funcional. As 13 imagens reais da R27 documentam estados selecionados da interface. Uma captura de chamado encerrado não comprova, sozinha, o bloqueio anterior à aprovação; essa afirmação depende do roteiro e do relatório. A figura funciona como apoio à interpretação do dado, com identificação do contexto e da natureza fictícia das informações (Silva, 2026c).

5.2 Histórico do projeto e alcance das mudanças

O formulário HEFESTO Works situa a origem declarada em 13/05/2023, em Linhares, Espírito Santo. A proposta inclui planejamento de manutenção, informações de máquinas, acesso por perfis e possibilidades de uso em diferentes setores. O documento identifica Hyago Piona Bazelato, Fiffer Silviero dos Santos e Ramon Ferreira da Silva na equipe histórica. O roteiro de apresentação de 2024 também menciona Pedro. Essa atribuição registra a construção coletiva da proposta e distingue o contexto de negócio da autoria dos trabalhos acadêmicos assinados por Ramon (Hefesto Works, 2023; Hefesto, 2024).

Os registros preservados de 2024 incluem convite de edição do arquivo HEFESTO WEB no Figma, datado de 29 de janeiro, roteiro de pitch e apresentação nomeada 05.06.24. O roteiro concentra o problema em marcenarias e fábricas de móveis de pequeno e médio porte. As telas e o discurso comercial documentam intenção de produto e decisões visuais. As promessas de redução de custos e os valores de perdas por parada não constituem resultados deste estudo. Não foram recuperados, nesse acervo, registros suficientes para demonstrar a operação de uma aplicação web naquele período (Hefesto, 2024).

O trabalho Hefesto Simplificado registra 2024 na capa e declara entrega em 06/12/2024. Descreve interface CMD em Python, usuários, fábricas, máquinas e manutenções, com persistência em arquivos pickle. Foram preservadas as variantes CORRIGIR e VALE ESTE. A cópia de código Hefesto_CMD_Version.py possui 361 linhas e termina com uma string incompleta; por isso, esse exemplar não permite reproduzir integralmente a execução histórica. A limitação da cópia recuperada não invalida os registros do trabalho nem demonstra que todas as cópias existentes naquele momento tinham o mesmo defeito (Silva, 2024).

Em 2025, o trabalho de Laboratório de Programação descreve Python, SQLite, autenticação, vínculo com empresas, máquinas, histórico, agenda e checklist, além do empacotamento para Windows. O fonte hefesto_cmd.py declara APP_VERSION igual a v0.1.0-beta. Sua estrutura confirma a existência de uma implementação em terminal, mas a inspeção não equivale a executar todos os seus fluxos ou a comprovar a identidade entre fonte e instalador. A mudança de arquivos pickle para um banco relacional caracteriza uma evolução técnica documentada entre os exemplares, sem evidência de migração automática de dados (Silva, 2025b, 2025d).

Os outros dois trabalhos de 2025 têm contribuições diferentes. Projeto de Sistemas organiza 25 requisitos funcionais, cinco não funcionais e uma arquitetura proposta. Análise de UI/UX do protótipo HEFESTO CMD examina dificuldades de interação e propõe interfaces gráficas. A avaliação descrita é qualitativa e heurística. Não foram demonstrados testes com participantes nem ganhos percentuais de usabilidade. O programa em terminal, a especificação e o design são artefatos relacionados, com estágios de maturidade próprios (Silva, 2025a, 2025c).

O manuscrito preliminar de 25 de outubro de 2026 consolidou parte desses antecedentes. A versão web 0.5.0/R27-textos-local utiliza serviços HTTP em Node.js, interface HTML/CSS/JavaScript e SQLite. Ela dá continuidade ao domínio e às decisões de produto, com reorganização da implementação. Não se deduz dos números de versão uma sequência completa de lançamentos, uma conversão automática do código Python ou a equivalência integral dos requisitos. A contribuição atual combina a implementação web com rastreabilidade e registros de verificação controlada (Silva, 2026a, 2026b, 2026c).

As limitações registradas no código Python pertencem aos exemplares inspecionados. Em particular, a incompatibilidade entre tipos de agenda e histórico no legado não foi atribuída ao código web sem nova evidência. Os registros de 2026 não são utilizados como comprovação retroativa dos programas anteriores.

A evolução web amplia a forma de acesso e reorganiza a autorização em sessões e serviços HTTP. O armazenamento persistente e o vínculo com empresas já apareciam no CMD. A contribuição atual inclui a organização de orçamento, aprovação e conferência no ciclo de atendimento examinado. As evidências de 30

de outubro sustentam esse recorte, preservando a distinção entre os mecanismos anteriores e o comportamento verificado na implementação atual (Silva, 2025b, 2026b, 2026c).

Os itens parciais da matriz são resultados relevantes da análise. Eles delimitam o que ainda precisa de decisão ou avaliação: relações multiempresa, política administrativa, paginação, busca por fabricante e aceitação humana de filtros e cadastro. A manutenção desses itens oferece orientação de desenvolvimento mais útil do que uma declaração de conclusão integral sem critérios equivalentes.

5.3 Limitações e implicações para o uso

Não houve intervenção em máquina real, coleta de vibração, temperatura ou corrente, nem comparação com instrumento calibrado. Curvas sintéticas e alertas por limiar podem demonstrar a interface, mas não produzem evidência de desempenho de um sistema de monitoramento. Para avaliar predição seriam necessários dados identificados, eventos de falha e um protocolo que separasse desenvolvimento e avaliação do modelo.

O uso de energia no painel precisa conservar a distinção entre potência em kW e energia em kWh. Estimativas baseadas em potência, horas e tarifa podem explicar um cálculo ilustrativo, mas não substituem medição. A versão estudada não sustenta conclusão sobre consumo da fábrica, economia energética ou precisão de instrumento.

A reprodução local depende de runtime e dependências compatíveis. O arquivo de lock ajuda a fixar a resolução, mas não torna idênticos sistemas operacionais, hardware, políticas do navegador e serviços externos. O estudo registra o ambiente do log e limita as afirmações ao procedimento documentado. A ausência de eventos de erro nas capturas não demonstra estabilidade em jornadas prolongadas.

A estabilidade operacional inclui questões não resolvidas pelo teste simples de reinício: backup consistente, restauração em outra instância, indisponibilidade de e-mail, expiração de sessão durante uma edição e concorrência de usuários. Esses casos devem receber critérios próprios. Não se estabeleceu acordo de nível de serviço nem latência percentil 95 para uma carga representativa.

A validação humana deverá distinguir realização da tarefa, compreensão do estado e satisfação percebida. Um ensaio conduzido apenas pelo autor pode localizar problemas funcionais, mas não representa uma amostra de operadores. A quantidade de participantes, o recrutamento e os procedimentos institucionais precisam ser definidos antes da coleta. Esta etapa do trabalho não envolveu participantes humanos.

5.4 Proposta de continuidade verificável

A primeira continuidade recomendada é um aceite de tarefas essenciais: cadastrar e conferir uma máquina, abrir chamado, avaliar proposta, registrar execução e encerrar com histórico. Para cada tarefa, devem ser registrados papel, dispositivo, resultado esperado, resultado observado, auxílio recebido e falha encontrada. Esses dados permitirão discutir limitações de interação sem atribuir causalidade industrial a um teste de interface.

Em seguida, uma avaliação de implantação pode acompanhar um conjunto definido de equipamentos durante período previamente estabelecido. Antes de medir melhoria, será necessário registrar a forma anterior de trabalho e indicadores comparáveis. O protocolo deverá diferenciar mudança de processo, treinamento e efeito do software. Não há, neste estudo, valores iniciais suficientes para calcular ganhos.

O Apêndice C reúne um protocolo proposto, ainda não executado, que organiza as evidências necessárias. Sua inclusão documenta continuidade e não amplia artificialmente os resultados. O sistema de monitoramento por sensores, se desenvolvido, exigirá uma etapa própria de integração e calibração, mantendo independentes os testes de comunicação, qualidade do sinal e utilidade para a manutenção.

6 CONCLUSÃO

O trabalho desenvolveu e examinou uma versão web utilizável do HEFESTO para gestão de manutenção, em continuidade à trajetória iniciada em 2023. A implementação 0.5.0, revisão local R27, reorganiza fluxos presentes nos programas em terminal de 2024 e 2025 e nas especificações e propostas de interface. A pergunta de pesquisa foi respondida no recorte dos cenários controlados: o fluxo selecionado relaciona cadastro de equipamento, autorização por perfil, etapas do atendimento, custo e histórico persistido, com bloqueios e recuperação após reinício registrados.

Os objetivos de relacionar requisitos e implementação, descrever a arquitetura e analisar evidências foram atendidos dentro do corpus. A suíte de 203 testes e o cenário complementar de 16 verificações oferecem suporte ao comportamento examinado, enquanto a matriz explicita requisitos parciais e mudanças de escopo. A versão constitui uma base funcional para continuidade do projeto; a evidência disponível não autoriza concluir adoção industrial, redução de custos ou qualidade global de uso.

Como continuidade, propõem-se avaliação com usuários, ensaios de continuidade operacional e estudo de implantação em ambiente industrial. A integração com sensores e a avaliação de manutenção preditiva dependem de aquisição real de dados e de critérios próprios. Preservar essas fronteiras permite desenvolver o aplicativo sem confundir potencial de aplicação com resultado já demonstrado.

REFERÊNCIAS

HEFESTO. Acervo de apresentação e prototipação do HEFESTO de 2024. Linhares, 2024. Roteiro de pitch, apresentação nomeada 05.06.24 e registros preservados de comunicação sobre Figma. Acervo histórico do projeto.

HEFESTO WORKS. Formulário ES Hub: HEFESTO Works. Linhares, 2023. Documento de apresentação do projeto; fundação declarada em 13 maio 2023. Texto preservado no acervo do projeto.

HEVNER, Alan R. et al. Design science in information systems research. MIS Quarterly, v. 28, n. 1, p. 75–105, 2004. DOI: 10.2307/25148625. Disponível em: https://aisel.aisnet.org/misq/vol28/iss1/6/. Acesso em: 30 out. 2026. Resumo consultado.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO 9241-11:2018: ergonomics of human-system interaction: part 11: usability: definitions and concepts. Geneva: ISO, 2018. Disponível em: https://www.iso.org/standard/63500.html. Acesso em: 30 out. 2026. Resumo público consultado.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO/IEC 25010:2023: systems and software engineering: systems and software Quality Requirements and Evaluation (SQuaRE): product quality model. Geneva: ISO, 2023. Disponível em: https://www.iso.org/standard/78176.html. Acesso em: 30 out. 2026. Resumo público consultado.

NIELSEN, Jakob. 10 usability heuristics for user interface design. Nielsen Norman Group, 24 abr. 1994. Revisado em: 30 jan. 2024. Disponível em: https://www.nngroup.com/articles/ten-usability-heuristics/. Acesso em: 30 out. 2026.

NODE.JS. HTTP. Node.js v24.19.0 documentation. [S.l.], 2026a. Disponível em: https://nodejs.org/download/release/v24.19.0/docs/api/http.html. Acesso em: 30 out. 2026.

NODE.JS. SQLite. Node.js v24.19.0 documentation. [S.l.], 2026b. Disponível em: https://nodejs.org/download/release/v24.19.0/docs/api/sqlite.html. Acesso em: 30 out. 2026.

OWASP FOUNDATION. Authorization cheat sheet. OWASP Cheat Sheet Series, [s.d.]a. Disponível em: https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html. Acesso em: 30 out. 2026.

OWASP FOUNDATION. Cross-site request forgery prevention cheat sheet. OWASP Cheat Sheet Series, [s.d.]b. Disponível em: https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html. Acesso em: 30 out. 2026.

OWASP FOUNDATION. Password storage cheat sheet. OWASP Cheat Sheet Series, [s.d.]c. Disponível em: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html. Acesso em: 30 out. 2026.

PACIFIC NORTHWEST NATIONAL LABORATORY. O&M best practice issue discussion: maintenance approaches. [S.l.]: PNNL, [s.d.]. Disponível em: https://www.pnnl.gov/projects/om-best-practices/maintenance-approaches. Acesso em: 30 out. 2026.

PERCIVAL, Colin; JOSEFSSON, Simon. The scrypt password-based key derivation function. RFC 7914. [S.l.]: RFC Editor, 2016. DOI: 10.17487/RFC7914. Disponível em: https://www.rfc-editor.org/rfc/rfc7914. Acesso em: 30 out. 2026.

SILVA, Ramon Ferreira da. Hefesto Simplificado: sistema de gestão de manutenção de máquinas de marcenaria em Python. Santo Amaro: Universidade Santo Amaro, 2024. Trabalho de Laboratório de Programação, Engenharia de Software. Variantes CORRIGIR e VALE ESTE preservadas no acervo do autor.

SILVA, Ramon Ferreira da. Análise de UI/UX do protótipo HEFESTO CMD. São Paulo: Universidade Santo Amaro, 2025a. Projeto Integrador de Projeto de UI e UX, Engenharia de Software. Documento do acervo do autor.

SILVA, Ramon Ferreira da. HEFESTO CMD v0.1.0-beta. 2025b. Código-fonte Python, arquivo hefesto_cmd.py. Acervo técnico do projeto.

SILVA, Ramon Ferreira da. HEFESTO: projeto e análise de requisitos de um sistema para gerenciamento de manutenção. São Paulo: Universidade Santo Amaro, 2025c. Projeto Integrador de Projeto de Sistemas, Engenharia de Software. Documento do acervo do autor.

SILVA, Ramon Ferreira da. Hefesto Simplificado: sistema de gestão de manutenção de máquinas de marcenaria em Python. Santo Amaro: Universidade Santo Amaro, 2025d. Trabalho de Laboratório de Programação, Engenharia de Software. Versão com SQLite e interface de linha de comando. Acervo do autor.

SILVA, Ramon Ferreira da. HEFESTO requisitos e proposta de interface para gerenciamento de manutenção. São Paulo, 25 out. 2026a. Manuscrito preliminar HEFESTO_TCC_Revisado_2026_10_25.docx. Acervo do projeto.

SILVA, Ramon Ferreira da. HEFESTO 0.5.0 e registros de verificação R25. 30 out. 2026b. Código-fonte, matrizes e evidências no pacote HEFESTO_HANDOFF_ATUAL_TCC_2026-10-30.zip. Publicação R25 de 29 out. 2026. Acervo técnico do projeto.

SILVA, Ramon Ferreira da. HEFESTO 0.5.0: revisão R27-textos-local e evidências de verificação. 30 out. 2026c. Pacote HEFESTO_HANDOFF_TCC_R27_TEXTOS_2026-10-30.zip. Capturas, logs e aplicação de teste; revisão local não publicada. Registro anterior de publicação R26 preservado separadamente. Acervo técnico do projeto.

SOUPPAYA, Murugiah; SCARFONE, Karen; DODSON, Donna. Secure Software Development Framework (SSDF) version 1.1: recommendations for mitigating the risk of software vulnerabilities. Gaithersburg: NIST, 2022. NIST SP 800-218. DOI: 10.6028/NIST.SP.800-218. Disponível em: https://csrc.nist.gov/pubs/sp/800/218/final. Acesso em: 30 out. 2026. Resumo público consultado.

SQLITE. Atomic commit in SQLite. [S.l.], [s.d.]. Disponível em: https://sqlite.org/atomiccommit.html. Acesso em: 30 out. 2026.

WASHIZAKI, Hironori (ed.). Guide to the Software Engineering Body of Knowledge. Version 4.0a. IEEE Computer Society, 2026. Disponível em: https://ieeecs-media.computer.org/media/education/swebok/swebok-v4.pdf. Acesso em: 30 out. 2026. Seções sobre requisitos, rastreabilidade e testes consultadas.

WORLD WIDE WEB CONSORTIUM. Web Content Accessibility Guidelines (WCAG) 2.2. W3C Recommendation, 12 dez. 2024. Disponível em: https://www.w3.org/TR/2024/REC-WCAG22-20241212/. Acesso em: 30 out. 2026.

APÊNDICE A RASTREABILIDADE DOS REQUISITOS

Esta matriz preserva os 25 requisitos funcionais e os cinco não funcionais do corpus de 2025 e a análise histórica registrada na R25. Não constitui nova auditoria integral dos 30 requisitos na R27. A rodada atual reexamina o fluxo selecionado e a revisão textual; o CSV original permanece inalterado (Silva, 2025c, 2026b, 2026c).

Quadro 4 – Correspondência entre requisitos históricos e análise da R25

ID	Requisito histórico	Estado e evidência	Limite
RF01	Cadastrar usuário com nome, e-mail e senha.	Verificado em cenário controlado. Cadastro HTTP, confirmação e persistência; suíte access/email/accounts	Entrega real de e-mail e aceite humano pendentes
RF02	Autenticar e distinguir usuário comum/admin.	Verificado em cenários controlados. accounts/admin-control/suite; isolamento e perfis via HTTP	MFA de produção não refeito no fluxo complementar
RF03	Recuperar conta com “Esqueci minha senha”.	Implementado; testes automatizados atuais. access/email-accounts/persistent-security	Entrega SMTP real não revalidada hoje
RF04	Alterar a própria senha.	Implementado; testes automatizados atuais. accounts.test e trusted-devices.test	Aceite humano pendente
RF05	Logout seguro.	Implementado; testes automatizados atuais. accounts.test: logout, sessão, revogação	Aceite humano pendente
RF06	Usuário cadastra uma ou mais empresas.	Parcial / contrato diferente. Servidor vincula conta de empresa a company_id; criação múltipla existe no modo local	Usuário com várias empresas no backend não confirmado; não marcar atendido integral
RF07	Selecionar empresa ativa.	Parcial / contexto controlado. Servidor fixa empresa autorizada; modo local possui seletor	Troca multiempresa da mesma conta produtiva não confirmada
RF08	Admin lista, edita e exclui empresas.	Parcial / política alterada. Administração e suspensão/reativação com testes	Editar/excluir livremente empresas do requisito antigo não comprovado
RF09	Admin vincula usuário a empresas.	Parcial / política alterada. Convites individuais por papel/máquina e revogação testados	Não há prova de vínculo multiempresa equivalente ao CLI

ID	Requisito histórico	Estado e evidência	Limite
RF10	Cadastrar máquina vinculada à empresa ativa.	Núcleo verificado; aceite integral pendente. Inventário persistido via HTTP e captura 02; foto/componentes no código/suíte	Cenário novo não enviou foto/anexo real
RF11	Listar máquinas da empresa com paginação.	Parcial. Listas e levantamento renderizados, escopo por empresa testado	Paginação do requisito original não confirmada; testar volume
RF12	Buscar por nome, modelo ou fabricante.	Parcial. Busca da grade por nome/tag/modelo; código inspecionado	Fabricante não consta no filtro principal e normalização completa de acentos não comprovada
RF13	Admin edita máquina.	Implementado com papel redefinido. Gestor gerencia inventário; servidor valida mantenedor/escopo	Admin da plataforma não é gestor da empresa; validar política de edição com usuário
RF14	Admin exclui máquina.	Implementado como arquivamento. domain.test: preserva histórico e impede novas operações	Documentar substituição de exclusão física por arquivamento
RF15	Admin altera status da máquina.	Implementado; testes automatizados atuais. domain.js + validações de status e inventário	Operação humana/estados de equipamentos reais não validados
RF16	Registrar manutenção com datas, tipo, técnico, descrição e custo.	Núcleo verificado. domain.test + fechamento HTTP e histórico R$ 290	Intervenção física não executada
RF17	Listar histórico da empresa ativa.	Núcleo verificado. Isolamento HTTP e captura 07	Filtros e ordenação precisam aceite humano
RF18	Filtrar histórico por máquina.	Implementado; testes e código. Filtro de histórico e escopo em domain/suite	Não foi feita bateria humana de busca nesta rodada
RF19	Agendar manutenção futura.	Implementado; testes automatizados atuais. domain/accounts/evolution tests para agenda e geração de ordem	Demonstração humana pendente
RF20	Criar checklist por agendamento.	Implementado; testes automatizados atuais. domain/evolution: checklist e plano aprovado	Chamado extra de 30/10 não usou checklist
RF21	Listar pendentes e finalizados.	Implementado; testes e código. planStatus, listas e atrasos	Aceite dos filtros em todas as telas pendente

ID	Requisito histórico	Estado e evidência	Limite
RF22	Marcar tarefas como concluídas.	Implementado; testes automatizados atuais. domain/evolution: etapas e bloqueio de conclusão	Desmarcar e registros devem ser revistos conforme protocolo final
RF23	Concluir agenda e gerar histórico.	Verificado em cenários controlados. Todos os tipos atuais e deduplicação na suíte; histórico único no fluxo adicional	Não equivale a execução do antigo CLI Python
RF24	Consultar e buscar fornecedores.	Implementado; código e suíte. Fornecedores no domínio e interface	Busca/contatos e catálogo real precisam aceite
RF25	Admin mantém fornecedores.	Implementado com papel redefinido. Cadastro/edição escopados; gestor no backend	Rever regra de remoção e papel do admin no texto
RNF01	Hash forte para senhas, ex.: bcrypt.	Implementado; testes controlados. scrypt, sal e testes de autenticação/isolamento	Sem auditoria externa de segurança; não usar bcrypt como tecnologia atual
RNF02	Interface intuitiva e consistente.	Pendente de validação com usuários. Interface/capturas disponíveis	Sem estudo formal de usabilidade/acessibilidade
RNF03	Consultas e listas eficientes.	Parcial; sem SLA validado. Compressão/cache e testes de contratos	Sem ensaio de carga representativo, latência P95 ou concorrência industrial
RNF04	Instalador Windows sem Python prévio.	Mudança de escopo. Distribuição web/PWA e versão local	Instalador Windows do CLI antigo não foi validado; web não satisfaz literalmente mesmo requisito
RNF05	Código modular e manutenível.	Parcialmente sustentado por inspeção. Módulos de domínio, serviço, UI, testes e pacote reproduzível	Não há métrica formal de manutenibilidade; migração/backup operacional precisam validação

Fonte: elaboração do autor com base no acervo técnico do HEFESTO (Silva, 2026b).

APÊNDICE B IDENTIFICAÇÃO E REPRODUÇÃO

O corpus atual é HEFESTO_HANDOFF_TCC_R27_TEXTOS_2026-10-30.zip, com 36.460.290 bytes e SHA-256 4ac292796bce049b705d2f0331dfd91b4233305d8a3ae0e7e26f970e09aeae83. Seus 447 registros de manifesto foram conferidos sem divergência. O acervo técnico anterior e a revisão acadêmica R3 permanecem preservados. A conferência não substitui reexecução independente dos ensaios.

A versão declarada é 0.5.0, revisão R27-textos-local, não publicada. A rodada fornecida utiliza Node.js v24.19.0 em Windows. O relatório HTTP registra início em 30/10/2026 às 15:55:54 UTC e término às 15:59:52 UTC, correspondentes a 12:55:54 e 12:59:52 BRT. As capturas finais estendem-se até 13:07 BRT.

Os registros atuais estão em verificacoes/, incluindo tests-full.txt e fluxo-controlado-r27.json; capturas/ reúne as 13 imagens e seus metadados. O catálogo de modos está em aplicativo/dist/demo-catalog.js. O registro de publicação R26 encontra-se em registros_anteriores/. As instruções constam de 05_REPRODUCAO.txt (Silva, 2026c).

Para reproduzir a suíte, deve-se extrair uma cópia do pacote, utilizar Node.js compatível com node:sqlite e instalar as dependências pelo pnpm com o arquivo de lock. No diretório aplicativo, executar em sequência pnpm install --frozen-lockfile, node build.mjs e node --test tests/*.test.mjs. Interromper o procedimento se uma etapa falhar e preservar os registros do novo ambiente.

Para o cenário autenticado, executar node qa-tcc-r27.mjs na cópia de trabalho, conforme 05_REPRODUCAO.txt. O roteiro cria SQLite temporário e escuta em 127.0.0.1:4198; suas condições de preparação são exclusivas de teste. As capturas da rodada fornecida foram obtidas separadamente na interface real, pelo navegador. Uma nova coleta deve registrar ambiente, resolução, horário e hash. O pacote não inclui banco privado ou segredos de produção.

A abertura de aplicativo/dist/index.html explora a demonstração em memória, descartada ao recarregar; não reproduz persistência do servidor. Novos ensaios devem preservar o corpus original e registrar ambiente, data, entradas e resultados, inclusive em consultas públicas.

APÊNDICE C PROTOCOLO PROPOSTO DE AVALIAÇÃO

Este protocolo proposto não foi executado. O planejamento com pessoas e dados de operação deve ser alinhado à orientação e aos procedimentos institucionais antes da coleta.

Quadro 5 – Procedimentos propostos e evidências a registrar

ID e tarefa	Procedimento proposto	Critério de observação
H01 Cadastro e recuperação	Cadastrar com e-mail de teste, confirmar, sair e recuperar	Conta única, mensagem compreensível, código de uso único, entrada com nova senha
H02 Levantamento e conferência	Cadastrar equipamento e conferir componentes com fonte técnica	Dados corretos, código estável, etapa rascunho/revisado compreendida
H03 Ocorrência e papéis	Abrir chamado da máquina atribuída; tentar acesso indevido	Chamado chega ao gestor; outra empresa/máquina permanece inacessível
H04 Atendimento completo	Atribuir, aceitar, orçar, aprovar, executar, conferir e fechar	Transições claras, registro de autoria e histórico único; custo confere
H05 Deslocamento físico	Testar autorização/negação, posição, perda de sinal e finalização em trajeto seguro	Atualização mensurável, privacidade e remoção da localização; registrar navegador/aparelho
H06 Rolagem e responsividade	Rolar cards/tabelas, navegar e trocar orientações	Conteúdo alcançável sem salto indevido; sem sobreposição relevante
H07 Foto real	Selecionar galeria, abrir câmera, negar/cancelar, capturar e salvar	Foto correta; mídia encerrada; erros compreensíveis
H08 Estoque e custos	Reservar, entregar, consumir, devolver e encerrar	Saldo/custo conferem; sem dupla baixa ou ferramenta pendente
H09 Usabilidade	Executar tarefas sem instrução passo a passo; registrar dificuldades	Critérios e instrumentos definidos antes da coleta; tempo/erros/sucesso por tarefa
H10 Energia/sensores	Comparar aquisição com medidor/referência calibrada	Unidades, precisão, lacunas e estimativa documentadas
H11 Continuidade de operação	Restaurar backup em ambiente separado e simular falhas de rede/processo	Integridade e recuperação com critérios objetivos

Fonte: elaboração do autor com base no acervo técnico do HEFESTO (Silva, 2026b).