



## ANÁLISE DO SOFTWARE PARA GERENCIAMENTO DE MANUTENÇÕES

Franco, Lucas e Ulbrich<sup>1</sup>  
Bueno, Victor e Cardoso<sup>2</sup>

### RESUMO:

Nesse resumo expandido será contemplado toda a parte de desenvolvimento do software, tendo como seu objetivo o gerenciamento de manutenções de equipamentos dentro das fábricas, utilizando linguagens de programação como Javascript, esperando ter por sucesso a criação do sistema, a fim de entregar para o cliente um software que atenda suas necessidades.

**Palavras-chave:** JavaScript. Node.js. Gerenciamento. Manutenção. Software.

### 1 INTRODUÇÃO

O software Thermal Tech surge como uma solução inovadora com o propósito de simplificar e aprimorar a gestão das manutenções internas nos sistemas de refrigeração das grandes empresas. Com uma abordagem centrada na eficiência, o sistema oferece a capacidade de planejar e executar manutenções preventivas de forma otimizada, além de permitir um controle preciso das horas de trabalho atribuídas a cada técnico.

Neste artigo, exploraremos em detalhes a trajetória de desenvolvimento do Thermal Tech, desde as escolhas fundamentais de linguagens de programação até as melhores práticas adotadas na construção do software.

<sup>1</sup> Lucas Ulbrich Franco 4º período do curso de Engenharia de Software; Ugv – Centro Universitário.

<sup>2</sup> Victor Luis Cardoso Bueno 4º período do curso de Engenharia de Software; Ugv – Centro Universitário.

<sup>3</sup> Docente do curso de Bacharelado em Engenharia de Software, Ugv - Centro Universitário. Bacharel em Sistemas de Informação e Especialista em Didática e Docência do Ensino Superior pelo Centro Universitário do Vale do Iguaçu. E-mail: prof\_andreatomko@ugv.edu.br.



## 2 DESENVOLVIMENTO

### 2.1 REQUISITOS E PLANEJAMENTO

Para criar uma base sólida para o desenvolvimento do software, foi identificando e especificando os requisitos funcionais e não funcionais. Os requisitos funcionais delinearão as principais funcionalidades que o software deve oferecer aos usuários. Isso inclui recursos como registro de usuários, autenticação, criação de perfis, busca avançada e interação com outros usuários por meio de mensagens.

Além dos requisitos funcionais, também documentamos os requisitos não funcionais. Estes definiram aspectos cruciais, como desempenho, segurança e escalabilidade. Por exemplo, determinamos que o tempo de resposta das consultas de pesquisa não deveria exceder 2 segundos, e que os dados do usuário seriam criptografados para garantir a privacidade.

Com base na especificação de requisitos, criamos um plano detalhado para o desenvolvimento do projeto. Estabelecemos marcos e metas claras para cada fase do processo, desde o design até a implantação. Nosso cronograma foi projetado para ser realista e flexível, permitindo ajustes à medida que o projeto progredia.

A equipe foi dividida em grupos de trabalho especializados, cada um focado em áreas como desenvolvimento frontend, backend, banco de dados e design de interface. Isso permitiu uma abordagem ágil, onde cada equipe poderia trabalhar em paralelo, garantindo que todas as partes do software fossem desenvolvidas de forma coesa.

De acordo com o escritor Mike Cohn (2023):

“O trabalho em equipe faz parte da essência de qualquer processo ágil. O Manifesto Ágil proclama que devemos preferir “pessoas e interações a processos e ferramentas” (Beck et al., 2001), ou seja, software bom vem de boas equipes. O próprio Scrum ganhou seu nome da ideia de que uma equipe de desenvolvimento de produtos deve se comportar de maneira parecida a de um time de rugby – um grupo de pessoas movendo a bola pelo campo de forma coesa. Considerando-se a importância central das equipes para um desenvolvimento ágil bem-sucedido, você não deve ficar surpreso ao encontrar um capítulo chamado “Trabalho em equipe”.”(Mike Cohn, 2011, p. 209).



A fase de especificação de requisitos e planejamento foi fundamental para definir o escopo do software e estabelecer uma estratégia clara para o desenvolvimento. Isso permitiu que a equipe trabalhasse de maneira eficaz, cumprisse prazos e entregasse um produto final que atendesse às necessidades dos usuários e aos padrões de qualidade estabelecidos.

## 2.2 ARQUITETURA E TECNOLOGIAS UTILIZADAS

A arquitetura do nosso software foi cuidadosamente projetada para atender às necessidades de escalabilidade, desempenho e manutenção. Optamos por uma arquitetura de microserviços devido à flexibilidade que ela proporciona, permitindo que diferentes componentes do software fossem desenvolvidos, implantados e escalados independentemente. Isso facilitou a manutenção e atualizações contínuas.

Para o desenvolvimento backend, escolhemos o Node.js como plataforma principal. A escolha do Node.js foi baseada na sua capacidade de lidar com um grande número de conexões simultâneas de forma assíncrona, o que era crucial para a escalabilidade do nosso software. Além disso, o uso do Express.js, um framework para Node.js, simplificou a criação de APIs robustas e eficientes. “O Node.js é um ambiente de servidor de código aberto que possibilita a execução do JavaScript como uma linguagem de programação server side”. (OLIVEIRA, Cláudio Luís V.; ZANETTI, Humberto Augusto P, 2020, p. 113).

No lado do design, utilizamos a linguagem CSS3 para criar um estilo visual consistente e atraente. Optamos por adotar as diretrizes de design responsivo para garantir que o software fosse facilmente acessível em dispositivos móveis e desktops.

Além disso, para oferecer uma experiência de desktop aos nossos usuários, utilizamos o Electron para empacotar nosso software,

Segundo o desenvolvedor Thiago Marinho, Engenheiro de Software (2023):

“Com Electron podemos desenvolver aplicações desktop usando HTML, CSS e Javascript – As famosas tecnologias da Web. Possui integração com Node.js,



podemos usá-lo para construir não apenas as telas mas toda a lógica de um BackEnd que acessa recursos do sistema operacional — diretórios locais, banco de dados de maneira mais simples.”

Sendo assim, transformando-o em um aplicativo desktop de fácil instalação e utilização.

### 3 CONSIDERAÇÕES FINAIS

Portanto, a escolha das tecnologias e a arquitetura adotada foram fundamentais para a criação de um software eficiente, escalável e de alta qualidade. As decisões foram tomadas com base nas necessidades específicas do projeto e na busca por uma experiência de usuário excepcional. O software segue em desenvolvimento, onde o foco passará a ser o JavaScript, assim que o banco de dados estiver funcionando corretamente.

### 4 REFERÊNCIAS

COHN, Mike. **Desenvolvimento de software com Scrum**. Grupo A, 2011. E-book. ISBN 9788577808199.

Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788577808199/>. Acesso em: 04 set. 2023.

OLIVEIRA, Cláudio Luís V.; ZANETTI, Humberto Augusto P. **JAVASCRIPT DESCOMPLICADO - PROGRAMAÇÃO PARA WEB, IOT E DISPOSITIVOS MÓVEIS**. São Paulo: Editora Saraiva, 2020. E-book. ISBN 9788536533100.

Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788536533100/>. Acesso em: 04 set. 2023.

MARINHO. Thiago. **Electron - construindo aplicações desktop com JavaScript**. Rocketseat, 2023. Disponível em:

<https://blog.rocketseat.com.br/electron-construindo-aplicacoes-desktop-com-javascript/>. Acesso em: 04 set. 2023.