

# Setup Smart: Uma ferramenta interativa para recomendação de hardware baseada em necessidades do usuário.

**Matheus Biesdorf, Ana Karina Dourado Salina de Oliveira**

Faculdade de Computação - Universidade Federal de Mato Grosso do Sul (UFMS)  
Campo Grande - MS - Brasil

matheus\_biesdorf@ufms.br, ana.salina@ufms.br

**Resumo.** Este artigo apresenta o desenvolvimento da aplicação web Setup Smart, criada com o intuito de suprir a falta de informações acessíveis sobre hardware e especificações de computadores, principalmente entre usuários que não possuem conhecimentos técnicos em informática. A aplicação tem como objetivo auxiliar o usuário na escolha da configuração de computador ou laptop mais adequada às suas necessidades, oferecendo uma ferramenta simples, intuitiva e educativa para explorar componentes de hardware e compreender seus usos. A aplicação foi desenvolvida integralmente em JavaScript, utilizando Node.js no back-end, React no front-end e MySQL para o armazenamento dos dados. Este artigo descreve a estrutura, a arquitetura e o funcionamento da aplicação.

**Abstract.** This article presents the development of the web application “Setup Smart”, created with the purpose of addressing the lack of accessible information about computer hardware and specifications, especially among users who do not possess technical knowledge in computing. This application aims to assist users in selecting the most suitable computer or laptop configuration for their needs, offering a simple, intuitive and educational tool to explore hardware components and understand their functions. The system was developed entirely in JavaScript, using Node.js for the back-end, React for the front-end and MySQL for data storage. This article describes the structure, architecture and functioning of the application.

## 1. Introdução

A escolha de uma configuração adequada de computador para uso pessoal, profissional ou corporativo é um desafio comum devido à grande variedade de componentes disponíveis e à complexidade das especificações técnicas. Esse cenário dificulta que usuários leigos, compradores ocasionais e até mesmo empresas identifiquem quais peças oferecem o melhor desempenho para suas necessidades, o que frequentemente resulta em escolhas inadequadas e gastos desnecessários. Nesse contexto, diferentes perfis de usuários enfrentam dificuldades semelhantes, desde indivíduos que necessitam de um computador para atividades pessoais ou profissionais até empresas que precisam adquirir vários equipamentos e desejam otimizar custos. Em ambos os casos, decisões equivocadas podem gerar desperdício financeiro, baixo desempenho e insatisfação a longo prazo.

Com o objetivo de solucionar essas dificuldades, foi desenvolvida a aplicação web Setup Smart, uma ferramenta que centraliza informações essenciais sobre hardware e auxilia o usuário a selecionar uma configuração de computador ou laptop adequada ao seu perfil de uso. Para isso, foram levantados os requisitos mínimos de sistema necessários para executar cada tipo de jogo e software, e realizada uma média que indica o que é necessário para rodar esses programas com uma margem de segurança.

Além disso, a aplicação oferece um conjunto de recursos voltados ao aprendizado e à orientação do usuário durante o processo de escolha de um equipamento de informática. A Setup Smart disponibiliza artigos informativos sobre *hardware*, permitindo que o usuário comprehenda, de forma acessível, as funções e características dos principais componentes de um computador. Também é possível consultar individualmente as especificações técnicas e a faixa de preço de diversos produtos, facilitando comparações entre alternativas disponíveis no mercado. O diferencial da aplicação é um questionário guiado que coleta informações sobre o perfil e as necessidades do usuário; com base nas respostas fornecidas, é recomendada automaticamente uma configuração de computador ou *laptop* adequada ao seu uso, tornando o processo de decisão mais rápido, assertivo e econômico.

Este artigo está estruturado da seguinte maneira: Seção 2: Aborda trabalhos relacionados. Seção 3: Aborda requisitos e arquitetura. Seção 4: Aborda a implementação. Seção 5: Aborda as considerações finais. Seção 6: Aborda trabalhos futuros. Referências: Lista as referências utilizadas no desenvolvimento deste trabalho.

## 2. Trabalhos Relacionados

Diversas ferramentas e serviços disponíveis na internet auxiliam usuários interessados em adquirir computadores ou componentes de *hardware*. No entanto, a maioria dessas soluções exige que o indivíduo já possua algum conhecimento técnico para interpretar especificações, avaliar preços e identificar compatibilidade entre peças. Nesta seção são apresentados dois exemplos relevantes nesse contexto e como cada um deles contribui para o processo de escolha de uma configuração de computador, bem como as limitações que justificam a criação da Setup Smart.

### 2.1 MEUPC.NET

MEUPC.NET [1] é um site voltado para montagem personalizada de computadores, no qual o usuário escolhe manualmente cada componente do sistema, como processador, placa mãe, memórias, placa de vídeo, armazenamento, etc. A plataforma oferece recursos importantes, como verificação automática de compatibilidade entre peças e acompanhamento de preços, permitindo que o usuário compare diferentes opções de forma prática. Apesar disso, sua utilização demanda um conhecimento prévio sobre *hardware*, já que todas as decisões são tomadas pelo próprio usuário e não há qualquer mecanismo de recomendação automática. Além disso, o serviço é direcionado exclusivamente para computadores de mesa, não abrangendo *laptops*.

### 2.2 PC Build Wizard

O PC Build Wizard [2] não é uma aplicação formal, mas sim um serviço gratuito de curadoria de *hardware* disponibilizado por um criador de conteúdo em plataformas como X, Telegram e Discord. O responsável pelo serviço publica diariamente sugestões de componentes, indicando produtos com boa reputação, preço favorável ou abaixo da média histórica dos últimos 30 dias. Diferentemente de ferramentas tradicionais, o PC Build Wizard também abrange outras categorias, como smartphones e tablets. Entretanto, embora forneça recomendações úteis, cabe ao usuário interpretar as informações, comparar modelos e decidir qual opção atende às suas necessidades específicas, o que ainda demanda compreensão básica de *hardware*.

## 2.3 Análise comparativa e contribuição da Setup Smart

As soluções analisadas fornecem suporte valioso para consumidores que já possuem familiaridade com o universo de *hardware*, oferecendo informações sobre preços, compatibilidade e qualidade dos produtos. No entanto, ambas apresentam uma limitação comum: a necessidade de conhecimento técnico por parte do usuário. Assim, grande parte do público leigo permanece dependente de terceiros ou sujeito a escolhas inadequadas.

Nesse contexto, a Setup Smart se diferencia ao adotar uma abordagem orientada às necessidades individuais do usuário. Por meio de um questionário guiado, a aplicação identifica o perfil de uso de cada pessoa e gera automaticamente uma recomendação de computador ou *laptop* adequado às suas demandas, eliminando a necessidade de conhecimento técnico avançado. Além disso, estão disponíveis aos usuários diversos artigos explicativos e consultas diretas a especificações e preços, combinando aprendizagem acessível com orientação prática. Dessa forma, a Setup Smart fornece uma experiência com algumas funcionalidades diferenciadas, especialmente para usuários iniciantes, além de beneficiar empresas que desejam otimizar investimentos ao selecionar equipamentos de forma mais assertiva.

## 2.4 Comparação entre a Setup Smart e outros serviços e aplicações

A Tabela 1 apresenta uma comparação entre os aplicativos ou serviços Setup Smart, MEUPC.NET e PC Build Wizard.

Características	MEUPC.NET	PC Build Wizard	Setup Smart
Objetivo principal	Permitir que o usuário monte manualmente um computador escolhendo cada componente	Oferecer recomendações diárias de produtos com bom custo-benefício.	Recomendar automaticamente uma configuração completa com base nas necessidades do usuário.
Público alvo	Usuários com conhecimento intermediário ou avançado em <i>hardware</i> .	Consumidores que acompanham promoções e possuem noções básicas de especificações.	Usuários leigos, intermediários, profissionais e empresas que desejam otimizar compras.
Modo de seleção	Escolha manual de cada peça.	Lista de recomendações publicadas pelo criador.	Questionário guiado que identifica o perfil de uso do usuário.
Compatibilidade	Verificação automática entre componentes escolhidos.	Não verifica compatibilidade pois recomenda produtos avulsos.	Recomendação já baseada em componentes compatíveis.
Abrangência	Somente desktops.	Desktops,	Desktops e <i>laptops</i> .

		dispositivos móveis e acessórios.	
Consulta de preços	Monitora preços e histórico.	Considera preços atuais e médias.	Exibe o menor preço atualizado de todos os produtos.
Educação do usuário	Não possui conteúdo educativo.	Não possui conteúdo educativo.	Possui artigos informativos sobre <i>hardware</i> , especificações e informática

**Tabela 1: Comparação entre serviços de auxílio sobre *hardware* e tecnologia**

### 3. Requisitos e Arquitetura

#### 3.1 Análise de contexto

##### 3.1.1 Visão geral

A Setup Smart é uma aplicação web desenvolvida com o objetivo de auxiliar usuários na escolha de configurações de computador e *laptop* adequadas às suas necessidades. A plataforma centraliza informações essenciais sobre *hardware*, permitindo que o usuário consulte especificações técnicas, faixa de preço e o desempenho esperado de diferentes componentes. Além disso, são disponibilizados artigos informativos que introduzem conceitos fundamentais de informática, facilitando o aprendizado de pessoas com pouco ou nenhum conhecimento técnico. Um elemento central da aplicação é o questionário guiado, que coleta dados sobre o perfil e o uso pretendido pelo usuário e, a partir disso, gera automaticamente recomendações de configurações compatíveis e otimizadas. Dessa forma, a Setup Smart atua tanto como ferramenta educativa quanto como suporte para tomadas de decisão relacionadas à compra de equipamentos de informática.

##### 3.1.2 Condições restritivas

O funcionamento da aplicação depende de alguns fatores técnicos essenciais. Por se tratar de uma aplicação baseada em arquitetura *web*, é necessária conexão ativa com a internet para acessar suas funcionalidades principais, realizar consultas ao banco de dados e gerar recomendações. Além disso, parte das informações exibidas ao usuário, como preços atualizados de produtos, é obtida por meio de *web scraping* [5] ou consultas a páginas de lojas e fornecedores. Dessa forma, o funcionamento correto desses recursos depende da disponibilidade e estabilidade dos sites consultados, bem como da manutenção de suas estruturas de página. Alterações significativas nessas fontes externas podem exigir certas adaptações. A aplicação também depende de navegadores modernos compatíveis com HyperText Markup Language (HTML), Cascading Style Sheets (CSS) [3] e JavaScript [4], não sendo garantida a execução adequada em navegadores desatualizados ou sem suporte a tais tecnologias.

### 3.1.3 Benefícios

A Setup Smart oferece uma série de benefícios para diferentes perfis de usuários. Pessoas leigas em informática podem compreender melhor os componentes de um computador e identificar qual configuração atende às suas atividades diárias, reduzindo a probabilidade de adquirir equipamentos inadequados ou com desempenho inferior ao esperado. Usuários intermediários e avançados encontram na plataforma uma forma prática de comparar componentes e consultar preços atualizados. Empresas que necessitam adquirir múltiplos computadores podem utilizar as recomendações para otimizar custos e padronizar configurações conforme suas demandas internas. De maneira geral, a aplicação contribui para decisões de compra mais informadas, econômicas e alinhadas às necessidades reais, tornando o processo mais acessível e eficiente para todos os tipos de usuários.

## 3.2 Análise de requisitos

### 3.2.1 Metodologia

Para o levantamento dos requisitos da Setup Smart, foram realizadas análises preliminares do problema, identificação do público-alvo e avaliação das principais dificuldades enfrentadas por usuários na escolha de configurações de computador. Entre as principais dificuldades identificadas estão a seleção adequada de componentes de acordo com o desempenho desejado, a verificação de compatibilidade entre peças e a avaliação do custo-benefício das possíveis combinações de hardware.

O processo de elicitação seguiu princípios apresentados por Wiegers (2013) [34], que enfatiza a importância de compreender o problema de forma ampla, identificar claramente as necessidades dos usuários e revisar continuamente os requisitos conforme novas demandas surgem durante o desenvolvimento. Essa abordagem se mostrou adequada ao projeto por envolver um sistema em constante refinamento, especialmente devido à inclusão progressiva de funcionalidades como o questionário orientado e a consulta de preços em tempo real.

As etapas envolveram discussões sobre as funcionalidades essenciais da aplicação, estudo de ferramentas semelhantes e prototipagem inicial das telas e fluxos de uso. Durante o desenvolvimento, novos requisitos foram incorporados conforme necessidades práticas surgiram, especialmente relacionadas ao questionário orientado e à consulta de preços em tempo real.

### 3.2.2 Classes de usuários

- **Cliente (usuário comum) não logado:** Tem acesso à página inicial com os artigos educativos, ao catálogo de produtos bem como suas especificações detalhadas e ao questionário interativo. Pode se cadastrar e logar na aplicação.
- **Cliente (usuário comum) logado:** Tem acesso aos mesmos privilégios de um usuário comum, podendo acessar suas informações pessoais de cadastro, alterar sua senha e adicionar produtos à sua lista de favoritos.
- **Administrador:** Possui permissões de administrador, podendo cadastrar, editar ou excluir produtos e usuários.

### 3.2.3 Lista de funcionalidades

As funcionalidades desta aplicação e os atores estão exibidas na Tabela 2:

Funcionalidade(s)	Usuário(s)
Acessar conteúdos informativos e artigos	Qualquer usuário
Visualizar produtos	Qualquer usuário
Ver detalhes de um produto	Qualquer usuário
Favoritar um produto	Cliente logado
Criar uma conta	Cliente não logado
Fazer login	Cliente não logado e Administrador
Acessar perfil	Cliente logado
Alterar senha	Cliente logado
Responder questionário	Qualquer usuário
Acessar painel do administrador	Administrador
Cadastrar, editar ou excluir um usuário	Administrador
Cadastrar, editar ou excluir um produto	Administrador
Cadastrar, editar ou excluir imagens	Administrador
Atualizar preço de produtos	Administrador

**Tabela 2: funcionalidades por nível de acesso da aplicação**

### 3.3 Especificação dos requisitos de software

Os requisitos da Setup Smart são divididos em Requisitos Funcionais e Requisitos Não Funcionais. Os Requisitos Funcionais descrevem as ações e comportamentos que devem obrigatoriamente ser oferecidos ao usuário, enquanto os Requisitos Não Funcionais definem características de qualidade necessárias para garantir uma boa experiência de uso e o correto funcionamento da aplicação.

#### 3.3.1 Requisitos Funcionais (RF)

- **RF01.** A aplicação deve disponibilizar artigos informativos sobre *hardware*, permitindo que o usuário aprenda conceitos básicos e avançados sobre componentes de computador.
- **RF02.** A aplicação deve permitir que o usuário visualize e pesquise componentes diversos, incluindo processadores, placas mãe, memórias, placas de vídeo, fontes, SSD (Solid-State Drive), gabinetes, monitores e *laptops*.

- **RF03.** A aplicação deve permitir que o usuário acesse a página específica de cada produto e visualize detalhes técnicos, descrições e o menor preço atualizado obtido por *web scraping*.
- **RF04.** O usuário pode criar uma conta na aplicação fornecendo nome, e-mail, senha, gênero, data de nascimento, profissão, cidade, estado e país onde mora.
- **RF05.** O usuário pode realizar login na aplicação com seu e-mail e senha cadastrados
- **RF06.** A aplicação deve permitir que o usuário adicione ou remova produtos da sua lista de favoritos.
- **RF07.** A aplicação deve permitir que o usuário visualize, edite e pesquise dentro da sua lista de favoritos.
- **RF08.** A aplicação deve permitir que o usuário visualize suas informações de cadastro e altere sua senha.
- **RF09.** A aplicação deve permitir que o usuário responda a um questionário destinado a identificar suas necessidades e preferências de *hardware*.
- **RF10.** A aplicação deve processar as informações coletadas pelo questionário e gerar automaticamente uma recomendação de configuração para computador ou laptop, exibindo os produtos sugeridos e orientações adicionais sobre *hardware*.
- **RF11.** A aplicação deve operar integrado ao banco de dados para armazenar, atualizar e recuperar informações sobre produtos, usuários e favoritos.

## Requisitos do Administrador

- **RF11.** A aplicação deve permitir que o usuário administrador acesse o Painel do Administrador para gerenciar a aplicação.
- **RF12.** A aplicação deve permitir que o administrador gerencie usuários, incluindo cadastro, edição e exclusão.
- **RF13.** A aplicação deve permitir que o administrador gerencie produtos, incluindo cadastro, edição e exclusão.
- **RF14.** A aplicação deve permitir que o administrador gerencie imagens vinculadas aos produtos.
- **RF15.** A aplicação deve permitir que o administrador atualize os preços de todos os produtos de uma categoria simultaneamente.

### 3.3.2 Requisitos Não Funcionais (RNF)

- **RNF01 Desempenho.** A aplicação deve responder a consultas, buscas e geração de recomendações de forma rápida e eficiente, minimizando tempos de carregamento. Para isso, adota-se como referência os indicadores de desempenho definidos pelo Google Web Vitals [35], que consideram um carregamento rápido quando o *Largest Contentful Paint* (LCP) ocorre em até 2,5 segundos, desempenho aceitável entre 2,5 e 4 segundos e lento acima de 4 segundos. Esses parâmetros servem como base para garantir uma experiência de uso adequada e compatível com padrões modernos de desempenho.
- **RNF02 Usabilidade.** A interface deve ser intuitiva e de fácil navegação, permitindo que usuários leigos compreendam seu funcionamento sem necessidade de conhecimento técnico prévio. Para isso, a aplicação utiliza um design limpo e organizado, com layout não poluído e termos acessíveis, evitando o uso de jargões técnicos. Além disso, botões e elementos interativos fazem uso de imagens e ícones que facilitam a identificação das ações disponíveis, contribuindo para que o usuário encontre rapidamente o que procura sem se sentir perdido durante a navegação.

- **RNF03 Responsividade.** A aplicação deve ser totalmente responsivo, garantindo boa usabilidade em diferentes tamanhos de tela e dispositivos, incluindo computadores, tablets e smartphones.
- **RNF04 Compatibilidade.** A aplicação deve ser compatível com navegadores modernos que suportem HTML5, CSS e JavaScript, não sendo garantido o funcionamento adequado em versões desatualizadas.
- **RNF05 Confiabilidade.** A aplicação deve garantir a integridade e consistência dos dados armazenados no banco de dados durante operações de leitura, escrita e atualização.
- **RNF06 Segurança.** A aplicação deve implementar boas práticas de segurança, protegendo dados de usuários e evitando acessos não autorizados, especialmente em operações de login e gerenciamento administrativo. Para isso, o sistema utiliza diferentes níveis de acesso e verificações de permissão que restringem funcionalidades conforme o perfil do usuário. Além disso, as credenciais são armazenadas de forma segura por meio do uso de *hash*, garantindo que senhas não sejam salvas em texto puro e reduzindo riscos de exposição em caso de violação. Essas medidas contribuem para a integridade, confidencialidade e proteção geral das informações tratadas pela aplicação.
- **RNF07 Disponibilidade.** A aplicação deve permanecer acessível durante a maior parte do tempo, considerando que funcionalidades de consulta de preços dependem da disponibilidade de lojas e serviços externos.
- **RNF08 Tratamento de falhas externas.** Quando sites de terceiros estiverem indisponíveis ou retornarem informações incompletas para *web scraping*, devem ser exibidas mensagens informativas e lidar com a situação sem comprometer o funcionamento geral.
- **RNF09 Manutenibilidade.** A aplicação deve ser estruturada de forma modular, facilitando a manutenção, atualização de componentes, expansão de funcionalidades e ajustes futuros.
- **RNF10 Escalabilidade.** A aplicação deve permitir aumento gradual no volume de dados, produtos e acessos simultâneos, sem comprometer o desempenho.

## 4. Solução Implementada

### 4.1 Implementação

O desenvolvimento da aplicação web Setup Smart foi dividido em duas partes principais: front-end e back-end. A seguir, entraremos em maiores detalhes a respeito de cada uma destas etapas, abordando tecnologias utilizadas e funcionalidades presentes no projeto.

#### 4.1.1 Front-end

O front-end da Setup Smart foi desenvolvido utilizando React [6], uma biblioteca JavaScript voltada para a criação de interfaces web dinâmicas e reutilizáveis. A escolha pelo React se deve à sua capacidade de escalabilidade, ao modelo baseado em componentes e ao bom desempenho na atualização da interface. Conforme apresentado por Banks (2020) [36], o React estrutura a interface em componentes independentes que podem ser reutilizados em diferentes partes da aplicação, facilitando a manutenção e reduzindo redundâncias. Além disso, o uso do Virtual DOM permite atualizar apenas os elementos necessários, tornando o processo de renderização mais eficiente em comparação ao JavaScript puro. Essa abordagem é complementada pelo uso de estados reativos, que controlam quais partes da página devem ser re-renderizadas a cada alteração.

A arquitetura do front-end foi organizada em dois grupos principais: páginas e componentes. Os componentes consistem em unidades reutilizáveis presentes em diferentes áreas da aplicação, como barra de navegação, rodapé, botões, cards, formulários e outros elementos visuais. Já as páginas representam seções completas da aplicação, responsáveis por estruturar a interface e processar as interações do usuário.

A comunicação entre front-end e back-end segue o modelo cliente-servidor por meio de requisições HTTP. Essa estrutura é fundamentada nos princípios do estilo arquitetural REST descritos por Fielding (2000) [37], no qual recursos são acessados por operações padronizadas sobre o protocolo HTTP. Assim, a aplicação mantém uma divisão clara entre responsabilidades e garante uma interação eficiente entre interface e serviços.

A estilização do projeto foi realizada integralmente em CSS, permitindo manipular cores, fontes, espaçamentos, alinhamentos, transições e efeitos visuais. Para garantir melhor organização e evitar conflitos de classes, foi utilizada a técnica de modularização através de CSS Modules [7], na qual cada arquivo `.module.css` possui um escopo isolado. Com isso, cada classe é automaticamente transformada em um identificador único, facilitando a manutenção e prevenindo sobreposições indesejadas.

O código do front-end foi hospedado na plataforma Vercel [8], que oferece suporte nativo a aplicações React e integra-se diretamente ao GitHub [9]. A plataforma é estável, gratuita para projetos pessoais e extremamente eficiente na implantação contínua, sendo uma opção ideal para aplicações *web* modernas.

A utilização do React contribuiu para uma navegação mais fluida, carregamentos rápidos e maior compatibilidade com os principais navegadores, como Google Chrome, Firefox, Safari e Opera. Além disso, foram adotados conceitos de *mobile-first* [10] durante o desenvolvimento, assegurando que a aplicação apresentasse boa usabilidade em dispositivos móveis, que representam grande parte do público que acessa a aplicação.

#### 4.1.2 Back-end

O back-end da Setup Smart foi desenvolvido utilizando Node.js [11], adotando o *framework* Express [12] como base para a construção da Application Programming Interface (API) responsável por conectar o front-end ao banco de dados e às funcionalidades principais.

O Node.js foi escolhido para o desenvolvimento do back-end por oferecer alto desempenho e eficiência na execução de operações assíncronas, graças ao seu modelo de processamento não bloqueante e ao motor V8 do Google. Além disso, por utilizar JavaScript no servidor, o desenvolvimento *full stack* torna-se mais simples e integrado ao restante da aplicação. Seu ecossistema amplo, disponibilizado pelo Node Package Manager (npm) [13], fornece milhares de bibliotecas que aceleram o desenvolvimento e facilitam integrações com serviços externos. O Node.js também se destaca pela escalabilidade e pela capacidade de lidar com um grande volume de requisições simultâneas, características essenciais para aplicações *web* modernas.

A arquitetura do back-end foi organizada em alguns grupos principais: *controllers*, *routes* e *services*. As *routes* são responsáveis por definir os *endpoints* acessíveis ao front-end, mapeando cada caminho da API para a funcionalidade correspondente. Já os *controllers* executam efetivamente a lógica associada a cada rota, recebendo dados enviados pelo usuário, interagindo com o banco de dados, realizando validações e retornando respostas estruturadas

em formato JavaScript Object Notation (JSON). Os *services* ficam encarregados de lidar com toda a lógica do questionário e realizar a busca de preços via *web scraping*.

#### 4.1.2 Questionário

O questionário constitui a funcionalidade central da Setup Smart, sendo responsável por coletar informações sobre o perfil de uso e as necessidades do usuário para gerar uma recomendação personalizada de computador ou laptop. O processo inicia-se no front-end, onde o usuário responde a uma sequência de perguntas de múltipla escolha, todas obrigatórias. Ao término do questionário, todas as respostas são enviadas ao back-end agrupadas em um único objeto.

O back-end recebe as respostas, percorre todas uma a uma e, para cada pergunta, atribui níveis de exigência para diferentes componentes do computador. Esses níveis funcionam como indicadores abstratos de desempenho. Por exemplo, uma resposta que sugira uso profissional intenso pode atribuir um nível mais alto ao processador, memória RAM e placa de vídeo, enquanto um uso mais leve atribuiria níveis menores. Após todos os níveis serem definidos, o sistema os converte em valores concretos de requisitos mínimos, como quantidade de memória RAM, potência mínima da fonte, desempenho mínimo da placa de vídeo, resolução ideal do monitor, etc.

Um ponto fundamental dessa lógica é que nenhum requisito mínimo pode ser reduzido ao longo do processo. Isso significa que, uma vez que um componente atinge determinado nível necessário, respostas posteriores nunca podem gerar uma exigência inferior. Esse mecanismo garante que a recomendação final nunca fique abaixo da necessidade real indicada pelo usuário.

Com o objeto de requisitos mínimos montado, inicia-se a etapa de busca no banco de dados. No caso de notebooks, o sistema procura diretamente um modelo que atenda a todos os requisitos mínimos calculados. Para computadores de mesa, a seleção ocorre componente por componente, seguindo uma ordem lógica que evita incompatibilidades técnicas e garante coerência na montagem final.

A ordem de busca dos componentes é estruturada da seguinte forma:

- 1. Processador** – É o primeiro item selecionado porque define o desempenho base do sistema e determina a compatibilidade com a placa-mãe e com o tipo de memória RAM.
- 2. Placa-mãe** – Escolhida exclusivamente entre modelos compatíveis com o socket do processador. Sua arquitetura também influencia o tipo de memórias que podem ser utilizadas, assim como o formato suportado pelo gabinete.
- 3. Memória RAM** – Deve ser compatível tanto com a placa-mãe quanto com o processador. A quantidade recomendada deriva diretamente do nível de desempenho identificado nas respostas do usuário. Também determina a compatibilidade com o gabinete.
- 4. Placa de vídeo** – Selecionada independentemente dos componentes anteriores, mas baseada no nível mínimo de desempenho calculado. Para classificar o desempenho da GPU, utiliza-se uma métrica de desempenho relativo obtida no TechPowerUp [14]. A placa de vídeo, por sua vez, influencia diretamente na escolha da fonte e do gabinete.

**5. Fonte de alimentação** – Sua potência recomendada é determinada principalmente pelo consumo da placa de vídeo.

**6. Gabinete** – Deve ser compatível com o formato da placa-mãe (por exemplo, ATX, Micro-ATX) e com o comprimento da placa de vídeo.

**7. SSD** – Selecionado separadamente, exigindo apenas compatibilidade com os padrões aceitos pela placa-mãe. Como a maioria das placas modernas suporta os formatos mais comuns, essa etapa não costuma ser restritiva.

**8. Monitor** – Escolhido independentemente, considerando principalmente a resolução e o nível de qualidade visual definidos como necessários pelo usuário.

Sempre que duas ou mais opções atendem aos requisitos mínimos, o sistema seleciona a de menor preço, garantindo uma recomendação eficiente e economicamente vantajosa. Ao final do processo, o back-end retorna ao front-end um objeto contendo todos os componentes recomendados, concluindo assim o fluxo completo do questionário.

O questionário é constituído por 12 perguntas de múltipla escolha, são elas:

**Pergunta 1:** Define se o usuário receberá uma recomendação de computador de mesa (com componentes individuais) ou um notebook. Essa escolha altera completamente o fluxo de seleção no back-end.

Pergunta 1 de 12

**Você precisa do desempenho bruto de um computador de mesa ou a praticidade de um notebook?**



Computador de mesa, vou usar só em casa mesmo



Notebook, preciso levar ele comigo quando sair

**Pergunta 2:** Determina o nível de desempenho necessário para uso profissional. Afeta diretamente requisitos mínimos de processador, memória RAM, placa de vídeo, armazenamento e qualidade do monitor.

Pergunta 2 de 12

**Com que tipo de softwares ou serviços você trabalha? (Escolha algo que se aproxime daquilo que você faz mesmo que não esteja na lista)**



Escritório, navegador, Documentos e Softwares leves



Arquitetura, Engenharia, Edição de fotos ou Desenvolvimento de Software



Edição de vídeos, Inteligência artificial e aplicações mais pesadas

**Pergunta 3:** Define o nível de desempenho necessário para jogos. Ajusta requisitos de processador, placa de vídeo, memória RAM, armazenamento e monitor com base no perfil de jogos do usuário.

Pergunta 3 de 12

**O que você gosta de jogar?**



Não gosto muito de jogos



Jogos mais leves de forma casual



Jogos competitivos



Jogos pesados com boa qualidade gráfica

**Pergunta 4:** Determina a quantidade mínima de memória RAM recomendada de acordo com o nível de multitarefa realizado pelo usuário.

Pergunta 4 de 12

**Você costuma utilizar muitas coisas ao mesmo tempo enquanto mexe no computador?**



Apenas o básico, uma ou duas abas do chrome no máximo



Sim, adoro jogar com o spotify aberto ao fundo



Com certeza, uso softwares pesados e deixo várias coisas rodando em segundo plano

**Pergunta 5:** Define a quantidade de armazenamento necessário, variando de usos básicos a cenários que exigem grande volume de espaço.

Pergunta 5 de 12

**Armazenamento/espaco é importante para você?**



Não preciso de muito espaço, apenas o básico



Quero uma margem pra trabalhar e instalar um joguinhos



Espaço é sempre bom, costumo instalar uns joguinhos mais pesados de vez em quando



Preciso de bastante armazenamento

**Pergunta 6:** Determina a resolução adequada do monitor, com base no quanto o usuário valoriza nitidez e qualidade de imagem.

Pergunta 6 de 12

**A resolução de imagem é importante pra você? (Quanto maior a resolução da tela, mais nítida é a imagem.)**



POV: YOU SET YOU VIDEO  
QUALITY TO 144P

Não muito, quero o que estiver mais baratinho



Sim, gostaria de algo com um pouco mais de qualidade



Muito, trabalho com fotos, vídeos e imagens de alta resolução

**Pergunta 7:** Define a fidelidade de cores necessária no monitor, influenciando desde opções simples até modelos profissionais.

Pergunta 7 de 12

**Fidelidade de cores é algo que impacta no seu trabalho ou na sua gameplay?**



Não me importo, quero algo barato apenas



Um pouco, quero algo com um pouco mais de qualidade



Sim, gosto de cores fíeis e bonitas para jogar ou trabalhar



Preciso da melhor qualidade e fidelidade de cores possível, independente do preço

**Pergunta 8:** Avalia se o ambiente é quente, permitindo recomendar soluções de resfriamento mais robustas para melhorar a durabilidade e o desempenho do sistema.

Pergunta 8 de 12

**No lugar onde você mora faz muito calor ou o ambiente em que seu computador irá ficar é muito quente?**



Não, é bem tranquilo



Mais ou menos, de vez em quando faz um pouco de calor



Sim, aqui faz bastante calor

**Pergunta 9:** Define recomendações relacionadas ao uso de webcam, variando entre modelos básicos, intermediários ou avançados conforme a necessidade do usuário.

Pergunta 9 de 12

**Você costuma gravar vídeos, fazer lives ou participar de reuniões?**



Não é algo muito relevante pra mim



Participo de algumas reuniões na faculdade ou trabalho



Sim, pretendo gravar alguns conteúdos ou fazer lives

**Pergunta 10:** Determina o nível de qualidade dos periféricos recomendados (mouse, teclado, mousepad, fone e microfone), com opções simples ou de maior desempenho.

Pergunta 10 de 12

**Sobre periféricos (mouse, teclado, mousepad, fone e microfone):**



Quero algo mais simples e barato



Quero algo de mais qualidade para jogar ou trabalhar

**Pergunta 11:** Define o sistema operacional preferido pelo usuário. Para notebooks, determina o sistema do modelo recomendado; para desktops, orienta sobre como obter o sistema escolhido.

Pergunta 11 de 12

**Qual sistema operacional você mais gosta?**



Windows



Linux

**Pergunta 12:** Define a plataforma recomendada para o computador. Caso o usuário deseje realizar upgrades futuramente, recomenda-se uma plataforma mais recente e com maior longevidade; caso contrário, prioriza-se uma opção mais acessível.

Pergunta 12 de 12

**Você pretende fazer upgrades a curto/médio prazo?**



Não, quero algo para usar agora mesmo



Sim, pretendo fazer alguns upgrades e preciso de longevidade

Ao final desse processo, o back-end retorna ao front-end um objeto contendo todos os componentes recomendados, concluindo o fluxo completo do questionário.

#### 4.1.3 *Web Scraping* e preços

*Web scraping* é uma técnica utilizada para extrair informações automaticamente de páginas da web, automatizando o acesso às páginas, a extração e o processamento dos dados em um formato estruturado para uso posterior (SAURKAR, PATHARE e GODE, 2018) [38]. Na Setup Smart, essa técnica é empregada para coletar os preços dos produtos cadastrados a partir de grandes lojas do varejo brasileiro, como Kabum, Pichau, Terabyte, Amazon, Magazine Luiza, Carrefour, Acer, Lenovo, Avell e Dell.

Na aplicação, todos os produtos cadastrados possuem as URLs das lojas onde podem ser encontrados. Assim, a partir de cada URL é possível acessar diretamente a página correspondente ao produto e extrair o preço exibido. Durante o cadastro, o administrador informa esses endereços, que passam a servir como pontos de coleta durante o processo de atualização de valores.

Para implementar a função de web scraping, é necessário analisar previamente o código-fonte da página de cada produto e identificar em qual elemento HTML o preço é exibido. Uma vez localizado o seletor correto, o sistema consegue extrair o valor desejado. Como cada loja utiliza estruturas, tecnologias e mecanismos distintos (incluindo proteções contra automações), foi necessário desenvolver uma lógica de extração específica para cada uma delas. Em alguns casos, como no site da Pichau, torna-se indispensável simular o comportamento de um usuário real para contornar verificações e permitir o carregamento adequado das informações, o que exige o uso de ferramentas capazes de executar páginas dinâmicas.

Para garantir que os preços exibidos ao usuário permaneçam atualizados, a aplicação permite executar o web scraping em lote para todos os produtos pertencentes a uma mesma categoria. No back-end, essa atualização ocorre por meio de processamento concorrente. Para cada produto, as requisições destinadas às diversas lojas são iniciadas simultaneamente por meio de Promises [39], permitindo que a extração dos preços aconteça em paralelo. Após a conclusão de todas as operações assíncronas, os resultados são consolidados e o menor preço é determinado.

Além disso, o sistema utiliza uma estratégia de paralelismo controlado para atualizar vários produtos de forma eficiente. Em vez de executar todas as extrações simultaneamente (o que poderia sobrecarregar o servidor e comprometer a estabilidade), as atualizações são divididas em grupos processados paralelamente. Cada grupo contém uma quantidade fixa de produtos, cujas operações assíncronas são executadas em conjunto, enquanto os demais aguardam sua vez. Esse mecanismo, baseado em lotes de Promises, equilibra desempenho e uso eficiente dos recursos, permitindo maior velocidade de atualização sem risco de instabilidade na aplicação.

#### 4.1.4 Banco de Dados

Para o armazenamento de dados, foi utilizado o MySQL [15], um sistema de gerenciamento de banco de dados relacional *open-source* amplamente adotado devido à sua confiabilidade, desempenho e capacidade de escalabilidade. A comunicação entre o MySQL e o back-end desenvolvido em Node.js é realizada por meio da biblioteca mysql2 [16], utilizando um *pool* de conexões configurado com variáveis de ambiente para garantir segurança e flexibilidade.

O banco de dados foi estruturado com uma tabela específica para cada tipo de componente, além de tabelas auxiliares relacionadas a usuários, favoritos e imagens. As tabelas principais são:

- **processador**: armazena as especificações dos processadores
- **memoria**: armazena as especificações das memórias RAM
- **placa\_mae**: armazena as especificações das placas-mãe
- **placa\_de\_video**: armazena as especificações das placas de vídeo
- **fonte**: armazena as especificações das fontes de alimentação
- **ssd**: armazena especificações de unidades de armazenamento
- **gabinete**: armazena as especificações dos gabinetes
- **monitor**: armazena as especificações dos monitores
- **notebook**: armazena as especificações dos laptops
- **usuarios**: registra os dados dos usuários cadastrados
- **favoritos**: guarda a relação entre usuários e produtos favoritados
- **imagens**: armazena as URLs públicas das imagens

A tabela **favoritos** armazena o *id* do usuário, o *id* do produto e o nome da tabela à qual esse produto pertence, permitindo associar qualquer item cadastrado à lista de favoritos de um usuário. Já a tabela **imagens** registra apenas a URL da imagem, retornada pelo serviço externo Cloudinary [17], além do *id* do produto e da tabela correspondente. O fluxo de armazenamento é o seguinte: o usuário realiza o upload pela aplicação, o arquivo é enviado ao Cloudinary, que retorna uma URL pública, e essa URL é então salva no banco de dados.

Tanto a tabela **favoritos** quanto a tabela **imagens** fazem uso de chaves estrangeiras, pois dependem do identificador do usuário ou do produto ao qual estão vinculadas.

Por fim, todo o banco de dados está hospedado na nuvem através da plataforma Railway [18], que oferece infraestrutura confiável e de fácil integração com aplicações Node.js, garantindo disponibilidade e acesso remoto durante todo o desenvolvimento da aplicação.

A Figura 1 representa graficamente a estrutura do banco de dados, enquanto a figura 2 mostra como funciona o armazenamento de imagens.

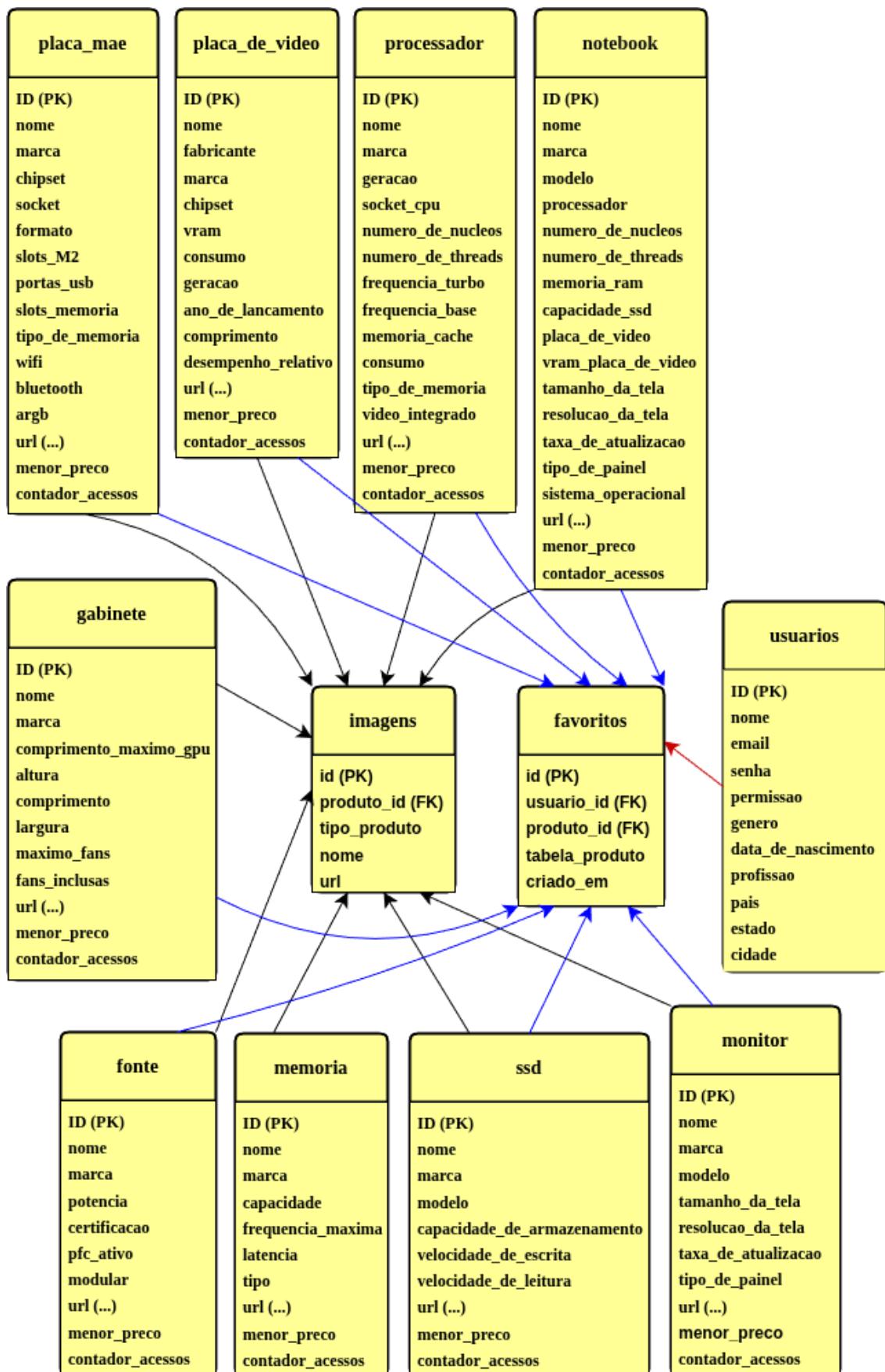
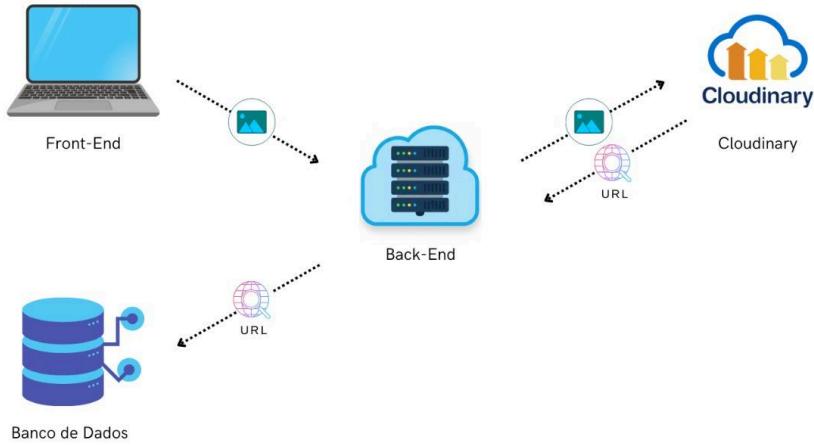


Figura 1: estrutura e tabelas do banco de dados



**Figura 2: Fluxo de dados do armazenamento de imagens usando o Cludinary**

#### 4.2 Login

Os usuários podem criar uma conta, embora o cadastro não seja obrigatório. Algumas funcionalidades, como favoritar itens e acessar a lista de favoritos, estão disponíveis apenas para usuários autenticados. No momento do cadastro, a senha é criptografada utilizando bcrypt [19], sendo armazenada no banco de dados em formato *hash* para garantir segurança e integridade dos dados.

A autenticação é realizada por meio de JSON Web Tokens (JWT) [20], permitindo identificar o usuário logado e determinar suas permissões dentro da aplicação. Esse modelo garante um controle de acesso seguro e eficiente ao longo de toda a navegação.

#### 4.3 Ambiente de desenvolvimento e ferramentas

O ambiente de desenvolvimento utilizado para a implementação da aplicação foi o Visual Studio Code [21], uma Integrated Development Environment (IDE) gratuita e multiplataforma desenvolvida pela Microsoft. Ela oferece grande variedade de extensões, possui excelente desempenho e é compatível com praticamente todas as linguagens e tecnologias modernas, além de contar com integração nativa com o GitHub, facilitando o versionamento do código.

Para o gerenciamento do banco de dados MySQL, foi utilizada a ferramenta MySQL Workbench [22], que permite criar, visualizar e administrar tabelas, executar consultas e monitorar o estado do banco de dados de maneira prática e eficiente.

Foi utilizado o GitHub Desktop [23], que oferece uma interface simples e amigável para operações comuns de versionamento, como criação de repositórios, *commits* e sincronização com a nuvem. Embora execute internamente as mesmas operações disponíveis pelo terminal, sua interface visual torna o fluxo de trabalho mais intuitivo e acessível.

#### 4.4 Bibliotecas e Dependências

Durante o desenvolvimento da aplicação, diversas bibliotecas e dependências foram utilizadas tanto no front-end quanto no back-end. A seguir, estão listadas as principais, acompanhadas de uma breve descrição de suas funções:

##### Front-end

- **react** – Biblioteca base utilizada para construção das interfaces web de forma dinâmica e componentizada.
- **react-dom** – Responsável pela renderização dos componentes React no navegador [24].
- **react-router-dom** – Gerencia as rotas e a navegação entre páginas, permitindo a criação de uma Single Page Application (SPA) [25].
- **react-hook-form** – Ferramenta para criação e gerenciamento de formulários com maior desempenho e organização [26].
- **axios** – Utilizado para realizar requisições HTTP ao back-end [27].
- **react-scripts** – Scripts essenciais fornecidos pelo Create React App, responsáveis por build, testes e execução do projeto [28].

##### Back-end

- **express** – *Framework* principal da API, responsável pela definição de rotas, utilização de middlewares e gerenciamento de requisições HTTP.
- **mysql2** – Permite estabelecer conexão com o banco de dados MySQL e executar consultas SQL.
- **dotenv** – Carrega variáveis de ambiente a partir de um arquivo `.env`, garantindo segurança e flexibilidade [29].
- **cors** – Permite que o servidor aceite requisições provenientes do front-end, mesmo em domínios diferentes [30].
- **axios** – Utilizado também no back-end para realizar requisições externas, especialmente no processo de *web scraping*.
- **cheerio** – Biblioteca para *web scraping* em páginas estáticas (HTML puro), semelhante ao jQuery no lado do servidor [31].
- **puppeteer** – Ferramenta de *web scraping* avançado, capaz de controlar um navegador real (*Headless Chrome*) para extrair dados de páginas dinâmicas [32].
- **bcrypt** – Utilizado para criptografia de senhas, garantindo segurança ao armazená-las no banco de dados.
- **jsonwebtoken** – Implementa autenticação por meio de JWT, permitindo validar usuários e permissões.
- **multer** – Responsável pelo upload de arquivos (como imagens) enviados ao back-end [33].
- **cloudinary** – Software Development Kit (SDK) utilizado para armazenamento e gerenciamento de imagens na nuvem, retornando URLs públicas.

## 4.5 Testes

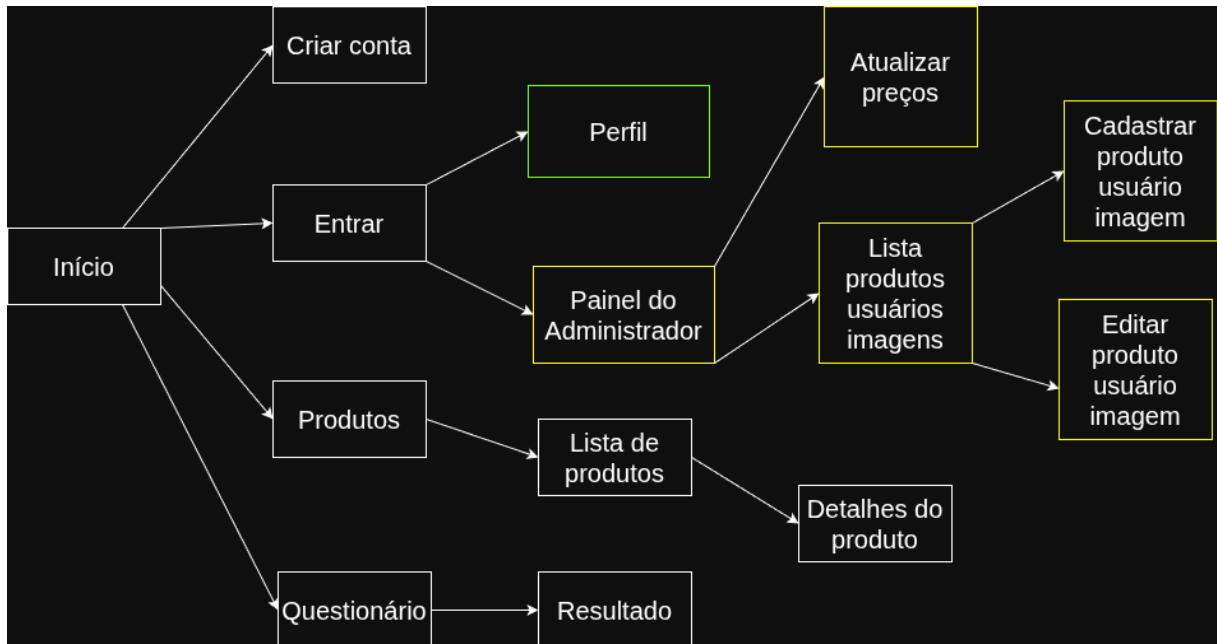
Durante o desenvolvimento da aplicação, foram realizados testes básicos para garantir o correto funcionamento das principais funcionalidades. No back-end, o Postman [34] foi utilizado para validar manualmente as rotas da API, permitindo verificar respostas, códigos de status e o comportamento da autenticação. Esses testes englobam operações como criação, listagem, atualização e remoção de registros, além da verificação do fluxo de login e autorização por token JWT.

No front-end, foram conduzidos testes manuais para assegurar a integridade da navegação entre páginas, o funcionamento dos formulários, a exibição correta dos dados e a responsividade da interface. Também foram observadas métricas gerais de desempenho, como tempo de carregamento inicial e fluidez na renderização dos componentes.

Embora não tenham sido empregados testes automatizados em profundidade, as verificações realizadas ao longo da implementação foram suficientes para garantir a estabilidade da aplicação e o funcionamento consistente das principais funcionalidades.

## 4.6 Diagrama de navegação e páginas da aplicação

O diagrama de navegação da aplicação representa os fluxos principais que o usuário pode percorrer, destacando quais páginas podem ser acessadas por usuários não autenticados, usuários logados como clientes e usuários com permissão de administrador. A aplicação possui variações na barra de navegação de acordo com o estado de autenticação do usuário e seu nível de permissão, garantindo acesso controlado às funcionalidades apropriadas de cada perfil. A figura 3 ilustra o fluxo de navegação da aplicação.



**Figura 3: Fluxo de navegação da aplicação Setup Smart**

## 4.6.1 Início

A página Início funciona como a página inicial da aplicação. Nela, o usuário encontra uma apresentação do site, imagens ilustrativas, textos informativos e links que direcionam para artigos sobre tecnologia e *hardware*. Essa página é pública e pode ser acessada por todos os usuários, independentemente de estarem logados ou não.

## 4.6.2 Criar conta

A página Criar Conta apresenta um formulário para cadastro de novos usuários. É necessário informar, obrigatoriamente, um e-mail e uma senha, havendo também campos opcionais. Esta página só pode ser acessada por usuários não autenticados. Após o cadastro ser concluído com sucesso, o usuário é automaticamente redirecionado para a página de login.

## 4.6.3 Entrar

A página Entrar contém o formulário de autenticação, permitindo que usuários cadastrados façam login utilizando e-mail e senha. Assim como a página de criação de conta, ela só pode ser acessada por usuários que não estejam logados. Após a autenticação, o usuário é redirecionado para a página inicial.

## 4.6.4 Produtos e lista de produtos

A página *Produtos* apresenta categorias de componentes, permitindo que o usuário selecione qual tipo de item deseja visualizar (processadores, memórias, *notebooks*, etc.). Ao escolher uma categoria, o usuário é levado para a página de listagem correspondente, onde todos os itens cadastrados são exibidos. Nessa página é possível realizar buscas por nome ou características, favoritar ou desfavoritar produtos e acessar os detalhes de qualquer item. A página pode ser acessada por qualquer usuário, mas a funcionalidade de favoritar é restrita a usuários autenticados.

## 4.6.5 Detalhes do Produto

Exibe todas as informações de um produto específico, incluindo imagem, preço atualizado e especificações técnicas. Essa página é pública e pode ser acessada por qualquer usuário.

## 4.6.6 Perfil

A página Perfil exibe os dados de cadastro do usuário e sua lista de produtos favoritos. É possível visualizar, buscar e remover itens favoritados, além de acessar a página de detalhes de cada produto. Essa tela é exclusiva para usuários logados com a permissão de cliente.

## 4.6.7 Painel de Administrador

O *Painel de Administrador* é destinado exclusivamente para usuários com permissão administrativa. Nele, o administrador pode visualizar suas informações de cadastro e acessar listagens de produtos, usuários e imagens. A partir dessa página também é possível acessar a funcionalidade de atualização de preços.

#### **4.6.8 Listas**

A página Listas reúne as listas de produtos, usuários e imagens cadastradas. Permite realizar buscas, cadastrar novos itens, editar registros existentes ou excluí-los. Ao selecionar cadastrar ou editar, o administrador é redirecionado para páginas específicas de formulário.

#### **4.6.9 Cadastrar**

Apresenta um formulário para cadastro de novos produtos, usuários ou imagens. Para imagens, o administrador deve selecionar o tipo de produto, associar a imagem ao item correspondente e realizar o upload do arquivo, que será enviado ao serviço de armazenamento externo. Essa página é exclusiva para administradores.

#### **4.6.10 Editar**

Apresenta um formulário semelhante ao de cadastro, porém com os campos preenchidos com os dados já existentes do item selecionado. Isso facilita a atualização de informações pelo administrador. Assim como as demais páginas administrativas, esta página só pode ser acessada por administradores.

#### **4.6.11 Atualizar preços**

Esta página contém botões que atualizam de uma só vez todos os preços de determinada categoria de produtos. A atualização ocorre de forma automática e é guardada no banco de dados. A página só pode ser acessada por administradores.

#### **4.6.12 Questionário e resultado**

Essa página apresenta um questionário interativo composto por perguntas de múltipla escolha. Conforme o usuário responde, avança para a próxima etapa. Ao concluir o questionário, a aplicação processa as respostas e exibe a configuração recomendada na página de resultado, acompanhada de informações complementares. Essa funcionalidade está disponível para qualquer usuário, logado ou não.

## 5. Considerações Finais

A aplicação web Setup Smart foi desenvolvida com o propósito de oferecer uma ferramenta acessível para auxiliar usuários na escolha da melhor solução de *hardware* conforme suas necessidades profissionais e pessoais. A plataforma busca atender diferentes perfis de público, em especial usuários leigos, que na maioria das vezes não possuem conhecimento técnico sobre computadores, ou até mesmo usuários mais experientes que desejam consultar especificações, comparar componentes ou aprofundar seu entendimento sobre *hardware*. Assim, a Setup Smart contribui tanto para a aprendizagem quanto para a tomada de decisão, demonstrando que é possível adquirir um computador adequado gastando o mínimo necessário.

A Setup Smart tem potencial para auxiliar em diversas situações práticas: apoiar pais na compra de um computador gamer para seus filhos, orientar um professor na escolha de um laptop para trabalho, ajudar empresas na renovação de seus equipamentos ou até mesmo servir como referência entre amigos que buscam recomendações de *hardware* confiáveis. A aplicação foi construída com o objetivo de proporcionar ao usuário uma compra segura, coerente e economicamente vantajosa, assegurando que todos os componentes recomendados sejam compatíveis entre si e adequados ao perfil informado.

Em suma, a aplicação cumpre seu papel de orientar, educar e facilitar o acesso à informação, contribuindo para que cada usuário possa realizar escolhas inteligentes, compatíveis com seu uso real e com o melhor custo-benefício possível.

## 6. Trabalhos Futuros

Assim como qualquer aplicação em desenvolvimento contínuo, a Setup Smart possui amplo espaço para expansão e aprimoramento. Entre as funcionalidades que podem ser implementadas em versões futuras, destacam-se algumas propostas relevantes.

Uma possibilidade é a criação de uma seção dedicada à recomendação de smartphones, considerando que muitos usuários têm dificuldade em compreender especificações técnicas e diferenças de desempenho entre modelos. Essa funcionalidade ampliaria o escopo da plataforma e alcançaria um público ainda maior.

Outra melhoria potencial envolve a implementação de um sistema de comentários e avaliações de produtos, permitindo que usuários compartilhem suas experiências e opiniões. O feedback da comunidade pode ser um fator decisivo no processo de escolha de *hardware*, agregando valor às recomendações fornecidas pela aplicação.

Além disso, a inclusão de novos tipos de componentes, como coolers, periféricos (mouse, teclado, fone, webcam, microfone) e acessórios em geral, contribuiria para tornar a aplicação ainda mais completa e abrangente. Da mesma forma, a expansão da seção de artigos educativos, abordando novos temas e aprofundando conteúdos já existentes, pode fortalecer o caráter informativo da plataforma.

Tais melhorias demonstram o potencial de evolução da Setup Smart e reforçam seu objetivo de oferecer uma solução cada vez mais completa, educativa e útil para diferentes perfis de usuários.

## Referências

- [1] MEUPC.NET. <https://meupc.net/>. Acesso em 30 nov. 2025.
- [2] PC Build Wizard <https://x.com/pcbuildwizard?lang=pt>. Acesso em 30 nov. 2025.
- [3] CSS Mozilla Developer Network <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em 30 nov. 2025.
- [4] JavaScript Mozilla Developer Network <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em 30 nov. 2025.
- [5] O guia definitivo para Web Scraping com Node.js <https://www.freecodecamp.org/portuguese/news/o-guia-definitivo-para-web-scraping-com-nodejs/>. Acesso em 30 nov. 2025.
- [6] React <https://react.dev/>. Acesso em 30 nov. 2025.
- [7] Por que utilizar css-modules no React <https://dev.to/gmantovani97/porque-utilizar-css-modules-no-react-1a93>. Acesso em 30 nov. 2025.
- [8] Vercel <https://vercel.com/>. Acesso em 30 nov. 2025.
- [9] GitHub <https://github.com/?locale=pt-BR>. Acesso em 30 nov. 2025.
- [10] Mobile First: o conceito e sua aplicabilidade <https://blog.apiki.com/mobile-first-o-conceito-e-sua-aplicabilidade/>. Acesso em 30 nov. 2025.
- [11] Documentação Node.js <https://nodejs.org/docs/latest/api/>. Acesso em 30 nov. 2025.
- [12] Express - Node.js web application framework <https://expressjs.com/>. Acesso em 30 nov. 2025.
- [13] Documentação npm <https://docs.npmjs.com/>. Acesso em 30 nov. 2025.
- [14] GPU Database TechPowerUp <https://www.techpowerup.com/gpu-specs/>. Acesso em 30 nov. 2025.
- [15] Documentação MySQL <https://dev.mysql.com/doc/>. Acesso em 30 nov. 2025.
- [16] mysql2 <https://www.npmjs.com/package/mysql2>. Acesso em 30 nov. 2025.
- [17] Cloudinary <https://cloudinary.com/>. Acesso em 30 nov. 2025.
- [18] Railway <https://railway.com/>. Acesso em 30 nov. 2025.
- [19] bcrypt <https://www.npmjs.com/package/bcrypt>. Acesso em 30 nov. 2025.
- [20] jsonwebtoken <https://www.npmjs.com/package/jsonwebtoken>. Acesso em 30 nov. 2025.
- [21] Visual Studio Code <https://code.visualstudio.com/>. Acesso em 30 nov. 2025.

- [22] MySQL Workbench <https://www.mysql.com/products/workbench/>. Acesso em 30 nov. 2025.
- [23] GitHub Desktop <https://desktop.github.com/download/>. Acesso em 30 nov. 2025.
- [24] ReactDOM <https://pt-br.legacy.reactjs.org/docs/react-dom.html>. Acesso em 30 nov. 2025.
- [25] React Router <https://reactrouter.com/>. Acesso em 30 nov. 2025.
- [26] React Hook Form <https://react-hook-form.com/>. Acesso em 30 nov. 2025.
- [27] Axios <https://axios-http.com/ptbr/docs/intro>. Acesso em 30 nov. 2025.
- [28] react-scripts <https://www.npmjs.com/package/react-scripts>. Acesso em 30 nov. 2025.
- [29] dotenv <https://www.npmjs.com/package/dotenv>. Acesso em 30 nov. 2025.
- [30] cors <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Guides/CORS>. Acesso em 30 nov. 2025.
- [31] cheerio <https://cheerio.js.org/>. Acesso em 30 nov. 2025.
- [32] puppeteer <https://pptr.dev/>. Acesso em 30 nov. 2025.
- [33] multer <https://www.npmjs.com/package/multer>. Acesso em 30 nov. 2025.
- [34] WIEGERS, Karl; BEATTY, Joy. *Software Requirements*. 3. ed. Redmond: Microsoft Press, 2013. Acesso em 30 nov 2025.
- [35] Google Web Vitals <https://web.dev/articles/vitals>. Acesso em 30 nov. 2025.
- [36] BANKS, Alex; PORCELLO, Eve. *Learning React: Modern Patterns for Developing React Apps*. 2. ed. Sebastopol: O'Reilly Media, 2020. Acesso em 30 nov. 2025.
- [37] FIELDING, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures*. 2000. Acesso em 30 nov. 2025.
- [38] SAURKAR, A. V.; PATHARE, K. G.; GODE, S. A. An overview on web scraping techniques and tools. International Journal on Future Revolution in Computer Science & Communication Engineering, v. 4, n. 4, p. 363-367, 2018. Disponível em: [https://www.researchgate.net/publication/357401723\\_Web\\_Scraping\\_or\\_Web\\_Crawling\\_States\\_of\\_Art\\_Techniques\\_Approaches\\_and\\_Application](https://www.researchgate.net/publication/357401723_Web_Scraping_or_Web_Crawling_States_of_Art_Techniques_Approaches_and_Application). Acesso em 30 nov. 2025.
- [39] Promises no JavaScript  
[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global\\_Objects/Promise](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Promise). Acesso em 30 nov. 2025.