CENTRO UNIVERSITÁRIO DE MACEIÓ

CURSO DE CIÊNCIA DA COMPUTAÇÃO

GABRIEL CARDOSO CARVALHO

JOSÉ MÁRCIO DE ALBUQUERQUE EMÍDIO

LUÍS EDUARDO MARINHO CAVALCANTE LUZ

SISTEMA DE COMPARAÇÃO DE PREÇOS COM AGENTE INTELIGENTE PARA CRIAÇÃO DE CONFIGURAÇÕES DE COMPUTADOR

MACEIÓ - AL

GABRIEL CARDOSO CARVALHO

JOSÉ MÁRCIO DE ALBUQUERQUE EMÍDIO

LUÍS EDUARDO MARINHO CAVALCANTE LUZ

SISTEMA DE COMPARAÇÃO DE PREÇOS COM AGENTE INTELIGENTE PARA CRIAÇÃO DE CONFIGURAÇÕES DE COMPUTADOR

Trabalho de Conclusão de Curso apresentado ao Centro Universitário de Maceió como um dos pré-requisitos para a obtenção de grau de Bacharel em Ciência da Computação.

Orientador: Esp. Victor Brunno Dantas de Souza Rosas

Coorientador: Esp. Ícaro Santos Ferreira

MACEIÓ - AL

GABRIEL CARDOSO CARVALHO

JOSÉ MÁRCIO DE ALBUQUERQUE EMÍDIO

LUÍS EDUARDO MARINHO CAVALCANTE LUZ

SISTEMA DE COMPARAÇÃO DE PREÇOS COM AGENTE INTELIGENTE PARA CRIAÇÃO DE CONFIGURAÇÕES DE COMPUTADOR

Monografia apresentada ao Centro Universitário de Maceió como um dos pré-requisitos para a obtenção de grau de bacharel em Ciência da Computação.

Aprovada em 17/06/2025

Banca Examinadora

Documento assinado digitalmente
gov.br VICTOR BRUNNO DANTAS DE SOUZA ROSAS
Data: 02/07/2025 12:23:58-0300
Verifique em https://validar.iti.gov.br

Esp. Victor Brunno Dantas de Souza Rosas

Centro Universitário de Maceió

Documento assinado digitalmente
gov.br ICARO SANTOS FERREIRA
Data: 03/07/2025 09:59:51-0300
Verifique em https://validar.iti.gov.br

Esp. Ícaro Santos Ferreira

Centro Universitário de Maceió

Documento assinado digitalmente
gov.br MARCOS VINICIUS SILVA BENTO
Data: 15/07/2025 12:10:53-0300
Verifique em https://validar.iti.gov.br

Me. Marcos Vinícius Silva Bento

Centro Universitário de Maceió

Documento assinado digitalmente
gov.br IZAAC DUARTE DE ALENCAR
Data: 07/07/2025 14:20:25-0300
Verifique em https://validar.iti.gov.br

Me. Izaac Duarte De Alencar

Centro Universitário de Maceió

Catalogação na fonte: Biblioteca do Centro Universitário de Maceió, Unima | Afya

C331s

Carvalho, Gabriel Cardoso

Sistema de comparação de preços com agente inteligente para criação de configurações de computador / Gabriel Cardoso Carvalho, José Márcio de Albuquerque Emídio, Luís Eduardo Marinho Cavalcante Luz ; orientação [de] Victor Brunno Dantas de Souza Rosas ; coorientação [de] Ícaro Santos Ferreira. – Maceió, 2025.

60 f. : il.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) -Centro Universitário de Maceió – Unima | Afya, Maceió, 2025.

Inclui Bibliografias: p. 55-57.

1. Comparação de preços. 2. Algoritmo genético. 3. Configuração de computador. I. Emídio, José Márcio de Albuquerque. II. Luz, Luís Eduardo Marinho Cavalcante. III. Rosas, Victor Brunno Dantas de Souza. (orient.). IV. Ferreira, Ícaro Santos. V. Centro Universitário de Maceió. VI. Título.

CDU: 004

Bibliotecária responsável: Adriele da Silva Lima CRB-4/1898

AGRADECIMENTOS

A realização deste Trabalho de Conclusão de Curso representa não apenas o encerramento de uma etapa acadêmica, mas também a concretização de um projeto construído com o apoio e incentivo de muitas pessoas.

Agradecemos primeiramente a Deus, por nos conceder força, saúde e perseverança ao longo dessa jornada.

Aos nossos pais e familiares, pelo amor incondicional, apoio constante e por acreditarem em nós mesmos nos momentos mais difíceis.

Aos colegas e amigos que estiveram ao meu lado, compartilhando experiências, desafios e conquistas durante toda a trajetória universitária.

Aos professores do curso, que, com seu conhecimento e compromisso, contribuíram para minha formação pessoal e profissional.

Por fim, agradecemos a todos que, de alguma forma, fizeram parte deste caminho. A cada um de vocês, o nosso sincero obrigado.

RESUMO

Este trabalho descreve o desenvolvimento de um site comparador de preços de peças de computador com um sistema de recomendação baseado em um algoritmo genético. O objetivo principal é auxiliar consumidores a montar configurações de hardware de acordo com suas necessidades e orçamento, garantindo a compatibilidade entre os componentes. O sistema utiliza tecnologias como React para o front-end, Django para o back-end e PostgreSQL para o banco de dados. Para a demonstração, os preços foram simulados, pois o acesso a APIs de lojas reais não estava disponível. O algoritmo genético, implementado com a biblioteca DEAP, otimiza a seleção de peças considerando desempenho, compatibilidade e custo. O trabalho conclui que o sistema atingiu seu objetivo, demonstrando a viabilidade da integração das tecnologias e do algoritmo genético para a criação de configurações de computador personalizadas e eficientes.

Palavras-chave: Comparação de preços, Algoritmo genético, Configuração de computador

ABSTRACT

This work describes the development of a computer parts price comparison website with a recommendation system based on a genetic algorithm. The main goal is to help consumers build hardware configurations according to their needs and budget, ensuring compatibility between components. The system uses technologies such as React for the front end, Django for the back end, and PostgreSQL for the database. For demonstration purposes, prices were simulated, as access to real store APIs was not available. The genetic algorithm, implemented with the DEAP library, optimizes the selection of parts by considering performance, compatibility, and cost. The study concludes that the system achieved its goal, demonstrating the feasibility of integrating technologies and the genetic algorithm to create customized and efficient computer configurations.

Keywords: Price comparison, Genetic Algorithm, Computer configuration

LISTA DE ABREVIATURAS E SIGLAS

TCC: Trabalho de Conclusão de Curso

IA: Inteligência Artificial

B2B: Business to Business (Negócio para Negócio)

B2C: Business to Consumer (Negócio para Consumidor)

API: Application Programming Interface (Interface de Programação de Aplicações)

JWT: JSON Web Token

DEAP: Distributed Evolutionary Algorithms in Python

CPU: Central Processing Unit (Unidade Central de Processamento)

GPU: Graphics Processing Unit (Unidade de Processamento Gráfico)

iGPU: Integrated Graphics Processing Unit (Unidade de Processamento Gráfico Integrada)

PSU: Power Supply Unit (Unidade de Fonte de Alimentação)

RAM: Random Access Memory (Memória de Acesso Aleatório)

HDD: Hard Disk Drive (Unidade de Disco Rígido)

SSD: Solid State Drive (Unidade de Estado Sólido)

SGBD: Sistemas de Gerenciamento de Bancos de Dados

RNAs: Redes Neurais Artificiais

SUMÁRIO

1 INTRODUÇÃO

Com o recente crescimento do mercado de tecnologia e a popularização de produtos como videogames, a demanda por computadores aumentou significativamente. Entretanto, muitos consumidores acabam adquirindo computadores com configurações de hardware que não condizem com o preço cobrado. Esse problema decorre, em grande parte, da complexidade envolvida em montar um computador equilibrado e proporcional ao custo, agravada pela constante variação de preços das peças e pela dificuldade de avaliar se um componente está atualizado. Segundo Yamakami (2024), em dez anos, a indústria de jogos eletrônicos ficou sete vezes maior no Brasil, o que reforça a crescente importância de soluções tecnológicas que auxiliem o consumidor nesse processo. Nesse contexto, tecnologias como Inteligência Artificial (IA) e comércio eletrônico (e-commerce) tornam-se fundamentais para facilitar a tomada de decisão. Sistemas de comparação de preços aliados a algoritmos de recomendação podem ajudar o consumidor a encontrar configurações de hardware que atendam ao seu orçamento e objetivos, sem exigir um conhecimento técnico aprofundado, além de assegurar a compatibilidade entre os componentes recomendados.

Na literatura, pesquisas como a de Someshkumar Mishra et al. (2021) já exploram sistemas de recomendação voltados à configuração de PCs utilizando técnicas de filtragem colaborativa, evidenciando a viabilidade dessa abordagem para montar builds equilibradas. Os autores demonstram que abordagens envolvendo análise de preferências e similaridade entre usuários podem gerar recomendações relevantes de componentes.

Sistemas de recomendação funcionam como consultores inteligentes, analisando dados como preferências, histórico de compras e padrões de navegação para sugerir produtos ou soluções personalizadas. No contexto da montagem de computadores, eles são capazes de indicar combinações de peças que ofereçam o melhor desempenho dentro de um determinado orçamento, considerando fatores como compatibilidade, desempenho e preço. Ao aplicarem-se a plataformas de montagem de PCs, tais sistemas facilitam a personalização e a decisão de compra mais assertiva e econômica, mesmo para usuários sem conhecimento técnico aprofundado.

No âmbito da otimização de orçamento e desempenho em hardware, técnicas avançadas como redes neurais artificiais e algoritmos genéticos podem ser empregadas com

sucesso para recomendar configurações. A integração de inteligência artificial no processo, com capacidade de adaptar recomendações a partir dos dados de uso e mercado, representa um diferencial inovador deste projeto.

Este trabalho propõe o desenvolvimento de um site comparador de preços de peças de computador, integrado a um sistema de recomendação baseado em algoritmo genético. O objetivo é sugerir configurações ideais conforme o perfil do usuário e o orçamento disponível, utilizando tecnologias modernas de desenvolvimento web, manipulação de dados e IA. O sistema visa simplificar a experiência do consumidor, oferecendo recomendações inteligentes que garantam o melhor custo-benefício e compatibilidade entre os componentes.

O problema de pesquisa abordado refere-se à dificuldade de coletar dados consistentes e completos no mercado de hardware, fragmentado e com informações inconsistentes. Como inovação, propõe-se a utilização de dados sintéticos para modelar o mercado, criando uma base controlada que permita validar o algoritmo genético com segurança e robustez, mesmo na ausência de dados reais abrangentes.

A relevância técnica e científica do projeto reside na adoção de metodologias de geração de dados sintéticos, combinadas a algoritmos de IA e engenharia de software, para criar um sistema robusto e acessível de recomendação de builds, contribuindo para a pesquisa em recomendação e para soluções práticas voltadas ao consumidor comum.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Desenvolver um sistema web para comparação de preços e recomendação automatizada de configurações de hardware para computadores, utilizando um algoritmo genético capaz de otimizar a seleção de componentes com base nas necessidades do usuário e em um orçamento previamente definido.

2.2 OBJETIVOS ESPECÍFICOS

- Projetar e implementar um sistema de recomendação baseado em algoritmo genético, capaz de gerar configurações personalizadas de hardware que

maximizem o desempenho e a compatibilidade entre os componentes, respeitando os limites orçamentários definidos pelo usuário.

- Modelar e estruturar uma base de dados sintética representativa do mercado de hardware, contemplando variáveis como especificações técnicas, faixas de preço e relações de compatibilidade, para permitir o desenvolvimento e validação do algoritmo sem depender de dados reais incompletos ou inconsistentes.
- Avaliar a eficácia do algoritmo genético por meio de métricas de desempenho e custo-benefício, testando sua capacidade de adaptação a diferentes perfis de uso (como jogos, trabalho e edição de vídeo) e cenários orçamentários diversos.

3 ASPECTOS TEÓRICOS

Neste capítulo, serão abordadas as principais características do e-commerce (comércio eletrônico), posteriormente o uso da inteligência artificial, manipulação de dados e elaboração de site.

3.1 E-COMMERCE (COMÉRCIO ELETRÔNICO)

O comércio eletrônico, ou e-commerce, consiste na compra e venda de produtos e serviços por meio da internet. Embora já existissem formas primitivas de transações eletrônicas desde a década de 1960, como o EDI (Electronic Data Interchange), foi somente com a criação da web comercial nos anos 1990 que o e-commerce ganhou tração global (MARQUET, 2024). A partir de 1994, com o uso de criptografia para transações seguras e o surgimento de empresas como Amazon e eBay, o e-commerce entrou em uma fase de crescimento acelerado. Posteriormente, nos anos 2000, a expansão da banda larga, dos dispositivos móveis e dos meios de pagamento digitais consolidou-se como força propulsora do comércio virtual.

Os modelos de negócio predominantes no comércio eletrônico são o B2B (Business-to-Business), o B2C (Business-to-Consumer) e o C2C (Consumer-to-Consumer).

No modelo B2B, empresas vendem produtos ou serviços para outras empresas, com transações geralmente em grandes volumes e ciclos de compra mais prolongados. Já o B2C foca na venda direta ao consumidor final, com decisões de compra mais rápidas, preços fixos e um alto volume de transações individuais. O modelo C2C, por sua vez, envolve consumidores vendendo para outros consumidores, comumente através de plataformas digitais intermediárias. Segundo RESEARCHGATE (2021), esses modelos diferem quanto à natureza da transação, público-alvo, volume de compras e estrutura de relacionamento comercial.

Os comparadores de preços ocupam papel central nesse cenário, especialmente no modelo B2C, ao oferecer ao consumidor ferramentas para comparação rápida de preços e características de produtos entre diferentes fornecedores. Estudos indicam que aproximadamente 80% dos consumidores realizam pesquisas online antes de concluir uma compra, seja em lojas virtuais ou físicas (SPRINGEROPEN, 2025), o que evidencia a importância crescente dessas plataformas na jornada de decisão do consumidor. Comparadores podem ser classificados como genéricos, abrangendo diversos setores do varejo, ou especializados, voltados a nichos específicos, como eletrônicos ou hardware de computador. Seu funcionamento se dá, em geral, por meio de técnicas como web scraping, integração com APIs de parceiros comerciais ou inserção manual de dados.

Contudo, a maioria dos comparadores limita-se à comparação de itens individuais, sem oferecer suporte à montagem de sistemas completos e otimizados, como no caso de computadores pessoais. Neste trabalho, propõe-se justamente uma solução diferenciada, que utiliza inteligência artificial para sugerir configurações completas de hardware, priorizando compatibilidade entre os componentes, desempenho e custo-benefício, a partir do perfil e orçamento do usuário.

3.2 INTELIGÊNCIA ARTIFICIAL (IA)

A inteligência artificial (IA) é o campo da ciência da computação dedicado ao desenvolvimento de sistemas capazes de realizar tarefas que, normalmente, requerem inteligência humana, tais como reconhecimento de voz, aprendizado, raciocínio e tomada de decisão (GOODFELLOW et al., 2016). Entre suas subáreas, destacam-se o aprendizado de máquina (machine learning), visão computacional, processamento de linguagem natural e

sistemas especialistas. O aprendizado de máquina, por exemplo, inclui técnicas que permitem que sistemas aprendam padrões a partir de dados, enquanto sistemas especialistas são programas que simulam o conhecimento humano para tomada de decisões específicas (GOODFELLOW et al., 2016).

Inspirados na teoria da evolução natural proposta por Charles Darwin, os algoritmos genéticos foram introduzidos por John Holland na década de 1970 como uma técnica de otimização baseada em princípios biológicos, como seleção natural, reprodução e mutação (HOLLAND, 1975). Esses algoritmos simulam o processo evolutivo para buscar soluções ótimas em espaços de busca complexos, especialmente quando métodos tradicionais se mostram ineficazes.

Os AG trabalham com uma população de soluções potenciais chamadas indivíduos, representadas por cromossomos que codificam os parâmetros do problema em questão. Cada cromossomo é composto por unidades básicas chamadas genes. O ciclo evolutivo ocorre ao longo de várias gerações, nas quais a qualidade de cada indivíduo é avaliada por uma função de aptidão (fitness function), que mensura o quão adequada é a solução para o problema.

Os principais operadores genéticos utilizados para evoluir a população incluem:

- Seleção: Método pelo qual os indivíduos mais aptos são escolhidos para reprodução. Exemplos incluem a seleção por roleta (probabilidade proporcional à aptidão) e a seleção por torneio (competição entre subconjuntos da população).
- Crossover: Processo que combina partes dos cromossomos de dois indivíduos para gerar descendentes. Pode ser ponto único, múltiplo ou uniforme.
- Mutação: Alteração aleatória de um ou mais genes em um cromossomo, introduzindo diversidade genética e evitando a convergência prematura.

Os algoritmos genéticos apresentam diversas vantagens, como a capacidade de explorar grandes espaços de busca e lidar com problemas não lineares e multimodais. Entretanto, também possuem desvantagens, como o risco de convergência para ótimos locais e uma possível lentidão na convergência dependendo da parametrização e do problema (MITCHELL, 1998).

Devido à sua flexibilidade e robustez, os AG são aplicados em diversos problemas de otimização, incluindo:

- Roteamento: Otimização de caminhos em redes e sistemas logísticos.
- Agendamento: Alocação eficiente de recursos e tarefas.
- Design de engenharia: Projeto automático de componentes e sistemas complexos.
- Otimização de portfólio: Seleção de investimentos que maximizem retorno e minimizem risco.

No contexto deste trabalho, os algoritmos genéticos são utilizados para recomendar configurações otimizadas de hardware de computadores, levando em conta múltiplas restrições e objetivos, como desempenho, custo e compatibilidade (MITCHELL, 1998).

3.3 ELABORAÇÃO DE UM SITE

O desenvolvimento de um website que compara preços de peças de hardware para computadores, integrado a um algoritmo genético capaz de recomendar as melhores configurações conforme o orçamento e perfil do usuário, demanda um processo complexo e multidisciplinar. Esse processo envolve desde o planejamento inicial até a implementação, testes e manutenção contínua da plataforma.

Para garantir uma experiência de usuário satisfatória, o site deve priorizar usabilidade e design responsivo, assegurando que a plataforma funcione adequadamente em diversos dispositivos, como desktops, tablets e smartphones. A interface deve ser intuitiva, facilitando a inserção das preferências do usuário, como finalidade do computador e orçamento disponível.

A atualização dos dados, apesar de realizada inicialmente com dados sintéticos, deve contemplar mecanismos para incorporar informações reais sempre que possível, seja por meio de integrações futuras com APIs ou web scraping, respeitando limitações técnicas e legais.

Além disso, aspectos de segurança são considerados para proteger os dados armazenados, mesmo que sejam predominantemente técnicos, garantindo a integridade da plataforma e a confiança do usuário.

Assim, o projeto oferece uma solução inovadora que facilita a montagem de computadores otimizados e personalizados, mesmo diante dos desafios na coleta de dados do mercado.

4 METODOLOGIA

A metodologia deste trabalho descreve os métodos, técnicas e ferramentas utilizadas para o desenvolvimento do sistema, abordando desde o tipo de pesquisa até os critérios de avaliação de software. O objetivo principal é garantir que todas as etapas deste projeto sejam bem definidas, permitindo que ele seja replicado para fins de estudo e proporcionando uma maior confiabilidade nos resultados obtidos.

4.1 TECNOLOGIAS UTILIZADAS

Para o desenvolvimento deste sistema, foram escolhidas as seguintes tecnologias:

- **Front-end**: A biblioteca React foi escolhida para o desenvolvimento da interface, pois possui diversas soluções da comunidade que facilitam a criação de componentes reutilizáveis. O projeto será iniciado com Vite, que agiliza o desenvolvimento e melhora o desempenho da aplicação.
- **Back-end**: O framework Django será utilizado para a criação da API, que se comunicará com o front-end por meio de chamadas RESTful. Django foi escolhido por ser escrito em Python, o que facilita a implementação do agente inteligente e garante escalabilidade no desenvolvimento da API.
- **Banco de Dados**: O PostgreSQL foi escolhido por ser um banco de dados escalável, confiável e gratuito para uso comercial.
- **Autenticação**: Será implementada utilizando JWT (JSON Web Token) com cookies HTTP-only, garantindo maior segurança na comunicação entre cliente e servidor.
- **Coleta de Preços**: Para fins de demonstração, os preços utilizados no sistema foram simulados por um algoritmo. No entanto, em uma versão futura, espera-se que os preços sejam obtidos por meio de APIs externas fornecidas por lojas parceiras, garantindo dados atualizados e integração com o mercado real.
- **Algoritmo Genético (DEAP):** O sistema utiliza a biblioteca DEAP (Distributed Evolutionary Algorithms in Python) para implementar um algoritmo genético

responsável por gerar combinações otimizadas de componentes de hardware. Através de operadores genéticos como seleção, cruzamento e mutação, o algoritmo busca configurações que atendam aos critérios definidos pelo usuário, como orçamento máximo, compatibilidade entre peças e prioridade de desempenho. O uso do DEAP proporciona flexibilidade e eficiência na modelagem do processo evolutivo, permitindo explorar um grande espaço de soluções possíveis de forma inteligente e automatizada.

4.2 GERAÇÃO DE PEÇAS E PREÇOS

Conforme estabelecido na fundamentação teórica, a opção por dados sintéticos representa uma solução metodológica robusta diante das limitações técnicas e legais do scraping de dados reais. A implementação prática deste sistema exigiu o desenvolvimento de algoritmos capazes de simular as características individuais dos componentes e suas inter-relações, buscando aproximar-se das dinâmicas encontradas no mercado real de hardware.

O sistema desenvolvido baseia-se em uma arquitetura modular que gera componentes seguindo distribuições estatísticas estimadas para o mercado brasileiro. Esta abordagem busca criar um ambiente de simulação que se aproxime das características do mercado real, oferecendo o controle necessário para testes reprodutíveis e validação científica.

4.2.1 Geração de processadores

A geração de processadores constitui o núcleo mais complexo do sistema, considerando que estes componentes determinam grande parte das compatibilidades e limitações de uma configuração. O algoritmo implementado considera as distribuições estimadas entre Intel e AMD, bem como as correlações técnicas entre gerações, sockets e especificações. Para Intel, o sistema simula a evolução das gerações, onde processadores da 8ª e 9ª geração utilizam socket LGA1151, 10ª e 11ª geração empregam LGA1200, e 12ª a 14ª geração adotam LGA1700. Esta correlação temporal-técnica busca garantir que os dados gerados reflitam de forma aproximada a realidade de compatibilidade entre componentes.

As especificações técnicas seguem distribuições modeladas com base nas características técnicas de cada série de processadores. Core i3 é modelado com 4 núcleos nas gerações mais antigas, evoluindo para 4-6 núcleos nas gerações 12+ com a introdução da arquitetura híbrida. Core i5 é simulado progredindo de 6 núcleos para 6-8 núcleos, enquanto Core i7 e i9 incorporam configurações mais complexas com 8-16 núcleos. Para AMD, o sistema busca simular a filosofia de design da empresa, onde Ryzen 3 é modelado com 4-6 núcleos, Ryzen 5 com 6-8 núcleos, Ryzen 7 com 8-12 núcleos, e Ryzen 9 com 12-16 núcleos. A linha Threadripper é tratada separadamente, com 24-64 núcleos e socket sTRX4 específico.

A nomenclatura dos processadores busca seguir as convenções de cada fabricante. Para Intel, o sistema gera nomes como "Intel Core i5-12400F" ou "Intel Core i7-13700K", onde o primeiro dígito após o hífen indica a geração, os três dígitos seguintes representam o modelo dentro da série, e o sufixo indica características especiais (K para desbloqueado, F para sem gráficos integrados, T para baixo consumo). Para AMD, nomes como "AMD Ryzen 5 5600X" ou "AMD Ryzen 7 7700" seguem o padrão observado, onde o primeiro dígito indica a geração, o segundo a série, e os dois últimos o modelo específico.

O sistema de gráficos integrados (iGPU) adiciona uma camada de realismo, considerando que processadores Intel tradicionalmente incluem gráficos integrados (exceto modelos com sufixo F), enquanto AMD tende a limitar esta característica aos modelos com sufixo G.

4.2.2 Placas de vídeo e componentes

Para placas de vídeo, o sistema implementa uma base de dados de chipsets com especificações técnicas aproximadas. Chipsets como RTX 3060, RTX 4090, RX 6800 ou RX 7900 XTX são associados a especificações de TDP, memória VRAM, frequências base e boost estimadas. O algoritmo então simula as variações que podem ser introduzidas por diferentes fabricantes, onde ASUS, MSI, Gigabyte e outros podem adicionar suas próprias implementações de cooling, overclocking de fábrica e designs de PCB. Esta abordagem busca refletir como o mercado real opera, onde um mesmo chip gráfico resulta em diversos produtos diferentes.

A geração de memória RAM considera a transição tecnológica entre DDR4 e DDR5, implementando uma distribuição estimada de 60% DDR4 e 40% DDR5 que busca aproximar-se do estado atual do mercado. As velocidades são geradas dentro de faixas tecnicamente viáveis: DDR4 entre 2133MHz e 3600MHz, DDR5 entre 4800MHz e 7200MHz. O sistema considera também configurações comerciais comuns, como módulos únicos para builds econômicas ou kits de dois módulos para configurações dual-channel otimizadas.

O subsistema de armazenamento simula a diversidade do mercado, gerando HDDs tradicionais para grandes capacidades a baixo custo, SSDs SATA como opção intermediária, e NVMe M.2 para máxima performance. As capacidades seguem progressões comerciais estimadas, onde HDDs escalam de 1TB a 8TB, adequados para armazenamento em massa, enquanto SSDs concentram-se na faixa de 256GB a 2TB, buscando balancear performance e custo.

4.2.3 Sistema de precificação

O sistema de precificação representa talvez o aspecto mais complexo e limitado da simulação, reconhecendo que a formação de preços no mercado real envolve dinâmicas muito mais complexas do que apenas especificações técnicas. Cada componente recebe um preço base calculado através de funções simplificadas que consideram principalmente suas especificações técnicas primárias, embora seja reconhecido que fatores como demanda de mercado, estratégias de marketing, disponibilidade de estoque, ciclos de lançamento de produtos, posicionamento de marca e condições econômicas globais exercem influência significativa na precificação real. Para processadores, a função incorpora número de núcleos, frequências base e turbo, e envelope térmico (TDP), numa tentativa de aproximação que, embora limitada, busca estabelecer alguma correlação entre capacidade computacional e valor de mercado. GPUs são precificadas considerando memória VRAM, frequências de processamento e consumo energético, reconhecendo que esta abordagem representa uma simplificação considerável dos fatores que realmente determinam os preços no mercado real.

A variação de preços entre diferentes lojas é simulada através de múltiplas camadas de aleatoriedade controlada, numa tentativa de aproximar-se das complexas dinâmicas de

precificação do varejo. Cada loja virtual recebe um viés de precificação que pode variar de -15% a +25% em relação ao preço base, numa simplificação que busca simular diferentes estratégias comerciais: lojas com viés negativo representam estabelecimentos focados em competitividade de preços, enquanto viés positivo simula lojas premium ou especializadas. Uma segunda camada de variação temporal (-8% a +12%) tenta aproximar-se das flutuações de mercado, embora seja reconhecido que as variações reais de preço são influenciadas por um conjunto muito mais amplo e complexo de fatores, incluindo condições macroeconômicas, sazonalidade específica de cada produto, estratégias de inventory management, acordos comerciais com fornecedores, e dinâmicas competitivas locais que não podem ser adequadamente capturadas por um modelo simplificado.

O sistema de promoções adiciona uma camada adicional de simulação através da implementação de descontos temporários em 25% dos produtos, com reduções que variam de 8% a 30% e datas de término aleatórias. Esta funcionalidade representa uma aproximação bastante simplificada das campanhas promocionais do varejo online, que na realidade envolvem estratégias muito mais sofisticadas baseadas em análise de comportamento do consumidor, gestão de estoque, calendário promocional coordenado, e objetivos específicos de vendas que não podem ser adequadamente modelados através de distribuições aleatórias.

4.3 MOTIVAÇÃO PARA A ESCOLHA DO ALGORITMO GENÉTICO

A criação automática de configurações de computadores personalizados (builds de PC) representa um problema de otimização combinatória multiobjetivo, no qual se equilibram três critérios principais: orçamento máximo, desempenho dos componentes e finalidade da máquina, como jogos, aplicações em inteligência artificial ou edição de vídeo. A necessidade de compatibilidade entre CPU, GPU, RAM, armazenamento, placa-mãe e fonte, aliada à grande variedade de modelos disponíveis, configura um espaço de busca de altíssima granularidade, inviável para métodos exaustivos ou determinísticos em tempo real (Yang, Xin e Guo, 2012).

Diante disso, opta-se pela utilização de algoritmos genéticos (AGs), pertencentes à classe de metaheurísticas evolucionárias, que se mostram eficazes em problemas com múltiplos objetivos, restrições complexas e variáveis discretas.

Os operadores utilizados no processo evolutivo incluem seleção por torneio, crossover de ponto único e mutação com verificação de viabilidade, que juntos garantem o equilíbrio entre exploração do espaço de busca e convergência para soluções otimizadas. Com isso, o algoritmo é capaz de gerar múltiplas alternativas de builds próximas ao ótimo global, respeitando simultaneamente os objetivos e restrições do problema (Purnaprajna, Reformat e Pedrycz, 2007).

Em comparação com outras abordagens de inteligência artificial, os algoritmos genéticos apresentam vantagens consideráveis. Redes neurais artificiais, por exemplo, são eficazes em tarefas de predição e classificação, mas não são adequadas para otimização combinatória direta, pois exigem bases de dados extensas e não garantem o cumprimento de restrições rígidas como compatibilidade física entre componentes (Baeldung, 2024). Algoritmos baseados em lógica fuzzy permitem a modelagem de preferências subjetivas, como “baixo custo” ou “alto desempenho”, mas, isoladamente, não lidam bem com a explosão combinatória do espaço de soluções. O método proposto por Combs e Andrews (1998) reduz parcialmente esse problema, ao limitar o crescimento exponencial do número de regras, mas ainda requer métodos complementares para uma busca global eficiente.

Sistemas especialistas baseados em regras fixas são frágeis frente à alta dinamicidade do mercado de hardware, além de exigirem atualizações manuais constantes. Por fim, heurísticas tradicionais como Hill Climbing ou Simulated Annealing podem encontrar soluções locais aceitáveis, mas não oferecem a mesma robustez dos AGs quanto à diversidade de soluções e à capacidade de escapar de mínimos locais (Konak, Coit e Smith, 2006).

Assim, os algoritmos genéticos destacam-se como a abordagem mais adequada para a seleção automática de builds de PC, uma vez que permitem a otimização simultânea de múltiplos critérios, respeitam restrições técnicas e financeiras, adaptam-se dinamicamente a variações nos modelos disponíveis e proporcionam ao usuário final diferentes opções viáveis e personalizadas de montagem.

4.4 APLICAÇÃO DO ALGORITMO GENÉTICO PARA OTIMIZAÇÃO DE MONTAGEM DE COMPUTADORES

Neste trabalho foi utilizado um algoritmo genético para realizar a otimização da montagem de computadores pessoais (PCs), considerando múltiplos critérios como

desempenho, compatibilidade entre componentes e restrições de orçamento. A implementação foi realizada com a biblioteca **DEAP (Distributed Evolutionary Algorithms in Python)**, amplamente utilizada em pesquisa e aplicações práticas envolvendo computação evolutiva.

4.4.1 Representação de Indivíduos

Cada indivíduo na população representa uma configuração de computador, composta por uma lista ordenada com os seguintes componentes:

- CPU
- GPU (ou iGPU vinculada à CPU, quando aplicável)
- Fonte de alimentação (PSU)
- Placa-mãe (Motherboard)
- Memória RAM
- Armazenamento (HDD/SSD)

Utilizou-se a classe creator.Individual da biblioteca DEAP para modelar os indivíduos. Cada componente é representado por uma instância dos modelos correspondentes no banco de dados. A avaliação da aptidão (fitness) considera fatores como compatibilidade entre peças, desempenho geral e restrição orçamentária.

4.4.2 Avaliação de Aptidão (Fitness)

A função de fitness é o componente central do algoritmo genético, responsável por orientar a evolução da população a cada geração. A função avalia as configurações com base em três critérios principais:

4.4.2.1 Verificação de compatibilidade

Antes de qualquer cálculo de desempenho, são aplicadas penalidades severas a indivíduos com configurações inviáveis, incluindo:

- CPU incompatível com o soquete da placa-mãe;
- iGPU que não corresponde à CPU que a contém;

- RAM com tipo incompatível com a placa-mãe;
- Fonte de alimentação com potência insuficiente para suportar o consumo estimado.

4.4.2.2 Cálculo de desempenho

O desempenho de cada componente é normalizado com base nos valores máximos disponíveis no banco de dados.

Para o processador (CPU), o desempenho é calculado pela média entre as frequências base e turbo, multiplicada pelo número de núcleos, e normalizada pelo valor máximo:

$$\mathbf{CPU_{norm}} = \frac{\frac{\mathrm{FREQ_{base}}+\mathrm{FREQ_{turbo}}}{\mathbf{2}} \cdot \mathrm{nucleos}}{\mathrm{MAXCPU}}$$

Para a GPU, o desempenho bruto é baseado na média das frequências multiplicada por uma função exponencial da quantidade de memória de vídeo e ajustada por um multiplicador baseado na arquitetura do chipset:

$$\mathbf{GPU_{raw}} = \left(\frac{\mathrm{FREQ_{base}} + \mathrm{FREQ_{turbo}}}{\mathbf{2}}\right) \cdot \mathrm{Memoria}^{\mathbf{0.7}} \cdot \mathrm{Multiplicador_{chipset}}$$

E o valor é normalizado:

$$\mathbf{GPU_{norm}} = \frac{\mathbf{GPU_{raw}}}{\mathbf{MAXGPU}}$$

Para a memória RAM, o desempenho é obtido com base na capacidade total e na velocidade da memória, e normalizado:

$$\mathbf{RAM_{norm}} = \frac{\mathrm{Capacidade\ total} \cdot \mathrm{Velocidade}}{\mathbf{MAXRAM}}$$

A fórmula de desempenho pondera cada componente com um peso configurável pelo usuário:

$$\mathbf{Desempenho\ Total} = \omega_{\mathbf{cpu}} \cdot \mathbf{CPU_{norm}} + \omega_{\mathbf{gpu}} \cdot \mathbf{GPU_{norm}} + \omega_{\mathbf{ram}} \cdot \mathbf{RAM_{norm}}$$

4.4.2.3 Penalização por orçamento

Caso o preço total da configuração ultrapasse o orçamento definido pelo usuário, aplica-se uma penalização proporcional ao excesso. A penalização é calculada de forma não linear, com crescimento mais suave:

$$\mathbf{Penalizacao} = \mathbf{1000} \cdot \left(\frac{\text{Preco Total} - \text{Orcamento}}{\text{Orcamento}} \right)^{\mathbf{1.5}}$$

A função de fitness final é definida como:

$$\mathbf{Fitness} = (\mathbf{Desempenho\ Total} \cdot \mathbf{1000}) - \mathbf{Penalizacao}$$

Essa estrutura garante que indivíduos com bom desempenho e dentro do orçamento tenham uma aptidão elevada, enquanto configurações inviáveis ou muito caras sejam penalizadas severamente.

4.4.3 Operadores Genéticos

Os operadores genéticos definidos foram:

- Seleção por torneio: seleciona os melhores indivíduos com base na aptidão.
- Cruzamento (crossover): realizado em ponto único, trocando parte da lista de componentes entre dois indivíduos.
- Mutação: substitui aleatoriamente um componente por outro compatível. A lógica da mutação leva em conta a viabilidade da nova configuração, como compatibilidade de socket, tipo de RAM e potência da fonte.

4.4.4 Inicialização e Execução

A população inicial é composta por builds geradas aleatoriamente com base em regras de compatibilidade. Durante o processo evolutivo, são utilizados os seguintes parâmetros:

- Gerações: 50
- População: 100
- Taxa de cruzamento: 80%
- Taxa de mutação: 20%

Em cada geração:

1. Indivíduos são avaliados e classificados por aptidão;

2. Realiza-se a seleção por torneio;
3. Cruzamento e mutação são aplicados;
4. A nova geração é formada a partir dos melhores resultados.

Ao final do processo, o algoritmo retorna a melhor configuração encontrada.

O resultado do algoritmo é convertido em um dicionário contendo os dados de cada componente, preço individual, preço total da configuração e valor da aptidão final. Esse formato facilita a integração com APIs ou interfaces web, como o front-end do sistema.

Exemplo de saída do algoritmo (A saída completa está disponível no Apêndice A):

Apêndice A - Saída do algoritmo


```
{
    "cpu": {
        "name": "AMD science",
        "image": "https://dummyimage.com/514x450",
        "brand": "AMD",
        "igpu": "02a5e859-ae16-4205-81c5-ff72d6d0f85c",
        "socket": "AM4",
        "tdp": 42,
        "cores": 16,
        "speed": 2298,
        "turbo": 4624,
        "price": {
            "store": "218539ba-207f-4919-bacb-2fe39dd6276a",
            "url_product":
"https://jones-howell.com/produtos/amd-science?ref=5388",
            "sale": false,
            "price": 4694.31,
            "old_price": null,
            "sale_percent": null,
            "sale_end": null,
            "content_type": 10,
            "object_id":
"057ef6e6-ba66-4338-a1f0-13a49d7d4e1b",
            "uid": "1381d49a-f962-431d-8415-4d104dcf1d26"
        }
    }
...
```


Fonte: Elaborado pelos autores (2025)

4.5 CASOS DE USO

O sistema proposto possui dois tipos de usuários: o usuário não logado e o usuário logado, sendo que cada um possui diferentes permissões de acesso às funcionalidades do sistema. A Figura 3 apresenta o diagrama de casos de uso, que representa graficamente essas interações.

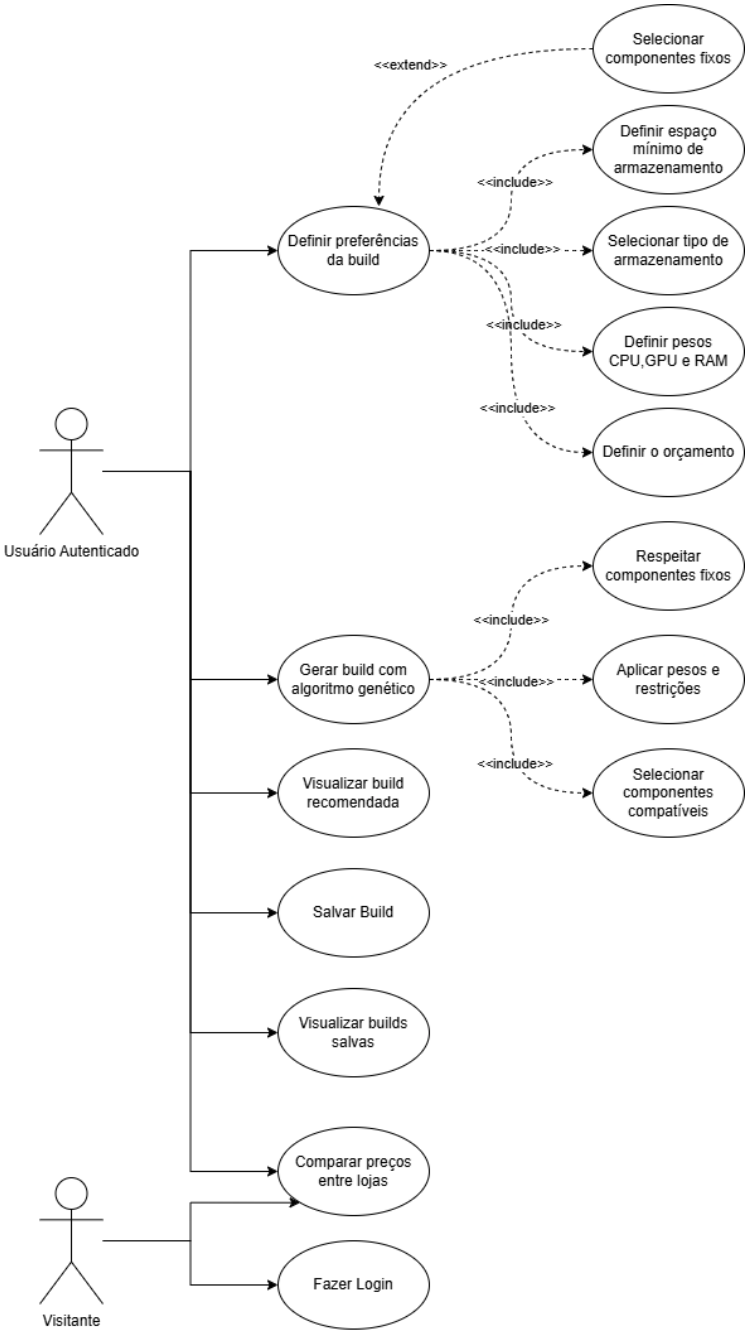
O usuário não logado possui acesso restrito, podendo apenas comparar preços entre lojas e realizar login ou cadastro. Essa limitação tem como objetivo oferecer uma prévia das funcionalidades do sistema, ao mesmo tempo em que incentiva o usuário a realizar autenticação para obter acesso completo.

Após o login, o usuário logado pode acessar todas as funcionalidades do sistema. Dentre elas, destaca-se a possibilidade de definir preferências da build, onde o usuário informa requisitos como orçamento disponível, tipo e espaço de armazenamento, pesos atribuídos à CPU, GPU e memória RAM, além de componentes que deseja manter fixos. Com base nessas preferências, o sistema aplica um algoritmo genético para gerar uma build recomendada, respeitando as restrições definidas.

O usuário também pode visualizar a build gerada, salvá-la para uso posterior e consultar builds salvas anteriormente. A funcionalidade de comparação de preços entre lojas permite ao usuário analisar os valores dos componentes selecionados em diferentes fornecedores, contribuindo para uma escolha mais econômica.

Esse modelo de interação visa proporcionar uma experiência personalizada, eficiente e acessível, integrando técnicas de inteligência artificial para auxiliar na montagem de computadores conforme as necessidades específicas de cada usuário.

Figura 1 - Diagrama de casos de uso



Fonte: Elaborado pelos autores (2025)

4.6 REQUISITOS

4.6.1 Requisitos Funcionais

4.6.1.1 Entrada de Dados

- RF01 – O sistema deve permitir que o usuário defina um orçamento máximo.
- RF02 – O sistema deve utilizar um banco de dados com peças de hardware (CPU, GPU, RAM, SSD, PSU, Placa-Mãe).
- RF03 – O sistema deve aplicar restrições de compatibilidade entre componentes.

4.6.1.2 Algoritmo Genético

- RF04 – O sistema deve inicializar a população com builds aleatórias.
- RF05 – O sistema deve avaliar a função de fitness com base em desempenho e custo.
- RF06 – O sistema deve realizar a seleção dos indivíduos mais aptos, utilizando o método de torneio ou roleta.
- RF07 – O sistema deve aplicar crossover para gerar novas configurações.
- RF08 – O sistema deve aplicar mutação nas builds para introduzir diversidade.
- RF09 – O sistema deve aplicar elitismo para preservar as melhores soluções entre gerações.

4.6.1.3 Funcionalidades Adicionais

- RF10 – O sistema deve controlar a compatibilidade entre componentes, como o socket do processador e da placa-mãe.
- RF11 – O sistema deve gerenciar o consumo de energia, garantindo que a fonte (PSU) seja compatível com GPU e CPU.
- RF12 – O sistema deve armazenar o histórico de builds para aprendizado contínuo e sugestões futuras.

4.6.2 Requisitos Não Funcionais

4.6.2.1 Desempenho

- RNF01 – O sistema deve ser otimizado para executar em tempo razoável, mesmo com grandes volumes de dados.
- RNF02 – O sistema deve ser escalável para diferentes perfis de usuários (como usuários casuais, gamers e profissionais).

4.6.2.2 Confiabilidade

- RNF03 – O sistema deve verificar a compatibilidade entre todos os componentes antes de sugerir uma build.
- RNF04 – O sistema deve garantir que o orçamento definido pelo usuário não seja excedido.

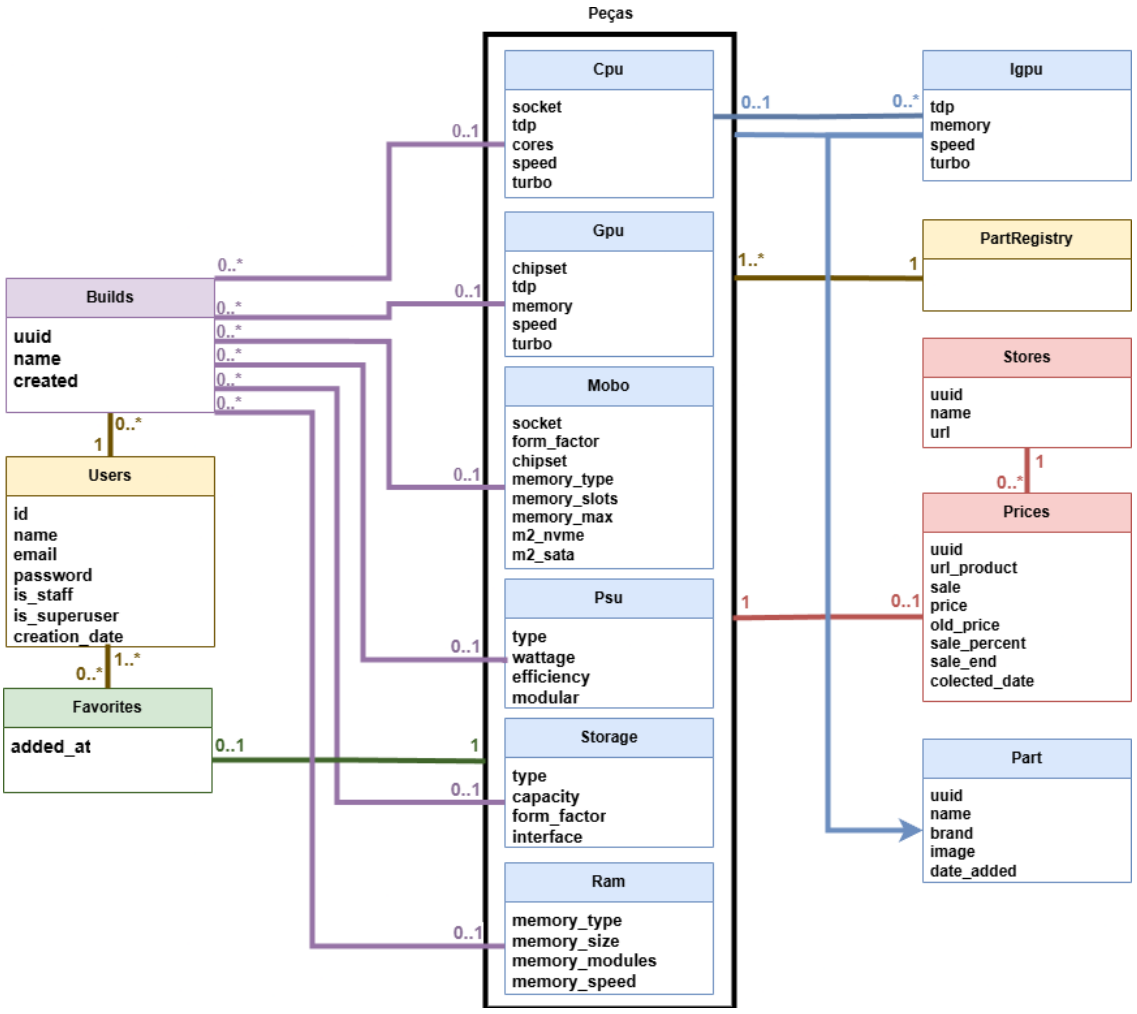
4.6.2.3 Usabilidade

- RNF05 – O sistema deve oferecer uma interface simples e intuitiva para que o usuário defina suas necessidades e orçamento.
- RNF06 – O sistema deve gerar um relatório detalhado da configuração final sugerida, contendo justificativas das escolhas realizadas.

4.7 DIAGRAMA DE CLASSES

O diagrama de classes da Figura 4 descreve as entidades e relacionamentos do sistema. A classe Users possui uma associação com Builds, que por sua vez é composta por várias Peças (componentes). Entre as peças, a Cpu pode ter uma relação opcional com uma iGpu (placa integrada). As Peças também se conectam a Prices e Stores para dados comerciais, e, por fim, a classe Favorites estabelece um relacionamento opcional entre um usuário e uma peça de seu interesse.

Figura 2 - Diagrama de Classes



Fonte: Elaborado pelos autores (2025)

5 RESULTADOS E DISCUSSÕES

5.1 DESENVOLVIMENTO DO SISTEMA

O site foi implementado implementando as tecnologias definidas na metodologia: React no front-end, Django no back-end e PostgreSQL para armazenar os dados. A comunicação entre cliente e servidor se deu por meio de uma api RESTful, com autenticação segura via JWT armazenado em cookies HTTP-only.

O algoritmo genético foi integrado à aplicação, possibilitando a geração de configurações de hardware personalizadas com base no orçamento e nas preferências indicadas pelo usuário. Critérios como desempenho, compatibilidade entre componentes e custo-benefício foram utilizados como parâmetros de avaliação das soluções geradas.

5.2 COLETA DE DADOS E ATUALIZAÇÃO DE PREÇOS

Durante os testes do sistema, a coleta de dados foi realizada por meio de um algoritmo que gerou produtos e preços simulados, com base em faixas e categorias compatíveis com o mercado real. Essa estratégia foi adotada devido à indisponibilidade de acesso a APIs públicas de lojas reais, que normalmente exigem parcerias comerciais, e também para evitar práticas como web scraping, que podem violar os termos de uso de muitos sites.

Apesar de os dados serem fictícios, eles foram estruturados de forma a representar cenários realistas de mercado, o que permitiu avaliar de maneira eficaz as funcionalidades do sistema. Entre essas funcionalidades, destaca-se o monitoramento de variações de preço, que permite ao usuário acompanhar reduções nos valores de produtos previamente selecionados.

A utilização de dados simulados demonstrou ser suficiente para validar os mecanismos de atualização de preços, geração de notificações e sugestões personalizadas, comprovando que o sistema é funcional e aplicável, caso venha a ser integrado futuramente com dados reais.

5.3 ANÁLISE DA EVOLUÇÃO DAS MÉTRICAS DE FITNESS

Tabela 1 - Estatísticas das gerações com pesos para IA

(continua)

Gerações	Indivíduos Avaliados	Tempo	Média	Desvio Padrão	Fitness Mínimo	Fitness Máximo
0	150	0	-494.469	395.305	-1000	46.9124
1	120	0.153795	-214.719	264.81	-1000	46.9124
2	131	0.108524	-96.371	226.646	-1000	46.9124
3	131	0.105044	-30.3707	186.329	-1000	64.2808
4	119	0.0875289	-51.6255	268.222	-1000	64.2808
5	113	0.112082	-62.3035	292.49	-1000	72.2479

Tabela 1 - Estatísticas das gerações com pesos para IA

(continua)

Gerações	Indivíduos Avaliados	Tempo	Média	Desvio Padrão	Fitness Mínimo	Fitness Máximo
6	134	0.143113	-17.2837	222.727	-1000	72.2479
7	126	0.0792856	-23.6153	261.32	-1000	72.2479
8	125	0.0932271	-11.9408	250.284	-1000	72.2479
9	133	0.130512	-1.10598	238.299	-1000	107.122
10	122	0.095048	-5.33951	254.401	-1000	89.9151
11	129	0.160043	36.6067	159.201	-1000	94.7628
12	132	0.121711	0.111246	243.786	-1000	94.7628
13	121	0.0925236	41.5829	154.093	-1000	94.7628
14	125	0.0963471	50.6711	175.4	-1000	94.7628
15	126	0.0860257	45.4883	198.257	-1000	94.7628
16	130	0.0988584	19.4265	259.398	-1000	101.312
17	135	0.0967667	22.6256	259.974	-1000	101.312
18	124	0.108057	32.1235	245.677	-1000	102.034
19	134	0.111622	10.3271	284.958	-1000	109.339
20	129	0.0610173	45.6162	231.811	-1000	112.818
21	129	0.0935454	73.4191	155.032	-1000	112.818
22	137	0.100515	45.0841	232.156	-1000	112.818
23	115	0.0785263	64.5439	179.056	-1000	112.818
24	120	0.0855298	19.5353	287.441	-1000	113.162
25	138	0.0755343	57.6191	217.111	-1000	113.162
26	120	0.0719988	46.661	249.539	-1000	113.162
27	116	0.143141	37.3204	263.606	-1000	113.162
28	121	0.0776458	47.947	249.881	-1000	140.927
29	124	0.0770168	70.5442	200.016	-1000	140.927
30	125	0.0665603	56.0773	234.863	-1000	140.927
31	123	0.107432	72.0815	200.848	-1000	140.927
32	115	0.0955527	70.2767	220.007	-1000	140.927
33	139	0.0918372	85.6656	203.445	-1000	149.062
34	127	0.0961127	102.842	183.739	-1000	149.062
35	127	0.0956962	80.952	240.418	-1000	149.062
36	129	0.0805175	94.2348	205.403	-1000	149.062

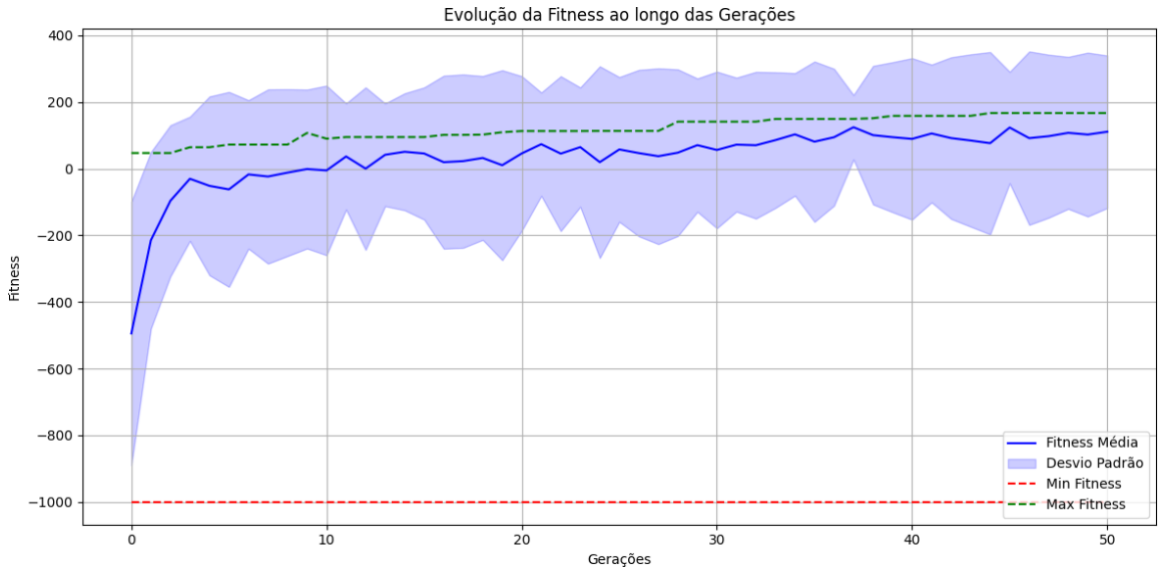
Tabela 1 - Estatísticas das gerações com pesos para IA

(conclusão)

Gerações	Indivíduos Avaliados	Tempo	Média	Desvio Padrão	Fitness Mínimo	Fitness Máximo
37	112	0.0975146	124.244	96.7835	-1000	149.206
38	128	0.0840979	100.524	207.688	-1000	151.214
39	128	0.0715399	94.4002	224.94	-1000	158.334
40	128	0.0655429	89.2504	242.128	-1000	158.334
41	125	0.0846233	105.498	206.569	-1000	158.334
42	136	0.0731461	91.6688	242.389	-1000	158.334
43	134	0.0770326	84.4463	258.411	-1000	158.334
44	126	0.12037	76.4148	273.347	-1000	166.847
45	125	0.0705512	123.315	167.251	-1000	166.847
46	130	0.0505507	91.513	260.069	-1000	166.847
47	118	0.0799983	97.387	244.445	-1000	166.847
48	129	0.0965576	107.406	227.88	-1000	166.847
49	122	0.0881047	102.286	245.707	-1000	166.847
50	138	0.079668	110.701	228.614	-1000	166.847

Fonte: Elaborado pelos autores (2025)

Gráfico 1 - Evolução do fitness com peso para IA



Fonte: Elaborada pelos autores (2025)

O Gráfico 1 apresenta a evolução da métrica de fitness ao longo de 50 gerações, com o objetivo de encontrar a melhor configuração de hardware possível, respeitando o orçamento do usuário e os critérios de compatibilidade entre componentes. O algoritmo genético utilizado foi desenvolvido com operadores personalizados de cruzamento e mutação que garantem, sempre que possível, a viabilidade das soluções geradas.

A execução foi realizada com os seguintes parâmetros:

- Orçamento máximo: R$ 3.000,00
- Tipo de armazenamento: SSD
- Capacidade mínima de armazenamento: 512 GB
- Pesos de desempenho para perfil de IA: CPU (30%), GPU (60%) e RAM (10%)

A função de avaliação (fitness) penaliza duramente indivíduos inviáveis (com valor fixo de -1000), enquanto recompensa configurações válidas com base em um escore ponderado de desempenho, considerando os pesos definidos para o perfil de uso em inteligência artificial. Penalizações adicionais são aplicadas proporcionalmente caso o orçamento seja ultrapassado, permitindo certa flexibilidade para soluções próximas ao limite de custo.

A aptidão máxima da população evoluiu de 46,91 na geração inicial até 166,84 na geração 49, demonstrando um progresso contínuo na identificação de builds mais eficientes. Esse crescimento indica que o algoritmo foi capaz de combinar peças compatíveis e de alto desempenho de maneira progressivamente mais eficaz ao longo do tempo.

Já a aptidão média, que iniciou em -494,46, apresentou melhora significativa nas primeiras gerações. Por volta da geração 10, observa-se uma inflexão, com a média oscilando entre valores positivos e negativos, refletindo o equilíbrio entre soluções válidas e indivíduos penalizados. A partir da geração 20, a média estabiliza entre 50 e 110 pontos, revelando uma população com desempenho crescente, embora ainda afetada por variações internas.

A presença recorrente de indivíduos com fitness igual a -1000 indica que, apesar das verificações de viabilidade embutidas nos operadores genéticos, ainda ocorrem combinações de componentes incompatíveis ou configurações que ultrapassam o limite de potência da fonte. Essa ocorrência é esperada, dado que tanto a mutação quanto o cruzamento possuem um componente estocástico. No entanto, a função de fitness suaviza penalizações por

orçamento excedido, permitindo que soluções promissoras, embora mais caras, continuem participando da população.

O desvio padrão da fitness se manteve elevado durante toda a execução, com valores variando entre 150 e 270, e picos superiores a 300. Essa dispersão reflete a coexistência de soluções excelentes e configurações penalizadas severamente. A diversidade é fomentada pelos operadores personalizados: a mutação respeita restrições técnicas como tipo de memória e compatibilidade de soquete, enquanto o cruzamento realiza trocas entre componentes apenas quando são viáveis.

Embora essa diversidade seja positiva nas fases iniciais, por promover uma ampla exploração do espaço de busca, sua persistência nas gerações finais indica falta de convergência total e sugere a presença de múltiplos ótimos locais.

A execução demonstrou que os operadores de crossover consciente (com verificação de compatibilidade entre CPU, GPU, RAM, armazenamento e placa-mãe) e mutação dirigida por restrições são eficazes para manter uma taxa razoável de indivíduos válidos. A presença constante de indivíduos inviáveis, no entanto, aponta para possíveis melhorias no algoritmo, como:

- Aplicação de técnicas de reparo após mutações destrutivas;
- Estratégias de substituição elitista para evitar perda de boas soluções;
- Redução progressiva da taxa de mutação conforme as gerações avançam.

Apesar disso, o aumento progressivo da média e do valor máximo de fitness ao longo das gerações comprova que o algoritmo consegue evoluir soluções cada vez mais adequadas, mesmo em um espaço de busca restrito e com múltiplas condicionantes técnicas.

5.3.1 Comparação com pesos balanceados

Tabela 2 - Estatísticas das gerações com pesos balanceados

(continua)

Gerações	Indivíduos Avaliados	Tempo	Média	Desvio Padrão	Fitness Mínimo	Fitness Máximo
0	150	0	-498.676	376.789	-1000	57.2707
1	128	0.111614	-257.999	282.038	-1000	57.2707
2	134	0.10005	-134.654	269.995	-1000	57.2707

Tabela 2 - Estatísticas das gerações com pesos balanceados

(continua)

Gerações	Indivíduos Avaliados	Tempo	Média	Desvio Padrão	Fitness Mínimo	Fitness Máximo
3	126	0.0851629	-47.6672	233.19	-1000	57.2707
4	114	0.0955138	-9.87828	220.665	-1000	57.2707
5	119	0.0895531	-31.5278	285.773	-1000	57.2707
6	126	0.0950778	-84.8715	358.984	-1000	57.2707
7	134	0.183112	-7.24111	250.88	-1000	57.2725
8	125	0.0770125	-35.8472	297.092	-1000	57.2725
9	127	0.0985348	-35.7658	297.12	-1000	57.2727
10	131	0.0985579	-21.5628	275.393	-1000	57.2727
11	125	0.0595202	-43.2086	307.148	-1000	57.2727
12	117	0.0840685	-21.4638	275.354	-1000	57.2727
13	117	0.0941703	-27.748	286.747	-1000	57.2727
14	116	0.0720208	-0.253676	237.465	-1000	57.2727
15	116	0.0925536	-0.775411	237.422	-1000	57.2727
16	129	0.0835176	0.0766581	237.44	-1000	57.2732
17	123	0.117524	-85.9356	358.693	-1000	57.2732
18	126	0.0855596	13.8904	207.041	-1000	57.2732
19	129	0.0795319	-71.2268	343.082	-1000	57.2732
20	133	0.0925303	-7.90423	250.794	-1000	57.2732
21	122	0.0755377	-35.6997	297.169	-1000	57.2732
22	125	0.160346	-29.5262	286.393	-1000	57.2732
23	124	0.0760267	-35.4401	297.19	-1000	57.2732
24	125	0.0700758	-42.1515	307.386	-1000	57.2732
25	127	0.0775373	-27.8868	286.696	-1000	57.2732
26	133	0.078007	-57.4625	325.843	-1000	57.2732
27	118	0.0665641	-28.6747	286.559	-1000	57.2732
28	115	0.0825403	-56.9141	325.964	-1000	57.2732
29	126	0.0735373	-36.4898	297.018	-1000	57.2732
30	130	0.0890326	-77.9194	351.264	-1000	57.2732
31	121	0.145091	-49.8015	316.818	-1000	57.2732
32	126	0.0705266	-29.159	286.442	-1000	57.2732
33	126	0.0609903	-49.9626	316.812	-1000	57.2732

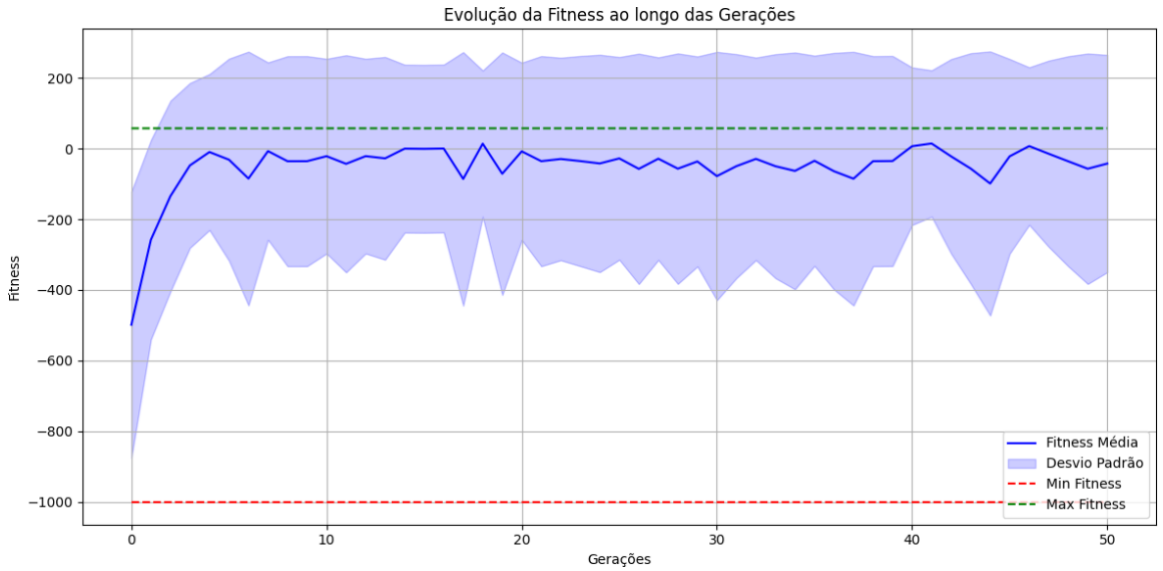
Tabela 2 - Estatísticas das gerações com pesos balanceados

(conclusão)

Gerações	Indivíduos Avaliados	Tempo	Média	Desvio Padrão	Fitness Mínimo	Fitness Máximo
34	128	0.0725598	-63.2168	334.945	-1000	57.2732
35	128	0.0680819	-34.5838	297.395	-1000	57.2732
36	127	0.127729	-64.185	334.689	-1000	57.2732
37	127	0.0802634	-85.2973	358.893	-1000	57.2732
38	120	0.0935318	-35.7431	297.146	-1000	57.2732
39	130	0.0740063	-35.2843	297.239	-1000	57.2732
40	116	0.113719	6.5946	222.858	-1000	57.2732
41	124	0.103508	14.3468	207.106	-1000	57.2732
42	117	0.101844	-22.1476	275.239	-1000	57.2732
43	127	0.0759871	-56.4724	326.107	-1000	57.2732
44	125	0.0645247	-98.9241	373.665	-1000	57.2732
45	131	0.0636632	-21.993	275.307	-1000	57.2732
46	127	0.147453	6.97857	222.876	-1000	57.2732
47	119	0.0554981	-14.748	263.446	-1000	57.2732
48	111	0.0930486	-36.0338	297.077	-1000	57.2732
49	113	0.0709896	-57.1129	325.952	-1000	57.2732
50	114	0.0725727	-42.4851	307.29	-1000	57.2732

Fonte: Elaborado pelos autores (2025)

Gráfico 2 - Evolução do fitness com peso equilibrado



Fonte: Elaborado pelos autores (2025)

Para efeito de comparação, foi realizada uma nova execução do algoritmo genético com os mesmos parâmetros de orçamento, tipo e capacidade mínima de armazenamento, porém com pesos de desempenho balanceados entre CPU e GPU(CPU (45%), GPU (45%) e RAM (10%)), resultados mostrados acima pela tabela 2 e o gráfico 2.

Esta variação busca avaliar como o algoritmo se comporta quando os pesos de desempenho deixam de favorecer fortemente a GPU (como no perfil de IA anterior) e passam a distribuir importância de forma mais uniforme entre o processador e a placa de vídeo.

Nesta execução, o valor máximo de fitness obtido foi 57,27, alcançado pela primeira vez já na geração inicial e mantido sem alterações até a geração final. Isso indica que uma solução com desempenho razoável foi encontrada precocemente, mas o algoritmo não conseguiu superá-la nas gerações seguintes. Diferente do observado na análise anterior, onde houve uma clara evolução do valor máximo (de 46,91 para 166,84), aqui não há evidência de progresso após o início da execução.

A fitness média, que começou em -498,67, oscilou entre valores negativos e levemente positivos ao longo das gerações. Em momentos pontuais, como nas gerações 16, 40, 41 e 46, a média ultrapassou zero, atingindo máximas próximas de 14 pontos. No entanto, esses picos foram esporádicos e não sustentaram uma tendência de melhora contínua, como ocorreu no perfil voltado para IA.

O desvio padrão, constantemente elevado (variando entre 220 e 375), reforça a presença de grande variabilidade na população, com indivíduos de excelente desempenho coexistindo com soluções inviáveis ou penalizadas.

Ao comparar os resultados com a execução anterior, nota-se que o perfil com ênfase em GPU (60%) favoreceu a identificação de configurações de alto desempenho em menos tempo, com crescimento constante do fitness máximo e médio. Já o perfil balanceado, embora mais neutro em relação aos componentes, parece ter reduzido o direcionamento evolutivo, dificultando a convergência para soluções superiores.

Isso pode ser explicado pela própria natureza do espaço de busca: o peso mais alto para GPU, anteriormente, orientava o algoritmo a priorizar placas de vídeo mais potentes, o que aumentava a seletividade e gerava pressão evolutiva mais intensa. No cenário atual, o equilíbrio entre CPU e GPU diminui o impacto individual de cada componente, ampliando o conjunto de possíveis soluções "razoáveis", mas diluindo os critérios de seleção.

Além disso, a persistência do valor máximo inicial durante toda a execução sugere que o algoritmo pode ter caído em um ótimo local precoce, e que os operadores de crossover e mutação, embora conscientes de compatibilidade, não foram suficientes para escapar dessa estagnação.

O experimento com pesos balanceados revela um aspecto importante da formulação do problema: os pesos atribuídos aos componentes de desempenho influenciam diretamente a capacidade do algoritmo genético em evoluir soluções superiores. No caso em questão, a uniformidade dos pesos reduziu a pressão seletiva sobre componentes críticos, o que comprometeu a eficiência da evolução ao longo das gerações.

Esse resultado reforça a importância de se ajustar os pesos de acordo com o perfil de uso desejado, e indica que o algoritmo, embora robusto, responde melhor a objetivos mais definidos (como desempenho em IA), onde pode explorar com mais foco os pontos fortes dos componentes.

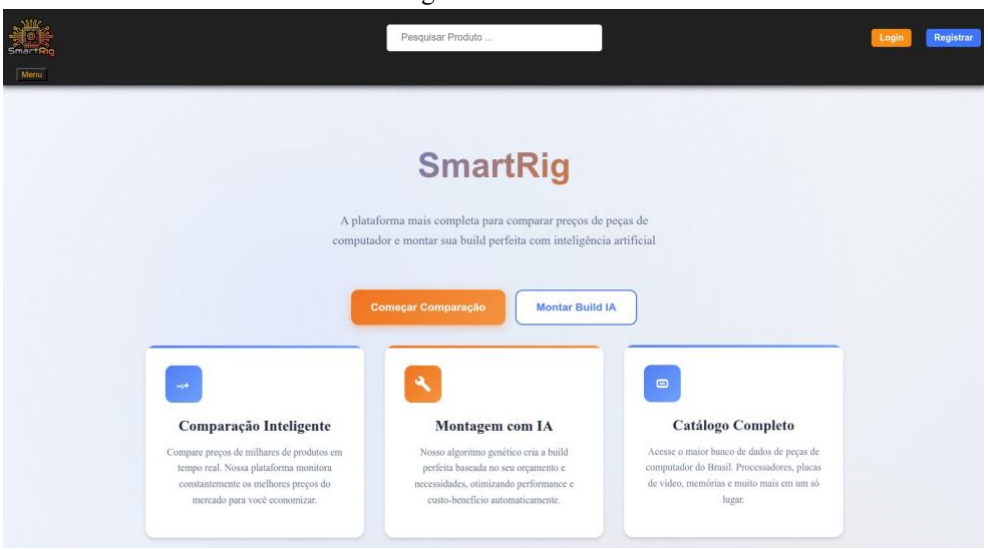
5.4 DESENVOLVIMENTO DO SITE

Para o desenvolvimento deste sistema, foi utilizada a biblioteca React, baseada em JavaScript, que permite a criação de interfaces de usuário interativas e facilita a incorporação

de HTML no código JavaScript. O React será responsável pela construção do visual (front-end) do site, bem como pela responsividade da aplicação. Toda a lógica de funcionamento será gerenciada por meio de uma API RESTful, implementada com o Django, framework em Python voltado para o desenvolvimento web e a criação de APIs robustas e escaláveis. O banco de dados PostgreSQL será utilizado para armazenar tanto os dados dos usuários quanto às informações provenientes das APIs externas que serão utilizadas pelo algoritmo genético.

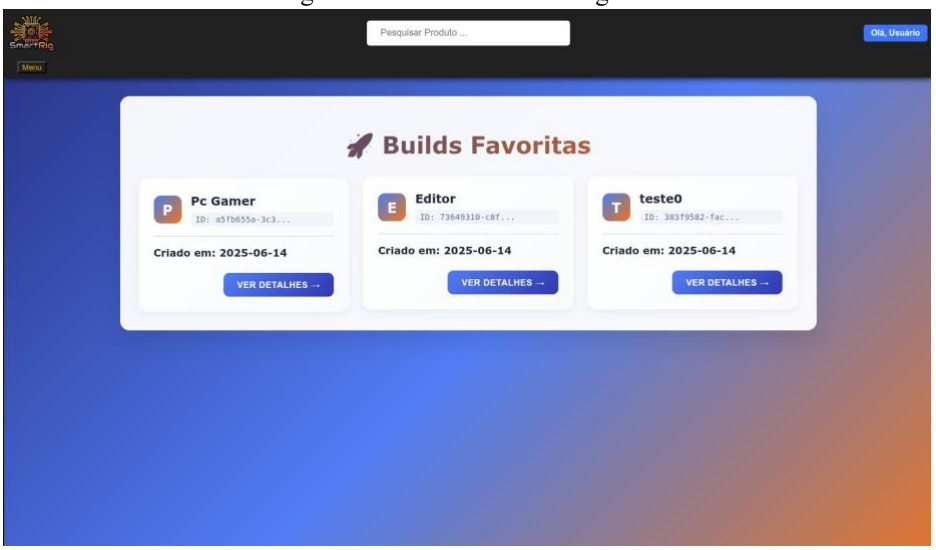
Para o desenvolvimento da interface deste sistema, foi utilizado o Figma, uma ferramenta de design colaborativo que permite a criação de protótipos interativos e interfaces de usuário com alta precisão visual. O Figma foi essencial na etapa de concepção e validação do layout do front-end, possibilitando a prototipagem rápida e a colaboração entre designers e desenvolvedores em tempo real. Através dele, foi possível definir a estrutura visual da aplicação, incluindo a responsividade para diferentes dispositivos, antes mesmo da implementação com a biblioteca React. Dessa forma, o Figma serviu como base para garantir uma experiência do usuário consistente e bem planejada. Abaixo seguem imagens das interfaces finalizadas:

Figura 3 - Home



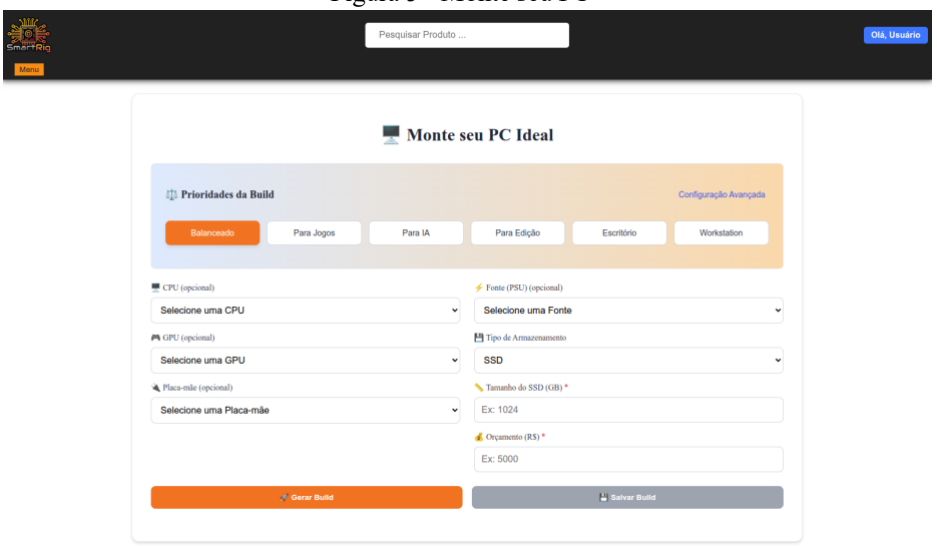
Fonte: Elaborado pelos autores (2025)

Figura 4 - Menu do usuário logado



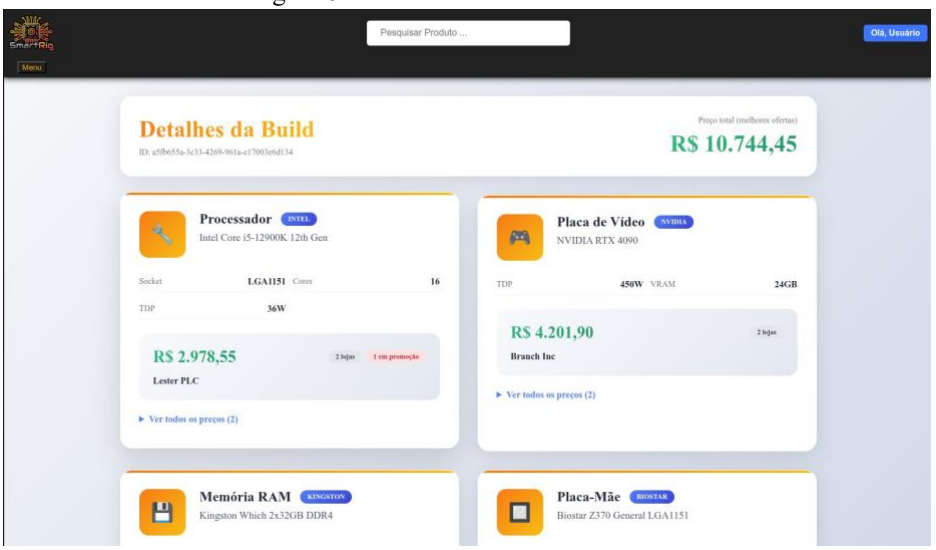
Fonte: Elaborado pelos autores (2025)

Figura 5 - Monte seu PC



Fonte: Elaborado pelos autores (2025)

Figura 6 - Detalhes de um build favorita



Fonte: Elaborado pelos autores (2025)

5.5 COMPARAÇÃO COM OUTROS PROJETOS

O sistema desenvolvido neste projeto distingue-se das soluções tradicionais de comparação de preços ao integrar, de forma automática, a montagem de builds de computador com apoio de algoritmos genéticos. Plataformas como Zoom, Buscapé e PCPartPicker limitam-se à apresentação de produtos, busca manual e verificação de compatibilidade; em contraste, o sistema aqui proposto gera automaticamente configurações otimizadas, respeitando orçamento, compatibilidade técnica e perfil de uso.

Utilizando operadores genéticos personalizados, crossover consciente e mutação restrita, foi possível alcançar, no perfil de inteligência artificial com peso de 60% para GPU, um aumento significativo do fitness máximo, de 46,91 na geração inicial para 166,84 na geração 49. A média de fitness, que começou em –494,46, estabilizou-se entre 50 e 110 a partir da geração 20, mostrando progresso consistente. Já no perfil balanceado (CPU 45%/GPU 45%), o fitness máximo estagnou em 57,27 desde o início, indicando que o algoritmo caiu em um ótimo local sem conseguir evoluir, evidência clara da influência decisiva das definições de peso no desempenho da evolução.

Essa dinâmica confirma que o algoritmo genético se beneficia de objetivos bem delineados, como o perfil de IA, onde a pressão seletiva sobre componentes cruciais orienta melhor a busca. Ainda que a diversidade populacional tenha se mantido alta (desvio-padrão entre 150 e 300), a persistência de indivíduos inviáveis com fitness fixo de –1000 ressalta os desafios impostos pelas restrições de compatibilidade e orçamento.

A literatura recente confirma a eficácia de algoritmos evolutivos em contextos semelhantes. No estudo de Muthu et al. (2018), os autores aplicaram uma combinação de algoritmos genéticos com técnicas de forrageamento bacteriano para otimização de escalonamento e alocação de recursos em ambientes de computação em nuvem, obtendo melhores resultados do que abordagens tradicionais em termos de eficiência e adaptabilidade. Já o artigo de Wang et al. (2022) explora o uso de algoritmos genéticos em contextos de projeto e otimização computacional, reforçando a robustez dessas técnicas em problemas com múltiplas restrições e critérios conflitantes.

Esses trabalhos reforçam a relevância técnica e científica da abordagem adotada neste projeto, especialmente em problemas do tipo combinatório e com objetivos múltiplos, como é o caso da montagem de computadores personalizados. Em comparação com soluções como BuildMyPC e DreamPCBuilder, que operam com regras fixas ou aprendizado supervisionado , o uso de algoritmos genéticos conscientes de compatibilidade e guiados por objetivos definidos oferece maior flexibilidade e melhor capacidade de adaptação, conforme demonstrado empiricamente nos experimentos realizados.

Com isso, o sistema desenvolvido não apenas incorpora métodos modernos de inteligência computacional, mas também oferece um diferencial prático e validado por dados reais, consolidando-se como uma solução eficiente e inteligente para a seleção automática de componentes de hardware.

6 CONCLUSÃO

O presente Trabalho de Conclusão de Curso conseguiu obter êxito em atingir seu objetivo geral de desenvolver um sistema de comparação de preços com agente inteligente para a criação de configurações de computador. A plataforma web, construída com React para

o front-end, Django para o back-end e PostgreSQL como banco de dados, demonstrou a viabilidade da integração dessas tecnologias para a finalidade proposta.

O componente central, o algoritmo genético implementado com a biblioteca DEAP, mostrou-se eficaz na geração de configurações de hardware personalizadas, otimizando a seleção de peças com base no orçamento, preferências do usuário, desempenho e, crucialmente, na compatibilidade entre os componentes. Os objetivos específicos foram alcançados, incluindo a criação de um banco de dados (embora com dados simulados para esta etapa), o desenvolvimento de um sistema de recomendação funcional, a implementação de uma interface web interativa e a garantia de compatibilidade entre as peças sugeridas. A simulação da atualização de preços também validou o mecanismo proposto para futuras integrações.

A principal limitação encontrada foi a ausência de acesso a APIs de lojas reais para a coleta de preços em tempo real, o que levou à utilização de dados simulados. No entanto, essa abordagem permitiu validar a lógica e a funcionalidade do sistema, que se mostrou robusto e capaz de cumprir suas funções caso integrado a fontes de dados reais.

Os resultados obtidos são promissores, indicando que o sistema desenvolvido representa uma ferramenta valiosa para auxiliar usuários, especialmente aqueles com conhecimento técnico limitado, no complexo processo de montagem de computadores. Ao automatizar a verificação de compatibilidade e otimizar a relação custo-benefício, o projeto contribui para uma tomada de decisão mais informada e eficiente.

7 TRABALHOS FUTUROS

Como trabalhos futuros, uma das principais direções a serem consideradas é a integração com APIs de e-commerces reais. Essa conexão permitiria a obtenção dinâmica e em tempo real de informações sobre disponibilidade, preços atualizados e promoções de componentes, tornando o sistema mais útil e aplicável no contexto prático. A automatização dessa coleta de dados também contribuiria para a escalabilidade da aplicação, reduzindo a necessidade de atualizações manuais no banco de dados.

Outro aspecto importante para o aprimoramento do projeto é a ampliação e o enriquecimento do banco de dados de componentes. Isso inclui não apenas o aumento do

número de itens cadastrados, mas também a inserção de informações mais detalhadas sobre especificações técnicas, compatibilidades e histórico de desempenho. Além disso, o refinamento contínuo do algoritmo genético, com a introdução de novas heurísticas e critérios de otimização, permitirá recomendações mais personalizadas e eficientes para diferentes perfis de usuários.

Por fim, a inclusão de avaliações e feedbacks de usuários pode agregar uma camada qualitativa à análise dos componentes recomendados, aproximando o sistema das preferências reais do público. A exploração de outras técnicas de inteligência artificial, como redes neurais ou sistemas híbridos de recomendação, também representa uma oportunidade promissora para o futuro do projeto. Essas inovações têm o potencial de elevar significativamente a qualidade das recomendações e de proporcionar uma experiência mais fluida, interativa e inteligente ao usuário final.

REFERÊNCIAS

YAMAKAMI, L. Indústria de games no Brasil cresceu sete vezes em dez anos.Veja, 16 ago. 2024. Disponível em: <https://veja.abril.com.br/economia/industria-de-games-no-brasil-cresceu-sete-vezes-em-dez-anos/>. Acesso em: 20 mar. 2025

BROWN, Megan A.; GRUEN, Andrew; MALDOFF, Gabe; MESSING, Solomon;SANDERSON, Zeve; ZIMMER, Michael. Web Scraping for research: legal, ethical, institutional, and scientific considerations. 30 Out. 2024. **arXiv.org**. Disponível em: https://arxiv.org/abs/2410.23432. Acesso em: 19 jun. 2025

FOERDERER, Jens. Should we trust web-scraped data? 4 Ago. 2023. arXiv.org. Disponivel em: https://arxiv.org/abs/2308.02231. Acesso em: 19 jun. 2025

LU, Yingzhou; CHEN, Lulu; ZHANG, Yuanyuan; SHEN, Minjie; WANG, Huazheng; WANG, Xiao; CAPUCINE, Van Rechem; FU, Tianfan; WEI, Wenqi. Machine Learning for Synthetic Data Generation: A review. 8 Feb. 2023. arXiv.org. Disponível em: https://arxiv.org/abs/2302.04062. Acesso em: 19 jun. 2025

MISHRA, Someshkumar; BANE, Shreyas; PANDIT, Rahul; PHADNIS, Neelam. PC Configuration and component Recommendation System. 2021. Disponível em: https://www.ijcaonline.org/archives/volume183/number11/31969-2021921411/. Acesso em: 19 jun. 2025

MARQUET. The Evolution of E-Commerce. Disponível em: https://www.marquet.cloud/the-evolution-of-e-commerce. Acesso em: 19 jun. 2025.

RESEARCHGATE. A History of E-Commerce. Disponível em: https://www.researchgate.net/publication/353718685_A_History_of_e-Commerce. Acesso em: 19 jun. 2025.

SPRINGEROPEN. A historical overview and analysis of e-commerce's milestones and its growing connection with air transport. Disponível em: https://jshippingandtrade.springeropen.com/articles/10.1186/s41072-025-00203-5. Acesso em: 19 jun. 2025.

GOODFELLOW, Ian et al. Deep Learning. MIT Press, 2016. Disponível em: https://www.deeplearningbook.org/. Acesso em: 20 jun. 2025.

HOLLAND, John H. Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press, 1975.

MITCHELL, Melanie. An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press, 1998.

COMBS; ANDREWS. Combinatorial rule explosion eliminated by a fuzzy rule configuration. 28 Fev. 1998. IEEE Xplore. Disponível em: https://ieeexplore.ieee.org/document/660804. Acesso em: 20 jun. 2025

KONAK, A.; COIT, D. W.; SMITH, A. E. Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering & System Safety, [s. l.], vol. 91, no. 9, p. 992–1007, 10 Jan. 2006. DOI 10.1016/j.ress.2005.11.018. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0951832005002012. Acesso em: 20 jun. 2025

PURNAPRAJNA, M.; REFORMAT, M.; PEDRYCZ, W. Genetic algorithms for hardware–software partitioning and optimal resource allocation. Journal of Systems Architecture, [s. l.], vol. 53, no. 7, p. 339–354, 9 Dec. 2006. DOI 10.1016/j.sysarc.2006.10.012. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S1383762106001342. Acesso em: 20 jun. 2025

BAELDUNG. Genetic Algorithms vs Neural Networks. 18 Mar. 2024. Baeldung. Disponível em: https://www.baeldung.com/cs/genetic-algorithms-vs-neural-networks. Acesso em: 20 jun. 2025

YANG, F.; XIN, L.; GUO, P. Y. A Multi-Objective Optimization genetic Algorithm for SOPC Hardware-Software partitioning. Advanced Materials Research, [s. l.], vol. 457–458, p. 1142–1148, 1 Jan. 2012. DOI 10.4028/www.scientific.net/amr.457-458.1142. Disponível em: https://www.researchgate.net/publication/283034429_A_Multi-Objective_Optimization_Genetic_Algorithm_for_SOPC_Hardware-Software_Partitioning. Acesso em: 20 jun. 2025

WANG, Y.; SHI, W. An automated approach to computer hardware design using genetic algorithm optimization. 10 jun. 2024. Disponível em: https://journal.esrgroups.org/jes/article/view/4338. Acesso em: 20 jun. 2025

MUTHU, A.; ENOCH, S. Optimized scheduling and resource allocation using evolutionary algorithms in cloud environment. International Journal of Intelligent Engineering and Systems, [s. l.], vol. 10, no. 5, p. 125–133, 19 Aug. 2017. DOI 10.22266/ijies2017.1031.14. Disponível em: https://www.researchgate.net/publication/320144230_Optimized_Scheduling_and_Resource_Allocation_Using_Evolutionary_Algorithms_in_Cloud_Environment. Acesso em: 20 jun. 2025

APÊNDICE

APÊNDICE A - Retorno do algoritmo genético


```
{
    "cpu": {
        "name": "AMD science",
        "image": "https://dummyimage.com/514x450",
        "brand": "AMD",
        "igpu": "02a5e859-ae16-4205-81c5-ff72d6d0f85c",
        "socket": "AM4",
        "tdp": 42,
        "cores": 16,
        "speed": 2298,
        "turbo": 4624,
        "price": {
            "store": "218539ba-207f-4919-bacb-2fe39dd6276a",
            "url_product":
"https://jones-howell.com/produtos/amd-science?ref=5388",
            "sale": false,
            "price": 4694.31,
            "old_price": null,
            "sale_percent": null,
            "sale_end": null,
            "content_type": 10,
            "object_id": "057ef6e6-ba66-4338-a1f0-13a49d7d4e1b",
            "uid": "1381d49a-f962-431d-8415-4d104dcf1d26"
        }
    },
    "gpu": {
        "name": "Radeon Vega 3",
        "brand": "AMD",
        "memory": 2,
        "speed": 1000,
        "turbo": 1200,
        "price": 0
    },
    "psu": {
        "name": "Pearson-Summers believe",
        "image": "https://dummyimage.com/73x461",
        "brand": "EVGA",
        "type": "SFX",
        "wattage": 450,
        "efficiency": "80+ Gold",
        "modular": "None",
        "price": {
            "store": "218539ba-207f-4919-bacb-2fe39dd6276a",
            "url_product":
"https://jones-howell.com/produtos/pearson-summers-believe?ref=6652",
            "sale": false,
            "price": 755.63,
            "old_price": null,
            "sale_percent": null,
            "sale_end": null,
```

```
            "content_type": 4,
            "object_id": "daa3ce2f-79c8-408e-a495-90b0e92416c2",
            "uid": "61032ad9-46ca-4364-8436-51f4c1d67c07"
        }
    },
    "mobo": {
        "name": "Palmer Group appear",
        "image": "https://dummyimage.com/97x454",
        "brand": "Biostar",
        "socket": "AM4",
        "form_factor": "Micro ATX",
        "memory_max": 128,
        "memory_type": "DDR4",
        "memory_slots": 4,
        "chipset": "B450",
        "m2_nvme": 1,
        "m2_sata": 0,
        "price": {
            "store": "5a3e296c-7285-48ff-ad20-01ee00cc9e86",
            "url_product":
"https://rollins-cortez.com/produtos/palmer-group-appear?ref=4630",
            "sale": false,
            "price": 892.44,
            "old_price": null,
            "sale_percent": null,
            "sale_end": null,
            "content_type": 3,
            "object_id": "109845e2-f4d2-41dc-9d2e-0a68d8d70154",
            "uid": "fef65843-d74c-4c4d-93a3-0fe752d10916"
        }
    },
    "ram": {
        "name": "G.Skill card",
        "image": "https://dummyimage.com/458x991",
        "brand": "G.Skill",
        "memory_type": "DDR4",
        "memory_size": 32,
        "memory_modules": 2,
        "memory_speed": 3595,
        "price": {
            "store": "749494b3-68fb-4405-8e0f-ad9f09349fb3",
            "url_product":
"https://white,caldwellandtaylor.com/produtos/g.skill-card?ref=3046",
            "sale": false,
            "price": 1093.67,
            "old_price": null,
            "sale_percent": null,
            "sale_end": null,
            "content_type": 5,
            "object_id": "79070c00-0d6d-4b2a-a97c-9b928b2c95f4",
            "uid": "572eb54d-fcbd-4d37-b7d4-2b855f47c3cf"
        }
    },
    "storage": {
        "name": "Meyers PLC better",
        "image": "https://dummyimage.com/417x644",
```

```
        "brand": "Christian, Wood and Ramos",
        "type": "HDD",
        "capacity": 1024,
        "form_factor": "3.5",
        "interface": "SATA",
        "price": {
            "store": "c4d8933d-2dfe-4500-8f07-32db507342a6",
            "url_product":
"https://morales-ramirez.com/produtos/meyers-plc-better?ref=4261",
            "sale": true,
            "price": 416.83,
            "old_price": 557.93,
            "sale_percent": 25,
            "sale_end": "2025-06-02T23:20:46.588Z",
            "content_type": 6,
            "object_id": "0af29d69-679a-4032-b5ba-79ebbbd00761",
            "uid": "14b0330f-f75d-4b3d-b8b1-a4f096753998"
        }
    },
    "total_price": 7852.880000000001,
    "fitness": 401.2158095824205
}
```


Fonte: Elaborada pelos autores (2025)

APÊNDICE B - REPOSITÓRIO DE CÓDIGO-FONTE DO PROJETO

Endereço de Acesso: https://github.com/Eduardo12305/SmartRig

Código QR para acesso rápido: