



UNIVERSIDADE FEDERAL DO MARANHÃO  
BACHARELADO INTERDISCIPLINAR EM CIÊNCIA E TECNOLOGIA  
EECP0011 - PROJETO E DESENVOLVIMENTO DE SOFTWARE

Arlison Gaspar de Oliveira

Ítalo Francisco Almeida de Oliveira

Gustavo de Oliveira Rego Moraes

Joao Pedro Miranda Sousa

Cauã Gabriel Santos Barros

**GRUPO 6 – Montagem de PC Online**

**São Luis – MA**

**Junho/2025**

Arlison Gaspar de Oliveira(2022017213)

Ítalo Francisco Almeida de Oliveira(2022017750)

Gustavo de Oliveira Rego Moraes(2021053091)

Joao Pedro Miranda Sousa(2022011087)

Cauã Gabriel Santos Barros(20240045292)

## **DOCUMENTAÇÃO - SISTEMA DE MONTAGEM DE PC ONLINE**

Documento apresentado como requisito parcial de avaliação da disciplina Projeto e Desenvolvimento de Software - Turma 01, no curso Bacharelado Interdisciplinar em Ciência e Tecnologia da Universidade Federal do Maranhão.

Orientador: Prof. Dr. Thales Levi Azevedo Valente.

**São Luis – MA**

**Junho/2025**

**RESUMO**

Este projeto tem como objetivo desenvolver uma plataforma online inteligente para montagem de PCs, utilizando integração com APIs de IA para oferecer recomendações personalizadas de hardware. O sistema analisa orçamento, finalidade de uso (games, edição, trabalho) e localização do usuário para sugerir configurações otimizadas, considerando desempenho, custo-benefício e disponibilidade regional de peças. Além disso, realiza verificações automáticas de compatibilidade entre componentes (CPU, placa-mãe, RAM, GPU), calcula o TDP total e a demanda térmica do sistema, garantindo um equilíbrio entre potência e refrigeração. A plataforma oferece tanto montagem assistida por IA quanto modo manual com alertas em tempo real, tornando o processo acessível para iniciantes enquanto mantém flexibilidade para usuários avançados.

**Palavras-chave:** montagem de PC, IA, compatibilidade, TDP, demanda térmica, orçamento, configuração otimizada, plataforma online, hardware

## SUMÁRIO

1 INTRODUÇÃO	5
1.1 Visão geral do Documento	5
1.2 Escopo do projeto	6

1.2.1 Problema	6
1.2.2 Justificativa	7
1.3 Concepção do Sistema	7
1.3.1 Entrevista	7
1.3.2 Análise de mercado	8
1.3.3 Requisitos	8
2 OBJETIVO	8
3 ELICITAÇÃO DE REQUISITOS	9
3.1 Requisitos Funcionais (RF)	9
3.2 Requisitos Não Funcionais (RNF)	11
4 DIAGRAMS DE CASO DE USO	12
4.1 Diagramação	12
4.2 Descrição dos casos de uso	17
4.2.1 Criar e Gerenciar Perfil de Usuário	17
4.2.2 Coletar Requisitos e Preferências do Usuário	19
4.2.3 Gerar Configuração Automatizada de PC	22
4.2.5 Recomendação de Sistema de Refrigeração Inteligente	26
5 DIAGRAMA DE ATIVIDADE	29
6 DIAGRAMA DE ESTADOS	41
7 DIAGRAMA DE SEQUENCIA	55
8 DIAGRAMA DE CLASSES	70
9 TECNOLOGIAS	87
10 ESCOPO DA PROTOTIPAÇÃO	89
11 CONCLUSÃO	90
12 REFERENCIAS	91

## **1. INTRODUÇÃO**

### **1.1 Visão geral do Documento**

Este documento tem como objetivo apresentar de forma clara e estruturada os aspectos fundamentais do projeto Montador de PC Online, idealizado a partir de uma entrevista com o Product Owner. São descritos o escopo, o problema abordado, a justificativa da proposta, a concepção inicial do sistema, a análise de mercado e os principais requisitos funcionais e não funcionais. Essa documentação servirá como base para o desenvolvimento técnico, comunicação entre stakeholders e acompanhamento do progresso do projeto.

## 1.2 Escopo do projeto

O projeto visa desenvolver uma plataforma web interativa para recomendação, personalização e exportação de configurações de computadores, com base no orçamento, perfil de uso e ambiente físico do usuário. A plataforma contará com recursos de inteligência artificial, coleta de requisitos via chatbot, análise de compatibilidade entre componentes e sugestão de builds otimizadas, incluindo links diretos para compra.

### 1.2.1 Problema

Atualmente, a montagem de computadores é um processo complexo para usuários comuns e até mesmo para empresas, exigindo tempo, conhecimento técnico e comparação entre inúmeras variáveis. A dificuldade em encontrar peças compatíveis e otimizadas para diferentes perfis de uso (jogos, escritório, servidores, etc.) dentro de um orçamento limitado representa um entrave significativo para muitos consumidores.

**Tabela 1** – Problemas indentificados

Problema Identificado	Descrição
Dificuldade na escolha de componentes compatíveis	Usuários sem conhecimento técnico enfrentam desafios para montar PCs com peças que funcionem bem entre si.
Falta de otimização com base no perfil de uso	Plataformas atuais não adaptam sugestões de hardware a diferentes perfis (gamer, escritório, edição de vídeo etc.).
Montagens fora do orçamento	Usuários muitas vezes recebem sugestões que ultrapassam o valor que podem gastar, sem alternativas viáveis.
Ausência de consideração de fatores ambientais	Nenhum sistema comum considera temperatura, poeira ou ventilação do ambiente onde o PC será usado.
Processo de escolha demorado e confuso	A busca manual por peças e comparação de preços é cansativa, especialmente para leigos.
Falta de orientação e suporte durante a montagem	Usuários não recebem alertas ou instruções claras sobre erros ou incompatibilidades ao montar o PC.

Dificuldade em encontrar os melhores preços	Não há integração direta com lojas para encontrar as ofertas mais vantajosas em tempo real.
Necessidade de conhecimento técnico prévio	Usuários iniciantes não conseguem montar configurações sem aprender conceitos avançados de hardware.
Impossibilidade de salvar ou exportar a configuração criada	Plataformas atuais não permitem salvar, compartilhar ou exportar builds de forma conveniente.
Falta de automação na coleta de requisitos	A definição das necessidades do usuário costuma ser manual, sem um processo guiado ou inteligente.

**Fonte:** Autoria propria,2025

### 1.2.2 Justificativa

O projeto busca democratizar o acesso à tecnologia e otimizar o processo de montagem de PCs personalizados, oferecendo sugestões inteligentes com base em IA. A solução se destaca por considerar não apenas fatores como orçamento e desempenho, mas também aspectos físicos do ambiente, como temperatura e fluxo de ar, diferencial ausente em concorrentes. Essa abordagem visa atrair tanto usuários finais quanto empresas que desejam eficiência na aquisição de hardware.

## 1.3 Concepção do Sistema

O sistema será desenvolvido como uma aplicação web responsiva, com um fluxo de uso guiado (wizard). Inicialmente, o usuário poderá escolher entre duas abordagens: recomendação automática com base em requisitos fornecidos ou montagem manual. A plataforma permitirá salvar e exportar builds, e utilizará integração com APIs externas e técnicas de web scraping para exibir os preços atualizados dos componentes.

### 1.3.1 Entrevista

A entrevista realizada com o Product Owner Arlison Gaspar evidenciou a necessidade de um sistema que una praticidade, inteligência e precisão na montagem de PCs. Foram identificadas funcionalidades essenciais para o MVP, como coleta de requisitos via chatbot, avaliação de compatibilidade, sugestão de builds e exportação em formatos populares (PDF, XLSX). Além disso, definiu-se que o cadastro será opcional, mas necessário para o salvamento das configurações.

### 1.3.2 Análise de mercado

A proposta se diferencia de outras soluções existentes no mercado por incluir variáveis ambientais na recomendação das configurações, aplicar regras de negócio adaptáveis a diferentes perfis de usuário e oferecer atualizações periódicas de preços. O sistema não pretende competir diretamente com lojas virtuais, mas sim atuar como um intermediador inteligente que facilita o processo de escolha e aquisição dos componentes.

### 1.3.3 Requisitos

Foram definidos requisitos funcionais como cadastro de usuário, recomendação automatizada de peças, montagem manual, avaliação de compatibilidade e sugestão de upgrades orçamentários. Entre os requisitos não funcionais, destacam-se: suporte a IA, responsividade, sincronização semanal de dados, tempo de resposta inferior a 6 segundos, segurança conforme LGPD e alta disponibilidade. A modelagem térmica e a integração com dados climáticos também são aspectos inovadores incorporados ao projeto.

## 2. OBJETIVO

O projeto Montador de PC Online tem como finalidade principal oferecer uma solução prática, inteligente e acessível para montagem personalizada de computadores, atendendo diferentes perfis de uso e respeitando as limitações orçamentárias e ambientais dos usuários. A seguir, são listados os objetivos principais do sistema:

**Tabela 2** – Objetivos específicos

Nº	Objetivo Específico
1	Coletar requisitos do usuário de forma simples e intuitiva, utilizando uma interface de chatbot.
2	Sugerir configurações completas de PC baseadas em IA, respeitando o orçamento e as preferências do usuário.
3	Avaliar, em tempo real, a compatibilidade entre os componentes selecionados automaticamente ou manualmente.
4	Permitir a substituição manual das peças sugeridas, mantendo alertas de compatibilidade.
5	Apresentar alternativas quando o orçamento for insuficiente, sugerindo upgrades de verba ou builds reduzidas.



6	Ajustar sugestões conforme o ambiente físico (temperatura, ventilação, poeira) onde o PC será instalado.
7	Oferecer links de compra atualizados, priorizando preço e confiabilidade das lojas.
8	Disponibilizar recursos para salvar e exportar builds em formatos como PDF e XLSX.
9	Garantir uma experiência de navegação fluida e responsiva em dispositivos móveis e desktop.
10	Assegurar a privacidade e segurança dos dados dos usuários, em conformidade com a LGPD.

**Fonte:** Aatoria propria,2025

### 3. ELICITAÇÃO DE REQUISITOS

Nesta seção são apresentados os requisitos essenciais levantados durante o processo de entrevista com o Product Owner (PO). Eles estão organizados em **Requisitos Funcionais (RF)** e **Requisitos Não Funcionais (RNF)**, representando respectivamente as funcionalidades esperadas do sistema e as qualidades que ele deve apresentar.

#### 3.1 Requisitos Funcionais (RF)

Os Requisitos Funcionais descrevem os comportamentos e funções específicas que o sistema deverá executar:

**Tabela 3** – Requisitos funcionais

ID	Nome	Descrição
RF-01	Cadastro de Usuário	Permitir que o usuário se registre (de forma opcional) para salvar builds e receber promoções.
RF-02	Definição de Orçamento	Coletar o valor máximo que o usuário deseja gastar na montagem do computador.
RF-03	Recomendação Automatizada	Gerar sugestões de peças com base nos requisitos e no orçamento informados.
RF-04	Montagem Manual	Permitir que o usuário edite ou substitua peças manualmente após a sugestão inicial.

RF-05	Avaliação de Compatibilidade	Verificar, em tempo real, se as peças seleccionadas são compatíveis entre si.
RF-06	Salvar Build	Permitir salvar a configuração de PC no perfil do usuário.
RF-07	Exportar Build	Gerar arquivos PDF ou XLSX com as peças, preços e links de compra.
RF-08	Sugerir Upgrade de Orçamento	Sugerir aumento de orçamento médio quando a configuração ultrapassar o valor definido.
RF-09	Build Alternativa Inferior	Oferecer alternativa com menor desempenho caso o orçamento seja insuficiente.
RF-10	Links de Compra	Apresentar links atualizados com as melhores ofertas para os componentes sugeridos.
RF-11	Perfil de Uso	Coletar ou inferir o perfil do usuário (gamer, escritório, etc.) para ajustar recomendações.
RF-12	Sugestão por Ambiente	Adaptar recomendações com base em dados ambientais (temperatura, poeira, ventilação).
RF-13	Chatbot	Utilizar um chatbot para guiar a coleta de requisitos de forma intuitiva e interativa.
RF-14	Deteccção de Localização (IP)	Estimar a cidade do usuário via GeoIP para fins de sugestão térmica, com consentimento.
RF-15	Coleta de Dados Ambientais	Perguntar se o ambiente do PC é climatizado, fechado ou empoeirado.
RF-16	Validação de Gabinete	Verificar se o gabinete escolhido é compatível com a configuração sugerida (dimensões, airflow etc.).
RF-17	Recomendação de Refrigeração	Calcular demanda térmica e sugerir ventilação adequada ou soluções como water cooler.

Fonte: Autoria Propria,2025

### 3.2 Requisitos Não Funcionais (RNF)

Os Requisitos Não Funcionais definem atributos de qualidade que o sistema deve possuir, garantindo confiabilidade, desempenho e usabilidade:

**Tabela 4 – Requisitos não funcionais**

ID	Nome	Descrição
RNF-01	Suporte de IA	Utilizar modelos de linguagem (LLMs) ou heurísticas para recomendação automatizada.
RNF-02	Responsividade	O sistema deve funcionar corretamente em dispositivos móveis, tablets e desktops.
RNF-03	Atualização de Dados	Os dados de preços e estoque devem ser atualizados semanalmente (fallback: quinzenal/mensal).
RNF-04	Tempo de Resposta	O tempo total para geração de recomendações e validações não deve ultrapassar 6 segundos.
RNF-05	Fluxo Guiado	O sistema deve seguir um modelo "wizard", com etapas claras e uma visão de resumo ao final.
RNF-06	Segurança e LGPD	Implementar criptografia de dados e garantir conformidade com a Lei Geral de Proteção de Dados.
RNF-07	Compatibilidade de Navegadores	Garantir suporte aos navegadores atuais (Chrome, Firefox, Edge, Safari).
RNF-08	Disponibilidade	O sistema deve estar disponível no mínimo 99% do tempo, com monitoramento de uptime.
RNF-09	Backup e Recuperação	Realizar backups diários dos dados com testes periódicos de recuperação.
RNF-10	Escalabilidade	Suportar picos de acesso sem comprometer desempenho.
RNF-11	Integração de Fontes	Implementar retentativas e logging para falhas em APIs ou scraping.
RNF-12	Pré-orçamento	Solicitar teto de gastos antes de iniciar o processo de recomendação.

RNF-13	Organização Visual	Interface clara: chatbot à direita, tabela de peças à esquerda.
RNF-14	Latência Clima	Utilizar pop-up de consentimento para coleta automática de dados climáticos; caso negado, entrada manual.
RNF-15	Consentimento de Localização	Solicitar autorização para uso da localização aproximada do usuário.
RNF-16	Modelagem Térmica Extensível	Permitir configuração externa de TDP e regras térmicas para ajustes futuros.

**Fonte:** Autoria própria

## 4. DIAGRAMAS DE CASO DE USO

### 4.1 Diagramação

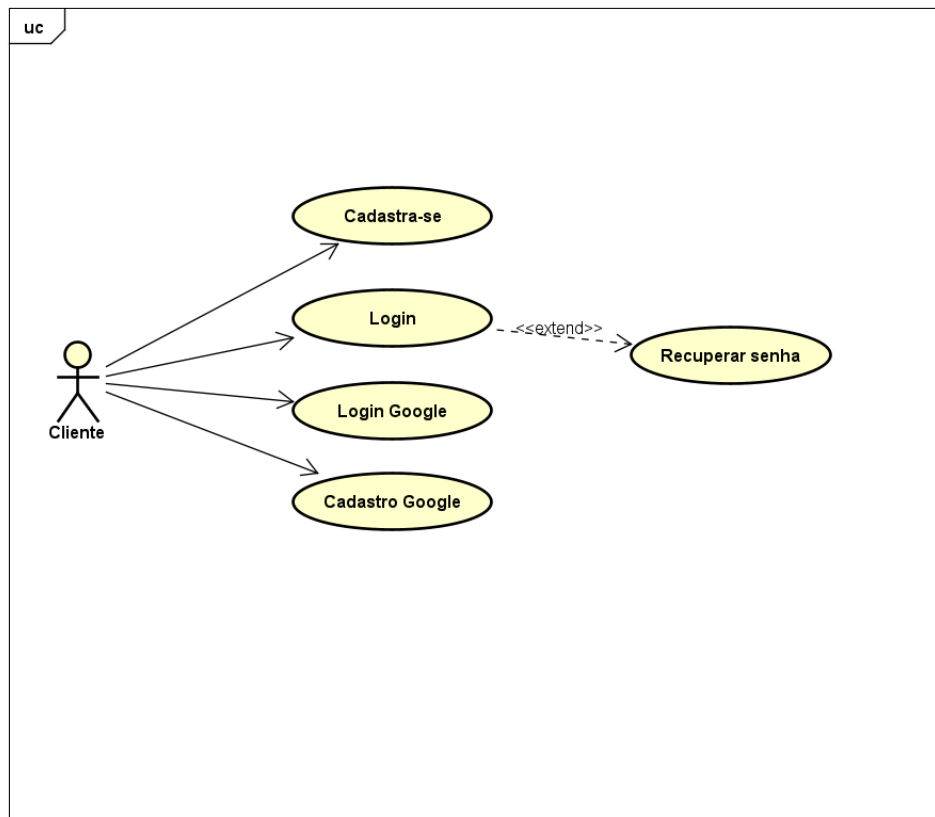
Os diagramas de caso de uso representam as funcionalidades do sistema e as interações entre os atores (usuários ou sistemas externos) e o sistema. Eles são essenciais para capturar os requisitos funcionais e definir o escopo do projeto.

Os componentes Principais dos diagramas de caso de uso são:

1. Atores: Representam os usuários ou outros sistemas que interagem com o sistema.
2. Casos de Uso: Representam as funcionalidades oferecidas pelo sistema.
3. Relacionamentos: Mostram como os atores interagem com os casos de uso.

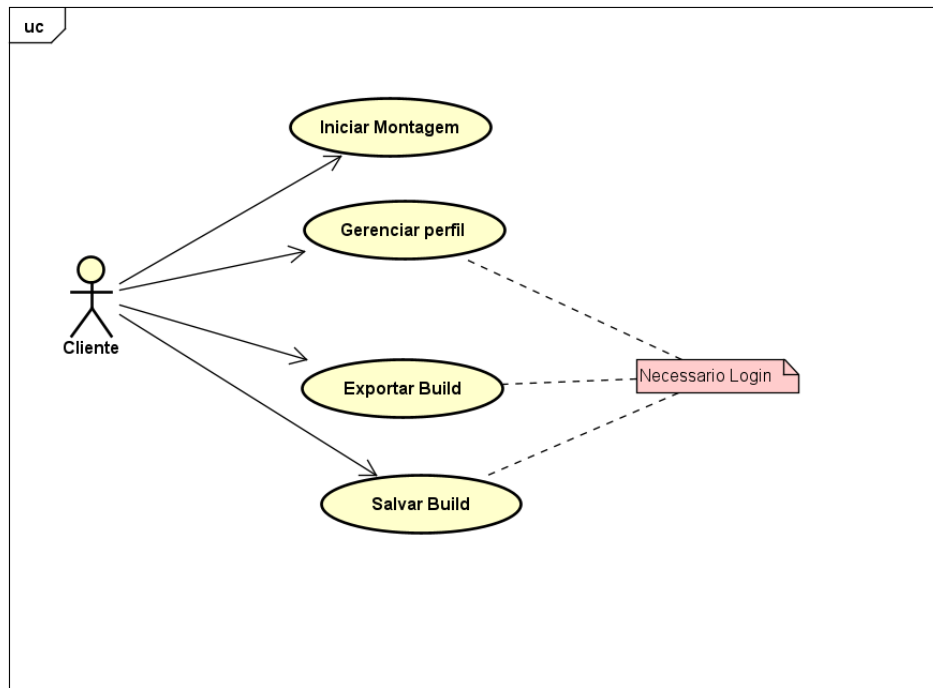
A seguir temos os diagramas dos casos de uso:

**Figura 1 – Cadastro e login**



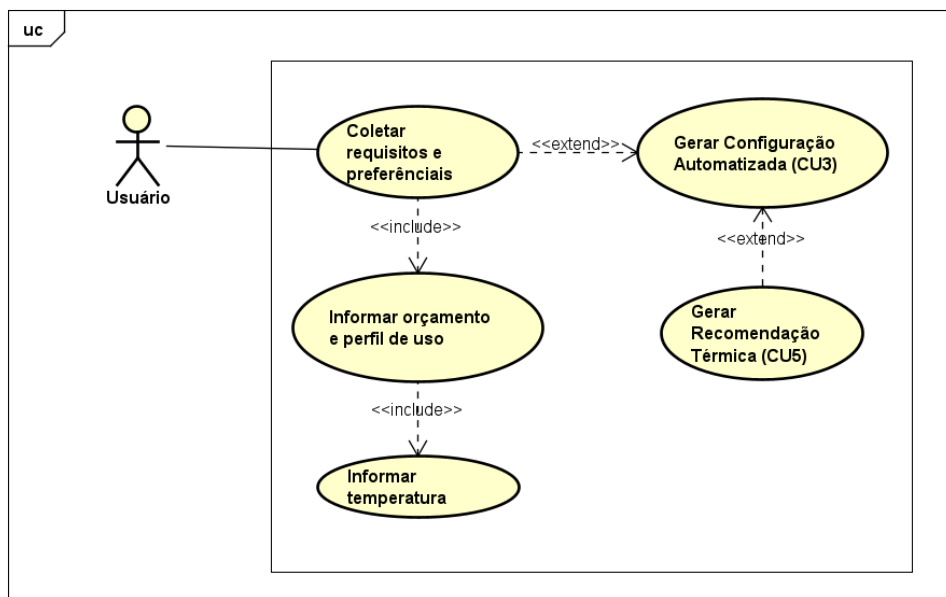
**Fonte:** Autoria própria

**Figura 2:** Ações após o login ou cadastro



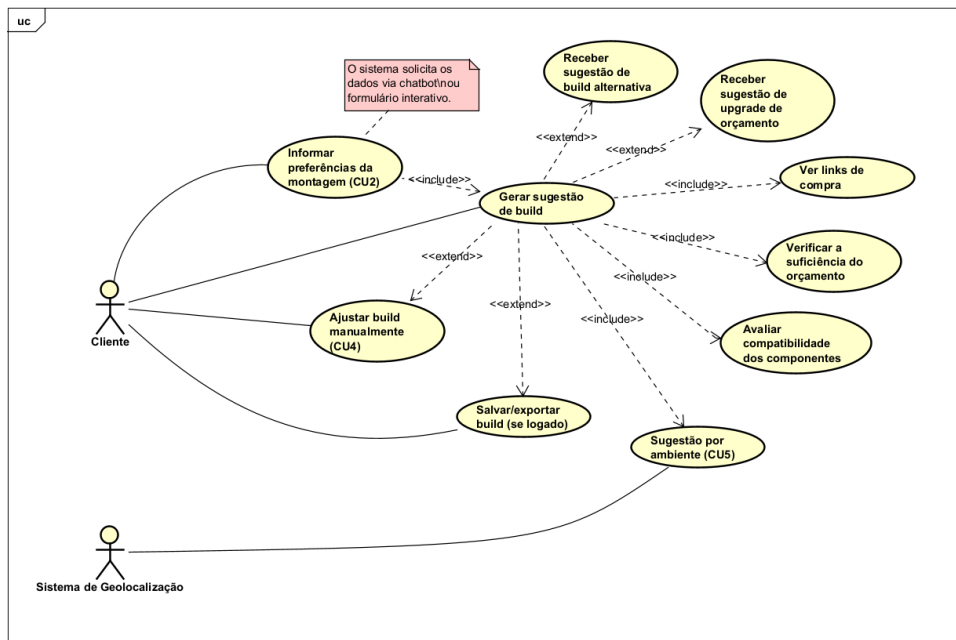
Fonte: Autoria própria

Figura 3 – Coleta de requisitos do usuário



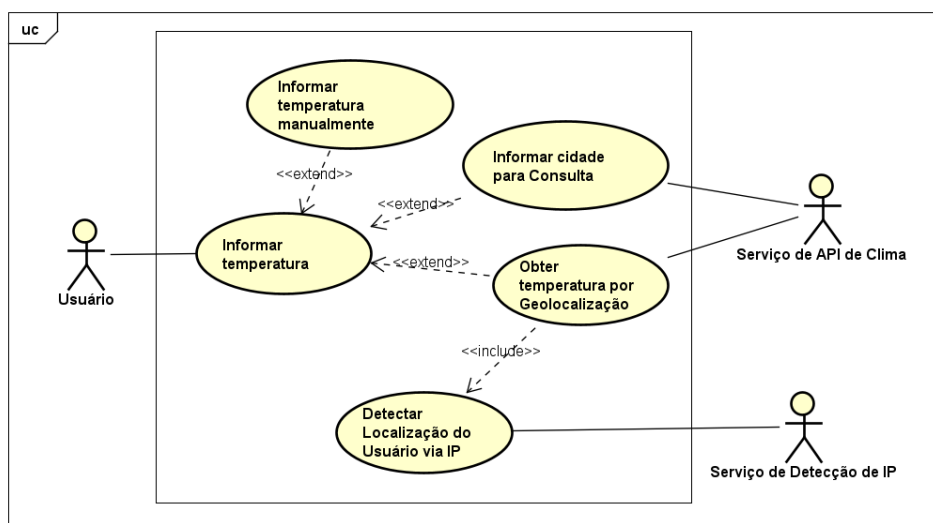
Fonte: Autoria propria

Figura 4 - Configuração automatizada de pc



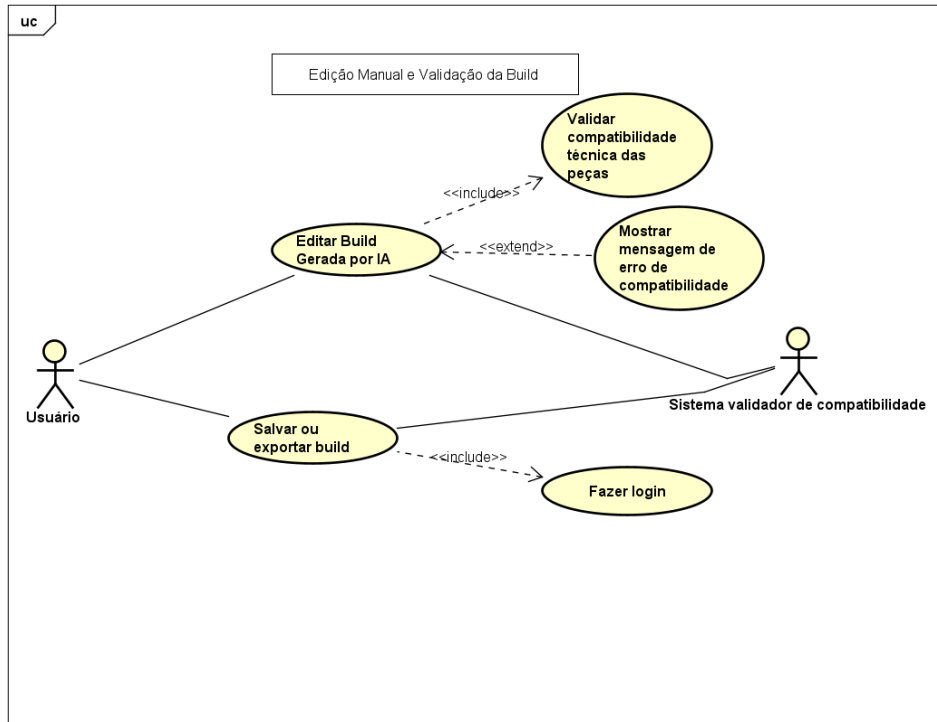
Fonte: Autoria propria

Figura 5 – Coleta de temperatura



Fonte: Autoria propria

Figura 6 – Montagem manual



Fonte: autoria propria



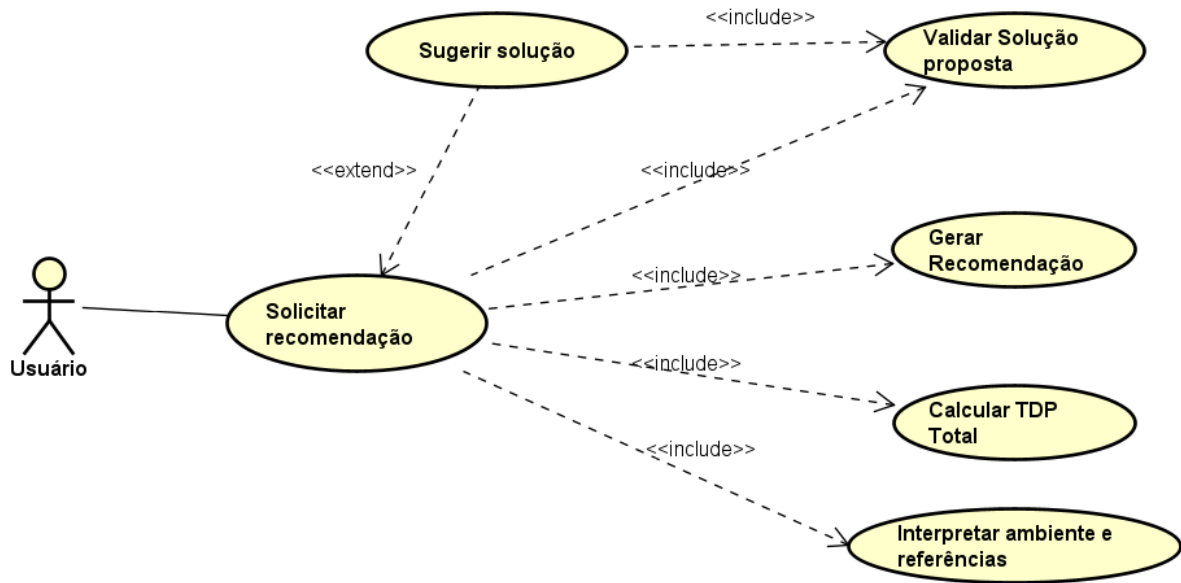


Figura 7 – Sistema de refrigeração

## 4.2 Descrição dos casos de uso

A descrição de casos de uso é uma documentação que descreve um caso de uso de forma detalhada. Ela fornece informações sobre o objetivo do caso de uso, os atores envolvidos, as pré-condições, as pós-condições, as ações dos atores e do sistema, o fluxo principal e os fluxos alternativos.

### 4.2.1 Criar e Gerenciar Perfil de Usuário

#### Descrição:

Permite que o usuário crie um perfil na plataforma para salvar builds, exportar configurações e receber promoções. O sistema oferece duas formas de acesso: por cadastro com e-mail/senha ou autenticação via conta Google (login automático). Usuários autenticados podem gerenciar seus dados pessoais e configurações salvas.

#### Atores:

- Usuário (pessoa física ou representante de empresa)
- Sistema (inclui integração com Google OAuth)

### **Pré-Condições:**

- Para salvar builds e exportar arquivos, o usuário deve estar autenticado.
- O usuário pode se cadastrar com:
  - a) E-mail e senha válidos, ou
  - b) Conta Google (OAuth 2.0).

### **Fluxo Principal:**

- O usuário acessa a opção “Entrar” ou “Criar conta”.
- O sistema oferece duas opções:
  - Cadastro/Login tradicional:
  - Usuário preenche e-mail e senha.
  - Sistema valida e autentica.
  - Login com Google:
  - Sistema redireciona para autenticação via Google.
  - Com autorização, o login é feito automaticamente e o perfil é criado (caso não exista).
- Após autenticação, o usuário pode:
  - Salvar configurações de montagem (builds).
  - Exportar builds em XLSX ou PDF.
  - Editar dados do perfil (nome, telefone, etc.).

### **Fluxo Alternativo:**

- Recuperação de senha (caso de login tradicional):
- Usuário seleciona “Esqueci minha senha”.
- Sistema envia link de redefinição para o e-mail informado.
- Usuário redefine a senha e retorna ao login.

### **Fluxos de Exceção:**

- E-mail já cadastrado (modo tradicional):
  - Sistema exibe: *“Este e-mail já está em uso. Faça login ou use outra conta.”*

- Login inválido (tradicional):
  - Mensagem: *“E-mail ou senha incorretos.”*
- Falha na autenticação com Google:
  - Mensagem: *“Erro ao conectar com o Google. Tente novamente.”*
- Falha na exportação da build:
  - Exibir: *“Erro ao gerar arquivo. Tente novamente mais tarde.”*

#### **Pós-Condições:**

- Usuário autenticado pode salvar e exportar builds.
- Dados pessoais e builds são armazenados com segurança (criptografia).
- Backups periódicos são realizados (quando aplicável).

#### **Requisitos Funcionais Relacionados:**

- RF-01 – Cadastro de Usuário
- RF-06 – Salvar Build
- RF-07 – Exportar Build

#### **Regras de Negócio:**

- RN01: Apenas usuários autenticados podem salvar builds.
- RN02: E-mails devem ser únicos.
- RN03: Senhas (para login tradicional) devem ter no mínimo 8 caracteres.
- RNxx (novo): A autenticação via conta Google deve seguir o protocolo OAuth 2.0 com consentimento do usuário.

#### **Relacionamento com Outros Casos de Uso:**

- Montar Computador Personalizado: o usuário precisa estar autenticado para salvar ou exportar builds.
- Gerenciar Orçamento e Builds: as configurações associadas ao perfil são usadas para recomendações e análises.

#### **4.2.2 Coletar Requisitos e Preferências do Usuário**

**Descrição:**

Permite ao sistema entender as necessidades do usuário para montar uma configuração de PC personalizada. As informações são coletadas por meio de chatbot e/ou formulários interativos, incluindo teto de orçamento, perfil de uso (ex.: jogos, escritório, edição de vídeo) e ambiente físico onde o computador será instalado (ex.: local quente, empoeirado, com ou sem climatização). Os dados coletados são usados para gerar recomendações automatizadas ou dar suporte à montagem manual.

**Atores:**

- Usuário (logado ou não)
- Sistema (inclui IA, chatbot, detecção de IP e APIs de clima)

**Pré-Condições:**

- O usuário acessa a funcionalidade de montagem de PC.
- Para salvar configurações futuras, o login é necessário.

**Fluxo Principal:**

- O usuário acessa a opção “Montar PC”.
- O sistema apresenta um assistente interativo via chatbot.
- O sistema solicita ao usuário:
  - Orçamento máximo disponível (RF-02).
  - Perfil de uso (RF-11), como “gamer”, “escritório”, “edição de vídeo”, “servidor”, etc.
  - Ambiente físico (RF-015), perguntando se o local é fechado, quente, empoeirado, climatizado, etc.
- O sistema pergunta se pode estimar a localização do usuário (via IP):
  - Se o usuário aceitar, coleta dados da cidade e busca temperatura ambiente máxima anual via API de clima (RF-14).
  - Se o usuário recusar, pede entrada manual da temperatura média.
- O sistema armazena as preferências em memória temporária ou no perfil (caso o usuário esteja autenticado).
- As informações são então encaminhadas ao mecanismo de recomendação.

**Fluxo Alternativo:**

- Repetir o processo:
- Usuário pode reiniciar a coleta de requisitos a qualquer momento.

#### **Fluxo de Exceção:**

- Resposta inválida (ex: texto em vez de número):
  - Sistema exibe mensagem: *“Informe um valor numérico válido para o orçamento.”*
- Recusa de localização:
  - Sistema bloqueia o avanço e solicita temperatura estimada.

#### **Pós-Condições:**

- O sistema possui dados suficientes para gerar uma recomendação personalizada.
- Os dados são armazenados com segurança (se logado) ou mantidos temporariamente na sessão.

#### **Requisitos Funcionais Relacionados:**

- RF-02 – Definição de Orçamento
- RF-11 – Perfil de Uso
- RF-13 – Chatbot
- RF-14 – Detecção de Localização (IP + API)
- RF-015 – Coleta de Dados Ambientais

#### **Regras de Negócio Relacionadas:**

- RN03: Sistema ajusta o tipo de build conforme o perfil (ex.: para “gamer”, prioriza GPU).
- RN05: Perfis-tipo como “Edição de Vídeo” ou “Servidor” influenciam distribuição do orçamento.
- RN-04: Caso o orçamento seja insuficiente, o sistema pode sugerir aumento ou build inferior (tratado no Caso de Uso 3).

#### **Relacionamento com Outros Casos de Uso:**

- Gerar Configuração Automatizada de PC (CU3): os dados coletados aqui são usados para montar a build.

- Gerenciar Perfil (CU1): se logado, as preferências podem ser salvas no perfil do usuário.
- Recomendação Térmica (CU5): os dados ambientais alimentam o cálculo de refrigeração ideal.

#### **4.2.3 Gerar Configuração Automatizada de PC**

##### **Descrição:**

Gera automaticamente uma configuração personalizada de computador com base nos dados fornecidos pelo usuário (orçamento, perfil de uso e ambiente). O sistema utiliza inteligência artificial para sugerir as peças mais adequadas, avalia a compatibilidade entre os componentes, detecta se o orçamento é suficiente e, quando necessário, sugere alternativas com menor custo ou recomenda o aumento do teto de gastos. Também associa links de compra das peças sugeridas.

##### **Atores:**

- Usuário (logado ou não logado)
- Sistema (IA de recomendação, validador de compatibilidade, integrador de preços)

##### **Pré-Condições:**

- O usuário deve ter fornecido as informações básicas: orçamento, perfil de uso e ambiente (ver CU2).
- Para salvar ou exportar a build, o usuário precisa estar autenticado.

##### **Fluxo Principal:**

- O sistema recebe os dados do usuário (orçamento, perfil, ambiente).
- A IA processa as informações e sugere uma build completa com base nos critérios:
  - Compatibilidade dos componentes
  - Prioridade de desempenho para o perfil informado
  - Custo-benefício dentro do orçamento disponível
- O sistema avalia se o orçamento é suficiente:
  - Se for suficiente, apresenta a build com peças, preços, links de compra e justificativa técnica.
  - Se for insuficiente, segue para o fluxo de sugestão (RF-08 e RF-09).

- O usuário visualiza a build e pode:
  - Aceitar a recomendação e prosseguir
  - Ajustar as peças manualmente (ligação com CU4)
  - Salvar ou exportar a configuração (se logado – CU1)

#### **Fluxo Alternativo:**

- Orçamento Insuficiente:
  - O sistema identifica que o orçamento não cobre os requisitos.
  - Sugere aumento do teto com explicação clara (RF-08), ou
  - Oferece uma build alternativa com desempenho inferior e aviso sobre as limitações (RF-09).
  - Usuário escolhe qual caminho seguir.

#### **Fluxo de Exceção:**

- Falha na IA de recomendação:
  - Exibe mensagem: *“Não foi possível gerar uma sugestão automática. Tente novamente ou opte pela montagem manual.”*
- Erro ao acessar links de lojas/parceiros:
  - Sistema omite link e marca peça com alerta: *“Preço temporariamente indisponível”*.

- **Pós-Condições:**
- Uma build é gerada automaticamente com base nas preferências e condições do usuário.
- O sistema fornece todos os dados necessários para salvar, exportar ou ajustar a configuração.

#### **Requisitos Funcionais Relacionados:**

- RF-03 – Recomendação Automatizada
- RF-08 – Sugerir Upgrade de Orçamento
- RF-09 – Build Alternativa Inferior
- RF-10 – Links de Compra
- RF-12 – Sugestão por Ambiente (com base na temperatura, poeira etc.)

#### **Regras de Negócio Relacionadas:**

- RN01: Distribuir o orçamento priorizando as peças mais importantes para o perfil de uso.
- RN02: Ordem de decisão da IA → compatibilidade > desempenho > custo-benefício.
- RN04: Quando orçamento é insuficiente, o sistema deve sempre oferecer explicações claras.

### **Relacionamento com Outros Casos de Uso:**

- Coletar Requisitos (CU2): usa os dados coletados para gerar a recomendação.
- Gerenciar Perfil (CU1): usuário logado pode salvar ou exportar a configuração.
- Montagem Manual (CU4): usuário pode ajustar a sugestão automática.
- Refrigeração e Ambiente (CU5): condições ambientais impactam a escolha dos componentes térmicos.

#### **4.2.4 Montagem Manual e Validação da Build**

##### **Descrição:**

Permite ao usuário personalizar a configuração de seu computador substituindo ou escolhendo manualmente os componentes desejados (CPU, GPU, RAM, etc.). O sistema realiza validações em tempo real para garantir a compatibilidade entre as peças selecionadas, incluindo o gabinete. Esse processo pode ocorrer a partir de uma sugestão automática previamente gerada ou desde o início da montagem.

##### **Atores:**

- Usuário (logado ou não logado)
- Sistema (validador de compatibilidade e validação física de gabinete)

##### **Pré-Condições:**

- O usuário deve ter iniciado o processo de montagem (via recomendação automática ou montagem manual direta).
- Para salvar ou exportar a build, é necessário estar logado.

##### **Fluxo Principal:**



- O usuário acessa a opção de montagem manual ou edita uma build gerada automaticamente.
- O sistema exibe as categorias de componentes (CPU, GPU, placa-mãe, RAM, armazenamento, gabinete, etc.).
- Para cada alteração ou adição de peça:
  - O sistema valida em tempo real a compatibilidade técnica (RF-05), como:
    - Soquete da CPU e placa-mãe
    - Capacidade e frequência da RAM suportada
    - Potência da fonte versus consumo estimado
- Se houver incompatibilidade, o sistema informa com uma mensagem clara (ex.: “*A placa-mãe escolhida não suporta esse processador*”).
- Quando o gabinete for selecionado ou modificado:
  - O sistema valida se ele comporta os componentes escolhidos (RF-016), considerando:
    - Tamanho da placa-mãe
    - Comprimento/altura da GPU
    - Altura do cooler ou radiador
    - Número e posição das ventoinhas
- O usuário pode finalizar a montagem, exportar ou salvar a build, desde que não haja incompatibilidades críticas.

#### **Fluxo Alternativo:**

- Alteração da Placa-Mãe para Formato Maior (ex.: ATX → E-ATX):
  - O sistema verifica novamente o gabinete e, se necessário, alerta que o gabinete atual é incompatível e oferece sugestões alternativas.
- Substituição de uma peça incompatível:
  - O sistema sugere modelos compatíveis com base nas peças já escolhidas.

#### **Fluxo de Exceção:**

- Incompatibilidade Crítica:
  - O sistema bloqueia o avanço e exibe: “*A configuração contém conflitos técnicos que impedem o funcionamento do PC. Corrija os itens destacados.*”
- Erro no carregamento de especificações de peças:

- Sistema avisa: *“Erro ao carregar dados técnicos da peça. Tente novamente ou selecione outro componente.”*

#### **Pós-Condições:**

- O usuário obtém uma configuração tecnicamente válida e personalizada.
- O sistema assegura que a build final seja funcional, com base nas validações feitas.

#### **Requisitos Funcionais Relacionados:**

- RF-04 – Montagem Manual
- RF-05 – Avaliação de Compatibilidade
- RF-016 – Validação de Gabinete

#### **Regras de Negócio Relacionadas:**

- RN02: A prioridade da IA e do sistema segue a ordem: compatibilidade > desempenho > custo.
- RN06: A compatibilidade técnica é obrigatória para salvar ou finalizar uma build.

#### **Relacionamento com Outros Casos de Uso:**

- Gerar Configuração Automatizada (CU3): o usuário pode ajustar manualmente a sugestão feita pela IA.
- Gerenciar Perfil (CU1): para salvar ou exportar a build personalizada.
- Refrigeração e Ambiente (CU5): validações de gabinete e espaço podem influenciar na escolha dos sistemas de resfriamento.

### **4.2.5 Recomendação de Sistema de Refrigeração Inteligente**

#### **Descrição:**

Este caso de uso permite que o sistema gere uma recomendação térmica personalizada e validada para a build do usuário, levando em conta os componentes escolhidos, o ambiente de uso e as preferências pessoais (como nível de ruído, temperatura e poeira).

A recomendação considera o TDP total, o tipo de gabinete, as condições do ambiente e o perfil térmico do usuário. O sistema então propõe uma solução composta por ventoinhas, air coolers ou AIOs, além de posições ideais de instalação.

A proposta gerada é automaticamente validada internamente quanto à viabilidade técnica (espaço físico, fluxo de ar, ruído, potência da fonte). Em caso de falha, o sistema analisa o motivo e sugere uma alternativa coerente, como trocar o gabinete, ajustar o número de ventoinhas ou recomendar um cooler mais silencioso.

O usuário pode aceitar a proposta, rejeitar, ou sugerir alterações via chatbot, que serão automaticamente reavaliadas.**Atores:**

- Usuário (logado ou não)
- Sistema (módulo térmico com validador embutido)

#### **Pré-Condições:**

- A build do computador já foi definida (CPU, GPU, gabinete, etc.).
- Os dados ambientais foram fornecidos (manualmente ou via localização, CU2).

#### **Fluxo Principal:**

- O sistema calcula o TDP total da build.
- Analisa os parâmetros do ambiente físico:
  - Temperatura ambiente
  - Climatização ou ventilação natural
  - Nível de poeira
  - Preferência de ruído
- Classifica a demanda térmica como Leve, Moderada ou Alta.
- Gera uma recomendação de sistema de refrigeração, incluindo:
  - Tipo de cooler (air, torre ou AIO)
  - Quantidade e tamanho de ventoinhas (ex.: 3x120 mm)
  - Posições ideais de instalação
  - Soluções de baixo ruído ou com filtros, se necessário
- O sistema valida automaticamente a solução quanto a:
  - Espaço físico no gabinete
  - Potência da fonte (PSU)

- Fluxo de ar coerente
  - Ruído adequado ao ambiente
- Se a recomendação for válida:
  - O sistema exibe a solução ao usuário.
- Se for inválida:
  - O sistema identifica o motivo da incompatibilidade
  - Propõe um ajuste pontual (ex.: novo gabinete, cooler alternativo, troca de PSU)
  - Revalida automaticamente e exibe a nova sugestão
- **Fluxos Alternativos:****Dados ambientais ausentes:** Se o usuário não preencher nem autorizar coleta automática, o sistema usa um valor padrão (ex.: 30 °C) e exibe um aviso de precisão reduzida.
- **Sugestão do usuário:** O usuário pode sugerir um ajuste específico (ex.: “quero um AIO 240 mm”). O sistema revalida automaticamente a nova proposta e exibe o resultado.
- 

### **Fluxos de Exceção:**

Inviabilidade técnica sem solução possível:

Se nenhum ajuste for possível com os componentes atuais, o sistema informa: “A refrigeração adequada não é compatível com o gabinete/fonte atual. Considere revisar sua build.” **Pós-**

### **Condições:**

- O usuário recebe uma recomendação térmica personalizada e validada.
- A solução respeita todos os limites físicos, térmicos e contextuais.

### **Requisitos Funcionais Relacionados:**

- **RF-017 – Recomendação de Refrigeração**
- **RF-016 – Validação de Gabinete**
- **RF-015 – Coleta de Dados Ambientais**
- **RF-014 – Detecção de Localização**

### **Regras de Negócio Relacionadas:**

- RN05: A recomendação térmica deve considerar o TDP total + ambiente de uso.
- RNF-01: O sistema deve utilizar IA ou heurísticas para geração e ajustes.
- RNF-16: A lógica de recomendação e validação deve ser extensível.

#### **Relacionamentos com Outros Casos de Uso:**

- CU1 – Gerenciar Perfil: registra preferências e histórico térmico.
- CU2 – Coletar Requisitos Ambientais: fornece os dados do ambiente.
- CU3 – Gerar Configuração Automatizada: pode acionar CU5 automaticamente
- CU4 – Montagem Manual: exige análise térmica conforme peças escolhidas.

## **5. DIAGRAMA DE ATIVIDADE**

O diagrama de atividade descreve o fluxo de atividades em um processo, representando a lógica do negócio e o fluxo de controle de operações. Ele é especialmente útil para entender e modelar processos complexos dentro do sistema.

Componentes Principais:

1. Atividades: Representam as tarefas ou ações realizadas.
2. Fluxos de Controle: Indicadores de transição entre atividades.
3. Decisões: Pontos de ramificação no fluxo de atividades com base em condições.
4. Swimlanes: Divisões que mostram quais atores ou sistemas executam quais atividades.

A seguir temos os diagramas de atividade:

Figura 8 – Cadastro e login

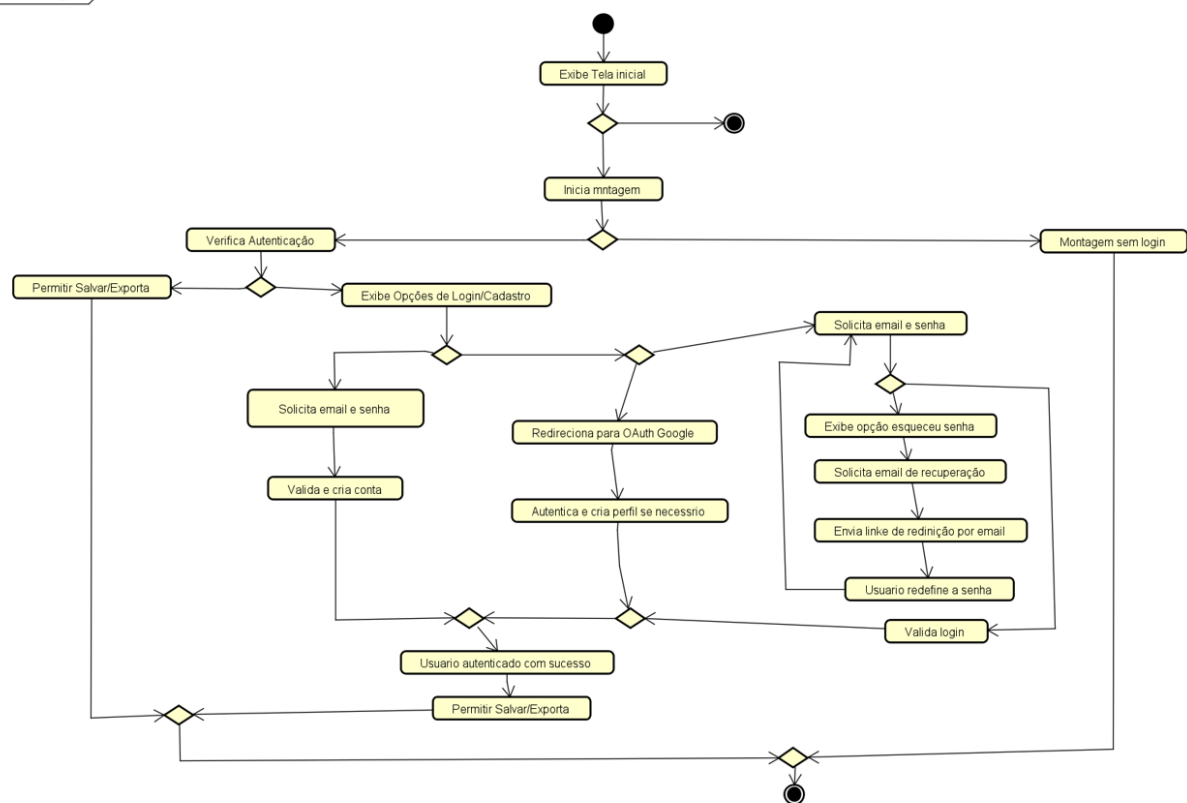


Figura 9 - Configuração automatizada

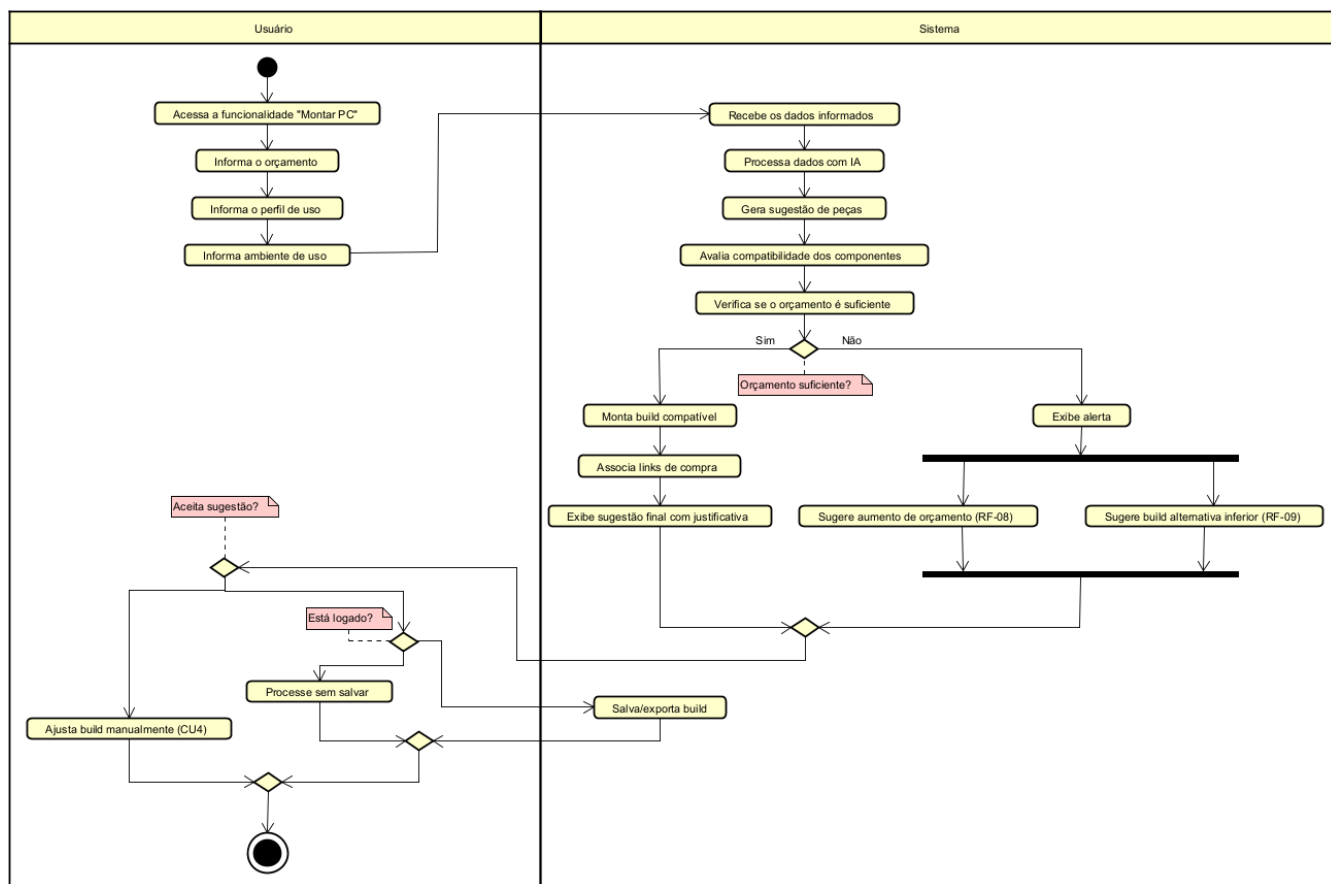


Figura 10 – Sistema de refrigeração

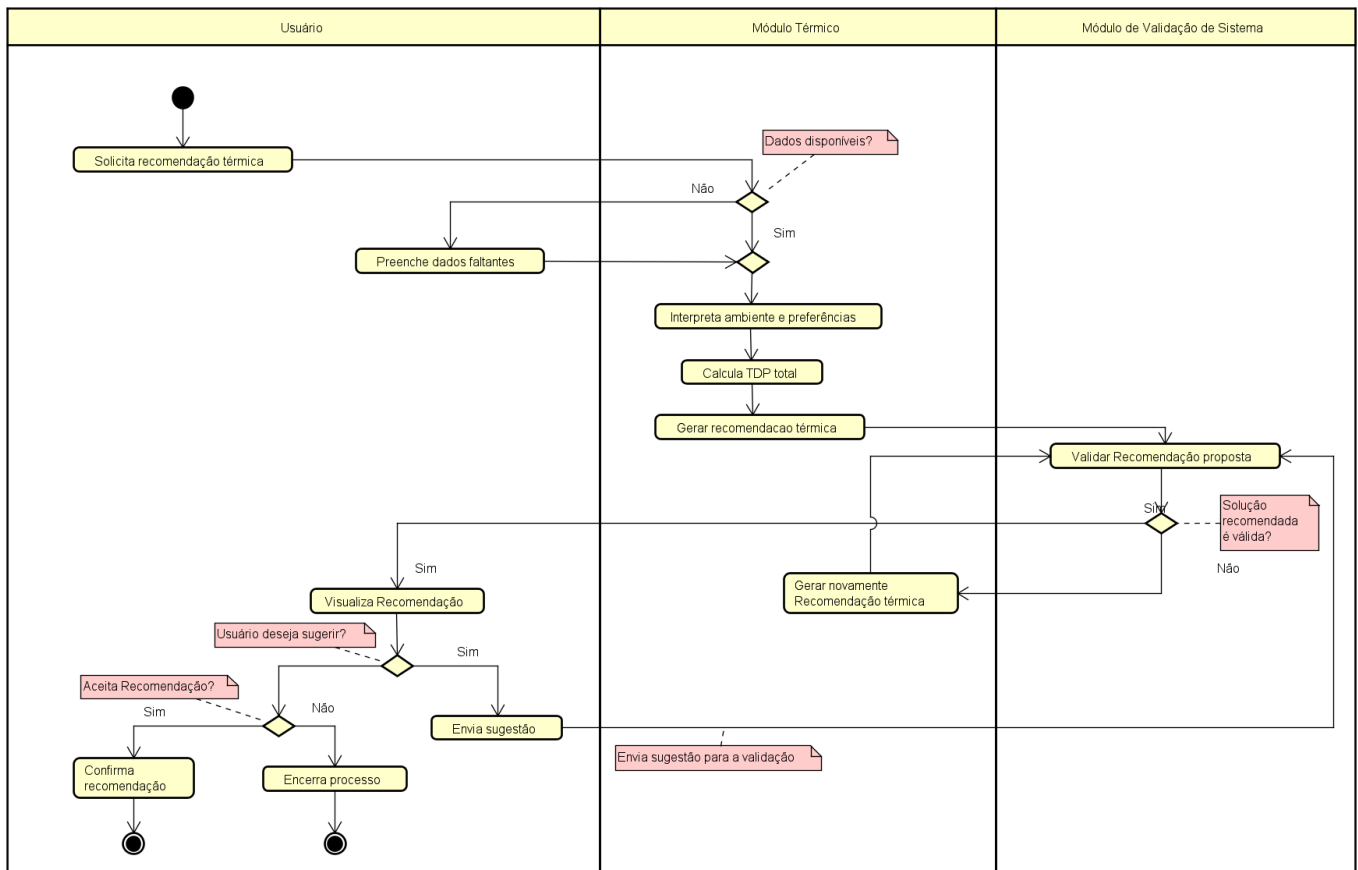


Figura 11 – Montagem manual



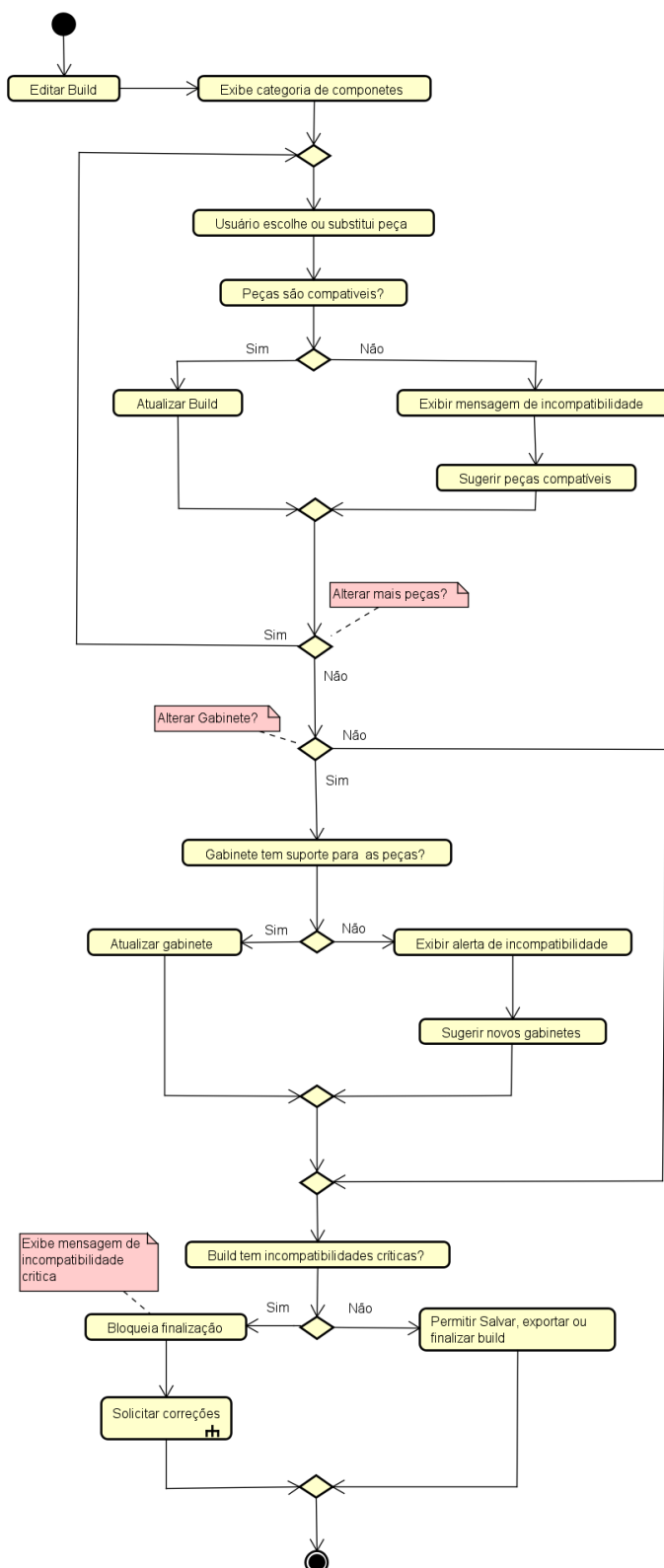
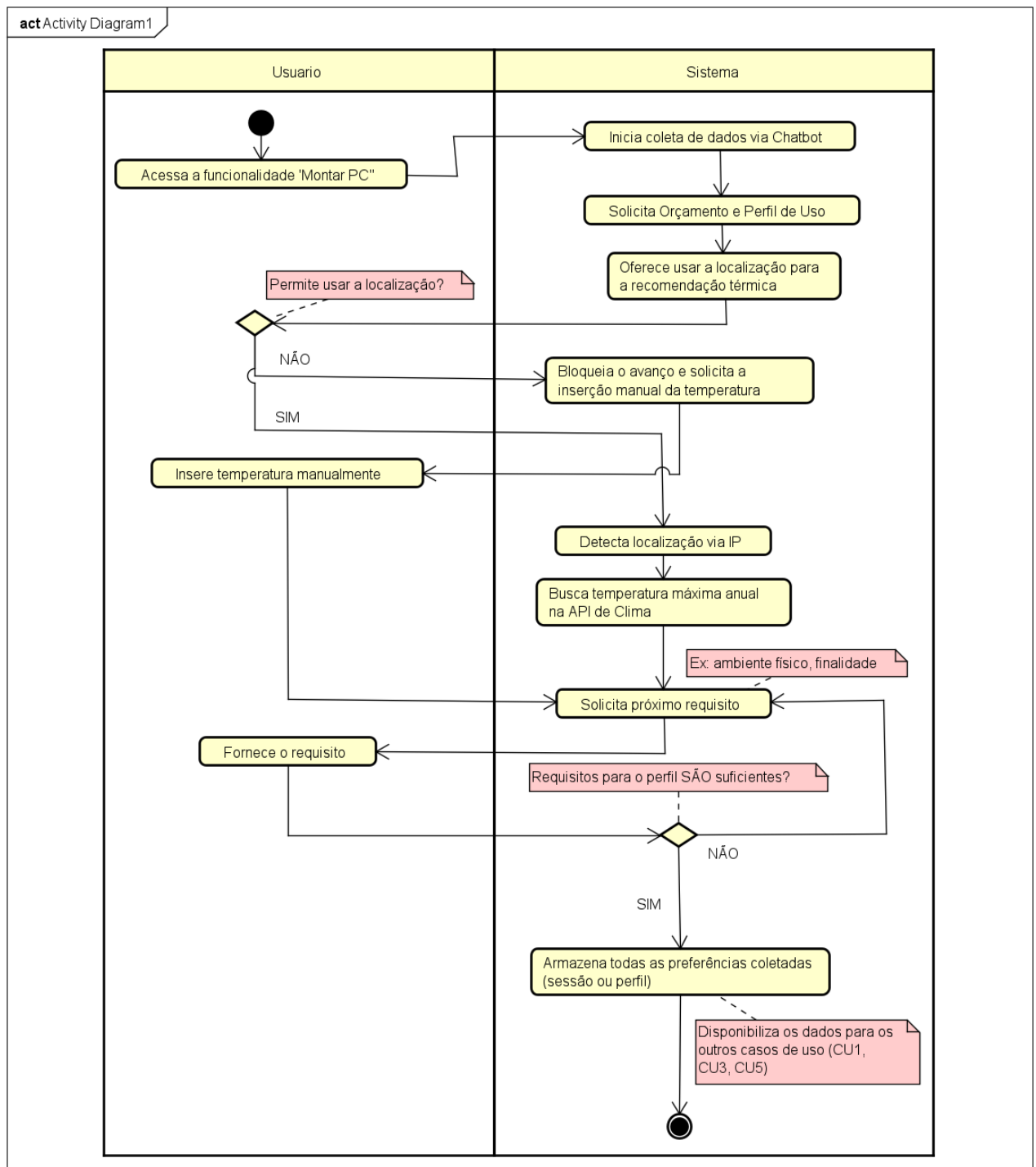


Figura 12 – Coleta de requisitos



### 5.1. Descrição dos diagramas de atividade

A descrição do diagrama de atividade é uma documentação que modela o fluxo de trabalho de um processo. Ela detalha as ações e atividades, a ordem em que ocorrem e as condições que determinam caminhos alternativos. O diagrama de atividade é fundamental para visualizar a sequência de passos de um caso de uso ou um processo de negócio, sendo

capaz de representar fluxos paralelos e concorrentes por meio de ramificações e pontos de sincronização.

### 5.1.1. Descrição Detalhada do Fluxo de Cadastro e Login

#### Início do Processo

O fluxo começa no nó inicial, e a primeira atividade executada é **Exibe Tela inicial**, que corresponde à exibição da interface principal do sistema.

#### Atividades Paralelas

A partir da tela inicial, o fluxo se divide em dois caminhos concorrentes através de um nó de ramificação (Fork):

- **Verificação de Autenticação:** A atividade **Verifica Autenticação** é executada para determinar se o usuário já possui uma sessão ativa no sistema.
- **Montagem da Interface:** A atividade **Inicia montagem** é executada, que é o processo de carregamento dos componentes e dados iniciais da interface.

#### Caminhos Condicionais de Autenticação

A partir de **Verifica Autenticação**, um nó de decisão direciona o fluxo:

- **Se a autenticação for positiva (usuário logado):** O fluxo vai para a atividade **Permite Salvar/Exporta**. Esta atividade representa a liberação de funcionalidades que dependem de um usuário autenticado. O fluxo então se conecta diretamente a um nó de sincronização final.
- **Se a autenticação for negativa (usuário não logado):** O fluxo se conecta a um nó de sincronização (Join), que aguarda a conclusão da atividade **Inicia montagem** para prosseguir.

#### Opções de Login/Cadastro

Após a sincronização, a atividade **Exibe Opções de Login/Cadastro** é apresentada. Um novo nó de decisão oferece três caminhos possíveis:

- **Acesso sem Login:** O fluxo se desvia para a atividade **Montagem sem login**, permitindo o acesso limitado ao sistema. Este caminho se une ao fluxo principal mais adiante.
- **Login/Cadastro com e-mail e senha:** A atividade **Solicita email e senha** é executada. Um nó de decisão interno oferece duas opções: login bem-sucedido, que leva à

atividade **Valida e cria conta**, ou recuperação de senha, que dispara um sub-processo que inclui o envio de um e-mail de recuperação e a redefinição da senha pelo usuário.

- **Login com OAuth Google:** O fluxo vai para a atividade **Redireciona para OAuth Google**, onde o usuário é enviado para a plataforma de autenticação externa. Após a autenticação, o sistema executa a atividade **Autentica e cria perfil se necessario**.

### Convergência e Pós-Authenticação

Os fluxos de login bem-sucedido (tradicional e Google) convergem em um nó de sincronização e, em seguida, na atividade **Usuário autenticado com sucesso**. Após a autenticação, a atividade **Permite Salvar/Exporta** é executada, concedendo as permissões necessárias.

### Sincronização Final e Conclusão

Todos os fluxos possíveis (usuário pré-autenticado, recém-autenticado e acesso sem login) convergem em um nó de sincronização final. O fluxo se encerra no nó final, indicando que o processo de autenticação ou acesso foi concluído com sucesso e o sistema está pronto para a interação do usuário.

## 5.1.2. Descrição Detalhada do Fluxo de Configuração Automatizada de PC

### Partições

O diagrama está dividido em duas partições que representam os papéis dos atores no processo:

- **Usuário:** Responsável por fornecer as informações e tomar decisões.
- **Sistema:** Responsável por receber, processar os dados e gerar as sugestões.

### Fluxo de Entrada de Dados (Usuário)

O processo se inicia no nó inicial, onde o usuário acessa a funcionalidade **"Montar PC"**. Sequencialmente, o usuário fornece ao sistema as seguintes informações:

- **Informa o orçamento** disponível.
- **Informa o perfil de uso** (por exemplo, jogos, trabalho, edição de vídeo).
- **Informa ambiente de uso** (por exemplo, tamanho do espaço, necessidades específicas).

### Processamento e Geração de Sugestão (Sistema)

Após receber os dados, o fluxo passa para a partição "Sistema", que executa as seguintes atividades:

- **Recebe os dados informados** do usuário.
- **Processa dados com IA**, utilizando inteligência artificial para analisar as informações.
- **Gera sugestão de peças** com base na análise.
- **Avalia compatibilidade dos componentes** para garantir que as peças sugeridas funcionem juntas.

### Verificação de Orçamento e Fluxos Alternativos

Um nó de decisão verifica a condição **Verifica se o orçamento é suficiente**:

- **Se o orçamento for suficiente ("Sim")**: O sistema executa as atividades **Monta build compatível**, **Associa links de compra** e **Exibe sugestão final com justificativa**.
- **Se o orçamento for insuficiente ("Não")**: O sistema exibe um **Exibe alerta** e oferece duas opções ao usuário: **Sugere aumento de orçamento (RF-08)** ou **Sugere build alternativa inferior (RF-09)**. Ambos os fluxos levam de volta a uma sugestão final para o usuário.

### Decisão do Usuário e Opções Pós-Sugestão

Após a exibição da sugestão, o fluxo retorna para a partição "Usuário" em um nó de decisão "Aceita sugestão?":

- **Se o usuário aceita a sugestão ("Sim")**: Um novo nó de decisão **Está logado?** é ativado. Se a resposta for "Sim", o sistema prossegue para **Salva/exporta build**. Se a resposta for "Não", o usuário pode optar por **Processo sem salvar** ou, em um caminho alternativo, **continuar sem salvar**.
- **Se o usuário não aceita a sugestão ("Não")**: O usuário opta por **Ajusta build manualmente (CU-04)**, que é um caso de uso para modificações manuais na configuração.

### Finalização do Fluxo

Os fluxos de **Ajusta build manualmente (CU-04)**, **Salva/exporta build** e **Processo sem salvar** convergem para o nó final do diagrama, indicando o encerramento do processo de configuração do PC.

## 5.1.3. Descrição Detalhada do Fluxo do Sistema de Recomendação Térmica

### Partições

O diagrama está dividido em três partições que representam os papéis dos atores e módulos do processo:

- **Usuário:** Responsável por iniciar o processo e interagir com as recomendações.
- **Módulo Térmico:** Responsável por processar os dados e gerar as recomendações.
- **Módulo de Validação de Sistema:** Responsável por verificar a validade das soluções propostas.

### Fluxo de Entrada e Processamento de Dados

O processo se inicia no nó inicial, onde o usuário executa a atividade **Solicita recomendação térmica**. O fluxo segue para o **Módulo Térmico**, que verifica se os **Dados disponíveis?**.

- **Se os dados não estiverem completos ("Não"):** O fluxo retorna à partição do usuário, que deve Preenche dados faltantes antes de o processo continuar.
- **Se os dados estiverem disponíveis ("Sim") ou forem preenchidos:** O Módulo Térmico executa as atividades Interpreta ambiente e preferências, Calcula TDP total e, em seguida, Gerar recomendação térmica.

### Validação da Recomendação e Loop de Geração

Após a recomendação ser gerada, o fluxo é transferido para o Módulo de Validação de Sistema, que executa a atividade “Validar Recomendação proposta”. Um nó de decisão verifica a condição “Solução recomendada é válida?”:

- **Se a solução for válida ("Sim"):** O fluxo retorna à partição do usuário.
- **Se a solução não for válida ("Não"):** O fluxo retorna ao **Módulo Térmico**, que executa a atividade **Gerar novamente Recomendação térmica**, criando um ciclo de validação e ajuste até que uma solução válida seja encontrada.

### Interação e Decisão do Usuário

Uma vez que uma solução válida é proposta, o fluxo retorna ao usuário, que executa a atividade **Visualiza Recomendação**. Dois nós de decisão sequenciais são apresentados:

- **Usuário deseja sugerir?:**
  - **Se a resposta for "Sim":** O usuário **Envia sugestão**, que é então enviada para a validação no **Módulo de Validação de Sistema**, reiniciando o ciclo de validação.
  - **Se a resposta for "Não":** O fluxo continua para a próxima decisão.
- **Aceita Recomendação?:**
  - **Se a resposta for "Sim":** O usuário executa a atividade **Confirma recomendação**.
  - **Se a resposta for "Não":** O usuário executa a atividade **Encerra processo**.

## Finalização do Fluxo

A finalização do processo ocorre após a decisão final do usuário. Tanto a atividade Confirma recomendação quanto a atividade Encerra processo conduzem o fluxo para o nó final do diagrama. Isso demonstra que o processo é encerrado de forma controlada, independentemente de o usuário ter aceitado ou não a solução proposta.

### 5.1.4. Descrição Detalhada de Montagem Manual de PC

O diagrama de atividade modela o processo interativo de um usuário ao montar um computador manualmente, com o sistema atuando como um guia de validação em tempo real para garantir a compatibilidade dos componentes escolhidos.

#### Início e Seleção de Componentes

O fluxo começa com a atividade "Editar Build". O sistema inicia o processo exibindo as categorias de componentes através da atividade "Exibe categoria de componentes". O usuário entra em um ciclo de seleção, onde a atividade "Usuário escolhe ou substitui peça" é o ponto central. A cada escolha, o sistema realiza uma validação de compatibilidade na condição "Peças são compatíveis?":

- **Caso de Sucesso:** Se as peças forem compatíveis ("Sim"), o sistema executa a atividade "Atualizar Build".
- **Caso de Falha:** Se houver incompatibilidade ("Não"), o sistema notifica o usuário através da atividade "Exibir mensagem de incompatibilidade" e auxilia a escolha com "Sugerir peças compatíveis".

#### Loop de Alterações e Verificação do Gabinete

Após cada atualização ou sugestão, o usuário é questionado na decisão "Alterar mais peças?". Se a resposta for "Sim", o fluxo retorna para o início do ciclo de seleção, permitindo múltiplas modificações. Se a resposta for "Não", o fluxo prossegue para uma etapa específica de verificação do gabinete, na decisão "Alterar Gabinete?".

- **Alteração do Gabinete:** Se o usuário optar por "Sim", o sistema verifica a condição "Gabinete tem suporte para as peças?".
  - **Compatibilidade:** Se o gabinete for compatível ("Sim"), a atividade "Atualizar gabinete" é executada.
  - **Incompatibilidade:** Se não for compatível ("Não"), o sistema exibe o "Exibir alerta de incompatibilidade" e executa a atividade "Sugerir novos gabinetes", fornecendo alternativas ao usuário.

- **Manter o Gabinete:** Se o usuário optar por "Não", o fluxo ignora a etapa de verificação do gabinete, seguindo diretamente para a validação final.

### Validação Final e Finalização do Build

Todos os caminhos convergem para a validação final do build. Um nó de decisão rigoroso verifica a condição "Build tem incompatibilidades críticas?".

- **Incompatibilidade Crítica:** Se houver incompatibilidade crítica ("Sim"), o sistema notifica o usuário com "Exibe mensagem de incompatibilidade crítica", "Bloqueia finalização" e "Solicitar correções". Esse fluxo direciona o usuário a um ponto anterior do processo para que as correções obrigatórias sejam realizadas.
- **Build Válido:** Se não houver incompatibilidade crítica ("Não"), a atividade "Permitir Salvar, exportar ou finalizar build" é executada, concedendo ao usuário a opção de concluir o processo.

O fluxo de "Solicitar correções" e o fluxo de finalização bem-sucedida convergem para o nó final do diagrama, garantindo que o processo seja sempre encerrado de maneira controlada, seja por sucesso ou por necessidade de correção.

### 5.1.5. Descrição Detalhada do Fluxo de Coleta de Requisitos

O diagrama de atividade modela o processo de coleta de requisitos e preferências do usuário para a funcionalidade de montar um PC, utilizando um fluxo interativo e automatizado. O processo é dividido entre as ações do usuário e as respostas do sistema.

#### Partições

O diagrama é dividido em duas partições que representam os papéis dos atores no processo:

- **Usuário:** Atua fornecendo informações e tomando decisões.
- **Sistema:** Responsável por iniciar a coleta, processar dados e fazer perguntas.

#### Início e Coleta Inicial de Dados

O fluxo começa com a atividade "Acessa a funcionalidade 'Montar PC'", iniciada pelo usuário. O controle é então transferido para o sistema, que executa as atividades "Inicia coleta de dados via Chatbot" e "Solicita Orçamento e Perfil de Uso", iniciando a interação.

#### Decisão de Localização e Temperatura

O sistema apresenta ao usuário a decisão "Permite usar a localização?".



- Permissão Concedida ("SIM"): O sistema "Detecta localização via IP" e, em seguida, "Busca temperatura máxima anual na API de Clima", obtendo os dados necessários de forma automatizada.
- Permissão Negada ("NÃO"): O sistema "Bloqueia o avanço e solicita a inserção manual da temperatura". O usuário, então, executa a atividade "Insere temperatura manualmente".

Após a coleta da temperatura (de forma automática ou manual), os fluxos se unem para prosseguir na coleta de outros requisitos.

### **Loop de Coleta de Requisitos**

A coleta continua com o sistema executando a atividade "Solicita próximo requisito", que pode incluir informações como "ambiente físico, finalidade". O usuário responde com a atividade "Fornece o requisito". A seguir, o sistema avalia a condição "Requisitos para o perfil SÃO suficientes?".

- Se os requisitos não forem suficientes ("NÃO"): O fluxo retorna para "Solicita próximo requisito", reiniciando o ciclo para coletar mais informações.
- Se os requisitos forem suficientes ("SIM"): O fluxo prossegue para a etapa final, saindo do loop.

### **Finalização e Disponibilização de Dados**

O processo se encerra com o sistema executando duas atividades essenciais:

- Armazena todas as preferências coletadas (sessão ou perfil): Esta atividade salva os dados de forma persistente ou temporária.
- Disponibiliza os dados para os outros casos de uso (CU1, CU3, CU5): Esta atividade torna os dados coletados acessíveis para outros módulos ou funcionalidades do sistema, como o de montagem de PC.

O fluxo termina no nó final, indicando o encerramento do processo de coleta de requisitos.

## **6. DIAGRAMA DE ESTADOS**

O Diagrama de Estados (ou Diagrama de Máquina de Estados) é um artefato da UML (Unified Modeling Language) utilizado para descrever os estados possíveis de um objeto durante seu ciclo de vida, bem como os eventos ou condições que provocam a transição entre esses estados. Ele é especialmente útil para modelar o comportamento de sistemas interativos

ou com regras de fluxo bem definidas, como assistentes de montagem, formulários com múltiplas etapas e fluxos guiados de usuário.

No projeto Montador de PC Online, esse diagrama é usado para representar as mudanças de estado do objeto "Build de PC" ao longo do uso da plataforma — desde sua criação até a exportação ou descarte.

A seguir, são listados os principais elementos encontrados em um diagrama de estados:

Tabela 5 - Principais Elementos do Diagrama

Elemento	Descrição
<b>Estado Inicial</b>	Indica o ponto de partida do objeto. Representado por um círculo sólido.
<b>Estado</b>	Representa uma situação ou condição em que o objeto pode estar. Mostrado como um retângulo com cantos arredondados.
<b>Transição</b>	Uma seta que liga dois estados, indicando uma mudança que ocorre devido a um evento.
<b>Evento</b>	Um gatilho que provoca a transição de um estado para outro (ex: "Orçamento definido").
<b>Condição/ Guarda</b>	Uma verificação lógica que deve ser verdadeira para que a transição ocorra (ex: [orcamento >= mínimo]).
<b>Ação</b>	Uma atividade executada durante a transição ou ao entrar/sair de um estado.
<b>Estado Final</b>	Indica o término do ciclo de vida do objeto. Representado por um círculo com um anel externo.

**Fonte:** Autoria propria,2025

### Aplicação no Projeto

No contexto do Montador de PC Online, o objeto modelado é a *Build de PC* do usuário. O diagrama descreve como ela é criada, modificada, validada e exportada, incluindo estados como:

- Nova Build Criada
- Requisitos Definidos
- Build Recomendado
- Build Editado Manualmente

- Compatibilidade Avaliada
- Build Salva
- Build Exportada
- Build Descartada

Esse modelo permite prever comportamentos esperados do sistema diante das ações do usuário e garantir consistência na lógica de funcionamento.

A seguir temos os digramas do sistema:

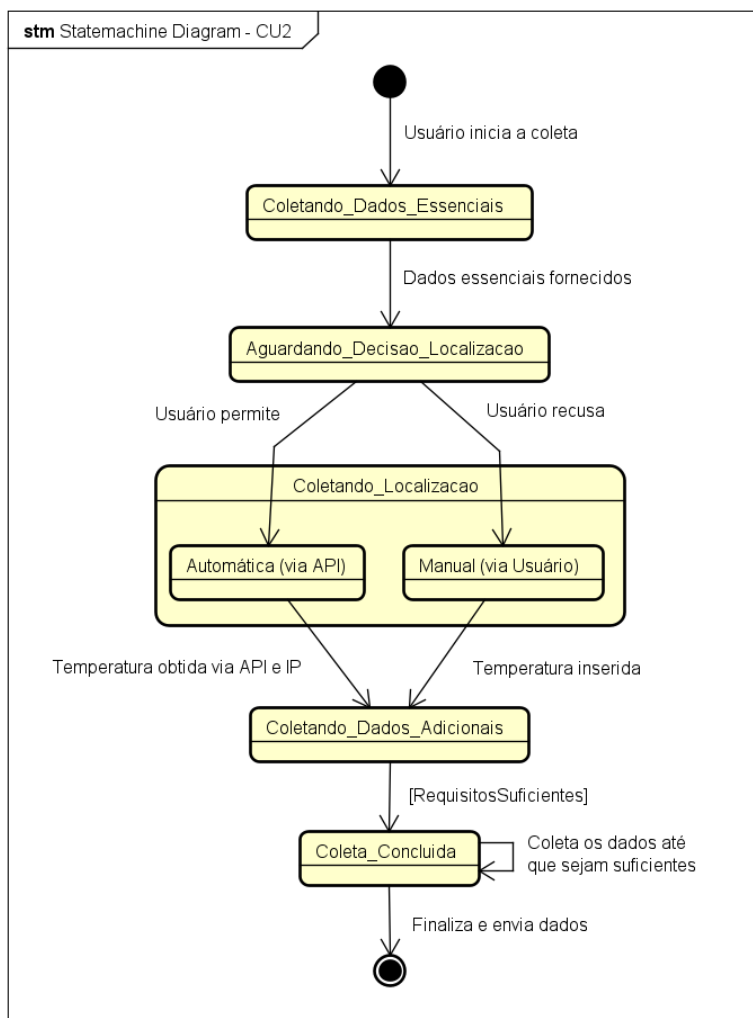
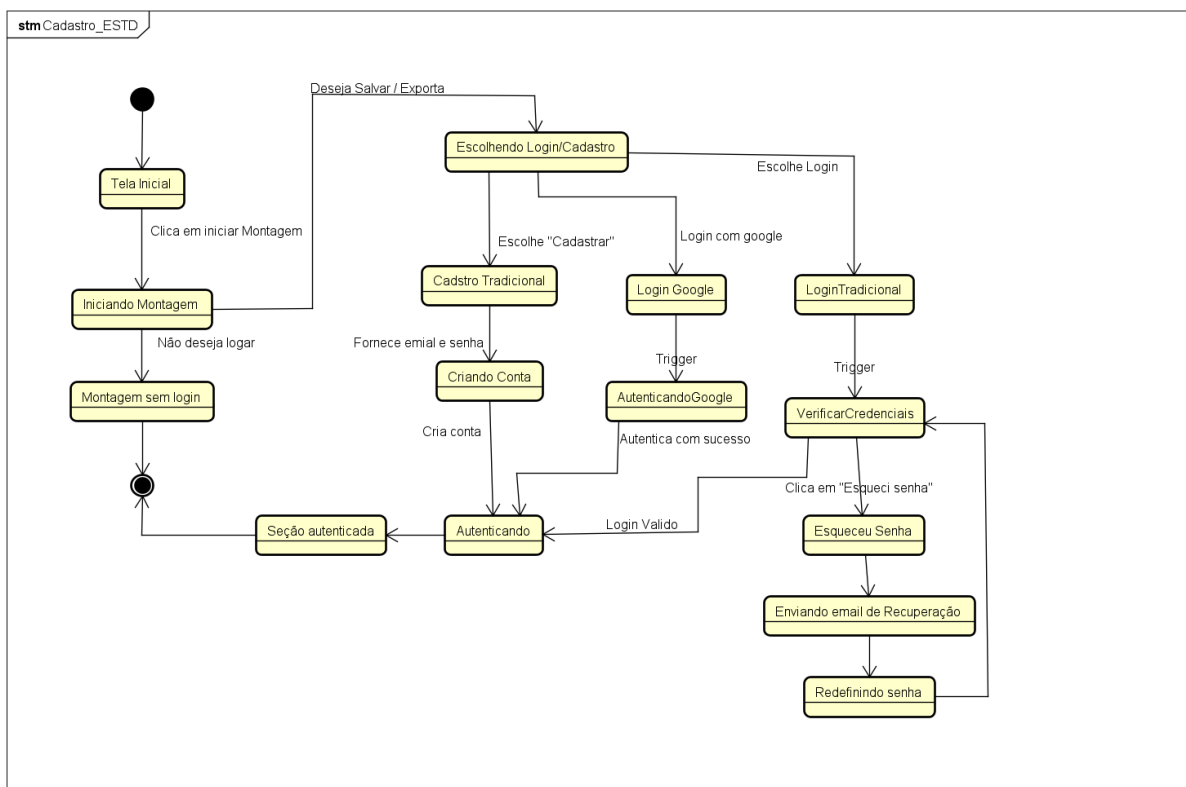


Figura 13 – Coleta de requisitos

Figura 14 – Cadastro e login



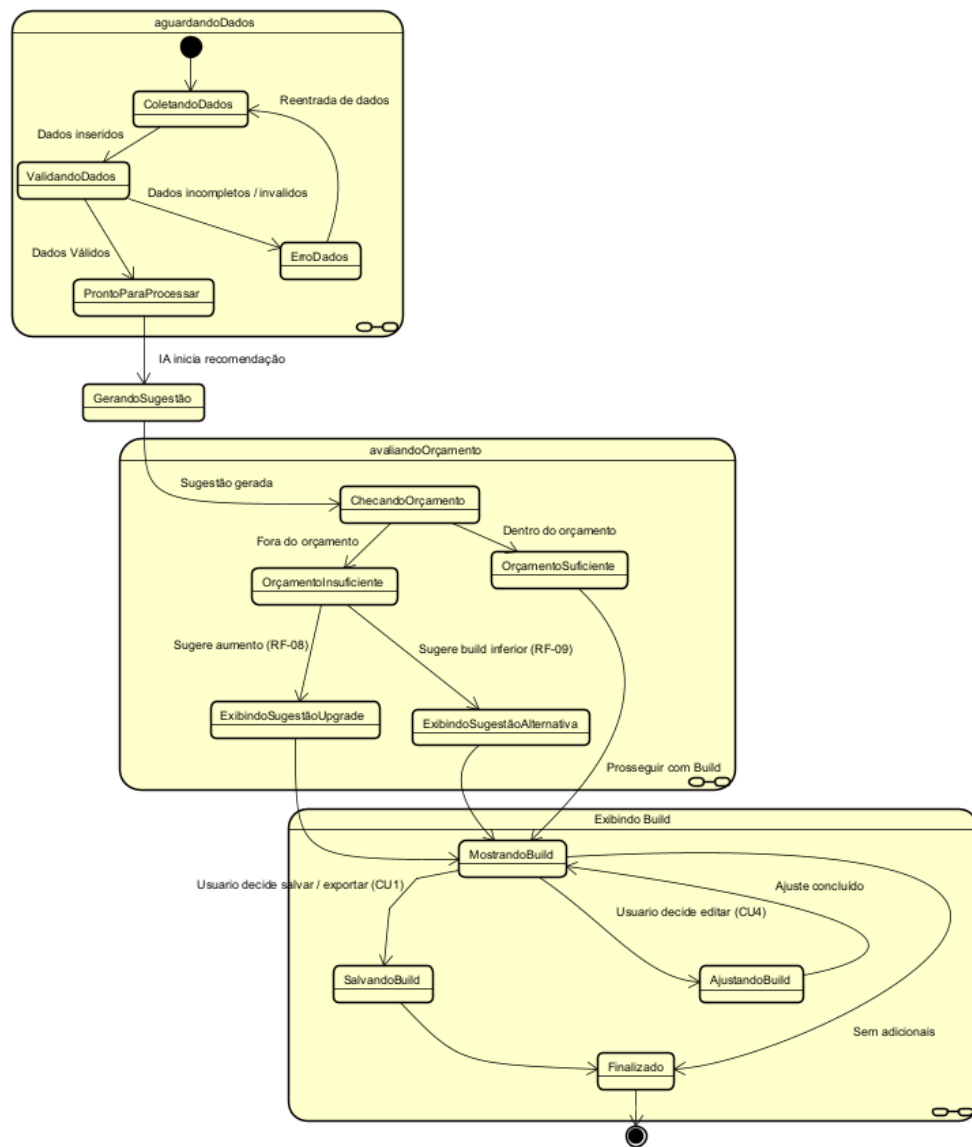


Figura 15: Configuração automatizada

Figura 16 – Sistema de refrigeração

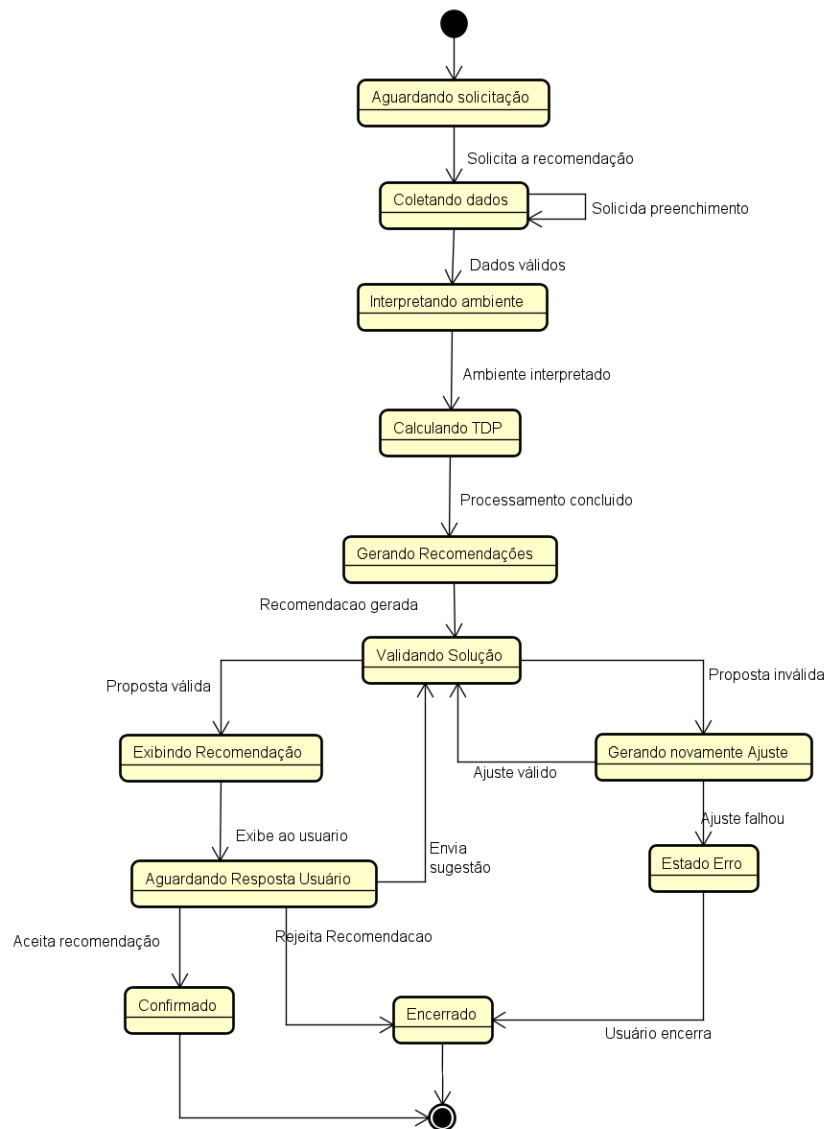
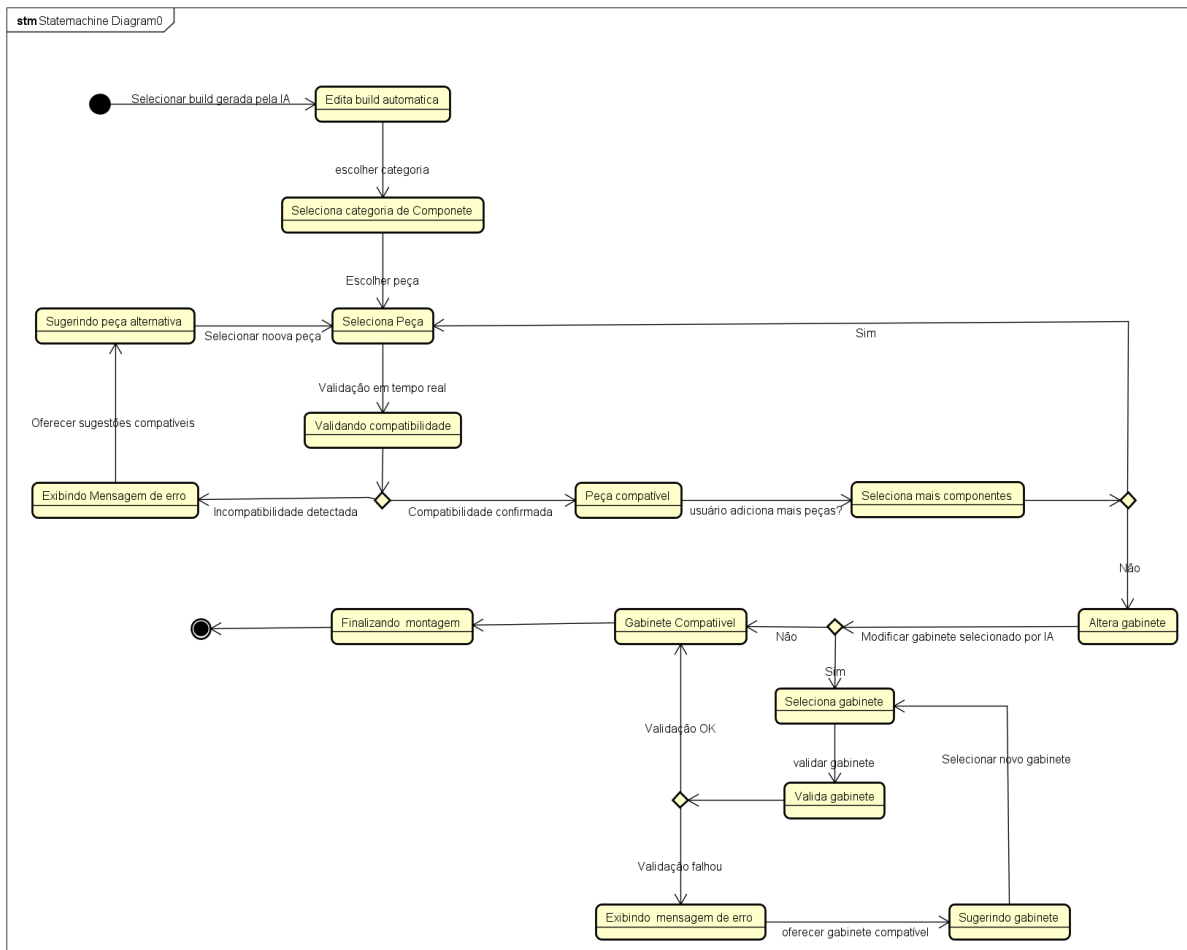


Figura 17 – Montagem Manual



## 6.1. Descrição Do Diagrama de Estados

A descrição do diagrama de estados é uma documentação que modela o comportamento dinâmico de um objeto ao longo de seu ciclo de vida. Ela detalha os diferentes estados pelos quais o objeto pode passar, as transições que ocorrem entre eles e os eventos que as disparam. O diagrama de estados é fundamental para visualizar a lógica por trás de sistemas reativos, sendo capaz de representar comportamentos complexos com a inclusão de sub-estados e condições de transição.

### 6.1.1. Descrição do Diagrama de Estados de Coleta de requisitos

O diagrama de estados stm Statemachine Diagram - CU2 modela o comportamento da funcionalidade de coleta de requisitos do usuário para a montagem de um PC. O diagrama descreve os estados sequenciais pelos quais a sessão do usuário transita até que todos os dados necessários sejam coletados.

## Estados de Coleta e Transições

O fluxo começa no nó inicial, com o evento "Usuário inicia a coleta", que leva a sessão ao estado "Coletando\_Dados\_Essenciais".

- **Coleta Essencial:** Nesse estado, o sistema espera que o usuário forneça os dados essenciais (como orçamento e perfil de uso). A transição para o próximo estado, "Aguardando\_Decisao\_Localizacao", ocorre quando o evento "Dados essenciais fornecidos" é disparado.
- **Decisão de Localização:** O estado "Aguardando\_Decisao\_Localizacao" representa o momento em que o sistema solicita permissão para usar a localização. A partir daqui, duas transições são possíveis:
  - Se o evento "Usuário permite" ocorrer, a sessão entra no sub-estado "Automática (via API)".
  - Se o evento "Usuário recusa" ocorrer, a sessão entra no sub-estado "Manual (via Usuário)".
- **Coleta de Localização:** O estado "Coletando\_Localizacao" é um estado composto que representa a coleta de temperatura. Os sub-estados "Automática" e "Manual" modelam as duas formas de obtenção dos dados. Ambas as transições para o próximo estado, "Coletando\_Dados\_Adicionais", são acionadas quando a temperatura é obtida: "Temperatura obtida via API e IP" ou "Temperatura inserida".
- **Coleta Adicional e Loop de Validação:** No estado "Coletando\_Dados\_Adicionais", o sistema solicita mais informações. Há uma transição de auto-retorno dentro do estado, que representa o loop de coleta de dados até que a condição de guarda [RequisitosSuficientes] seja atendida.
- **Coleta Concluída:** Quando a condição é atendida, a transição para o estado "Coleta\_Concluida" é ativada.

## Finalização

A partir do estado "Coleta\_Concluida", a transição final é disparada pelo evento "Finaliza e envia dados", levando ao nó final do diagrama. Isso representa o encerramento da coleta de requisitos e a disponibilização dos dados para o restante do sistema.



### 6.1.2. Descrição do Diagrama de Estados de Cadastro e Login Título e Propósito

O diagrama de estados de Cadastro e Login modela o comportamento dinâmico de uma sessão de usuário em um sistema, especificamente durante os processos de autenticação (login), cadastro e recuperação de senha. O diagrama descreve os estados pelos quais a sessão do usuário transita em resposta a eventos específicos.

#### Estrutura de Estados

O diagrama é composto por estados que podem ser categorizados da seguinte forma:

- **Estados de Início e Navegação:** "Tela Inicial", "Iniciando Montagem", "Montagem sem login".
- **Estados de Escolha:** "Escolhendo Login/Cadastro".
- **Estados de Cadastro e Autenticação:** "Cadastro Tradicional", "Criando Conta", "Login Tradicional", "Login Google", "Autenticando Google", "VerificarCredenciais", "Autenticando".
- **Estados de Recuperação de Senha:** "Esqueceu Senha", "Enviando email de Recuperação", "Redefinindo senha".
- **Estado Final de Sucesso:** "Seção autenticada".

#### Fluxo de Transições Detalhado

**a) Transições Iniciais** O processo começa no nó inicial, levando ao estado "Tela Inicial". A partir daí, duas transições são possíveis:

- Para o estado "Iniciando Montagem", disparada pelo evento "Clica em iniciar Montagem".
- Para o estado "Montagem sem login", disparada pelo evento "Não deseja logar", que leva ao nó final.

**b) Fluxo de Seleção e Opções de Autenticação** O estado "Iniciando Montagem" leva ao estado "Escolhendo Login/Cadastro". Este estado é central para as opções de autenticação e permite as seguintes transições:

- Para "LoginTradicional", disparada pelo evento "Escolhe Login".
- Para "Login Google", disparada pelo evento "Login com google".

- Para "Cadastro Tradicional", disparada pelo evento "Escolhe "Cadastrar"".

Há também uma transição direta de "Tela Inicial" para "Escolhendo Login/Cadastro", que ocorre sob a condição "Deseja Salvar / Exporta".

#### **c) Fluxo de Cadastro Tradicional**

- Do estado "Cadastro Tradicional", a transição para "Criando Conta" é disparada pelo evento "Fornece email e senha".
- A transição de "Criando Conta" para "Autenticando" é disparada pelo evento "Cria conta".

#### **d) Fluxo de Autenticação Tradicional e Recuperação de Senha**

- Do estado "LoginTradicional", uma transição para "VerificarCredenciais" é disparada.
- Do estado "VerificarCredenciais", duas transições são possíveis:
  - Para "Autenticando", se a condição "Login Válido" for atendida.
  - Para "Esqueceu Senha", se o evento "Clica em 'Esqueci senha'" ocorrer.
- O fluxo de recuperação de senha segue os estados "Enviando email de Recuperação" e "Redefinindo senha", com uma transição de retorno para "VerificarCredenciais" após a redefinição, para revalidar a nova senha.

#### **e) Fluxo de Autenticação via Google**

- Do estado "Login Google", uma transição para "Autenticando Google" é disparada.
- Do estado "Autenticando Google", a transição para o estado "Autenticando" é disparada pelo evento "Autentica com sucesso".

#### **f) Transição para o Estado Final**

- O estado "Autenticando" representa o processamento final da autenticação e transita para o estado "Seção autenticada".
- O estado "Seção autenticada" e o estado "Montagem sem login" representam os dois estados finais válidos e levam ao nó final do diagrama, indicando o encerramento do fluxo de autenticação/acesso.

## Conclusão

O diagrama modela de forma clara e granular o ciclo de vida da sessão do usuário durante a autenticação. A distinção entre os diferentes estados, as transições controladas por eventos e as condições de guarda criam um fluxo lógico e robusto para o tratamento das interações do usuário, desde a escolha inicial até o acesso finalizado. A inclusão do fluxo de recuperação de senha e do acesso sem login demonstra a abrangência da modelagem.

### 6.1.3. Descrição do Diagrama de Estados de Configuração Automatizada

O diagrama de estados modela o ciclo de vida de uma sessão de configuração de PC, desde a entrada de dados até a finalização do build. O diagrama utiliza estados compostos para agrupar as fases principais do processo: coleta de dados, avaliação do orçamento e interação com o build.

#### Estados e Transições Principais

O processo se inicia no estado composto "AguardandoDados", que gerencia a entrada e a validação das informações do usuário. Uma vez que os dados são validados, a sessão transita para o estado "GerandoSugestão" pelo evento "IA inicia recomendação".

A recomendação gerada leva a sessão ao estado composto "avaliandoOrçamento", onde o sistema interage com o usuário para garantir que o orçamento seja atendido. Em seguida, a transição para o estado composto "MostrandoBuild" ocorre com o evento "Prosseguir com Build". O fluxo termina no nó final, que é alcançado a partir do sub-estado "Finalizado".

#### Detalhamento do Estado Composto "AguardandoDados"

Este estado gerencia a coleta inicial de dados e é composto por três sub-estados:

- **"ColetandoDados"**: É o estado inicial para a entrada de dados. A transição para "ValidandoDados" é disparada pelo evento "Dados inseridos".
- **"ValidandoDados"**: O sistema verifica se os dados são completos e válidos. Se houver problemas, o evento "Dados incompletos / invalidos" retorna a sessão para "ColetandoDados". Se os dados forem válidos, a transição para "ProntoParaProcessar" ocorre com o evento "Dados Válidos".
- **"ProntoParaProcessar"**: Este sub-estado é o ponto de saída do estado composto.

#### Detalhamento do Estado Composto "avaliandoOrçamento"

Este estado lida com a avaliação do orçamento e a interação com o usuário sobre as sugestões:

- O fluxo começa no sub-estado "ChecandoOrçamento", acionado pelo evento "Sugestão gerada".
- A partir de "ChecandoOrçamento", duas transições são possíveis, com base na condição do orçamento ser "Dentro do orçamento" ou "Fora do orçamento".
  - **"OrçamentoSuficiente"**: A transição para este estado ocorre se o orçamento for suficiente.
  - **"OrçamentoInsuficiente"**: Se o orçamento for insuficiente, o sistema oferece alternativas nos estados "ExibindoSugestaoUpgrade" (evento "Sugere aumento (RF-08)") ou "ExibindoSugestaoAlternativa" (evento "Sugere build inferior (RF-09)"). Esses fluxos podem retornar a "ChecandoOrçamento", criando um ciclo de sugestão e avaliação.

### Detalhamento do Estado Composto "MostrandoBuild"

Neste estado, o usuário interage com a configuração proposta:

- O fluxo começa em "MostrandoBuild", onde o build é exibido.
- A partir de "MostrandoBuild", o usuário pode optar por:
  - Salvar o build, transicionando para "SalvandoBuild" com o evento "Usuario decide salvar / exportar (CU1)".
  - Editar o build, transicionando para "AjustandoBuild" com o evento "Usuario decide editar (CU4)".
- O sub-estado "AjustandoBuild" possui um loop de auto-retorno até que o "Ajuste concluído".
- Os estados "SalvandoBuild" e "AjustandoBuild" convergem para o estado "Finalizado", que representa o encerramento da interação com o build.

A transição final para o nó final do diagrama ocorre a partir do sub-estado "Finalizado".

#### 6.1.4. Descrição do Diagrama de Estados de Sistema de Refrigeração

O diagrama de estados modela o comportamento de um sistema de recomendação de soluções térmicas, desde a solicitação inicial do usuário até a finalização do processo. Ele detalha as transições entre os estados, incluindo ciclos de validação e feedback.

## Estados e Transições Principais

O processo se inicia no nó inicial, levando ao estado "Aguardando solicitacao". A transição para o estado "Coletando dados" é disparada pelo evento "Solicita a recomendacao". A partir daí, o fluxo segue uma sequência de processamento e validação:

- **Coleta e Processamento:** Após os dados serem validados, o sistema passa pelos estados "Interpretando ambiente", "Calculando TDP" e "Gerando Recomendacoes", cada um representando uma etapa de análise e processamento da solicitação do usuário.
- **Validação:** A transição para o estado "Validando Solucao" é disparada pelo evento "Recomendacao gerada". Este estado é um ponto de controle crucial do fluxo.

## Fluxos Condicionais e Ciclos de Validação

O estado "Validando Solucao" determina o próximo passo com base na validade da proposta:

- **Proposta Válida:** Se a proposta for válida, o evento "Proposta valida" leva a sessão ao estado "Exibindo Recomendacao".
- **Proposta Inválida:** Se a proposta for inválida, o evento "Proposta invalida" leva a sessão ao estado "Gerando novamente Ajuste".
  - O estado "Gerando novamente Ajuste" tem um ciclo de feedback: se o ajuste for válido, a transição "Ajuste valido" retorna a sessão para "Validando Solucao". Se o ajuste falhar, o evento "Ajuste falhou" leva ao estado terminal de erro, "Estado Erro".

## Interação com o Usuário e Finalização

Após a exibição de uma recomendação válida, o sistema transita para o estado "Aguardando Resposta Usuario", onde espera a decisão do usuário:

- **Aceitação:** Se o usuário "Aceita recomendacao", o fluxo segue para o estado "Confirmado".
- **Rejeição:** Se o usuário "Rejeita Recomendacao", o fluxo segue para o estado "Encerrado".

O estado de erro, "Estado Erro", também leva ao estado "Encerrado" através do evento "Usuario encerra", garantindo uma finalização controlada mesmo em caso de falha. Os estados "Confirmado" e "Encerrado" representam os dois pontos finais do fluxo, cada um levando ao nó final do diagrama.

### 6.1.5. Descrição do Diagrama de Estados de Montagem Manual

O diagrama de estados modela o ciclo de vida de uma sessão de montagem manual de PC, onde o usuário é guiado por um sistema de validação em tempo real. O processo é focado na seleção e confirmação da compatibilidade dos componentes, incluindo a escolha do gabinete, antes da finalização do build.

#### Fluxo de Seleção de Componentes

O fluxo se inicia após o evento "Selecionar build gerada pela IA", levando ao estado "Edita build automatica". O usuário então segue uma sequência de seleção de componentes:

- **Seleção de Categoria:** O estado inicial é "Seleciona categoria de Componente", que representa a escolha do tipo de peça a ser alterada.
- **Seleção da Peça:** Em seguida, a sessão transita para "Seleciona Peça". O evento "Validação em tempo real" leva ao estado de validação.

#### Validação e Sugestão de Alternativas

O estado "Validando compatibilidade" é um ponto de controle que gera duas transições:

- **Incompatibilidade:** Se ocorrer o evento "incompatibilidade detectada", a sessão transita para "Exibindo Mensagem de erro", que por sua vez leva ao estado "Sugerindo peça alternativa". Este estado oferece opções ao usuário, que pode "Selecionar nova peça", retornando ao estado de validação, criando um ciclo de ajuste e verificação.
- **Compatibilidade:** Se a condição "Compatibilidade confirmada" for atendida, a sessão entra no estado "Peça compatível".

#### Loop de Adição de Componentes

O estado "Peça compatível" é o ponto de decisão para continuar a montagem manual. Com base no evento "usuario adiciona mais peças?", a sessão pode:

- **Adicionar mais peças:** Se a resposta for "Sim", a sessão transita para o estado "Seleciona mais componentes", que retorna ao início do ciclo de seleção de componentes.
- **Prosseguir:** Se a resposta for "Não", a sessão avança para o fluxo de verificação do gabinete.

### Verificação do Gabinete

O diagrama apresenta um fluxo de validação específico para o gabinete, que pode ser acessado de diferentes maneiras, como a transição do estado "Seleciona mais componentes" ou a partir do evento "Altera gabinete". O estado "Valida gabinete" é central, verificando a compatibilidade do gabinete com as demais peças.

- **Compatível:** Se a condição for "Validacao OK", a sessão entra no estado "Gabinete Compatível".
- **Incompatível:** Se a condição "Validação falhou" ocorrer, a sessão transita para "Exibindo mensagem de erro" e, em seguida, para "Sugerindo gabinete", onde um ciclo de ajuste se inicia.

### Finalização da Montagem

Uma vez que o gabinete é considerado compatível, o estado "Gabinete Compatível" leva a sessão ao estado "Finalizando montagem". Este estado representa a conclusão do processo de edição manual, e a transição a partir dele leva ao nó final do diagrama, encerrando a atividade.

## 7. DIAGRAMA DE SEQUÊNCIA

O Diagrama de Sequência é um dos diagramas comportamentais da UML utilizado para descrever a interação entre os objetos ou componentes de um sistema ao longo do tempo. Ele mostra a ordem em que as mensagens são trocadas entre os participantes para realizar uma funcionalidade específica.

Esse tipo de diagrama é extremamente útil para representar casos de uso complexos, identificar dependências entre módulos e detalhar o fluxo de execução de processos críticos.

No projeto Montador de PC Online, o diagrama de sequência ajuda a visualizar como o sistema se comporta, por exemplo, durante o processo de recomendação de peças, verificação de compatibilidade ou salvamento de builds.

Tabela 6 - Principais Elementos do Diagrama

Elemento	Descrição
<b>Ator</b>	Representa um usuário ou sistema externo que interage com o sistema (ex: "Usuário").
<b>Participante</b>	Um componente interno ou entidade envolvida na execução do processo (ex: "Sistema", "IA", "Banco de Dados").
<b>Linha de Vida</b>	Linha vertical que indica a existência de um participante durante a execução da funcionalidade.
<b>Mensagem</b>	Seta horizontal entre participantes que representa uma chamada de método ou troca de informação.
<b>Mensagem Síncrona</b>	Indica uma chamada que espera resposta (linha cheia com ponta cheia).
<b>Mensagem Assíncrona</b>	Indica uma chamada que não espera resposta (linha cheia com ponta aberta).
<b>Resposta</b>	Representada por uma seta pontilhada retornando ao chamador, indicando o retorno de uma função.
<b>Bloco de Ativação</b>	Retângulo fino sobre a linha de vida, mostrando quando o participante está "ativo".
<b>Notas / Comentários</b>	Informações adicionais sobre interações específicas.

**Fonte:** Autoria própria, 2025

#### Aplicação no Projeto

No contexto do Montador de PC Online, o diagrama de sequência pode ser aplicado para modelar diferentes cenários de uso, como:

- O fluxo de coleta de requisitos e geração de recomendação automatizada.



- O processo de verificação de compatibilidade entre peças.
- A interação entre o sistema, o banco de dados e a IA durante a montagem.
- A exportação de uma build para PDF ou XLSX.

Esses diagramas permitem ao time visualizar o encadeamento correto de chamadas e identificar gargalos, dependências ou comportamentos inesperados.

A seguir temos os diagramas de sequência

Figura 18 – Cadastro e Login

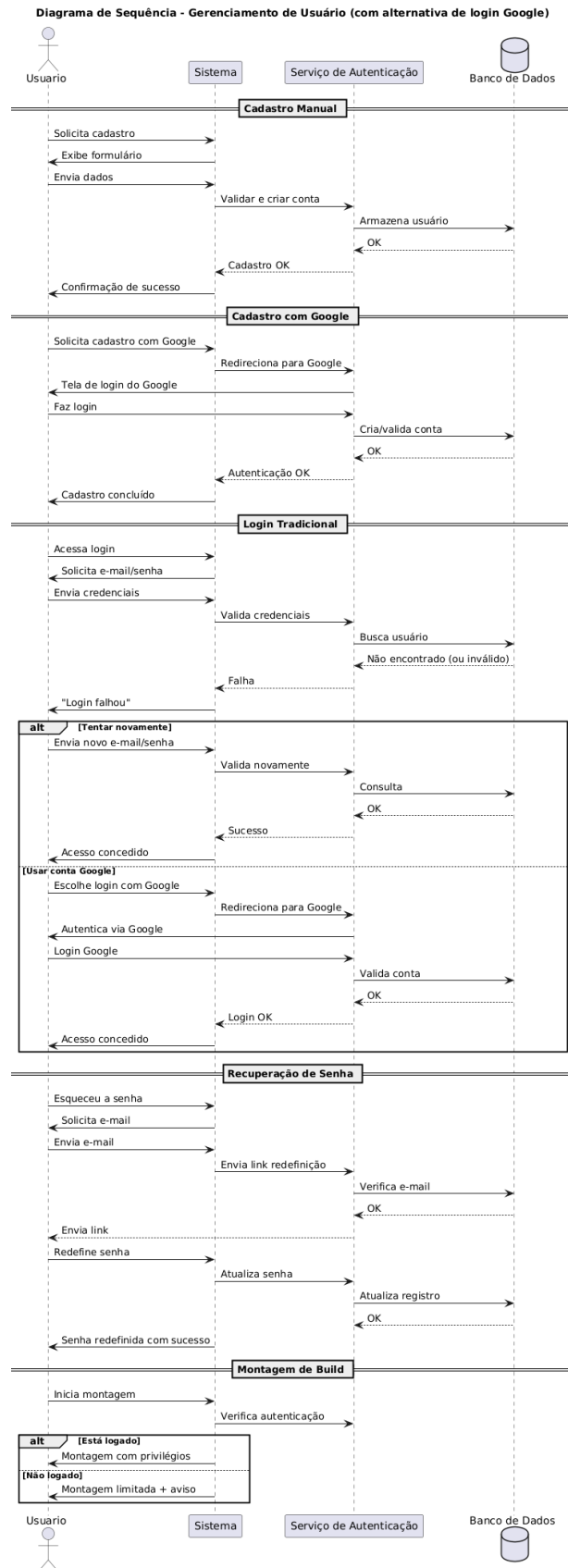


Figura 19 – Coleta de requisitos

Diagrama de Sequência: Coletar Requisitos e Preferências

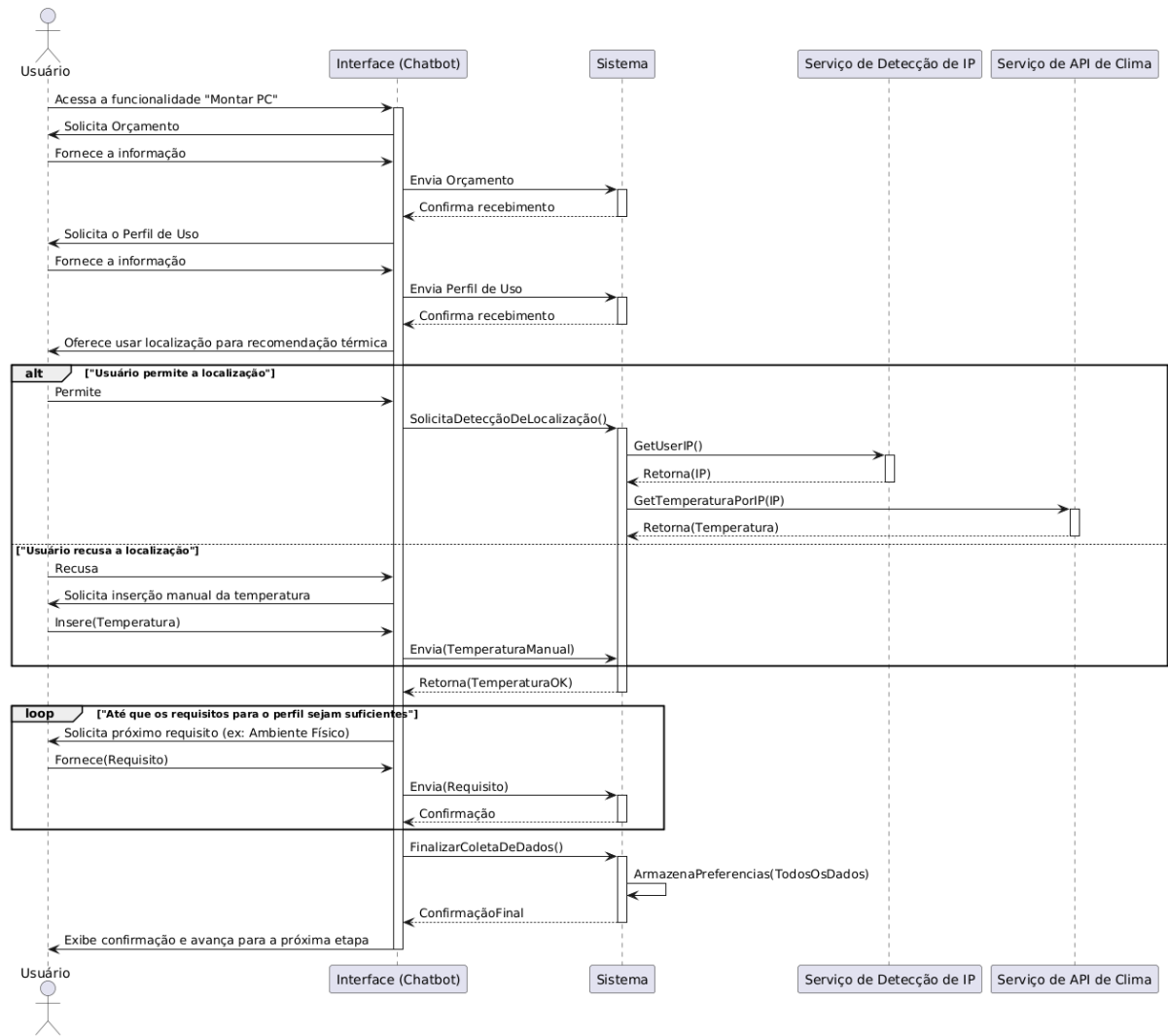
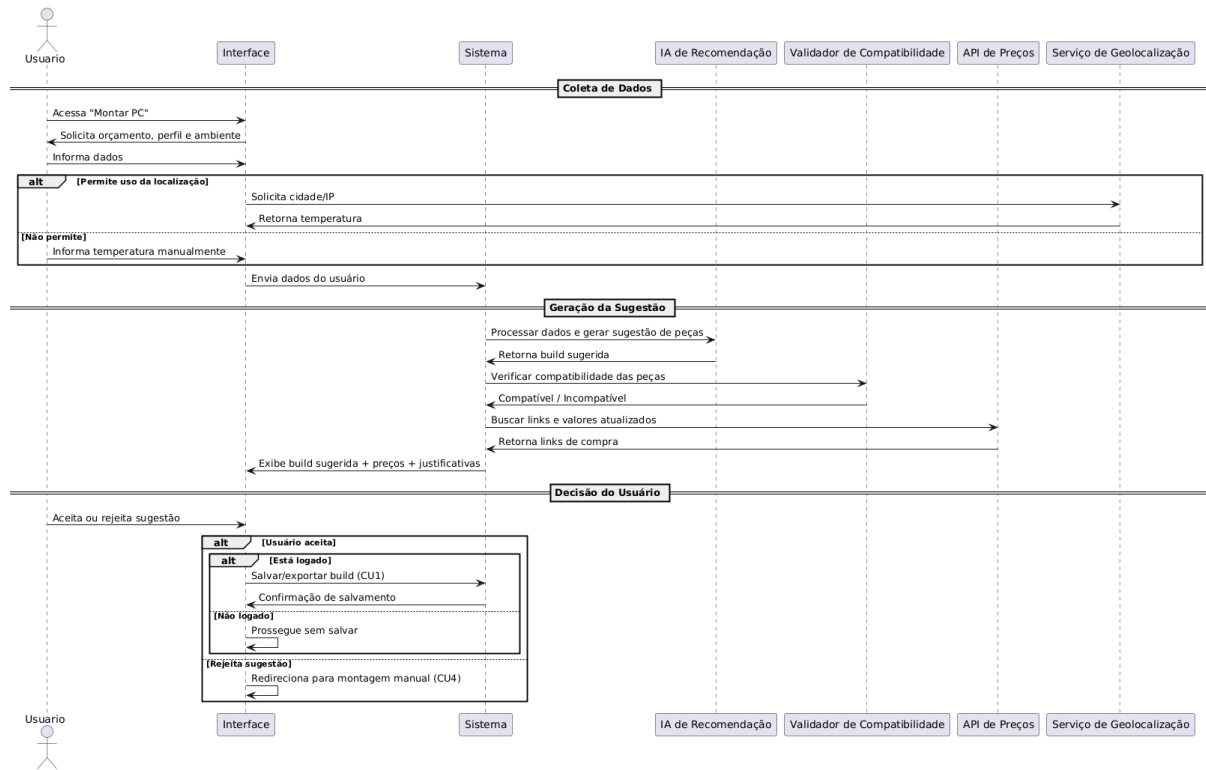


Figura 20 - Configuração automatizada



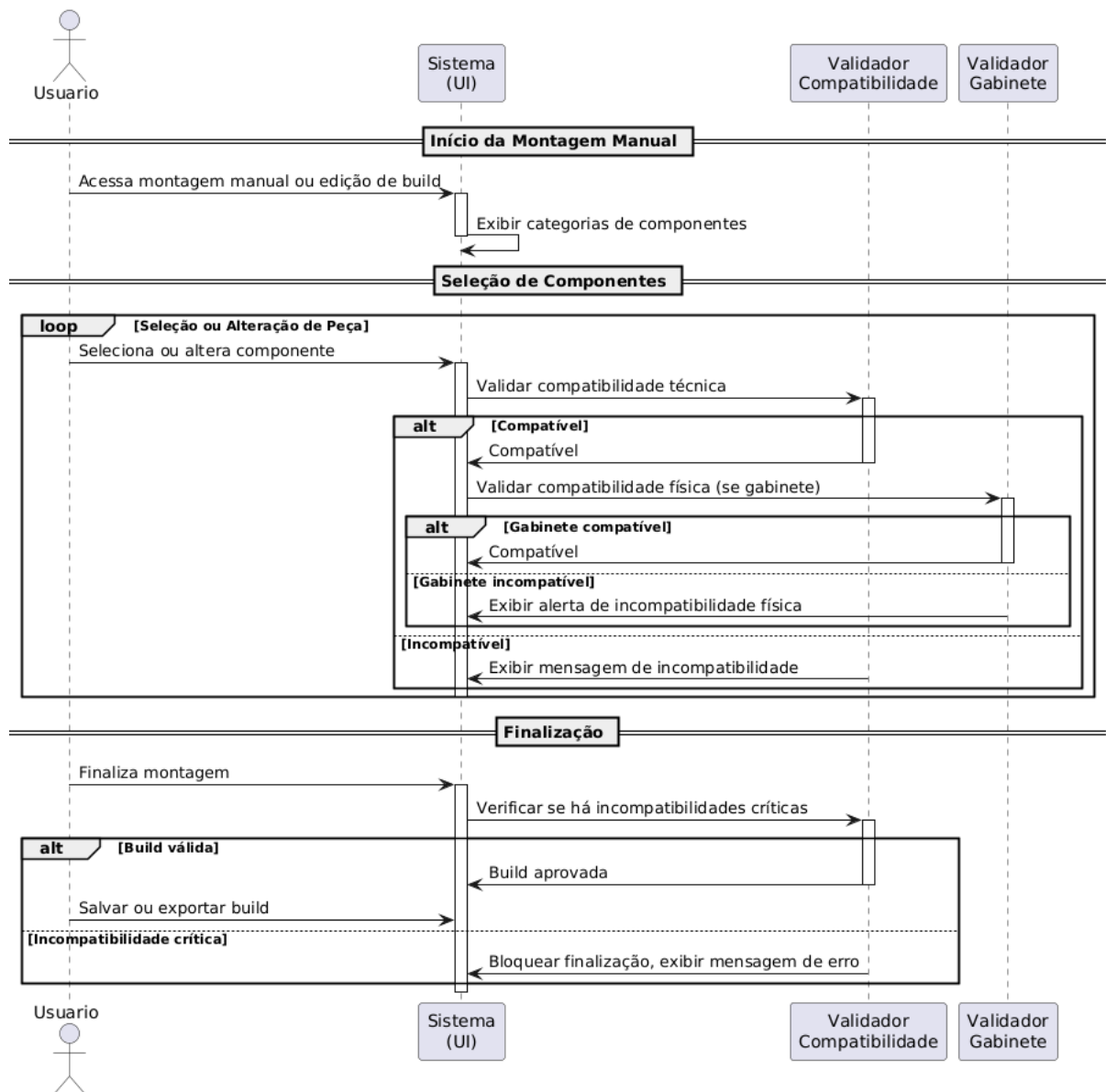
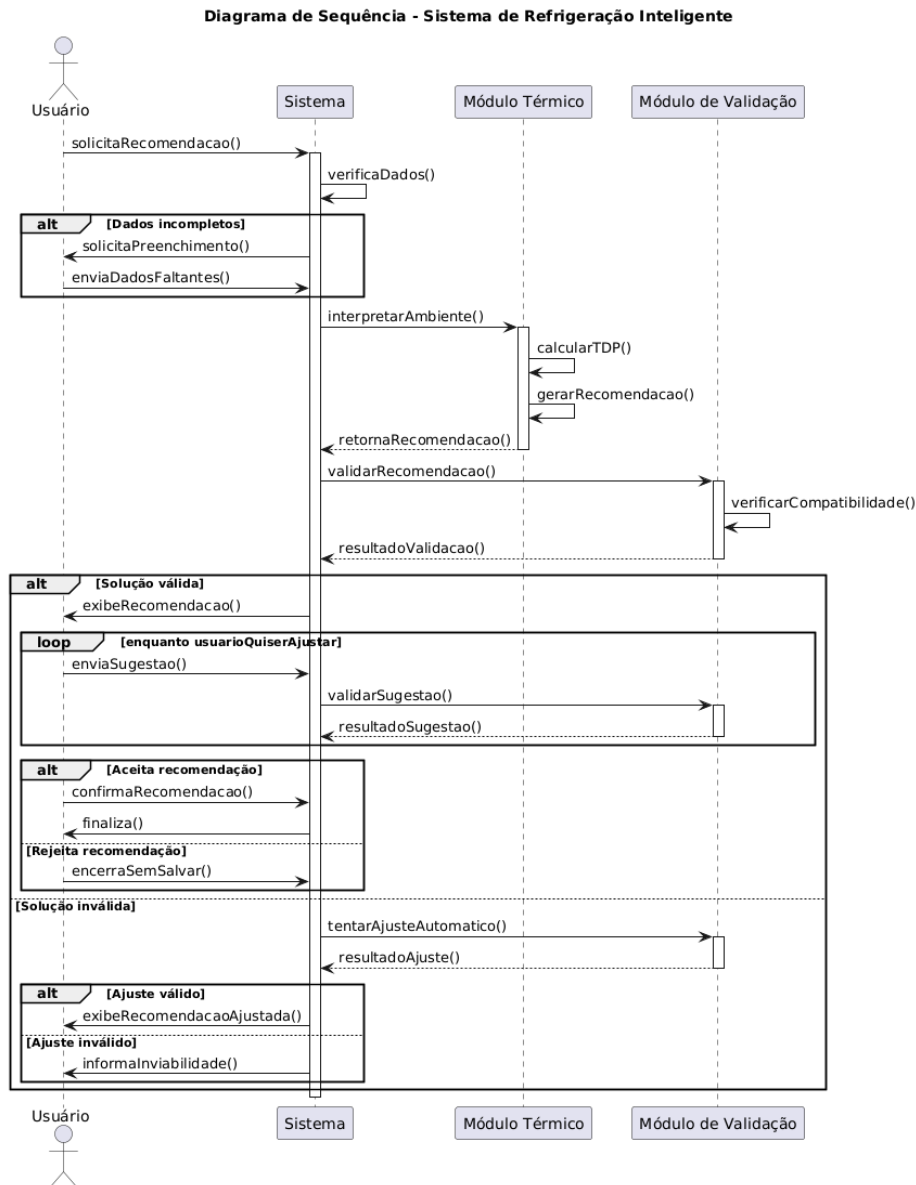


Figura 21 – Montagem manual

Figura 22 – Sistema de Refrigeração



## 7.1 Descrição dos Diagramas de Sequência

A descrição do diagrama de sequência é uma documentação que modela o comportamento dinâmico de um sistema, focando na interação entre objetos. Ela detalha a sequência de mensagens trocadas, a ordem temporal em que ocorrem e as linhas de vida que representam a existência de cada objeto ao longo do tempo. O diagrama de sequência é fundamental para visualizar a colaboração entre os objetos para a realização de um caso de uso, sendo capaz de representar a ordem exata das chamadas e das respostas, além de fluxos alternativos e laços de interação.

### 7.1.1. Descrição Detalhada de Cadastro e Login:

O diagrama de sequência modela as interações dinâmicas entre os objetos "Usuário", "Sistema", "Serviço de Autenticação" e "Banco de Dados" para todos os processos de gerenciamento de conta, organizados em fluxos distintos.

#### Fluxo de Cadastro Manual

- **1. Solicitação:** O "Usuário" envia a mensagem "Solicita cadastro" ao "Sistema".
- **2. Exibição:** O "Sistema" exibe um "Exibe formulario" para o "Usuário".
- **3. Criação de Conta:** O "Usuário" envia os "Envia dados" ao "Sistema".
- **4. Validação:** O "Sistema" envia a mensagem "Validar e criar conta" para o "Serviço de Autenticação".
- **5. Persistência:** O "Serviço de Autenticação" envia a mensagem "Armazena usuario" ao "Banco de Dados".
- **6. Confirmação:** O "Banco de Dados" retorna "OK" para o "Serviço de Autenticação", que responde com "Cadastro OK" ao "Sistema".
- **7. Notificação:** O "Sistema" envia uma "Confirmação de sucesso" para o "Usuário".

#### Fluxo de Cadastro com Google

- **1. Solicitação:** O "Usuário" envia a mensagem "Solicita cadastro com Google" para o "Sistema".
- **2. Redirecionamento:** O "Sistema" responde com "Redireciona para Google".
- **3. Autenticação:** Após o "Usuário" realizar o login, o "Sistema" envia "Cria/valida conta" ao "Serviço de Autenticação".
- **4. Confirmação:** O "Serviço de Autenticação" retorna "Autenticação OK" ao "Sistema".
- **5. Notificação:** O "Sistema" envia a mensagem "Cadastro concluido" para o "Usuário".

#### Fluxo de Login Tradicional (Bloco alt)

- **1. Solicitação:** O "Usuário" envia a mensagem "Acessa login" ao "Sistema", que responde com "Solicita e-mail/senha".

- **2. Validação:** O "Usuário" envia "Envia credenciais" e o "Sistema" envia "Valida credenciais" ao "Serviço de Autenticação".
- **Cenário de Sucesso:**
  - **3. Confirmação:** O "Serviço de Autenticação" responde "Sucesso".
  - **4. Conclusão:** O "Sistema" informa "Acesso concedido" ao "Usuário".
- **Cenário de Falha:**
  - **3. Erro:** O "Serviço de Autenticação" responde "Falha" ao "Sistema".
  - **4. Notificação:** O "Sistema" informa "Login falhou" ao "Usuário".

### Fluxo de Recuperação de Senha

- **1. Solicitação:** O "Usuário" envia "Esqueceu a senha", e o "Sistema" solicita "Solicita e-mail".
- **2. Verificação:** O "Usuário" envia o e-mail, e o "Sistema" aciona o "Serviço de Autenticação" para "Verifica e-mail" no "Banco de Dados".
- **3. Envio:** O "Serviço de Autenticação" envia um "link redefinição" ao "Sistema", que o envia ao "Usuário".
- **4. Redefinição:** O "Usuário" acessa o "Envia link" e "Redefine senha", e o "Sistema" envia "Atualiza senha" ao "Serviço de Autenticação".
- **5. Confirmação:** O serviço atualiza o registro no "Banco de Dados" e retorna "OK" ao "Sistema".
- **6. Notificação:** O "Sistema" informa "Senha redefinida com sucesso" ao "Usuário".

### Fluxo de Montagem de Build (Bloco alt)

- **1. Solicitação:** O "Usuário" envia "Inicia montagem" ao "Sistema".
- **2. Verificação:** O "Sistema" verifica a autenticação com o "Serviço de Autenticação".
- **Cenário "Está logado":**
  - **3. Funcionalidade:** O "Sistema" responde com "Montagem com privilégios".
- **Cenário "Não logado":**
  - **3. Funcionalidade:** O "Sistema" responde com "Montagem limitada + aviso".



### 7.1.2. Descrição Detalhada de Coleta de Requisitos:

O diagrama de sequência modela as interações entre o "Usuário", a "Interface (Chatbot)", o "Sistema" e os serviços de detecção de localização e clima para coletar todos os requisitos necessários para a montagem de um PC.

#### Fluxo de Coleta de Dados Essenciais (Orçamento e Perfil)

- **1. Início:** O "Usuário" envia a mensagem "Acessa a funcionalidade 'Montar PC'".
- **2. Solicitação:** A "Interface (Chatbot)" envia a mensagem "Solicita Orçamento" para o "Usuário", que responde com "Fornece a informação".
- **3. Processamento:** A "Interface (Chatbot)" envia a mensagem "Envia Orçamento" para o "Sistema", que responde com uma "Confirma recebimento". Este ciclo se repete para o "Solicita o Perfil de Uso".

#### Fluxo de Coleta de Localização (Bloco alt)

- **4. Decisão:** A "Interface (Chatbot)" envia a mensagem "Oferece usar localização para recomendação térmica" para o "Usuário", que deve decidir entre "Permite" ou "Recusa".
- **Cenário "Usuário permite a localização":**
  - **5. Detecção:** A "Interface (Chatbot)" envia "SolicitaDeteccaoDeLocalizacao()" para o "Sistema", que aciona o "Serviço de Detecção de IP" com a mensagem "GetUserID()".
  - **6. Obtenção de Temperatura:** O "Serviço de Detecção de IP" retorna o "Retorna(IP)", que o "Sistema" usa para enviar a mensagem "GetTemperaturaPorIP(IP)" para o "Serviço de API de Clima".
  - **7. Resposta:** O "Serviço de API de Clima" retorna a "Retorna(Temperatura)" para o "Sistema".
- **Cenário "Usuário recusa a localização":**
  - **5. Solicitação Manual:** A "Interface (Chatbot)" envia a mensagem "Solicita insercao manual da temperatura" para o "Usuário".
  - **6. Resposta Manual:** O "Usuário" responde com "Insere(Temperatura)", e a "Interface (Chatbot)" envia "Envia(TemperaturaManual)" para o "Sistema".
  - **7. Confirmação:** O "Sistema" retorna "Retorna(TemperaturaOK)" para a "Interface (Chatbot)".

## Fluxo de Coleta de Requisitos Adicionais (Bloco loop)

- **8. Loop de Coleta:** A "Interface (Chatbot)" inicia um loop que dura "até que os requisitos para o perfil sejam suficientes". Dentro do loop, o "Chatbot" envia a mensagem "Solicita próximo requisito" e o "Usuário" responde com "Fornece(Requisito)".
- **9. Processamento Interno:** A "Interface (Chatbot)" envia "Envia(Requisito)" para o "Sistema", que retorna uma "Confirmação". Este ciclo se repete até que a condição do loop seja satisfeita.

## Finalização do Fluxo

- **10. Conclusão:** Após o loop, o "Sistema" executa a mensagem "FinalizarColetaDeDados()" e, em seguida, a mensagem "ArmazenaPreferencias(TodosOsDados)" em si mesmo, para persistir as informações.
- **11. Notificação Final:** O "Sistema" retorna uma "ConfirmaçãoFinal" para a "Interface (Chatbot)", que exibe a mensagem "Exibe confirmação e avança para a próxima etapa" para o "Usuário".

### 7.1.3. Descrição Detalhada de Configuração Automatizada:

O diagrama de sequência detalha o fluxo de interações entre os objetos "Usuário", "Interface", "Sistema" e diversos serviços (IA, Validador, API de Preços e Geolocalização) para a funcionalidade de configuração automatizada de um PC. O processo é dividido em três fases principais: coleta de dados, geração da sugestão e a decisão do usuário.

## Fluxo de Coleta de Dados

- **1. Início e Solicitação:** O "Usuário" envia a mensagem "Acessa 'Montar PC'", e a "Interface" responde com a mensagem "Solicita orçamento, perfil e ambiente".
- **2. Envio de Dados:** O "Usuário" envia a mensagem "Informa dados" para a "Interface", que a repassa para o "Sistema" com "Envia dados do usuário".
- **3. Coleta de Localização (Bloco alt):** O "Sistema" solicita a temperatura do ambiente.
  - **Cenário Permissão Concedida:** O "Sistema" envia a mensagem "Solicita cidade/IP" para o "Serviço de Geolocalização", que retorna a "Retorna temperatura".

- **Cenário Permissão Negada:** O "Usuário" fornece a "Informa temperatura manualmente".

### Fluxo de Geração da Sugestão

- **4. Processamento da IA:** O "Sistema" envia a mensagem "Processar dados e gerar sugestão de peças" para a "IA de Recomendação", que retorna a "Retorna build sugerida".
- **5. Validação:** O "Sistema" envia a mensagem "Verificar compatibilidade das peças" para o "Validador de Compatibilidade", que retorna "Compatível / Incompatível".
- **6. Busca de Preços:** O "Sistema" envia a mensagem "Buscar links e valores atualizados" para a "API de Preços", que responde com "Retorna links de compra".
- **7. Exibição:** O "Sistema" exibe a "Exibe build sugerida + preços + justificativas" na "Interface" para o "Usuário".

### Fluxo de Decisão do Usuário (Bloco alt)

- **8. Decisão do Usuário:** O processo aguarda a decisão do "Usuário" com a mensagem "Aceita ou rejeita sugestão".
- **Cenário de Aceitação (Bloco alt aninhado):**
  - **Cenário "Está logado":** O "Sistema" envia "Salvar/exportar build (CU1)" para o banco de dados e retorna uma "Confirmação de salvamento" para a "Interface".
  - **Cenário "Não logado":** O "Sistema" responde com "Prossegue sem salvar" para a "Interface".
- **Cenário de Rejeição:**
  - **9. Redirecionamento:** O "Sistema" envia a mensagem "Redireciona para montagem manual (CU4)" para a "Interface", que inicia um novo processo.

#### 7.1.4. Descrição Detalhada de Montagem Manual:

O diagrama de sequência detalha as interações entre o "Usuário", o "Sistema (UI)" e os serviços de validação para o processo de montagem e edição manual de um build de PC. O fluxo é segmentado em etapas de seleção, validação e finalização, com validações em tempo real.

## Fluxo de Início da Montagem Manual

- **1. Início:** O "Usuário" envia a mensagem "Acessa montagem manual ou edição de build" para o "Sistema (UI)".
- **2. Exibição:** O "Sistema (UI)" responde com a mensagem "Exibir categorias de componentes" para o "Usuário", iniciando o processo de seleção.

## Fluxo de Seleção de Componentes (Bloco loop)

- **3. Seleção:** O "Usuário" envia a mensagem "Seleciona ou altera componente" para o "Sistema (UI)", que aciona o "Validador Compatibilidade" com a mensagem "Validar compatibilidade técnica".
- **4. Validação (Bloco alt):** O sistema processa a resposta de validação técnica.
  - **Cenário [Compatível]:** O "Sistema (UI)" envia uma mensagem para o "Validador Gabinete" para "Validar compatibilidade física (se gabinete)".
    - **Validação Física (Bloco alt aninhado):**
      - **Cenário [Gabinete compatível]:** O processo continua sem interrupções.
      - **Cenário [Gabinete incompatível]:** O "Sistema (UI)" exibe "Exibir alerta de incompatibilidade física" para o "Usuário".
  - **Cenário [Incompatível]:** O "Sistema (UI)" exibe "Exibir mensagem de incompatibilidade" para o "Usuário". Este ciclo se repete até que o "Usuário" termine a seleção de componentes.

## Fluxo de Finalização

- **5. Finalização:** O "Usuário" envia a mensagem "Finaliza montagem" para o "Sistema (UI)", que solicita ao "Validador Compatibilidade" a mensagem "Verificar se há incompatibilidades críticas".
- **6. Verificação Crítica (Bloco alt):**
  - **Cenário [Build válida]:** O "Validador Compatibilidade" retorna "Build aprovada" para o "Sistema (UI)". O sistema então executa a mensagem "Salvar ou exportar build", finalizando o processo com sucesso.

- **Cenário [Incompatibilidade crítica]:** O "Sistema (UI)" exibe a mensagem "Bloquear finalização, exibir mensagem de erro" para o "Usuário", impedindo a conclusão do processo até que as correções sejam feitas.

#### 7.1.5. Descrição Detalhada de Sistema de Refrigeração:

O diagrama de sequência detalha o fluxo de interações entre os objetos "Usuário", "Sistema" e os serviços "Módulo Térmico" e "Módulo de Validação" para gerar e validar uma recomendação de refrigeração. O fluxo inclui a coleta de dados, a geração de sugestões, a validação e a interação com o usuário, com lógicas de ajuste e finalização.

#### Fluxo de Coleta e Geração da Recomendação

- **1. Solicitação:** O "Usuário" envia a mensagem "solicitaRecomendacao()" para o "Sistema".
- **2. Verificação de Dados (Bloco alt):** O "Sistema" executa "verificaDados()". Se os dados estiverem incompletos, o "Sistema" envia "solicitaPreenchimento()" e recebe "enviaDadosFaltantes()", criando um loop de coleta.
- **3. Processamento Térmico:** Após a validação dos dados, o "Sistema" envia as mensagens "interpretarAmbiente()", "calcularTDP()" e "gerarRecomendacao()" para o "Módulo Térmico".
- **4. Validação:** O "Módulo Térmico" retorna a "retornaRecomendacao()" para o "Sistema", que envia "validarRecomendacao()" para o "Módulo de Validação". O "Módulo de Validação" verifica a compatibilidade e retorna o "resultadoValidacao()".

#### Fluxo de Interação com o Usuário e Ajustes (Bloco alt principal)

- **Cenário [Solução válida]:** O "Sistema" exibe a recomendação ao "Usuário" com "exibeRecomendacao()". Um **bloco loop** para ajustes é iniciado:
  - O "Usuário" pode enviar uma "sugestao()", que o "Sistema" valida e retorna o "resultadoSugestao()".
  - Dentro do loop, um **bloco alt** é usado para a decisão final do usuário:
    - **Aceitação:** Se o "Usuário" "Aceita recomendacao", o "Sistema" "confirmaRecomendacao()" e "finaliza()".
    - **Rejeição:** Se o "Usuário" "Rejeita recomendacao", o "Sistema" executa "encerraSemSalvar()".

- **Cenário [Solução inválida]:** O "Sistema" tenta um ajuste automático com a mensagem "tentarAjusteAutomatico()" para o "Módulo Térmico". O resultado do ajuste é tratado em um **bloco alt aninhado**:
  - **Ajuste Válido:** Se o resultado for válido, a mensagem "exibeRecomendacaoAjustada()" é enviada.
  - **Ajuste Inválido:** Se o ajuste falhar, o "Sistema" envia a mensagem "informarInviabilidade()" ao "Usuário".

## 8. DIAGRAMA DE CLASSES

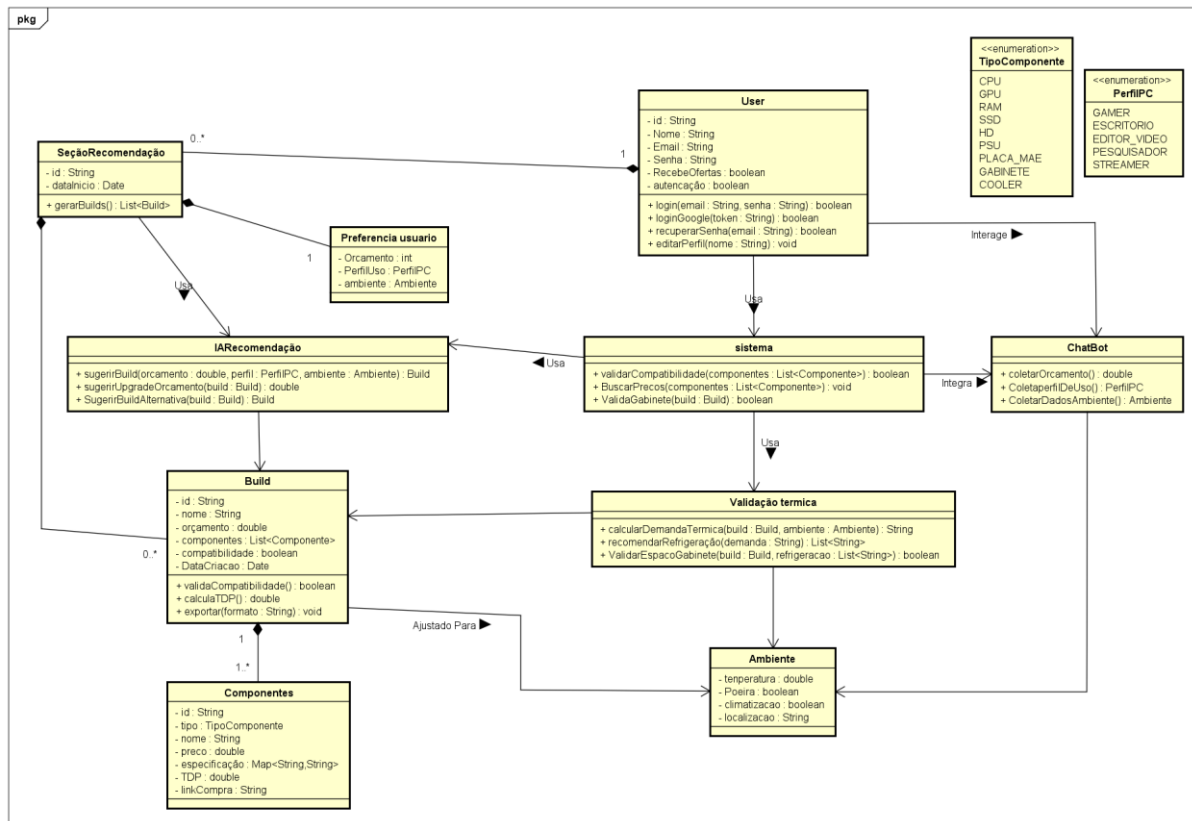
O Diagrama de Classes é um dos principais diagramas estruturais da UML. Ele representa a estrutura estática do sistema, modelando as classes, seus atributos, métodos e os relacionamentos entre elas. Esse diagrama é essencial para o planejamento da arquitetura do software, auxiliando na identificação dos componentes que formarão a base do código-fonte.

Os componentes Principais dos diagramas de classe são:

- **Classes:** Representam as entidades e os objetos do sistema.
- **Atributos:** São as propriedades e características das classes.
- **Métodos:** Representam o comportamento e as funcionalidades das classes.
- **Relacionamentos:** Mostram as conexões e interações entre as classes (como associações, herança e dependências).
- **Multiplicidade:** Define o número de instâncias de uma classe que se relacionam com outra.

A seguir temos o diagrama de classes do sistema:

Figura 23- Diagrama de Classe do sistema



Fonte: Autoria Própria.

## 8.1 Diagrama de Classes

### 8.1.1 Título e Metadados

- **Título:** Diagrama de Classes do Módulo de Recomendação de Builds e Validação Térmica
- **Identificador:** CD-001
- **Versão:** 1.0
- **Data:** 2025-05-030
- **Autor:** Arlison Gaspar
- **Contexto:**

Este diagrama modela o módulo de recomendação de configurações (builds) de computadores de um sistema que combina IA, validação térmica e interação via chatbot. Abrange as entidades responsáveis por representar usuários, preferências, builds e componentes, além dos serviços de recomendação, validação de

compatibilidade e cálculo térmico. Destina-se à equipe de desenvolvimento como especificação para implementação e testes.

### 8.1.2 Descrição das Classes

**Nota sobre visibilidade:** + = público, - = privado.

#### 8.1.2.1 SeçãoRecomendação

**Responsabilidade:** Agrupar recomendações geradas (conjunto de builds) em uma sessão com data de criação.

##### Atributos

Nome	Tipo	Visibilidade	Descrição	Restrições
<b>id</b>	String	-	Identificador único da seção de recomendações.	{unique}, {not null}
<b>dataInicio</b>	Date	-	Data/hora de criação da seção.	{not null}

##### Métodos

Nome	Parâmetros	Retorno	Visibilidade	Descrição
gerarBuilds()	—	List<Build>	+	Gera e retorna uma lista de Build com base nas preferências associadas.

#### 8.1.2.2 PreferenciaUsuario

**Responsabilidade:** Armazenar as preferências do usuário para guiar a recomendação (orçamento, perfil de uso e ambiente).

##### Atributos



Nome	Tipo	Visibilidade	Descrição	Restrições
<b>orcamento</b>	int	-	Orçamento máximo pretendido (moeda padrão do sistema).	{>=0}, {not null}
<b>perfilUso</b>	Perfil PC	-	Perfil de uso do PC (enum).	{not null}
<b>ambiente</b>	Ambiente	-	Objeto Ambiente com condições físicas do local.	{not null}

## Métodos

- Sem métodos públicos explícitos no diagrama (classe de modelo).

### 8.1.2.3 IARecomendação

**Responsabilidade:** Gerar sugestões de builds e alternativas usando IA/heurísticas.

## Atributos

- Não especificados no diagrama (assume-se estado interno opcional).

## Métodos

Nome	Parâmetros	Retorno	Visibilidade	Descrição
<b>sugerirBuild(orcamento, perfil, ambiente)</b>	orcamento: double, perfil: PerfilPC, ambiente: Ambiente	Build	+	Gera uma Build inicial considerando parâmetros fornecidos.
<b>sugerirUpgradeOrcamento(build)</b>	build: Build	double	+	Estima orçamento adicional necessário para upgrade.
<b>sugerirBuildAlternativa(build)</b>	build: Build	Build	+	Retorna uma alternativa balanceada para

	a Build fornecida.
--	--------------------

#### 8.1.2.4 Build

**Responsabilidade:** Representar uma configuração completa de computador (componentes, custo e metadados). Sujeita a validações de compatibilidade e térmicas.

##### Atributos

Nome	Tipo	Visibilidade	Descrição	Restrições
<b>id</b>	String	-	Identificador único da build.	{unique}, {not null}
<b>nome</b>	String	-	Nome/alias da build.	{not null}
<b>orcamento</b>	double	-	Custo total estimado.	{>=0}
<b>componentes</b>	List<Componente>	-	Lista de componentes que compõem a build.	{not empty}
<b>compatibilidade</b>	boolean	-	Indica se a build passou na validação de compatibilidade.	{default=false}
<b>dataCriacao</b>	Date	-	Data/hora de criação da build.	{not null}

##### Métodos

Nome	Parâmetros	Retorno	Visibilidade	Descrição
<b>validaCompatibilidade()</b>	—	boolean	+	Verifica compatibilidade entre

				componentes (soquete, dimensões, PSU, etc.).
<b>calculaTDP()</b>	—	double	+	Soma os TDPs dos componentes relevantes e retorna demanda térmica total.
<b>exportar(formato)</b>	formato: String	void	+	Exporta os dados da Build em formato solicitado (ex.: JSON, CSV, PDF).

### 8.1.2.5 Componente

**Responsabilidade:** Representar um componente de hardware (CPU, GPU, RAM, etc.) com suas propriedades técnicas e preço.

#### Atributos

Nome	Tipo	Visibilidade	Descrição	Restrições
<b>id</b>	String	-	Identificador único do componente.	{unique}, {not null}
<b>tipo</b>	TipoComponente	-	Tipo do componente (enum).	{not null}
<b>nome</b>	String	-	Nome comercial do componente.	{not null}
<b>preco</b>	double	-	Preço unitário do componente.	{>=0}
<b>especificacao</b>	Map<String,String>	-	Mapa de especificações (soquete, clock, etc.).	—
<b>TDP</b>	double	-	Potência de dissipação térmica (Watts).	{>=0}

<b>linkCompra</b>	String	-	URL de referência/compra (opcional).	{format=url}
-------------------	--------	---	--------------------------------------	--------------

## Métodos

- Nenhum método público especificado no diagrama.

### 8.1.2.6 Ambiente

**Responsabilidade:** Modelar as condições do local de uso, utilizadas nos cálculos térmicos e recomendações.

## Atributos

Nome	Tipo	Visibilidade	Descrição	Restrições
<b>temperatura</b>	doub le	-	Temperatura ambiente média (°C).	{not null}
<b>poeira</b>	boole an	-	Indica presença/risco de poeira.	{default=false }
<b>climatizacao</b>	boole an	-	Indica presença de climatização (ar-condicionado).	{default=false }
<b>localizacao</b>	String	-	Descrição da localização (ex.: sala, data center).	—

## Métodos

- Não há métodos públicos especificados no diagrama.

### 8.1.2.7 User

**Responsabilidade:** Representar o usuário do sistema com dados de autenticação e preferências de comunicação.

## Atributos

Nome	Tipo	Visibilidade	Descrição	Restrições
------	------	--------------	-----------	------------

<b>id</b>	String	-	Identificador único do usuário.	{unique}, {not null}
<b>nome</b>	String	-	Nome completo do usuário.	{not null}
<b>email</b>	String	-	E-mail de login/contato.	{unique}, {not null}, {format=email}
<b>senha</b>	String	-	Hash da senha do usuário.	{not null}
<b>recebeOfertas</b>	boolean	-	Indica interesse em receber ofertas.	{default=false}
<b>autenticacao</b>	boolean	-	Estado de autenticação (sessão ativa).	{default=false}

## Métodos

Nome	Parâmetros	Retorno	Visibilidade	Descrição
<b>login(email, senha)</b>	email: String, senha: String	boolean	+	Autentica usuário por credenciais.
<b>loginGoogle(token)</b>	token: String	boolean	+	Autentica via token OAuth (Google).
<b>recuperarSenha(email)</b>	email: String	boolean	+	Inicia processo de recuperação de senha (envia e-mail).
<b>editarPerfil(nome)</b>	nome: String	void	+	Atualiza o nome do usuário (e outros campos por extensão).

#### 8.1.2.8 Sistema

**Responsabilidade:** Serviços centrais para validação de compatibilidade, busca de preços e validação de gabinete.

##### Atributos

- Não explicitados no diagrama (classe de serviço/fachada).

##### Métodos

Nome	Parâmetros	Retorno	Visibilidade	Descrição
<b>validarCompatibilidade(componentes)</b>	componentes: List<Componente>	boolean	+	Verifica compatibilidade técnica entre componentes.
<b>buscarPrecos(componentes)</b>	componentes: List<Componente>	void	+	Atualiza/preenche preços consultando fontes externas (APIs/DB).
<b>validaGabinete(build)</b>	build: Build	boolean	+	Verifica se gabinete suporta componentes e refrigeração.

#### 8.1.2.9 Validação Térmica

**Responsabilidade:** Calcular demanda térmica, recomendar refrigeração e validar espaço físico.

##### Atributos

- Não explicitados no diagrama (serviço).

##### Métodos

---

N P R V D

o a e i e

n r t s s

e â o i c

n r b r

e n i i

t o l ç

r i ã

o d o

s a

d

e

---

**c b S + C**

**a u t a**

**l i r l**

**c l i c**

**u d n u**

**l : g l**

**a B a**

**r u d**

**D i e**

**e l n**

**n d a**

**a , n**

**n a d**

**d n a**

**a b t**

**T i é**

**e e r**

**r n n**

**n t i**

**i e c**

**c : a**

**a A (**

---

---

( n c  
b b a  
u i t  
i e e  
l n g  
d t o  
, e r  
a i  
n a  
b /  
i d  
e e  
n s  
t c  
e r  
) i  
ç  
ã  
o  
)  
c  
o  
n  
b  
a  
s  
e  
e  
n  
T  
D  
P  
s  
e

---



---

a  
n  
b  
i  
e  
n  
t  
e  
.  
**r d L + R**  
**e e i e**  
**c n s t**  
**o a t o**  
**n n < r**  
**e d S n**  
**n a t a**  
**d : r l**  
**a S i i**  
**r t n s**  
**R r g t**  
**e i > a**  
**f n d**  
**r g e**  
**i s**  
**g o**  
**e l**  
**r u**  
**a ç**  
**c ã**  
**a e**  
**o s**  
**( d**  
**d e**  
**e r**

---

<b>n</b>	e
<b>a</b>	f
<b>n</b>	r
<b>d</b>	i
<b>a</b>	g
<b>)</b>	e
	r
	a
	ç
	ã
	o
	r
	e
	c
	o
	n
	e
	n
	d
	a
	d
	a
	s
	.
<b>v b b + V</b>	
<b>a u o</b>	e
<b>l i o</b>	r
<b>i l l</b>	i
<b>d d e</b>	f
<b>a : a</b>	i
<b>r B n</b>	c
<b>E u</b>	a
<b>s i</b>	e
<b>p l</b>	s

---

<b>a</b>	d	p
<b>c</b>	,	a
<b>o</b>	r	ç
<b>C</b>	e	o
<b>a</b>	f	f
<b>b</b>	r	í
<b>i</b>	i	s
<b>n</b>	g	i
<b>e</b>	e	c
<b>t</b>	r	o
<b>e</b>	a	n
<b>(</b>	c	o
<b>b</b>	a	g
<b>u</b>	o	a
<b>i</b>	:	b
<b>l</b>	L	i
<b>d</b>	i	n
<b>,</b>	s	e
<b>r</b>	t	t
<b>e</b>	<	e
<b>f</b>	S	p
<b>r</b>	t	a
<b>i</b>	r	r
<b>g</b>	i	a
<b>e</b>	n	a
<b>r</b>	g	s
<b>a</b>	>	s
<b>c</b>		o
<b>a</b>		l
<b>o</b>		u
<b>)</b>		ç
		õ
		e
		s

---

---

s  
u  
g  
e  
r  
i  
d  
a  
s  
.

---

#### 8.1.2.10 ChatBot

**Responsabilidade:** Coletar dados do usuário interativamente e integrar com o sistema de recomendação.

##### Atributos

- Não explicitados.

##### Métodos

Nome	Parâmetros	Retorno	Visibilidade	Descrição
<b>coletarOrcamento()</b>	—	doubl e	+	Interage para obter orçamento do usuário.
<b>coletarPerfilDeUso()</b>	—	Perfil PC	+	Determina perfil de uso do usuário por diálogo.
<b>coletarDadosAmbiente()</b>	—	Ambiente	+	Coleta dados de ambiente (temperatura, poeira, climatização).

#### 8.1.2.11 Enumerações

##### 8.1.2.11.1 TipoComponente

- Valores: CPU, GPU, RAM, SSD, HD, PSU, PLACA\_MAE, GABINETE, COOLER
- Descrição: Classifica o tipo funcional do componente.

#### 8.1.2.11.2 PerfilPC

- Valores: GAMER, ESCRITORIO, EDITOR\_VIDEO, PESQUISADOR, STREAMER
- Descrição: Perfil de uso que orienta prioridades (ex.: desempenho gráfico para GAMER).

### 8.1.3 Descrição dos Relacionamentos

#### 8.1.3.1 SeçãoRecomendação — Build

- **Tipo:** Composição/Associação
- **Classes:** SeçãoRecomendação ↔ Build
- **Papel:** SeçãoRecomendação agrupa builds geradas.
- **Multiplicidade:** SeçãoRecomendação 0..\* → Build (um para muitos).
- **Navegabilidade:** Bidirecional.

#### 8.1.3.2 SeçãoRecomendação — PreferenciaUsuario

- **Tipo:** Associação
- **Classes:** SeçãoRecomendação ↔ PreferenciaUsuario
- **Papel:** Preferências definem parâmetros para geração da seção.
- **Multiplicidade:** 1 → 1.
- **Navegabilidade:** Unidirecional (Seção usa Preferência).

#### 8.1.3.3 IARecomendação → Build

- **Tipo:** Dependência/Uso
- **Papel:** IA gera e modifica objetos Build.
- **Multiplicidade:** N/A.
- **Navegabilidade:** Unidirecional (IA → Build).

#### **8.1.3.4 Build ↔ Componente**

- **Tipo:** Composição
- **Papel:** Build agrega Componente.
- **Multiplicidade:** Build 1 → Componente 1..\* .
- **Navegabilidade:** Bidirecional.

#### **8.1.3.5 Build — Ambiente**

- **Tipo:** Associação (Ajustado para)
- **Papel:** Ambiente influencia recomendações e validações térmicas.
- **Multiplicidade:** Build 0..\* → Ambiente 1 (referência).
- **Navegabilidade:** Unidirecional.

#### **8.1.3.6 User → Sistema**

- **Tipo:** Associação/Uso
- **Papel:** Usuário invoca operações do sistema (login, solicita recomendações).
- **Multiplicidade:** 1 → 1.
- **Navegabilidade:** Unidirecional.

#### **8.1.3.7 Sistema → Validação Térmica**

- **Tipo:** Dependência/Uso
- **Papel:** Sistema delega cálculos térmicos.
- **Navegabilidade:** Unidirecional.

#### **8.1.3.8 ChatBot ↔ Sistema**

- **Tipo:** Integração
- **Papel:** ChatBot coleta e envia dados para o Sistema.
- **Navegabilidade:** ChatBot → Sistema.

#### **8.1.3.9 Validação Térmica — Build / Ambiente**

- **Tipo:** Associação/Dependência

- **Papel:** Avalia Build considerando Ambiente.
- **Navegabilidade:** Unidirecional (Validação Térmica lê Build e Ambiente).

#### 8.1.4 Regras de Negócio e Restrições

1. **IDs únicos:** id de User, Build, Componente e Seção Recomendação devem ser únicos e não nulos.
2. **Orçamento:** IA Recomendação.sugerirBuild não deve exceder o orçamento informado; se impossível, retornar alternativa ou sinalizar necessidade de upgrade (sugerirUpgradeOrçamento).
3. **Compatibilidade obrigatória:** Build.validaCompatibilidade() deve verificar soquete CPU/placa-mãe, tipo/freq. de memória, potência da PSU com margem (ex.: 25%), dimensões físicas e conectores.
4. **Cálculo térmico:** Build.calculaTDP() soma TDPs críticos; Validação Térmica ajusta por temperatura ambiente e presença/ausência de climatização; recomendações devem considerar risco de poeira.
5. **Validação de gabinete:** Sistema.validaGabinete(build) verifica espaço e suporte para a refrigeração sugerida.
6. **Armazenamento seguro:** User.senha deve ser armazenada como hash; autenticação deve suportar credenciais e OAuth (loginGoogle).
7. **Consentimento de marketing:** User.recebeOfertas controla comunicações (respeito a opt-in).
8. **ChatBot:** Deve validar entradas do usuário (ex.: numericidade do orçamento) antes de converter para PreferenciaUsuario/Ambiente.
9. **Exportação:** Build.exportar(formato) ao menos JSON e CSV; PDF opcional.
10. **Margens de segurança:** Ao calcular requisitos de PSU/refrigeração, aplicar margem (ex.: PSU recomendada = somatório\_TDP \* 1.25).

## 9. TECNOLOGIAS

O projeto Montador de PC Online será desenvolvido com base em tecnologias modernas e compatíveis com aplicações web interativas e escaláveis. A arquitetura da solução

está sendo planejada para proporcionar uma experiência fluida ao usuário, com forte integração a serviços externos e facilidade de manutenção por parte da equipe técnica.

Na camada de frontend, será utilizado HTML estruturado em conjunto com TypeScript, proporcionando maior robustez, segurança em tempo de desenvolvimento e melhor controle sobre os tipos de dados manipulados na aplicação. A interface será construída com foco em responsividade e usabilidade, seguindo um modelo de fluxo guiado (wizard) para facilitar a interação com o sistema.

No backend, será mantida a utilização de Node.js com Express.js para a construção de APIs REST que darão suporte à lógica de negócio, manipulação de dados e integração com serviços externos. A persistência dos dados será feita com o banco MongoDB, adequado para armazenar informações flexíveis como builds personalizadas, perfis de usuário e preferências específicas.

Para a funcionalidade de recomendação automatizada de peças, o sistema utilizará a API de inteligência artificial da Gemini (Google) por meio do Gemini Studio, que permitirá interpretar os requisitos informados pelo usuário (como orçamento, perfil de uso e ambiente físico) e retornar sugestões de configuração otimizadas. Essa abordagem evita a necessidade de desenvolver IA própria, reduz a complexidade do backend e aproveita a capacidade cognitiva de uma IA de última geração.

A plataforma também fará uso de APIs externas de clima (como Open-Meteo ou Meteostat) e de geolocalização (como ipinfo.io ou MaxMind) para ajustar automaticamente as sugestões térmicas conforme a cidade e o ambiente do usuário, com total respeito à privacidade e à legislação de proteção de dados.

Outros recursos técnicos incluirão login opcional com Google (OAuth 2.0), exportação de builds em PDF e XLSX por meio de bibliotecas como jsPDF e xlsx, e hospedagem da aplicação em plataformas como Vercel ou Render, que oferecem integração contínua, fácil escalabilidade e alta disponibilidade.

Durante o processo de desenvolvimento, serão utilizadas ferramentas como Git e GitHub para controle de versão, além de Trello ou Notion para organização das tarefas e acompanhamento das entregas. O monitoramento da aplicação em produção será realizado com Sentry ou ferramentas equivalentes para detecção de falhas e análise de desempenho.



Essa seleção tecnológica garante um desenvolvimento sólido, modular e aderente às melhores práticas atuais, além de permitir a evolução contínua do sistema com futuras integrações e funcionalidades mais avançadas.

## **10. ESCOPO DA PROTOTIPAÇÃO**

O protótipo do sistema Montador de PC Online tem como objetivo validar as principais funcionalidades da plataforma em sua versão inicial (MVP – Produto Mínimo Viável), permitindo testes de usabilidade, compreensão do fluxo e avaliação da experiência do usuário antes da implementação completa do sistema.

Esse protótipo será desenvolvido com foco nas funcionalidades essenciais, representando de forma fiel a interface, os fluxos de interação e os comportamentos esperados do sistema, ainda que sem a integração completa com bancos de dados, APIs externas ou funcionalidades finais.

### **Funcionalidades incluídas no protótipo:**

- Tela de início com opção de montar PC com ou sem login.
- Coleta de requisitos do usuário através de um assistente interativo (chatbot simulado).
- Entrada de orçamento e perfil de uso, com campos básicos.
- Recebimento de uma configuração recomendada (gerada de forma simulada ou com chamada à API da Gemini).
- Opção de montagem manual, permitindo substituir componentes sugeridos.
- Verificação visual de compatibilidade entre as peças escolhidas.
- Tela de resumo da build, exibindo as peças escolhidas, preço estimado e links fictícios.
- Função de exportação simulada, permitindo gerar uma visualização em PDF ou planilha.

### **Funcionalidades fora do escopo do protótipo:**

- Integração real com bancos de dados (persistência de builds).
- Cadastro e login funcional de usuários.
- Comunicação real com lojas de varejo ou scraping de preços.
- Acesso real à API da Gemini (poderá ser simulado com respostas pré-definidas).
- Ajustes de recomendação térmica com base em localização e temperatura real.

- Sistema avançado de recomendação térmica (substituído por versão simplificada no protótipo).
- Compartilhamento de builds ou painel administrativo.

### **Objetivos do protótipo:**

- Validar a navegação e a estrutura das interfaces.
- Coletar feedback do usuário quanto à clareza do fluxo e à apresentação das recomendações.
- Demonstrar o valor da solução proposta para possíveis stakeholders.
- Servir de base para a implementação iterativa da versão funcional do sistema.

## **11. CONCLUSÃO**

O projeto **Montador de PC Online** propõe uma solução inovadora e acessível para um problema comum enfrentado por usuários e empresas: a montagem eficiente de computadores personalizados. Através da combinação de tecnologias modernas, integração com uma API de IA da Gemini e um fluxo de navegação guiado, a plataforma pretende oferecer uma experiência inteligente, prática e ajustada às necessidades reais do usuário.

Com a definição clara de escopo, requisitos e objetivos, a equipe está preparada para iniciar as etapas de prototipagem, validação e desenvolvimento iterativo do sistema. Este documento serve como base sólida para orientar o time técnico, alinhar expectativas com stakeholders e garantir que a solução proposta atenda aos critérios de qualidade, usabilidade e inovação esperados.

## 12 REFERENCIAS

[1] PRESSMAN, R. S.; MAXIM, B. Engenharia de software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

[2] JORNAL DA USP. A expansão do mercado de games brasileiro se deve a mudanças no modo tradicional do trabalho. São Paulo, 2023. Disponível em: <https://jornal.usp.br/radio-usp/a-expansao-do-mercado-de-games-brasileiro-se-deve-a-mudancas-no-modo-tradicional-do-trabalho/>. Acesso em: 02 jun. 2025.

[3] KINGSTON. Os 10 principais erros que os iniciantes cometem ao montar um PC. 2023. Disponível em: <https://www.kingston.com/br/blog/gaming/top-10-pc-build-mistakes-beginners-make>. Acesso em: 30 mai. 2025.

[4] SOMMERVILLE, I. Engenharia de Software. 9. ed. São Paulo: Pearson Addison Wesley, 2011.

[5] BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. 2. ed. Rio de Janeiro: Campus, 2006.

[6] GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos. Porto Alegre: Bookman, 2000.

[7] Object Management Group (OMG). OMG Unified Modeling Language (OMG UML), Superstructure, V2.5.1. Disponível em: <https://www.omg.org/spec/UML/>. Acesso em: 05 jun. 2025.

[8] ADRENALINE. Guia rápido de montagem de PC! 2021. Disponível em: <https://www.adrenaline.com.br/hardware/guia-rapido-de-montagem-de-pc/>. Acesso em: 12 mai. 2025.

[9] PICHAU ARENA. Como Montar um PC Gamer: Guia Completo para 2025. 2025. Disponível em: <https://www.pichauarena.com.br/pichau-arena/como-montar-pc-gamer/>. Acesso em: 12 mai. 2025.

Este projeto visa facilitar a montagem de PCs personalizados com o uso de IA. O problema é a dificuldade dos usuários em escolher peças compatíveis e otimizadas. A solução é um site com IA que coleta preferências e gera builds automaticamente. Foi desenvolvido com TypeScript, Node.js e Gemini AI.

Para executar o projeto localmente, é necessário ter o Node.js instalado e uma chave de API do Gemini. O processo começa acessando a pasta raiz do projeto com o comando `cd codigo/MONTAGEM_DE_PC`. Em seguida, é preciso instalar as dependências utilizando os seguintes comandos no terminal: `npm install`, `npm install --save-dev @types/react @types/react-dom`, `npm install vite --save-dev` e `npm install jspdf`.

Como etapa opcional, recomenda-se rodar `npm audit fix --force` para corrigir automaticamente vulnerabilidades encontradas nas dependências.

Após isso, deve-se configurar a chave da API Gemini. Para isso, crie um arquivo chamado `.env.local` na raiz do projeto e adicione a variável `GEMINI_API_KEY` com sua respectiva chave.

Com tudo pronto, basta iniciar o projeto localmente executando o comando `npm run dev`.

**autor:** Arlison Gaspar de Oliveira, Ítalo Francisco Almeida de Oliveira, Gustavo de Oliveira Rego Moraes, Joao Pedro Miranda Sousa, Cauã Gabriel Santos Barros **contato:** [Arlison.go@discente.ufma.br](mailto:Arlison.go@discente.ufma.br)

**data última versão:** 18/07/2025

**versão:** 6.4

**outros repositórios:** <https://github.com/gustvo-olive>

**Agradecimentos:** Universidade Federal do Maranhão (UFMA), Professor Doutor Thales Levi Azevedo Valente, e colegas de curso.

Copyright/License

Este material é resultado de um trabalho acadêmico para a disciplina PROJETO E DESENVOLVIMENTO DE SOFTWARE, sob a orientação do professor Dr. THALES LEVI AZEVEDO VALENTE, semestre letivo 2025.1, curso Engenharia da Computação, na Universidade Federal do Maranhão (UFMA). Todo o material sob esta licença é software livre: pode ser usado para fins acadêmicos e comerciais sem nenhum custo. Não há

papelada, nem royalties, nem restrições de "copyleft" do tipo GNU. Ele é licenciado sob os termos da Licença MIT, conforme descrito abaixo, e, portanto, é compatível com a GPL e também se qualifica como software de código aberto. É de domínio público. Os detalhes legais estão abaixo. O espírito desta licença é que você é livre para usar este material para qualquer finalidade, sem nenhum custo. O único requisito é que, se você usá-los, nos dê crédito.

Licenciado sob a Licença MIT. Permissão é concedida, gratuitamente, a qualquer pessoa que obtenha uma cópia deste software e dos arquivos de documentação associados (o "Software"), para lidar no Software sem restrição, incluindo sem limitação os direitos de usar, copiar, modificar, mesclar, publicar, distribuir, sublicenciar e/ou vender cópias do Software, e permitir pessoas a quem o Software é fornecido a fazê-lo, sujeito às seguintes condições:

Este aviso de direitos autorais e este aviso de permissão devem ser incluídos em todas as cópias ou partes substanciais do Software.

O SOFTWARE É FORNECIDO "COMO ESTÁ", SEM GARANTIA DE QUALQUER TIPO, EXPRESSA OU IMPLÍCITA, INCLUINDO MAS NÃO SE LIMITANDO ÀS GARANTIAS DE COMERCIALIZAÇÃO, ADEQUAÇÃO A UM DETERMINADO FIM E NÃO INFRINGÊNCIA. EM NENHUM CASO OS AUTORES OU DETENTORES DE DIREITOS AUTORAIS SERÃO RESPONSÁVEIS POR QUALQUER RECLAMAÇÃO, DANOS OU OUTRA RESPONSABILIDADE, SEJA EM AÇÃO DE CONTRATO, TORT OU OUTRA FORMA, DECORRENTE DE, FORA DE OU EM CONEXÃO COM O SOFTWARE OU O USO OU OUTRAS NEGOCIAÇÕES NO SOFTWARE.

Para mais informações sobre a Licença MIT: <https://opensource.org/licenses/MIT>

---