**Framework**

**Front-End com**

**Consumo de API**

## **Documento de visão do negócio**

|  |  |
| --- | --- |
| **Elaboração:** | **Matheus B.R** |

## Finalidade

Este documento tem como finalidade apresentar uma visão abrangente do projeto

SENAI GAMES, uma loja virtual de jogos e produtos eletrônicos desenvolvida utilizando o

framework Angular. O documento descreve o escopo, objetivos, funcionalidades e

requisitos do sistema, fornecendo uma base sólida para o entendimento do projeto por

parte de desenvolvedores, stakeholders e usuários finais.

## Escopo

O escopo do projeto SENAI GAMES abrange o desenvolvimento de uma aplicação web para comércio eletrônico de jogos e produtos eletrônicos, com funcionalidades de navegação por categorias, visualização de produtos, sistema de login/cadastro e carrinho de compras. A aplicação foi desenvolvida utilizando o framework Angular, seguindo os princípios de Single Page Application (SPA), com componentes reutilizáveis e uma arquitetura modular. A documentação abrange: - Visão geral do produto e seu posicionamento no mercado Descrição da equipe desenvolvedora e público-alvo - Análise de alternativas e concorrências - Restrições técnicas e de negócio - Requisitos funcionais e não funcionais- Modelagem de casos de uso - Definições técnicas e referências

## Visão geral

O documento está estruturado para fornecer uma visão completa do projeto, abordando desde aspectos técnicos da implementação até os requisitos funcionais e não funcionais do sistema. Ele serve como referência para o desenvolvimento, manutenção e evolução do projeto, garantindo que todos os envolvidos tenham uma compreensão clara dos objetivos e funcionalidades do SENAI GAMES.

A documentação abrange:

- Visão geral do produto e seu posicionamento no mercado Descrição da equipe desenvolvedora e público-alvo

- Análise de alternativas e concorrências

- Restrições técnicas e de negócio

- Requisitos funcionais e não funcionais

- Modelagem de casos de uso

- Definições técnicas e referência

## Visão geral do produto

O SENAI GAMES é uma aplicação web desenvolvida em Angular que oferece uma interface intuitiva para compra de jogos e produtos eletrônicos. O sistema foi projetado seguindo os princípios de design responsivo e experiência do usuário, garantindo uma navegação fluida e agradável em diferentes dispositivos.

Principais características:

* **Página inicial com carrossel de destaques**: Apresenta banners promocionais e produtos em destaque por meio de um carrossel interativo implementado com Bootstrap.
* **Navegação por categorias**: Permite aos usuários filtrar produtos por categorias como Monitores, Notebooks, Jogos, Desktops e Peças/Acessórios.
* **Exibição de produtos mais vendidos:** Seção dedicada aos produtos mais populares, organizados em uma grade responsiva.
* **Sistema de login e cadastro de usuários:** Interface para autenticação e registro de novos usuários, com validação de formulários.
* **Layout responsivo:** Design adaptável, ajustando-se automaticamente a diferentes tamanhos de tela e dispositivos.
* **Componentes reutilizáveis:** Estrutura modular com componentes Angular independentes para navbar, footer, home e login.

A arquitetura do sistema segue o padrão de componentes do Angular, separando claramente apresentação (templates HTML e estilos CSS) e lógica de negócio (TypeScript), facilitando manutenção e evolução do código.

## Posicionamento no mercado

**SENAI GAMES** se posiciona como uma plataforma de e-commerce especializada em **jogos e produtos eletrônicos**, focada em oferecer uma **experiência de usuário fluida e moderna**, utilizando **tecnologias atuais** como Angular. O diferencial do produto está na combinação de:

1. **Interface moderna e intuitiva**: Design **clean e organizado** que facilita a navegação e descoberta de produtos.
2. **Foco no público gamer**: Conteúdo e produtos direcionados especificamente para **entusiastas de jogos e tecnologia**.
3. **Tecnologia de ponta**: Utilização do **framework Angular** para garantir uma **experiência de usuário dinâmica e responsiva**.
4. **Arquitetura escalável**: Estrutura de código modular que permite **fácil expansão e adição de novas funcionalidades**.

O sistema busca atender a crescente demanda por plataformas especializadas em produtos eletrônicos e jogos, oferecendo uma alternativa focada e otimizada para este nicho de mercado

## Descrição da equipe desenvolvedora e dos clientes

A equipe desenvolvedora é composta por um estudante do SENAI, com o projeto sendo desenvolvido como parte do currículo educacional. A equipe possui conhecimentos em desenvolvimento web front-end, com foco em Angular, HTML, CSS e TypeScript.

Público-alvo (clientes)

- Gamers: Entusiastas de jogos eletrônicos que buscam uma plataforma especializada para adquirir jogos e equipamentos.

- Entusiastas de tecnologia: Pessoas interessadas em produtos eletrônicos de última geração, como computadores, monitores e periféricos.

- Consumidores online: Usuários habituados a realizar compras pela internet, que valorizam uma experiência de compra fluida e segura.

- Profissionais de TI: Pessoas que necessitam de equipamentos específicos para trabalho ou estudo na área de tecnologia.

## Alternativas e concorrências

O mercado de e-commerce de jogos e produtos eletrônicos conta com grandes players como Steam, Epic Games Store, Amazon e Kabum. Cada um destes concorrentes possui características específicas:

Steam e Epic Games Store:

• Foco exclusivo em jogos digitais

• Plataformas próprias para execução dos jogos

• Comunidades ativas de jogadores

• Promoções sazonais

Amazon:

• Variedade ampla de produtos, não apenas eletrônicos

• Logística avançada e entregas rápidas

• Sistema de avaliações e recomendações

• Marketplace com múltiplos vendedores

Kabum e Pichau:

• Especialização em hardware e periféricos

• Foco no público gamer e entusiastas de tecnologia

• Conteúdo técnico especializado

• Promoções frequentes

## Restrições

O desenvolvimento e operação do SENAI GAMES está sujeito às seguintes restrições:

Restrições técnicas:

• Conectividade: O sistema requer conexão com internet para funcionamento, tanto para carregamento da aplicação quanto para futuras integrações com APIs.

• Compatibilidade: A aplicação é otimizada para navegadores modernos (Chrome, Firefox, Edge), podendo apresentar limitações em navegadores desatualizados.

• Hospedagem: Necessidade de servidor para hospedagem da aplicação Angular, com suporte a aplicações SPA.

• Integrações: Para funcionalidade completa, o sistema precisará de integração com APIs de pagamento, autenticação e gerenciamento de estoque (a serem implementadas).

Restrições de negócio:

• Escopo inicial: A primeira versão foca na interface de usuário e navegação, deixando para versões futuras a implementação completa do processo de compra.

• Recursos de desenvolvimento: Limitações de equipe e tempo por se tratar de um projeto educacional.

• Regulamentações: Necessidade de conformidade com leis de proteção de dados (LGPD) para implementação completa.

• Manutenção: Necessidade de atualizações periódicas para acompanhar as evoluções do framework Angular.

Estas restrições foram consideradas durante o desenvolvimento do projeto e devem ser observadas em futuras evoluções do sistema.

## Requisitos do produto

Requisitos Funcionais:

• RF001: Navegação por categorias

O sistema deve permitir a visualização de produtos organizados por categorias Implementado através de componentes Angular na página inicial

Prioridade: Alta

• RF002: Autenticação de usuários

O sistema deve permitir cadastro e login de usuários Implementado através do componente de login com validação de formulários

Prioridade: Alta

• RF003: Exibição de produtos em destaque

O sistema deve exibir produtos em destaque na página inicial através de carrossel Implementado utilizando componente de carrossel do Bootstrap

Prioridade: Média

• RF004: Navegação entre páginas

• O sistema deve permitir navegação fluida entre diferentes páginas sem recarregamento completo

Implementado através do sistema de rotas do Angular

Prioridade: Alta

• RF005: Responsividade

O sistema deve se adaptar a diferentes tamanhos de tela e dispositivos • Implementado através de CSS responsivo e classes do Bootstrap

Prioridade: Alta

• RF006: Botão "Voltar ao Topo"

O sistema deve oferecer um botão para retornar ao topo da página quando o usuário rolar para baixo • Implementado através de JavaScript/TypeScript com detecção de scroll

Prioridade: Baixa

Requisitos Não Funcionais:

• RNF001: Framework Angular

O sistema deve ser desenvolvido utilizando o framework Angular

Versão utilizada: Angular 17+

Prioridade: Alta

RNF002: Desempenho

O sistema deve ter tempo de carregamento inicial inferior a 3 segundos em conexões padrão

Implementado através de otimizações de build do Angular

Prioridade: Média

RNF003: Acessibilidade

O sistema deve seguir padrões básicos de acessibilidade web

Implementado através de atributos ARIA e estrutura semântica de HTML

Prioridade: Média

RNF004: Compatibilidade

O sistema deve ser compatível com os principais navegadores (Chrome, Firefox, Edge, Safari)

Testado em múltiplas plataformas

Prioridade: Alta

RNF005: Modularidade

O código deve seguir princípios de modularidade e componentização

Implementado através da arquitetura de componentes do Angular

Prioridade: Alta

## Modelagem de caso de uso

**Caso de Uso: Realizar Login**

**Ator Principal:** Usuário não autenticado  
**Pré-condições:** Usuário possui cadastro no sistema

**Fluxo Principal**

1. Usuário acessa a página de login através do link na barra de navegação.
2. Sistema exibe formulário de login com campos para usuário e senha.
3. Usuário preenche os campos de usuário e senha.
4. Usuário clica no botão "Login".
5. Sistema valida as credenciais.
6. Sistema redireciona para a página inicial com usuário autenticado.

**Fluxos Alternativos**

* Se as credenciais forem inválidas, o sistema exibe uma mensagem de erro.
* Se o usuário não preencher todos os campos obrigatórios, o sistema destaca os campos faltantes.

**Implementação Técnica**

* **Componente Angular:** LoginComponent
* **Validação de formulário:** Implementada com Angular Forms
* **Feedback visual:** Classes CSS de validação

**Caso de Uso: Realizar Cadastro**

**Ator Principal:** Visitante  
**Pré-condições:** Nenhuma

**Fluxo Principal**

1. Usuário acessa a página de login.
2. Usuário clica no botão "Não é cadastrado ainda? Cadastre-se".
3. Sistema exibe formulário de cadastro.
4. Usuário preenche os campos de usuário, e-mail e senha.
5. Usuário clica no botão "Cadastrar".
6. Sistema valida os dados do formulário.
7. Sistema registra o novo usuário.
8. Sistema exibe mensagem de sucesso.

**Fluxos Alternativos**

* Se o e-mail já estiver cadastrado, o sistema exibe uma mensagem de erro.
* Se a senha não atender aos requisitos mínimos, o sistema exibe um alerta.

**Implementação Técnica**

* **Componente Angular:** LoginComponent com toggle para exibição do formulário de cadastro.
* **Validação:** Verificação de formato de e-mail e comprimento mínimo de senha.
* **Interação:** Implementada com TypeScript e Angular Forms.

**Caso de Uso: Navegar por Categorias**

**Ator Principal:** Qualquer usuário  
**Pré-condições:** Nenhuma

**Fluxo Principal**

1. Usuário acessa a página inicial.
2. Sistema exibe seção de categorias com imagens representativas.
3. Usuário seleciona uma categoria clicando na imagem.
4. Sistema exibe produtos da categoria selecionada.

**Implementação Técnica**

* **Componente Angular:** HomeComponent
* **Layout:** Grid responsivo com CSS/Bootstrap
* **Efeitos visuais:** Transição de escala ao passar o mouse sobre as categorias

## Definição, acrônimos e abreviações

* **Angular**: Framework para desenvolvimento de aplicações web mantido pelo Google.
* **API (Application Programming Interface)**: Interface de programação que permite a comunicação entre diferentes sistemas.
* **SPA (Single Page Application)**: Aplicação de página única, onde a navegação ocorre sem recarregar a página inteira.
* **HTML (HyperText Markup Language)**: Linguagem de marcação utilizada para estruturar o conteúdo web.
* **CSS (Cascading Style Sheets)**: Linguagem de estilização utilizada para definir a aparência dos elementos HTML.
* **TypeScript**: Linguagem de programação que estende o JavaScript, adicionando tipagem estática.
* **Bootstrap**: Framework CSS para desenvolvimento de interfaces responsivas.
* **Componente**: Unidade básica de interface no Angular, encapsulando HTML, CSS e lógica.
* **Roteamento**: Sistema de navegação entre diferentes views em uma SPA.
* **DOM (Document Object Model)**: Representação em memória da estrutura de um documento HTML.
* **Responsivo**: Design que se adapta a diferentes tamanhos de tela e dispositivos.

## Referências

**Referências**

* **Documentação oficial do Angular**: [Angular Dev](https://angular.dev/)
* **Documentação do Bootstrap**: [Bootstrap Docs](https://getbootstrap.com/docs/)
* **Repositório do projeto**: [GitHub - LOJA GAMES SENAI](https://github.com/MatheusBenvindo/LOJA-GAMES-SENAI)
* **MDN Web Docs (Mozilla Developer Network)**: [MDN Web Docs](https://developer.mozilla.org/)
* **TypeScript Documentation**: [TypeScript Docs](https://www.typescriptlang.org/docs/)