

Relatório da aplicação web para venda de jogos digitais “Edugames”

Usabilidade e Desenvolvimento Web
UNIFACS

Semestre 2/2025

Salvador, Bahia, Brasil

Informações do Projeto

Autores:

Amanda Kelly Brito Cerqueira (12722211832) Carolina
Paiva Carvalho Brinço (1272225024)

Luís André Gomes de Oliveira Silva (12722131007) Paulo
Henrique Pereira Araujo Piedade (12722116801)

Disciplina: Usabilidade e Desenvolvimento Web

Instituição: UNIFACS Semestre:

2/2025

Repositório GitHub:

<https://github.com/luis-a-silva/A3UCUsabilidadeReact>

Introdução

Com o avanço da sociedade digital, o acesso a produtos, serviços e informações ocorre, em grande parte, por meio de interfaces web. Nesse contexto, a qualidade da experiência do usuário e a acessibilidade das aplicações tornam-se fatores determinantes para a inclusão digital. Aplicações com baixa usabilidade podem dificultar ou até impedir o acesso de determinados grupos de pessoas, reforçando barreiras tecnológicas e sociais.

Com base nesses princípios, o projeto EduGames foi desenvolvido como uma aplicação web voltada para a venda de jogos digitais, com foco em design intuitivo, acessibilidade e responsividade. A proposta visa unir entretenimento e tecnologia em uma plataforma funcional e inclusiva.

Objetivo

O principal objetivo do EduGames é oferecer uma plataforma web simples e acessível para a compra de jogos digitais. O sistema busca proporcionar uma experiência de navegação agradável, intuitiva e inclusiva, permitindo que o usuário encontre, avalie e adquira jogos de forma prática e visualmente atraente.

Justificativa

O mercado de jogos digitais é um dos setores mais relevantes da economia tecnológica contemporânea. Entretanto, muitas plataformas de vendas não priorizam a acessibilidade e a experiência do usuário. O EduGames surge como uma alternativa inclusiva, com design limpo, responsivo e compatível com diversos dispositivos, incluindo smartphones e tablets.

Tecnologias e metodologias utilizadas

- **HTML5:** estruturação semântica e organização do conteúdo das páginas.
- **CSS3:** definição do estilo visual, cores, tipografia e layout responsivo.
- **JavaScript:** implementação de funcionalidades interativas e validações.
- **Font Awesome:** biblioteca de ícones utilizada para aprimorar a comunicação visual e a navegabilidade.
- **Google Fonts:** biblioteca utilizada para importação e aplicação de fontes personalizadas, garantindo melhor legibilidade e estética.
- **React:** Framework JavaScript para construir interfaces de usuário (UIs) através de **Componentes** reutilizáveis, encapsulando estrutura, estilo e comportamento, e otimizando a interatividade.

Relatório – Aplicação Web Edugames

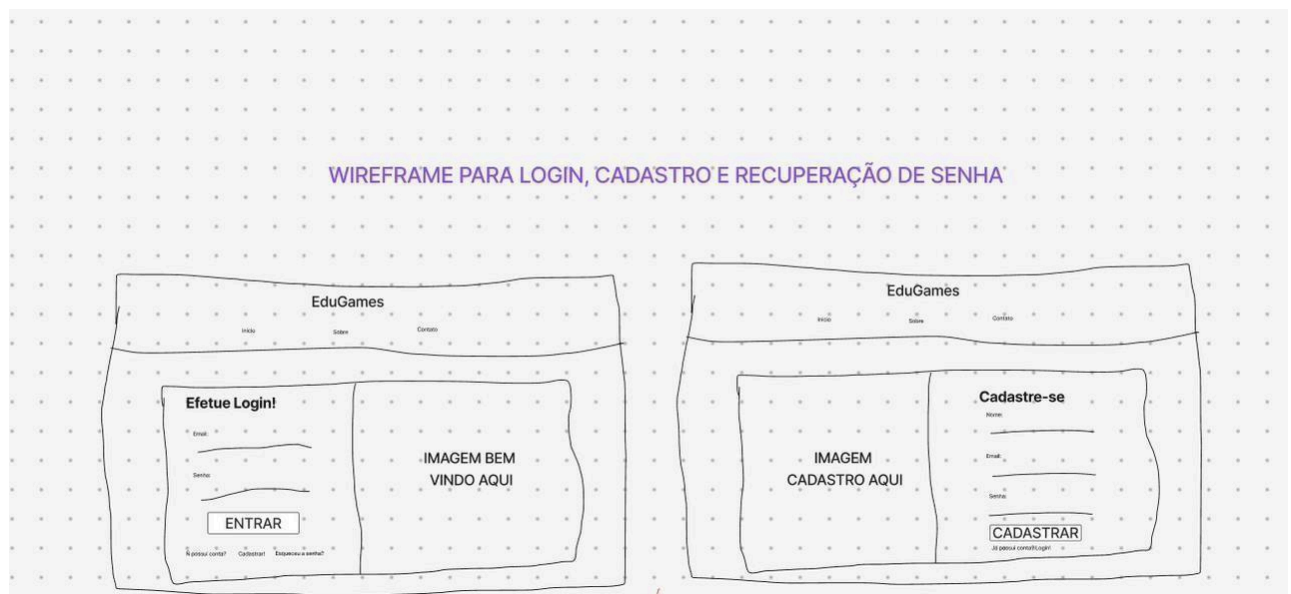
- **Vite:** Ferramenta de *build* ultra-rápida que serve como **servidor de desenvolvimento** moderno, acelerando o ambiente e o *hot reloading* para um desenvolvimento ágil.
- **Axios:** Biblioteca JavaScript utilizada para realizar **requisições HTTP** simples e padronizadas, facilitando a comunicação com APIs e o carregamento de conteúdo dinâmico.
- **Router:** Implementa o **roteamento** em aplicações de página única (SPA), gerenciando a URL para exibir o Componente correto e garantindo a navegabilidade fluida.

Wireframes

Antes do desenvolvimento da interface definitiva, foram criados wireframes representando a estrutura visual e o fluxo de navegação do EduGames.

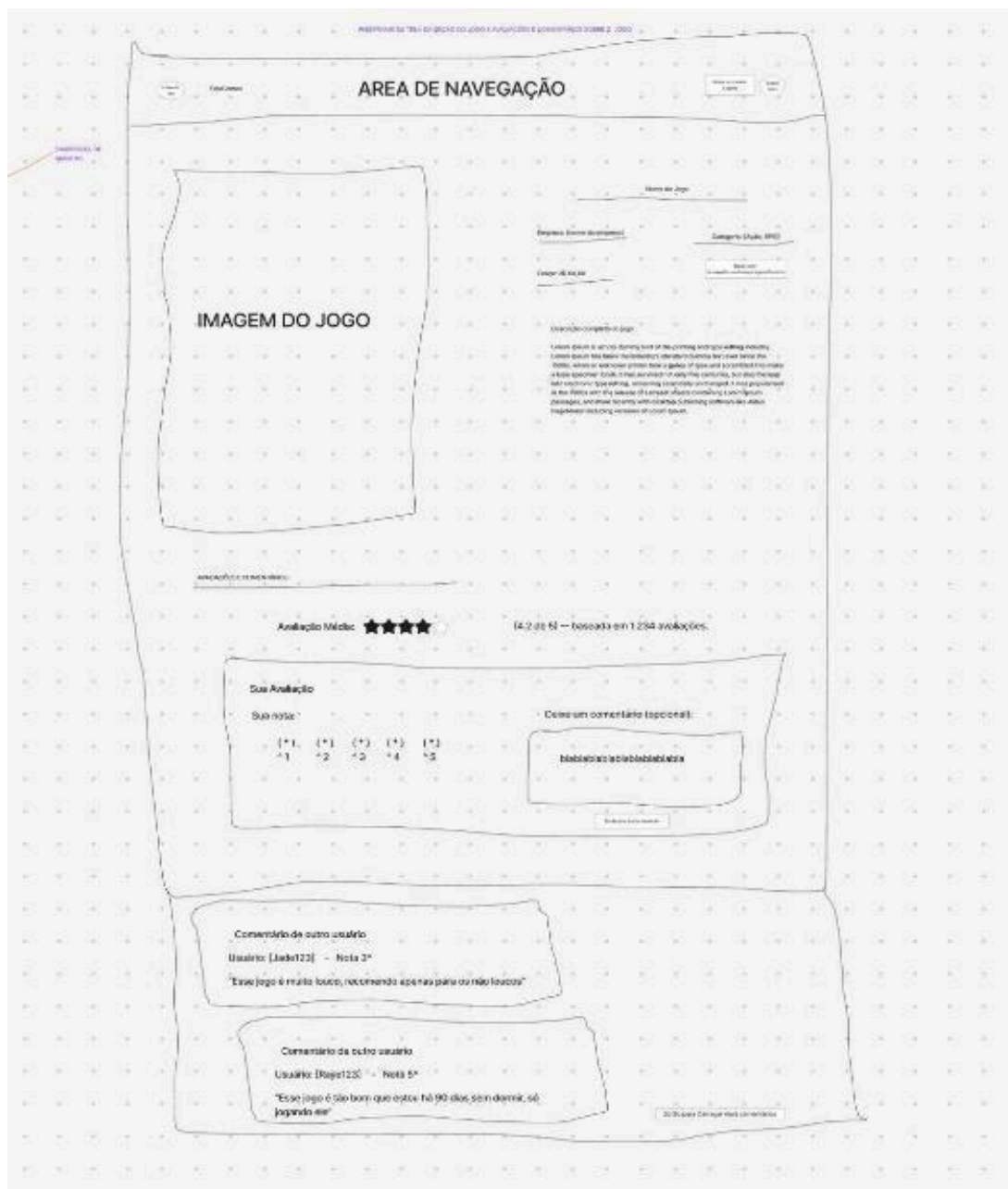
Os wireframes foram desenvolvidos com base em princípios de design centrado no usuário (DCU), priorizando a clareza, a simplicidade e a coerência entre as telas. A criação desses protótipos permitiu identificar possíveis ajustes de layout, posicionamento de botões, distribuição de conteúdo e harmonia visual entre as seções.

5.1 Wireframe para login, cadastro e recuperação de senha



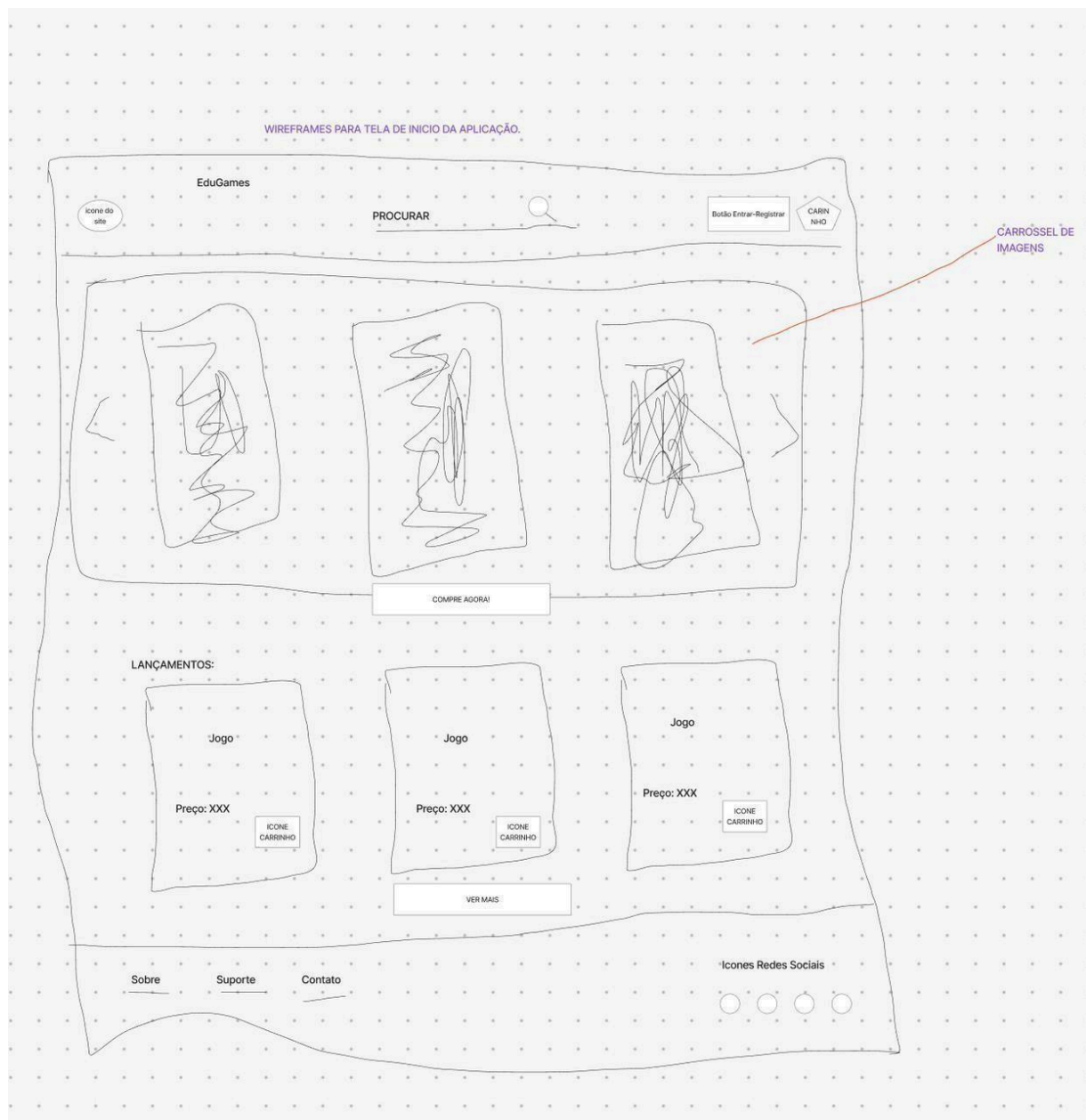
O wireframe de login, cadastro e recuperação de senha do EduGames foi desenvolvido para ilustrar a tela inicial de acesso do usuário à plataforma. Ele apresenta uma estrutura simples e intuitiva, dividida em duas seções principais: uma para efetuar o login, com campos de e-mail e senha e botão de entrada, e outra para realizar o cadastro, com campos semelhantes e botão de registro. Há também áreas reservadas para imagens ilustrativas que tornam a interface mais acolhedora e visualmente agradável. O objetivo desse wireframe foi validar o layout e o fluxo de autenticação do usuário, garantindo clareza, usabilidade e coerência com o restante do site.

5.1.2 Wireframe da tela da seção do jogo e avaliações e comentários sobre o jogo



O wireframe da tela de avaliações e comentários do EduGames foi criado para representar a forma como o usuário interage com as avaliações dos jogos dentro da plataforma. Ele demonstra a disposição dos elementos principais — imagem e descrição do jogo, média de avaliações, campo para inserir nota e comentário, além dos feedbacks de outros usuários — com foco em clareza, acessibilidade e coerência visual. O objetivo foi validar o fluxo de interação, garantindo uma experiência intuitiva e responsiva alinhada às heurísticas de Nielsen e aos princípios de design para melhor interação do usuário.

5.1.3 Wireframes para tela de início da aplicação

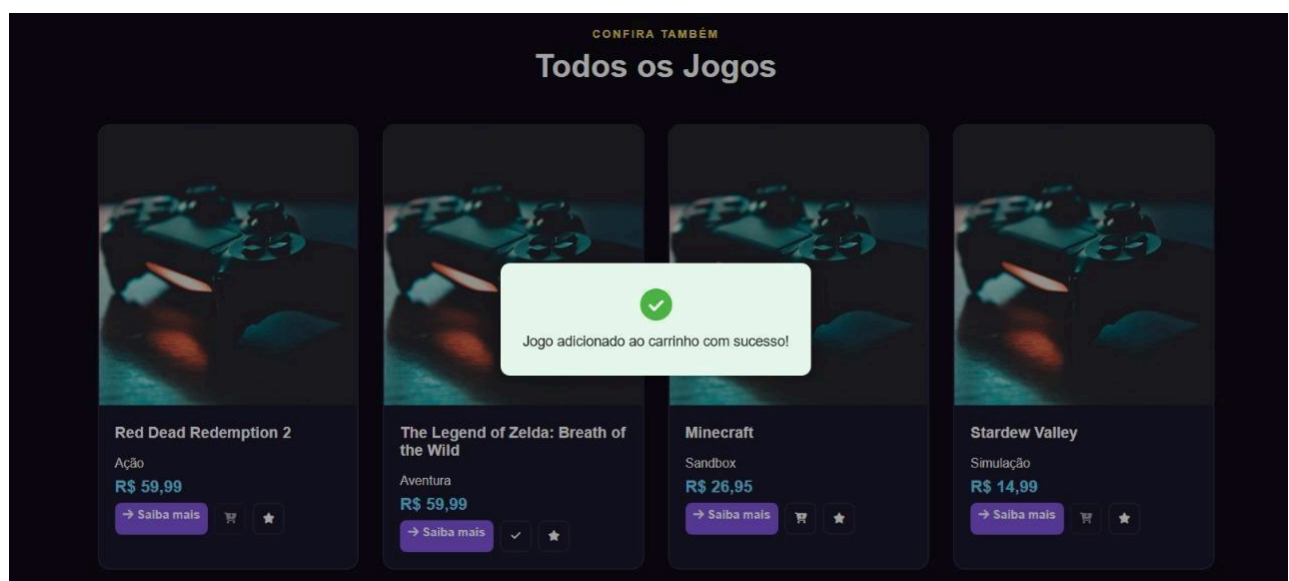


O wireframe da tela inicial da aplicação EduGames foi criado para representar a página principal que o usuário visualiza após o login. Ele apresenta uma estrutura organizada e intuitiva, com barra de navegação superior contendo o logotipo, campo de busca, botão de login/cadastro e ícone de carrinho, facilitando o acesso rápido às principais funções. Logo abaixo, há um carrossel de imagens que destaca promoções e jogos em destaque, seguido por seções de lançamentos e destaques, onde cada jogo é exibido em um card com imagem, nome, preço e botão de compra. No rodapé, estão os links de suporte, contato e redes sociais, reforçando a credibilidade e acessibilidade do site. O objetivo desse wireframe foi planejar o layout e o fluxo de navegação da página inicial, priorizando a usabilidade, clareza visual e responsividade, de acordo com os princípios do design centrado no usuário aplicados no projeto EduGames.

Mapeamento das 10 heurísticas de Nielsen no site (onde já aplicar/sugestões práticas)

6.1 Visibilidade do status do sistema

Para assegurar a visibilidade do status do sistema, foram implementados mecanismos de alerta baseados em componentes do tipo *toast*, exibidos automaticamente a cada ação relevante do usuário. Esses alertas possuem duração de 1,5 segundo e fornecem feedback imediato sobre operações como adicionar itens ao carrinho ou aos favoritos. A solução também previne cliques repetidos, garantindo que a mesma ação não seja disparada de forma duplicada durante o período de exibição do alerta. Essa estratégia mantém o usuário continuamente informado sobre o estado atual do sistema e evita comportamentos indevidos.



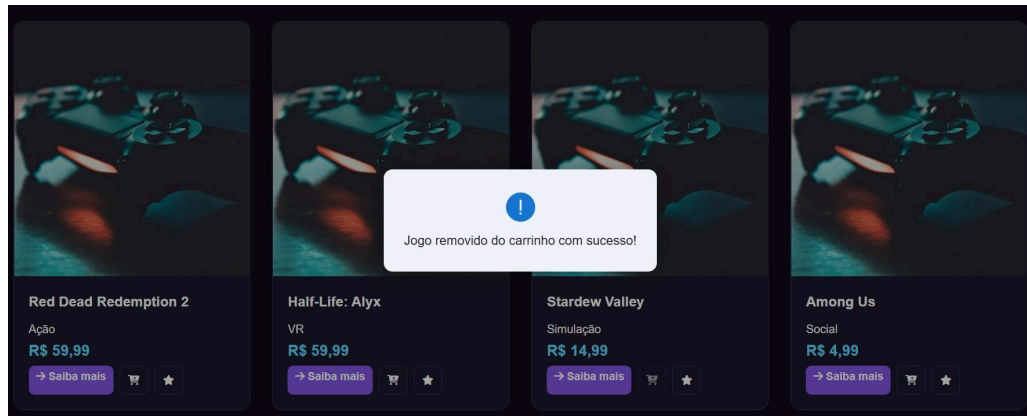
6.2 Correspondência entre o sistema e o mundo real

A interface foi padronizada para utilizar linguagem simples, direta e alinhada às convenções conhecidas pelo usuário. Textos de botões, rótulos, cores de elementos interativos e iconografia foram ajustados para refletir padrões amplamente reconhecíveis, como o uso de ícones de carrinho para ações de compra. As informações monetárias são apresentadas em formato padronizado (R\$ + duas casas decimais), reforçando a correspondência com práticas do mundo real. A uniformidade no tamanho, cor e hierarquia dos elementos visuais facilita a compreensão e reduz a carga cognitiva necessária para interpretar a tela.



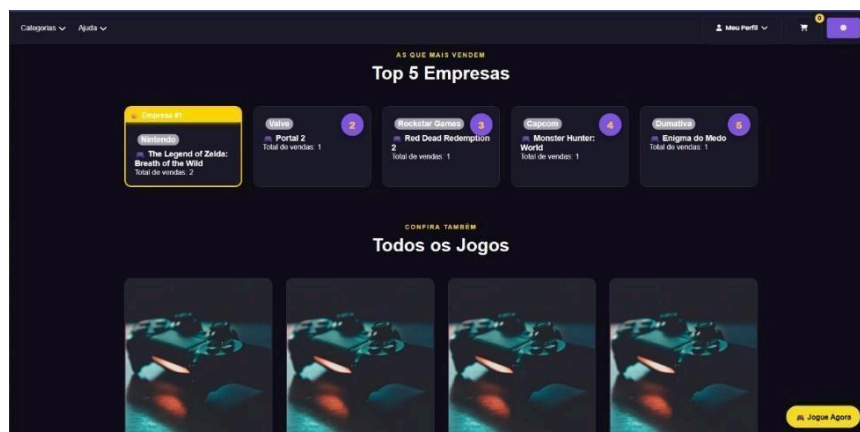
6.3 Controle e liberdade do usuário

O sistema foi projetado para permitir o cancelamento imediato de ações, garantindo maior autonomia ao usuário. Sempre que um item é adicionado ao carrinho ou à lista de favoritos, o próprio botão de interação muda automaticamente sua funcionalidade para remover o item, sem a necessidade de recarregar a página ou acessar uma tela distinta. Essa alternância instantânea entre adicionar e remover caracteriza um mecanismo eficiente de controle reversível, reduzindo frustrações e facilitando correções rápidas.



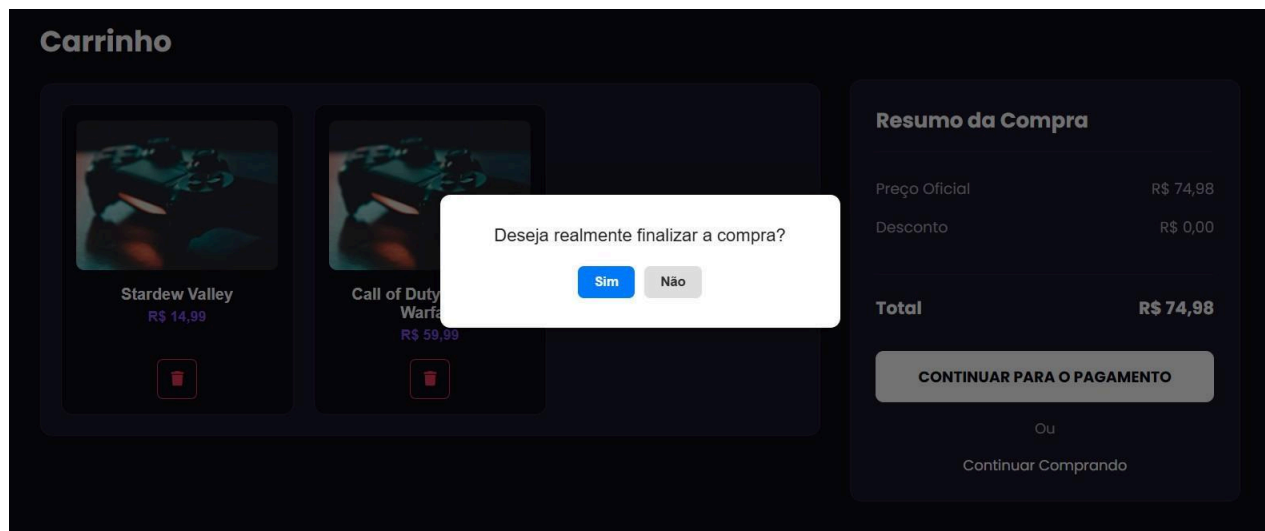
6.4 Consistência e padronização

A consistência da interface foi assegurada pela unificação de classes, labels e cores para elementos com a mesma função. Todo botão responsável pela ação de compra utiliza a mesma classe `.btn-comprar`, garantindo uniformidade visual em todas as instâncias. A posição do preço e do botão dentro dos cards segue o padrão definido em `.preco-comprar`, reforçando previsibilidade e familiaridade. A padronização contribui para o entendimento imediato das ações disponíveis e facilita a navegação em diferentes seções do sistema.



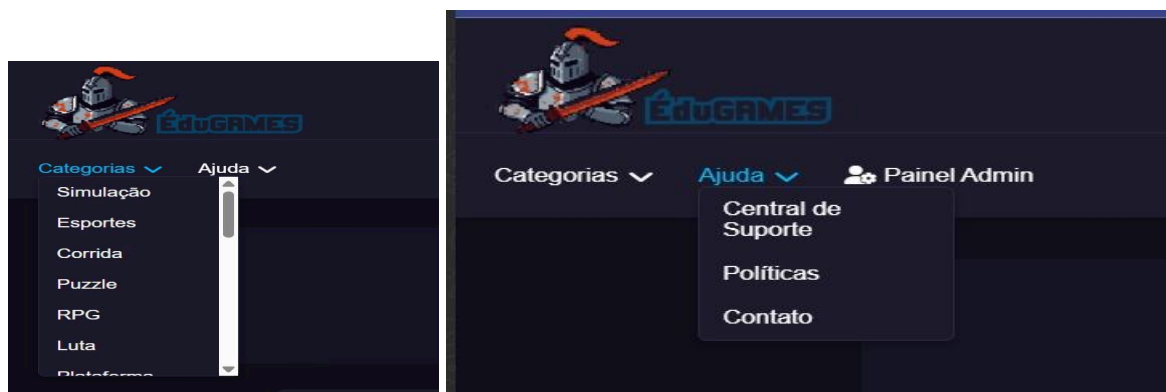
6.5 Prevenção de erros

Diversos mecanismos de prevenção ativa de erros foram aplicados. Botões são automaticamente desabilitados quando a ação correspondente está em execução, evitando comandos repetidos. Ações destrutivas, como remoção de itens, exigem confirmação para evitar operações indesejadas. Além disso, elementos já adicionados ao carrinho ou aos favoritos passam a exibir ícones e cores diferenciadas (ex.: carrinho vermelho), informando visualmente que a ação já foi concluída. O sistema ainda impede redirecionamentos quando dados obrigatórios não foram preenchidos, garantindo que o fluxo de interação permaneça consistente e seguro.



6.6 Reconhecimento em vez de memorização

A interface foi reorganizada de modo a reduzir a necessidade de memorização por parte do usuário. A barra de navegação principal tornou-se *sticky*, permanecendo fixa no topo da tela durante toda a rolagem. Tal navbar contém os principais menus e um filtro de categorias, permitindo acesso constante às funções essenciais sem que o usuário precise retornar ao início da página. A visibilidade persistente dos elementos de navegação facilita a localização de conteúdos e reduz a carga cognitiva associada à lembrança de caminhos internos.



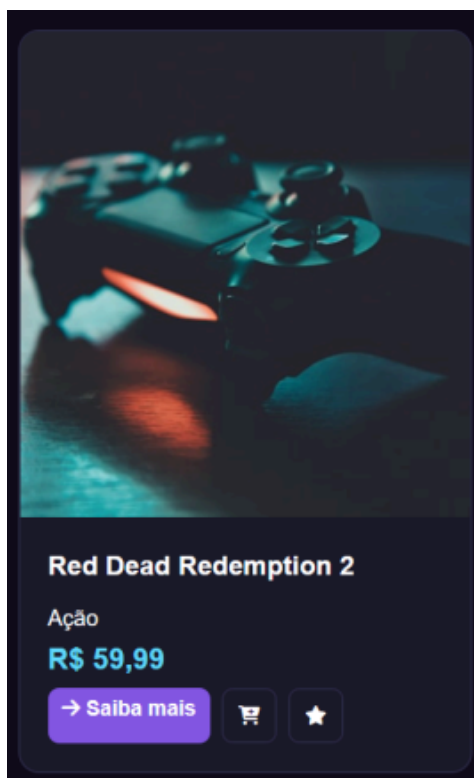
6.7 Flexibilidade e eficiência de uso

A interface foi projetada para oferecer eficiência tanto a usuários iniciantes quanto a usuários experientes. Para iniciantes, os botões de interação permanecem claramente visíveis nos cards, com ações explícitas como “Saiba mais”, “Adicionar ao carrinho” e “Adicionar aos favoritos”. Esses comandos estão posicionados logo abaixo do título e do preço, facilitando a compreensão de que são elementos clicáveis e acionáveis.

Para usuários experientes, foram implementados mecanismos de navegação acelerada. Além do carrossel de jogos mais vendidos, que funciona como um atalho visual direto aos itens de maior relevância, o próprio **card do jogo agora possui comportamento inteligente de redirecionamento**. A imagem principal do jogo se tornou clicável e leva diretamente à página de detalhes, reproduzindo o padrão amplamente conhecido de plataformas digitais, onde usuários habituados esperam que a imagem funcione como um link.

Dessa forma, o sistema permite duas formas distintas de acessar a página do jogo:

1. **Via botão “Saiba mais”**, recomendado para usuários iniciantes e mais explícito.
2. **Via clique direto na imagem**, atalho preferido por usuários experientes, que já têm familiaridade com o padrão de navegação em cards de produtos.



6.8 Design estético e minimalista

A interface segue princípios de design minimalista, apresentando somente informações essenciais em cada elemento da tela. Nos cards principais, destacam-se apenas o nome do jogo, o preço e a indicação de estado da ação (já adquirido ou disponível para compra).

Informações secundárias ou redundantes foram suprimidas para evitar sobrecarga visual. Essa abordagem privilegia clareza e objetividade, direcionando a atenção do usuário para os elementos de maior relevância.



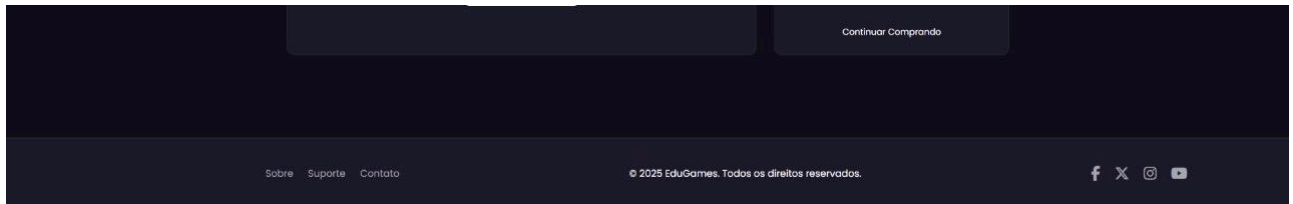
6.9 Ajude os usuários a reconhecer, diagnosticar e recuperar erros

Mensagens de erro foram implementadas de forma clara e objetiva, orientando o usuário quanto às possíveis causas e soluções. Quando um filtro aplicado não retorna resultados, o sistema exibe mensagem informativa indicando a ausência de itens correspondentes e sugerindo que o usuário ajuste ou altere os filtros. Essa comunicação direta permite diagnóstico imediato e fornece instruções para recuperação da tarefa sem interrupções no fluxo de navegação.



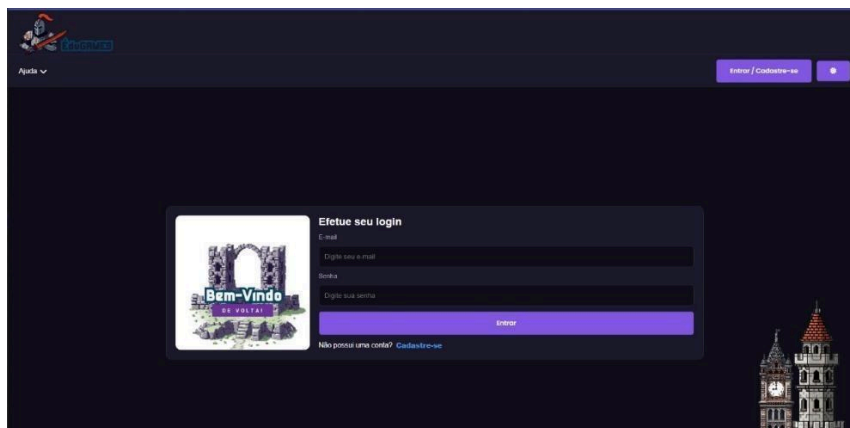
6.10 Ajuda e documentação

Todas as telas incluem no rodapé links de ajuda, redes sociais e contato institucional. Esses elementos garantem que o usuário tenha acesso contínuo a canais de suporte e comunicação, possibilitando esclarecimento de dúvidas ou obtenção de assistência sempre que necessário. O posicionamento fixo no *footer* mantém esses recursos acessíveis independentemente da seção navegada. (*Em implementação*)

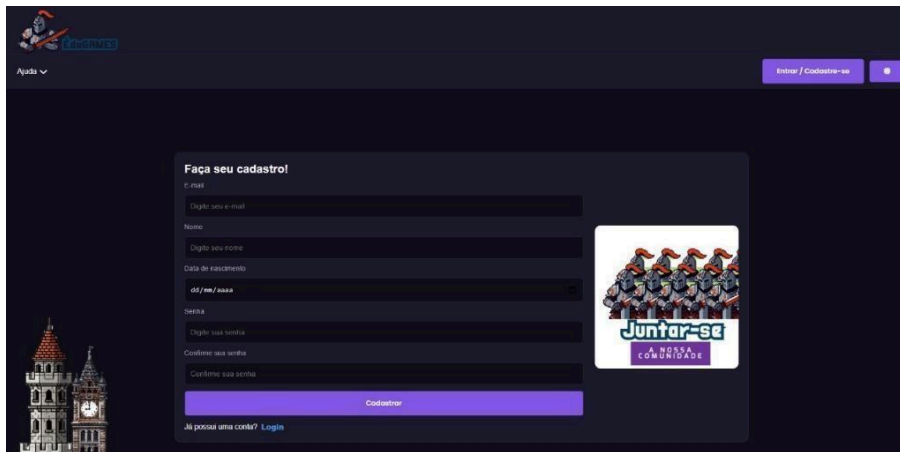


Prototipação

Página de login e cadastro



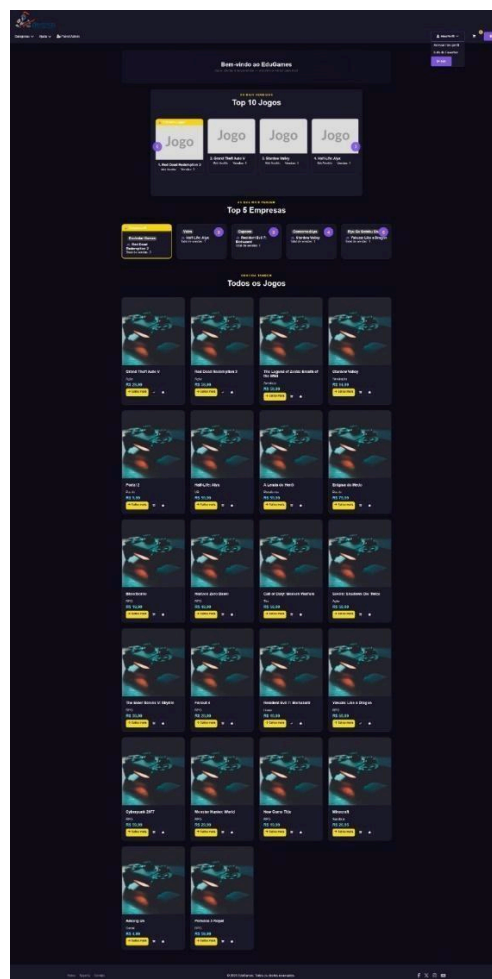
Relatório – Aplicação Web Edugames



Esta tela inicial permite que o usuário acesse a plataforma por meio de login com e-mail e senha, ou realize seu cadastro. A interface é minimalista e utiliza ícones da biblioteca **Font Awesome**, melhorando a comunicação visual.

O uso de fontes importadas do **Google Fonts** reforça a identidade visual e garante uma leitura confortável.

Página Inicial - Catálogo de jogos



Relatório – Aplicação Web Edugames

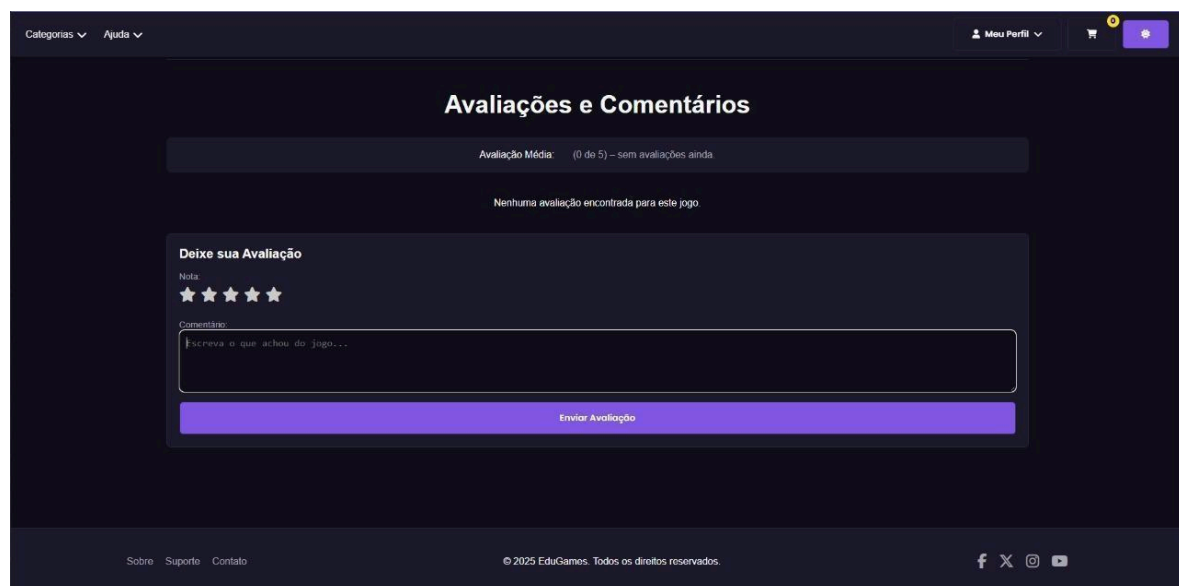
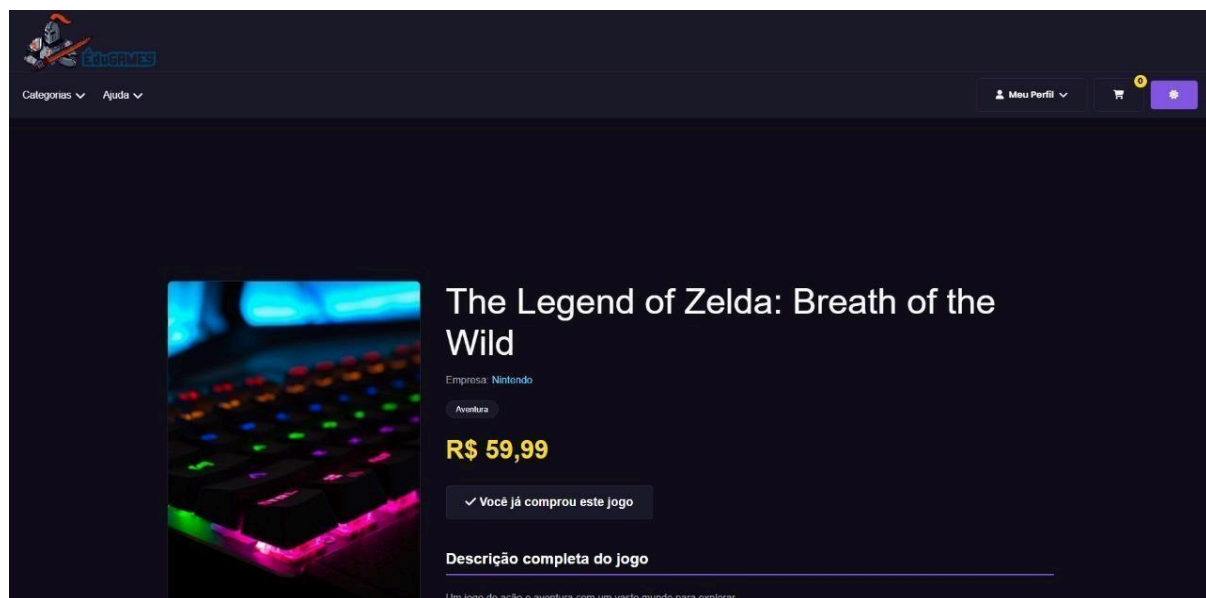
Após o login, o usuário tem acesso à página principal, onde é apresentado o catálogo de jogos digitais. Os jogos são exibidos em **cards interativos**, contendo imagem, nome, preço e botão de compra.

A interface apresenta seções como:

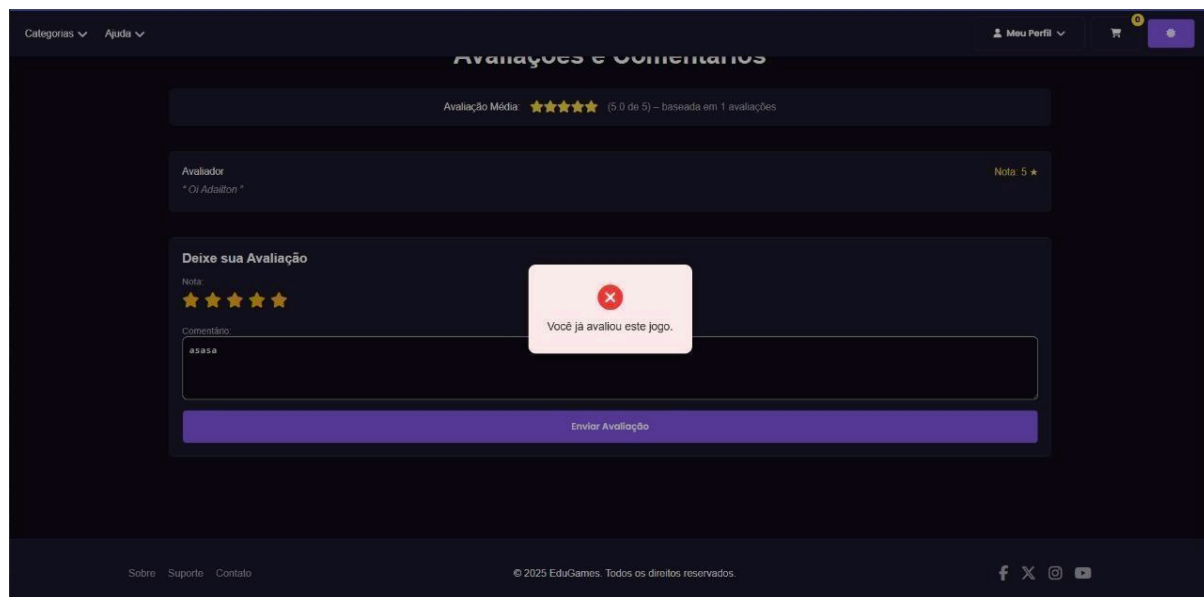
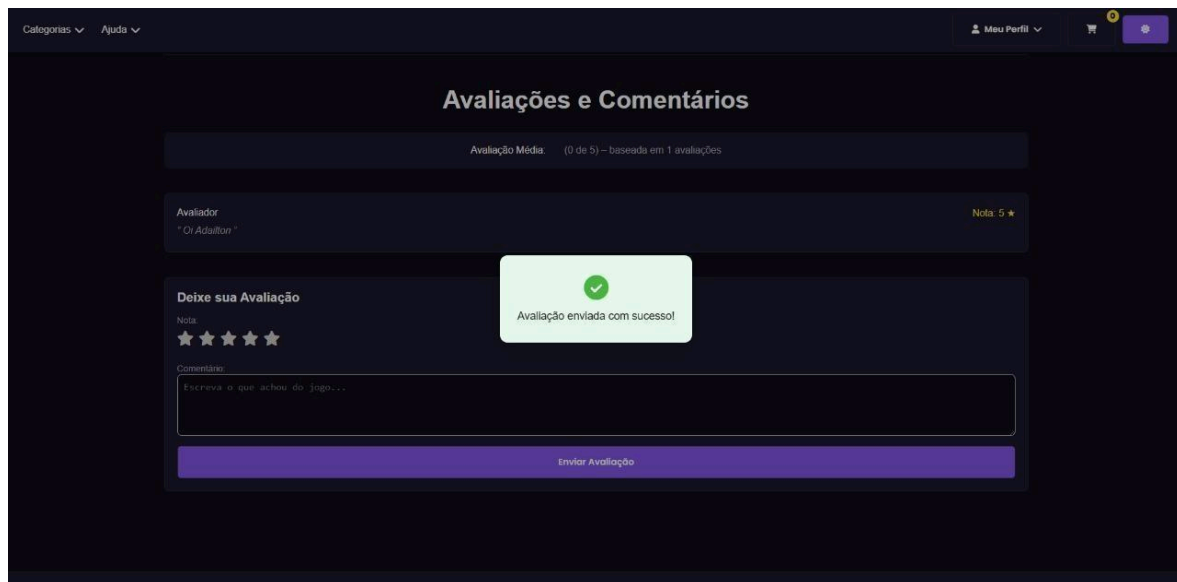
- Top 10 Jogos
- Top 5 empresas
- Todos os Jogos

Há também barra de busca e filtros dinâmicos que permitem localizar jogos por gênero, plataforma e promoção. O design é **responsivo**, adaptando-se automaticamente a telas de diferentes tamanhos.

Página de Avaliações



Relatório – Aplicação Web Edugames



Esta página permite que os usuários atribuam notas e escrevam comentários sobre os jogos adquiridos. O sistema de estrelas de avaliação oferece retorno visual imediato. O comentário uma vez realizado não pode ser realizado novamente, sinalizando mensagem de erro/aviso, uma vez que tal comando tenta ser novamente executado. Todos os elementos seguem padrões de acessibilidade, com contraste adequado e mensagens de feedback claras.

Escolha do modo claro/escuro

A escolha de incluir a opção de modo claro e escuro no projeto EduGames foi feita para melhorar a experiência do usuário e a acessibilidade visual da plataforma. Essa funcionalidade permite que o usuário personalize a interface de acordo com suas preferências ou necessidades, reduzindo o cansaço visual em ambientes com diferentes níveis de iluminação. O modo escuro é especialmente útil para usuários que acessam o site à noite ou em locais com

pouca luz, pois diminui o brilho da tela e torna a leitura mais confortável. Já o modo claro é mais adequado para ambientes bem iluminados, garantindo melhor visibilidade do conteúdo. Além disso, a alternância entre os modos contribui para a inclusão e o conforto de pessoas com sensibilidades visuais, além de seguir uma tendência moderna de design adaptativo presente nas principais plataformas digitais.

Em síntese, essa escolha reforça o compromisso do EduGames com a usabilidade, acessibilidade e personalização da experiência do usuário, valores centrais do projeto.

Jogos interativos gratuitos

A disponibilização de jogos interativos gratuitos dentro da EduGames, representa uma inovação significativa ao transformar a experiência de navegação em algo mais imersivo, envolvente e estratégico. Em vez de seguir o modelo tradicional de vitrine estática, a EduGames passa a oferecer ao usuário a oportunidade de experimentar mecânicas, estilos visuais e narrativas antes mesmo de considerar uma compra, criando valor imediato sem custo. Essa abordagem reduz barreiras de entrada, aumenta a confiança do consumidor e fortalece o engajamento, já que o usuário permanece mais tempo explorando o ecossistema da plataforma. Além disso, funciona como uma “degustação digital”, semelhante a um trial inteligente, estimulando decisões de compra mais conscientes e ampliando a percepção de qualidade da marca, que demonstra transparência, acessibilidade e foco na experiência do jogador. Dessa forma, a EduGames não apenas eleva o nível de usabilidade e entretenimento, mas também constrói um diferencial competitivo sólido no mercado de jogos online.

Escolha da fonte/design

A escolha da fonte e do design no projeto EduGames foi guiada pelos princípios de usabilidade, legibilidade e coerência visual. Optou-se por fontes simples e modernas, importadas do Google Fonts, para garantir uma leitura fluida e agradável em diferentes dispositivos e tamanhos de tela. Esse cuidado melhora a acessibilidade e reduz a fadiga visual, principalmente em interfaces com grande volume de texto, como descrições de jogos e comentários de usuários.

No design, priorizou-se um estilo limpo e minimalista, com cores equilibradas e elementos bem espaçados, facilitando a navegação e destacando o conteúdo principal. Essa escolha reforça a identidade visual moderna e inclusiva do EduGames, mantendo harmonia com a proposta de um ambiente digital intuitivo e voltado à melhor experiência do usuário.

A escolha de um tema mais medieval para o design do EduGames foi pensada para criar uma identidade visual envolvente e temática, remetendo ao universo de muitos jogos populares que exploram esse estilo. Elementos como tonalidades terrosas, tipografia com traços mais robustos e ícones inspirados em símbolos antigos ajudam a transmitir uma atmosfera de aventura e fantasia, aproximando o usuário do contexto dos jogos disponíveis na plataforma. Essa estética não apenas torna o site mais atrativo visualmente, mas também reforça a imersão e a coerência temática entre o conteúdo e a experiência oferecida. Além disso, o tema medieval contribui para diferenciar o EduGames de outras plataformas de e-commerce de jogos, unindo criatividade, identidade e usabilidade em um mesmo conceito visual.

Implementação do BackEnd

1. Introdução

A aplicação Edu-Games consiste em um sistema web desenvolvido com tecnologias modernas de front-end e integrado a uma API REST externa responsável pelo processamento de dados, vendas e gerenciamento das informações dos jogos. A arquitetura segue o padrão cliente-servidor, no qual o frontend solicita dados ao backend por meio de requisições HTTP, recebendo respostas estruturadas em formato JSON.

O sistema permite listar jogos, filtrar por categorias, adicionar itens ao carrinho, favoritar produtos, visualizar detalhes, registrar avaliações, consultar rankings e acessar funcionalidades administrativas.

2. Tecnologias Utilizadas

2.1 React

React é uma biblioteca JavaScript voltada para a construção de interfaces de usuário, baseada no conceito de componentes reutilizáveis. Ela utiliza o paradigma declarativo, permitindo que o desenvolvedor descreva como a interface deve se comportar de acordo com o estado da aplicação. Sua arquitetura segue o modelo de árvore de componentes e utiliza o Virtual DOM para otimizar atualizações na interface.

2.2 Vite

Vite é uma ferramenta de build e servidor de desenvolvimento que proporciona inicialização rápida do projeto e recarregamento em tempo real. Sua principal característica é o uso de módulos ES nativos do navegador, evitando etapas de empacotamento durante o desenvolvimento e acelerando o processo de compilação.

2.3 Axios

Axios é uma biblioteca utilizada para realizar requisições HTTP. Ela simplifica a comunicação entre o frontend e o backend, abstraindo métodos como GET, POST, PUT e DELETE. Axios trabalha de forma assíncrona, permitindo manipulação de respostas e tratamento de erros de maneira organizada.

2.4 React Router

React Router é uma biblioteca que gerencia a navegação entre as páginas da aplicação sem recarregamento total do site (Single Page Application). Através de rotas definidas, permite exibir diferentes componentes conforme o caminho acessado pelo usuário.

2.5 CSS Modules

CSS Modules é uma técnica que encapsula estilos em escopos específicos, evitando conflitos de classes. Cada arquivo CSS gera identificadores únicos, garantindo isolamento e mantendo a

organização dos estilos.

2.6 LocalStorage

O LocalStorage é um mecanismo de armazenamento local no navegador que permite salvar dados de forma persistente no dispositivo do usuário. No sistema, é utilizado para manter sessões, carrinhos e informações de autenticação mesmo após recarregamentos da página.

2.7 JavaScript ES2023

A aplicação utiliza recursos modernos do JavaScript, como funções assíncronas, módulos nativos, desestruturação, arrow functions e operadores lógicos aprimorados, garantindo maior legibilidade e eficiência no código.

2.8 API REST Externa

A API REST utilizada pelo sistema é responsável por fornecer os dados consumidos pelo frontend. Ela segue princípios de Representational State Transfer, retornando dados em JSON e utilizando métodos HTTP padronizados.

3. Conceito e Funcionamento das Requisições HTTP REST

A arquitetura REST utiliza métodos HTTP para representar ações sobre recursos. Cada método possui um propósito específico:

GET: Recupera informações de um recurso. Ex.: listar jogos.

POST: Envia dados para criação de um novo recurso. Ex.: registrar avaliação.

PUT: Atualiza um recurso existente.

DELETE: Remove um recurso do servidor.

A comunicação ocorre por meio de requisições enviadas pelo cliente (frontend) e respostas devolvidas pelo servidor (backend). Geralmente, o backend retorna respostas no formato JSON, contendo dados, status e mensagens de erro ou sucesso.

No Edu-Games, a biblioteca Axios é responsável por realizar essas operações.

4. Estrutura do Projeto Frontend

A estrutura do diretório principal segue um padrão modularizado para facilitar manutenção e escalabilidade:

Relatório – Aplicação Web Edugames

```
src/
├── api/           # Serviços de comunicação com a API
│   ├── authService.jsx
│   ├── carrinho.jsx
│   ├── empresa.jsx
│   ├── favoritos.jsx
│   ├── jogos.jsx
│   └── vendas.jsx
├── assets/       # Imagens, logos e ícones
├── components/   # Componentes reutilizáveis
│   ├── Footer/
│   └── Header/
├── pages/        # Páginas renderizadas
│   ├── Admin/
│   ├── Cart/
│   ├── Favoritos/
│   └── GameInfo/
├── utils/        # Funções auxiliares
├── App.jsx
├── main.jsx
└── index.css
```

Cada arquivo dentro da pasta /api é responsável por abstrair uma parte da comunicação com o backend, mantendo o código organizado e separando as responsabilidades da interface.

5. Processo de Execução do Projeto

Para executar a aplicação em ambiente local, o procedimento recomendado é:

Clonar o repositório:

```
git clone https://github.com/luis-a-silva/A3UCUsabilidadeReact.git
```

```
cd A3UCUsabilidadeReact
```

Instalar

dependências: npm

```
install
```

Executar o servidor de desenvolvimento:

npm run dev

A aplicação fica disponível em:

http://localhost:5173

6. Comunicação entre Frontend e Backend

A comunicação ocorre por meio de serviços definidos na pasta /api, que utilizam Axios para consumir os endpoints do servidor. Um exemplo típico é o método responsável por obter todos os jogos:

```
export async function getAllJogos() {  
  const res = await axios.get(`${API_URL}/jogos`);  
  return res.data.dados;  
}
```

O fluxo é composto pelas seguintes etapas:

O frontend envia uma requisição GET ao backend.

O servidor processa a solicitação e consulta o banco de dados.

A API retorna um objeto JSON contendo:

dados solicitados

mensagem de status

código de resposta

O frontend recebe os dados, atualiza o estado do componente e renderiza as informações na interface.

Esse modelo garante desacoplamento, escalabilidade e facilidade de manutenção.

Documentação das Rotas e Telas do Módulo Administrativo – Edu-Games

1. Introdução

O painel administrativo do sistema *Edu-Games* foi desenvolvido para permitir que

usuários com perfil de administrador realizem ações de gestão relacionadas a usuários, empresas, categorias, jogos e monitoramento de vendas. O acesso às funcionalidades administrativas é restrito e baseado em autenticação via token JWT (JSON Web Token), armazenado no navegador e decodificado pelo frontend para validação do perfil do usuário.

2. Autenticação e Autorização do Administrador

2.1 Obtenção e Armazenamento do Token

Após o login bem-sucedido, o backend retorna ao frontend um token JWT, que é salvo no localStorage sob a chave "token". Esse token contém informações essenciais para a autorização, incluindo:

- id: identificador do usuário.
- nome: nome completo do usuário autenticado.
- perfil: perfil de acesso (Administrador ou Usuário).
- iat: data de emissão (*issued at*).
- exp: data de expiração (*expiration time*).

Exemplo de token utilizado:

```
{
  "alg": "HS256",
  "typ": "JWT"
}.
{
  "id": 1,
  "nome": "Admin",
  "perfil":
  "Administrador", "iat":
  1764005403,
  "exp": 1764009003
}.
[Signature]
```

2.2 Decodificação do Token

A decodificação é realizada no frontend utilizando a biblioteca jwt-decode. A função getDecodedToken() é responsável por recuperar o token armazenado e convertê-lo para um objeto JavaScript legível:

```
import { jwtDecode } from "jwt-decode";
```

```
export function getDecodedToken() {  
  const token = getToken("token");  
  if (!token) return null;  
  
  try {  
    return jwtDecode(token);  
  } catch (err) {  
    console.error("Erro ao decodificar token:", err);  
    return null;  
  }  
}
```

Caso o token seja inválido ou esteja ausente, a função retorna null.

2.3 Verificação do Perfil de Administrador

A autorização é realizada pela função `isAdmin()`, que verifica se o campo "perfil" do token corresponde ao valor "Administrador":

```
export function isAdmin() {  
  const decoded = getDecodedToken();  
  return decoded?.perfil === "Administrador";  
}
```

Se a condição for verdadeira, o usuário é autorizado a acessar:

- O atalho principal do painel administrativo.
- As rotas e telas exclusivas de administração.

Essa validação é aplicada tanto no componente visual quanto nas rotas protegidas.

3. Proteção de Rotas – AdminRoute

O componente `AdminRoute` encapsula a lógica de segurança, impedindo que usuários não autorizados acessem páginas administrativas. O comportamento geral é:

1. Verificar se o token existe.
2. Decodificar o token.
3. Validar se `perfil === "Administrador"`.

4. Se verdadeiro → renderizar a página de administração.
5. Se falso → redirecionar o usuário para a página padrão ou de erro.

Esse mecanismo garante que o conteúdo administrativo não seja exibido a usuários comuns, mesmo que tentem acessar a rota pela URL.

4. Módulo Administrativo – Telas e Funcionalidades

A área administrativa é composta por quatro módulos principais, cada um contendo páginas específicas e operações permitidas apenas para administradores.

4.1 Tela de Usuários

4.1.1 Listagem de Usuários

Exibe todos os usuários cadastrados no sistema, incluindo seus perfis, nomes e identificadores.

Permite também acessar ações de edição.

4.1.2 Criação de Usuário (Create)

Tela destinada ao cadastro de novos usuários com os seguintes campos:

- Nome
- E-mail
- Senha
- Data de Nascimento

Ao confirmar a operação, um POST é enviado à API.

4.1.3 Atualização de Usuário (Update)

Permite alterar dados previamente cadastrados, enviando requisições PUT ao backend.

4.2 Dashboard Administrativo

A dashboard reúne indicadores de análise e relatórios gerenciais:

4.2.1 Jogos Mais Vendidos

Gráfico e tabela exibindo os jogos com maior volume de vendas.

4.2.2 Jogos Mais Vendidos por Empresa

Relatório que correlaciona empresas produtoras com desempenho de vendas dos seus jogos.

Os dados são obtidos via chamadas à API /vendas e /jogos.

4.3 Tela de Empresas

4.3.1 Criar Empresa

Formulário para inclusão de novas empresas de jogos.

Envia dados via POST.

4.3.2 Listar Empresa

Exibe todas as empresas cadastradas, com id, nome e ações disponíveis.

4.3.2 Atualizar Empresa

Permite modificar atributos existentes e envia PUT com os dados alterados.

4.4 Tela de Jogos (CRUD)

4.4.1 Listagem de Jogos

Exibe todos os jogos do sistema com informações como:

- Nome
- Empresa
- Categoria
- Preço
- Total de vendas

4.4.2 Criação de Jogo (Create)

Campos utilizados:

- Nome
- Descrição

Relatório – Aplicação Web Edugames

- Preço
- Ano
- Categoria
- Empresa

Ao enviar, o sistema executa um POST para /jogos.

4.4.3 Atualização de Jogo (Update)

Permite modificar atributos existentes e envia PUT com os dados alterados.

4.4.4 Remoção de Jogo (Delete)

Operação protegida que exige confirmação. Envia DELETE para o backend.

4.4.5 Visualização Detalhada (Read)

Exibe todos os atributos do jogo em formato expandido.

5. Fluxo de Acesso ao Painel Administrativo

O fluxo técnico funciona da seguinte forma:

1. O usuário efetua login.
2. O backend retorna um token JWT contendo o perfil.
3. O token é armazenado no localStorage.
4. O menu principal verifica isAdmin() e exibe o atalho para o painel.
5. Ao tentar acessar rotas administrativas, o AdminRoute valida o token.
6. Se for administrador → rota liberada.
7. Se não for administrador → usuário é redirecionado.

6. Conclusão

O módulo administrativo do *Edu-Games* foi projetado seguindo princípios de segurança, segregação de responsabilidades e boas práticas de autenticação. A verificação do token no cliente garante que apenas usuários autorizados acessem operações sensíveis como criação de jogos, edição de usuários e gerenciamento de categorias. O uso de rotas protegidas, componentes dedicados e validação clara do perfil reforça a integridade do

sistema e organiza a experiência do administrador.

Resultados

O resultado final foi uma aplicação **leve, acessível e visualmente atrativa**, capaz de simular um ambiente real de e-commerce de jogos digitais.

O sistema apresenta bom desempenho, layout responsivo e fluidez de navegação, proporcionando ao usuário uma experiência agradável e coerente com as boas práticas de desenvolvimento frontend.

Conclusão

O projeto EduGames demonstrou, na prática, a importância da integração entre usabilidade, acessibilidade e design visual no desenvolvimento de aplicações web. O uso das linguagens HTML, CSS e JavaScript, aliado às bibliotecas Font Awesome Google Fonts, possibilitou a criação de uma interface funcional, moderna e inclusiva.

A realização do projeto reforça a relevância do design centrado no usuário e da acessibilidade digital como pilares fundamentais para a construção de experiências web mais democráticas e eficientes.

Referências

- [Bem-vindo\(a\) ao Steam](#)