

# RELATÓRIO

USABILIDADE, DESENVOLVIMENTO *WEB*,  
*MOBILE* E JOGOS

Universidade Salvador – UNIFACS  
Salvador, BA  
2025

# RELATÓRIO

## USABILIDADE, DESENVOLVIMENTO *WEB*, *MOBILE* E JOGOS

### **Equipe:**

George Gebers Brizolla  
Júlio César de Jesus Barreto Souza  
Luis Eduardo da Silva Belinelli  
Marcelo Miranda Cavalcanti  
Pedro Lima Vergne  
Zanoni Campos Fernandes

### **Professores:**

Adailton de Jesus Cerqueira Junior  
Wellington Lacerda Silveira da Silva

Universidade Salvador – UNIFACS  
Salvador, BA  
2025

## 1. INTRODUÇÃO

Trata-se de relatório da segunda entrega (final) do projeto da terceira avaliação (A3), no âmbito da unidade curricular (UC) de Usabilidade, Desenvolvimento WEB, Mobile e Jogos, da Universidade Salvador (UNIFACS), segundo semestre de 2025 (2025.2), ministrada pelos professores Adailton de Jesus Cerqueira Junior e Wellinton Lacerda Silveira da Silva.

O projeto consiste em uma aplicação de loja de jogos online, desenvolvida utilizando a biblioteca de interfaces de usuário *React* (<https://react.dev/>), que irá consumir uma API externa responsável pelo *backend* e persistência dos dados. Essa aplicação seguiu os princípios de acessibilidade e usabilidade propostos na UC.

Como tarefas da segunda entrega, deve-se entregar o código fonte da aplicação, desenvolvido com a biblioteca *React* em questão, além de um relatório mostrando o progresso da equipe entre a primeira e a atual entrega, bem como informações gerais sobre os critérios de usabilidade endereçados no projeto e detalhamento da integração com o *backend*.

## 2. LINKS IMPORTANTES

### i. Github:

i.i Repositório da equipe: <https://github.com/facs-web-usabilidade/project/>

### ii. Protótipos de baixa fidelidade (*wireframe*):

#### ii.i No Figma:

<https://www.figma.com/design/HGveF5Rudnh7owBssgA0LL/Figma-Wireframe?node-id=11-2&t=feuMHXENWAVOcVoh-1>

#### ii.ii No Github: [https://github.com/facs-web-](https://github.com/facs-web-usabilidade/project/blob/dev/relatorio/Figma-Wireframe.png)

[usabilidade/project/blob/dev/relatorio/Figma-Wireframe.png](https://github.com/facs-web-usabilidade/project/blob/dev/relatorio/Figma-Wireframe.png)

### iii. Protótipos de alta fidelidade (*high*):

#### iii.i No Figma:

<https://www.figma.com/design/D1dTaLV1GChj8Gn1eboMem/Figma-High?node-id=0-1&t=feuMHXENWAVOcVoh-1>

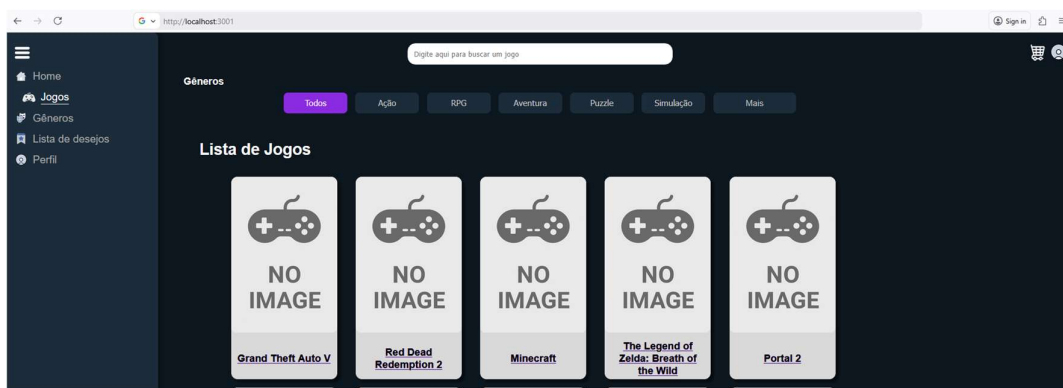
iii.ii No Github: <https://github.com/facs-web-usabilidade/project/blob/dev/relatorio/Figma-High.png>

### 3. HEURÍSTICAS DE NIELSEN UTILIZADAS

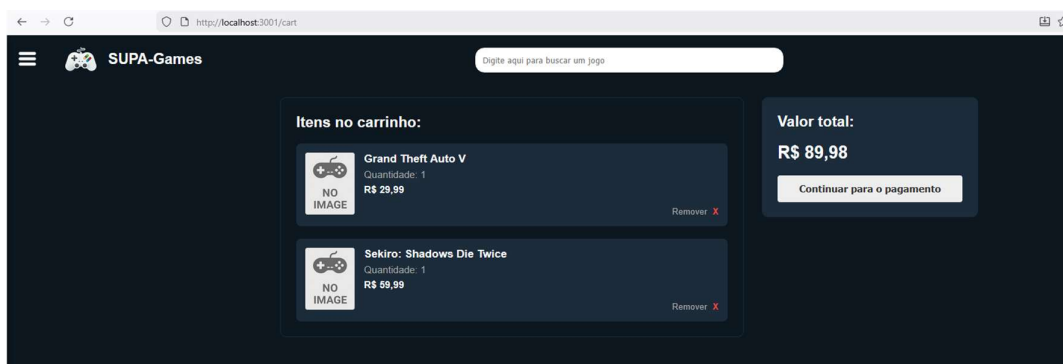
#### i. Visibilidade do Status do Sistema:

Essa heurística estabelece que o sistema deve manter o usuário constantemente informado sobre o estado atual da aplicação, fornecendo feedback adequado e em tempo hábil.

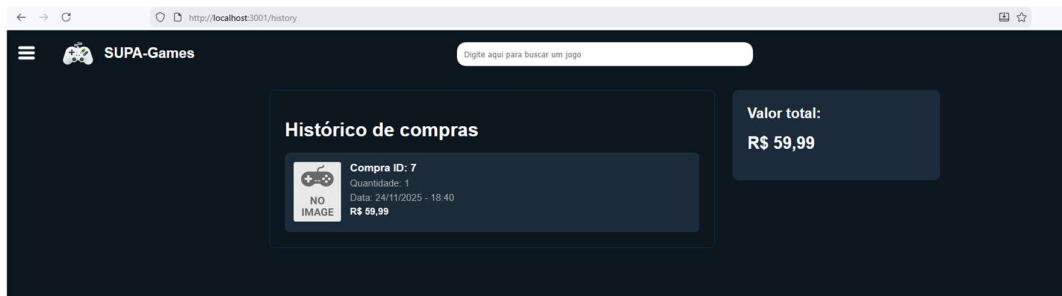
No sistema desenvolvido, essa diretriz é observada na barra de navegação, onde a página ativa é destacada visualmente por meio de realce, permitindo ao usuário compreender facilmente sua localização no ambiente digital. Tela:



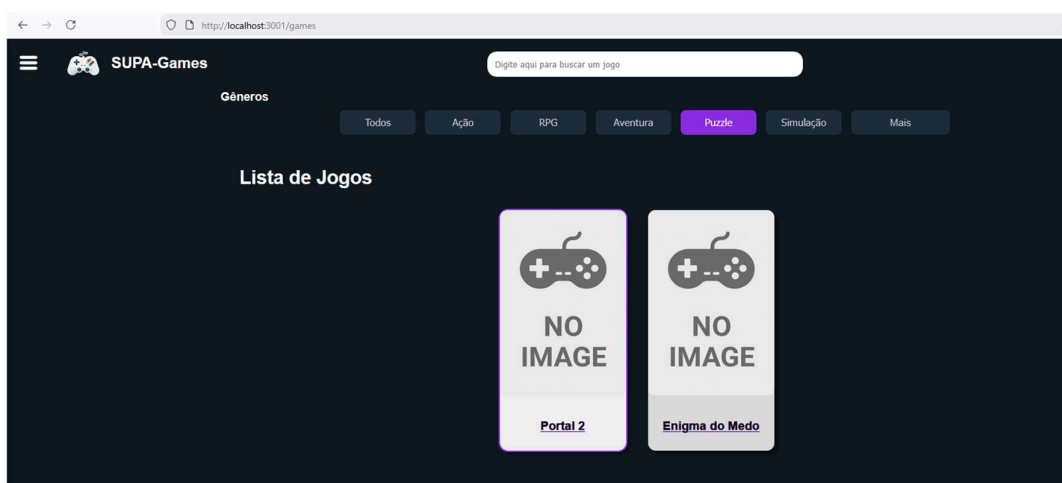
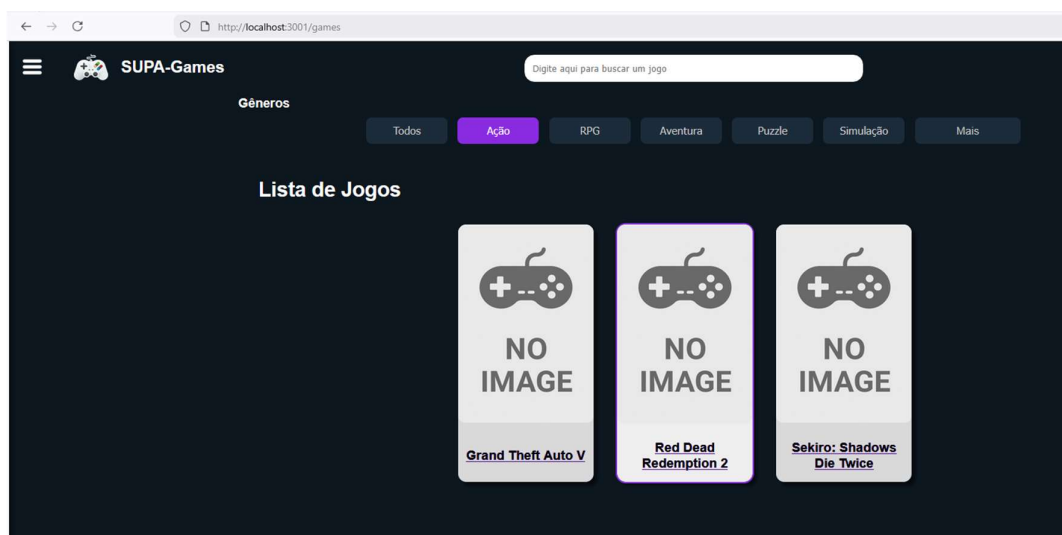
Outro exemplo é o carrinho de compras, que exibe o subtotal dos itens adicionados, garantindo que o usuário tenha ciência do progresso de sua ação e da experiência de compra. Tela:



Há ainda uma outra tela de visibilidade do status do usuário, responsável por mostrar a ele seu histórico de compras e valor total gasto, em tela que possui um padrão com a anterior: Exemplo:



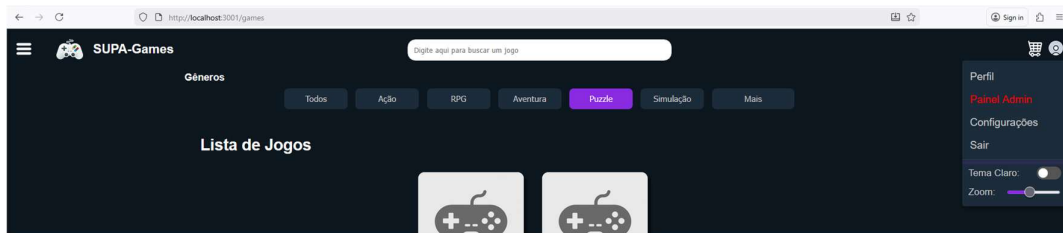
O sistema ainda mantém o usuário informado sobre a operacionalidade da aplicação, onde está sua navegação e filtros eventualmente selecionados, por meio de cursores dinâmicos e *hover* em pontos de interesse, como botões, caixas e *cards*. Telas:



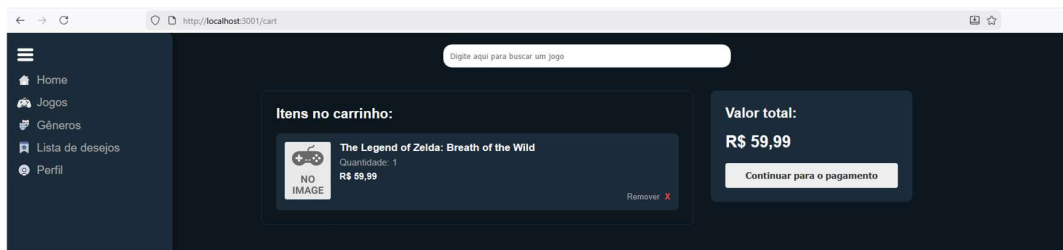
## ii. Compatibilidade entre o sistema e o mundo real:

De acordo com Nielsen, o sistema deve adotar a linguagem e os conceitos familiares ao usuário, aproximando-se da forma como ele percebe e interage com o mundo real.

Essa heurística é aplicada no projeto por meio do uso de ícones e símbolos universalmente reconhecíveis, como o ícone de carrinho para o módulo de compras, um input em branco para realização de buscas e o símbolo da área do usuário, todos posicionados em destaque e nos locais de costume de mercado. Esses elementos reduzem a necessidade de instruções adicionais, tornando a interação mais intuitiva e eficiente. Tela:



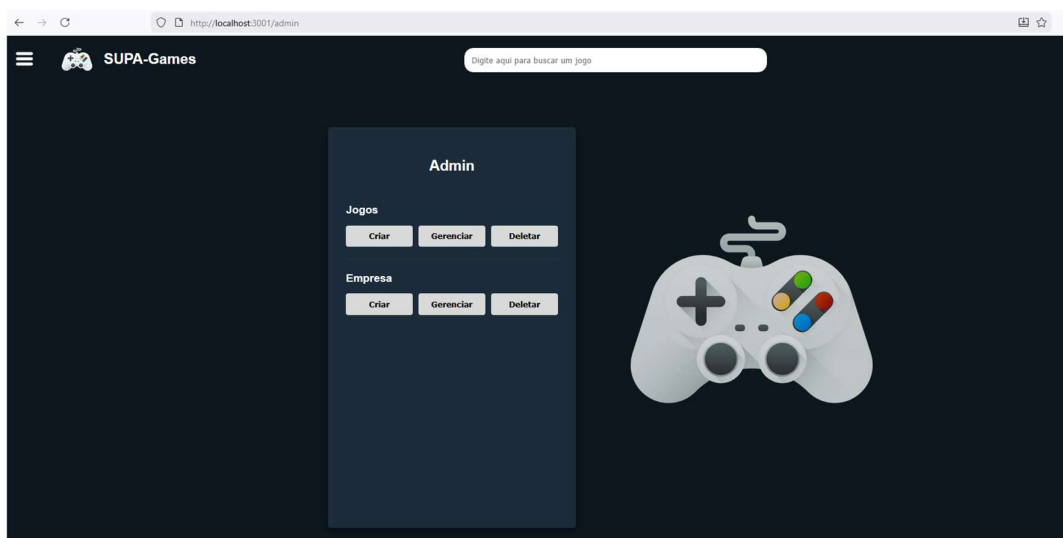
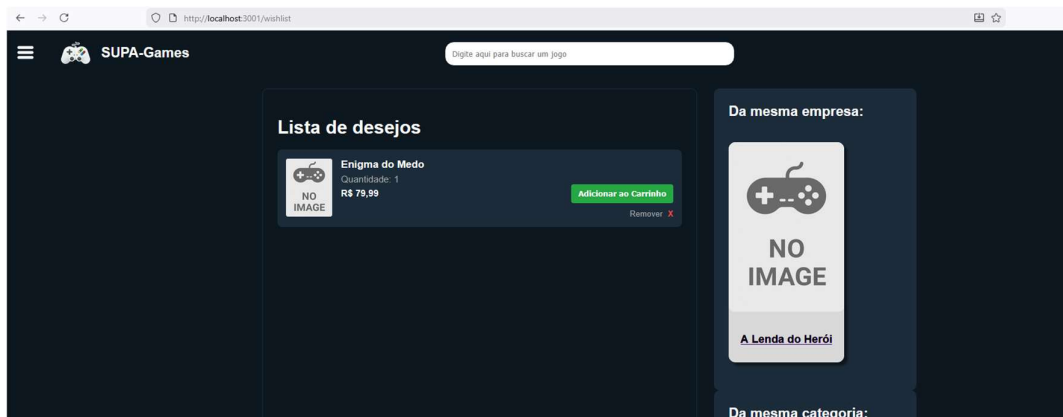
Outro exemplo é a utilização do linguajar comum do domínio, como “jogos”, “gêneros”, “lista de desejo” (“*wishlist*”), “ofertas”, “avaliados”, “carrinho”, enfim. Vide tela de exemplo:



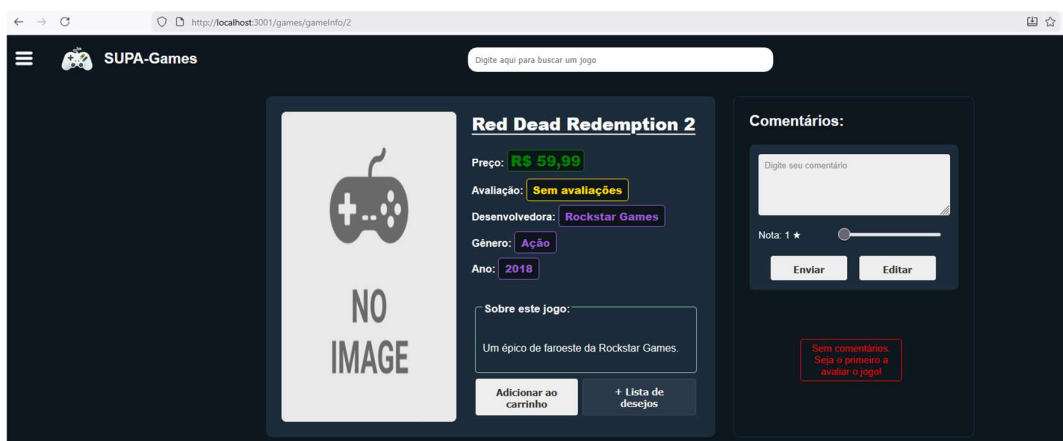
### iii. Controle e liberdade para o usuário:

O sistema deve permitir que o usuário desfça ações e recupere-se facilmente, transmitindo segurança durante a utilização.

No sistema tal princípio é garantido pelos botões de remoção e cancelamento disponíveis em diferentes seções, como o carrinho, a lista de desejos e a área administrativa. Essa abordagem garante autonomia e flexibilidade, minimizando frustrações decorrentes de ações não intencionais. Tela da página de desejos (a página do carrinho já foi apresentada acima) e do controle do admin:

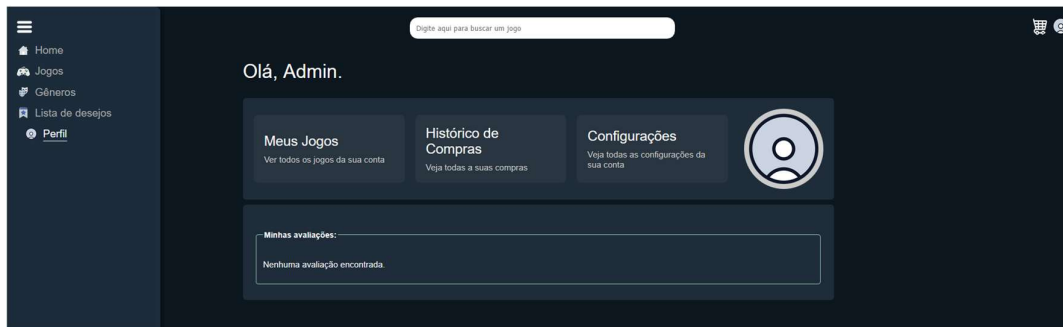


Essa liberdade ainda é garantida na tela de um jogo específico, em que o usuário pode adicionar o jogo na lista de desejos ou, caso prefira, mandar diretamente para o carrinho:



Da mesma forma, outras funcionalidades garantem autonomia ao usuário, como navegabilidade facilitada na aplicação - seja pelo menu *hamburger* no canto esquerdo

superior, ou nos ícones de ir ao carrinho de compras e ao perfil do usuário no canto direito superior, ambos disponíveis em todas as telas da aplicação:

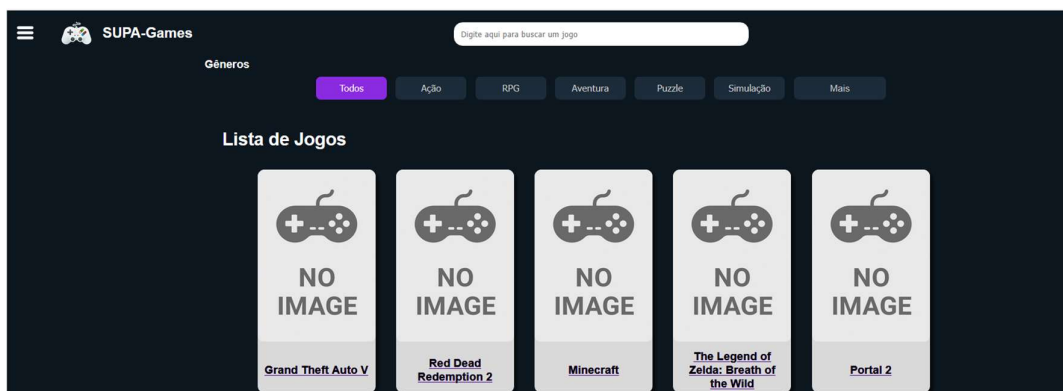


#### iv. Consistência e Padronização:

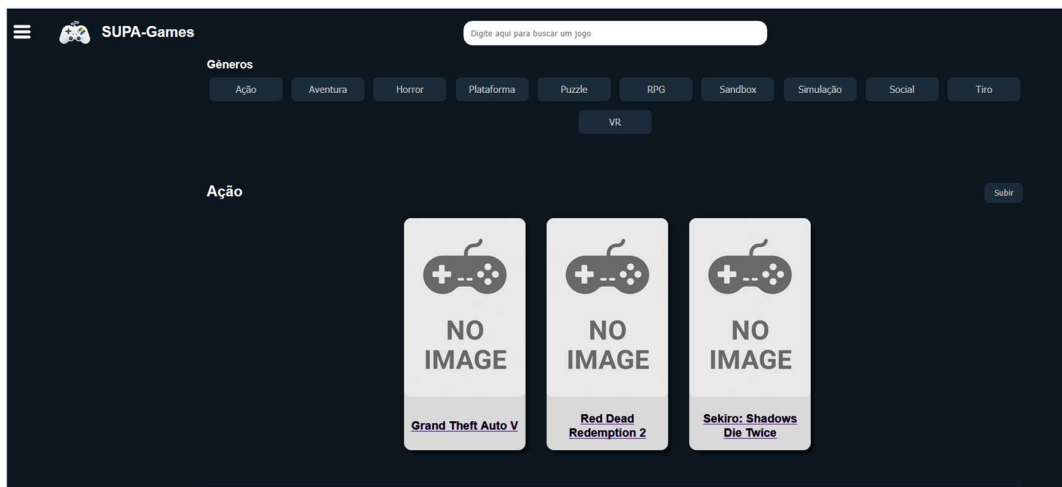
A consistência é essencial para evitar que o usuário precise reaprender padrões de interação a cada nova tela. O sistema apresenta uniformidade na disposição dos elementos, mantendo o mesmo padrão de cores, botões, tipografia e posicionamento em todas as interfaces. Há também reutilização de componentes ao longo do projeto, como caixas de conteúdo e pesquisa, botões, *cards* de objetos (jogos), enfim.

Essa padronização contribui para a previsibilidade das ações e reduz a carga cognitiva do usuário durante a navegação. O layout mantém o tema ao longo das páginas (originalmente escuro, mas com opções de tema claro), fontes, margens e estilo de botões igualmente em todas as páginas.

Ademais, a aplicação segue padrões introduzidos e geralmente aceitos em outras soluções similares de mercado. Telas em exemplo, que demonstram o reaproveitamento de cards em páginas distintas:





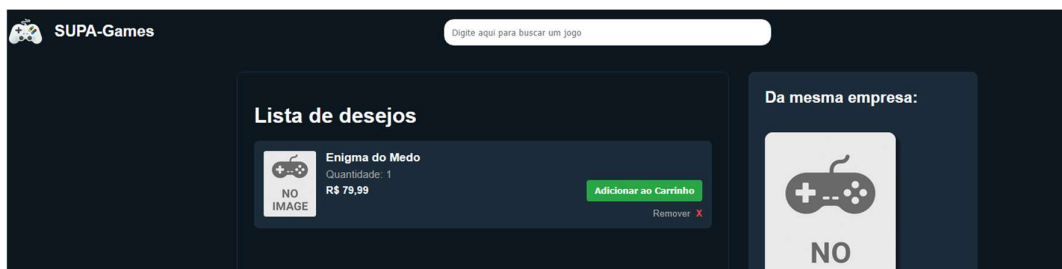


## v. Prevenção de erros:

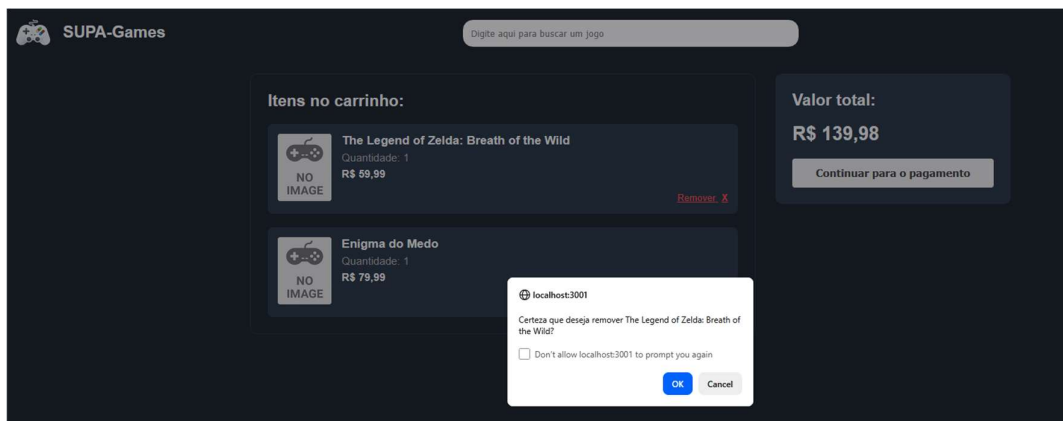
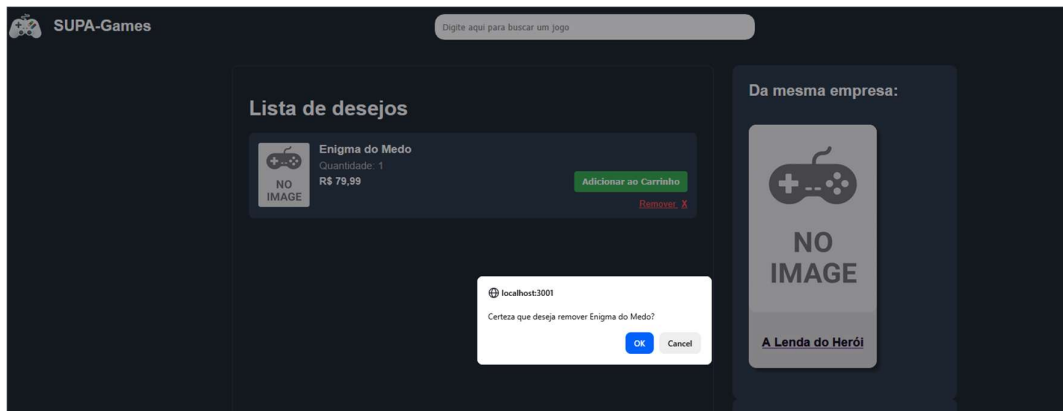
De acordo com essa heurística, melhor do que comunicar o erro ao usuário é ter um *design* e implementação que previne erros de acontecerem em primeiro lugar, que podem ser originados de deslizos ou enganos.

Deslizes são erros inconscientes causados por desatenção do usuário ou da equipe da solução. Enganos são erros conscientes baseados em incompatibilidades entre o modelo mental do usuário para aquela solução e o design.

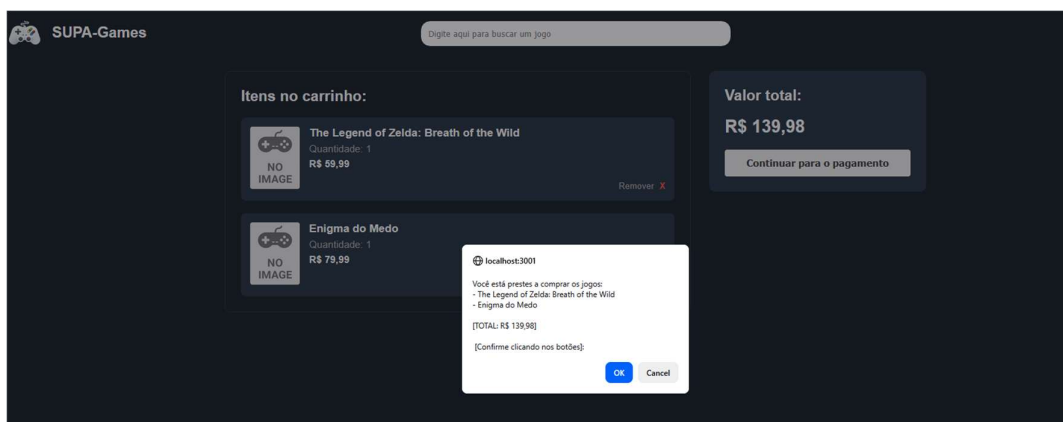
Na aplicação os botões possuem ações claras e individuais, que reduz erros cometidos pelo usuário. Ainda é possível prosseguir ou desfazer ações relacionadas aos objetos (jogos) em diversas telas, como por exemplo na lista de desejos - é possível avançar e adicionar o jogo ao carrinho, ou gerenciar a lista de desejos removendo os jogos inseridos. Tela:



Ações performadas sobre os itens nas páginas de lista de desejo e carrinho são sempre confirmadas com prompts ao usuário - comunicando com a heurística de visibilidade do status - sendo ainda necessário confirmação no caso de ações permanentes, como tirar um jogo da lista de desejos ou do carrinho:



O mesmo se dá na hora da compra, em que é exigida uma confirmação antes do *checkout* completo, de forma que o usuário não realize uma compra sem querer:

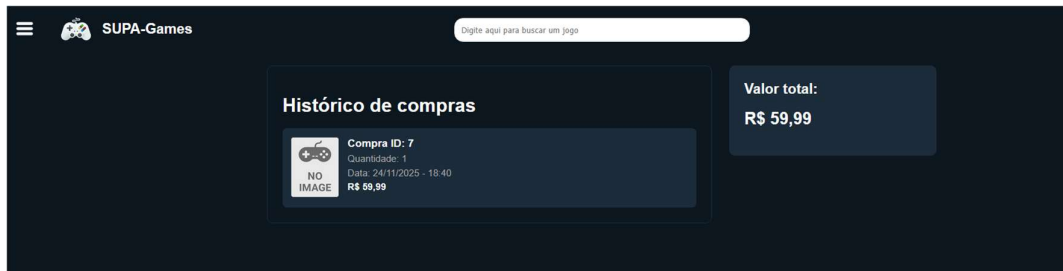


## vi. Reconhecimento em vez de memorização:

Conforme essa heurística, o sistema deve priorizar o reconhecimento visual e funcional de elementos em detrimento da memorização de comandos.

No sistema, esse princípio é aplicado pela utilização de padrões de interface já consolidados, como o ícone do carrinho e o ícone de perfil, ambos localizados no canto superior direito da tela. Essa familiaridade facilita a navegação e reduz a necessidade de aprendizado adicional.

Além desse menu *dropdown* e o menu *hamburger* lateral dinâmico, que já diminuem a carga mental do usuário, as telas que são reaproveitadas ou reutilizadas durante a aplicação possuem descrição clara do seu objetivo, facilitando assim que o usuário se situe e reconheça onde está, além da marcação da página atual de navegação já mencionada acima. Tela:

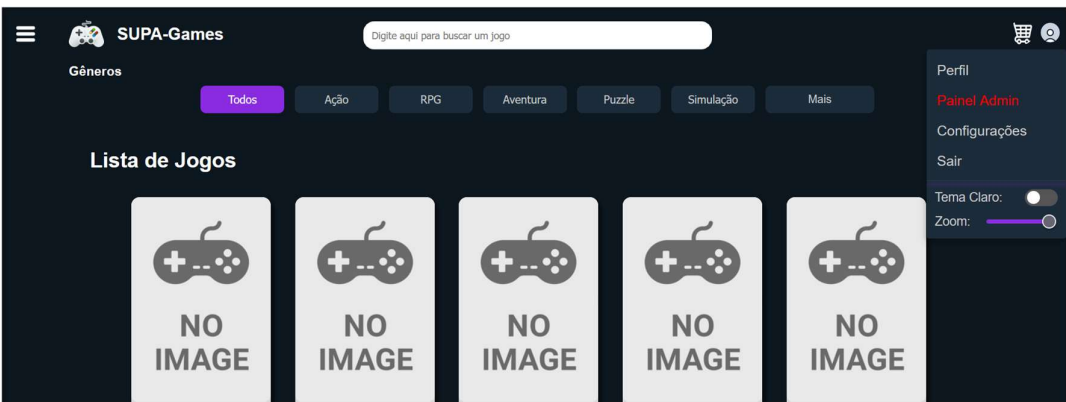
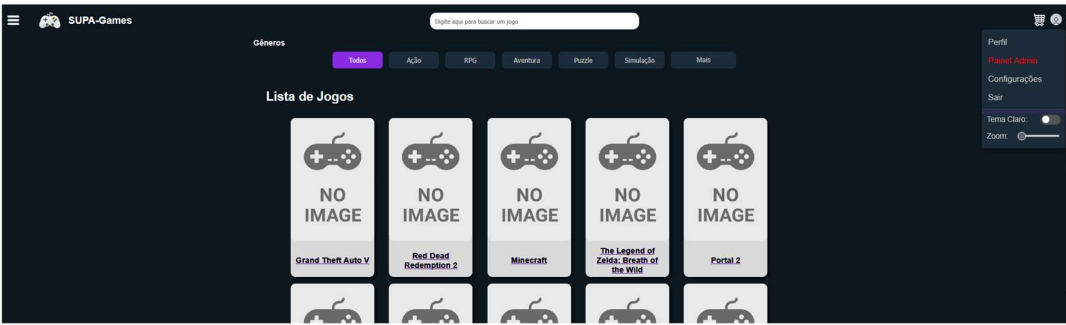
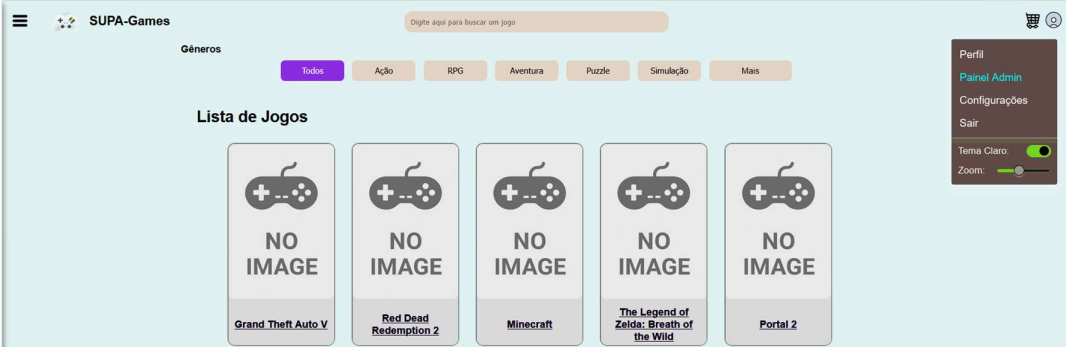
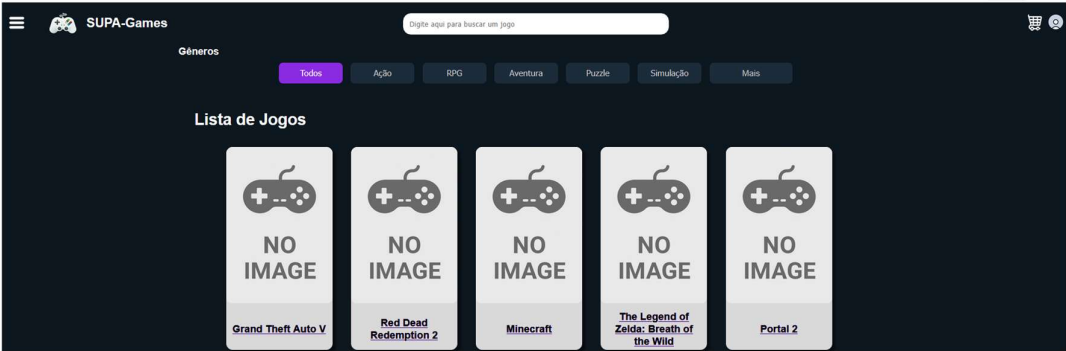


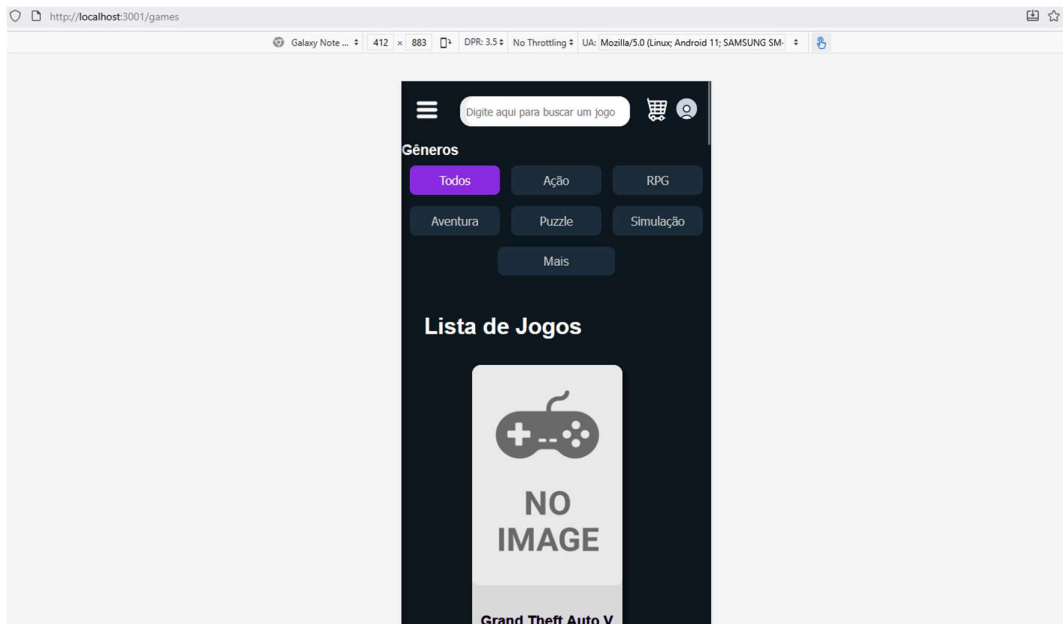
## vii. Eficiência e flexibilidade de uso:

De acordo com essa heurística, a interface deve atender tanto a usuários iniciantes quanto a usuários experientes, permitindo múltiplas formas de execução de uma mesma ação.

Um exemplo dessa aplicação está na área de “perfil”, onde usuários iniciantes podem acessá-la por meio da barra lateral de navegação dinâmica do menu *hamburger*, enquanto usuários mais experientes utilizam diretamente o atalho do ícone de perfil, promovendo agilidade e eficiência.

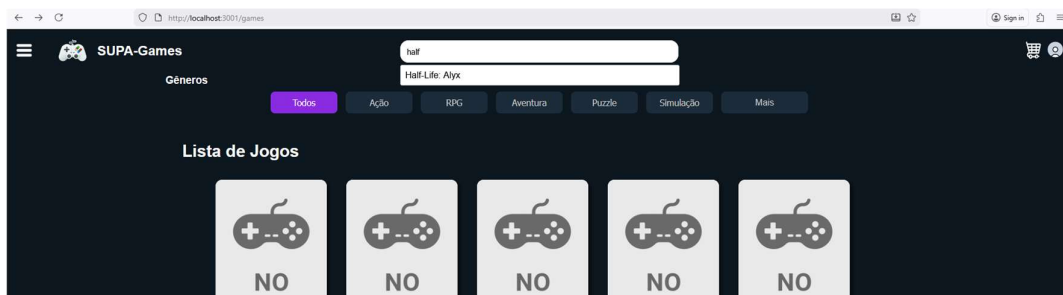
A aplicação ainda garante flexibilidade no uso por meio de *design* responsivo, possibilitando a utilização em dispositivos móveis e tradicionais (*desktop*), além de possuir opção de zoom da tela e de tema alternativo (claro, uma vez que o padrão é escuro). Telas:





Tais possibilidades ainda garantem maior acessibilidade para usuários diversos, com algum nível de deficiência visual ou não.

Outro exemplo é a barra de pesquisa presente em diversas telas da aplicação, disponibilizando ao usuário que encontre jogos procurados mais rapidamente, já filtrando as possibilidades. Tela:



### **viii. Estética e design minimalista:**

O design de uma interface deve ser simples, limpo e funcional, evitando a sobrecarga de informações visuais e textuais.

Assim, o site foi desenvolvido seguindo essa heurística, priorizando clareza, equilíbrio visual, limpeza e hierarquia de informações. Elementos desnecessários foram reduzidos, favorecendo a legibilidade e o foco do usuário nas ações principais.

Por padrão, foi dada preferência pelo modo escuro, em cor sólida, sem elementos desnecessários visíveis. O espaçamento, margens e bordas são todos padronizados, o que faz com que o usuário foque muito mais nas informações principais.

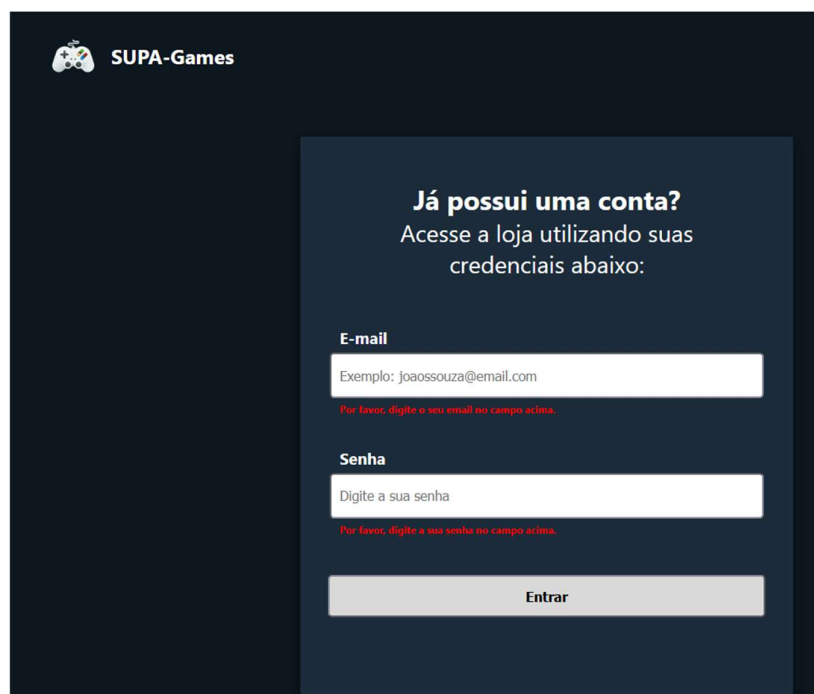
Os cards dos jogos usam designs simples e agradáveis, com os nomes dos jogos localizados abaixo deles, priorizando a legibilidade do usuário, ainda que a imagem não seja apresentada por algum defeito de carregamento.

Deixamos de apresentar telas nessa seção, uma vez que se trata de uma abordagem mais geral e que já está incluída/exemplificada nas demais imagens apresentadas.

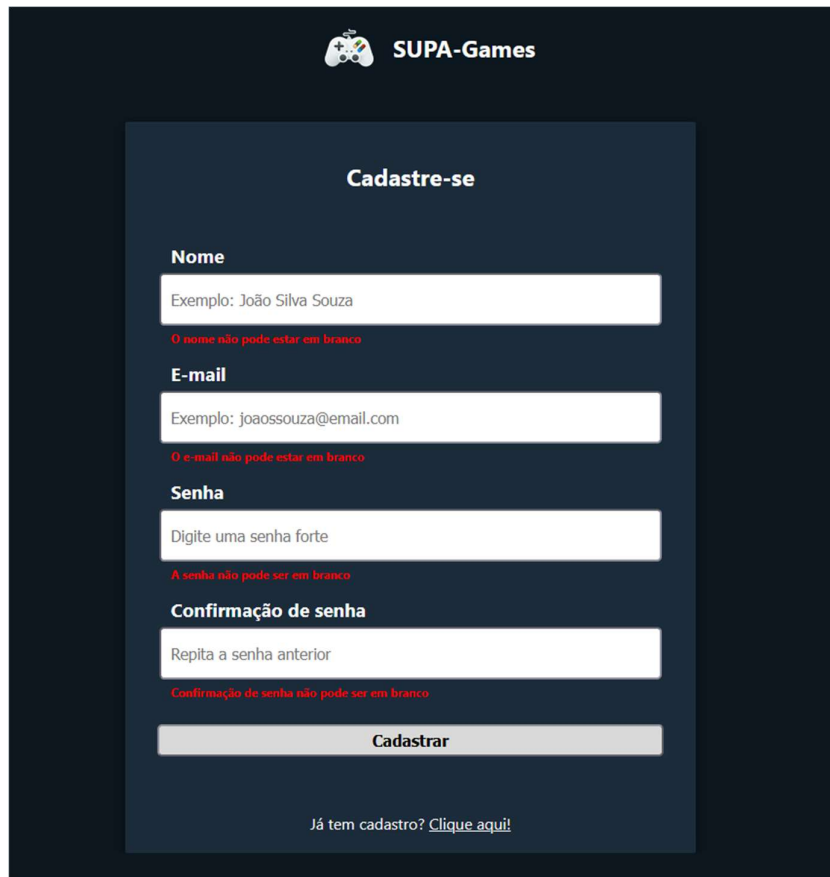
## **ix. Ajude os usuários a reconhecerem, diagnosticarem e recuperarem-se de erros:**

O sistema deve fornecer mensagens de erro claras, precisas e construtivas, auxiliando o usuário na identificação e resolução de problemas.

No processo de cadastro e autenticação, por exemplo, validações automáticas detectam erros comuns, como e-mails inválidos ou senhas fracas, exibindo mensagens explicativas e impedindo o prosseguimento até que o problema seja corrigido. Essa prática reduz falhas futuras e melhora significativamente a experiência do usuário. Telas:



A imagem mostra a interface de login do sistema SUPA-Games. No topo, há um ícone de controle de jogo e o texto "SUPA-Games". O formulário centralizado pergunta "Já possui uma conta?" e instrui o usuário a acessar a loja com suas credenciais. Ele contém campos para "E-mail" (com o exemplo "joaossouza@email.com") e "Senha", ambos com mensagens de erro em vermelho: "Por favor, digite o seu email no campo acima." e "Por favor, digite a sua senha no campo acima." respectivamente. Um botão "Entrar" está na base do formulário.



The image shows a registration form for 'SUPA-Games'. At the top, there is a logo of a game controller and the text 'SUPA-Games'. The form is titled 'Cadastre-se' (Register). It contains four input fields: 'Nome' (Name) with the example 'João Silva Souza' and a red error message 'O nome não pode estar em branco'; 'E-mail' with the example 'joaossouza@email.com' and a red error message 'O e-mail não pode estar em branco'; 'Senha' (Password) with the prompt 'Digite uma senha forte' and a red error message 'A senha não pode ser em branco'; and 'Confirmação de senha' (Confirm password) with the prompt 'Repita a senha anterior' and a red error message 'Confirmação de senha não pode ser em branco'. Below the fields is a 'Cadastrar' button. At the bottom, there is a link: 'Já tem cadastro? [Clique aqui!](#)'.

#### x. Ajuda e documentação:

No futuro, como parte de entregas posteriores, será implementada uma página de ajuda com documentação adicional sobre a aplicação e sua utilização, FAQ (*frequently asked questions*) e artigos de exemplo de utilização.

Há ainda a previsão de ícones de ajuda com mais descrições em opções eventualmente não tão claras ou que necessitam de explicações adicionais.

## 4. INTEGRAÇÃO COM BACKEND

Seguindo uma arquitetura MVC (*Model-View-Controller*), nossa aplicação estava integrada com um backend, desenvolvido em *Node.JS*, acessado por meio de uma *API* (*Application programming interface*) que roda em outra máquina ou porta de uma mesma máquina. Isso garante maior segurança, além de permitir flexibilidade e escalabilidade melhoradas, quando comparada a uma solução de *host* único.

Todas as chamadas para esta *API* eram feitas por meio do protocolo HTTP e o padrão de trocas de informação escolhido foi o *JSON (Javascript Object Notation)*, moderno e ideal para aplicações do tipo *web*, como a presente. Para as requisições junto ao *backend*, foi adotada a biblioteca *Axios*, presente no NPM, que auxilia na montagem de chamadas HTTP de forma mais legível.

Além das verificações no *front*, essa aplicação do *backend* ainda era responsável por, além de persistir e manipular os dados de importância junto ao banco de dados - do tipo relacional *SQL*, rodando em *SQLITE* - reforçar as regras de negócio pertinentes.

## 5. CONCLUSÃO

O design do sistema apresenta uma base sólida para a usabilidade do usuário, seguindo as melhores práticas estabelecidas nas heurísticas de Nielsen, referência sobre o tema. Demonstra ainda cuidado com a experiência do usuário, possuindo uma interface minimalista, que atende a todos os requisitos, sem distrações desnecessárias ou poluição visual, sendo agradável e sem sobrecarga sensorial.

Apresenta um uso consistente de cores, componentes visuais adequados e já familiares aos usuários em outros *websites*, atendendo às normas e padrões utilizados em outras plataformas e sistemas amplamente conhecidos.

Finalmente, e apesar de estarmos seguros nos melhores padrões de usabilidade seguidos, reconhece-se que sempre há espaço para incrementos e melhorias, que certamente serão avaliados e eventualmente implementados na solução.