

Laboratório de Engenharia de Software

LES

Desenvolvimento de Software

Versão: 21/11/2017

*Prof. Antonio Sergio Bernardo*

RESUMO 3

1. INTRODUÇÃO.......,..... 4

2. ANÁLISE DE NEGÓCIOS

2.1 Análise da Situação Atual 4

2.1.1 Descrição do Macro-Processo 4

2.1.2 Diagrama de Processos As Is 5

2.1.3 Diagnóstico da Situação Atual 5

2.2 Planejamento de Melhorias 6

2.2.1 Diagrama de Processos To Be 7

2.2.2 Descrição dos Processos To Be 7

2.2.3 Plano de Implementação (Word) 8

3. SISTEMA DE INFORMAÇÃO

3.1 Requisitos Não Funcionais...................................................................................................................... 9

3.1.1 Requisitos Organizacionais............................................................................................................ 9

3.1.2 Requisitos de Produto 9

3.1.3 Requisitos de Externos 9

3.2 Requisitos Funcionais 10

3.2.1 Diagramas de Casos de Uso 10

3.2.2 Descrição dos Casos de Uso 11

3.3 Projeto Detalhado do Software 14

3.3.1 Arquitetura da Aplicação 14

3.3.2 Tecnologias a Serem Utilizadas 14

3.3.3 Componentes do Sw 15

3.3.4 Modelo de Dados 25

3.3.5 Protótipo Back-End 25

3.3.6 Protótipo Front-End 36

**Nome do Projeto**

RESUMO

O projeto criado é um e- commerce de jogos eletrônicos, sendo voltada mais para a aplicação web, onde um usuário pode criar uma conta, selecionar jogos ao qual ele deseja, adiciona-los no carrinho de compras, e em seguida ele pode editar o carrinho de compra, adicionando ou removendo algum jogo, ou somente realizando a compra dos selecionados.

1. INTRODUÇÃO

A To play oferece um grande número de opções de jogos para compra, de todos gêneros, incluindo lançamentos.

Porém, eles usando de um caderno para anotar as baixas, assim como um sistema de organização dos jogos, sendo por data de lançamento, o que acaba causando uma grande dificuldade em encontrar um título especifico.

É usado uma máquina de cartão, que nem sempre funciona como deveria, o que acaba por gerar um transtorno desnecessário, causando o desinteresse do cliente em realiza a compra.

1. ANÁLISE DE NEGÓCIOS
   1. Análise da Situação Atual

2.1.1 Descrição do Macro-Processo To play

. Requisito Tipo: Map (Macro-Processo) As Is

. Sub Requisitos As Is

- Seleção do jogo (Processo Primário)

- Compra do jogo (Processo Primário)

Nome: To play

Telefone: (15) 3368-4963

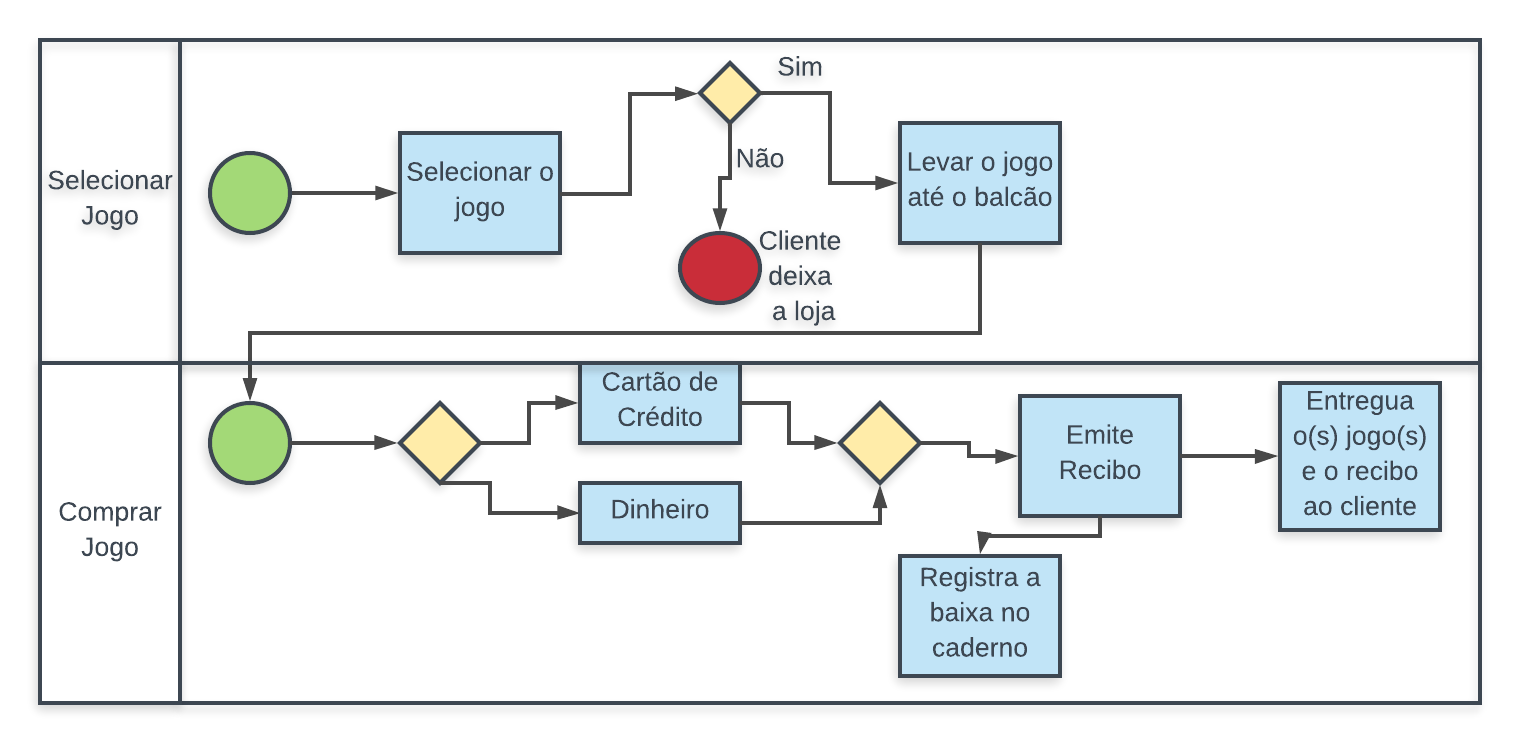
Site: Não tem

Horário de funcionamento: todos os dias das 8h às 23h

Formas de pagamento: Dinheiro, Cartões de Crédito.

Perfil: A To play oferece um grande número de opções de jogos para compra, de todos gêneros, incluindo lançamentos.

* + 1. Diagrama de Processos As Is



* + 1. Diagnóstico da Situação Atual

Nome da Análise: Revisão de Procedimentos

Descrição:

Na Seleção:

1 - Os jogos estão organizados por data de lançamento, o que acaba causando uma dificuldade para o cliente selecionar o que ele deseja, já que nem sempre o cliente sabe a data de lançamento do seu jogo, o que leva ao cliente, fica um tempo procurando o jogo, o que pode causar a perca de interesse do mesmo.

Na Compra:

2 – A loja usa de uma máquina de cartão, que é o meio mais usado, mas o mesmo, nem sempre está funcionando, pois, essa máquina, depende de sinal telefônico para realizar as operações.

* 1. Planejamento de Melhorias
     1. Especificação das Mudanças

Nome da Análise: Revisão de Procedimentos

Descrição:

1 – Criar uma ferramenta web para que o usuário possa seleciona e comprar a distância.

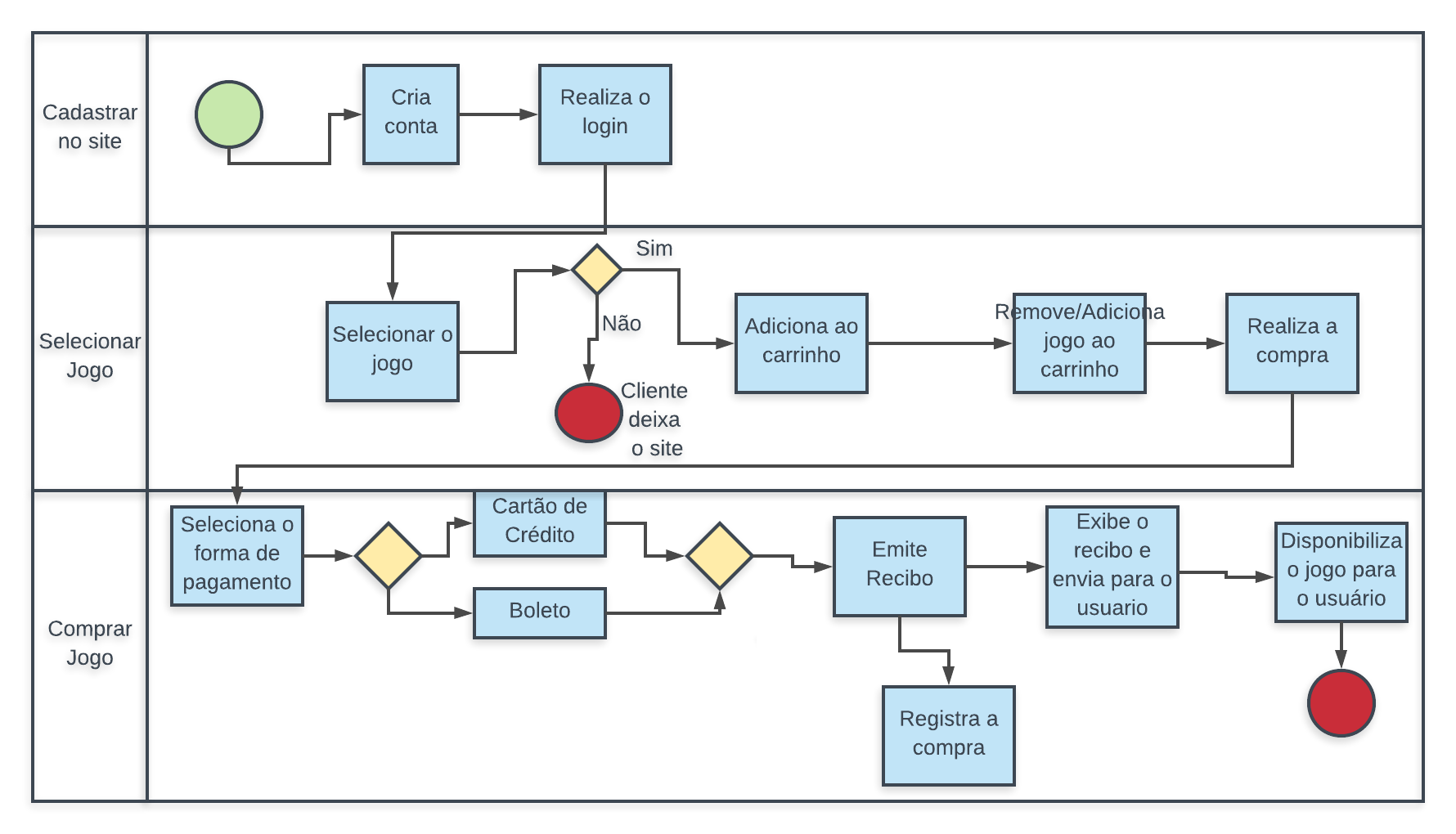
As Melhorias estão especificadas no Modelo Proposto (To Be)

a) Controle de compra será informatizado, evitando confusões nas anotações sobre as baixas realizadas no caderno.

b) Os clientes poderão procurar os jogos usando vários filtros como, nome, gênero.

c) Toda comprar se realizada online, sem a necessidade de uma máquina de cartão.

2.2.4 Diagrama de Processos To Be



2.2.5 Descrição dos Processos To Be

Cadastrar no site

O usuário acessa o site, e cria uma conta.

Selecionar Jogo

O usuário navega pelo site e seleciona o(s) jogo(s) que ele deseja e os adiciona em seu carrinho.

Comprar Jogo

O usuário acessa o seu carrinha e seleciona o método de pagamento.

* + 1. Plano de Implementação

1 Toda mídia física será substituída pela mídia digital.

2 Caso, seja optado por manter o estabelecimento físico, os funcionário serão treinados para operar o site, para que, mesmo que o cliente possa comprar a distância, ele opte por ir fisicamente na loja para comprar, mesmo que não seja vendida mídia física.

3 No primeiro mês de uso da nova aplicação, será estudado, de acordo com o feedback, e será feita novas melhorias.

4 A partir do segundo mês, usando o feedback, assim como a procura de determinados títulos e gêneros, será colocado em prática uma melhor exibição dos mais procurados.

6 Será realizada manutenções periódicas todo mês, visando mantes o site atualizado e operante.

1. SISTEMA DE INFORMAÇÃO

3.1 Requisitos Não Funcionais

3.1.1 Requisitos Organizacionais

RNF1 – Compatibilidade com navegadores Internet

O sistema funcionará nos navegadores de internet Internet Explorer, Google Chrome e Mozilla Firefox.

3.1.2 Requisitos de Produto

RNF2 - Carga rápida de dados

Carregamento rápido de dados do servidor, as páginas do sistema devem carregar no máximo em 3 segundos.

RNF3 - Espaço

A aplicação rodará via web, e possui informações que são armazenadas em um datacenter, portanto não é necessário um hardware específico para que o sistema seja operado, bastando ter apenas conexão com internet, para pode operar o sistema.

RNF4 – Interface Intuitiva

Interface desenvolvida para facilitar a agilidade e seu uso.

3.1.3 Requisitos Externos

RNF5 – Acesso e alteração do banco de dado somente por usuários com permissão

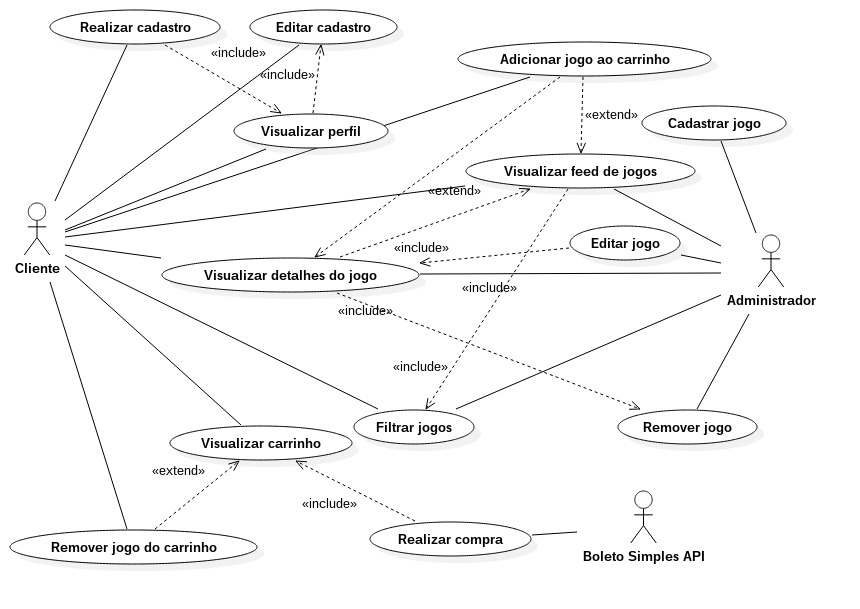
Todo e qualquer dado do banco de dados, só poderá ser alterado por aquela com permissão para tal.

RNF6 – Privacidade

Todo e qualquer dado inserido no banco de dado, que, referente aos dados do usuário, por razões legais, não serão divulgados.

* 1. Requisitos Funcionais

3.2.1 Diagramas de Casos de Uso



3.2.2 Descrição dos Casos de Uso

|  |
| --- |
| **Caso de uso:** Realizar cadastro |
| **Atores:** Cliente |
| **Interessados:** Cliente que deseja ter acesso as compras |
| **Pré-condições:** Cliente não possui cadastro no sistema |
| **Pós-condições:** Usuário é redirecionado para o feed de jogos |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O cliente entra em cadastro de usuário 2. O cliente insere seus dados pessoais 3. O sistema valida os dados 4. O cliente salva os dados no sistema |
| **Tratamento de exceções:**  **2.** **Os dados estão inválidos**  2.1 Volta passo 2 |

|  |
| --- |
| **Caso de uso:** Editar cadastro |
| **Atores:** Cliente |
| **Interessados:** Cliente que deseja atualizar seu cadastro |
| **Pré-condições:** Cliente deve possuir cadastro no sistema |
| **Pós-condições:** Mensagem de atualização é mostrada ao usuário |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O cliente visualiza o perfil 2. O cliente insere seus dados pessoais 3. O sistema valida os dados 4. O cliente salva os dados no sistema |
| **Tratamento de exceções:**  **2.** **Os dados estão inválidos**  2.1 Volta passo 2 |

|  |
| --- |
| **Caso de uso:** Visualizar perfil |
| **Atores:** Cliente |
| **Interessados:** Cliente que deseja visualizar seu perfil |
| **Pré-condições:** Estar cadastrado no sistema |
| **Pós-condições:** Perfil é mostrado ao usuário |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O cliente entra em visualizar perfil |
| **Tratamento de exceções:** |

|  |
| --- |
| **Caso de uso:** Adicionar jogo ao carrinho |
| **Atores:** Cliente |
| **Interessados:** Cliente que deseja comprar um jogo |
| **Pré-condições:** O cliente deve possuir cadastro no sistema |
| **Pós-condições:** O cliente deve poder continuar a navegação |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O cliente seleciona adicionar ao carrinho 2. O jogo é adicionado ao carrinho |
| **Tratamento de exceções:** |

|  |
| --- |
| **Caso de uso:** Cadastrar jogo |
| **Atores:** Administrador |
| **Interessados:** O administrador do sistema |
| **Pré-condições:** O administrador deve possuir todos os dados do jogo |
| **Pós-condições:** Mensagem de jogo cadastrado deve ser mostrado |
| **Requisitos correlacionados:** |
| **Variações tecnológicas:** |
| **Questões em aberto** |
| **Fluxo principal:**   1. O administrador deve selecionar cadastrar jogo 2. O administrador deve inserir os dados 3. O sistema deve validar os dados 4. O administrador salva os dados |
| **Tratamento de exceções**  **2 O sistema deve validar os dados**  2.1 Volta ao passo 2 |

* 1. Projeto Detalhado do Sofware
     1. Arquitetura da Aplicação
     2. FrameWork (front-end)

AngularJS: Framework para desenvolvimento do front-end de aplicações web.

* + 1. Banco de Dados

MongoDB: software de banco de dados orientado a documentos livre, de código aberto e multiplataforma, escrito na linguagem C++. Classificado como um programa de banco de dados NoSQL

* + 1. Back-End

NODE.js: é uma plataforma construída sobre o motor JavaScript do Google Chrome para facilmente construir aplicações de rede rápidas e escaláveis.

3.3.2 Tecnologias a Serem Utilizadas

As tecnologias usadas nesse projeto são: Angular, MongoDB e NODE.js

3.3.3 Componentes do Sw

#### **Add-Game.component**

|  |
| --- |
| <div class="container body-container"> |
|  | <form> |
|  | <div class="form-group"> |
|  | <label>Nome</label> |
|  | <input name="name" [(ngModel)]="this.game.name" type="text" class="form-control" placeholder="Enter game name"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Sinopse</label> |
|  | <input name="synopse" [(ngModel)]="this.game.synopse" type="text" class="form-control" placeholder="Enter game synopse"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Preço</label> |
|  | <input name="preco" [(ngModel)]="this.game.price" type="number" class="form-control" placeholder="Enter game price"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Release date</label> |
|  | <input name="date" [(ngModel)]="this.game.releaseDate" type="date" class="form-control" placeholder="Enter game release date"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Developer</label> |
|  | <input name="developer" [(ngModel)]="this.game.developer" type="text" class="form-control" placeholder="Enter game developer"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Editor</label> |
|  | <input name="editor" [(ngModel)]="this.game.editor" type="text" class="form-control" placeholder="Enter game editor"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Image</label> |
|  | <input name="image" [(ngModel)]="this.game.imgsrc" type="text" class="form-control" placeholder="Enter image link of the game"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Gênero</label> |
|  | <select class="form-control"> |
|  | <option \*ngFor="let gender of genders">{{gender}}</option> |
|  | </select> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Tag</label> |
|  | <select class="form-control"> |
|  | <option \*ngFor="let tag of tags">{{tag}}</option> |
|  | </select> |
|  | </div> |
|  | <button class="btn btn-primary" (click)="addGame()">Adicionar jogo</button> |
|  | <a router-link="/home"> |
|  | <button class="btn btn-primary" routerLink="/home">Cancelar</button> |
|  | </a> |
|  |  |
|  | </form> |
|  | </div> |

**Cart.Component**

|  |
| --- |
| <div class="container"> |
|  | <div class="card"> |
|  | <div class="card-header d-flex flex-row justify-content-between"> |
|  | <div>Produto</div> |
|  | <div class="w-25 text-center">Preço</div> |
|  | </div> |
|  |  |
|  | <div class="card-body d-flex flex-xs-column flex-sm-column flex-md-row flex-lg-row flex-xl-row justify-content-between" \*ngFor="let game of cart"> |
|  |  |
|  | <div class="d-flex flex-row align-items-center align-content-end"> |
|  | <img class="img-cart d-none d-sm-block" src="{{game.imgsrc}}"> |
|  | <div class="game-title-cart">{{game.name}}</div> |
|  | </div> |
|  | <div class="d-flex flex-row align-items-center justify-content-end w-25"> |
|  | <div class="pr-lg-5 pr-xg-5">R$ {{game.price}},00</div> |
|  | <img class="pl-3" style="width: 50px; height: 30px" src="https://cdn4.iconfinder.com/data/icons/defaulticon/icons/png/256x256/cancel.png" alt="remove" (click)="removeGame(game)"> |
|  | </div> |
|  | </div> |
|  | </div> |
|  | <button type="button" class="btn btn-light btn-lg btn-block mt-2" [ngClass]="{'d-none':this.cart.length == 0}" (click)="buyGames()">Comprar</button> |
|  | </div> |

**Create-account.component**

|  |
| --- |
| <div class="container body-container"> |
|  | <form> |
|  | <div class="form-group"> |
|  | <label>Email</label> |
|  | <input name="email" [(ngModel)]="this.user.email" type="email" class="form-control" aria-describedby="emailHelp" placeholder="Enter email"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Password</label> |
|  | <input name="pass" [(ngModel)]="this.user.pass" type="password" class="form-control" placeholder="Password"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Username</label> |
|  | <input name="username" [(ngModel)]="this.user.name" type="text" class="form-control" placeholder="Username"> |
|  | </div> |
|  | <button class="btn btn-primary" (click)="createAccount()">Criar conta</button> |
|  | </form> |
|  | </div> |

**Game.component**

|  |
| --- |
| <div class="card card-spacer"> |
|  | <div class="card-header" (click)="openModal(game)"> |
|  | <h3 class="card-title"> |
|  | {{game.name}} |
|  | </h3> |
|  | </div> |
|  | <div class="card-body" (click)="openModal(game)"> |
|  | <img class="card-game-img" src="{{game.imgsrc}}"> |
|  | <p> |
|  | <span class="badge badge-dark">{{game.tags}}</span> |
|  | </p> |
|  | <p>{{game.synopse}}</p> |
|  | <p>{{game.releasedate}}</p> |
|  | <p>{{game.developer}}</p> |
|  | <p>{{game.editor}}</p> |
|  | <p>{{game.gender}}</p> |
|  | <p>{{game.rating}}</p> |
|  | <h4>R$ {{game.price}},00</h4> |
|  | </div> |
|  | <div class="card-footer"> |
|  | <button class="card-btn-footer" [ngClass]="{'bg-success':isInCart(game)}" (click)="addCart(game)"> |
|  | </button> |
|  |  |
|  | </div> |
|  | </div> |

**Home.component**

|  |
| --- |
| <div class="container body-container"> |
|  | <div class="row"> |
|  | <aside class="col-12 col-sm-6 col-lg-3 "> |
|  | <div class="sticky-top"> |
|  | <button id="toggler" class="navbar-toggler filter-toggler d-sm-block d-xs-block d-md-none d-lg-none d-xl-none" type="button" data-toggle="collapse" data-target="#filter" aria-controls="filter" |
|  | aria-expanded="true" aria-label="Toggle navigation"> |
|  | <i class="material-icons">filter\_list</i> |
|  | </button> |
|  | <div id="filter" class="card collapse show"> |
|  | <div class="card-header card-filtro-spacer"> |
|  | <a class="card-filtro-title">Filtro</a> |
|  | <a class="btn card-filter-button" (click)="searchTitle()"> |
|  | <img class="card-img" src="../assets/img/search.png"> |
|  | </a> |
|  | </div> |
|  | <div class="card-body"> |
|  | <input type="search" placeholder="Pesquisar" [(ngModel)]="this.searchBox" > |
|  | <ul class="list-group list-group-flush" \*ngFor="let gender of genders"> |
|  | <li class="list-group-item"> |
|  | <input type="checkbox" (click)="setFilter(gender)"> |
|  | <label>{{gender}}</label> |
|  | </li> |
|  | </ul> |
|  | </div> |
|  | </div> |
|  | </div> |
|  |  |
|  | </aside> |
|  | <section class="col-12 col-sm-6 col-lg-9"> |
|  | <div class="row"> |
|  | <div class="col-12 col-lg-6 d-flex flex-row align-content-center" \*ngFor="let game of games"> |
|  | <div class="card card-spacer"> |
|  | <div class="card-header" (click)="openModal(game)"> |
|  | <h3 class="card-title"> |
|  | {{game.name}} |
|  | </h3> |
|  | </div> |
|  | <div class="card-body" (click)="openModal(game)"> |
|  | <img class="card-game-img" src="{{game.imgsrc}}"> |
|  | <p> |
|  | <span class="badge badge-dark">{{game.tags}}</span> |
|  | </p> |
|  | <p>{{game.synopse}}</p> |
|  | <p>{{game.rating}}</p> |
|  | <h4>R$ {{game.price}},00</h4> |
|  | </div> |
|  | <div class="card-footer"> |
|  | <button class="card-btn-footer" [ngClass]="{'bg-success':isInCart(game)}" (click)="addCart(game)"> |
|  | </button> |
|  | </div> |
|  | </div> |
|  | </div> |
|  | </div> |
|  | </section> |
|  | </div> |
|  | </div> |

**login.component**

|  |
| --- |
| <div class="container"> |
|  | <form (ngSubmit)="login()"> |
|  | <div class="form-group"> |
|  | <label>Email</label> |
|  | <input name="email" [(ngModel)]="this.user.email" type="email" class="form-control" aria-describedby="emailHelp" placeholder="Enter email"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Password</label> |
|  | <input name="pass" [(ngModel)]="this.user.pass" type="password" class="form-control" placeholder="Password"> |
|  | </div> |
|  | <button type="submit" class="btn btn-primary" (click)="login()">Login</button> |
|  | <button class="btn btn-primary" (click)="criarConta()">Criar conta</button> |
|  | </form> |
|  |  |
|  | </div> |

**popup-confirm.component**

|  |
| --- |
| <div class="card"> |
|  | <div class="card-body"> |
|  | Confirmar compra dos jogos? |
|  | </div> |
|  | <div class="card-footer"> |
|  | <button type="button" class="btn btn-success btn-lg btn-block"> |
|  | Confirmar |
|  | </button> |
|  | <button type="button" class="btn btn-danger btn-lg btn-block"> |
|  | Cancelar |
|  | </button> |
|  | </div> |
|  |  |
|  | </div> |

**app.component**

|  |
| --- |
| <header class="navbar navbar-expand-lg navbar-dark bg-dark justify-content-between"> |
|  | <a class="navbar-brand" routerLink="/home"> |
|  | <img class="img-navbar" src="../assets/img/logo.ico"> |
|  | </a> |
|  | <div> |
|  | <ul class="navbar-nav mr-auto flex-row"> |
|  | <li class="nav-item pr-4" \*ngIf="this.dataStorage.user.adm"> |
|  | <a routerLink="/addGame"> |
|  | <button type="button" class="btn btn-outline-light" >Cadastrar jogo</button> |
|  | </a> |
|  | </li> |
|  | <li class="nav-item pr-4" \*ngIf="!this.dataStorage.user.adm"> |
|  | <a routerLink="/cart" > |
|  | <img class="img-navbar" src="../assets/img/cart.png"> |
|  | </a> |
|  | </li> |
|  | <li class="nav-item" \*ngIf="!this.dataStorage.logged"> |
|  | <a routerLink="/login"> |
|  | <button type="button" class="btn btn-outline-light">Login</button> |
|  | </a> |
|  | </li> |
|  | <li class="nav-item" \*ngIf="this.dataStorage.logged"> |
|  | <img class="img-navbar" src="../assets/img/user.png"> |
|  | </li> |
|  |  |
|  | </ul> |
|  | </div> |
|  |  |
|  | </header> |
|  | <div class="alert alert-success" role="alert" \*ngIf="this.dataStorage.compra" > |
|  | Compra realizada com sucesso!!! |
|  | </div> |
|  | <router-outlet></router-outlet> |

3.3.4 Modelo de Dados

User:

"\_id": "5afba392c390fa4df05ec85b",

"email": "example@example.com",

"username": "example",

"password": "example123",

"type": 1,

"\_\_v": 0

Game:

"\_id": "5b22fd4960860437638be767",

"name": "corrida rapida",

"synopse": "corra o mais rapido ",

"tags": "Corrida",

"price": 12.35,

"releasedate": "2013-03-01T04:10:00.000Z",

"developer": "lero dvpl",

"editor": "lero edts",

"genre": "Corrida",

"rating": 0,

"imgsrc": "https://images.livrariasaraiva.com.br/imagemnet/imagem.aspx/?pro\_id=9171307&qld=90&l=430&a=-1",

"\_\_v": 0

* + 1. Protótipo Front-End

#### **Add-Game.component**

|  |
| --- |
| <div class="container body-container"> |
|  | <form> |
|  | <div class="form-group"> |
|  | <label>Nome</label> |
|  | <input name="name" [(ngModel)]="this.game.name" type="text" class="form-control" placeholder="Enter game name"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Sinopse</label> |
|  | <input name="synopse" [(ngModel)]="this.game.synopse" type="text" class="form-control" placeholder="Enter game synopse"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Preço</label> |
|  | <input name="preco" [(ngModel)]="this.game.price" type="number" class="form-control" placeholder="Enter game price"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Release date</label> |
|  | <input name="date" [(ngModel)]="this.game.releaseDate" type="date" class="form-control" placeholder="Enter game release date"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Developer</label> |
|  | <input name="developer" [(ngModel)]="this.game.developer" type="text" class="form-control" placeholder="Enter game developer"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Editor</label> |
|  | <input name="editor" [(ngModel)]="this.game.editor" type="text" class="form-control" placeholder="Enter game editor"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Image</label> |
|  | <input name="image" [(ngModel)]="this.game.imgsrc" type="text" class="form-control" placeholder="Enter image link of the game"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Gênero</label> |
|  | <select class="form-control"> |
|  | <option \*ngFor="let gender of genders">{{gender}}</option> |
|  | </select> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Tag</label> |
|  | <select class="form-control"> |
|  | <option \*ngFor="let tag of tags">{{tag}}</option> |
|  | </select> |
|  | </div> |
|  | <button class="btn btn-primary" (click)="addGame()">Adicionar jogo</button> |
|  | <a router-link="/home"> |
|  | <button class="btn btn-primary" routerLink="/home">Cancelar</button> |
|  | </a> |
|  |  |
|  | </form> |
|  | </div> |

**Cart.Component**

|  |
| --- |
| <div class="container"> |
|  | <div class="card"> |
|  | <div class="card-header d-flex flex-row justify-content-between"> |
|  | <div>Produto</div> |
|  | <div class="w-25 text-center">Preço</div> |
|  | </div> |
|  |  |
|  | <div class="card-body d-flex flex-xs-column flex-sm-column flex-md-row flex-lg-row flex-xl-row justify-content-between" \*ngFor="let game of cart"> |
|  |  |
|  | <div class="d-flex flex-row align-items-center align-content-end"> |
|  | <img class="img-cart d-none d-sm-block" src="{{game.imgsrc}}"> |
|  | <div class="game-title-cart">{{game.name}}</div> |
|  | </div> |
|  | <div class="d-flex flex-row align-items-center justify-content-end w-25"> |
|  | <div class="pr-lg-5 pr-xg-5">R$ {{game.price}},00</div> |
|  | <img class="pl-3" style="width: 50px; height: 30px" src="https://cdn4.iconfinder.com/data/icons/defaulticon/icons/png/256x256/cancel.png" alt="remove" (click)="removeGame(game)"> |
|  | </div> |
|  | </div> |
|  | </div> |
|  | <button type="button" class="btn btn-light btn-lg btn-block mt-2" [ngClass]="{'d-none':this.cart.length == 0}" (click)="buyGames()">Comprar</button> |
|  | </div> |

**Create-account.component**

|  |
| --- |
| <div class="container body-container"> |
|  | <form> |
|  | <div class="form-group"> |
|  | <label>Email</label> |
|  | <input name="email" [(ngModel)]="this.user.email" type="email" class="form-control" aria-describedby="emailHelp" placeholder="Enter email"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Password</label> |
|  | <input name="pass" [(ngModel)]="this.user.pass" type="password" class="form-control" placeholder="Password"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Username</label> |
|  | <input name="username" [(ngModel)]="this.user.name" type="text" class="form-control" placeholder="Username"> |
|  | </div> |
|  | <button class="btn btn-primary" (click)="createAccount()">Criar conta</button> |
|  | </form> |
|  | </div> |

**Game.component**

|  |
| --- |
| <div class="card card-spacer"> |
|  | <div class="card-header" (click)="openModal(game)"> |
|  | <h3 class="card-title"> |
|  | {{game.name}} |
|  | </h3> |
|  | </div> |
|  | <div class="card-body" (click)="openModal(game)"> |
|  | <img class="card-game-img" src="{{game.imgsrc}}"> |
|  | <p> |
|  | <span class="badge badge-dark">{{game.tags}}</span> |
|  | </p> |
|  | <p>{{game.synopse}}</p> |
|  | <p>{{game.releasedate}}</p> |
|  | <p>{{game.developer}}</p> |
|  | <p>{{game.editor}}</p> |
|  | <p>{{game.gender}}</p> |
|  | <p>{{game.rating}}</p> |
|  | <h4>R$ {{game.price}},00</h4> |
|  | </div> |
|  | <div class="card-footer"> |
|  | <button class="card-btn-footer" [ngClass]="{'bg-success':isInCart(game)}" (click)="addCart(game)"> |
|  | </button> |
|  |  |
|  | </div> |
|  | </div> |

**Home.component**

|  |
| --- |
| <div class="container body-container"> |
|  | <div class="row"> |
|  | <aside class="col-12 col-sm-6 col-lg-3 "> |
|  | <div class="sticky-top"> |
|  | <button id="toggler" class="navbar-toggler filter-toggler d-sm-block d-xs-block d-md-none d-lg-none d-xl-none" type="button" data-toggle="collapse" data-target="#filter" aria-controls="filter" |
|  | aria-expanded="true" aria-label="Toggle navigation"> |
|  | <i class="material-icons">filter\_list</i> |
|  | </button> |
|  | <div id="filter" class="card collapse show"> |
|  | <div class="card-header card-filtro-spacer"> |
|  | <a class="card-filtro-title">Filtro</a> |
|  | <a class="btn card-filter-button" (click)="searchTitle()"> |
|  | <img class="card-img" src="../assets/img/search.png"> |
|  | </a> |
|  | </div> |
|  | <div class="card-body"> |
|  | <input type="search" placeholder="Pesquisar" [(ngModel)]="this.searchBox" > |
|  | <ul class="list-group list-group-flush" \*ngFor="let gender of genders"> |
|  | <li class="list-group-item"> |
|  | <input type="checkbox" (click)="setFilter(gender)"> |
|  | <label>{{gender}}</label> |
|  | </li> |
|  | </ul> |
|  | </div> |
|  | </div> |
|  | </div> |
|  |  |
|  | </aside> |
|  | <section class="col-12 col-sm-6 col-lg-9"> |
|  | <div class="row"> |
|  | <div class="col-12 col-lg-6 d-flex flex-row align-content-center" \*ngFor="let game of games"> |
|  | <div class="card card-spacer"> |
|  | <div class="card-header" (click)="openModal(game)"> |
|  | <h3 class="card-title"> |
|  | {{game.name}} |
|  | </h3> |
|  | </div> |
|  | <div class="card-body" (click)="openModal(game)"> |
|  | <img class="card-game-img" src="{{game.imgsrc}}"> |
|  | <p> |
|  | <span class="badge badge-dark">{{game.tags}}</span> |
|  | </p> |
|  | <p>{{game.synopse}}</p> |
|  | <p>{{game.rating}}</p> |
|  | <h4>R$ {{game.price}},00</h4> |
|  | </div> |
|  | <div class="card-footer"> |
|  | <button class="card-btn-footer" [ngClass]="{'bg-success':isInCart(game)}" (click)="addCart(game)"> |
|  | </button> |
|  | </div> |
|  | </div> |
|  | </div> |
|  | </div> |
|  | </section> |
|  | </div> |
|  | </div> |

**login.component**

|  |
| --- |
| <div class="container"> |
|  | <form (ngSubmit)="login()"> |
|  | <div class="form-group"> |
|  | <label>Email</label> |
|  | <input name="email" [(ngModel)]="this.user.email" type="email" class="form-control" aria-describedby="emailHelp" placeholder="Enter email"> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label>Password</label> |
|  | <input name="pass" [(ngModel)]="this.user.pass" type="password" class="form-control" placeholder="Password"> |
|  | </div> |
|  | <button type="submit" class="btn btn-primary" (click)="login()">Login</button> |
|  | <button class="btn btn-primary" (click)="criarConta()">Criar conta</button> |
|  | </form> |
|  |  |
|  | </div> |

**popup-confirm.component**

|  |
| --- |
| <div class="card"> |
|  | <div class="card-body"> |
|  | Confirmar compra dos jogos? |
|  | </div> |
|  | <div class="card-footer"> |
|  | <button type="button" class="btn btn-success btn-lg btn-block"> |
|  | Confirmar |
|  | </button> |
|  | <button type="button" class="btn btn-danger btn-lg btn-block"> |
|  | Cancelar |
|  | </button> |
|  | </div> |
|  |  |
|  | </div> |

**app.component**

|  |
| --- |
| <header class="navbar navbar-expand-lg navbar-dark bg-dark justify-content-between"> |
|  | <a class="navbar-brand" routerLink="/home"> |
|  | <img class="img-navbar" src="../assets/img/logo.ico"> |
|  | </a> |
|  | <div> |
|  | <ul class="navbar-nav mr-auto flex-row"> |
|  | <li class="nav-item pr-4" \*ngIf="this.dataStorage.user.adm"> |
|  | <a routerLink="/addGame"> |
|  | <button type="button" class="btn btn-outline-light" >Cadastrar jogo</button> |
|  | </a> |
|  | </li> |
|  | <li class="nav-item pr-4" \*ngIf="!this.dataStorage.user.adm"> |
|  | <a routerLink="/cart" > |
|  | <img class="img-navbar" src="../assets/img/cart.png"> |
|  | </a> |
|  | </li> |
|  | <li class="nav-item" \*ngIf="!this.dataStorage.logged"> |
|  | <a routerLink="/login"> |
|  | <button type="button" class="btn btn-outline-light">Login</button> |
|  | </a> |
|  | </li> |
|  | <li class="nav-item" \*ngIf="this.dataStorage.logged"> |
|  | <img class="img-navbar" src="../assets/img/user.png"> |
|  | </li> |
|  |  |
|  | </ul> |
|  | </div> |
|  |  |
|  | </header> |
|  | <div class="alert alert-success" role="alert" \*ngIf="this.dataStorage.compra" > |
|  | Compra realizada com sucesso!!! |
|  | </div> |
|  | <router-outlet></router-outlet> |

3.3.6 Protótipo Back-End

* + 1. API

#### /users

* **POST**

Inserts a user into database. A Json body is required as shown next.

$ curl -s -XPOST "localhost:8080/users" -H "Content-Type: application/json" -d '{

"email": "example@example.com",

"username": "example",

"password": "example123",

"type": 1

}' | jq

{

"\_id": "5afba392c390fa4df05ec85b",

"email": "example@example.com",

"username": "example",

"password": "example123",

"type": 1,

"\_\_v": 0

}

#### /users/:userId

* **PUT**

Change a user's data. Id must be passed as param (**:userId**), a Json body with location data is also required as shown.

$ curl -s -XPUT "localhost:8080/users/5afba392c390fa4df05ec85b" -H "Content-Type: application/json" -d '{

"email": "example@changed.com",

"username": "anotherexample",

"password": "examplero",

"type": 2

}' | jq

{

"\_id": "5afba392c390fa4df05ec85b",

"email": "example@changed.com",

"username": "anotherexample",

"password": "examplero",

"type": 2,

"\_\_v": 0

}

#### /users

* **GET**

get all users data

$ curl -s "localhost:8080/users" -H "Content-Type: application/json" | jq

[

{

"\_id": "5b2194cbe383a5303d0c8d0b",

"email": "example@example.com",

"username": "example",

"password": "example123",

"type": 1,

"\_\_v": 0

},

{

"\_id": "5b2195b10ec977309985cb5f",

"email": "example@example.com",

"username": "example",

"password": "example123",

"type": 1,

"\_\_v": 0

}

]

#### /users/:userId

* **GET**

get single user data. userId's data.

$ curl -s "localhost:8080/users/5b2194cbe383a5303d0c8d0b" -H "Content-Type: application/json" | jq

{

"\_id": "5b2194cbe383a5303d0c8d0b",

"email": "example@example.com",

"username": "example",

"password": "example123",

"type": 1,

"\_\_v": 0

}

### /users/:userId

* **DELETE**

delete specified users data from db.

$ curl -s -XDELETE "localhost:8080/users/5b2194cbe383a5303d0c8d0b" -H "Content-Type: application/json" | jq

{

"message": "User deleted."

}

## Login

#### /login

* **POST**

send user's username and login. if it matches with db data, returns users data.

$ curl -s -XPOST "localhost:8080/login" -H "Content-Type: application/json" -d '{

"username": "example",

"password": "example123"

}' | jq

{

"\_id": "5b2195b10ec977309985cb5f",

"email": "example@example.com",

"username": "example",

"password": "example123",

"type": 1,

"\_\_v": 0

}

wrong input

{

"error": "username or password incorrect."

}

## Games endpoints

#### /games

* **POST**

Inserts a game into database. A Json body is required as shown next.

$ curl -s -XPOST "localhost:8080/games" -H "Content-Type: application/json" -d '{

"name": "corrida rapida",

"synopse": "corra o mais rapido ",

"tags": "Corrida",

"price": 12.35,

"releasedate": "2013-03-01T01:10:00",

"developer": "lero dvpl",

"editor": "lero edts",

"genre": "Corrida",

"rating": 0,

"imgsrc": "https://images.livrariasaraiva.com.br/imagemnet/imagem.aspx/?pro\_id=9171307&qld=90&l=430&a=-1"

}' | jq

{

"\_id": "5b22fd4960860437638be767",

"name": "corrida rapida",

"synopse": "corra o mais rapido ",

"tags": "Corrida",

"price": 12.35,

"releasedate": "2013-03-01T04:10:00.000Z",

"developer": "lero dvpl",

"editor": "lero edts",

"genre": "Corrida",

"rating": 0,

"imgsrc": "https://images.livrariasaraiva.com.br/imagemnet/imagem.aspx/?pro\_id=9171307&qld=90&l=430&a=-1",

"\_\_v": 0

}

#### /games/:gameId

* **PUT**

Change a games's data. Id must be passed as param (**:gameId**), a Json body with location data is also required as shown.

$ curl -s -XPUT "localhost:8080/games/5b22f72bee1b383613e6c8f2" -H "Content-Type: application/json" -d '{

"name": "sanic gotta go slow",

"synopse": "ouriço",

"tags": "corrida",

"price": 12.35,

"releasedate": "2013-03-01T04:10:00.000Z",

"developer": "lero dvpl",

"editor": "lero edts",

"genre": "horror",

"rating": 0,

"imgsrc": "https://http2.mlstatic.com/sonic-forces-ps4-midia-fisica-lacrado-novo-original-D\_NQ\_NP\_658382-MLB26464235963\_112017-F.webp"

}' | jq

{

"\_id": "5b22f72bee1b383613e6c8f2",

"name": "sanic gotta go slow",

"synopse": "ouriço",

"tags": "corrida",

"price": 12.35,

"releasedate": "2013-03-01T04:10:00.000Z",

"developer": "lero dvpl",

"editor": "lero edts",

"genre": "horror",

"rating": 0,

"imgsrc": "https://http2.mlstatic.com/sonic-forces-ps4-midia-fisica-lacrado-novo-original-D\_NQ\_NP\_658382-MLB26464235963\_112017-F.webp",

"\_\_v": 0

}

#### /games

* **GET**

get all games data

$ curl -s "localhost:8080/games" -H "Content-Type: application/json" | jq

[

{

"\_id": "5b22f72bee1b383613e6c8f2",

"name": "sanic gotta go slow",

"synopse": "ouriço",

"tags": "corrida",

"price": 12.35,

"releasedate": "2013-03-01T04:10:00.000Z",

"developer": "lero dvpl",

"editor": "lero edts",

"genre": "horror",

"rating": 0,

"imgsrc": "https://http2.mlstatic.com/sonic-forces-ps4-midia-fisica-lacrado-novo-original-D\_NQ\_NP\_658382-MLB26464235963\_112017-F.webp",

"\_\_v": 0

},

{

"\_id": "5b22f8c286969c3646ef8c41",

"name": "Roller Coaster",

"synopse": "Construa seu parque",

"tags": "Construcao",

"price": 12.35,

"releasedate": "2013-03-01T04:10:00.000Z",

"developer": "lero dvpl",

"editor": "lero edts",

"genre": "Construçao",

"rating": 0,

"imgsrc": "https://http2.mlstatic.com/rollercoaster-tycoon-world-dvd-pc-midia-fisica-D\_NQ\_NP\_782650-MLB26799855909\_022018-F.webp",

"\_\_v": 0

}

]

#### /games/:gameId

* **GET**

get single game data. gameId's data.

$ curl -s "localhost:8080/games/5b22f8c286969c3646ef8c41" -H "Content-Type: application/json" | jq

{

"\_id": "5b22f8c286969c3646ef8c41",

"name": "Roller Coaster",

"synopse": "Construa seu parque",

"tags": "Construcao",

"price": 12.35,

"releasedate": "2013-03-01T04:10:00.000Z",

"developer": "lero dvpl",

"editor": "lero edts",

"genre": "Construçao",

"rating": 0,

"imgsrc": "https://http2.mlstatic.com/rollercoaster-tycoon-world-dvd-pc-midia-fisica-D\_NQ\_NP\_782650-MLB26799855909\_022018-F.webp",

"\_\_v": 0

}

### /games/:gameId

* **DELETE**

delete specified games data from db.

$ curl -s -XDELETE "localhost:8080/games/5b22f8c286969c3646ef8c41" -H "Content-Type: application/json" | jq

{

"message": "Game deleted."

}

#### /genres

* **GET**

get all genres from games

$ curl -s "localhost:8080/genres" -H "Content-Type: application/json" | jq

[

"horror",

"Corrida"

]

3.3.6.1 Routes

3.3.6.1.1 GameRoutes.js

|  |
| --- |
| 'use strict'; |
|  |  |
|  | module.exports = function(app) { |
|  | var Game = require('../controllers/GameController'); |
|  |  |
|  | //game routes "for bunch of data" |
|  | app.route('/games') |
|  | .get(Game.list\_all\_games) |
|  | .post(Game.insert\_a\_game); |
|  |  |
|  | //game routes "for sigle data" |
|  | app.route('/games/:gameId') |
|  | .get(Game.read\_a\_game) |
|  | .put(Game.update\_a\_game) |
|  | .delete(Game.delete\_a\_game); |
|  |  |
|  | //game routes "for bunch of data" |
|  | app.route('/genres') |
|  | .get(Game.list\_all\_genres); |
|  | }; |

3.3.6.1.2 UserRoutes.js

|  |
| --- |
| 'use strict'; |
|  |  |
|  | module.exports = function(app) { |
|  | var User = require('../controllers/UserController'); |
|  |  |
|  | //user routes "for bunch of data" |
|  | app.route('/users') |
|  | .get(User.list\_all\_users) |
|  | .post(User.insert\_a\_user); |
|  |  |
|  | //user routes "for sigle data" |
|  | app.route('/users/:userId') |
|  | .get(User.read\_a\_user) |
|  | .put(User.update\_a\_user) |
|  | .delete(User.delete\_a\_user); |
|  |  |
|  | //login data |
|  | app.route('/login') |
|  | .post(User.login); |
|  | }; |

3.3.6.2 Models

3.3.6.2.1 GameModel.js

|  |
| --- |
| 'use strict'; |
|  |  |
|  | var mongoose = require('mongoose'); |
|  | var Schema = mongoose.Schema; |
|  |  |
|  | //creating game's base struct |
|  | var GameSchema = new Schema({ |
|  | name: { |
|  | type: String, |
|  | required: 'Enter the game\'s name' |
|  | }, |
|  | synopse: { |
|  | type: String |
|  | }, |
|  | tags: { |
|  | type: String |
|  | }, |
|  | price: { |
|  | type: Number |
|  | }, |
|  | releasedate: { |
|  | type: Date |
|  | }, |
|  | developer: { |
|  | type: String |
|  | }, |
|  | editor: { |
|  | type: String |
|  | }, |
|  | genre: { |
|  | type: String |
|  | }, |
|  | rating: { |
|  | type: Number |
|  | }, |
|  | imgsrc: { |
|  | type: String |
|  | } |
|  | }); |
|  |  |
|  | module.exports = mongoose.model('Games', GameSchema); |

3.3.6.2.2 UserModel.js

|  |
| --- |
| 'use strict'; |
|  |  |
|  | var mongoose = require('mongoose'); |
|  | var Schema = mongoose.Schema; |
|  |  |
|  | //creating user's base struct |
|  | var UserSchema = new Schema({ |
|  | email: { |
|  | type: String, |
|  | required: 'Enter the user\'s email' |
|  | }, |
|  | username: { |
|  | type: String |
|  | }, |
|  | password: { |
|  | type: String |
|  | }, |
|  | type: { |
|  | type: Number |
|  | } |
|  | }); |
|  |  |
|  | module.exports = mongoose.model('Users', UserSchema); |

3.3.6.3 Controllers

3.3.6.3.1 GameController.js

|  |
| --- |
| 'use strict'; |
|  |  |
|  | var mongoose = require('mongoose'); |
|  | Game = mongoose.model('Games'); |
|  |  |
|  | //GET at /games |
|  | exports.list\_all\_games = function(req, res) { |
|  | Game.find({}, function(err, game) { |
|  | if (err){ |
|  | res.status(500).send(err); |
|  | } else { |
|  | res.status(200).json(game); |
|  | } |
|  | }); |
|  | }; |
|  |  |
|  | //POST at /games |
|  | exports.insert\_a\_game = function(req, res) { |
|  | var new\_game = new Game(req.body); |
|  | new\_game.save(function(err, game){ |
|  | if (err){ |
|  | res.status(500).send(err); |
|  | } |
|  | res.status(201).json(game); |
|  | }); |
|  | }; |
|  |  |
|  | //GET at /games/:gameId |
|  | exports.read\_a\_game = function(req, res) { |
|  | Game.findById(req.params.gameId, function(err, game) { |
|  | if (err){ |
|  | res.status(500).send(err); |
|  | } else { |
|  | res.status(200).json(game); |
|  | } |
|  | }); |
|  | }; |
|  |  |
|  | //PUT at /games/:gameId |
|  | exports.update\_a\_game = function(req, res) { |
|  | Game.findOneAndUpdate( |
|  | {\_id: req.params.gameId}, req.body, {new: true}, function(err, game){ |
|  | if (err){ |
|  | res.status(500).send(err); |
|  | } else { |
|  |  |
|  | res.status(200).json(game); |
|  | } |
|  | }) |
|  | } |
|  |  |
|  | //DELETE at /games/:gameId |
|  | exports.delete\_a\_game = function(req, res) { |
|  | Game.remove({\_id: req.params.gameId}, function(err, game){ |
|  | if (err){ |
|  | res.status(500).send(err); |
|  | } else { |
|  | res.status(200).json({message: "Game deleted."}); |
|  | } |
|  | }); |
|  | }; |
|  |  |
|  | //GET at /genres |
|  | exports.list\_all\_genres = function(req, res) { |
|  | Game.distinct("genre", function(err, game) { |
|  | if (err){ |
|  | res.status(500).send(err); |
|  | } else { |
|  | res.status(200).json(game); |
|  | } |
|  | }); |
|  | }; |

**3.3.6.3.2 UserController.js**

|  |
| --- |
| 'use strict'; |
|  |  |
|  | var mongoose = require('mongoose'); |
|  | User = mongoose.model('Users'); |
|  |  |
|  | //GET at /users |
|  | exports.list\_all\_users = function(req, res) { |
|  | console.log('alan') |
|  | User.find({}, function(err, user) { |
|  | if (err){ |
|  | res.status(500).send(err); |
|  | } else { |
|  | res.status(200).json(user); |
|  | } |
|  | }); |
|  | }; |
|  |  |
|  | //POST at /users |
|  | exports.insert\_a\_user = function(req, res) { |
|  | var new\_user = new User(req.body); |
|  | new\_user.save(function(err, user){ |
|  | if (err){ |
|  | res.status(500).send(err); |
|  | } else { |
|  | res.status(201).json({"id": new\_user.\_id, |
|  | "email": new\_user.email, |
|  | "username": new\_user.username, |
|  | "type": new\_user.type}); |
|  | } |
|  | }); |
|  | }; |
|  |  |
|  | //GET at /users/:userId |
|  | exports.read\_a\_user = function(req, res) { |
|  | User.findById(req.params.userId, function(err, user) { |
|  | if (err){ |
|  | res.status(500).send(err); |
|  | } else { |
|  | res.status(200).json(user); |
|  | } |
|  | }); |
|  | }; |
|  |  |
|  | //PUT at /users/:userId |
|  | exports.update\_a\_user = function(req, res) { |
|  | User.findOneAndUpdate( |
|  | {\_id: req.params.userId}, req.body, {new: true}, function(err, user){ |
|  | if (err){ |
|  | res.status(500).send(err); |
|  | } else { |
|  | res.status(200).json(user); |
|  | } |
|  | }) |
|  | } |
|  |  |
|  | //DELETE at /users/:userId |
|  | exports.delete\_a\_user = function(req, res) { |
|  | User.remove({\_id: req.params.userId}, function(err, user){ |
|  | if (err){ |
|  | res.status(500).send(err); |
|  | } else { |
|  | res.status(200).json({message: "User deleted."}); |
|  | } |
|  | }); |
|  | }; |
|  |  |
|  | //Login |
|  | exports.login = function(req, res) { |
|  | User.findOne({username: req.body.username}, function(err, user){ |
|  | if(err){ |
|  | res.status(500).send(err); |
|  | } else { |
|  | if(user != null) |
|  | { |
|  | if(req.body.password == user.password) |
|  | { |
|  | res.status(200).json(user); |
|  | } else { |
|  | res.status(401).json({error: "username or password incorrect."}) |
|  | } |
|  | } else { |
|  | res.status(401).json({error: "username or password incorrect."}) |
|  | } |
|  |  |
|  | } |
|  | }) |
|  | } |