

**Sistema de Controle**  
*Catálogo Filatélico*  
*Selos*

Solução simples e prática para colecionadores de Selos que desejam manter uma organização e controle em sua coleção.

Page: <https://vjungles.github.io/controle-de-selos/>

**1. Visão geral do sistema**



O objetivo principal deste sistema é facilitar a organização e a coleção de selos filatélicos.

O software é um Sistema Web que gerencia duas entidades: "Album" e "Selo".  
Permite aos usuários criar, visualizar, editar e excluir álbuns e selos.  
Cada álbum pode conter vários selos e cada selo pertence somente a um álbum.

Apresenta uma interface gráfica intuitiva e fácil de usar, com as seguintes funcionalidades principais:

*Login*  
*Criar um novo Álbum*  
*Editar um Álbum*  
*Excluir um Álbum*  
*Listar todos os Álbuns*  
*Adicionar um novo Selo a um Álbum*  
*Editar um Selo de um Álbum*  
*Excluir um Selo de um Álbum*  
*Listar todos os Selos de um Álbum*

Permite aos usuários visualizar todos os álbuns e selos existentes e atualizar suas informações a qualquer momento. Além disso, os usuários podem facilmente excluir um álbum ou selo que não precisam mais.

O frontend da aplicação foi desenvolvido com Angular e o backend foi simulado pela implementação de uma API Fake, usando o JSON Server.

## 2. Entidades:

- Album** Representa um álbum de selos.  
Atributos:
- id - identificador exclusivo do álbum.
  - name - nome do álbum.
  - description - descrição do álbum.
  - userId - id do usuário que criou o álbum..
  - selos - selos do álbum.
- Selo** Representa um selo.  
Atributos:
- id - identificador exclusivo do selo.
  - name - nome do selo.
  - tipo - tipo do selo.
  - description - descrição do selo.
  - year - ano de fabricação do selo.
  - country - país do selo.
  - stamp - se possuir carimbo.
  - filigram - se o selo possui marca d'água.
  - type - tipo do selo.
  - link - url na internet.
- User** Representa um usuário.  
Atributos:
- id- identificador usuário.
  - username - nome do usuário.
  - password - senha do usuário.

## 3. Funcionalidades:

### Login

Permite ao usuário logar no sistema.

Parâmetros de entrada:

- usuario
- senha

Comportamento: Logar no sistema.



USUÁRIO:

SENHA:

Acessar



[Tudo sobre Filatelia](#)

Sistema de Controle - Catálogo Filatélico - Selos  
João Valtter Jungles Junior  
UTPR - Frameworks Web

Retorno: Usuário logado.

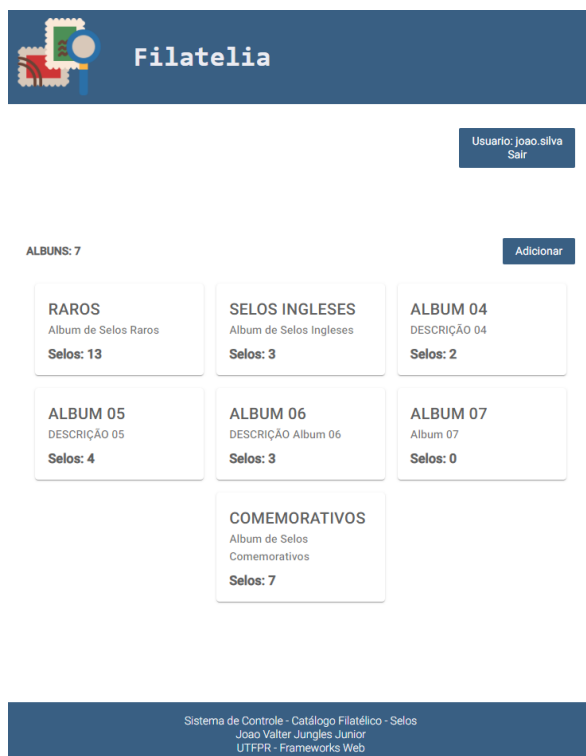
## Lista de álbuns

Permite ao usuário visualizar todos os álbuns existentes.

Parâmetros de entrada:

- id do usuário logado

Comportamento: Retorna uma lista com todos os álbuns existentes para o usuário.



Retorno: Lista de álbuns.

## Novo álbum

Permite ao usuário criar um novo álbum.

Parâmetros de entrada:

- nome
- descricao

Comportamento: Cria um novo álbum com os dados fornecidos.

O formulário 'Novo Álbum' possui o seguinte layout:

- Título: **Novo Álbum**
- Campos de entrada:
  - NOME \* :** Campo de texto obrigatório.
  - DESCRIÇÃO:** Área de texto para descrição.
- Botões de ação: **Fechar** e **Salvar**.

Retorno: álbum criado.

### Editar álbum

Permite ao usuário atualizar os dados de um álbum existente.

Parâmetros de entrada:

- albumId
- nome
- descricao

Comportamento: Atualiza o álbum com os dados fornecidos.



Formulário de Edição de Álbum. O formulário possui o título "Editar Álbum". Abaixo dele, há o campo "NOME\*" com o valor "SELOS INGLESES". Abaixo disso, há o campo "DESCRIÇÃO:" com o valor "Album de Selos Ingleses". No rodapé do formulário, há dois botões: "Fechar" e "Salvar".

Retorno: álbum atualizado.

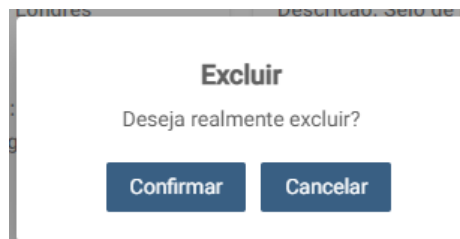
### Excluir álbum

Permite ao usuário excluir um álbum existente.

Parâmetros de entrada:

- albumId

Comportamento: Exclui o álbum com o id fornecido.



Modal de Exclusão. O modal possui o título "Excluir". Abaixo dele, há o texto "Deseja realmente excluir?". No rodapé do modal, há dois botões: "Confirmar" e "Cancelar".

Retorno: álbum excluído.

## Novo Selo

Permite ao usuário criar um novo selo a um álbum existente.

Parâmetros de entrada:

- tipo
- nome
- descricao
- anoFabricacao
- pais
- carimbo
- filigrama
- link
- albumId

Comportamento: Cria um novo selo com os dados fornecidos.

### Novo Selo

TIPO:

NOME\*:

DESCRIÇÃO:

ANO DE FABRICAÇÃO\*:

PAÍS:

CARIMBO: ☐

FILIGRAMA: ☐ SIM ☐ NÃO

LINK\*:

Salvar

Fechar

Retorno: selo criado.

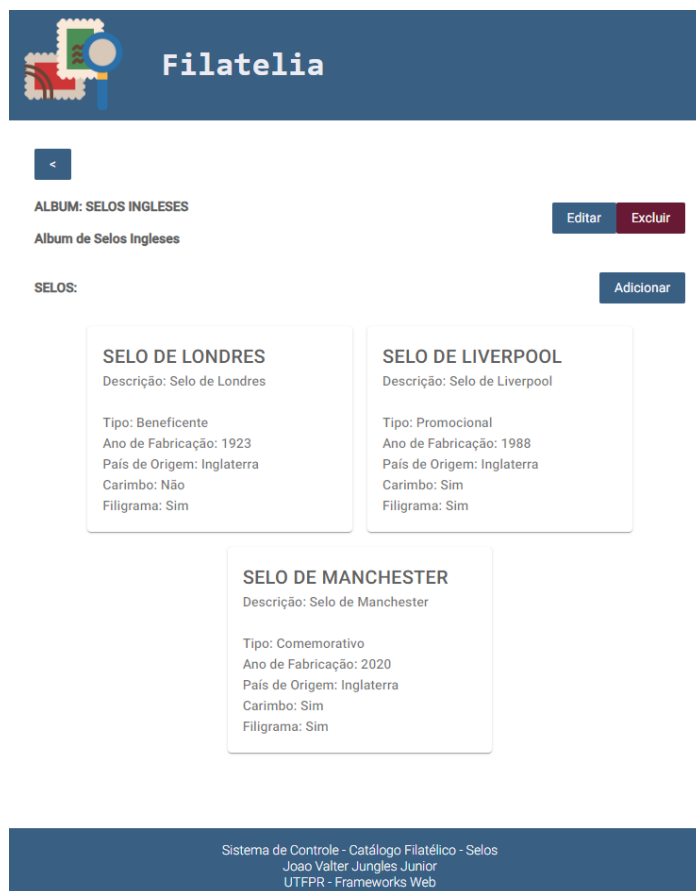
## Listar Selos

Permite ao usuário visualizar todos os selos de um álbum existente.

Parâmetros de entrada:

- albumId

Comportamento: Retorna uma lista com todos os selos de um álbum existente.



Retorno: Lista de selos.

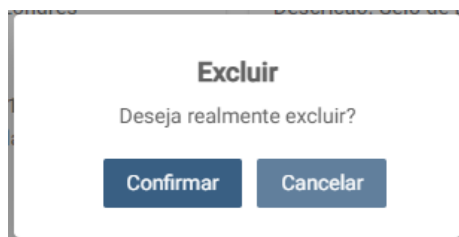
## Excluir Selo

Permite ao usuário excluir um selo de um álbum existente.

Parâmetros de entrada:

- seloid

Comportamento: Exclui o selo com o id fornecido.



Retorno: selo excluído.

## Editar Selo

Permite ao usuário atualizar os dados de um selo de um álbum existente.

Parâmetros de entrada

- seloid
- nome
- tipo
- descricao
- anoFabricacao
- pais
- carimbo
- filigrama
- albumId

Comportamento: Atualiza o selo com os dados fornecidos.

### Editar Selo

**TIPO:**

Beneficente

**NOME \*:**

SELO DE LONDRES

**DESCRIÇÃO:**

Selo de Londres

**ANO DE FABRICAÇÃO \*:**

1923

**PAÍS:**

Inglaterra

**CARIMBO:** ☒

**FILIGRAMA:** ☒ SIM ☐ NÃO

**LINK \*:**

https://www.selosefilatelia.com

Salvar

Excluir

Fechar

Retorno: selo atualizado.

#### 4. Mensagens:

##### Usuário não encontrado

USUÁRIO:

SENHA:

Usuário não encontrado

Acessar

##### Senha inválida

USUÁRIO:

SENHA:

Senha inválida

Acessar

##### Álbum salvo com sucesso

Album salvo com sucesso

ok

##### Selo salvo com sucesso

Selo salvo com sucesso

ok

##### Álbum excluído com sucesso

Album excluido com sucesso

ok

##### Selo excluído com sucesso

Selo excluido com sucesso

ok

##### Formato do link

LINK \*:

Necessário formato de link - http://

##### Formato do ano de fabricação

ANO DE FABRICAÇÃO \*:

Necessario formato YYYY (exemplo: 2023)

#### 5. Protótipo:

- Figma

<https://www.figma.com/proto/XarePPgeODdsLiyjL1r02e/controle-de-selos?type=design&node-id=115-3&scaling=min-zoom&page-id=0%3A1&starting-point-node-id=115%3A3>

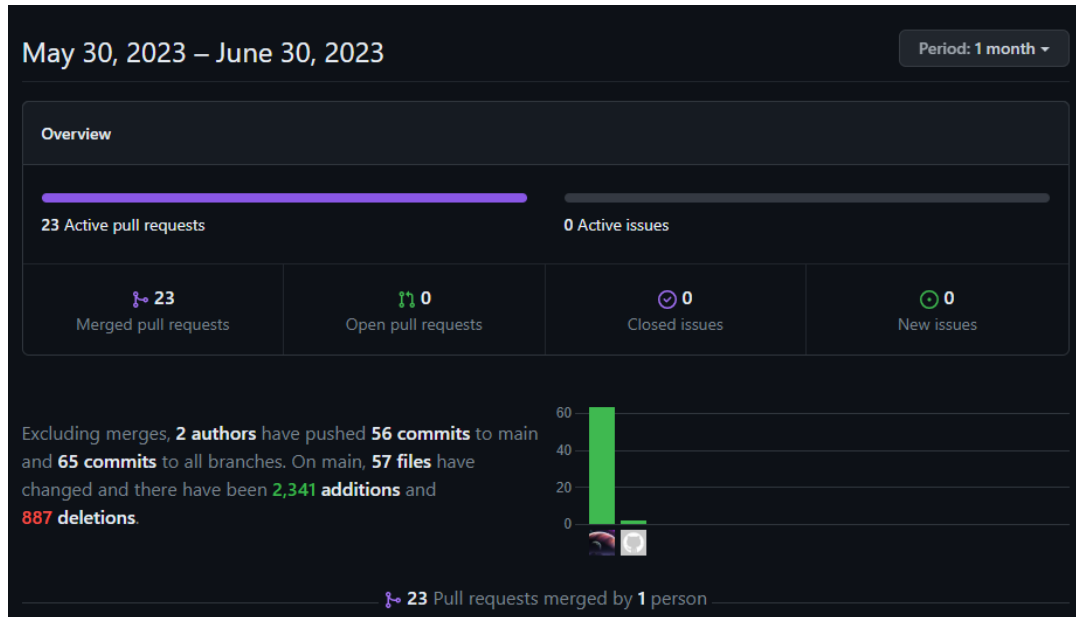


## 6. Repositório:

- Github

<https://github.com/jvjungles/controle-de-selos>

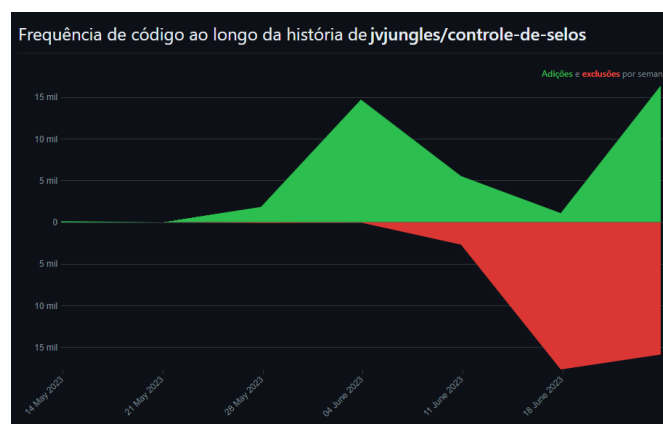
- Overview (last month)



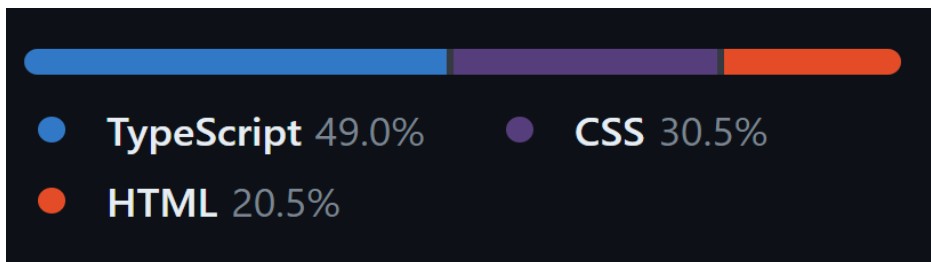
- Network



- Code frequency



## 7. Linguagens de programação:



## 8. Tecnologias:

Node v18.16.0

<https://nodejs.org/>



Npm v9.5.1

<https://www.npmjs.com/>



Typescript v5.0.4

<https://www.typescriptlang.org/>



Angular v16.0.0

<https://angular.io/>



Angular Material v16.0.1

<https://material.angular.io/>



Concurrently v8.2.0

<https://github.com/open-cli-tools/concurrently>

open-cli-tools/  
**concurrently**



Json Server v0.17.3

<https://www.npmjs.com/package/json-server>

**JSON Server**

Angular-cli-ghpages v1.0.0

<https://pages.github.com/>



## 9. Manual de execução

- Clonar o repositório com *git clone*
- Fazer checkout no branch *develop* que contém as modificações mais recentes
- Abrir o projeto no editor Visual Studio Code (VS Code)
- Abrir um terminal pelo VSCode apontando para o diretório raiz do projeto
- Instalar as dependências contidas no *package.json*
  - Comando: *npm install*
- Para rodar a aplicação, temos duas opções:
  - Opção 01 - rodar local:
    - Executar o comando *npm start* que automaticamente executará os comandos *ng serve* e *npm run json:server* através do *concurrently*
      - Por padrão, a aplicação JSON Server executa no endereço *localhost:3000*
      - A aplicação executa no endereço *localhost:4200*
    - Opção 02 - rodar pelo Github pages:
      - Executar a API Fake (JSON Server)
        - Execução o comando *npm run json:server*
        - Por padrão, a aplicação JSON Server executa no endereço *localhost:3000*
      - Acessar a aplicação
        - <https://jvjungles.github.io/controle-de-selos/>

## 10. Json Server

<http://localhost:3000/>

### Resources

[/users](#) 4x  
[/albums](#) 18x

To access and modify resources, you can use any HTTP method:

GET POST PUT PATCH DELETE OPTIONS

undefined

<http://localhost:3000/users>

```
[
  {
    "id": "2",
    "username": "neymar.jr",
    "password": "paris"
  },
  {
    "id": "3",
    "username": "ayrton.senna",
    "password": "ferrari"
  },
  {
    "id": "1",
    "username": "joao.silva",
    "password": "123456"
  },
  {
    "id": "4",
    "username": "admin",
    "password": "admin"
  }
]
```

<http://localhost:3000/albums>

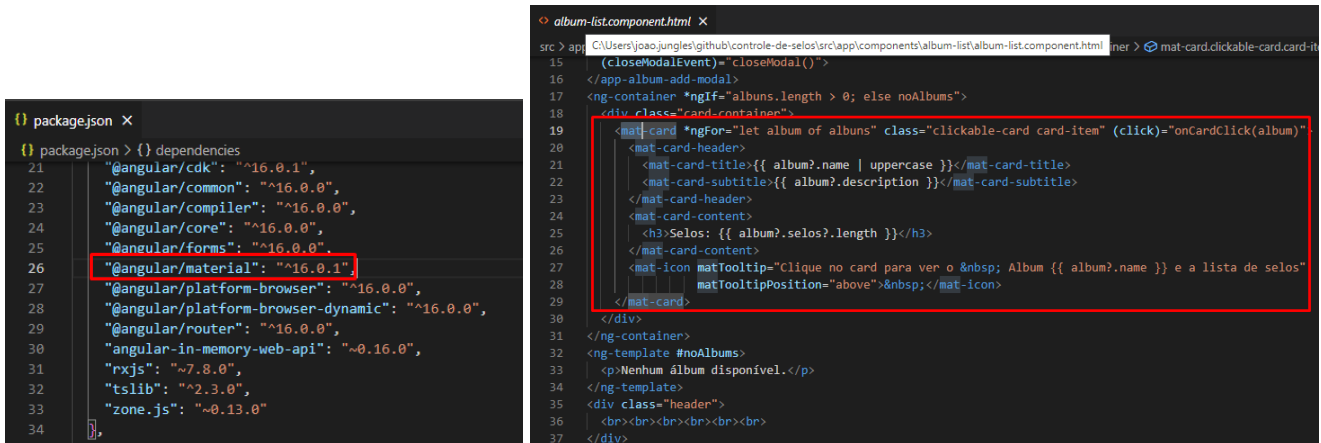
```
[
  {
    "name": "RAROS",
    "description": "Album de Selos Raros",
    "userId": "1",
    "id": 1,
    "selos": [
      {
        "id": 1,
        "name": "SELO 01",
        "description": "DESCRIÇÃO selo 01 ",
        "type": "Outros",
        "year": "1988",
        "country": "Brasil",
        "stamp": true,
        "filigram": "Não",
        "link": ""
      },
      {
        "id": 2,
        "name": "SELO 02",
        "description": "DESCRIÇÃO selo 02",
        "type": "Comemorativo",
        "year": "2020",
        "country": "Brasil",
        "stamp": true,
        "filigram": "Sim"
      }
    ]
  }
]
```

## 11. Check List:

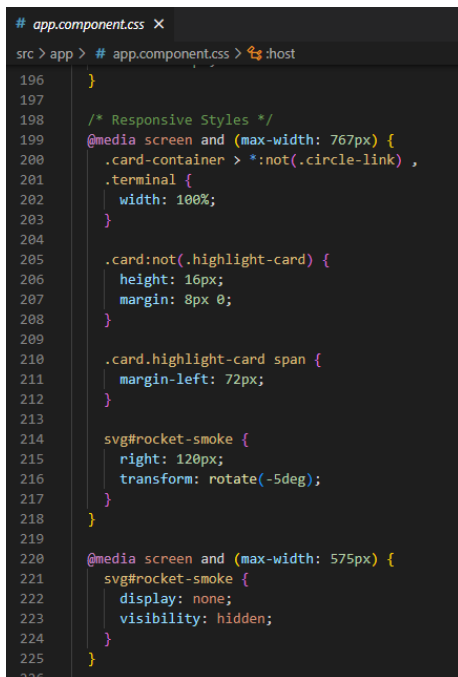
- [X] Criar o repositório no GitHub com a estrutura do Gitflow, ou seja, branches main e develop.

<https://github.com/jvjungles/controle-de-selos>

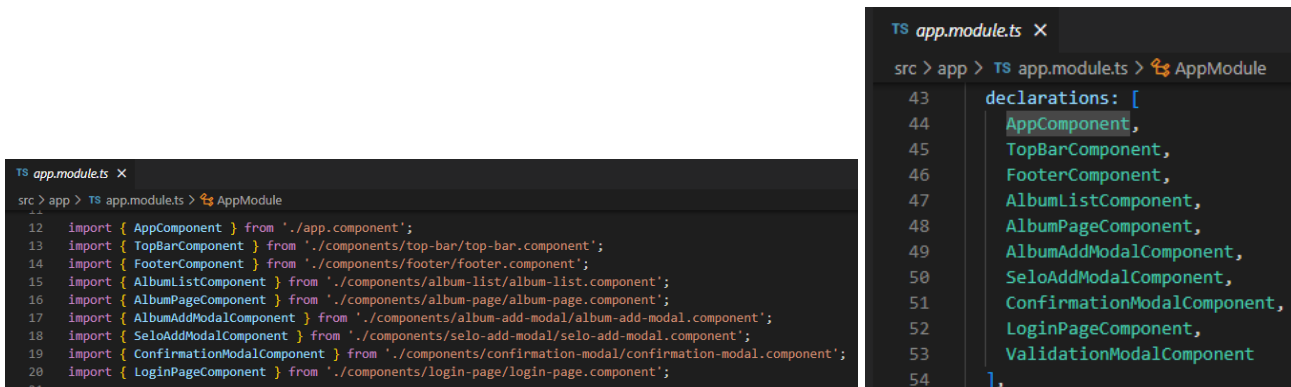
- [X] Usar componentes de algum framework CSS (Bootstrap, Materialize ou outro)

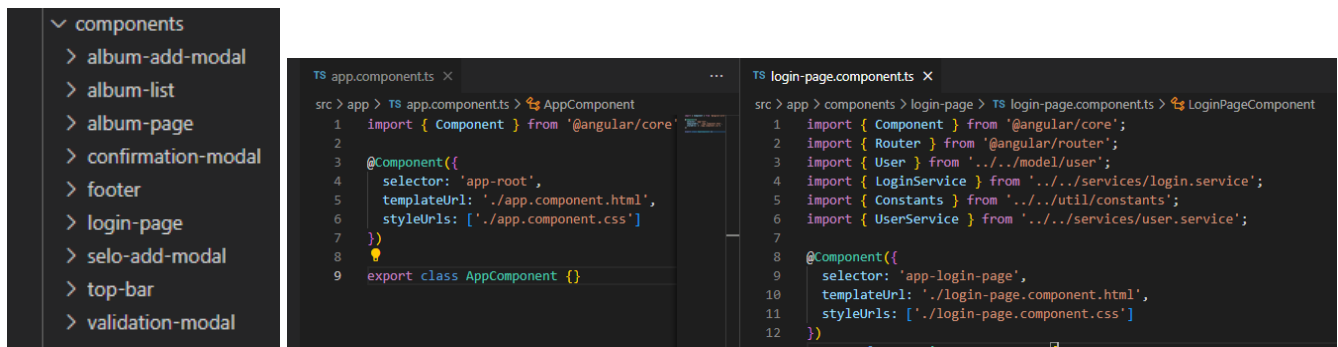


- [X] Apresentar as telas com layout responsivo usando ou não algum framework CSS.

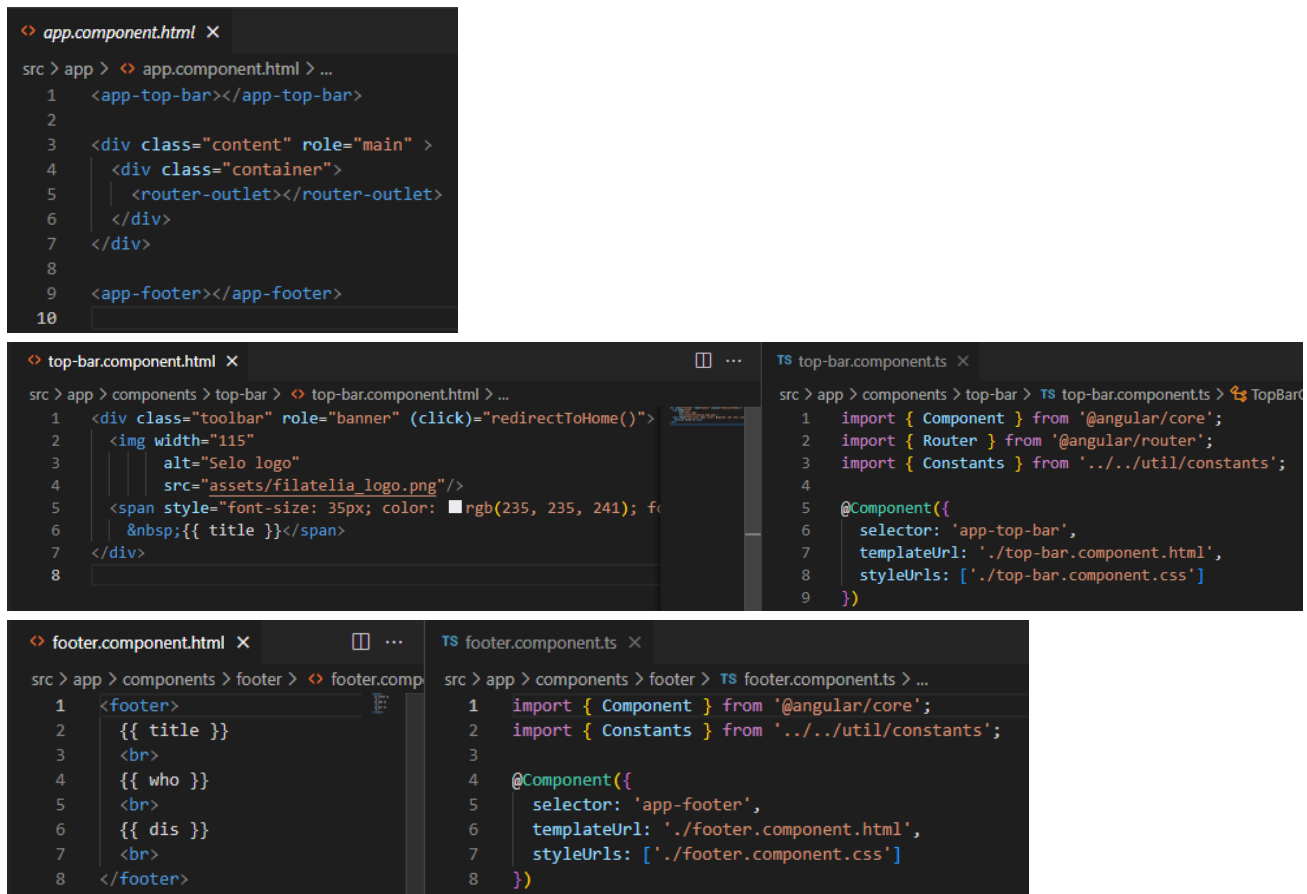


- [X] Construir páginas web com o conceito de componentes.

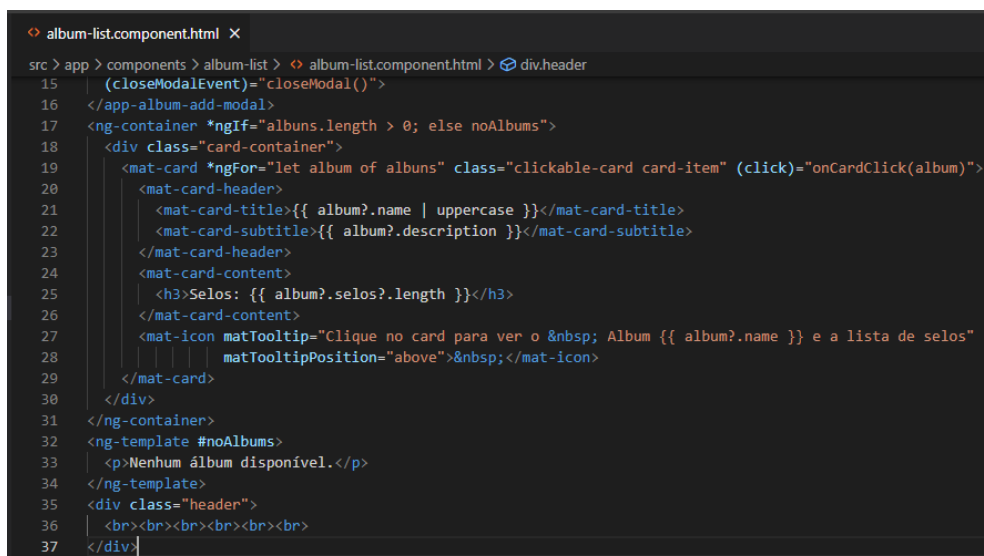




- [X] Criar o layout da aplicação com componentes, ou seja, o cabeçalho e rodapé precisam ser componentes.



- [X] Usar pelo menos dois tipos de data-binding (Interpolation, Property Binding, Event Binding e Two Way Data Binding).



```

album-add-modal.component.html X
src > app > components > album-add-modal > album-add-modal.component.html > div.modal-overlay > div.modal-content > div.button-container
1 <div class="modal-overlay" [class.show]="showModal">
2 <div class="modal-content">
3 <h1>{{ albumTitle }} Album</h1>
4 <div class="input-container">
5 <label for="nome">Nome * </label>
6 <input class="custom-text" type="text" id="nome" [(ngModel)]="nome" (input)="isNomeValid()"
7 | (mouseenter)="showTooltip = true" (mouseleave)="showTooltip = false" />
8 <span class="tooltip" *ngIf="showTooltip" role="tooltip">Será salvo em letras maiúsculas</span>
9 </div>
10 <div class="input-container">
11 <label for="descricao">Descrição:</label>
12 <textarea class="custom-textarea" id="descricao" [(ngModel)]="descricao"></textarea>
13 </div>
14 <div class="button-container">
15 <button class="close-button" (click)="closeModal()">Fechar</button>
16 <button class="save-button" (click)="saveModal()" [disabled]="!isNomeValid()">Salvar</button>
17 </div>
18 </div>
19 </div>

```

```

album-add-modal.component.html X
src > app > components > album-add-modal > album-add-modal.component.html > div.modal-overlay > div.modal-content > div.input-container
1 <div class="modal-overlay" [class.show]="showModal">
2 <div class="modal-content">
3 <h1>{{ albumTitle }} Album</h1>
4 <div class="input-container">
5 <label for="nome">Nome * </label>
6 <input class="custom-text" type="text" id="nome" [(ngModel)]="nome" (input)="isNomeValid()"
7 | (mouseenter)="showTooltip = true" (mouseleave)="showTooltip = false" />
8 <span class="tooltip" *ngIf="showTooltip" role="tooltip">Será salvo em letras maiúsculas</span>
9 </div>
10 <div class="input-container">
11 <label for="descricao">Descrição:</label>
12 <textarea class="custom-textarea" id="descricao" [(ngModel)]="descricao"></textarea>
13 </div>
14 <div class="button-container">
15 <button class="close-button" (click)="closeModal()">Fechar</button>
16 <button class="save-button" (click)="saveModal()" [disabled]="!isNomeValid()">Salvar</button>
17 </div>
18 </div>
19 </div>
20

```

```

TS album-add-modal.component.ts X
src > app > components > album-add-modal > TS album-add-modal.component.ts
49
50 closeModal() {
51   if (this.albumTitle !== 'Editar') {
52     this.nome = '';
53     this.descricao = '';
54   }
55   this.showModal = false;
56   this.closeModalEvent.emit();
57 }
58
59 saveModal() {
60   if (this.albumid === -1) {
61     this.save();
62   } else {
63     this.update();
64   }
65 }
66

```

- [X] Passar dados via hierarquia de componentes, ou seja, usando @Input ou @Output.

```

TS selo-add-modal.component.ts X
src > app > components > selo-add-modal > TS selo-add-modal.component.ts > SeloAddModalComponent > closeModalEvent
1 import { Component, Input, Output, EventEmitter } from '@angular/core';
2 import { MatDialog } from '@angular/material/dialog';
3 import { ConfirmationModalComponent } from '../confirmation-modal/confirmation-modal.component';
4 import { ValidationModalComponent } from '../validation-modal/validation-modal.component';
5 import { AlbumService } from '../../services/album.service';
6 import { SeloTypes } from '../../model/selotype';
7 import { Album } from '../../model/album';
8 import { Selo } from '../../model/selo';
9 import { Constants } from '../../util/constants';
10
11 @Component({
12   selector: 'app-selo-add-modal',
13   templateUrl: './selo-add-modal.component.html',
14   styleUrls: ['./selo-add-modal.component.css']
15 })
16 export class SeloAddModalComponent {
17
18   constructor(private dialog: MatDialog, private service: AlbumService) { }
19
20   @Input() showSeloModal: boolean = false;
21   @Input() seloTitle: String = '';
22   @Input() albumid: number = -1;
23   @Input() selo: Selo | undefined;
24   @Output() closeModalEvent = new EventEmitter<void>();
25

```

```

TS album-add-modal.component.ts X
src > app > components > album-add-modal > TS album-add-modal.component.ts > AlbumAddModalComponent > closeModal
1 import { Component, Input, Output, EventEmitter } from '@angular/core';
2 import { ActivatedRoute } from '@angular/router';
3 import { MatDialog } from '@angular/material/dialog';
4 import { AlbumService } from '../../services/album.service';
5 import { UserService } from '../../services/user.service';
6 import { ValidationModalComponent } from '../validation-modal/validation-modal.component';
7 import { Album } from '../../model/album';
8 import { User } from '../../model/user';
9 import { Constants } from '../../util/constants';
10
11 @Component({
12   selector: 'app-album-add-modal',
13   templateUrl: './album-add-modal.component.html',
14   styleUrls: ['./album-add-modal.component.css']
15 })
16 export class AlbumAddModalComponent {
17
18   constructor(private dialog: MatDialog, private userService: UserService, private route: ActivatedRoute, private service: AlbumService) { }
19
20   @Input() showModal: boolean = false;
21   @Input() albumTitle: String = '';
22   @Input() albumid: number = -1;
23   @Output() closeModalEvent = new EventEmitter<void>();
24

```

```

album-page.component.html X
src > app > components > album-page > album-page.component.html > div >
23     </div>
24     <app-album-add-modal
25       [showModal]="showModal"
26       [albumTitle]="albumTitle"
27       [albumid]="albumid"
28       (closeModalEvent)="closeModal()">
29     </app-album-add-modal>
30     <app-selo-add-modal
31       [showSeloModal]="showSeloModal"
32       [seloTitle]="seloTitle"
33       [albumid]="albumid"
34       [selo]="selo"
35       (closeModalEvent)="closeSeloModal()">
36     </app-selo-add-modal>
37     <br>
38   </div>
39

```

- [X] Mapear componentes à rotas no módulo de rotas.

```

26
27 @NgModule({
28   imports: [
29     BrowserModule,
30     ReactiveFormsModule,
31     MatSlideToggleModule,
32     FormsModule,
33     MatCardModule,
34     MatDialogModule,
35     MatTooltipModule,
36     HttpClientModule,
37     RouterModule.forRoot([
38       { path: '', component: LoginPageComponent }, // pagina login
39       { path: 'album-list', component: AlbumListComponent }, // pagina inicial - lista de albuns
40       { path: 'album-page', component: AlbumPageComponent }, // pagina do album - lista de selos
41     ])
42   ],
43 })

```

- [X] Criar navegação entre páginas por meio de rotas.

TS album-page.component.ts
src > app > components > album-page > TS album-page.component.ts > AlbumPageComponent
90 redirectToHome() {
91 this.getNomeUsuario();
92 if (this.nomeUsuario === '') {
93 this.router.navigate(['/']);
94 } else {
95 this.router.navigate(['/album-list']);
96 }
97 }
98
99 back() {
100 this.router.navigate(['/album-list']);
101 }
102

TS login-page.component.ts X
src > app > components > login-page > TS login-page.component.ts > LoginPageComponent
28
29 async login() {
30 this.usuarioEncontrado = true;
31 this.senhaValida = true;
32 try {
33 const user: User = await this.service.getByNome(this.username);
34 if (user) {
35 if (user.password === this.password) {
36 localStorage.setItem(Constants.USER, this.username);
37 this.userService.setUser(user);
38 this.router.navigate(['/album-list']);
39 } else {
40 this.senhaValida = false;
41 }
42 } else {
43 this.usuarioEncontrado = false;
44 }
45 } catch (error) {
46 this.usuarioEncontrado = false;
47 }
48 }
49

- [X] Passar dados entre componentes que representam diferentes telas via parâmetros de rotas.

TS album-list.component.ts X
src > app > components > album-list > TS album-list.component.ts > AlbumListComponent
22
23 ngOnInit() {
24 this.findAlbumList();
25 const userId = localStorage.getItem(Constants.USER);
26 this.nomeUsuario = userId ? String(userId) : '';
27 this.albumQtde = +localStorage.getItem(Constants.ALBUNS\_SIZE)!;
28 }
29
30 onCardClick(album: any) {
31 this.router.navigate(['/album-page', album]);
32 }
33

TS album-page.component.ts X
src > app > components > album-page > TS album-page.component.ts > AlbumPageComponent
30
31 ngOnInit() {
32 this.route.params.subscribe(params => {
33 this.service.getById(+params['id']).subscribe(album => {
34 this.album = album;
35 });
36 });
37 }
38

- [X] Validar campos do formulário com REGEX e apresentar os erros.

```
TS selo-add-modal.component.ts
src > app > components > selo-add-modal > TS selo-add-modal.component.ts > SeloAddModalComponent > 123
124   isYearValid(): boolean {
125     const pattern = /[0-9]{4}/;
126     return this.selo?.year !== undefined && pattern.test(this.selo.year);
127   }
128
129   isValidLink(): boolean {
130     const regex = /^http(s)?:\//\S+$/;
131     return this.selo?.link !== undefined && regex.test(this.selo?.link);
132   }
133 }

selo-add-modal.component.html X
src > app > components > selo-add-modal > selo-add-modal.component.html > div.modal-overlay.modal-auto-height > div.modal-content > div.input-cont
19 </div>
20 <div class="input-container">
21   <label for="anoFabricacao">Ano de Fabricação * :</label>
22   <input class="custom-text" type="text" id="anoFabricacao" *ngIf="selo" [(ngModel)]="selo.year" required />
23 </div>
24 <div class="error-message error-message-small" *ngIf="selo && !isYearValid() && (selo.year || selo.year === '')">
25   Necessario formato YYYY (exemplo: 2023) <br><br><br><br>
26 </div>

selo-add-modal.component.html X
src > app > components > selo-add-modal > selo-add-modal.component.html > div.modal-overlay.modal-auto-height > div.modal-content
42 <div class="input-container">
43   <label for="nome">Link * :</label>
44   <input class="custom-text" type="text" id="link" *ngIf="selo" [(ngModel)]="selo.link" required />
45 </div>
46 <div class="error-message error-message-small" *ngIf="selo?.link && !isValidLink()">
47   Necessário formato de link - http://
48 </div>
```

- [X] Desabilitar o botão de submit enquanto o formulário está inválido.

```
album-add-modal.component.html X
src > app > components > album-add-modal > album-add-modal.component.html > div.modal-overlay > div.modal-content > div.button
1 <div class="modal-overlay" [class.show]="showModal">
2   <div class="modal-content">
3     <h1>{{ albumTitle }} Álbum</h1>
4     <div class="input-container">
5       <label for="nome">Nome * :</label>
6       <input class="custom-text" type="text" id="nome" [(ngModel)]="nome" (input)="isNomeValid()"
7         (mouseenter)="showTooltip = true" (mouseleave)="showTooltip = false" />
8       <span class="tooltip" *ngIf="showTooltip" role="tooltip">Será salvo em letras maiúsculas</span>
9     </div>
10    <div class="input-container">
11      <label for="descricao">Descrição:</label>
12      <textarea class="custom-textarea" id="descricao" [(ngModel)]="descricao"></textarea>
13    </div>
14    <div class="button-container">
15      <button class="close-button" (click)="closeModal()">Fechar</button>
16      <button class="save-button" (click)="saveModal()" [disabled]="!isNomeValid()">Salvar</button>
17    </div>
18  </div>
19 </div>

TS album-add-modal.component.ts X
src > app > components > album-add-modal > TS album-add-modal.component.ts
111
112   isNomeValid(): boolean {
113     return this.nome.trim().length > 0;
114   }
115 }
```

- [X] Cadastrar uma entidade no Web Storage.

```
TS shared.ts X
src > app > util > TS shared.ts > Shared
1 import { Constants } from './constants';
2
3 export class Shared {
4
5   constructor() {}
6
7   // WebStorage
8   public static initializeWebStorage(albums: any[]): void {
9     localStorage.setItem(Constants.ALBUNS_SIZE, albums.length.toString());
10    localStorage.setItem(Constants.ALBUNS, JSON.stringify(albums));
11  }
12 }
```



- [X] Fazer requisições a API com tratamento da resposta com Promises ( Fetch API, async/await ou lastValueFrom do RXJS).

```
TS login.service.ts
src > app > services > TS login.service.ts > ...
1 import { HttpClient, HttpHeaders } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { RoutesAPI } from '../util/routes-api';
4 import { User } from '../model/user';
5 import { lastValueFrom } from 'rxjs';
6
7 @Injectable()
8 export class LoginService {
9   URL = RoutesAPI.USERS;
10
11   httpOptions = {
12     headers: new HttpHeaders({ 'Content-Type': 'application/json' }),
13   };
14
15   constructor(private httpClient: HttpClient) {}
16
17   async listUser(): Promise<User[]> {
18     try {
19       const users: User[] = await lastValueFrom(
20         this.httpClient.get<User[]>(this.URL)
21       );
22       return users;
23     } catch (error) {
24       throw new Error('Erro ao obter a lista de usuários');
25     }
26   }
27
28   async getByName(usuario: string): Promise<User> {
29     try {
30       const users: User[] = await lastValueFrom(
31         this.httpClient.get<User[]>(`${this.URL}?username=${usuario}`)
32       );
33
34       if (users.length > 0) {
35         return users[0];
36       } else {
37         throw new Error('Usuário não encontrado');
38       }
39     } catch (error) {
40       throw error;
41     }
42   }
43 }
```

```
TS login-page.component.ts X
src > app > components > login-page > TS login-page.component.ts > LoginPageComponent >
28
29 async login() {
30   this.usuarioEncontrado = true;
31   this.senhaValida = true;
32   try {
33     const user: User = await this.service.getByName(this.username);
34     if (user) {
35       if (user.password === this.password) {
36         localStorage.setItem(Constants.USER, this.username);
37         this.userService.setUser(user);
38         this.router.navigate(['/album-list']);
39       } else {
40         this.senhaValida = false;
41       }
42     } else {
43       this.usuarioEncontrado = false;
44     }
45   } catch (error) {
46     this.usuarioEncontrado = false;
47   }
48 }
```

- [X] Cadastrar uma entidade no JSON Server.

```
() package.json
() package.json > {} devDependencies
1 {
2   "name": "controle-de-selos",
3   "version": "1.0.0",
4   "description": "Sistema de Controle - Catálogo Filatélico - Selos.",
5   "license": "MIT",
6   "scripts": {
7     "ng": "ng",
8     "start": "concurrently \"ng serve\" \"json-server --watch db.json\"",
9     "startng": "ng serve",
10    "build": "ng build --base-href=/controle-de-selos/",
11    "ghpages": "angular-cli-ghpages --dir-dist/controle-de-selos",
12    "deploy": "npm run build & npm run ghpages",
13    "watch": "ng build --watch --configuration development",
14    "test": "ng test",
15    "e2e": "npx e2e",
16    "json:server": "json-server --watch db.json"
17  },
18  "devDependencies": {
19    "@angular-devkit/build-angular": "^16.0.0",
20    "@angular/cli": "^16.0.0",
21    "@angular/compiler-cli": "^16.0.0",
22    "@types/jasmine": "~4.3.0",
23    "@types/node": "^16.11.35",
24    "json-server": "^0.17.3",
25    "concurrently": "^8.2.0",

```

← → ↺ ⓘ localhost:3000

## Resources

[/users](#) 4x

[/albums](#) 18x

To access and modify resources, you can use any HTTP method:

GET POST PUT PATCH DELETE OPTIONS

undefined

```
TS album.service.ts X
src > app > services > TS album.service.ts > AlbumService > saveWithValidation > switchMap() callback > catch
54 saveWithValidation(album: Album): Observable<Album> {
55   album.name = album.name?.toUpperCase();
56   return this.checkIfAlbumExists(album).pipe(
57     switchMap(albumExists => {
58       if (albumExists) {
59         return this.httpClient.post<Album>(this.URL, album, this.httpOptions).pipe(
60           catchError((error) => {
61             console.error('Erro ao salvar o álbum:', error);
62             throw error;
63           })
64         );
65       } else {
66         return throwError('Já existe um álbum com o mesmo nome');
67       }
68     })
69   );
70 }
```

- [X] Apresentar uma lista de dados com a diretiva estrutural ngFor.

```
album-list.component.html X
src > app > components > album-list > album-list.component.html > ng-container > div.card-container > mat-card.clickable-card.card-item
17 <ng-container *ngIf="albums.length > 0; else noAlbums">
18   <div class="card-container">
19     <mat-card *ngFor="let album of albums" class="clickable-card card-item" (click)="onCardClick(album)">
20       <mat-card-header>
21         <mat-card-title>{{ album?.name | uppercase }}</mat-card-title>
22         <mat-card-subtitle>{{ album?.description }}</mat-card-subtitle>
23       </mat-card-header>
24       <mat-card-content>
25         <h3>Selos: {{ album?.selos?.length }}</h3>
26       </mat-card-content>
27       <mat-icon matTooltip="Clique no card para ver o &nbsp; Album {{ album?.name }} e a lista de selos"
28         matTooltipPosition="above">&nbsp;</mat-icon>
29     </mat-card>
30   </div>
31 </ng-container>
32 <ng-template #noAlbums>
33   <p>Nenhum álbum disponível.</p>
34 </ng-template>
35 <div class="header">
36   <br><br><br><br><br><br>
37 </div>
```

- [X] Usar a diretiva ngIf

```
album-page.component.html X
src > app > components > album-page > album-page.component.html > div > ng-container
45 <ng-container *ngIf="album && album.selos && album.selos.length > 0; else noSelos">
46   <div class="card-container">
47     <mat-card *ngFor="let selo of album?.selos" class="clickable-card card-item" (click)="onCardClick(selo)">
48       <mat-card-header>
49         <br>
50         <mat-card-title>{{ selo?.name | uppercase }}</mat-card-title>
51         <mat-card-subtitle>Descrição: {{ selo?.description }}</mat-card-subtitle>
52         <mat-card-subtitle>&nbsp;</mat-card-subtitle>
53         <mat-card-subtitle>Tipo: {{ selo?.type }}</mat-card-subtitle>
54         <mat-card-subtitle>Ano de Fabricação: {{ selo?.year }}</mat-card-subtitle>
55         <mat-card-subtitle>País de Origem: {{ selo?.country }}</mat-card-subtitle>
56         <mat-card-subtitle>Carimbo: {{ selo?.stamp ? 'Sim' : 'Não' }}</mat-card-subtitle>
57         <mat-card-subtitle>Filigrama: {{ selo?.filigram }}</mat-card-subtitle>
58         <mat-card-subtitle>
59           <a [href]="selo?.link" target="_blank" (click)="$event.stopPropagation()">{{ selo?.link }}</a>
60         </mat-card-subtitle>
61       </mat-card-header>
62       <mat-card-content>
63       </mat-card-content>
64     </mat-card>
65   </div>
66 </ng-container>
67 <ng-template #noSelos>
68   <br><br>
69   <p class="text-center">Nenhum Selo disponível.</p>
70 </ng-template>
71 </div>
```

- [X] Formatar a apresentação de dados com Pipes.

```
album-list.component.html X
src > app > components > album-list > album-list.component.html > ng-container > div.card-container > mat-card.clickable-card.card-item
17 <ng-container *ngIf="albums.length > 0; else noAlbums">
18   <div class="card-container">
19     <mat-card *ngFor="let album of albums" class="clickable-card card-item" (click)="onCardClick(album)">
20       <mat-card-header>
21         <mat-card-title>{{ album?.name | uppercase }}</mat-card-title>
22         <mat-card-subtitle>{{ album?.description }}</mat-card-subtitle>
23       </mat-card-header>
24       <mat-card-content>
25         <h3>Selos: {{ album?.selos?.length }}</h3>
26       </mat-card-content>
27       <mat-icon matTooltip="Clique no card para ver o &nbsp; Album {{ album?.name }} e a lista de selos"
28         matTooltipPosition="above">&nbsp;</mat-icon>
29     </mat-card>
30   </div>
31 </ng-container>
32 <ng-template #noAlbums>
33   <p>Nenhum álbum disponível.</p>
34 </ng-template>
```

- [X] Fazer requisições a API com tratamento da resposta com Observables.

```
TS album.service.ts X
src > app > services > TS album.service.ts > AlbumService > saveWithValidation > switchMap() callback > catch

54 saveWithValidation(album: Album): Observable<Album> {
55   album.name = album.name?.toUpperCase();
56   return this.checkIfAlbumExists(album).pipe(
57     switchMap(albumExists => {
58       if (albumExists) {
59         return this.httpClient.post<Album>(this.URL, album, this.httpOptions).pipe(
60           catchError((error) => {
61             console.error('Erro ao salvar o álbum:', error);
62             throw error;
63           })
64         );
65       } else {
66         return throwError('Já existe um álbum com o mesmo nome');
67       }
68     })
69   );
70 }
```

```
TS album-add-modal.component.ts X
src > app > components > album-add-modal > TS album-add-modal.component.ts > AlbumAddModalComp

74
75 this.service.saveWithValidation(album).subscribe({
76   next: (savedAlbum) => {
77     this.nome = '';
78     this.descricao = '';
79     this.showModal = false;
80     this.closeModalEvent.emit();
81   },
82   error: (error: any) => {
83     this.openValidationModal(Constants.ALBUM_NAME_VALIDATE);
84   },
85   complete: () => {
86     this.openValidationModal(Constants.ALBUM_SAVED);
87   },
88 });
89 }
```

- [X] Build e deploy da aplicação.

<https://github.com/jvjungles/controle-de-selos/tree/gh-pages>

<https://github.com/jvjungles/controle-de-selos/deployments>

```
() packagejson •
() package.json > {} devDependencies

1 {
2   "name": "controle-de-selos",
3   "version": "1.0.0",
4   "description": "Sistema de Controle - Catálogo Filatélico - Selos.",
5   "license": "MIT",
6   "scripts": {
7     "ng": "ng",
8     "start": "concurrently \"ng serve\" \"json-server --watch db.json\"",
9     "startng": "ng serve",
10    "build": "ng build --base-href=/controle-de-selos/",
11    "ghpages": "angular-cli-ghpages --dir=dist/controle-de-selos",
12    "deploy": "npm run build & npm run ghpages",
13    "watch": "ng build --watch --configuration development",
14    "test": "ng test",
15    "e2e": "ng e2e",
16    "json:server": "json-server --watch db.json"
17  },
18  "devDependencies": {
19    "@angular-devkit/build-angular": "^16.0.0",
20    "@angular/cli": "^16.0.0",
21    "@angular/compiler-cli": "^16.0.0",
22    "@types/jasmine": "~4.3.0",
23    "@types/node": "^16.11.35",
24    "angular-cli-ghpages": "^1.0.0",
25    "concurrently": "^8.2.0",
```

```
joao.jungles@WINB029083CWB MINGW64 ~/github/controlre-de-selos (feature/build-deploy)
$ npm run deploy

> controlre-de-selos@1.0.0 deploy
> npm run build & npm run ghpages

> controlre-de-selos@1.0.0 build
> ng build --base-href=/controlre-de-selos/

✓ Browser application bundle generation complete.
✓ Copying assets complete.
✖ Generating index.html...1 rules skipped due to selector errors:
  router-outlet+* -> Cannot read properties of undefined (reading 'type')
✓ Index.html generation complete.

Initial Chunk Files | Names | Raw Size | Estimated Transfer Size
main.99b5b72da8964dc7.js | main | 419.71 kB | 98.56 kB
styles.2eab8d7b928aa3e6.css | styles | 112.97 kB | 11.10 kB
Initial Chunk Files | Names | Raw Size | Estimated Transfer Size

main.99b5b72da8964dc7.js | main | 419.71 kB | 98.56 kB

styles.2eab8d7b928aa3e6.css | styles | 112.97 kB | 11.10 kB

polyfills.902a9b684e7e3593.js | polyfills | 33.00 kB | 10.63 kB
runtime.527749ff98af9bdc.js | runtime | 912 bytes | 515 bytes

| Initial Total | 566.57 kB | 120.79 kB

> controlre-de-selos@1.0.0 ghpages
> angular-cli-ghpages --dir=dist/controlre-de-selos

⬢ Uploading via git, please wait...
⬢ Successfully published via angular-cli-ghpages! Have a nice day!
```

pages build and deployment #10

Re-run all jobs Latest #3

Summary

Jobs

- build
- report-build-status
- deploy

Run details

- Usage

Re-run triggered 3 days ago

Status

Total duration

Artifacts

jjungles

49a77f2

Success

22s

1

pages-build-deployment

on: dynamic

build25s

report-build-status2s

deploy7s

https://jjungles.github.io/controlre-de-sel...

Artifacts

Produced during runtime

Name

Size

github-pagesExpired780 KB

## 12. Aplicação rodando em Produção:

<https://jvjungles.github.io/controle-de-selos/>

