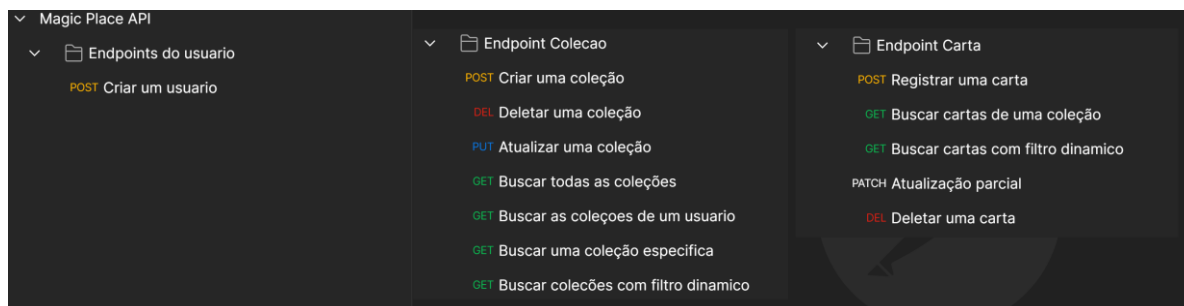


Terminando o crud

Quando terminei o CRUD básico dos recursos primordiais da API (As Coleções e as Cartas) eu já tinha Postman lotado de endpoint, mas tinha conseguido cumprir meu objetivo principal que era ter um crud que validava operações que não eram "R":



Uma coisa me deixou receoso nessa parte, tudo que é atributo pode virar critérios de queries, mas o prazo é curto pra fazer tudo isso. Então como uma cereja do bolo decidi implementar uma pesquisa dinâmica nas coleções e nas cartas, para que elas pudessem ser encontradas mais facilmente, pensando no ponto de vista de um usuário comum:

```
@GetMapping("/por-nome-edicao/{filtroDinamico}")
public ResponseEntity<Page<CartaDTO>> buscarCartasPorNomeOuEdicao(@PathVariable String filtroDinamico,
    @PageableDefault(sort = "nomeCarta", direction = Sort.Direction.ASC, page = 0, size = 5) Pageable paginacao){

    Page<Carta> paginaCartas = cartaRepository.findAll(Specification
        .where(CartaSpecification.nomeOuCartaParecidaCom(filtroDinamico))
        .or(CartaSpecification.edicaoOuCartaParecidaCom(filtroDinamico)), paginacao);

    Page<CartaDTO> paginaCartaDto = paginaCartas.map(CartaDTO::new);

    return ResponseEntity.ok(paginaCartaDto);
}
```

```
@GetMapping("/por-nome-descricao/{filtroDinamico}")
public ResponseEntity<Page<ColecaoDTO>> buscarColecoesPorNomeOuDesc(@PathVariable String filtroDinamico,
    @PageableDefault(sort = "nomeColecao", direction = Sort.Direction.ASC, page = 0, size = 5) Pageable paginacao){

    Page<Colecao> paginaColecao = colecaoRepository.findAll(Specification
        .where(ColecaoSpecification.nomeOuColecaoParecidaCom(filtroDinamico))
        .or(ColecaoSpecification.descOuColecaoParecidaCom(filtroDinamico)), paginacao);

    Page<ColecaoDTO> paginaDtoDeColecao = paginaColecao.map(ColecaoDTO::new);

    return ResponseEntity.ok(paginaDtoDeColecao);
}
```

Na parte de ordenação e paginação eu usei das duas formas possíveis, em **uma eu usei os parâmetros via URL de maneira explícita** e na outra eu deixei um **valor padrão com ajuda do Spring** (mas que podem ser substituídos pelo cliente da API a vontade, desde que ele passe as variáveis em inglês):

```
@GetMapping("/da-colecao/idColecao={id}&pagina={pg}&qtdPagina={qtd}&ordenarPor={ordenacao}")
public ResponseEntity<Page<CartaDTO>> buscarCartasDeUmaColecao(@PathVariable Long id,
    @PathVariable int pg, @PathVariable int qtd,
    @PathVariable String ordenacao){
```

```
@GetMapping("/por-nome-edicao/{filtroDinamico}")
public ResponseEntity<Page<CartaDTO>> buscarCartasPorNomeOuEdicao(@PathVariable String filtroDinamico,
    @PageableDefault(sort = "nomeCarta", direction = Sort.Direction.ASC, page = 0, size = 5) Pageable paginacao){
```