

Inatel

CENTRO DE INTEGRAÇÃO CIENTÍFICA,
CULTURAL E TECNOLÓGICA DO INATEL



Controle de Cartas

Davi Rosim

Felipe Inácio

João Vitor Scodeler

Descrição

- Usada pelos administradores do jogo e deve permitir o cadastro de cartas (definindo o valor de poder e narrativa¹), além de permitir a atualização dos valores e remoção de cartas cadastradas.
- A aplicação enviar cartas aleatórias sempre que requisitado e receber quais cartas foram escolhidas.
- É esperado que a aplicação possa gerar relatórios de cartas mais usadas.

MVC

- **O MVC é utilizado em muitos projetos devido a arquitetura que possui**, o que possibilita a divisão do projeto em camadas muito bem definidas. Cada uma delas, o Model, o Controller e a View, executa o que lhe é definido e nada mais do que isso.
- Linguagem Java
- Framework Spring

Esboço

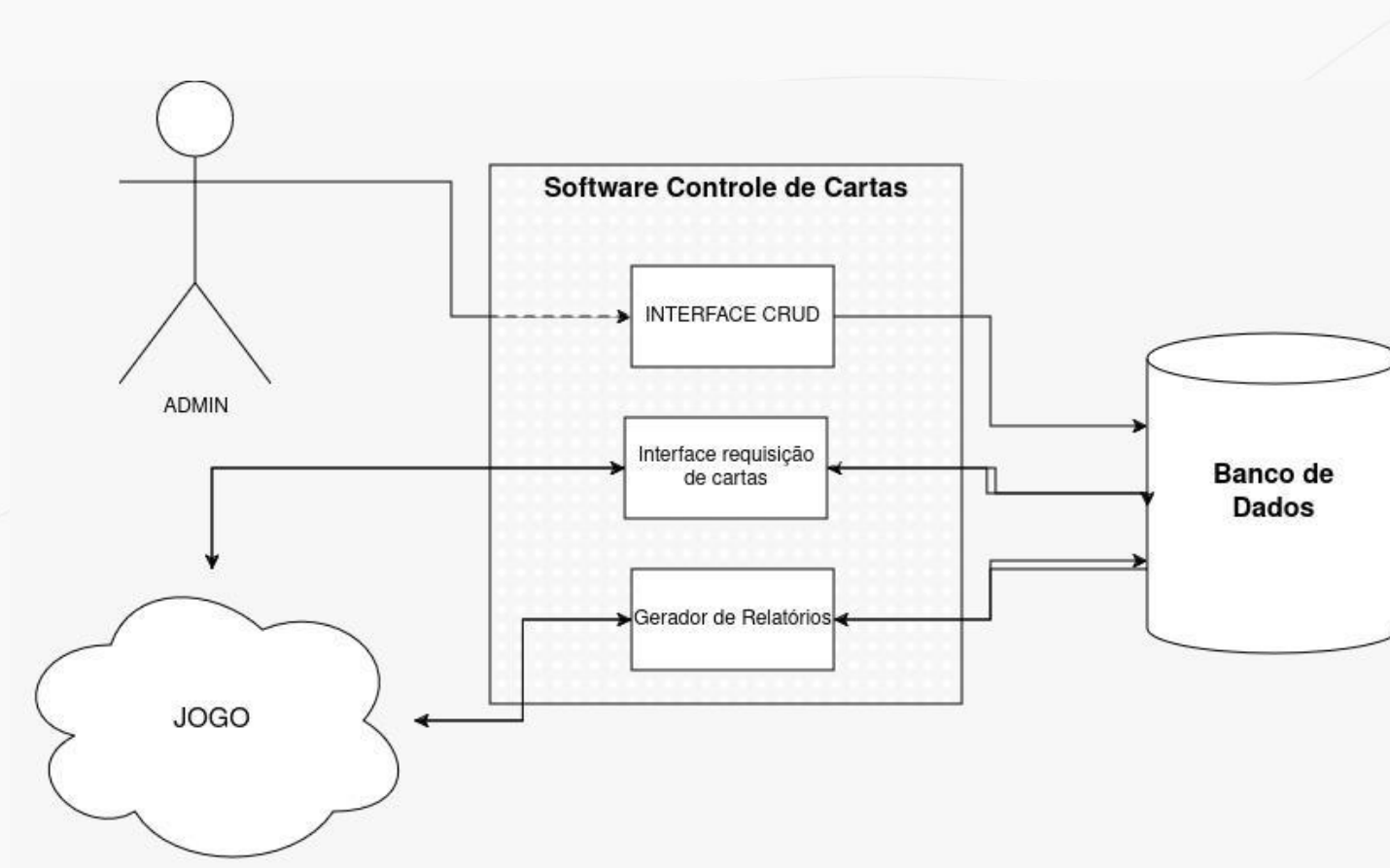


Diagrama de blocos

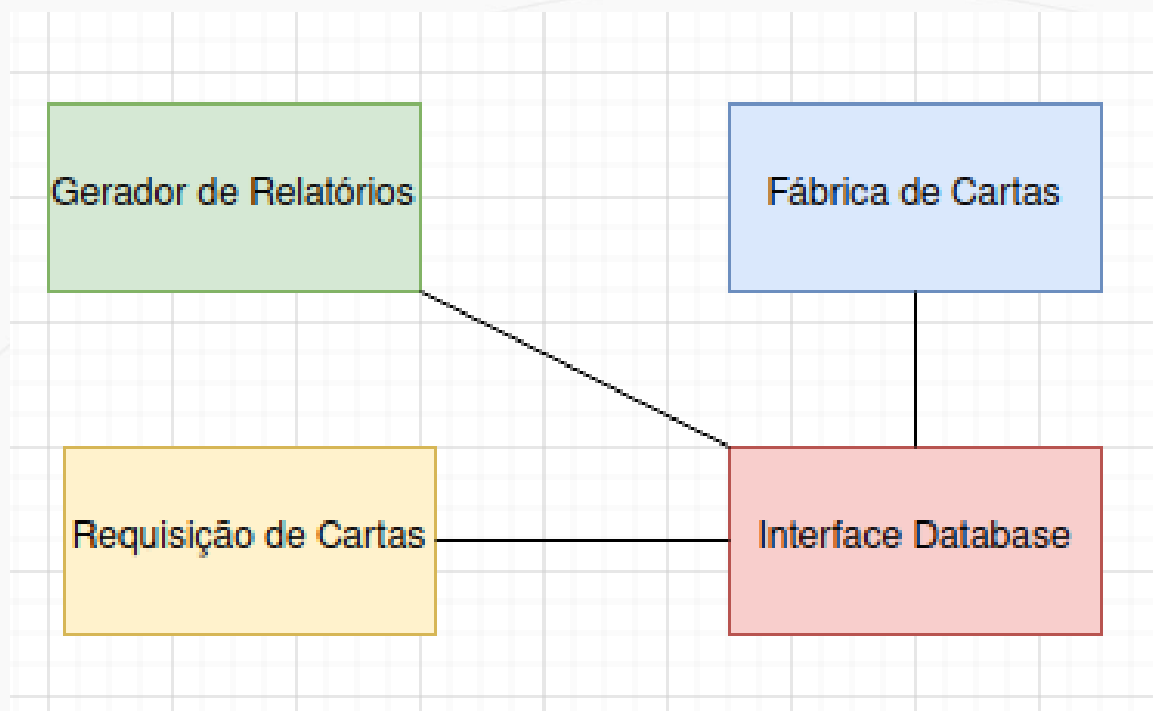


Diagrama Estrutural

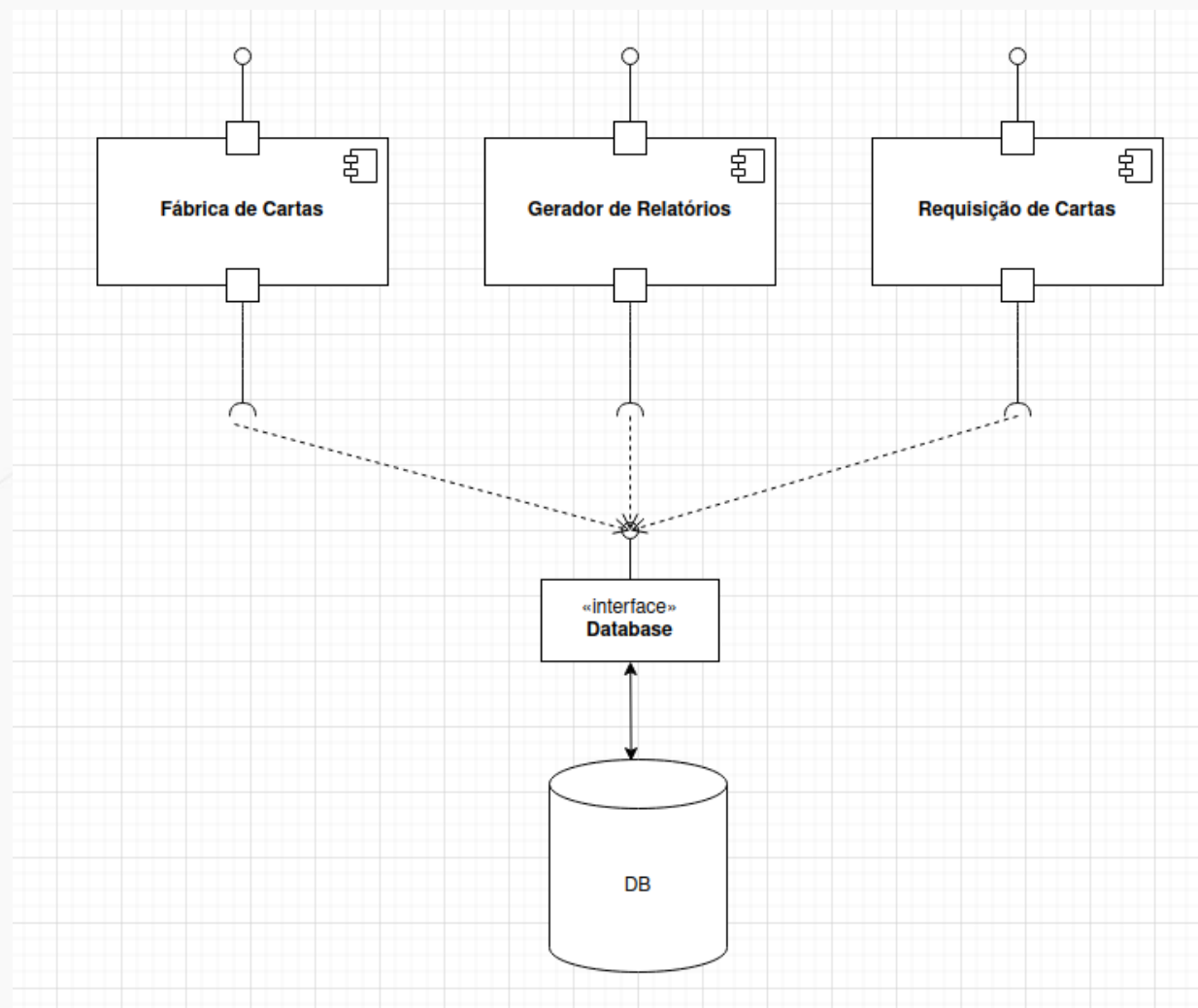
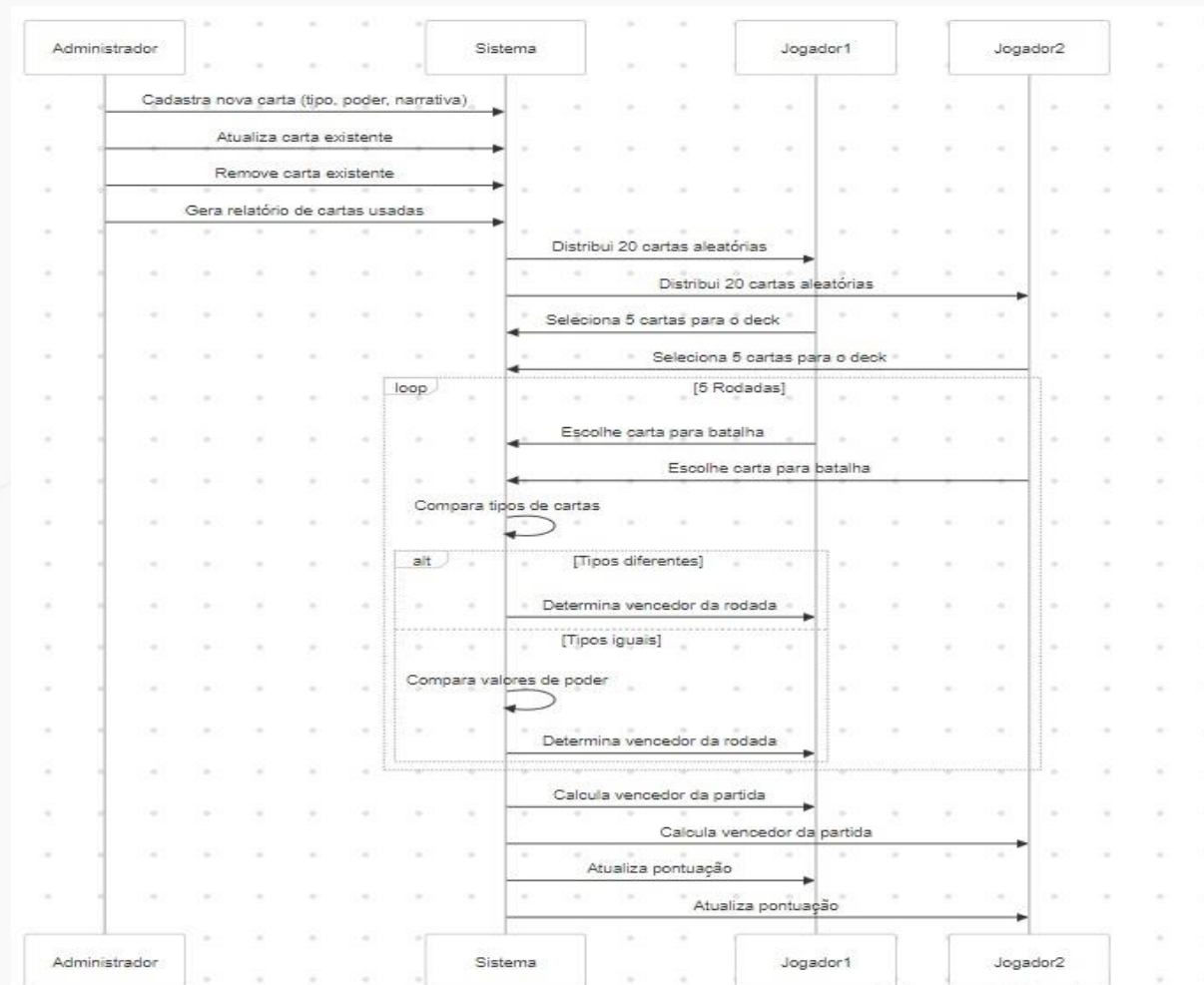


Diagrama Comportamental



Código

```
1  import org.springframework.data.annotation.Id;
2  import org.springframework.data.mongodb.core.mapping.Document;
3
4  @Document(collection = "cartas")
5  public class Carta {
6
7      @Id
8      private String id;
9      private String nome;
10     private String descricao;
11     private int poder;
12
13     // Getters e Setters
14     public String getId() {
15         return id;
16     }
17
18     public void setId(String id) {
19         this.id = id;
20     }
21
22     public String getNome() {
23         return nome;
24     }
25
26     public void setNome(String nome) {
27         this.nome = nome;
28     }
29
30     public String getDescricao() {
31         return descricao;
32     }
33
34     public void setDescricao(String descricao) {
35         this.descricao = descricao;
36     }
37
38     public int getPoder() {
39         return poder;
40     }
41
42     public void setPoder(int poder) {
43         this.poder = poder;
44     }
45 }
```

Código

```
10  public class CartaController {
11
12      @Autowired
13      private CartaService cartaService;
14
15      // Listar todas as cartas
16      @GetMapping
17      public List<Carta> listarCartas() {
18          return cartaService.listarCartas();
19      }
20
21      // Buscar carta por ID
22      @GetMapping("/{id}")
23      public ResponseEntity<Carta> buscarCartaPorId(@PathVariable String id) {
24          Optional<Carta> carta = cartaService.buscarCartaPorId(id);
25          if (carta.isPresent()) {
26              return ResponseEntity.ok(carta.get());
27          } else {
28              return ResponseEntity.notFound().build();
29          }
30      }
31
32      // Adicionar nova carta
33      @PostMapping
34      public Carta adicionarCarta(@RequestBody Carta carta) {
35          return cartaService.salvarCarta(carta);
36      }
37
38      // Atualizar carta existente
39      @PutMapping("/{id}")
40      public ResponseEntity<Carta> atualizarCarta(@PathVariable String id, @RequestBody Carta cartaAtualizada) {
41          Optional<Carta> carta = cartaService.buscarCartaPorId(id);
42          if (carta.isPresent()) {
43              cartaAtualizada.setId(id); // Mantém o ID da carta original
44              Carta cartaSalva = cartaService.salvarCarta(cartaAtualizada);
45              return ResponseEntity.ok(cartaSalva);
46          } else {
47              return ResponseEntity.notFound().build();
48          }
49      }
50
51      // Deletar carta
52      @DeleteMapping("/{id}")
53      public ResponseEntity<Void> deletarCarta(@PathVariable String id) {
54          if (cartaService.buscarCartaPorId(id).isPresent()) {
55              cartaService.deletarCarta(id);
56              return ResponseEntity.noContent().build();
57          } else {
58              return ResponseEntity.notFound().build();
59          }
60      }
61  }
```

Código

```
1 import org.springframework.data.mongodb.repository.MongoRepository;
2 import org.springframework.stereotype.Repository;
3
4 @Repository
5 public interface CartaRepository extends MongoRepository<Carta, String> {
6     // Aqui você pode adicionar consultas customizadas se necessário
7 }
```

```
1 import org.springframework.beans.factory.annotation.Autowired;
2 import org.springframework.stereotype.Service;
3 import java.util.List;
4 import java.util.Optional;
5
6 @Service
7 public class CartaService {
8
9     @Autowired
10     private CartaRepository cartaRepository;
11
12     public List<Carta> listarCartas() {
13         return cartaRepository.findAll();
14     }
15
16     public Optional<Carta> buscarCartaPorId(String id) {
17         return cartaRepository.findById(id);
18     }
19
20     public Carta salvarCarta(Carta carta) {
21         return cartaRepository.save(carta);
22     }
23
24     public void deletarCarta(String id) {
25         cartaRepository.deleteById(id);
26     }
27 }
```

Obrigado!