

Essa equipe é pequena,
tipo char

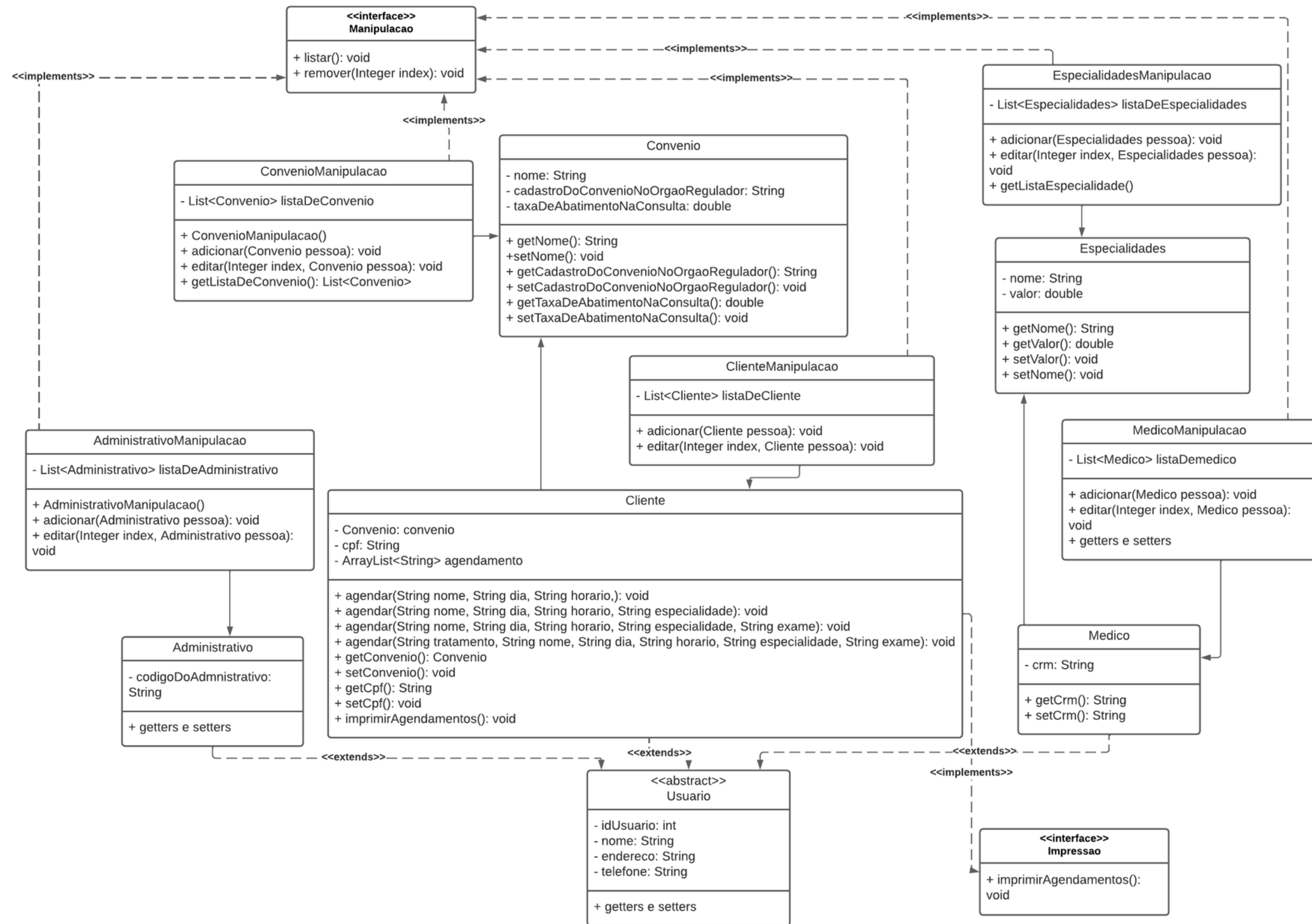
Danilo Torquato - QA
Diego Vieira - QA



Requisitos do projeto

- Cadastro de usuários (administrador, cliente, médico)
- Cadastro de convênios, especialidades médicas e tratamentos especiais
- Agendamento de consultas e outros serviços de saúde
- Consulta de valores por tipo de serviço + tipo de convênio





Encapsulamento

```
Usuario.java x
3 usages 3 inheritors DaniloTorquato +1 *
1 public abstract class Usuario {
    1 usage
2     private int idUsuario;
    2 usages
3     private String nome;
    2 usages
4     private String endereco;
    2 usages
5     private String telefone;
    no usages Diego
6     public int getIdUsuario() { return idUsuario; }
    DaniloTorquato
9     public String getNome() { return nome; }
12
    DaniloTorquato
13     public void setNome(String nome) { this.nome = nome; }
16
    6 usages DaniloTorquato
17     public String getEndereco() { return endereco; }
20
    9 usages DaniloTorquato
21     public void setEndereco(String endereco) { this.endereco = endereco; }
24
    6 usages DaniloTorquato
25     public String getTelefone() { return telefone; }
28
    9 usages DaniloTorquato
29     public void setTelefone(String telefone) { this.telefone = telefone; }
```

Encapsulamento

```
Usuario.java x
3 usages 3 inheritors DaniloTorquato +1 *
1 public abstract class Usuario {
    1 usage
2     private int idUsuario;
    2 usages
3     private String nome;
    2 usages
4     private String endereco;
    2 usages
5     private String telefone;
    no usages DaniloTorquato
6     public int getIdUsuario() { return idUsuario; }
    DaniloTorquato
9     public String getNome() { return nome; }
12
    DaniloTorquato
13     public void setNome(String nome) { this.nome = nome; }
16
    6 usages DaniloTorquato
17     public String getEndereco() { return endereco; }
20
    9 usages DaniloTorquato
21     public void setEndereco(String endereco) { this.endereco = endereco; }
24
    6 usages DaniloTorquato
25     public String getTelefone() { return telefone; }
28
    9 usages DaniloTorquato
29     public void setTelefone(String telefone) { this.telefone = telefone; }
```

Encapsulamento

```
Usuario.java x
3 usages 3 inheritors DaniloTorquato +1 *
1 public abstract class Usuario {
2     private int idUsuario;
3     private String nome;
4     private String endereco;
5     private String telefone;
6     public int getIdUsuario() { return idUsuario; }
7
8     public String getNome() { return nome; }
9
10    public void setNome(String nome) { this.nome = nome; }
11
12    public String getEndereco() { return endereco; }
13
14    public void setEndereco(String endereco) { this.endereco = endereco; }
15
16    public String getTelefone() { return telefone; }
17
18    public void setTelefone(String telefone) { this.telefone = telefone; }
```

Herança

```
Cliente.java x
1  import java.util.ArrayList;
2
3  10 usages DaniloTorquato
   public class Cliente extends Usuario implements Impressao{
4      2 usages
       private Convenio convenio= new Convenio();
5      2 usages
       private String cpf;
6      23 usages
       ArrayList<String> agendamento = new ArrayList<>();
7      3 usages DaniloTorquato
       public Cliente() {
8
9     }
10
11    no usages DaniloTorquato
       public Convenio getConvenio() {
12        return convenio;
13    }
14
15    no usages DaniloTorquato
       public void setConvenio(Convenio convenio) { this.convenio = convenio; }
16
17
18    2 usages DaniloTorquato
       public String getCpf() { return cpf; }
19
20
21    3 usages DaniloTorquato
       public void setCpf(String cpf) { this.cpf = cpf; }
22
23
```


Herança

```
Cliente.java x
1  import java.util.ArrayList;
2
3  public class Cliente extends Usuario implements Impressao{
4      private Convenio convenio= new Convenio();
5      private String cpf;
6      ArrayList<String> agendamento = new ArrayList<>();
7      public Cliente() {
8      }
9
10
11     public Convenio getConvenio() {
12         return convenio;
13     }
14
15     public void setConvenio(Convenio convenio) { this.convenio = convenio; }
16
17
18     public String getCpf() { return cpf; }
19
20
21     public void setCpf(String cpf) { this.cpf = cpf; }
```

Polimorfismo

```
Cliente.java x
32
33 // Polimorfismo
no usages DaniloTorquato
34 public void agendar(String nome, String dia, String horario) {
35     agendamento.add(nome);
36     agendamento.add(dia);
37     agendamento.add(horario);
38     agendamento.add("|| ");
39 }
40
1 usage DaniloTorquato
41 public void agendar(String nome, String dia, String horario, String especialidade) {
42     agendamento.add(nome);
43     agendamento.add(dia);
44     agendamento.add(horario);
45     agendamento.add(especialidade);
46     agendamento.add("|| ");
47 }
48
3 usages DaniloTorquato
49 public void agendar(String nome, String dia, String horario, String especialidade, String exame) {
50     agendamento.add(nome);
51     agendamento.add(dia);
52     agendamento.add(horario);
53     agendamento.add(especialidade);
54     agendamento.add(exame);
55     agendamento.add("|| ");
56 }
```

Polimorfismo

```
Cliente.java x
32
33 // Polimorfismo
no usages DaniloTorquato
34 public void agendar(String nome, String dia, String horario) {
35     agendamento.add(nome);
36     agendamento.add(dia);
37     agendamento.add(horario);
38     agendamento.add("|| ");
39 }
40
1 usage DaniloTorquato
41 public void agendar(String nome, String dia, String horario, String especialidade) {
42     agendamento.add(nome);
43     agendamento.add(dia);
44     agendamento.add(horario);
45     agendamento.add(especialidade);
46     agendamento.add("|| ");
47 }
48
3 usages DaniloTorquato
49 public void agendar(String nome, String dia, String horario, String especialidade, String exame) {
50     agendamento.add(nome);
51     agendamento.add(dia);
52     agendamento.add(horario);
53     agendamento.add(especialidade);
54     agendamento.add(exame);
55     agendamento.add("|| ");
56 }
```

Collections

```
ClienteManipulacao.java x
1  import ...
3
2 usages DaniloTorquato
4  public class ClienteManipulacao implements Manipulacao {
5      9 usages
6      private List<Cliente> listaDeCliente;
7
8      1 usage DaniloTorquato
9      public ClienteManipulacao() { this.listaDeCliente = new ArrayList<>(); }
10
11      1 usage DaniloTorquato
12      public void adicionar(Cliente pessoa) { this.listaDeCliente.add(pessoa); }
13
14      5 usages DaniloTorquato
15      public void remover(Integer index) { this.listaDeCliente.remove(index.intValue()); }
16
17      1 usage DaniloTorquato
18
19      @
20      public void editar(Integer index, Cliente pessoa) {
21          Cliente pessoaProcurada = listaDeCliente.get(index);
22          pessoaProcurada.setCpf(pessoa.getCpf());
23          pessoaProcurada.setNome(pessoa.getNome());
24          pessoaProcurada.setTelefone(pessoa.getTelefone());
25          pessoaProcurada.setEndereco(pessoa.getEndereco());
26      }
27
28      15 usages DaniloTorquato
29      public void listar() {
30          for (int i = 0; i < listaDeCliente.size(); i++) {
31              //this.imprimir();
32          }
33      }
34  }
```

Collections

```
ClienteManipulacao.java x
1  import ...
3
2 usages DaniloTorquato
4  public class ClienteManipulacao implements Manipulacao {
5      9 usages
6      = private List<Cliente> listaDeCliente;
7
1 usage DaniloTorquato
7  public ClienteManipulacao() { this.listaDeCliente = new ArrayList<>(); }
10
1 usage DaniloTorquato
11 public void adicionar(Cliente pessoa) { this.listaDeCliente.add(pessoa); }
14
5 usages DaniloTorquato
15 public void remover(Integer index) { this.listaDeCliente.remove(index.intValue()); }
18
1 usage DaniloTorquato
19 @ public void editar(Integer index, Cliente pessoa) {
20     Cliente pessoaProcurada = listaDeCliente.get(index);
21     pessoaProcurada.setCpf(pessoa.getCpf());
22     pessoaProcurada.setNome(pessoa.getNome());
23     pessoaProcurada.setTelefone(pessoa.getTelefone());
24     pessoaProcurada.setEndereco(pessoa.getEndereco());
25 }
26
15 usages DaniloTorquato
27 public void listar() {
28     for (int i = 0; i < listaDeCliente.size(); i++) {
29         //this.imprimir();

```