

# Sprint 1 - Java - Evotech

Link do repositório:

[https://github.com/jvs4nt/SPRINT1\\_JAVA\\_EVOTECH](https://github.com/jvs4nt/SPRINT1_JAVA_EVOTECH)

## Integrantes da equipe:

RM554328 - João Vitor de Santana dos Santos

- Desenvolvimento do backend
- Versionamento do projeto

RM552671 - Willian Daniel Oliveira Dantas

- Desenvolvimento do backend
- Estruturação dos endpoints

RM554021 - Sara Ingrid da Silva

- Modelagem das entidades
- Liderança da entrega
- Desenvolvimento do backend

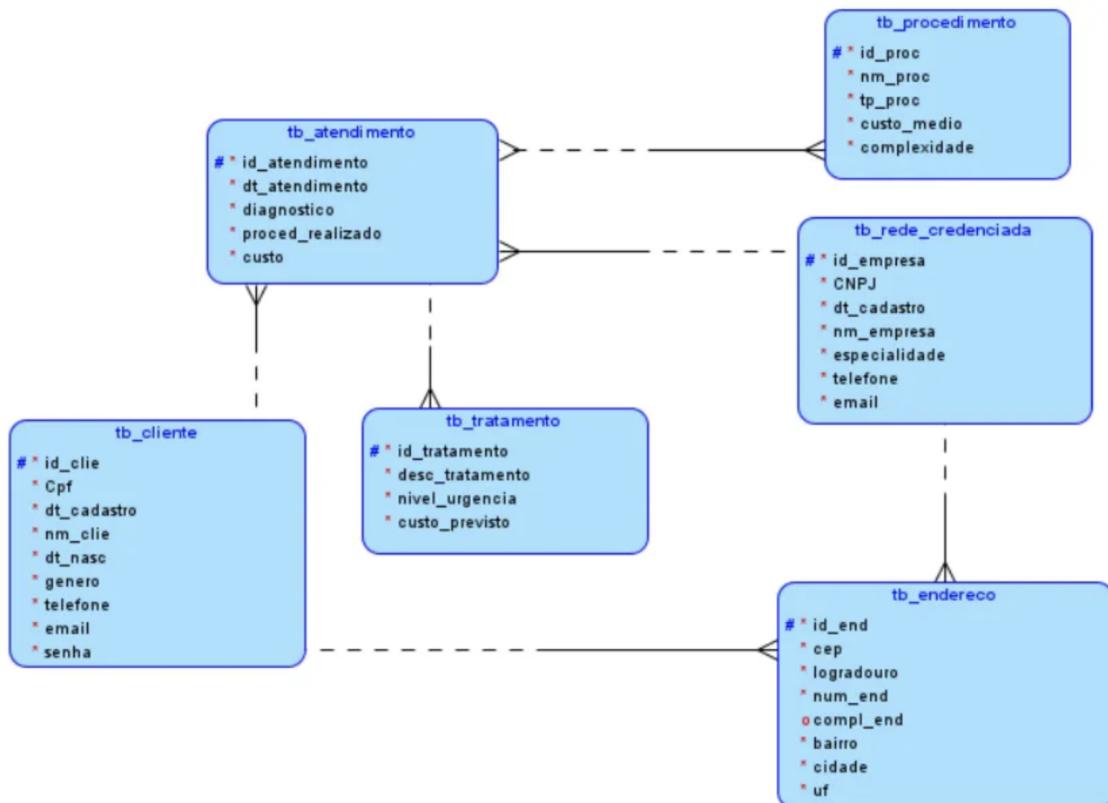
## Cronograma de desenvolvimento:

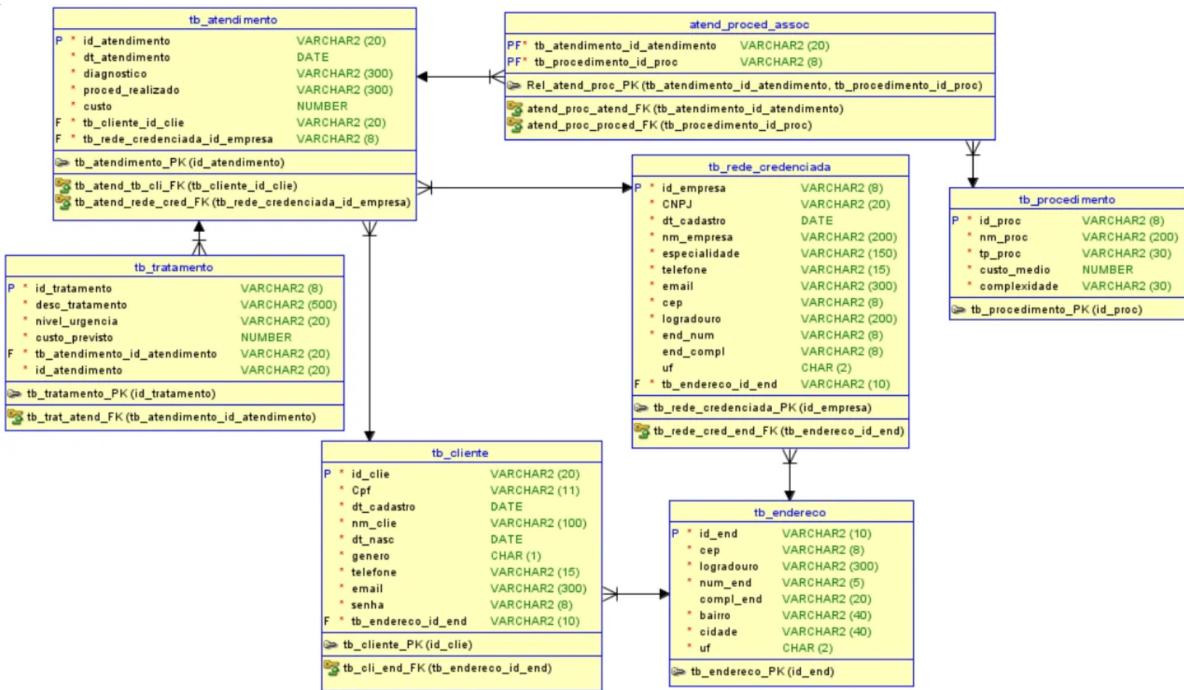
Fase	Duração	Atividades
Planejamento	2 horas	Definição de requisitos, arquitetura e design do sistema
Configuração do Ambiente	20 minutos	Instalação e configuração do Java, Spring Framework, e ferramentas de desenvolvimento

Desenvolvimento do Backend	6 horas	Implementação de modelos, repositórios, serviços e controladores REST
Testes	1 hora	Testes unitários, de integração e de sistema
Revisão e Refinamento	1 hora	Correção de bugs, otimização de desempenho e melhorias de código
Documentação	30 minutos	Elaboração da documentação técnica e do usuário

Observação: Este cronograma é uma estimativa geral e pode ser ajustado conforme as necessidades específicas durante nosso desenvolvimento.

## Diagrama de classes:





Observação: Este é o diagrama completo da nossa solução, ou seja, na primeira sprint nem todas as entidades serão implementadas, apenas o necessário para uma primeira apresentação.

O propósito dessa parte da nossa solução, é organizar o processo de atendimento dos clientes Odontoprev, armazenando o máximo de detalhes possíveis dos atendimentos, para que com análise de IA possamos melhorar cada vez mais o fluxo de clientes e acompanhar de perto desde a identificação de um sinistro até a resolução do mesmo. Em uma próxima etapa, faremos uma nova API para alimentar a gamificação do aplicativo "Odontoprev Points", esse será o nosso principal apresentável ao final da sprint 4.

## Evidências dos testes realizados:

OBS. Sinta-se livre para testar os endpoints, todos seguem o mesmo padrão então basta mudar a entidade passada na URL e se atentar aos campos necessários

## Endpoints

## Cliente

- **GET** `/clientes` : Retorna todos os clientes.
- **GET** `/clientes/{id}` : Retorna um cliente pelo ID.
- **POST** `/clientes` : Cria um novo cliente.
- **PUT** `/clientes/{id}` : Atualiza um cliente existente.
- **DELETE** `/clientes/{id}` : Deleta um cliente pelo ID.

## Atendimento

- **GET** `/atendimentos` : Retorna todos os atendimentos.
- **GET** `/atendimentos/{id}` : Retorna um atendimento pelo ID.
- **POST** `/atendimentos` : Cria um novo atendimento.
- **PUT** `/atendimentos/{id}` : Atualiza um atendimento existente.
- **DELETE** `/atendimentos/{id}` : Deleta um atendimento pelo ID.

## Tratamento

- **GET** `/tratamentos` : Retorna todos os tratamentos.
- **GET** `/tratamentos/{id}` : Retorna um tratamento pelo ID.
- **POST** `/tratamentos` : Cria um novo tratamento.
- **PUT** `/tratamentos/{id}` : Atualiza um tratamento existente.
- **DELETE** `/tratamentos/{id}` : Deleta um tratamento pelo ID.

## Clientes:

Insomnia

POST localhost:8080/clientes

201 Created | 583 ms | 1 Minute Ago

Params Body Auth Headers Scripts Docs

JSON

```

1 {
2   "endereco": {
3     "idEndereco": null,
4     "logradouro": "Rua das Flores",
5     "numero": "123",
6     "complemento": "Apto 456",
7     "bairro": "Jardim Primavera",
8     "cidade": "São Paulo",
9     "estado": "SP",
10    "cep": "01234-567"
11  },
12  "nomeCompleto": "João da Silva",
13  "email": "joao.silva@example.com",
14  "telefone": "11987654321"
15 }

```

```

1 {
2   "idCliente": "6a294849-c6a6-46b5-97b4-6cd9e0fa7b35",
3   "endereco": {
4     "idEndereco": "b1d6b5ab-d32e-4b4d-b671-3b6cc7eb27c",
5     "logradouro": "Rua das Flores",
6     "numero": "123",
7     "complemento": "Apto 456",
8     "bairro": "Jardim Primavera",
9     "cidade": "São Paulo",
10    "estado": "SP",
11    "cep": "01234-567"
12  },
13  "nomeCompleto": "João da Silva",
14  "email": "joao.silva@example.com",
15  "telefone": "11987654321",
16  "atendimentos": null
17 }

```

master | Beautify JSON | \$store.books[\*].author

Preferences | Online | Made with ❤ by Kong

Oracle Version control

Database Explorer

RM554328

Output RM554328.CLIENTE

ID_CLIENTE	EMAIL	NOME_COMPLETO	TELEFONE
6a294849-c6a6-46b5-97b4-6cd9e0fa7b35	joao.silva@example.com	João da Silva	11987654321

Services

Database Consoles > @oracle.flap.com.br > console

CHALLENGE / Sprint 1 - Java - Evotech

Compartilhar ⋮

Insomnia

Invite joaoavit342@gmail.com

Search... ⌘ P

**JAVA**

GET localhost:8080/clientes

Send ⋮ 200 OK 245 ms 379 B Just Now

Params Body Auth Headers (3) Scripts Docs

URL PREVIEW http://localhost:8080/clientes

QUERY PARAMETERS Import from URL Bulk Edit

+ Add Delete all Description

name	value

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

```

1 [
2 {
3   "idCliente": "6a294849-c6a6-
4 46b5-97b4-6cd9e0f7ab35",
5   "endereco": {
6     "idEndereco": "b1d6b5ab-d32e-
7 4b4d-b671-3b6cc7eb7e7c",
8     "logradouro": "Rua das
9     Flores",
10    "numero": "123",
11    "complemento": "Apto 456",
12    "bairro": "Jardim Primavera",
13    "cidade": "São Paulo",
14    "estado": "SP",
15    "cep": "01234-567"
16  },
17  "nomeCompleto": "João da Silva",
18  "email":
19  "jaoao.silva@example.com",
20  "telefone": "11987654321",
21  "atendimentos": []
22 }
23 ]

```

\$.store.books[\*].author

Online Made with ❤ by Kong

master Preferences

## Tratamentos

CHALLENGE / Sprint 1 - Java - Evotech

Compartilhar ⋮

Insomnia

Invite joaoavit342@gmail.com

Search... ⌘ P

**JAVA**

GET localhost:8080/tratamentos

Send ⋮ 200 OK 91 ms 194 2 Minutes Ago

Params Body Auth Headers (3) Scripts Docs

URL PREVIEW http://localhost:8080/tratamentos

QUERY PARAMETERS Import from URL Bulk Edit

+ Add Delete all Description

name	value

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. ':id'

```

1 [
2 {
3   "idTratamento": "4dce52bc-b1eb-
4 4cd4-a776-fc8e06a24e56",
5   "descricao": "Tratamento para
6   canais estágio II"
7 },
8   {
9     "idTratamento": "74ed4faf-c463-
10 46fd-9c52-9e45adc50551",
11     "descricao": "Avaliação Geral"
12   }
13 ]

```

\$.store.books[\*].author

Online Made with ❤ by Kong

master Preferences

The screenshot shows the Insomnia API client interface. A POST request is made to `localhost:8080/tratamentos`. The response status is **201 Created**, with a response time of 166 ms and a timestamp of Just Now. The JSON body sent was:

```
1= {  
2 "descricao": "Manutenção de aparelho"  
3 }
```

The response received is:

```
1= {  
2 "idTratamento": "f6bf1d61-68ec-  
454a-a8fb-1d127a621b79",  
3 "descricao": "Manutenção de  
aparelho"  
4 }
```

The screenshot shows the Insomnia API client interface. A PUT request is made to `localhost:8080/tratamentos/f6bf1d61-68ec-454a-a8fb-1d127a621b79`. The response status is **200 OK**, with a response time of 360 ms and a timestamp of Just Now. The JSON body sent was:

```
1= {  
2 "descricao": "Manutenção de aparelho atualizado"  
3 }
```

The response received is:

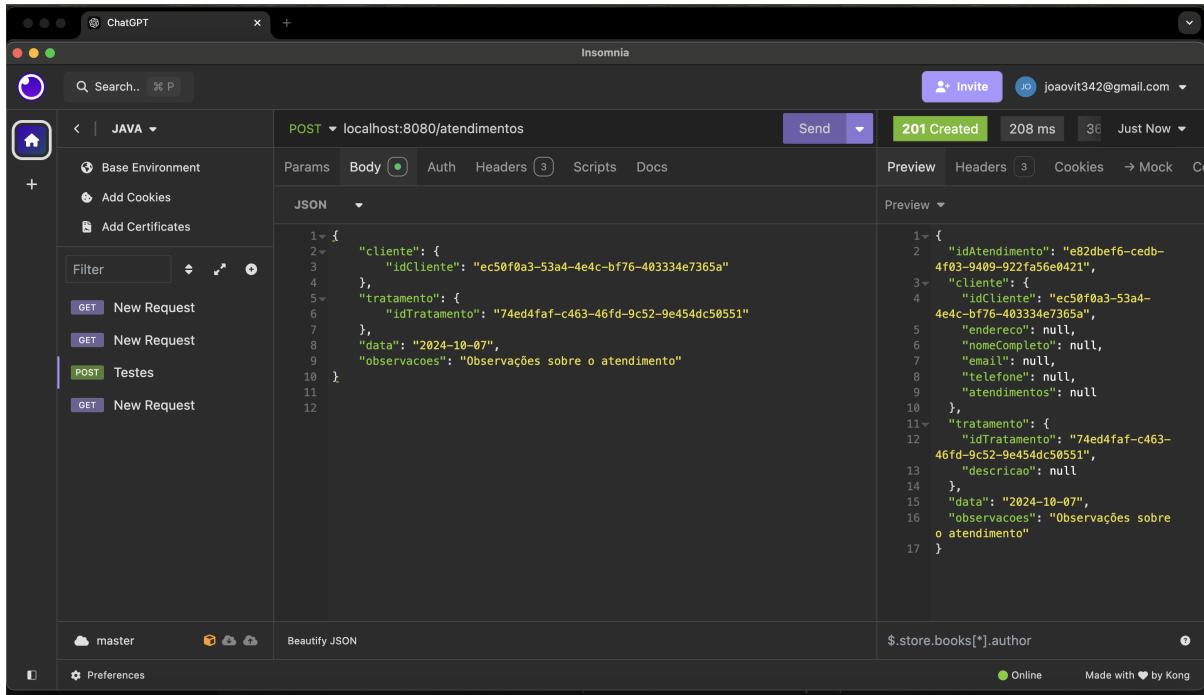
```
1= {  
2 "idTratamento": "f6bf1d61-68ec-  
454a-a8fb-1d127a621b79",  
3 "descricao": "Manutenção de  
aparelho atualizado"  
4 }
```

The screenshot shows the Insomnia API client interface. A DELETE request is made to `localhost:8080/tratamentos/4dce52bc-b1eb-4cd4-a776-fc8e06a2`. The response status is **204 No Content**, with a response time of 134 ms and a timestamp of Just Now. The JSON body sent was:

```
1= {  
2 "descricao": "Manutenção de aparelho atualizado"  
3 }
```

The response message is: **No body returned for response**.

# Atendimentos:

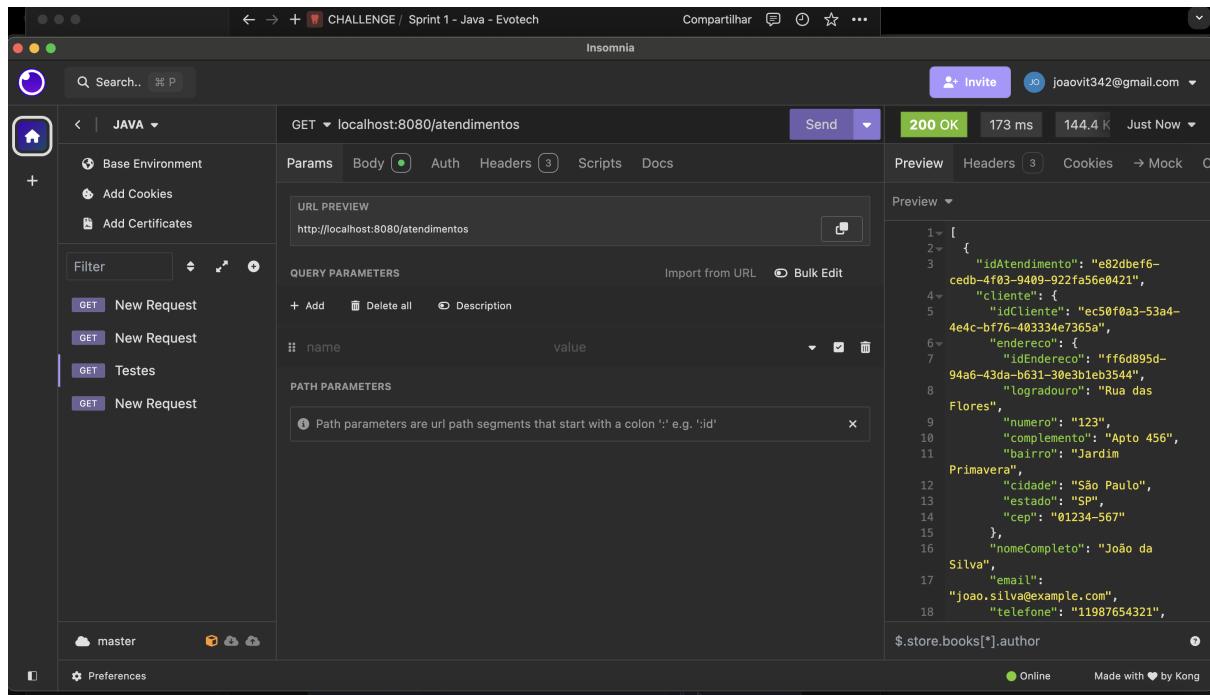


The screenshot shows the Insomnia REST client interface. A POST request is made to `localhost:8080/atendimentos`. The response status is `201 Created` with a response time of `208 ms` and timestamped as `Just Now`. The JSON response body is displayed in the preview pane:

```
1~ {  
2~   "cliente": {  
3~     "idCliente": "ec50f0a3-53a4-4e4c-bf76-403334e7365a"  
4~   },  
5~   "tratamento": {  
6~     "idTratamento": "74ed4faf-c463-46fd-9c52-9e454dc50551"  
7~   },  
8~   "data": "2024-10-07",  
9~   "observacoes": "Observações sobre o atendimento"  
10 }  
11  
12 }
```

The JSON response body is also shown in the preview pane:

```
1~ {  
2~   "idAtendimento": "e82dbef6-cedb-  
3~   4f03-9409-922fa56e0421",  
4~   "cliente": {  
5~     "idCliente": "ec50f0a3-53a4-  
6~     4e4c-bf76-403334e7365a",  
7~     "endereco": null,  
8~     "nomeCompleto": null,  
9~     "email": null,  
10~    "telefone": null,  
11~    "atendimentos": null  
12~  },  
13~  "tratamento": {  
14~    "idTratamento": "74ed4faf-c463-  
15~    46fd-9c52-9e454dc50551",  
16~    "descricao": null  
17~  },  
18~  "data": "2024-10-07",  
19~  "observacoes": "Observações sobre  
o atendimento"  
20 }
```



The screenshot shows the Insomnia REST client interface. A GET request is made to `localhost:8080/atendimentos`. The response status is `200 OK` with a response time of `173 ms` and size of `144.4 K`, timestamped as `Just Now`. The JSON response body is displayed in the preview pane:

```
1~ [  
2~   {  
3~     "idAtendimento": "e82dbef6-  
4~     cedb-4f03-9409-922fa56e0421",  
5~     "cliente": {  
6~       "idCliente": "ec50f0a3-53a4-  
7~       4e4c-bf76-403334e7365a",  
8~       "endereco": {  
9~         "idEndereco": "ff6d895d-  
10~        94a6-43da-b631-30e3b1eb3544",  
11~        "logradouro": "Rua das  
12~        Flores",  
13~        "numero": "123",  
14~        "complemento": "Apto 456",  
15~        "bairro": "Jardim  
16~        Primavera",  
17~        "cidade": "São Paulo",  
18~        "estado": "SP",  
19~        "cep": "01234-567"  
20~      },  
21~      "nomeCompleto": "João da  
22~      Silva",  
23~      "email":  
24~      "joao.silva@example.com",  
25~      "telefone": "11987654321",  
26~    }  
27~  ]
```