

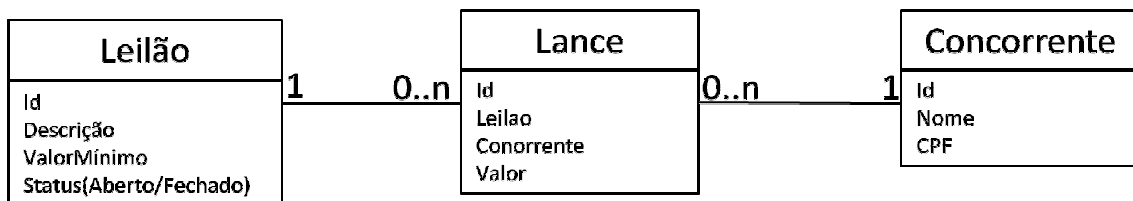
# Leilão Secreto Online

Um **Leilão** é um processo de alienação (venda) de algum bem de forma que os interessados concorrem entre si e o que oferecer a maior oferta financeira pelo referido bem ganha e paga o valor que ofereceu.

**Leilão Secreto** é uma modalidade em que cada concorrente faz seu lance sem o conhecimento do lance dos demais.

Seu trabalho é implementar o back-end para atender às requisições de um sistema de leilão online secreto.

Modelo do sistema.



Vamos implementar quatro endpoints com ações diversas em cada um.

**TODOS OS DADOS SERÃO TRANSFERIDOS EM ARQUIVOS NO FORMATO JSON!**

## Endpoint /leilao/

### Verbo GET

Com um Id (/leilao/{id}): devolve um Json com o DTO do leilão indicado pelo Id. O DTO consta todos os dados do leilão.

Caso o Id seja inválido, devolve o status 404 (Not Found)

Sem um Id (/leilao/): devolve um json com os DTOs de todos os leilões cadastrados.

Caso a operação seja bem sucedida, retorna o status 200 (Ok)

### Verbo POST

Segue um DTO (Form) com os dados de um Leilão (mas sem o ID) dentro do corpo da requisição.

O sistema deverá incluir o referido leilão e retornar o status 201 (Created)

Caso algum problema ocorra, deve retornar o status 400 (Bad Request)

Verbo **PUT**

Com um Id (/leilao/{id}): altera o dado de um leilão indicado pelo Id e devolve o DTO com o dado atualizado

Segue um DTO (Form) com os dados de um Leilão (mas sem o ID) dentro do corpo da requisição.

Caso Id seja inválido, ou não seja passado, devolve o status 404 (Not Found)

Caso a operação seja bem sucedida, retorna o status 200 (Ok) e o DTO com o dado do leilão atualizado.

#### Verbo DELETE

Com um Id (/leilao/{id}): exclui um leilão indicado pelo Id e devolve o DTO com o dado atualizado

Caso o Id seja inválido, ou não seja passado, devolve o status 404 (Not Found)

Caso a operação seja bem sucedida, retorna o status 200 (Ok)

### Endpoint /concorrente/

#### Verbo GET

Com um Id (/concorrente/{id}): devolve um Json com o DTO do concorrente indicado pelo Id. O DTO consta apenas o nome do concorrente.

Caso o Id seja inválido, devolve o status 404 (Not Found)

Sem um Id (/concorrente/): devolve um json com os DTOs de todos os concorrentes cadastrados.

Caso a operação seja bem sucedida, retorna o status 200 (Ok)

#### Verbo POST

Segue um DTO (Form) com os dados de um Concorrente (mas sem o ID) dentro do corpo da requisição.

O sistema deverá incluir o referido Concorrente e retornar o status 201 (Created)

Caso algum problema ocorra, deve retornar o status 400 (Bad Request)

#### Verbo PUT

Com um Id (/concorrente/{id}): altera o dado de um Concorrente indicado pelo Id e devolve o DTO com o dado atualizado

Segue um DTO (Form) com os dados de um Concorrente (mas sem o ID) dentro do corpo da requisição.

Caso Id seja inválido, ou não seja passado, devolve o status 404 (Not Found)

Caso a operação seja bem sucedida, retorna o status 200 (Ok) e o DTO com o dado do Concorrente atualizados.

### Verbo DELETE

Com um Id (/concorrente/{id}): exclui um Concorrente indicado pelo Id e devolve o DTO com o dado atualizado.

Caso o Id seja inválido, ou não seja passado, devolve o status 404 (Not Found)

Caso a operação seja bem sucedida, retorna o status 200 (Ok)

## Endpoint /lance/

### Verbo GET

Com um Id (/lance/{id}): devolve um Json com o DTO do lance indicado pelo Id.

O DTO consta o Id do Leilao, o Id do Concorrente e o valor do lance.

Caso o Id (parâmetro) seja inválido, devolve o status 404 (Not Found)

Sem um Id (/lance/): devolve um json com os DTOs de todos os lances cadastrados.

Com parâmetros

/lance/leilao={Id} - devolve DTOs com todos os lances do leilão indicado por ID (Status 200: OK) ou status 404 (Not Found)

/lance/concorrente={Id} - devolve DTOs com todos os lances do concorrente indicado por ID (Status 200: OK) ou status 404 (Not Found)

Caso a operação seja bem sucedida, retorna o status 200 (Ok)

### Verbo POST

Segue um DTO (Form) com os dados de um Lance (Id do Leilão, Id do Concorrente e valor) (mas sem o ID) dentro do corpo da requisição.

O sistema deverá incluir o referido Lance e retornar o status 201 (Created)

Caso o Id do Concorrente não exista, retorna status 403 (Forbidden)

Caso o Id do Leilão não exista, retorna o status 400 (Bad Request)

Caso o Id do Leilão refira a um leilão que está com status fechado, retorna o status 404 (Forbidden)

### Verbo PUT

Com um Id (/lance/{id}): altera o dado de um Lance indicado pelo Id e devolve o DTO com o dado atualizado

Segue um DTO (Form) com os dados de um Lance (mas sem o ID) dentro do corpo da requisição.

Caso Id seja inválido, ou não seja passado, devolve o status 404 (Not Found)

Caso o Id do Concorrente não exista, retorna status 403 (Forbidden)

Caso o Id do Leilão não exista, retorna o status 400 (Bad Request)

Caso o Id do Leilão refira a um leilão que está com status fechado, retorna o status 404 (Forbidden)

Caso a operação seja bem sucedida, retorna o status 200 (Ok) e o DTO com o dado do lance atualizados.

### **Verbo DELETE**

Com um Id (/lance/{id}): exclui um Concorrente indicado pelo Id e devolve o DTO com o dado atualizado.

Caso o Id seja inválido, ou não seja passado, devolve o status 404 (Not Found)

Caso o id refira a um lance cujo leilão está com status "Fechado", reorna o status 403 (Forbidden)

Caso a operação seja bem sucedida, retorna o status 200 (Ok)

### **Endpoint /vencedor\_leilao/**

Apenas o verbo GET deve ser implementado

Com um Id (/leilao/{id}): devolve um Json que inclui os dados do do leilão indicado pelo Id, mais o valor do maior lance que aquele leilão recebeu, mais o DTO do Concorrente que fez o referido lance.

Caso o Id seja inválido, devolve o status 404 (Not Found)

Caso o Id não seja passado, devolve o status 400 (Bad Request)

Caso o id refira a um lance cujo leilão está com status "Fechado", reorna o status 403 (Forbidden)