DOCUMENTO DE TESTES EXERCICIO 2

Autor: Carlos Herráez

Data: 09/11/2015

**Funcionalidade:** Salvar o endereço de um usuário, seguindo as regras CRUD (Create, Read, Update e Delete. Criando um serviço que responda e receba informação em formato JSON. Se deve verificar se o CEP passado pelo usuário é válido usando o serviço do exercício 1. Em caso de algum erro deve retornar a mensagem de erro e não realizar inclusão nem alteração. Os campos obrigatórios são: Rua, Número, Cep, Cidad e Estado.

**Procedimento de testes:** Iremos testar o serviço criado a través da ferramenta SOAPUI 5.2.1

**Serviço a ser testado:**

**Serviço 1: Inclusão:**

[**http://localhost:8080/Exercicios1e2/rest/crudServices/incluirEndereco**](http://localhost:8080/Exercicios1e2/rest/crudServices/incluirEndereco)

**Caso 1:** CEP Válidoe todos os campos obrigatórios preechidos**.**

**Resposta esperada:** Retorno de mensagem incluído com êxito.

**Resposta obtida:** Retorno de mensagem incluído com êxito.

**Entrada:** {

"cepNumber": "26176080",

"rua": "Rua Marabu",

"bairro": "Parque Real",

"cidade": "Belford Roxo",

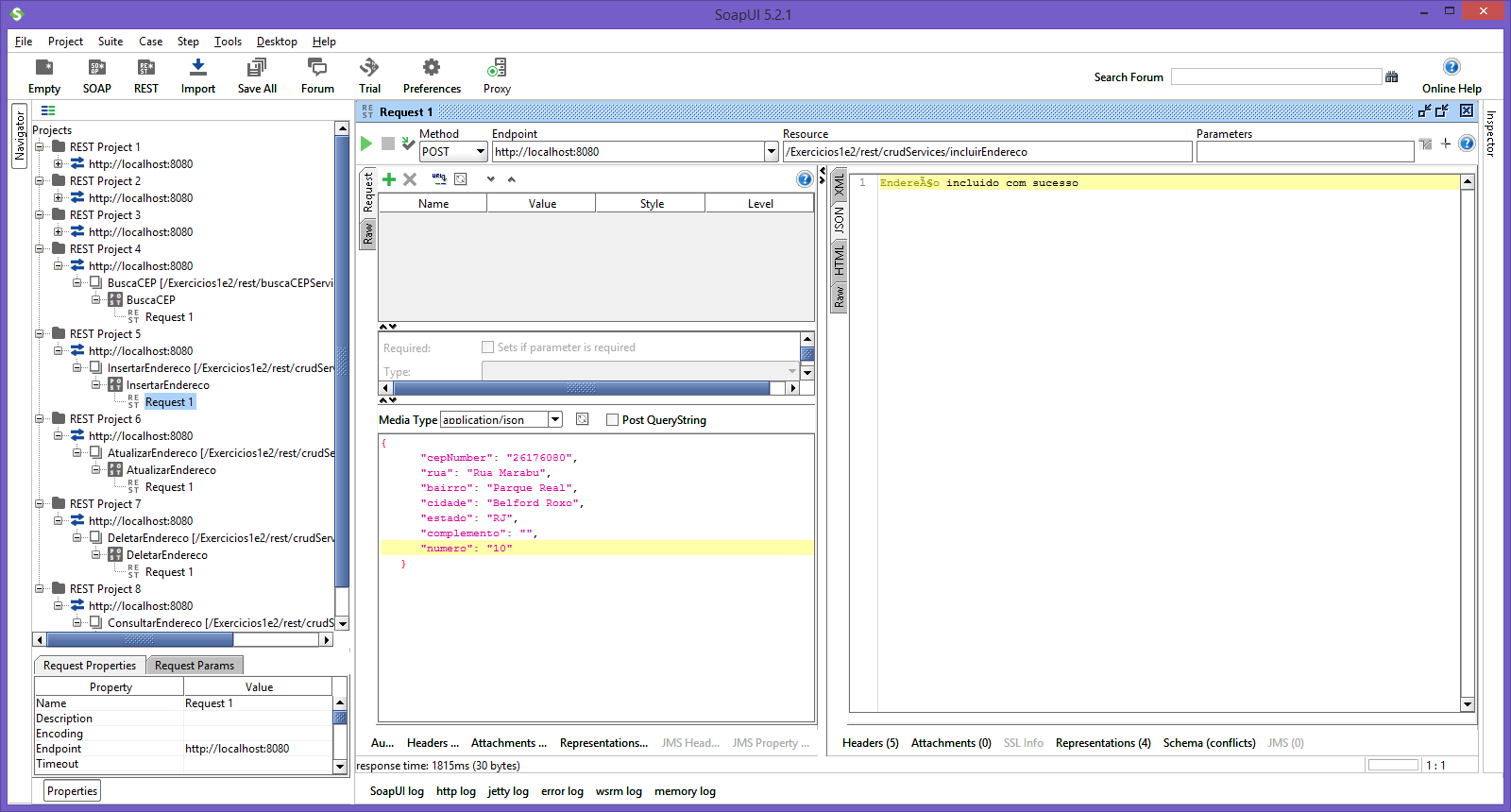
"estado": "RJ",

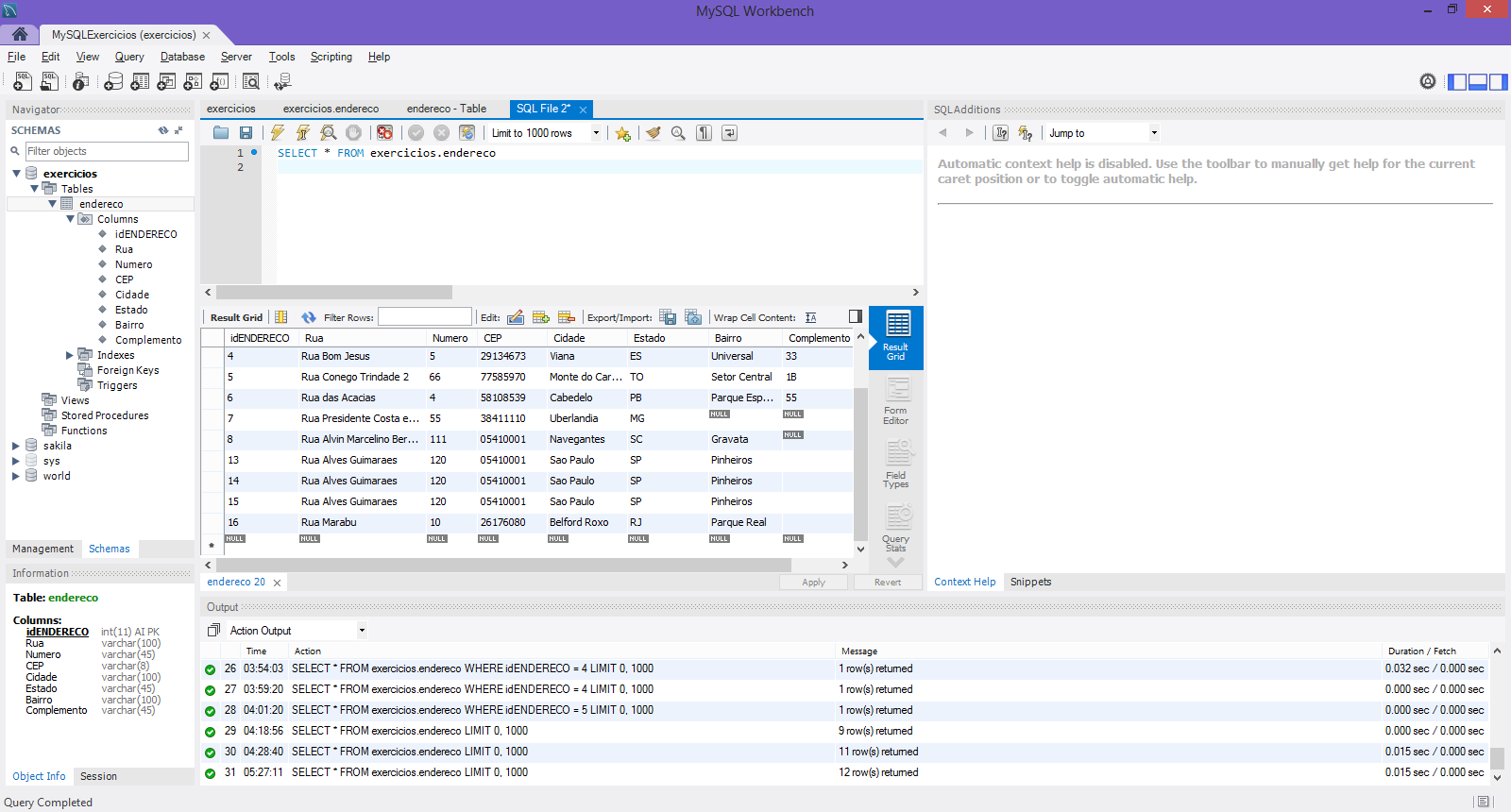
"complemento": "",

"numero": "10"

}

**Saida:** Endereço incluído com sucesso



****

Registro 16 da tabela. Inserido corretamente. CASO OK

**Caso 2:** CEP inválido**.**

**Resposta esperada:** Retorno de mensagem de CEP inválido.

**Resposta obtida:** Retorno de mensagem de CEP inválido.

**Entrada:** {

"cepNumber": "261760",

"rua": "Rua Marabu",

"bairro": "Parque Real",

"cidade": "Belford Roxo",

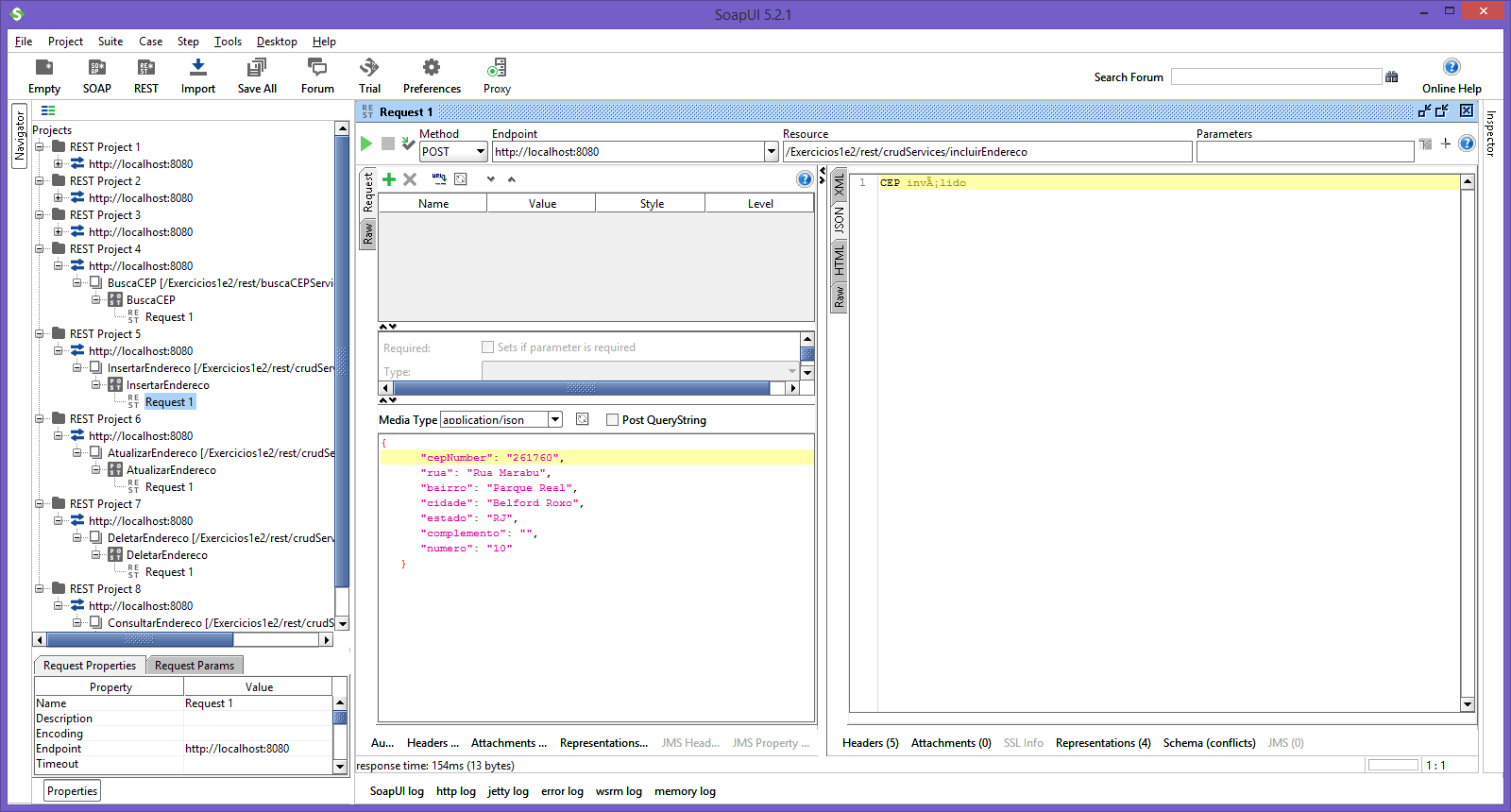
"estado": "RJ",

"complemento": "",

"numero": "10"

}

**Saída**: CEP inválido



CASO OK

**Caso 3:** Sem algum dos campos obrigatórios preenchidos**.**

**Resposta esperada:** Retorno de mensagem informando os campos obrigatórios.

**Resposta obtida:** Retorno de mensagem informando os campos obrigatórios.

**Entrada:** {

"cepNumber": "26176080",

"rua": "",

"bairro": "Parque Real",

"cidade": "Belford Roxo",

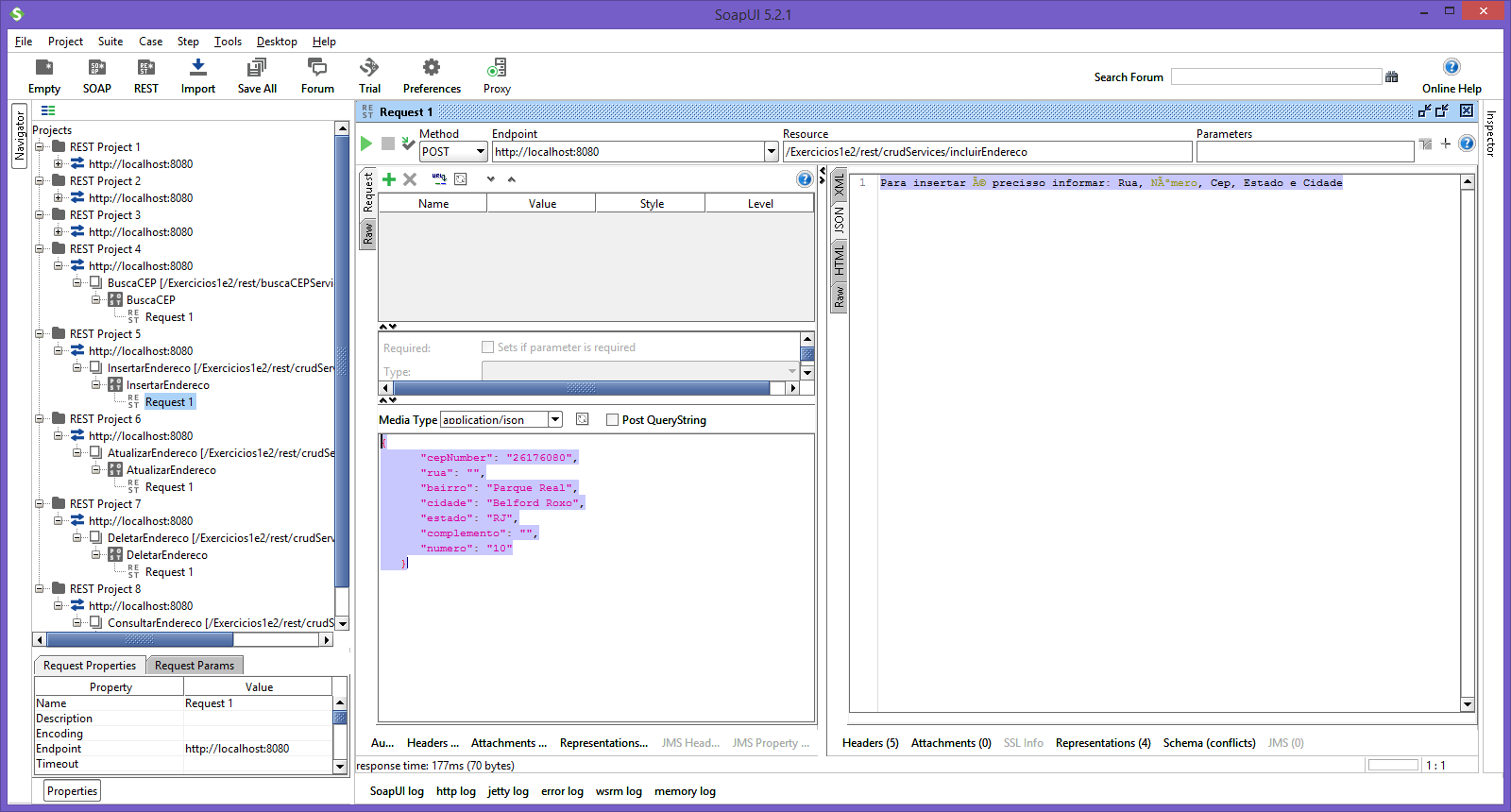
"estado": "RJ",

"complemento": "",

"numero": "10"

}

**Saída:** Para insertar é precisso informar: Rua, Número, Cep, Estado e Cidade



CASO OK

**Serviço 2: Atualização:**

[**http://localhost:8080/Exercicios1e2/rest/crudServices/atualizarEndereco**](http://localhost:8080/Exercicios1e2/rest/crudServices/atualizarEndereco)

**Caso 4:** CEP Válido**.** Id existente**.**

**Resposta esperada:** Retorno de mensagem atualizado com êxito.

**Resposta obtida:** Retorno de mensagem atualizado com êxito.

**Entrada**: {

"id": 14,

"cepNumber": "05410001",

"rua": "Rua teste",

"bairro": "",

"cidade": "",

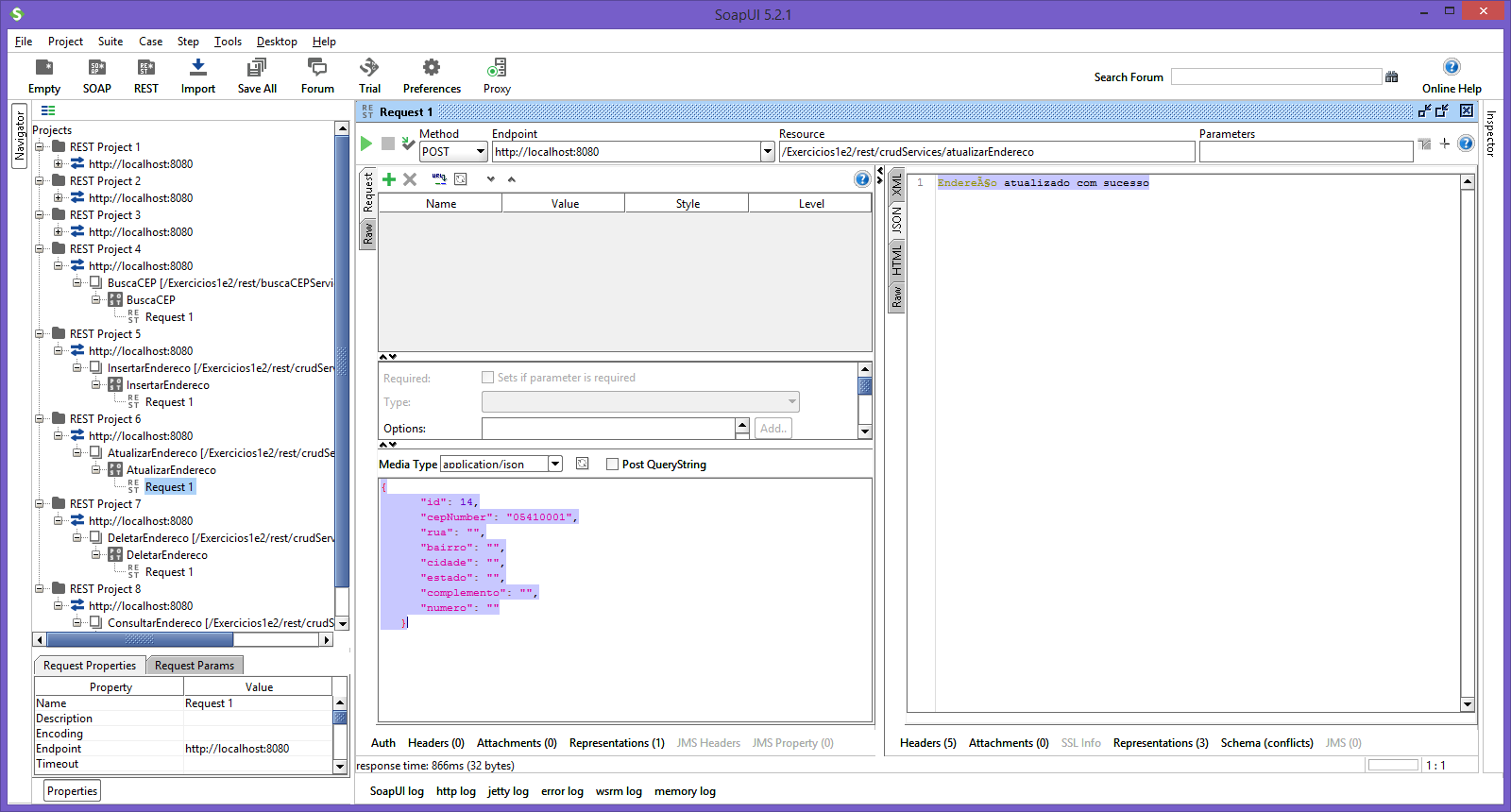
"estado": "",

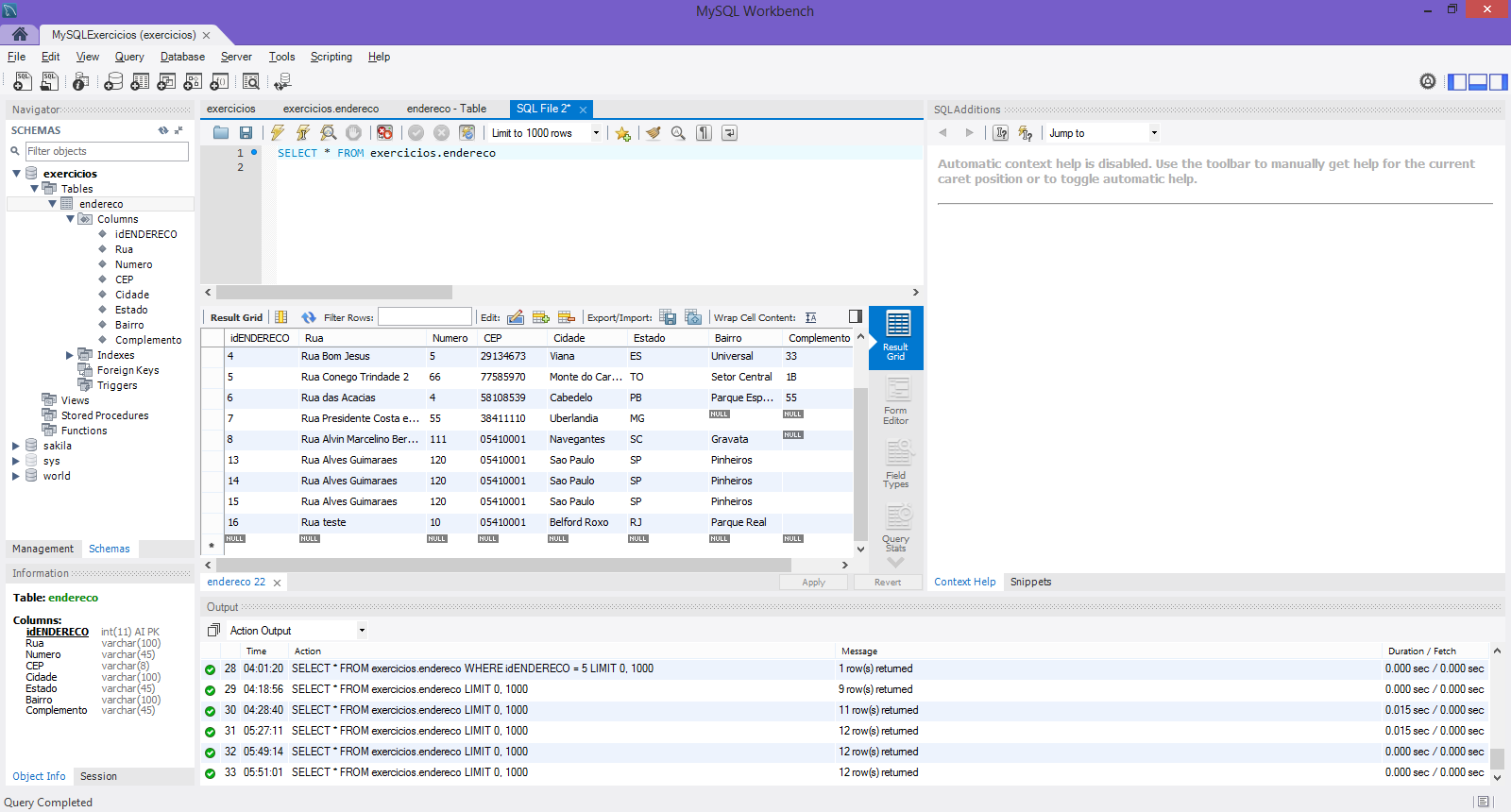
"complemento": "",

"numero": ""

}

**Saída:** Endereço atualizado com sucesso

****

****

Registro 16 da tabela. Atualizado corretamente. CASO OK

**Caso 5:** CEP Inválido**.**

**Resposta esperada:** Retorno de mensagem CEP inválido.

**Resposta obtida:** Retorno de mensagem CEP inválido.

**Entrada:** {

"id": 14,

"cepNumber": "0541000",

"rua": "",

"bairro": "",

"cidade": "",

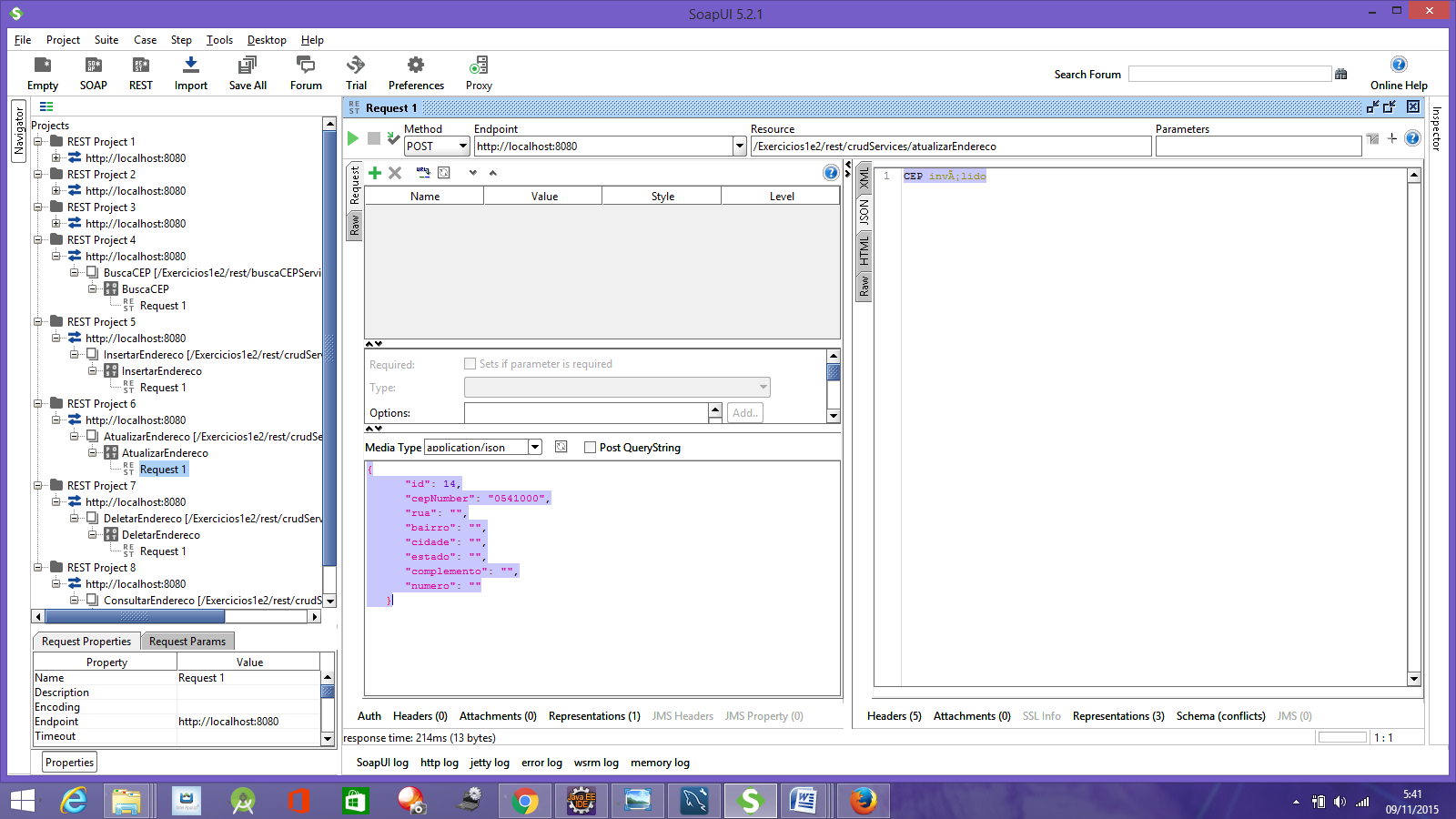
"estado": "",

"complemento": "",

"numero": ""

}

**Saída: CEP inválido**

****

CASO OK

**Caso 6:** Nenhum campo preenchido (Somente o id)**.**

**Resposta esperada:** Retorno de mensagem informando que se deve preencher pelo menos um.

**Resposta obtida:** Retorno de mensagem informando que se deve preencher pelo menos um.

**Entrada:** {

"id": 14,

"cepNumber": "",

"rua": "",

"bairro": "",

"cidade": "",

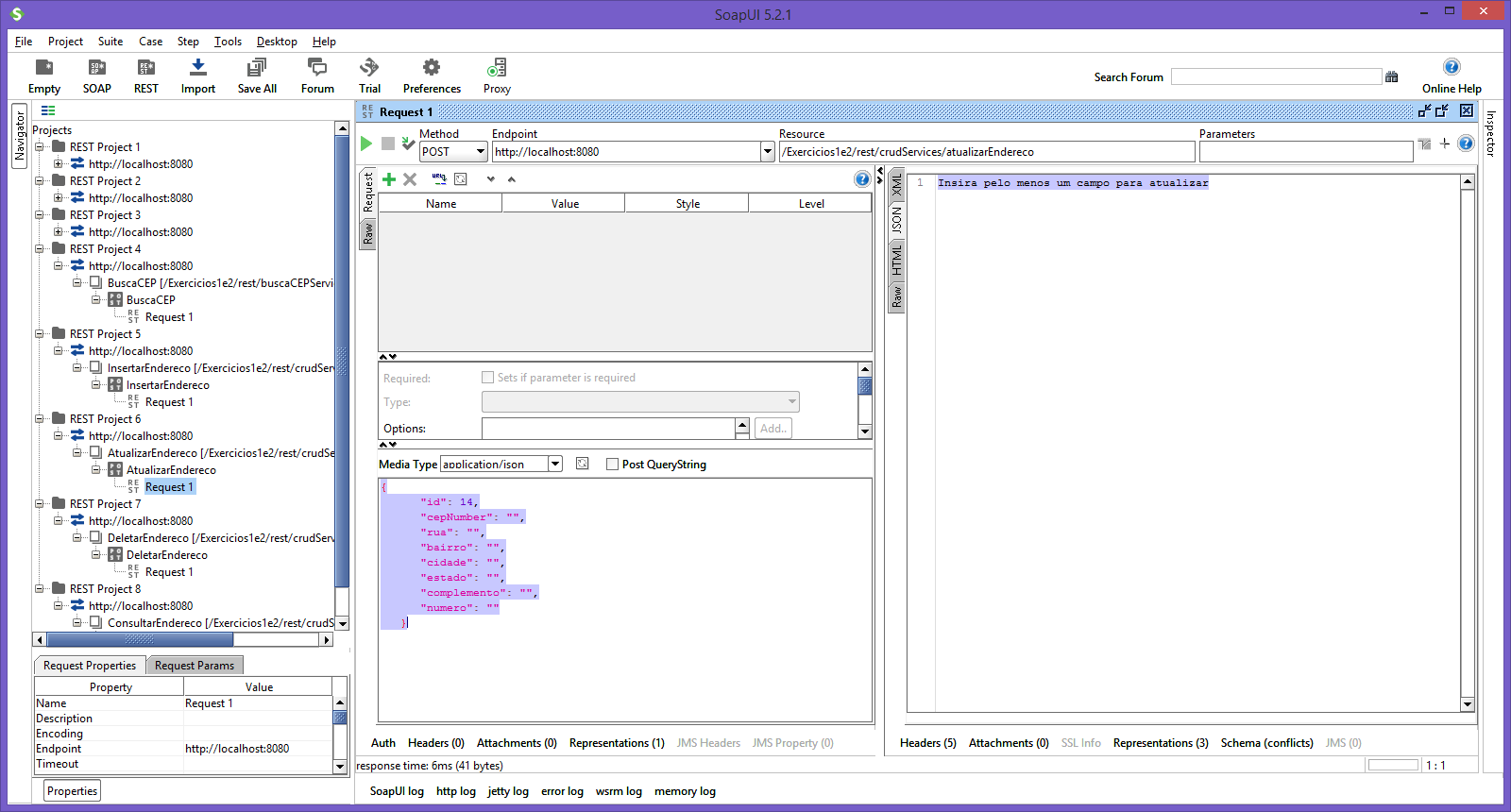
"estado": "",

"complemento": "",

"numero": ""

}

**Saída:** Insira pelo menos um campo para atualizar.



CASO OK.

**Caso 7:** Id não existe**.**

**Resposta esperada:** Retorno de mensagem informando que o endereço não foi atualizado ou não existe.

**Resposta obtida:** Retorno de mensagem informando que o endereço não foi atualizado ou não existe.

**Entrada:** {

"id": 20,

"cepNumber": "",

"rua": "Rua teste",

"bairro": "",

"cidade": "",

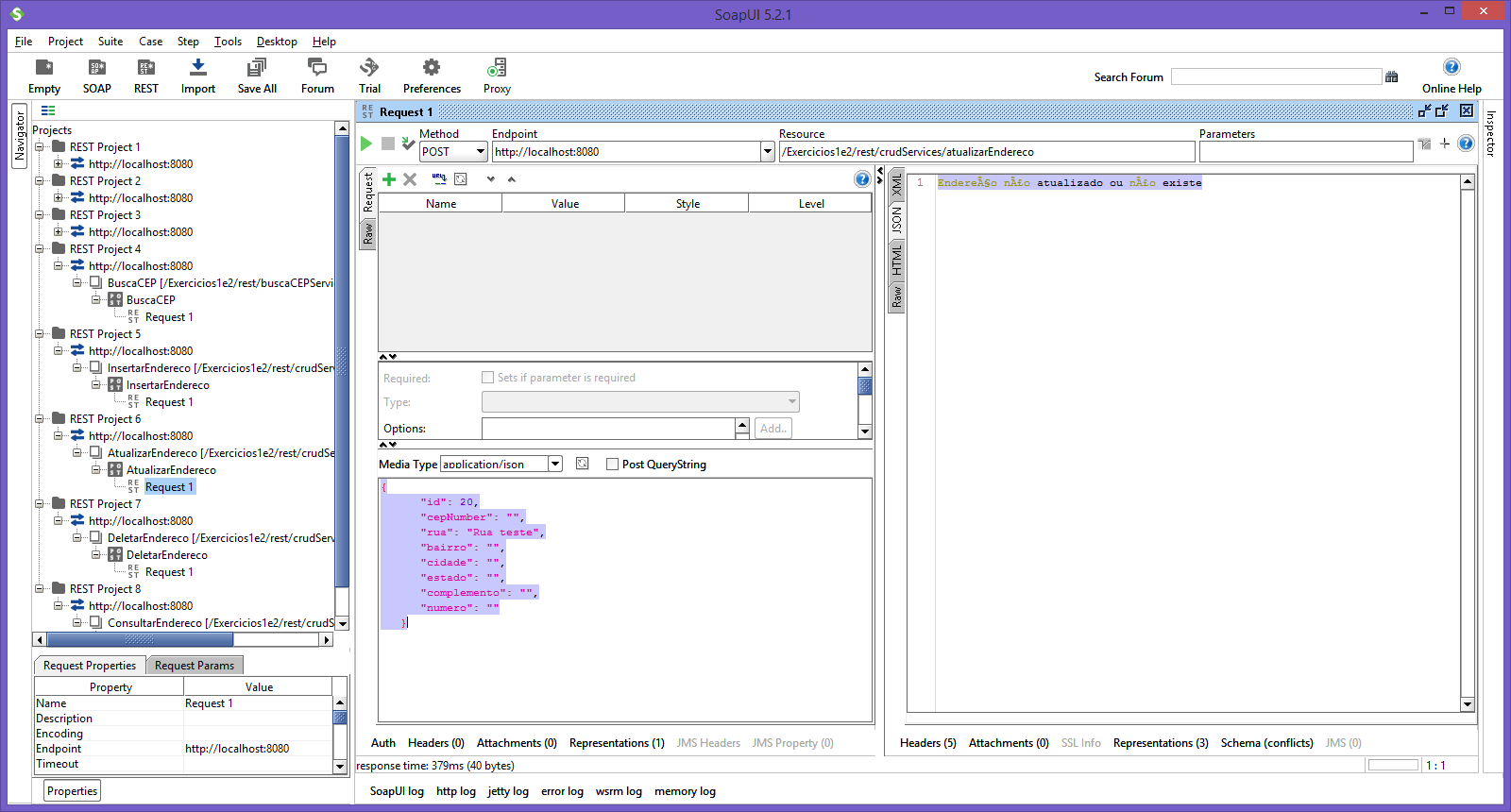
"estado": "",

"complemento": "",

"numero": ""

}

**Saída: Endereço não atualizado ou não existe**

****

CASO OK

**Serviço 3: Consultar:**

[**http://localhost:8080/Exercicios1e2/rest/crudServices/consultarEndereco**](http://localhost:8080/Exercicios1e2/rest/crudServices/consultarEndereco)

**Caso 8:** Id existente**.**

**Resposta esperada:** Retorno de mensagem informando o endereço solicitado.

**Resposta obtida:** Retorno de mensagem informando o endereço solicitado.

**Entrada**: {

"id": 16,

"cepNumber": "",

"rua": "",

"bairro": "",

"cidade": "",

"estado": "",

"complemento": "",

"numero": ""

}

**Saída:** {

"id": 16,

"endereco": {

"id": 16,

"cepNumber": "05410001",

"rua": "Rua teste",

"bairro": "Parque Real",

"cidade": "Belford Roxo",

"estado": "RJ",

"complemento": "",

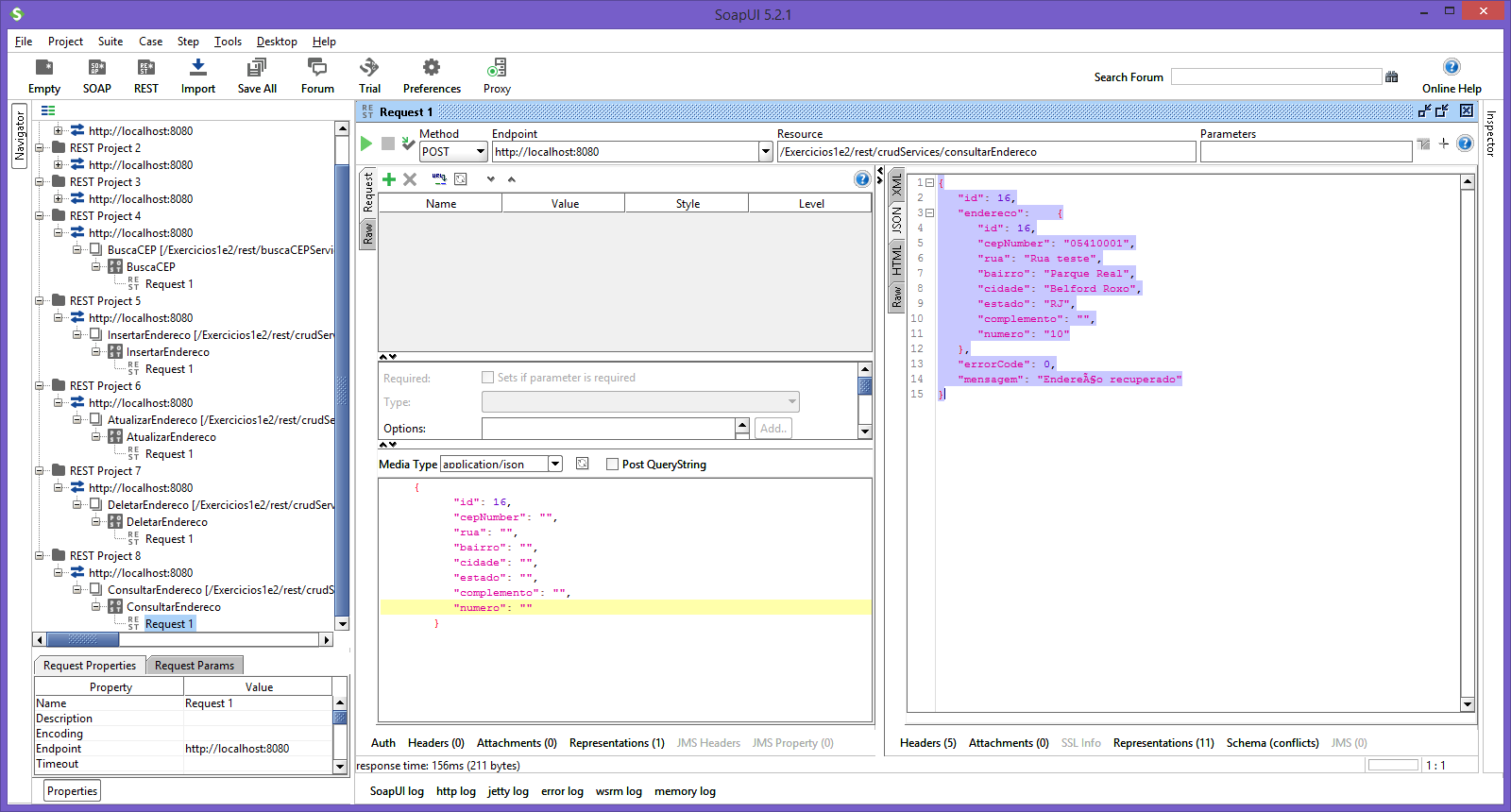
"numero": "10"

},

"errorCode": 0,

"mensagem": "Endereço recuperado"

}



CASO OK

**Caso 9:** Id não existente**.**

**Resposta esperada:** Retorno de mensagem informando que o endereço não existe.

**Resposta obtida:** Retorno de mensagem informando que o endereço não existe.

**Entrada**: {

"id": 20,

"cepNumber": "",

"rua": "",

"bairro": "",

"cidade": "",

"estado": "",

"complemento": "",

"numero": ""

}

**Saída:** {

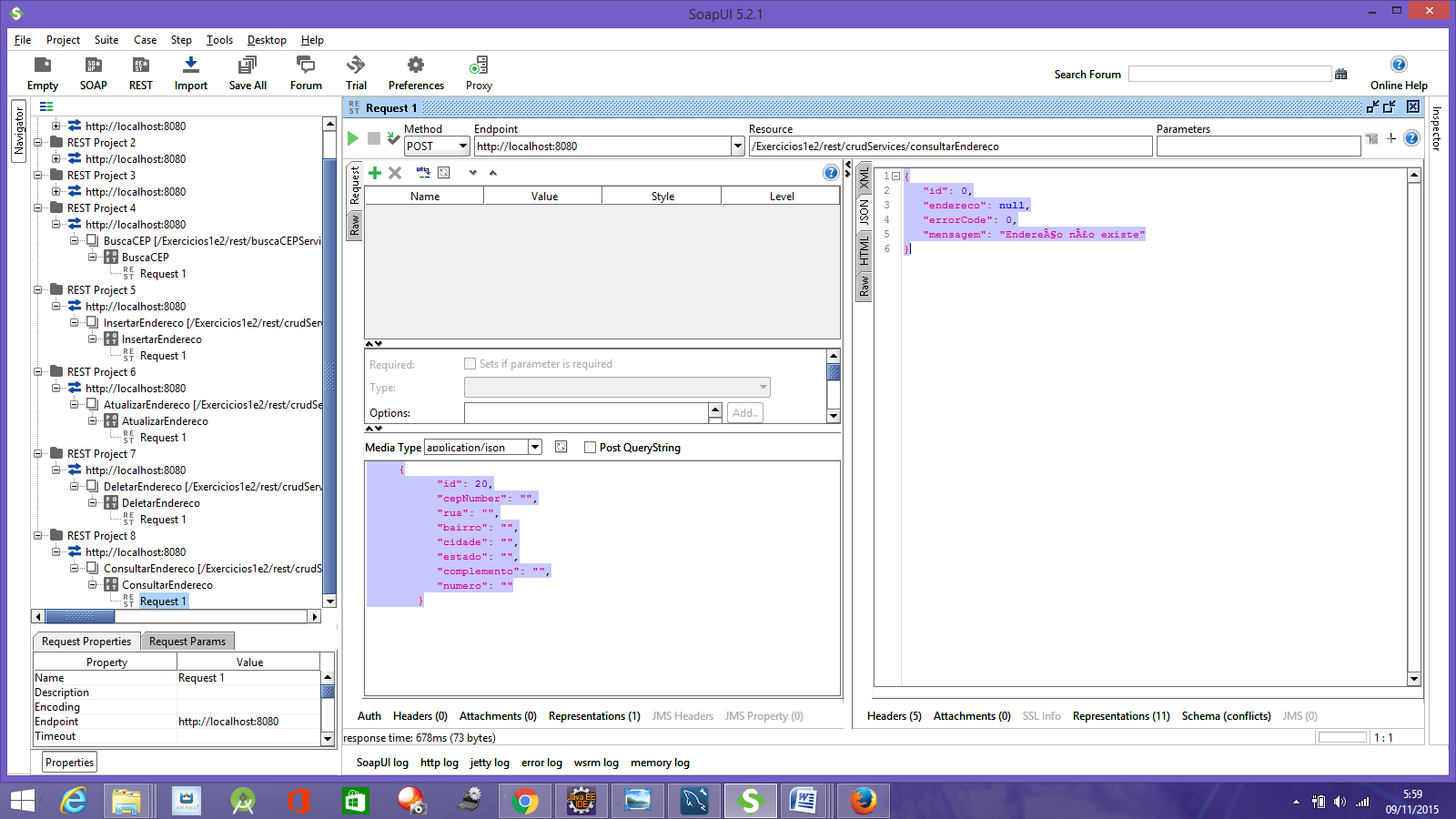
"id": 0,

"endereco": null,

"errorCode": 0,

"mensagem": "Endereço não existe"

}



CASO OK

**Serviço 4: Deletar:**

[**http://localhost:8080/Exercicios1e2/rest/crudServices/deletarEndereco**](http://localhost:8080/Exercicios1e2/rest/crudServices/deletarEndereco)

**Caso 10:** Id existente**.**

**Resposta esperada:** Retorno de mensagem informando que o endereço solicitado foi deletado.

**Resposta obtida:** Retorno de mensagem informando que o endereço solicitado foi deletado.

**Entrada**:

{

"id": 16,

"cepNumber": "",

"rua": "",

"bairro": "",

"cidade": "",

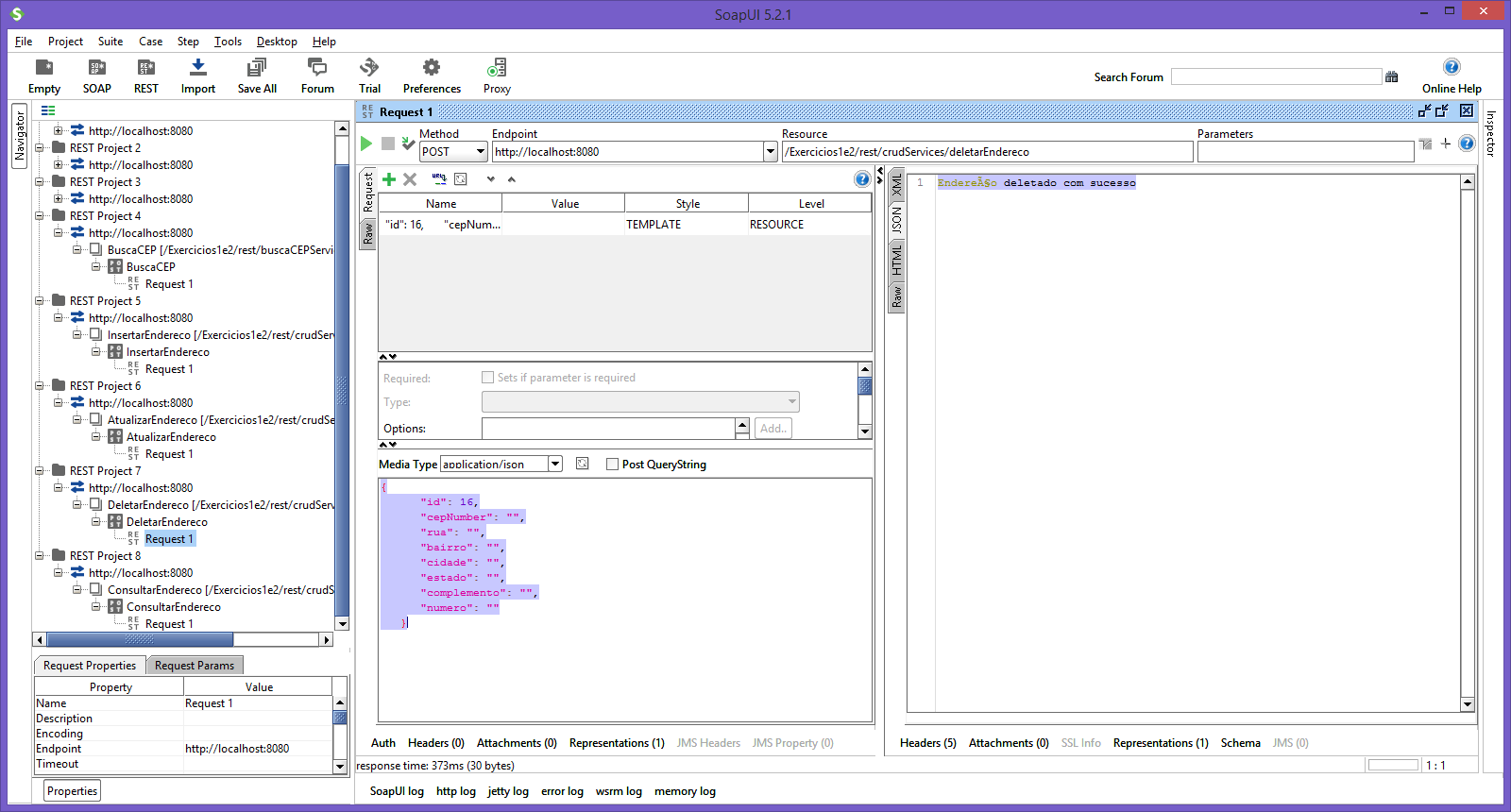
"estado": "",

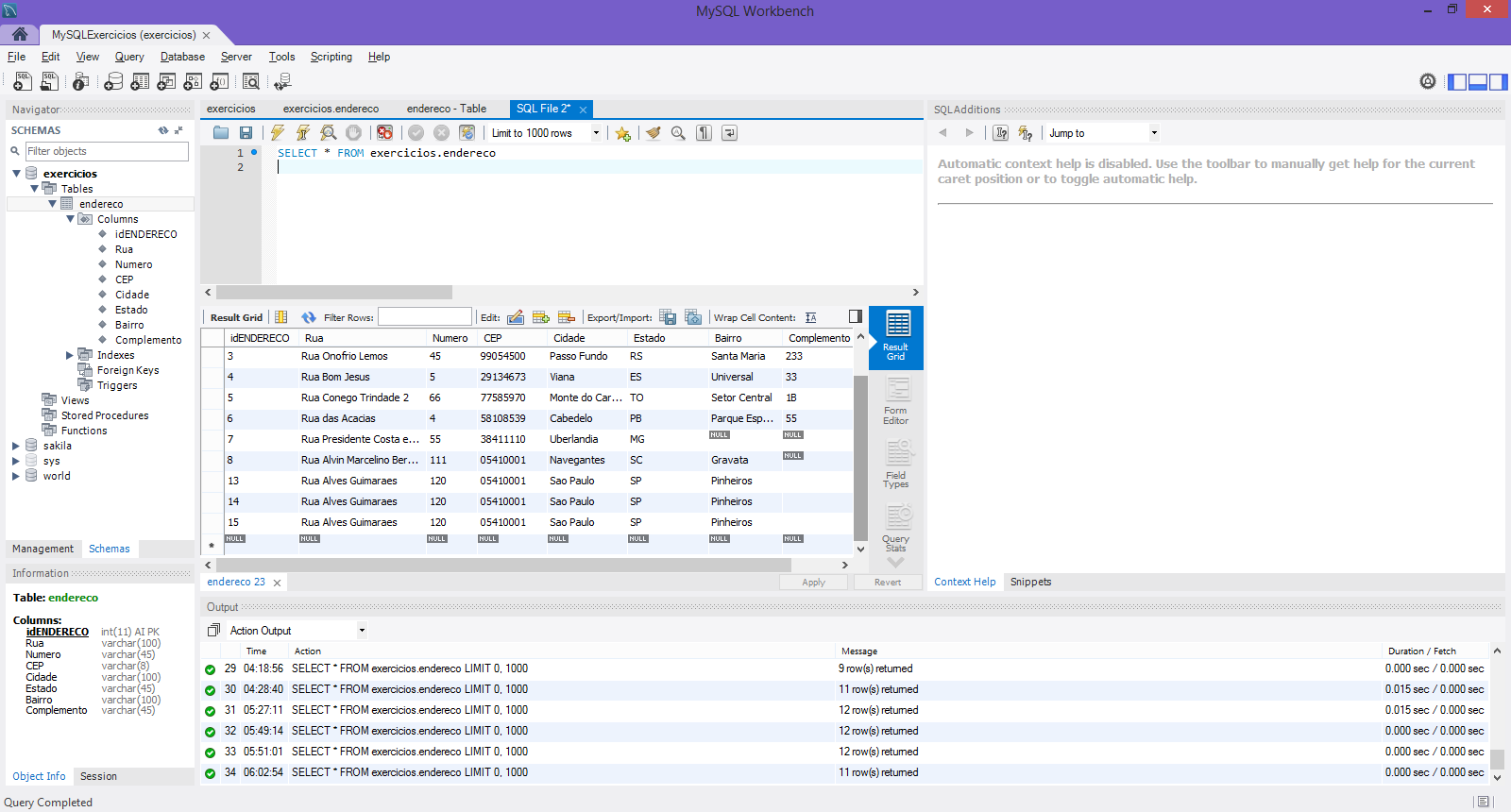
"complemento": "",

"numero": ""

}

**Saída:** Endereço deletado com sucesso

****

****

CASO OK.