



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS – PUC MINAS
***campus* POÇOS DE CALDAS – MG**
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Projeto de Sistemas Distribuídos

Projeto Cliente REST

Sistemas Distribuídos e Comunicação Cliente/Servidor

Aluno: Lucas Omar Andrade Leal

Orientador: Udo Fritzke Junior

1º semestre/2025

2025-S1: TAREFA AVALIATIVA 2 - RELATÓRIO SOBRE A OBSERVAÇÃO DA EXPERIÊNCIA DOS ALUNOS AO USAREM O ÓCULOS DE VR (TRABALHO EM DUPLA) - 5 PONTOS + 5 PONTOS EXTRA (OPCIONAL)

Vencimento 26 jun por 23:59

Pontos 30

Enviando uma caixa de entrada de texto, um URL de site, ou um upload de arquivo

Implementar uma aplicação cliente que utilize algum serviço Web que forneça uma API que siga o estilo arquitetural REST (REpresentational State Transfer).

A aplicação deverá incluir:

- um módulo cliente (desktop ou mobile) que acesse a API REST;
- acessos incluindo no mínimo chamadas de métodos GET;

Deverá ser entregue um documento com:

- descrição geral da API REST do serviço Web utilizado;
- descrição de cada chamada do cliente à API. Esta descrição deverá incluir: a URL do recurso, a forma de autenticação (como chaves) e exemplos de representação do recurso;
- instruções para a execução de um teste operacional da implementação (execução do cliente).

Deverá ser entregue o projeto completo em uma das seguintes formas:

- Projeto NetBeans compactado com arquivo jar executável (com instruções de execução);
- Repositório Github com o projeto clonável para o NetBeans;

A apresentação deverá ser feita através da gravação de um pitch de no máximo 5 minutos. O link para o pitch deve ser informado na entrada de texto.

Esta tarefa poderá ser feita em equipe de no máximo 3 alunos.

Dicas:

O hub <https://rapidapi.com/hub> possui vários serviços disponíveis por APIs REST. Os projetos abaixo utilizam uma das APIs (GeoDB Cities):

- [RESTCliente.zip](#) (projeto Java no Netbeans)
- Repositório GitHub de app Android de acesso à API GeoDB Cities: https://github.com/udofritzke/AcessoGeoDB_v2.
- Instruções sobre acesso ao hub Rapidapi podem ser encontrados nos slides [9 Serviços Web.pdf](#).

SUMÁRIO

1 DESCRIÇÃO GERAL DA API REST UTILIZADA	4
2 DESCRIÇÃO DE CADA CHAMADA À API	4
Recursos da API	4
Exemplo de chamada realizada pela aplicação	4
Parâmetros utilizados	4
Autenticação	4
Exemplo de Resposta (JSON)	4
3 INSTRUÇÕES PARA EXECUÇÃO DE UM TESTE OPERACIONAL DA IMPLEMENTAÇÃO	5
Pré-requisitos	5
Passos para executar o cliente:	5
4 CONSIDERAÇÕES FINAIS	6

1 DESCRIÇÃO GERAL DA API REST UTILIZADA

A aplicação cliente Java utiliza a API GeoDB Cities, disponibilizada pela RapidAPI, para consultar informações sobre cidades com base em um prefixo de nome fornecido pelo usuário. A API oferece dados como nome da cidade, país, código do país, região, latitude, longitude e população.

A comunicação com a API é feita através de requisições HTTP do tipo GET, com cabeçalhos de autenticação necessários para garantir acesso seguro ao serviço.

2 DESCRIÇÃO DE CADA CHAMADA À API

Recursos da API

- URL base da API:

```
https://wft-geo-db.p.rapidapi.com/v1/geo/cities
```

Exemplo de chamada realizada pela aplicação

```
GET
https://wft-geo-db.p.rapidapi.com/v1/geo/cities?namePrefix=São%20José%20do%20Vale%20do%20Rio%20Preto&limit=5
```

Parâmetros utilizados

Parâmetro	Descrição
namePrefix	Prefixo do nome da cidade a ser pesquisada.
limit	Quantidade máxima de resultados retornados (opcional, padrão 10).

Autenticação

A API exige autenticação via chaves de acesso nos cabeçalhos da requisição HTTP.

Cabeçalho	Valor
X-RapidAPI-Key	1bfdcb4a86mshf4ee4ce59811ec7p188cecjsn49c81557acb2
X-RapidAPI-Host	wft-geo-db.p.rapidapi.com

Exemplo de Resposta (JSON)

```
{
  "data": [
    {
      "name": "São José do Vale do Rio Preto",

```

```
    "country": "Brazil",  
    "countryCode": "BR",  
    "region": "Rio de Janeiro",  
    "regionCode": "RJ",  
    "latitude": -22.1525,  
    "longitude": -42.9328,  
    "population": 22214  
  }  
  // ... outras cidades (até 5)  
]  
}
```

A aplicação formata e exibe cada item retornado com os seguintes campos:

- Nome da cidade;
- País e código do país;
- Região e código da região;
- Coordenadas geográficas;
- População.

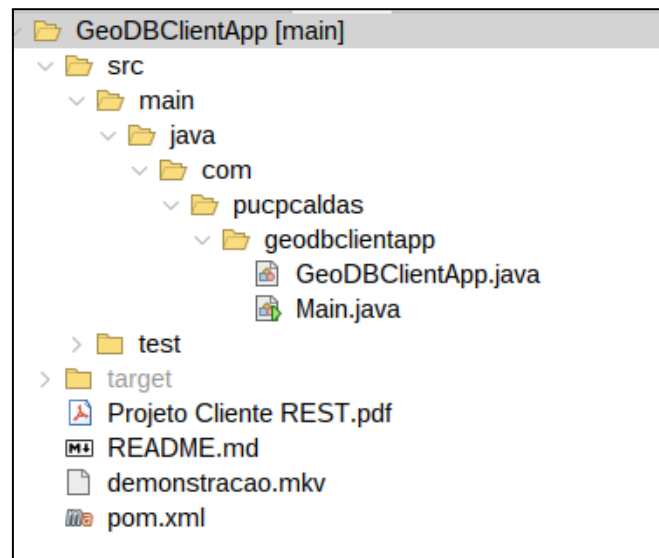
3 INSTRUÇÕES PARA EXECUÇÃO DE UM TESTE OPERACIONAL DA IMPLEMENTAÇÃO

Pré-requisitos

- JDK 21.0.7 ou superior instalado
- Internet ativa
- Dependência da biblioteca org.json (pode ser incluída manualmente ou via Maven/Gradle)

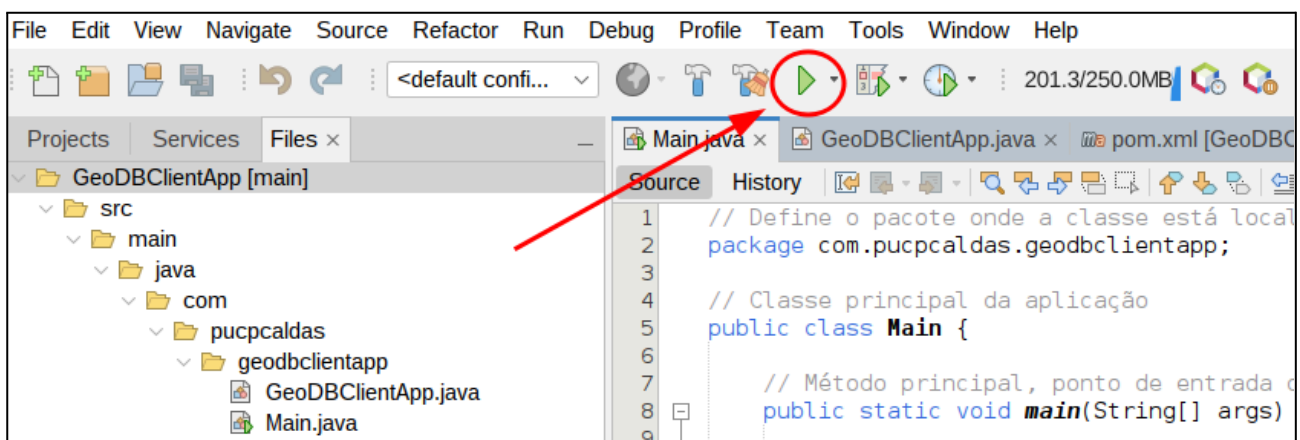
Passos para executar o cliente:

I. Crie a estrutura de pacotes

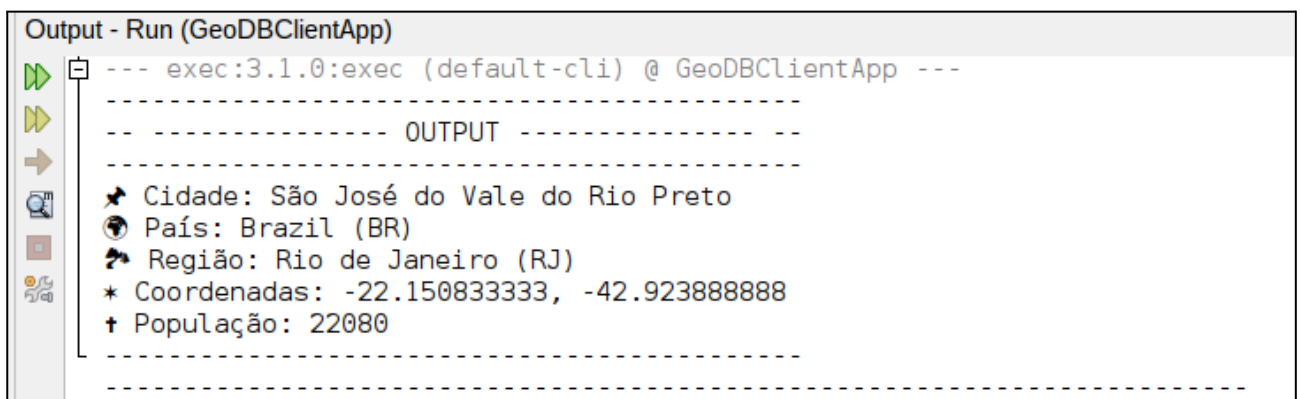


II. Copie o conteúdo dos arquivos Main.java e GeoDBClientApp.java nos respectivos locais.

III. Execute a aplicação:



IV. Saída esperada (exemplo):



4 ANEXOS

- GitHub: <https://github.com/lucasleall/puc-sd-20260626-java-client-rest>
- Download vídeo demonstração:
<https://github.com/lucasleall/puc-sd-20260626-java-client-rest/raw/refs/heads/main/demonstracao.tracao.mkv>