

MÓDULO 1 - EXTRA NETWORKING API



LENDO DE UMA API REST EM JAVA

CLIENTE

VIACEP

GET <https://viacep.com.br/ws/01001000/json/>

```
{  
  "cep": "01001-000",  
  "logradouro": "Praça da Sé",  
  "complemento": "lado ímpar",  
  "unidade": "",  
  "bairro": "Sé",  
  "localidade": "São Paulo",  
  "uf": "SP",  
  "estado": "São Paulo",  
  "regiao": "Sudeste",  
  "ibge": "3550308",  
  "gia": "1004",  
  "ddd": "11",  
  "siafi": "7107"  
}
```

java.net

URL e URLConnection

JAVA 1.0

CLIENTE

URL e URLConnection

```
String viaCepURL = "https://viacep.com.br/ws/01001000/json/";
```

```
URL url = new URL(viaCepURL);
```

```
URLConnection urlConnection = url.openConnection();
```

Do pacote java.net, desde o Java 1.0

```
InputStream inputStream = urlConnection.getInputStream();
```

Abre a conexão aqui

```
Scanner scanner = new Scanner(inputStream);
```

```
while(scanner.hasNextLine()) {
```

```
    String line = scanner.nextLine();
```

```
    System.out.println(line);
```

```
}
```

```
scanner.close();
```

Usando Scanner como se fosse qualquer outro InputStream

Fechando conexão corretamente

```
String viaCepURL = "https://viacep.com.br/ws/01001000/json/";
```

```
Scanner scanner = null;  
try {  
    scanner = new Scanner(new URL(viaCepURL).openStream());  
    while(scanner.hasNextLine()) {  
        String line = scanner.nextLine();  
        System.out.println(line);  
    }  
} finally {  
    if (scanner != null) {  
        scanner.close();  
    }  
}
```

*Abre o InputStream de
uma vez*

*Fechando Scanner tanto em caso
de sucesso como de exceção*

Try-with-resources

```
String viaCepURL = "https://viacep.com.br/ws/01001000/json/";  
  
try (Scanner scanner = new Scanner(new URL(viaCepURL).openStream())) {  
    while (scanner.hasNextLine()) {  
        String line = scanner.nextLine();  
        System.out.println(line);  
    }  
}
```



Fecha o InputStream de maneira automática

java.net.http

HttpClient

JAVA 11

CLIENTE

HttpClient

URI ao invés de URL

```
URI viaCepURI = URI.create("https://viacep.com.br/ws/01001000/json/");
```

```
try (HttpClient client = HttpClient.newBuilder().build()) {
```

Do pacote java.net.http

```
HttpRequest request = HttpRequest.newBuilder().uri(viaCepURI).build();
```

```
HttpResponse<String> response = client.send(request,
```

```
    HttpResponse.BodyHandlers.ofString());
```

```
System.out.println(response.statusCode());
```

```
System.out.println(response.body());
```

```
}
```

*Conceitos HTTP: request,
response, status code, body*

com.sun.net.httpserver

jwebserver e HttpServer

JAVA 18

SERVIDOR

Salvando itens do cardápio em JSON

```
Database database = new Database();
List<ItemCardapio> itensCardapio = database.listaItensCardapio();
```

```
Gson gson = new Gson();
String json = gson.toJson(itensCardapio);
Path arquivoItens = Path.of("itensCardapio.json");
Files.writeString(arquivoItens, json);
```

Arquivo
itensCardapio.json
salvo na raiz do projeto

Na Main

```
[{"id":1,"nome":"Refresco do Chaves", "descricao":"Suco de limão que parece de tamarindo e tem gosto de groselha.", "categoria":"BEBIDAS", "preco":2.99}, {"id":2, "nome":"Sanduíche de Presunto do Chaves", "descricao":"Sanduíche de presunto simples, mas feito com muito amor.", "categoria":"PRATOS_PRINCIPAIS", "preco":3.50, "precoPromocional":2.99}, {"id":5, "nome":"Torta de Frango da Dona Florinda", "descricao":"Torta de frango com recheio cremoso e massa crocante."}, ...]
```

Subindo um jwebserver

\$ jwebserver

Binding to loopback by default. For all interfaces use "-b 0.0.0.0" or "-b ::".

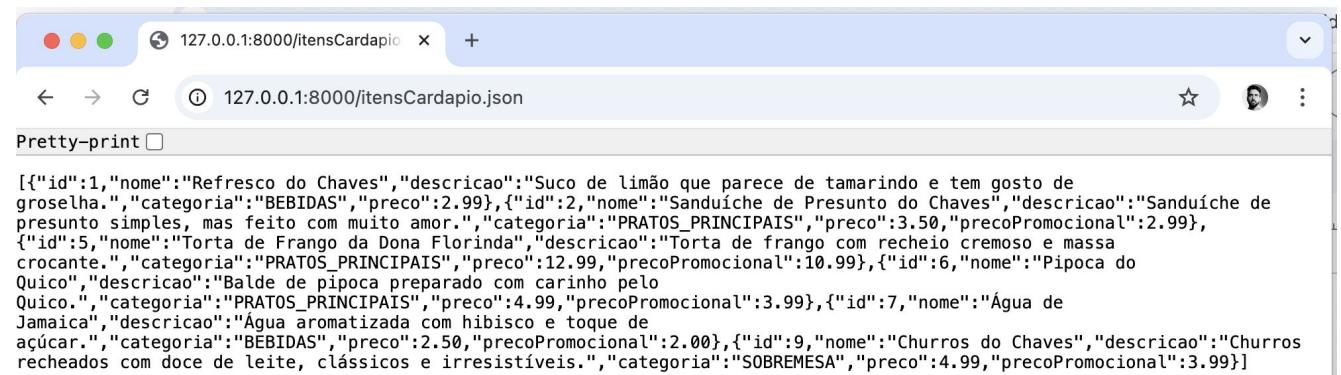
Serving /Users/alexandreaquiles/Documents/unipds/curso-modulo1-extra/cardapio and subdirectories on 127.0.0.1 port 8000

URL <http://127.0.0.1:8000/>



Directory listing for /

- [itemsCardapio.json](#)
- [README.md](#)
- [gradle/](#)
- [gradlew](#)
- [build.gradle](#)
- [build/](#)
- [gradlew.bat](#)
- [settings.gradle](#)
- [src/](#)



Chamando o servidor com HttpClient

```
URI viaCepURI = URI.create("http://127.0.0.1:8000/itensCardapio.json");
```

Endereço do jwebserver

```
200
[{"id":1,"nome":"Refresco do Chaves","descricao":"Suco de limão que parece de tamarindo e tem gosto de groselha.", "categoria":"BEBIDAS", "preco":2.99}, {"id":2,"nome":"Sanduíche de Presunto do Chaves", "descricao":"Sanduíche de presunto simples, mas feito com muito amor.", "categoria":"PRATOS_PRINCIPAIS", "preco":3.50, "precoPromocional":2.99}, {"id":5,"nome":"Torta de Frango da Dona Florinda", "descricao":"Torta de frango com recheio cremoso e massa crocante.", "categoria":"PRATOS_PRINCIPAIS", "preco":12.99, "precoPromocional":10.99}, {"id":6,"nome":"Pipoca do Quico", "descricao":"Balde de pipoca preparado com carinho pelo Quico.", "categoria":"PRATOS_PRINCIPAIS", "preco":4.99, "precoPromocional":3.99}, {"id":7,"nome":"Água de Jamaica", "descricao":"Água aromatizada com hibisco e toque de açúcar.", "categoria":"BEBIDAS", "preco":2.50, "precoPromocional":2.00}, {"id":9,"nome":"Churros do Chaves", "descricao":"Churros recheados com doce de leite, clássicos e irresistíveis.", "categoria":"SOBREMESA", "preco":4.99, "precoPromocional":3.99}]
```

Subindo um HttpServer

```
HttpServer httpServer = HttpServer.create(new InetSocketAddress(8000), 0);

httpServer.createContext("/itensCardapio.json", exchange -> {
    Path path = Path.of("itensCardapio.json");
    String json = Files.readString(path);

    byte[] bytes = json.getBytes();

    Headers responseHeaders = exchange.getResponseHeaders();
    responseHeaders.add("Content-Type", "application/json; charset=UTF-8");

    exchange.sendResponseHeaders(200, bytes.length);
    try(OutputStream responseBody = exchange.getResponseBody()){
        responseBody.write(bytes);
    }
});

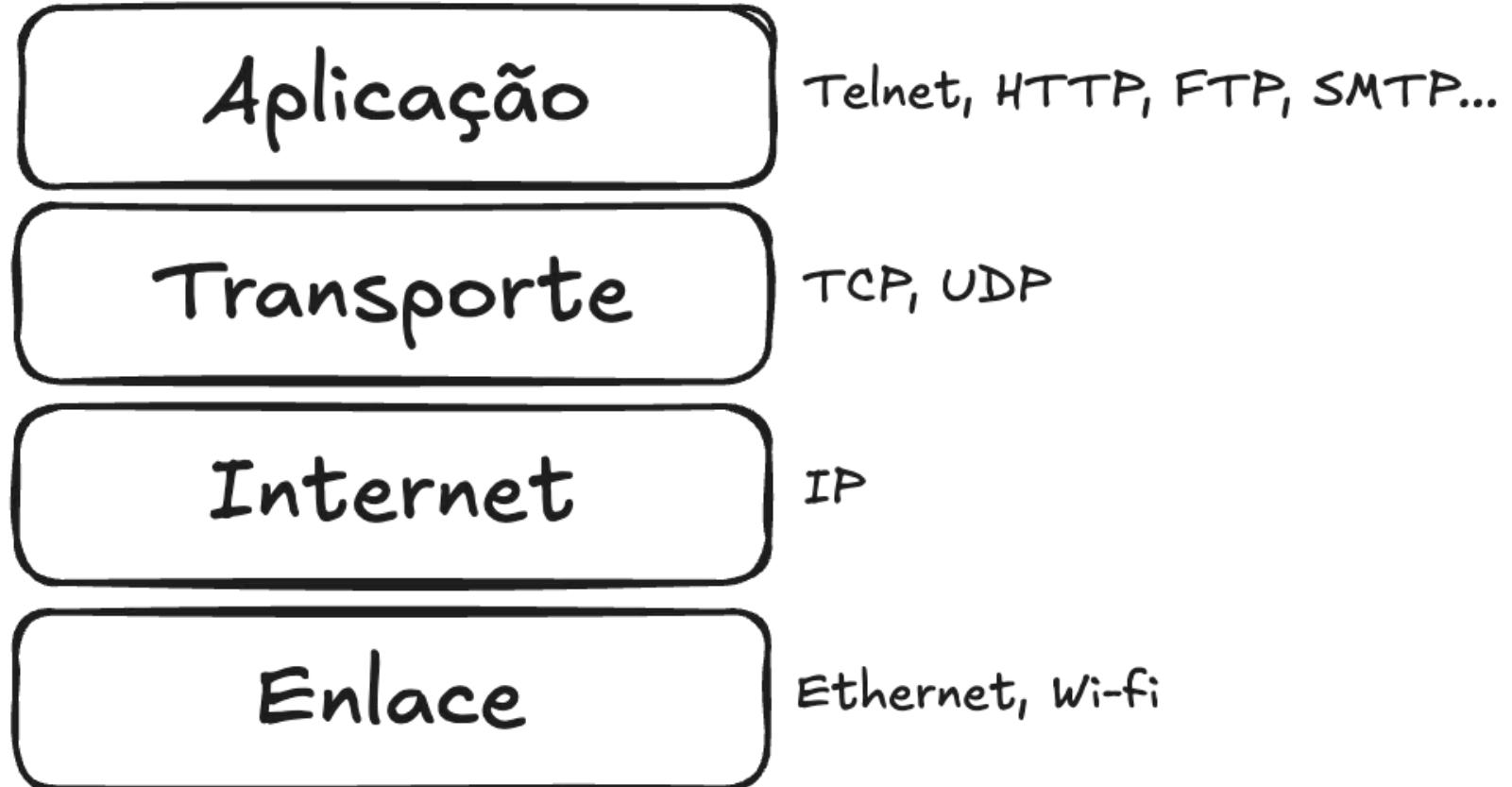
System.out.println("Subiu servidor http! ");
httpServer.start();
```

Testando com cURL

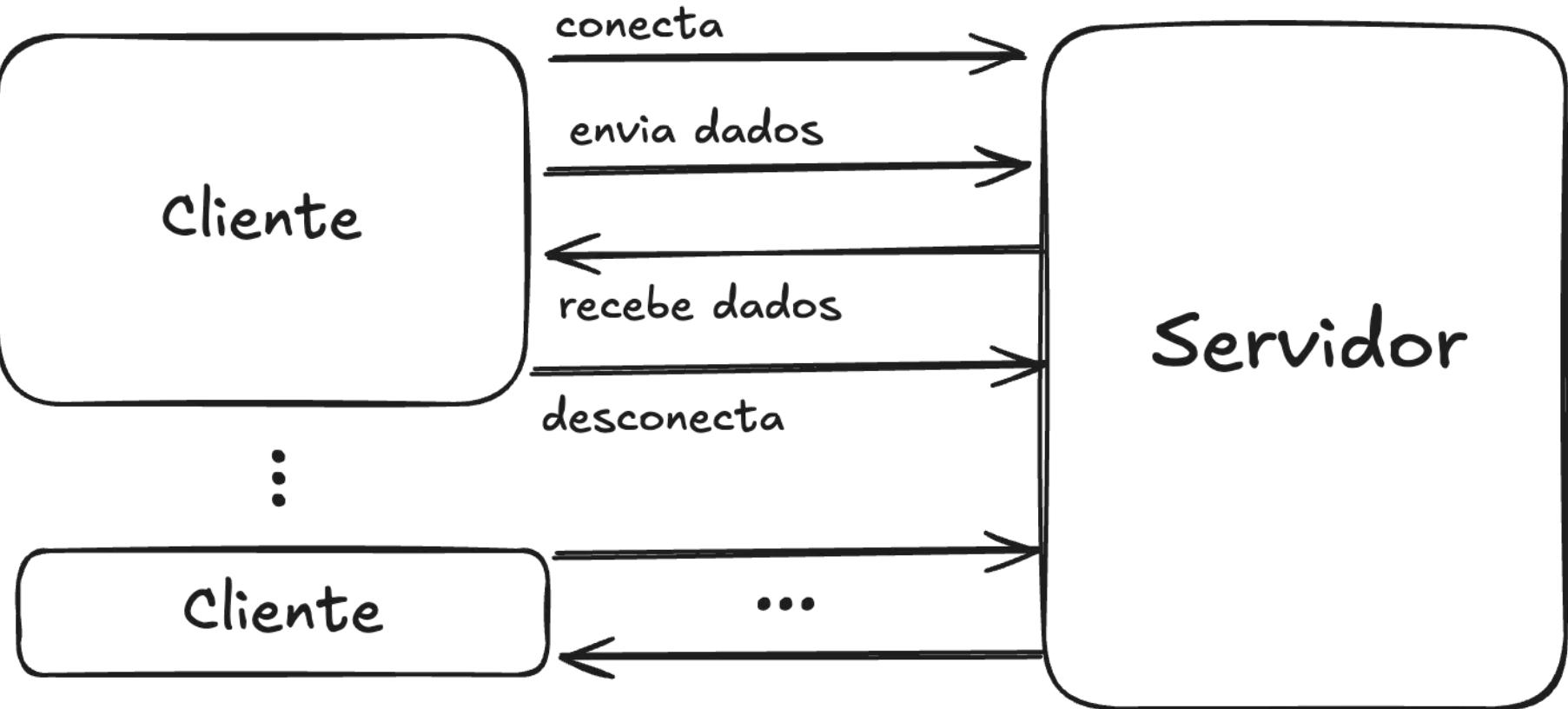
```
$ curl -v 'http://localhost:8000/itensCardapio.json'
* Host localhost:8000 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
* Trying [::1]:8000...
* Connected to localhost (::1) port 8000
> GET /itensCardapio.json HTTP/1.1
> Host: localhost:8000
> User-Agent: curl/8.7.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< Date: Wed, 13 Aug 2025 00:17:02 GMT
< Content-type: application/json; charset=UTF-8
< Content-length: 1036
<
* Connection #0 to host localhost left intact
[{"id":1,"nome":"Refresco do Chaves","descricao":"Suco de limão que parece de tamarindo e tem gosto de groselha.", "categoria":"BEBIDAS", "preco":2.99}, {"id":2,"nome":"Sanduíche de Presunto do Chaves", "descricao":"Sanduíche de presunto simples..."}]%
```

TCP/IP e Sockets

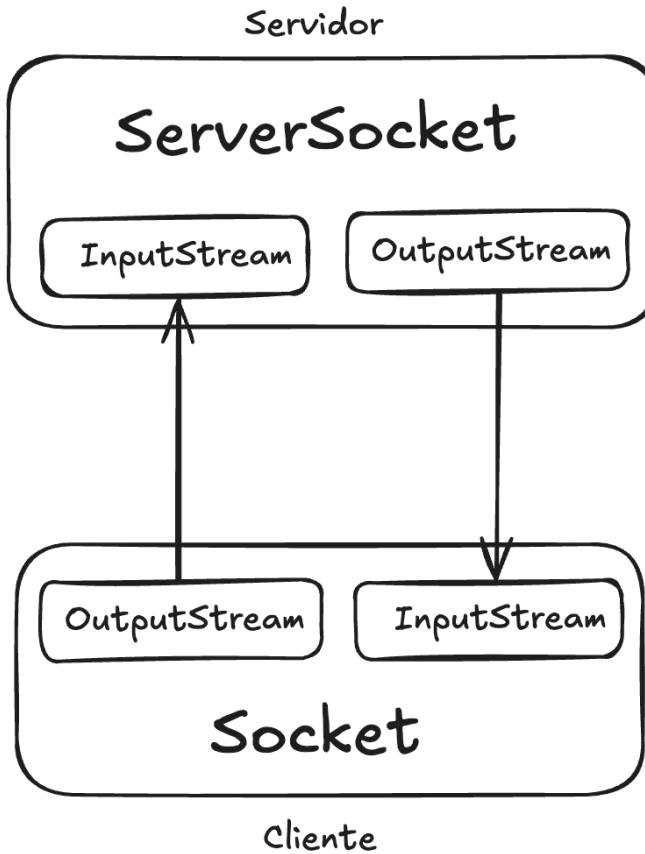
Modelo TCP/IP



Fluxo TCP



Sockets



endereço físico

IP

172.217.28.196 : 80

www.google.com

endereço da aplicação

Porta

RFC 2616: o protocolo HTTP/1.1

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

Method = "OPTIONS" ; [Section 9.2](#)
| "GET" ; [Section 9.3](#)
| "HEAD" ; [Section 9.4](#)
| "POST" ; [Section 9.5](#)
| "PUT" ; [Section 9.6](#)
| "DELETE" ; [Section 9.7](#)
| "TRACE" ; [Section 9.8](#)
| "CONNECT" ; [Section 9.9](#)
| extension-method

extension-method = token

Request-URI = "*" | absoluteURI | abs_path | authority

GET /pub/WWW/TheProject.html HTTP/1.1

Testando com Telnet

```
$ telnet
```

```
telnet> open localhost 8000
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
```

```
Escape character is '^]'.
```

Conectando ao servidor

```
GET /itensCardapio.json HTTP/1.1
```

```
HTTP/1.1 200 OK
```

```
Date: Sun, 10 Aug 2025 18:15:29 GMT
```

```
Content-type: application/json; charset=UTF-8
```

```
Content-length: 1036
```

```
[{"id":1,"nome":"Refrresco do Chaves","descricao":"Suco de limão que parece de tamarindo e tem gosto de ...
```

*Solicitando o recurso
/itensCardapio.json*

Usando um ServerSocket

```
int porta = 8000;
try(ServerSocket serverSocket = new ServerSocket(porta)) {
    System.out.println("Iniciando servidor na porta " + porta);

    try (Socket clientSocket = serverSocket.accept()) {
        System.out.println("Conectou com cliente " +
                           clientSocket.getInetAddress());

        InputStream in = clientSocket.getInputStream();

        StringBuilder requestBuilder = new StringBuilder();
        int data;
        do {
            data = in.read();
            requestBuilder.append((char) data);
        } while(in.available()>0);

        String request = requestBuilder.toString();
        System.out.println(request);
    }
}
```

Cria
ServerSocket

Nova conexão com um
cliente

Jeito mais baixo nível
de ler dados

Testando o ServerSocket

localhost:8000



This site can't be reached

localhost refused to connect.

Try:

- Checking the connection
- [Checking the proxy and the firewall](#)

ERR_CONNECTION_REFUSED

Details

```
Iniciando servidor na porta 8000
Conectou com cliente /0:0:0:0:0:0:0:1
GET / HTTP/1.1
Host: localhost:8000
Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Not;A=Brand";v="99", "Google Chrome";v="139",
"Chromium";v="139"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "macOS"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: en-US,en;q=0.9,pt;q=0.8
```

Response no ServerSocket

```
Database database = new Database();
List<ItemCardapio> listaItensCardapio = database.listaItensCardapio();
Gson gson = new Gson();
String json = gson.toJson(listaItensCardapio);

PrintStream out = new PrintStream(clientSocket.getOutputStream());
out.println("HTTP/1.1 200 OK");
out.println("Content-Type: application/json; charset=UTF-8");
out.println("");
out.print(json);
```

Lista itens e gera JSON

Envia o response

Testando com cURL

```
$ curl -v 'http://localhost:8000/itensCardapio.json'
* Host localhost:8000 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
* Trying [::1]:8000...
* Connected to localhost (::1) port 8000
> GET /itensCardapio.json HTTP/1.1
> Host: localhost:8000
> User-Agent: curl/8.7.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< Content-Type: application/json; charset=UTF-8
<
* no chunk, no close, no size. Assume close to signal end
* Closing connection
[{"id":1,"nome":"Refresco do Chaves","descricao":"Suco de limão que parece de tamarindo e tem gosto de groselha.","categoria":"BEBIDAS","preco":2.99}, {"id":2,"nome":"Sanduíche de Presunto do Chaves","descricao":"Sanduíche de presunto simples, mas feito com muito amor.","categoria":"PRATOS_PRINCIPAIS","preco":3.50,"precoPromocional":2.99}, ... ]%
```

Socket do lado do Cliente

```
try(Socket clientSocket = new Socket("localhost", 8000)) {  
    try (PrintStream out =  
        new PrintStream(clientSocket.getOutputStream())) {  
  
        out.println("GET /itensCardapio.json HTTP/1.1\\n");  
    }  
}
```

Cria Socket

Lendo do Socket no Cliente

```
try(Socket clientSocket = new Socket("localhost", 8000)) {  
    try (PrintStream out =  
        new PrintStream(clientSocket.getOutputStream());  
  
    Scanner in = new Scanner(clientSocket.getInputStream())); {  
  
        out.println("GET /itensCardapio.json HTTP/1.1\\n");  
  
        while (in.hasNextLine()) {  
            String line = in.nextLine();  
            System.out.println(line);  
        }  
    }  
}
```

*Lendo do
InputStream*

*Imprime na saída padrão o
que for enviado do Servidor*