



**Instituto Politécnico Nacional**

ESCUELA SUPERIOR DE CÓMPUTO

# CLASE 21 - CLOUD TRANSLATION API

*Sistemas distribuidos*

*Profesor: Ukranio Coronilla Contreras*

Luciano Hernández Jonathan

Abril 1, 2025

# 1 Captura de pantalla con respuesta.

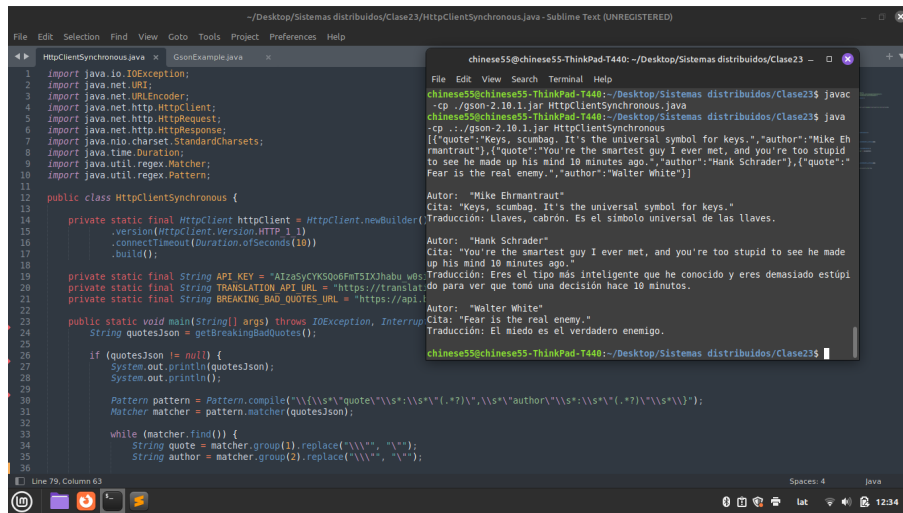


Figure 1: Caption

## 2 Codigo del cliente.

```

1 import java.io.IOException;
2 import java.net.URI;
3 import java.net.URLEncoder;
4 import java.net.http.HttpClient;
5 import java.net.http.HttpRequest;
6 import java.net.http.HttpResponse;
7 import java.nio.charset.StandardCharsets;
8 import java.time.Duration;
9 import java.util.regex.Matcher;
10 import java.util.regex.Pattern;
11
12 public class HttpClientSynchronous {
13
14     private static final HttpClient httpClient = HttpClient.newBuilder()
15         .version(HttpClient.Version.HTTP_1_1)
16         .connectTimeout(Duration.ofSeconds(10))
17         .build();
18
19     private static final String API_KEY = "AIZA5YCYKSQo6FmT5IXJhabu_w0siFnQTd5Up-E";
20     private static final String TRANSLATION_API_URL = "https://translation.googleapis.com/";
21     private static final String BREAKING_BAD_QUOTES_URL = "https://api.breakingbadquotes.xyz/";
22
23     public static void main(String[] args) throws IOException, InterruptedException {
24         String quotesJson = getBreakingBadQuotes();
25
26         if (quotesJson != null) {
27             System.out.println(quotesJson);
28             System.out.println();
29
30             Pattern pattern = Pattern.compile("\\{\\s*\"quote\"\\s*:\\s*\"(.*)\"\\s*,\\s*\"author\"\\s*:\\s*\"(.*)\"\\s*\\}");
31             Matcher matcher = pattern.matcher(quotesJson);
32
33             while (matcher.find()) {
34                 String quote = matcher.group(1).replace("\\\\", "\\");
35                 String author = matcher.group(2).replace("\\\\", "\\");
36
37                 String translated = translateText(quote);
38
39                 System.out.println("Autor: " + author + " " + translated);

```

```

40         System.out.println("Cita: \"" + quote + "\"");
41         System.out.println("Traduccin: " + translated);
42         System.out.println(); // Separador
43     }
44     } else {
45         System.out.println("No se pudo obtener el JSON.");
46     }
47 }
48
49 private static String getBreakingBadQuotes() throws IOException, InterruptedException {
50     HttpRequest request = HttpRequest.newBuilder()
51         .GET()
52         .uri(URI.create(BREAKING_BAD_QUOTES_URL))
53         .setHeader("User-Agent", "Java 11 HttpClient Bot")
54         .build();
55
56     HttpResponse<String> response = httpClient.send(request, HttpResponse.BodyHandlers.
ofString());
57
58     return (response.statusCode() == 200) ? response.body() : null;
59 }
60
61 private static String translateText(String text) throws IOException, InterruptedException
62 {
63     String encodedText = URLEncoder.encode(text, StandardCharsets.UTF_8);
64     String apiUrl = TRANSLATION_API_URL + "?key=" + API_KEY + "&target=es&q=" +
encodedText;
65
66     HttpRequest request = HttpRequest.newBuilder()
67         .POST(HttpRequest.BodyPublishers.noBody())
68         .uri(URI.create(apiUrl))
69         .header("Content-Type", "application/json")
70         .header("Accept", "application/json")
71         .build();
72
73     HttpResponse<String> response = httpClient.send(request, HttpResponse.BodyHandlers.
ofString());
74
75     if (response.statusCode() == 200) {
76         String responseBody = response.body();
77         Pattern translationPattern = Pattern.compile("\"translatedText\"\\s*:\\s*\"(.*)\"");
78
79         Matcher matcher = translationPattern.matcher(responseBody);
80         if (matcher.find()) {
81             return matcher.group(1).replace("&#39;", "");
82         }
83     }
84     return "Error en la traduccin.";
85 }

```