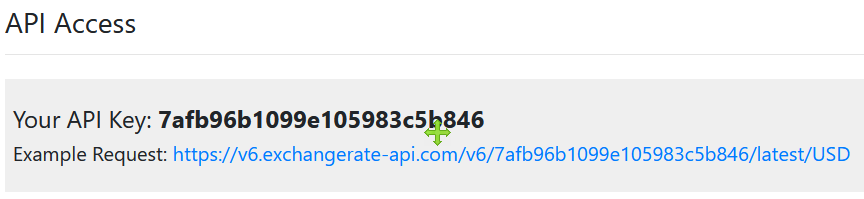
# Pasos Para Construir El Conversor De Monedas.

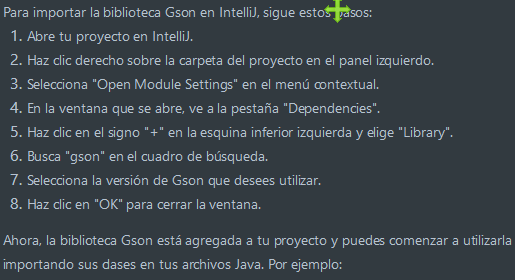
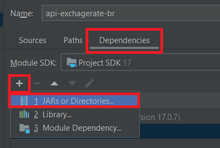
**1. Crear el repositorio en github.**

**2. Configurar el entorno JAVA.**

**3.Conociendo la API Rest:** <https://app.exchangerate-api.com/dashboard/confirmed>

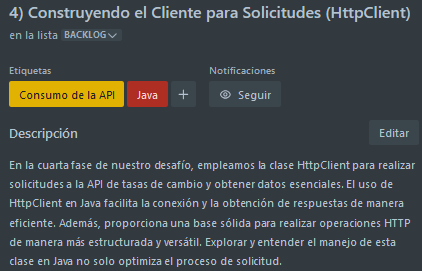
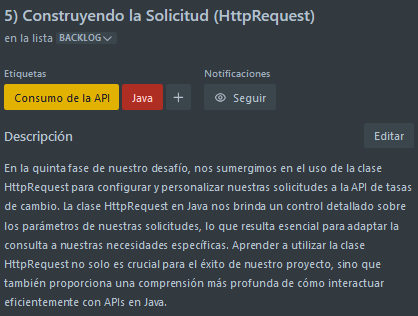


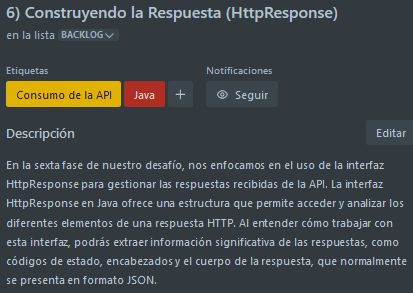
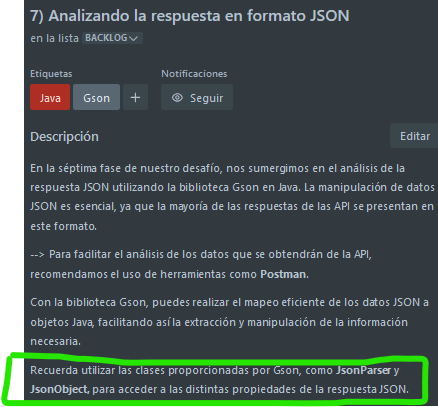
<https://v6.exchangerate-api.com/v6/7afb96b1099e105983c5b846/latest/USD>

**4. Importar la biblioteca Gson en el proyecto:**

**4.1 hacer búsqueda en google**

**4.2 descargar Gson en archivo .jar**

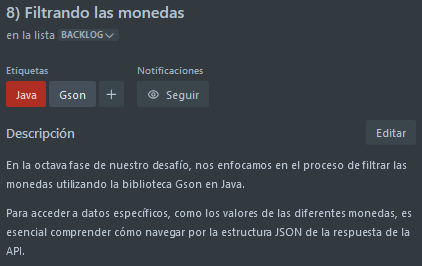
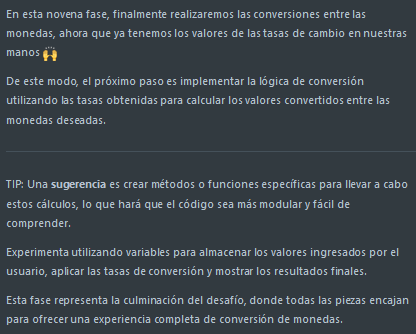
**5. Construir el cliente para solicitudes (HttpClient): 6. Construir solicitud (HttpRequest):**

**7. Construir respuesta (HttpClient): 8. Analizar la respuesta recibida en formato Json:**

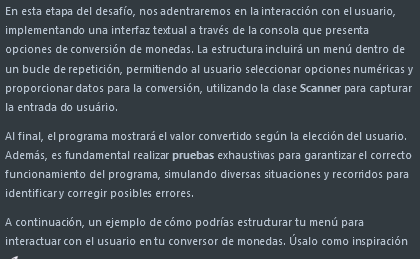
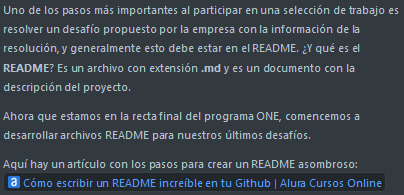
**Códigos de los pasos 1 al 7:**

import com.google.gson.Gson;  
import java.io.IOException;  
import java.net.URI;  
import java.net.http.HttpClient;  
import java.net.http.HttpRequest;  
import java.net.http.HttpResponse;  
  
public class Principal {  
 public static void main(String[] args) throws IOException, InterruptedException {  
  
 System.*out*.println("hola mundo");  
 HttpClient client = HttpClient.*newHttpClient*();  
 HttpRequest request = HttpRequest.*newBuilder*()  
 .uri(URI.*create*("https://v6.exchangerate-api.com/v6/7afb96b1099e105983c5b846/latest/USD"))  
 .build();  
 HttpResponse<String> response = client  
 .send(request, HttpResponse.BodyHandlers.*ofString*());  
 System.*out*.println(response.body());  
 }  
}

**9. Filtrando las monedas: 10. Convirtiendo los valores:**

****

**11. Interactuando con el usuario: 12. Hacer un README**

****

ANEXOS

explicame el código y si tambien tiene algo que ver con la consulta de atributos de un Json:

JsonParser parser = new JsonParser();

JsonElement jsonElement = parser.parse(response.body()); System.out.println(jsonElement);

¡Claro! Vamos a desglosar el código y ver cómo se relaciona con la consulta de atributos de un JSON.

### Explicación del Código

1. \*\*Crear un Parser JSON\*\*:

```java

JsonParser parser = new JsonParser();

```

Aquí, creamos una instancia de `JsonParser`. Esta clase de la biblioteca Gson se utiliza para analizar cadenas JSON y convertirlas en objetos de tipo `JsonElement`.

2. \*\*Analizar la Respuesta JSON\*\*:

```java

JsonElement jsonElement = parser.parse(response.body());

```

En esta línea, utilizamos el método `parse` del `JsonParser` para convertir la cadena JSON de la respuesta (`response.body()`) en un objeto de tipo `JsonElement`.

3. \*\*Imprimir el Elemento JSON\*\*:

```java

System.out.println(jsonElement);

```

Finalmente, imprimimos el `jsonElement` en la consola. Esto mostrará la representación JSON del objeto.

### Relación con la Consulta de Atributos de un JSON

El uso de `JsonParser` y `JsonElement` de la biblioteca Gson está muy relacionado con la consulta de atributos de un JSON. Una vez que tienes un `JsonElement`, puedes convertirlo en otros tipos de objetos Gson para acceder a sus atributos.

Por ejemplo, puedes convertir un `JsonElement` en un `JsonObject` si sabes que el JSON representa un objeto:

```java

JsonObject jsonObject = jsonElement.getAsJsonObject();

String nombre = jsonObject.get("nombre").getAsString();

int edad = jsonObject.get("edad").getAsInt();

String correo = jsonObject.get("correo").getAsString();

System.out.println("Nombre: " + nombre);

System.out.println("Edad: " + edad);

System.out.println("Correo: " + correo);

```

### Código Completo con Consulta de Atributos

Aquí te dejo el código completo con la consulta de atributos de un JSON:

```java

import com.google.gson.JsonElement;

import com.google.gson.JsonObject;

import com.google.gson.JsonParser;

public class Main {

public static void main(String[] args) {

String jsonResponse = "{\"nombre\":\"Juan Pérez\", \"edad\":30, \"correo\":\"juan.perez@example.com\"}";

// Crear una instancia de JsonParser

JsonParser parser = new JsonParser();

// Analizar el JSON

JsonElement jsonElement = parser.parse(jsonResponse);

// Convertir JsonElement a JsonObject

JsonObject jsonObject = jsonElement.getAsJsonObject();

// Consultar los atributos

String nombre = jsonObject.get("nombre").getAsString();

int edad = jsonObject.get("edad").getAsInt();

String correo = jsonObject.get("correo").getAsString();

// Imprimir los valores

System.out.println("Nombre: " + nombre);

System.out.println("Edad: " + edad);

System.out.println("Correo: " + correo);

}

}

```

### Resumen

- \*\*`JsonParser`\*\*: Analiza la cadena JSON y la convierte en un `JsonElement`.

- \*\*`JsonElement`\*\*: Representa cualquier elemento JSON (objeto, matriz, valor primitivo, nulo).

- \*\*Consultar Atributos\*\*: Convertir `JsonElement` en `JsonObject` para acceder a los atributos específicos utilizando métodos como `getAsString`, `getAsInt`, etc.

Este enfoque te permite trabajar con JSON sin necesidad de convertirlo directamente en una clase Java, aunque convertirlo a una clase puede ser más conveniente en casos de estructuras JSON más complejas o cuando se trabaja con datos de manera repetitiva.

Si tienes más preguntas o necesitas más ayuda, ¡aquí estoy para asistirte! 🚀 ¿Hay algo más específico en lo que pueda ayudarte?