

# Programación Multimedia y Dispositivos Móviles Práctica 7

Servicios en red II.

Fecha límite de entrega 20/01/2023

# Ejercicio 1. ¿Qué diferencias y similitudes existen entre las conexiones HTTP y HTTPS?

HTTP (Protocolo de Transferencia de Hipertexto) y HTTPS (Lo mismo pero *Seguro*) son dos protocolos para transmitir datos por Internet, pero HTTPS utiliza una capa adicional de seguridad llamada SSL (Capa de Sockets Seguros) o TLS (Capa de Transporte Seguro).

Similitudes:

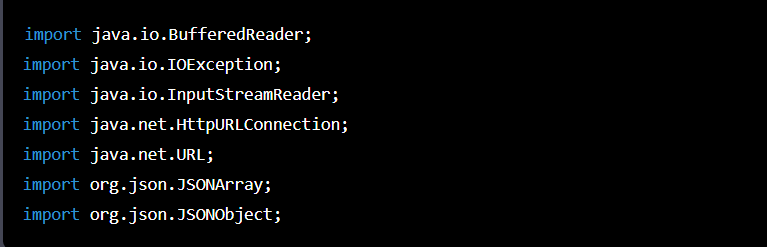
* Tanto HTTP como HTTPS utilizan el mismo conjunto de métodos de solicitud (GET, POST, otros...).
* Ambos utilizan los mismos códigos de estado (por ejemplo 404 Not Found)
* Ambos utilizan el mismo formato URI (Identificador Uniforme de Recursos).
* Ambos son protocolos de la capa de aplicación del modelo OSI.

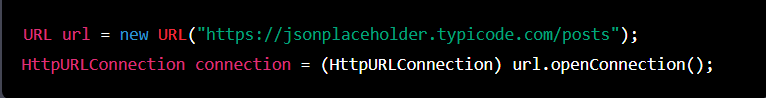
Diferencias:

* HTTPS utiliza un certificado SSL/TLS para cifrar los datos que se envían, mientras que HTTP no.
* Las conexiones HTTPS utilizan un puerto por defecto diferente (443) al HTTP (80).
* Las conexiones HTTPS se identifican por la URL que empieza por "https://" en lugar de "http://".
* Los navegadores suelen mostrar un icono de candado o un indicador de "seguro" en la barra de direcciones para las conexiones HTTPS, lo que puede ayudar a los usuarios a saber que su conexión es segura.

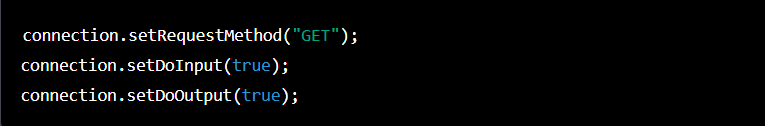
# Ejercicio 2. Realiza un pequeño tutorial de cómo procesar datos JSON con conexión a internet.

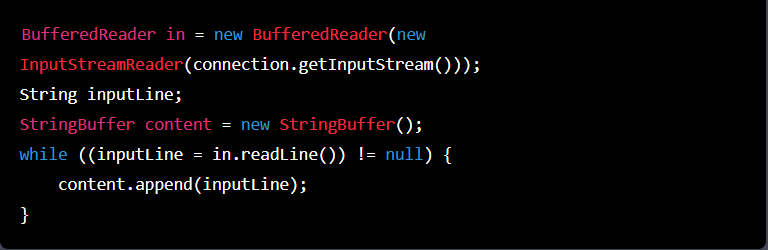
Primero, necesitamos importar las librerías:

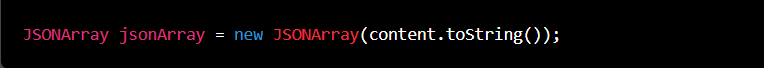
Después, abriremos una conexión con el punto final JSON del que queremos. Creando un nuevo objeto URL y pasando la URL del punto final como una cadena. Luego, abrimos una conexión llamando al método openConnection() en el objeto URL, y convertir el resultado en un objeto HttpURLConnection.



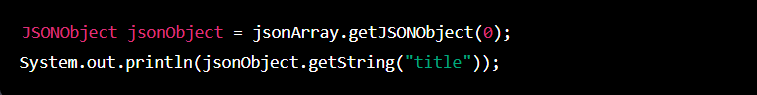
Ahora establecemos el método de solicitud como "GET" y configuramos la conexión para permitir la entrada y la salida.

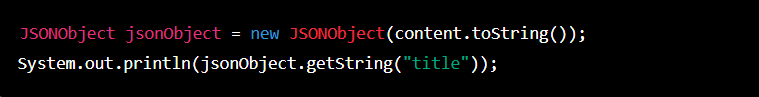
Ahora podemos leer la respuesta desde el punto final creando un nuevo objeto BufferedReader y pasándole un objeto InputStreamReader creado a partir del flujo de entrada de la conexión. A continuación, usamos el método readLine() para leer la respuesta línea por línea.

Ahora analizamos los datos JSON creando un nuevo Array JSONArray o JSONObject y pasándole la cadena de respuesta.



Ahora accedemos a los datos JSON tratándolos como un JSONArray. Por ejemplo, usamos el método get() para acceder a valores específicos o el método length() para obtener el número de elementos del array JSON.

Si usamos JSONObject



# Ejercicio 3. Realiza una infografía con el régimen de permisos necesario en el desarrollo de aplicaciones Android.



# Ejercicio 4. Realiza una diapositiva en genially con los pasos a seguir en el punto 9.

La práctica se entregará en formato pdf con las siguientes consideraciones:

Entrega en fecha, hasta un máximo de 10 punto.

Retraso de 2 días desde la fecha máxima de entrega, hasta un máximo de 7 puntos.

Retraso de 4 días desde la fecha máxima de entrega, hasta un máximo de 5 puntos.

Retraso de más de 4 días desde la fecha máxima de entrega, pierde derecho a evaluación.