



27/11/2025

Segundo parcial práctico

- 1) Implemente en Java un servidor HTTP (HTTPServer.jar) que acceda al Servicio Meteorológico Nacional para obtener información actual del clima de una localidad. La aplicación deberá responder a solicitudes GET en el puerto 8081.

Servicio a utilizar:

https://ws.smn.gob.ar/map_items/weather

Servicio a implementar

clima?provincia=NOMBRE_PROVINCIA

clima?provincia=NOMBRE_PROVINCIA&localidad=NOMBRE_LOCALIDAD

Ejemplo de uso

<http://localhost:8081/clima?provincia=Buenos Aires>

<http://localhost:8081/clima?provincia=Buenos Aires&localidad=Mar del Plata>

Salida

```
<?xml version="1.0" encoding="UTF-8"?>
<provincia>
    <nombre>Buenos Aires</nombre>
    <localidades>
        <localidad>
            <nombre>Mar del Plata</nombre>
            <temperatura>25.2</temperatura>
            <humedad>60</humedad>
            <viento>5</viento>
            <presion>1024</presion>
        </localidad>
        ...
    </localidades>
</provincia>
```



-
- 2) Implemente una página dinámica utilizando JSP (que se ejecute en un servidor Apache Tomcat) que genere una página html que contenga una tabla con la lista de localidades y su información del clima. Esta página deberá consumir el servicio provisto por la aplicación implementada en el punto 1.

Servicio a implementar

```
/app/clima.jsp?provincia=NOMBRE_PROVINCIA  
/app/clima.jsp?provincia=NOMBRE_PROINVICA&localidad=NOMBRE_LOCALIDAD
```

Ejemplo de uso

```
http://localhost:8080/app/clima.jsp?provincia=Buenos Aires  
http://localhost:8080/app/clima.jsp?provincia=Buenos Aires&localidad=Mar del Pla
```

El Clima

Provincia: Buenos Aires

Localidad	Temperatura	Humedad	Presión	Viento
Mar del Plata	25.2	60	1024	5
Miramar	22.1	65	1020	2

- 3) Implemente una aplicación web que permita al usuario seleccionar una provincia y opcionalmente una localidad de listas desplegables y al confirmar se muestre la tabla con información del clima de la localidad/localidades seleccionadas.
Implementar mediante servlets y Ajax.

Recurso a utilizar:

Para obtener la lista de provincias

<https://apis.datos.gob.ar/georef/api/provincias>

Para obtener la lista de localidades de una provincia

<https://apis.datos.gob.ar/georef/api/localidades?provincia=PROVINCIA&campos=nombre&max=1000>

Ejemplo

<https://apis.datos.gob.ar/georef/api/localidades?provincia=Buenos%20Aires&campos=nombre&max=1000>



El Clima

Provincia:

Localidad:

Mostrar

- 4) Mostrar en la página principal la cantidad de usuarios que se encuentran en ese momento conectados a la aplicación. Utilizar websocket para llevar el control de la cantidad de conexiones. Si la cantidad de usuarios conectados supera una cantidad máxima permitida se deberá bloquear a ese usuario hasta que se libere el cupo. Mostrar el mensaje “En este momento hay demasiados usuarios utilizando el servicio”.

Usuarios: 4

El Clima

Provincia:

Localidad:

Mostrar

Cuando la cantidad de usuarios conectados supera la cantidad máxima permitida:

Usuarios: 4

El Clima

En este momento hay demasiados usuarios utilizando el servicio

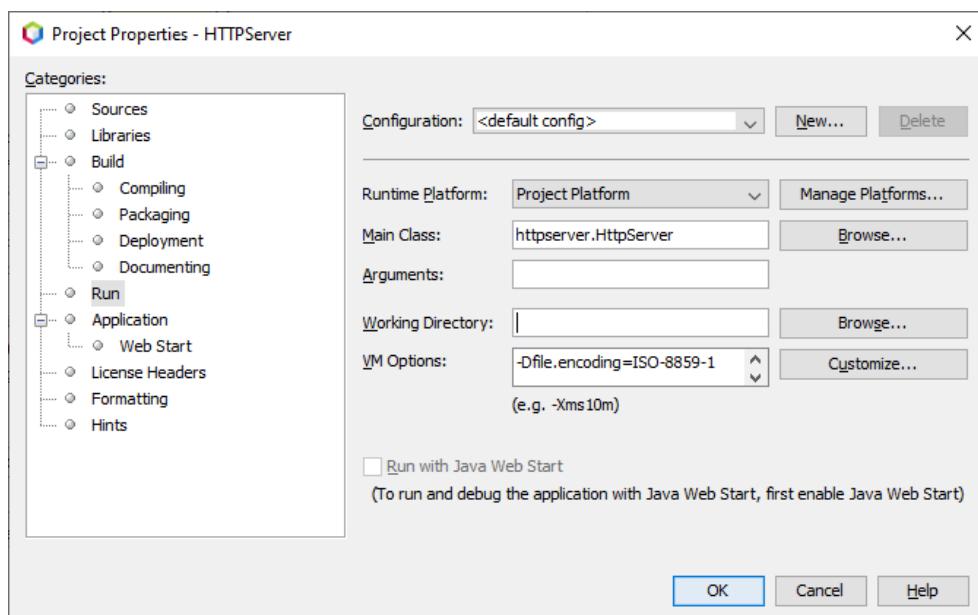


Se recomienda utilizar la librería json-simple-1.1.jar para parsear el documento JSON.

Incluirlo en las librerías externas del proyecto: Propiedades/Librerías

```
Object obj = JSONValue.parse(inputLine);
JSONArray array = (JSONArray) obj;
for (int i = 0; i < array.size(); i++) {
    JSONObject o = (JSONObject) array.get(i);
    String provincia = (String) o.get("province");
    String localidad = (String) o.get("name");
    JSONObject weather = (JSONObject) o.get("weather");
    String temperatura = weather.get("temp").toString();
    .....
}
```

Configurar VM Options con -Dfile.encoding=ISO-8859-1 para solucionar problemas con caracteres especiales.



- 1) 2 puntos
 - a) Servicio por provincia
 - b) Servicio por localidad
- 2) 3 puntos
 - a) tabla por provincia
 - b) tabla por localidad
- 3) 3 puntos
 - a) carga provincias
 - b) carga localidades
 - c) muestra tabla clima
- 4) 2 puntos
 - a) Muestra usuarios conectados
 - b) Bloquea pantalla