



**INSTITUTO TÉCNICO
DE ESTUDIOS
PROFESIONALES**

Programación Multimedia y Dispositivos Móviles Práctica 6

Servicios en red I.

Miguel Ángel Rodríguez Dalí

**Fecha límite de entrega
09/01/2023**

Ejercicio 1. ¿Qué es la codificación JSON y para qué se usa? Enumera los datos que soporta.

JSON, que significa notación de objetos de JavaScript, es un formato de intercambio de datos que es muy fácil de leer y escribir para los programadores, así como fácil de interpretar y crear para las empresas. Es un formato de texto totalmente independiente del lenguaje de programación.

Y se usa porque es un formato común para serializar y deserializar objetos en la mayoría de los idiomas.

Ejemplo de un JSON:

```
{
  "squadName": "Super hero squad",
  "homeTown": "Metro City",
  "formed": 2016,
  "secretBase": "Super tower",
  "active": true,
  "members": [
    {
      "name": "Molecule Man",
      "age": 29,
      "secretIdentity": "Dan Jukes",
      "powers": [
        "Radiation resistance",
        "Turning tiny",
        "Radiation blast"
      ]
    },
    {
      "name": "Madame Uppercut",
      "age": 39,
      "secretIdentity": "Jane Wilson",
      "powers": [
        "Million tonne punch",
        "Damage resistance",
        "Superhuman reflexes"
      ]
    },
    {
      "name": "Eternal Flame",
      "age": 1000000,
      "secretIdentity": "Unknown",
      "powers": [
        "Immortality",
        "Heat Immunity",
        "Inferno",
        "Teleportation",
        "Interdimensional travel"
      ]
    }
  ]
}
```

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

Para hacer esto recibiremos los datos desde un JSON a nuestro Java, usando la clase `JsonReader`. Crearemos una clase `Animal` de ejemplo desde la que trabajaremos y haremos las cosas:

```
public class Animal {
    private String especie;
    private String descripcion;
    private String imagen;

    public Animal(String especie, String descripcion, String imagen) {
        this.especie = especie;
        this.descripcion = descripcion;
        this.imagen = imagen;
    }

    public String getEspecie() {
        return especie;
    }

    public void setEspecie(String especie) {
        this.especie = especie;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public String getImagen() {
        return imagen;
    }

    public void setImagen(String imagen) {
        this.imagen = imagen;
    }
}
```

Luego leeremos el formato del JSON almacenándolo en un List del tipo `Animal`:

```
public List<Animal> readJsonStream(InputStream in) throws IOException {
    // Nueva instancia JsonReader
    JsonReader reader = new JsonReader(new InputStreamReader(in, "UTF-8"));
    try {
        // Leer Array
        return leerArrayAnimales(reader);
    } finally {
        reader.close();
    }
}
```

Continuaremos leyendo el array del lector de objetos JSON:

```
public List leerArrayAnimales(JsonReader reader) throws IOException {  
    // Lista temporal  
    ArrayList animales = new ArrayList();  
  
    reader.beginArray();  
    while (reader.hasNext()) {  
        // Leer objeto  
        animales.add(leerAnimal(reader));  
    }  
    reader.endArray();  
    return animales;  
}
```

Leemos los atributos de cada objeto y lo asignamos a un objeto de la clase Animal:

```
public Animal leerAnimal(JsonReader reader) throws IOException {  
    String especie = null;  
    String descripcion = null;  
    String imagen = null;  
  
    reader.beginObject();  
    while (reader.hasNext()) {  
        String name = reader洗洗Name();  
        switch (name) {  
            case "especie":  
                especie = reader.nextString();  
                break;  
            case "descripcion":  
                descripcion = reader.nextString();  
                break;  
            case "imagen":  
                imagen = reader.nextString();  
                break;  
            default:  
                reader.skipValue();  
                break;  
        }  
    }  
    reader.endObject();  
    return new Animal(especie, descripcion, imagen);  
}
```

Ejercicio 3. Realiza un pequeño tutorial de cómo se gestionan las bases de datos en un servidor remoto y un servidor local. Añade imágenes y/o capturas de pantalla y explícalas con casos prácticos.

Base de datos en un server remoto:

Para ello hay que conectarse desde el servidor web a la BD para obtener su información requerida en la aplicación. Necesitando unas configuraciones específicas:

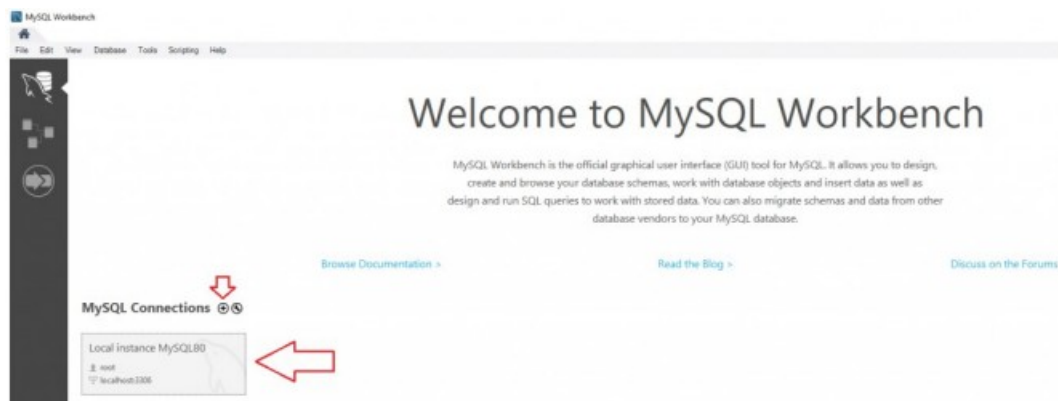
- Apertura del puerto 3306 u otro, siendo el susodicho el predeterminado de MySQL.
- Creación de un usuario en la BD con los permisos pertinentes.
- Configuración del MySQL para que acepte conexiones remotas:
 - Hay que editar el archivo de configuración “mysqld.cnf”.
 - En el archivo anterior cambiaremos la línea mostrada a continuación comentándola con “#” para dar acceso desde cualquier ip.

```
bind-address = 127.0.0.1
```

- Después tendremos que guardar los cambios y reiniciar el servidor para que se complete el cambio.

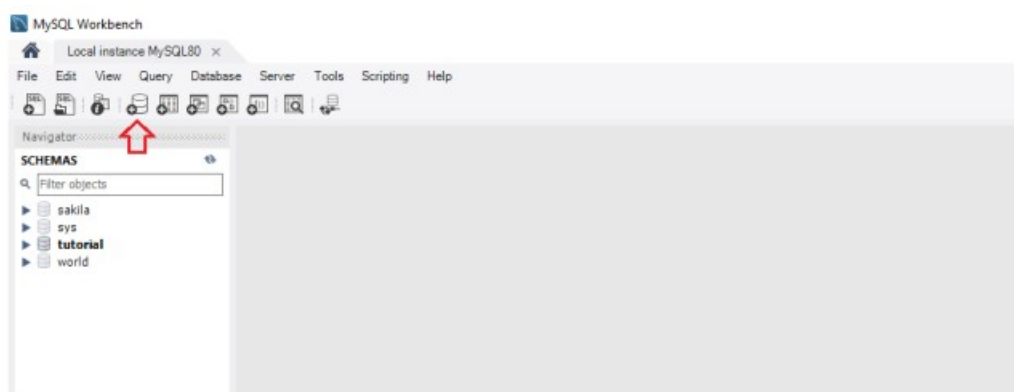
Base de datos en un server local:

Usaremos MySQL Workbench en este proceso, requiriendo lo primero crear la BD:

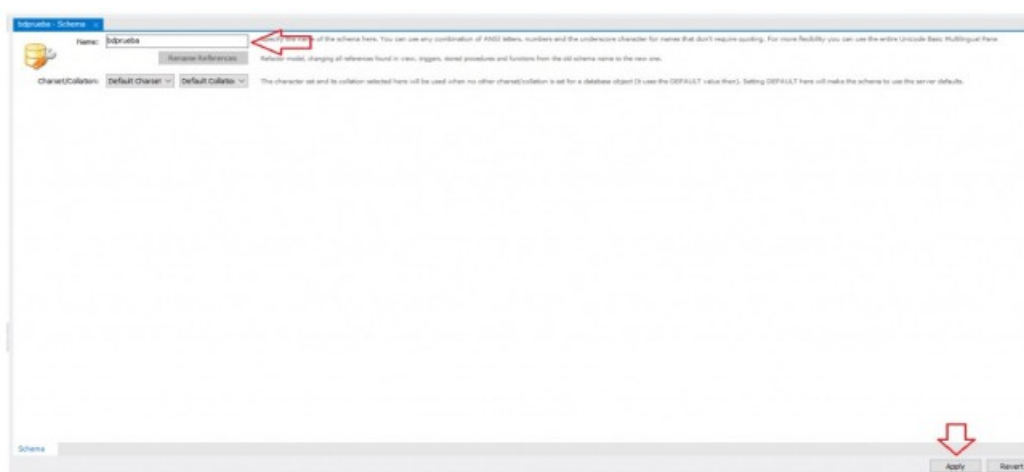


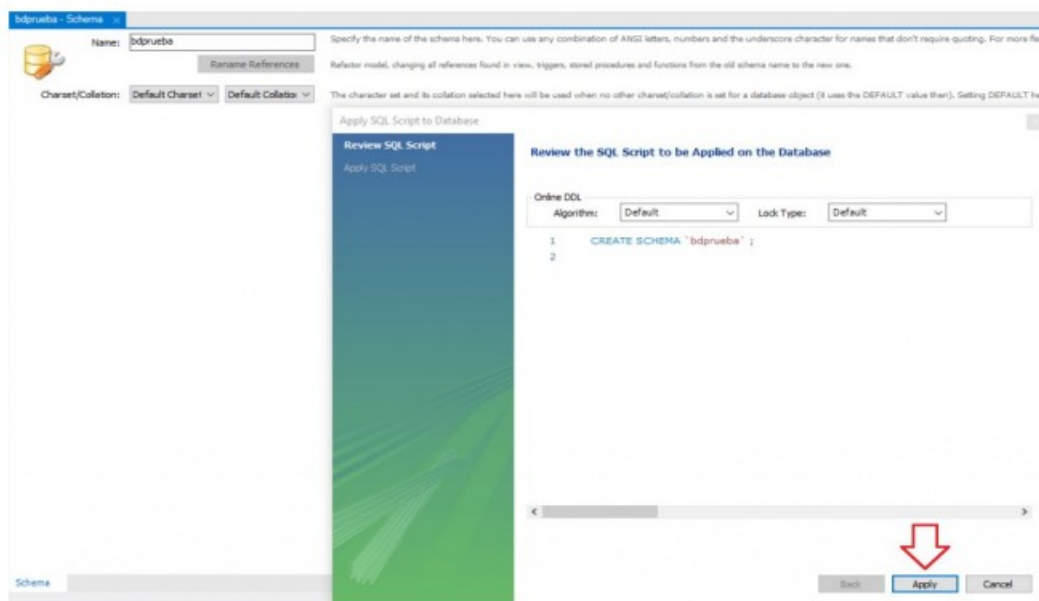
Nuestro usuario y puerto son: root y 3306.

- Para crear la BD hay que dar click en el icono de los discos:

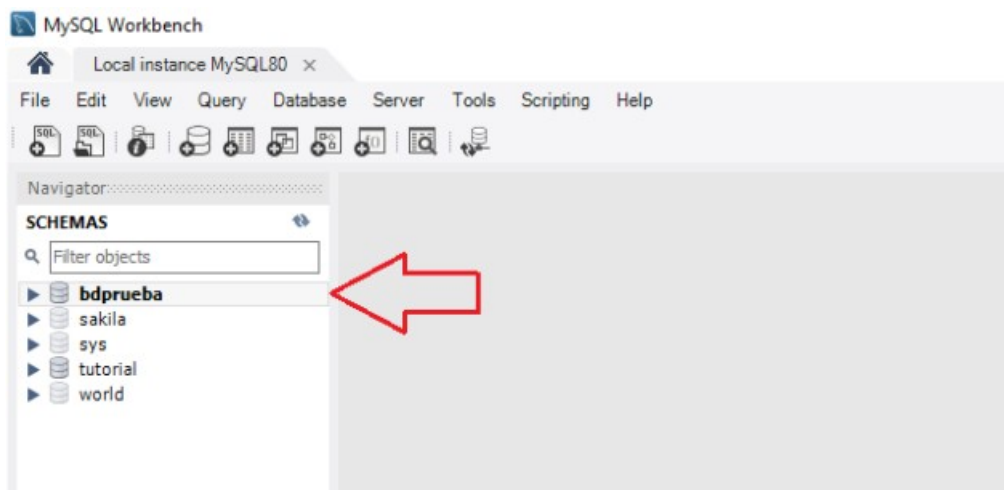


- Después escribiremos su nombre y aplicaremos:





- Finalizaremos y ya estaría creada la BD:



Ejercicio 4. Realiza una diapositiva en genially para cada una de las operaciones más importantes que podemos realizar desde PHP7 sobre nuestra base de datos MySQL. Pon ejemplos para cada una de ellas.

<https://view.genial.ly/63bc5bcc0277720018b15151/presentation-presentacion-digital-basica>

Entrega

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.