

# DataWeb Wizard



# DataWeb Wizard : Ejemplo de una GUI

Field name:	nombre	Type:	Varchar ▾
Field name:	apellidos	Type:	Varchar ▾
Field name:	edad	Type:	Integer ▾
Field name:	salario	Type:	Double ▾
<input type="button" value="Add field"/>		<input type="button" value="Remove field"/>	
		<input type="button" value="Save fields"/>	
<input type="button" value="Create database"/>			

Deberían/Podrían  
haber más **inputs**



WebApplicationFormGeneration  
Apache Tomcat & Java DB

# DataWeb Wizard



- DataWeb Wizard **creará base datos dinámicamente.**
- Deberá contar con manejo de **sesiones** utilizando HttpSession
  - Permitirá el registro de **nuevos usuarios**
  - No permitirá el acceso a ninguna página a usuarios no registrados.
  - Dependiendo el usuario será la base de datos a la que tenga acceso.

# DataWeb Wizard



- Para crear una base de datos, el usuario **sólo** deberá capturar (i) el **nombre** de la base datos, (ii) el **usuario** y (iii) su **contraseña**.
- El usuario podrá crear **N tablas dinámicamente**
- Para crear una tabla, el usuario indicará
  - El nombre, tipo (de una lista de opciones) y posiblemente la longitud del campo, entre otros.
  - Indicar la llave primaria.

# DataWeb Wizard



- El usuario podrá **elegir entre las tablas** que ha creado con anterioridad. Para cada tabla se podrá:
  - **Borrar** registros.
  - **Crear** registros.
  - **Desplazarse** al registro siguiente, al anterior, al registro inicial y al final.

# DataWeb Wizard

## Requerimientos técnicos



- **Utilizar AJAX**
- **Utilizar servicios web RESTful**  
que utilicen los métodos PUT, GET, DELETE y POST  
y si regresan información que sea en XML  
(se tendrá que implementar el empaquetamiento y  
desempaquetamiento de XML)
- **Crear servicios web SOAP**  
los cuales serán accedidos vía los servicios web RESTful
- **Restricción:** No se pueden utilizar frameworks como  
JavaServer Faces, Struts, Spring, Hibernate o similares.

# DataWeb Wizard



DatabaseMetaData

SQL: CREATE table

```
"jdbc:derby://localhost:1527/myNewDatabase;create=true;
```



w3schools.com SQL Tutorial

<http://www.w3schools.com/sql/default.asp>



# DataWeb Wizard



- ¿Qué debo entregar?
  - Proyecto de netbeans con todo el código fuente y las clases.
  - Archivo readme.txt con información clara y detallada de cómo ejecutar/desplegar su proyecto
  - Reporte de implementación



# DataWeb Wizard



- **Reporte de implementación:**

- Introducción
- Documentación UML
  - Diagramas de secuencia
  - Diagramas de clases
  - Diagrama de bases de datos
- Conclusiones

# DataWeb Wizard



- Equipo de 3 personas.
- Evaluación:
  - Peso total del proyecto equivalente al **20%** de su calificación final
    - Implementación del proyecto (16%).
    - Calidad y presentación del reporte autocontenido (4%)

# DataWeb Wizard



- Fecha y hora de entrega: *15 de Mayo de 2015 a las 4 pm*
- Se deberá entregar en persona en clase acompañado del reporte impreso (y en versión digital).
- **NOTA 1:** 20% menos por cada día natural de retraso.
- **NOTA 2:** Si se entrega después de la hora de entrega, en automático aplica un día menos.

# Código de utilidad

## Crear dinámicamente base de datos y tablas

```
Class.forName("org.apache.derby.jdbc.ClientDriver");
Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/mySixthDatabase;create=true;",
    "root", "root");
Statement query = con.createStatement();
String QueryString = "create table MySixthTable (id int not null, name " +
    "varchar(25),gender varchar(20),address varchar(50),phone varchar(20), primary key(id))";
query.executeUpdate(QueryString);
```



WebApplicationDBExamples  
Apache Tomcat & Java DB

# Código de utilidad

## Listado de tablas de una base de datos

```
Class.forName("org.apache.derby.jdbc.ClientDriver");
Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/myFirstDatabase", "root", "root");
DatabaseMetaData meta = con.getMetaData();
ResultSet res = meta.getTables(null, null, null, new String[] {"TABLE"});
System.out.println("List of tables: ");
while (res.next()) {
    out.println(res.getString("TABLE_NAME"));
}
res.close();
```



WebApplicationDBExamples  
Apache Tomcat & Java DB

# Código de utilidad

## Metadatos de una tabla

```
Class.forName("org.apache.derby.jdbc.ClientDriver");
Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/myFirstDatabase", "root", "root");
Statement query = con.createStatement();
ResultSet rs = query.executeQuery("SELECT * FROM CUSTOMERS");
out.println("<p>");

if (rs != null) {
    if (rs.next()) {
        ResultSetMetaData rsmd = rs.getMetaData();
        for (int i = 1; i <= rsmd.getColumnCount(); i++) {
            System.out.println("column name: " + rsmd.getColumnName(i));
            System.out.println("column type: "+rsmd.getColumnTypeName(i));
        }
    }
}
```



WebApplicationDBExamples  
Apache Tomcat & Java DB