

# Librerías propias de Groovy - Ejercicios

## Índice

1 MarkupBuilder (I).....	2
2 MarkupBuilder (II).....	2
3 Creación de DSLs.....	2
4 Leyendo fuentes RSS.....	2
5 Bases de datos (I).....	2
6 Bases de datos (II).....	3

## 1. MarkupBuilder (I)

Escribir el código para generar la siguiente página web utilizando MarkupBuilder. El archivo debe llamarse ej41.groovy.

```
<html>
  <head>
    <title>Sesiones Groovy</title>
  </head>
  <body>
    <h1>Sesiones Groovy</h1>
    <ul>
      <li>Introducción Groovy</li>
      <li>El lenguaje Groovy</li>
      <li>Aspectos avanzados en Groovy</li>
      <li>Librerías propias de Groovy</li>
    </ul>
  </body>
</html>
```

## 2. MarkupBuilder (II)

Escribir el código de una fuente RSS con diferentes noticias. Puedes tomar como ejemplo la fuente RSS de la web <http://groovy.org.es>. El archivo debe llamarse ej42.groovy.

## 3. Creación de DSLs

Crear un ejemplo de DSL que permite realizar operaciones del tipo `5.dias + 6.horas`. Las unidades que vamos a poder convertir serán segundos, minutos, horas, días, semanas y años. Vamos a suponer que un año tiene 53 semanas y 365 días. El archivo debe llamarse ej43.groovy.

## 4. Leyendo fuentes RSS

Mostrar por pantalla la información extraída de la fuente RSS generada en el ejercicio 2 (MarkupBuilder II). El archivo debe llamarse ej44.groovy.

## 5. Bases de datos (I)

Crear una tabla en una nueva base de datos llamada *Libros* que contenga los campos *id*, *título*, *isbn* y *año de edición*.

Realizar las cuatro operaciones básicas sobre la nueva tabla creada directamente con los métodos `execute()`, `executeUpdate()`, `eachRow()`, `query()`, `rows()` y `firstRow()`. El archivo debe llamarse ej45.groovy.

## 6. Bases de datos (II)

---

Crear una nueva tabla llamada *Usuarios*. Esta tabla tendrá la típica información de los usuarios de cualquier aplicación, es decir, *nombre de usuario*, *contraseña*, *nombre completo* y *correo electrónico*.

Posteriormente realizar inserciones de nuevos usuarios en la tabla recién creada y mostrar los usuarios por pantalla utilizando los métodos `each()` y `findAll()`. El archivo debe llamarse `ej46.groovy`.

