

# Introducción a Groovy - Ejercicios

## Índice

1 De Java a Groovy.....	2
2 GroovyBeans.....	2
3 Comprobación mediante asserts.....	3
4 Setters en Groovy.....	3
5 Métodos en Groovy.....	3
6 Más métodos en Groovy.....	3

## 1. De Java a Groovy

Modificar la siguiente clase en Java para convertirla en una clase en Groovy y que quede lo más simplificada posible con la misma funcionalidad. El archivo creado debe llamarse `ej11.groovy`.

```
import java.util.List;
import java.util.ArrayList;
import java.util.Iterator;

public class Todo {
    private String titulo;
    private String descripcion;

    public Todo() {}

    public Todo(String tit, String des) {
        this.titulo = tit;
        this.descripcion = des;
    }

    public String getTitulo(){
        return titulo;
    }

    public void setTitulo(String tit){
        this.titulo = tit;
    }

    public String getDescripcion(){
        return descripcion;
    }

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

    public static void main (String[] args){
        List todos = new ArrayList();
        todos.add(new Todo("Lavadora","Poner lavadora"));
        todos.add(new Todo("Impresora","Comprar cartuchos impresora"));
        todos.add(new Todo("Películas","Devolver películas videoclub"));

        for (Iterator iter = todos.iterator();iter.hasNext();) {
            Todo todo = (Todo)iter.next();
            System.out.println(todo.getTitulo()+"
"+todo.getDescripcion());
        }
    }
}
```

## 2. GroovyBeans

Crear la clase *Hotlink* con la información necesaria para almacenar todos los enlaces favoritos. La información a almacenar de estos enlaces será el *nombre* de la página, la *URL* de la misma y una cadena de *etiquetas* que categorizan el contenido del enlace.

Simplemente crearemos un constructor de la clase *Hotlink* que permita especificar las tres propiedades de la clase (nombre, url y etiquetas). Para comprobar que nuestra clase funciona correctamente, insertaremos los siguientes ejemplos de direcciones web:

- Nombre: DCCIA, url: <http://www.dccia.ua.es>, etiquetas: departamento universidad alicante
- Nombre: Google, url: <http://www.google.es>, etiquetas: buscador calendario docs
- Nombre: El País, url: <http://www.elpais.es>, etiquetas: periódico noticias sucesos

El archivo creado se debe llamar `ej12.groovy`.

### 3. Comprobación mediante asserts

Comprobar que se han insertado correctamente cada uno de los enlaces anteriormente indicados mediante el uso de `asserts` y las funciones tipo `getter()` generadas automáticamente en Groovy. Este archivo debe llamarse `ej13.groovy`.

### 4. Setters en Groovy

Modificar la información de las etiquetas de cada uno de enlaces utilizando para ello los métodos tipo `setter()` que Groovy ha generado automáticamente. Comprobar posteriormente que los cambios se han efectuado correctamente mediante la utilización de `asserts`. Este archivo debe llamarse `ej14.groovy`.

### 5. Métodos en Groovy

Crear el método `getUrl()` para que en lugar de devolver simplemente la propiedad `url`, devuelva el texto HTML necesario para mostrar un enlace con el nombre del enlace. Por ejemplo:

```
<a href="http://www.google.es">Google</a>
```

Comprobar que el nuevo método funciona correctamente mediante la utilización de `assert`. Este archivo debe llamarse `ej15.groovy`.

### 6. Más métodos en Groovy

Crear un método que devuelva un resumen de la información de cada hotlink con toda la información del mismo. No es necesario producir el HTML, simplemente una salida similar a la siguiente:

Nombre	URL	Etiquetas	
DCCIA	<a href="http://www.dccia.ua.es">http://www.dccia.ua.es</a>	departamento	universidad

		alicante
--	--	----------

El fichero debe llamarse ej16.groovy.

