



IMÁGENES.

1

Presentación Laboratorio de Programación en Internet.

Gonzalo París Graíño.

RESUMEN

- Cómo subir una imagen.
- Cómo tratar una imagen.
- Cómo guardar una imagen
- Librerías utilizadas
- Ejemplo práctico

CÓMO SUBIR UNA IMAGEN

- El primer problema a la hora de tratar el tema de las imágenes, es saber cómo el usuario nos la puede facilitar.
- La forma de interactuar con el usuario es mediante formularios, por lo que deberemos modificar las opciones de los formularios utilizados hasta ahora para poder incluir imágenes.

CÓMO SUBIR UNA IMAGEN

<FORM>

Method tiene que ser **POST**.

Enctype tiene que ser “**multipart/form-data**”.

<INPUT>

Type tiene que ser “**file**”

EJEMPLO

```
<form method ="post" enctype="multipart/form-data"  
  action="/contexto/ProcesarImagen">
```

```
  Nombre: <input name="nombre"><br>
```

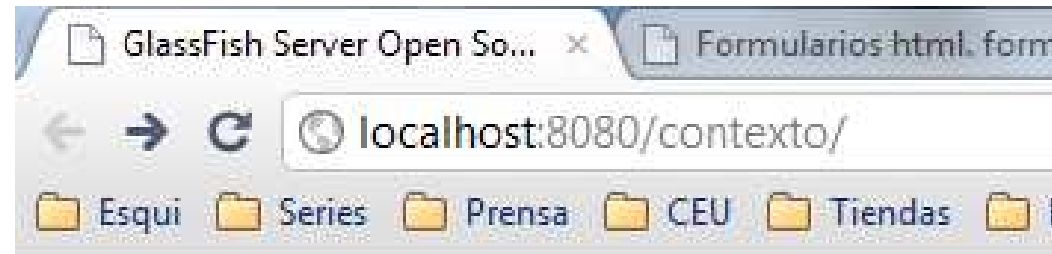
```
  <br>Selecciones la imagen:<br>
```

```
  <input type="file" name="archivo">
```

```
  <br><button>enviar</button>
```

```
</form>
```

CÓMO SUBIR UNA IMAGEN



Enviar una foto.

Nombre:

Selecciones la imagen:

No se ha...archivo

CÓMO TRATAR UNA IMAGEN.

- Para trabajar con una imagen (file), haremos uso de la librería FileUpload de Apache Commons.
- ❖ De manera resumida, vamos a generar una lista de items que contenga todos los parámetros de la petición, incluidos los files. De tal forma, podremos trabajar tanto con los files como con los campos de texto de una forma sencilla y unificada.
- ❖ Además, esta clase proporciona información adicional de cada parámetro que nos será muy útil

CÓMO TRATAR UNA IMAGEN.

PARSEAMOS LA PETICIÓN.

```
/*Comprobamos que se trata de un formulario file.*/
```

```
    boolean isMultipart =  
ServletFileUpload.isMultipartContent(request);
```

```
/* Nos irá generando los items y según su tamaño los guardará en  
   memoria, en un fichero temporal o no los aceptará si son muy  
   grandes.*/
```

```
FileItemFactory factory = new DiskFileItemFactory();
```

```
/* Creamos un manejador para todos los items de la petición.*/  
ServletFileUpload upload = new ServletFileUpload(factory);
```

```
/* Obtenemos la lista con todos los items de la petición.*/  
List /* FileItem */ items = upload.parseRequest(request);
```

CÓMO TRATAR UNA IMAGEN.

- Una vez tenemos la lista, deberemos procesar todos los parámetros. Para saber si se trata de un file o texto, la librería nos proporciona el siguiente método.

```
Iterator iter = items.iterator();
    while (iter.hasNext()) {
FileItem item = (FileItem) iter.next();
    if (item.isFormField()) {
        processFormField(item);
    } else {
        processUploadedFile(item);
    }
}
```


CÓMO TRATAR UNA IMAGEN.

- Parámetro solo texto:

String name = **item.getFieldName();**

String value = **item.getString();**

- Parámetro File:

String fieldName = **item.getFieldName();**

String fileName = **item.getName();**

String contentType = **item.getContentType();**

boolean isInMemory = **item.isInMemory();**

long sizeInBytes = **item.getSize();**

CÓMO GUARDAR UNA IMAGEN

- FileUpload, facilita la forma de guardas una imagen (file) haciendo uso de la clase File.

```
File uploadedFile = new File(...);  
item.write(uploadedFile);
```

- Recomendado guardar dentro del directorio Web Pages, para poder acceder a ellas de una forma simple y relativa. Para ello, hago uso de la función:

```
/*Nos devuelve el path de Web-Pages.*/  
String pathWeb = contexto.getRealPath("/");
```

LIBRERÍAS UTILIZADAS

org.apache.commons.fileupload

<http://commons.apache.org/fileupload/>

EJEMPLO PRÁCTICO

- A continuación se mostrará un proyecto realizado en NetBeans que permite subir una imagen y mostrar por pantalla junto información útil que nos proporciona FileUpload.