

Configuración de Spring Boot para subir archivos

➤ Agregar las siguientes propiedades en el archivo `application.properties`

```
# CONFIGURACION MULTIPART (SUBIDA DE ARCHIVOS)
# ¿Habilitamos subida de archivos?
spring.servlet.multipart.enabled=true
# Directorio intermedio para subir archivos (Linux/MAC)
# spring.servlet.multipart.location=/tmp
# Directorio temporal para subir archivos (Windows)
spring.servlet.multipart.location=c:/tmp
# Maximo tamaño de archivos que se pueden subir
spring.servlet.multipart.max-file-size=2MB
```

➤ Crear un directorio para guardar los archivos que se subirán. Por ejemplo:

✓ Windows

- C:\empleos\img-vacantes

✓ Linux/MAC: El directorio **DEBE TENER PERMISOS DE ESCRITURA/LECTURA** para el usuario que ejecuta la aplicación de Spring Boot.

```
$ sudo mkdir -p /empleos/img-vacantes
$ sudo chown -R ivan:ivan /empleos/img-vacantes
$ ls -l /empleos
drwxr-xr-x 2 ivan ivan 4096 feb 13 09:23 img-vacantes
```

Clase Helper para guardar el archivo a disco duro

Controlador

```
@PostMapping(value = "/save")
public String guardar(... @RequestParam("archivoImagen") MultipartFile multiPart ...) {
    ...
    if (!multiPart.isEmpty()) {
        //String ruta = "/empleos/img-vacantes/"; // Linux/MAC
        String ruta = "c:/empleos/img-vacantes/"; // Windows
        String nombreImagen = Utileria.guardarArchivo(multiPart, ruta);
        if (nombreImagen != null){ // La imagen si se subio
            // Procesamos la variable nombreImagen
            vacante.setImagen(nombreImagen);
        }
    }
    ...
}
```

Formulario HTML

```
<form action="#" enctype="multipart/form-data">

    <input type="file" name="archivoImagen" />
    <input type="submit">

</form>
```

Nota: En este caso el valor del atributo name del input de tipo file (archivoImagen) no es el nombre de la propiedad de la clase de modelo, porque no se puede convertir un tipo File a String.

```
public class Utileria {
    public static String guardarArchivo(MultipartFile multiPart, String ruta) {
        // Obtenemos el nombre original del archivo.
        String nombreOriginal = multiPart.getOriginalFilename();
        try {
            // Formamos el nombre del archivo para guardarlo en el disco duro.
            File imageFile = new File(ruta + nombreOriginal);
            System.out.println("Archivo: " + imageFile.getAbsolutePath());
            //Guardamos fisicamente el archivo en HD.
            multiPart.transferTo(imageFile);
            return nombreOriginal;
        } catch (IOException e) {
            System.out.println("Error " + e.getMessage());
            return null;
        }
    }
}
```

Método (helper) para guardar el archivo en el disco duro.