



2.

SERVICIO DE ALMACENAMIENTO

NUESTRO EJEMPLO

- ▶ Tomamos un ejemplo de Spring
<https://spring.io/guides/gs/uploading-files/>
- ▶ La idea es crear un servicio de almacenamiento estándar (interface) y una implementación en local.

INTERFACE **StorageService**

```
public interface StorageService {  
    void init();  
    String store(MultipartFile file, long id);  
    Stream<Path> loadAll();  
    Path load(String filename);  
    Resource loadAsResource(String filename);  
    void deleteAll();  
}
```

Métodos que debería proporcionarnos un servicio de almacenamiento de ficheros.

Algunos están modificados sobre el ejemplo.

CLASE `FileSystemStorageService`

- ▶ Almacenamiento en nuestro sistema de ficheros.
- ▶ Método *store*
 - ▶ Recibe el multiparte y el ID del nuevo empleado.
 - ▶ Suponemos que estamos subiendo el avatar.
 - ▶ Lo almacenamos como ID.ext (por ejemplo, 5.png, o 4.jpg).

CLASE `FileSystemStorageService`

- ▶ Método *load*
 - ▶ Devuelve la ruta de un fichero desde su nombre.
- ▶ Método *loadAsResource*
 - ▶ Recibe el nombre de un fichero.
 - ▶ Busca el fichero, y lo devuelve como una instancia de *Resource* (envoltorio conveniente para un fichero)

CLASE `FileSystemStorageService`

- ▶ Método *init*
 - ▶ Inicializa el sistema de ficheros
- ▶ Método *deleteAll*
 - ▶ Elimina todos los ficheros

La clase `FileSystemStorageService` utiliza algunas clases de **error** y de **configuración**, y necesita ser **invocado al lanzar el proyecto**.

CLASES DE ERROR

- ▶ *StorageException*
Error general de almacenamiento.
- ▶ *StorageFileNotFoundException*
Fichero no encontrado.

CLASES DE CONFIGURACIÓN

- ▶ *StorageProperties*
 - ▶ Anotada con *@ConfigurationProperties*
 - ▶ Envoltorio conveniente para uso de properties.
 - ▶ Inyectable mediante *@Autowired*
 - ▶ Tenemos que añadir *@EnableConfigurationProperties*
(StorageProperties.class)
en una clase con *@Configuration*

INVOCAR AL LANZAR EL PROYECTO

- ▶ Creación de un *CommandLineRunner* en la clase *Application*.
- ▶ Permite encapsular un código que se ejecutará en el lanzamiento de la aplicación.
- ▶ Usamos una expresión lambda (muy habitual en este contexto).