

## MANEJO DE FICHEROS

---

### PROCESO DE DESARROLLO

#### Objetivos

- Utilizar clases para la gesti n de ficheros y directorios
- Conocer los principales m todos de la clase File de Java
- Manejar el API de Java

#### Recursos

##### V deos:

- [La clase String](#)
- [La clase StringBuilder](#)
- [La clase File](#)
- [Configuraci n de Eclipse para proyecto de ficheros](#)
- [Funcionalidades de la clase File](#)
- [Preparaci n de la actividad](#)
- [V deo de ayuda: Tratamiento de excepciones](#)
- [V deo de ayuda: Lanzamiento de excepciones](#)

##### Librer as y c digo:

- [GestionFicherosApp](#)

##### Referencias:

- Ian F. Darwin. 2014. *Java Cookbook, 3rd Edition*.
- <http://docs.oracle.com/javase/7/docs/api/java/io/File.html>
- <https://docs.oracle.com/javase/7/docs/api/java/lang/StringBuilder.html>
- <https://docs.oracle.com/javase/7/docs/api/java/lang/Exception.html>

## MANEJO DE FICHEROS

---

### Pasos a seguir

En este primer proyecto vamos a trabajar con las funcionalidades de la clase **File** de Java. Aunque el proyecto entero excede los objetivos que se quieren trabajar, se pretende que se configure el entorno con las indicaciones para poder visualizar las funcionalidades que tenemos que implementar.

Mira los v deos, descarga las librer as y prepara el entorno tal y como se indica. Puedes utilizar como base, el c digo que se ofrece. Una vez hecho esto, lo que se espera es realices la implementaci n correcta del m todo **getInformacion** para que tenga la siguiente funcionalidad:

- Deber  mostrar el nombre, tipo (fichero o directorio), ubicaci n (path completo), fecha de  ltima modificaci n en formato fecha y si es oculto o no
- Si es un fichero deber  mostrar su tama o en bytes
- Si es un directorio deber  mostrar el n mero de elementos que contiene, espacio libre, espacio disponible y espacio total
- Deber  haber una gesti n de errores (como m nimo, controlar que exista y que se tienen permisos).

Para ello, deber s localizar los m todos de la clase **File** que nos permiten realizar la funcionalidad deseada a trav s del API de Java. El resultado de las llamadas a los distintos m todos, se puede concatenar en forma de un objeto **StringBuilder**, que se utilizar  para mostrar la informaci n desde la interfaz. En el v deo de la preparaci n de la actividad se puede ver con detalle lo que se espera que se entregue. Deber s comentar este m todo debidamente. La entrega incluir  esta actividad y la siguiente.

**Dedicaci n estimada: 3 horas**