

# FileSystem

---

FileSystem es un programa que gestiona un sistema de archivos. En el archivo zip adjunto a la tarea tienes un programa en Java que está sin terminar de implementar.

Diseñarás el programa en varias fases:

## Fase 1. (30 puntos)

Tienes que hacer que la clase FileSystemMain1.java funcione correctamente.

Para ello deberás implementar la clase File.

En el archivo File.java tienes instrucciones para su implementación.

Ejecuta la clase FileSystemMain1 con el depurador de Visual Studio para comprobar que tu clase File funciona correctamente.

Tienes que implementar los métodos de la interfaz FileEntry.

Ojo el método isDirectory debe siempre devolver false, puesto que esta clase se usa para representar a un archivo

[Ver vídeo de esta fase](#)

## Fase 2. (10 puntos)

Tienes que crear todas las excepciones que se elevan en las diferentes funciones de la interfaz FolderEntry.

Para ello crea los archivos:

FolderIsFullException.java

FileAlreadyExistsException.java

FileDoesNotExistsException.java

FolderIsNotEmptyException.java

Dentro de cada archivo tienes que crear la clase correspondiente heredando de la clase Exception tal y como hemos visto en clase.

No hace falta que añadas nada dentro de la clase.

## Fase 3. (20 puntos)

Tienes que hacer que la clase `FileSystemMain2.java` funcione correctamente.

Para ello deberás implementar la clase `Folder`.

La clase `Folder` hereda de la clase `File`. Una carpeta es un tipo de archivo pero sin contenido que puede albergar otros archivos y carpetas.

En el archivo `Folder.java` tienes instrucciones para su implementación.

Ejecuta la clase `FileSystemMain2` con el depurador de Visual Studio para comprobar que tu clase `Folder` funciona correctamente.

En esta fase, solo es necesario implementar el método `toString` de la interfaz `FolderEntry` y el método `isDirectory` de la interfaz `FileEntry`, pero tendrás que incluir la implementación del resto de métodos.

El método `isDirectory` indicará en la clase `Folder` un `true` ya que esta clase es la que representa a una carpeta.

Cómo lógicamente no los vas a implementar en esta fase, lo que harás es sobrescribir el método de la interfaz y si el método debe devolver por ejemplo un boolean pues devuelves `false`, si por ejemplo debes devolver un entero, pues devuelves un `0`, si tienes que devolver un Objeto `File` o `Folder`, pues devuelves un `null`.

De esta manera se cumpliría que todos los métodos de la interfaz están implementados en la clase `Folder`.

[Ver vídeo de esta fase](#)

## Fase 4. (40 puntos)

Tienes que hacer que la clase `FileSystemMain3.java` funcione correctamente.

Para ello deberás terminar de implementar la clase `Folder`.

Deberás implementar correctamente todos los métodos que faltaban por implementar en la fase 3.

En el archivo `Folder.java` tienes instrucciones para su implementación.

Ejecuta la clase `FileSystemMain3` con el depurador de Visual Studio para comprobar que tu clase `Folder` funciona correctamente.

[Ver vídeo de esta fase](#)