

# Especificación de la clase File

## Arquitectura del sistema

La clase `File` representa un archivo almacenado en el sistema de archivos del ordenador. Cada `file` individual representa un recurso utilizado por un sistema de software (por ejemplo, para mostrar una imagen en una página web). Un `file` puede ser de dos tipos: un `archivo PROPERTY` (que contiene pares clave=valor, como `DATE=20250919`) o una `IMAGE` (en formato PNG de 8 bits).

**Figura 1.** Diagrama de clases

La arquitectura del sistema de software se muestra en Figura 1.

El `content` del `file` se implementa como un `ArrayList<Character>`. Esta estructura de datos permite tanto la representación de:

- caracteres Unicode (UTF-16) para los pares clave=valor, y
- Los caracteres UTF-8 necesarios para el formato PNG de 8 bits, es decir, solo se usa el byte menos significativo<sup>1</sup> (usando valores en el intervalo [0, 255]).

---

<sup>1</sup> El byte menos significativo puede obtenerse enmascarando un carácter con `0xFF`:

## Implementación requerida de la clase File

**La clase File toma los siguientes valores durante la instanciaión:**

- content deberá estar vacío, pero no null.

**Se deben implementar los siguientes métodos de la clase File:**

- void addProperty(char[] newcontent)

**Este método añade un par clave=valor al content de un file**

- newcontent se añade al content existente.
    - Si newcontent es null, se lanzará una InvalidContentException.
    - Si el type del file es IMAGE, se lanzará una excepción WrongFileTypeException.
  - long getCRC32()
- Este método calcula el CRC32 (código de redundancia cíclica) del content del file**
- Se devuelve el valor CRC32. Este CRC32 se proporciona como un long mediante el método calculateCRC32() de la clase FileUtils. El método calculateCRC32() no puede manejar directamente un ArrayList<Character>.content debettransformarseenun byte[] antes de usar el método calculateCRC32().

---

```
char c = 'A'; // example: 'A' = 65 (0x0041)
// get least significant byte as int
int lsb = c & 0x00FF;
// get most significant byte as int
int msb = (c >>> 8) & 0xFF;
// if you need it specifically as a byte type
byte lsbByte = (byte) (c & 0x00FF);
```

- Dado que calculateCRC32() no admite arrays vacíos, si content está vacío, getCRC32() devolverá el valor 0 (cero).