

**1. Autores de la clase String:**

Los autores son **Arthur van Hoff**, **Lee Boynton**, **Martin Buchholz** y **Ulf Zibis**, como se indica en la sección "Author" del Javadoc.

**2. Métodos deprecated y sus problemas:**

- **Constructores deprecated:**

- `String(byte[] ascii, int hibyte)`: Usa un byte alto fijo para convertir bytes a caracteres, ignorando la codificación estándar.
- `String(byte[] ascii, int hibyte, int offset, int count)`: Similar al anterior, pero con offset y longitud.

- **Problemas:** Estos métodos asumen una codificación específica (ISO-Latin-1 si hibyte es cero), lo que puede generar resultados incorrectos en entornos con codificaciones diferentes (como UTF-8). Su uso causa errores de portabilidad y conversión incorrecta de caracteres.

**3. Funcionalidad de `lastIndexOf(String str)`:**

Devuelve el índice de la **última ocurrencia** de la subcadena str dentro del String. Si no se encuentra, devuelve -1.

- Ejemplo: `"Java es divertido, Java es útil".lastIndexOf("Java")` → devuelve 16 (posición del segundo "Java").

**4. Parámetros de `replace(char oldChar, char newChar)`:**

- **Parámetros de entrada:**

- `oldChar`: Carácter a reemplazar.
- `newChar`: Carácter que sustituye a `oldChar`.

- **Ejemplos del Javadoc:**

1. `"mesquite in your cellar".replace('e', 'o')` → "mosquito in your cellar".
2. `"the war of baronets".replace('r', 'y')` → "the way of bayonets".

**5. Parámetro de salida de `split(String regex, int limit)`:**

El método devuelve un **arreglo de Strings (`String[]`)**. Este array contiene las subcadenas resultantes de dividir el String original según la expresión regular regex, aplicada hasta un límite de veces.