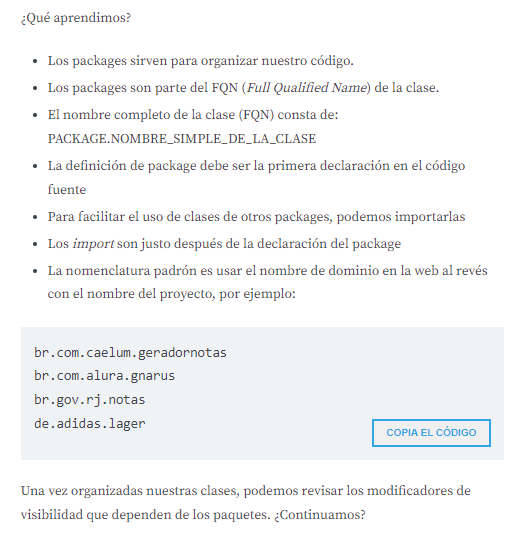
# Organizando clases con paquetes

## Conclusión 1: Organizando clases con paquetes



|  |
| --- |
|  |

## Conclusión 2: Todos los modificadores de acceso

¿Qué aprendimos?

En esta clase volvimos a hablar sobre visibilidad y aprendimos:

* Hay 3 palabras clave relacionadas con la visibilidad: private, protected, public
* Hay 4 niveles de visibilidad (de menor a mayor):
  + private (visible solo en clase)
  + <<package private>> (visible en la clase y en cualquier otro miembro del mismo paquete, que puede ser llamado de default)
  + protected (visible en la clase y en cualquier otro miembro del mismo paquete y para cualquier hijo)
  + public (visible en cualquier paquete)
* Los modificadores pueden ser usados en la definición de la clase, atributo, constructor y método.

|  |
| --- |
|  |

## Conclusión 3: Distribución de código

En esta sección más ligera vimos y aprendimos:

* Qué comentarios y tags (anotaciones) usar para definir el javadoc
* Cómo generar javadoc en Eclipse
* Que javadoc es una documentación para desarrolladores
* Que las clases estándar de Java también usan javadoc
* Cómo crear nuestra propia librería a través de JAR (J\*ava \*ARchive)
* Cómo importar librerías al nuevo proyecto
* Cómo crear un JAR ejecutable

En la siguiente clase conoceremos el paquete java.lang.

|  |
| --- |
|  |

## Conclusión 4: El paquete java.lang

En esta clase aprendimos y sabemos:

El package java.lang es el único paquete que no necesita ser importado

* Tiene clases fundamentales que cualquier aplicación necesita, como la clase String y System
* Los objetos de la clase String son inmutables y usamos una sintaxis literal para crear (objeto literal)
* Cualquier método para cambiar la clase String devuelve un nuevo String que representa el cambio
* La clase String es una CharSequence
* Si necesitamos concatenar muchos String debemos usar la clase StringBuilder
* Vimos varios métodos de la clase String como trim, charAt, contains, isEmpty, length, indexOf, replace

En la siguiente clase veremos otra clase fundamental: java.lang.Object

|  |
| --- |
|  |

## Conclusión 5: La clase Object



|  |
| --- |
|  |

## Anexos:

* Cuando se repiten nombres, se convino que se establece la extensión del dominio, luego web, luego la calisificacion package
* <https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>
* <https://docs.oracle.com/javase/10/docs/api/java/lang/String.html>.

Ar.com.bytebank

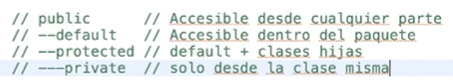
## Código:

* Import
* Out: según lo que me dice aquí es PrintStream, es un objeto del tipo PrintStream, entonces ¿yo qué puedo inferir por esto? Out es una referencia a algún objeto dentro de la clase System, y esta referencia es el tipo PrintStream.

podemos ir dándonos cuenta que out es un objeto que existe dentro de system y que además debería ser estático. ¿Y por qué digo debería ser estático? Si ustedes recuerdan sobre los atributos estáticos que hemos visto en cursos anteriores, yo no necesito crear una instancia de un objeto para acceder a estos atributos estáticos.

## Vocabulario:

* Paquetes:
* Modificador de acceso default: cuando no se determina un modi de acceso, se delimita uno publico por default pero solo en el package que se encuentra



* Documentación o java doc: Se puede poner a nivel de la clase y a nivel del contructor. Siempre se pone sobre el elemento que se quiere comentar
  + Método
  + Constructor
  + Clase
* Método sobrecargado
* Métodos estaticos