**Seccion 23**

**Paquete Java**

En Java lo que organizamos son clases, y básicamente nos permite agrupar clases ya sea por su función, por herencia, o por cualquier característica que deseemos, lo importante es que tengamos una organización de nuestras clases, ya que solo en la versión estándar de Java SE encontramos más de 4000 clases, y por lo tanto es importante definir la forma en que organizaremos nuestras clases.

Al tener tantas clases, es normal que existan clases con el mismo nombre, sin embargo como cada clase pertenece a un paquete distinto, los paquetes también nos servirán para evitar problemas de nombres entre clases.

**CONVECIÓN EN NOMBRE DE PAQUETES EN JAVA**

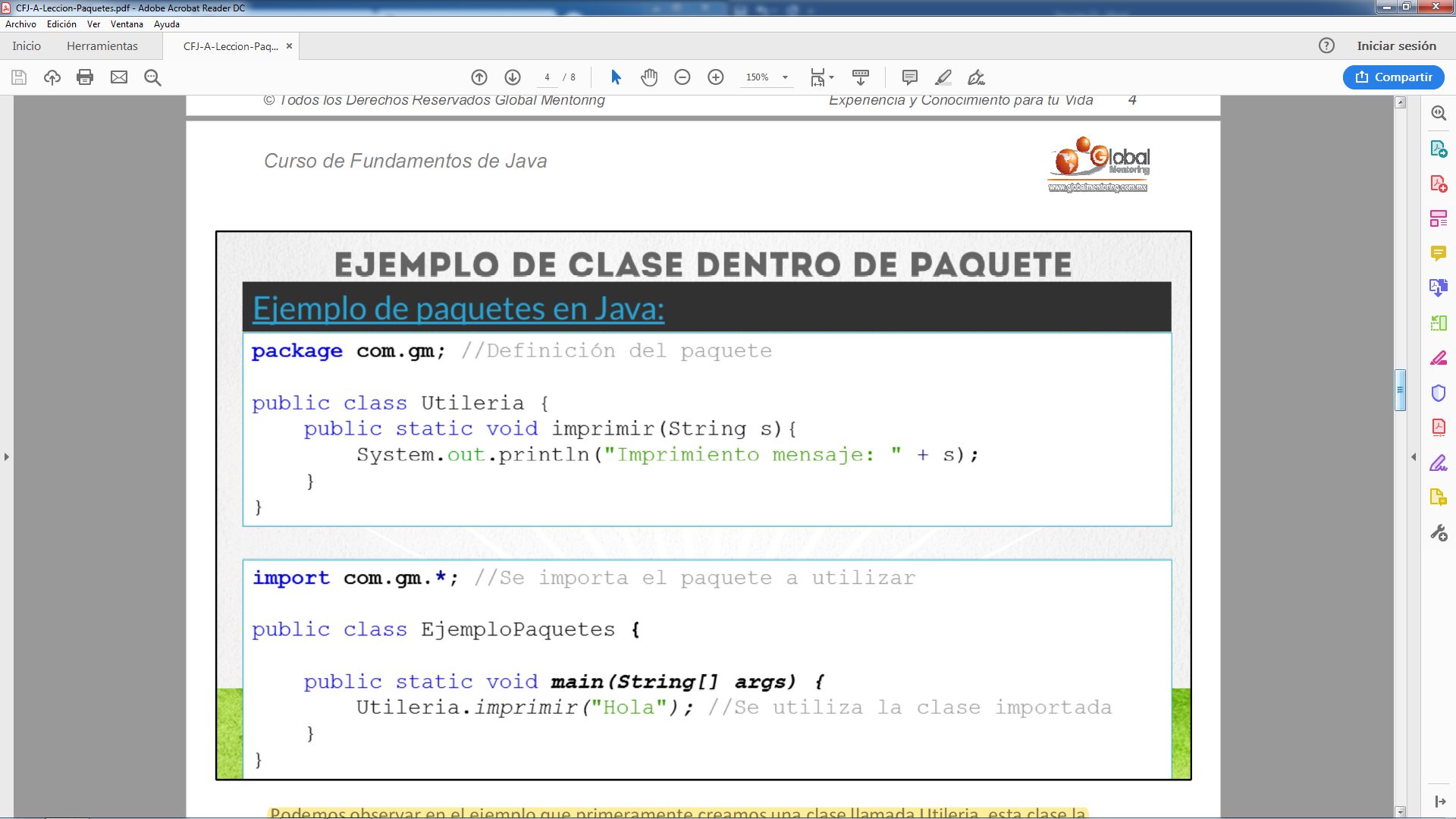
* **Todo el nombre debe estar en minúsculas.**
* **Se acostumbra escribir el nombre del dominio web de manera invertida.**

**Ejem. Si el nombre del dominio es: globalmentoring, entoces el nombre del paquete será: mx.com. globalmentoring**

* **Ejemplo de un proyecto mx.com. globalmentoring.mipryecto**
* **Ejemplo de subpaquetes**

**mx.com. globalmentoring.contabilidad.miproyecto**

**mx.com. globalmentoring.administracion.miproyecto**



**Podemos observar en el ejemplo que primeramente creamos una clase llamada Utilería, esta clase la estamos agregando a un paquete llamado com.gm.**

**En este caso creamos la clase EjemploPaquetes en el paquete por default de Java, es decir, en ningún paquete.**

**Ahora, para poder utilizar la clase Utileria definida en el paquete com.gm, lo que debemos hacer es utilizar la palabra import, la cual la podemos utilizar de dos maneras, una es importando todas las clases usando el**

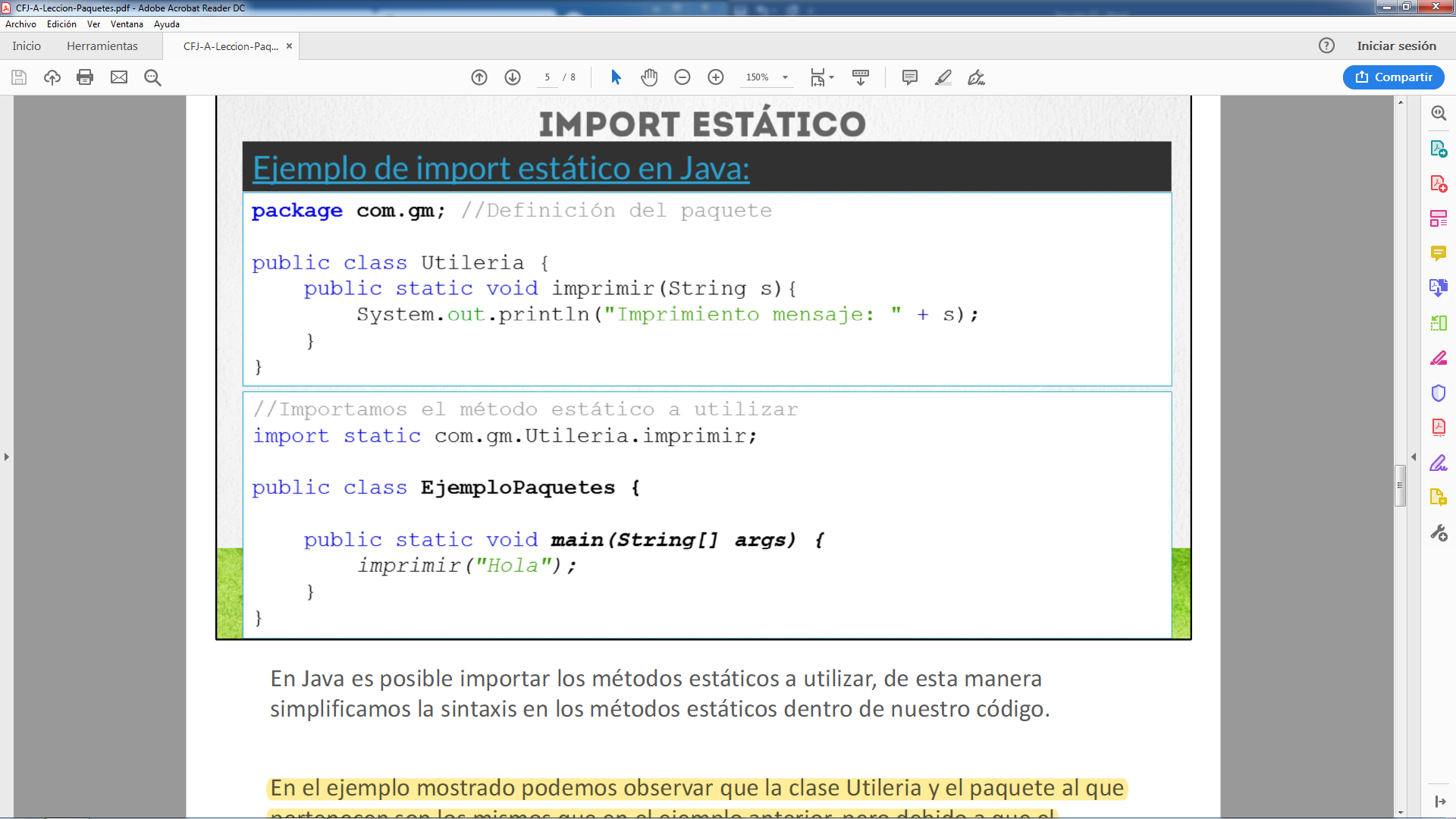
**\*, o la otra es especificando el nombre de la clase, es decir: import com.gm.Utileria. Sin embargo, esta última opción nos obliga a hacer un import por cada clase que deseemos utilizar del paquete com.gm, y si**

**fueran muchas clases serian muchas líneas de código, por lo que en muchas ocasiones la notación import com.gm.\* es la más utilizada. Cabe mencionar que el import no afecta la memoria, ya que no es que se carguen todas las clases de un paquete en memoria, una clase de carga en memoria hasta que se usa el nombre de la misma dentro del programa y no antes.**

**Finalmente podemos ver cómo estamos utilizando el método estático llamado imprimir, el cual pertenece a la clase Utilería. Observamos que ya no estamos indicando a qué paquete corresponde dicha clase, ya**

**que hemos hecho el import. Sin embargo, si hubiera más de una clase llamada Utilería, podríamos eliminar el import e indicar en la línea de código donde vamos a utilizar la clase, directamente el paquete al cual**

**corresponde la clase, quedando com.gm. Utileria.imprimir(“Hola”); con esto sabríamos exactamente a qué paquete corresponde la clase que estamos utilizando.**



**En el ejemplo mostrado podemos observar que la clase Utileria y el paquete al que pertenecen son los mismos que en el ejemplo anterior, pero debido a que el método imprimir es un método estático, podemos aprovechar la sintaxis de import static que nos brinda java para importar el método estático a utilizar, y de esta manera se simplifica la sintaxis del uso del método imprimir, ya que como podemos**

**observar el método imprimir ya no debe indicar a qué clase pertenece, sino que es suficiente con indicar el nombre del método, ya que en el import static ya se ha indicado que pertenece a la clase Utileria.**