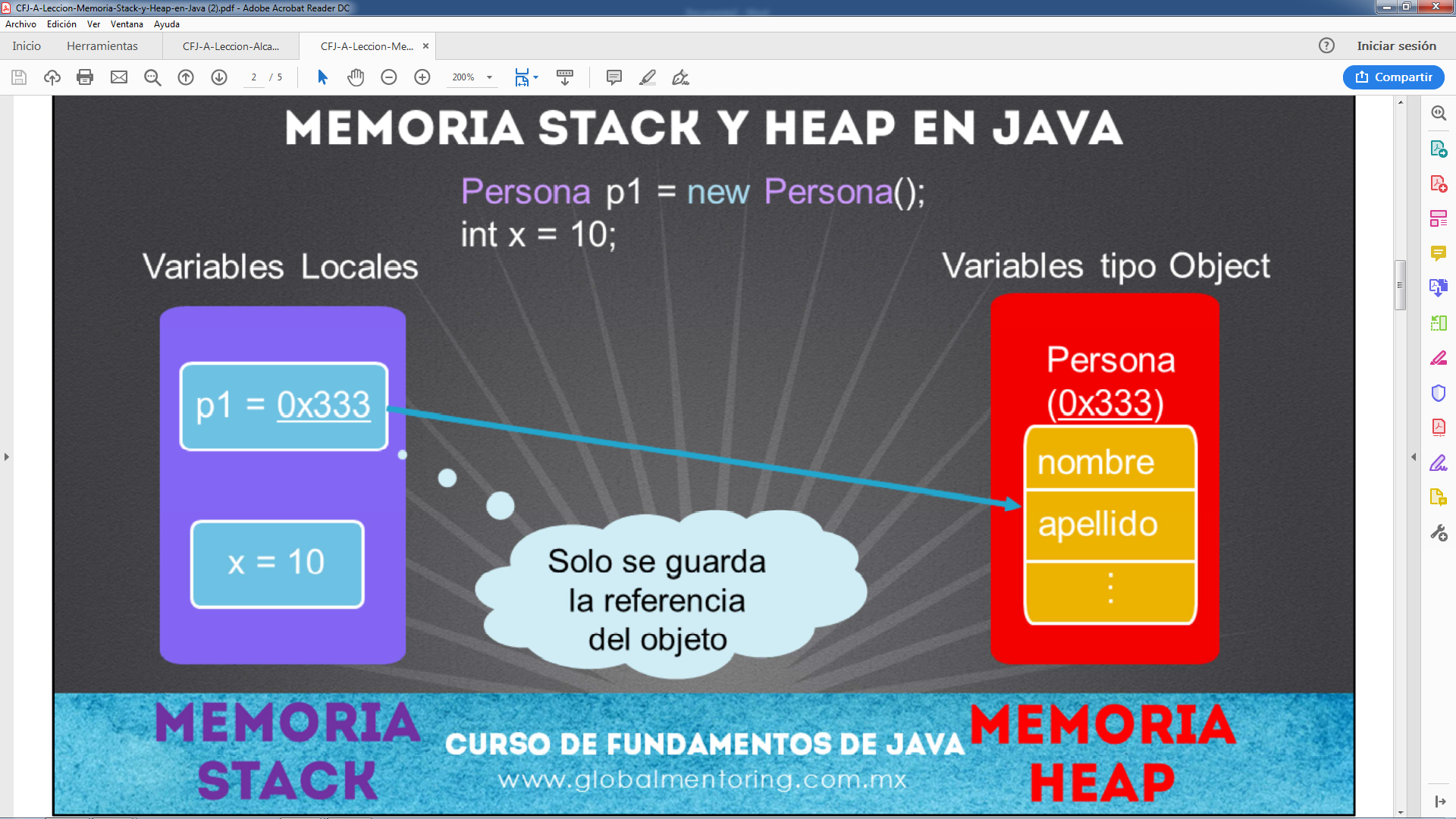
**SECCIÓN 13**

**MEMORIA STACK Y HEAP EN JAVA**

Como en muchos lenguajes de programación, la memoria RAM (Random Access Memory) se utiliza para almacenar la información de nuestro programa mientras éste se ejecuta. Java ejecuta un proceso, y a su vez internamente existen dos clasificaciones para almacenar los valores de nuestros programas, conocidos como memoria Stack y Memoria Heap.



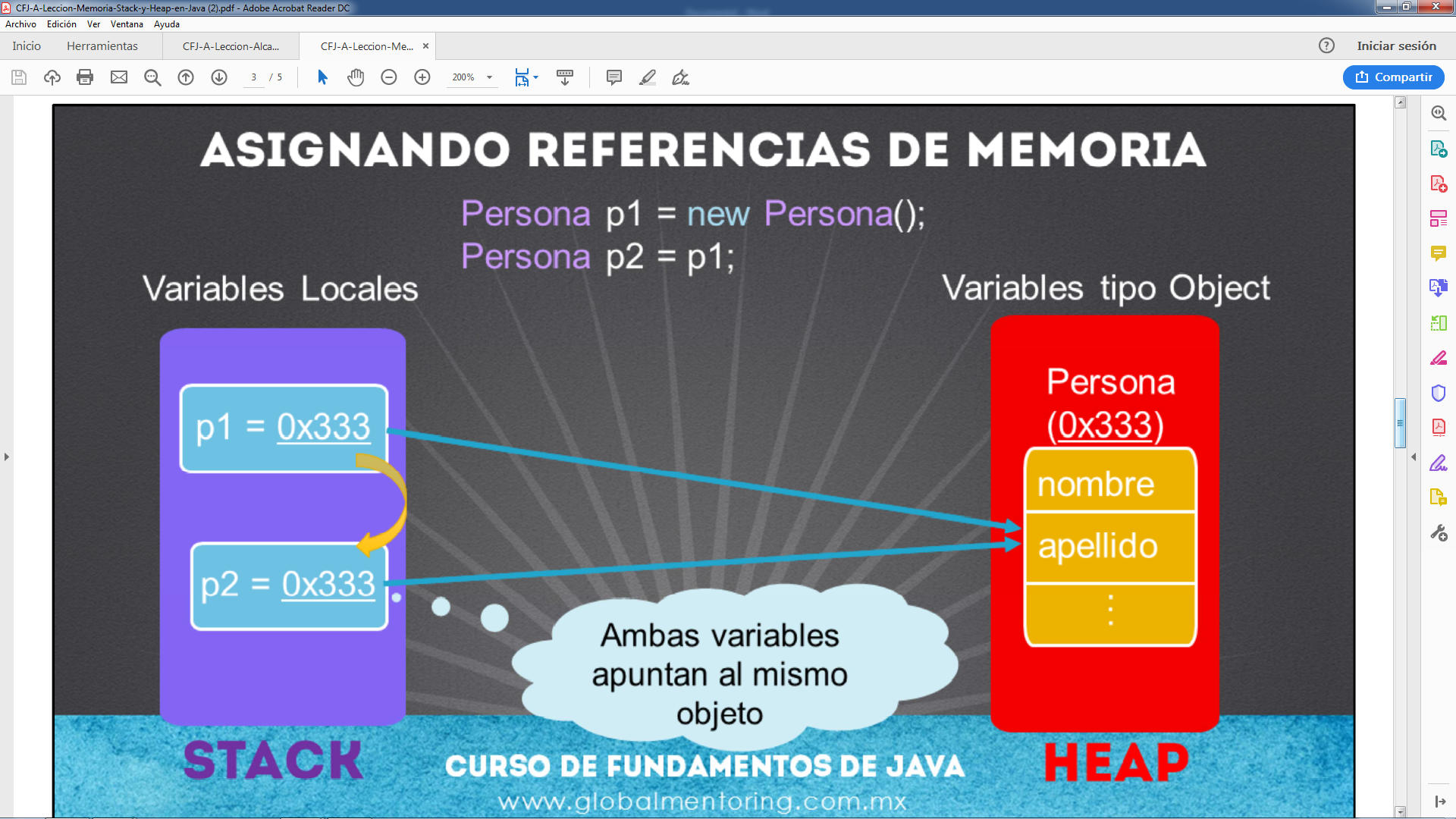
La memoria Stack se utiliza para almacenar las variables locales y las llamadas de funciones en Java. Las variables almacenadas en este espacio de memoria normalmente tienen un periodo de vida corto, únicamente mientras termina la función o método en el que se están ejecutando.

La memoria Heap se utiliza para almacenar los objetos Java, incluyendo sus atributos. Los objetos almacenados en este espacio de memoria normalmente tienen un tiempo de duración más prolongado.

En la figura podemos observar que la referencia del objeto se representa por un valor hexadecimal (0x333), el cual contiene la dirección de memoria donde está almacenado el objeto, y por lo tanto la variable local p1 almacena únicamente esta

referencia de memoria.

De esta manera el recolector de basura (garbage collector) puede buscar aquellos objetos en la memoria heap que ya no estén siendo referenciados por ninguna otra variable y finalmente liberar el espacio en memoria que ocupaba dicho objeto.



Es importante observar que no se está creando un segundo objeto, ya que solamente hay una llamada de la palabra new, y por lo tanto solo hay un objeto creado. Lo que almacena la variable p2 es sólo el valor de la referencia en memoria del objeto Persona creado. Cada que utilizamos la palabra new recibiremos una nueva referencia de memoria, es importante notar esto, ya que hasta ese momento es que se creará un nuevo objeto y no antes.