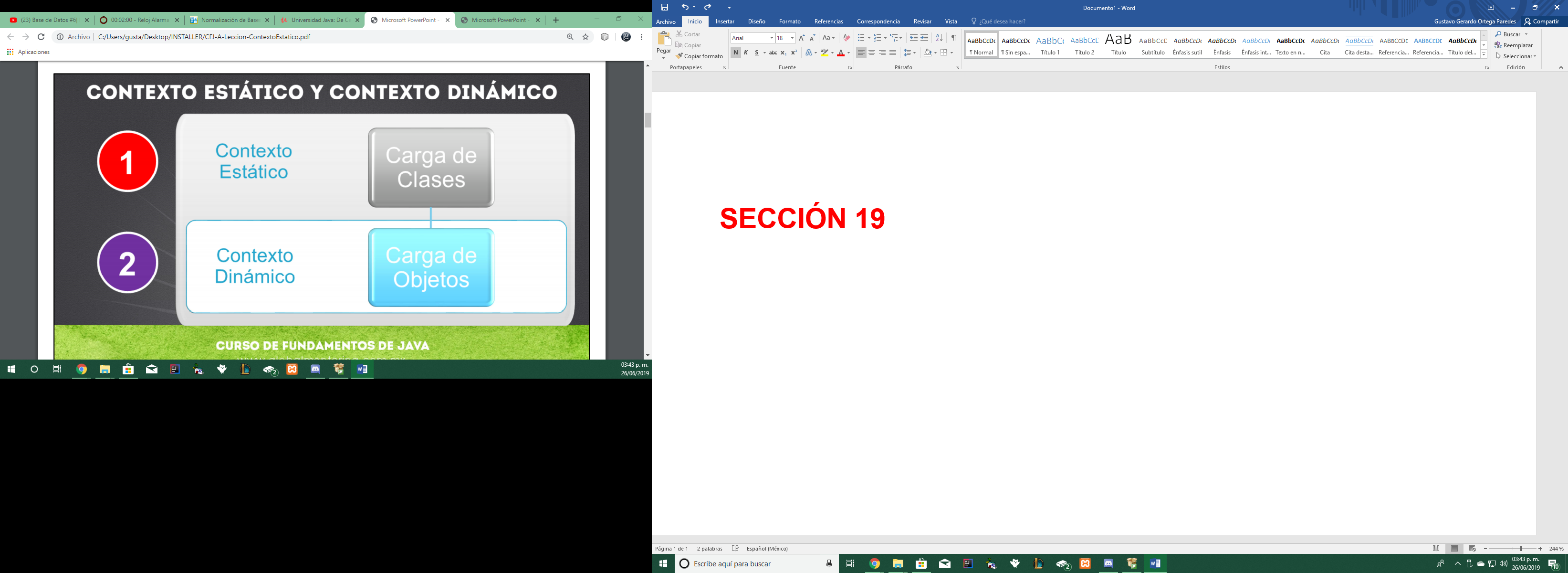
**SECCIÓN 19**

**CONTEXTO ESTÁTICO**

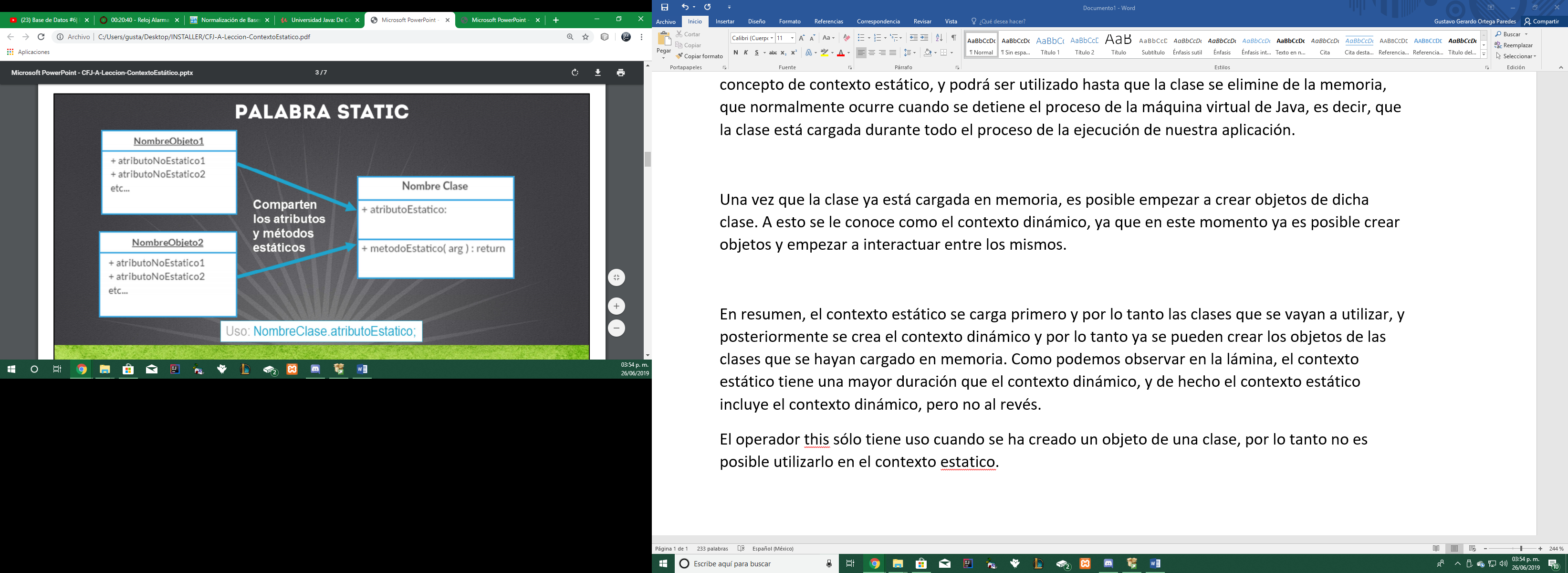


Para comenzar a crear objetos de una clase, primero se debe cargar la clase en memoria por medio de lo que se conoce como ClassLoader (cargador de clases). Aquí entra por primera vez el concepto de contexto estático, y podrá ser utilizado hasta que la clase se elimine de la memoria, que normalmente ocurre cuando se detiene el proceso de la máquina virtual de Java, es decir, que la clase está cargada durante todo el proceso de la ejecución de nuestra aplicación.

Una vez que la clase ya está cargada en memoria, es posible empezar a crear objetos de dicha clase. A esto se le conoce como el contexto dinámico, ya que en este momento ya es posible crear objetos y empezar a interactuar entre los mismos.

En resumen, el contexto estático se carga primero y por lo tanto las clases que se vayan a utilizar, y posteriormente se crea el contexto dinámico y por lo tanto ya se pueden crear los objetos de las clases que se hayan cargado en memoria. Como podemos observar en la lámina, el contexto estático tiene una mayor duración que el contexto dinámico, y de hecho el contexto estático incluye el contexto dinámico, pero no al revés.

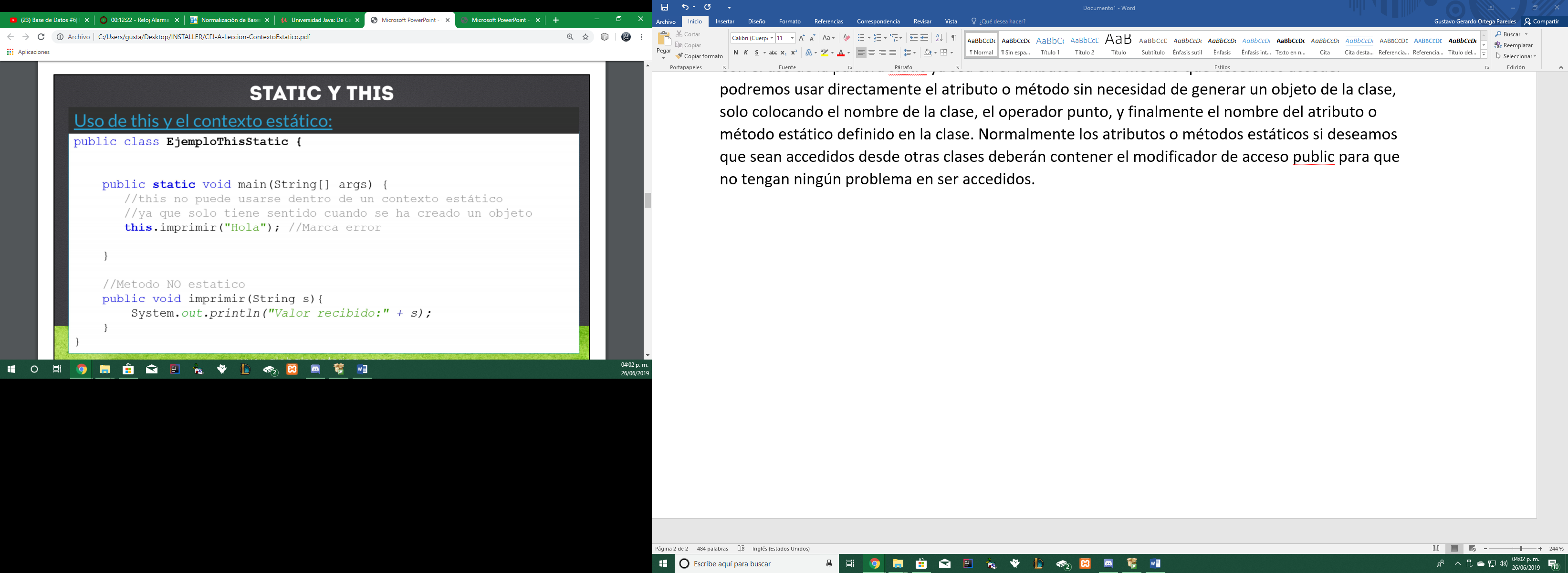
El operador this sólo tiene uso cuando se ha creado un objeto de una clase, por lo tanto no es posible utilizarlo en el contexto estatico.



Podemos utilizar la palabra **static** para interactuar con el contexto estático. Por ejemplo, si definimos un atributo o un método como estático, lo que estamos indicamos es que el atributo o método pertenecen a la clase y no al objeto.

Por ejemplo, si creamos un atributo sin usar la palabra static, que es como normalmente los definimos, cada que creamos un objeto se creará también una variable asociada al objeto que se crea, pero si definimos el atributo como estático, estamos indicando que el atributo solo se crea una vez, sin importar cuantos objetos se creen, sólo habrá una variable la cual se asocia a la clase y no al objeto. Y si un objeto accede al valor de la variable estática leerá el mismo valor que los demás objetos, y si un objeto modifica el valor estático, todos los demás objetos accederán al mismo valor ya que este valor está almacenado en la clase y no en los objetos de dicha clase.

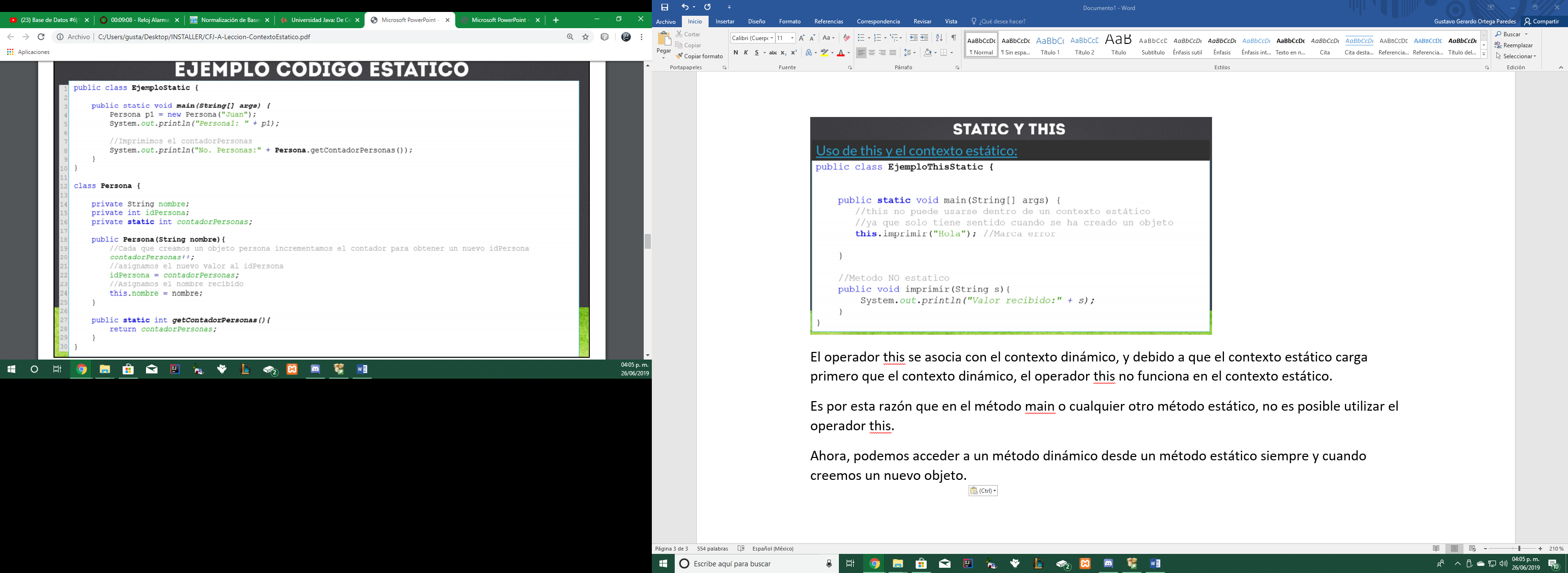
Con el uso de la palabra static ya sea en el atributo o en el método que deseamos acceder podremos usar directamente el atributo o método sin necesidad de generar un objeto de la clase, solo colocando el nombre de la clase, el operador punto, y finalmente el nombre del atributo o método estático definido en la clase. Normalmente los atributos o métodos estáticos si deseamos que sean accedidos desde otras clases deberán contener el modificador de acceso public para que no tengan ningún problema en ser accedidos.



El operador this se asocia con el contexto dinámico, y debido a que el contexto estático carga primero que el contexto dinámico, el operador this no funciona en el contexto estático.

Es por esta razón que en el método main o cualquier otro método estático, no es posible utilizar el operador this.

Ahora, podemos acceder a un método dinámico desde un método estático siempre y cuando creemos un nuevo objeto.



Por un lado declaramos una variable de tipo static llamada contadorPersonas. Al declarar esta variable como estática, lo que buscamos es que cada vez que creamos un objeto de tipo Persona, vamos a incrementar el valor de esta variable. Esto lo podemos lograr utilizando el constructor de la clase, ya que cada que creamos un objeto de tipo Persona, se mandará llamar el constructor de la clase, y allí podemos incrementar esta variable con cada llamada realizada al constructor (línea 20).

Una vez que se ha incrementado esta variable estática, este valor aplica para todos los objetos, por lo tanto cada que incrementamos esta variable puede ser consultada por todos los objetos que creemos y de esta manera es que esta variable no comienza desde cero cada que creamos un nuevo objeto, sino que se queda en el último valor y se incrementa una vez más, esto debido a que esta variable pertenece a la clase y no a los objetos.

Ahora, en el método main, el cual es un método estático, vamos a hacer uso de la clase Persona. En primer lugar crearemos un objeto de tipo Persona, y asignamos el valor de Juan al atributo nombre con ayuda del constructor de la clase.

Desde el momento que estamos utilizando la palabra new, quiere decir que ya estamos trabajando con el contexto dinámico, esto permite trabajar ya con el contexto dinámico (objetos) y el contexto estático (clases).

En este código estamos omitiendo la implementación del método toString para la clase Persona, el cual se debe agregar para observar el estado del objeto persona impreso en la línea 5.

Finalmente, una vez que hemos terminado de crear objetos de tipo persona, podemos mandar a imprimir el número total de objetos Persona que se han creado, y aquí es donde podemos observar la notación que hemos platicado, para mandar a llamar un método estático público de una clase basta con colocar el nombre de la clase, el operador punto y el nombre del método estático, es decir que no debemos utilizar el objeto, de hecho no es buena práctica y si insistimos en usar una variable de tipo object para mandar a llamar un método estático nos mostrará un warning, avisándonos que es una mala práctica.