****

**INFORME USO DE HERRAMIENTAS DE PROGRAMACION**

****

**PRESENTADO POR:**

**ANGIE PAOLA VILLADA ORTIZ**

**UNIVERSIDAD TECNOLOGICA DE PEREIRA**

**FACULTAD DE INGENIERIAS**

**INGENIERIA DE SISTEMAS Y COMPUTACION**

**PEREIRA, RISARALDA**

**2019**

**PRESENTADO A:**

**LEONARDO TAFFURHT**

**UNIVERSIDAD TECNOLOGICA DE PEREIRA**

**FACULTAD DE INGENIERIAS**

**INGENIERIA DE SISTEMAS Y COMPUTACION**

**PEREIRA, RISARALDA**

**2019**

****

**INFORME USO DE HERRAMIENTAS DE PROGRAMACION**

**PRESENTADO POR:**

**ANGIE PAOLA VILLADA ORTIZ**

**PRESENTADO A:**

**JULIO CESAR CHAVARRO**

**UNIVERSIDAD TECNOLOGICA DE PEREIRA**

**FACULTAD DE INGENIERIAS**

**INGENIERIA DE SISTEMAS Y COMPUTACION**

**PEREIRA, RISARALDA**

**2019**

MAKE:

 es un programa para la construcción de código. Su función principal es evitarnos recompilar archivos innecesariamente. Se basa en reglas de dependencia, que especifican qué ficheros dependen de cuales, y órdenes de construcción.

VENTAJAS:

* compilación de programas grandes.
* Automatización de tareas de reconstrucción o regeneración de ficheros.
* Es capaz de saber qué cosas hay que recompilar. Si cuando estamos depurando nuestro programa tocamos un fichero fuente, al compilar con **make** sólo se recompilarán aquellos ficheros que dependan del que hemos tocado. Si compilamos a mano con CC, (o el compilador que sea), o tenemos en la cabeza esas dependencias para compilar sólo lo que hace falta, o lo compilamos todo. Si el proyecto es grande, se nos olvidará alguna dependencia o nos pasaremos horas compilando.
* Nos guarda los comandos de compilación con todos sus parámetros para encontrar librerías, ficheros de cabecera (.h), etc., etc. No tendremos que escribir largas líneas de compilación con montones de opciones que debemos saber de memoria o, al menos, sólo tendremos que hacerlo una vez.

DESVENTAJAS:

* La preparación del *makefile* es laboriosa. Incluso contando con herramientas para detección de dependencias, hace falta en muchos casos introducir manualmente gran parte de la información: listas de ficheros, opciones de compilación, etc.
* La reconstrucción no es segura, ya que no se detectan cambios hacia atrás en el tiempo. Esto ocurre cuando se reemplaza un fichero deficiente por una versión anterior correcta.
* La reconstrucción puede no ser tan eficiente como cabría esperar, a pesar de que la eficiencia es uno de los objetivos de la herramienta *make*. Por ejemplo, se reconstruyen ficheros finales a partir de otros intermedios, aunque éstos realmente no cambien, como ocurre con los ficheros objeto al recompilar después de cambiar sólo comentarios en el texto fuente.
* Por otra parte, y en relación con lo anterior, las dependencias se establecen e nivel de ficheros, y no de partes de ficheros. Esto hace en algunos casos que se recompile prácticamente toda una aplicación tras modificar ligeramente un fichero de cabecera básico, de bajo nivel.
* Los ficheros *makefile* son poco transportables. En particular las acciones son específicas para cada plataforma: ‘*Shell*’, compilador, etc.

ANT:

Ant es una herramienta Open-Source utilizada en la compilación y creación de programas Java; si está familiarizado con Linux, Ant es considerado un make para Java.

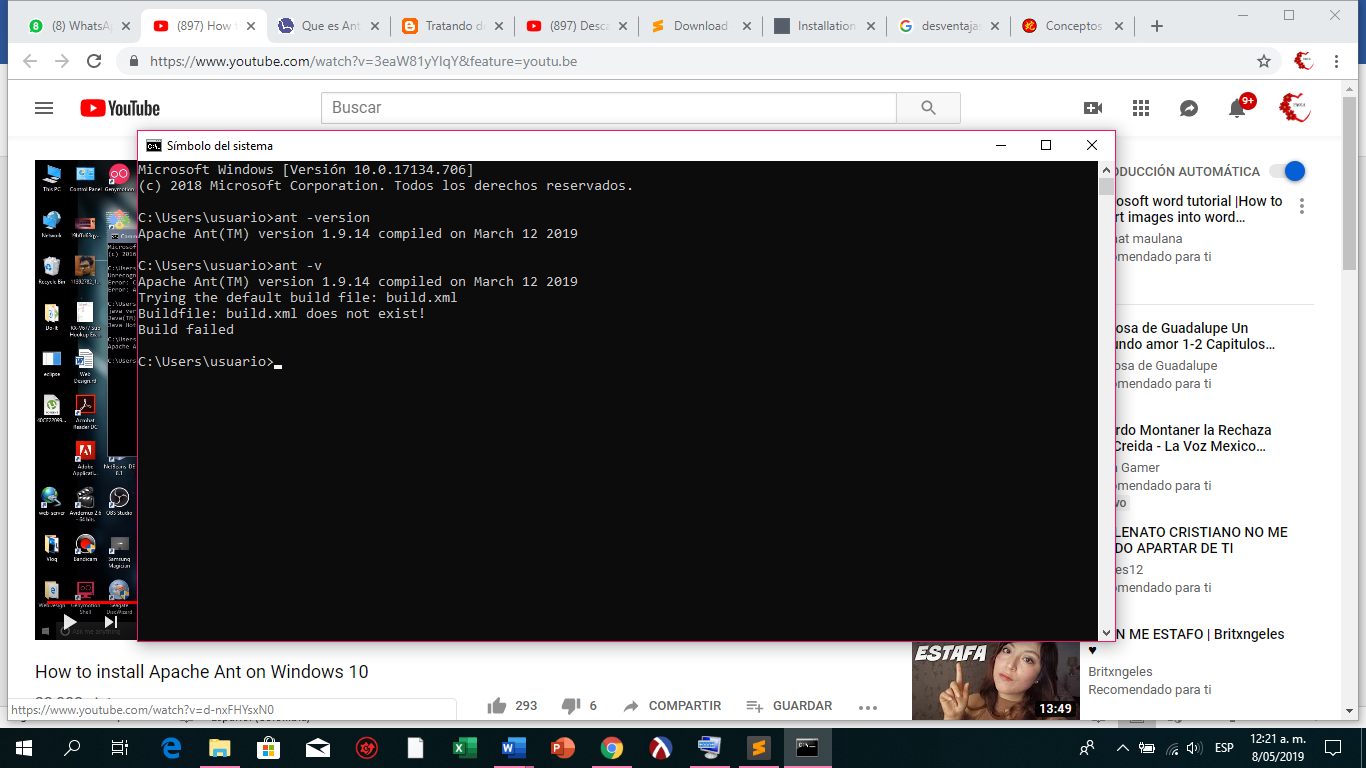
VENTAJAS:

* Ant está escrito en XML y Java, esto permite ofrecer una solución interoperable al nivel de sistema operativo (debido a Java) y configuraciones descriptivas (debido a XML).
* utiliza el leguaje demarcado para poder ejecutar los demás comandos muy diferente de Make que utiliza los Makefile

DESVENTAJAS:

* El tamaño de los proyectos afectaba el buen manejo por medio de una sintaxis.

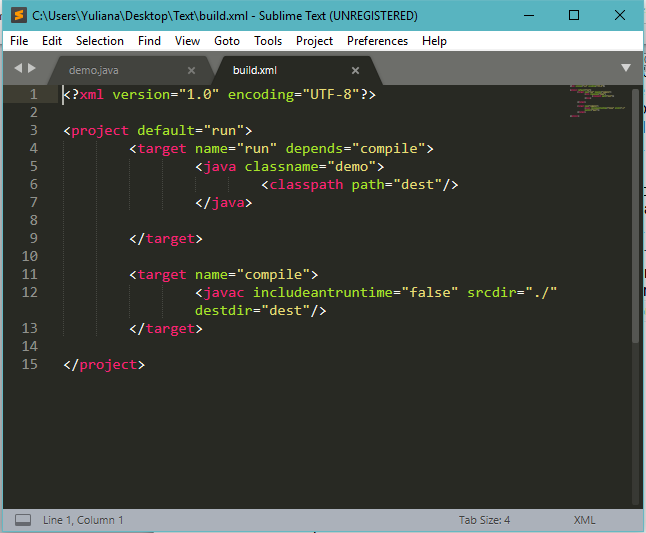
INSTALACION:

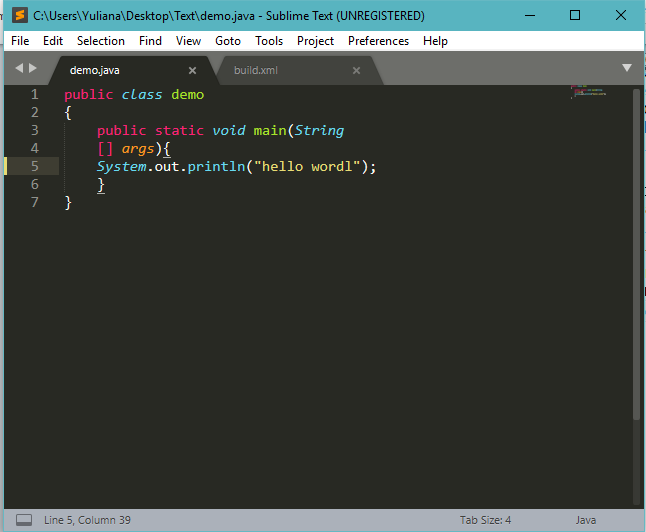


* Después de haber instalado ant, procederemos a desarrollar los siguientes pasos con el fin de ejecutar hola mundo.

1 PASO:

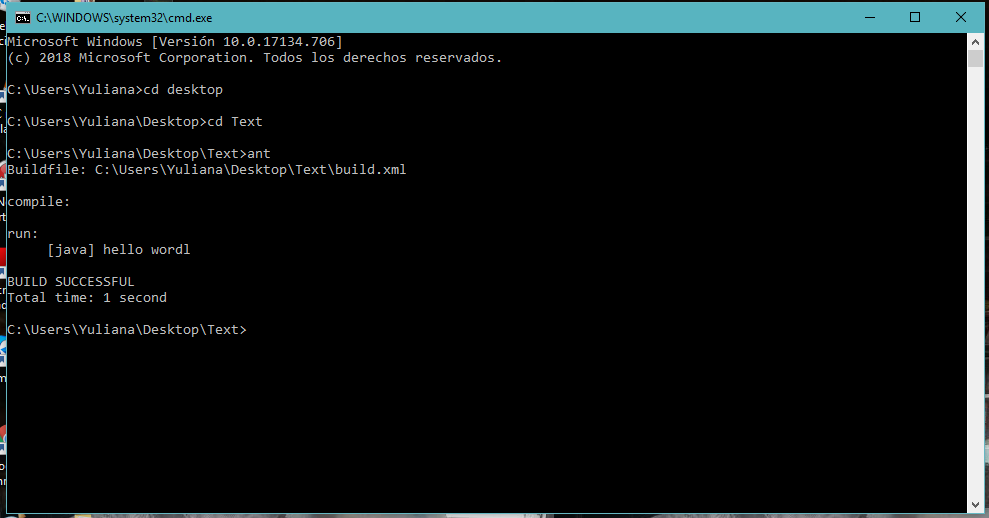
* Descargamos un editor de texto llamado Sublime 3, donde se crearan las extensiones XML y java .





2 PASO:

* Después de que hallamos escrito las líneas de código anteriores procedemos a ejecutar el comando con las siguientes instrucciones.



* Teniendo como resultado nuestro objetivo planteado, el cual era generar “hola mundo”.

MAVEN:

Maven es una herramienta de gestión de proyectos que se basa en la “convención sobre configuración”. Es decir, asume un comportamiento por defecto que permite empezar a trabajar sin necesidad de configuración. Por ejemplo, Maven asume la estructura de los ficheros del proyecto, con lo cual a la hora de compilar no se pierde tiempo indicando donde se encuentra el código fuente o las librerías como en Ant, Maven define, entre otras cosas:

1. Un modelo de objeto del proyecto.
2. Un sistema de gestión de dependencias.
3. El ciclo de vida del proyecto.
4. La lógica para ejecutar nuevas tareas en determinadas fases del ciclo de vida.

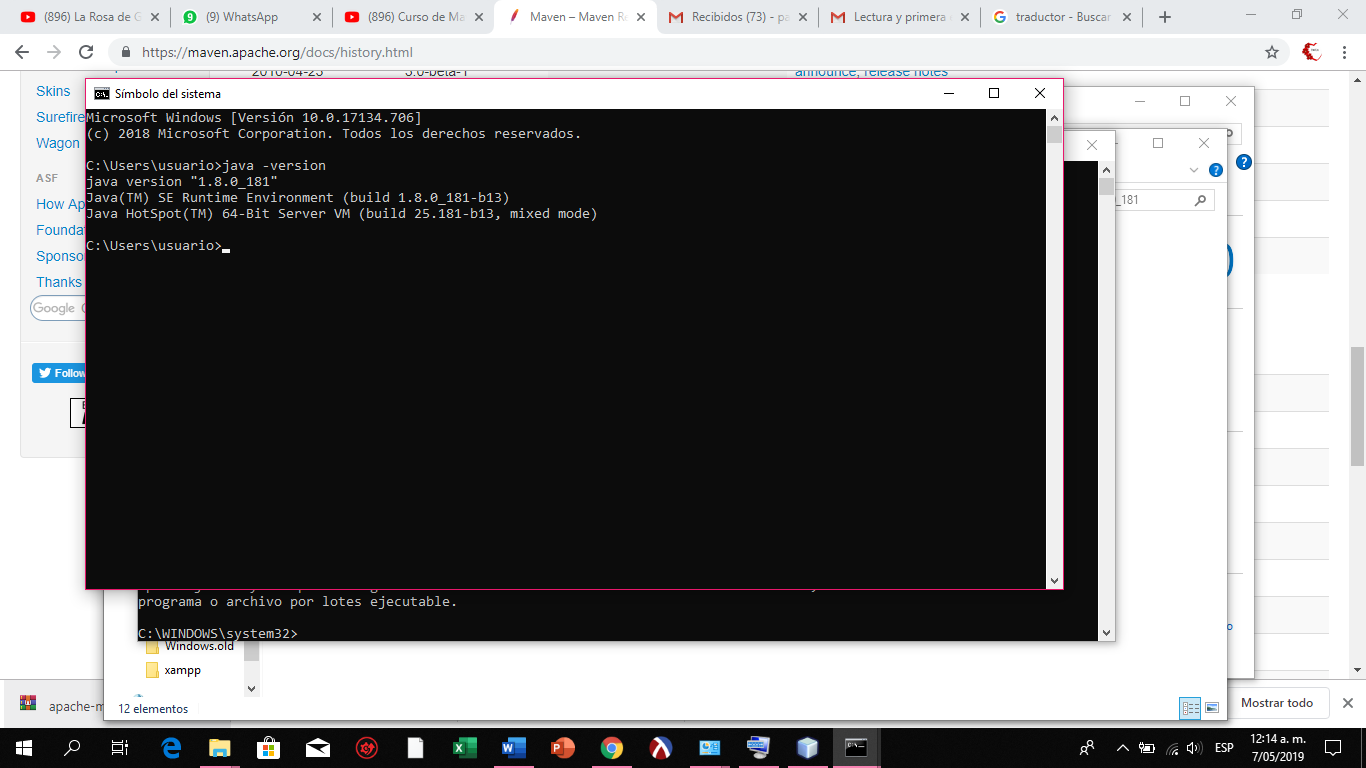
VENTAJAS:

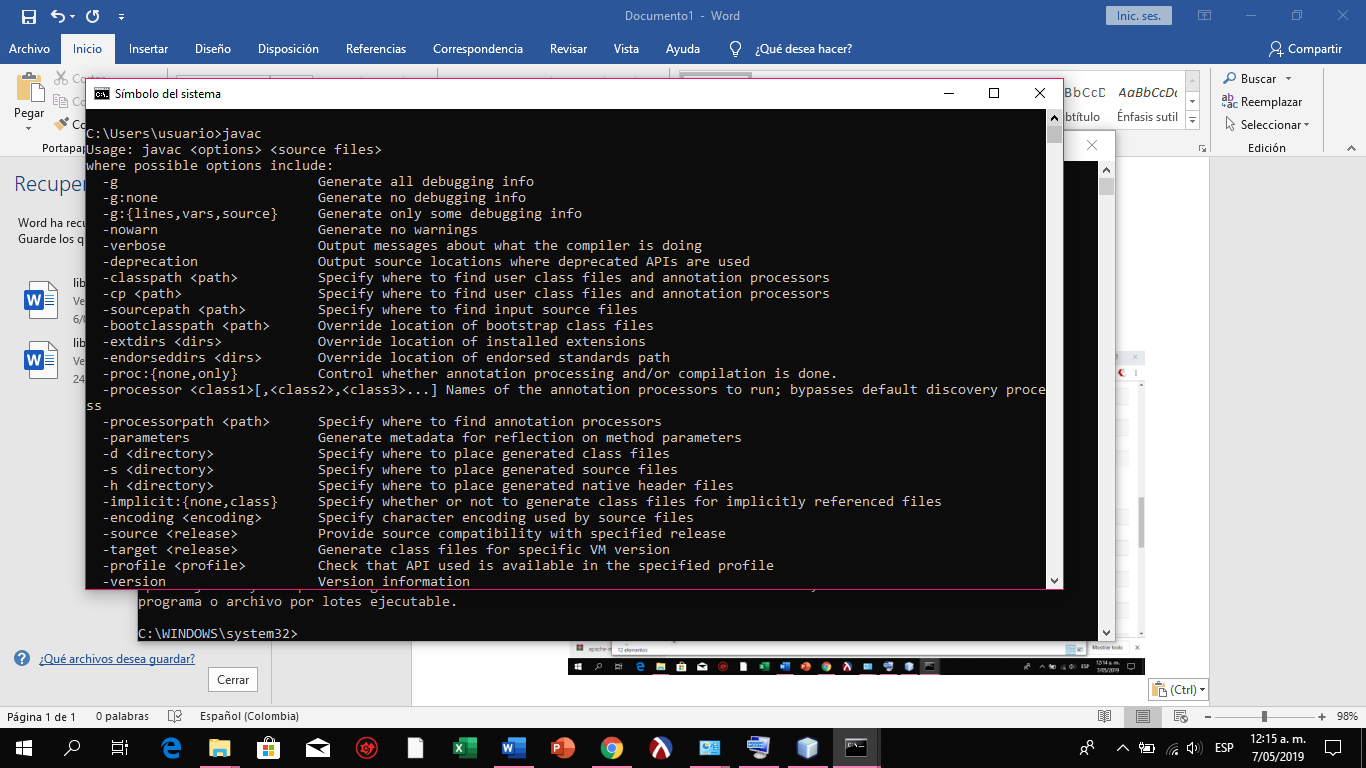
* se evitan tiempos de configurar e implementar el entorno de desarrollo para poder centrarnos en lo realmente interesante de un proyecto Java: desarrollar y documentar nuestro código

DESVENTAJAS:

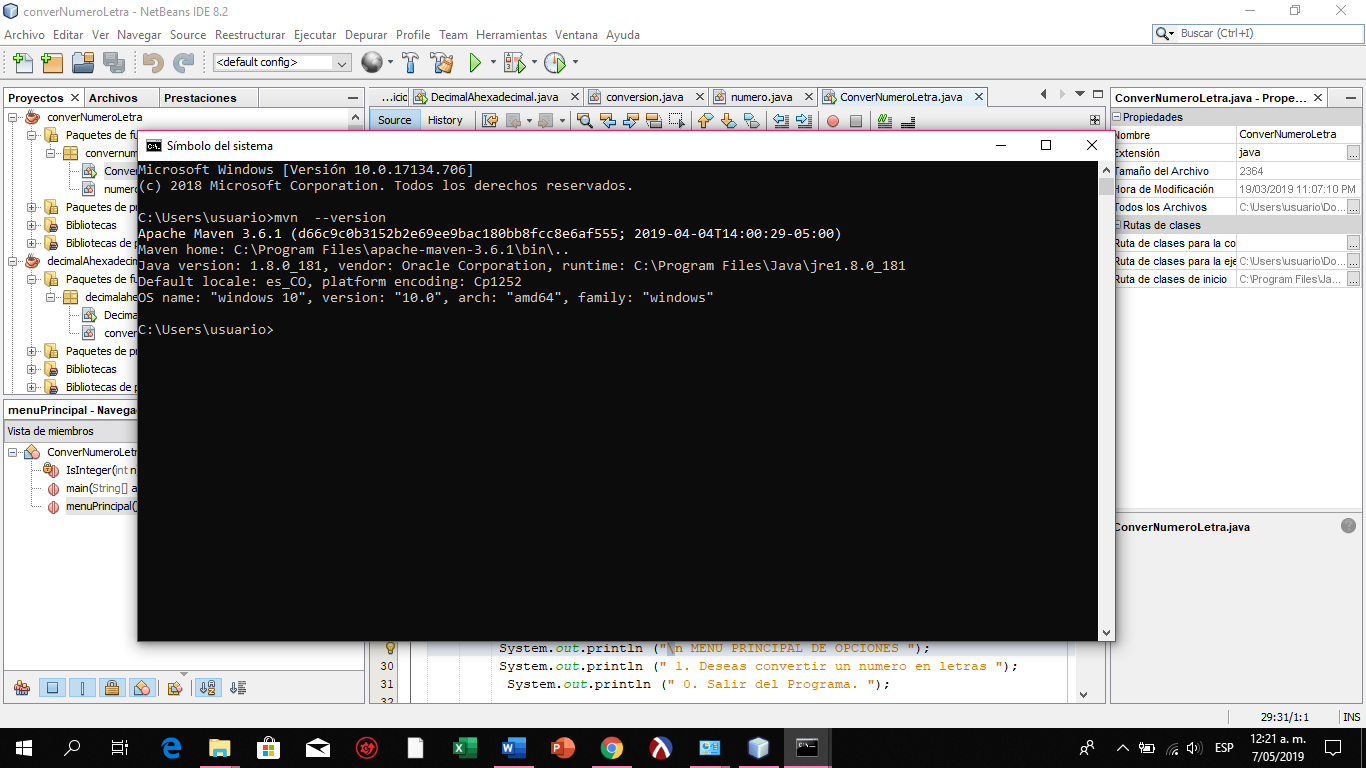
* ya que este maneja todo el ciclo completo de compilación volviéndose así un poco complejo y extenso, además Trabaja en xml.

INSTALACION DE JDK:





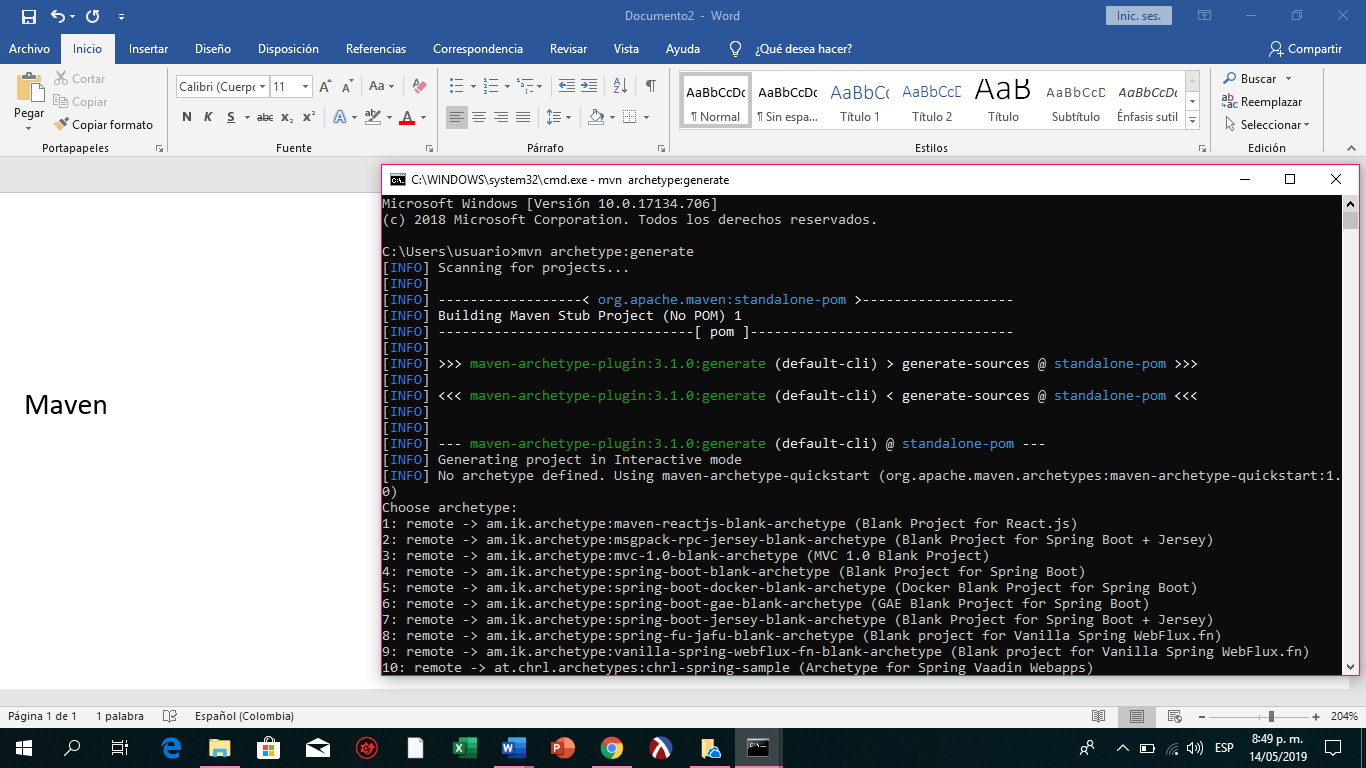
INSTALACION DE MAVEN:

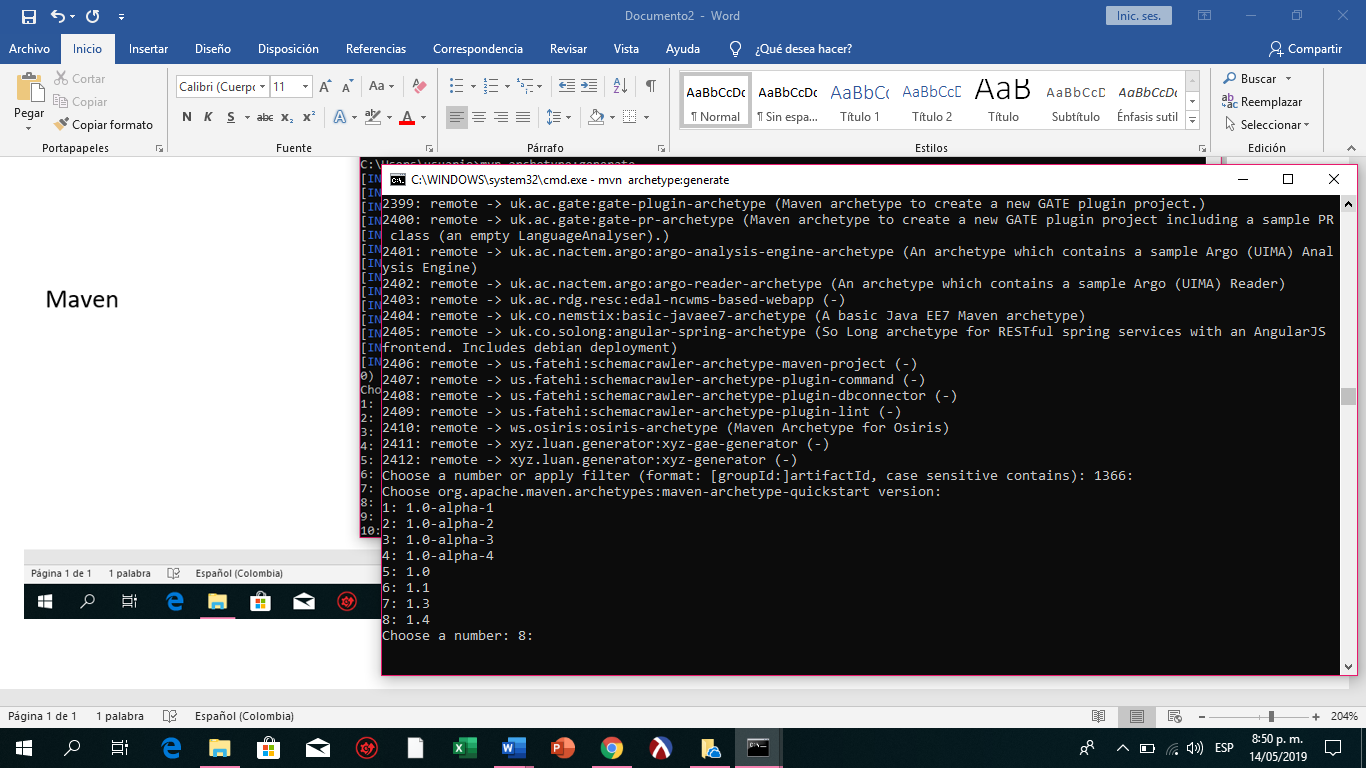


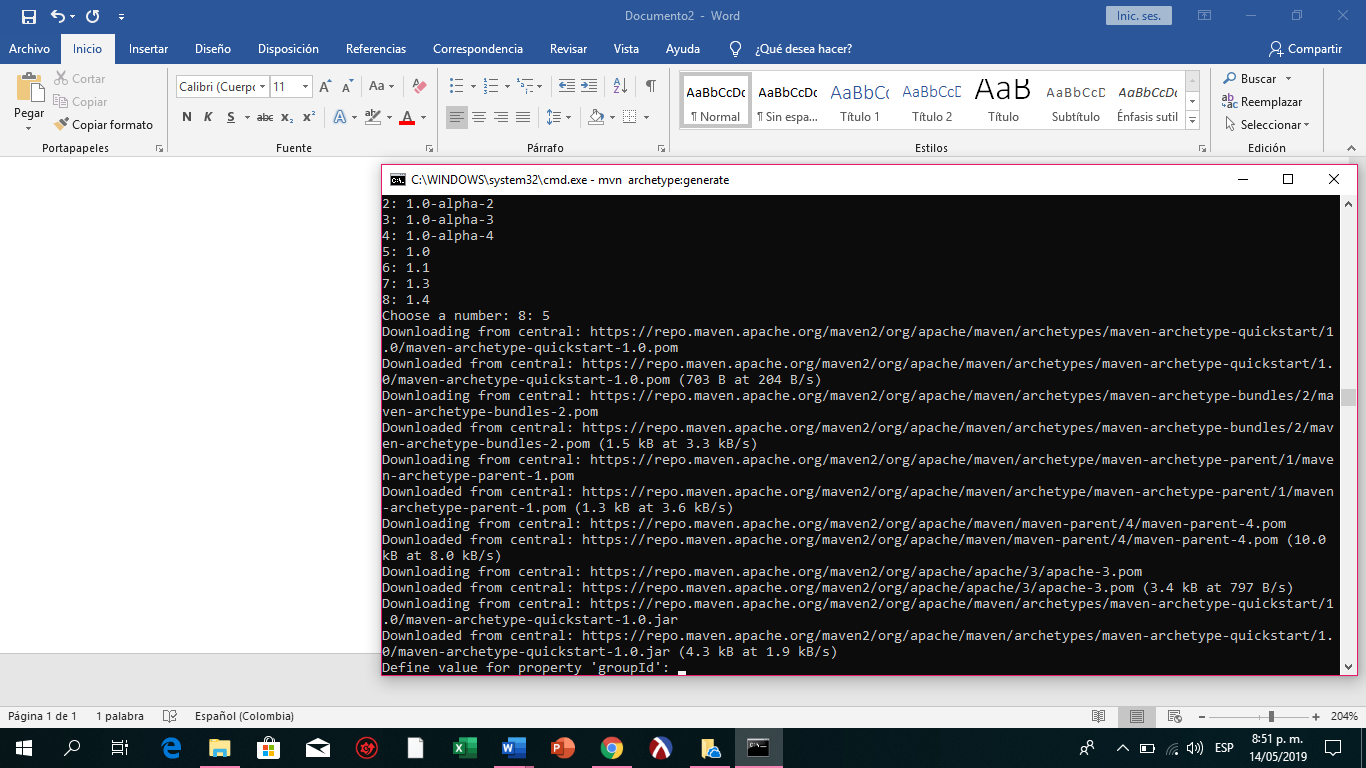
USO DE MAVEN:

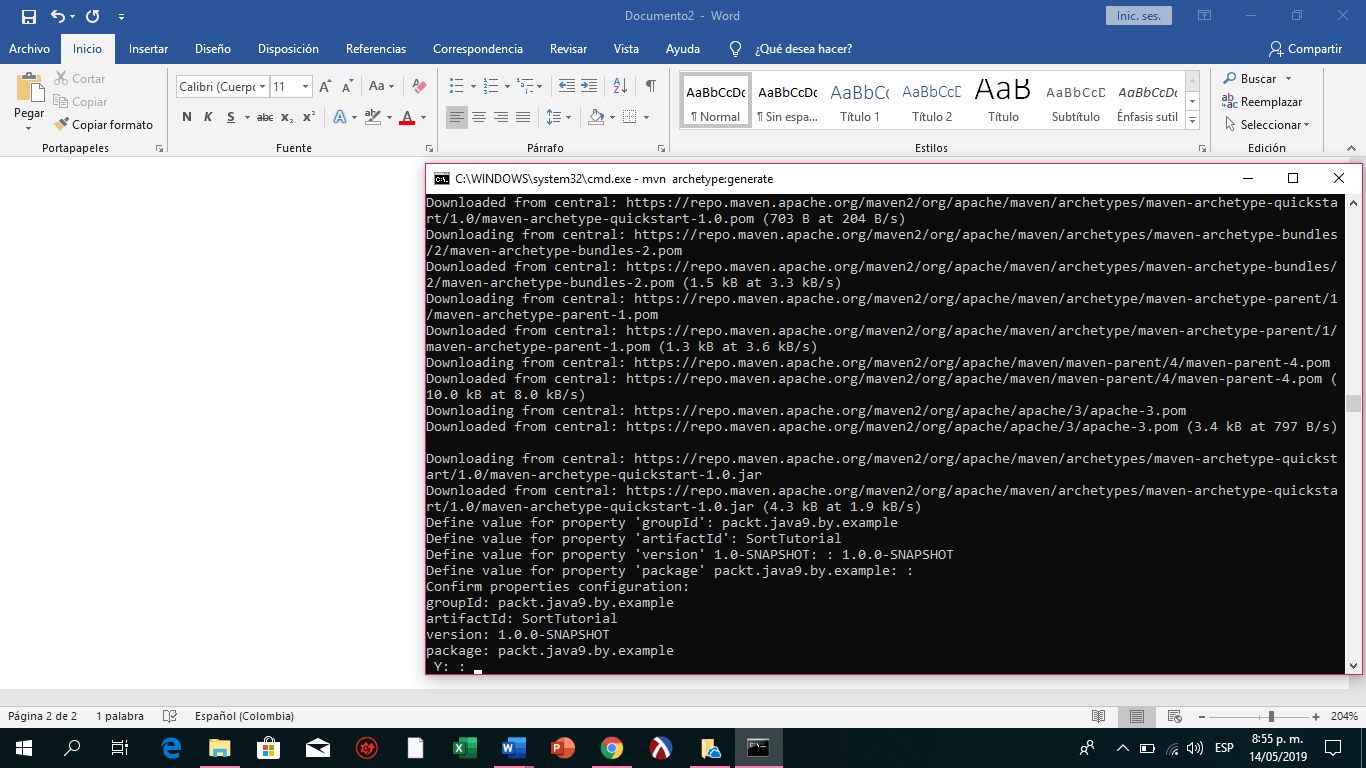
1 PASO:

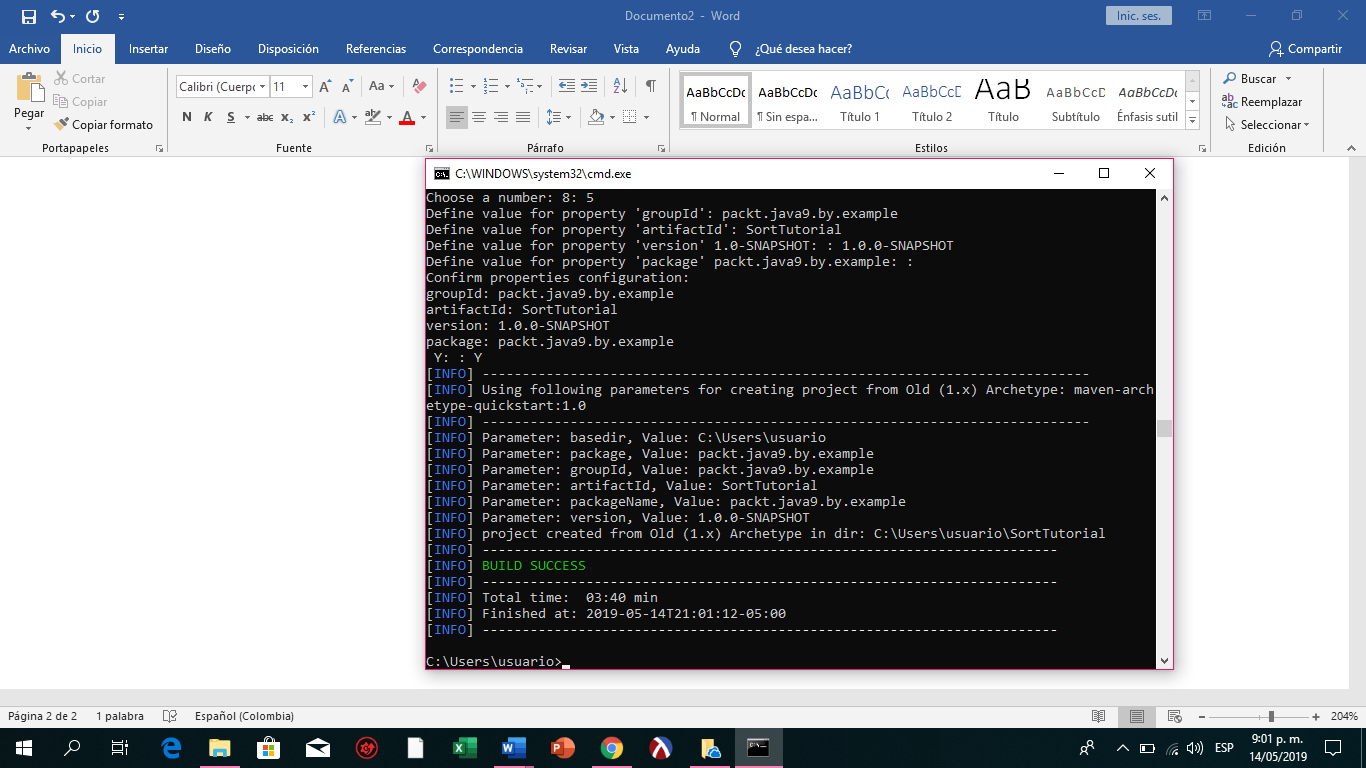
* crearemos el proyecto.





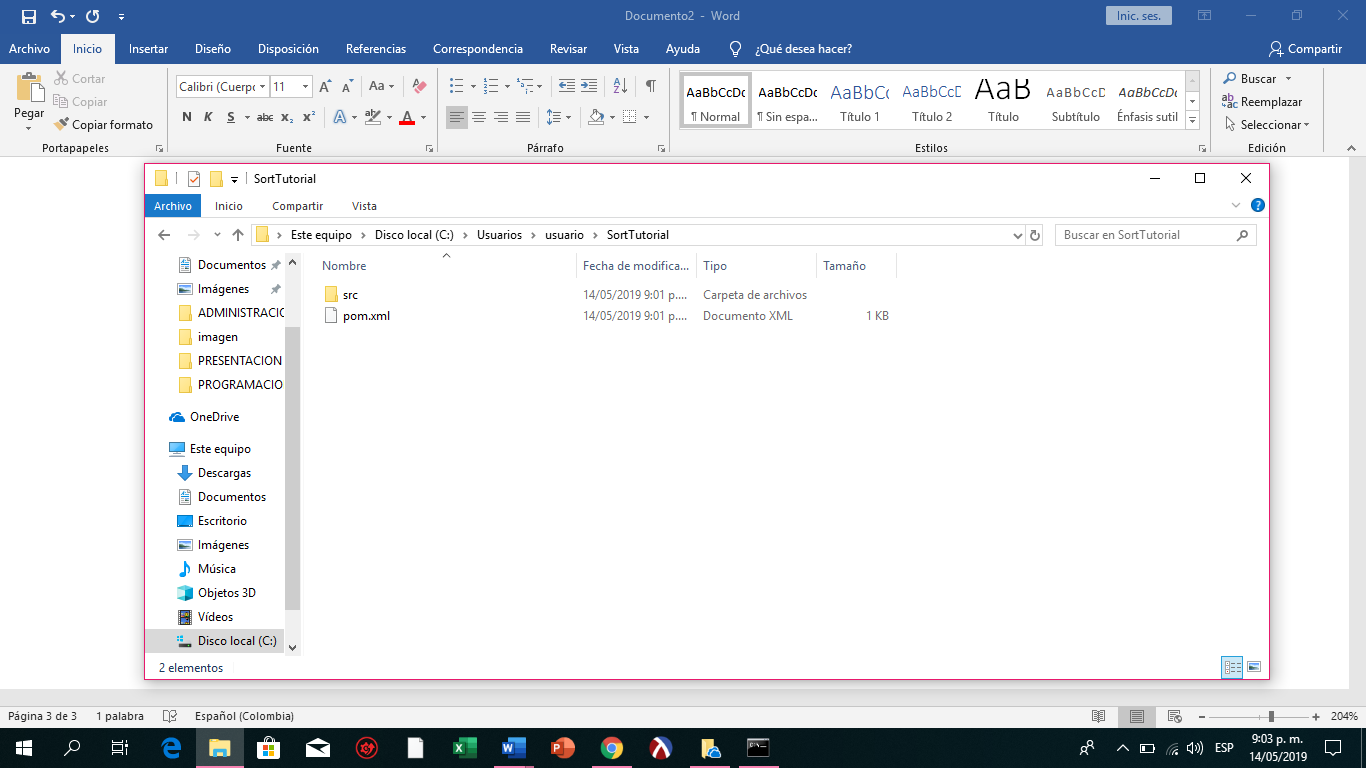






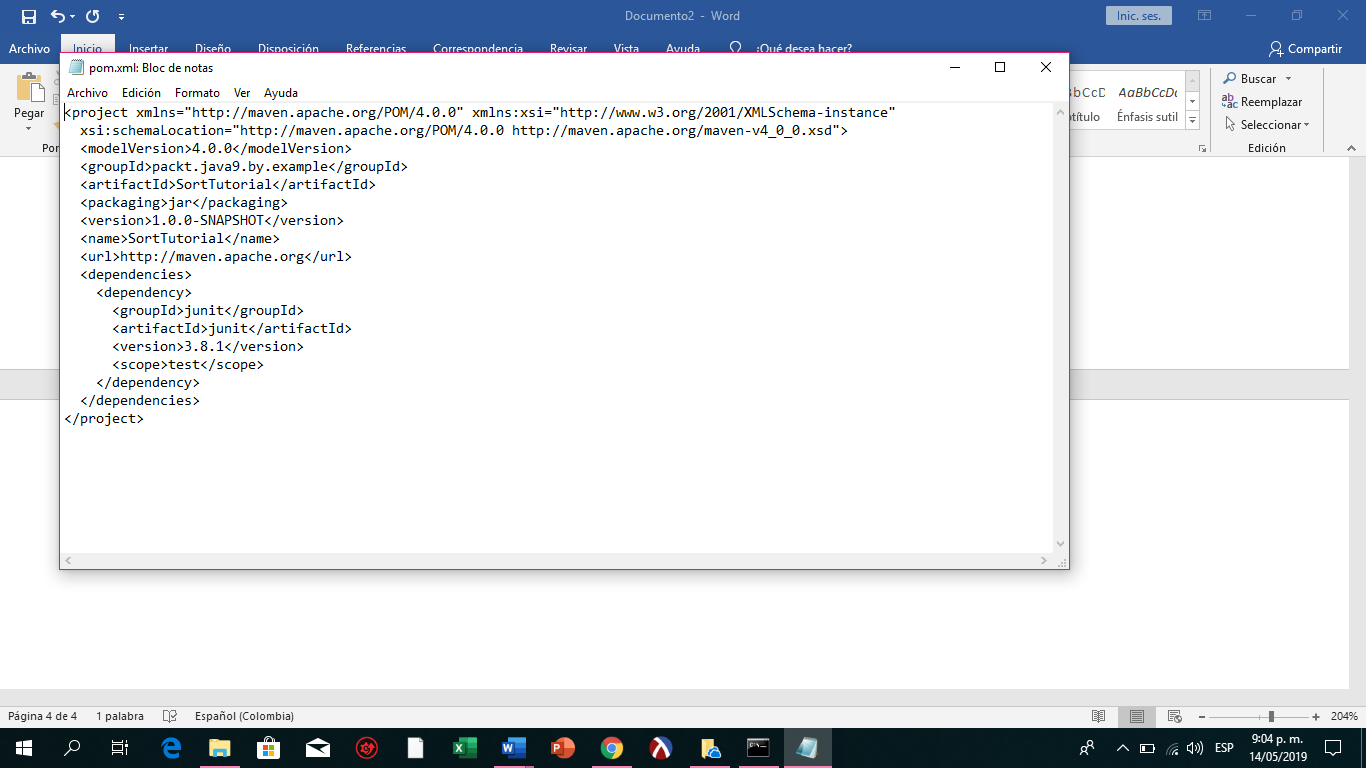
2 PASO:

* una ves que hallamos seguido los pasos anteriores procedemos a ir a disco local/usuarios/usuario/sorftTutorial donde encontraremos esto:



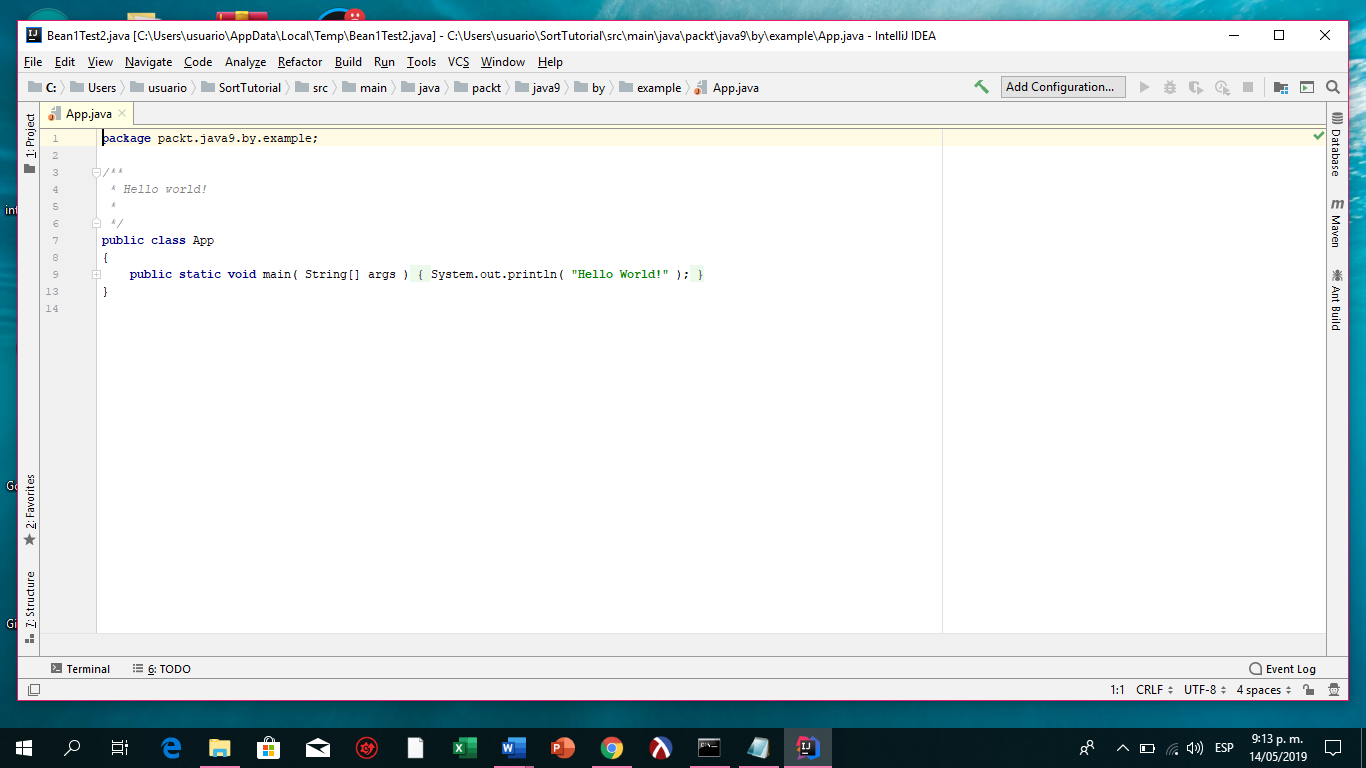
3 PASO:

* Para verificar si se creo el proyecto abrimos pom.XML.



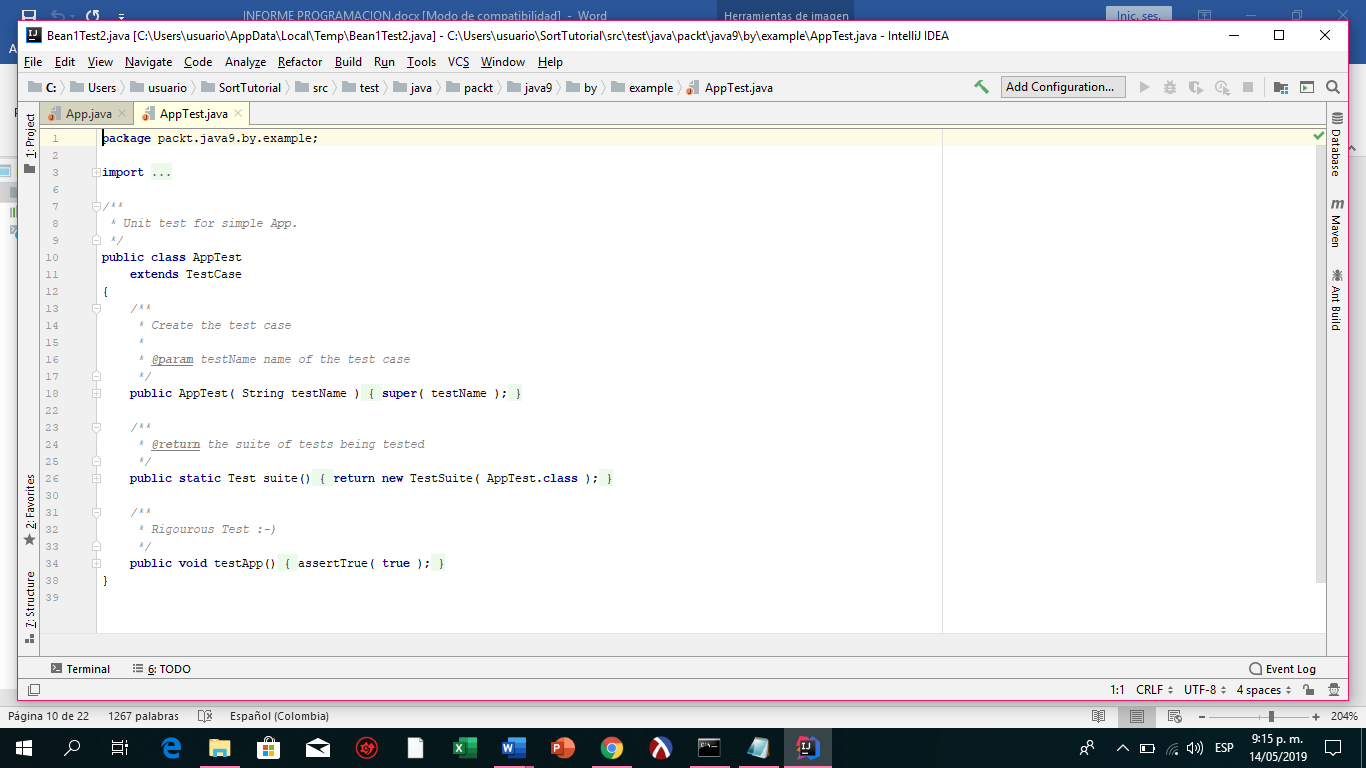
4 PASO:

* Después de haber comprobado que se creó el proyecto.
* Descargaremos intellij.
* Abrimos sorftTutorial/src/main/java/packt/java9/by/example/App.java.



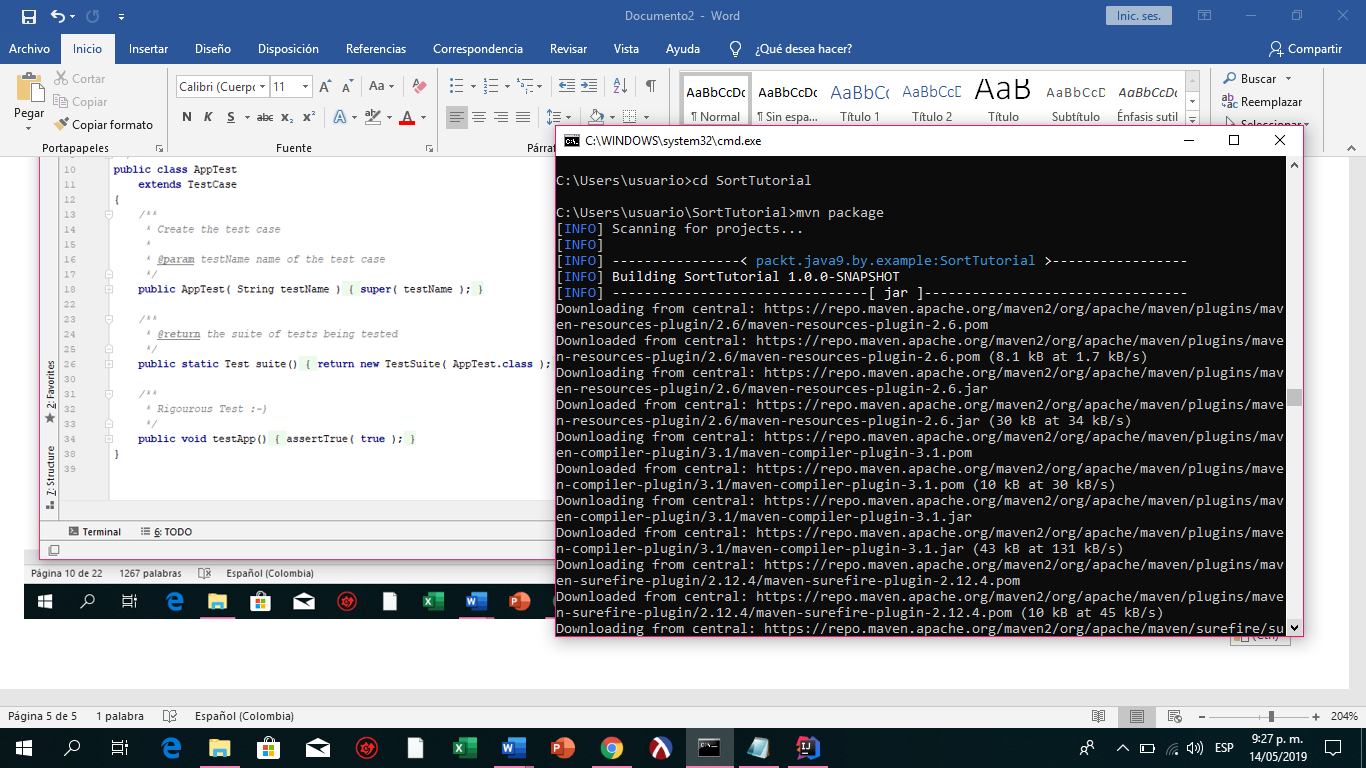
5 PASO:

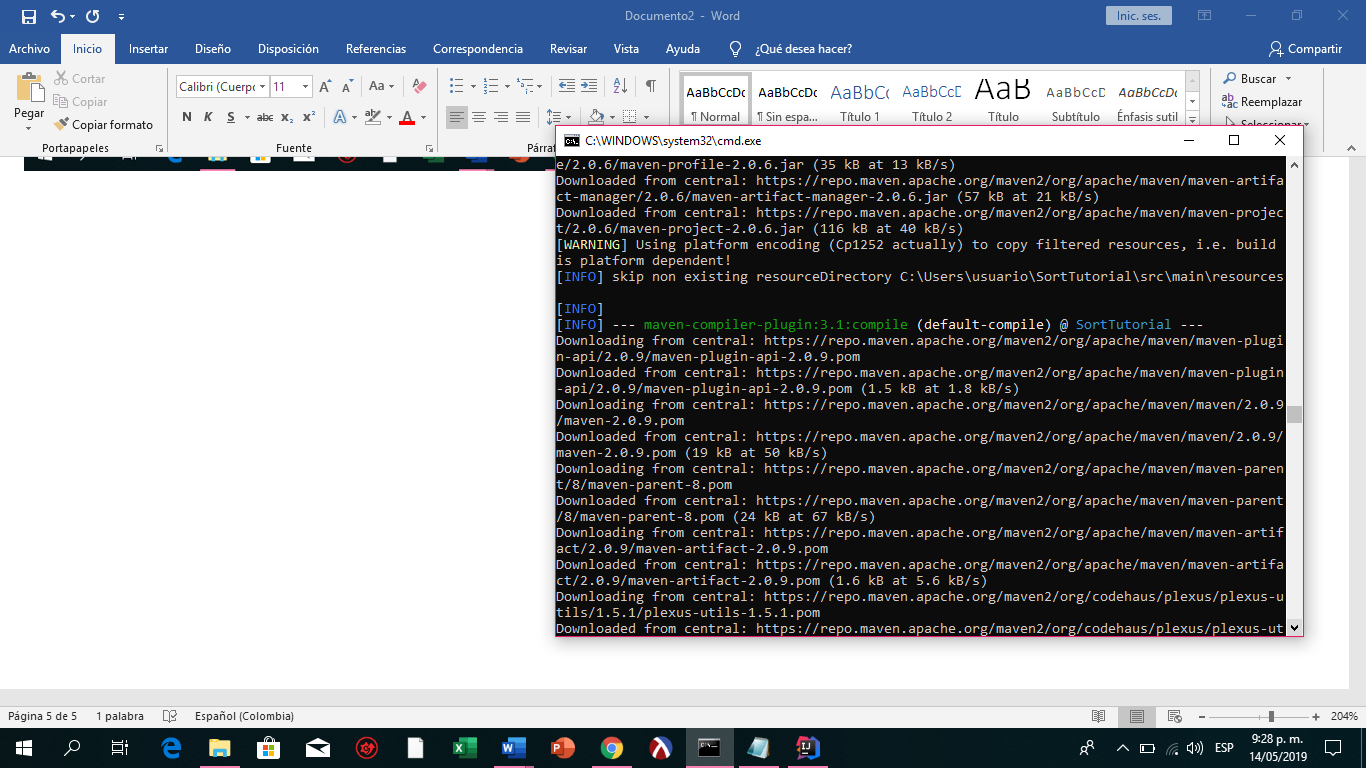
* Abrimos sorftTutorial/Text/java/packt/java9/by/example/AppTest.java.

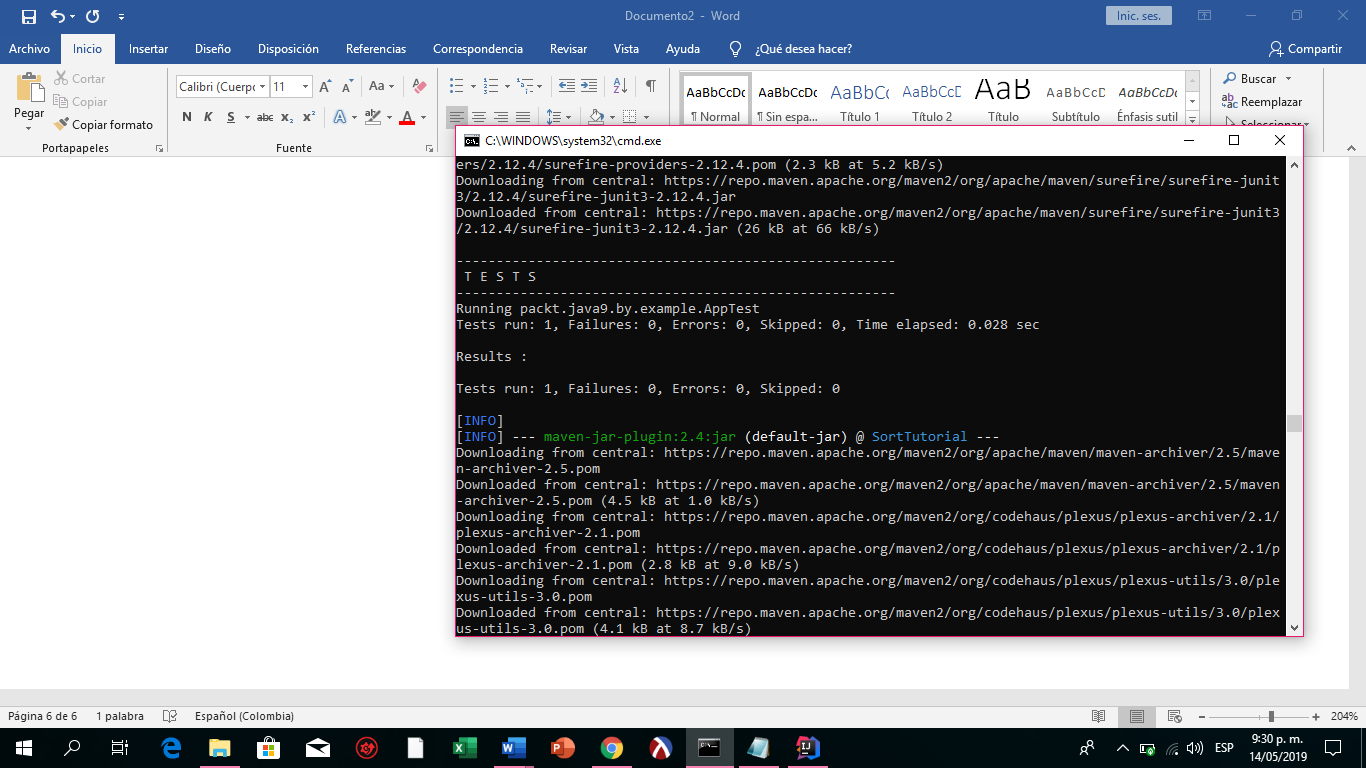


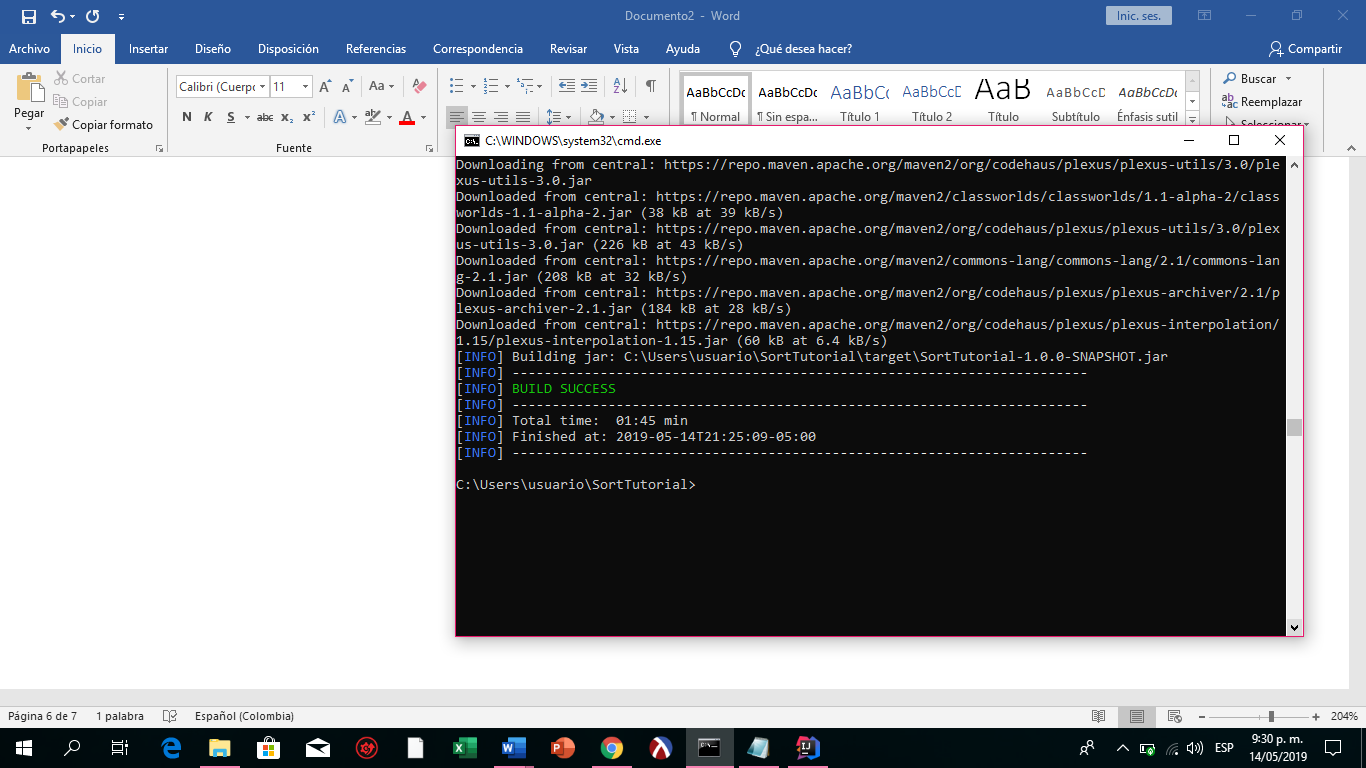
6 PASO:

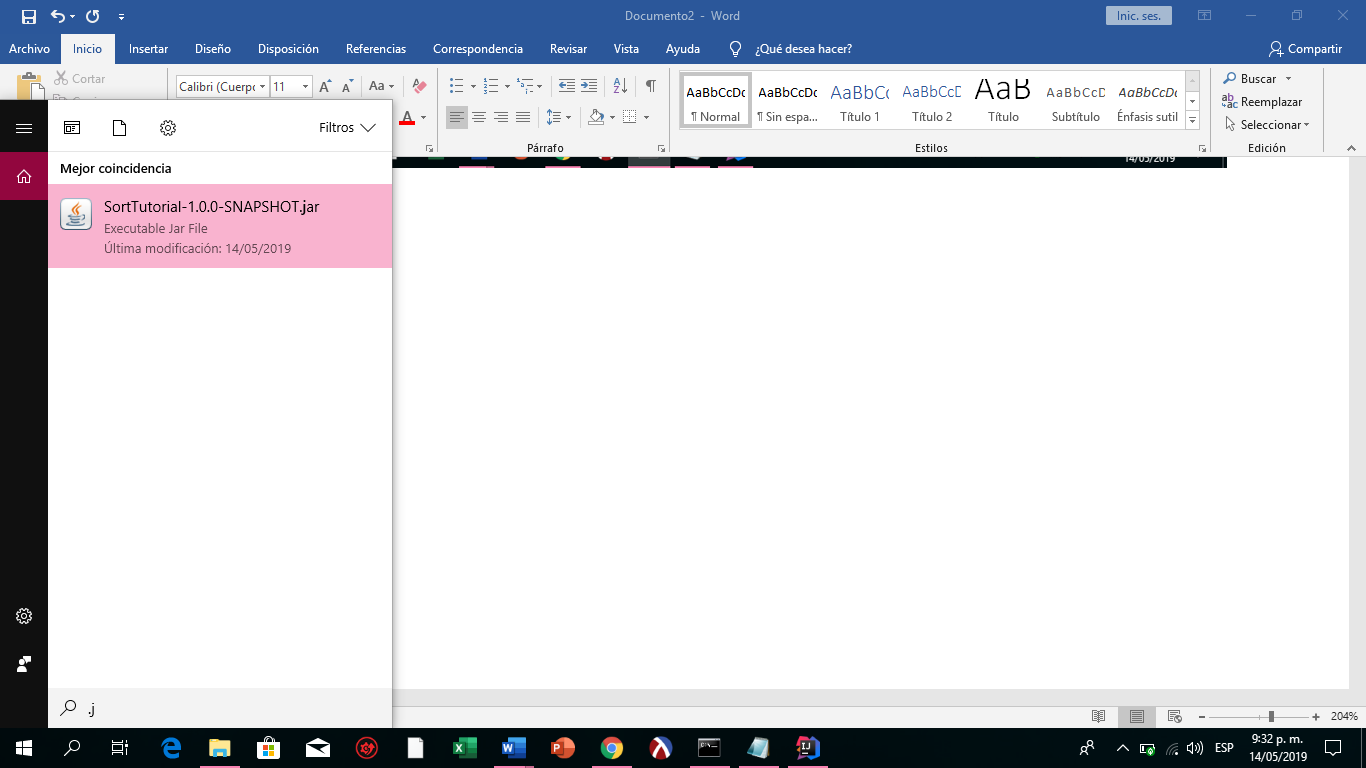
* Ejecutamos el comando(cmd).
* Crearemos el (.jar) del proyecto.





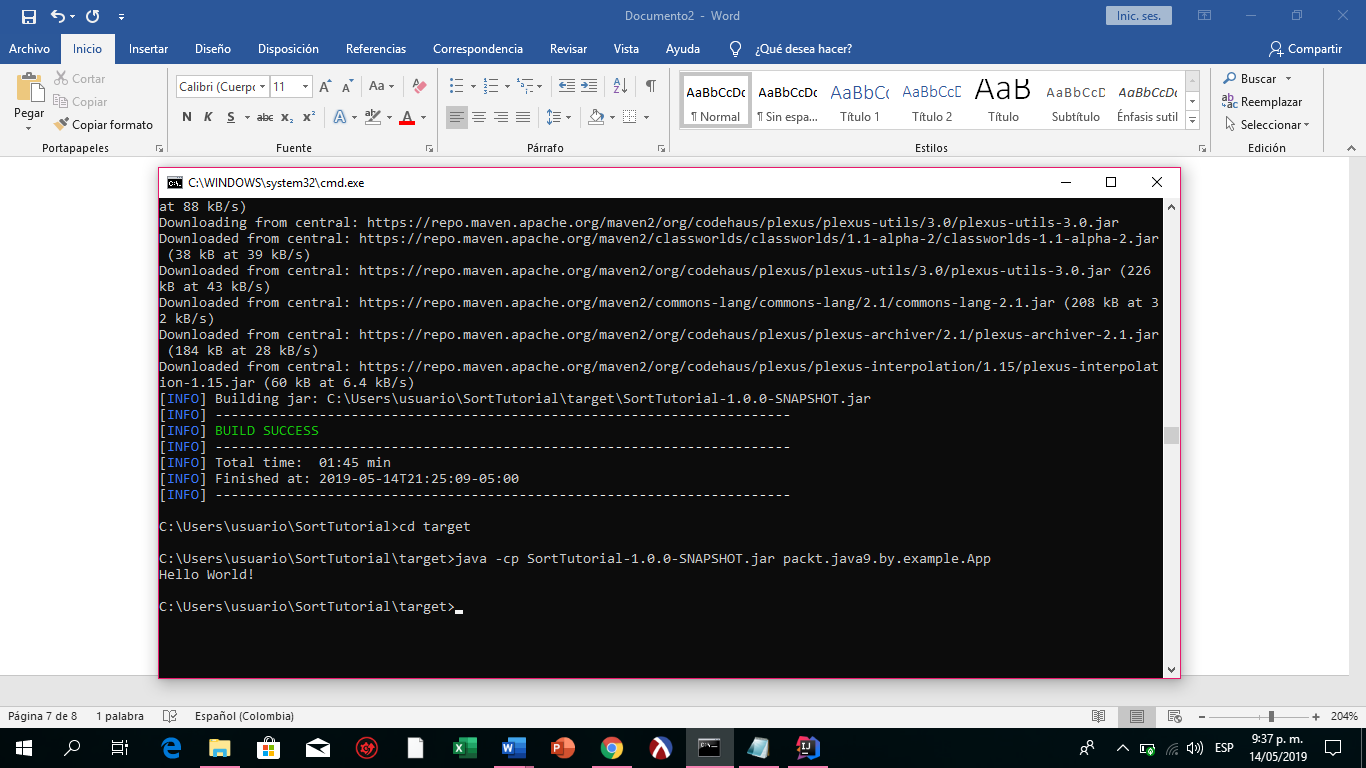






7 PASO:

* Procedemos a la ejecución de hola mundo



GRADLE:

Es una herramienta para la construcción de proyectos basados en características de herramientas de Maven y Ant, también se puede decir que es un conjunto de jar que corre sobre la máquina virtual.

VENTAJAS:

* Capacidad de añadir complementos para incorporar funcionalidades extras, entré ellos compilar proyectos (gradle, java, gradle con java).
* Agrega objectos al repositorio.
* Herramienta multiplataforma, no requiere de un entorno de desarrollo especifico.
* El proceso de instalación es de código abierto.
* Ejecuta múltiples tareas.
* Si existe un error el sigue compilando.
* Minimiza el tiempo de ejecución en los diferentes compiladores para distintos sistemas operativos.

DESVENTAJAS:

* No es una herramienta que, al momento de su instalación, no se instala en el ordenador por si sola.

INSTALACION:

