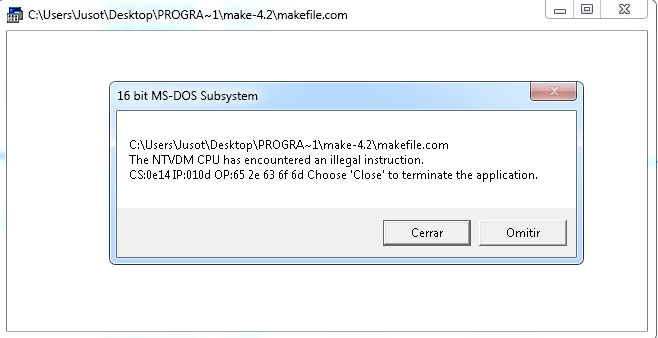
**Informe sobre diferentes Software de compilación Java.**

**Julian Giraldo Cardona | 1004752912 | Ing Sistemas | UTP**

**MAKE**

Fue imposible su ejecución ya que este programa fue desarrollado en los años antes del 2000 entonces Windows por políticas y por cuidarse del virus no permite la ejecución de comandos tan antiguos.

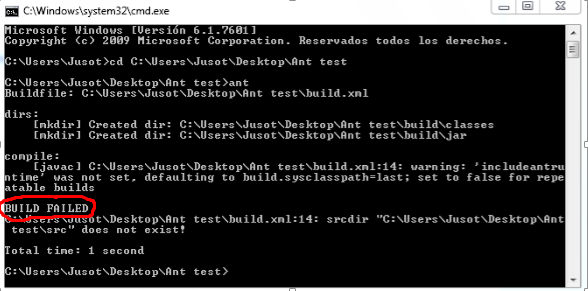


**ANT**

Manualmente creamos archivo build.xml para implantar el código ejecutable en la cmd.

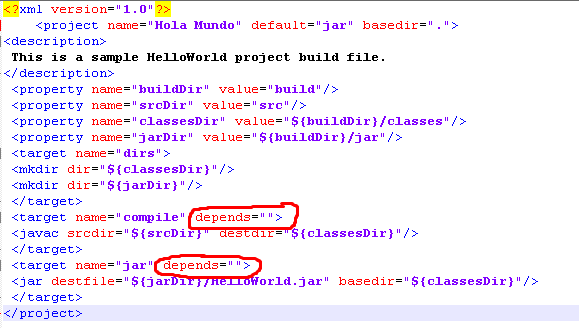


Luego al tener este archivo build.xml lo que se hizo fue agregar el directorio en el cmd y después ejecutarlo con “ant” para así crear las clases en su respectivo directorio.

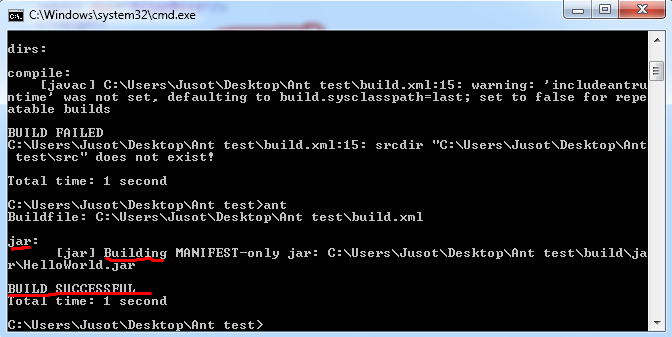


En esta versión de código aparecía un error que no permitía compilar los archivos de java por una dependencia de carpetas creadas, entonces lo que yo hice fue eliminar esta dependencia para poder ejecutar todo el código completo, se sabe por lógica que la dependencia es para verificar si están creadas las carpetas de Classes y java entonces podemos verificar esto en la ubicación local del archivo build.xml

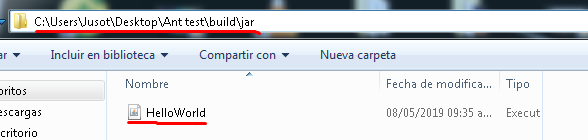
Al final después de eliminar estas dependencias el código en build.xml queda así.



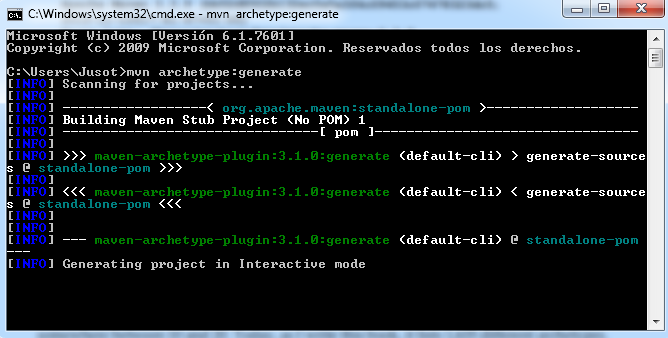
Después de quitar las dependencias, el código ejecutado queda así.

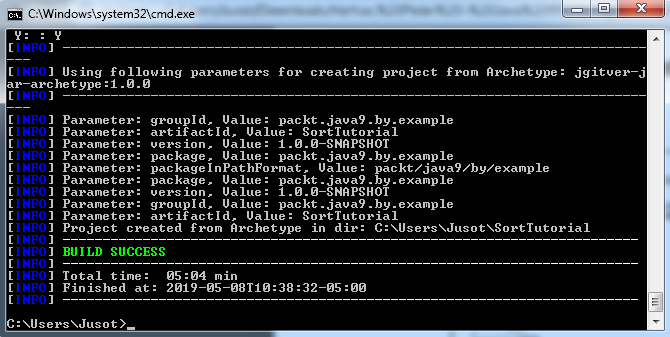


Aquí se crea el archivo ejecutable .jar que nosotros llamamos HelloWorld como lo indica el libro didáctico.



**MAVEN**

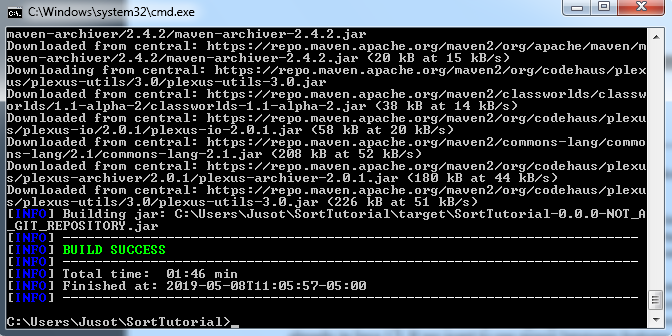


Después de haber configurado Maven con sus respectivas variables de entorno para su ejecución en la cmd procedemos a descargar los proyectos y seguiremos los pasos del libro. 

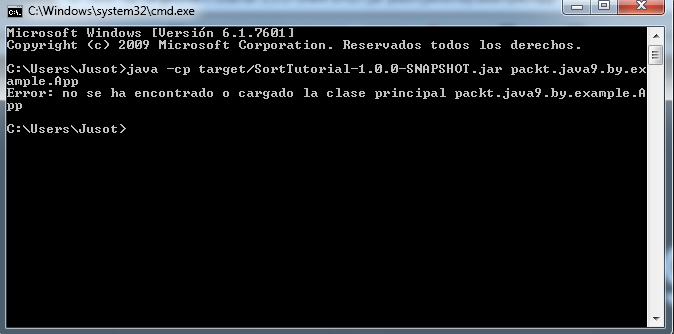
Luego de seguir los pasos previos, creamos el proyecto y después nos dirigimos al archivo generado .xml



Con esta ubicación del archivo lo que hacemos es crear la estructura del proyecto para así mostrar el ejemplo de programación utilizando hola mundo.



**ERROR**

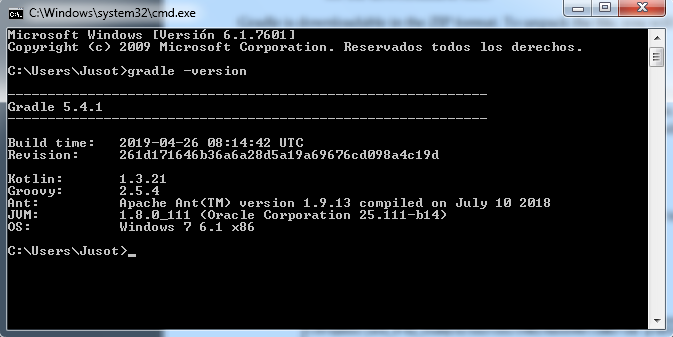


Este error puede ser debido a la diferente ubicación del archivo.

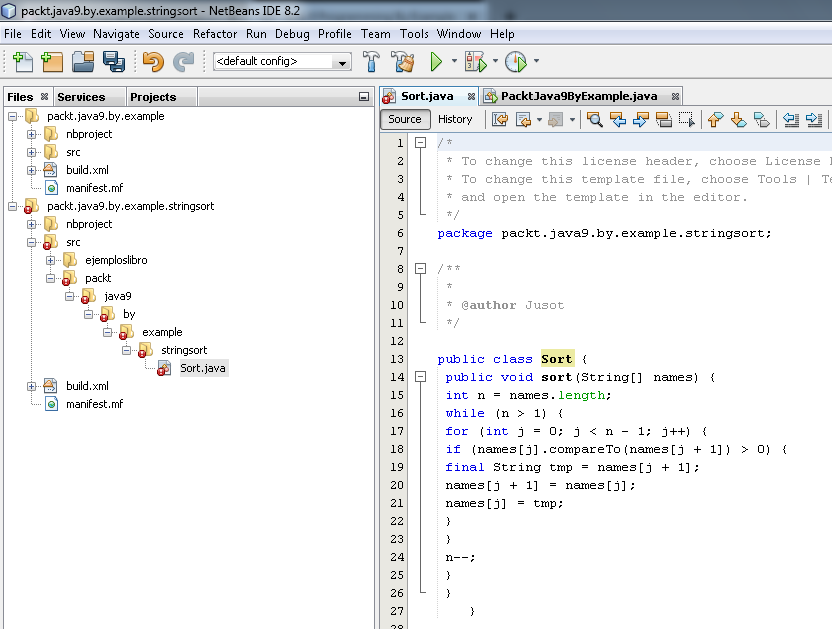
**GRADLE**

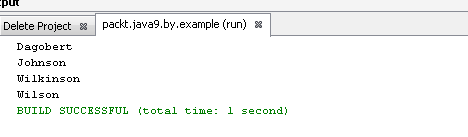
Se descarga gradle de la página oficial y luego se extrae el archivo zip obtenido.

Se crean rutas de environment para escribir algunas líneas de código en la cmd y para instalar Gradle.

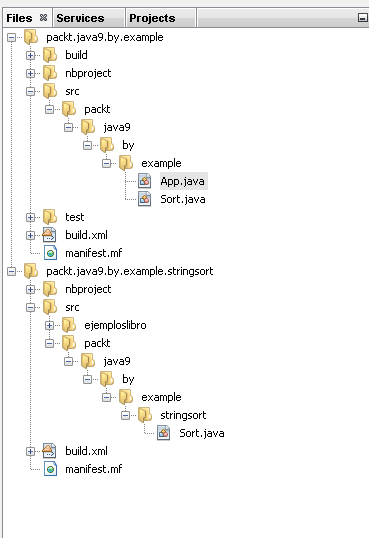


**CODING THE SORT**





En este código se utilizó el compilador IDE junto con los archivos de Maven, se crearon unas clases que hacen un llamado a cada carpeta donde estaba el código de Maven pasado.



|  |  |  |
| --- | --- | --- |
| **DIFERENCIAS** | | |
| **Ant** | **Maven** | **Gradle** |
| Más antigua respecto a las dos. | Más antigua que Gradle. | Más moderna. |
| Ant no tiene convenciones formales como una estructura de directorio de proyecto común o comportamiento predeterminado. | Maven tiene convenciones. Sabe dónde se encuentra el código fuente y siguió la convención de construcción de un proyecto. | Gradle aprendió de los errores de Maven, porque finalmente realiza muchas de las mismas tareas, tiene la misma finalidad, pero eliminaron la complejidad de Maven. |
| Se tiene que decirle a Ant exactamente dónde encontrar la fuente y dónde colocar la salida. | El plugin compilador de Maven pone las clases compiladas en un target, y produce un archivo JAR en un destino. | Gradle ofrece un bootstrap para que puedas ponerlo dentro de tu distribución de fuentes (en tu repo de git, por ejemplo) para que otros devs puedan construir tu proyecto sin tener que instalar Gradle. |
| Ant es procesal. Se tiene que decirle a Ant exactamente qué hacer y cuándo hacerlo. Se tiene que decirle indicarle el compilar, copiar y luego comprimir. | Maven es declarativo. Todo lo que se tiene que hacer es crear un archivo pom.xml y poner su fuente en un directorio predeterminado. Maven se encarga del resto. | Gradle tiene un daemon que ayuda a agilizar mucho el tiempo de construcción de proyectos porque es un proceso que ya está corriendo siempre y por tanto elimina el tiempo de arranque de la JVM. |

**CONCLUSIONES**

* El mejor software de estos 4 (aunque probé 3) es Gradle ya que es una versión moderna y es la mejora de Maven entonces es más cómodo su trabajo.
* A pesar de que las 4 distribuciones son un poco distintas estas son de Java y hay que crear un environment para trabajar en la cmd.
* Estos programas son más para desarrollar en Linux porque Windows es para personas del hogar y que no tienen que ver mucho con la programación, entonces también por eso se complica el trabajo en este equipo.
* La mejor opción para programar con Java en Windows es IDE.