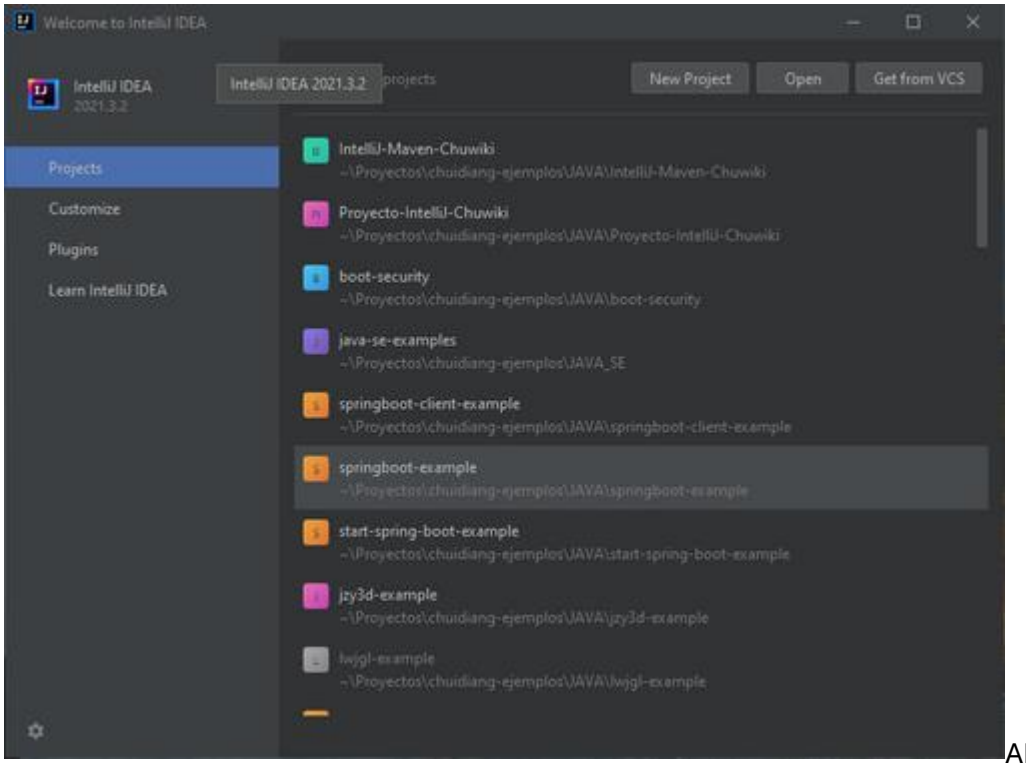
Crear proyecto maven en IntelliJ IDEA
Vamos a crear en este ejemplo un proyecto Maven con IntelliJ IDEA desde cero

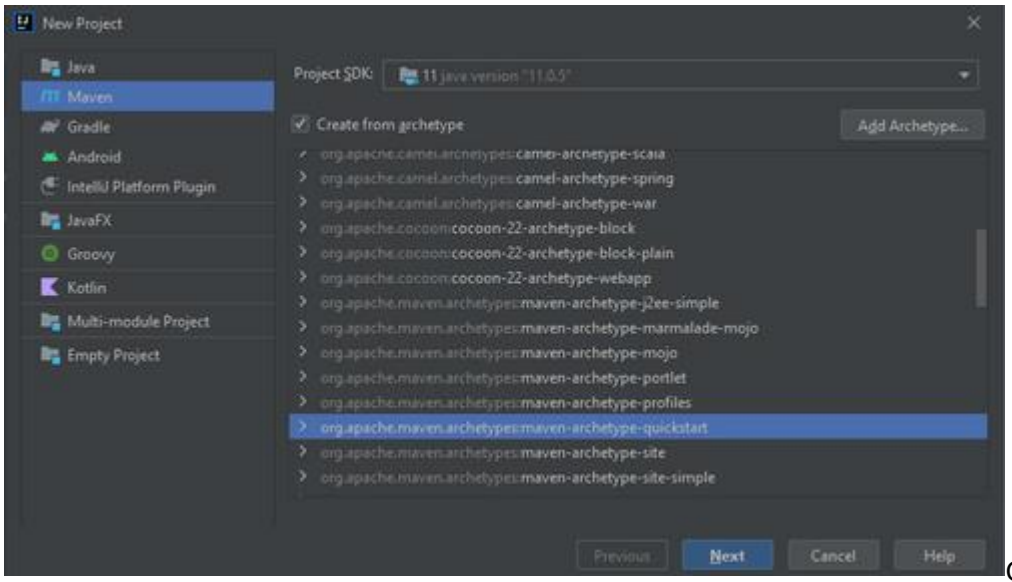
Creación del proyecto maven

Abrimos IntelliJ IDEA. Dependiendo si ya has trabajado con él, te saldrá el último proyecto en el que hayas trabajado abierto, o bien si no has trabajado nunca antes o cerraste el proyecto con el que habías estado trabajando, te saldrá una ventana como la siguiente



Abrir un proyecto Intellij existente

En el primer caso, "File"->"new project". En el caso que aparece en la imagen, simplemente botón "New Project". Debe aparecer una ventana como la siguiente



Creación de un proyecto nuevo con Intellij

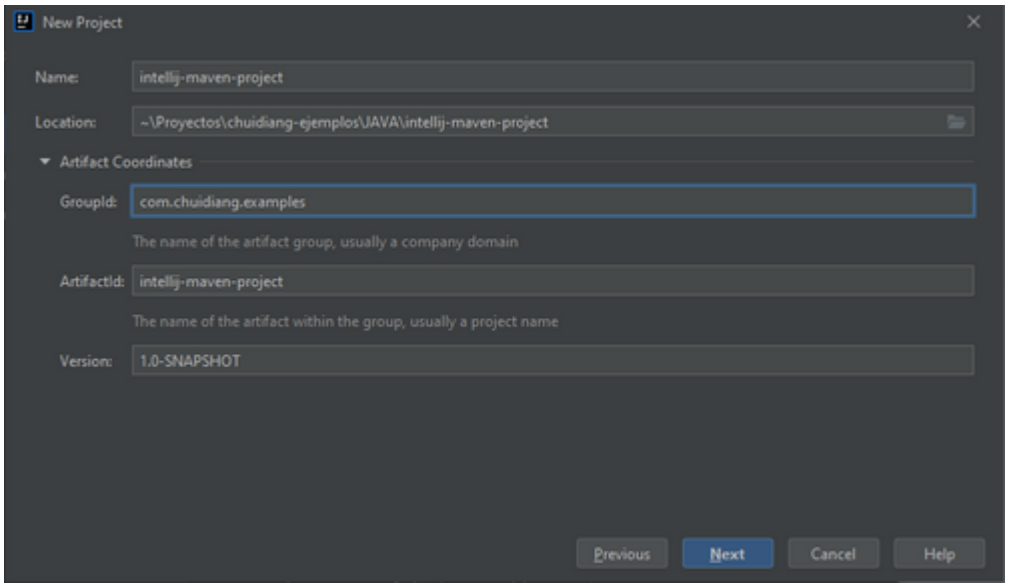
Seleccionamos Maven en la columna de la izquierda.

En "Project SDK" sale un combo con las versiones del SDK de java que IntelliJ IDEA sabe que tenemos en el ordenador. También da opción dicho combo a descargarse una versión nueva o buscar en nuestro disco una versión que ya tengamos instalada y que IntelliJ IDEA no sabe que la tenemos. En mi ejemplo, ya tengo instalada y selecciono la 11.

Podríamos dar ya al botón "next", pero podemos elegir también antes alguno de los "archetype" que maven pone a nuestra disposición. Un "archetype" es una plantilla para un tipo determinado de proyecto. Por ejemplo, hay "archetypes" para crear proyectos j2ee, proyectos portlet, etc, etc. Si no elegimos "archteype", se nos creará el proyecto vacío. Si elegimos uno de los "archetype" se nos rellenará automáticamente el proyecto con las dependencias necesarias y el código básico para empezar.

Como vamos a hacer un "hola mundo" simple y para ver el efecto de elegir un "archetype", marcamos el check "create from archetype" y en la lista de abajo buscamos y seleccionamos "maven-archetype-quickstart". Esta plantilla nos creará directamente una clase con el "Hola Mundo".

Damos al botón "next" y aparece la siguiente ventana



Parámetros del nuevo proyecto Maven en IntelliJ

Aquí rellenamos el nombre del proyecto, "intellij-maven-project" en mi ejemplo. Elegimos el directorio donde queremos crear dicho proyecto. Aparece una opción por defecto que podemos dejar si nos vale o bien cambiarla.

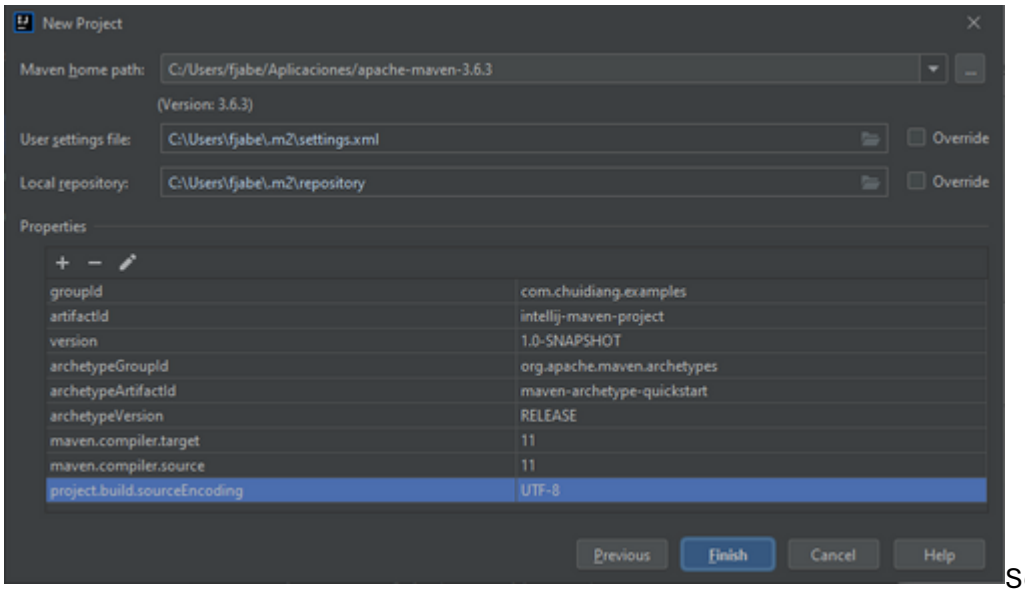
Desplegamos si no lo está la caja "artifact coordinates" y aparecen tres valores rellenos por defecto, pero que podemos modificar a nuestro gusto

- groupId: En maven los proyectos puede agruparse en grupos. Todos los proyectos que estén relacionados entre sí de alguna manera podemos ponerlos en el mismo grupo. Proyectos que no tengan nada que ver unos con otros podemos ponerlos en grupos distintos. Para evitar que desarrolladores distintos coincidan y pongan el mismo nombre de grupo, suele se habitual poner como nombre de grupo un dominio o algo muy específico del desarrollador o empresa, en mi ejemplo

"com.chuidiang" y luego alguna palabra más que identifique el grupo de proyectos, en mi caso "examples" es el grupo que pongo a casi todos mis proyectos de ejemplo de chuwiki. Podemos poner el grupo que queramos, únicamente hay que ser consciente que la plantilla maven creará las clases que cree en el package con el nombre del grupo, por lo que el nombre debe ser algo que también sea válido como paquete, es decir, no puede tener espacios o determinados caracteres.

- artifactId: En maven es el nombre del jar y del proyecto. Podemos poner lo que nos guste pero siendo conscientes que por defecto será el nombre del fichero jar que se genere cuando compilemos.
- version: Versión de nuestro proyecto. Puede ser lo que queramos que se parezca a un número de versión. Maven reconoce versiones terminadas en "-SNAPSHOT" como versiones que están en desarrollo, por lo que suele ser habitual terminar el número de versión con esta coletilla y dejarlo hasta que nuestro proyecto esté terminado, donde haríamos un último compilado quitando esta coletilla.

Pulsamos "next" y aparece la siguiente ventana



Selección la versión de Maven para nuestro proyecto Intellij

En "maven project home" elegimos la instalación de maven que tengamos en nuestro ordenador. Aquí aparecerán las que IntelliJ IDEA tenga conocimiento, pero podemos elegir el otro directorio donde tengamos instalada una y que IntelliJ IDEA no conozca, o bien elegir una que ya vienen embebida en IntelliJ IDEA.

El fichero settings.xml es el fichero de configuración de maven. Habitualmente viene uno predefinido en la instalación, podemos elegir ese si no sale ya por defecto. En mi caso, tengo uno copiado en C:\Users\fjabe\.m2\settings.xml con la configuración particular que a mí me interesa. fjabe es mi usuario de windows.

El local respository es donde maven dejará los jar que se baje de internet o donde dejará los jar de tus proyectos cuando le digas a maven que los instale. En mi caso están en C:\Users\fjabe\.m2\repository, que es el sitio por defecto.

En el panel de abajo aparecen una serie de properties. Todas ellas salen por defecto, excepto las 3 últimas que he añadido yo a mano. Se añaden pulsando el botón "+" que aparece justo debajo de la palabra "Properties". Esto abre una ventana que te pide el nombre de la propiedad y su valor. Las tres que he añadido son

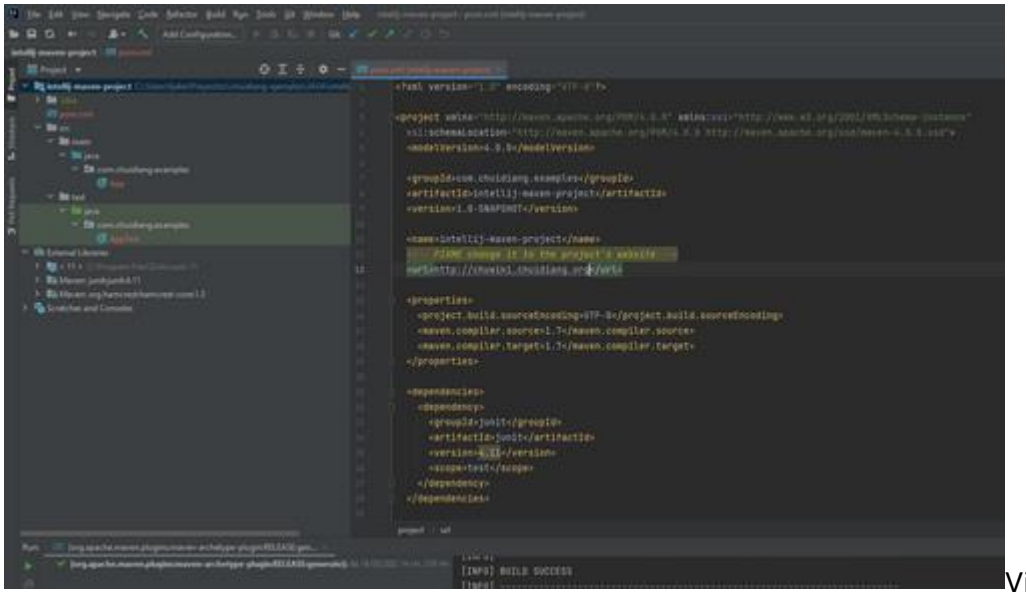
- maven.compiler.target -> 11. Esto indica a maven que genere jar de la versión 11. Aunque usemos un sdk de la 11, podríamos generar jar compatibles, por ejemplo, con la versión 1.8. Es una forma de asegurar que los jar que generamos luego se pueden ejecutar en versiones más antiguas de java.
- maven.compiler.source -> 11. Esto indica a maven que considere que los fuentes java pueden tener cosas específicas de la versión 11. Aunque usemos un sdk de la 11, podríamos decirle que use fuentes de la 1.8, de forma que si en esos fuentes metemos algo que no esté en la versión 1.8, el compilador de un error. Es una forma de garantizar que generamos fuentes compatibles con compiladores más antiguos.
- project.build.sourceEncoding -> UTF-8. Si vamos a usar caracteres extraños en nuestro código, por ejemplo, textos o comentarios con acentos, es importante tener claro que tipo de codificación se va a usar en los ficheros. Windows por defecto para español pone "CP1252". A mi no me gusta por que esa codificación es específica para español. Prefiero cambiarla a UTF-8, que es más internacional.

Pulsamos "Finish" y se crea el proyecto

Ver el proyecto creado

El proyecto se crea, pero debemos ser un poco pacientes y esperar que termine del todo.

Se abre el proyecto en IntelliJ IDEA y poco se van añadiendo cosas. Una vez termina, se ejecuta el "archetype" maven que hemos elegido y entonces se crean más cosas. Debemos esperar que todo el proceso termine para tenerlo todo completo. Deberíamos ver algo como lo de la siguiente imagen



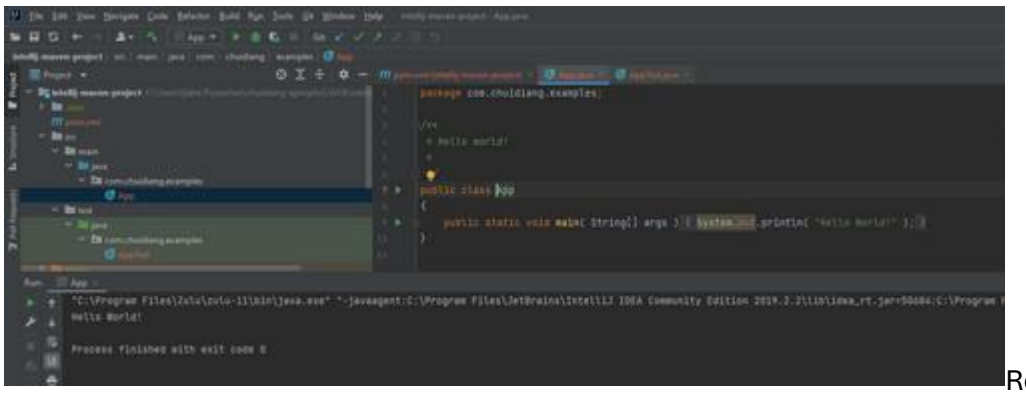
Vista del proyecto Maven recién creado en Intellij

En el árbol de la izquierda, vemos el nombre de nuestro proyecto con varias cosas debajo.

- .idea es un directorio específico de IntelliJ IDEA. Podemos ignorarlo.
- pom.xml es el fichero principal de maven para este proyecto. Es además el que ves en la imagen en el lado derecho. Un fichero xml con una serie de cosas. En mi caso he cambiado el valor del tag url por el de mi página web. Según vayas avanzando en tu proyecto es posible que tengas que tocar este fichero para añadir o modificar dependencias, añadir información adicional al proyecto, etc.
- src es el directorio donde estarán los fuentes. Lo desplegamos. Maven mete en src/main/java los fuentes del proyecto y en src/test/java los ficheros java que hacen de test, junit en concreto. Como hemos usado el "archetype" quickstart, se han creado dos ficheros. Un "App.java" con el código "Hola mundo" y un test "AppTest.java" que no hace nada, salvo dar OK.
- External Librerias contiene todas las librerías de las que dependemos:
    - El sdk 11
    - junit que es la librería para hacer test de junit, el "AppTest.java" que el "archetype" maven nos ha creado
    - hamcrest-core, una librería que necesita junit.
- Scratches and consoles. Un directorio donde podemos poner todo tipo de ficheros que nos sean útiles para pruebas nuestras, notas, pequeños main java que nos sirvan para verificar algo, etc. Si los ponemos aquí no formarán parte del proyecto.

Ejecutar el proyecto

Para ejecutar el proyecto, hacemos doble click en el fichero App.java para que se abra y veremos lo de la siguiente imagen



Resultado de ejecutar el proyecto Maven Hola Mundo en Intellij

Bastará con hacer click en uno de los triángulos verdes que aparece a la izquierda del código, bien en la línea del "public class", bien en la línea del "main()". Esto abre un menú con "Run App.main()", la seleccionamos y esto compilará y ejecutará el programa.

Ves en la imagen anterior, en la parte inferior, la salida de nuestro "Hello World"