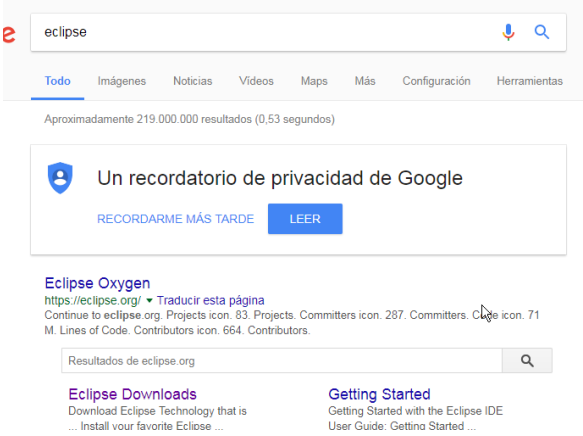
Mediante ECLIPSE realizaremos todos los desarrollos de los proyectos JAVA.

Uno de las grandes virtudes (y a la vez problema) de los desarrollos en JAVA es que se utilizan multitud de librerías externas, las cuales a su vez pueden llegar a tener dependencias de otras. Esto junto al problema del versionado puede ser muy complicado de gestionar.

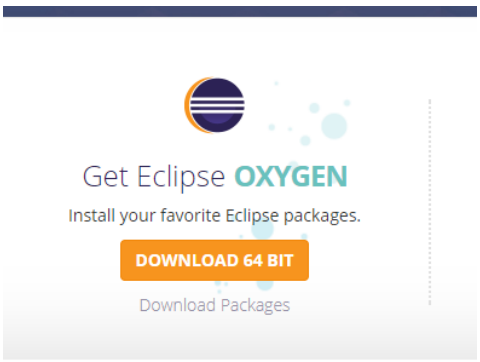
La solución es la utilización de herramientas de construcción para montar nuestras aplicaciones JAVA. Una de las primeras fue ANT, después apareció MAVEN, mucho más potente, y últimamente GRADLE, por su simplicidad y potencia, es de los más utilizados.

1) Instalamos la última versión de ECLIPSE, en nuestro caso ECLIPSE OXYGEN.

   Ponemos “eclipse” en el navegador..



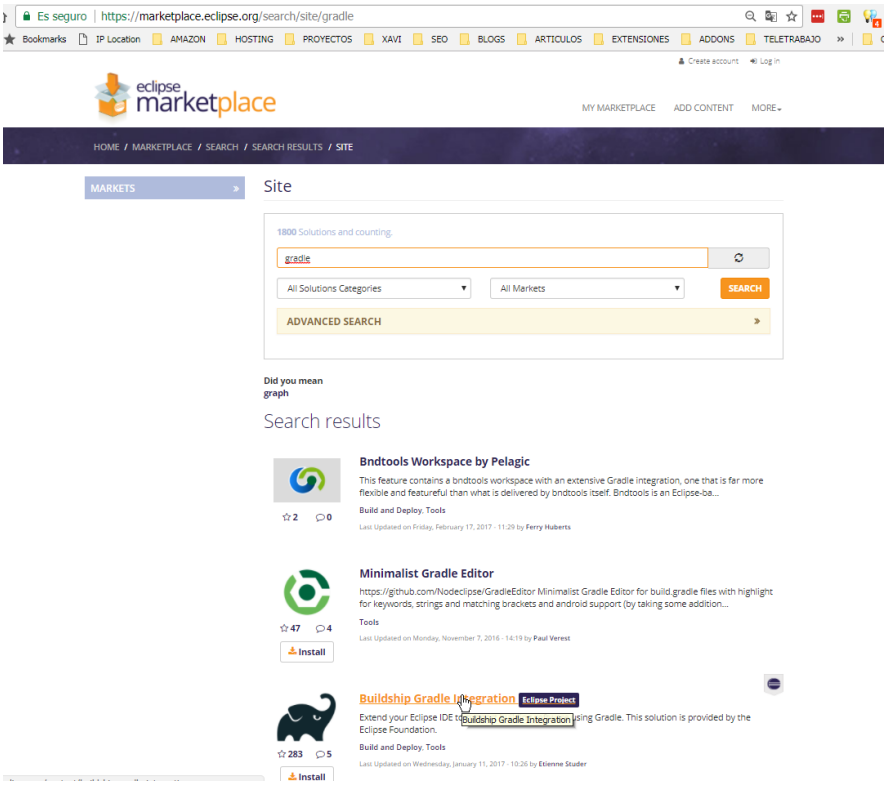
   Y a continuación nos lo bajamos..



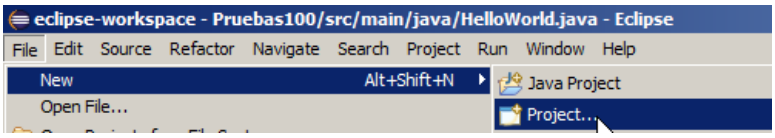
2) ECLIPSE tiene problemas en EMPRESA (firewall) con la bajada de datos desde los repositorios de MAVEN, para ello utilizaremos GRADLE que es mucho más sencillo y que vemos que funciona perfectamente.

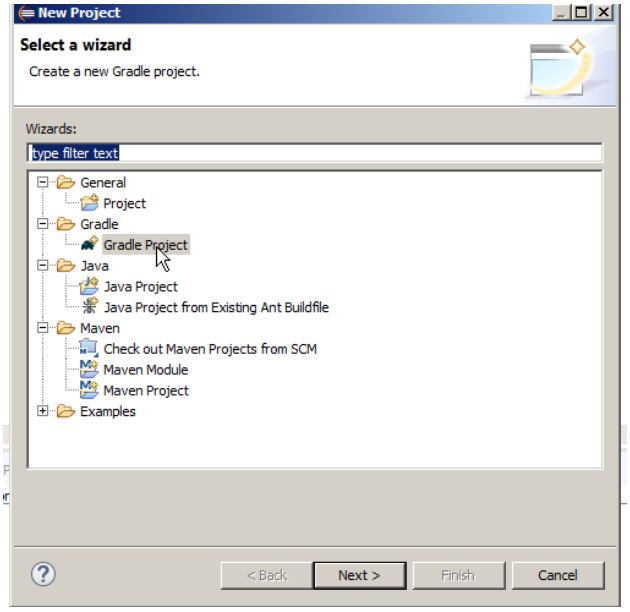
Así que antes que nada nos bajamos GRADLE. Este está como PLUGIN de eclipse.

Para ellos iremos al MARKETPLACE de ECLIPSE y mediante DRAG AND DROP contra la aplicación de escritorio lo instalaremos.



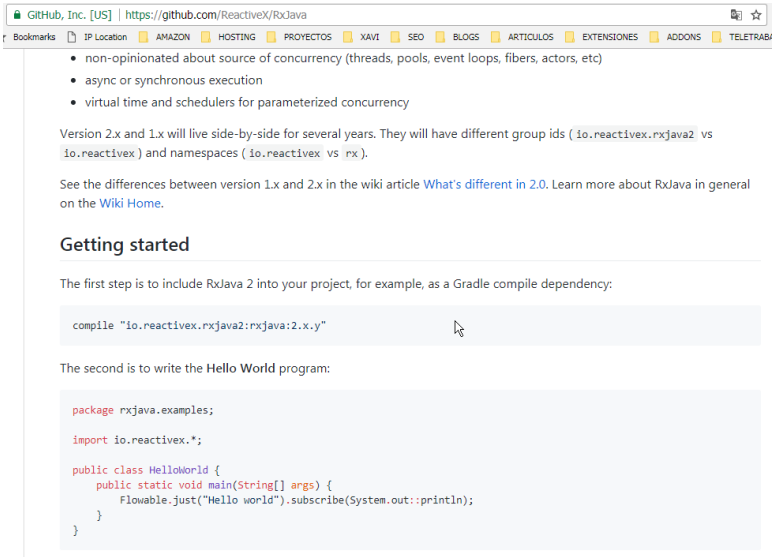
3) Una vez instalamos ya podemos hacer un “New Project” GRADLE.



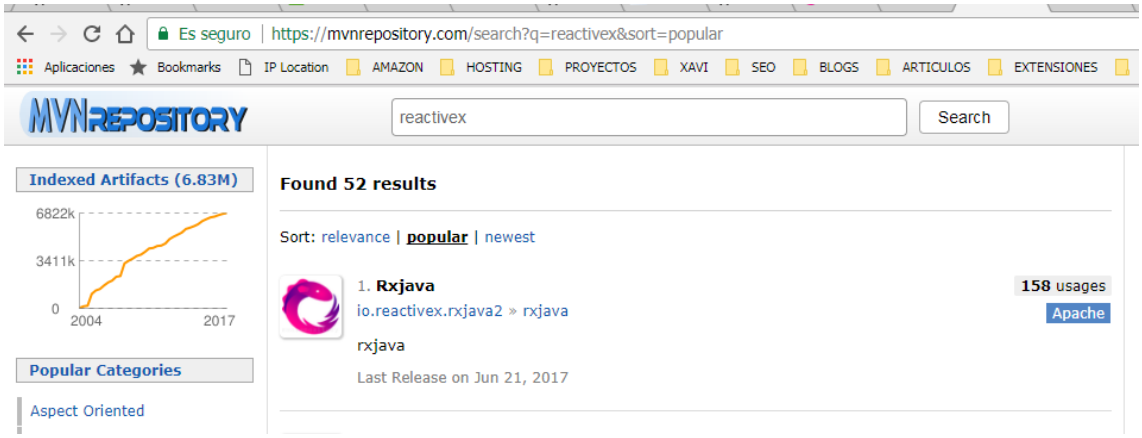


4) En nuestro caso queremos empezar a trabajar con la librería de REACTIVEX.IO para JAVARX. Vemos que en la documentación nos dice que está todo en GITHUB.

Donde además nos indica cual es el paquete a instalar en GRADLE y una simple clase de HELLOWORLD para que podamos ver que se ha instalado correctamente.



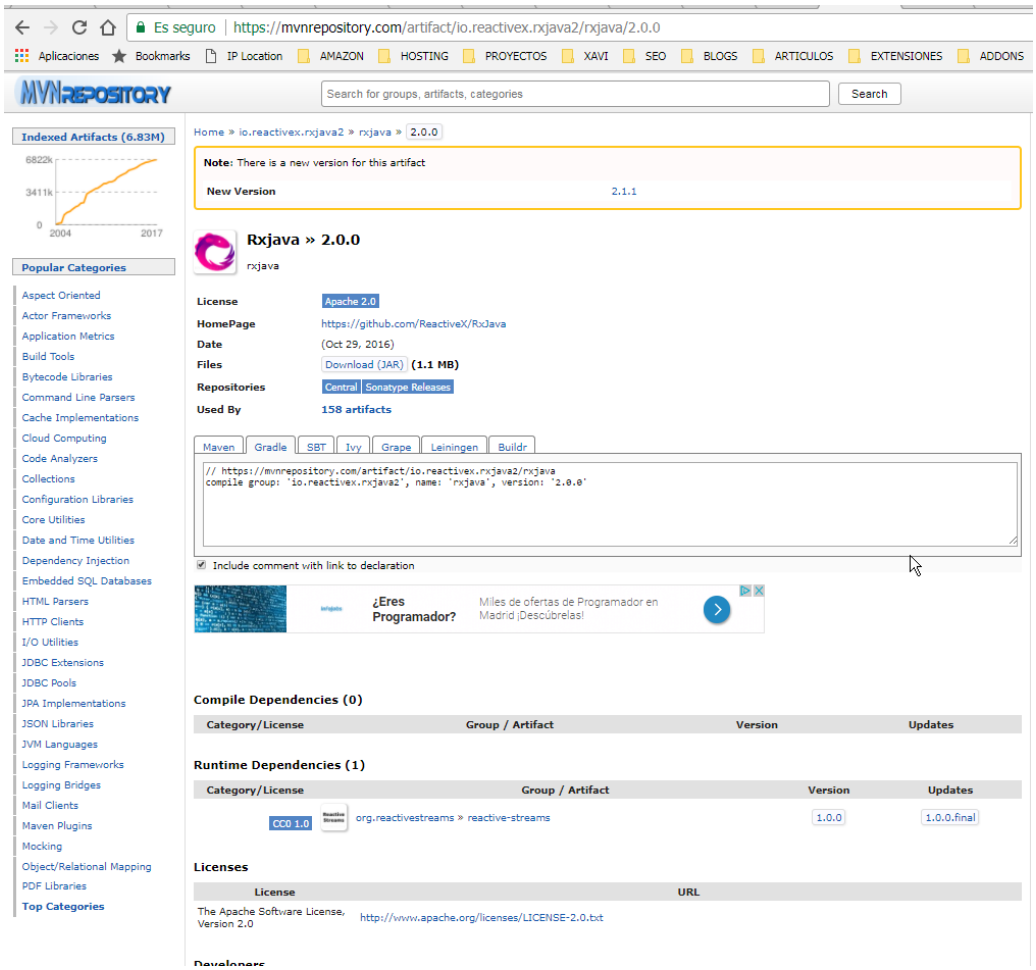
Si vamos al repositorio de MAVEN podemos ver que está aquí.



Podemos ver para la version 2.0.0 que nos indica que la ultima es la 2.1.1.

Donde están los fuentes en GITHUB y que sentencias hemos de poner en GRADLE (o cualquier otro de las herramientas constructoras) para que nos lo podamos bajar.

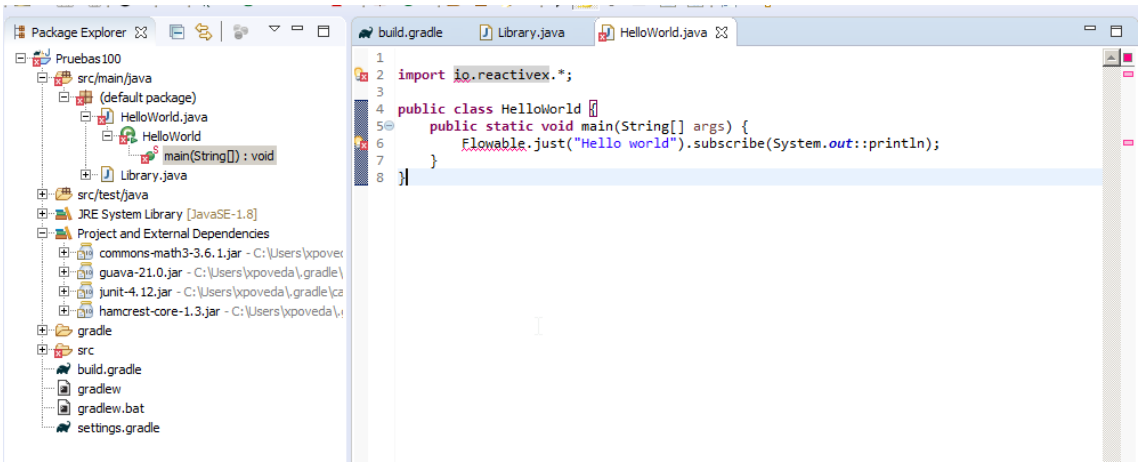
Aquí tambien nos podríamos bajar el “jar” de la librería y tambien se indican las librerias de las que depende (y que por lo tanto tambien necesitamos).



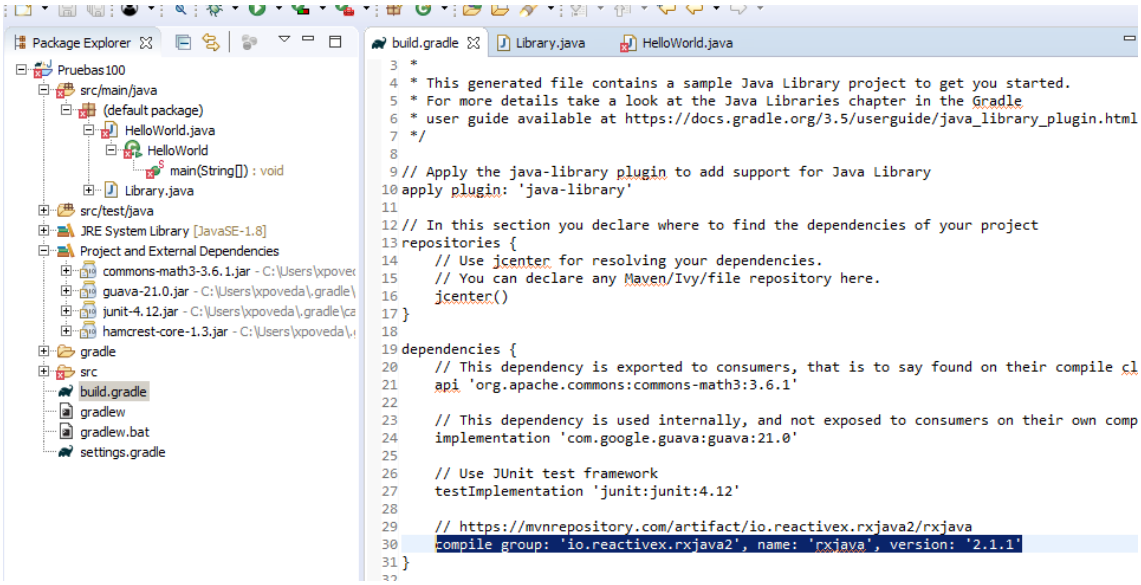
Aquí podemos ver el HELLOWORD que estamos intentando probar.

Como aun no nos hemos bajado las librerias da error en las llamadas a la librería y en el uso de una clase que forma parte de ella.

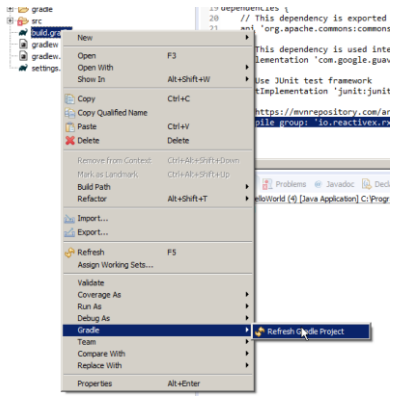
```
import io.reactivex.*;

public class HelloWorld {
    public static void main(String[] args) {
        Flowable.just("Hello world").subscribe(System.out::println);
    }
}
```



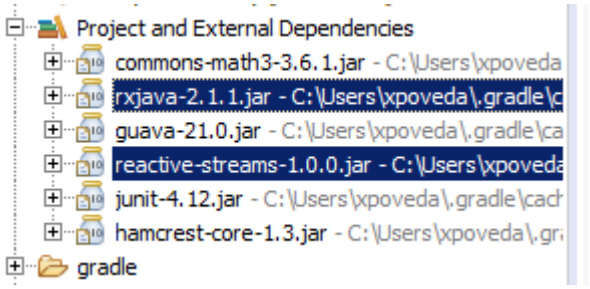
Ahora modificamos el build.gradle



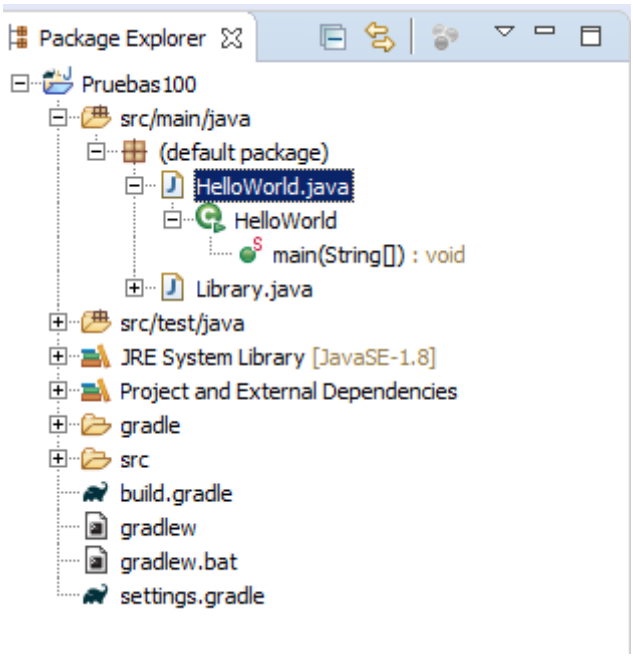
Y hacemos un refresh.



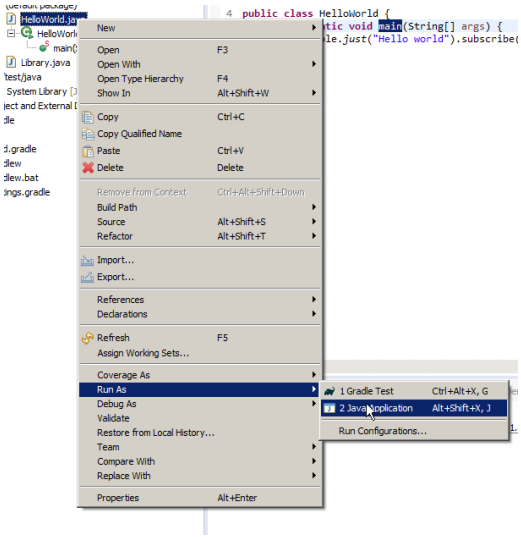
Con esto la lista de dependencias de nuestro proyecto se ha modificado.



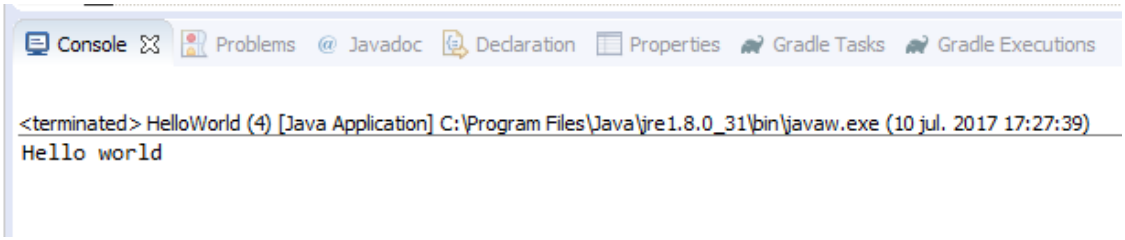
Han desaparecido las marcas rojas.



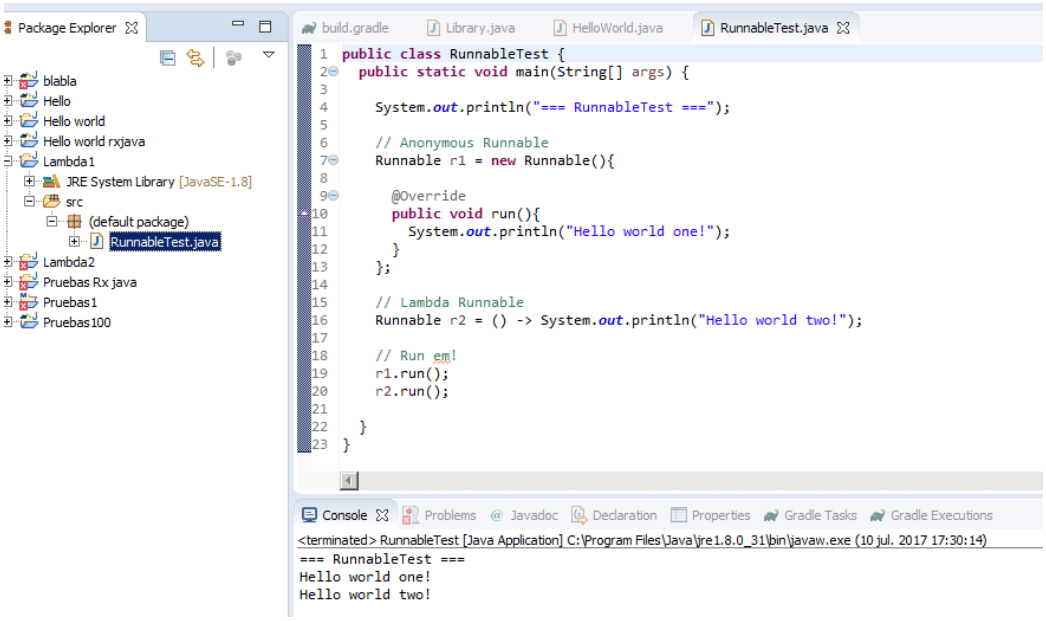
Y ya podemos ejecutar el método de la clase Main.



Como resultado:



Aquí podemos ver un ejemplo con expresiones lambda.



Y aquí otro que nos hemos bajado con unas pruebas automatizadas y generación con ANT.

http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/Lambda-QuickStart/index.html#section3