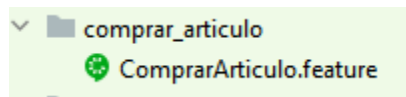
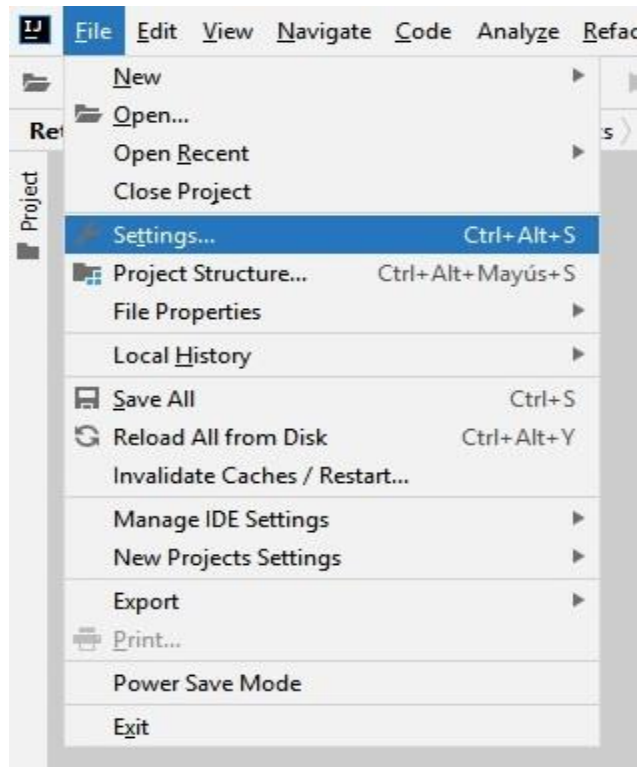


Instalación de Cucumber

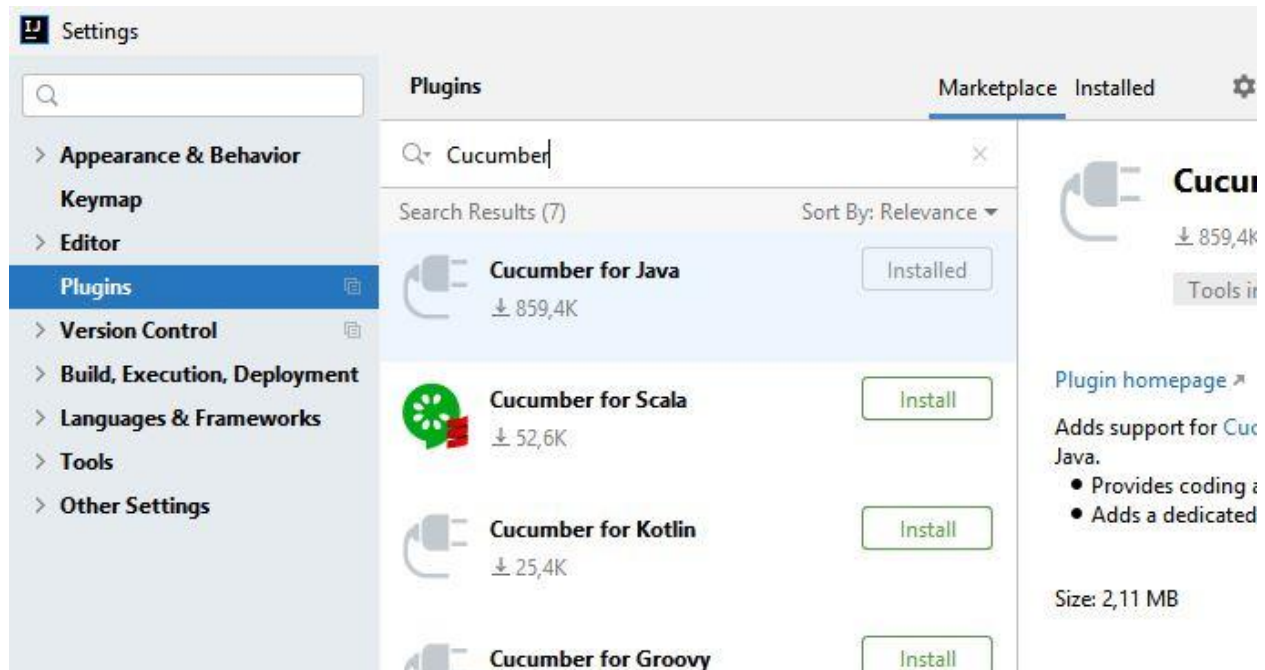
Se necesita descargar el Plugins para que pueda reconocer el archivo Feature cada que se indique el nombre de archivo con la extensión **.feature**



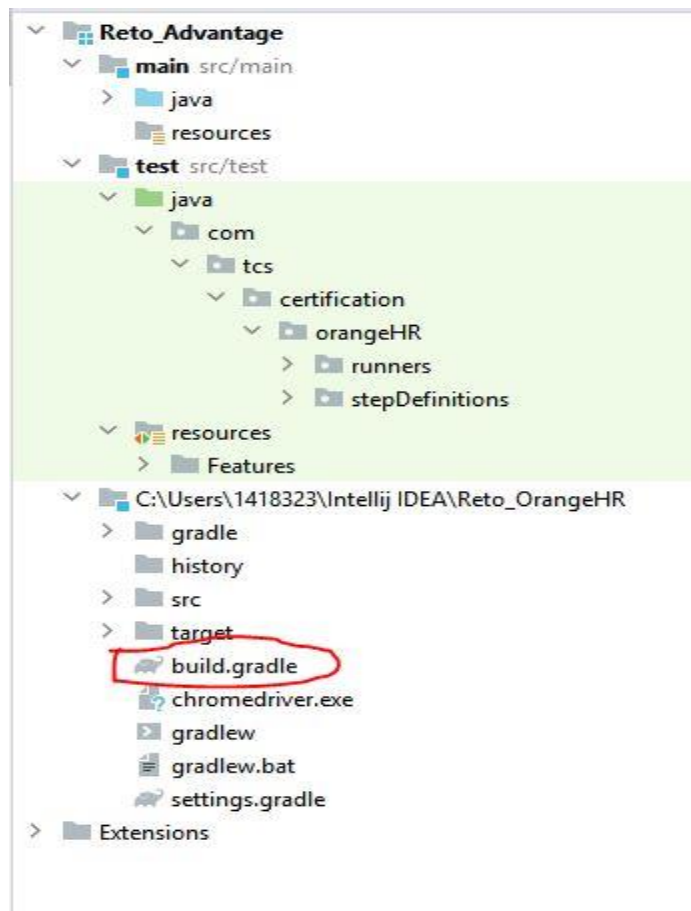
1. Ir a IntelliJ a la opción de Setting



2. Se dirige a la opción de Plugins, ahí les debe de aparecer todo lo que tienen instalado, de lo contrario van a Buscar en la opción Market Place y lo instalan dando clic en Install.



Configuracion de Build Gradle



build.gradle: Archivo de texto plano en el que especificamos los tipos de empaquetados y el producto que requerimos al compilar (jar, war, etc). También especificamos los repositorios y dependencias que se necesitarán al momento de compilar. Además, puedes definir propiedades del archivo.



```
1 plugins {
2     id 'java' // le indica que utilizaremos el complemento para compilar java
3 }
4
5 group 'org.example'
6 version '1.0-SNAPSHOT'
7
8 repositories { // le dice a gradle los repositorios donde buscara las librerias
9     jcenter() //le indica a gradle el repositorio donde ira a buscar las librerias necesarias para el proyecto
10 }
11
12 dependencies { /* le indicamos a Gradle las dependencias que necesitará nuestra aplicación para funcionar
13                 y que seran descargadas del repositorio indicado (jcenter())*/
14
15     testImplementation 'org.junit.jupiter:junit-jupiter-api:5.6.0'
16     testRuntimeOnly 'org.junit.jupiter:junit-jupiter-engine'
17     compile group: 'net.serenity-bdd', name: 'serenity-junit', version: '2.1.2'
18     compile group: 'net.serenity-bdd', name: 'serenity-cucumber', version: '1.9.51'
19 }
20
21 test {
22     useJUnitPlatform()
23 }
```

Entonces se crea un Proyecto de Build Gradle Nuevo como se ha hecho hasta ahora, nos dirigimos a la opción de build.gradle para configurar el archivo Build Gradle y vamos a borrar lo que tiene actualmente y luego vamos a copiar y pegar lo siguiente:

```
repositories {
    mavenLocal()
    jcenter()
}

buildscript {
    repositories {
        mavenLocal()
        jcenter()
    }
}

dependencies {
```

```

        classpath("net.serenity-bdd:serenity-gradle-plugin:2.1.2")
    }
}
apply plugin: 'java'
//    apply plugin: 'eclipse'
//    apply plugin: 'idea'
apply plugin: 'net.serenity-bdd.aggregator'

dependencies {
    implementation 'junit:junit:4.12'
    implementation 'junit:junit:4.12'
    testCompile 'net.serenity-bdd:serenity-core:2.1.2'
    testCompile 'net.serenity-bdd:serenity-junit:2.1.2'
    compile group: 'net.serenity-bdd', name: 'serenity-cucumber', version:
'1.9.51'
    compile group: 'net.serenity-bdd', name: 'serenity-screenplay-webdriver',
version: '2.1.2'
    compile group: 'net.serenity-bdd', name: 'serenity-screenplay', version:
'2.1.2'
    testCompile('junit:junit:4.12')
    compile group: 'net.serenity-bdd', name: 'serenity-core', version:
'2.1.2'
    compile group: 'net.serenity-bdd', name: 'serenity-junit', version:
'2.1.2'
    testCompile('org.slf4j:slf4j-simple:1.7.7')
    compile group: 'net.serenity-bdd', name: 'serenity-rest-assured',
version: '2.1.2'
    compile group: 'net.serenity-bdd', name: 'serenity-screenplay-rest',
version: '2.1.2'
    compile group: 'org.json', name: 'json', version: '20200518'
}
gradle.startParameter.continueOnFailure = true

```

Cerramos el IntelliJ y lo volvemos a abrir con el Proyecto que creamos en el cual le cambiamos la configuración al archivo Build Gradle para que se actualice y descargue las librerías y configuraciones indicadas para el funcionamiento de Cucumber con la creación del Feature, StepDefinitions y la ejecución con el Runner.