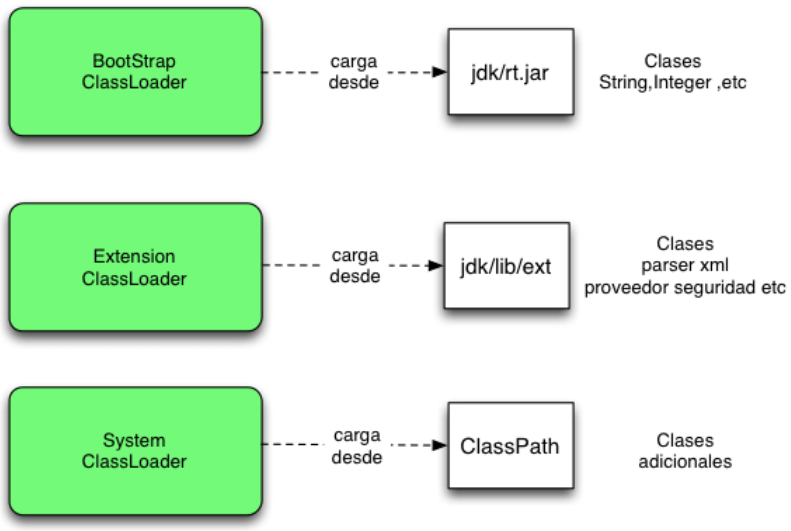
Este es uno de los conceptos de la plataforma Java menos entendidos por la gente .Vamos a dedicar algunos post a hablar del tema. En principio un Classloader es una clase encargada de cargar otras clases en memoria según la necesidad que tenga la maquina virtual java .

Tipos de ClassLoader

Cuando nosotros lanzamos un programa sencillo desde por ejemplo la consola nos encontramos con la siguiente estructura de cargadores de clase .



BootStrapClassLoader :Este es el cargador de clases principal encargado de cargar las clases core del JDK como por ejemplo String, Integer, Date etc. Estas clases están ubicadas en la libreria rt.jar dentro del JDK.

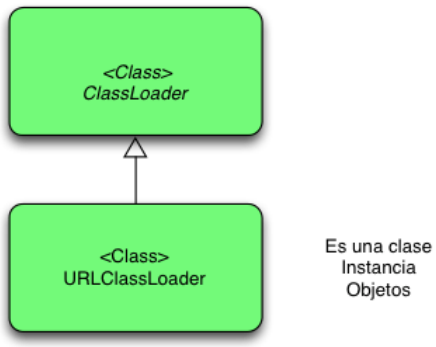
ExtensionClassLoader :Este es otro cargador de clases que se encarga de cargar las clases que se encuentran ubicadas en la carpeta jdk/lib/ext y hacen referencia a extensiones que necesita el JDK tipo proveedores de seguridad, parser XML etc .

SystemClassLoader: Este cargador de clases es uno de los mas habituales ya que se encarga

de cargar las clase que se encuentran dentro del classpath de la aplicación .Normalmente librerias que nosotros necesitamos tipo Hibernate,Spring etc.

Clases y Objetos

Muchas veces la gente me pregunta sobre si los cargadores de clase son clases u objetos. Vamos a detallar un poco el tema. Si revisamos un poco el JDK nos encontraremos con una estructura de clases como la siguiente (simplificada).

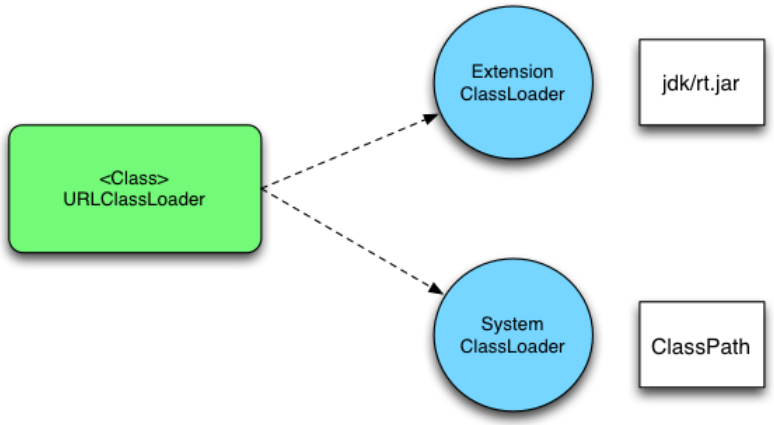


Lamentablemente no tenemos ni BootStrapClassLoader , ni ExtensionClassLoader ni SystemClassLoader . ¿Entonces como funciona esto realmente?. Vamos a ir paso a paso . En primer lugar es normal que no exista una clase BootStrapClassLoader ya que no esta definida en código java sino en código nativo es la que se encarga de cargar las clases principales Java.



Sin embargo nos quedan dos cargadores por encontrar. Eso se debe a que los ClassLoaders son tanto clases como objetos de la maquina virtual . En nuestro caso disponemos de una

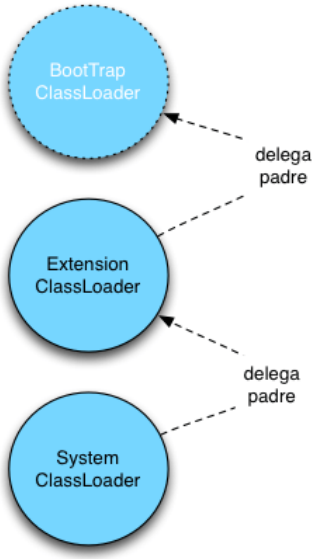
clase denominada URLClassLoader la cual será la encargada de instanciar dos objetos de tipo “URLClassLoader” el ExtensionClassLoader y el SystemClassLoader cada uno de los cuales apunta a URLs distintas.



De esta manera ya hemos solventado el problema de como la JVM gestiona cada uno de los ClassLoaders que disponemos al arrancar un programa.

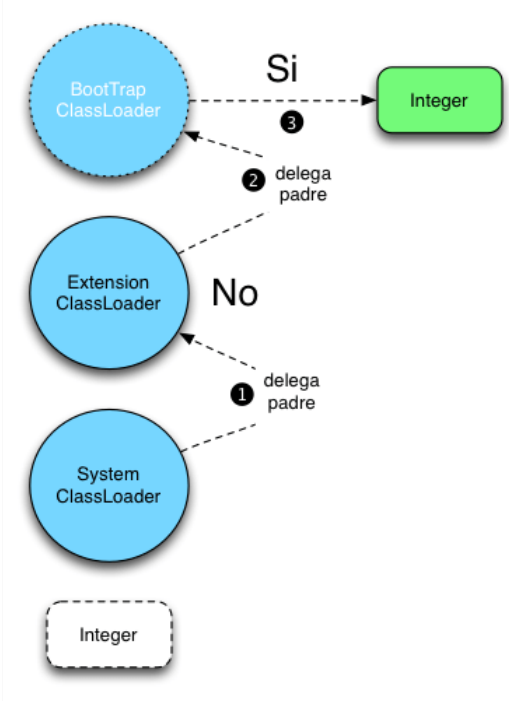
Relación de ClassLoaders

Otra de las cosas mas importantes a conocer a nivel de estos objetos ClassLoader es que están relacionados entre si con una relación padre/hijo.

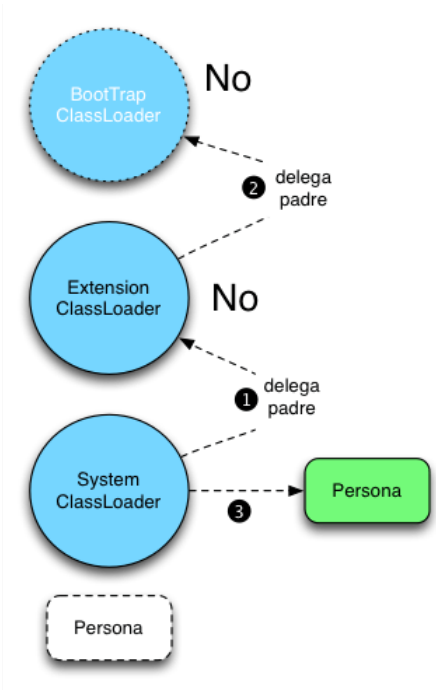


Esto implica que cuando un ClassLoader va a cargar una clase primero pregunta a su ClassLoader padre si el la tiene disponible en tal caso este la carga. En el caso de que el ClassLoader padre no la pueda cargar delega la petición en el siguiente padre hasta llegar al BootStrapClassLoader .Si ninguno de los ClassLoader padre puede cargar la clase , el cargador de clases actual intentará cargarla de sus rutas disponibles . Sino la encuentra se producirá un ClassNotFoundException.

Por ejemplo si solicitamos cargar la clase Integer sucederá lo siguiente:



En cambio si queremos cargar la clase persona del classpath



Bueno hemos hecho una breve introducción al concepto cargador de clases mas adelante cubriremos estos conceptos en entornos web.