# Mapa de herramientas

|  |  |  |
| --- | --- | --- |
| **Propósito** | **Nombre** | **Versión** |
| Entorno de desarrollo | Eclipse Luna |  |
| Máquina virtual | Oracle VM Virtual Box |  |
| Servidor web | Apache Tomcat | 7 |
| Control de versiones | Git |  |
| Repositorio | GitHub |  |
| Plugin para utilizar GitHub en eclipse | EGit |  |
| Servidor de base de datos | MySQL |  |
| Software | JDK | 8 |
| Gestión de incidencias | GitHub |  |
| Framework | Spring |  |
| Framework | Angula JS |  |
| Herramienta de mapeo objeto-relacional | Hibernate |  |
| Librería | Faster XML/Jackson core |  |

Como ya se ha hecho en apartados anteriores, a continuación procederemos a explicar cómo vamos a utilizar cada una de estas herramientas.

En primer lugar, hemos utilizado la máquina virtual y en ella hemos instalado todas las herramientas necesarias para construir nuestro proyecto. Para realizar la gestión del código se han utilizado las siguientes herramientas:

Git

Egit

GitHUb

Eclipse

Hemos creado un repositorio en GitHub para poder hacer uso del sistema de control de versiones de Git y tal y como ya se ha explicado en el punto 6 – Gestión del código fuente hemos utilizado la herramienta EGit para disponer de todas las funcionalidades de GitHub de una manera más sencilla. El segundo paso es, por tanto, la realización del código fuente, para lo que se han utilizado las siguientes herramientas:

Angular JS

Tomcat

Srping

Eclipse

Hibernate

JDK

Maven

MySQL

Hemos optado por utilizar el entorno de desarrollo Eclipse Luna ya que utilizaremos lenguaje JAVA y además permite la instalación y el uso de muchas de las herramientas necesarias para nuestro proyecto. Eclipse hace uso de JDK, que puesto que engloba JRE hemos decidido no hacer referencia a él en el diagrama. Este software nos proporciona herramientas para el desarrollo de programas java como un intérprete y un compilador.

Para crear el proyecto hemos utilizado Maven, el cual nos da la posibilidad de describir el proyecto a construir, sus dependencias con otros módulos y el orden en el que se deben de construir los elementos. Hemos podido hacer uso de Maven gracias al plugin para Eclipse M2E-WTP, el cual ya está incluido en la versión de Eclipse que hemos utilizado.

Hemos decidido seguir utilizando los framework Angular JS y Spring que ya se propusieron y utilizaron el año pasado para facilitar el uso del patrón Modelo-Vista-Controlador. Angula JS es un framework para aplicaciones web que amplía el HTML tradicional permitiendo que las aplicaciones sean más ligeras gracias a una serie de inyecciones de dependencias. Por otro lado Spring es otro framework para el desarrollo de aplicaciones Java que permite, entre otros muchos servicios, gestionar el flujo de información web, proporciona soporte con la capa de acceso a datos implementada con JPA, gestiona el control de acceso y autentificación y se encarga de los detalles de bajo nivel.

Hemos utilizado Apache como servidor web junto con el contenedor web Tomcat. A pesar de que ambos pueden utilizarse por separado como servidores web nosotros hemos decido utilizarlos en conjunto para aprovechar sus ventajas y sobretodo porque hemos trabajado con ellos en otras asignaturas.

Por otro lado hemos utilizado MySQL como servidor de Base de Datos junto con el conector MySQL 5.1 para que nuestra aplicación se conecte al dicho servidor. Junto a esta herramienta hemos utilizado Hibernate que permite transformar las entidades de la base de datos en objetos JAVA para utilizarlos en nuestro proyecto, tal y como ya se había indicado en el punto 5 – Entorno de desarrollo.

El tercer y último objetivo, y del cual ya se ha hablado de manera más detallada en el punto 8 – Gestión del cambio, incidencias y depuración, es la gestión de incidencias. Para este proceso hemos hecho uso de los mecanismos que nos provee GitHub. Esta herramienta nos permite crear nuevas incidencias o “issues”, asignarlas a los miembros del equipo y cerrarlas una vez solucionadas.

Cabe destacar que puesto que es necesaria la integración con otros subsitemas, se ha utilizado la librería Jackson para proporcionar un JSON con el que poder utilizar nuestros servicios (Faster XML/Jackson core).