

# INDIVIDUAL REPORTS - E3.03

**Nombre:** José Antonio Márquez López

**DNI:** 80236202N

**Correo:** [josmarlop16@alum.us.es](mailto:josmarlop16@alum.us.es)

## Índice

<b>INDIVIDUAL REPORTS - E3.03</b>	<b>1</b>
<b>Development configuration</b>	<b>2</b>
Introducción	2
Java	2
Firefox & Gecko driver	2
Maven	3
Maria DB & DBeaver	3
Eclipse & Lombok	4
Anexo	4
<b>Describing a WIS</b>	<b>5</b>
<b>Testing a WIS</b>	<b>5</b>

# Development configuration

## Introducción

Para el desarrollo de la asignatura “Diseño y Pruebas 2” es necesario tener un entorno de trabajo determinado para que funcione correctamente el proyecto base con el que trabajaremos a lo largo de la asignatura.

## Java

Lo primero que necesitábamos instalar es “Java jdk 8 Update 281” que será el lenguaje de programación en el que está el proyecto base. Para la comprobación de la correcta instalación es necesario ejecutar los comandos “java -version”.

```
[(base) joseantonio@MacBook-Pro-de-Jose ~ % java -version  
java version "1.8.0_281"  
Java(TM) SE Runtime Environment (build 1.8.0_281-b09)  
Java HotSpot(TM) 64-Bit Server VM (build 25.281-b09, mixed mode)  
(base) joseantonio@MacBook-Pro-de-Jose ~ %
```

## Firefox & Gecko driver

Para realizar las pruebas necesitamos instalar el navegador Firefox y , gracias a Gecko Driver podremos automatizarlas. Para comprobar su correcta instalación debemos escribir los comandos “firefox --version | more” y “geckodriver --version”

```
[(base) joseantonio@MacBook-Pro-de-Jose ~ % firefox -version | more  
Mozilla Firefox 96.0  
[(base) joseantonio@MacBook-Pro-de-Jose ~ % geckodriver -version  
geckodriver: error: Found argument '-e' which wasn't expected, or isn't valid in  
this context  
  
USAGE:  
    geckodriver -v  
  
For more information try --help  
[(base) joseantonio@MacBook-Pro-de-Jose ~ % geckodriver --version  
geckodriver 0.30.0  
  
The source code of this program is available from  
testing/geckodriver in https://hg.mozilla.org/mozilla-central.  
  
This program is subject to the terms of the Mozilla Public License 2.0.  
You can obtain a copy of the license at https://mozilla.org/MPL/2.0/.  
(base) joseantonio@MacBook-Pro-de-Jose ~ %
```

## Maven

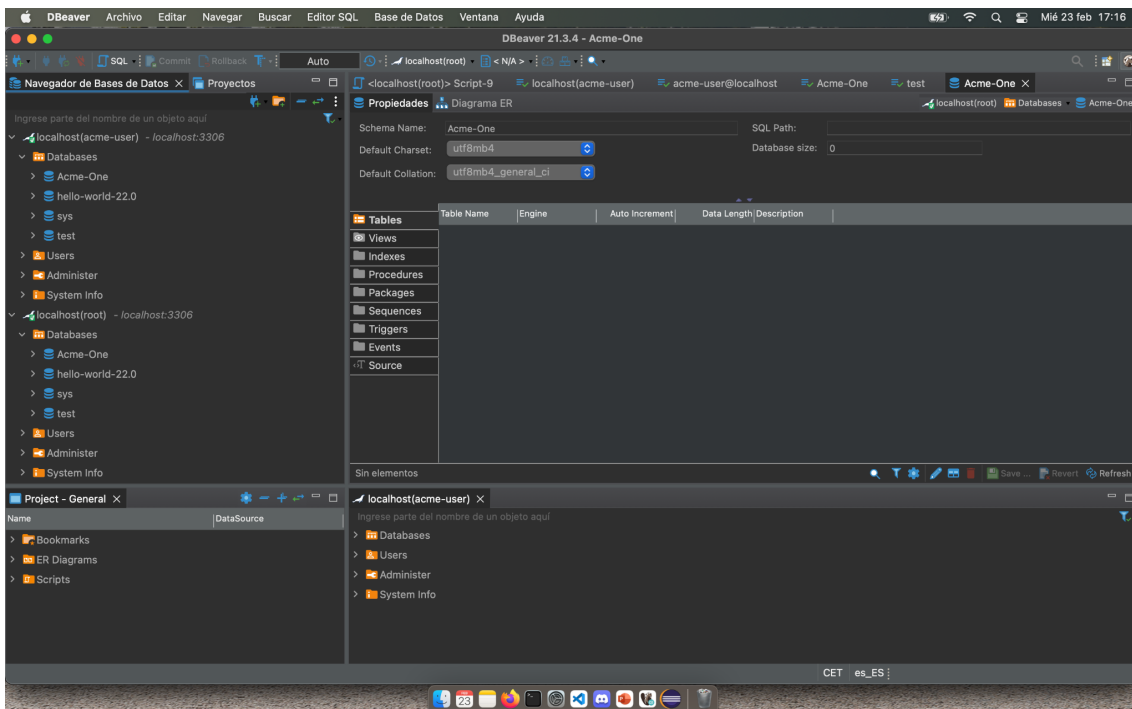
Para administrar nuestros componentes de software y empaquetar nuestros proyectos, trabajaremos con Maven. Es una herramienta de administración de compilaciones. Comprobamos su correcta instalación ejecutando el comando “mvn -version”.

En mi caso, solo tengo instalado maven en el directorio que voy a usar para la asignatura, para no tener conflictos con posibles proyectos futuros

```
[(base) joseantonio@MacBook-Pro-de-Jose Workspace-22.0 % mvn -version]
Apache Maven 3.8.4 (9b656c72d54e5bacbed989b64718c159fe39b537)
Maven home: /usr/local/Cellar/maven/3.8.4/libexec
Java version: 1.8.0_281, vendor: Oracle Corporation, runtime: /Library/Java/Java
VirtualMachines/jdk1.8.0_281.jdk/Contents/Home/jre
Default locale: es_ES, platform encoding: UTF-8
OS name: "mac os x", version: "10.16", arch: "x86_64", family: "mac"
```

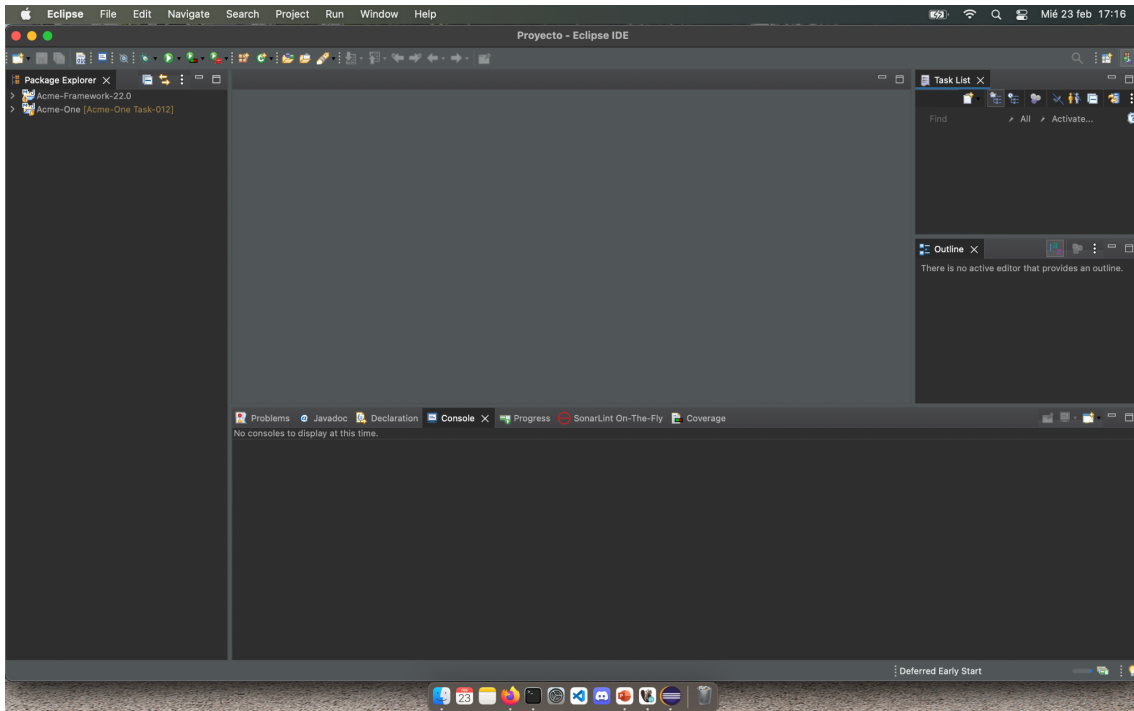
## Maria DB & DBeaver

Usaremos MariaDB, que es el servidor de la base de datos. Como herramienta de gestión de base de datos usaremos DBeaver. En mi caso es una herramienta muy familiar, pues durante IISSI 1 e IISSI 2 tuve que usarla, al no poder usar HeidiSQL que era la que nos requerían en ese momento. La instalación y puesta a punto fue muy sencilla por lo mencionado anteriormente. Para que ambas aplicaciones funcionen es necesario primero ejecutar MariaDB y posteriormente DBeaver. Para comprobar que la conexión se ha realizado correctamente en DBeaver se debe mostrar un tick verde sobre la conexión creada (en el ejemplo localhost:3306).



## Eclipse & Lombok

Para escribir y depurar nuestros proyectos, la herramienta utilizada será Eclipse. Entorno de desarrollo integrado (IDE). Además, gracias a Lombok, conseguimos simplificar la escritura de código Java, y varios complementos que simplifican el desarrollo web. Para que Eclipse funcione correctamente es necesario usar el workspace proporcionado en la asignatura, el java jdk que se nos indica en las transparencias e importar el archivo de preferencias "Eclipse-Preferences.epf" que lo encontramos también en la carpeta que nos proporciona la asignatura. Al igual que DBeaver, es una herramienta que estoy acostumbrado a usar en anteriores asignaturas.



## Anexo

Todas las herramientas que utilizaremos en esta asignatura las hemos instalado a partir de la carpeta "Tools" que viene en el fichero "Workspace-22.0" descargado desde la página web de la asignatura. Yo al ser el único usuario de MacOS de mi grupo, tuve que indagar un poco en determinados archivos para crear los launchers mientras se subía la reciente guía de instalación para MacOS.

## Describing a WIS

Para mí, un WIS o Web Information System (Sistema de Información Web) es un conjunto de elementos que interactúan entre sí con un objetivo común, que puede ser desde recolectar, almacenar y manipular información hasta ser un elemento de difusión de información, por ejemplo.

## Testing a WIS

La importancia de los test para un sistema de información web es muy importante, debido a que los test comprueban todas las funcionalidades del sistema, lo que nos permite conforme realizamos actualizaciones de nuestro sistema, determinar y prevenir que posibles fallos se introduzcan en producción, así como ahorrar costes, ya que es más fácil y barato solucionar un error o un problema en fases tempranas de desarrollo, que una vez ya está el producto lanzado. Por ello es muy interesante, y casi obligatorio hoy en día, hacer uso de mecanismos de automatización de tests y despliegue, y así evitar que se actualice nuestra versión en producción con errores.

Existen muchos tipos de test entre los más comunes son los test unitarios, que prueban de manera muy aislada una determinada funcionalidad, y los test E2E (end to end) que prueban el sistema web de manera más completa, y normalmente comprueban varias historias de usuario juntas, como por ejemplo: un usuario sube un video a la web, lee los diversos comentarios de otros usuarios, o directamente interactúa con ellos mediante un sistema de comunicación. Todo eso, desde el punto de vista del desarrollador, es testeable, lo que le permitirá identificar distintos errores rápidamente y corregirlos en el menor tiempo posible.