Workgroup C1.067

Set up of the development configuration

20/02/2025

Aarón Jesús Acuña Bellido - [aaracubel@alum.us.es](mailto:aaracubel@alum.us.es)

Javier Arellano López – [javarelop@alum.us.es](mailto:javarelop@alum.us.es)

Raúl Calero Capote – [raucalcap@alum.us.es](mailto:raucalcap@alum.us.es)

Tudor Cristian Lacatus Cosma [-ionlac@alum.us.es](mailto:-ionlac@alum.us.es)

Miguel Prado Jiménez – [migprajim@alum.us.es](mailto:migprajim@alum.us.es)

Table of Contents

[INTRODUCTION 4](#_Toc190971988)

[DEVELOPMENT CONFIGURATION PROCESS 4](#_Toc190971989)

[Workspace 4](#_Toc190971990)

[The Java platform 4](#_Toc190971991)

[Firefox and Gecko Driver 4](#_Toc190971992)

[MariaDB and DBeaver 5](#_Toc190971993)

[Eclipse, Lombok, and plug-ins 5](#_Toc190971994)

[CONCLUSIONS 5](#_Toc190971995)

[BIBLIOGRAPHY 6](#_Toc190971996)

Executive Summary:

This report presents the set up of the development configuration that we will use throughout the entire project. This is not a guideline, but an explanation of the tools used. The programming language selected is Java, meanwhile the formal testing browser chosen is Firefox Developer Edition. In order to work programmatically with the browser we use Gecko driver. For the database we use as tools, Maria DB and DBeaver, whereas the integrated development environment used to work with the code is Eclipse. Lombok, Sonar’s Lint, CSVEdit and other plug-ins are used to simplify development. This report explains in more detail each of the tools used and how they were set up, preparing the necessary configuration in order to work in the project.

Revision Table

|  |  |  |
| --- | --- | --- |
| Revision Number | Date | Description |
| 1.0 | 20/02/2025 | First description of the development configuration |
| 1.1 | 20/02/2025 | Justify content and eliminate blank spaces |
|  |  |  |

# INTRODUCTION

This report is intended to provide an explanation on how we have set up the development configuration. As a group, all of us have followed the same steps in order to have the same working environment so tools do not conflict with each other. This configuration was carried out by all team members following a well described guideline and with the support of a workspace given to us. The tools we needed to set up cover the programming language, the browser used for formal testing (as the project is WIS, we need the web in order to see how it looks like), the database system, and the IDE with the respective plug-ins which facilitate work.

# DEVELOPMENT CONFIGURATION PROCESS

## Workspace

With the objective of helping us set up a functional environment, we were provided with a workspace in which we had already downloaded everything we may need in order to carry out this project.

With this prior help, we only needed to configure the following tools in order to set up everything.

## The Java platform

Java is the programming language selected in order to develop and run the application, and the version selected for our work is *Java 21.0.4*. In order to use it, the environment variables of our computers were updated correctly including the directory where this version is stored

## Firefox and Gecko Driver

In order to fulfill the necessity of formal testing of our WIS, we have installed Firefox developer’s version. It is important to remark that this version can co-exits with the end-user one. Gecko driver is a web browser engine that works as a link between our tests and the Firefox browser. It allows us to use it programmatically. The version we use is *geckodriver-0.35.0*. As we did before with Java, for set up we included both tools in our environment variables

## MariaDB and DBeaver

MariaDB is a supported fork of the MySQL relational database management system. Our project data is stored in this database server. In order to execute the SQL scripts and administrate the database, we use DBeaver (version 24.3.1). For the server to be available and running, some commands must be executed in the terminal, first a setup and afterwards a start. Once the server has already been setup, it can be stopped or started easily also from the command line.

DBeaver didn’t require any previous installation, it came in the workspace mentioned above. The database connection to the MariaDB server is created using DBeaver, entering the credentials provided by our teacher. The connection needs an installation of MariaDB’s driver. This connection is the root one, so in order not to work always in it, we also created a new connection renamed as user conn.

## Eclipse, Lombok, and plug-ins

For writing, compiling and debugging our code, we use Eclipse IDE for Java EE, the difference with the classic Java IDE is that the first one comes with plugins preinstalled which makes it ideal for working with application servers. The version used is *eclipse-jee-2024-12R*. To simplify development we also use Lombok (version 1.18.36).

After launching eclipse, some changes must be done before working. We selected the preferences we needed: the plugins we installed in eclipse are CSVEdit (version 2.0.0), for viewing and editing csv files, and Sonar’s Lint (version 10.6.0), whose functionality is helping to write clean code. Then we configured and selected the JDK adding the “*-ea”* option to active the “assert” statement, and we set the compiler compliance level to 21.

Finally to install Lombok, we just launched the installer and selected the location of Eclipse in the workspace

# CONCLUSIONS

To sum up, the tools we have used to set up our development configuration are: Java as programming language, Firefox and Gecko Driver for formal testing on the web, MariaDB and DBeaver for database management, Eclipse as IDE and some plugins to facilitate the work.

# BIBLIOGRAPHY

We obtain the information from the material provided to us by the subject DP2

<https://ev.us.es/ultra/courses/_89154_1/cl/outline>