

# Development Configuration Set-up Report



**Acme AirNav Solutions, Inc**

Group Number: C1.066

Repository: <https://github.com/mquirosq/DP2-C1.066>

## Members:

María Quirós Quiroga, [marquiqui@alum.us.es](mailto:marquiqui@alum.us.es)

Guillermo Rodríguez Narbona, [guirodnar@alum.us.es](mailto:guirodnar@alum.us.es)

Ignacio Mora Pérez, [ignmorper1@alum.us.es](mailto:ignmorper1@alum.us.es)

Daniel Herrera Urbano, [danherurb@alum.us.es](mailto:danherurb@alum.us.es)

Alejandro Parody Quirós, [aleparqui@alum.us.es](mailto:aleparqui@alum.us.es)

February 19, 2025

# Contents

<b>Executive Summary</b>	<b>2</b>
<b>Revision History</b>	<b>3</b>
<b>1 Introduction</b>	<b>4</b>
<b>2 Workspace Set-up</b>	<b>5</b>
2.1 Firefox Developer Edition . . . . .	5
2.2 Java Development Kit . . . . .	6
<b>3 Eclipse Configuration</b>	<b>6</b>
3.1 Preferences . . . . .	6
3.2 Plug-in and agent installation . . . . .	6
3.3 Importing the framework . . . . .	7
<b>4 Database Configuration</b>	<b>7</b>
4.1 MariaDB . . . . .	7
4.2 DBeaver . . . . .	7
<b>5 Conclusions</b>	<b>8</b>
<b>6 Bibliography</b>	<b>9</b>

## Executive Summary

This report covers the process we followed to configure the development environment for the Acme AirNav Solutions (Acme ANS for short) project.

Said process is divided into three parts in this report. The first one covers the basic set-up of the directory structure of the workspace, which in turn includes the installation of the browser used to run the application to be developed as part of the project and the configuration of the Java platform development kit used to develop the software.

The second part comprises the configuration of the Eclipse Integrated Development Environment, used to produce the code for the project (including installation of related plug-ins and agents used to ease the development process) and the set-up of the software development framework used as scaffolding upon which to build our application.

Finally, this report also covers the configuration of the database management system that will be used to store the information required for the application to be developed.

## Revision History

Revision	Date	Description
1.0	2025-02-16	Initial version
1.1	2025-02-17	Drafted document sections
1.2	2025-02-18	Started writing definitive document sections
1.3	2025-02-19	Wrote definitive document sections

# 1. Introduction

The configuration of the development environment constitutes a crucial part of any software development process, as it consists of performing every necessary step to properly configure every tool to be used for the duration of the project in the development process.

Therefore, the aim of this report is to discuss the process we followed to configure the development environment for the Acme AirNav Solutions project. The goal of this project is to develop a Web Information System (or WIS, for short) to assist airports in the management of charter flights; therefore, tools are required to produce, compile and execute the source code for said system (which, in our case, are the Eclipse Integrated Development Environment, or IDE for short, and the Java Development Kit, or JDK), as well as to visualize and interact with the client-side presentation of the system (in the case of this project, the Mozilla Firefox Developer Edition web browser) and to store and manage the information required to perform the system's functions, which consisted of the MariaDB database and the DBeaver database management tool.

Most of these tools were provided as part of a workspace folder supplied in the course materials. Said workspace folder also provided a directory structure for the resources used throughout the project. A guide was also provided that instructed us in the process of configuring the environment.

The structure of the document will be the following: firstly, the configuration of the directory to be used as a workspace for the project will be covered, with subsections for both the web browser installation and the development kit configuration; secondly, the configuration of the Eclipse IDE will be discussed, including subsections dedicated to the configuration of program preferences, extensions, and the software framework to be used for the project; lastly, the database system configuration will be addressed, with two subsections devoted to the database system itself and the database management system respectively.

## 2. Workspace Set-up

In order to set-up the general workspace, we started by downloading and decompressing the provided files that conform the **Workspace-25** directory. This way, we ended up with the following directory structure:

```
Workspace-25
├── Depot
├── Frameworks
├── Projects
├── Scrapbook
├── Starters
├── Tools
│   ├── Agents
│   ├── IDEs
│   ├── Infrastructure
│   └── Plugins
```

Where:

- The **Depot** folder will contain the projects that are not being actively worked on.
- The **Frameworks** folder contains the frameworks to be used in the software development process. For now, it only contains the Acme framework.
- The **Projects** folder contains the projects in which work is being carried out.
- The **Scrapbook** folder contains miscellaneous items.
- The **Starters** folder contains example projects.
- The **Tools** folder contains the various programs and extensions to be used in the software development process:
  - The **Agents** folder contains artifacts that will alter the Java code loaded into the Java Virtual Machine.
  - The **IDEs** folder contains software development tools.
  - The **Infrastructure** folder contains programs used as the underlying infrastructure for the developed application.
  - The **Plugins** folder contains extensions for the development tools.

### 2.1. Firefox Developer Edition

In order to be able to test the application's user interface, a web browser is needed (as the application to be developed is a Web Information System). Therefore, we used the

provided installer to install the Mozilla Firefox Developer Edition web browser on our computers.

We then added both the Firefox Developer Edition directory and GeckoDriver (a tool that allows connecting Java and Firefox) to the PATH environment variable so as to make them recognizable from any directory using the command line.

## **2.2. Java Development Kit**

Finally, we also added the `bin` directory of the provided Java Development Kit (JDK) version to the PATH environment variable and set the `JAVA_HOME` environment variable to the root directory of the provided JDK version, so as to make it the default JDK installation on our computers. The JDK will be used to compile and execute the resulting application, as it will be developed in the Java platform.

## **3. Eclipse Configuration**

Eclipse is the tool that will primarily be used in this project to produce the source code of the application. It is an Integrated Development Environment (or IDE), meaning it is also responsible of invoking the compiler and has an integrated debugger used to execute and test the application.

A full installation of Eclipse was provided as part of the workspace (in the `IDEs` folder). Therefore, no installation was required prior to launching the application.

### **3.1. Preferences**

The Eclipse IDE comes bundled with a version of the Java Development Kit, which is different from the one we were to use for the project. Therefore, we changed it in the IDE preferences to the Java version 21 Development Kit that exists as part of the workspace.

Furthermore, we also enabled assertions in the compiler settings, which are used to perform critical checks on program variables. We also had to change the compiler compliance level to 21, so as to enable the latest features available in the JDK.

### **3.2. Plug-in and agent installation**

Then, we proceeded to install two Eclipse plug-ins we will use to ease the development process:

- The CSVEdit plug-in allows editing `.csv` files in the Eclipse editor, which are used to store the initial data used to populate the database.
- The SonarLint plug-in statically analyzes the code being written in order to detect potential errors prior to compile-time.

Afterwards, we closed Eclipse and installed the Lombok agent (which eases the programming process by auto-generating certain common types of methods) using the installer provided as part of the workspace.

### 3.3. Importing the framework

We then launched Eclipse again and imported the ACME framework (provided as part of the workspace in the **Frameworks** folder) (making sure to specify the `[artifactId]-[version]` naming template when importing it so as to ensure correct detection of project properties).

## 4. Database Configuration

MariaDB is the relational database used to store the necessary information for the Information System resulting from the project. In order to easily manage the database, we will use the DBeaver tool, which provides a graphical interface for managing databases.

Hence, Configuring the database was a two-part process: first, we had to set up the database itself (MariaDB), then set up the database management tool (DBeaver), including connecting it to the database.

### 4.1. MariaDB

In order to configure MariaDB, all we had to do was execute the `setup-mariadb` script provided as part of the workspace. Launching a database instance is now a matter of executing the `start-mariadb` script, also provided in the workspace.

### 4.2. DBeaver

With the MariaDB instance still executing, we launched DBeaver (of which an installation was provided as part of the workspace)

We created two DBeaver connections:



- An administrator connection, for which we created a user with username `root` and password `r00t-P@$$w0rd`. This connection will be used to create the necessary databases.
- A regular connection, for which we created a user with username `acme-user` and password `ACME-Us3r-P@ssw0rd`. This connection will be the one utilized to make use of the database at execution time.

## 5. Conclusions

After following the process described above to configure our development environment, the team is more than ready to begin work on the application. The process was turned into an easier undertaking thanks to the guide provided as part of the course materials, which thoroughly describes the minutiae of setting up each of the required development tools.

Testing the configured environment with the example projects provided in the workspace is an appropriate way of testing its correctness. In our case, each member of the team launched the supplied example projects and obtained positive results (every member was able to launch the resulting application and interact with it through the web user interface in the browser), with no issues being reported regarding the workspace configuration insofar.

Therefore, it can be concluded that the configuration process was a success, thus enabling the team to perform their project-related tasks in an easier, more streamlined way.

## 6. Bibliography

Intentionally blank.