**Group Number:** DP2-C1-027  
 **Repository:**[AirNav-Logistics Repository](https://github.com/DP2-C1-027/AirNav-Logistics.git) **Workgroup Members:**

-Nora Peñaloza Friqui (norpennfri@alum.us.es)

-Jose Garcia de Tejada Delgado (josgardel8@alum.us.es)

-Lucia Campos Diez (luccamdie@alum.us.es)

-Nicolas Gomez Claraco (nicgomcla@alum.us.es)

-Manuel Jesus Niza Cobo (mannizcob@alum.us.es)

**Date**: February 16, 2025

**Table of Contents**

1. **Executive Summary..........................................................Page 2**
2. **Revision Table...............................................................Page 3**
3. **Introduction.................................................................Page 4**
4. **Project Setup and Configuration........................................Page 5**
5. **Database Configuration....................................................Page 6**
6. **Application Execution.....................................................Page 7**
7. **Conclusions.................................................................Page 8**
8. **Bibliography.................................................................Page 9**

**Executive Summary**

This document outlines the steps involved in setting up and configuring the AirNav-Logistics project for local development. The process begins with cloning the repository into the local directory and configuring the development environment using Eclipse as the Integrated Development Environment (IDE). Following the setup of the IDE, MariaDB is initialized, and the necessary databases are created using DBeaver for proper configuration. Launchers are created, and the database is populated to ensure the application has the necessary data before execution. After successful initialization, the application is launched and accessible via a provided URL. This document provides a detailed guide on how to successfully configure the project environment and ensure that all systems work efficiently for development and testing purposes.

**Revision Table**

|  |  |  |
| --- | --- | --- |
| **Revision Number** | **Date** | **Description** |
| **1.0** | **February 16, 2025** | **Initial release of the document outlining the project setup.** |

**Introduction**

This document describes the detailed steps taken to set up and configure the AirNav-Logistics project for local development. It covers the process of cloning the repository, configuring the development environment, and initializing the necessary databases. Additionally, it includes instructions on launching the application and confirming its successful execution. This guide is intended for developers seeking to set up the project locally and run it in a development environment.

The structure of this document is as follows: Section 1 provides an executive summary of the project setup process. Section 2 outlines the revision history. Section 3 provides an introduction to the document and its content. Section 4 explains the steps for cloning the project repository and setting up the development environment in Eclipse. Section 5 describes the process of configuring the database. Section 6 details the application execution steps. Finally, Section 7 offers conclusions, and Section 8 includes the bibliography, if applicable.

**Project Setup and Configuration**

The initial step involved cloning the AirNav-Logistics repository into the “projects” directory to ensure the availability of the codebase for local development. The following Git command was executed in the Git console, located in the “projects” folder:

git clone https://github.com/DP2-C1-027/AirNav-Logistics.git

Eclipse was selected as the Integrated Development Environment (IDE) to support the coding, debugging, and execution processes. The choice of Eclipse provided a suitable environment for efficient development and navigation through the project.

**Database Configuration**

The next step was to start MariaDB, which was essential for the proper functioning of the application. To begin this process, the start-mariadb.cmd file was located and executed. Alternatively, the command prompt was directed to the folder containing the script and the following command was run:

start-mariadb

DBeaver was utilized for database configuration. A new SQL script was created and executed to establish the necessary databases for the project. Below is the script used to create the databases and assign appropriate privileges to the user:

-- create-database.sql

-- Copyright (c) 2012-2025 Rafael Corchuelo.

-- In keeping with the traditional purpose of furthering education and research, it is

-- the policy of the copyright owner to permit non-commercial use and redistribution of

-- this software. It has been tested carefully, but it is not guaranteed for any particular

-- purposes. The copyright owner does not offer any warranties or representations, nor do

-- they accept any liabilities with respect to them.

-- HINT: replace "Project-Version" by your project name and version.

drop database if exists `Acme-ANS-D01-25.1.0`;

create database `Acme-ANS-D01-25.1.0`

character set = 'utf8mb4'

collate = 'utf8mb4\_unicode\_ci';

grant select, insert, update, delete, create, drop, references, index, alter,

create temporary tables, lock tables, create view, create routine,

alter routine, execute, trigger, show view

on `Acme-ANS-D01-25.1.0`.\* to 'acme-user'@'%';

drop database if exists `Acme-ANS-D01-25.1.0-Test`;

create database `Acme-ANS-D01-25.1.0-Test`

character set = 'utf8mb4'

collate = 'utf8mb4\_unicode\_ci';

grant select, insert, update, delete, create, drop, references, index, alter,

create temporary tables, lock tables, create view, create routine,

alter routine, execute, trigger, show view

on `Acme-ANS-D01-25.1.0-Test`.\* to 'acme-user'@'%';

This script created two databases—Acme-ANS-D01-25.1.0 for production and Acme-ANS-D01-25.1.0-Test for testing—and granted the necessary privileges to the user acme-user.

**Application Execution**

Once the database was configured, the next step was to create the required launchers for database population and application execution. The create-launchers command was run from the command prompt in the workspace to automate this process.

After setting up the project within Eclipse, the database was populated using the development-populator#sample command. This ensured that the database contained the necessary data for the application to function correctly. Following this, the application was initiated using the development-application#run command, starting the development environment.

Upon successful startup, the terminal displayed a URL that provided access to the application’s interface. This URL was copied and pasted into a web browser to verify that the application was functional and that the setup process had been completed successfully.

**Conclusions**

In conclusion, this document outlines the key steps required to set up and configure the AirNav-Logistics project for local development. The process involves cloning the repository, configuring the development environment in Eclipse, and initializing MariaDB. By creating the necessary databases and launching the application, the environment was fully prepared for development and testing. Successful completion of the setup confirmed that the application was functioning correctly.

The process outlined in this document ensures that developers can replicate the setup, manage project dependencies efficiently, and contribute to the development of the AirNav-Logistics application.

**Bibliography**

Intentionally blan