# Amarradero Llanero

Amarradero Llanero focuses on integrating Camunda, a business process automation platform, with a .NET Core application. The integration allows for the management of workflow and tasks in Camunda while connecting to a PostgreSQL database for storing and retrieving relevant information.

## Index

1. [Description](#description)

2. [Prerequisites](#prerequisites)

3. [Usage](#usage)

4. [Development Summary](#development-summary)

## Description

The Camunda and .NET Core integration is crucial for enhancing automation and efficiency in business processes. \*\*Amarradero Llanero\*\* is built upon Docker containers to ensure portability and scalability of the application. A PostgreSQL database is used to store relevant data, and processes and tasks are configured in Camunda to coordinate activities.

## Prerequisites

Before running the program, ensure that you have the following prerequisites installed:

1. \*\*Code editor\*\*: if you want to modify the program you must have[JCHV1], we recommend using Visual Studio Code (VS Code). You can download it from the [official website](https://code.visualstudio.com/download).

2. \*\*Version control system\*\*: Install GIT from the [official website](https://git-scm.com/downloads).

3. \*\*Clone the repository\*\*: Use the following command to clone the repository: `git clone https://github.com/BPMN-sw-evol/AmarraderoLlanero.git`.

4. \*\*Docker Desktop\*\*: Install Docker from the [official website](https://www.docker.com/products/docker-desktop/)

5. \*\*WSL2\*\*: Use the following command to install WSL2

````

wsl --install

wsl --set-default-version 2

````

To download a specific distribution, we use `wsl --list --online` to list the available distributions, then use `wsl --install -d "distribution-version"`.

## Usage

To execute the program:

1. Create and save an .env file in the root of the project [JCHV2]with the following structure:

````json

DATABASE\_SERVER=db

DATABASE\_PORT=5432[JCHV3]

DATABASE\_DB=AmarraderoLlanero

DATABASE\_USER=postgres

DATABASE\_PASSWORD=admin

````

2. Execute the following command, in a command console located in the root of the project, to build the images and raise the containers: `docker-compose up`.

NOTE: If you want to know whether the images have been correctly built, in a new terminal or in the command prompt, run: `docker images`.

xxx

3. XXXX In the browser, enter the following address to create the database: `localhost/dbconexion`.

If you want to enter to check the database, in a console (terminal or DOS prompt) run: `docker exec -it database psql -U postgres`.

NOTE: Review (and add) how to consult de db with applications as PgmAdmin

4. In the camunda platform, we deploy the \*\*AmarraderoLlanero\*\* model including \*\*the other files located\*\* in the \*\*BPMN-Models\*\* folder of the project.

5. In the browser, enter the following address to enter the database engine and start an instance of the BPM model: `http://localhost:8080/camunda-welcome/index.html`

At this point you can:

1. Start a BPM model instance
2. Interact with it
3. Review the records in the database. For this you can:

````sql

\c AmarraderoLlanero

SELECT \* FROM "Users";

````

if you want to check the order records in the database, run:

````sql

\c AmarraderoLlanero

SELECT \* FROM "Orders";

````

## Development Summary

The project focuses on creating a restaurant order management application using Docker, ASP.NET Core, Camunda BPM, and PostgreSQL. The application allows users to register, place orders from different areas of the restaurant, and automates workflow processes. User and order information is stored in the PostgreSQL database.

The database store the following attributes for each instance of BPM model:

\*\*Table Users\*\*

| Id | Username | Fullname | Email | Password |

| ------------- | --------------------|-----------------------|--------------------|--------------- |

|Unique identifier| User's username | Full name | Email address | Password |

\*\*Table Orders\*\*

| Id | CurrentDate | PedidoAsadero | CantPedidoAsadero | PedidoCocina | CantPedidoCocina | pedidoBar| cantPedidoBar |

| ------------- | ---------|-----------|-----------|------------|------------|--------|--------------- |

|Unique identifier for each record.| The date when the order was placed. | The order from the grill section. | The quantity of the grill order. | The order from the kitchen section. | The quantity of the kitchen order. | The order from the bar section. | The quantity of the bar order. |

[JCHV1]¿se refieren al programa de .NET? (program.cs?) ...Revisar la forma de decirlo.

[JCHV2](the cloned folder)

[JCHV3]Agregar que éstos no se pueden modificar por xxxxx