

ReadMe

2026-01-18

PROJECT EXECUTION INSTRUCTIONS

This project is distributed as two pre-built Docker containers: A REST API A Shiny web application that consumes the API No source code or build step is required to run the project. Two Docker Compose configurations are provided, depending on the container registry used.

REQUIREMENTS

Docker installed (Docker Desktop or Docker Engine) Docker Compose

To verify installation: `docker --version` `docker compose version`

DOCKER COMPOSE OPTIONS

There are two equivalent Docker Compose files:

COMPOSE 1 – Docker Hub images

This version uses images hosted on Docker Hub. Images: `ivanlg95/archivosdocker-api:v1` `ivanlg95/archivosdocker-shiny:v1`

COMPOSE 2 – GitHub Container Registry (GHCR) images

This version uses images hosted on GitHub Container Registry. Images: `ghcr.io/oscarlopezll/archivosdocker-api:latest` `ghcr.io/oscarlopezll/archivosdocker-shiny:latest` Both versions provide the same functionality. Only one of them should be used.

EXECUTION STEPS

Choose the appropriate `docker-compose.yml` file (Compose 1 or Compose 2) and place it in a local directory. Open a terminal in that directory. Start the services with: `docker compose up` Docker will automatically pull the required images from the selected registry if they are not available locally.

ACCESSING THE APPLICATION

Once the containers are running:

Shiny application: `http://localhost:3838`

API: `http://localhost:8000`

CONTAINER COMMUNICATION

The Shiny application communicates with the API through Docker's internal network using the service name "api". Localhost is not used for container-to-container communication.

STOPPING THE APPLICATION

To stop the containers, press: `Ctrl + C`

Then run: `docker compose down`

NOTES No additional configuration is required. The project has been tested on recent Docker versions on macOS. Docker Hub and GitHub Container Registry versions are functionally identical.