

Escuela Politécnica Nacional

Facultad de ingeniería de Software

1. Introducción

El documento describe la implantación del sistema AntCiberDron, desarrollado con la metodología Extreme Programming (XP). Esta fase busca verificar que el sistema esté correctamente integrado y funcione de manera estable, cumpliendo los requerimientos. Se valida la operación completa, desde la autenticación de usuarios hasta el procesamiento de coordenadas y la evaluación del arsenal mediante el autómata determinista. La implantación no incluye pruebas unitarias ni validación interna del código, ya que estas pertenecen al desarrollo en XP. Su objetivo es garantizar que el sistema esté listo para las pruebas de aceptación sin fallos críticos.

2. Objetivo

El objetivo de la implantación del sistema AntCiberDron es verificar que el sistema se ejecute correctamente de forma integral, asegurando la correcta integración de sus módulos y el funcionamiento estable del flujo principal, desde la autenticación de usuarios hasta la evaluación del tipo de arsenal mediante el autómata.

Esta etapa permite confirmar que el sistema se encuentra listo para su uso y preparado para la realización de las pruebas de aceptación, de acuerdo con la metodología Extreme Programming (XP).

3. Alcance

La implantación del sistema AntCiberDron comprende la verificación de la ejecución general del sistema y el correcto funcionamiento del flujo principal, incluyendo el inicio del sistema, la autenticación de usuarios, la carga del archivo *Grupo##.csv*, la visualización del proceso de carga y la evaluación del tipo de arsenal.

Esta etapa no incluye pruebas internas del código ni pruebas unitarias, ya que estas forman parte de la fase de desarrollo en la metodología Extreme Programming (XP). El alcance de la implantación se limita a confirmar que el sistema se encuentra correctamente integrado y operativo, previo a la ejecución de las pruebas de aceptación.

4. Entorno de Implantación

La implantación del sistema AntCiberDron se realizó en un entorno **contenedorizado**, utilizando **Docker** como plataforma de despliegue, lo que permitió garantizar un entorno de ejecución controlado, reproducible y consistente.

El entorno de implantación utilizado es el siguiente:

- **Lenguaje de programación:** Java
- **Entorno de desarrollo (IDE):** Visual Studio Code
- **Plataforma de contenedores:** Docker
- **Tipo de ejecución:** Contenedor Docker ejecutado de forma local
- **Sistema operativo anfitrión:** Windows

Previo a la ejecución del sistema, se realizó la **instalación y configuración de Docker**, posterior a lo cual se procedió a construir y ejecutar el contenedor que aloja el sistema AntCiberDron. Este enfoque permitió validar que el sistema puede ejecutarse correctamente de manera independiente del entorno local del desarrollador.

5. Procedimiento de Implantación

La implantación del sistema AntCiberDron se realizó utilizando una arquitectura basada en contenedores Docker, permitiendo el despliegue controlado del frontend y backend del sistema. El procedimiento seguido fue el siguiente:

1.1. Preparación del entorno

- Se verificó la instalación y correcto funcionamiento de **Docker** y **Docker Compose** en el equipo anfitrión.
- Se confirmó la disponibilidad del proyecto **WORKSHOP-XP** en el entorno local, con su estructura organizada por fases de XP.

1.2. Estructura del proyecto

El proyecto se encuentra organizado de la siguiente manera:

- server/: backend del sistema.
- src/: código fuente del frontend.
- XP/: documentación organizada por fases de Extreme Programming.
- Grupo03.csv: archivo de coordenadas, horarios y tipo de arsenal.
- Dockerfile: definición de la imagen del sistema.
- docker-compose.yml: orquestación de los servicios.
- nginx.conf: configuración del servidor web.
- Scripts de arranque: start.ps1 y start-all.ps1.

1.3. Construcción de los contenedores

- Desde la raíz del proyecto se procedió a construir las imágenes definidas en el archivo docker-compose.yml, utilizando las configuraciones del Dockerfile y del servidor **Nginx**.
- Se verificó que el proceso de construcción finalice correctamente, sin errores críticos.

1.4. Ejecución del sistema

- Se levantaron los servicios definidos mediante **Docker Compose**, iniciando el backend y el frontend del sistema.
- Se confirmó el arranque correcto de los contenedores mediante los logs de Docker.
- El sistema quedó disponible para su uso a través del navegador web, sirviendo la aplicación mediante **Nginx**.

1.5. Verificación funcional

- Se accedió al sistema AntCiberDron desde la interfaz web.
- Se realizó la autenticación de usuarios.
- Se verificó la visualización de los datos del equipo.
- Se ejecutó la lectura del archivo Grupo03.csv, validando la carga de coordenadas, horarios y tipo de arsenal.
- Se comprobó la visualización del **loading** durante el procesamiento.
- Se validó la evaluación del tipo de arsenal mediante el **autómata determinista**, observando el resultado True/False y la presentación de la coordenada cuando correspondía.

1.6. Finalización

Se verificó que el sistema opere de manera estable, sin cierres inesperados ni fallos críticos, quedando correctamente implantado y listo para la ejecución de las pruebas de aceptación.

6. Resultados

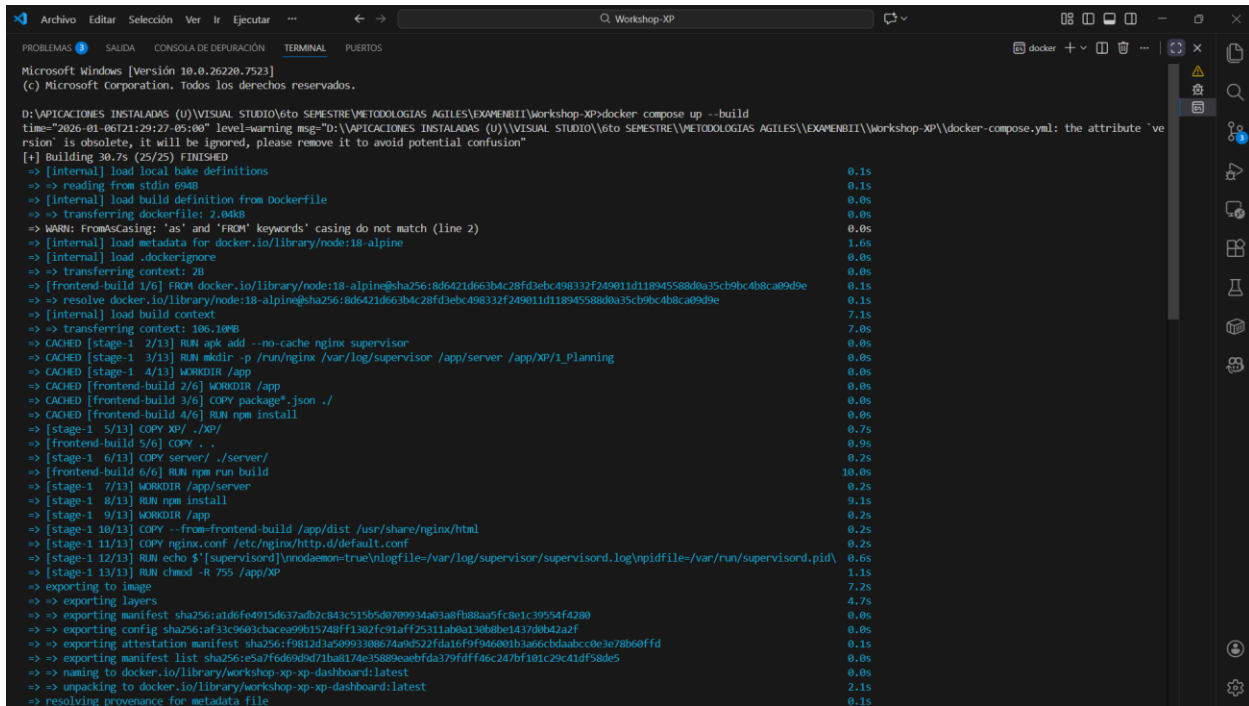
Como resultado del proceso de implantación, el sistema **AntCiberDron** fue ejecutado correctamente dentro del entorno contenedorizado definido con **Docker**, logrando el despliegue exitoso de los servicios correspondientes al frontend y backend del sistema.

Durante la ejecución se verificó que:

- Los contenedores definidos en el archivo docker-compose.yml se iniciaron de manera correcta.
- El sistema fue accesible a través del navegador web mediante el servidor **Nginx**.
- La autenticación de usuarios se realizó sin inconvenientes.
- La información del equipo se visualizó correctamente.
- El archivo Grupo03.csv fue leído y procesado exitosamente.
- El proceso de carga (**loading**) se mostró durante la lectura y procesamiento de las coordenadas.
- La evaluación del tipo de arsenal mediante el **autómata determinista** produjo los resultados esperados (True/False), mostrando la coordenada correspondiente cuando aplicaba.

No se presentaron errores críticos ni fallos de ejecución durante la implantación, por lo que el sistema se considera **correctamente implantado y listo para la ejecución de las pruebas de aceptación**, conforme a la metodología Extreme Programming (XP)

7. Evidencias



```
Microsoft Windows [Versión 10.0.26220.7523]
(c) Microsoft Corporation. Todos los derechos reservados.

D:\VAPICACIONES INSTALADAS (U)\VISUAL STUDIO\6to SEMESTRE\METODOLOGIAS AGILES\EXAMENBIII\Workshop-XP>docker compose up --build
time="2026-01-06T21:29:27-05:00" level=warning msg="D:\VAPICACIONES INSTALADAS (U)\VISUAL STUDIO\6to SEMESTRE\METODOLOGIAS AGILES\EXAMENBIII\Workshop-XP\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Building 30.7s (25/25) FINISHED
-> [internal] load local bake definitions                                0.1s
-> reading from stdin 694B                                             0.1s
-> [internal] load build definition from Dockerfile                    0.0s
-> transferring dockerfile: 2.04kB                                     0.0s
-> WARN: FromCasing: 'as' and 'FROM' keywords' casing do not match (line 2) 0.0s
-> [internal] load metadata for docker.io/library/node:18-alpine      1.6s
-> [internal] load .dockerignore                                         0.0s
-> transferring context: 2B                                             0.0s
-> [frontend-build 1/6] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d8a35cb9bc4b8ca09d9e 0.1s
-> resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d8a35cb9bc4b8ca09d9e 0.1s
-> [internal] load build context                                         7.1s
-> transferring context: 106.10kB                                       7.0s
-> CACHED [stage-1 2/13] RUN apk add --no-cache nginx supervisor      0.0s
-> CACHED [stage-1 3/13] RUN mkdir -p /run/nginx /var/log/supervisor /app/server /app/XP/1_Planning 0.0s
-> CACHED [stage-1 4/13] WORKDIR /app                                  0.0s
-> CACHED [frontend-build 2/6] WORKDIR /app                           0.0s
-> CACHED [frontend-build 3/6] COPY package*.json ./                  0.0s
-> CACHED [frontend-build 4/6] RUN npm install                         0.0s
-> [stage-1 5/13] COPY XP/ ./XP/                                       0.7s
-> [frontend-build 5/6] COPY . .                                        0.9s
-> [stage-1 6/13] COPY server/ ./server/                               0.2s
-> [frontend-build 6/6] RUN npm run build                              10.0s
-> [stage-1 7/13] WORKDIR /app/server                                  0.2s
-> [stage-1 8/13] RUN npm install                                       9.1s
-> [stage-1 9/13] WORKDIR /app                                          0.2s
-> [stage-1 10/13] COPY --from=frontend-build /app/dist /usr/share/nginx/html 0.2s
-> [stage-1 11/13] COPY nginx.conf /etc/nginx/httpd/default.conf      0.2s
-> [stage-1 12/13] RUN echo $'[supervisord]/nodaemon=true\nlogfile=/var/log/supervisor/supervisord.log\npidfile=/var/run/supervisord.pid' 0.6s
-> [stage-1 13/13] RUN chmod -R 755 /app/XP                             1.1s
-> exporting to image                                                  7.2s
-> exporting layers                                                    4.7s
-> exporting manifest sha256:a1d6fe4915d637ad82c843c515b5d8709934a03a8fb88aa5fc0e1c39554f4280 0.0s
-> exporting config sha256:af3c5603cbace99b15748ff1302fc91aff25311ab0a13688be1437d0b42a2f 0.0s
-> exporting attestation manifest sha256:f08124f90993308e74a9d5271da16f9f946001b3a6c6dabac0e3e70b60ffd 0.1s
-> exporting manifest list sha256:e5a7f6d69dbd71ba8174e35889e0ebfda379f9ff45c247bf101c29c41df58de5 0.0s
-> naming to docker.io/library/workshop-xp-xp-dashboard:latest       0.0s
-> unpacking to docker.io/library/workshop-xp-xp-dashboard:latest    2.1s
-> resolving provenance for metadata file                              0.1s
```

Figura 1. Evidencia – Construcción y ejecución del contenedor Docker

La imagen evidencia la ejecución del comando de construcción y despliegue del sistema mediante **Docker Compose**, donde se observa la creación exitosa de la imagen del proyecto AntCiberDron. Durante el proceso se cargaron correctamente las definiciones del *Dockerfile*, se construyó el frontend, se copiaron los archivos del backend y se configuró el servidor **Nginx** dentro del contenedor.

El proceso finalizó sin errores críticos, confirmando que el contenedor fue generado y ejecutado correctamente, permitiendo la implantación exitosa del sistema en un entorno controlado y reproducible.

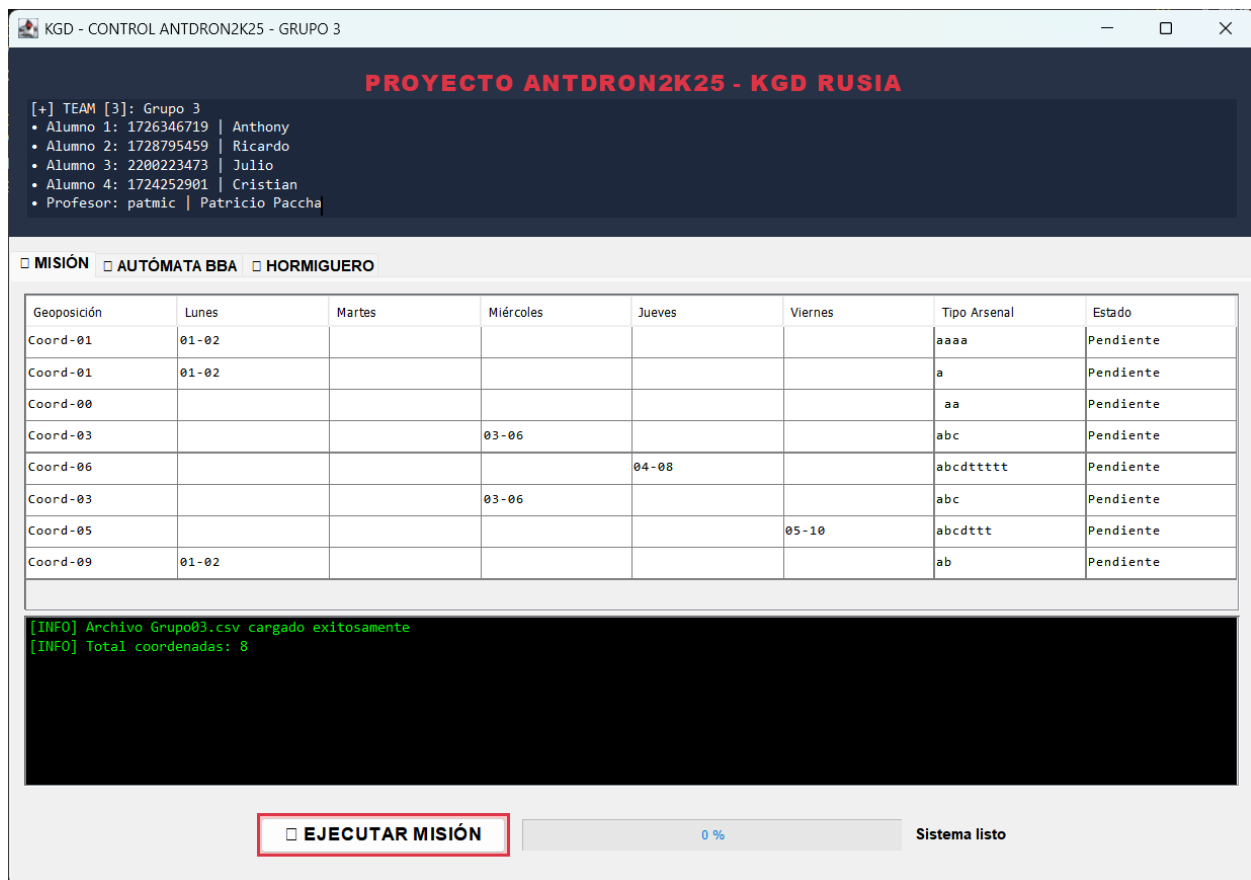


Figura 2. Evidencia – Sistema AntCiberDron en ejecución

La imagen evidencia la ejecución correcta del sistema AntCiberDron, mostrando la carga exitosa del archivo Grupo03.csv, la visualización de las coordenadas y la información del equipo. El sistema se encuentra operativo y listo para la ejecución de las misiones, confirmando una implantación exitosa.

8. Conclusiones

El proceso de implantación del sistema AntCiberDron se realizó de manera satisfactoria utilizando un entorno contenedorizado con Docker, permitiendo la correcta ejecución e integración de los componentes del sistema.

Se verificó que el sistema inicia correctamente, carga el archivo de coordenadas correspondiente al grupo, presenta la información del equipo y muestra las misiones de forma adecuada, sin presentar errores críticos durante su ejecución.

En consecuencia, se concluye que el sistema se encuentra **correctamente implantado y operativo**, listo para la ejecución de las **pruebas de aceptación**, conforme a la metodología Extreme Programming (XP).