

Corporación Universitaria del Huila – CORHUILA

Asignatura: Arquitectura de software

INFORME – FASE 1

Montaje del Sistema y Pruebas Funcionales

Docente:

Luis Ángel Vargas Narvaez

Integrantes:

Angie Valentian Flórez Vargas - Avflorez-2023a@corhuila.edu.co

Sebastián Puentes Gonzales – Spuentes-2022b@corhuila.edu.co

Sergio Alejandro Muñoz Cabrera - Samunoz-2023a@corhuila.edu.co

Miguel Ángel Rivera Lozano – marivera-2023a@corhuila.edu.co

Neiva – Huila

17 de febrero de 2026

Introducción

La presente fase tiene como finalidad documentar el despliegue técnico del sistema de encuestas suministrado por el taller de Arquitectura de Software, así como verificar su comportamiento funcional bajo un entorno local controlado.

El objetivo no se limita únicamente a ejecutar la aplicación, sino a comprender su estructura de despliegue, validar la interacción entre sus componentes y confirmar el estado actual del sistema antes de proceder con su análisis arquitectónico.

Esta fase constituye la línea base sobre la cual se desarrollarán las siguientes etapas del taller.

Descripción del entorno de ejecución

Para garantizar una ejecución estable del sistema, se empleó un entorno basado en contenedores Docker, permitiendo aislar los servicios y evitar conflictos de dependencias locales.

- Sistema Operativo: Windows 10/11
- Docker Desktop (motor de los contenedores)
- Git (Control de versiones)
- Navegador Google Chrome
- IDE para revisión de código

El uso de contenedores facilita la replicabilidad del entorno y asegura que todos los participantes trabajen bajo condiciones equivalentes.

Actividad 1.1 – Procedimiento de Despliegue

1. Obtención del código fuente

Se descargó el repositorio del proyecto mediante Git y se accedió al directorio raíz que contiene el archivo de configuración docker-compose.yml.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19045.6466]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\--\Documents>git clone https://github.com/lanvargas94/Semana-2.git
Cloning into 'Semana-2'...
remote: Enumerating objects: 49, done.
remote: Counting objects: 100% (49/49), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 49 (delta 0), reused 49 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (49/49), 10.70 KiB | 304.00 KiB/s, done.

C:\Users\--\Documents>cd Semana-2
```

Imagen 1. Clonar repositorio

En la imagen se observa la ejecución de la clonación del repositorio del proyecto “Espaguetti de encuestas”, permitiendo así obtener de manera local los módulos del sistema, siendo estos el backend, frontend y la configuración del Docker Compose,

Seguidamente se validó la estructura de las carpetas, Encontrando los módulos correspondientes al backend, frontend y el archivo de configuración “docker-compose.yml”.

1-2 Construcción y levantamiento de servicios

Se ejecutó el comando:

```
docker-compose up --build
```

Este comando nos permitió construir las imágenes en docker, al igual que los contenedores para así luego iniciar los servicios necesarios.

Los servicios que se desplegaron fueron:

Backend → Puerto 8080

Frontend → Puerto 4200

Base de datos PostgreSQL → Puerto 5432

```
C:\Windows\System32\cmd.exe - docker compose up --build
#34 [backend] exporting to image
#34 exporting layers
#34 exporting layers 3.ls done
#34 exporting manifest sha256:32bc136258c9cc17bf0ea5216702f422ec5a8b0e562256c73c17cfdf55e82c8e2 0.1s done
#34 exporting config sha256:863a0b895c3e6f5bb413d3be26653503cef1c49e0f35bd2c1d0989fb0fd9dc 0.0s done
#34 exporting attestation manifest sha256:df8f813a71393d1d0dfc9bc68db6332b7f77ce0db9aa18fa8d0d7085593b97ca8 0.1s done
#34 exporting attestation manifest sha256:df8f813a71393d1d0dfc9bc68db6332b7f77ce0db9aa18fa8d0d7085593b97ca8 0.1s done
#34 naming to docker.io/library/semana-2-backend:latest
#34 naming to docker.io/library/semana-2-backend:0.0s done
#34 naming to docker.io/library/semana-2-backend:0.0s done
#34 unpacking to docker.io/library/semana-2-backend:latest
#34 unpacking to docker.io/library/semana-2-backend:0.5s done
#34 DONE 4.1s

#35 [backend] resolving provenance for metadata file
#35 DONE 0.1s
[+] up 6/6
  Image semana-2-frontend      Built
  0.2s
  └ Container semana-2-db-1     Created
  └ Container semana-2-backend-1 Created
  └ Container semana-2-frontend-1 Created
Attaching to backend-1, db-1, frontend-1
db-1 | PostgreSQL Database directory appears to contain a database; Skipping initialization
db-1 |
db-1 | 2026-02-18 01:29:13.502 UTC [1] LOG:  starting PostgreSQL 15.16 (Debian 15.16-1.pgdg13+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 14.2.0-19) 14.2.0, 64-bit
db-1 | 2026-02-18 01:29:13.503 UTC [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
db-1 | 2026-02-18 01:29:13.505 UTC [1] LOG:  listening on IPv6 address "::", port 5432
db-1 | 2026-02-18 01:29:13.564 UTC [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
db-1 | 2026-02-18 01:29:13.617 UTC [29] LOG:  database system was shut down at 2026-02-17 13:59:57 UTC
db-1 | 2026-02-18 01:29:13.724 UTC [1] LOG:  database system is ready to accept connections
frontend-1 | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
frontend-1 | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
frontend-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
frontend-1 | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
frontend-1 | /10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
frontend-1 | /docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
frontend-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
frontend-1 | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
frontend-1 | /docker-entrypoint.sh: Configuration complete; ready for start up
frontend-1 | 2026/02/18 01:29:15 [notice] #1: using the "epoll" event method
```

Imagen 2. Docker Build

La imagen muestra la construcción de las imágenes Docker, mediante el comando Docker-Compose up –build, mostrando que cada servicio (frontend, backend y base de datos) fue compilado y preparado para su despliegue sin errores previos.

Imagen 3. Contenedor exitoso

En esta imagen se visualiza la creación e iniciación exitosa de los contenedores de cada módulo del sistema, validando que los servicios están listos para la ejecución y prueba funcional.

1.3 Verificación de servicios en ejecución

Una vez terminada la construcción, se verificó que los tres contenedores se encuentran en ejecución y sin errores de funcionamiento.

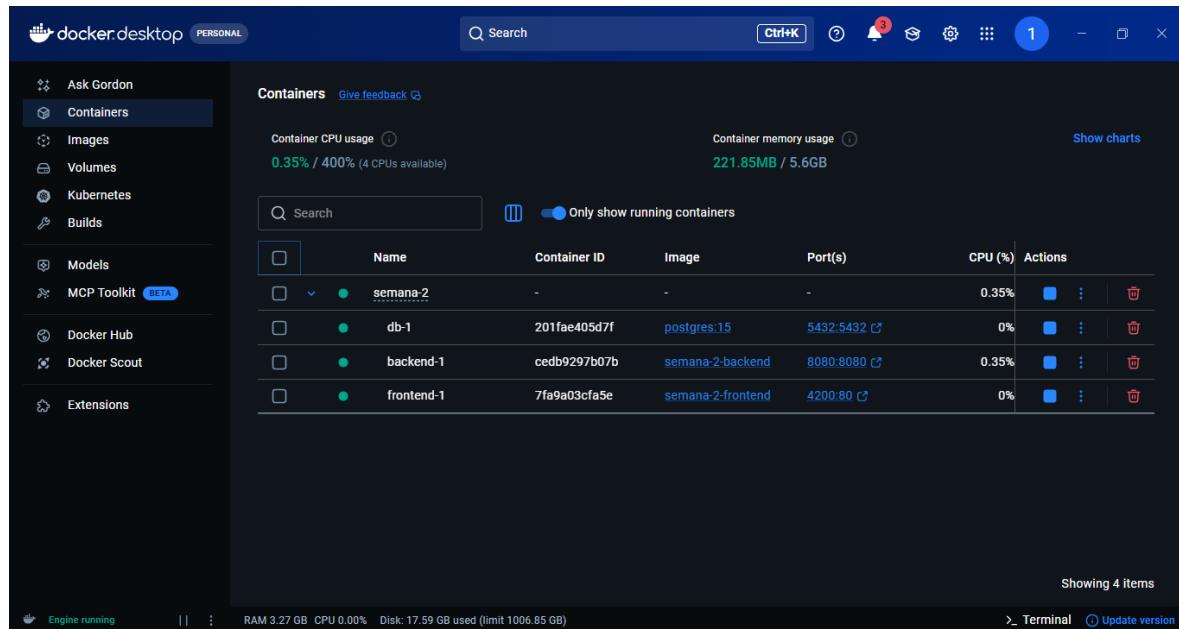


Imagen 4. Contenedores e imágenes en docker

La imagen muestra los 3 contenedores (Frontend, Backend y base de datos), evidenciando que si se encuentran activos y sin ningún error en la terminal, lo que comprueba que tenemos un entorno estable para realizar pruebas funcionales.

Actividad 1.2 – Pruebas Funcionales

Con el entorno corriendo, procedemos a realizar las pruebas para comprobar que funcionalidades tiene este programa.

| CASO | DESCRIPCIÓN | RESULTADO ESPERADO |
|-------|--------------------|--------------------|
| CP-01 | Crear una encuesta | Funciona ✓ |
| CP-02 | Mostrar error | No muestra error |
| CP-03 | Incrementa SI | Funciona ✓ |
| CP-04 | Incrementa NO | Funciona ✓ |
| CP-05 | Listar encuestas | Funciona ✓ |
| CP-06 | SQL Injection | No funciona |

CP-01 Crear encuesta válida

En esta captura se observa la creación de una nueva encuesta ingresando una pregunta válida con más de tres caracteres. El sistema permite publicar la encuesta correctamente y redirige a la vista de resultados, confirmando que la funcionalidad de creación de encuestas opera de manera adecuada.

The screenshot shows the 'Crear Encuesta' (Create Survey) page. At the top, there is a navigation bar with links to 'Encuestas Espagueti', 'Home', and 'Respuestas'. The main form has a title 'Crear Encuesta'. A text input field contains the question '¿Te gusta la programación?'. Below it is a 'Publicar Encuesta' (Publish Survey) button. Underneath the button, there is a section for 'Respuestas automáticas:' (Automatic responses) with a 'SI | NO' (Yes | No) button. The background of the page is light gray.

Imagen 5. Crear encuesta válida

CP-02 Crear encuesta con texto <3 caracteres

En esta captura se muestra el intento de crear una encuesta con un texto menor a tres caracteres. El sistema permite la creación de la encuesta sin mostrar un mensaje de error, lo que indica que no existe una validación adecuada en el backend o frontend para restringir este tipo de entradas.

The screenshot shows the 'Crear Encuesta' (Create Survey) page. The navigation bar at the top includes 'Encuestas Espagueti', 'Home', and 'Respuestas'. The main form has a title 'Crear Encuesta'. A text input field contains the question '>#'. Below the input field is a 'Publicar Encuesta' (Publish Survey) button. Underneath the button, there is a red error message 'Escribe algo mas largo' (Write something longer). There is also a section for 'Respuestas automáticas:' (Automatic responses) with a 'SI | NO' (Yes | No) button. The background of the page is light gray.

Imagen 6. Texto < 3 caracteres

CP-03 Votar “SI” en una encuesta / CP-04 Votar “NO” en una encuesta

En esta captura se visualiza el proceso de votación seleccionando la opción "SI" y el "NO" en una encuesta existente. Se observa que el contador de votos incrementa correctamente, lo que confirma que el sistema registra y actualiza los votos de forma adecuada.

Encuestas Espagueti [Home](#) [Respuestas](#)

¿Te gustan los perros calientes? ???

SI: 14
NO: 17

This screenshot shows a survey application interface. At the top, there's a navigation bar with links for 'Encuestas Espagueti', 'Home', and 'Respuestas'. Below that, a question is displayed: '¿Te gustan los perros calientes? ???'. Underneath the question are two buttons: 'SI' and 'NO'. Below the buttons, the counts are shown: 'SI: 14' and 'NO: 17'. The background is light gray.

Imagen 7. Se muestra cp-03/cp-04

CP-05 Listar todas las encuestas

En esta captura se observa la lista completa de encuestas disponibles en el sistema. Se muestran las preguntas junto con el número de votos positivos, negativos y el total, lo que confirma que el sistema recupera y presenta correctamente la información almacenada en la base de datos.

| Respuestas de Encuestas | | | |
|----------------------------|--------|-------|-----------|
| Hq | Si: 0 | NO: 0 | Total: 0 |
| Ver | | | |
| ¿Te gusta la programación? | Si: 5 | NO: 5 | Total: 10 |
| Ver | | | |
| te gustan los bgatos? | Si: 10 | NO: 0 | Total: 10 |
| Ver | | | |
| te gustan los perros? | Si: 3 | NO: 4 | Total: 7 |
| Ver | | | |

This screenshot shows a list of surveys. At the top, there's a header 'Respuestas de Encuestas'. Below it, four survey entries are listed. Each entry contains the question, the count of 'Si' votes, the count of 'No' votes, and the total count. There are also 'Ver' (View) links next to each entry. The background is light gray.

Imagen 8. Listar encuestas

CP-06 Probar SQL Injection: 'OR' 1='1

En esta captura se muestra el intento de realizar una inyección SQL ingresando el texto ' OR '1'='1 como pregunta en la encuesta. El sistema no presenta errores ni comportamientos anormales, lo que indica que la entrada fue tratada como texto normal. Aunque en esta prueba básica no se evidenció alteración del comportamiento, la construcción dinámica de consultas SQL representa un riesgo estructural que podría ser explotado bajo escenarios

más complejos.

The screenshot shows a web page titled "Encuestas Espagueti" with a sub-menu "Home" and "Respuestas". The main content area is titled "Crear Encuesta". A text input field contains the value "' OR '1'='1". Below it is a button labeled "Publicar Encuesta". A red error message "Error raro" is displayed. Underneath, there is a section titled "Respuestas automáticas:" with two buttons: "SI" and "NO".

Imagen 9. SQL Injection

Evaluación general del sistema

El despliegue del sistema fue exitoso y las funcionalidades principales operan conforme a lo esperado. Sin embargo, se identificaron aspectos que requieren mejora:

- Falta de validación robusta en entradas de datos.
- Riesgo potencial de vulnerabilidad por construcción manual de consultas SQL.
- Arquitectura backend con alta concentración de responsabilidades.

Estos hallazgos justifican la necesidad de un análisis arquitectónico más profundo en la siguiente fase del proyecto.

Conclusión

La ejecución de esta fase permitió comprender el funcionamiento actual del sistema desde una perspectiva técnica y funcional. Se verificó que el entorno puede desplegarse correctamente mediante Docker y que las funcionalidades principales están operativas.

No obstante, se identificaron debilidades relacionadas con validación de datos y seguridad estructural, lo cual evidencia la necesidad de aplicar principios de arquitectura limpia y patrones de diseño adecuados en fases posteriores.