

Corporación Universitaria del Huila – CORHUILA

Asignatura: Arquitectura de Software

**INFORME – FASE 1**

**Montaje, Exploración y Pruebas Funcionales**

Docente:

Luis Ángel Vargas Narvaez

Integrantes:

Juan José Torrejano Rojas - jjtorrejano-2032b@corhuila.edu.co

Jhon Edinson Marín Tapias - jemarin-2032b@corhuila.edu.co

Karina Cantillo Plaza - kcantillo-2032b@corhuila.edu.co

Danay Mariana Pereira Ospina - dmpereira-2032b@corhuila.edu.co

Neiva, Huila

15 de febrero de 2026

## Introducción

En la presente fase se realizó el montaje y verificación del sistema "Espagueti de Encuestas", una aplicación web desarrollada bajo una arquitectura monolítica compuesta por backend en Spring Boot, frontend en Angular y base de datos PostgreSQL, desplegada mediante Docker Compose.

El objetivo de esta fase fue configurar correctamente el entorno de ejecución, validar el funcionamiento de los servicios y ejecutar pruebas funcionales que permitieran comprobar el comportamiento general del sistema.

## Actividad 1.1 – Montaje del Entorno

### 1.1 Clonación del Repositorio

El proceso inició con la clonación del repositorio oficial del proyecto utilizando el siguiente comando:

```
Microsoft Windows [Version 10.0.19045.6466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Juan\Documents\arquitectura>git clone https://github.com/lanvargas94/Semana-2.git
fatal: destination path 'Semana-2' already exists and is not an empty directory.
```

*Imagen 1. Clonar repositorio*

En la figura se observa la ejecución del comando de clonación del repositorio del proyecto "Espagueti de Encuestas", permitiendo obtener localmente todos los módulos del sistema, incluyendo backend, frontend y configuración de Docker Compose.

Posteriormente, se verificó la estructura de carpetas, identificando los módulos correspondientes al backend, frontend y archivo de configuración docker-compose.yml.

### 1.2 Construcción y Ejecución de Contenedores

Para el despliegue del sistema se ejecutó el siguiente comando: *docker-compose up --build*. Este comando permitió construir las imágenes Docker, crear los contenedores e iniciar los servicios necesarios.

Los servicios desplegados fueron:

Frontend → Puerto 4200

Backend → Puerto 8080

Base de datos PostgreSQL → Puerto 5432

```
(C:) Microsoft Corporation. All rights reserved.

C:\Users\Juan\Documents\arquitectura\Semana-2>docker-compose up --build
times="2026-02-14T19:02:55-05:00" Level=warning msg="C:\\Users\\Juan\\Documents\\arquitectura\\Semana-2\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 15/15
   db Pulled
   0c8d55a45c0d Pull complete
   03cf9c337134 Pull complete
   d64dbdf800a2 Pull complete
   2718e764d89a Pull complete
   c50a02fe2f5f Pull complete
   08ace4c9833e Pull complete
   b64f280cf2b Pull complete
   ea5e741a30ce Pull complete
   e60549838c5a Pull complete
   6771773ef0da Pull complete
   049b2237836a Pull complete
   7b098c79e38c Pull complete
[+] Building 29.3s (5/13)
=> [backend internal] load build definition from Dockerfile
=> => transferring dockerfile: 307B
=> [backend internal] load metadata for docker.io/library/maven:3.9-eclipse-temurin-17
=> [backend internal] load metadata for docker.io/library/eclipse-temurin:17-jre
=> [backend internal] load .dockerignore
=> => transferring context: 2B
=> [backend build 1/5] FROM docker.io/library/maven:3.9-eclipse-temurin-17@sha256:e2532c643efbc52e339b24b6a47196b46ec8c269e08da3cd49365db41d9d252
=> => resolve docker.io/library/maven:3.9-eclipse-temurin-17@sha256:e2532c643efbc52e339b24b6a47196b46ec8c269e08da3cd49365db41d9d252
=> => sha256:a3629ac5b9f4e80dc2032439ff2354e73b06a6cc2e08f0035a2d7c001c8b4114 27.26MB / 29.73MB
=> => sha256:e2532c643efbc52e339b24b6a47196b46ec8c269e08da3cd49365db41d9d252 7.94kB / 7.94kB
=> => sha256:4dee09e08c8af0e0c27accfa1fa19a6e63d1da08188fada53b2478b7d60ae08 2.91kB / 2.91kB
=> => sha256:56d763df0b22cf6884d1b80f5ce8a562da72c0f329b13744417b478ec388adf6 10.05kB / 10.05kB
=> [backend internal] load build context
=> => transferring context: 6.61kB
=> [backend stage-1 1/3] FROM docker.io/library/eclipse-temurin:17-jre@sha256:c81561cbc09964364afa412133d8eddf55b83ef5d39a64d0a5f109359bbfccc
=> => resolve docker.io/library/eclipse-temurin:17-jre@sha256:c81561cbc09964364afa412133d8eddf55b83ef5d39a64d0a5f109359bbfccc
=> => sha256:c81561cbc09964364afa412133d8eddf55b83ef5d39a64d0a5f109359bbfccc 8.48kB / 8.48kB
=> => sha256:76f14cf9f2632f5098f855dca88bc3163c9f4a206b37433e9f33897e85ffe8f 1.94kB / 1.94kB
=> => sha256:3531757c955e47cd3d8d9df4a9f31efd1007b31bef47b3e2335831d713d9b0 6.08kB / 6.08kB
=> => sha256:a3629ac5b9f4e80dc2032439ff2354e73b06a6cc2e08f0035a2d7c001c8b4114 27.26MB / 29.73MB
[+] Running 0/1cdd4191aa259e84c498d6407f4f2b9cc91e84f53667da421ceaa3755af5efz 10.49MB / 25.47MB
- Service backend Building
=> => sha256:3531757c955e47cd3d8d9df4a9f31efd1007b31bef47b3e2335831d713d9b0 6.08kB / 6.08kB
7:05 PM 0%
```

Imagen 2. Docker build

La figura muestra la construcción de las imágenes Docker mediante el comando docker-compose up --build, evidenciando que cada servicio (frontend, backend y base de datos) fue compilado y preparado para su despliegue sin errores.

```
C:\Windows\System32\cmd.exe - docker-compose up --build

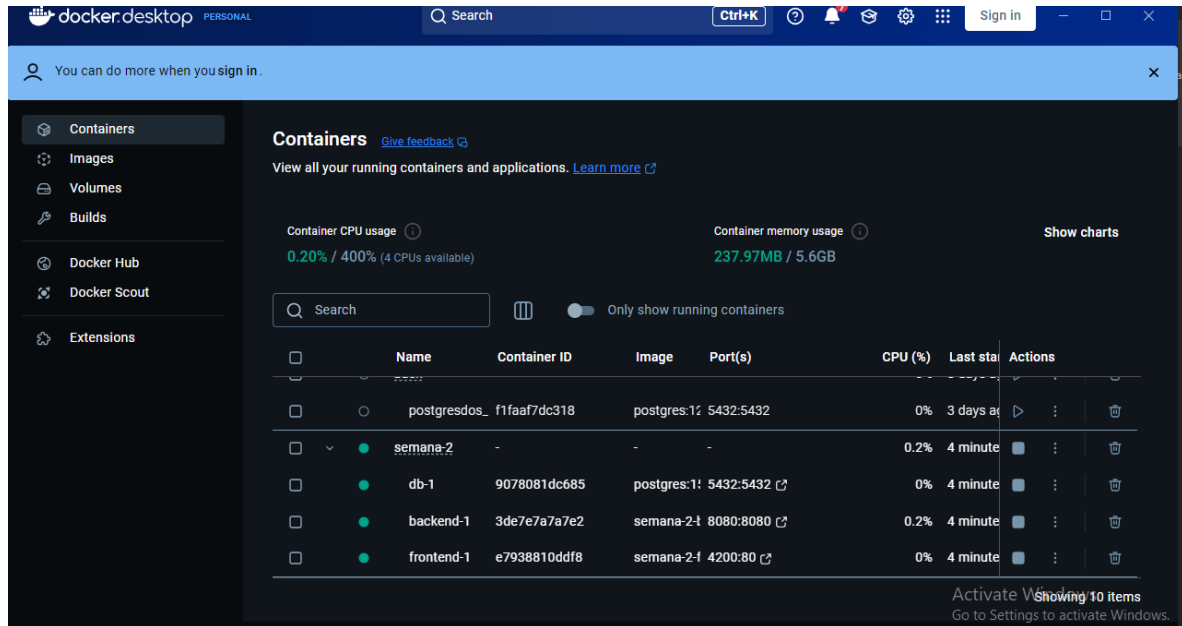
db-1 | PostgreSQL init process complete; ready for start up.
db-1 |
db-1 | 2026-02-15 00:19:35.082 UTC [1] LOG: starting PostgreSQL 15.16 (Debian 15.16-1.pgdg13+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 14.2.0-19) 14.2.0, 64-bit
db-1 | 2026-02-15 00:19:35.092 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
db-1 | 2026-02-15 00:19:35.093 UTC [1] LOG: listening on IPv6 address "::", port 5432
db-1 | 2026-02-15 00:19:35.131 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
db-1 | 2026-02-15 00:19:35.186 UTC [66] LOG: database system was shut down at 2026-02-15 00:19:34 UTC
db-1 | 2026-02-15 00:19:35.234 UTC [1] LOG: database system is ready to accept connections
backend-1 | 2026-02-15T00:19:35.431Z INFO 1 --- [ main ] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
backend-1 | 2026-02-15T00:19:35.486Z INFO 1 --- [ main ] o.apache.catalina.core.StandardService : Starting service [Tomcat]
backend-1 | 2026-02-15T00:19:35.487Z INFO 1 --- [ main ] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.18]
backend-1 | 2026-02-15T00:19:35.722Z INFO 1 --- [ main ] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
backend-1 | 2026-02-15T00:19:35.740Z INFO 1 --- [ main ] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 6203 ms
backend-1 | 2026-02-15T00:19:37.691Z INFO 1 --- [ main ] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context path '/'
backend-1 | 2026-02-15T00:19:37.736Z INFO 1 --- [ main ] com.espagueti.controller.Application : Started Application in 10.769 seconds (process running for 13.689s)
db-1 | 2026-02-15 00:24:35.657 UTC [64] LOG: checkpoint starting: time
db-1 | 2026-02-15 00:24:38.525 UTC [64] LOG: checkpoint complete: wrote 31 buffers (0.2%); 0 WAL file(s) added, 0 removed, 0 recycled; write=2.817 s, sync=0.35 s, total=2.869 s; sync files=13, longest=0.010 s, average=0.003 s; distance=113 kB, estimate=113 kB
frontend-1 | 172.21.0.1 - - [15/Feb/2026:00:24:41 +0000] "GET / HTTP/1.1" 200 520 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36" "-"
frontend-1 | 172.21.0.1 - - [15/Feb/2026:00:24:41 +0000] "GET /main.js HTTP/1.1" 200 281228 "http://localhost:4200/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36" "-"
frontend-1 | 172.21.0.1 - - [15/Feb/2026:00:24:42 +0000] "GET /styles.css HTTP/1.1" 200 54 "http://localhost:4200/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36" "-"
frontend-1 | 2026/02/15 00:24:43 [error] 30#30: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.21.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost:4200", referer: "http://localhost:4200/"
frontend-1 | 172.21.0.1 - - [15/Feb/2026:00:24:43 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost:4200/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36" "-"
```

Imagen 3. Contenedor exitoso

En la figura se visualiza la creación e inicio exitoso de los contenedores correspondientes a cada módulo del sistema, confirmando que los servicios están listos para la ejecución y prueba funcional.

### 1.3 Verificación de Servicios en Ejecución

Una vez finalizada la construcción, se verificó que los tres contenedores se encontraran en ejecución y sin errores.



*Imagen 4. Muestra de contenedor e imágenes*

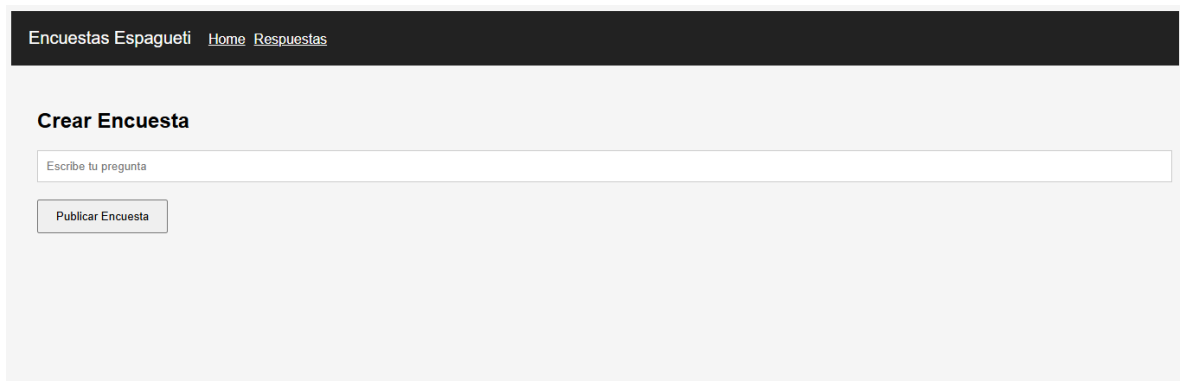
La imagen evidencia la ejecución simultánea de los tres contenedores (frontend, backend y base de datos), mostrando que se encuentran activos y sin errores en la terminal, lo que garantiza un entorno estable para realizar pruebas funcionales.

### Actividad 1.2 – Pruebas Funcionales

Con el entorno correctamente desplegado, se procedió a ejecutar los casos de prueba definidos en la guía del taller.

Código	Descripción	Resultado Esperado	Resultado Obtenido	Estado
CP-01	Crear una encuesta válida	Encuesta creada y redirección al detalle	La encuesta fue creada correctamente sin presentar errores	Exitoso
CP-02	Crear encuesta con texto menor a 3 caracteres	Mostrar mensaje de validación	El sistema impidió la creación y mostró mensaje de error	Exitoso
CP-03	Votar "SI" en una encuesta	Incremento del contador de votos SI	El contador incrementó correctamente en cada clic	Exitoso
CP-04	Votar "NO" en una encuesta	Incremento del contador de votos NO	El contador incrementó correctamente en cada clic	Exitoso

CP-05	Listar todas las encuestas	Mostrar listado completo	Se visualizó correctamente la lista de encuestas creadas	Exitoso
CP-06	Probar SQL Injection 'OR '1'='1	El sistema no debería permitir manipulación de la consulta	El sistema generó un error y no permitió la creación de la encuesta	Parcialmente Exitoso



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

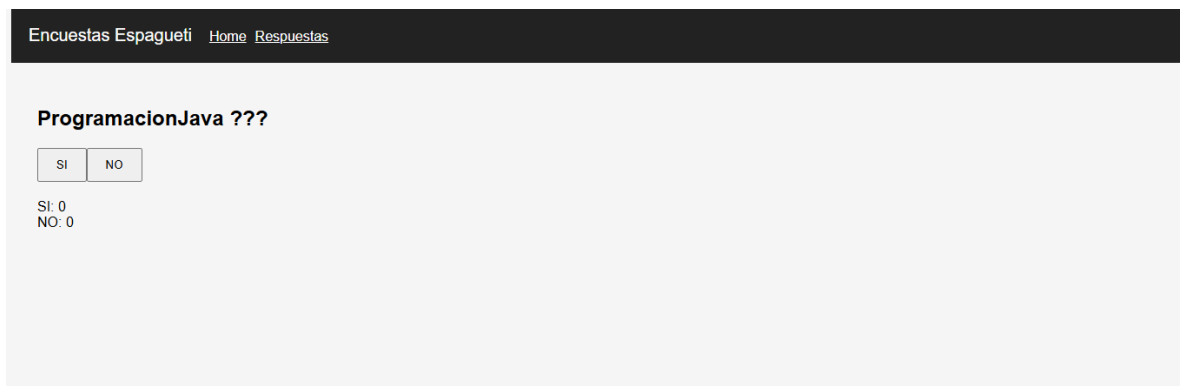
### Crear Encuesta

Escribe tu pregunta

Publicar Encuesta

*Imagen 5. CP-01*

La figura ilustra la creación de una encuesta válida dentro del sistema, mostrando que el registro se realiza correctamente y el usuario es redirigido al detalle de la encuesta, cumpliendo con los requerimientos funcionales esperados.



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

### ProgramacionJava ???

SI NO

SI: 0  
NO: 0

*Imagen 6. CP-01*

Luego de ser redirigido, en esta imagen se observa el detalle de la encuesta recién creada evidenciando que la información se presenta correctamente en la interfaz de usuario.

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

### Crear Encuesta

i

Escribe algo mas largo

Respuestas automáticas:

This screenshot shows the 'Crear Encuesta' (Create Survey) form. The title input field contains the character 'i', which is less than the required three characters. A red message 'Escribe algo mas largo' (Write something longer) is displayed below the input field. The 'Publicar Encuesta' (Publish Survey) button is visible, along with the 'Respuestas automáticas' (Automatic responses) section containing 'SI' and 'NO' buttons.

*Imagen 7. CP-02*

La figura muestra el mensaje de validación que impide crear una encuesta con menos de tres caracteres, asegurando que el sistema cumpla con las reglas de validación establecidas y prevenga la entrada de datos incorrectos.

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

### ProgramacionJava ???

SI: 4  
NO: 1

This screenshot shows the 'ProgramacionJava ???' survey. The 'SI' (Yes) button is selected, and the vote counter has been updated to 'SI: 4' and 'NO: 1'. The 'NO' button is also visible.

*Imagen 8. CP-03*

La imagen evidencia el incremento del contador de votos “SI” al interactuar con la encuesta, confirmando que el sistema registra correctamente la participación de los usuarios y actualiza los resultados en tiempo real.

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

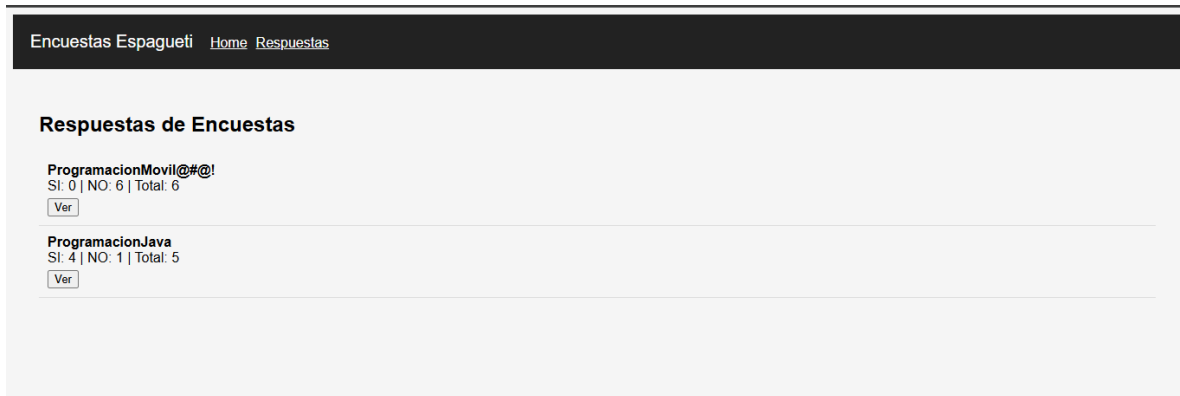
### ProgramacionMovil@#@! ???

SI: 0  
NO: 6

This screenshot shows the 'ProgramacionMovil@#@! ???' survey. The 'SI' (Yes) button is selected, and the vote counter has been updated to 'SI: 0' and 'NO: 6'. The 'NO' button is also visible.

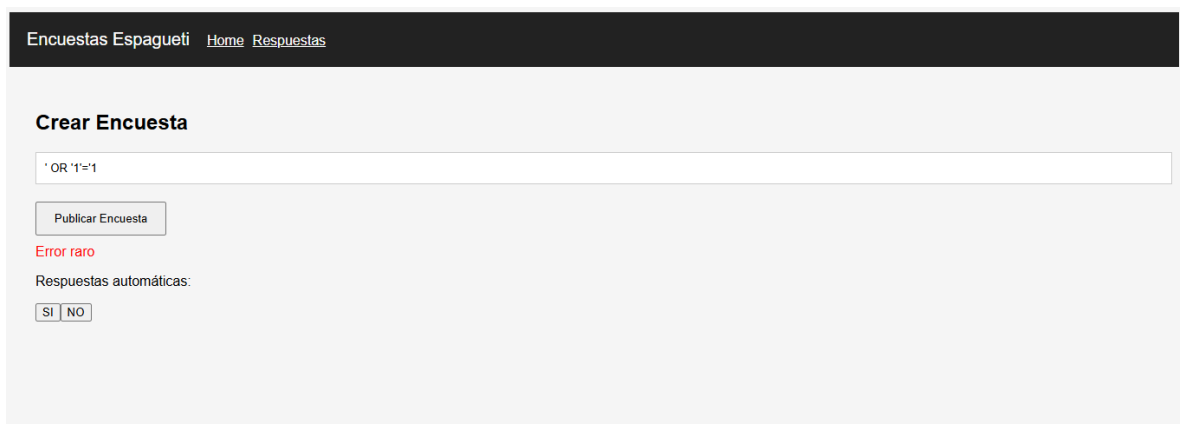
### Imagen 9. CP-04

La figura muestra el incremento del contador de votos “NO”, demostrando que el sistema maneja de manera correcta las opciones de respuesta y mantiene la consistencia de los datos durante la votación.



### Imagen 10. CP-05

En la imagen se visualiza el listado completo de todas las encuestas creadas, evidenciando que el sistema permite consultar y presentar la información de manera organizada, facilitando la navegación y gestión de las encuestas.



### Imagen 11. CP-06

La figura muestra la respuesta del sistema ante un intento de inyección SQL. El sistema bloqueó la creación de la encuesta, pero generó un error inesperado, evidenciando manejo parcial de la entrada.

## **Conclusiones de la Fase 1**

El entorno fue desplegado exitosamente mediante Docker Compose, verificando la correcta ejecución de los tres servicios principales del sistema.

Las pruebas funcionales demostraron que el sistema cumple con las funcionalidades básicas de creación de encuestas, registro de votos y visualización de resultados. Sin embargo, durante las pruebas de seguridad se identificaron comportamientos que sugieren posibles vulnerabilidades, lo cual será abordado en la fase de análisis de anti-patrones.

Esta fase permitió comprender el funcionamiento general del sistema y sentar las bases para el análisis arquitectónico posterior.