PROCESO
MÓDULO
NOMBRE DEL FORMATO
INFORME DE ACTIVIDADES SEMANALES

CLASIFICACIÓN DE LA INFORMACIÓN		
Pública	Pública Clasificada	Pública Reservada
	X	

PROYECTO FONDO EMPRENDER SENA

SEMANA DEL 24 DE FEBRERO AL 2 DE MARZO DE 2026

Presentado por:

NELSON RICARDO GÓMEZ MARTÍNEZ

OFICINA DE SISTEMAS DIRECCIÓN GENERAL

SENA

Bogotá DC

2026

ENTREGABLES:

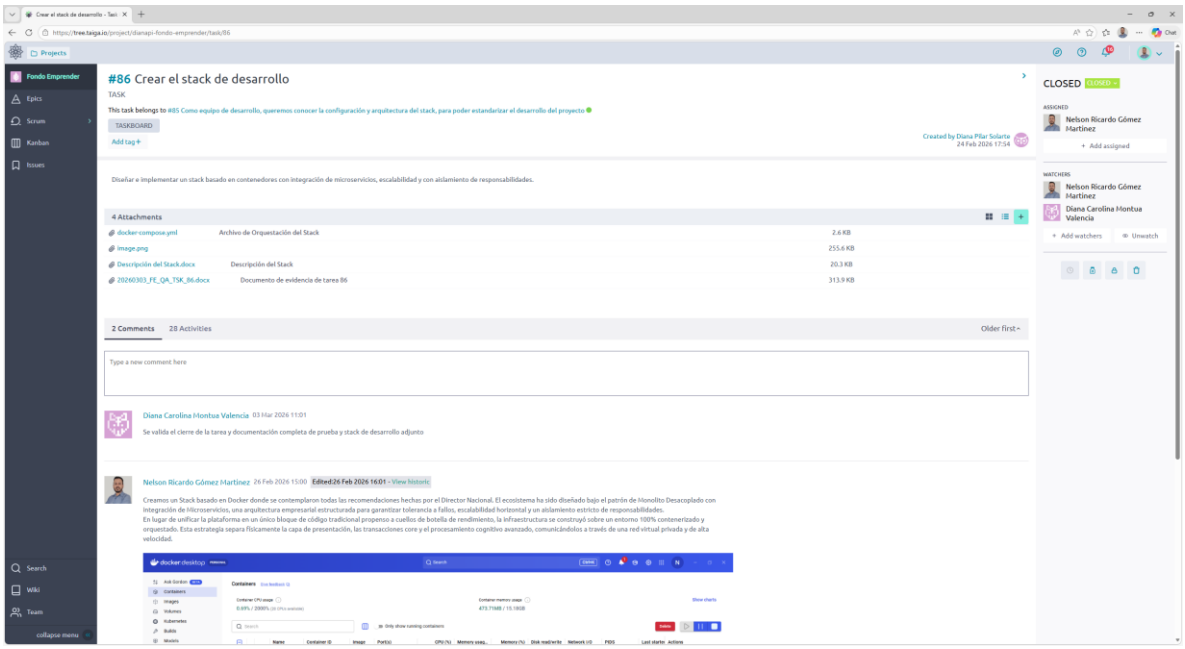
→ Tarea #86: Crear el stack de desarrollo

Creamos un Stack basado en Docker donde se contemplaron todas las recomendaciones hechas por el Director Nacional. El ecosistema ha sido diseñado bajo el patrón de Monolito Desacoplado con integración de Microservicios, una arquitectura empresarial estructurada para garantizar tolerancia a fallos, escalabilidad horizontal y un aislamiento estricto de responsabilidades.
En lugar de unificar la plataforma en un único bloque de código tradicional propenso a cuellos de botella de rendimiento, la infraestructura se construyó sobre un entorno 100% contenerizado y orquestado. Esta estrategia separa físicamente la capa de presentación, las transacciones core y el procesamiento cognitivo avanzado, comunicándolos a través de una red virtual privada y de alta velocidad.

EVIDENCIAS DEL ENTREGABLE

A continuación, se puede observar las pruebas referentes a los entregables:

[Crear el stack de desarrollo - Task #86 - Fondo Emprender]



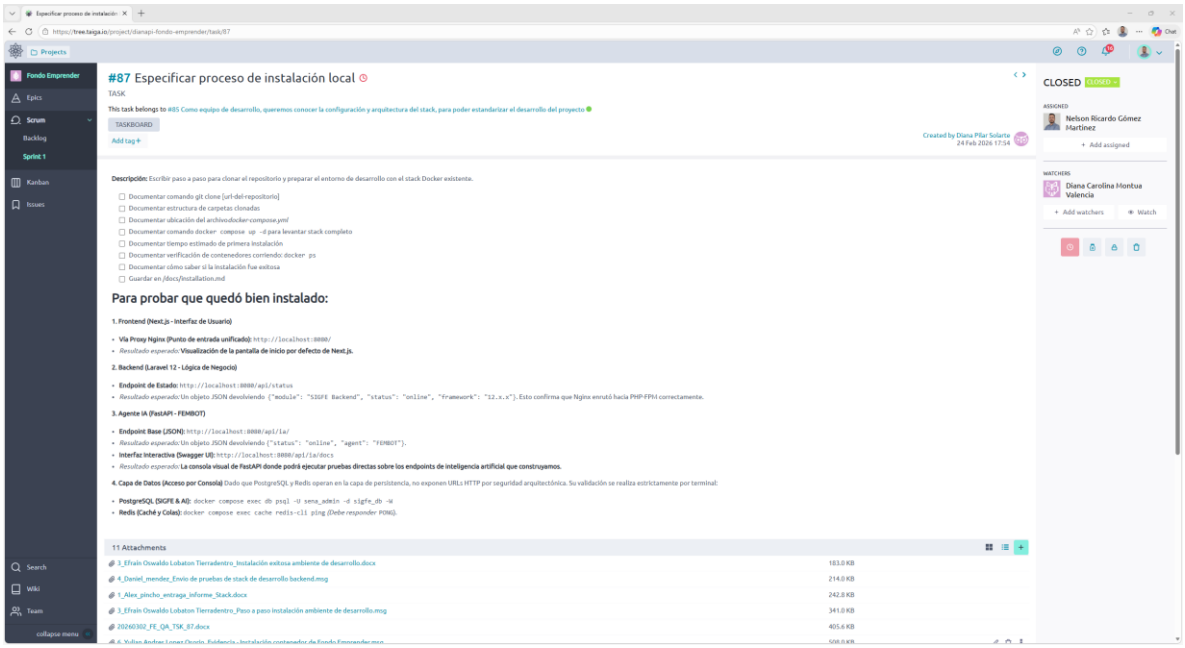
→ Tarea # 87: Especificar proceso de instalación local del Stack

La tarea consistió en documentar de manera detallada el proceso de instalación y verificación de un entorno de desarrollo basado en Docker, comenzando con la clonación del repositorio mediante git clone, la descripción de la estructura de carpetas y la ubicación del archivo docker-compose.yml. Se registraron los comandos necesarios para levantar el stack completo con docker compose up -d, el tiempo estimado de instalación inicial y la verificación de contenedores en ejecución con docker ps, además de los criterios para confirmar que la instalación fue exitosa. Finalmente, se incluyeron pruebas de funcionamiento para cada componente: frontend en Next.js accesible vía Nginx en http://localhost:8080/, backend en Laravel 12 con el endpoint /api/status, agente de IA en FastAPI con sus endpoints y Swagger UI, y validaciones de la capa de datos en PostgreSQL y Redis por consola

EVIDENCIAS DEL ENTREGABLE

A continuación, se puede observar las pruebas referentes a los entregables:

Especificar proceso de instalación local - Task #87 - Fondo Emprender

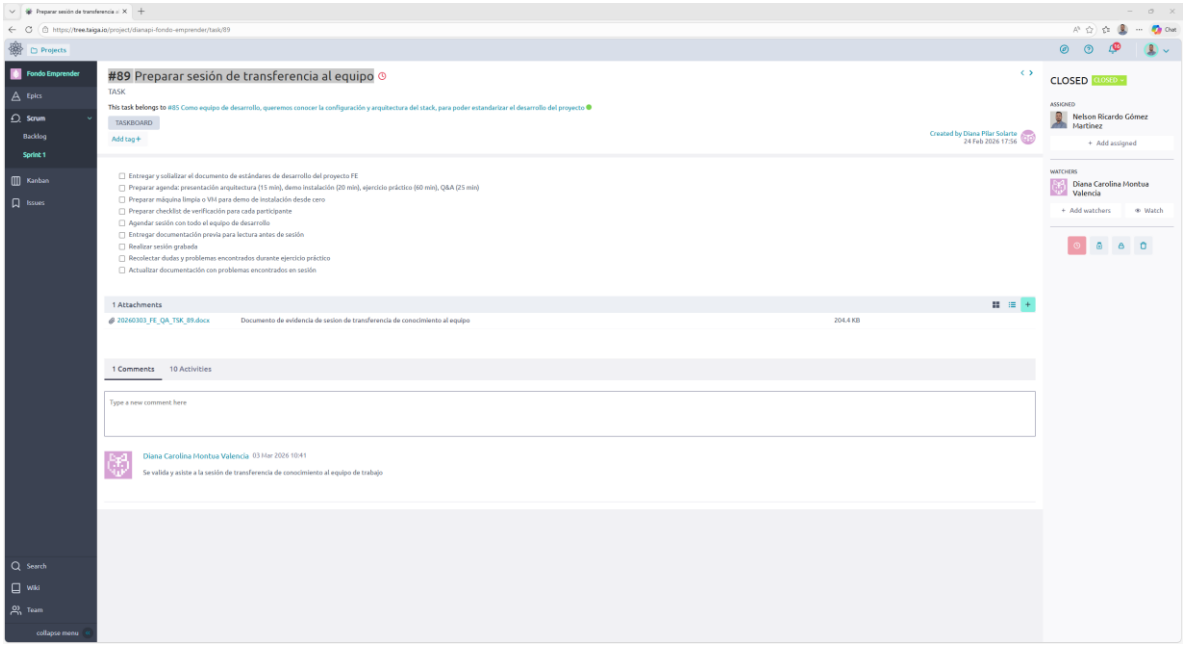


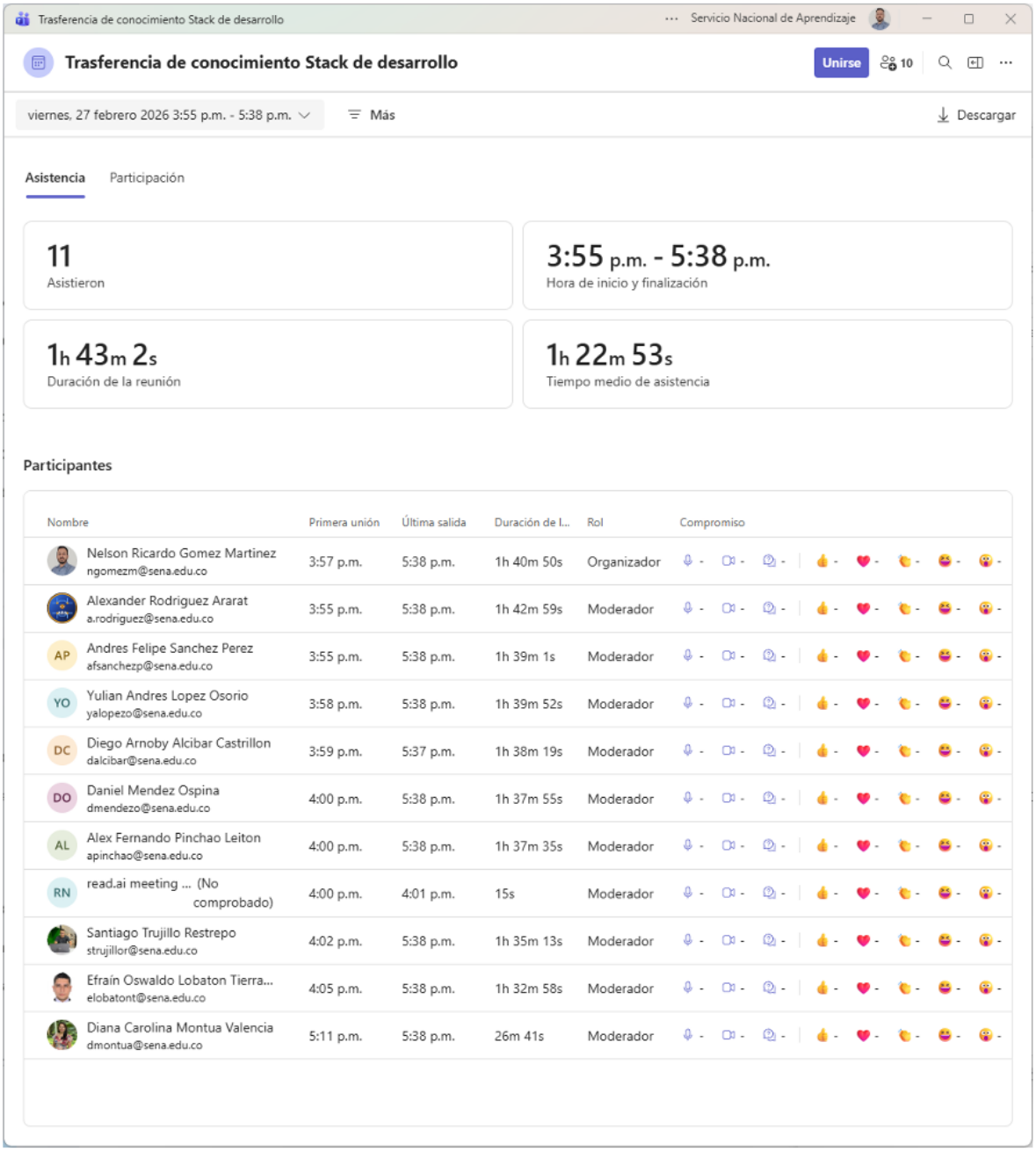
→ Tarea #89: Preparar sesión de transferencia de conocimiento al equipo

La tarea #89 consistió en preparar y ejecutar una sesión de transferencia de conocimiento al equipo de desarrollo, con el objetivo de socializar la configuración y arquitectura del stack y estandarizar prácticas de trabajo. Para ello se organizó una agenda con bloques definidos: presentación de arquitectura (15 min), demo de instalación en una máquina limpia o VM (20 min), ejercicio práctico guiado (60 min) y espacio de preguntas y respuestas (25 min). Se elaboró un checklist de verificación para cada participante, se entregó documentación previa para lectura, se agendó la sesión con todo el equipo y se grabó la jornada para referencia futura. Finalmente, se recolectaron dudas y problemas surgidos durante el ejercicio práctico, asegurando que el equipo contara con insumos claros para continuar con el desarrollo bajo estándares comunes.

EVIDENCIAS DEL ENTREGABLE

A continuación, se puede observar las pruebas referentes a los entregables:





Trasferencia de conocimiento Stack de desarrollo

Servicio Nacional de Aprendizaje

Trasferencia de conocimiento Stack de desarrollo

Unirse 10

viernes, 27 febrero 2026 3:55 p.m. - 5:38 p.m. | Más | Descargar

Asistencia | Participación

11
Asistieron

3:55 p.m. - 5:38 p.m.
Hora de inicio y finalización

1h 43m 2s
Duración de la reunión

1h 22m 53s
Tiempo medio de asistencia

Participantes

Nombre	Primera unión	Última salida	Duración de l...	Rol	Compromiso
Nelson Ricardo Gomez Martinez ngomezm@sena.edu.co	3:57 p.m.	5:38 p.m.	1h 40m 50s	Organizador	
Alexander Rodriguez Ararat a.rodriguez@sena.edu.co	3:55 p.m.	5:38 p.m.	1h 42m 59s	Moderador	
AP Andres Felipe Sanchez Perez afsanchezp@sena.edu.co	3:55 p.m.	5:38 p.m.	1h 39m 1s	Moderador	
YO Yulian Andres Lopez Osorio yalopezo@sena.edu.co	3:58 p.m.	5:38 p.m.	1h 39m 52s	Moderador	
DC Diego Arnoby Alcibar Castrillon dalcibar@sena.edu.co	3:59 p.m.	5:37 p.m.	1h 38m 19s	Moderador	
DO Daniel Mendez Ospina dmendezo@sena.edu.co	4:00 p.m.	5:38 p.m.	1h 37m 55s	Moderador	
AL Alex Fernando Pinchao Leiton apinchao@sena.edu.co	4:00 p.m.	5:38 p.m.	1h 37m 35s	Moderador	
RN read.ai meeting ... (No comprobado)	4:00 p.m.	4:01 p.m.	15s	Moderador	
Santiago Trujillo Restrepo strujillor@sena.edu.co	4:02 p.m.	5:38 p.m.	1h 35m 13s	Moderador	
Efraín Oswaldo Lobaton Tierra... elobatont@sena.edu.co	4:05 p.m.	5:38 p.m.	1h 32m 58s	Moderador	
Diana Carolina Montua Valencia dmontua@sena.edu.co	5:11 p.m.	5:38 p.m.	26m 41s	Moderador	

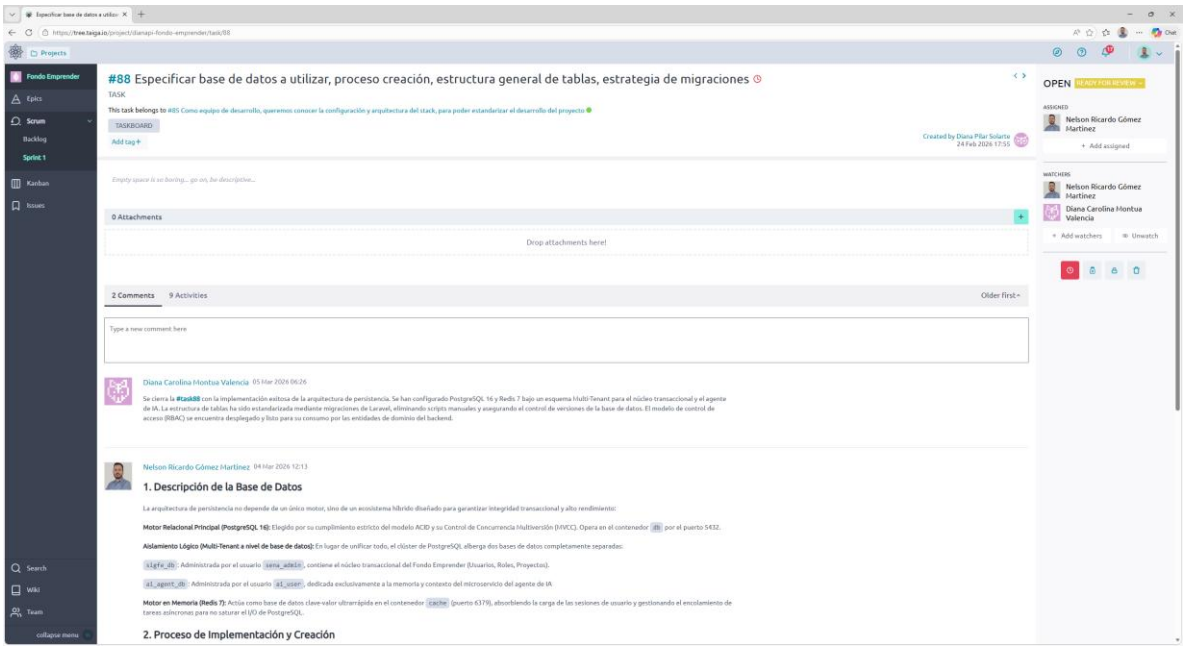
→ Tarea #88: Especificar base de datos a utilizar, proceso creación, estructura general de tablas, estrategia de migraciones

La tarea se centró en definir y desplegar la arquitectura de persistencia del proyecto, especificando los motores de base de datos y la estrategia de migraciones. Se implementó un esquema híbrido con PostgreSQL 16 como motor relacional principal y Redis 7 como base en memoria para sesiones y colas, bajo un modelo Multi-Tenant que separa el núcleo transaccional (sigfe_db) del agente de IA (ai_agent_db). El proceso de creación se realizó mediante Docker Compose y scripts de inicialización, garantizando replicabilidad y desacoplamiento de servicios. La evolución del esquema se estandarizó con migraciones de Laravel, eliminando la dependencia de scripts manuales y asegurando trazabilidad y control de versiones. El estado actual incluye la implementación del modelo de Control de Acceso Basado en Roles (RBAC) con tablas de roles, permisos y relaciones pivote, lo que consolida la seguridad y prepara la base para mapear entidades de dominio en la arquitectura hexagonal.

EVIDENCIAS DEL ENTREGABLE

A continuación, se puede observar las pruebas referentes a los entregables:

Especificar base de datos a utilizar, proceso creación, estructura general de tablas, estrategia de migraciones - Task #88 - Fondo Emprender



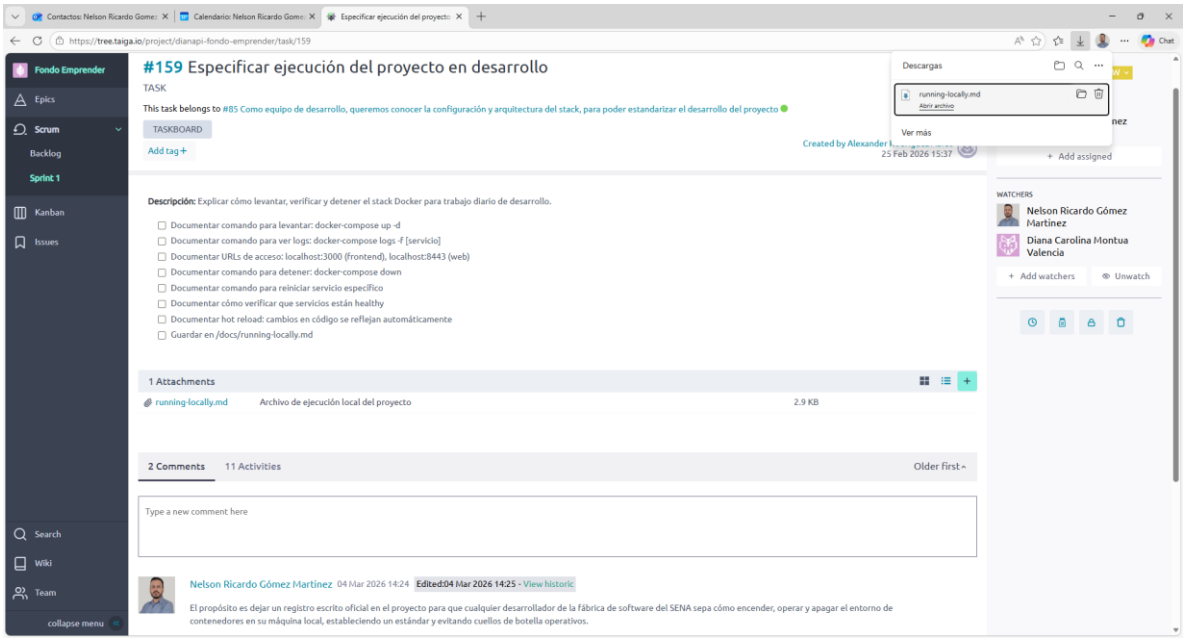
→ Tarea #159: Especificar ejecución del proyecto en desarrollo

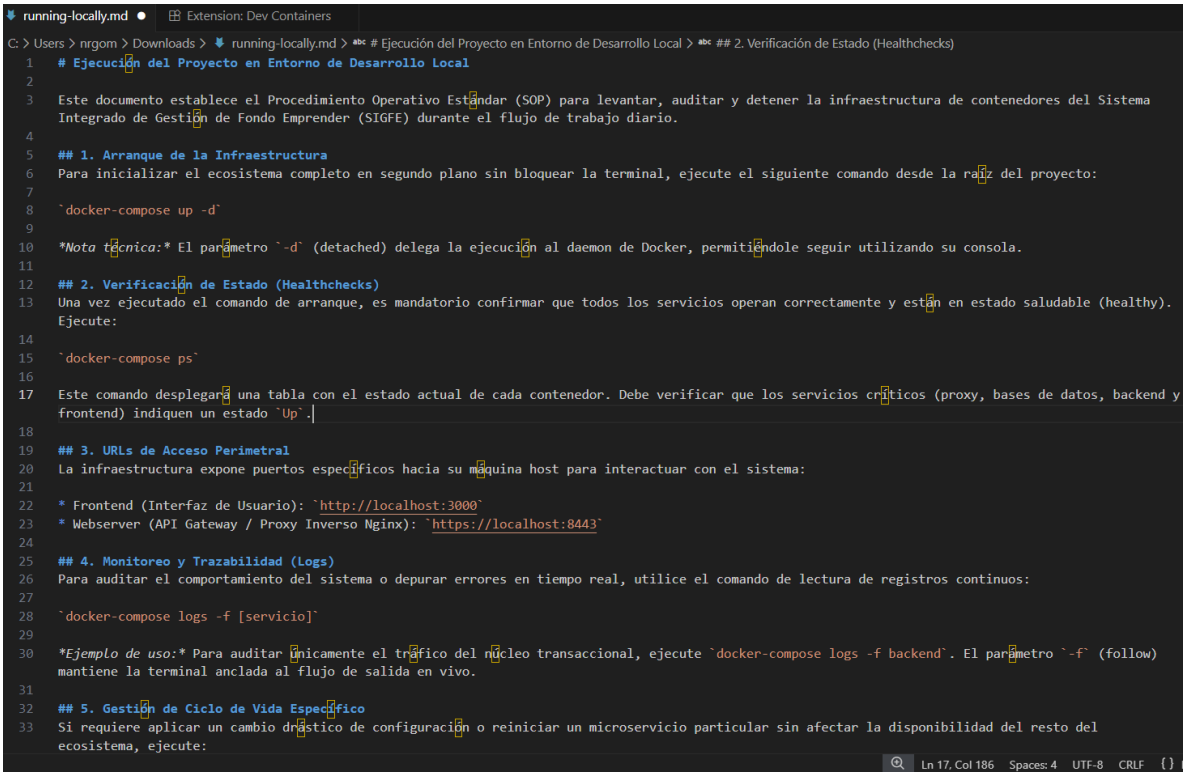
Para trabajar diariamente con el stack de desarrollo en Docker, es necesario conocer los comandos básicos y la forma de verificar que todo funcione correctamente. El proyecto se levanta con docker-compose up -d, lo que inicia los servicios en segundo plano. Para inspeccionar la salida de un servicio específico se utiliza docker-compose logs -f [servicio], lo que permite seguir los registros en tiempo real. Los puntos de acceso principales son http://localhost:3000 para el frontend y https://localhost:8443 para la aplicación web. Cuando se requiere detener todo el entorno, se ejecuta docker-compose down, y si solo se necesita reiniciar un servicio puntual, se usa docker-compose restart [servicio]. La verificación del estado de los contenedores se realiza con docker ps y observando que los servicios aparezcan como healthy. Además, el proyecto está configurado con hot reload, lo que significa que cualquier cambio en el código se refleja automáticamente sin necesidad de reiniciar manualmente los contenedores. Toda esta información debe quedar registrada en el archivo /docs/running-locally.md para estandarizar el proceso dentro del equipo de desarrollo.

EVIDENCIAS DEL ENTREGABLE

A continuación, se puede observar las pruebas referentes a los entregables:

[Especificar ejecución del proyecto en desarrollo - Task #159 - Fondo Emprender]




running-locally.md
Extension: Dev Containers
C: > Users > nrgom > Downloads > running-locally.md > # Ejecución del Proyecto en Entorno de Desarrollo Local > ## 2. Verificación de Estado (Healthchecks)
Ejecución del Proyecto en Entorno de Desarrollo Local
Este documento establece el Procedimiento Operativo Estándar (SOP) para levantar, auditar y detener la infraestructura de contenedores del Sistema Integrado de Gestión de Fondo Emprender (SIGFE) durante el flujo de trabajo diario.
1. Arranque de la Infraestructura
Para inicializar el ecosistema completo en segundo plano sin bloquear la terminal, ejecute el siguiente comando desde la raíz del proyecto:
`docker-compose up -d`
Nota técnica: El parámetro `-d` (detached) delega la ejecución al daemon de Docker, permitiéndole seguir utilizando su consola.
2. Verificación de Estado (Healthchecks)
Una vez ejecutado el comando de arranque, es mandatorio confirmar que todos los servicios operan correctamente y están en estado saludable (healthy). Ejecute:
`docker-compose ps`
Este comando desplegará una tabla con el estado actual de cada contenedor. Debe verificar que los servicios críticos (proxy, bases de datos, backend y frontend) indiquen un estado `Up`.
3. URLs de Acceso Perimetral
La infraestructura expone puertos específicos hacia su máquina host para interactuar con el sistema:
* Frontend (Interfaz de Usuario): `http://localhost:3000`
* Webserver (API Gateway / Proxy Inverso Nginx): `https://localhost:8443`
4. Monitoreo y Trazabilidad (Logs)
Para auditar el comportamiento del sistema o depurar errores en tiempo real, utilice el comando de lectura de registros continuos:
`docker-compose logs -f [servicio]`
Ejemplo de uso: Para auditar únicamente el tráfico del núcleo transaccional, ejecute `docker-compose logs -f backend`. El parámetro `-f` (follow) mantiene la terminal anclada al flujo de salida en vivo.
5. Gestión de Ciclo de Vida Específico
Si requiere aplicar un cambio drástico de configuración o reiniciar un microservicio particular sin afectar la disponibilidad del resto del ecosistema, ejecute:
Ln 17, Col 186
Spaces: 4
UTF-8
CRLF