PROCESO
MÓDULO
NOMBRE DEL FORMATO
INFORME DE ACTIVIDADES SEMANALES
CLASIFICACIÓN DE LA INFORMACIÓN

Pública		Pública Clasificada	X	Pública Reservada	

PROYECTO FONDO EMPRENDER SENA

SEMANA DEL 3 AL 9 DE MARZO DE 2026

Presentado por:

NELSON RICARDO GÓMEZ MARTÍNEZ

OFICINA DE SISTEMAS DIRECCIÓN GENERAL

SENA

Bogotá DC

2026

ENTREGABLES:

→ **Tarea 1: Especificar variables de entorno necesaria.**

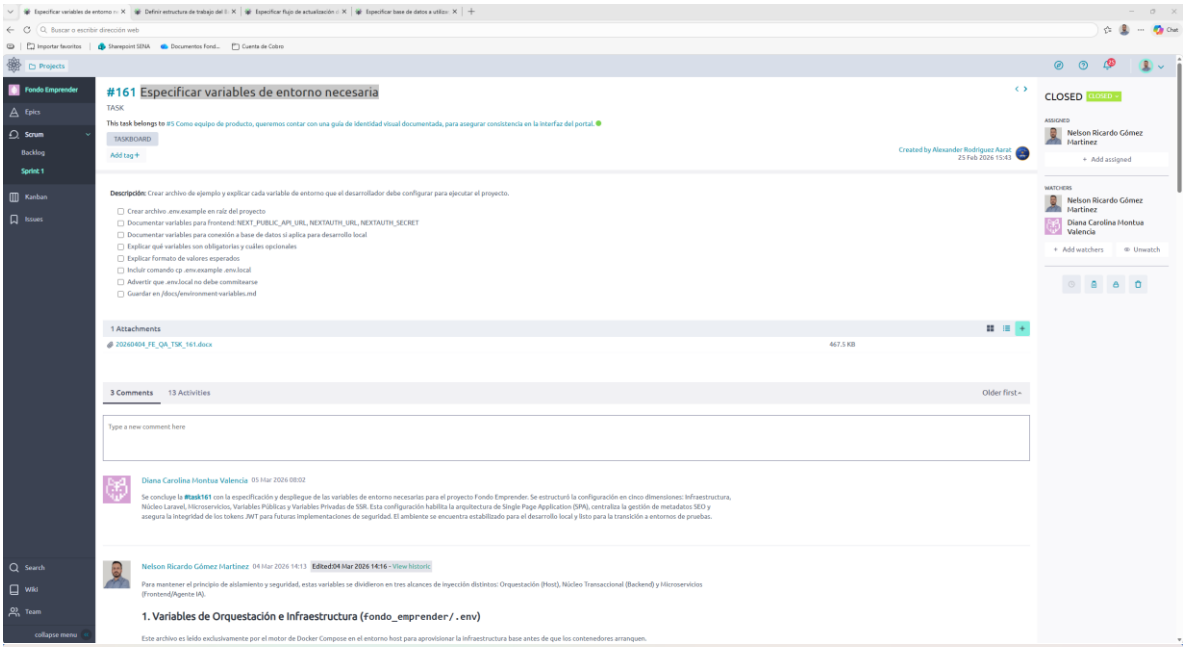
EVIDENCIAS DEL ENTREGABLE

Para abordar la tarea era fundamental establecer una base clara que permitiera a cualquier desarrollador configurar el proyecto sin fricciones. La creación de un archivo .env.example en la raíz del repositorio servirá como plantilla estándar, donde se definan todas las variables requeridas para la ejecución del frontend y la conexión con servicios externos. Este archivo no solo facilitará la instalación inicial, sino que también garantizará que todos los miembros del equipo trabajen bajo las mismas condiciones, evitando inconsistencias y errores derivados de configuraciones locales distintas.

Además, la documentación en /docs/environment-variables.md explica de manera detallada el propósito de cada variable, indicando cuáles son obligatorias y cuáles opcionales, así como el formato esperado de sus valores. Incluir instrucciones prácticas, como el uso del comando cp .env.example .env.local, permitirá a los desarrolladores replicar rápidamente la configuración base en sus entornos locales. Es igualmente importante advertir que el archivo .env.local nunca debe ser versionado en el repositorio, ya que contiene credenciales sensibles o configuraciones específicas de cada máquina. Con esta guía, se asegura un proceso de desarrollo más ordenado, seguro y consistente.

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

Especificar variables de entorno necesaria - Task #161 - Fondo Emprender



#161 Especificar variables de entorno necesaria

TASK

This task belongs to #5 Como equipo de producto, queremos contar con una guía de identidad visual documentada, para asegurar consistencia en la interfaz del portal.

TASKBOARD

Add tag +

Created by Alexander Rodriguez Aarat 25 Feb 2026 15:43

CLOSED

Nelson Ricardo Gómez Martinez

+ Add assigned

Nelson Ricardo Gómez Martinez

Diana Carolina Montua Valencia

+ Add watchers Unwatch

Descripción: Crear archivo de ejemplo y explicar cada variable de entorno que el desarrollador debe configurar para ejecutar el proyecto.

- [ ] Crear archivo .env.example en raíz del proyecto
- [ ] Documentar variables para frontend: NEXT_PUBLIC_API_URL, NEXTAUTH_URL, NEXTAUTH_SECRET
- [ ] Documentar variables para conexión a base de datos si aplica para desarrollo local
- [ ] Explicar qué variables son obligatorias y cuáles opcionales
- [ ] Explicar formato de valores esperados
- [ ] Incluir comando cp .env.example .env.local
- [ ] Advertir que .env.local no debe commitearse
- [ ] Guardar en /docs/environment-variables.md

1 Attachments

20260404_FE_QA_TSK_161.docx 467.5 KB

3 Comments 13 Activities Older first

Type a new comment here

Diana Carolina Montua Valencia 05 Mar 2026 08:02

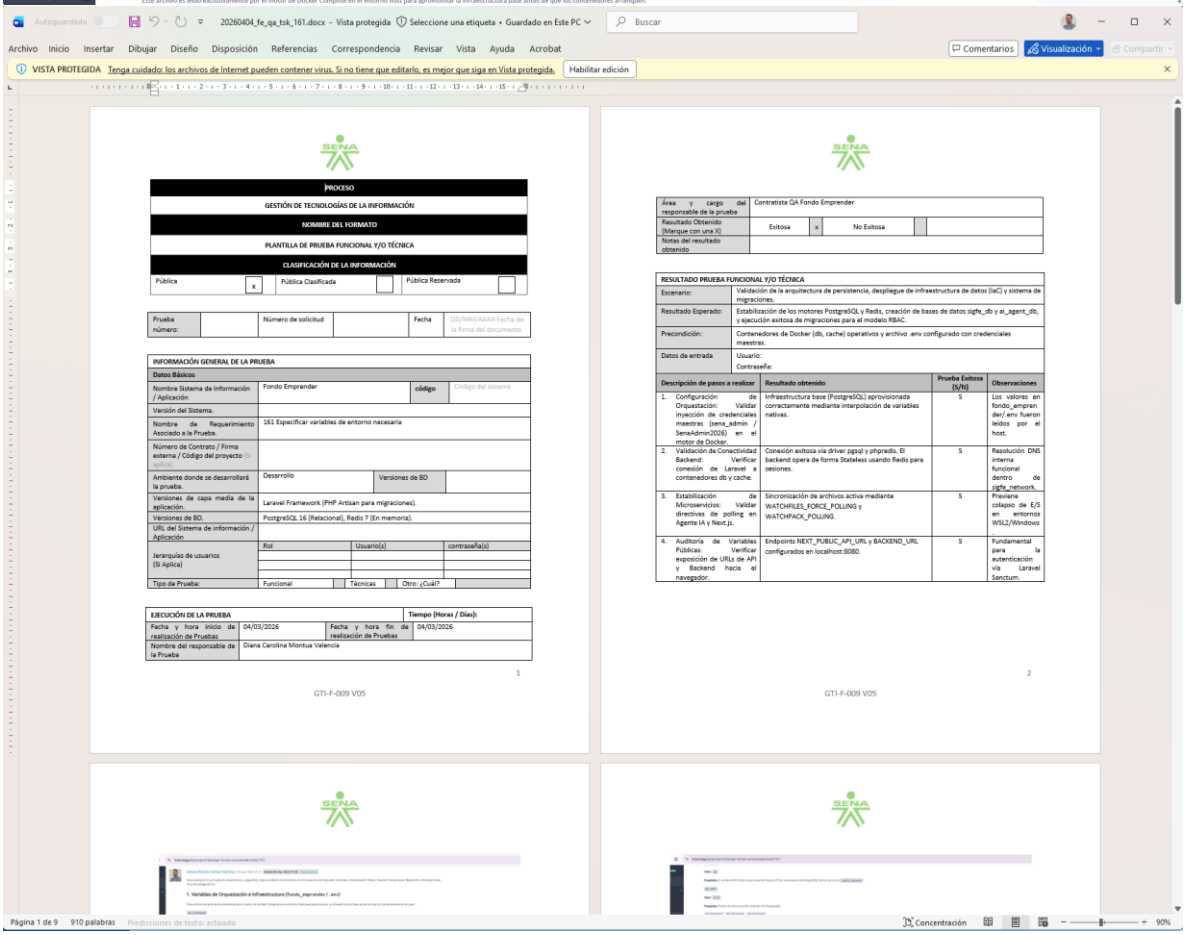
Se concluye la #task161 con la especificación y despliegue de las variables de entorno necesarias para el proyecto Fondo Emprender. Se estructuró la configuración en cinco dimensiones: Infraestructura, Núcleo Laravel, Microservicios, Variables Públicas y Variables Privadas de SSR. Esta configuración habilita la arquitectura de Single Page Application (SPA), centraliza la gestión de metadatos SEO y asegura la integridad de los tokens JWT para futuras implementaciones de seguridad. El ambiente se encuentra estabilizado para el desarrollo local y listo para la transición a entornos de pruebas.

Nelson Ricardo Gómez Martinez 04 Mar 2026 14:13 Edited:04 Mar 2026 14:16 - View historic

Para mantener el principio de aislamiento y seguridad, estas variables se dividieron en tres alcances de inyección distintos: Orquestación (Host), Núcleo Transaccional (Backend) y Microservicios (Frontend/Agente IA).

1. Variables de Orquestación e Infraestructura (fondo_emprender/.env)

Este archivo es leído exclusivamente por el motor de Docker Compose en el entorno host para aprovisionar la infraestructura base antes de que los contenedores arranquen.



20260404_fe_qa_tsk_161.docx - Vista protegida

VISTA PROTEGIDA Tenga cuidado: los archivos de Internet pueden contener virus. Si no tiene que editarlo, es mejor que siga en Vista protegida. Habilitar edición

PROCESO
GESTIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN
NOMBRE DEL FORMATO
PLANTILLA DE PRUEBA FUNCIONAL Y/O TÉCNICA
CLASIFICACIÓN DE LA INFORMACIÓN

Pública	x	Pública Clasificada		Pública Reservada	

Prueba número:		Número de solicitud		Fecha	DD/MM/AAAA Fecha de la firma del documento

INFORMACIÓN GENERAL DE LA PRUEBA			
Datos Básicos			
Nombre Sistema de información / Aplicación	Fondo Emprender	código	Código del sistema
Versión del Sistema.			
Nombre de Requerimiento Asociado a la Prueba.	161 Especificar variables de entorno necesaria		
Número de Contrato / Firma externa / Código del proyecto (Si aplica)			
Ambiente donde se desarrollará la prueba.	Desarrollo	Versiones de BD	
Versiones de capa media de la aplicación.	Laravel Framework (PHP Artisan para migraciones).		
Versiones de BD.	PostgreSQL 16 (Relacional), Redis 7 (En memoria).		
URL del Sistema de información / Aplicación			
Jerarquías de usuarios (Si Aplica)	Rol	Usuario(s)	contraseña(s)
Tipo de Prueba:	Funcional	Técnicas	Otro: ¿Cuál?

EJECUCIÓN DE LA PRUEBA			Tiempo (Horas / Días):
Fecha y hora inicio de realización de Pruebas	04/03/2026	Fecha y hora fin de realización de Pruebas	04/03/2026
Nombre del responsable de la Prueba	Diana Carolina Montua Valencia		

1

GTI-F-009 V05

Área y cargo del responsable de la prueba	Contratista QA Fondo Emprender				
Resultado Obtenido (Marque con una X)	Exitosa	x	No Exitosa		
Notas del resultado obtenido					

RESULTADO PRUEBA FUNCIONAL Y/O TÉCNICA	
Escenario:	Validación de la arquitectura de persistencia, despliegue de infraestructura de datos (IaC) y sistema de migraciones.
Resultado Esperado:	Estabilización de los motores PostgreSQL y Redis, creación de bases de datos sigfe_db y ai_agent_db, y ejecución exitosa de migraciones para el modelo RBAC.
Precondición:	Contenedores de Docker (db, cache) operativos y archivo .env configurado con credenciales maestras.
Datos de entrada	Usuario: Contraseña:

Descripción de pasos a realizar	Resultado obtenido	Prueba Exitosa (S/N)	Observaciones
1. Configuración de Orquestación: Validar inyección de credenciales maestras (sena_admin / SenaAdmin2026) en el motor de Docker.	Infraestructura base (PostgreSQL) aprovisionada correctamente mediante interpolación de variables nativas.	S	Los valores en fondo_emprender/.env fueron leídos por el host.
2. Validación de Conectividad Backend: Verificar conexión de Laravel a contenedores db y cache.	Conexión exitosa vía driver pgsql y phpredis. El backend opera de forma Stateless usando Redis para sesiones.	S	Resolución DNS interna funcional dentro de sigfe_network.
3. Estabilización de Microservicios: Validar directivas de polling en Agente IA y Next.js.	Sincronización de archivos activa mediante WATCHFILES_FORCE_POLLING y WATCHPACK_POLLING.	S	Previene colapso de E/S en entornos WSL2/Windows
4. Auditoría de Variables Públicas: Verificar exposición de URLs de API y Backend hacia el navegador.	Endpoints NEXT_PUBLIC_API_URL y BACKEND_URL configurados en localhost:8080.	S	Fundamental para la autenticación vía Laravel Sanctum.

2

GTI-F-009 V05

Página 1 de 9 910 palabras Predicciones de texto: activado Concentración 90%

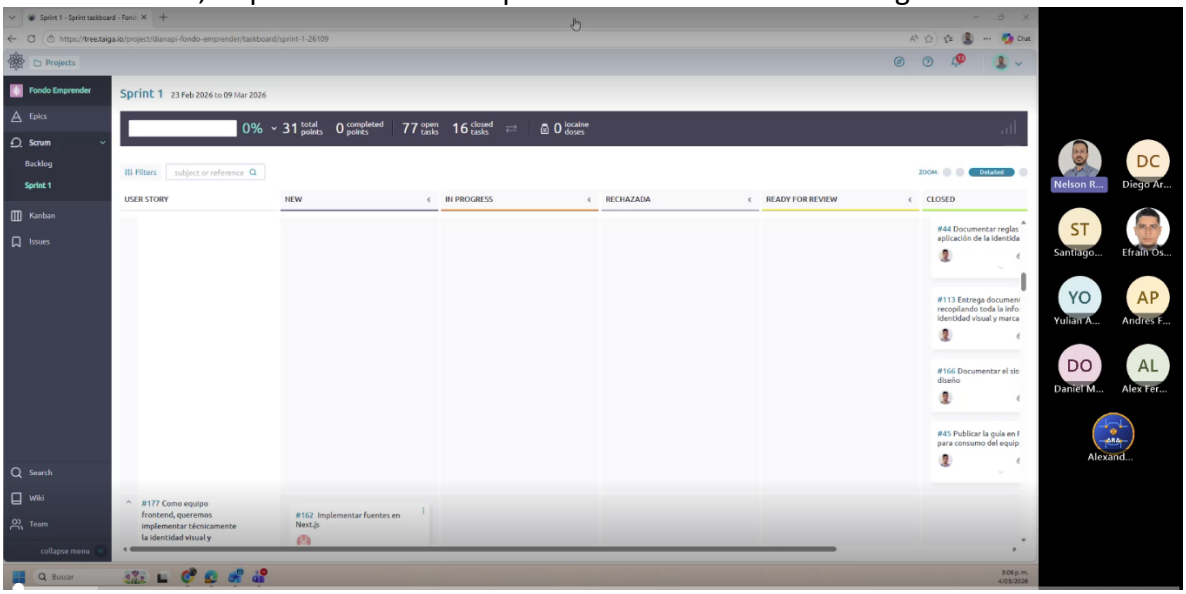
→ Tarea 2: Sesión de seguimiento metodológico del equipo y gestión de tareas en Taiga.

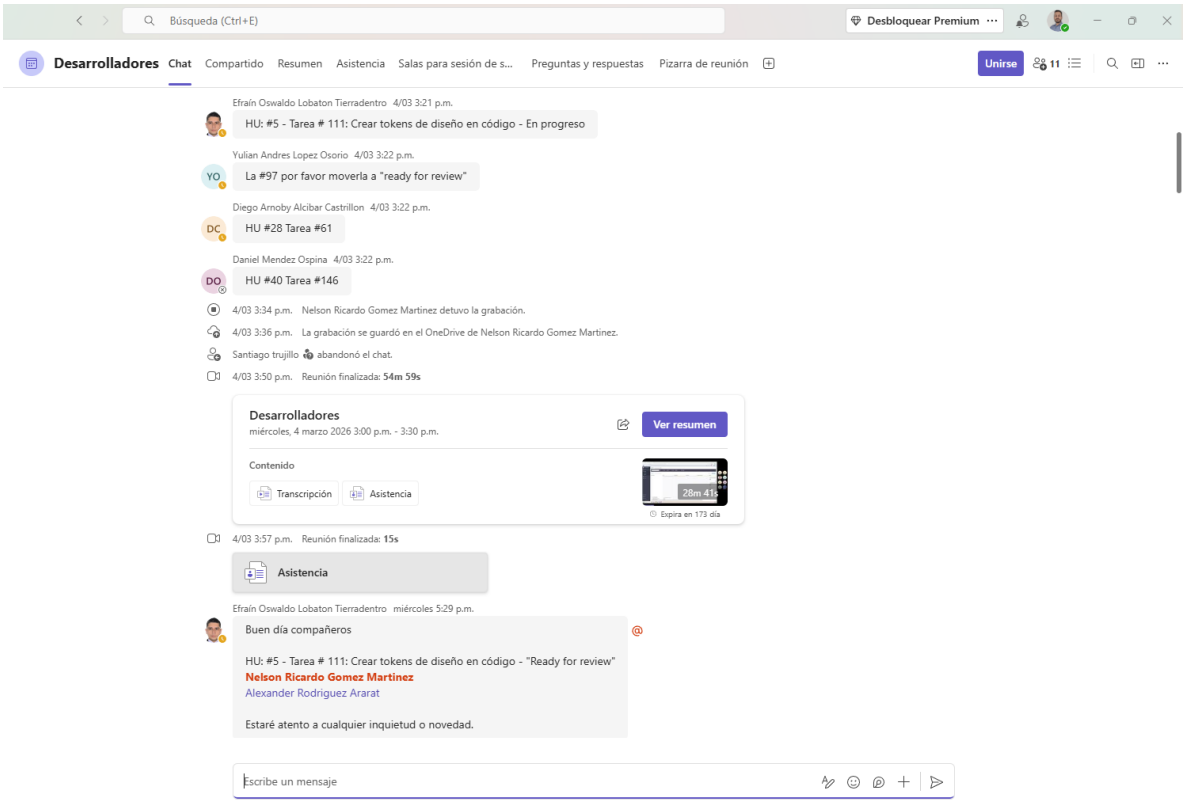
La reunión se desarrolló como un espacio de seguimiento y ajuste a la metodología de trabajo del equipo, particularmente en relación con el uso del tablero Kanban dentro del sprint. Durante la sesión se revisó la forma en que las tareas están siendo registradas y actualizadas en el tablero, destacando la importancia de que cada actividad refleje su estado real, especialmente en la columna “En progreso”. Se observó que actualmente muchas tareas aparecen únicamente en estado creado y no en ejecución, lo cual genera una percepción de baja actividad. Asimismo, se enfatizó que el tablero debe permitir a los directivos visualizar de manera clara el avance del equipo y detectar posibles desequilibrios en el flujo de trabajo.

Adicionalmente, se identificaron algunas limitaciones en la herramienta utilizada para la gestión de tareas (Taiga), que en ciertos casos impiden que los miembros del equipo puedan mover directamente las tareas entre estados. Como medida provisional para fortalecer la disciplina en la actualización de las actividades, se propuso habilitar un canal de comunicación (por ejemplo, WhatsApp) donde los integrantes del equipo informen cuándo inician o finalizan una tarea. También se mencionó la creación de un equipo de prueba denominado Backend Team, con el propósito de experimentar mecanismos de coordinación y consolidar prácticas más efectivas de seguimiento del trabajo.

EVIDENCIAS DEL ENTREGABLE

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





→ Tarea 3: Definir estructura de trabajo del Backend.

Se avanzó en la propuesta de una arquitectura sólida que permita escalar el proyecto y mantener la organización del código de manera clara. Se optó por una Arquitectura Hexagonal orientada por Dominios (DDD - Domain-Driven Design), lo que implica que los módulos no se agrupan por tipo de archivo, sino por Contextos Delimitados (Bounded Contexts). Esta decisión busca que cada parte del sistema tenga independencia y responsabilidades bien definidas, facilitando la evolución del backend sin comprometer la estabilidad general.

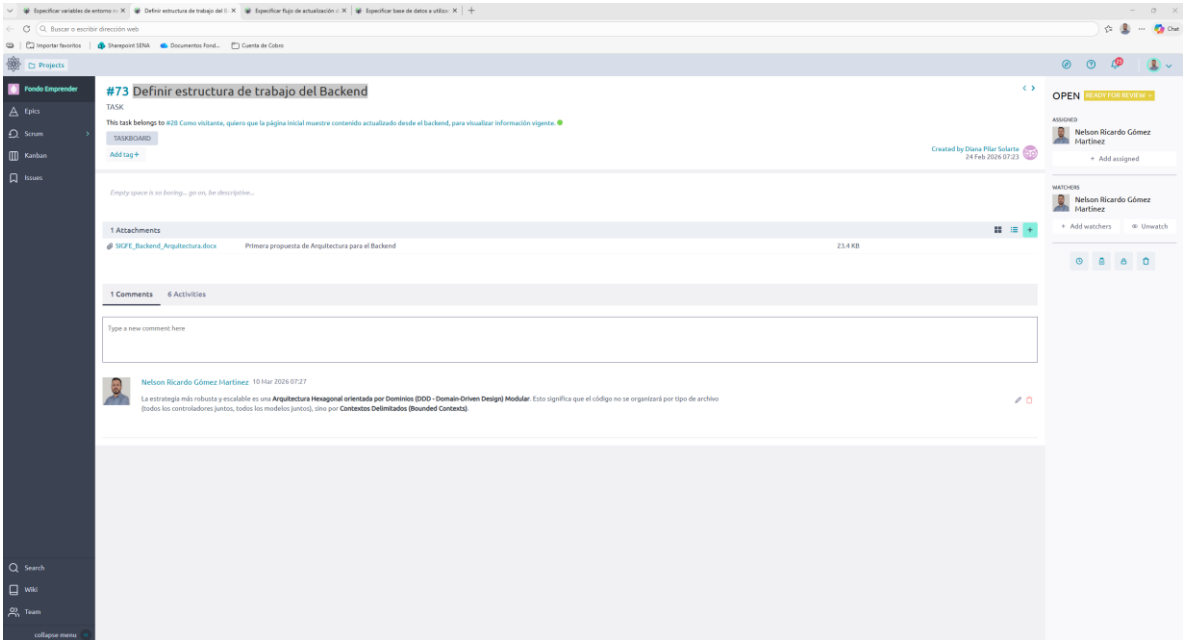
Como evidencia del trabajo, se elaboró el documento SIGFE_Backend_Arquitectura.docx, donde se describe la primera propuesta de organización modular y se detallan los principios que guiarán la implementación. Este enfoque asegura que el backend pueda responder de manera eficiente a las necesidades del frontend, garantizando que la página inicial muestre contenido actualizado y vigente. La estructura planteada también abre la puerta a integrar

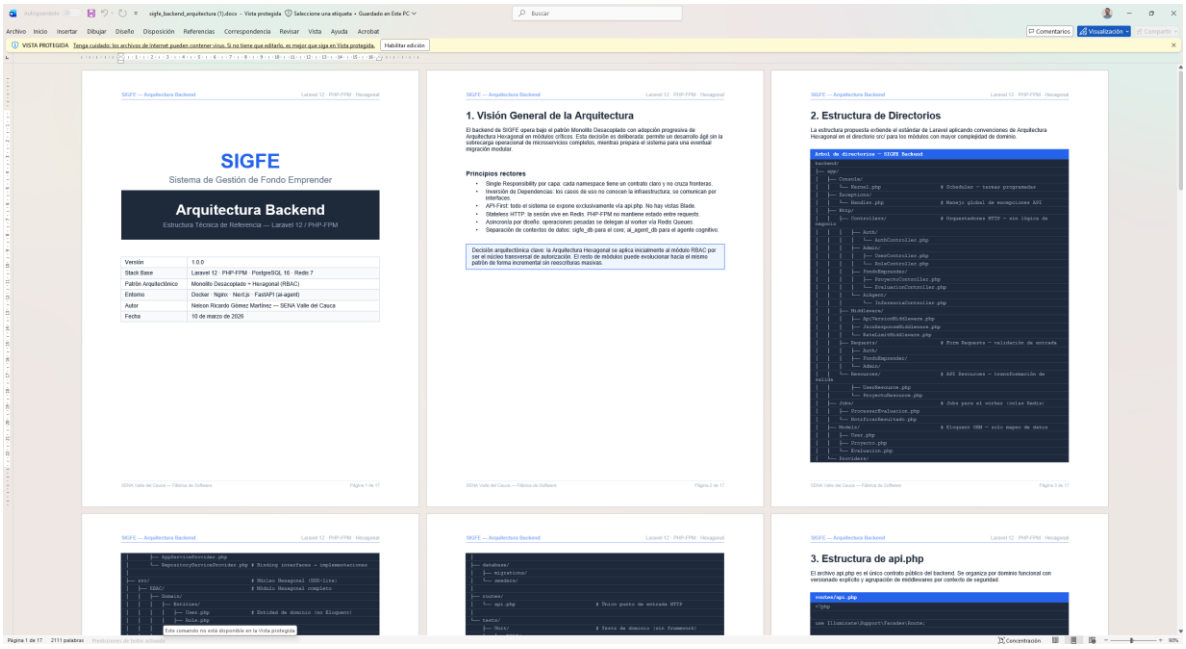
nuevas funcionalidades sin generar deuda técnica, manteniendo la coherencia y escalabilidad del sistema.

EVIDENCIAS DEL ENTREGABLE

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

Definir estructura de trabajo del Backend - Task #73 - Fondo Emprender





→ Tarea 4: Especificar flujo de actualización del stack.

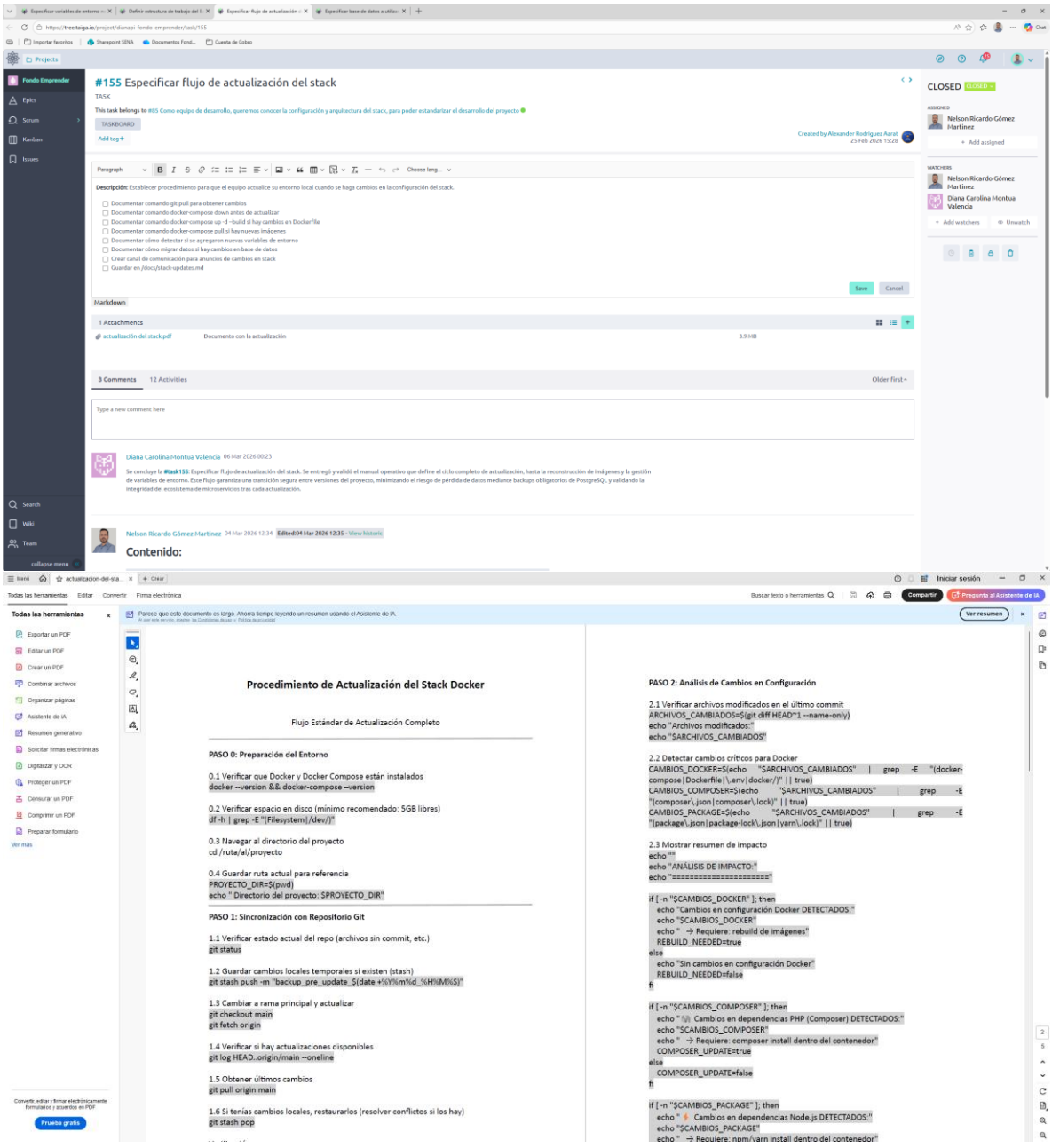
Se definió un procedimiento claro y estandarizado para que el equipo de desarrollo pueda mantener sus entornos locales alineados con los cambios en la configuración del stack. Se documentaron los comandos esenciales, como git pull para obtener las últimas actualizaciones, docker-compose down antes de aplicar cambios, y docker-compose up -d --build en caso de modificaciones en el Dockerfile. También se incluyó el uso de docker-compose pull para actualizar imágenes y se explicó cómo verificar si se agregaron nuevas variables de entorno, así como los pasos necesarios para migrar datos cuando se presentan cambios en la base de datos.

Como evidencia, se entregó el documento actualización del stack.pdf, que consolida el manual operativo y fue validado por el equipo. Este flujo garantiza una transición segura entre versiones, minimizando riesgos mediante la práctica obligatoria de respaldos en PostgreSQL y validando la integridad de los microservicios tras cada actualización. Además, se estableció un canal de comunicación para anunciar cambios en el stack, asegurando que todos los desarrolladores estén informados y puedan aplicar las actualizaciones de manera ordenada y consistente.

EVIDENCIAS DEL ENTREGABLE

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

[Especificar flujo de actualización del stack - Task #155 - Fondo Emprender]



→ Tarea 5: Normalizar base de datos de la tarea #88.

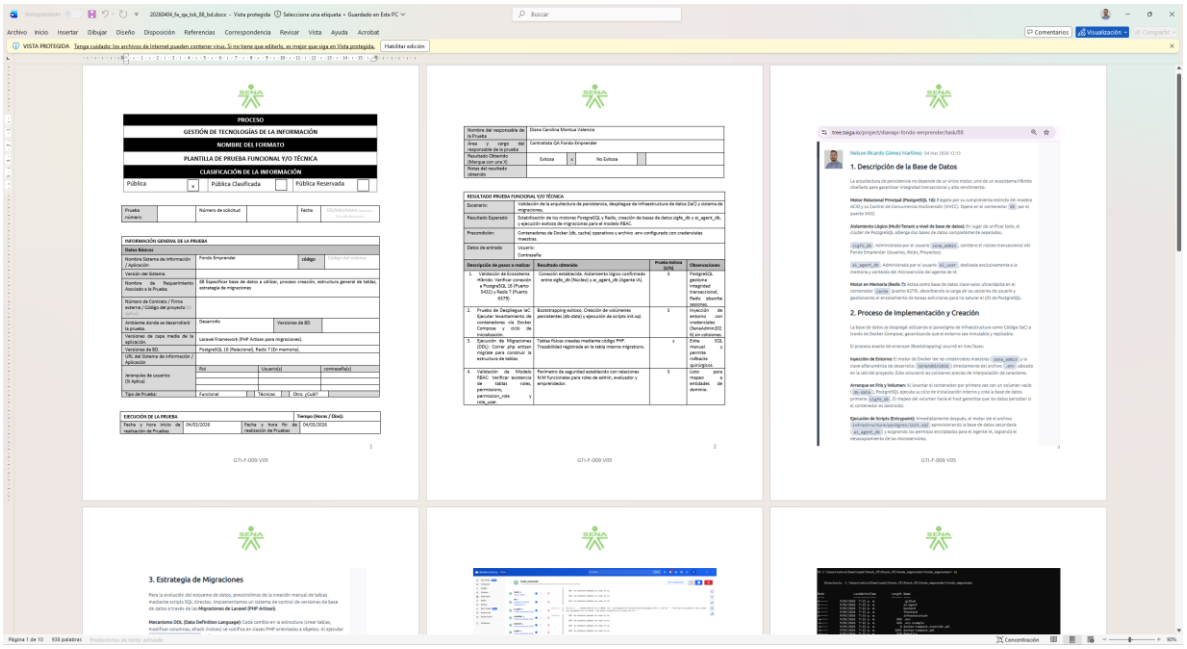
Se definió la arquitectura de persistencia del proyecto con un enfoque híbrido. Se seleccionó PostgreSQL 16 como motor relacional principal, garantizando integridad transaccional mediante MVCC y cumplimiento estricto del modelo ACID. Adicionalmente, se incorporó Redis 7 como base de datos en memoria para gestionar sesiones y tareas asíncronas, optimizando el rendimiento del stack. Se implementó un esquema Multi-Tenant, con dos bases de datos separadas: sigfe_db para el núcleo transaccional del Fondo Emprender y ai_agent_db para el microservicio del agente de IA. Como parte del trabajo, se normalizó el primer modelo de datos, asegurando consistencia y evitando redundancias en las entidades principales.

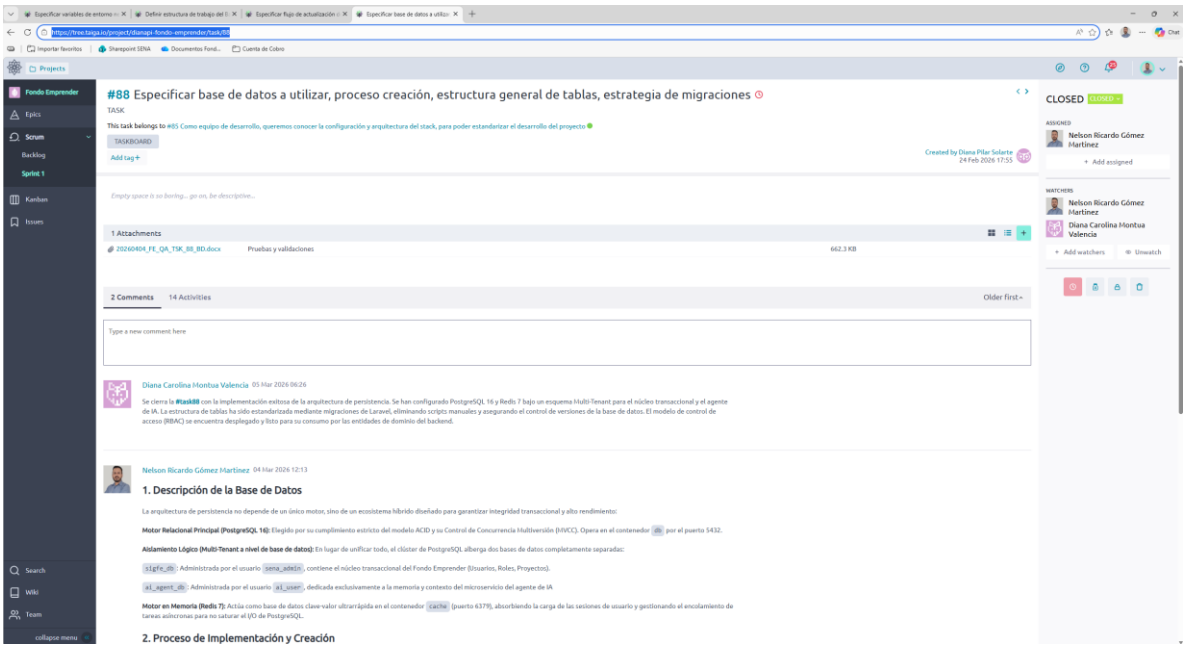
El proceso de creación y configuración se documentó en detalle, incluyendo la inicialización de contenedores, usuarios y permisos. La estructura general de tablas se estandarizó con migraciones de Laravel, eliminando la necesidad de scripts manuales y permitiendo control de versiones sobre la base de datos. Se desplegó el modelo de control de acceso (RBAC), listo para ser consumido por las entidades de dominio del backend. La estrategia de migraciones asegura que cualquier cambio en el esquema pueda aplicarse de forma ordenada, con soporte para rollback y validación de integridad. Con esta implementación, el equipo cuenta con un entorno robusto y escalable, alineado con la arquitectura del stack y preparado para futuras evoluciones del sistema.

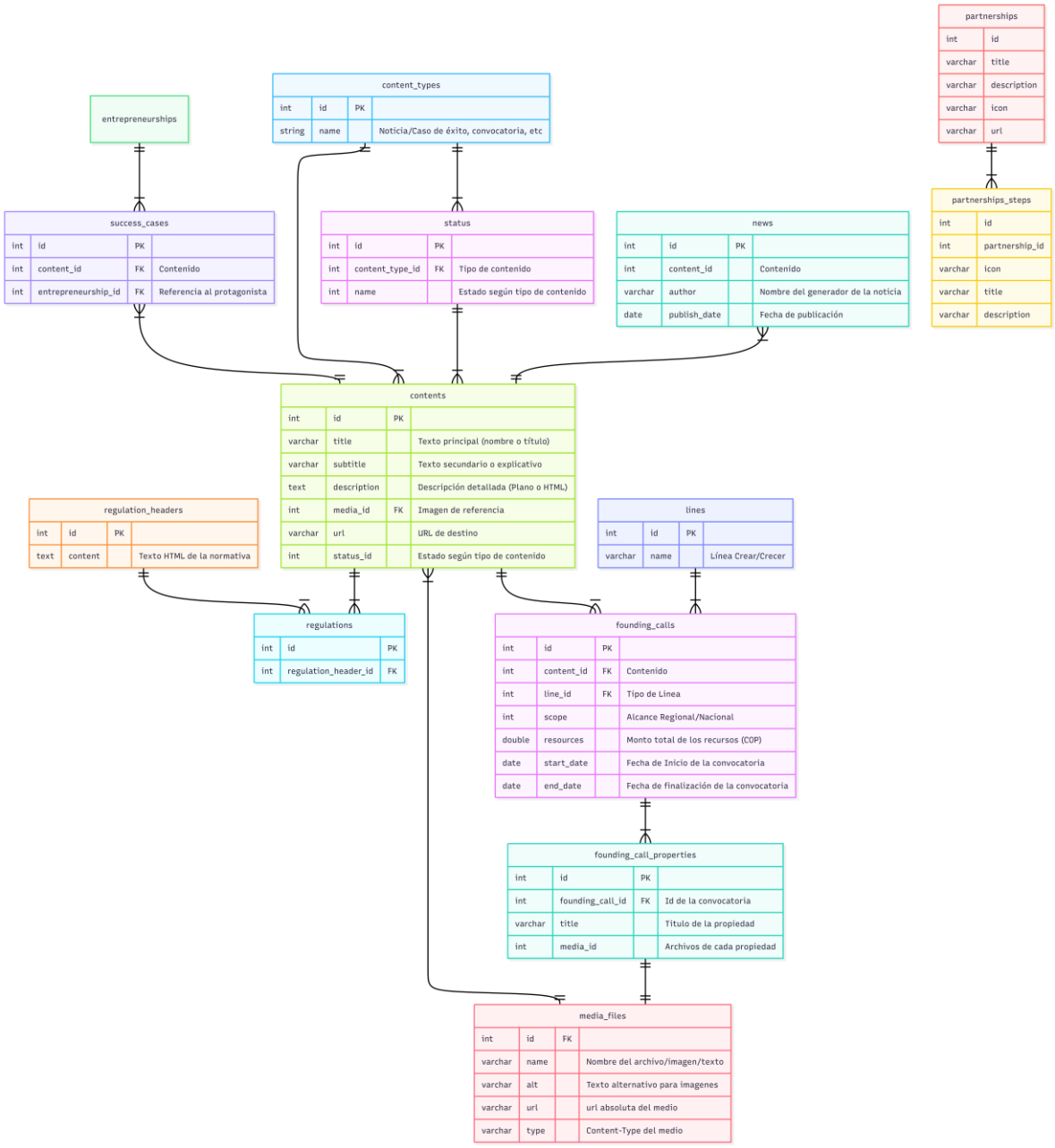
EVIDENCIAS DEL ENTREGABLE

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




#88 Especificar base de datos a utilizar, proceso creación, estructura general de tablas, estrategia de migraciones
CLOSED
1. Descripción de la Base de Datos
2. Proceso de Implementación y Creación


partnerships
int id
varchar title
varchar description
varchar icon
varchar url
content_types
int id PK
string name Noticia/Caso de éxito, convocatoria, etc
entrepreneurships
partnerships_steps
int id
int partnership_id
varchar icon
varchar title
varchar description
success_cases
int id PK
int content_id FK Contenido
int entrepreneurship_id FK Referencia al protagonista
status
int id PK
int content_type_id FK Tipo de contenido
int name Estado según tipo de contenido
news
int id PK
int content_id Contenido
varchar author Nombre del generador de la noticia
date publish_date Fecha de publicación
contents
int id PK
varchar title Texto principal (nombre o título)
varchar subtitle Texto secundario o explicativo
text description Descripción detallada (Plano o HTML)
int media_id FK Imagen de referencia
varchar url URL de destino
int status_id Estado según tipo de contenido
regulation_headers
int id PK
text content Texto HTML de la normativa
lines
int id PK
varchar name Línea Crear/Crecer
regulations
int id PK
int regulation_header_id FK
founding_calls
int id PK
int content_id FK Contenido
int line_id FK Tipo de Línea
int scope Alcance Regional/Nacional
double resources Monto total de los recursos (COP)
date start_date Fecha de Inicio de la convocatoria
date end_date Fecha de finalización de la convocatoria
founding_call_properties
int id PK
int founding_call_id FK Id de la convocatoria
varchar title Título de la propiedad
int media_id Archivos de cada propiedad
media_files
int id FK
varchar name Nombre del archivo/imagen/texto
varchar alt Texto alternativo para imagenes
varchar url url absoluta del medio
varchar type Content-Type del medio