Carlos Daniel Martinez García - 19001064
Douglas Daniel Pérez Hernández - 22003865

Delivery #1: Discovery & Reverse Engineering

TASK #2 AI-Driven Discovery & Context Map:

Vamos a colocar tal cual el LLM nos devolvió la información que le solicitamos del código.

Funcionalidad del Proyecto

Utilizamos Gemini, Copilot y Chatgpt, las 3 nos devolvieron resultados muy parecidos, pero nos quedamos con la respuesta de Chatgpt ya que la consideramos una de las más completas.

¿Qué es este proyecto?

Este proyecto es una implementación backend completa del RealWorld / Conduit App usando Django + Django REST Framework.
Es una API REST realista que simula una plataforma tipo Medium, donde los usuarios pueden publicar artículos e interactuar entre ellos.

No es un proyecto de ejemplo simple, sino una referencia profesional de cómo estructurar un backend real.

¿Para qué sirve?

Sirve para:

- Demostrar buenas prácticas en Django
- Mostrar cómo construir una API REST bien diseñada
- Practicar arquitectura limpia y DDD
- Usarse como base para proyectos reales
- Aprender cómo separar dominios en apps

Carlos Daniel Martinez García - 19001064
Douglas Daniel Pérez Hernández - 22003865

¿Qué funcionalidades implementa?

El proyecto permite:

Usuarios

- Registro de usuarios
- Login y autenticación con tokens (JWT)
- Actualización de datos del usuario

Perfiles

- Ver perfiles públicos
- Seguir y dejar de seguir usuarios

Artículos

- Crear, editar y eliminar artículos
- Listar artículos globales
- Ver feed personalizado (según a quién sigues)
- Marcar artículos como favoritos

Comentarios

- Agregar comentarios a artículos
- Listar comentarios por artículo
- Eliminar comentarios propios

Tags

- Listar tags disponibles
- Asociar tags a artículos

¿Cómo está estructurado?

```
django-realworld-example-app/
│
├── conduit/          ← Proyecto Django principal
│   ├── apps/         ← Apps del dominio (usuarios, artículos, etc.)
│   ├── settings.py   ← Configuración general
│   ├── urls.py       ← Rutas de la API
│   └── wsgi.py
│
├── manage.py         ← Comando principal de Django
├── requirements.txt  ← Dependencias del proyecto
├── db.sqlite3        ← Base de datos de desarrollo
└── README.md         ← Documentación del proyecto
```

Carlos Daniel Martinez García - 19001064
Douglas Daniel Pérez Hernández - 22003865

Qué lo hace interesante

✔ Separación clara de responsabilidades
✔ Uso correcto de Django Rest Framework
✔ Dominio bien definido (no solo CRUD)
✔ Fácil de extender o refactorizar
✔ Ideal para practicar DDD en Django

Principales Bounded Contexts

User Management Context (Usuarios y Autenticación)

App: `users`

Responsabilidad

- Registro de usuarios
- Login / logout
- JWT tokens
- Actualización del perfil propio

Conceptos del dominio

- User
- Email
- Password
- Token

Por qué es un Bounded Context

- Tiene reglas propias (password, autenticación)
- No le importa cómo funcionan artículos o comentarios
- Expone solo lo necesario al resto del sistema

Profile Context (Relaciones sociales)

App: `profiles`

Responsabilidad

- Ver perfiles de otros usuarios
- Seguir / dejar de seguir usuarios

Conceptos

- Profile
- Follow
- Following

Carlos Daniel Martinez García - 19001064
Douglas Daniel Pérez Hernández - 22003865

Importante

- No gestiona login
- No crea usuarios
- Solo maneja **relaciones entre usuarios**

Aquí se ve muy bien la separación:

- `users` = identidad
- `profiles` = relaciones sociales

Article Context (Publicación de contenido)

App: `articles`

Responsabilidad

- Crear, editar, eliminar artículos
- Listar artículos
- Feed personalizado
- Favoritos (likes)

Conceptos

- Article
- Slug
- Body
- Favorites
- Feed

Reglas típicas

- Solo el autor puede editar
- Un usuario puede marcar favorito
- Un artículo pertenece a un autor

Carlos Daniel Martinez García - 19001064
Douglas Daniel Pérez Hernández - 22003865

Comment Context (Interacciones sobre artículos)

App: `comments`

Responsabilidad

- Crear comentarios
- Listar comentarios por artículo
- Eliminar comentarios propios

Conceptos

- Comment
- Author
- Article

Por qué es un contexto separado

- Tiene reglas distintas a artículos
- Depende de Article, pero no lo modifica
- Vida útil diferente

Tag Context (Clasificación)

App: `tags`

Responsabilidad

- Listar tags
- Asociar tags a artículos

Conceptos

- Tag
- Name

Contexto pequeño pero claro

- No tiene reglas complejas
- No depende de usuarios directamente

Carlos Daniel Martinez García - 19001064
Douglas Daniel Pérez Hernández - 22003865

Relación entre Bounded Contexts

[Users] ──┐

├──> [Profiles]

├──> [Articles] ──> [Comments]

│ │

│ └──> [Tags]

Tipo de relaciones

- **Users** → **Articles**: Partnership
- **Articles** → **Comments**: Customer / Supplier
- **Users** → **Profiles**: Shared Kernel (User ID)
- **Articles** → **Tags**: Simple dependency

Context Map

En base a los Bounded Context identificados realizamos el Context Map: