Detalle por entrega

Entrega 1 — REQUERIMIENTOS (nota 1)

Qué entregar

- `REQUERIMIENTOS.md` en la raíz del repositorio.
- Repositorio público (GitHub) con Gitflow inicializado y rama `develop` creada.
- Issues creados que representen los requisitos (al menos 8 issues: usuarios, historias, tareas).

Contenido mínimo de `REQUERIMIENTOS.md`

- Descripción del cliente y problema principal.
- Lista de **usuarios** del sistema.
- **Tipos de usuarios y perfiles** (roles) con permisos por rol (mínimo: Administrador, Usuario/Cliente, Vendedor).
- Funciones indispensables por perfil (lista priorizada).
- Datos básicos a almacenar (entidades y atributos).
- Lista de **Requisitos funcionales** y **No funcionales**.
- Definición del **MVP** (qué incluye / qué queda para futuras versiones).

Checklist mínimo

- Grupo formado (3–4 integrantes) y listado en README del repo.
- `REQUERIMIENTOS.md` completo.
- Issues creados y asignados a integrantes.
- Branching mínimo: `main` y `develop` presentes.
- Al menos 5 commits descriptivos en `develop`.

Criterio de evaluación (20% del total del trabajo) — para esta nota

- Claridad y completitud del levantamiento (50% de la nota de entrega).
- Correcta definición de roles/perfiles (30%).
- Uso inicial de Git/Gitflow + issues + commits (20%).

Entrega 2 — DISEÑO FUNCIONAL & TÉCNICO (nota 2)

Qué entregar

- `DISENO.md` con:
    - Diagrama de casos de uso / mapa de funcionalidades.
    - Descripción de módulos y flujo por rol.
    - Diagrama de arquitectura (API, MongoDB, Redis, Docker).
    - Modelo de datos (colecciones y esquemas Mongoose).

    - Descripción de estrategia de caché en Redis.
    - Lista de endpoints principales (rutas, método, input/output, permisos).
- Mockups o wireframes simples (pueden ser imágenes en `/docs`).
- PR en `develop` agregando `DISENO.md` (con revisión por otro integrante).

Elementos técnicos mínimos a especificar

- Colecciones principales con campos obligatorios y relaciones (ej.: `users`, `products`, `orders`).
- Ejemplo de esquema Mongoose para `users`.
- Endpoints mínimos: `POST /auth/login`, `POST /auth/register`, `GET /users/:id`, CRUD para recurso principal (ej. `products`), `GET /orders`.

Checklist mínimo

- `DISENO.md` completo y versionado en repo.
- Diagrama de arquitectura subido (PNG/SVG).
- Esquemas Mongoose para al menos 3 colecciones.
- Endpoints documentados con métodos y permisos.
- PR con comentarios de al menos otro integrante (revisión).

Criterio de evaluación (20% del total)

- Cohesión entre funcional y técnico (40%).
- Calidad del modelo de datos y endpoints (30%).
- Claridad de la arquitectura y manejo de caché (20%).
- Evidencia de trabajo colaborativo (PRs/reviews) (10%).

Entrega 3 — MVP FUNCIONAL + DOCKER (nota 3)

Qué entregar

- Código fuente completo en rama `develop` / merge a `main` al final.
- `Dockerfile` y `docker-compose.yml` que levanten:
    - Servicio API (Node/Express).
    - MongoDB.
    - Redis.
- `README.md` con instrucciones de ejecución (comando de docker compose).
- Implementación mínima de roles y autenticación (JWT).
- Cacheo con Redis en al menos un endpoint (ej.: listado `GET /products` cacheado).
- Historial de commits y uso de ramas feature por integrante (evidencia en repo).

README — ejemplo (poner exactamente en repo)

```
git clone <repositorio>
cd <repositorio>
docker compose up --build
```

Requisitos funcionales mínimos en el MVP

- Registro e inicio de sesión (JWT).
- Roles aplicados (al menos dos: admin y usuario).
- CRUD del recurso principal.
- Conexión a MongoDB y uso básico de Redis como cache.
- API accesible en `http://localhost:3000`.

Checklist mínimo

- `docker compose up --build` levanta API + Mongo + Redis.
- Autenticación JWT funcionando.
- Roles verificados en al menos 2 endpoints.
- Cache funcionando (ej.: segunda llamada al endpoint devuelve rápido y con header que indique cache).
- Documentación de endpoints en `README.md` o `docs/api.md`.
- Merge final a `main` (o etiqueta de release).

Criterio de evaluación (60% del total)

- Implementación técnica y estabilidad (50% de la nota de entrega).
- Uso funcional de Redis y Mongo (20%).
- Docker Compose y reproducibilidad (20%).
- Calidad del README y documentación (10%).