

# SIMULACRO HU3 — Catálogo de Productos

## Objetivo general

Desarrollar una API REST en .NET con arquitectura en capas (**API, Application, Domain, Infrastructure**) que implemente:

- CRUD de **Usuarios** (con roles básicos).
- CRUD de **Productos**.
- Autenticación **JWT** (registro y login).
- Documentación con **Swagger**.
- **Dockerización** (API + MySQL).

Cada estudiante deberá **subir la solución completa a GitHub** y **desplegar la API en algún servicio** (por ejemplo: Render, Railway o Azure).

---

## Alcance (esta HU)

### Usuarios

- `POST /api/auth/register` → registrar usuario (username, email, password, role)
- `POST /api/auth/login` → login → devuelve JWT
- `GET /api/users` → listar (protegido, solo Admin)
- `GET /api/users/{id}` → obtener por id (protegido)
- `PUT /api/users/{id}` → actualizar (protegido)
- `DELETE /api/users/{id}` → eliminar (protegido)

### Productos

- `POST /api/products` → crear producto (protegido)
- `GET /api/products` → listar productos (protegido)
- `GET /api/products/{id}` → obtener por id (protegido)
- `PUT /api/products/{id}` → actualizar (protegido)
- `DELETE /api/products/{id}` → eliminar (protegido)

### Protección JWT:

Todas las rutas excepto `register` y `login` requieren un token válido.

Algunas rutas (ej. eliminar usuario o producto) deben ser restringidas al rol **Admin**.

---

## Entregables

Cada estudiante debe entregar:

1. **Repositorio GitHub** con la solución en C# (4 proyectos):
  - `webProductos.Api`
  - `webProductos.Application`
  - `webProductos.Domain`
  - `webProductos.Infrastructure`
2. **docker-compose.yml** funcional (API + MySQL + Adminer opcional)
3. **Migraciones EF Core** y `DbContext` configurado
4. **Colección Postman** exportada
5. **2 pruebas unitarias** en la capa `Application` (por ejemplo:
  - Validación de creación de producto
  - Verificación de login de usuario)
6. **Despliegue funcional online**