Laboratorio Guiado

Comprensión de Codebase con GitHub Copilot

Proyecto: `dotnet-architecture/eShopOnWeb`
Stack: ASP.NET Core · Clean Architecture

https://github.com/NimblePros/eShopOnWeb.git

Objetivo del laboratorio

Usar GitHub Copilot Chat para:

- Comprender un sistema basado en **Clean Architecture**
- Identificar responsabilidades reales vs declaradas
- Analizar flujos de negocio y dependencias
- Detectar supuestos implícitos y zonas ambiguas del dominio

Escenario

> Te incorporas como developer/arquitecto a un e-commerce existente.
> Debes entender rápidamente el dominio y la arquitectura antes de extenderla.

Restricción

> No modificar código en este laboratorio.
> Solo análisis asistido.

BLOQUE 1 — Comprensión básica de archivos, clases y funciones

Objetivo

Confirmar que Copilot entiende correctamente **capas, roles y responsabilidades**.

Ejercicio 1.1 — ¿Qué hace este archivo?

Archivo sugerido

- `Program.cs` (Web)

Prompt básico

> “Explícame qué hace este archivo y cuál es su rol dentro de la aplicación”

Resultado esperado

- Descripción del bootstrap
- Registro de dependencias
- Configuración del pipeline HTTP

Ejercicio 1.2 — ¿Qué hace esta clase de dominio?

Clase

- `CatalogItem`

Prompt

> “Describe la responsabilidad de esta clase dentro del dominio”

Refinamiento

> “¿Esta clase contiene lógica de negocio o solo datos?”

Discusión

- Entidades ricas vs anémicas

Ejercicio 1.3 — ¿Qué hace este método?

Método

- `AddItemToBasket(...)`

Prompt

“Describe paso a paso qué hace este método y qué reglas aplica”

Validación

- Revisar invariantes del dominio
- Detectar reglas implícitas

BLOQUE 2 — Responsabilidades y flujos principales

Objetivo

Entender **cómo fluye un caso de uso** a través de las capas.

Ejercicio 2.1 — Identificación de capas

Prompt

“Enumera las capas principales de esta solución y su propósito”

Refinamiento

“¿Qué capa debería contener la lógica de negocio?”

Resultado esperado

- Web
- Application/Core
- Infrastructure

Ejercicio 2.2 — Flujo funcional completo

Caso

Agregar un producto al carrito

Prompt

“Describe el flujo completo desde la petición HTTP hasta la persistencia cuando se agrega un producto al carrito”

Actividad

- Navegar por:
    - Controller / Endpoint

    - Application logic
    - Domain
    - Infrastructure

Ejercicio 2.3 — Flujo alternativo / edge case

Prompt

> "¿Qué ocurre si el producto no existe o el carrito no está creado?"

Aprendizaje

- Manejo de errores
- Supuestos no validados explícitamente

BLOQUE 3 — Relaciones entre módulos y capas

Objetivo

Evaluar si la arquitectura **se respeta en la práctica**.

Ejercicio 3.1 — Dependencias permitidas

Prompt

> "¿Qué dependencias tiene esta capa y cuáles no debería tener según Clean Architecture?"

Discusión

- Dependencias hacia afuera
- Infraestructura aislada

Ejercicio 3.2 — Cruces indebidos

Prompt avanzado

> "¿Detectas posibles violaciones a Clean Architecture en este código?"

Actividad

- Validar si las violaciones son reales o solo aparentes

Ejercicio 3.3 — Mapa arquitectónico

Prompt

> “Describe un diagrama C4 nivel contenedores basado en esta solución”

Actividad

- Dibujar el diagrama
- Ajustar manualmente

BLOQUE 4 — Supuestos implícitos y zonas ambiguas

Objetivo

Entrenar **pensamiento crítico de dominio y arquitectura**.

Ejercicio 4.1 — Supuestos del dominio

Prompt

> “¿Qué supuestos de negocio están implícitos en el dominio de este e-commerce?”

Ejemplos

- Stock infinito
- Precios inmutables
- Usuario autenticado

Ejercicio 4.2 — Ambigüedades del modelo

Prompt

> “¿Qué partes del modelo de dominio podrían generar confusión o mal uso?”

Discusión

- Nombres
- Invariantes no explícitas
- Reglas dispersas

Ejercicio 4.3 — Qué no se puede inferir

Prompt avanzado

> “¿Qué decisiones de negocio no pueden inferirse solo a partir del código?”

Aprendizaje clave

- El código no reemplaza conocimiento de negocio
- Copilot ayuda a identificar vacíos

Cierre del laboratorio

Reflexión guiada

- ¿Dónde Copilot fue más útil?
- ¿Dónde fue ambiguo?
- ¿Qué preguntas quedarían para el PO o arquitecto?