ARQUITECTURA DE SOFTWARE · CLASE 8 · EVALUACIÓN

Cómo se califica este taller

Léanla antes de empezar. Aquí está todo lo que se evalúa y cómo.

Este taller es el parcial del segundo corte. No hay examen escrito. Lo que se califica es su sistema, sus decisiones documentadas, y la prueba de que esas decisiones se hacen cumplir solas.

Los tres componentes

QUÉ	PESO	CÓMO SE MIDE
Los tres commits verde · rojo · verde	**40%**	**Objetivo.** Está en el historial de Actions o no está. No hay interpretación.
Los dos ADR	**30%**	Las cinco secciones completas, y que «qué pagamos» diga algo concreto.
La sustentación próxima clase	**30%**	Individual, 5 minutos, **sin notas**, con el repositorio en pantalla.

El 40% de los commits · así se reparte

LO QUE HAY EN EL HISTORIAL	PUNTOS
Verde, rojo y verde, en ese orden	**40 / 40**
Verde y rojo, pero no volvieron a verde	**30**
Solo verde — nunca probaron la regla	**20**
El pipeline nunca corrió	**0**

Este 40% no requiere talento: requiere hacerlo. Si salen hoy de clase con los tres commits, ya tienen casi la mitad de la nota asegurada.

El 30% de los ADR · así se reparte

QUÉ SE VE EN EL REPOSITORIO	PUNTOS
Dos ADR, cinco secciones, «qué pagamos» concreto en los dos	**30 / 30**
Dos ADR completos pero «qué pagamos» genérico	**22**
Un solo ADR bien hecho	**18**
ADR sin la sección «qué pagamos»	**10**

El criterio duro: un ADR sin factura no es un ADR. Si la sección «qué pagamos» dice «ninguno» o «nada relevante», la decisión era obvia y no había que documentarla.

La sustentación · próxima clase

Cinco minutos por grupo. El profesor elige **en el momento** quién del grupo responde. Solo se permite tener el repositorio abierto en pantalla — **nada de notas**.

Hoy, al final de la clase, cada grupo saca **una de las cinco preguntas de abajo**. La conocen desde hoy. Lo que no saben es a quién le va a tocar hablar, así que **prepárenla todos**.

Las cinco preguntas

1

Muéstreme su **commit rojo** y explíqueme, sin leer, qué se rompía y por qué eso le importaba al negocio.

2

¿Qué pagaron por su ADR-002? Y si alguien dijera que ese costo es muy alto, ¿qué le responde?

3

Si **borro su regla** de `reglas.json`, el pipeline pasa a verde. ¿Por qué eso no resuelve nada?

4

¿Cuál de sus cuatro módulos es el menos confiable, y qué hicieron para que su caída no tumbe al resto?

5

La IA escribió su sistema. ¿Qué parte de este trabajo no habría podido hacer ella, y por qué?

Qué se considera una buena respuesta

SI DICE…	SE LEE COMO
Lee el ADR en voz alta	Memorizó, no entendió
«El import viola la regla»	Respuesta técnica, incompleta
«Si el correo se cae, el préstamo se caía con él»	**Respuesta de arquitecto**
«La IA no sabe qué le importa al negocio»	**Entendió el semestre**

Y una advertencia honesta: si la respuesta suena memorizada, la repregunta va a ser **«¿y qué habría pasado si hubieran decidido lo contrario?»**. Eso no se puede memorizar.

Lista de verificación antes de salir del salón

☐ El sistema corre al abrir `index.html`
☐ Los **dos drivers** están escritos como propiedades, no como funcionalidades
☐ Los **dos ADR** tienen las cinco secciones, incluida «qué pagamos»
☐ `reglas.json` tiene el campo `porque` en lenguaje de negocio
☐ En Actions se ven **tres corridas: verde, roja, verde**
☐ El repositorio es **público**
☐ Mandamos el enlace por el canal
☐ Apuntamos **la pregunta que nos sortearon**