Ejercicio integrador: Hotel Mirador del Lago

Un enunciado largo, de un dominio nuevo, para encontrar vos solo todo lo que vimos de POO: clases, encapsulación, constructores, static, herencia, clases abstractas, polimorfismo, interfaces y relaciones. Se puede leer acá, imprimir o guardar como PDF.

Cómo usar esta guía

EL MÉTODO, EL MISMO DEL TRABAJO FINAL

Es el recorrido de la presentación de repaso del trabajo final (módulo 13), en el mismo orden:

1. **Enunciado.** Leelo entero y escribí el problema con tus palabras.
2. **Sustantivos y verbos.** Sustantivos → clase o atributo; verbos → operación o relación.
3. **Clases.** Cada clase con sus atributos y métodos.
4. **Jerarquía: lo que ES.** Lo repetido sube a un padre, abstracto si no existe solo.
5. **Contratos: lo que PUEDE HACER.** Capacidades que cruzan familias: interfaces.
6. **Relaciones.** Asociación, agregación, composición, dependencia; con multiplicidad.
7. **Diagrama.** El diagrama de clases completo, antes de programar.
8. **Código.** Recién ahora Java, clase por clase.

ANALOGÍA: EL PLANO DEL HOTEL

Nadie reforma un hotel sin el plano: primero se marca qué es cada ambiente, qué paredes son del edificio y qué muebles se pueden sacar. Con el enunciado

pasa lo mismo: cada oración esconde una decisión de diseño. Tu trabajo es encontrarla y marcarla antes de escribir una línea de código.

Trabajá con lápiz y papel. Cada vez que tomes una decisión, anotá **la oración del enunciado que la justifica**: si no encontrás la oración, probablemente la decisión sobra.

El enunciado

ENUNCIADO: HOTEL MIRADOR DEL LAGO

El **Hotel Mirador del Lago** quiere un sistema propio para dejar el papel. Del hotel se guarda el nombre, la dirección y la cantidad de estrellas. Está frente al lago, mira a las montañas y los domingos hay música en vivo en el lobby.

Las habitaciones son parte del edificio: si algún día se demoliera, desaparecerían con él. Todo hotel tiene por lo menos una. Cada habitación tiene un número, un piso, un precio base por noche y una capacidad máxima de personas, y ninguna puede darse de alta sin esos datos. No hay habitaciones «a secas»: cada una es simple, doble o suite. La simple aloja a una persona y cobra el precio base por noche. La doble aloja a dos y, desde la séptima noche, descuenta un 10 % del total. La suite aloja hasta cuatro y, además del precio base, cobra por cada noche un cargo fijo por el jacuzzi. Cada habitación sabe calcular, a su manera, cuánto cuesta quedarse unas noches.

La piscina y el gimnasio también son parte del hotel. De los dos se guarda nombre y superficie en metros cuadrados; de la piscina, además, la profundidad máxima, y del gimnasio, la cantidad de máquinas. No abren igual: la piscina funciona de 9 a 20 y el gimnasio, a toda hora. A los dos se les puede preguntar si están abiertos a una hora dada.

Las camareras de piso limpian habitaciones, pero también la piscina y el gimnasio. Sea una suite o la piscina, a cada lugar se le pregunta si necesita limpieza y se le avisa cuando quedó limpio. Una habitación necesita limpieza cuando el huésped la desocupa; la piscina y el gimnasio, una vez por día.

Del personal se registra el documento, el nombre y un sueldo base, que nunca puede ser negativo. Nadie trabaja como «empleado» en general: hay recepcionistas, camareras de piso y cocineros. Cada recepcionista cubre un turno (mañana, tarde o noche), y el turno noche cobra un 20 % más. Cada camarera tiene pisos asignados y suma un monto fijo por piso. Cada cocinero tiene una especialidad, por la que cobra un plus fijo. A fin de mes se liquidan los sueldos recorriendo la lista del personal, sin preguntar qué hace cada uno. El personal trabaja en el hotel, pero no le pertenece: una recepcionista puede renunciar y emplearse en otro hotel, y un hotel recién inaugurado puede no tener personal todavía.

De cada huésped se guarda el documento, el nombre y un email, que tiene que contener una arroba. Un huésped puede tener ninguna, una o varias reservas a lo largo del tiempo; cada reserva es de un solo huésped y de una sola habitación, y una misma habitación aparece en muchas reservas. Una reserva nace con su huésped, su habitación, la fecha de entrada, la cantidad de noches y la cantidad de personas: es de una noche como mínimo y no puede alojar más personas de las que entran en la habitación. Cada reserva recibe un código correlativo que no se repite en todo el hotel: R-0001, R-0002, R-0003…

Lo que un huésped come o toma en el restaurante se anota como un consumo (descripción, fecha y monto) y se carga a su cuenta.

En el check-in, la recepcionista pasa la tarjeta del huésped por el lector del mostrador, que autoriza el pago; el lector no es de ella ni lo guarda: solo lo usa en ese momento. Al irse, el huésped paga juntas sus reservas y sus consumos. En recepción no les importa de qué es cada cargo: a cada uno le piden su total y los suman.

Consignas

Resolvelas en este orden: cada una usa lo que encontraste en la anterior.

1. Leer el enunciado

1. Leelo dos veces sin anotar nada. Después escribí, en dos o tres renglones, qué problema resuelve el sistema.
2. Subrayá los **sustantivos** con un color y los **verbos** con otro. Sustantivos → clase o atributo; verbos → operación o relación.
3. Tachá lo que ninguna regla usa: hay datos que están solo para ambientar.

2. Clases y atributos

1. Armá la lista de candidatos. Para cada uno decidí: ¿es una clase (tiene datos propios y comportamiento) o un atributo de otra clase?
2. Escribí cada clase con sus atributos y el tipo de cada uno. Vas a encontrar más de quince clases.
3. Nombrá dos objetos concretos de alguna clase, con valores inventados para sus atributos.

3. Encapsulación

1. Encontrá en el enunciado las **cuatro reglas** que un dato nunca puede romper.
2. Para cada regla, indicá qué atributo protege y qué método la hace cumplir (un setter o el constructor). Escribí la firma con su visibilidad.
3. Explicá qué pasaría si ese atributo fuera público.

4. Constructor y static

1. ¿Con qué datos nace cada habitación? ¿Y cada reserva? Escribí las firmas de esos constructores.
2. Hay un dato que no pertenece a ninguna reserva en particular, sino a todas juntas. ¿Cuál es? ¿Cómo lo declarás y cómo genera cada código?

5. Herencia y clases abstractas

1. Encontrá las **tres familias** del enunciado: para cada una, el padre y sus hijas.
2. ¿Qué atributos y métodos suben al padre? ¿Cuál conviene que sea `protected` y por qué?

3. ¿Cuáles padres no pueden existir solos? Buscá la frase que lo dice.
4. ¿Qué método se declara en el padre **sin cuerpo** en cada familia?

6. Polimorfismo

1. Encontrá tres preguntas que todos los de una familia responden, pero cada uno a su manera. Escribí cómo responde cada hija.
2. Encontrá dos lugares donde el sistema recorre una lista **sin preguntar de qué tipo es cada elemento**. ¿De qué tipo es cada lista?

7. Interfaces

1. Hay dos capacidades que comparten clases de familias distintas. ¿Cuáles son y qué métodos tiene cada una?
2. ¿Qué clases firman cada contrato?
3. ¿Por qué no alcanza con una clase padre para cada capacidad?

8. Relaciones con multiplicidad

1. Para cada par de clases que se conocen, decidí si es asociación, agregación, composición o dependencia, y copiá la oración que lo muestra.
2. Escribí la multiplicidad en los dos extremos de cada asociación, agregación y composición.
3. ¿Por qué las habitaciones y el personal, que están los dos «en» el hotel, no se relacionan igual con él?

9. El diagrama UML completo

Dibujá el diagrama de clases con todo lo anterior: atributos y métodos con visibilidad (`+`, `–`, `#`), lo abstracto en cursiva, lo static subrayado, las interfaces con «interface», y cada relación con su flecha y su multiplicidad. Si no entra en una hoja, partilo en dos: las jerarquías por un lado y las relaciones por otro.

10. El esqueleto en Java

1. Escribí las clases abstractas, una subclase de cada una con `@Override`, las dos interfaces y al menos una clase que las implemente.
2. Incluí el contador static y por lo menos un setter que valide.
3. Escribí un `main` que muestre el polimorfismo: un recorrido por una lista con todo lo que se le cobra a un huésped (¿de qué tipo tiene que ser esa lista?) que sume su cuenta y otro por una `List<Empleado>` que liquide los sueldos.

Tabla para completar

Una fila por concepto. En la segunda columna copiá la frase del enunciado; en la tercera, la clase, el método o la relación de tu modelo.

Concepto	**Dónde aparece en el enunciado**	**Clase o elemento**
Clase y objeto		
Atributo		
Encapsulación		
Constructor		
Miembro static		
Herencia		
Clase abstracta		
Método abstracto		
Polimorfismo		
Interfaz (1)		
Interfaz (2)		

Concepto	Dónde aparece en el enunciado	Clase o elemento
Asociación con multiplicidad		
Agregación		
Composición		
Dependencia		
Fuera de alcance		

Checklist: ¿tu modelo tiene todo?

Lo mismo que pide el trabajo final. Marcá cada punto recién cuando lo puedas señalar en tu diagrama.

- □ Al menos 8 clases
- □ 1 clase abstracta con 1 método abstracto
- □ Herencia con 2 subclases o más
- □ 2 interfaces, una implementada en familias distintas
- □ Polimorfismo: una llamada sobre una lista del tipo padre
- □ 1 asociación con multiplicidad
- □ 1 agregación
- □ 1 composición
- □ 1 dependencia
- □ Encapsulación con al menos un setter que valide
- □ 1 miembro static

> **Antes de comparar con otros**

> No hay un único modelo correcto: lo que tiene que estar bien es que cada decisión se apoye en una oración del enunciado. Si dos modelos difieren, comparen las oraciones que usó cada uno.