

INF-253 Lenguajes de Programación

Tarea 3 2025-2: Exploración Sub-acuática

Profesores: José Luis Martí Lara, Wladimir Ormazábal Orellana, Ricardo Salas Letelier
Ayudante coordinador: Bastián Ríos Lastarria
Ayudantes: Diego Duarte, Andrés Jablonca, Cristobal Tirado,
Martín Pardo, Carlos Bravo, Martín Palacios,
Blas Olivares, Sofía Ramirez, Benjamín Echeverría

6 de agosto de 2025

Índice

1. Contexto	2
2. Descripción de la tarea	2
2.1. Objetivo general	2
2.2. Modelo y restricciones	2
2.3. Bucle de juego e idea principal	2
2.4. Vehículos y cosas a considerar	3
2.5. Zonas y recursos	4
2.6. Economía de O ₂ y profundidad	4
2.7. Producción por recolección (loot)	5
2.8. Creación de objetos y costos	5
2.9. Reparación de la Nave Estrellada (victoria)	6
2.10. Condiciones de derrota y persistencia	6
3. Diagrama de clases y atributos mínimos	6
4. Sobre la entrega	7
5. Calificación	8
5.1. Entrega	8
5.1.1. Entrega Mínima (total 30 pts)	8
5.1.2. Asignación de puntajes general (total 70 pts)	9
5.2. Descuentos	10

1. Contexto

Tras varias noches sin dormir para terminar Space Defender, al fin apagas el PC y caes rendido. . . En tu sueño, la nave del juego celebra la victoria con un impulso final, pero el motor se descontrola y te lanza a un planeta cubierto por un vasto océano, utilizas tu cápsula de escape para amortiguar el impacto y al abrir la escotilla solo ves agua, rastros de que en algún momento hubieron estructuras alienígenas y tu nave a la distancia estrellada y echando humo.

Luego de este sueño tan loco, despiertas con una idea: convertir ese sueño en un juego en Java de supervivencia y exploración bajo el agua, con oxígeno limitado, presión y calor como enemigos, y con **tu Nave Estrellada** esperando ser reparada.

2. Descripción de la tarea

2.1. Objetivo general

Construir un prototipo en Java de exploración submarina por turnos donde el jugador gestiona oxígeno (O_2), presión y temperatura para recolectar recursos y planos, y finalmente reparar la Nave Estrellada para escapar del planeta. El mundo se compone de cuatro zonas únicas y persistentes: ZonaArrecife, ZonaProfunda, ZonaVolcanica y NaveEstrellada.

2.2. Modelo y restricciones

El prototipo es un juego de exploración submarina por turnos. El jugador gestiona oxígeno (O_2) y presión para recolectar recursos y planos con el fin de reparar la Nave Estrellada y escapar. Las zonas del mundo son únicas y persistentes (una instancia por zona): **ZonaArrecife**, **ZonaProfunda**, **ZonaVolcánica** y **NaveEstrellada**. Cada zona define un catálogo de *ItemTipo* disponibles y reglas propias (profundidad, calor, riesgos). El acceso por profundidad se valida con una *interfaz* de “acceso a profundidad” implementada por **Jugador** (a nado) y por la **NaveExploradora** (vehículo). El jugador posee banderas de progreso: **tienePlanos**, **mejoraTanque** (mitiga presión) y **trajeTermico** (mitiga calor). El inventario es una lista de *Item* (tipo + cantidad) que incluye recursos y piezas especiales (**PIEZA_TANQUE**, **PLANO_NAVES**, **MODULO_PROFUNDIDAD**).

2.3. Bucle de juego e idea principal

Al inicio del juego, el jugador contará con solo su cápsula de escape a la que puede volver por oxígeno, esta cápsula será la *Nave Exploradora* la cual te permitirá viajar entre zonas, almacenar y crear objetos a partir de los recursos encontrados.

Cada turno representa una decisión del jugador que **consume tiempo/oxígeno** al estar en el agua y puede modificar el estado del mundo o del jugador. Las opciones que tendrá el jugador son:

1. En el agua:
 - **Subir o descender en profundidad.** (debe señalar qué tan profundo y consume una cantidad moderada de oxígeno)
 - **Recolectar recursos.** (Consume oxígeno y la cantidad depende de la profundidad, debe permitir seleccionar qué recurso se obtiene)
 - **Explorar** (consume oxígeno y existe probabilidad de ítems exclusivos para la progresión)

- **Volver a la nave.**

2. Dentro de la nave exploradora:

- **Recargar O₂:** automático al entrar/
- **Subir o descender en profundidad.** (debe señalar qué tan profundo y validar que la nave permita esa profundidad)
- **Gestionar inventario:** mover/guardar ítems, instalar MODULO_PROFUNDIDAD en la nave (si se posee), etc.
- **Crear objetos.**
- **Elegir destino o salir de la nave.**

3. **Ingreso a zona:** Se valida profundidad: se permite entrar si *Jugador* o *NaveExploradora* soportan la **profundidad mínima** de la zona.

4. Mas adelante también se conseguirán algunas mejoras como el *Robot Excavador* con él se podrá:

- Excavar recursos. (recolecciones más grandes)
- Volver a la nave y descargar.
- Reparar. (costará recursos)

Si el O₂ llega a 0 o sucumbe ante las adversidades de la zona, el jugador **pierde su inventario** y reaparece en la nave.

2.4. Vehículos y cosas a considerar

Límites de profundidad. La Nave Exploradora opera con **500 m** de base (permite entrar a *ZonaProfunda* pero se debe negar el descenso a más de 500 m sin módulo) y con el MODULO_PROFUNDIDAD activo permite la exploración completa de todas las zonas con la nave.

Anclaje de profundidad (sin costo de O₂). La nave mantiene una *profundidad de anclaje* p_{nave} por zona, de forma que mientras el jugador esté dentro de la nave, puede **ascender/descender la nave libremente** (no consume O₂) y se fija p_{nave} . Al **salir de la nave**, el jugador aparece exactamente en p_{nave} . Al **volver a entrar**, la nave conserva p_{nave} .

Movimiento del jugador vs. movimiento de la nave. El *movimiento de la nave* no consume O₂ y no cuenta como acción de trabajo.

Validación de acceso. Se permite entrar a una zona si **Jugador** o **NaveExploradora** satisfacen la profundidad mínima: $Jugador.puedeAcceder(Z_{mín}) \vee Nave.puedeAcceder(Z_{mín})$. Además se hacen validaciones cada vez que se aumenta la profundidad, independiente si es la nave o jugador.

2.5. Zonas y recursos

Cada zona cuenta con sus propios recursos y estos harán más facil progresar la aventura, a continuación se detalla qué puede ocurrir en cada zona.

- **ZonaArrecife (0–199 m):** recursos básicos (*cuarzo, silicio, cobre*). Contiene **3** PIEZA_TANQUE en total; cada acción de *explorar* tiene **30 %** de probabilidad de encontrar una, hasta agotar stock.
- **ZonaProfunda (200–999 m):** *plata, oro, acero, diamante, magnetita*. La presión incrementa el consumo de O₂; con *mejoraTanque* este sobrecoste anula.
- **NaveEstrellada (0 m):** zona bimodal. *Sin planos*: se explora el interior (sin gasto de O₂) pero, sin *trajeTermico*, solo se permite **1** acción antes de sucumbir a las altas temperaturas. **25 %** de encontrar MODULO_PROFUNDIDAD (único) durante exploraciones, el cual te permitirá ingresar con tu nave a profundidades mayores a 1000. Si cuenta con *trajeTermico*, exploraciones ilimitadas y además se podrá extraer *cables y piezas de metal*. En caso de contar con la cantidad suficiente de planos, se permitirá apagar el incendio de la nave y repararla para escapar, costará muchos recursos.
- **ZonaVolcánica(1000-1500m):** Recursos: *titanio, sulfuro, uranio*. El jugador debe contar obligatoriamente con la mejora del tanque y el traje térmico para poder entrar a esta zona sin morir apenas salga de la nave, se recomienda traer el RobotExcavador dado que existe un 20 % de probabilidad de que el jugador pierda la consciencia al salir por su cuenta, perdiendo su inventario actual en el proceso. En esta zona, existe solo un 15 % de probabilidad de encontrar los planos para reparar la nave por cada exploración realizada, objeto único.

2.6. Economía de O₂ y profundidad

Capacidad inicial: $O_0 = 60$ unidades (1 unidad ≈ 1 s). Sea una zona con rango $[Z_{\min}, Z_{\max}]$ y profundidad actual del jugador z (acotada al rango). Definimos la profundidad normalizada:

$$d = \frac{z - Z_{\min}}{\max(1, Z_{\max} - Z_{\min})} \in [0, 1]$$

Presión por zona (solo si mejoraTanque=false):

$$\text{pres}(d) = \begin{cases} 0, & \text{Arrecife o NaveEstrellada} \\ 10 + 6d, & \text{ZonaProfunda} \\ \infty, & \text{ZonaVolcánica (No se puede bajar de la nave sin la mejora)} \end{cases}$$

si *mejoraTanque* = *true* \Rightarrow **pres(d)=0**.

Costos de O₂ por acción (sin término de temperatura):

$$C_{\text{explorar}}(d) = \lceil 12 + 10d + \text{pres}(d) \rceil$$

$$C_{\text{recolectar}}(d) = \lceil 10 + 6d + \text{pres}(d) \rceil$$

$$C_{\text{mover}}(d, \Delta z) = \left\lceil (3 + 3d) \cdot \frac{|\Delta z|}{50 \text{ m}} \right\rceil$$

Nota: (1) En **ZonaProfunda** la presión base es de 10 y es el valor que debe ir en su atributo único, es importante que en la fórmula del código utilicen este atributo y no el 10 de forma 'arbitraria' ya que es posible que los correctores modifiquen la presión para testear distintas cosas de su programa.

2.7. Producción por recolección (loot)

La cantidad de unidades obtenidas por **una** acción de *recolectar* aumenta con la profundidad:

$$n(d) = \text{máx}(1, \lfloor n_{\text{mín}} + (n_{\text{máx}} - n_{\text{mín}}) \cdot d \rfloor)$$

Parámetros por zona:

Zona	$n_{\text{mín}}$	$n_{\text{máx}}$
Arrecife (0–199 m)	1	3
Profunda (200–999 m)	2	6
Volcánica (1000–1500 m)	3	8
Nave Estrellada (0 m)	1	4

Ítems de progresión (en *explorar*) y sus probabilidades de obtención: Arrecife:

PIEZA_TANQUE (30 %, stock total 3). NaveEstrellada: MODULO_PROFUNDIDAD (25 %, único).

Volcánica: PLANO_NAVES (p. ej., 15 %, único). Una vez obtenido un ítem único, deja de sortearse.

En caso de no obtener el objeto de progresión en la exploración, el jugador obtiene $\text{máx}(1, \lfloor n_{\text{mín}} \times d \rfloor)$ **de un recurso aleatorio disponible en la zona.**

2.8. Creación de objetos y costos

Formato: cada receta consume ítems del inventario del jugador (o nave) y habilita una mejora o produce un nuevo ítem/estado.

- **Mejora de tanque:** Permite anular los efectos de la presión en zona profunda y volcánica y aumenta en un 100 % la capacidad de oxígeno base del jugador.
Requiere: $3 \times \text{PIEZA_TANQUE} \Rightarrow \text{mejoraTanque} = \text{true}$.
- **Mejora de oxígeno:** Una vez se cuenta con la mejora al tanque de oxígeno, se puede mejorar la capacidad de éste, otorgando 30 s adicionales a la capacidad máxima del tanque por cada mejora.
Requiere: $10 \times \text{plata} + 15 \times \text{cuarzo} \Rightarrow \text{capacidad máxima del tanque de oxígeno} + 30$.
- **Traje térmico:** Requiere: $10 \times \text{silicio} + 3 \times \text{oro} + 5 \times \text{cuarzo} \Rightarrow \text{trajeTermico} = \text{true}$.
- **Instalar módulo de profundidad:** usar MODULO_PROFUNDIDAD $\Rightarrow \text{moduloActivo} = \text{true}$.
- **Robot Excavador:** El Robot Excavador contará con una capacidad de carga base de 1000 unidades en recursos. En caso de superar esta cantidad, el robot necesitará ser reparado, es inmune a los efectos del calor y la presión. A continuación los objetos y mejoras relacionadas al robot:
 - **Crear robot:** Requiere: $15 \times \text{cobre} + 10 \times \text{magnetita} + 5 \times \text{diamante} + 20 \times \text{acero}$
 - **Mejorar capacidad de carga:** Requiere: $10 \times \text{titanio} + 20 \times \text{cuarzo}$
 $\Rightarrow \text{capacidadCarga} + 25\%$
 - **Reparar robot:** Requiere: $4 \times \text{cables} + 3 \times \text{piezas_metal} + 5 \times \text{magnetita}$

2.9. Reparación de la Nave Estrellada (victoria)

Requiere poseer **PLANO_NAVE** (único) y pagar un **costo de materiales** en la zona *NaveEstrellada*.

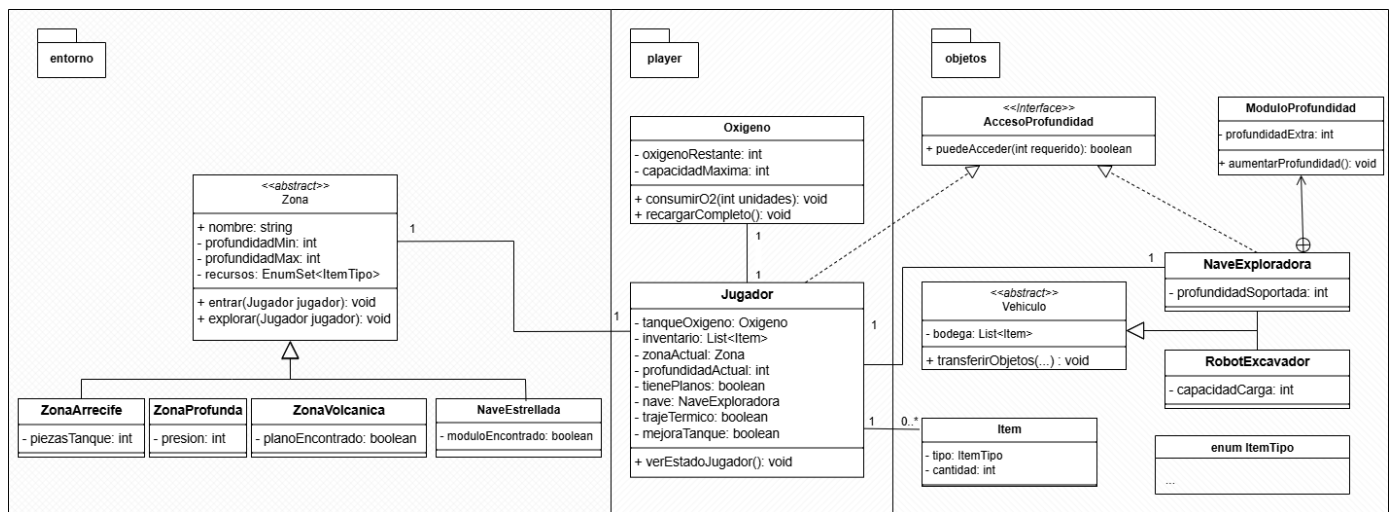
- Costo: $50 \times \textit{titanio} + 30 \times \textit{acero} + 15 \times \textit{uranio} + 20 \times \textit{sulfuro}$

Al confirmar, se dispara la **secuencia de escape** y termina la partida con éxito

2.10. Condiciones de derrota y persistencia

- **Derrota por O₂:** si el contador llega a 0 durante la inmersión, el jugador pierde su inventario y reaparece en la nave.
- **Sofocación:** en *Nave Estrellada* sin `trajeTermico`, al intentar realizar más de una acción, el jugador pierde su inventario y reaparece en la nave.
- **Persistencia de mundo:** cada zona conserva su estado (stock de `PIEZA_TANQUE`, bandera de `MODULO_PROFUNDIDAD` encontrado, etc.). Solo al reiniciar partida se reponen estos estados, no al morir. (se recomienda un archivo extra 'zonas' donde se almacenen estas únicas instancias para evitar crear adicionales sin querer)

3. Diagrama de clases y atributos mínimos



Notas: - Deben incluir en `enum ItemTipo` todos los tipos de objetos mencionados en el enunciado, estos son los recursos de cada zona y sus objetos de progresión tales como `PIEZA_TANQUE`, `PLANO_NAVE` y `MODULO_PROFUNDIDAD`.

- La relación existente entre `NaveExploradora` y `ModuloProfundidad` corresponde a una clase anidada.
- La no realización de la tarea de manera fiel al diagrama presentado acá corresponderá a nota mínima (0), esto incluye la correcta organización de los paquetes *entorno*, *player* y *objetos*, las relaciones presentes, herencias, interfaces, clases anidadas, subclasses y superclasses, atributos y métodos.

- Tienen libertad creativa en cuanto a las cosas que **no se mencionan** en este enunciado **ni** en el diagrama de clases, así como lo que se muestra por pantalla, pero deben ser coherentes con la tarea, representarse de manera clara y estable, y no romper con lo ya establecido, además de que todas las libertades deben quedar documentadas en el README para facilitar la revisión.

4. Sobre la entrega

- El código debe venir indentado y ordenado.
- Se permite que los estudiantes decidan qué argumentos debe contener el método 'void transferirObjetos(...)' con libertad.
- Se debe entregar los siguientes archivos:
 - Main.java
 - README.md
 - Makefile
 - player/
 - Jugador.java
 - Oxigeno.java
 - objetos/
 - AccesoProfundidad.java
 - Vehiculo.java
 - NaveExploradora.java
 - RobotExcavador.java
 - Item.java
 - ItemTipo.java
 - entorno/
 - Zona.java
 - ZonaArrecife.java
 - ZonaProfunda.java
 - ZonaVolcanica.java
 - NaveEstrellada.java
 - Zonas.java (opcional pero recomendado)
- **El Uso de IA está prohibido.** Si se detecta, la tarea recibirá nota 0 sin derecho a apelación. (aplica para la totalidad de la entrega)
- Si no existe orden en el código habrá descuento.
- Todas las funciones deben ir comentadas con el siguiente formato:

```
/*
 * Descripción de lo que realiza
 * @param Parámetro(s): Tipo
 * @return Tipo de retorno
 */
```

Se harán descuentos por funcion no comentada

- La entrega debe realizarse en un archivo comprimido en tar.gz y debe llevar el nombre: Tarea3LP_RolAlumno.tar.gz.
Ej.: Tarea3LP_202473000-k.tar.gz.
- El archivo README.txt debe contener el nombre y rol del alumno e instrucciones detalladas para la correcta utilización de su programa.
- La entrega será vía aula y el plazo máximo de entrega es hasta el día **Miércoles 15 de octubre, 23:59 horas, vía aula.**
- Por cada día de atraso se descontarán 20 puntos (10 puntos dentro de la primera hora)
- Las copias y Uso de código generado por IA serán evaluadas con nota 0 y se informarán a las respectivas autoridades.
- Solo se pueden realizar consultas respecto a la tarea hasta 2 días antes de la entrega.
- Las consultas del enunciado se debe realizarse mediante el foro de la tarea disponible en AULA.

5. Calificación

5.1. Entrega

Para la calificación de su tarea, debe realizar una entrega con requerimientos mínimos que otorgarán 30 pts base, luego se entregará puntaje dependiendo de los otros requerimientos que llegue a cumplir.

5.1.1. Entrega Mínima (total 30 pts)

1. **Creación de objetos + movimiento a nado y viaje entre zonas** [20 pts]
 - Se instancia correctamente **NaveExploradora** y **Jugador** al ejecutar el programa. (5 pts)
 - Todas las zonas son instanciadas como únicas (estado persiste al volver). (5 pts)
para la EM solo se verificará ZonaArrecife y ZonaProfunda de manera práctica, se reserva el derecho de limitar la nota a 30 en caso de no cumplir con el resto de zonas.
 - Se permite el viaje entre las zonas **ZonaArrecife** y **ZonaProfunda** ida y vuelta. (5 pts)
 - Subir/descender respetando costo de O₂ con $C_{\text{mover}}(d)$ en *ZonaArrecife*. (5 pts)
2. **Arrecife: explorar con fórmulas** [10 pts]
 - **Explorar:** aplica $C_{\text{explorar}}(d)$; otorga PIEZA_TANQUE por probabilidad (stock total 3) o *recurso aleatorio* si no cae el mencionado. (10 pts)

De obtener los 30 puntos de la entrega mínima, se procederá a la revisión de las siguientes secciones de la tarea.

5.1.2. Asignación de puntajes general (total 70 pts)

1. Zonas avanzadas [9 pts]
 - **NaveEstrellada:** *explorar* otorga MODULO_PROFUNDIDAD por probabilidad (único) o recurso aleatorio; permite *reparar* con planos+materiales. (3 pts)
 - **ZonaVolcánica:** ingreso validado solo si se cumplen las mejoras requeridas; *explorar* puede otorgar PLANO_NAVES (único). (3 pts)
 - **ZonaProfunda:** Permite el descenso de Nave Exploradora por debajo de los 500 m solo después de conseguir MODULO_PROFUNDIDAD. (3 pts)
2. Fórmulas de O₂ (aplicación global) [12 pts]
 - Implementación de $d(z)$ único y fórmulas correctas y consistentes en $C_{\text{explorar}}, C_{\text{recolectar}}, C_{\text{mover}}$. (8 pts)
 - Presión: $\text{pres}(d) = 10 + 6d$ en **Profunda**; anulada con *mejoraTanque*. (4 pts)
3. Nave Exploradora [12 pts]
 - Profundidad base 500 m; 1500 m con MODULO_PROFUNDIDAD instalado/activo. (2 pts)
 - Anclaje de profundidad por zona; salir/entrar reaparece exactamente en el anclaje; movimiento de la nave no consume O₂. (6 pts)
 - Instalación del módulo desde inventario (consumo del ítem y cambio efectivo del límite). (4 pts)
 - Sistema de almacenamiento de objetos incorporado en la nave, a salvo de pérdidas por derrota del Jugador. (2 pts)
4. Recolección (extraer) y loot [9pts]
 - **Recolectar:** descuenta O₂ con $C_{\text{recolectar}}(d)$; produce cantidades según $n(d)$ por zona. (6 pts)
 - Catálogo correcto de recursos por zona y catálogo global (enum ItemTipo) e inventario se actualiza según corresponda. (3 pts)
5. Inventario, crafteos y mejoras [12 pts]
 - 3 PIEZA_TANQUE \Rightarrow mejoraTanque=true; **Mejora de Oxígeno** +30 s cada una; trajeTermico según receta. (6 pts)
 - Cantidades de las recetas de fabricación/mejora correctas. (4 pts)
 - Consumo de ítems en recetas y validaciones de stock/estado (tienePlanos, etc.). (2 pts)
6. Robot Excavador [8 pts]
 - Opera integrado al bucle (extrae lotes grandes); respeta capacidadCarga; descarga en la nave actualiza inventario. (5 pts)
 - se permite reparación en caso de superar la capacidad de carga y se permite aumentar la capacidad de carga. (3 pts)
7. Victoria/derrota y flujo final [8 pts]
 - Reparación de **NaveEstrellada** con PLANO_NAVES + materiales \Rightarrow victoria. (3 pts)
 - Derrota por O₂=0 (o condiciones definidas) devuelve al jugador a la nave y pierde el botón de la salida. (5 pts)

5.2. Descuentos

- Falta de Makefile o no funcional (-100 pts)
- Falta de clases y/o paquetes del diagrama (-100 pts)
- Código no compila (-100 pts)
- Falta de atributos y/o métodos mínimos del diagrama (-20 pts c/u)
- Firmas y/o argumentos incorrectos en los métodos solicitados (-5 pts c/u)
- Falta de README (-20 pts)
- Falta de orden (Max -20 pts)
- interfaz confusa, poco clara o inestable (máx -30 pts)
- Atributo público que no corresponde (cada uno, máx -30, -5 pts c/u)
- Warnings de compilación (-5 pts c/u)
- Falta de comentarios (cada método, máx -50, -10 pts c/u)
- Falta de información en el README (nombre, rol, instrucciones, etc) (-5 pts c/u)
- Falta de orden o mala indentación en el código (-5 a -20 pts según gravedad)
- Mal nombre en los archivos entregados (-5 pts c/u)
- Uso de código y/o texto generado por IA implicará un 0 y se le informará a las respectivas autoridades.
- Entrega tardía (-10 pts si es dentro de la primera hora; -20 pts por cada día o fracción de retraso)
- En caso de existir nota negativa, esta será remplazada por un 0.