

# INF-253 Lenguajes de Programación

## Tarea 3: Java

Profesor: José Luis Martí Lara, Jorge Ariel Díaz Matte,  
Wladimir Elías Ormazabal Orellana.

Ayudante Cátedra: Norman Cortés Vega, Lucas Morrison Osorio,  
Jhossep Martínez Velásquez.

Ayudante Coordinador: Álvaro Rojas Valenzuela.

Ayudante Tareas: Fernanda Araya Zárate, Bryan González Ramírez,  
Andrés Jablonca Peña, Blas Olivares Gutiérrez, Nicolás Paz Tralma,  
Bastián Salomon Ávalos, Cristian Tapia Llantén, Cristóbal Tirado Morales.

23 de Septiembre de 2024

*So much universe, and so little time...*

## 1. No Java's Sky

Después de la exitosa defensa de Po Inter City, decides explorar otra ciudad de los lenguajes de programación. Al llegar a Class City, una empresa *Game's Java Class* te pide ayuda para desarrollar un juego sobre exploración espacial. Como buen informático/a, tomas esta oportunidad para aprender el lenguaje de las clases, herencias y polimorfismo llamado **Java**.

## 2. El Juego

*No Java's Sky* es un juego de supervivencia y de exploración planetaria. Donde puedes visitar diversos planetas, extraer recursos únicos y sobrevivir a las adversidades climáticas, con el objetivo de llegar al centro de la galaxia. Para ello se te introducen las siguientes componentes:

### 2.1. Mapa Galáctico y los Planetas

Los Planetas que existen en este universo son infinitos y generados al momento de explorar el Mapa Galáctico. Los Planetas tienen dos recursos en común, los Cristales de

Hidrógeno y las Flores de Sodio. Además, cada tipo de planeta enfrenta algún tipo de adversidad climática, que consumirá la energía de protección del exotraje del jugador. Esto depende del tamaño y tipo del planeta. A continuación se presentan los tipos de planetas:

■ **Helados:**

Estos planetas son una tundra congelada, donde uno siempre verá nieve y hielo. Son muy ricos en Cristales de Hidrógeno y puede haber asentamientos de especies alienígenas que están dispuestas a intercambiar recursos por mejoras para tu exotraje o para tu nave.

- Cristales de Hidrógeno:  $0.65 \cdot (4\pi \text{radio}^2)$  Unidades.
- Flores de Sodio:  $0.35 \cdot (4\pi \text{radio}^2)$  Unidades.
- Consumo de Energía:  $0.15 \cdot |\text{temperatura}|$  Unidades de energía.
- Temperatura:  $\text{rand}(-120, -30)$  °C.
- Radio:  $\text{rand}(10^3, 10^6)$  Metros.

■ **Oceánicos:**

Estos planetas son un gran océano, rico en vida acuática. Pero todos los recursos de valor están en las profundidades, por lo que decides aventurarte bajo el agua, consumiendo tu energía para poder respirar. Son ricos en Flores de Sodio y puede haber asentamientos de especies alienígenas que están dispuestos a intercambiar recursos por mejoras a tu exotraje o para tu nave. Estas son las características:

- Cristales de Hidrógeno:  $0.2 \cdot (4\pi \text{radio}^2)$  Unidades.
- Flores de Sodio:  $0.65 \cdot (4\pi \text{radio}^2)$  Unidades.
- Consumo de Energía:  $0.002 \cdot \text{profundidad}^2$  Unidades de energía.
- Profundidad:  $\text{rand}(30, 10^3)$  Metros.
- Radio:  $\text{rand}(10^4, 10^6)$  Metros.

■ **Radiactivos:**

Estos planetas están llenos de vida muy exótica. Pero la atmósfera es muy densa en componentes radiactivos incompatibles con tu sistema, por lo que el traje consume mucha energía para evitar los efectos letales. Esta radiación proviene de los grandes yacimientos ricos en Uranio, un elemento muy cotizado por los asentamientos de otros planetas, se puede intercambiar este elemento por mejoras en los asentamientos. Estas son las características:

- Cristales de Hidrógeno:  $0.2 \cdot (4\pi \text{radio}^2)$  Unidades.
- Flores de Sodio:  $0.2 \cdot (4\pi \text{radio}^2)$  Unidades.
- Uranio:  $0.25 \cdot (4\pi \text{radio}^2) \cdot \text{radiacion}$

- Consumo de Energía:  $0.3 \cdot \text{radiacion}$  Unidades de energía.
- Radiación:  $\text{rand}(10, 50)$  Rad.
- Radio:  $\text{rand}(10^4, 10^5)$  Metros.

■ **Volcánicos:**

Este planeta de lejos puede parecer un sol, si no fuese por la nube de cenizas que cubre toda la superficie. Los grandes volcanes producen un aumento en la temperatura de la superficie que es incapaz de albergar vida, por lo que el exotraje lucha por tu supervivencia. La lava de estos planetas al solidificarse es rica en Platino, un elemento muy cotizado por los asentamientos de otros planetas, se puede intercambiar este elemento por mejoras en los asentamientos. Estas son las características:

- Cristales de Hidrógeno:  $0.3 \cdot (4\pi \text{radio}^2)$  Unidades.
- Flores de Sodio: 0 Unidades.
- Platino:  $0.25 \cdot (4\pi \text{radio}^2) - 20.5 \cdot \text{temperatura}^2$
- Consumo de Energía:  $0.08 \cdot \text{temperatura}$  Unidades de energía.
- Temperatura:  $\text{rand}(120, 256)$  °C.
- Radio:  $\text{rand}(10^3, 10^5)$  Metros.

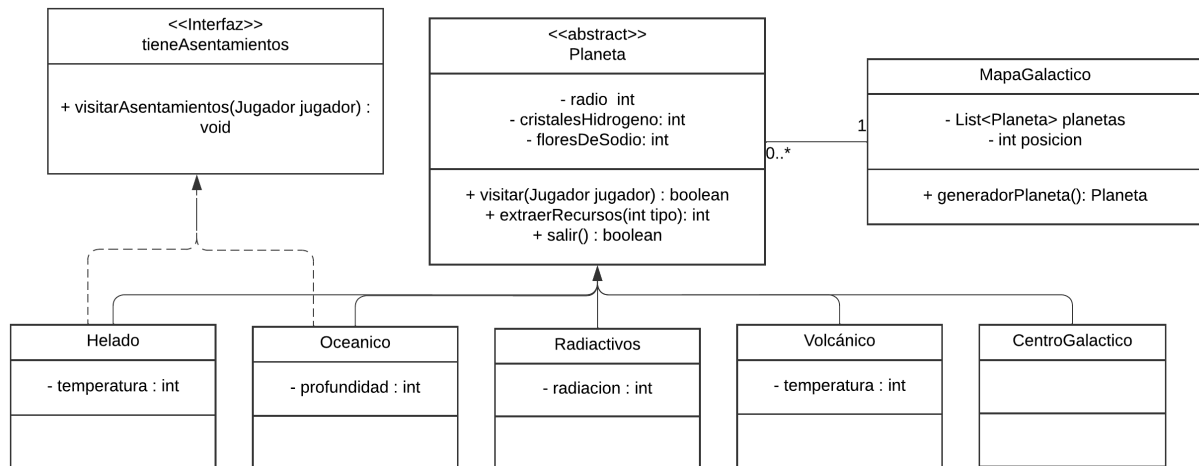
■ **Centro Galáctico:**

Solo existe un centro galáctico. Es un lugar muy hostil, estrellas parecen orbitar un objeto estelar desconocido, y solo las mejores naves espaciales pueden llegar. Para poder visitar este lugar es necesario una nave con una eficiencia de propulsor sobre del 50 %.

*La función  $\text{rand}(a, b)$  retorna un número entero aleatorio del intervalo  $[a, b]$ . Los valores se aproximan al entero más cercano.*

Los planetas son administrados por el Mapa Galáctico. A cada planeta se le asigna un índice único y son guardados en una lista de planetas. El planeta con índice 0 es el primero en la lista y es donde inicia la aventura. Si el jugador ha explorado  $n$  planetas y quiere viajar al planeta  $n + 1$ , el Mapa Galáctico se encarga de generar el nuevo planeta e insertarlo en su lista. En otras palabras, se van creando nuevos planetas a medida que se exploran y son guardados en la lista para visitarlos nuevamente. El Mapa Galáctico tiene registro en qué posición de la lista se encuentra el jugador.

La generación procedural de planetas es aleatorio, varía según su tipo. Las probabilidades son: Helados 30 %, Oceánicos 30 %, Radiactivos 20 %, Volcánicos 19 % y Centro Galáctico 1 %. Este último solamente se puede generar una vez. Para la implementación se tiene el siguiente diagrama de clases UML:



Donde se pueden ver los siguientes métodos:

- **Planeta.visitar(Jugador):** Considerado una acción, el jugador visita el planeta conociendo sus características y permitiendo poder extraer recursos.
- **Planeta.extraerRecursos(int tipo):** Considerado una acción, el jugador extraerá una cantidad de un tipo de recurso disponible en el planeta, los recursos son finitos. Entre más recursos extraigan, más energía de protección se consumirá.
- **Planeta.salir():** Permite abandonar el planeta para poder viajar a otro.
- **visitarAsentamientos(Jugador) (Interfaz):** Permite al jugador visitar a los locatarios del planeta, puede intercambiar recursos de valor por mejoras al exotraje o a la nave.
- **MapaGalactico.generadorPlaneta():** Permite generar un nuevo planeta.

## 2.2. El Jugador y su Nave Espacial

El jugador y la nave son especificados por las siguientes dos clases:

Nave	Jugador
- unidadesCombustible : float - eficienciaPropulsor : float	- unidadesEnergiaProteccion : float - eficienciaEnergiaProteccion : float
+ viajarPlaneta(MapaGalactico MG, int direccion, int tamanoSalto) : boolean + recargarPropulsores(int hidrogeno) : void	+ recargarEnergiaProteccion(int sodio)

#### ■ Jugador:

Es el jugador. Este viene equipado con el exotraje que permite protegerse del ambiente de un planeta. Este usa Sodio extraído de las Flores de Sodio como energía de protección. Tiene una capacidad inicial de 100.0 unidades de energía y una eficiencia inicial de 0.0 %. El consumo y recarga de energía por extraer recursos se calcula como:

$$\text{unidadesConsumidas} = 0.5 \cdot \text{unidadesRecurso} \cdot (\text{consumoEnergia}/100) \cdot (1 - \text{eficiencia})$$

$$\text{unidadesRecargadas} = 0.65 \cdot \text{unidadesSodio} \cdot (1 + \text{eficiencia})$$

Donde *unidadesRecurso* es la cantidad de recursos que se va a extraer en el planeta.

#### ■ Nave:

Es la Nave Espacial, te permite realizar viajes a través del Mapa Galáctico a costa de usar combustible en base hidrógeno extraídos de los Cristales de Hidrógeno, es capaz de curvar el espacio y tiempo generando un “salto” entre planetas. Tiene una capacidad inicial de 100.0 unidades de combustible y una eficiencia inicial de 0.0 %. Para viajar usando el Mapa Galáctico es necesario la dirección (positivo o negativo) y el tamaño del salto. En otras palabras, si estás en Planeta 15 y se quiere viajar al Planeta 10, la dirección es negativa y el tamaño del salto es de 5 Planetas. El consumo y recarga de combustible por viajar se calcula como:

$$\text{unidadesConsumidas} = 0.75 \cdot \text{tamañoSalto}^2 \cdot (1 - \text{eficiencia})$$

$$\text{unidadesRecargadas} = 0.6 \cdot \text{unidadesHidrogeno} \cdot (1 + \text{eficiencia})$$

### 2.3. Ciclo del Juego e Interfaz

Cuando el juego inicia, se debe crear el primer planeta 0 y posicionar al jugador en la órbita del planeta con la energía del exotraje y combustible de la nave llenos. Cada vez que está en la órbita de un planeta se le da la opción de visitar la superficie o de realizar un salto hacia otro planeta.

Una vez en el planeta, el jugador tiene la opción de elegir cuáles y cuantas unidades de recursos extraerá. Como también la opción de interactuar con los locatarios. Estos

intercambian recursos como el Uranio y el Platino por mejoras. El jugador necesita salir del planeta para poder visitar otro. El objetivo final es poder visitar el Centro Galáctico para concluir la exploración.

Es importante considerar que la energía del exotraje del jugador es limitada, y si se le acaba, va a sucumbir por las adversidades del planeta y perderá su inventario. Por suerte, la nave tiene un sistema de emergencias y llevará al jugador nuevamente al planeta 0 con energías y combustible recargadas.

Todas las interacciones se realizan mediante consola. El formato de la interfaz es libre, pero tiene que ser clara y puede esperar inputs según lo que diga. Es decir, si solicita que ingrese un número entre 1 y 5, va a recibir un número entre 1 y 5. Puede crear sus propias historias y narraciones cosméticas para los planetas y sus habitantes. Hay bonificación para las mejores experiencias de exploración y aventura.

### 3. Datos de Relevantes

- Todas las clases deben respetar fielmente el diagrama de clases.
- Es necesario crear un sistema de inventario para que el jugador almacene los recursos extraídos. El inventario queda vacío si el jugador sucumbe por las adversidades del planeta. La implementación queda a decisión del alumno.
- No puedes cambiar los atributos y métodos existentes en el diagrama de clases.
- Puedes añadir nuevos atributos, métodos y clases que estimes conveniente.
- Todos los atributos deben ser **private**, por lo que es necesario implementar **getters** y **setters** para cada atributo.
- Se debe crear una archivo **NoJavaSky.java** que contenga la función *main*.

### 4. Sobre la Entrega

- El código debe venir indentado y ordenado. De no existir orden, se realizará descuentos.
- Se debe entregar los siguientes archivos:
  - NoJavaSky.java, Planeta.java, Helado.java, Oceanico.java, Radiactivo.java, Volcanico.java, CentroGalactico.java, MapaGalactico.java, tieneAsentamientos.java, Nave.java, Jugador.java
  - Makefile
  - README.txt

- Las funciones deben ir comentadas, explicando clara y brevemente lo que realiza, los parámetros que recibe y los que devuelve (en caso de que devuelva algo). Se deja en libertad al formato del comentario.
- Debe estar presente el archivo **MAKEFILE** para que se efectúe la revisión, este debe compilar TODOS los archivos.
- Se usará el compilador javac versión 20 o superior. Se recomienda especificar la versión del compilador en el README.
- El trabajo es individual obligatoriamente.
- **La entrega debe realizarse en un archivo comprimido en tar.gz y debe llevar el nombre: Tarea3LP\_RolAlumno.tar.gz.**  
Ej.: Tarea3LP\_202273000-k.tar.gz.
- El archivo **README.txt** debe contener **nombre y rol del alumno**, junto a instrucciones detalladas para la correcta **compilación y ejecución** del programa.
- La entrega es vía aula y el plazo máximo de entrega es hasta el **16 de octubre a las 23:55 hora aula**.
- Las consultas se deben realizar mediante el foro de la tarea disponible en AULA.
- Por cada día (o fracción) de atraso se descontará 20 puntos. 10 puntos dentro de la primera hora.
- Las copias serán evaluadas con nota 0 y se informará a las respectivas autoridades.

## 5. Clasificación

### 5.1. Entrega

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

#### 5.1.1. Entrega Mínima

Para obtener el puntaje mínimo en la entrega, el programa debe cumplir con los siguientes requerimientos:

- Implementa la lista del Mapa Galáctico pre-generando al menos 10 planetas de al menos un tipo cada uno. (10 pts)

- Usa `MapaGalactico.planetas` para almacenar los planetas generados durante la exploración. (10 pts)
- Implementa movimiento a través del Mapa Galáctico con saltos de tamaño 1. (5 pts)
- Implementa una interfaz por consola que indique la posición en el mapa galáctico y el tipo del planeta en esa posición. (5 pts)

### 5.1.2. Entrega

Luego de cumplir con la Entrega Mínima, puede obtener más puntaje cumpliendo con los siguientes puntos (puede haber puntaje parcial por cada punto):

- **Planetas**
  - Implementa el generador de planetas procedural, permitiendo explorar nuevos planetas (10 pts)
  - Implementa todos los tipos de planetas de acuerdo al diagrama de clases (6 pts)
  - Implementa la generación de recursos y características según el enunciado (6 pts)
  - Implementa la Interfaz Asentamientos, pudiendo intercambiar recursos por mejoras (10 pts)
  - Implementa *getters* y *setters* (3 pts)
- **Jugador y Nave**
  - Implementa el consumo y recarga de energía del exotraje dependiendo de las características del planeta. (10 pts)
  - Implementa un sistema de inventario para el jugador según el enunciado (5 pts)
  - Implementa viajes interplanetarios de diferentes tamaños de saltos, consumiendo combustible correspondiente (4 pts)
  - Implementa recarga del combustible de la nave según el enunciado (3 pts)
  - Implementa *getters* y *setters* (3 pts)
- **Interfaz y Ciclo de Juego**
  - Condición de perdida y victoria (4 pts)
  - Implementa una interfaz con toda la información necesaria para las tomas de decisiones, esta es comprensible y ordenada (6 pts)
- **Bonificación**

Creatividad en la interfaz y experiencia de usuario. Crea una interfaz bonita, historias, arte ASCII y/o nuevas interacciones que generen un sentimiento de exploración. (10 pts extras)



## 5.2. Descuentos

- Falta de orden (Max -20 pts)
- Falta de comentarios (-10 pts c/u, Max -30 pts)
- Código no compila (-100 pts)
- Falta de Makefile (-100 pts)
- Atributo con acceso *public* (-5 pts c/u, Max -30 pts)
- Falta de README (-20 pts)
- Falta de alguna información obligatoria en el README (-5 pts c/u)
- Día de atraso (-20 pts por día (o fracción), -10 pts dentro de la primera hora)
- Mal nombre en algún archivo entregado o información errónea (-5 pts c/u)

En caso de existir nota negativa, esta será remplazada por un 0.