

[75.07 / 95.02]

# Algoritmos y programación III

Trabajo práctico 2 (enunciado)

Primer cuatrimestre del año 2019

# Índice

<b>1. Objetivo</b>	<b>2</b>
<b>2. Consigna general</b>	<b>2</b>
<b>3. Especificación de la aplicación a desarrollar</b>	<b>2</b>
3.1. Jugador	2
3.2. Materiales	3
3.3. Herramientas	
3.4. Mapa	6
<b>4. Interfaz gráfica</b>	<b>7</b>
<b>5. Herramientas</b>	<b>7</b>
<b>6. Entregables</b>	<b>8</b>
<b>7. Formas de entrega</b>	<b>8</b>
<b>8. Evaluación</b>	<b>9</b>
<b>9. Casos de prueba para cada entrega</b>	<b>9</b>
Entrega 0 (8 de noviembre)	9
Entrega 1 (Semana del 12 de noviembre)	9
Entrega 2 (Semana del 19 Noviembre)	10
Entrega 3 (Semana del 26 de Noviembre)	10
Entrega 4 - Final: (6 de Diciembre)	10
<b>10. Informe</b>	<b>10</b>
Supuestos	10
Diagramas de clases	11
Diagramas de secuencia	11
Diagrama de paquetes	11
Diagramas de estado	11
Detalles de implementación	11
Excepciones	11

# 1. Objetivo

Desarrollar una aplicación de manera grupal aplicando todos los conceptos vistos en el curso, utilizando un lenguaje de tipado estático (Java) con un diseño del modelo orientado a objetos y trabajando con las técnicas de TDD e Integración Continua.

## 2. Consigna general

Desarrollar la aplicación completa, incluyendo el modelo de clases e interfaz gráfica. La aplicación deberá ser acompañada por pruebas unitarias e integrales y documentación de diseño.

## 3. Especificación de la aplicación a desarrollar

La aplicación consiste en una versión muy simplificada (y en 2D) del famoso juego Minecraft ([Para mas informacion](#)). El juego es de un solo jugador, y debe permitir al usuario explorar el mapa, conseguir materiales y crear herramientas y a partir de ellos.

### 3.1. Jugador

Es la entidad controlable por el usuario, el cual puede moverse por el mapa, y conseguir materiales que haya en su camino.


Al iniciar la partida, el jugador inicia con un hacha de madera en el inventario.


El jugador debe poder desplazarse en todas las direcciones y puede moverse por todos los casilleros vacíos (es decir, sin ningún material).


### 3.2. Materiales


Los materiales pueden estar dispersas en el mapa, o bien, pueden pertenecer al usuario luego que este las recolecta.

Para recolectar un material el jugador tiene que usar una herramienta para desgastarlo y poder recoger la materia prima. Dependiendo de la durabilidad de cada material, éste va a ser más fácil o difícil de conseguir desgastarlo, y también depende del material con el que esté hecha la herramienta.

<b>Nombre:</b>	Madera
<b>Durabilidad:</b>	10
<b>Cómo conseguirla:</b>	Se puede usar un <b>Hacha</b>
<b>Unidad elemental:</b>	<div>Una unidad de madera</div> 

<b>Nombre:</b>	Piedra
<b>Durabilidad:</b>	30
<b>Cómo conseguirla:</b>	Se puede usar un <b>Pico</b>
<b>Unidad elemental:</b>	<div>  </div> Una unidad de piedra

<b>Nombre:</b>	Metal
<b>Durabilidad:</b>	50
<b>Cómo conseguirla:</b>	Se puede usar un <b>Pico de piedra</b>
<b>Unidad elemental:</b>	<div>  </div> Una unidad de metal

<b>Nombre:</b>	Diamante
<b>Durabilidad:</b>	100
<b>Cómo conseguirla:</b>	Se puede usar un <b>Pico fino de Metal y Piedra</b>
<b>Unidad elemental:</b>	<div>  </div> Una unidad de diamante

### 3.3. Herramientas

Las herramientas son necesarias para poder recolectar los distintos tipos de materiales. El jugador debe tener equipada la herramienta al momento de tratar de desgastar un material que lo requiera. Algunas de estas herramientas pueden ser construidas con distintos materiales, lo que aumenta su durabilidad y fuerza para desgastar más rápido los materiales.

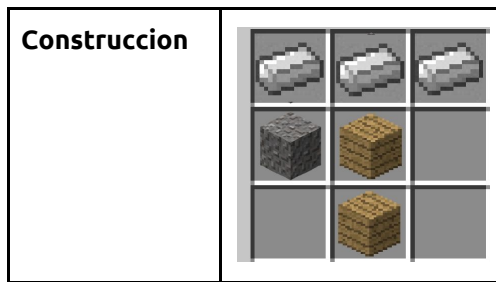
Cada vez que el Jugador usa una herramienta (tenga éxito o no al conseguir el material) la durabilidad de ésta se reduce (la forma de reducción depende de cada herramienta), y la durabilidad del material se reduce dependiendo de la fuerza de la herramienta.

**Construcción de una herramienta:** La construcción consta en poner los materiales necesarios en un orden específico que va a estar detallado en cada herramienta. Los mismos tiene que estar dispuestos en un tablero de 3x3 que el usuario puede acceder y acomodar ahí los materiales

<b>Nombre:</b>	Hacha		
<b>Material</b>	Madera	Piedra	Metal
<b>Durabilidad:</b>	100	200	400
<b>Fuerza</b>	2	5	10
<b>Forma de desgaste</b>	Durabilidad - = Fuerza	Durabilidad - = Fuerza	Durabilidad -= Fuerza / 2
<b>Construccion</b>			

<b>Nombre:</b>	Pico		
<b>Material</b>	Madera	Piedra	Metal
<b>Durabilidad:</b>	100	200	400
<b>Fuerza</b>	2	4	12
<b>Forma de desgaste</b>	Durabilidad - = Fuerza	Durabilidad - = Fuerza / 1.5	Al décimo golpe se rompe
<b>Construccion</b>			

<b>Nombre:</b>	Pico Fino
<b>Material</b>	Metal, Piedra
<b>Durabilidad:</b>	1000
<b>Fuerza</b>	20
<b>Forma de desgaste</b>	Durabilidad-= Durabilidad * 0.1



### 3.4. Mapa

El mapa es del estilo de un tablero, el cual cada casillero puede tener un solo ocupante (sea un jugador, o un material)

El tamaño y disposición del mapa es a elección de los alumnos.

	Met	Met			D	D						
	Met	Met			D	D						
	M	M	M	M								
	M	M	M	M								
	M	M	M	M								
	M	M	M	M			J					

## 4. Interfaz gráfica

La interacción entre el usuarios y la aplicación deberá ser mediante una interfaz gráfica intuitiva. Consistirá en una aplicación de escritorio utilizando **JavaFX** y se pondrá mucho énfasis y se evaluará como parte de la consigna su **usabilidad**.

## 5. Herramientas

1. **JDK (Java Development Kit):** Versión 1.7 o superior que incluya JavaFX.
2. **JUnit:** Framework de pruebas unitarias para Java.
3. **IDE (Entorno de desarrollo integrado):** Su uso es opcional y cada integrante del grupo puede utilizar uno distinto o incluso el editor de texto que más le guste. Lo importante es que el repositorio de las entregas no contenga ningún archivo de ningún IDE y que la construcción y ejecución de la aplicación sea totalmente independiente del entorno de desarrollo. Algunos de los IDEs más populares son:

- a. [Eclipse](#)
  - b. [IntelliJ](#)
  - c. [Netbeans](#)
4. **Apache Ant:** Se deberá incluir el archivo `build.xml` y todos los archivos adicionales necesarios (por ejemplo, `ivy.xml`) para la compilación y construcción automatizada de la aplicación. El informe deberá contener instrucciones acerca de los comandos necesarios (preferentemente también en el archivo `README.md` del repositorio).
5. **Repositorio:** Todas las entregas deberán ser subidas a un repositorio único para todo el grupo en donde quedarán registrados los aportes de cada miembro. Se podrá elegir entre los siguientes:
  - a. GitHub
  - b. GitLab
  - c. BitBucket
6. **Git:** Herramienta de control de versiones
7. **Herramienta de integración continua:** Deberá estar configurada de manera tal que cada *commit* dispare la compilación, construcción y ejecución de las pruebas unitarias automáticamente. Algunas de las más populares son:
  - a. Travis-CI
  - b. Jenkins
  - c. Gitlab CI

## 6. Entregables

Para cada entrega se deberá subir lo siguiente al repositorio:

1. Código fuente de la aplicación completa, incluyendo también: código de las pruebas, archivos de recursos
2. Script para compilación y ejecución (Ant)
3. Informe, acorde a lo especificado en este documento (en las primeras entregas se podrá incluir solamente un enlace a Overleaf o a Google Docs en donde confeccionen el informe e incluir el archivo PDF solamente en la entrega final).

No se deberá incluir ningún archivo compilado (formato `.class`) ni tampoco aquellos propios de algún IDE (por ejemplo `.idea`). Tampoco se deberá incluir archivos de diagramas UML propios de alguna herramienta. Todos los diagramas deben ser exportados como imágenes de manera tal que sea transparente la herramienta que hayan utilizado para crearlos.

## 7. Formas de entrega

Habrán **4 entregas formales** que tendrán una calificación de **APROBADO** o **NO APROBADO** en el momento de la entrega. Además se contará con una entrega 0 preliminar.

Aquel grupo que acumule 2 no aprobados, quedará automáticamente desaprobado con la consiguiente **pérdida de regularidad en la materia de todos los integrantes del grupo**. En cada entrega se deberá incluir el informe actualizado (preferentemente impreso).

## 8. Evaluación

El día del vencimiento de cada entrega, cada ayudante convocará a los integrantes de su grupo, solicitará el informe correspondiente e iniciará la corrección mediante una entrevista grupal.

**Es imprescindible la presencia de todos los integrantes del grupo el día de cada corrección.**

Se evaluará el trabajo grupal y a cada integrante en forma individual. El objetivo de esto es comprender la dinámica de trabajo del equipo y los roles que ha desempeñado cada integrante del grupo. Para que el alumno apruebe el trabajo práctico debe estar aprobado en los dos aspectos: grupal e individual (se revisarán los *commits* de cada integrante en el repositorio).

Dentro de los ítems a chequear el ayudante evaluará aspectos formales (como ser la forma de presentación del informe), aspectos funcionales: que se resuelva el problema planteado y aspectos operativos: que el TP funcione integrado.

## 9. Casos de prueba para cada entrega

*Se sobreentiende que cada entrega consta de las pruebas + el código que hace pasar dichas pruebas).*

*Los alumnos del curso 2 que tengan correctores asignados el martes a la noche tendrán la entrega presencial los días martes.*

### Entrega 1 (jueves 6 de junio)

- Tests de Herramienta:
  - Se crea hacha de madera con con su durabilidad y fuerza correspondiente
  - Se crea pico de madera con con su durabilidad y fuerza correspondiente
  - Se crea hacha de metal con con su durabilidad y fuerza correspondiente
  - Se crea Pico de metal con con su durabilidad y fuerza correspondiente
  - Se crea Pico Fino con con su durabilidad y fuerza correspondiente
  - Hacha se usa contra cada uno de los materiales y se reduce su durabilidad correspondiente al material del Hacha
  - Pico se usa contra cada uno de los materiales y se reduce su durabilidad correspondiente al material del Pico
  - Pico Fino se usa contra Diamante y se reduce su durabilidad correspondiente
  - Pico Fino se usa contra los demás materiales y no se reduce su durabilidad
- Tests de Jugador:
  - Correcta creación e inicialización del jugador
- Test de Material:
  - Madera se golpea con un Hacha de Madera y se verifica que se reduce la durabilidad del material.
  - Piedra se golpea con un Hacha y se verifica que no se reduce la durabilidad del material.
  - Metal se golpea con un Pico de madera y se verifica que no se reduce la durabilidad del material.

### Entrega 2(jueves 13 de junio)

- Construcción de herramienta:



- Se verifica la correcta construcción de cada herramienta con su correspondiente forma de creación y materiales.
- Jugador:
  - Se verifica la correcta inicialización del jugador en el mapa.
  - Se verifica que puede moverse para todas las direcciones vacías.
- Mapa:
  - Se verifica que se puede ocupar un casillero vacío del terreno.
  - Se verifica que no se puede ocupar un casillero ocupado.
  - Se verifica que un casillero puede ser ocupado por un material o personaje.
- Juego:
  - Se verifica que se inicia el juego con el terreno inicializado (materiales distribuidos en el mapa), el jugador inicializado.

### Entrega 3 (jueves 27 de junio)

1. Interfaz gráfica inicial básica: comienzo del juego, disposición de materiales en el tablero y menú inicial.

### Entrega 4 - Final: (jueves 27 de junio)

**Trabajo Práctico completo funcionando, con interfaz gráfica final, sonidos e informe completo.**

<p><b>Tiempo total de desarrollo del trabajo práctico:</b> cinco semanas</p>
--

## 10. Informe

El informe deberá estar subdividido en las siguientes secciones:

### Supuestos

Documentar todos los supuestos hechos sobre el enunciado. Asegurarse de validar con los docentes.

## Diagramas de clases

Varios diagramas de clases, mostrando la relación estática entre las clases. Pueden agregar todo el texto necesario para aclarar y explicar su diseño de manera tal que logre el modelo logre comunicarse de manera efectiva.

## Diagramas de secuencia

Varios diagramas de secuencia, mostrando la relación dinámica entre distintos objetos planteando una gran cantidad de escenarios que contemplen las secuencias más interesantes del modelo.

## Diagrama de paquetes

Incluir un diagrama de paquetes UML para mostrar el acoplamiento de su trabajo.

## Diagramas de estado

Incluir diagramas de estados, mostrando tanto los estados como las distintas transiciones para varias entidades del modelo.

## Detalles de implementación

Deben detallar/explicar qué estrategias utilizaron para resolver todos los puntos más conflictivos del trabajo práctico. Mencionar qué patrones de diseño fueron utilizados y por qué motivos.

## Excepciones

Explicar las excepciones creadas, con qué fin fueron creadas y cómo y dónde se las atrapa explicando qué acciones se toman al respecto una vez capturadas.