

Objetivos

Unidad 2: Diseño y Construcción de Estructuras de Datos

- OE2.1. Proponer y justificar un diseño para implementar una estructura de datos, siguiendo una metodología y considerando la flexibilidad en los tipos de datos y la complejidad temporal de las operaciones.
- OE2.2. Implementar estructuras de datos extensibles y generales utilizando interfaces, herencia y tipos de datos genéricos.
- OE2.3. Escribir el invariante de una clase e implementar los métodos necesarios para su verificación utilizando los elementos apropiados del lenguaje.
- OE2.4. Diseñar, adaptar y utilizar estructuras de datos de acceso directo por llave, las cuales están basadas en la capacidad de las funciones de hashing para localizar una posición física a partir de una llave lógica.
- OE2.5. Utilizar estructuras lineales FIFO, LIFO y diccionarios como parte de la solución de un problema.
- OE2.7. Diseñar y construir las pruebas unitarias de cada una de las estructuras de datos lineales implementadas.

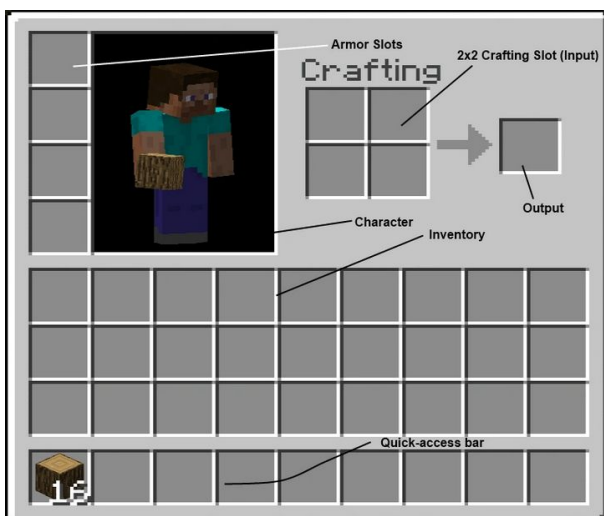
Enunciado

Minecraft es un videojuego de construcción muy famoso, de tipo mundo abierto o sandbox. Fue creado originalmente por Markus Persson (Notch), y posteriormente fue desarrollado por su empresa, Mojang AB. Fue lanzado en el año 2009 y ha superado más de los 90 millones de jugadores alrededor del mundo hasta el día de hoy.

Como se mencionó anteriormente, Minecraft es un juego de mundo abierto, por lo que no tiene un objetivo específico. Esto le permite al jugador tener libertad para escoger su forma de juego.

El juego se centra en el uso de bloques (colocarlos y destruirlos), los cuales representan distintos elementos de la naturaleza, como lo podrían ser piedras, troncos, carbón, entre otros. Minecraft posee un sistema de “crafteo” que le permite a los jugadores crear nuevos bloques a partir de otros correctamente ordenados mediante el uso de mesas de trabajo o desde una sección del inventario que posee la misma función, permitiendo al jugador tener una diversidad de bloques importante.

Esta gran variedad de items le permite a los jugadores hacer volar su creatividad, ya sea desde la construcción de una modesta casa, hasta la construcción de complejos circuitos de redstone para crear pequeñas computadoras funcionales.



A la izquierda, el inventario que posee el jugador, el cual le permite almacenar bloques de distintos tipos, hasta un máximo de 64 bloques del mismo tipo por casilla. Consta de una matriz 4x9, de la cual solo 9 casillas son de acceso rápido.

La empresa desarrolladora de este famoso y extraordinario videojuego ha estado analizando algunos problemas que tienen los jugadores al momento de construir utilizando bloques, al igual que otros inconvenientes de desarrollo en la rama principal de este proyecto.

Una problemática detectada por los desarrolladores más antiguos está relacionada con la referencia que hace cada casilla (que contiene un conjunto de bloques) en el inventario, hacia el contador interno de bloques del sistema, esto se da debido al uso de un algoritmo muy lento. Este consiste en verificar el tipo de bloque que es usado para luego recorrer uno por uno el total de bloques en el sistema, y de esta manera detectar que sí se obtiene un bloque de piedra, este debe agruparse con los bloques de piedra que están presentes en el inventario.

Esto hace que el consumo de memoria RAM al usar el juego sea muy alto. Los desarrolladores están buscando una solución óptima para que el rendimiento sea el mejor.

También, la sección encargada de la experiencia de usuario de Minecraft constantemente ha recibido peticiones para añadir *features* en la próxima versión para tener una modalidad nueva de acceso rápido y de construcción:

La modalidad de acceso rápido, según lo descrito constantemente por los jugadores, sería muy útil si tuviera una variante en la cual solo se pudieran almacenar bloques del mismo tipo en la barra de acceso rápido, siendo estos bloques transportados directamente desde el inventario interno hacia esta barra de manera automática. Esto con el fin de realizar construcciones de una manera más eficiente.



Esta es una de las imágenes que se han encontrado en la dark web sobre dicha función.

Y por último, la modalidad de construcción debería permitir el uso de n diferentes barras de acceso rápido, las cuales deberían variar al clicar una flecha al lado del menú de acceso rápido. De esta manera los jugadores incrementarían su productividad al momento de construir.



Este es el primer menú de acceso rápido.



Al clicar la flecha de la parte derecha despliega el segundo menú de acceso rápido.



Luego de llegar al n -ésimo, el siguiente sería de nuevo el primer menú de acceso rápido.

El equipo detrás de Minecraft se ha enterado de tus grandes habilidades para resolver problemas y tu gran conocimiento de estructuras de datos por lo cual ha decidido contratarle para que presentes una propuesta de cómo implementar estas nuevas características en su producto, éstas son:

1. Proponer una forma eficiente para acceder a los bloques del sistema.
2. Implementar las *features* de modalidad de acceso rápido y construcción, demostrando el uso adecuado de estructuras de datos.
3. Proponer una visualización intuitiva para percibir los nuevos cambios del sistema y de esta forma atraer a nuevos usuarios al mundo de Minecraft.

Para cada una de los anteriores ítems estamos esperando una muy buena descripción del procedimiento, con una breve explicación de lo que está haciendo y el por qué funciona. Usted debe utilizar el método de la ingeniería para resolver este problema y dejar evidencia en su informe de los resultados de cada fase. Recuerde revisar el [Resumen del Método de la Ingeniería](#) y el [ejemplo del Método de la Ingeniería aplicado a un problema](#).

Entregables:

1. Informe PSP0. Cada estudiante debe entregar el informe de su desarrollo.
2. Entrega de informe del método de la ingeniería.
3. Diseño del TAD para cada estructura de datos requerida.
4. Diseño del diagrama de clases desacoplado y utilizando generics.
5. Diseño de los casos de prueba. Adicionalmente debe explicar cómo se resuelven dos casos, paso a paso (con dibujos, si es necesario), por cada estructura de datos diseñada e implementada.
6. Diseño del diagrama de clases de pruebas unitarias automáticas.
7. Implementación de las estructuras de datos
8. Implementación de las *features* de las modalidades de acceso rápido y construcción.
9. Implementación de una interfaz gráfica que permita visualizar las funcionalidades descritas anteriormente.
10. Implementación de las pruebas unitarias automáticas.

El laboratorio debe ser desarrollado en grupos de máximo 3 estudiantes. Recuerde que todos los artefactos generados en la fase de diseño e implementación deben estar en inglés. Recuerde que puede encontrar la rúbrica en el siguiente [enlace](#).

Nota: Usted debe entregar un archivo comprimido en formato zip de un directorio con únicamente 2 archivos: 1 archivo de informe en formato pdf con toda la documentación (de cada una de las fases del método y el análisis) y otro archivo comprimido de un directorio con los archivos de codificación en sus respectivos paquetes.

Todos los archivos deben estar almacenados en GitHub y debe evidenciarse su uso desde el inicio del proyecto. El nombre del archivo comprimido debe tener el formato: PRIMERAPELLIDOEST1_PRIMERAPELLIDOEST2_PRIMERAPELLIDOEST3.zip (tenga en cuenta que el separador entre cada apellido es un guion al piso).