

Proyecto Final

BUSCAMINAS

Generalidades

Fecha de entrega: Sábado 16-08-2025

Modalidad: Individual o Grupal de máximo 2 personas

Objetivo

Aplicar los conocimientos relacionados a la programación básica, interfaz gráfica y de estructura de datos, para programas, en un juego buscaminas de baja complejidad en interfaz gráfica.

Requerimientos

El proyecto consiste en realizar un programa en el cual el usuario pueda jugar Buscaminas deal mejor estilo de Windows, teniendo en cuenta las siguientes consideraciones:

- El usuario debe ingresar el tamaño del lado del tablero (L). Este será un cuadrado perfecto. Y debe ser mayor a 2. Se debe validar que no se ingresen números menores.
- El programa generará aleatoriamente la ubicación (coordenadas) de $[2*L]$ minas. (Tener en cuenta que en cada casilla no podrá quedar más de una mina y que se deben ubicar todas las minas).
- Se debe presentar una interfaz en la cual no se muestre la ubicación de las minas, solo se mostrara lo que el usuario vaya marcando o destapando. La grilla puede ser hecha con una jTable o usar un gráfico de internet o propio.
- El usuario ingresa una coordenada y una instrucción (marcar (X), destapar o desmarcar).
 - Si una mina está marcada no se puede pisar.
 - Si una mina es destapada se pierde el juego.
 - El juego termina cuando se pierda o cuando se ubiquen exactamente todas las minas.
 - El número de marcas es igual al número de minas.
 - Si el usuario destapa una casilla que no tenga minas, debe aparecer en dicha casilla el número de minas que limitan con la coordenada ingresada.
 - Si el usuario pierde, se debe mostrar la ubicación de todas las minas
- Si el usuario destapa una casilla que no tiene minas ni minas que limiten con ella, se deben descubrir todas las casillas que no tengan minas ni vecinas a su alrededor y mostrar las vecinas de estas (en el ejemplo, si se da clic en una casilla 0, se deben destapar hasta los números vecinos a la zona de ceros).
- Para más información pueden ver el juego "WinMine" incluido en los sistemas operativos Windows anteriores a Windows 10.

Visualización Hipotética:

1	1	1	1	*	1
1	*	2	2	3	2
3	3	3	*	2	*
*	*	2	1	2	1
2	2	1	0	0	0
0	0	0	0	0	0

Guía de implementación

- Usted debe elaborar una propuesta de diagrama de clases. Considere que para la propuesta de diagrama de clases debe identificar las relaciones, los atributos y métodos necesarios para cada clase.
- Se debe contar con un Menú en JOptionPane o directamente en el JPanel, que permita el Juego Nuevo y el Salir del Juego. Si se pierde en el juego, debe haber un JOptionPane de tipo Error que indique se perdió, y luego otro, inmediatamente después de cerrado ese, que pregunte si se desea Jugar de Nuevo, o Salir del juego.
- Por último, considere que debe llevar una lista estadística de juegos jugados, perdidos y ganados. Que se mostrará al inicio y final de cada juego, en un JOptionPane o directamente con el JPanel.

Entregables

Aplicación en Java, sub-versionada usando GitHub con mínimo 10 commits.

Reporte en forma digital siguiendo el siguiente formato:

- Portada formal.
- Tabla de contenidos, correctamente vinculada y paginada.
- Resumen Ejecutivo.
-

- Descripción (breve de todo lo que abarca la documentación). El objetivo de este resumen es captar la atención del lector y motivarlo a aprender más sobre el proyecto. Menos de una página.
- Abstracto (Resumen de mínimo 350 palabras) - <http://www.contadordecaracteres.com/>
- Objetivo General y Objetivos Específicos.
- **Introducción:** ¿Por qué se hace el proyecto y qué se incluye?
- **Desarrollo:**
 - Descripción del problema: ¿Qué es lo que hay que resolver? Identificar pequeños problemas que deben resolverse en el proyecto.
 - ¿Cómo se va resolver el problema? La forma en que se planea resolver el problema.
 - Diagrama de clases en UML.
 - Análisis del problema: En dónde, con sus palabras, analice el problema que se desea solucionar desde el punto de vista de las implicaciones y los retos que tiene que realizar este proyecto.
- Análisis de la solución: Luego de la implementación, decir qué se logró implementar, lo que faltó y qué cosas se podrían mejorar de lo que se implementó. No sólo mencionar, si no explicar por qué. También, debe explicar detalles de implementación de la aplicación.
- Discusión de resultados obtenidos: En donde se analiza el grado de completitud y los problemas que se presentaron y si estos fueron corregidos o no y como se corrigieron si es del caso.
- **Conclusiones**
- Recomendaciones: Consejos o advertencias que se derivan de las conclusiones. Lecciones aprendidas durante el desarrollo del proyecto. Recomendaciones para personas que tengan que hacer el mismo trabajo. Así como posibles proyectos que podrían realizarse en el curso.
- Cronograma de Trabajo: distribución de las actividades y el tiempo asignado para su realización.
- Referencias Bibliográficas o Bibliografía: Si es necesario utilizar referencias, estas se realizaran utilizando el formato de APA. Se debe usar al menos un libro o manual directamente vinculado y bien identificado con el lenguaje JAVA y/o la herramienta NetBeans. El mismo puede ser digital.
 - Bitácora.
 - Anexos: (Si hubiesen)

Todo debe empacarse en un archivo compreso en formato ZIP con el nombre:

IIProyectoProgral_ Nombre1Apellidos1-Nombre2Apellidos2.zip

Detalles de la entrega:

- Cualquier proyecto que no compile o esté incompleto, la nota quedará a criterio del profesor.
- Este proyecto programado es únicamente en parejas o individual y debe realizarse en el lenguaje de programación Java, utilizando el IDE NetBeans. Bajo ninguna circunstancia se permitirán copias de trabajos.
- Si se presenta cualquier intento de fraude la calificación será de uno para todos los implicados. Esto incluye el uso de IA.
- El proyecto y su escrito, debe ser sub-versionado en GitHub, y debe constar de al menos 10 commits, que no sean consecutivos. Pierden los puntos de este factor, si se presenta un solo commit, o varios commits cercanos a la fecha de entrega, que sean de forma consecutiva.
- Se debe agregar al profesor al repositorio activo, con el correo AndrewJey@gmail.com.
- Ningún trabajo será recibido después de la fecha y hora establecidas. La entrega debe hacerse a través del Campus Virtual en Moodle, en el que adjuntan un archivo comprimido en ZIP con el proyecto y la documentación, y escriben el link a su repositorio en GitHub. Se recomienda que se empiece a trabajar desde hoy.
- En caso de detectar códigos iguales o muy parecidos entre proyectos las partes involucradas obtendrán la nota de 1 (uno).

Criterios de Evaluación

Aplicación – 60 pts.

Utilización de la Convención de Java – 20 pts.

Documentación – 20 pts.

Aplicación	Puntos
Diagrama y Definición de clases	5
Uso correcto de Interfaz Gráfica	5
Estructuras de datos y Menús	5
Manejo de Casillas	10
Registro de Juegos	10
Configuración de Matrices	10
Jugabilidad (<i>In Game-Play</i>)	10
Sub-versionamiento y Commits	5
Total	60

Documentación	Puntos
Totalidad de Partes	3
Abstracto	2
Tabla de Contenidos Correcta	2
Objetivos e Introducción	2
Desarrollo	3
Resultados, Recomendaciones y Conclusiones	3
Anexos y Datos Finales	2
Formato APA y redacción	3
Total	20

<i>Resumen</i>	<i>Puntos</i>
Aplicación	60
Utilización Convención de Java	20
Documentación	20
Total	100

Convención de Java	Puntos
Comentarios	5
Declaraciones (<i>solamente una declaración por línea, colocación</i>)	3
Nomenclatura de los nombres de: <i>clases, variables, métodos, etc...</i>	4
Estructura de las sentencias	3
Código funcional corto y limpio	5
Total	20