### **Fase 1: Identificación del Problema**

#### **Problema Identificado:**

Se busca diseñar un sistema estratégico que simule un campo de minas representado mediante un grafo. Los jugadores deben navegar este campo enfrentando decisiones bajo incertidumbre, optimizando rutas seguras y minimizando riesgos al tiempo que aplican conceptos avanzados de teoría de grafos.

#### **Enunciado del Problema:**

El problema consiste en desarrollar un sistema que permita a los jugadores atravesar un campo de minas bajo condiciones de riesgo e incertidumbre. Para alcanzar este objetivo, los jugadores deberán tomar decisiones estratégicas basadas en probabilidades de detonación y aplicar algoritmos de teoría de grafos para identificar las rutas más seguras o eficientes. Además, el juego debe incluir elementos dinámicos, como cambios en las posiciones de las minas o caminos bloqueados, para aumentar la complejidad y fomentar la adaptabilidad del jugador.

#### **Criterios Clave para la Definición del Problema:**

1. **Necesidad Real:**
   * Es fundamental diseñar un sistema educativo-lúdico que combine la práctica de conceptos matemáticos y de programación (teoría de grafos) con una experiencia interactiva y desafiante, relevante tanto para el aprendizaje como para el entretenimiento.
2. **Definición Objetiva de la Necesidad:**
   * El sistema debe permitir a los jugadores explorar diferentes rutas en un campo de minas, optimizando su desempeño mediante la aplicación de algoritmos como Dijkstra para rutas más seguras y Kruskal/Prim para optimizar recursos. El enfoque debe estar en la toma de decisiones estratégicas y la gestión de riesgos.
3. **Ausencia de Restricciones Innecesarias:**
   * El problema no limita el diseño del sistema a un tipo específico de mecánicas o visualización. Los únicos requisitos son la representación del campo de minas como un grafo con al menos 50 vértices y 50 aristas, y la incorporación de al menos dos algoritmos clásicos de teoría de grafos.
4. **Diferenciación de Problemas y Soluciones:**
   * El problema está planteado como la necesidad de navegar un campo de minas bajo incertidumbre, sin prejuiciar la solución en términos de interfaz o mecánicas específicas. Esto deja libertad para explorar diversos enfoques.

#### **Objetivo General del Proyecto:**

Diseñar un sistema basado en grafos que permita a los jugadores tomar decisiones estratégicas para navegar un campo de minas, aplicando algoritmos óptimos y enfrentando dinámicas de riesgo e incertidumbre. El sistema debe equilibrar el desafío, la interactividad y la aplicabilidad educativa.

#### **Problema en Términos Objetivos:**

¿Cómo desarrollar un sistema interactivo que combine la simulación de un campo de minas con la aplicación de teoría de grafos, permitiendo a los jugadores optimizar rutas y minimizar riesgos en un entorno dinámico y desafiante?

### **Contexto del Problema:**

1. **Incertidumbre y Riesgo:**
   * Cada nodo del grafo tiene una probabilidad de contener una mina. El jugador debe tomar decisiones sin certeza completa, simulando escenarios reales de optimización bajo incertidumbre.
2. **Elección de Caminos Óptimos:**
   * Existen múltiples rutas entre el punto de inicio y el objetivo, cada una con diferentes riesgos y recompensas. Elegir el camino adecuado requiere análisis estratégico y comprensión de los algoritmos de grafos.
3. **Dinamismo del Escenario:**
   * Las condiciones del campo pueden cambiar, por ejemplo, con minas que se mueven o caminos que se bloquean. Esto obliga al jugador a recalcular y adaptar su estrategia constantemente.
4. **Incorporación de Teoría de Grafos:**
   * El juego debe integrar algoritmos como Dijkstra para encontrar rutas más seguras o Kruskal/Prim para optimizar recursos, vinculando conceptos teóricos con una experiencia práctica.

### **Objetivos por Solucionar:**

1. **Optimizar rutas seguras:**
   * Diseñar herramientas y mecánicas que permitan a los jugadores encontrar caminos con menor probabilidad de detonación.
2. **Fomentar la toma de decisiones informada:**
   * Proporcionar información estratégica de manera accesible, sin sobrecargar al jugador, para que pueda evaluar riesgos y beneficios antes de cada movimiento.
3. **Incrementar el desafío:**
   * Incorporar elementos dinámicos, como minas móviles y caminos bloqueados, que obliguen al jugador a adaptarse y recalcular estrategias durante el juego.
4. **Promover el aprendizaje:**
   * Facilitar la comprensión y aplicación de conceptos avanzados de teoría de grafos mediante una interacción intuitiva y visual.

### **Fase 2: Recopilación de la Información Necesaria**

#### **Fuentes de Información:**

1. **Teoría de Grafos:**
   * **Conceptos Clave:**
     + Representaciones de grafos (matriz de adyacencia y lista de adyacencia).
     + Algoritmos relevantes: Dijkstra (caminos mínimos), BFS/DFS (recorridos), Prim/Kruskal (árboles de recubrimiento mínimo).
2. **Revisión de Proyectos Similares:**
   * Juegos educativos que aplican teoría de grafos o simulaciones con incertidumbre:
     + *Minesweeper*: Aunque no utiliza grafos explícitamente, presenta una base para comprender dinámicas relacionadas con minas.
     + Simuladores de rutas óptimas en tiempo real, como aplicaciones de navegación.
   * **Objetivo:** Identificar mecánicas efectivas y evitar replicar soluciones existentes sin innovación.
3. **Elicitación de Requerimientos:**
   * **Método:** Realizar un análisis estructurado para determinar los requerimientos funcionales y no funcionales del sistema.
     + **Requerimientos funcionales clave:**
       - Generación de grafos con al menos 50 nodos y 50 aristas.
       - Implementación de algoritmos como Dijkstra y Prim/Kruskal.
       - Mecanismos dinámicos como minas móviles o caminos bloqueados.
       - Modos de juego que ajusten la dificultad.
     + **Requerimientos no funcionales clave:**
       - Interfaz gráfica interactiva.
       - Rendimiento óptimo con grafos grandes.
4. **Estudio de Dinámicas de Juegos y Decisiones Estratégicas:**
   * **Dinámicas basadas en incertidumbre:**
     + Modelos probabilísticos para determinar si un nodo contiene una mina.
     + Impacto de la información incompleta en la toma de decisiones.
5. **Revisión de Herramientas de Desarrollo:**
   * **Tecnologías clave:**
     + Lenguaje de programación: Java (por compatibilidad con estructuras de datos y gráficos).
     + Librerías: JavaFX para la interfaz gráfica.
     + Gestión de datos: Estructuras eficientes para representar grafos dinámicos.

#### **Datos Recopilados Relevantes:**

1. **Estructura de los Grafos:**
   * Representaciones posibles (matriz y lista de adyacencia).
   * Comparación de eficiencia entre las dos implementaciones según el tamaño y densidad del grafo.
2. **Algoritmos Seleccionados:**
   * **Dijkstra:** Identificación de rutas seguras con menor probabilidad de explosión.
   * **Prim/Kruskal:** Optimización de recursos al elegir caminos con menor costo global.
3. **Mecánicas del Juego:**
   * Distribución aleatoria de minas en nodos del grafo.
   * Implementación de nodos especiales (escudos, pistas).
   * Incorporación de cambios dinámicos, como minas móviles o bloqueos.
4. **Interacción con el Usuario:**
   * Modalidades de juego (fácil, medio, difícil).
   * Visualización intuitiva del grafo y sus elementos.
5. **Requerimientos Técnicos:**
   * Capacidad para manejar grafos con al menos 50 nodos y 50 aristas.
   * Modularidad para intercambiar implementaciones de grafos (matriz/lista).
   * Pruebas unitarias que garanticen la funcionalidad de los algoritmos y el diseño del sistema.

### **Fase 3: Búsqueda de Soluciones Creativas**

#### **1. Lluvia de Ideas**

Se reunieron propuestas espontáneas para resolver el problema desde diferentes perspectivas:

* **Visualización Intuitiva del Grafo:** Crear un entorno gráfico donde los nodos representen posiciones y las aristas sean caminos posibles. Las minas, escudos y pistas deben estar representados de manera clara y visualmente atractiva.
* **Modos de Juego Variables:** Diseñar varios niveles de dificultad con características específicas:
  + **Fácil:** Menos minas, más pistas visibles.
  + **Medio:** Minas móviles y caminos que se bloquean dinámicamente.
  + **Difícil:** Minas ocultas, reubicación aleatoria constante y restricciones de tiempo.
* **Aplicación de Algoritmos:** Utilizar algoritmos de grafos para enriquecer la jugabilidad:
  + **Dijkstra:** Para sugerir rutas más seguras.
  + **Prim:** Para encontrar caminos estratégicamente óptimos al comienzo de cada nivel.
* **Incorporación de Decisiones Estratégicas:** Permitir que los jugadores inviertan recursos (como escudos) para reducir el riesgo o explorar nodos adicionales.

#### **2. Listas de Revisión**

Examinamos las áreas clave del diseño del sistema y su relación con el problema:

* **Generación del Grafo:**
  + ¿El sistema garantiza al menos un camino seguro entre el nodo inicial y el objetivo?
  + ¿Las aristas se generan con pesos adecuados para los algoritmos?
* **Dinamismo:**
  + ¿El juego incluye cambios en las minas o los caminos para mantener el desafío?
  + ¿Los nodos especiales agregan valor a la experiencia del jugador?
* **Interfaz:**
  + ¿Es fácil para el jugador interpretar el grafo y sus elementos?
  + ¿El diseño se adapta a los diferentes niveles de dificultad?
* **Implementación de Algoritmos:**
  + ¿Los algoritmos seleccionados están correctamente integrados en las mecánicas del juego?
  + ¿Se cumplen los objetivos educativos y de aprendizaje en el proceso?

#### **3. Lista de Atributos**

Se listaron las características principales del sistema, separándolas por funcionalidad:

* **Atributos del Grafo:**
  + Al menos 50 nodos y 50 aristas.
  + Representación mediante matriz o lista de adyacencia.
  + Etiquetas para nodos (mina, escudo, pista, normal).
* **Atributos del Juego:**
  + Modos de dificultad (fácil, medio, difícil).
  + Cambios dinámicos (minas móviles, caminos bloqueados).
  + Implementación de rutas seguras basadas en algoritmos.
* **Atributos Visuales:**
  + Representación gráfica del campo de minas.
  + Indicadores visuales para nodos especiales.
* **Atributos Educativos:**
  + Aplicación de algoritmos como Dijkstra, BFS y Prim.
  + Promoción del análisis estratégico y la toma de decisiones bajo incertidumbre.

#### **4. Relación Forzada**

Se exploraron combinaciones entre ideas aparentemente no relacionadas para generar propuestas innovadoras:

* **Algoritmos + Dinámica del Juego:**
  + Las minas móviles pueden moverse siguiendo reglas específicas basadas en algoritmos (e.g., BFS para simular movimientos "inteligentes").
* **Interacción Jugador + Escudos:**
  + Los jugadores podrían "desactivar" minas en un nodo si recolectan suficientes escudos en nodos anteriores, añadiendo un componente estratégico.
* **Modos Competitivos + Algoritmos:**
  + En un modo multijugador, los jugadores compiten para encontrar la ruta más segura usando Dijkstra, enfrentándose a minas dinámicas que afectan a ambos jugadores.

**Ideas Generadas:**

1. **Campo de Minas Visual Dinámico:**
   * Un grafo interactivo donde cada nodo y arista sea visible, pero las minas y sus probabilidades estén ocultas hasta que el jugador se acerque.
2. **Sugerencias Estratégicas Basadas en Algoritmos:**
   * Opcionalmente, los jugadores pueden solicitar rutas sugeridas usando Dijkstra, pero con un costo en recursos o tiempo.
3. **Modos de Juego Innovadores:**
   * **Modo Explorador:** Ver solo nodos cercanos y sus probabilidades.
   * **Modo Competitivo:** Dos jugadores en el mismo campo de minas, compitiendo para alcanzar el objetivo.
   * **Modo Supervivencia:** Minas que cambian dinámicamente de lugar tras un número de turnos.
4. **Escenarios Educativos:**
   * Cada nivel introduce un nuevo concepto de teoría de grafos, como caminos mínimos o árboles de recubrimiento, reforzando el aprendizaje mediante el juego.
5. **Personalización del Grafo:**
   * Permitir al jugador seleccionar parámetros, como el número de nodos, densidad de aristas y proporción de minas.

### **Fase 4: Transición de la Formulación de Ideas a los Diseños Preliminares**

### **Análisis Crítico de las Soluciones Propuestas**

Se evaluaron las ideas generadas en la Fase 3 en términos de viabilidad técnica, impacto en la jugabilidad y alineación con los objetivos educativos y lúdicos del sistema:

1. **Campo de Minas Visual Dinámico:**
   * **Viabilidad:** Factible con herramientas como JavaFX para la representación gráfica.
   * **Impacto:** Mejora la experiencia visual del jugador, haciéndola más interactiva.

**Resultado:** **Aceptada.**

1. **Sugerencias Estratégicas Basadas en Algoritmos:**
   * **Viabilidad:** Implementar Dijkstra para rutas óptimas es factible; sin embargo, podría simplificar demasiado el desafío del juego si se abusa.
   * **Impacto:** Ayuda a los jugadores en niveles iniciales, pero puede restar dificultad en niveles avanzados.

**Resultado:** **Aceptada con modificaciones.** Solo estará disponible en niveles fáciles o como opción limitada.

1. **Modos de Juego Innovadores:**
   * **Modo Explorador:** **Viabilidad:** Factible; requiere ocultar información de nodos y vecinos.

**Resultado:** **Aceptada.**

* + **Modo Competitivo:** **Viabilidad:** Complejo de implementar para el alcance del proyecto actual.

**Resultado:** **Descartada.**

* + **Modo Supervivencia:** **Viabilidad:** Factible; requiere lógica para mover minas dinámicamente.

**Resultado:** **Aceptada.**

1. **Escenarios Educativos:**
   * **Viabilidad:** Factible; puede integrarse como tutoriales iniciales o retos progresivos.

**Resultado:** **Aceptada.**

1. **Personalización del Grafo:**
   * **Viabilidad:** Complejo de implementar de manera flexible dentro de los límites actuales.

**Resultado:** **Descartada.**

### **Modelos Utilizados en el Proceso de Diseño**

Para avanzar hacia diseños preliminares, se emplearon tres tipos de modelos:

#### **1. Modelo de Simulación:**

* **Objetivo:** Probar dinámicas del juego sin necesidad de implementación completa.
* **Aplicación:**
  + Simular nodos y minas en un grafo con nodos etiquetados como normales, minas y escudos.
  + Probar la lógica de rutas óptimas y cambios dinámicos (e.g., minas móviles).
  + Herramientas: Scripts en Java para verificar comportamientos esperados antes de integrarlos en la interfaz gráfica.

#### **2. Modelo de Diseño Visual:**

* **Objetivo:** Planificar la interfaz y visualización del campo de minas.
* **Aplicación:** Boceto preliminar:
  + Nodos como círculos, con colores que indiquen estado (normal, mina, escudo).
  + Aristas como líneas entre nodos.
  + Indicadores visuales para probabilidades y rutas seguras.

### **Diseños Preliminares Resultantes**

1. **Estructura del Grafo:**
   * **Representación:**
     + Lista de adyacencia para manejo eficiente de nodos y aristas.
   * **Nodos Especiales:**
     + Minas: Nodos con una probabilidad pp definida.
     + Escudos: Nodos que reducen el impacto de detonaciones.
     + Normales: Nodos sin propiedades especiales.
2. **Algoritmos Seleccionados:**
   * **Dijkstra:** Ruta óptima para minimizar riesgo.
   * **BFS:** Explorar nodos vecinos y calcular probabilidades acumuladas.
   * **Lógica Dinámica:** Movimiento de minas tras un número fijo de turnos.
3. **Modos de Juego:**
   * **Explorador:** Nodo inicial visible, vecinos descubiertos al avanzar.
   * **Supervivencia:** Minas cambian de lugar periódicamente.
   * **Educativo:** Escenarios progresivos con introducción a algoritmos.
4. **Interfaz Gráfica:**
   * **Elementos:**
     + Nodo: Representado como círculo.
     + Arista: Línea con peso visible en modos educativos.
     + Indicadores: Colores o iconos para minas, escudos y pistas.
   * **Interacción:**
     + Selección de rutas con clics.
     + Información adicional desplegable sobre nodos y rutas.

### **Fase 5: Evaluación y Selección de la Mejor Solución**

#### **1. Factores de Evaluación**

Se utilizaron los siguientes criterios para analizar las posibles soluciones:

* **Viabilidad Técnica:**
  + ¿Es implementable dentro del alcance del proyecto?
  + ¿Se dispone de herramientas y recursos adecuados para su desarrollo?
* **Impacto Educativo:**
  + ¿Facilita la comprensión de la teoría de grafos y la aplicación de algoritmos?
  + ¿Promueve habilidades analíticas y estratégicas en los jugadores?
* **Experiencia del Usuario:**
  + ¿Es interactivo y atractivo para el jugador?
  + ¿Ofrece un nivel adecuado de desafío y adaptabilidad?
* **Complejidad y Costos:**
  + ¿El diseño es manejable dentro del tiempo y esfuerzo disponible?
  + ¿Evita complicaciones técnicas innecesarias?

#### **2. Alternativas Analizadas**

**A. Campo de Minas Visual Dinámico**

* **Ventajas:**
  + Mejora la visualización de nodos, rutas y riesgos.
  + Facilita la interacción y la comprensión de la estructura del grafo.
* **Desventajas:**
  + Requiere una implementación gráfica robusta (JavaFX o similar).

**Resultado:** **Aceptada.** Es fundamental para la experiencia del usuario.

**B. Sugerencias Estratégicas Basadas en Algoritmos**

* **Ventajas:**
  + Introduce un componente educativo directo (Dijkstra para rutas óptimas).
  + Útil en niveles iniciales para familiarizar al jugador con las mecánicas.
* **Desventajas:**
  + Puede simplificar demasiado el juego en niveles avanzados.
* **Resultado:** **Aceptada con restricciones.** Disponible solo en modos fáciles.

**C. Modos de Juego (Explorador y Supervivencia)**

* **Ventajas:**
  + Aumentan la variedad y el desafío.
  + El modo Supervivencia introduce dinamismo con minas móviles.
* **Desventajas:**
  + El modo Explorador podría requerir lógica adicional para ocultar nodos.
* **Resultado:** **Aceptados.** Ambos modos son factibles y enriquecen la jugabilidad.

**D. Escenarios Educativos Progresivos**

* **Ventajas:**
  + Refuerzan la enseñanza de conceptos como rutas óptimas y árboles mínimos.
  + Se integran fácilmente como tutoriales o retos.
* **Desventajas:**
  + Requieren diseño adicional de niveles y narrativas.
* **Resultado:** **Aceptados.** Se incluirán como parte del diseño.

**E. Personalización del Grafo**

* **Ventajas:**
  + Permitiría al jugador ajustar la dificultad de forma granular.
* **Desventajas:**
  + Alta complejidad técnica para el alcance actual del proyecto.
* **Resultado:** **Descartada.** No es prioritaria en esta versión.

#### **3. Retroalimentación y Refinamiento**

Con base en el análisis de las alternativas, se realizaron los siguientes ajustes:

* Las **sugerencias estratégicas (Dijkstra)** serán limitadas a niveles fáciles o como herramienta opcional con un costo (e.g., consumo de recursos del jugador).
* El **modo Supervivencia** incluirá un contador de turnos para la reubicación de minas, balanceando el dinamismo con la dificultad.
* La personalización del grafo será excluida en esta versión para priorizar otros elementos esenciales.

### **Solución Seleccionada**

**Diseño Final Propuesto:** Un sistema interactivo y estratégico que incluye:

1. **Grafo Dinámico:**
   * Representado visualmente con JavaFX.
   * Nodos etiquetados como normales, minas o especiales (escudos, pistas).
2. **Modos de Juego:**
   * **Modo Explorador:** Información limitada al nodo actual y vecinos.
   * **Modo Supervivencia:** Minas móviles y caminos bloqueados dinámicamente.
3. **Aplicación de Algoritmos:**
   * **Dijkstra:** Sugerencias estratégicas en niveles fáciles.
   * **Prim/Kruskal:** Retos específicos basados en optimización de caminos.
4. **Interfaz Intuitiva:**
   * Visualización clara del grafo, con colores e iconos representativos.
   * Interacción mediante clics para seleccionar rutas.
5. **Escenarios Educativos:**
   * Tutoriales y retos progresivos que introducen conceptos de grafos y algoritmos.

### **Evaluación Final**

La solución seleccionada cumple con los objetivos del proyecto, equilibrando jugabilidad, impacto educativo y viabilidad técnica. Además, permite futuras expansiones, como la personalización del grafo o modos competitivos, en iteraciones posteriores.

**FASE 6: PREPARACIÓN DE INFORMES Y ESPECIFICACIONES**

Una vez seleccionado el diseño final del juego de minas, se debe preparar la documentación necesaria para su aprobación, soporte y ejecución. En el caso de este proyecto, el informe incluye:

1. **Resumen del Proyecto**: El desarrollo de un juego interactivo estilo buscaminas que emplea estructuras de datos como grafos y una interfaz gráfica desarrollada en JavaFX.
2. **Especificaciones Técnicas**:
   * **Lenguaje de Programación**: Java (versión usada JDK 23).
   * **Interfaz Gráfica**: JavaFX.
   * **Estructuras de Datos**: Grafo basado en listas de adyacencia.
   * **Características Principales**:
     + Generación dinámica de minas en un tablero configurable.
     + Representación visual del jugador, minas y meta utilizando imágenes.
     + Mecanismo de detección de colisiones (mina o meta).
     + Selección de dificultad inicial (cantidad de minas).
3. **Resultados Esperados**:
   * Un juego funcional con una cuadrícula dinámica que simule un tablero de minas.
   * Experiencia interactiva que permita al usuario navegar y visualizar el progreso.
4. **Componentes Entregables**:
   * Código fuente organizado en paquetes (model, ui).
   * Recursos gráficos (images/bomba.png, images/diamante.png).
   * Documentación de uso e instalación.
5. **Consideraciones de Diseño**:
   * El tablero debe ser intuitivo para el jugador.
   * El uso de grafos asegura flexibilidad para futuros cambios en la lógica de conexiones.
   * Las imágenes deben escalarse para mantener la calidad visual en diferentes dispositivos.

**FASE 7: IMPLEMENTACIÓN DEL DISEÑO**

La fase final del proyecto consiste en transformar el diseño conceptual en un producto funcional. Para este juego, las etapas de implementación son las siguientes:

1. **Preparación del Entorno**:
   * Configurar un entorno de desarrollo integrado (IDE) como IntelliJ IDEA con soporte para JavaFX.
   * Asegurar la inclusión del archivo module-info.java para gestionar dependencias.
2. **Producción del Sistema**:
   * **Estructuración del Proyecto**: Seguir la estructura propuesta en los paquetes (model, ui, resources) para una mayor claridad y mantenibilidad.
   * **Implementación de la Lógica del Juego**: Asegurarse de que las conexiones entre nodos (grafos) y las reglas del juego (minas, meta) se reflejen correctamente.
   * **Desarrollo de la Interfaz**: Diseñar la interfaz gráfica que permita al usuario interactuar con el tablero mediante JavaFX, ajustando las escenas según las acciones del jugador.
3. **Pruebas**:
   * **Pruebas Funcionales**: Verificar que el jugador pueda navegar sin errores y que el juego termine correctamente al alcanzar la meta o pisar una mina.
   * **Pruebas de Interfaz**: Garantizar que las imágenes y los elementos gráficos sean visibles y estén correctamente alineados.
   * **Pruebas de Estrés**: Evaluar el rendimiento con diferentes configuraciones del tablero y cantidades de minas.
4. **Entrega del Producto Final**:
   * Entrega del proyecto en un repositorio de github**.**