Proyecto: Universo de Algoritmos

Aplicación de Autómatas y Técnicas de Programación

- **Modalidad de trabajo:** equipos de 2 o 3 estudiantes.
- **Entrega final:** lunes 9 de junio de 2025 (hora pactada con el profesor).
- **Lenguaje:** elegido por los estudiantes.
- **Entregables:**
  - Repositorio en GitHub con código versionado y evidencias de participación de cada integrante.
  - Manual de usuario y manual técnico del sistema.
  - Video explicativo en inglés (mínimo 5 minutos), donde todos los integrantes participen en cámara, con tiempos equitativos.

Módulo 1: Misión Interestelar – Resolución por Backtracking Recursivo

Desarrollar un sistema con interfaz gráfica que permita resolver un reto de exploración galáctica basado en una matriz MxN (mínimo 30x30), donde el objetivo es que una nave llegue desde una celda origen hasta una celda destino siguiendo ciertas reglas y restricciones complejas:

1. La matriz representará un mapa del universo cargado desde un archivo JSON, incluyendo:
   - Tamaño de la matriz (M y N).
   - Coordenadas de la **celda origen** y **celda destino**.
   - Posiciones de **agujeros negros** (celdas completamente bloqueadas).
   - Posiciones de **estrellas gigantes** (mínimo 5), que permiten destruir 1 agujero negro adyacente (si hay más, solo destruye uno).
   - Posiciones de **agujeros de gusano**, son túneles de una sola dirección que conectan dos celdas distantes. Se consumen al usarse.
   - Celdas que requieren un nivel de **carga mínima** para acceder (la nave empieza con cierta carga inicial, que puede cambiar durante el trayecto).
   - Celdas de **recarga de energía**, que permiten recargar X veces el valor actual de energía de la nave.
   - Cada celda tiene un valor de gasto de energía para la máquina. Cuando la lave agota su energía no puede continuar por ese camino y deberá retornar para probar un nuevo camino.
   - MatrizInicial: contiene las celdas ya definidas con la carga de energía que se gasta la nave al pasar por ellas.
2. El algoritmo debe:
   - Usar Backtracking recursivo para encontrar al menos una solución válida o informar que no existe una solución.
   - Registrar y mostrar la **cadena de pasos** seguida para llegar a la meta.
   - Validar y aplicar las restricciones de movimiento.
   - Mostrar gráficamente el universo, con íconos representando los tipos de celda (ver imagen 1).
   - Permitir ver la **animación paso a paso** de la ruta encontrada desde la interfaz.

ucaldas@ucaldas.edu.co www.ucaldas.edu.co PBX (57)(6) 878 15 00 Calle 65 # 26 - 10 | Manizales - Colombia

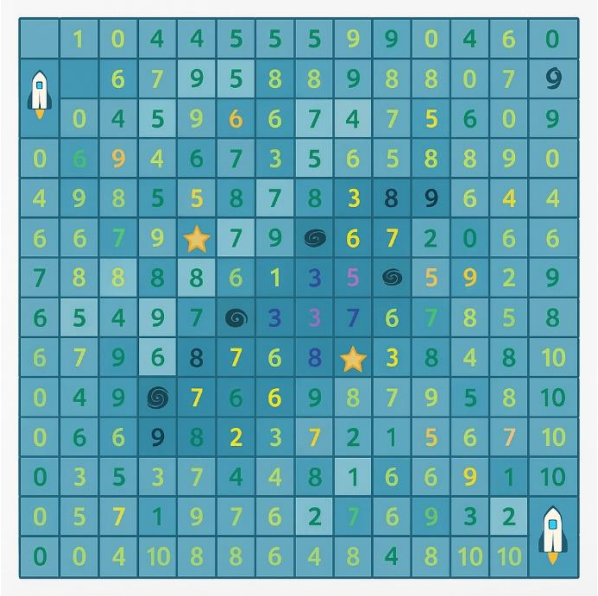

Imagen 1. Ejemplo gráfico de la matriz (solo sugerencia).

Formato de entrada JSON básico: (Adjunto al enunciado del proyecto se entrega un json de prueba que puede ser modificado según las necesidades de cada equipo).

```
{
  "matriz": { "filas": 35, "columnas": 40 },  "origen": [0, 0],  "destino": [34, 39],
  "agujerosNegros": [[3,5],[10,20],[8,8]], "estrellasGigantes": [[7,7],[14,14],[20,20]],
  "portales": [{ "desde": [5,10], "hasta": [25,30] }, { "desde": [12,3], "hasta": [2,39] }],
  "agujerosGusano": [{ "entrada": [11,11], "salida": [13,13] }, { "entrada": [18,5], "salida": [21,6] }],
  "zonasRecarga": [[4,4, 3],[15,15, 2]], "celdasCargaRequerida": [{ "coordenada": [9,9], "cargaGastada":
30 }, { "coordenada": [22,22], "cargaGastada": 22 }], "cargaInicial": 200,
“matrizInicial”: [[…],[…],[…],…,[…]]
}
```

En zonas de recarga se tendrán listas de 3 posiciones, donde la primera posición es la coordenada en X, la segunda es la coordenada en Y y la tercera es la cantidad de veces por la que se multiplica la carga actual de la nave. En dichas celdas no se aplica el gasto de energía que tenga asignada la celda.

La matriz inicial contiene la matriz NxM con las celdas llenas de números entre 1 y 10, lo que representa el gasto de energía que tiene la nave al pasar por dicha celda (el valor asignado se debe disminuir a la energía de la nave).

ucaldas@ucaldas.edu.co www.ucaldas.edu.co PBX (57)(6) 878 15 00 Calle 65 # 26 - 10 | Manizales - Colombia

Módulo 2 – "AFDs en la Vida Real: Patrullas Sintácticas para Detectar Cadenas Válidas"

Diseñar y programar dos AFDs que validen **cadenas reales** con **estructuras complejas**, basadas en reglas sintácticas rigurosas, pero procesables sin memoria adicional (es decir, sin necesidad de pilas ni gramáticas contextuales). El AFD debe ser capaz de determinar si una cadena respeta todas las reglas del formato que representa.

Reglas del Proyecto:

- Elige **2 de los 10 tipos de cadenas** listadas abajo.
- Para cada tipo:
  - Define el AFD formalmente: Σ, Q, q_0, F, δ
  - Dibuje el grafo o implementa su matriz de transiciones
  - Crea un programa que lea un archivo.txt con varias cadenas (una por línea) y diga si cada una es válida o no.
  - En caso de error: mostrar el número de línea, el carácter donde falla (si es posible), y la causa del error.

Listado de 10 patrones reales complejos (elija 2):

1. Número de tarjeta de crédito (16 dígitos agrupados 4 a 4, seguido de fecha de vencimiento, y finalizando con los tres dígitos del código de verificación)

- Formato: dddd dddd dddd dddd mm/aaaa cvv (donde d es dígito, m es mes, a es año y cvv es el código de verificación)
- Requiere validar espacios exactos y longitud
- Ejemplo válido: 1234 5678 9012 3456 03/2029 336

2. CURP de México (18 caracteres con estructura fija)

- Formato: AAAAmmddHXXCCCNNNN
  - 4 letras iniciales (mayúsculas)
  - 6 dígitos de fecha de nacimiento
  - 1 letra para sexo (H o M)
  - 2 letras para estado
  - 3 letras internas
  - 2 dígitos + 1 dígito verificador
- Ejemplo válido: GARC980512HDFLNS09

3. Dirección IPv4

- Formato: d.d.d.d (cada d es entre 0–255
- Regla básica: 4 bloques de 1 a 3 dígitos (hasta 255), separados por punto(.).
- Ejemplo válido: 192.168.100.1

ucaldas@ucaldas.edu.co www.ucaldas.edu.co PBX (57)(6) 878 15 00 Calle 65 # 26 - 10 | Manizales - Colombia

4. Placa vehicular con formato extendido

- Formato: ABC-1234-Q
  - 3 letras mayúsculas, guion, 4 dígitos, guion, una letra para tipo de vehículo
- Ejemplo válido: PQR-1298-M

5. Clave de producto alfanumérica estructurada

- Formato: XX-1234-YYZ
  - 2 letras, guion, 4 dígitos, guion, 2 letras y un dígito
- Ejemplo válido: AB-8472-XY3

6. Expresión aritmética completamente balanceada (con operandos y operadores)

- Solo expresiones como: num op num op num
- Formato: 123+45-7 o 4*5/2+1, sin espacios, mínimo 3 operandos y 2 operadores
- Solo se aceptan dígitos y + - * / en alternancia válida
- Ejemplo válido: 2+4*7/3-1

7. Formato de matrícula universitaria

- Formato: AÑO-TIPO-CÓDIGO
  - 4 dígitos de año, guion, dos letras (tipo de carrera), guion, 4 dígitos
- Ejemplo válido: 2023-IS-0841

8. Formato de código de seguimiento logístico internacional (12 caracteres)

- Formato: 2 letras + 9 dígitos + 1 letra final
- Ejemplo válido: BR123456789Z

9. Número de serie de producto electrónico

- Formato: SN-LLNN-YYMM
  - SN-, 2 letras, 2 números, guion, año y mes de fabricación
- Ejemplo válido: SN-AK45-2405

10. Formato de licencia de conducir por estado

- Formato: EST-XXXX-LL
  - 3 letras del estado, 4 dígitos, 2 letras
- Ejemplo válido: GYE-2847-AZ

ucaldas@ucaldas.edu.co www.ucaldas.edu.co PBX (57)(6) 878 15 00 Calle 65 # 26 - 10 | Manizales - Colombia