

# Algoritmos y Programación II

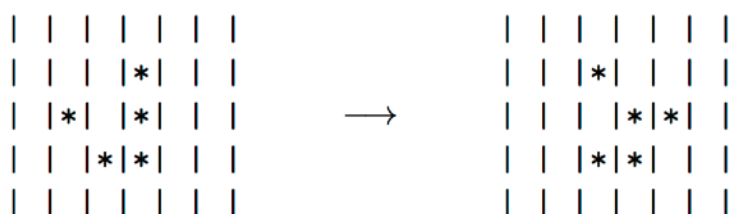
## Proyecto 1 (Grupos de 2)

### Juego de la vida

El juego de la vida , inventado por John Conway en 1970 , es un ejemplo de un "juego" cero - jugador conocido como un autómata celular . El juego consiste en un mundo de dos dimensiones que se extiende infinitamente en todas las direcciones , divididos en " células". Cada célula esta "muerta" o "viva" en una "generación". El juego consiste en un conjunto de reglas que describen cómo las células evolucionan de generación en generación. Estas reglas permiten calcular el estado de una célula en la próxima generación como una función de los estados de sus células vecinas en la generación actual. En un mundo en 2-D , los vecinos de una célula serán aquellas otras células verticales, horizontales y diagonales adyacentes a esa celda. El conjunto de reglas que rigen el juego de la vida son las siguientes:

1. Una célula viva con menos de dos vecinos vivos muere.
2. Una célula viva con más de tres vecinos vivos muere también.
3. Una célula viva con exactamente dos o tres vecinos vivos vive.
4. Una célula muerta con exactamente tres vecinos vivos vuelve a la vida.

Ejemplo



*Referenciado de: Massachusetts Institute of Technology (2010)*

### Requerimientos

Se desea que usted implemente el juego de la vida en lenguaje C bajo los siguientes parámetros:

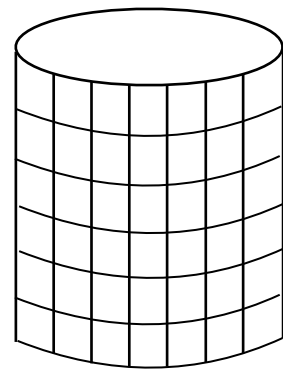
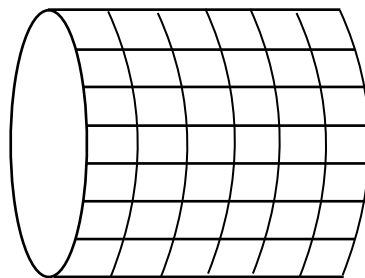
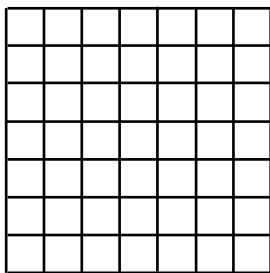
1. Se solicitará al usuario, desde el terminal, el nombre del archivo que contendrá la matriz de células inicial, el número de generaciones que se desea simular y el tiempo (en milisegundos) que ha de transcurrir entre una generación y otra.

2. El archivo que contiene la generación inicial tendrá el mismo formato de inicialización de una matriz en c. Donde los elementos marcados con cero representan las células muertas y los marcados con 1 representan las células vivas. En ningún momento se especificara explícitamente el tamaño del tablero por lo que este debe ser dinámico y debe ajustarse a lo que este contenido en el archivo dado. De igual forma pueden existir espacios de cualquier tipo en el archivo.

Ejemplo:

$\{\{0,0,0,1, \dots ,0,1\} , \dots , \{0,0,0,1, \dots ,0,1\}\}$

3. En el terminal se deberán visualizar de forma simultánea tres simulaciones. La primera es un tablero clásico que siga las restricciones del juego previamente dadas, la segunda será la de un tablero gráficamente igual al anterior pero lógicamente será como un cilindro horizontal y por ultimo la tercera será similar a la segunda pero lógicamente será igual a un cilindro vertical. La segunda y tercera simulación afectara por ende los vecinos de las primeras y ultimas filas respectivamente.



4. Se deberán generar al final de la simulación 3 archivos “clasico”, “horizontal” y “vertical” con la secuencia de tableros de las tres simulaciones anteriormente mencionadas.s
5. Sea cuidadoso con el uso de memoria, evitando desperdiciarla.

La entrega del proyecto constará de un informe escrito no mayor a 5 paginas que contenga el diseño de la solución implementada, el código fuente y una serie de resultados que se detallaran con mayor precision el día de la defensa.

**El plagio de código fuente o bien del contenido del informe implica la anulación del proyecto.**

**Fecha de defensa: Lunes 22 / 06 / 2020 7:30 am**