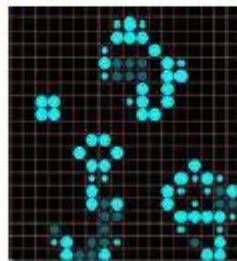
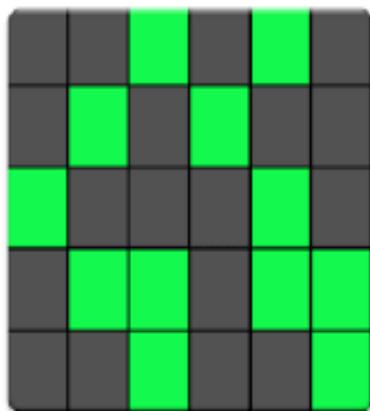


PROYECTO AUTÓMATA CELULAR ALIVE (ACA)



Descripción

La evolución de la vida tiene una variedad de patrones reconocidos que provienen de determinadas posiciones iniciales. La vida es un ejemplo de emergencia y auto-organización. Los autómatas celulares son autómatas simples que producen una salida a partir de varias entradas, modificando en el proceso su estado según una función de transición. Por lo general, en un autómata celular, al igual que en la evolución, el estado de una célula en una generación determinada depende única y exclusivamente de los estados de las células vecinas y de su propio estado en la generación anterior.



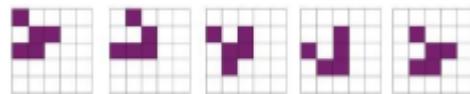
Esta simulación se basará en una matriz o tablero de un tamaño de $n \times m$ determinado (máximo 20×20), que podríamos considerar nuestro "caldo de cultivo", en la cual **mueren** y se **crean células**. Cada celda de este caldo podrá contener una célula **viva** o **estar vacía**

El estado del caldo evoluciona a lo largo de unidades de tiempo discretas o turnos. Cada turno representa una generación en la vida. El estado de todas las células se tiene en cuenta para calcular el estado de las mismas en la generación siguiente. Todas las células se actualizan simultáneamente en cada generación.

Una nueva generación será la entrada para volver a aplicar las reglas, y así sucesivamente.

Se pide desarrollar un programa AUTÓMATA CELULAR ACA que simule el proceso de evolución de un conjunto de células.

Reglas



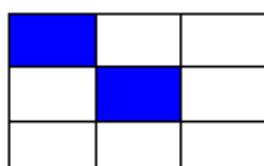
1) A efectos de aplicación de las reglas, una celda del tablero se considerará rodeada por **ocho celdas vecinas** (celdas a izquierda-derecha, arriba-abajo y las cuatro celdas contiguas en diagonal). Una celda en el borde del tablero evolucionará como si todos sus vecinos no accesibles fueran celdas vacías.

2) La supervivencia, nacimiento o muerte de las células deberá determinarse considerando sus vecinos en esa generación. Es decir, no deberán realizarse cambios sobre el tablero hasta que no se haya determinado el destino de todas las celdas en la siguiente generación.

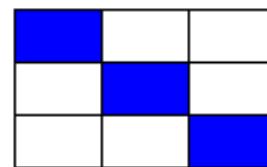
3) Las reglas explícitas de evolución del caldo de cultivo son:

- **REGLA 1:** Si una celda está ocupada por una célula y tiene **una sola célula vecina o ninguna**, esa célula **muere por soledad**.
- **REGLA 2:** Si una celda está ocupada por una célula y tiene **4 o más células vecinas**, **muere por superpoblación**.
- **REGLA 3:** Si una celda está ocupada por una célula y **tiene 2 ó 3 células vecinas**, sobrevive a la siguiente generación.
- **REGLA 4:** Si una **celda está vacía** (no está ocupada por una célula) y **tiene 3 células vecinas**, nace una nueva célula en su lugar para la siguiente generación.

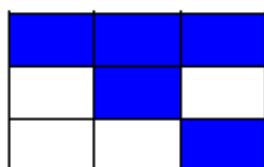
Algunos ejemplos:



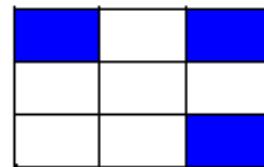
La célula central muere por soledad (1 vecino)



La célula central seguirá conteniendo una célula viva (sin cambios)



La célula central muere por superpoblación (4 vecinos)

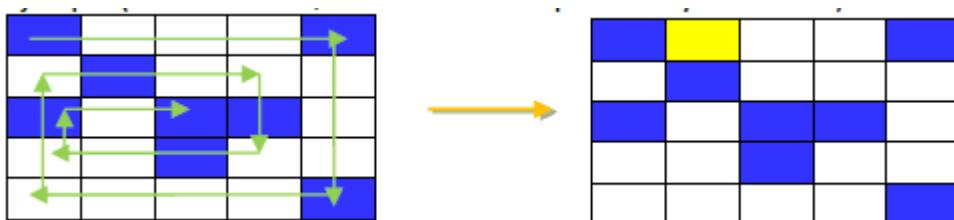


Una nueva célula nacerá en la casilla central por soledad (3 vecino)

Adicionalmente, a veces ocurre el **milagro de la vida**, una oportunidad que otorga Dios a algunas células ángel (amarillas) para nacer. Estos eventos milagrosos pueden generar un nacimiento espontáneo dependiendo de celdas libres en determinados recorridos:

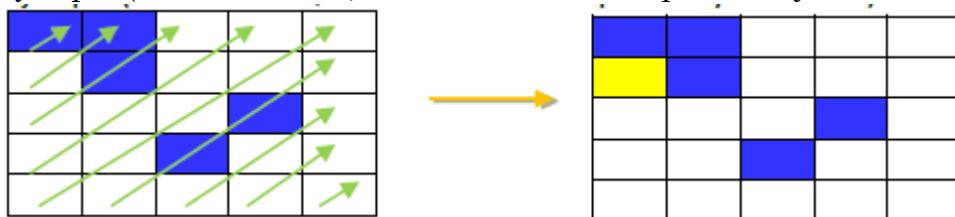
- **MILAGRO 1:** Si en un recorrido en espiral se encuentran libres el 50 % (parte entera) de las celdas que ambas coordenadas (x,y) son impares hay una celula, nace una célula ángel en la primera posición desocupada del recorrido

Ejemplo (son 25 casillas, #coordenadas impares = 9 y el 50% = 4):



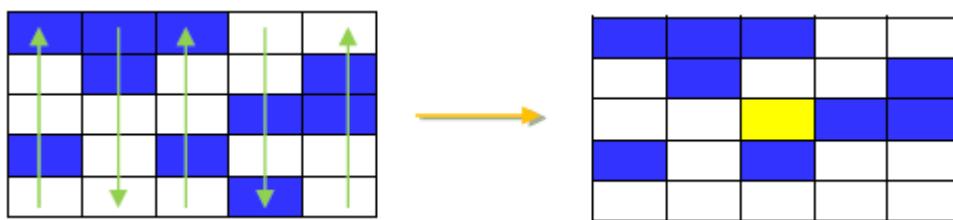
- **MILAGRO 2:** Si en el recorrido diagonal secundaria inferior, se encuentran libres el 70% (parte entera) de las celdas cuya coordenada x es par, nace una célula en la última posición desocupada del recorrido

Ejemplo (son 25 casillas, #coordenadas con x par = 10 y el 70% = 7):



- **MILAGRO 3:** Si en el recorrido zigzag vertical, se encuentran libres el 60% (parte entera) de las celdas cuya coordenada y es impar, nace una célula en la primera casilla desocupada que se encuentra en la segunda mitad del recorrido.

Ejemplo (son 25 casillas, #coordenadas con y impar = 15 y el 60% = 9):



Proyecto Final (Fecha de Entrega 7-11/06/25)

Objetivo.

Usted como programador de ACA deberá desarrollar un programa que cumpla con los siguientes requerimientos:

- Leer un archivo en disco una configuración inicial de células sobre el tablero que representa el caldo de cultivo.
- Modificar a solicitud del usuario la configuración del caldo de cultivo, eliminando células y/o creando otras nuevas.
- Generar la configuración inicial del caldo de cultivo en forma aleatoria. Para ello, el usuario introducirá por teclado el número de células vivas, el cual debe ser como mínimo “n” (número de filas de la matriz) y como máximo la cantidad de celdas de la matriz, el programa deberá distribuirlas aleatoriamente en la matriz. El usuario también pudiera elegir que el número de células vivas fuera aleatorio.
- Calcular y mostrar por pantalla la evolución del caldo de cultivo representado en la matriz, de generación en generación, en forma puntual.
- Permitir “milagros”, es decir, dar opción al usuario para ejecutar uno de los tres milagros descritos.
- Calcular y mostrar de una vez, el resultado tras X generaciones (parámetro suministrado por el usuario). Si en alguna generación anterior a X, ya no existe ni una célula viva, se termina en esa generación y el programa debe indicar esta condición.
- Finalizar ACA a solicitud del usuario
- Guardar la configuración final en un archivo de salida.

Entrada del Proyecto Final

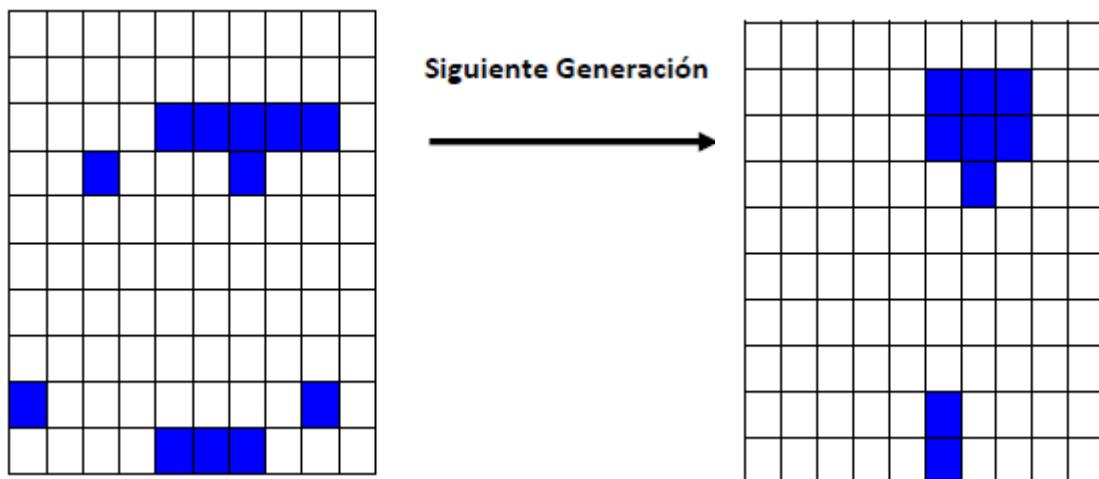
La carga inicial para ejecutar ACA, consistirá en una secuencia de líneas que se encuentran en un archivo de texto (ASCII) denominado **ACAENTRA.TXT**, que tiene el siguiente formato:

1. La línea número 1 contiene el número de filas (**n**) y columnas (**m**) del caldo de cultivo separados por una coma y sin espacios en blanco. Se puede asumir que **n** y **m** pueden ser hasta el valor 30 y no menos de 10 (aunque los ejemplos colocados son menores a 10).
2. Las siguientes líneas contendrán cada una, las coordenadas **i,j** del caldo donde se ubicarán células vivas. Se escribirá los valores “**i**” y “**j**”, separados por una coma y sin espacios en blanco
3. Adicionalmente, el programa debe permitir la entrada manual de los datos para formar un caldo de cultivo inicial.

Salida del Proyecto Final.

El resultado de la ejecución de ACA se mostrará por pantalla, pero adicionalmente se creará un archivo de texto (ASCII) con nombre **ACASALI.TXT** que contendrá una línea con cada coordenada **i, j** del caldo en donde quedaron células vivas después de la simulación con ACA. Se escribirá los valores “**i**” y “**j**”, separados por una coma y sin espacios en blanco.

Ejemplo:



Ejemplo de entrada
ACAENTRA.TXT

10,10
3,5
3,6
3,7
3,8
3,9
4,3
4,7
9,1
9,9
10,5
10,6
10,7

Ejemplo de salida
ACASALI.TXT

2,6
2,7
2,8
3,6
3,7
3,8
4,7
9,6
10,6

Requisitos para la evaluación del proyecto:

1. Equipos de máximo, cuatro personas de la misma sección de Teoría.
2. La entrega del Proyecto se hará a través de Módulo 7, en los espacios destinados para tal fin (**NO se aceptarán proyectos entregados por otra vía**).
No se aceptarán proyectos en fecha de entrega posterior a la prevista.
3. La corrección exige la presencia de **TODOS** los integrantes del equipo
4. Deben estar programados **TODOS** los requerimientos

IMPORTANTE : *Incurrir en alguno de los puntos anteriores, traerá como consecuencia **REPROBAR** el Proyecto y solo en situaciones **MUY PARTICULARES** y **ESPECIALES**, se podría considerar la corrección del mismo pero en base a una **Nota MAXIMA** de 12 puntos.*

5. Interfaz **amigable, fácil uso y entendible**.
6. Se evaluará positivamente lo siguiente:
 - Modularidad, estructuración y legibilidad del código
 - Buena documentación: cantidad y calidad de los comentarios.
 - Estructuras de datos apropiadas en la solución del problema.
 - Uso de estándares adecuados en la nominación de las variables.
 - Funcionalidad y eficiencia de los algoritmos

Requisitos para la entrega del Proyecto Final:

1. Deben entregarse los siguientes archivos :
 -) **Archivo . py** (que contiene TODO el Programa) cuyo nombre será APELLIDO de un de los integrantes del Grupo(el que tenga el apellido menos común) y la inicial del nombre del resto de los integrantes(sin dejar espacios en blanco) • <extensión>(**py**)
 -) **Archivo de entrada .txt** ([acaentra.txt](#)) que contiene la información del caldo inicial.