LABORATORIO N°1

CURSO DE MÉTODOS DE PROGRAMACIÓN

2-2020

UdeSantiago de Chile

CONTENIDO

Introducción **2**

Contexto **2**

Descripción del juego **2**

Trabajo a realizar **5**

- Características de entrada y salida 7
- Funcionalidades 8

Entregas y evaluación **9**

1. Introducción

En este documento se muestra el enunciado del primer laboratorio a desarrollar dentro del curso de Métodos de Programación, el cual consiste en una simulación de un autómata celular, llamado "*El juego de la vida*".

En este documento se explica el autómata en sí, para posteriormente indicar qué es lo que deberá desarrollar usted en este laboratorio. A continuación, se muestran las reglas que posee este trabajo y su forma de evaluación. Finalmente se informan de las fechas, los elementos entregables, la forma de hacerlo llegar y otros elementos necesarios.

2. Contexto

John Conway (1937 - 2020) es un matemático británico que lamentablemente nos dejó este año a causa del virus COVID-19, pero su legado no será fácilmente olvidado. Dentro de los trabajos que he investigaciones de John se encuentran temas acorde a la teoría de grupos, teoría de números, teoría de juegos y otros.

Dentro de los elementos popularmente más conocidos se encuentran el sistema numérico de los números surreales, el juego brotes, el phutball y el juego de la vida, siendo éste último en el cual se centrará el trabajo a realizar.

Los autómatas celulares corresponden a un modelo matemático que va evolucionando a través del tiempo, descubiertos inicialmente por John von Neumann en 1950. Estos, históricamente se pueden dividir en tres puntos históricos, los cuales son conocidos como la Era de Von Neumann, la Era de John Conway y la era de Stephen Wolfram, siendo cada uno de ellos un científico que marcó un cambio drástico en este campo.

Para la época de John Conway, en los años '70, se destaca el juego diseñado por él, y es conocido como "El juego de la vida", en el cual se centrará el desarrollo de este trabajo, el cual posee reglas simples, pero puede generar elementos muy divertidos.

3. Descripción del juego

El juego de la vida consiste en la representación de la interacción entre células en un espacio dado. Para la representación del espacio se utiliza una cuadrícula ortogonal bidimensional compuesta por espacios cuadrados, en donde cada uno de los espacios representa a una célula la cual puede tener solo dos estados, estar viva o estar muerta.

A través del tiempo las células van generando nuevas células vivas (reproducción), van muriendo otras (sobrepoblación) y otras se mantienen vivas o muertas, según correspondan las reglas que se coloquen.

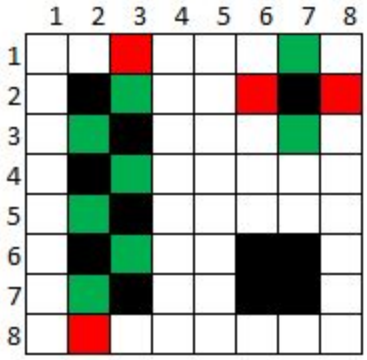
Las reglas del juego de la vida de Conway se va evaluando por edad o época, y son distintas para las células vivas y muertas. Estas reglas son:

- Para las células vivas en la :
  - Si en la época n la célula posee menos de 2 células vecinas, esta célula morirá para la época siguiente, por falta de población.
  - Si en la época n la célula posee 2 o 3 células vecinas, esta célula vivirá para la época siguiente.
  - Si en la época n la célula posee más de 3 células vecinas, esta célula morirá para la época siguiente., por sobrepoblación.
- Para las células muertas:
  - Si en la época n la célula posee 3 o más células vecinas, esta vivirá para la próxima generación, por reproducción.

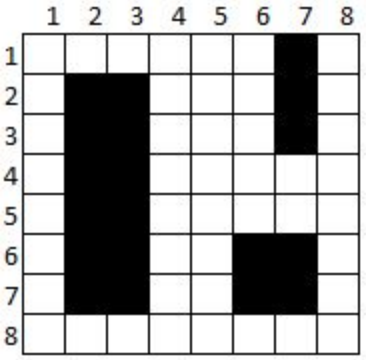
Para una explicación extra del juego, se les recomienda también ver el vídeo del canal de YouTube “Derivando” llamado “¿Conoces el juego de la vida?”[1].

De esta forma, por cada generación la distribución de la población general de células va variando generación en generación, por ejemplo, para una población distribuida en una matriz de 8x8, como se muestra en la Tabla N°1.

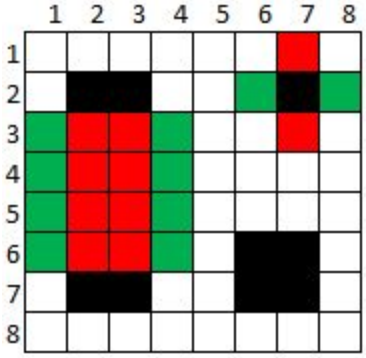
[1] El link directo al vídeo es:
https://www.youtube.com/watch?v=OWXD_wJxCKQ&ab_channel=Derivando



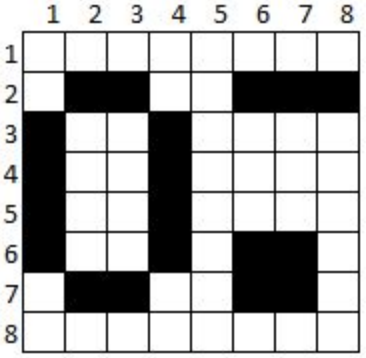
Generación N°2 (Cambios)



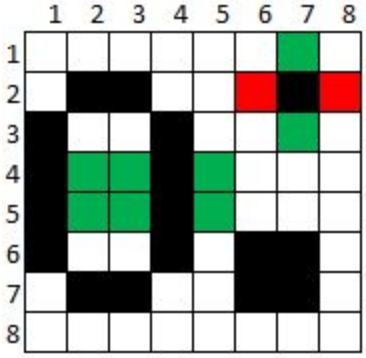
Generación N°2 (Final)



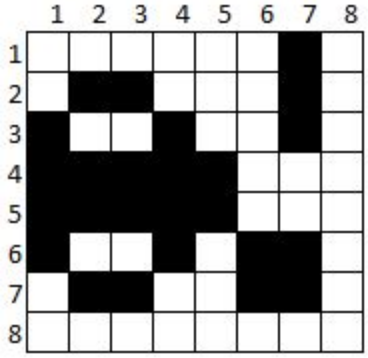
Generación N°3 (Cambios)



Generación N°3 (Final)



Generación N°4 (Cambios)



Generación N°4 (Final)

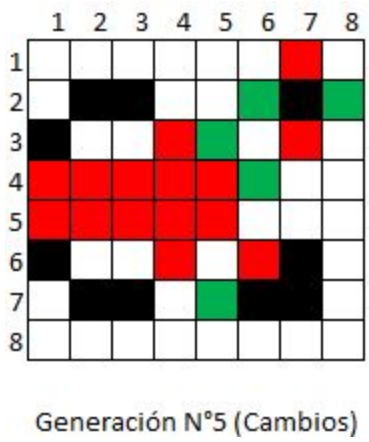
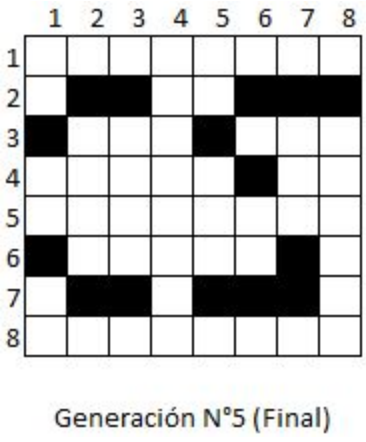
 Generación N°5 (Cambios)	 Generación N°5 (Final)

Tabla N°1: Proceso para 5 generaciones de una población de células iniciales

En la Tabla se puede apreciar como desde una población inicial de células distribuidas en una matriz de 8x8, donde las celdas en negro representan a las células vivas, mientras que las en blanco representan las células muertas, se ve como en cada una de las representaciones del tablero, llamadas *Generación N°X (Final)* (todas ubicadas al lado derecho de la tabla) el cómo se aplicaron las reglas antes mencionadas. Las imágenes llamadas *Generación N°X (Cambios)* (todas ubicadas al lado izquierdo de la tabla) sirven para mostrar cuales son las células que viven, nacen o mueren acorde a las reglas.

4. Trabajo a realizar

El trabajo a realizar consiste en crear un programa que permita realizar una simulación del Juego de la Vida de Conway a partir de una matriz de células vivas y muertas.

Para el desarrollo del trabajo, la matriz será entregada por el usuario mediante un archivo de texto llamado "Entrada.in" el cual se compone de lo siguiente:

1. La primera línea del archivo corresponde al tamaño de la matriz que representará la población. Este será siempre un valor numérico resultado de una potencia de 2.
2. Las siguientes n líneas representarán a las células células vivas y muertas, para esto se representará cada línea también con n caracteres de solo "_" (guión bajo) o "X" (carácter mayúscula x), donde los "_" representarán a las células muertas y las "X" son las vivas.

El usuario, mediante pantalla indicará cuántas generaciones deben pasar para el juego de Conway y el programa deberá escribir un archivo de texto mostrando el resultado final para la cantidad de generaciones que pasaron. El nombre del archivo de texto debe indicar la cantidad de generaciones que han pasado.

Como ejemplo, para una entrada de una matriz de 8x8 el archivo de entrada debe ser como el mostrado en la Figura 1. Mientras que en la Figura 2 muestra el resultado de la entrada en la generación 2 y la Figura 3 en la generación 4.

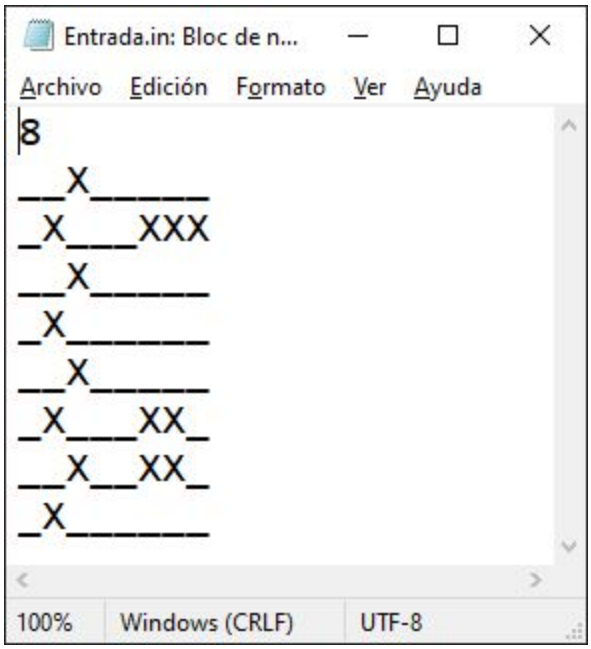


Figura 1: Ejemplo de entrada.

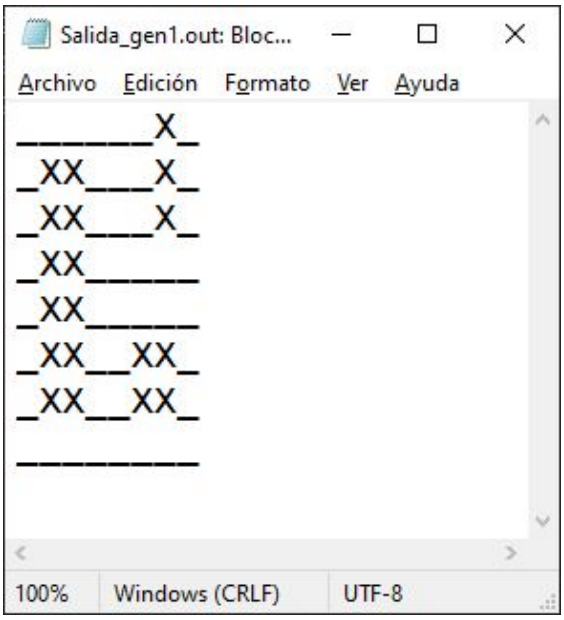


Figura 2: Salida para la 1° generación de la entrada de la Figura 1.

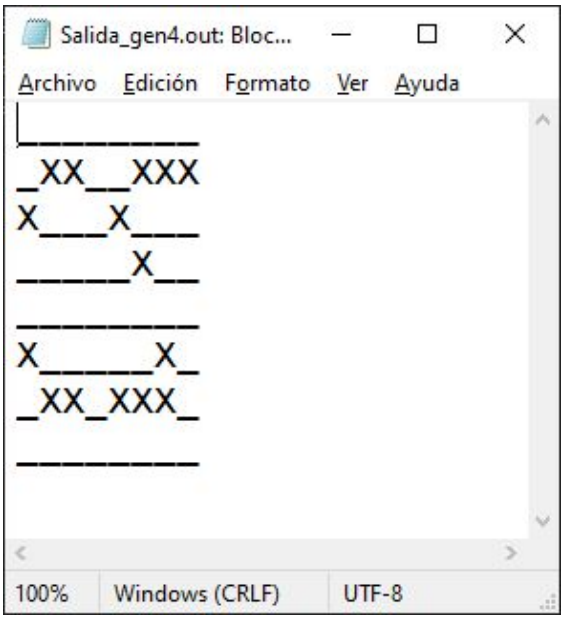


Figura 3: Salida para la 4° generación de la entrada de la Figura 1.

4.1. Características de entrada y salida

Dentro del desarrollo del trabajo se deben considerar explícitamente los siguientes elementos para su desarrollo.

Para los archivos se debe considerar qué:

- El archivo de texto siempre se llamará “Entrada.in”, y se encontrará en la misma carpeta donde se ubica el ejecutable de su programa.
- El archivo de salida se debe crear en la misma carpeta donde se encuentra el ejecutable de su programa.
- El nombre del archivo de salida debe ser “Salida_gen<X>.out”, donde X es la generación señalada por el usuario. De esta forma la salida para la quinceava generación deberá llamarse “Salida_gen15<X>.out”.

Con respecto a las generaciones, el archivo de entrada corresponderá a la Generación N°1, por lo que si el usuario indica que desea una sola generación, se deberá mostrar el mismo archivo como resultado.

La cantidad de generaciones a ejecutar serán señaladas por el usuario por entrada estándar.

4.2. FUNCIONALIDADES

Para el desarrollo del programa es necesario que se cumplan con el siguiente listado de funcionalidades:

1. El programa debe estar escrito en lenguaje de programación C.
2. El programa debe funcionar para Sistemas Operativos Unix, IOS y Windows.
3. La validación de la entrada: Se debe validar tanto el archivo de entrada, que cumpla con las características solicitadas en su contenido y se debe validar la cantidad de generaciones que se deben ejecutar. Alguno de los elementos a validar son[2]:
    a. Que la primera línea del archivo de entrada sea un valor numérico potencia de 2.
    b. Que la matriz del archivo de entrada contenga solo "_" y "X" y sea la cantidad de caracteres indicados en la primera línea.
    c. Que la cantidad de generaciones indicadas por el usuario sea un número entero positivo.
4. El programa debe poseer un menú, en el cual solicite al usuario la cantidad de generaciones a ejecutar o salir del programa.
5. El programa debe informar al usuario cuando finalice su ejecución y genere el archivo de salida.
6. El programa debe darle al usuario la opción de mostrar la evolución de generación en generación por pantalla, hasta la generación indicada por el usuario, señalada en el punto 4.
7. El programa debe funcionar para cualquier archivo de entrada que cumpla con las características mencionadas en el enunciado.
8. Para el paso de generación en generación se debe hacer uso de una función que utilice **Recursión** para solucionar el tema.
9. Para el cálculo de si una célula nace, vivirá o morirá se debe hacer utilizando **División y conquista** sobre el tablero. Considere para División y conquista el concepto vistos en clases y disponible dentro del curso de moodle en UdeSantiagoVirtual.

[2] Dentro de la revisión de su programa pueden ser considerados otros elementos a validar, que estén dentro de lo solicitado en el enunciado.

5. Entregas y evaluación

La entrega consiste en la subida al portal del curso de Métodos de Programación, en el sitio www.udesantiagovirtual.cl, en la actividad Entrega Laboratorio Individual N°1 el día **12 de diciembre del año 2020 a las 23:55 horas**.

En específico se deberá entregar un archivo comprimido en formato .zip o .rar que:

- Que tenga como nombre la siguiente estructura **<Sección Laboratorio>_<RUN>**:
  - Primero debe ir la sección del laboratorio, por ejemplo, L23.
  - Luego el RUN del creador del código, por ejemplo, 12345678-9
  - Entonces el nombre final será L23_12345678-9
- El código fuente del programa en extensión .c
- Los archivos de entrada de prueba de su programa, que cumplan con el formato solicitado.
  - Para esto los archivos deben llamarse Prueba1.in y además, debe incluir el archivo de prueba dado por la coordinación.
- Un archivo de texto plano llamado README, el cual debe indicar un paso a paso el cómo funciona su programa.
  - Este es un mini manual de usuario, explicando a grandes rasgos el cómo funcionan ciertos elementos, por ejemplo, si hay un menú, indicar si las opciones se ingresan de forma numérica, palabras, si estas van en mayúsculas o minúsculas, etc. Si el programa tiene que leer o escribir o interactuar con un archivo de texto, dónde debe estar ubicado. Si la interacción con el usuario es mediante entrada y salida estándar o mediante alguna interfaz gráfica.

Para la evaluación, se revisará en específico los siguientes puntos, con nota de 0 a 6 para cada uno, y el porcentaje correspondiente a la ponderación que tendrá cada punto, el cual está indicado entre paréntesis, posteriormente se sumará el punto base a su evaluación:

- Lectura y escritura del archivo de entrada de forma correcta (5%).
- Uso de menús (10%).
- Evolución de generación en generación de la población inicial de células acorde a las reglas declaradas en el enunciado y la aplicación correcta de los métodos a utilizar (25%).
- Generación de los archivos de salida correspondientes (20%).
- Uso de buenas prácticas, en dónde (10%):
  - El código fuente debe tener como comentario, al inicio de él, la siguiente información:

        - RUN del creador del código.
        - Sección del laboratorio a la cual pertenece el creador del código.
        - Fecha de creación del código.
    - Las funciones creadas deben indicar:
        - Entradas: Qué significa cada una de las entradas de las funciones, por ejemplo, si se tiene una función suma que recibe dos números enteros, indicar que la entrada corresponde a los operandos de la operación adición.
        - Salida: Qué significa la salida generada, y en caso de tener salidas que representen valores del tipo Verdadero o Falso, opciones 1, 2, 3, …, n; indicar qué significa que salga cada una de éstas.
        - Objetivo de la función.
    - Uso de nombres de variables representativos.
    - Uso de identación y orden del código.
- Pruebas (30 %):
    - Dos pruebas dadas por el equipo docente (5% ambas):
        - Ambas quedarán disponibles en el curso de moodle de UdeSantiagoVirtual.
    - Dos pruebas ocultas realizadas por el equipo docente (15% ambas):
        - Las cuales no serán mostradas hasta después del momento de revisar.
    - Dos pruebas entregadas por el creador del código (10% ambas)
        - Estas deben ir incluidas como ejemplos del archivo de entrada, dentro del archivo comprimido a subir.
    - La validez de las prueba no consistirá solamente en mostrar o dar el resultado solicitado, sino que también la aplicación de los métodos de resolución de problemas solicitados, en caso de aplicar el método solicitado el puntaje es válido, incluso cuando el método sea aplicado de forma errónea. En caso de no ver la aplicación de los métodos solicitados se considerará que no son pruebas válidas, por lo que serán evaluadas con puntaje 0.