# Game Of Life - Proyecto Final ICC

David Hernández Uriostegui

December 6, 2019

## 1 Desarrollo de la solución

En esta sección se explica en que consistió el proyecto y su solución.

El proyecto consiste en implementar una versión del ya viejo y conocido "Game of Life" diseñado por el matemático John Horton Conway el cual a partir de un automata celular y un conjunto de reglas diseñó un programa cuyo principio comenzó aplicarse en varias areas, una de ellas, el procesamiento de imagenes. El proposito del proyecto es que a partir de una imagen pudieramos manipular los pixeles, siendo esto lo equivalente a los automatas celulares, e igualmente a partir de un conjunto de reglas en base a los colores de la imagen se determinaba si un pixel vivía o moría. Cabe aclarar que se modifocó un poco las reglas de juego, en un momento más se explicará. Para poder darle solución al problema lo que se hizo fue crear 2 clases principales: La clase Cell y la clase Board, también las exepciones que se mencionarán.

**Cell:** Esta clase modela un pixel, es decir tiene como atributos los colores RGB que tiene un pixel y como estamos implementado una versión del juego de la vida, entonces también un atributo de estado que indica si está vivo o muerto, dependiendo de las reglas dadas.
**Board:** Este clase modela una matriz de Cells, la cual nos servirá para poder iterar sobre la imagen y ver el cambio de estado de los pixeles.

**NoImage:** Esta exepción nos permite saber si no se cargó una imagen o si la dirección de la imagen no se encuentra sobre el directorio que se necesita.

**SizeException:** Esta exepción nos indica que la imagen que si engresó es menor a 500x500 pixeles, y nos abre otra por defecto.

**InvalidFormat:** Esta exepción nos indica que la "imagen" ingresada no tiene fomato PNG, JPG o JPEG y nos carga otra por defecto

## 2 Herramientas utilizadas

Las herramientas utlizadas para la realización del proyecto son las que se mencionan a continuación:

1. Lenguaje de programación Java

2. Entorno gráfico Processing.

3. Imagenes de minimo 500x500 pixeles para realizar prubas

# 3 Estructura del proyecto

Para poder empezar con el proyecto, se tuvo que pensar en que clases deberían ser implementadas para que el proyecto fuera funcional; como se mencionó antes estas fueron:

1. Cell

2. Board

3. Sketch(aquí es donde se implementan los métodos para que comience la ejecución).

4. NoImage

5. SizeException

6. InvalidFormat

Como se mencionó antes la clase Cell modela un pixel de una imagen, y por lo tanto tiene color y un estado(vivo o muerto) y este se determinaba si la suma de sus componentes de color(rojo, azul y verde) era mayor o igual 500 el pixel está vivo, en otro caso está muerto.

Ahora bien, tenemos nuestra clase Board que modela una matriz de pixeles sobre la cual se va a iterar, y en esta clase tenemos un metodo llamado sumaColoresV() el cual permite calcular la suma de los colores de los pixeles que están alrededor de un pixel, y aquí entra otras reglas:

- Si el pixel está vivo y sumaColorV() oscila entre 1000 y 2295 el pixel sigue vivo, en caso de que no oscile entre estos valores al pixel se le sustrae una cantidad aleatoria sobre un canal aletorio hasta que el pixel muera

- Si el pixel está muerto, pero sumaColorV() oscila entre 500 y 755, entonces el pixel incrementa un valor aleatoria a cualquiera de sus canales hasta que el pixel renazaca

Por último en el Sketch se realizaron varios métodos para implementar las exepciones y así tratar que el programa fuera un programa lo suficentemente robusto, así como métodos para que se iterara sobre la imagen y los pixeles pudieran comenzar a aplicar las reglas, en estos métodos se usó 2 for para así poder iterar sobre casa pixel de la imagen

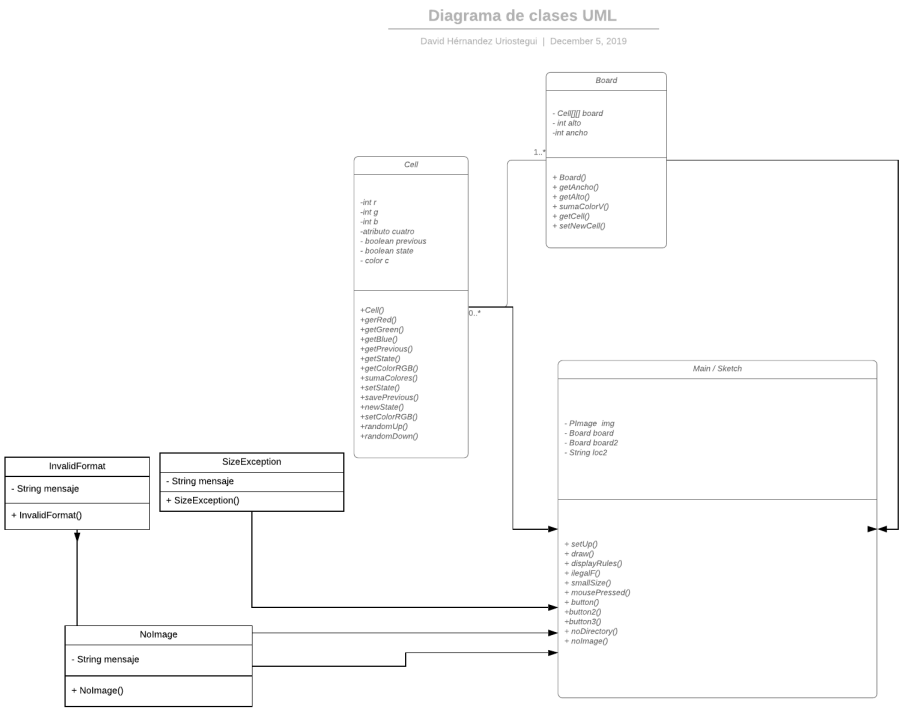


Figure 1: Diagrama UML del proyecto

# 4 Resultados obtenidos

A continuación se presentan los resultados obtenidos después de meter la imagen al programa

Figure 2: Figura original 1

Figure 3: Figura obtenida 1

Figure 4: Figura orginal 2

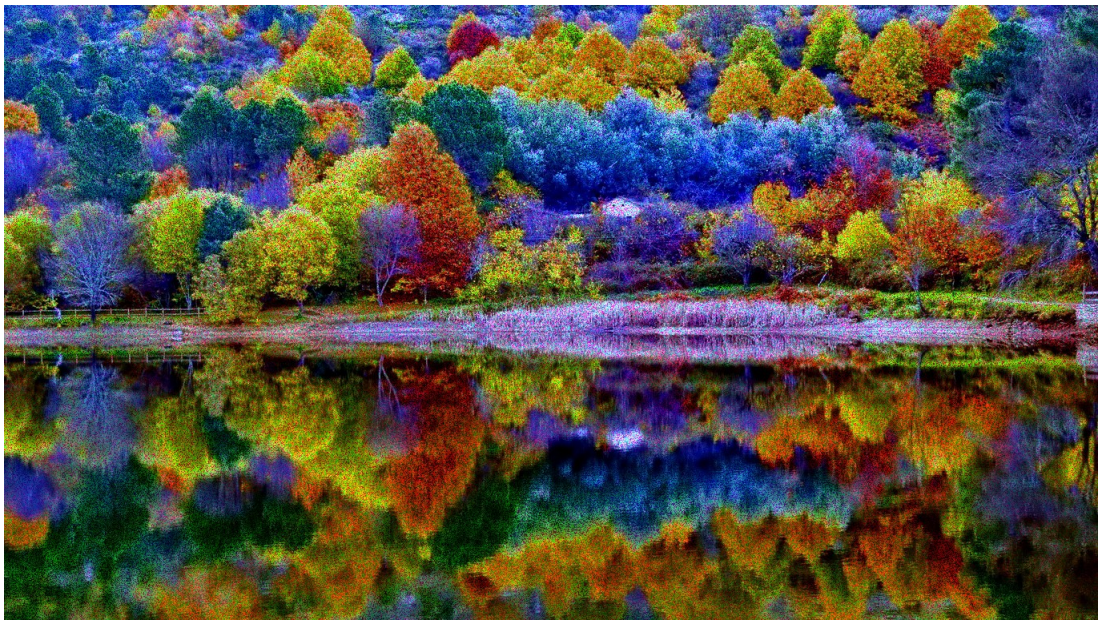

Figure 5: Figura obtenida 2

# 5 Conclusiones

Bueno como consluiones puedo decir que:

- Al poder iterar sobre cada pixel, esto nos grandes posibilidades de implementar algoritmos de procesamineto de imagenes más complejos
- Al aumentar el rango del valor aleatorio de aumento y descenso los pixeles tienden a tomar un color más oscura por lo cual se optó por dejarlo así, ya que se nota de mejor manera el cambio. Y a esto se referia que cambió un poco las reglas.
- Al usar estas reglas los pixeles no tienden a dar como resultado algo como lo que se mostró, pero tal vez las reglas básicas del Juego de la Vida podría mostrarse una imagen así, ya que uno de mis compañeros nos lo mostró.
- En las imagenes que no tiene mucha variedad de color no se nota de gran manera las iteraciones.
- Si tomaramos el brillo de cada pixel, tal vez podríamos obtener resultados más intersantes.
- Por último me parecec que haber hecho las métodos principales en el Sketch me permitió una mayor facilidad de realizar el proyecto.

# 6 Bibliografía

Para finalizar me gustaría decir en que bibliografía me basé para poder realizar el proyecto, de hecho sólo es una:

- Canal de YouTube: The Coding Train
- GitHub de The Coding Train, repositorio del clasico Game of Life
- En particular a partir del repositorio de The Coding Train me basé para hacer el metodo sumaColorV()