

Juan Pablo Garzon Parra
Nicolas Ricardo Valderrama

Ejercicios 2 documento

1. Observe sus comportamientos en la casa, en la universidad y en el medio de transporte que utiliza. Encuentre, para cada uno de estos escenarios sus reglas básicas.

Las reglas básicas para los diferentes entornos en los que nos desarrollamos, como lo dice la lectura, se ven impuestos por el concepto de "vecindad", significa que todo depende de nuestros comportamientos y de los comportamientos de las personas de nuestro entorno, algunas de las principales reglas básicas que se ven aplicadas en los escenarios planteados son:

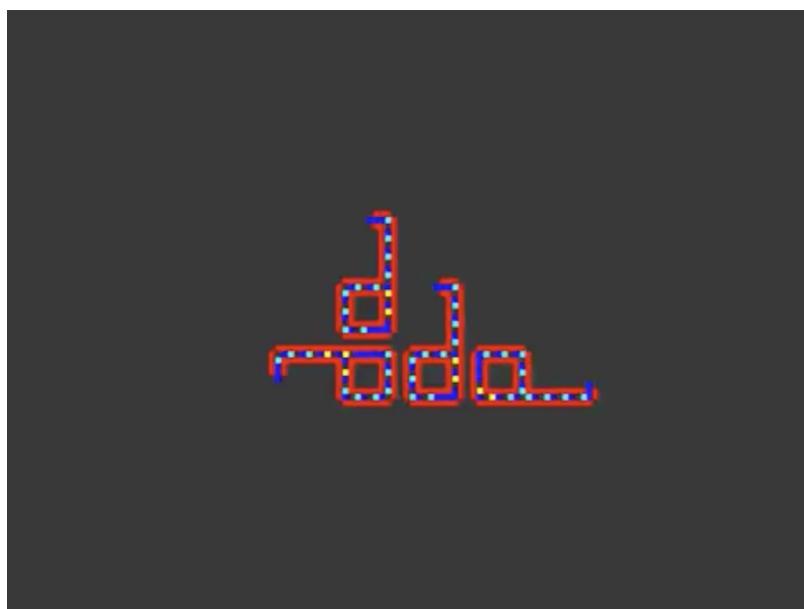
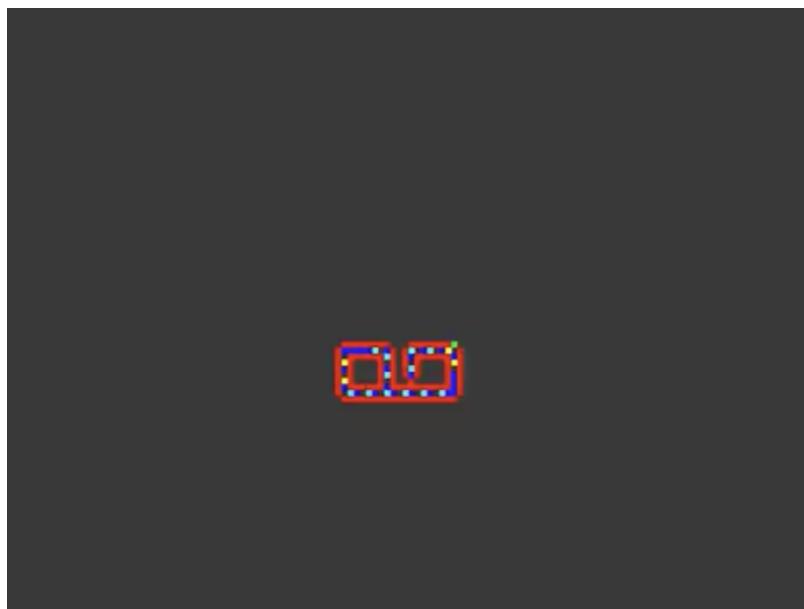
- Casa:
 - Cerrar con llave las puertas todo el tiempo por seguridad.
 - No desperdiciar luz o agua.
 - Procurar tener en orden los espacios comunes como la cocina o el baño.
- Universidad:
 - Respetar la opinión y punto de vista de los demás.
 - Tener las pertenencias (como la maleta) a la vista siempre.
 - Asegurar la bicicleta con candado.
 - Estar preparado para desalojar cuando hay manifestaciones.
- Transporte (a pie):
 - Estar precavido todo el tiempo.
 - Caminar rápido.
 - No usar el celular en la calle mientras camina.

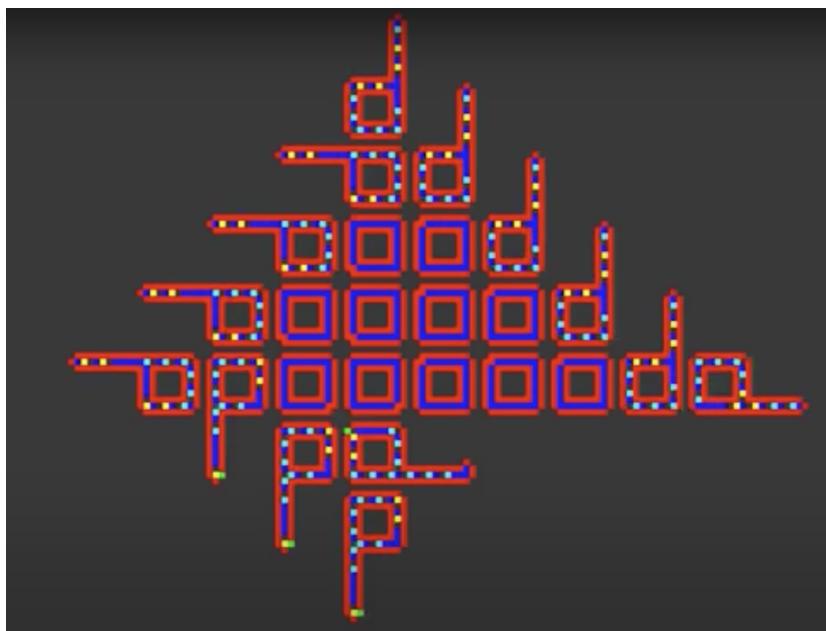
2. En la librería de modelos de NetLogo, encuentre una aplicación de AC, describa el modelo, córralo y haga un análisis del resultado.

En los modelos de la comunidad de NetLogo encontré el modelos de "**Loops de Langton**", estos loops representan una forma específica de vida artificial en un autómata celular, estos loops constan de una serie de células con información genética que circula de manera constante a lo largo de un bucle central y a través de un apéndice similar a un brazo, el cual eventualmente se transformará en un bucle descendente. Los "genes" programan la ejecución de tres giros hacia la izquierda, dando forma al bucle principal, que posteriormente se separa de la estructura madre.

Dentro de la Biblioteca de Modelos de NetLogo existe una categoría llamada "Cellular Automata", dentro de esta categoría se encuentran el modelo de "Brian's Brain", "Life" y "Life Turtles-Based", estos dos últimos modelos pertenecientes al juego de la vida.

Modelo en ejecución:





Al obtener los resultados de la simulación encontramos que esta genera vida artificial, creando diferentes células nuevas y formando patrones muy complejos partiendo de un conjunto de reglas muy simples, encontramos el cambio de estado de las células (activa o inactiva), este modelo puede ser fundamental como partida para el estudio de conceptos de autorganización, vida artificial y comportamiento emergente en la ciencia y la computación.

4. Tome una fotografía a color, conviértela en una fotografía en tonos grises. Ahora, toma cada píxel de la fotografía en grises y cada píxel de la fotografía a color. Aplique una vecindad de Moore de 9 vecinos al pixel gris y obtenga la regla para predecir el pixel a color. Guarde todas las reglas encontradas y con base en esas reglas, vuelva a generar la fotografía a color. Ahora, tome una foto antigua de su casa y conviértela a color con base en las reglas que encontraron.

Obtuvimos buenos resultados al realizar la nueva imagen, pero lo que hicimos fue promediar los resultados que se guardaban en el conjunto de reglas, pues al guardarlas exactas como estaba en la vecindad de cada píxel, no se coloreaba ninguna imagen diferente a la entrenada.

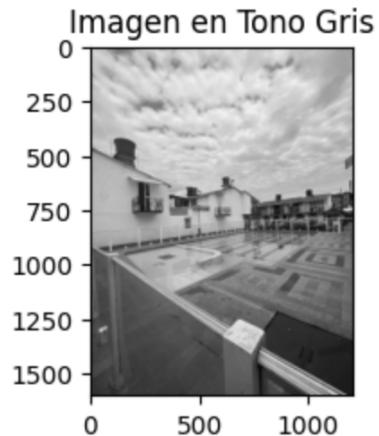
Esta es la imagen original con la que se obtienen las reglas:

```
↳ Text(0.5, 1.0, 'Imagen a Color Original')
```



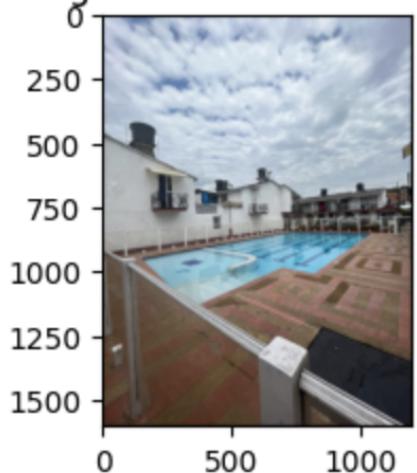
La misma en tono de grises:

```
Text(0.5, 1.0, 'Imagen en Tono Gris')
```



La imagen reconstruida:

```
Imagen a Color Reconstruida
```



Ahora vamos a probar con la imagen de la casa:



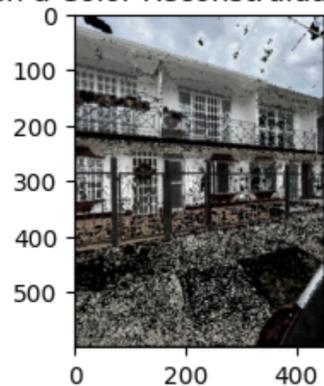
La casa en tono de grises:

```
↳ Text(0.5, 1.0, 'Casa en Tono Gris')
```



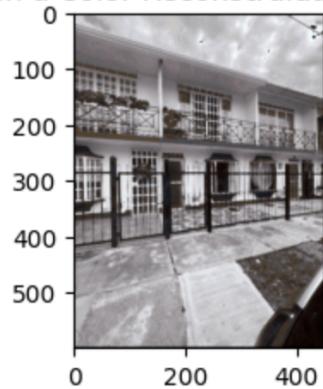
Al hacer la reconstrucción de la imagen con el conjunto de reglas encontramos dos formas, en la primera solo hallamos el promedio obtenido, esto hace que la imagen tenga los colores exactos pero que no se coloree completamente:

Imagen a Color Reconstruida de la casa



La segunda opción fue usar la función techo sobre el promedio, lo que nos coloreaba toda la imagen pero con una paleta de colores sepia:

Imagen a Color Reconstruida de la casa



Link al notebook de python: [🔗 T2Imagen.ipynb](#)