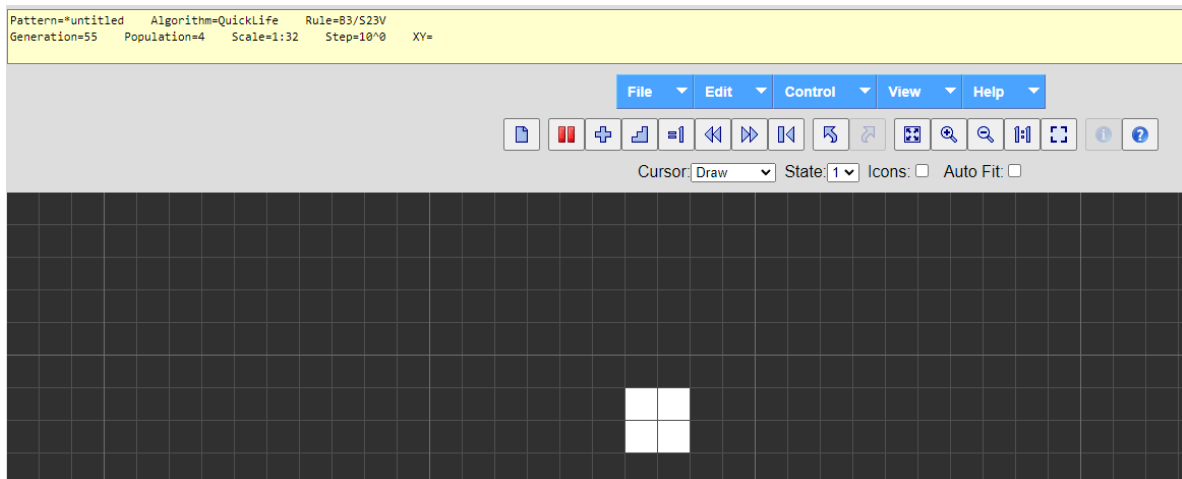
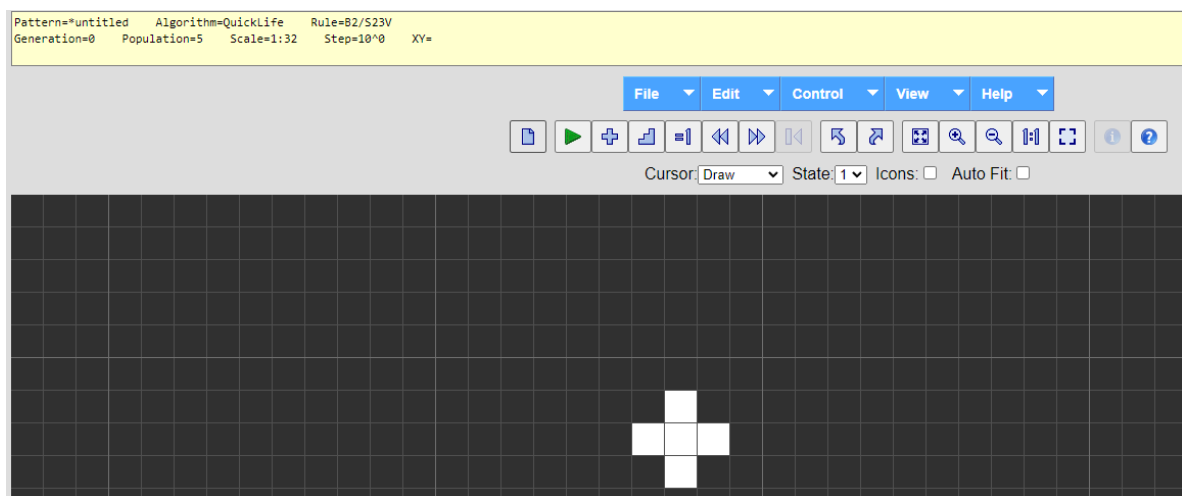
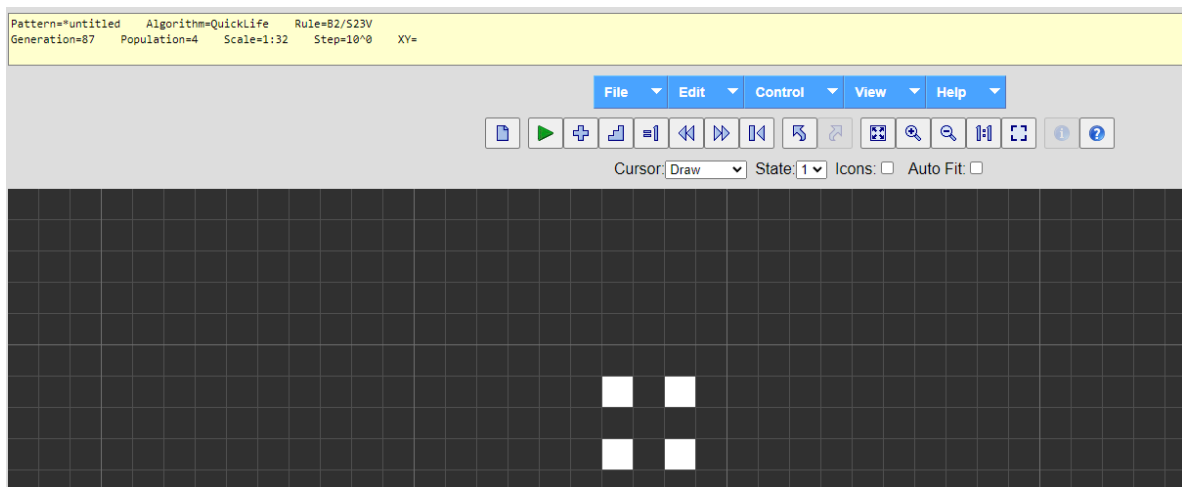


3.

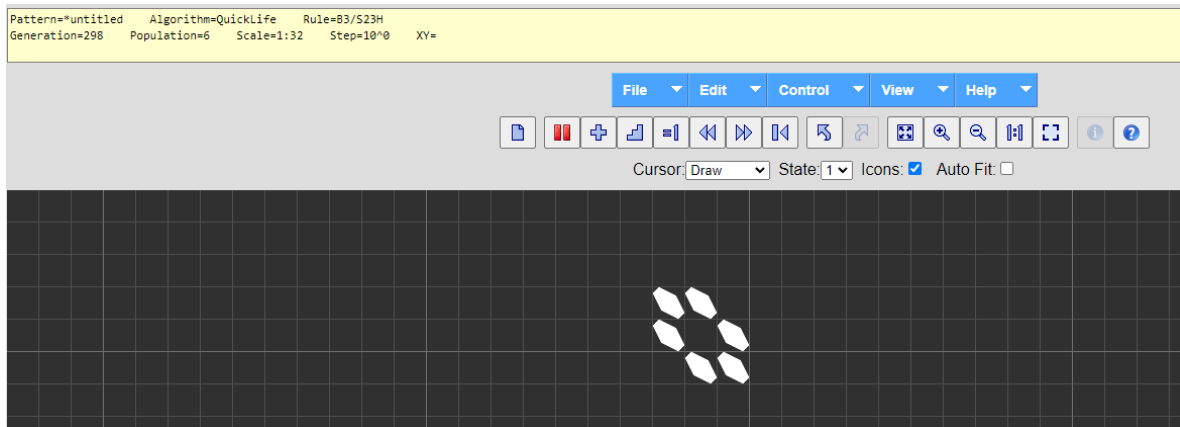
Still life



Oscillator



6. REGLA B3/s23H Numero de estudiante 30738, como tengo un 0 lo cambie por un 2.



7.

En el Circuit Mapping Problem (CMP), los componentes del circuito se representan como nodos de un grafo lógico, y las hormigas exploran posibles asignaciones de estos componentes a rutas específicas. El objetivo principal es optimizar métricas como el retardo, la longitud de las conexiones o el consumo energético.

Cada nodo o conexión tiene un valor de feromonas que refleja la calidad de las asignaciones. Inicialmente, las feromonas tienen valores iguales para garantizar exploración uniforme. Estas feromonas se actualizan en cada iteración del algoritmo.

Las decisiones de las hormigas se guían mediante una **heurística** que tiene en cuenta factores como:

- Distancia entre componentes conectados.
- Congestión en las rutas.
- Consumo energético asociado.

La probabilidad de que una hormiga elija una asignación específica se define como:

$$P(i,j) = \frac{[\tau(i,j)]^\alpha \cdot [\eta(i,j)]^\beta}{\sum_{k \in \text{vecindad}} [\tau(i,k)]^\alpha \cdot [\eta(i,k)]^\beta}$$

Donde:

- $\tau(i,j)$: Nivel de feromonas en la asignación i,j .
- $\eta(i,j)$: Valor heurístico que guía la asignación.
- α, β : Pesos que ajustan la importancia de las feromonas y la heurística.

La actualización de feromonas se realiza mediante:

$$\tau(i,j) = (1 - \rho) \cdot \tau(i,j) + \sum_{\text{hormiga } k} \Delta\tau_k(i,j)$$

Donde:

- ρ : Tasa de evaporación de las feromonas, para evitar que el algoritmo quede atrapado en soluciones subóptimas.
- $\Delta\tau_k(i,j)$: Incremento de feromonas basado en la calidad de la solución encontrada por la hormiga k .

El algoritmo termina cuando se alcanza un número predeterminado de iteraciones o cuando no hay mejora significativa en las soluciones encontradas.

8.

1. Estado actual (t).
2. Estado previo ($t-1$).
3. Estados actuales de los vecinos inmediatos.

Nueva regla:

1. Una célula estará viva (1) en la próxima generación ($t+1$) si:
 - Estaba viva (1) en $t-1$ y tiene exactamente 2 vecinos vivos en t .
 - O estaba muerta (0) en $t-1$ y tiene exactamente 1 vecino vivo en t .
2. En cualquier otro caso, estará muerta (0).

Patrón inicial:

- Generación 0 ($t=0$): 0 1 0 0 1 0 1
- Generación anterior ($t=-1$): 0 0 0 0 0 0 0

Evolución:

1. Generación $t+1$: 1 0 1 0 0 1 0
2. Generación $t+2$: 0 1 0 1 0 0 1
3. Generación $t+3$: 1 0 0 1 1 0 0
4. Generación $t+4$: 0 1 0 0 1 0 1 (regresa al estado inicial).