Manual de usuario - Interfaz gráfica ANNarchy

1. Introducción

El objetivo principal de esta interfaz gráfica es facilitar el uso del neurosimulador ANNarchy. Está dirigida especialmente a estudiantes que desean crear, simular y analizar redes neuronales sin necesidad de escribir código desde cero. A través de esta herramienta, el usuario puede construir redes neuronales de forma visual, ejecutar simulaciones con parámetros configurables y generar gráficos que ayuden a interpretar los resultados obtenidos.

2. Vista general del sistema

La interfaz gráfica está organizada en tres secciones principales: el **header**, el **lienzo** y el **sidebar**.

- **Header:**
  Ubicado en la parte superior de la pantalla, contiene el botón para iniciar la simulación, además de dos campos donde el usuario puede configurar:

    - **Tiempo de simulación:** Representa la duración total de la simulación en milisegundos. Su valor por defecto es 1000 ms.

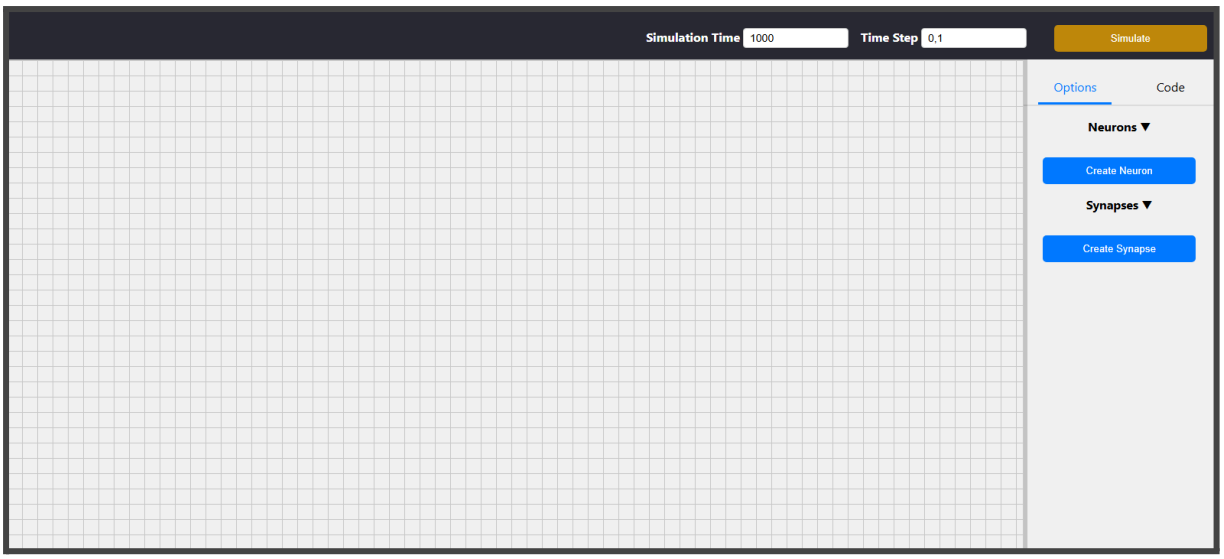
    - **Paso de tiempo:** Corresponde al valor del incremento temporal usado durante la simulación (timestep), con un valor predeterminado de 0.1 ms. Un valor más pequeño proporciona mayor precisión pero puede aumentar el tiempo de simulación (Dependiendo del modelo usado puede producir fallos).

- **Lienzo:**
  Es el área central y de mayor tamaño en la pantalla. Funciona como el espacio de trabajo donde se puede construir gráficamente una red neuronal. En este espacio se pueden disponer elementos como poblaciones neuronales, conexiones, sinapsis y monitores, mediante acciones como arrastrar, conectar y configurar visualmente cada componente.

- **Sidebar:**
  Es un menú lateral ubicado a la derecha de la pantalla, dividido en dos pestañas:

    - **Opciones:** Se muestra por defecto y presenta un listado de modelos de neuronas y sinapsis disponibles. Desde aquí el usuario puede crear modelos personalizados mediante los botones correspondientes.

    - **Code:** Permite visualizar el código ANNarchy generado automáticamente a partir de la red neuronal construida en el lienzo. También ofrece la opción de descargar este código para su uso o modificación fuera de la interfaz.



Vista general de la interfaz

Secciones de la interfaz

3. Funcionalidades principales

A continuación se describen las principales funcionalidades disponibles en la interfaz gráfica:

3.1 Construcción visual de la red neuronal

El usuario puede construir redes neuronales de forma gráfica mediante la disposición de elementos en el lienzo. Las funcionalidades principales incluyen:

- **Arrastrar y soltar**: Los modelos neuronales y sinápticos pueden arrastrarse desde el menú lateral hacia el lienzo para colocarlos, así como moverse libremente dentro del mismo.

- **Eliminación**: Para eliminar un modelo del lienzo, basta con arrastrarlo nuevamente hacia el sidebar.

- **Identificación**: Cada modelo en el lienzo se muestra con un identificador, que incluye su tipo y un número secuencial, permitiendo distinguirlos fácilmente.

- **Conexiones**: Los usuarios pueden crear conexiones entre poblaciones neuronales utilizando la herramienta de conexión. Las conexiones se visualizan con líneas:

    - **Azul** si la proyección es *excitatoria* (por defecto).

    - **Roja** si es *inhibitoria*.

    - Se indica además el tipo de conexión:

        - 1:1 para conexiones uno a uno.

        - n:n para conexiones *all-to-all*.

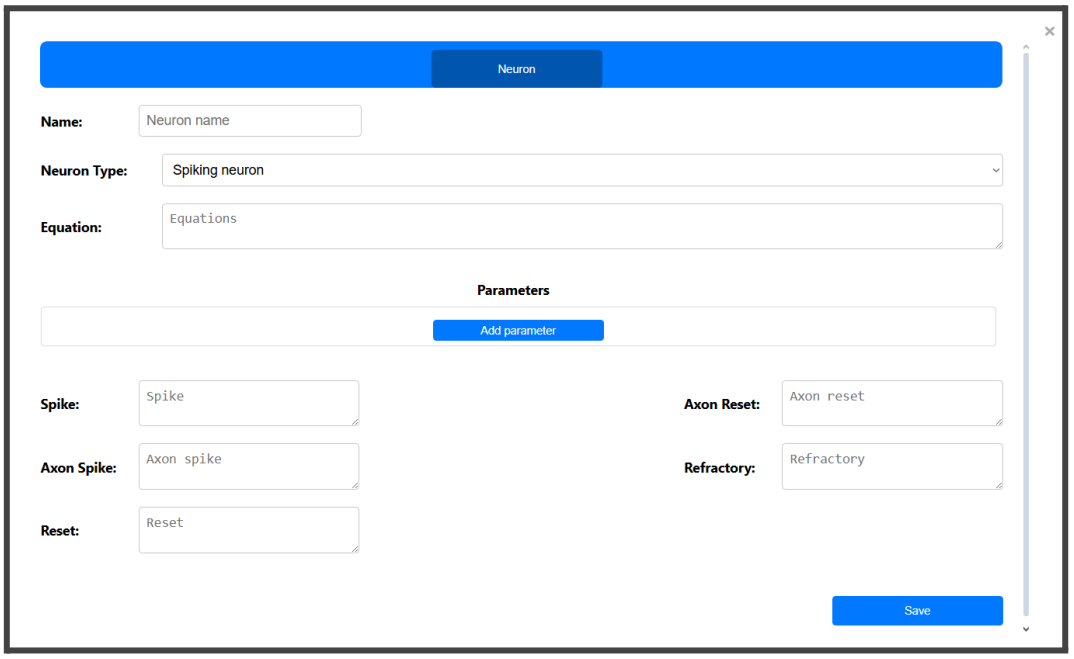
- De forma predeterminada, las poblaciones tienen tamaño 1, y las conexiones son excitatorias y de tipo all-to-all.

3.2 Creación de modelos personalizados

El usuario puede crear sus propios modelos personalizados de neuronas y sinapsis. Estos modelos se agregan automáticamente al listado del menú lateral para ser usados en la construcción de la red.

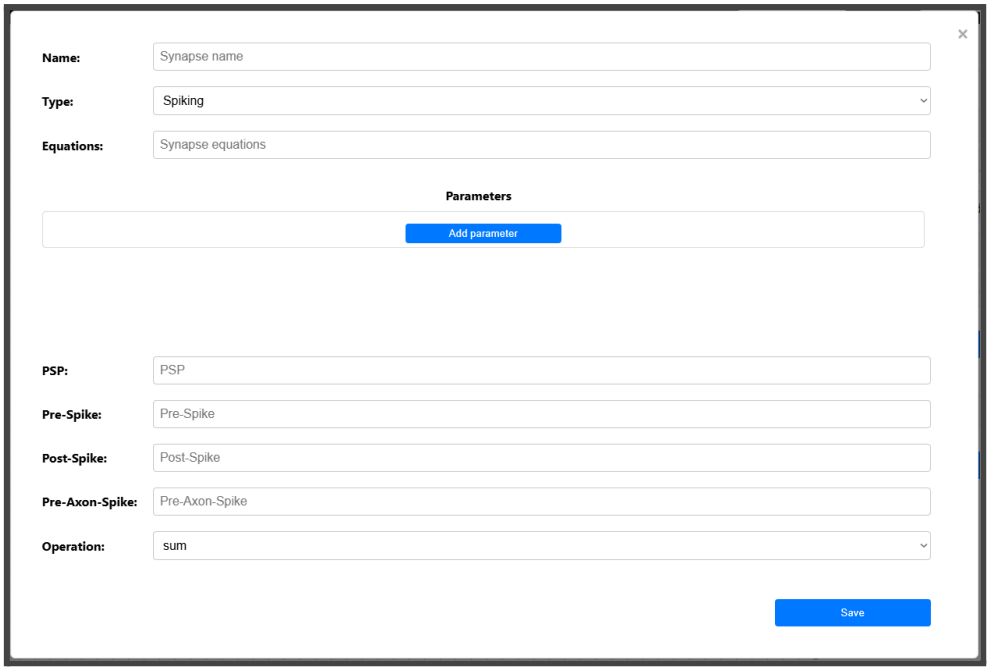
- **Modelos de neuronas**: Al crear un modelo neuronal, se despliega un panel que permite definir:

- Nombre del modelo
- Tipo de modelo (e.g. IF, HH)
- Ecuación de la neurona
- Parámetros asociados
- Eventos como: spike, reset, refractory, axon_spike, axon_reset



Vista de panel de creación de neuronas

- **Modelos de sinapsis**: Al crear un modelo sináptico, se presenta un panel para definir:
  - Nombre de la sinapsis
  - Parámetros generales
  - Eventos como: pre_spike, post_spike,PSP
  - Operación sináptica (por defecto suma)

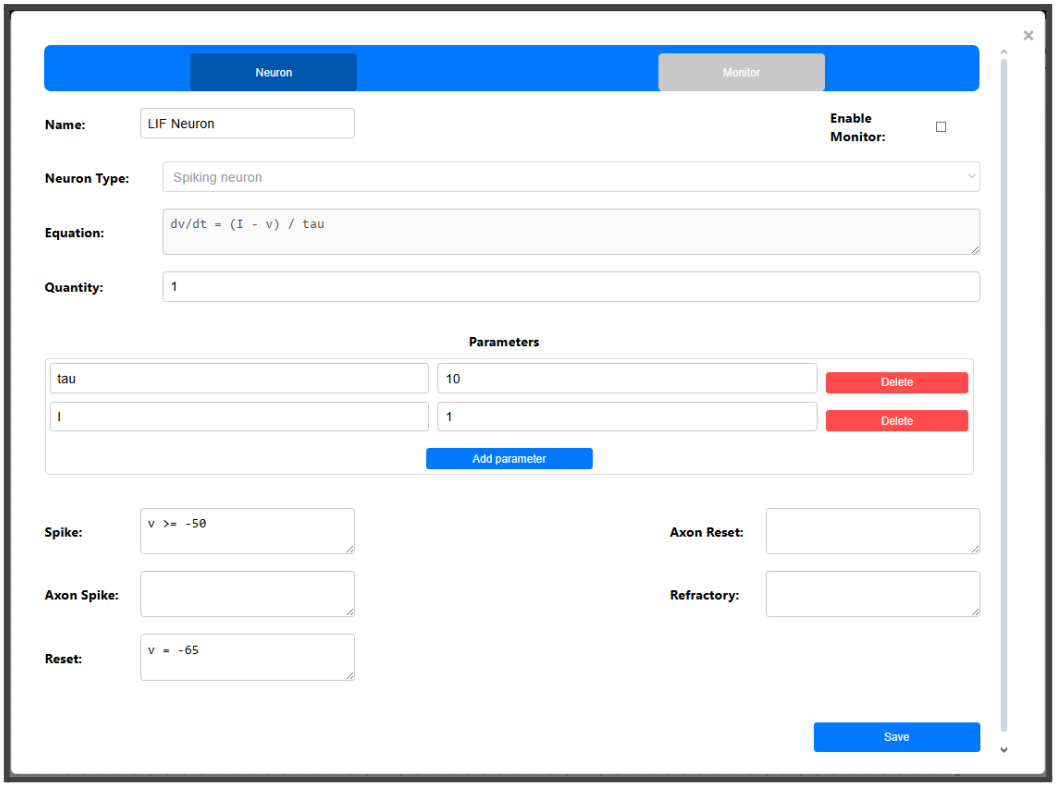


Vista de panel de creación de sinapsis

3.3 Gestión de poblaciones

Haciendo clic derecho sobre una población neuronal en el lienzo se abre un panel de gestión que permite editar configuraciones específicas de esa instancia:

- Nombre de la población
- Tipo de modelo neuronal asociado
- Ecuación correspondiente
- Tamaño de la población
- Parámetros específicos para esa población
- Eventos (spike, reset, refractory) que, aunque son los mismos definidos en el modelo, pueden ser personalizados para esa población.
- Casilla para habilitar el **monitor de actividad**:
    - Al habilitarlo, se activa una pestaña adicional donde se pueden seleccionar las variables que se desean registrar durante la simulación.

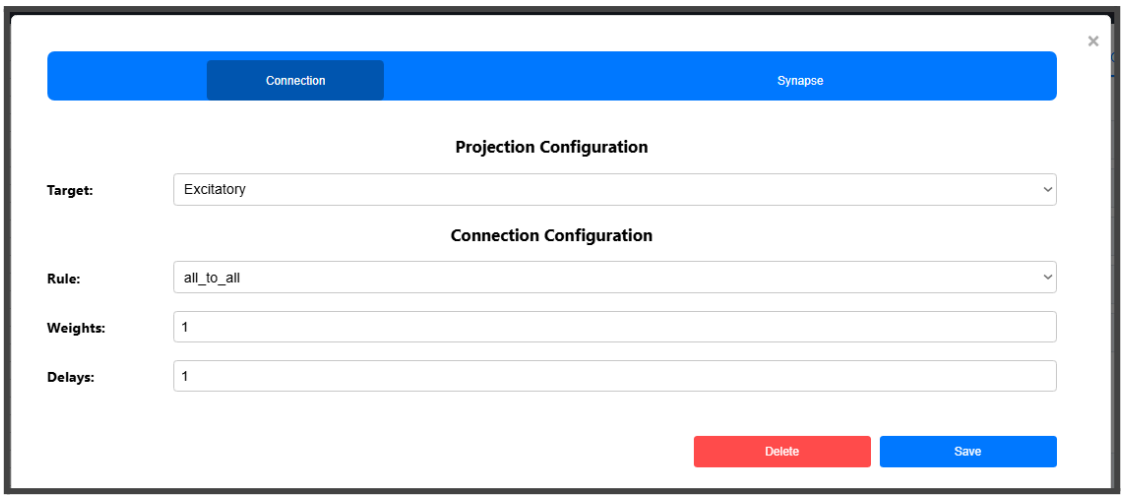


Vista de panel de gestión de población

3.4 Gestión de conexiones

Al hacer clic derecho sobre una conexión en el lienzo, se abre un panel de gestión compuesto por dos pestañas:

- **Conexión**:
    - Permite ajustar:
        - Tipo de conexión (1:1, all-to-all)
        - Peso sináptico
        - Retardo (delay)
        - Tipo de proyección (*excitatoria* o *inhibitoria*)
- **Sinapsis**:
    - Permite seleccionar y configurar el modelo de sinapsis utilizado en esa conexión.
    - Los campos a editar son los mismos que al momento de crear un modelo sináptico: parámetros, eventos (pre_spike, post_spike, PSP) y operación.



Vista de panel de gestión de conexión

3.5 Gestión de monitores

El uso de monitores en la interfaz permite registrar y visualizar la actividad de las poblaciones durante la simulación. Para activar un monitor, se debe acceder al **panel de configuración de la población** mediante clic derecho sobre ella, y luego marcar la **casilla de activación del monitor**.

Una vez activado, se habilitará una **pestaña adicional de Monitores** donde se podrán gestionar las variables a capturar durante la simulación.

- **Modelos predeterminados**: se mostrarán opciones para seleccionar directamente las variables disponibles para ese modelo.

- **Modelos personalizados**: será necesario **escribir manualmente el nombre de las variables** que se desea capturar separadas por una coma, respetando exactamente la forma en que fue declarada.

Al principio se mostrará un cuadro vacío indicando que se debe realizar la simulación para mostrar el gráfico. Después de ejecutar la simulación, se podrá volver a la pestaña de Monitores. Al hacer clic sobre una de las variables capturadas, se mostrará automáticamente un **gráfico con los datos registrados**, para modelos personalizados es necesario escribir una única variable registrada para observar.

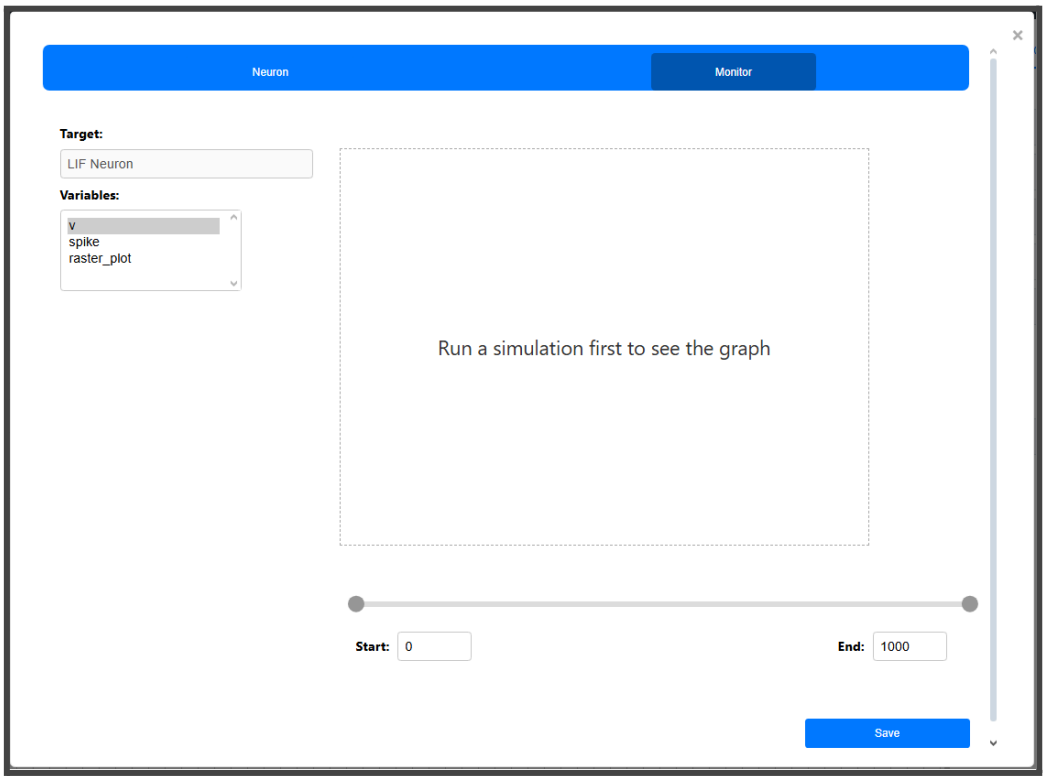
Esta sección también incluye:

- **Botones para mostrar/ocultar etiquetas** del gráfico.

- **Botón para descargar el gráfico** en formato de imagen.

- Campos para **seleccionar el grupo de neuronas** (cuando corresponda).
- Campos para **elegir el intervalo de tiempo** a visualizar.

Un control deslizante (slider) permite ajustar fácilmente el rango temporal, desde 0 hasta la duración completa de la simulación, actualizando el gráfico instantáneamente para una exploración rápida y efectiva de los resultados.



Vista de panel de población (Pestaña de monitor)

3.6 Generación de código

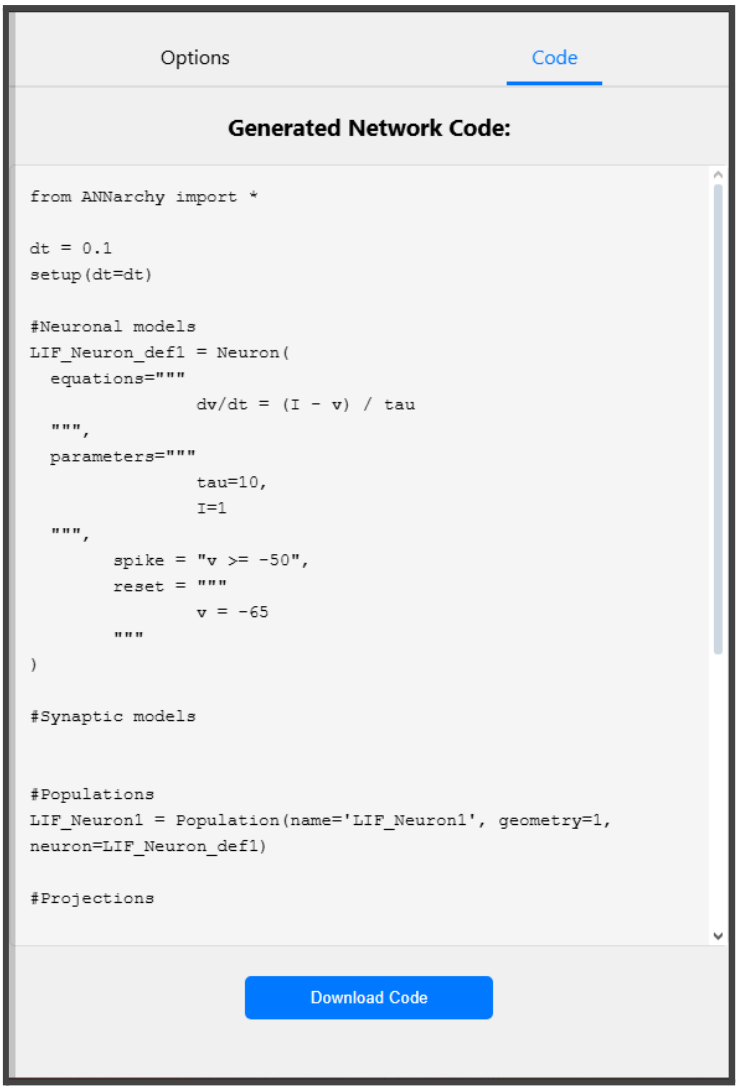
La interfaz genera automáticamente el **código ANNarchy** correspondiente a la red neuronal construida gráficamente en el lienzo. Esta generación ocurre **en tiempo real**, por lo que cualquier cambio que se realice en la red se verá reflejado inmediatamente en el código.

El código generado puede visualizarse en la pestaña **"Code"** del menú lateral. Desde allí, también se ofrece la opción de **descargar el archivo** con el código completo.

La estructura del código sigue una organización por bloques, que incluye:

- Definición de modelos de **neuronas** utilizados.
- Definición de modelos de **sinapsis**.

- Creación de **poblaciones neuronales**.
- Creación de **proyecciones** (conexiones sinápticas entre poblaciones).
- Configuración de **monitores** para registrar variables durante la simulación.



Vista de código de red

3.7 Ejecución de la simulación

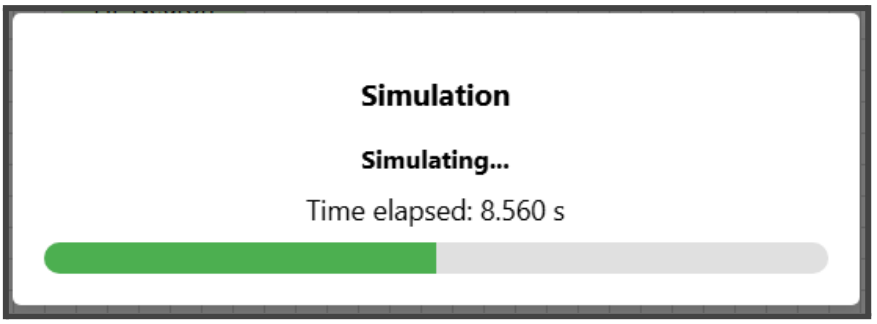
Para ejecutar la simulación, se debe presionar el botón **"Simulate"**, el cual ejecutará la red por la cantidad de tiempo definida en el campo **simulation time**, utilizando el **paso temporal (time step)** especificado **en el header**.

La duración de la simulación puede variar entre **segundos y varios minutos**, dependiendo de:

- La **complejidad de la red** neuronal.
- Los **parámetros utilizados**.
- La **carga del servidor** al momento de la ejecución.

Al iniciar la simulación, se mostrará un **mensaje de estado**, que puede indicar si la simulación:

- Se encuentra **en cola**.
- Está **siendo ejecutada**.
- Está **procesando resultados**.



Vista de mensaje de estado de simulación

Una vez finalizada, los **monitores activados** habrán registrado información de las variables seleccionadas, la cual se podrá visualizar desde la pestaña de Monitores.

En caso de que ocurra **algún error durante la simulación**, se mostrará el **mensaje de error entregado por el simulador** dentro del mismo cuadro de estado, para facilitar su análisis y corrección.

Ejemplo de error

4. Posibles errores durante la simulación

Durante la ejecución de simulaciones, pueden producirse errores que impidan que el sistema funcione correctamente. Estos errores se deben principalmente a problemas en la definición de modelos personalizados, en los parámetros asignados a las poblaciones o en la configuración de los monitores. A continuación, se describen algunos errores comunes y sus causas más frecuentes:

Error:
ANNarchy.core.Global.ANNarchyException: The value of the parameter is not a float.
Causa:
Alguna población tiene asignado un valor inválido como parámetro (por ejemplo, texto o un campo vacío en lugar de un número).
Solución:
Verificar que todos los parámetros definidos en las poblaciones sean valores numéricos válidos (con punto decimal si es necesario).

Error:
ANNarchy.core.Global.ANNarchyException: Compilation failed.
Causa:
Existe un error de sintaxis en algún campo del modelo, como por ejemplo escribir una asignación (=) donde debería ir una expresión booleana (por ejemplo, en condiciones de spike o reset).
Solución:
Revisar cuidadosamente las ecuaciones y condiciones ingresadas en los modelos personalizados. Usar siempre expresiones válidas según la sintaxis de ANNarchy.

Error:
ANNarchy.core.Global.ANNarchyException: Monitor: the object does not have an attribute named x
Causa:
Se intentó monitorizar una variable que no existe en la población seleccionada.
Solución:
Verificar que el nombre de la variable ingresado en el monitor coincida exactamente con una variable declarada en el modelo asociado (respetando mayúsculas y minúsculas).

Error:
ANNarchy.core.Global.ANNarchyException: Parameters are not recordable
Causa:
Se intentó registrar un parámetro que no puede ser monitorizado (como un valor fijo).
Solución:
Solo se pueden monitorizar variables dinámicas (definidas en el bloque variables). Evitar usar parámetros como objetivo de los monitores.

En general, los **códigos de error de ANNarchy que se muestran al finalizar una simulación son claros y fáciles de comprender**. Estos mensajes pueden copiarse directamente para buscar información adicional o ayuda en foros y recursos en línea, facilitando así la resolución de problemas.

5. Ejemplo de uso (Neurona Rate-coded)

A continuación, se describe un flujo típico de trabajo utilizando la interfaz gráfica, con base en un ejemplo de la documentación de ANNarchy, para este se usará un modelo de neurona y de sinapsis tipo Rate-coded.

5.1 Definición de modelos personalizados

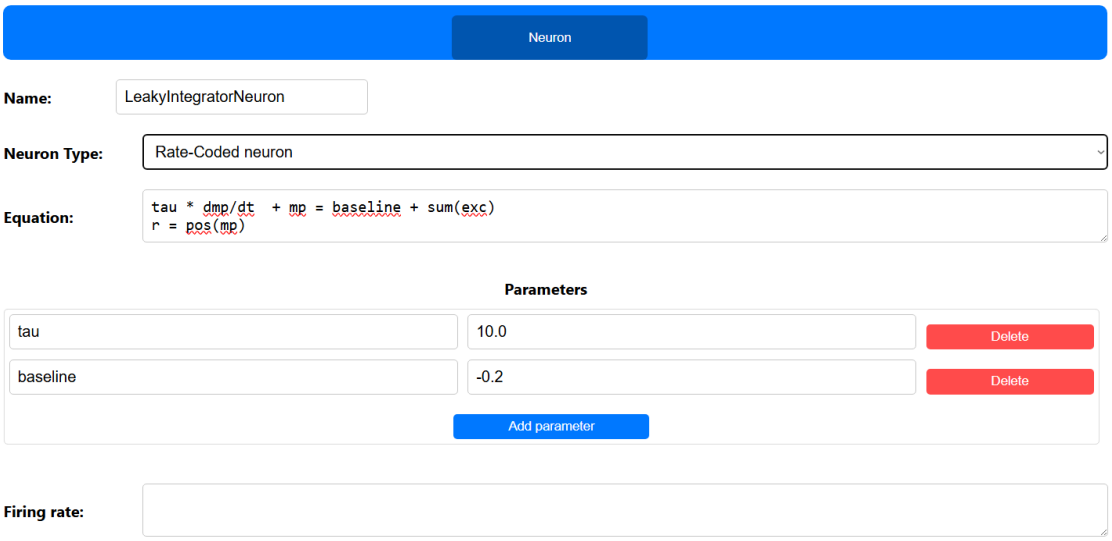
Antes de construir la red, es necesario definir los modelos personalizados de neurona y sinapsis.

Modelo neuronal: Leaky Integrator Neuron

Tipo de neurona	**Rate-coded**
Ecuaciones	**tau * dmp/dt + mp = baseline + sum(exc)** **r = pos(mp)**
Parámetros	**tau = 10.0** **baseline = -0.2**

Donde:

- tau: constante de tiempo de integración.
- mp: potencial de membrana.
- baseline: corriente de entrada base.
- sum(exc): suma de entradas excitatorias.
- r: tasa de activación, que se obtiene con la función pos() para limitarla a valores positivos.



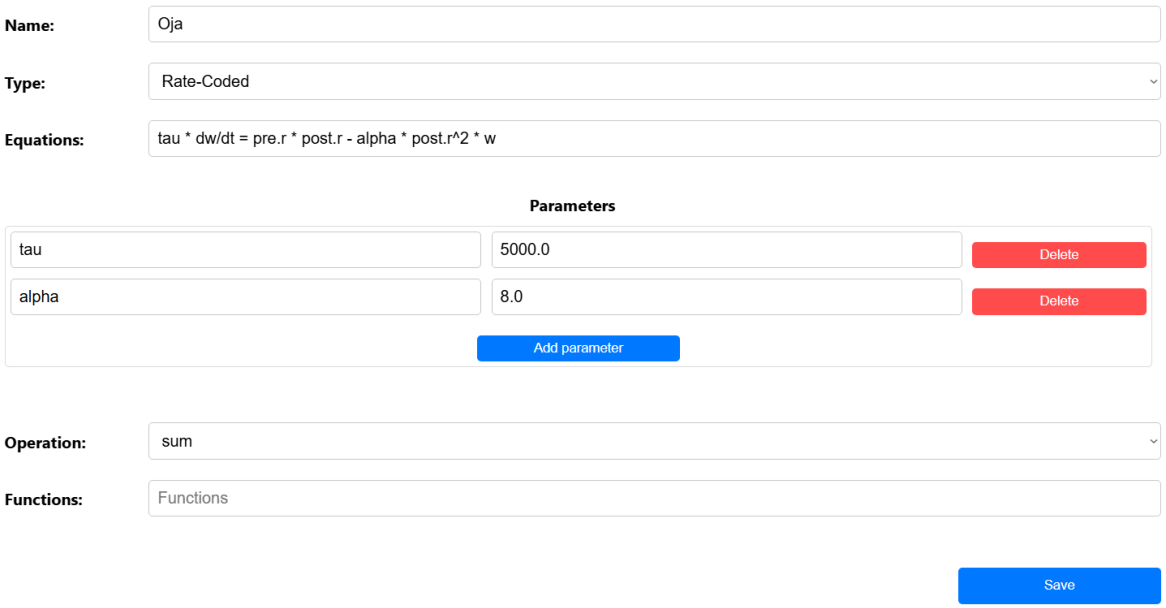
Definición de neurona

Modelo de sinapsis: Oja

Tipo de sinapsis	**Rate-coded**
Ecuaciones	**tau * dw/dt = pre.r * post.r - alpha * post.r^2 * w**
Parámetros	**tau = 5000.0** **alpha = 8.0**
Operación	**Suma**

Donde:

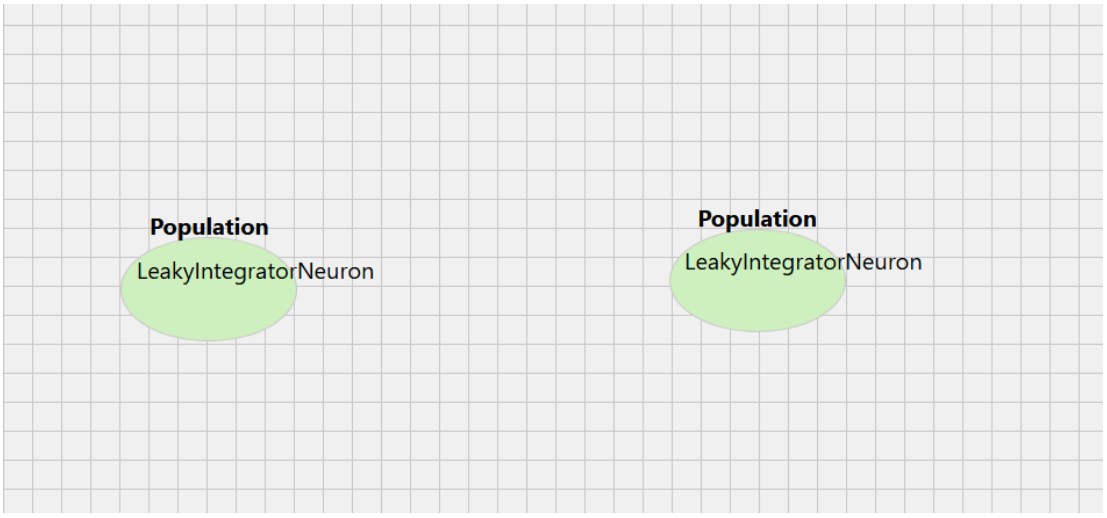
- pre.r: actividad de la neurona presináptica.
- post.r: actividad de la neurona postsináptica.
- w: peso sináptico.
- tau: constante de tiempo para la modificación del peso.
- alpha: factor de decaimiento de aprendizaje.



Definición de sinapsis

5.2 Construcción de la red

Desde el menú lateral, se arrastran dos instancias del modelo LeakyIntegratorNeuron al lienzo.

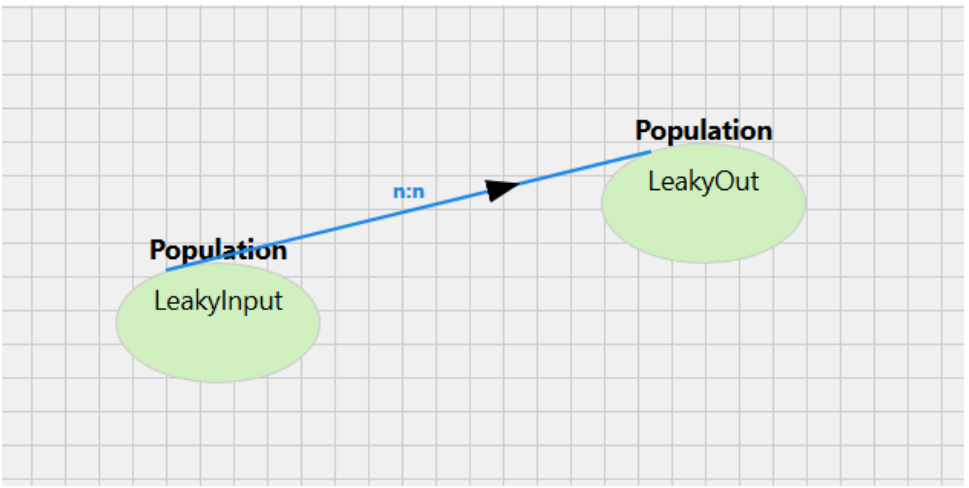


Neuronas en lienzo

Haciendo clic derecho sobre cada neurona, se ajusta su tamaño a **100**. Esto indica que cada población contiene 100 neuronas individuales. (También se puede cambiar el nombre de la población).

Ajustes población de entrada

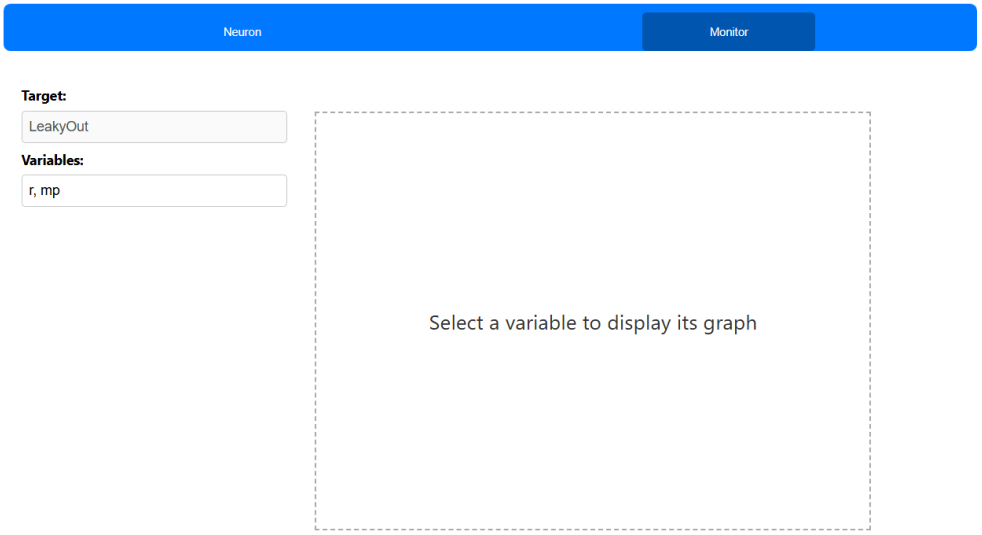
Se selecciona el botón de sinapsis del menú lateral y se conecta una neurona con la otra utilizando el modelo **Oja**, se deja en la configuración predeterminada (excitatoria, all to all, peso 1 y delay 1).



Conexión de neuronas Leaky Integrator

Se activa el monitor para la población que recibe la conexión sináptica.

- Se selecciona una o ambas variables para observar, como r (tasa de activación) o mp (potencial de membrana).

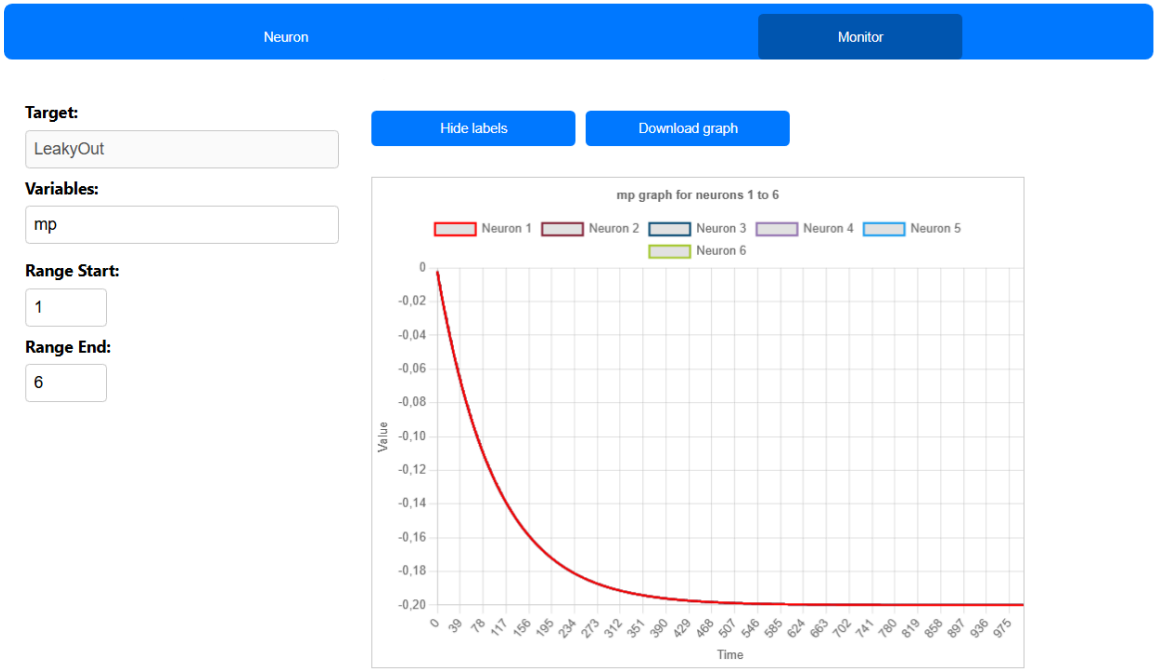


Selección de variables en monitor

5.3 Ejecutar la simulación

Se hace clic en el botón de simulación para ejecutar el modelo con los parámetros actuales. Luego se accede a la pestaña Monitores para visualizar el gráfico generado.

- Al oprimir el botón de mostrar etiquetas y poner el rango desde la neurona 1 a la 6 se ve una sola recta con las 6 etiquetas, esto se debe a que todas las neuronas de la población tienen el mismo comportamiento.



Primeros resultados de simulación

5.4 Edición de parámetros y nueva simulación

Ahora para cambiar el comportamiento de la red se cambiarán los parámetros y ajustes de los elementos de la red.

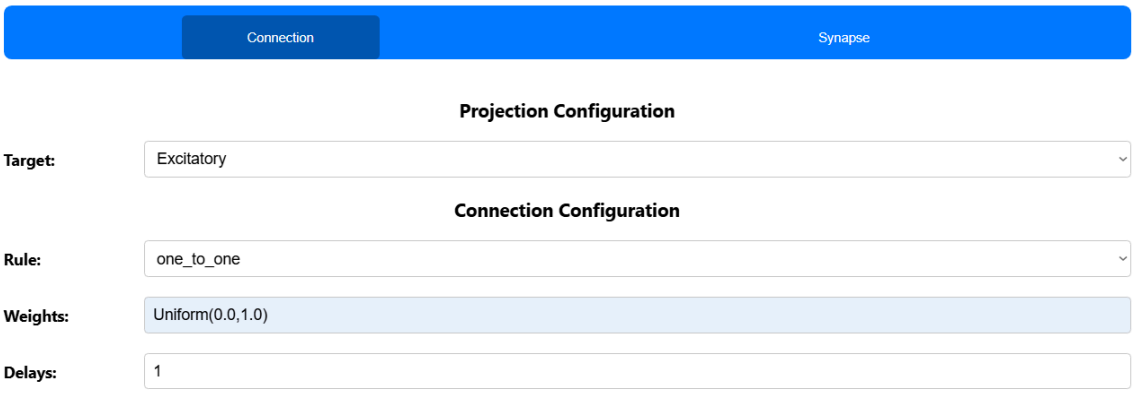
- Se edita el valor de baseline en la primera población (emisora) a un número positivo, por ejemplo: 0.7. Esto activa la población para que dispare espontáneamente.

- Se edita la sinapsis para:

    - Usar un valor de peso inicial aleatorio: Uniform(0.0, 1.0)

    - Establecer la regla de conexión en forma uno a uno (1 a 1), lo que significa que la neurona i en la primera población se conecta con la neurona i en la segunda.

Parámetros modificados de neurona de entrada



Parámetros modificados de conexión

Se vuelve a simular con los nuevos parámetros.

- Se escoge la misma variable que la vez anterior.
- Se cambia el rango de neuronas a visualizar, por ejemplo, desde 1 a 6.
- Se activan las etiquetas para ver el identificador de cada curva en el gráfico.

Resultados con los cambios de parámetros

5.5 Conclusión del ejemplo

El ejemplo práctico presentado demuestra cómo las configuraciones iniciales influyen directamente en el comportamiento dinámico de la red neuronal. En la primera simulación, al asignar valores fijos y homogéneos, todas las neuronas respondieron de forma idéntica, generando un gráfico plano con una tendencia estable hacia un mismo valor (aproximadamente -0.25).

Tras modificar parámetros clave (como establecer un baseline positivo en la neurona y aplicar una distribución Uniform(0.0, 1.0) para los pesos sinápticos) la segunda simulación mostró respuestas diferenciadas entre neuronas. Esta vez, el gráfico refleja un rango más variado y complejo de comportamientos, evidenciando que la red ahora presentaba una dinámica más compleja.

Este resultado pone de relieve la facilidad para experimentar y ajustar parámetros dentro del entorno, así como la capacidad de observar los resultados de forma ágil y dinámica, lo que permite explorar distintos comportamientos neuronales de manera intuitiva y eficaz.

6. Ejemplo de uso (Neurona Spikes)

Este ejemplo muestra cómo utilizar neuronas spiking basadas en el modelo Izhikevich, ya integrado en la interfaz, para construir una red simple, simular su comportamiento y observar los resultados mediante monitores.

6.1 Definición del modelo

El modelo **Izhikevich** implementa las siguientes ecuaciones:

$$I = g_{exc} - g_{inh} + noise \times Normal(0.0, 1.0)$$

$$\frac{dv}{dt} = 0.04v^2 + 5.0v + 140.0 - u + I$$

$$\frac{du}{dt} = a(bv - u)$$

Con:

- **Disparo (spike)** cuando v ≥ v thresh.
- **Reset**: v = c, u = u + d

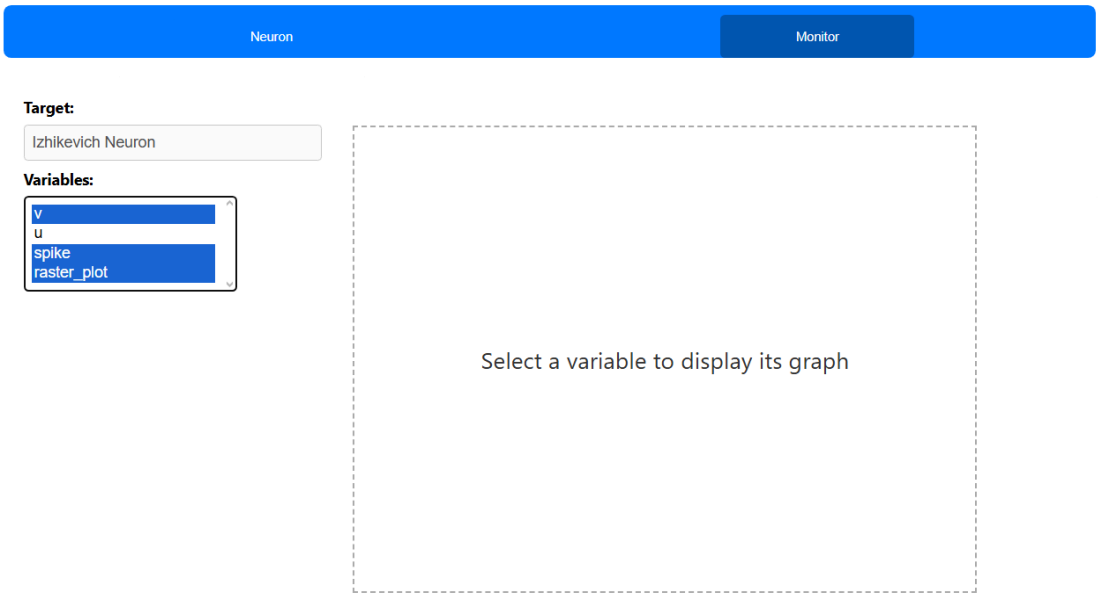
Parámetros predeterminados:

- a=0.02
- b=0.2
- d=2
- v thresh=30
- noise = 0

Este modelo viene implementado de manera predeterminada en la interfaz, con lo que no es necesario definirlo como personalizado.

6.2 Construcción de la red

Desde el menú lateral, se arrastra una sola instancia de la neurona predefinida Izhikevich al lienzo y se le asigna una población de 100. Después se le asigna un monitor a la población y se seleccionan variables las variables **V (Voltaje), Spikes(Tasa de disparo) y Raster plot.**



Monitor con variables seleccionadas

6.3 Ejecutar la simulación

Se ejecuta la simulación haciendo clic en el botón “Simulate” con los parámetros actuales. Luego se accede al monitor para ver los datos monitorizados, los cuales deberían asemejarse a las siguientes figuras:

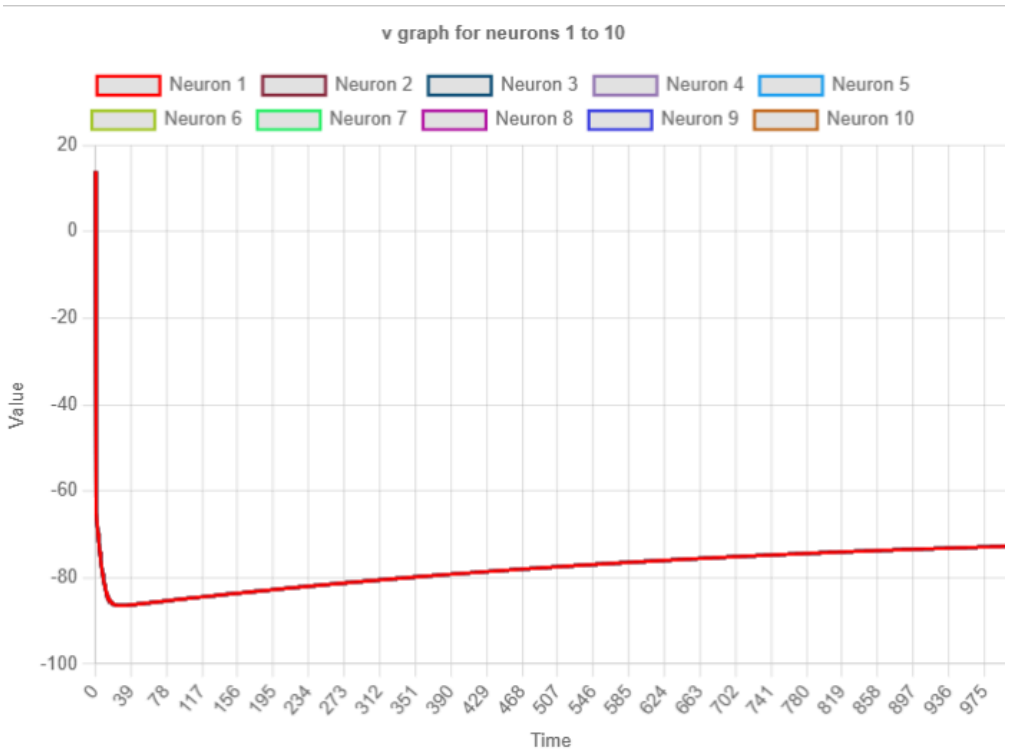


Gráfico de V para población Izhikevich (Configuración predeterminada)

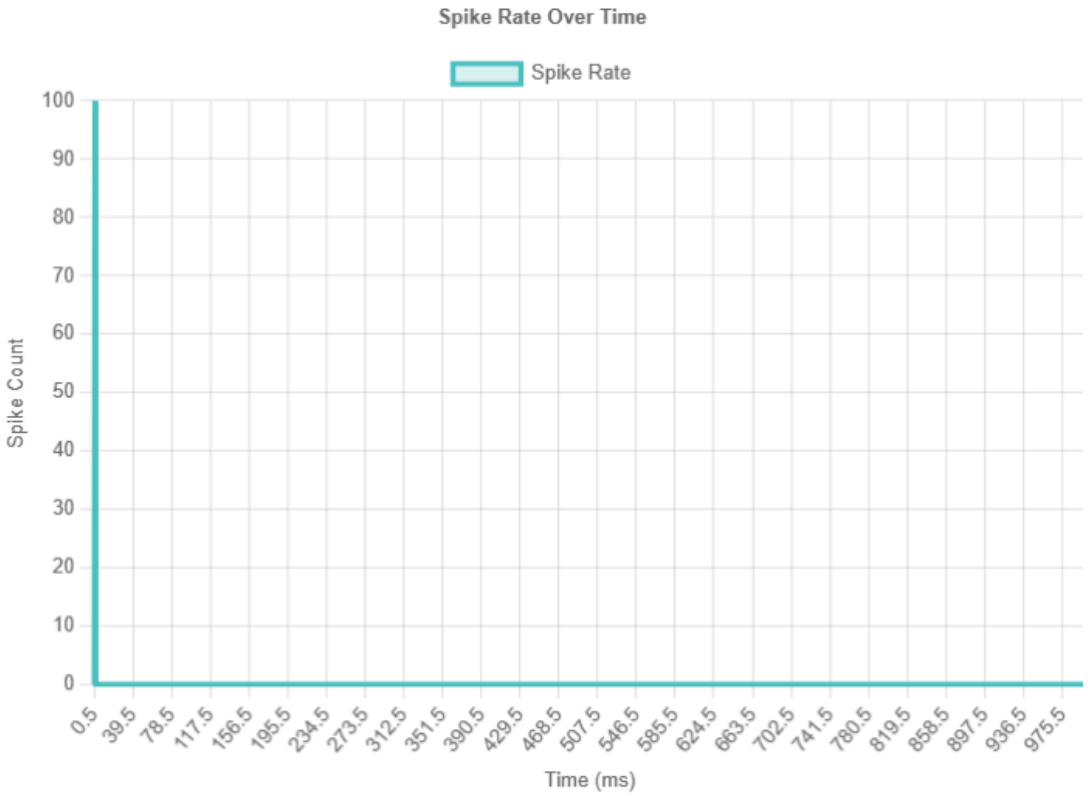


Gráfico de Spikes para la población de Izhikevich (Configuración predeterminada)

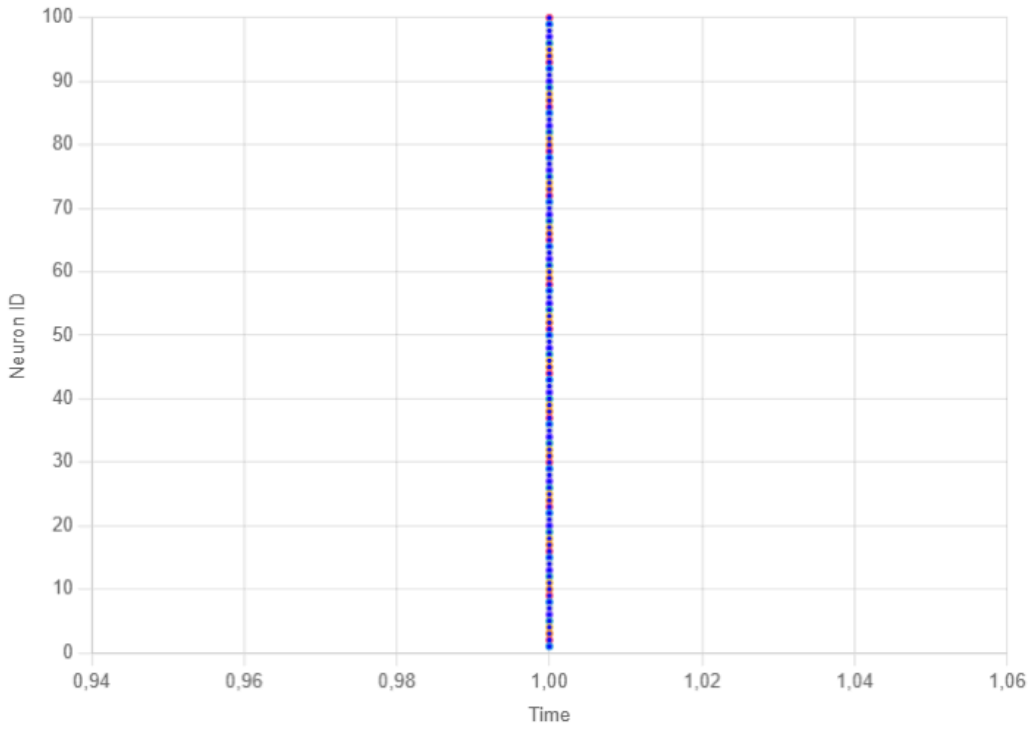


Gráfico Raster plot para la población de Izhikevich (Configuración predeterminada)

En los gráficos se puede apreciar el comportamiento en reposo de las neuronas de Izhikevich, el estado inicial de las neuronas provoca que en los primeros instantes de la simulación todas estas disparen y luego no hay actividad (No ocurren spikes) aunque V lentamente aumenta.

6.4 Ajustes en la red neuronal

Ahora para aumentar la actividad de la neurona se añade una neurona de Poisson y se conecta a la población de Izhikevich para generar estimulación

Se crea la población de poisson usando la neurona predefinida y se le pone un tamaño de 100, este modelo tiene el parámetro “rate” del cual depende su tasa de disparo, a mayor “rate”, mayor tasa de disparo, con lo que se le pondrá un valor de 1000. Se creará una conexión que la una a la neurona de Izhikevich con una configuración all to all con peso Uniform(1.0,5.0) para darle aleatoriedad y mayor actividad a la neurona objetivo.
Con los cambios hechos se vuelve a ejecutar la simulación para registrar los mismos datos.

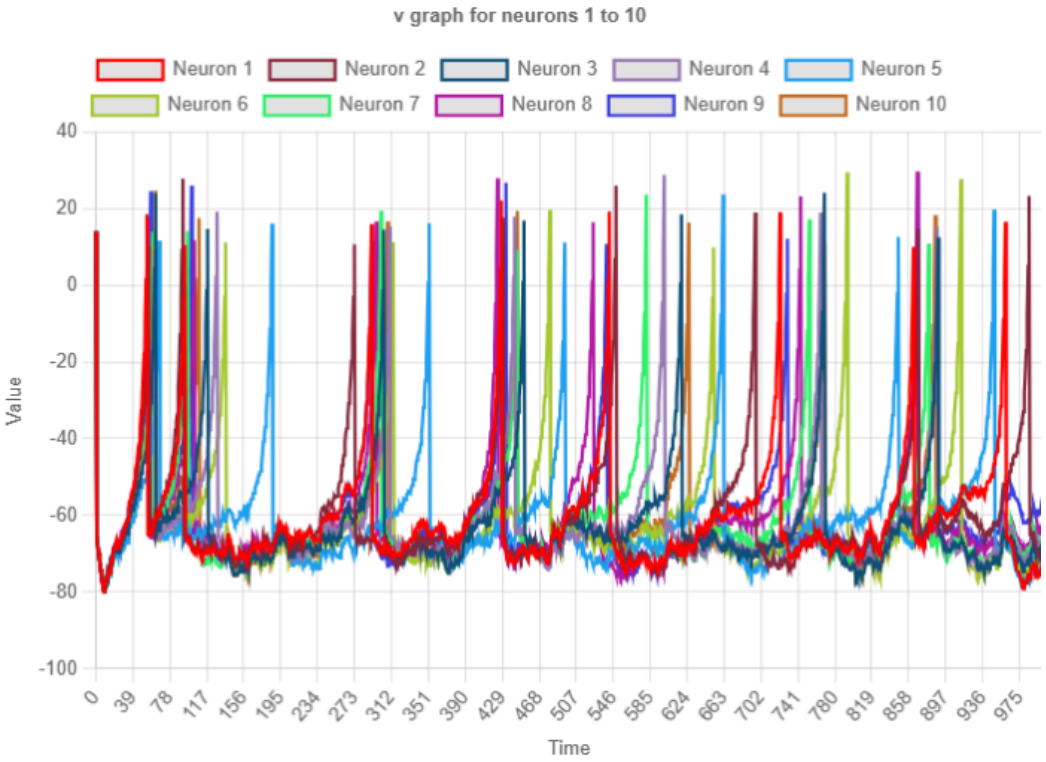


Gráfico de V para población Izhikevich (Configuración estimulada)

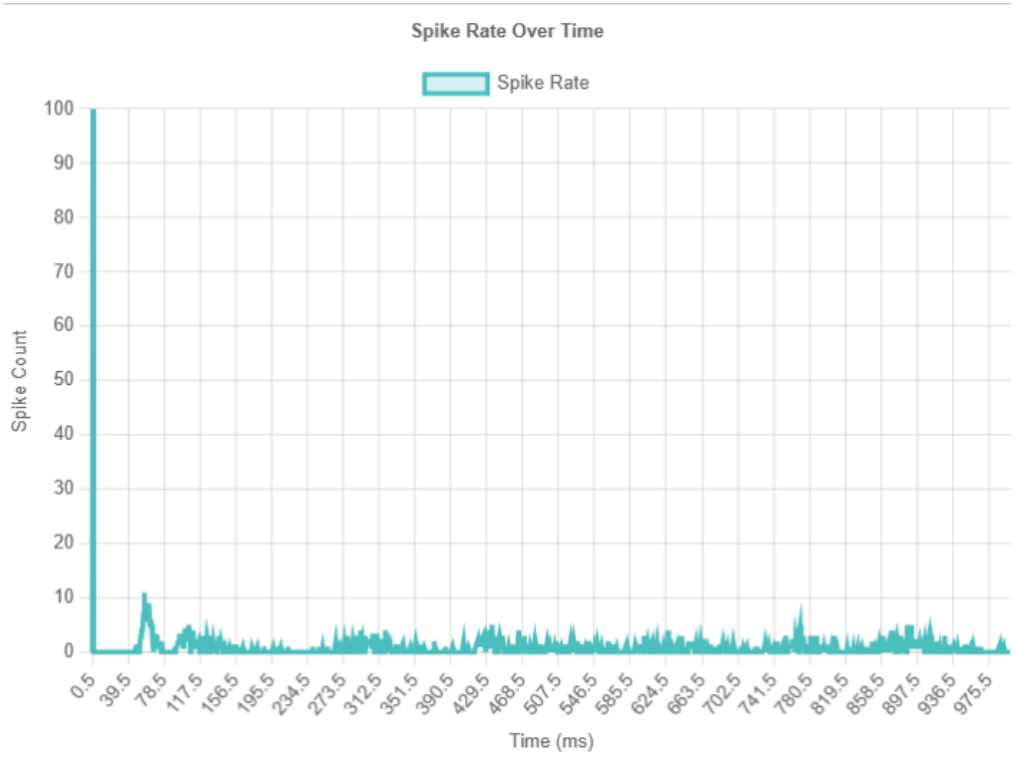


Gráfico de Spikes para la población de Izhikevich (Configuración estimulada)

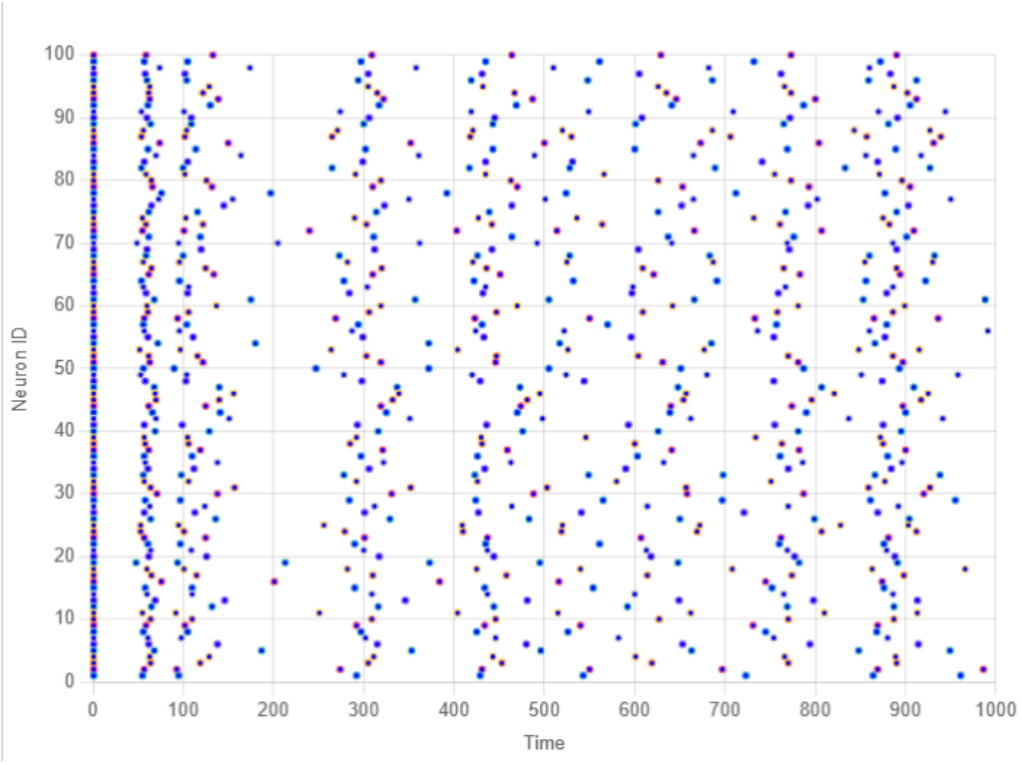


Gráfico Raster plot para la población de Izhikevich (Configuración estimulada)

Luego de haber simulado con los nuevos parámetros no solo se hace visible que aumenta mucho la actividad respecto el caso anterior, sino que los gráficos presentan patrones más ordenados y regulares. Este flujo de trabajo permite crear redes funcionales, de manera progresiva y visualizar cambios en de forma sencilla y dinámica.