

Guía práctica de laboratorio  
Comportamiento *closed-loop* con Bonsai

<b>Bonsai</b>	<b>2</b>
<b>Instalación de Bonsai</b>	<b>2</b>
<b>Arduino</b>	<b>3</b>
<b>Instalación del Arduino IDE</b>	<b>4</b>
<b>Comunicación la placa Arduino desde un programa externo</b>	<b>4</b>
<b>Uso de Bonsai</b>	<b>5</b>
Ej 1 Salida digital a una placa Arduino	6
<b>Uso del protoboard</b>	<b>6</b>
<b>Uso de Arduino desde Bonsai</b>	<b>7</b>
Ej 2 Salida digital con un circuito LED externo	7
Extra: entrada digital	8
<b>Construir el riel de alimentación del protoboard a partir de la fuente de 5V de la placa Arduino</b>	<b>9</b>
Ej 3 Salida digital PWM con un servomotor	10
Ej 4 Señales analógicas	11
<b>Uso de Bonsai para seguimiento</b>	<b>12</b>
Ej 5 Grabar un video	12
Ej 6 Conversión a escala de grises	12
Ej 7 Segmentación de colores	13
Ej 8 Seguimiento	14
Ej 9 Sustracción de fondo y segmentación del movimiento	15
Ej 10 Medición del movimiento	16
<b>Uso de Bonsai para closed-loop con Arduino</b>	<b>17</b>
Ej 11 Emisión de una señal digital a partir de la actividad en una región de interés	17
Ej 12 Emisión de una señal digital a partir de un umbral en una señal analógica	18
A partir de la actividad del ratón de la computadora	18
A partir de una entrada de audio	18

# Guía I

## Bonsai

En esta guía nos familiarizaremos con el lenguaje de programación visual [Bonsai](#) (Lopes *et al.* 2015). Este lenguaje permite combinar el funcionamiento de varios dispositivos en paralelo y diseñar soluciones experimentales personalizadas integradas y flexibles para el estudio del comportamiento. Las posibilidades incluyen el trackeo de los sujetos experimentales, la automatización de tareas comportamentales con múltiples sensores y actuadores, la integración con registros de electrofisiología e incluso la posibilidad de realizar experimentos de tipo bucle cerrado, más conocidos como *closed loop*.

Bonsai tiene la ventaja de ser una plataforma relativamente simple de usar, con requerimientos mínimos de conocimientos de programación para la adquisición de datos. Se caracteriza también, como otras herramientas abiertas, por su [comunidad](#) activa de usuarios dispuestos a entender y armar sus propias herramientas de trabajo, así como también a ayudar a otros miembros a lograr sus objetivos experimentales.

Utilizaremos Bonsai en conjunto con una placa microcontroladora Arduino, la cual nos va servir para emitir y registrar señales digitales (LED, servomotor, pulsos TTL) o analógicas (sonido, sensores) mediante puertos de entrada y de salida.

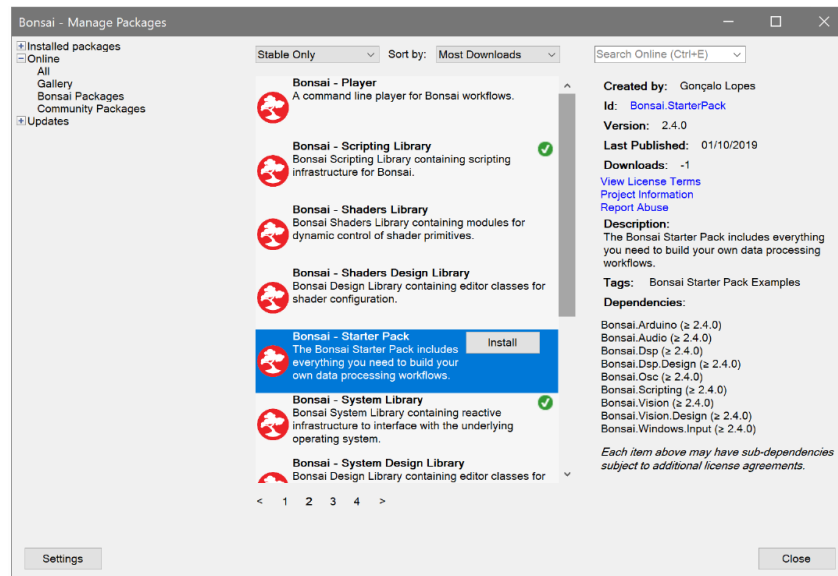
Para todo esto, debemos instalar Bonsai y el Arduino IDE (entorno de desarrollo integrado, por las siglas en inglés de integrated development environment), y lograr que se comuniquen entre sí. Empezemos:

## Instalación de Bonsai

1. Descargá **Bonsai** de <http://bonsai-rx.org> y ejecutalo.
2. Instalá **Bonsai - Starter Pack** desde el administrador de paquetes (*Tools > Manage Packages*).

## Guía práctica de laboratorio

### Comportamiento *closed-loop* con Bonsai



3. Hacé clic en la pestaña de **Updates** en el extremo izquierdo de la pantalla e instalá cualquier actualización que esté disponible.

## Arduino






Haremos uso de la placa microcontroladora Arduino desde Bonsai, para hacer uso de sus entradas/salidas analógicas y digitales. Para ello, necesitamos instalar el Arduino IDE y establecer la comunicación con Bonsai. También es posible cargar programas (*sketch*) en la placa con el Arduino IDE (entorno de desarrollo integrado), pero nos limitaremos a usarla desde Bonsai.

La placa ARDUINO UNO está conformada por un microcontrolador ATMEGA que posee entradas y salidas tanto analógicas como digitales que nos van a permitir interactuar con el mundo exterior a partir de las instrucciones que le enviemos. Para ello, necesitaremos conectarla a la computadora con su cable USB. Deberás referirte al esquema *pinout* para conocer la disposición de pines y las conexiones básicas de Arduino UNO:



## Uso de Bonsai

En este [link](#) podrás acceder a contenidos teóricos del lenguaje de programación visual Bonsai. Para un desarrollo más fluido de esta guía durante el curso, a continuación te comentamos algunos de los conceptos claves.

Categoría del operador	Descripción
 Source (fuente)	Genera flujos de datos desde dispositivos o desde archivos
 Transform (Transformar)	Convierte o procesa elementos individuales de datos
 Condition (condición)	Filtra elementos de datos que coincidan con alguna condición específica
 Sink (sumidero)	Guarda datos o activa salidas externas
 Combinator (combinador)	Administrar el flujo de información o sincroniza entradas paralelas


Sugerencias y comandos útiles:

- Los operadores pueden encontrarse en la ventana **Toolbox**. Doble clic en el operador de interés y aparecerá en el espacio **Workflow**.
- Para conectar nodos, clic izquierdo sobre uno > **Create Connection**, y clic sobre el nodo que querés unir al anterior.
- Al hacer clic izquierdo sobre un nodo, se despliega un menú de opciones, entre las cuales están habilitar (Ctrl D)/deshabilitar (Ctrl+Shift+D) dicho nodo y externalizar alguna de sus salidas o propiedades.
- En un *workflow* que esté corriendo, al hacer doble clic sobre un nodo, se abre una ventana con su salida.

Te recomendamos la introducción sobre [cómo usar la interfaz de usuario](#) en y la [lista de los comandos más usados](#).

## Ej 1 Salida digital a una placa Arduino

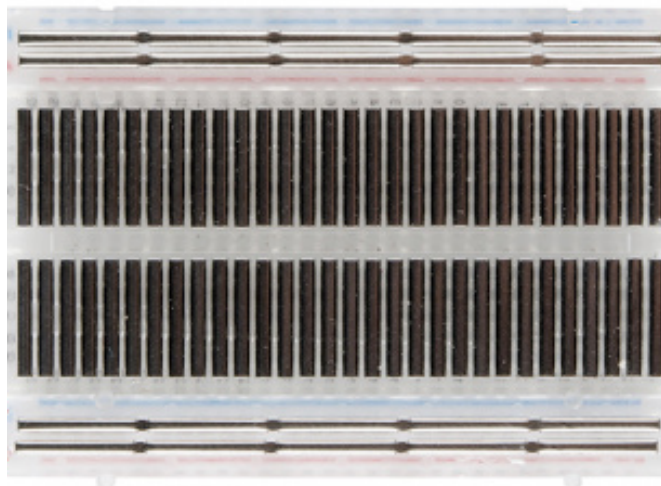


1. Insertá un nodo **Boolean (o Buleano)**. Este nodo puede tomar los valores TRUE o FALSE.
2. Insertá un nodo **DigitalOutput**.
3. Asegurate de tener conectada la placa Arduino a la computadora y configurá la propiedad **Pin** del nodo **DigitalOutput** al pin que corresponda para encender el LED que viene integrado en la placa (lo vas a poder encontrar en el esquema *pinout* que se encuentra más arriba con el dibujo de un diodo  ).
4. Configurá la propiedad **PortName** del nodo **DigitalOutput** para que apunte a donde se encuentra la placa Arduino.
5. Corré el *workflow* y explorá las propiedades de los nodos para lograr encender y apagar el LED.

## Uso del protoboard

El *protoboard* nos va a permitir interconectar los diferentes componentes electrónicos y las entradas/salidas del Arduino mediante cables, sin la necesidad de soldarlos.

Consiste de líneas conductoras (tiras metálicas) dispuestas según se ve en la foto a continuación. Se puede acceder a cada línea mediante agujeros o “pines” en los cuales encastran las patas de los componentes electrónicos. Todas los componentes que se colocan en la misma línea están conectados por igual.



*Las entrañas de un protoboard, enseñando sus conexiones.*

Los pines adyacentes a la línea roja están conectados entre sí. Lo mismo sucede con los pines adyacentes a la línea azul. Estos pines suelen utilizarse como “rieles” para las líneas de alimentación, ya que son accesibles a lo largo de todo el protoboard.

## Uso de Arduino desde Bonsai

### Ej 2 Salida digital con un circuito LED externo

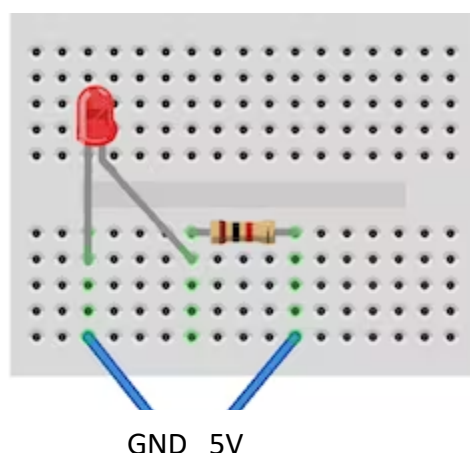
#### ¡Advertencia 1!

*Debés asegurarte de que los extremos de los cables que uses para la salida de tensión de 5V y GND no se toquen entre sí. Si estuvieran en cortocircuito, fluiría una corriente muy grande, la suficiente como para freír nuestro Arduino.*

#### ¡Advertencia 2!

*Tené cuidado al conectar a los pines del Arduino. Para no equivocarte de pin, mirá la placa desde arriba para ver bien las etiquetas, o contá los pines.*

1. Conectá los componentes para hacer un circuito LED externo.

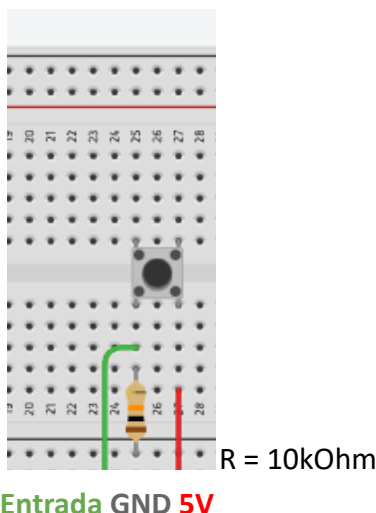


El circuito para conectar un LED necesita una fuente de alimentación (usaremos 5V de la placa Arduino), una resistencia para limitar la corriente que circula por el LED, y una tierra (GND). Usaremos una resistencia de 220 Ohms. El cátodo del LED (borde recto en la base del encapsulado plástico del LED) debe ir conectado a la tierra.

2. Controlá el encendido y apagado de la LED con el *workflow* de Bonsai que armaste en el Ej 1. ¿Cuál va a ser tu fuente de alimentación ahora?

### Extra: entrada digital



1. Un pulsador en un circuito que haga pasar 5V solo cuando está presionado nos va a poder hacer las veces de señal digital en un puerto de entrada. Nuevamente, podemos tomar esos 5V del Arduino. El circuito para conectar un pulsador es el siguiente:



*Notá que cada pulsador tiene 4 patas pero utilizamos solo 2 para un circuito.  
Usamos un puerto digital como entrada al Arduino.*

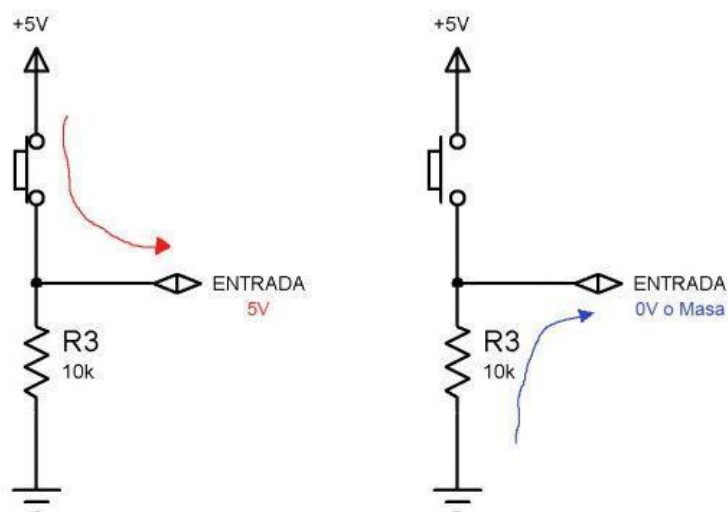
2. Si queremos que se encienda el LED cuando presionamos el pulsador (porque pasa la señal de 5V al Arduino) ¿qué nodo de Bonsai necesitás? ¿Cómo lo debés integrar con el nodo **DigitalOutput** que usaste en el Ej 2?

Veamos cómo funciona el circuito del pulsador:

El pulsador  está conectado a un puerto de entrada de la placa Arduino (cable verde) y al riel de 5V (cable rojo). Al presionarlo, la corriente fluye desde la alimentación hacia el puerto de entrada, y el puerto lee una señal HIGH (5V) (parte izquierda del diagrama de circuito a continuación). Al dejar de presionarlo, el circuito está abierto, ¿es suficiente para leer LOW en la entrada?. Para que esté apagado, queremos asegurarnos que el puerto de entrada lea 0V, entonces debemos proveerle un camino a tierra (GND)  (parte derecha del diagrama de circuito a continuación).



Guía práctica de laboratorio  
Comportamiento *closed-loop* con Bonsai



¿Y por qué ponemos una resistencia entre el pulsador y GND? Porque si no estuviera esa resistencia, no funcionaría el circuito cuando el pulsador está presionado, ¿podés explicar por qué?

Esta configuración se llama *pull-down*, porque el circuito “adicional” se ocupa de llevar a 0V la entrada cuando el circuito está abierto.

## Construir el riel de alimentación del protoboard a partir de la fuente de 5V de la placa Arduino

Para alimentar circuitos externos más cómodamente, solemos conectar un riel del protoboard al voltaje necesario para los componentes electrónicos que vamos a usar y otro riel a tierra.

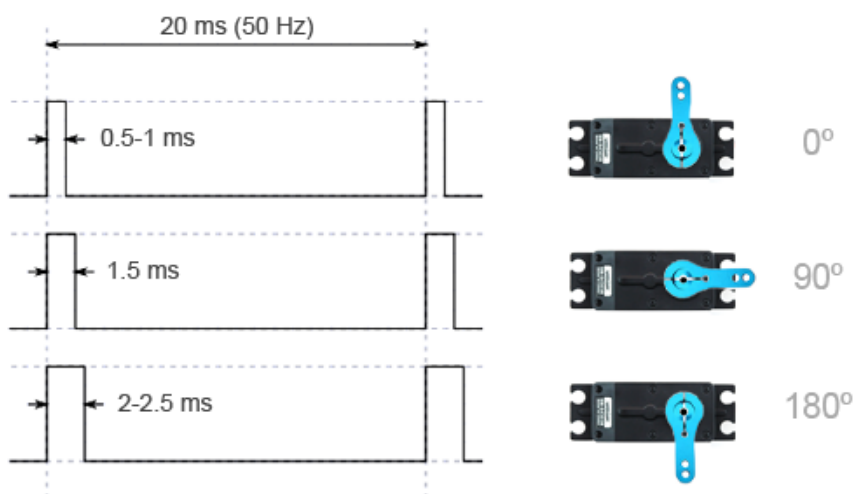
Para lo que sigue, asegurate de tener armados los rieles:

1. Conectá la salida de voltaje de 5V del Arduino mediante un cable macho-macho (rojo preferiblemente) a alguno de los orificios adyacentes a la línea roja del *protoboard*.
2. Conectá el pin de GND del Arduino mediante otro cable macho-macho (negro preferiblemente) a alguno de los orificios adyacentes a la línea azul del *protoboard*.

### Ej 3 Salida digital PWM con un servomotor

Un servomotor, o servo, es un motor al que le indicamos el ángulo en el cual debe posicionarse. Típicamente disponen de un rango de movimiento de entre 0 a 180° (no son capaces de dar una vuelta por completo).

La información del ángulo de giro deseado se realiza mediante la transmisión de una señal pulsada (PWM, Pulse Width Modulation) con un período de 20 ms. El ancho del pulso HIGH va a determinar la posición del servo, como se ve en el esquema a continuación. Esta señal se la entregaremos a través de un puerto digital PWM de la placa Arduino, utilizando el nodo **ServoOutput** de Bonsai.



#### ¡Advertencia!

*En el caso de utilizar servomotores, motores paso a paso, o cualquier otro dispositivo que pueda necesitar un consumo considerable de corriente, es muy importante utilizar una fuente de alimentación externa. Caso contrario podríamos exigirle demasiado corriente a nuestro Arduino, destruyéndolo e incluso dañando en el puerto USB de la computadora. En este ejercicio, al tratarse de un servo pequeño y no ejercer fuerza sobre el mismo (consumo mínimo de corriente) y por cuestiones didácticas, la corriente que nos brinda la placa Arduino es suficiente.*

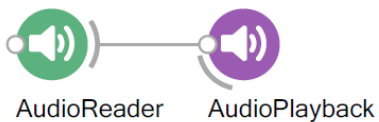
La conexión del servo es la siguiente: el cable rojo va en la línea de alimentación 5V, el marrón a GND y el amarillo, a un pin digital PWM de la placa Arduino que utilizaremos como salida.

1. ¿Puedes lograr mover el servo a un ángulo determinado usando Bonsai? Pista: “Int” quiere decir “número entero”.
2. ¿Qué necesitás agregar para alternar entre dos ángulos y lograr generar un movimiento oscilatorio? Tratá de imaginarlo y vemos cómo implementarlo en conjunto.

## Ej 4 Señales analógicas



1. Insertá un nodo **AudioCapture**
2. Insertá un nodo **AudioWriter** y conectalo al **AudioCapture**. Configurá la propiedad de **FileName** del **AudioWriter**. Acordate de agregar al nombre la extensión .WAV.
3. Revisá que las frecuencias de registro de ambos nodos sean iguales
4. Corré el *workflow* por unos segundos y chequeá que el archivo de audio se haya guardado correctamente.



5. Insertá un nodo fuente **AudioReader**.
6. Configurá la propiedad **FileName** para que sea el archivo de audio que acabás de guardar.
7. Insertá un nodo sumidero **AudioPlayback**.
8. Corré el workflow, escuchá el sonido y mirá la señal en el visor del nodo de AudioPlayback.

## Guía II

En esta parte de la guía, nos enfocaremos en las funcionalidades de Bonsai para realizar análisis y seguimiento de un objeto (“tracking”) en tiempo real con la cámara web. Luego, integraremos estas capacidades con lo aprendido anteriormente sobre control entradas y salidas de Arduino.

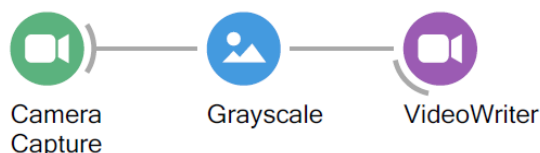
### Uso de Bonsai para seguimiento

#### Ej 5 Grabar un video



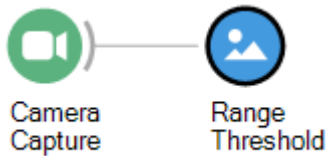
1. Insertá el nodo **CameraCapture**. Necesitas tener alguna cámara conectada a tu computadora. Si es una laptop podés usar la cámara integrada. Configurá el index según corresponda al de la cámara que querés usar.
2. Insertá el nodo **VideoWriter**.
3. Configurá la propiedad **FileName** del nodo **VideoWriter** con un nombre de archivo terminado en **.avi**. Podés apretar los tres puntos al costado de **FileName** para setear la dirección donde querés guardar el video.
4. Corré el *workflow* y controlá que se genere un vídeo válido.

#### Ej 6 Conversión a escala de grises



1. Insertá un transformador **Grayscale** entre los nodos **CameraCapture** y **VideoWriter**.
2. Corré el *workflow*. ¿Cómo se ve el video guardado?
3. Modificá el *workflow* para que ahora puedas guardar en simultáneo un video en color y en escala de grises.

## Ej 7 Segmentación de colores

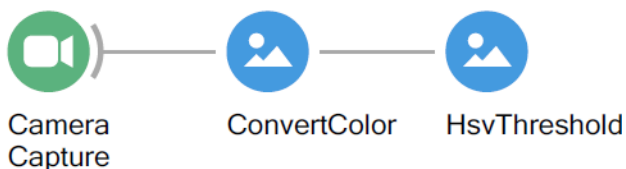


1. Inserta el nodo **CameraCapture**.
2. Inserta el nodo **RangeThreshold**.
3. Abre el visualizador del operador **RangeThreshold** haciendo doble clic sobre ese nodo.
4. Configura las propiedades **Lower** y **Upper** del **RangeThreshold** para detectar solo tu objeto de color.

*Pista 1:* al hacer clic en el símbolo + que se encuentra a la izquierda de cada propiedad, se expanden sus valores individuales.

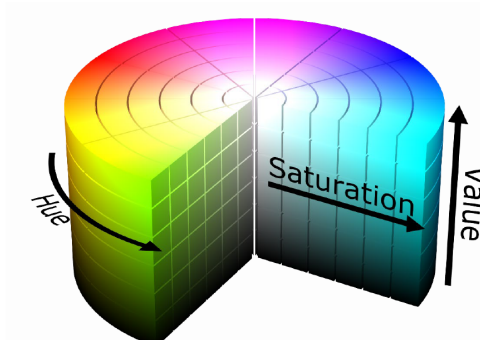
*Pista 2:* haciendo doble clic sobre cualquier nodo podrás visualizar la información que sale de ese nodo. Te recomendamos, al correr cada *workflow*, que hagas doble clic sobre los nodos, al menos sobre los más relevantes, para una inspección del proceso. Además, si haces clic izquierdo sobre la ventana de visualización de los nodos de imagen, en el margen inferior de la ventana verás no solo la posición xy del cursor sino también los valores de color en esa posición.

Este método segmenta los objetos de color estableciendo límites directamente en el espacio de color RGB. Se considera que este espacio de color es pobre para la segmentación de un color ¿Te das cuenta por qué?



Guía práctica de laboratorio  
Comportamiento *closed-loop* con Bonsai

1. Reemplazá el operador **RangeThreshold** por el **ConvertColor**. Este transformador convierte tu imagen desde el espacio de color RGB al espacio de color *Hue-Saturation-Value* (HSV, Tono - Saturación - Valor).



2. Insertá un nodo **HsvThreshold**.
3. Configurá las propiedades **Lower** y **Upper** de **HSVThreshold** para detectar solo a tu objeto de color.
4. Corroborá el seguimiento de tu objeto de color bajo distintas condiciones lumínicas.

## Ej 8 Seguimiento

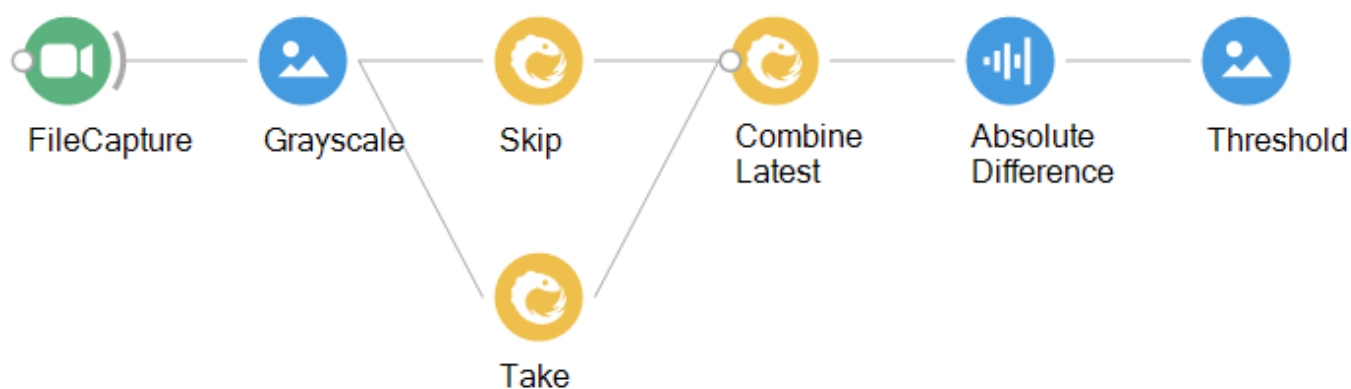


1. Usando el workflow del ejercicio anterior, insertá un nodo **FindContours**. Este operador traza los contornos de todos los objetos en una imagen blanco y negro. Definimos como objeto a una región de la imagen conectada por píxeles blancos.
2. Insertá un transformador **BinaryRegionAnalysis**. Este nodo permite calcular área, posición del centro de masa, orientación de todos los contornos detectados en la imagen.
3. Insertá un nodo **LargestBinaryRegion**. Este operador extrae el objeto más grande de los presentes en el nodo anterior. Haciendo clic izquierdo, seleccioná como *output* al *centroid*, que indica la posición del centro de masa del objeto (*Connected Component > Centroid*). Si arrastras el nodo **Centroid** hacia la ventana de visualización del nodo **CameraCapture** o **ConvertColor**, podrás ver el centroide marcado con un punto rojo, acompañando el movimiento del objeto. Si haces clic sobre el video, se marcará la trayectoria del centroide.
4. Grabá en un .csv la posición del centro de masa usando un nodo **CsvWriter** y abrílo (si podés, también graficalo) para verificar si tiene sentido.

**Nota:** El seguimiento de objetos no necesariamente debe ser con videos adquiridos *online*, sino que puede hacerse un análisis *offline* usando videos ya grabados y el operador **FileCapture**. Esto muchas veces sirve para hacer *troubleshooting* de un workflow que toma videos sin tener que estar en vivo con la cámara.

## Ej 9 Sustracción de fondo y segmentación del movimiento

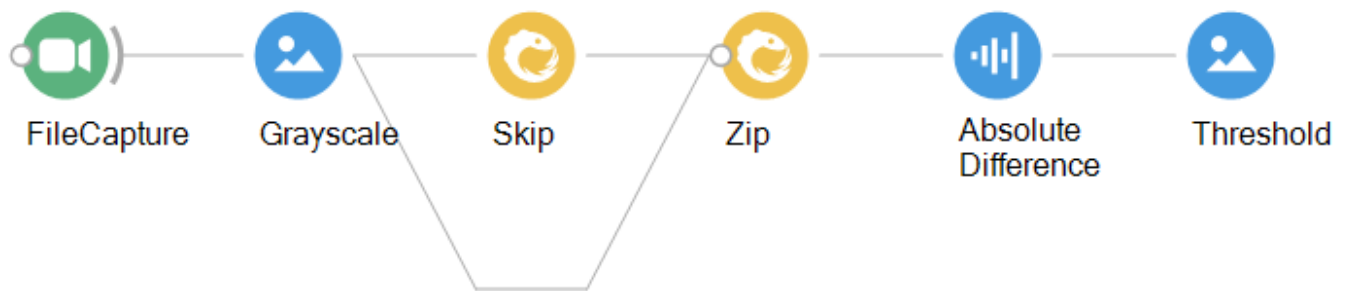
Antes de comenzar a probar distintos métodos para obtener la sustracción de fondo, necesitás tener un video con el cual probar los workflows. Basándote en el ej 5, grabá un video de un minuto de duración en el cual, en los primeros segundos, haya un fondo estático y, luego, aparezca un objeto en movimiento sobre ese mismo fondo.



1. Utilizá el nodo **FileCapture** para cargar el video que grabaste.
2. Pasá el video a escala de grises como hiciste en el ejercicio 6.
3. Insertá un operador **Skip**. Configurá su propiedad **Count** en 1.
4. En una nueva rama insertá un operador **Take**. Configurá su propiedad **Count** en 1.  
¿Qué hacen estos nodos? Buscá las descripciones en el recuadro de propiedades de la parte derecha de la pantalla.
5. Combiná las imágenes de las dos ramas usando el nodo combinador **CombineLatest**.
6. Insertá el nodo **AbsoluteDifference** después de **CombineLatest**.
7. Insertá un nodo transformador **Threshold (Vision)**. Visualiza la salida de ese nodo y ajusta la propiedad **ThresholdValue**.
8. Probá el resultado del workflow variando la propiedad Count en **Take** y **Skip**.

Describí con tus palabras qué está pasando.

Guía práctica de laboratorio  
Comportamiento *closed-loop* con Bonsai

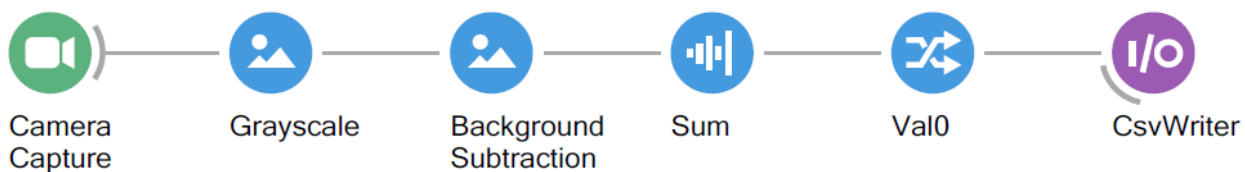


9. Reemplazá el operador **CombineLatest** por el nodo **Zip**.
10. Eliminá el operador **Take**.

Describí con tus palabras qué está pasando ahora. ¿Qué pensás que pasaría si aumentás la propiedad Count de **Skip**?

11. Aumentá la propiedad Count de **Skip** para comprobarlo. ¿Qué valor utilizarías para esta propiedad?

## Ej 10 Medición del movimiento



1. Generá un video en escala de grises como hiciste en el ejercicio 6.
2. Insertá el nodo **BackgroundSubtraction**. Configurá su propiedad **AdaptationRate** en 1. Ajustá la propiedad **ThresholdValue** para quitar el ruido de fondo.
3. Corré el *workflow*, ¿cual de los dos métodos de sustracción que viste en el Ej 9 está utilizando el nodo **BackgroundSubtraction**?
4. Cambiá el valor de **AdaptationRate** a 0 y fijate qué resultado obtenés ahora.

¿A qué conclusión llegás?

5. Insertá el operador **Sum**. Este operador va a sumar el valor de todos los píxeles de la imagen.
6. Hacé clic derecho sobre el operador **Sum**. Seleccioná *Scalar* > **Val0** desde el menú.

Nota: El operador **Sum** suma los valores de los píxeles en todos los canales de color de la imagen. Sin embargo, en el caso de las imágenes binarias en escala de grises, sólo hay un canal activo y su suma se almacena en el campo **Val0**.



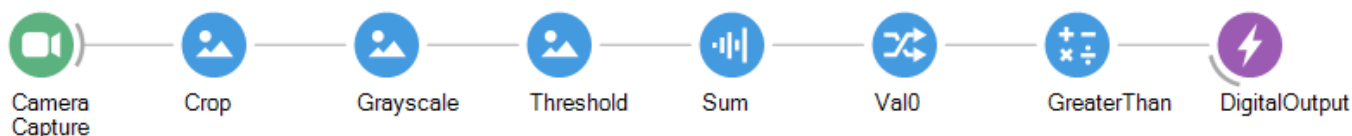
7. Corré el *workflow*, apunta tu cámara a un objeto en movimiento y mirá la salida del operador **Sum**. Compará movimientos pequeños con movimientos más marcados. ¿Qué pasa con la señal cuando el objeto se queda perfectamente quieto?
8. Clic derecho sobre el operador **Sum**. Seleccioná *Scalar > Val0* desde el menú.
9. Registrá el movimiento de un objeto usando el nodo **CsvWriter**.

## Uso de Bonsai para *closed-loop* con Arduino

En un experimento de tipo “circuito cerrado”, más conocido como *closed loop*, el objetivo es generar en tiempo real una acción como respuesta -o feedback- a un determinado evento. De esta manera, la salida de nuestro sistema depende de la detección de un evento. En el caso de experimentos comportamentales, este evento podría ser la ubicación de un animal en cierto lugar de la arena experimental, o el momento en que una señal de un sensor supera un umbral, entre muchos otros.

Usando Bonsai, es posible diseñar paradigmas que incluyen el procesamiento en tiempo real de los datos de las entradas (sensores, cámaras, micrófonos, electrodos, etc.) de manera tal de actuar sobre el animal o la arena experimental (estimulación cerebral, presentación de un estímulo, recompensa, cambio en la arena, etc).

### Ej 11 Emisión de una señal digital a partir de la actividad en una región de interés



1. Insertá un nodo **CameraCapture**
2. Insertá un nodo transformador **Crop**, que te va a permitir cortar una parte del espacio de la imagen que sea de tu interés.
3. Para establecer esa región de interés, que en este caso debe englobar el LED, corré el *workflow*. Recordá que haciendo doble clic sobre cada nodo podrás “ver” la información que sale de ese nodo. Como verás, inicialmente la imagen del **Crop** es igual a la imagen del **CameraCapture**. Andá a las propiedades del Crop, clickeá el símbolo “...” de **RegionOfInterest**. Aparecerá una muestra del espacio de la imagen el la que podrás recuadrar tu región de interés-

Nota: si estás usando la cámara integrada de una laptop podés pegar con cinta el protoboard a una caja de cartón que la sostenga vertical, de manera que puedas detectar el LED desde tu imagen.

Pista: La calibración visual de la región de interés también puede realizarse haciendo clic derecho en el nodo crop, seleccionando **ShowDefaultEditor**

4. Insertá un nodo **Grayscale** y un **Threshold(Vision)**
5. Insertá un operador **Sum(Dsp)** y seleccioná **Val0**
6. Insertá un nodo **GreaterThan** y configurá la propiedad **Value**

7. Insertá un nodo **DigitalOutput**. Configuralo al pin 13 del Arduino y al puerto adecuado.
8. Corré el *workflow* y verificá que “entrando” a la región de interés genera la emisión de una señal en el Arduino (luz en el LED integrado al Pin 13).

Extra: reemplazá el transformador Crop por uno **CropPolygon** (experimentalmente no siempre vamos a querer regiones de interés rectangulares). Este nodo también te permite seleccionar múltiples regiones.

## Ej 12 Emisión de una señal digital a partir de un umbral en una señal analógica

### a) A partir de la actividad del ratón de la computadora

En Bonsai también puedes capturar los periféricos de la computadora, como el ratón y el teclado, para incorporar como señales de entrada en un *workflow*.

1. Insertá el nodo **MouseMove (Window.Input)** y externalizá el output X o Y, con clic izquierdo sobre el nodo.
2. Usá nodos **GreaterThan**, **LessThan** u operadores equivalentes para conectar uno de los ejes del mouse con la salida.
3. Conectalo al nodo **ServoOutput**. ¿Este nodo recibe valores booleanos?

### b) A partir de una entrada de audio

1. Insertá un nodo de input analógico (por ejemplo **AudioCapture** o **AnalogInput** de Arduino)
2. Usá los nodos **Buffer(Dsp)** y **Average** para poder capturar el valor promedio de más de una muestra. **Average** devuelve 4 valores por defecto, pero los nodos subsiguientes toman 1 solo. Para obtener una sola salida del nodo **Average**, hacé clic derecho > *Output* > *Val0*.  
¿Qué función cumple **Average** en este *workflow*?
3. Usá los nodos que conocés para detectar cuándo el audio supera cierta amplitud.  
¿Te parece que es posible establecer correctamente el umbral? (pista: ¿La salida es estable?)
4. Hacé que la salida haga algo interesante (por ejemplo encender un LED o mover un servo)  
¿Para qué tipo de registros ligados al comportamiento te parece que puede utilizarse esto?

Este contenido fue preparado por Cecilia Herbert, Marcos Coletti y Juan Ignacio Sanguinetti Scheck para el laboratorio “Closed-loop behavior with Bonsai” del tercer módulo “NEURAL SYSTEMS, COGNITION, GENOMICS & BEHAVIOR” de la escuela “Latin American Training Program” de la Society for Neuroscience (SfN), con el apoyo de The Grass Foundation, el Latin American Regional Committee of the International Brain Research Organization (IBRO-LARC), el US/Canada Regional Committee (US/CRC) of IBRO y el Instituto de Investigaciones Biológicas Clemente Estable (IIBCE).

Parte de este contenido fue tomado del “Taller práctico abierto IBRO-LARC sobre adquisición de señales neuronales” organizado por Verónica de la Fuente, Cecilia Herbert, Azul Silva y Marcos Coletti, que contó con el apoyo de IBRO-LARC, la Sociedad Argentina de Investigación en Neurociencias y el Fondo Metropolitano de la Cultura, las Artes y las Ciencias de la Ciudad de Buenos Aires. El material del taller está disponible en <https://github.com/talleresopensource/taller-ibro-larc>.

A su vez, la parte de Bonsai de ese curso fue una traducción y adaptación de parte del contenido del curso “**Visual Reactive Programming with Bonsai**”, del “Cajal NeuroKit Advanced Neuroscience Training Programme”, y que fue desarrollado por NeuroGEARS, Ltd, cuyo material está disponible en <https://neurogears.org/vrp-2021/>.

Este contenido está bajo licencia [Atribución/Reconocimiento-CompartirIgual 4.0 Internacional \(CC BY-SA 4.0\)](#). Esto significa que:

#### **Sos libre de...**

- **Compartir** — copiar y redistribuir el material en cualquier medio o formato
- **Adaptar** — remezclar, transformar y construir a partir del material para cualquier propósito, incluso comercialmente.

#### **Bajo los siguientes términos:**

- **Atribución** — Debés dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Podés hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que vos o tu uso tienen el apoyo de la licenciante.
- **CompartirIgual** — Si remezclás, transformás o creás a partir del material, debés distribuir tu contribución bajo la misma licencia del original.
- **No hay restricciones adicionales** — No podés aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier uso permitido por la licencia.

Este resumen legible por humanos no es un sustituto de [la licencia](#). Para ver una copia de esta licencia, visitá el siguiente vínculo: <https://creativecommons.org/licenses/by-sa/4.0>.