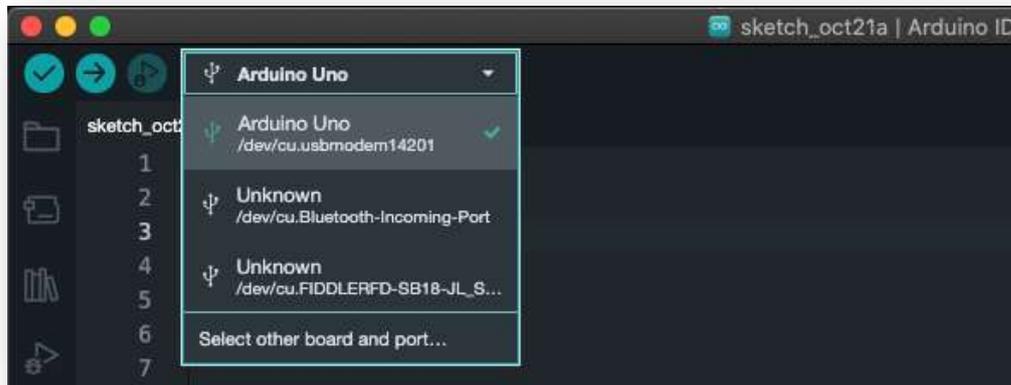
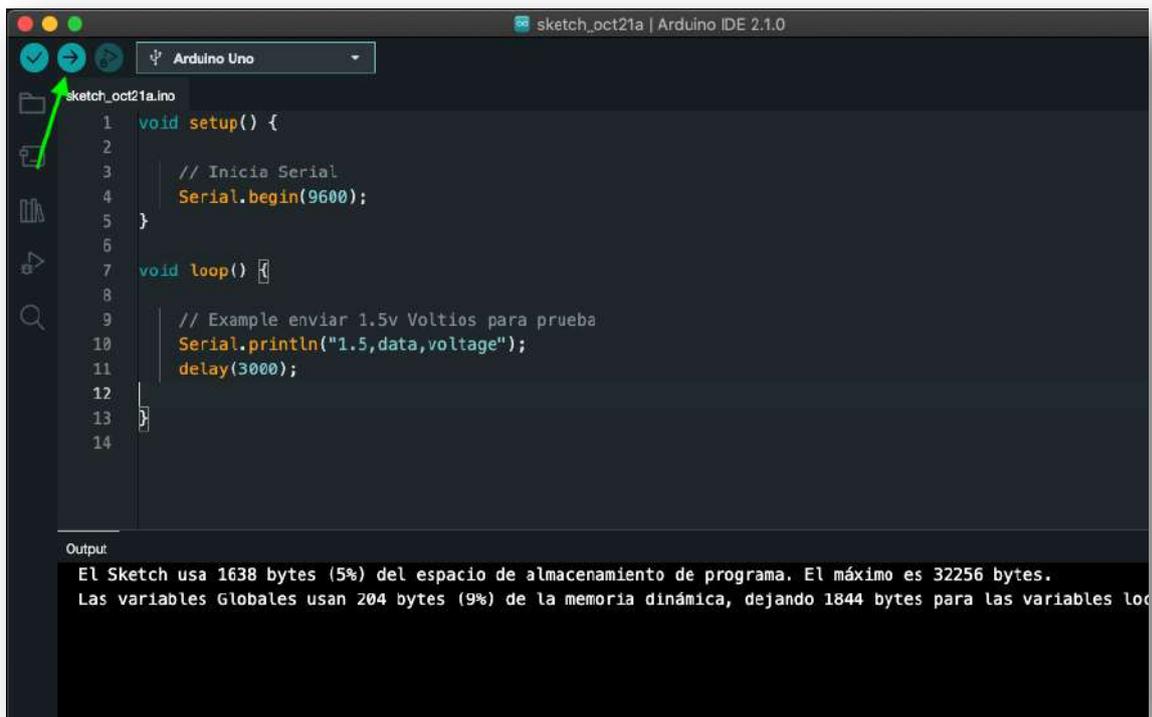


Conexión Hardware Arduino

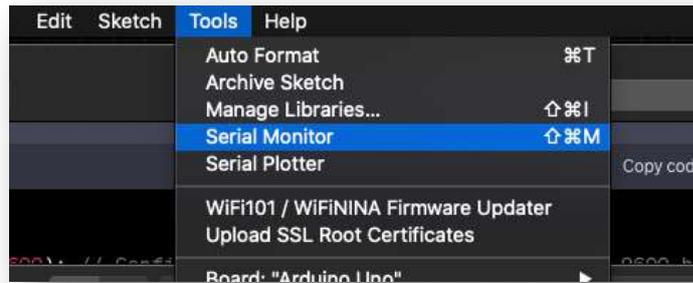
1. Alistando su arduino para enviar datos al software PCB Solution FBL
2. El primero ejemplo vamos a enviar datos de manera automática cada 3 segundos. La intención es ver que PCB Solution FBL reciba estos valores y los vaya agregando a las pinzas en la pantalla
3. Conecte su placa arduino y seleccione su placa en la lista



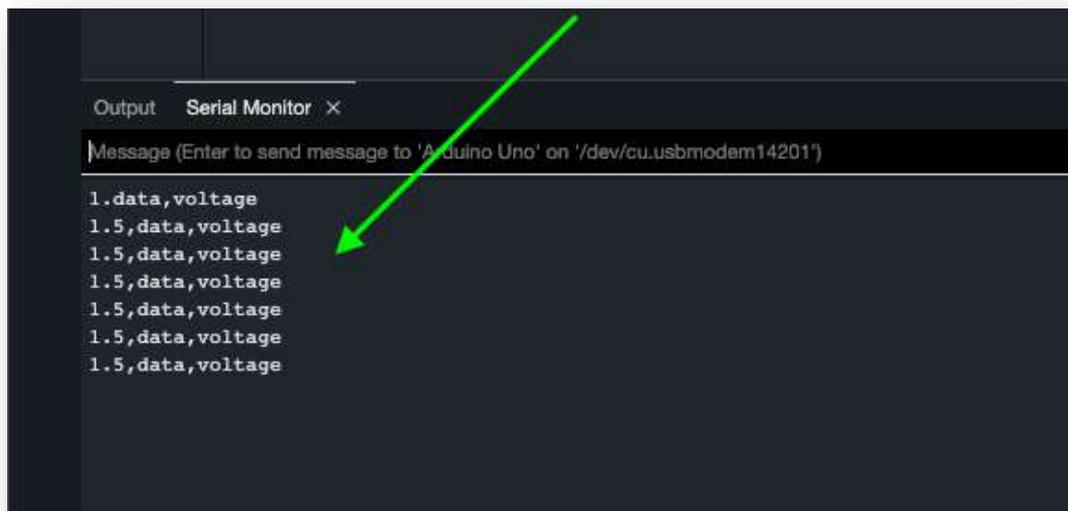
4. Grabe el siguiente código en su placa arduino, haciendo clic en botón cargar , que tiene la forma de una flecha a la derecha . (Todavía no conecte nada adicional a su placa arduino)



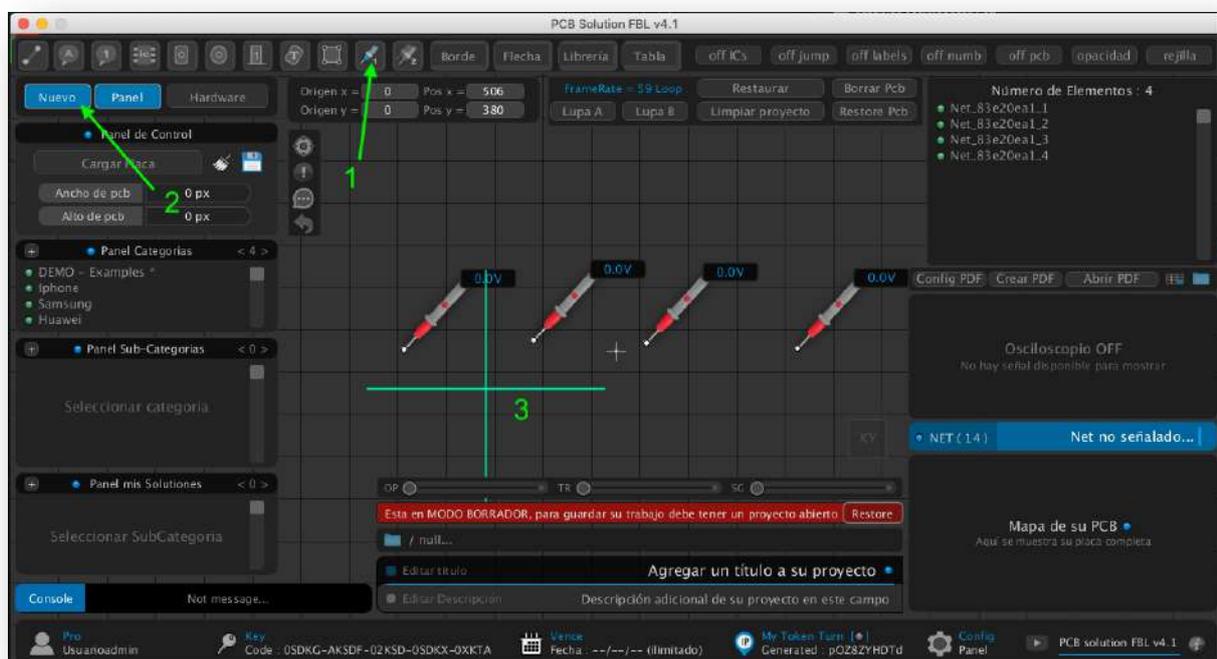
- 5 Ahora que su arduino esta con el código de prueba, Ejecute el monitor serial de arduino para ver si la placa envía datos cada 3 segundos. Le debe aparecer algo así



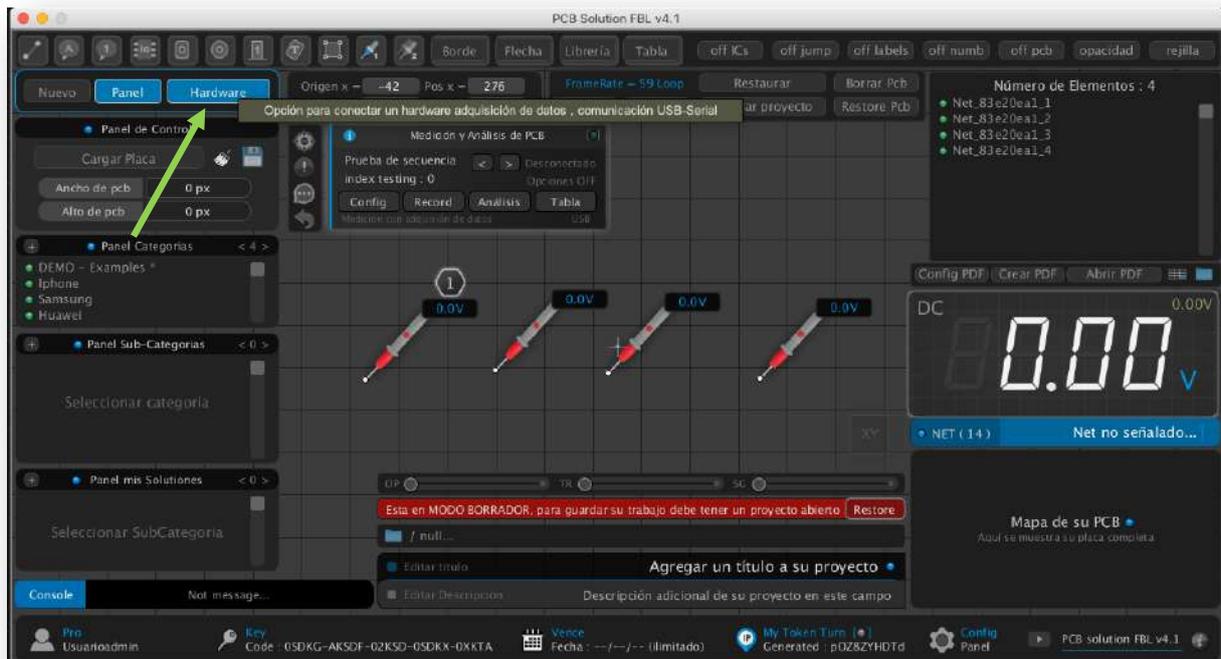
- 6 Esto debería de ver, que se imprima cada 3 segundos. Ya está listo su arduino ! . Cierre el programa de arduino



- 7 Ahora inicie PCB Solution FBL , debe crear unas cuantas pinzas, como ejemplo hay que crear 4 puntas de voltímetro en pantalla : Pinza > Nuevo > Clic pantalla



8 Ahora active el botón "HARDWARE"



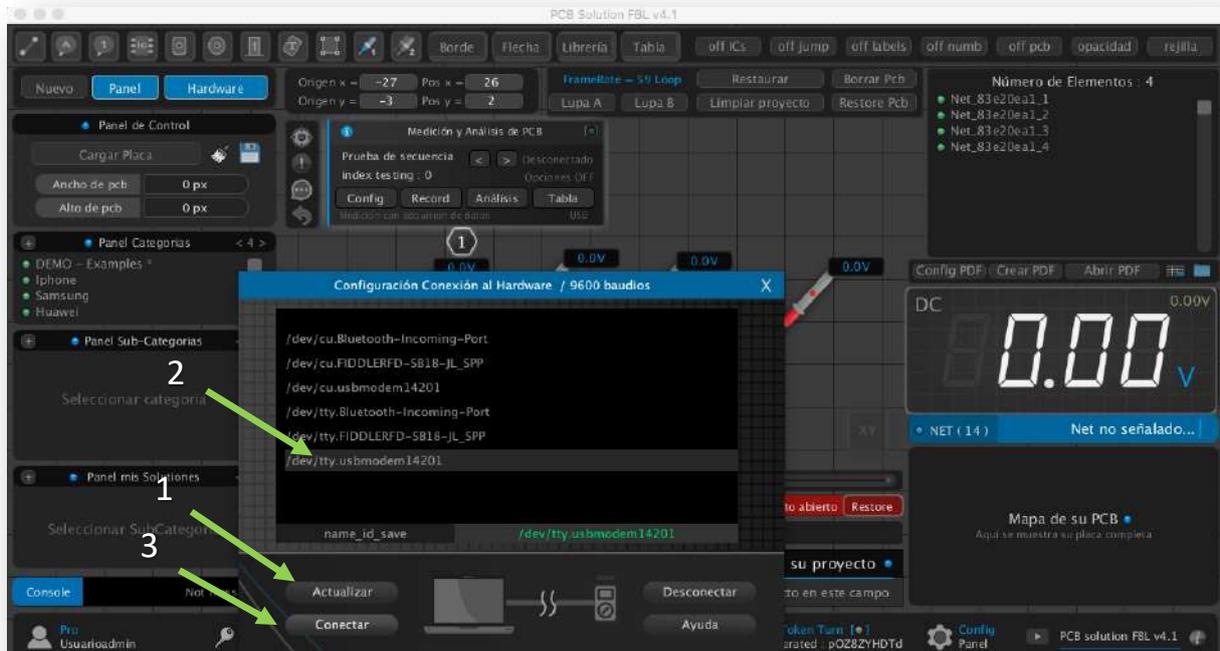
9 Conecte su placa arduino y presione el botón de "Configuración" del panel hardware que apareció



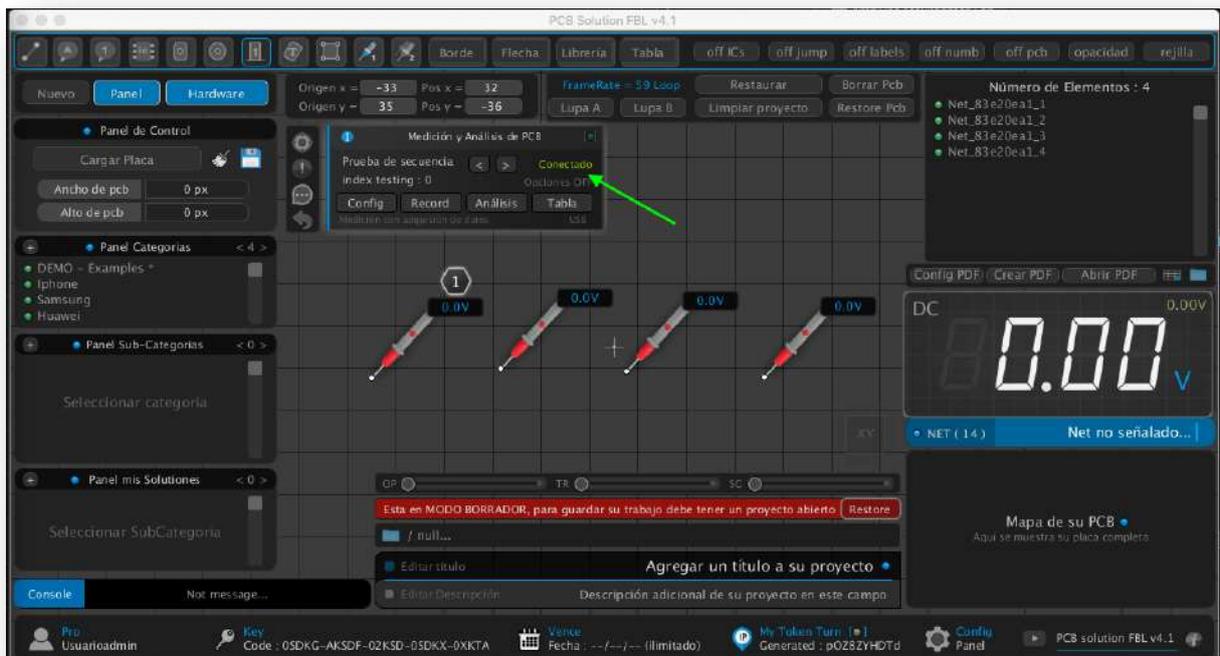
Cualquier placa arduino debería de funcionar para conectar al programa, nosotros hemos hecho test con las siguientes placas : (Arduino uno, Arduino mega, Arduino nano...)

10 presione **“Actualizar”** hasta que su arduino aparezca en la lista y lo selecciona su placa con clic izquierdo. Seguidamente haga clic en **“Conectar”** (Si conecta se cierra la ventana)

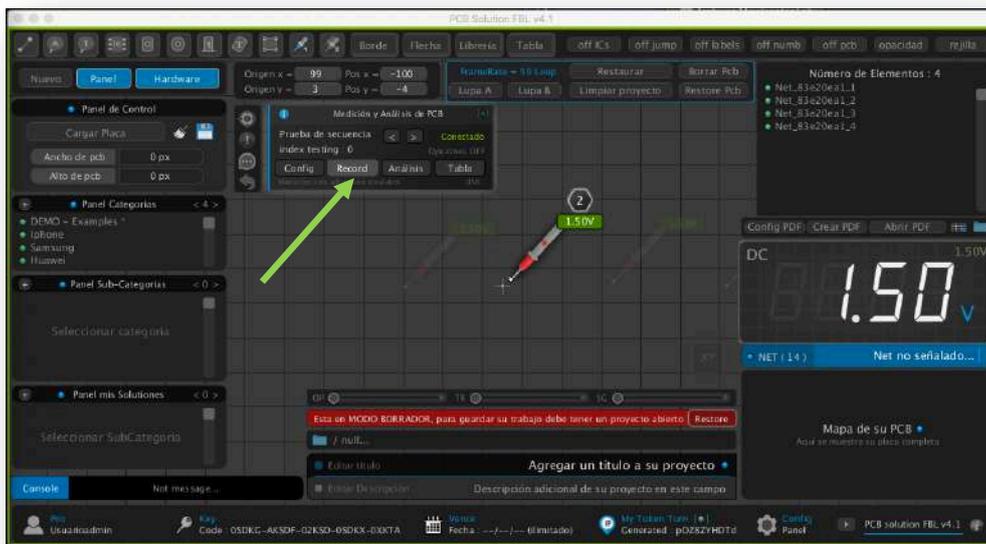
- WINDOWS : aparece como nombre **“COM”**
- MAC : aparece como **“/dev/tty.usbmodem ...”**



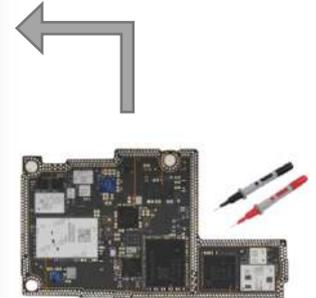
11 Ahora notará que dice **“Conectado”** en el panel de control



- 12 Ya está todo listo , debe ahora hacer clic en botón **“Record”** para recibir voltajes que arduino estará enviando. Podrás notar que cuando recibe un valor, el programa pasa automáticamente a la siguiente pinza disponible

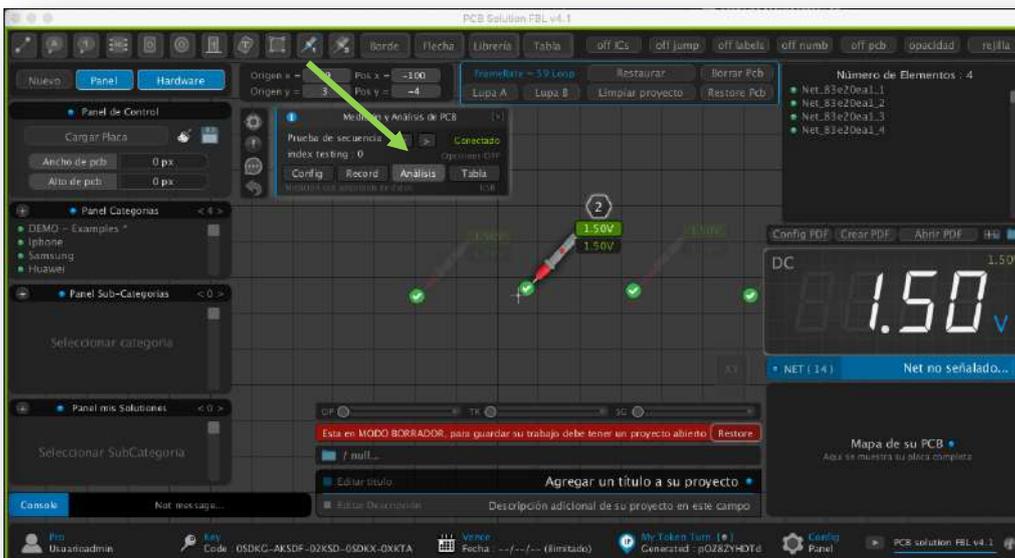


Enviando ...

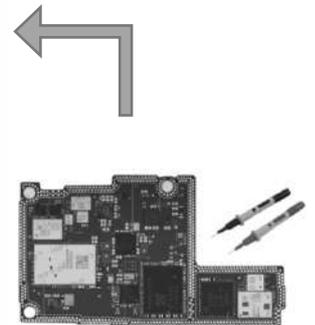


PCB buena
(Tomando muestras)

- 13 Ahora active el botón **“Análisis”** .Puede ver que se está tomando nuevas capturas de voltajes ,pero se guarda en un segundo campo. Aquí simulamos el análisis en otra placa defectuosa usando los voltajes buenos como referencia. Cuando salga rojo, es porque el programa detecto que el voltaje leído en su segunda placa es distinto, lo que puede revelar falla en la línea o componente



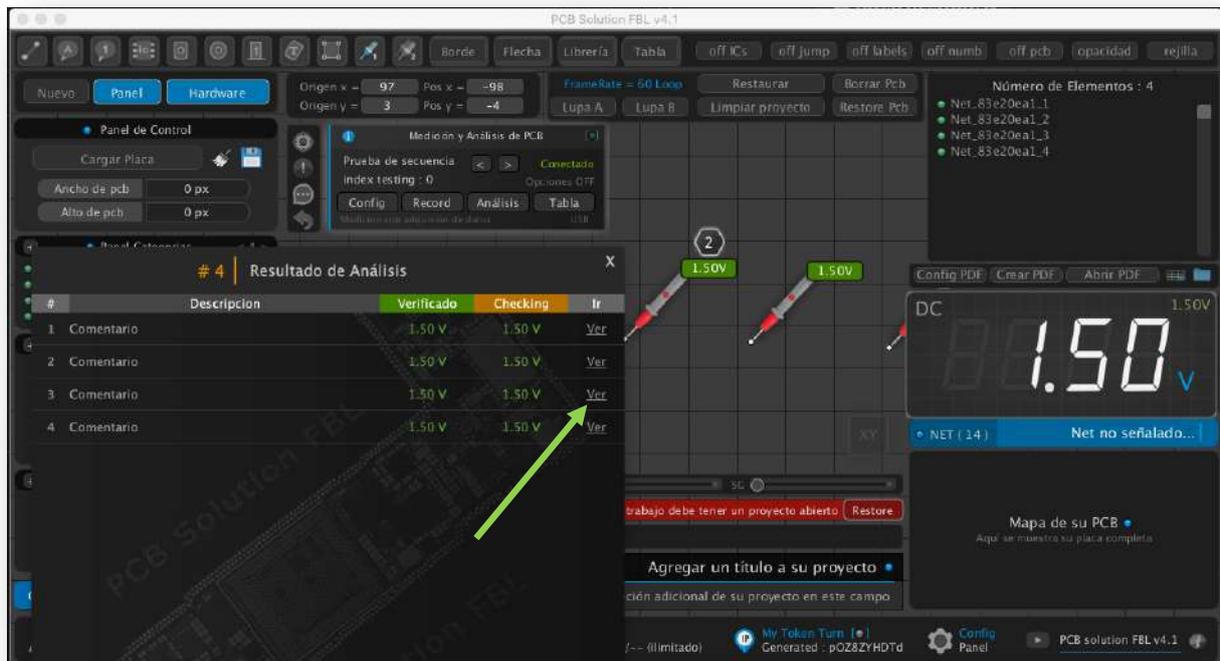
Enviando ...



PCB checking
(Captura y compara)

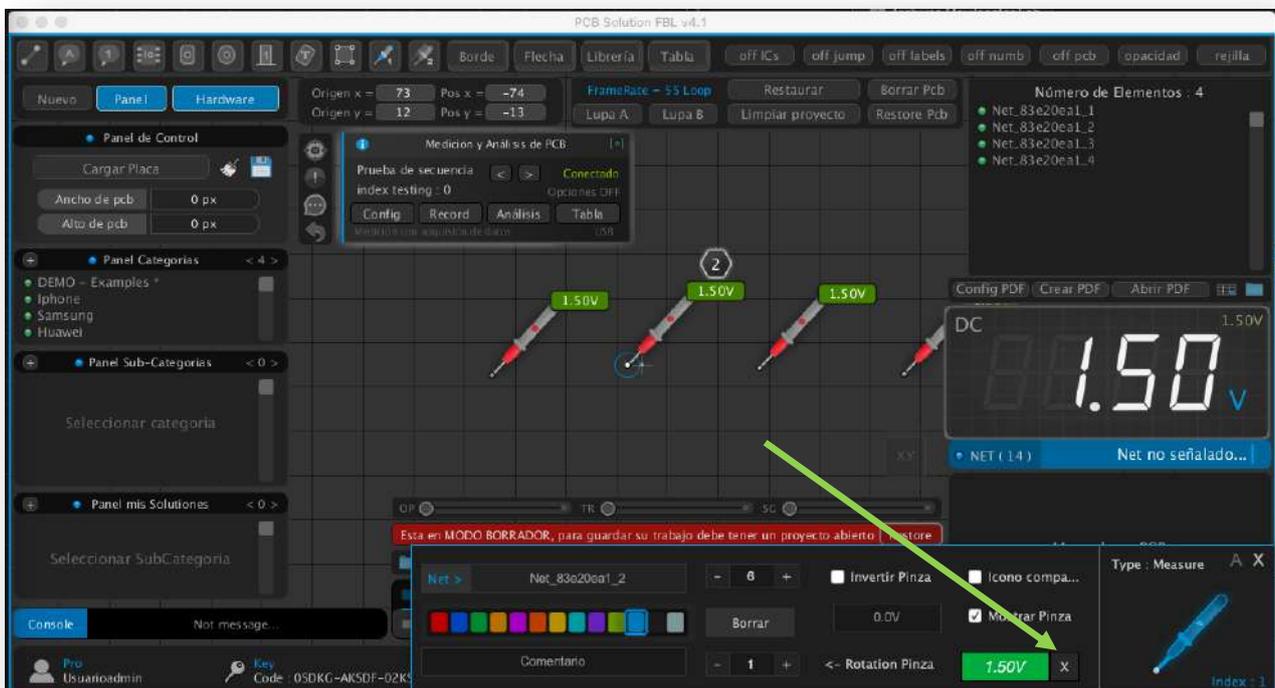
Espera que se haya capturado voltaje en todas las pinzas, y seguidamente ya puedes desactivar el botón de “Análisis”

14 Presiona el botón **"tabla"** para ver la lista de voltajes comparativos. La primera columna (verde) son los voltajes capturados de una placa en buen estado. Y la segunda columna son voltajes de una segunda placa que se busca encontrar algo extraño



NOTA : Dentro de la tabla puedes dar clic en **"ver"** para desplazar la pantalla donde se encuentra la pinza de medición.

NOTA ADICIONAL : Si quieres borrar la captura de voltaje hecha por hardware para la pinza, puedes hacerlo dando clic derecho sobre la pinza de medición y en propiedades clic en **"X"** donde está el voltaje capturado



EJEMPLO 2

Luego de probar el envío de datos de manera automática, ahora ya puedes probar este segundo código, en este ejemplo ahora si vamos a enviar un voltaje de arduino a PCB Solution FBL cada vez que presionemos un switch mecánico

```

cpp
Copy code

int switchPin = 2;
int analogPin = A0;
float voltage = 0.0;

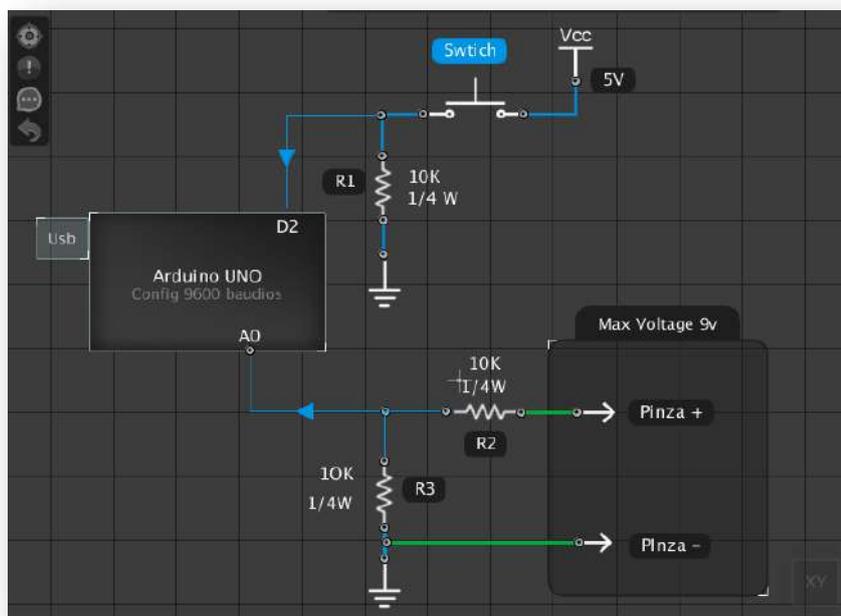
void setup() {
  pinMode(switchPin, INPUT); // Deja el pin 2 como entrada sin resistencia pull-up
  Serial.begin(9600);
}

void loop() {
  if (digitalRead(switchPin) == LOW) { // Si el interruptor se presiona (LOW)
    int rawValue = analogRead(analogPin); // Lee el valor analógico en A0
    voltage = (rawValue * 5.0 / 1023.0) * 2; // Convierte el valor a voltaje

    // Envía la cadena por el puerto serie
    Serial.print(voltage, 2); // Muestra el valor con 2 decimales
    Serial.print(",data,");
    Serial.println("voltage");
    delay(1000); // Espera 1 segundo para evitar múltiples lecturas
  }
}

```

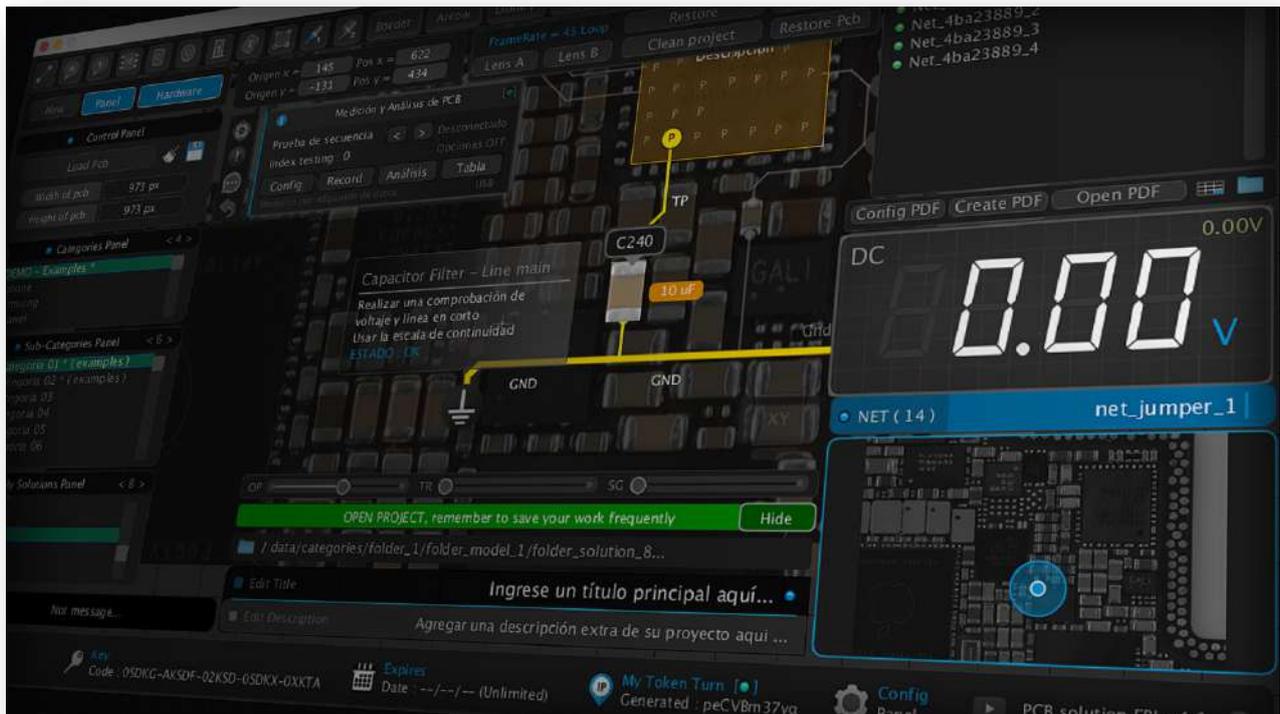
Arma este circuito básico de prueba, para un voltímetro muy básico



Importante : Este circuito debe medir voltajes de **0 a 9 voltios**. Asegúrese no medir voltajes superiores porque no tiene protección (OVP) y tampoco medir voltajes negativos. La finalidad solo en comprobar como puedes enviar voltajes usando una placa arduino.

El diseño del circuito más robusto para un medidor más completo dependerá del desarrollador que intenta usar esta función de PCB Solution FBL , en la cual le permite conectar un arduino como interface a su propio multímetro que ha desarrollado.

Una buena noticia , es que nosotros al tiempo que escribimos esta nota, ya nos encontramos desarrollando un voltímetro oficial para el software que va incluir, un diseño seguro con todo lo necesario como fusibles, resistencias de precisión , comparadores AMP OP, Condensadores de filtrado, protección sobre voltajes... y más detalles. Publicaremos pronto un medidor propio con otras novedades importantes, y la constante mejora de nuestro software PCB Solution FBL



Atte. Equipo Movilcenter Lab
Lima-Perú

Web oficial : www.movilcenter-lab.com