

## Electrónica para Todos con el Uso de Arduino: Experiencias Positivas en la Implementación de Soluciones Hardware-Software

Cristian Vidal-Silva<sup>(1)\*</sup>, Matías I. Lineros<sup>(2)</sup>, Guillermo E. Uribe<sup>(3)</sup> y César J. Olmos<sup>(3)</sup>

(1) Departamento de Administración, Facultad de Economía y Administración, Universidad Católica del Norte, Avenida Angamos 0610, Antofagasta – Chile (e-mail: cristian.vidal@ucn.cl)

(2) Ingeniería Civil Informática, Escuela de Ingeniería y Negocios, Universidad Viña del Mar, Campus Rodelillo, Agua Santa 7055, Viña del Mar – Chile (e-mail: matias.lineros.r@gmail.com)

(3) Ingeniería Civil en Minas, Escuela de Ingeniería y Negocios, Universidad Viña del Mar, Campus Rodelillo, Agua Santa 7055, Viña del Mar – Chile (e-mail: guribe@uvm.cl; cesar.olmos@uvm.cl)

\* Autor a quien debe ser dirigida la correspondencia.

*Recibido Feb.28, 2019; Aceptado Abr. 25, 2019; Versión final May. 27, 2019, Publicado Dic. 2019*

---

### Resumen

Este trabajo presenta experiencias para con el desarrollo de proyectos Arduino, de estudiantes de Ingeniería Civil Informática de la Universidad Viña del Mar (ICI-UVM) en Chile, en un módulo (curso) de su proceso formativo. Las áreas de informática y electrónica usualmente consideran rumbos formativos diferentes sin presentar una clara hegemonía en el proceso de enseñanza, con un bajo desarrollo de competencias integradoras para la producción de soluciones que combinen hardware y software para el procesamiento de señales, lo que Arduino permite disminuir en alto grado. Este trabajo destaca que, en tan sólo seis semanas, los estudiantes de Ingeniería Civil Informática de la Universidad Viña del Mar fueron capaces de desarrollar competencias para el uso de Arduino con el fin de implementar sistemas prototipo para la adquisición y visualización de variables ambientales. Esto es sólo una prueba de que la electrónica y la computación están más accesible para todos hoy en día, con bajas barreras de costo y aprendizaje.

*Palabras clave: Arduino; electrónica; circuitos electrónicos; programación; procesamiento de señales*

## Electronics for Everybody using Arduino: Positive Experience in the Implementation of Hardware-Software Solutions

### Abstract

This paper presents experiences for the development of Arduino projects of Computing Engineering students of the Viña del Mar University in Chile in a module (course) of their formative process. The computing and electronic areas usually consider different formative courses without presenting a clear hegemony in the teaching process. There exists low development of competences to produce solutions combining hardware and software for the signal processing, barriers that Arduino permits reducing in a high degree. This article highlights that in just six weeks, the students of Computing Engineering of the Viña del Mar University were able to develop competences for the use of Arduino to implement prototype systems for the acquisition of environmental variables. This is just a proof that electronics and computing are more accessible for everybody nowadays, with low barriers of cost and learning.

*Keywords: Arduino; electronic; electronic circuits; programming; signal processing*

## INTRODUCCIÓN

El diseño e implementación de circuitos analógicos y digitales para un trabajo integrado con sistemas computacionales es usualmente no trivial para estudiantes del área informática (Hertzog y Swart, 2016). Los cursos de formación profesional de temáticas relacionadas a circuitos digitales y electrónica usualmente demandaban el acceder y utilizar componentes cercanos a las teorías de electricidad y electrónica, y también enfrentar dificultades con las interfaces de entrada/salida de los sistemas computacionales tradicionales. Previamente, las interfaces tradicionales de los computadores era los puertos serial y paralelo, y luego el puerto USB, y no existía una compatibilidad de enlace directo a componentes hardware externos. Además, tampoco existían (o eran de poca o nula difusión) mecanismos de comunicación para con dichas interfaces hardware en los lenguajes de programación. A pesar de esto, los trabajos de Vidal y Pavesi (2005), Vidal y Gatica (2010), y Vidal et al. (2011) describen una experiencia exitosa de procesos de digitalización y procesamiento de señales analógicas, con módulos de amplificación, filtrado, y conversión de señales analógicas a digitales, necesarios para su manipulación y tratamiento computacional. Una compleja situación que hoy ya es casi una completa historia para que los estudiantes de ingeniería puedan trabajar en el diseño y desarrollo de soluciones en problemas de la vida real gracias a Arduino (Hertzog y Swart, 2016). Este permite incluso a jóvenes y adolescentes que no estudian en la universidad, desarrollar soluciones Arduino sin la completa necesidad de conocer todos los detalles técnicos de electricidad y electrónica (Martín-Ramos et al., 2016). Tal como lo presenta Martín-Ramos et al. (2017), Arduino también facilita el acceso a la programación y a las tutorías para jóvenes no universitarios.

Según Margolis (2011) y Jones (2017), Arduino fue creado durante 2005 por el entonces estudiante Massimo Banzhi del instituto IVRAE en Italia, para integrar y facilitar el aprendizaje integrado de computación y electrónica para estudiantes de dicho instituto, con el principio de Hardware abierto (del inglés Open-Hardware). Desde entonces, dado su fácil acceso, ya que inicialmente era gratis y luego de bajo costo, y por su gran éxito en el instituto IVRAE en Italia, se inició su distribución y comercialización tanto en Italia como en el resto del mundo. Gracias al éxito de Arduino, ya existen dispositivos hardware y aplicaciones software compatibles con Arduino. Por ejemplo, pantallas de cristal líquido y sensores de bajo costo hoy son accesibles en el mercado, así como también entornos de desarrollo integrado web como Tinkercad (2019). Dichos ejemplos tan sólo avalan y motivan el desarrollo de soluciones con Arduino.

Gracias a los avances tecnológicos, tanto en hardware como en software, en la búsqueda de una mayor integración de estos, cada día es más viable el desarrollo de soluciones integradas de hardware y software para estudiantes y profesionales de dichas áreas. Por ejemplo, tal y como se describe en el trabajo de Vidal y Pavesi (2005), una solución integrada de hardware y software tradicional requiere fragmentar su diseño en hardware y software sin perder de vista su integración. Aun cuando los sistemas de computación de escritorio reflejan grandes avances tanto en velocidad de cómputo y capacidad de memoria, estos son sistemas que involucran responsabilidades adicionales, tanto de sistema operativo como de usuario, además de la responsabilidad de la existencia de señales externas para su recepción y procesamiento. Es decir, existe un delta de tiempo entre captura y procesamiento de la señal que, para señales de alta variabilidad, según Teorema de Shannon y Nyquists (Antoniou, 2016), los sistemas computacionales tradicionales pueden resultar poco prácticos para trabajar con sistemas de recepción y procesamiento de señales.

Dada la amplia difusión y disponibilidad de Arduino, este trabajo presenta experiencias prácticas del uso de Arduino en el diseño y construcción de soluciones para la adquisición de señales del ambiente con el uso de sensores tanto analógicos como digitales, para la visualización de dichas señales y la realización de acciones para el control de estas según el contexto y problemática definida. Así, este trabajo demuestra la disponibilidad y cercanía de Arduino para; primero, disminuir y eliminar barreras de costo e incompatibilidad de hardware antes mencionadas; segundo, para facilitar la integración de componentes hardware y software para el desarrollo de soluciones integradas; y tercero, para facilitar el desarrollo de competencias de electrónica y computación para el desarrollo de soluciones integradoras para estudiantes con una baja o nula formación previa en dichas áreas. Es necesario destacar que los estudiantes de este experimento tenían cero formaciones previas en electrónica, y que fueron los primeros en esta casa de estudios en experimentar con Arduino. Este trabajo es una extensión de los resultados descritos en Lineros et al. (2018).

## OTROS ANTECEDENTES

La tecnología Arduino resulta en un sistema de computación con la integración directa de las áreas de hardware y software para el diseño e implementación de soluciones. Arduino hoy representa una compañía de código y hardware abierto para facilitar el acceso y uso conjunto de la electrónica y computación para el desarrollo de sistemas (Jones, 2017). Al igual que una computadora convencional, una placa Arduino puede realizar una multitud de funciones, y requiere de entradas y/o salidas para un mayor valor de su uso (Nusse, 2013). En la práctica, según (ArduinoChile, 2019), ya existen diferentes placas Arduino disponibles tales como

Arduino Uno R3, Arduino Leonardo y Arduino Mega, donde los dos últimos representan una evolución desde Arduino Uno R3, en términos de mejoras e inclusión de nuevas características hardware. Por ejemplo, una versión de Arduino Mega permite la interacción con teléfonos móviles Android. A continuación, se describen detalles de componentes esenciales de Arduino Uno R3 usado en el desarrollo de los proyectos base de este trabajo: microcontrolador ATmega328; voltaje de entrada 7-12V; 14 pines digitales de I/O, con 6 salidas PWM para la generación de señales analógicas; 6 entradas análogas; 32KB de memoria Flash; Reloj de 16MHz de velocidad.

La figura 1 muestra y detalla los componentes de Arduino Uno R3. Dadas las entradas y salidas de una placa Arduino, para una interacción directa con ella, diferentes componentes y dispositivos de electrónica con el nivel de voltaje y corriente de Arduino (Margolis, 2011). Como ejemplos de uso y aplicación de Arduino, ArduinoHome (2019) presenta ejemplos para el hogar (domótica) tales como el control de funcionamiento de luces y equipos electrónicos, y para la adquisición y procesamiento de señales con Arduino Uno R3. Para la programación de placas Arduino, se utiliza el marco de trabajo de programación de código abierto para microcontroladores Wiring (Wiring, 2019) que se basa en el lenguaje de programación Processing (Shiffman, 2015) que fue desarrollado en Java con sintaxis y semántica similares a C y C++.

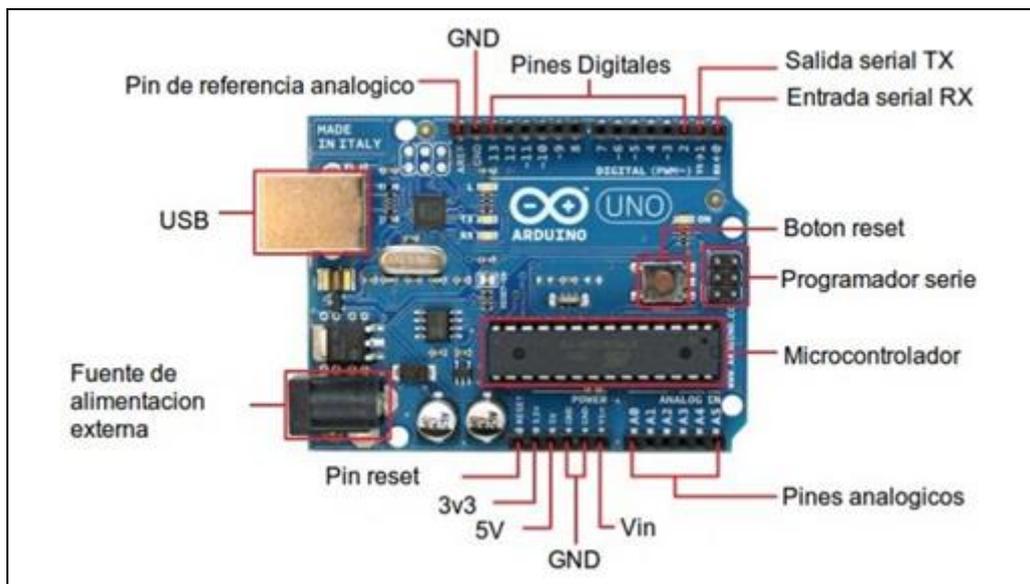


Fig. 1: Placa Arduino Uno R3 (Adaptada de Lineros et al., 2018).

a. Circuito Arduino ejemplo Hola Mundo.

```

int pinLed = 13;

void setup()
{
  pinMode(pinLed, OUTPUT);
}

void loop() {
  digitalWrite(pinLed, HIGH);
  delay(1000); // Esper 1000 milisegundos
  digitalWrite(pinLed, LOW);
  delay(1000); // Esper 1000 milisegundos
}
                    
```

b. Código Arduino ejemplo Hola Mundo.

Fig. 2: Hardware y Código ejemplo básico Hola Mundo con Arduino (Adaptada de Lineros et al., 2018).

Un programa Arduino consiste de dos funciones esenciales, tal como la función principal (main) en C y Java, pero esta vez con semánticas algo diferentes: la función setup y la función loop (Martín-Ramos et al., 2017). La función setup tiene su ocurrencia al inicio de funcionamiento de sistema Arduino (sólo al inicio), con el objetivo de establecer valores y/o condiciones iniciales de funcionamiento, mientras que la función loop es ejecutada de manera cíclica durante el funcionamiento del sistema Arduino (similar a un ciclo infinito). La figura 2 presenta el circuito y código de un ejemplo básico Arduino para encender un Led, ejemplo que sería análogo al Hola Mundo de lenguaje C. Entonces, para cada ejecución periódica de la función loop en este ejemplo, se encenderá el Led conectado por el pin de conexión pinLed (`digitalWrite(pinLed, HIGH)`), se envía un voltaje alto al pin 13. Luego, se realiza una pausa de 1 segundo (`delay(1000)`), para entonces enviar un voltaje bajo por el mismo pin de conexión pinLed para apagar el led (`digitalWrite(pinLed, LOW)`) y finalmente se procede a realizar una pausa de 1 segundo (`delay(1000)`). La velocidad de re-ejecución de la función loop se determina por la velocidad de la placa base Arduino, junto con la velocidad de trabajo e interacción entre los componentes que se usan en cada ejecución de dicha función.

Tal y como lo señalan Brock et al. (2009), por la accesibilidad y bajo costo de las placas Arduino y su accesible entorno de programación, es posible hoy en día la enseñanza de sistemas embebidos en cursos iniciales de programación y sistemas digitales. Por la relevancia y beneficios de la programación y el desarrollo de soluciones algorítmicas en la educación para niños, tales como el fomento de la creatividad, expresividad, autoconfianza y lógica, Merkouris et al. (2015) presentan experiencias de niños en el desarrollo de soluciones de computación ubicua con Arduino, esto es, Arduino con robots y computación móvil. Los resultados de dichos trabajos no hacen más que impulsar el uso de Arduino en la educación, independiente de la edad de los educandos.

## METODOLOGÍA

Este trabajo fue desarrollado por estudiantes del curso Arquitectura y Sistemas Operativos de quinto semestre ICI UVM. Los estudiantes, si bien tienen una base en programación, ellos poseen un bajo o nulo nivel en electrónica. Las competencias principales por desarrollar en dicho módulo son: 1. Conceptualizar los componentes fundamentales hardware y software de un sistema computacional moderno, y 2. Plantear soluciones prototipo concurrentes tanto hardware como software en una organización.

Los experimentos de este trabajo con Arduino Uno R3 requieren utilizar pines analógicos y digitales para la captura de señales de temperatura y ultrasonido, respectivamente, y pines digitales para realizar acciones de control definidas. A continuación, se describen los dos proyectos: (i) *proyecto Arduino de sistema para la detección de objetos cercanos para evitar colisiones*; y (ii) *proyecto Arduino de sistema para el control de la temperatura*.

### 1) Proyecto Arduino de sistema para la detección de objetos cercanos para evitar colisiones.

Este proyecto Arduino tiene como objetivo la implementación de un sistema para la detección de objetos y la cercanía de estos para evitar colisiones. Para su funcionamiento, este sistema incluye dos sensores de ultrasonido HC-SR04, con una lectura por turnos de estos, para almacenar los valores de cada sensor en vectores cíclicos, con índices para conocer la ubicación del primer y último valor en cada vector. El sistema Arduino del experimento 1, plantea un funcionamiento por turnos para la lectura de cada sensor, donde el turno inicial se determina de manera aleatoria. En cada iteración de funcionamiento, el sistema calculará la menor y mayor distancia obtenidas hasta el momento, junto con el promedio y la desviación estándar de los datos ya almacenados del sensor de turno, para su visualización en una pantalla de cristal líquido LCD. Aun cuando la lectura de ambos sensores se podría realizar en la misma iteración de la función loop, se solicita el trabajo por turnos para una coordinación en la labor de ambos sensores para el cálculo de sus valores y su impacto en el sistema.

Se asume que cada sensor puede estar en uno de los siguientes estados, según valores  $d_1$  y  $d_2$  de distancia o cercanía previamente definidos con  $d_2 \geq d_1$ : i) normal: se detecta un objeto a una distancia  $d_0$  mayor que  $d_2$  ( $d_0 > d_2$ ), ii) precaución: se detecta un objeto a una distancia  $d_0$  en el rango ( $d_1, d_2$ ], esto es, por sobre  $d_1$  pero menor o igual a  $d_2$  ( $d_1 < d_0 \leq d_2$ ) y iii) crítico: se detecta un objeto a una distancia  $d_0$  que es menor que  $d_1$  ( $d_0 < d_1$ ). Cada sensor tiene asociado un Led RGB y una alarma, y, dependiendo del estado de cada sensor, se enciende su Led RGB con un color definido: rojo para estado crítico, azul para estado precaución y verde para estado normal. La alarma de un sensor se enciende cuando este se encuentra en estado crítico. La figura 3 muestra el diseño del circuito para la solución de este experimento con la herramienta en línea Tinkercad (Tinkercad, 2019).

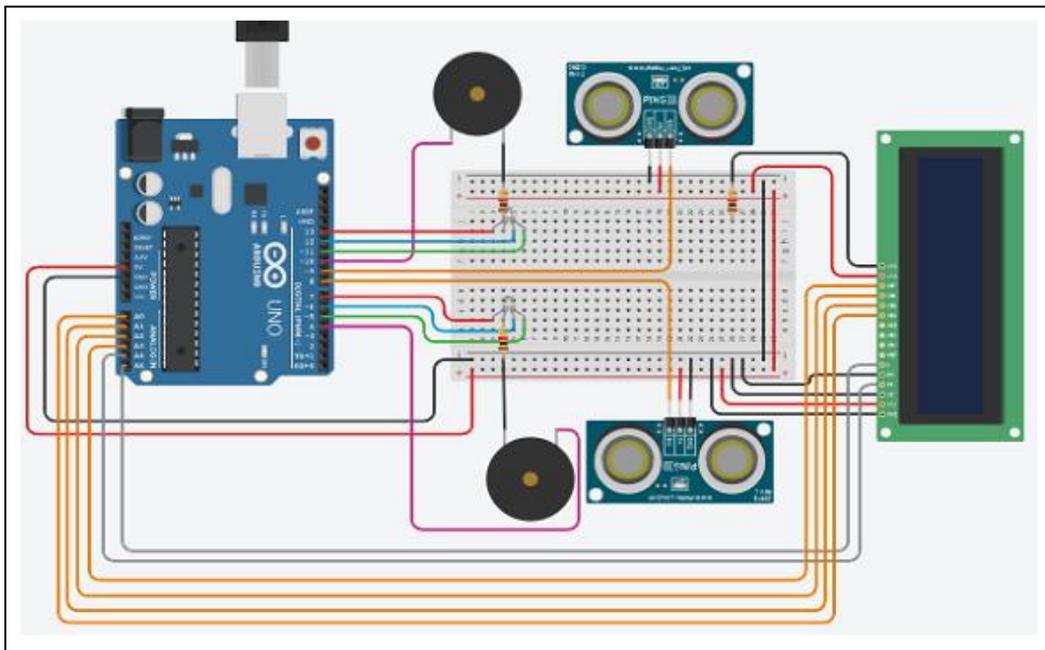


Fig. 3: Diseño de sistema Arduino de sistema para la detección de objetos cercanos para evitar colisiones (Adaptada de Lineros et al., 2018).

2) Proyecto Arduino de sistema para el control de la temperatura.

El segundo experimento Arduino de los estudiantes ICI UVM es análogo al primero ya que consiste en un sistema para el control de temperatura. Así, este sistema utiliza 2 sensores LM35DZ con 3 estados posibles, dos Leds RGB (uno asociado a cada sensor), una pantalla de cristal líquido LCD, y una alarma del sistema. Este sistema realiza lectura de cada sensor por turno, con el primer turno aleatorio. Para cada sensor se definen 2 valores  $T_1$  y  $T_2$ , con  $T_1 \leq T_2$ , los que permiten definir su estado según su último valor obtenido  $T_o$ . Así, cada sensor puede estar en uno de los 3 siguientes estados: i) normal: la última temperatura leída o sensada  $T_o$  está bajo  $T_1$  ( $T_o < T_1$ ) ii) precaución: la última temperatura leída  $T_o$  está en el rango ( $T_1, T_2$ ) ( $T_1 \leq T_o < T_2$ ), y iii) crítico: la última temperatura sensada  $T_o$  está sobre  $T_2$  ( $T_o > T_2$ ).

La pantalla LCD muestra el valor actual, mínimo, máximo, promedio y desviación estándar para el sensor de turno. Similar al experimento 1, la alarma del sistema sonará cuando alguno de los sensores se encuentre en un estado crítico. La figura 4 muestra el diseño del circuito con Tinkercad (Tinkercad, 2019) de este experimento. Las figuras 5, 6, 7 y 8 presentan un ejemplo de solución de código modular para el 1er experimento. El código para la solución del segundo experimento es análogo.

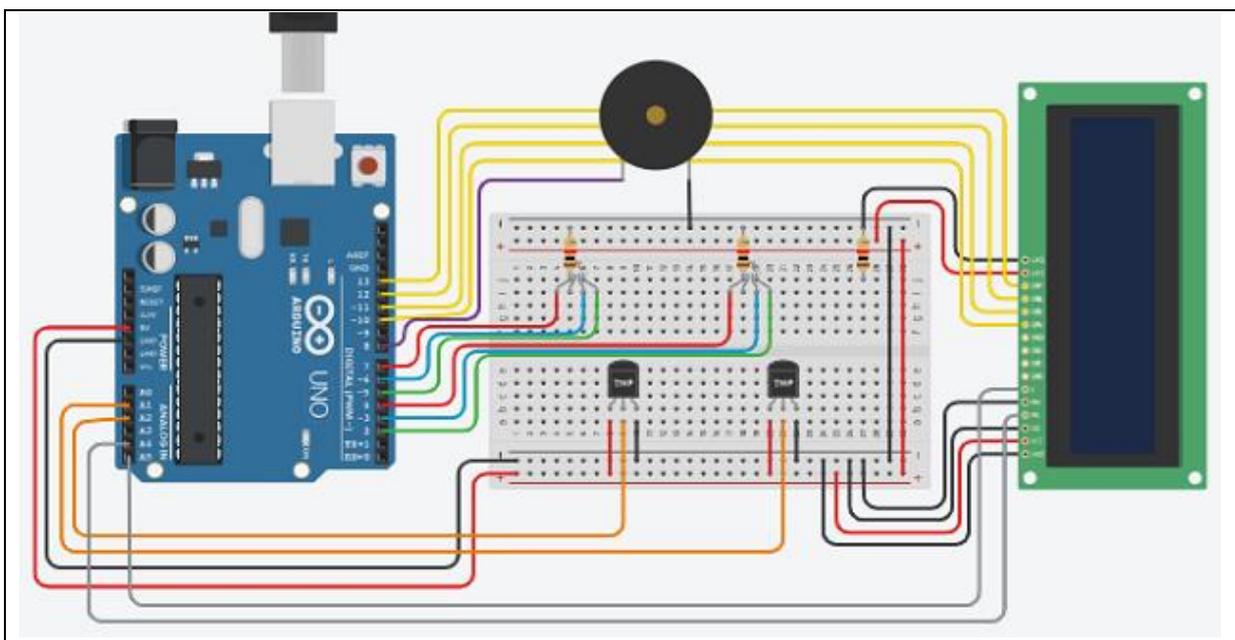


Fig. 4: Diseño de sistema Arduino de sistema para el control de la temperatura (Adaptada de Lineros et al., 2018).

```

#include <LiquidCrystal.h>
LiquidCrystal lcd(11, 10, 9, 8, 7, 6);

#define pinecho 2
#define pintrigger 3
#define pinecho2 4
#define pintrigger2 5

int valorMin1=15, valorMax1=40;
int valorMin2=15, valorMax2=40;

const int pinRojol = A3;
const int pinVerdel1 = A4;
const int pinAzul1 = A5;

const int pinRojol2 = A0;
const int pinVerde2= A1;
const int pinAzul2 = A2;

const int pinAlarm2 = 13;
const int pinAlarm1 = 12;

int vmin1 = 9999; int vmin2 = 9999;
int vmax1 = -1; int vmax2 = -1;

int tam = 100;
double valse1[100]; double valse2[100];
unsigned int tiempo, distancia;

void EncenderLed(int pin){
  digitalWrite(pin, HIGH);
  delay(500);
  digitalWrite(pin, LOW);
}

float CalcularSumatoria(String s){
  float prom = 0;

  if(s == "S1"){
    for(int i=0; i<tam; i++){
      prom = prom + valse1[i];
    }else{
      for(int i=0; i<tam; i++){
        prom = prom + valse2[i];
      }
    }

  return prom;
}

int CalcularCantidad(double valse[]){
  int cantidad = 0;

  for(int i=0; i<tam; i++){
    if(valse[i]> 0)
      cantidad = cantidad+1;
  }

  return cantidad;
}

```

Fig. 5: Variables y funciones para encender Led RGB, calcular sumatoria y lecturas (Adaptada de Lineros et al., 2018).

```

void MensajeLCD(int valor,String sensor,double vmin,double vmax){
  double sum, desvia; int promedio, cantidad;
  lcd.clear(); lcd.setCursor(0,0); lcd.print(sensor);

  if(sensor == "S1"){
    valse1[CalcularCantidad(valse1)] = valor;
    cantidad = CalcularCantidad(valse1);
    sum = CalcularSumatoria(sensor);
    promedio = round((sum/cantidad));
    desvia = CalcularDesviacion(promedio, cantidad, sensor);
    Serial.print("cantidad = ");Serial.println(cantidad);
    Serial.print("promedio = ");Serial.println(promedio);
    Serial.print("s =");Serial.println(desvia);
  }else{
    valse2[CalcularCantidad(valse2)] = valor;
    cantidad = CalcularCantidad(valse2);
    sum = CalcularSumatoria(sensor);
    promedio = round((sum/cantidad));
    desvia = CalcularDesviacion(promedio, cantidad, sensor);
    Serial.print("cantidad = ");Serial.println(cantidad);
    Serial.print("promedio = ");Serial.println(promedio);
    Serial.print("s =");Serial.println(desvia);
  }

  lcd.print(" DA: "); lcd.print(round(valor));
  lcd.print("-"); lcd.print(round(vmin));
  lcd.setCursor(0,1); lcd.print(round(vmax));
  lcd.print("-"); lcd.print(promedio);
  lcd.print("-"); lcd.print(desvia);
  delay(600);
}

```

Fig. 6: Función para enviar mensaje a pantalla LCD (Adaptada de Lineros et al., 2018).

```
float CalcularDesviacion(int prom,int cant, String s){
    double total = 0;

    if(s == "S1"){
        for(int i=0; i<tam; i++){
            if(valsen1[i]>0)
                total = total + pow((valsen1[i]-prom),2);
        }
    }else{
        for(int i=0; i<tam; i++){
            if(valsen2[i]>0)
                total = total + pow((valsen2[i]-prom),2);
        }
    }

    total = (total / (cant-1));
    total = sqrt(total);

    return total;
}
```

Fig. 7: Función para calcular la desviación estándar de valores leídos (Adaptada de Lineros et al., 2018).

<pre>void CalcularValMinMax(double valorCen,     String s){     if(s == "S1"){         if(valorCen &lt; vmin1)             vmin1 = round(valorCen);         if(valorCen &gt; vmax1)             vmax1 = round(valorCen);     }else{         if(valorCen &lt; vmin2)             vmin2 = round(valorCen);         if(valorCen &gt; vmax2)             vmax2 = round(valorCen);     } }  void CalcularDistancia(int dist,     int valorMax,     int valorMin, int pinVerde,     int pinAzul, int pinRojo,     int pinAlarm){     if(dist &gt; valorMax){         pinMode(pinAlarm, INPUT);         EncenderLed(pinVerde);     }     else if(dist &lt;= valorMax &amp;&amp;         dist &gt;= valorMin){         pinMode(pinAlarm, INPUT);         EncenderLed(pinAzul);     }     else{         if(pinAlarm == pinAlarm1){             pinMode(pinAlarm1, OUTPUT);             tone(pinAlarm1, 234, 2000);         }         else if(pinAlarm == pinAlarm2){             pinMode(pinAlarm2, OUTPUT);             tone(pinAlarm2, 954, 2000);         }     }      EncenderLed(pinRojo); } }</pre>	<pre>void setup(){     lcd.begin(16,2);     pinMode(pinecho, INPUT);     pinMode(pintrigger, OUTPUT);     pinMode(pinecho2, INPUT);     pinMode(pintrigger2, OUTPUT);     pinMode(pinRojo1, OUTPUT);     pinMode(pinAzul1, OUTPUT);     pinMode(pinVerde1, OUTPUT);     pinMode(pinRojo2, OUTPUT);     pinMode(pinAzul2, OUTPUT);     pinMode(pinVerde2, OUTPUT);      Serial.begin(9600); }  void loop(){     CalcularDistancia(         ObtenerValorSensor(pintrigger,             pinecho, "S1"), valorMax1,             valorMin1, pinVerde1,             pinAzul1, pinRojo1, pinAlarm1);      MensajeLCD(         ObtenerValorSensor(pintrigger,             pinecho, "S1"), "S1", vmin1, vmax1);      CalcularDistancia(         ObtenerValorSensor(pintrigger2,             pinecho2, "S2"), valorMax2,             valorMin2, pinVerde2,             pinAzul2, pinRojo2, pinAlarm2);      MensajeLCD(         ObtenerValorSensor(pintrigger2,             pinecho2, "S2"), "S2", vmin2, vmax2); } }</pre>
--	---

Fig. 8: Funciones para calcular distancias mínima y actual, funciones setup y loop (Adaptada de Lineros et al., 2018).

## RESULTADOS Y DISCUSIÓN

Desde un punto de vista práctico, los 7 grupos de trabajo de estudiantes ICI UVM lograron finalizar exitosamente sus proyectos, esto es, 3 grupos con el primer experimento y 4 grupos con el segundo. En la presentación final de estos trabajos, los grupos demostraron afinidad con Arduino y electrónica asociada, y en el desarrollo de soluciones modulares de programación. Por ejemplo, como el proyecto requería la presentación de avances, las primeras versiones de estos sólo consideraban el uso de variables globales con repetición de código extremo en algunos casos.

Junto con la presentación final de estos proyectos, los estudiantes entregaron un informe estilo artículo de revista para desarrollar competencias de escritura de informes científicos y desarrollaron habilidades para redactar en 3era persona con el uso de referencias y citas a ideas no propias. Las tablas 1 y 2 muestran resultados del funcionamiento de las soluciones de ambos experimentos. Para la tabla 1, los valores de cercanía son  $d_1 = 40\text{cm}$  y  $d_2 = 20\text{cm}$ . Los valores de los sensores en las 2 primeras iteraciones son de un estado normal (45cm y 50cm) y se enciende la luz verde de sus Leds RGB. Sin embargo, en las siguientes 4 iteraciones, ambos sensores determinan un estado de precaución (valores de 30cm y 35cm), y encienden la luz azul de sus Leds RGB. Finalmente, en las últimas 4 iteraciones ambos sensores encienden la alarma ya que sus valores son 15cm y 12cm, y encienden la luz roja de sus leds RGB.

Para la tabla 2, la temperatura de ambos sensores son  $T_1 = 20^\circ$  y  $T_2 = 30^\circ$ , respectivamente. Como se observa, el sistema Arduino representa de forma correcta todas las situaciones: i) en los primeros tres tiempos, ambos sensores leen  $24^\circ$  y se encuentran en estado de precaución con la luz azul encendida de sus leds RGB; ii) en los siguientes 3 lecturas, lecturas 4 a 6, el sensor 1 (S1) lee  $12^\circ$  y el sensor 2 (S2) lee  $35^\circ$ , entonces S1 activa la luz verde en su led RGB mientras que S2 activa la luz roja de su led RGB y se activa o enciende la alarma; iii) en las últimas 4 iteraciones, S1 lee  $12^\circ$  y S2 lee  $21^\circ$  con lo cual S1 se mantiene en un estado normal mientras S2 está ahora en un estado de precaución. Con estos resultados, se valida el funcionamiento de la solución del 2do experimento para su implementación física.

Tabla 1: Funcionamiento de solución para experimento 1 (Adaptada de Lineros et al., 2018).

#	1er Sensor de Prox. S1	2doSensor de Prox. S2	Luz Verde S1	Luz Verde S2	Luz Azul S1	Luz Azul S2	Luz Roja S1	Luz Roja S2	Alarma
1	45cm	50cm	X	X					
2	45cm	50cm	X	X					
3	30cm	35cm			X	X			
4	30cm	35cm			X	X			
5	30cm	35cm			X	X			
6	30cm	35cm			X	X			
7	10cm	12cm					X	X	X
8	10cm	12cm					X	X	X
9	10cm	12cm					X	X	X
10	10cm	12cm					X	X	X

Tabla 2: Funcionamiento de solución para experimento 2 (Adaptada de Lineros et al., 2018).

#	1er Sensor de Temp. S1	2doSensor de Temp. S2	Luz Verde S1	Luz Verde S2	Luz Azul S1	Luz Azul S2	Luz Roja S1	Luz Roja S2	Alarma
1	$24^\circ$	$24^\circ$			X	X			
2	$24^\circ$	$24^\circ$			X	X			
3	$24^\circ$	$24^\circ$			X	X			
4	$12^\circ$	$35^\circ$	X					X	X
5	$12^\circ$	$35^\circ$	X					X	X
6	$12^\circ$	$35^\circ$	X					X	X
7	$12^\circ$	$21^\circ$	X			X			
8	$12^\circ$	$21^\circ$	X			X			
9	$12^\circ$	$21^\circ$	X			X			
10	$12^\circ$	$21^\circ$	X			X			

En el núcleo de ambos experimentos Arduino, existe un procesamiento de señales. Esto es esencial para el funcionamiento de cada sistema y para trabajar con los valores que se obtienen de los sensores de ultrasonido y de temperatura. Estos sensores presentan valores que varían entre 0v y 5v sin considerar valores intermedios para el sensor digital de proximidad o cercanía, y de 0 a 2,5v para el sensor de temperatura. El sensor de proximidad, por su relación con la lógica binaria, tal y como ilustra la figura 9, presenta dos únicos valores o estados: estado alto (1) y estado bajo (0). Los beneficios de las señales digitales son la poca pérdida de calidad y una viable reconstrucción con procesos algorítmicos de regeneración. Además, las señales digitales no son altamente afectadas por el ruido del ambiente (Antoniu, 2016). Dada la disponibilidad de Arduino y su aplicabilidad en proyectos para la adquisición de señales y control del ambiente (Ait-Ahmed et al., 2017), actualmente, los autores de este trabajo, profesores de las carreras ICI e Ingeniería Civil en Minas UVM, trabajan en un sistema Arduino para el análisis y detección de fenómenos críticos en centros mineros en la región de Valparaíso, Chile.

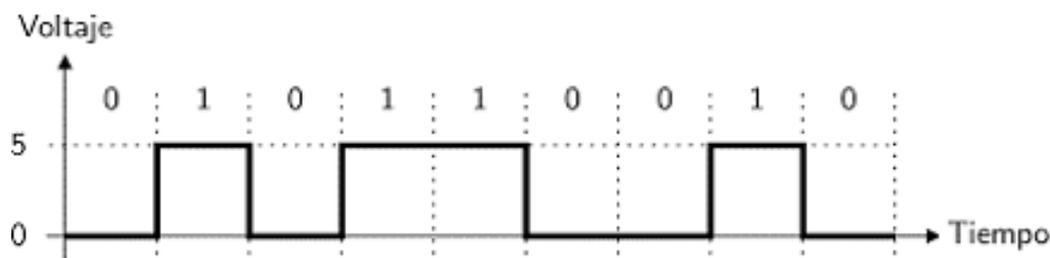


Fig. 9: Procesamiento de señales digitales (Adaptada de Lineros et al., 2018).

Es relevante destacar que esta experiencia es análoga a las de Brock et al. (2009), Martín-Ramos et al. (2017) y Tan et al. (2017) referente a la enseñanza de la programación, y de Byishimo y Garba (2016), y López-Rodríguez y Cuesta (2016) respecto a la enseñanza de electrónica básica para con estudiantes sin formación o con una mínima formación en estas áreas. Aun cuando la complejidad y dificultad de los experimentos diseñados no es muy alta, es destacable el dominio de los elementos de Arduino y tópicos asociados para la confección de informes estilo artículo de investigación mediante el formato de revistas CIT, así como la presentación formal de dichos experimentos y resultados prácticos de los mismos. Los resultados finales de los estudiantes fue una nota promedio de 5.8, con una nota mínima de 5.4 y máxima de 6.8.

Desde un punto de vista práctico, se puede argumentar que la primera competencia de este módulo (1. Conceptualiza los componentes fundamentales hardware y software de un sistema computacional moderno) se desarrolló en un alto nivel mediante los experimentos desarrollados, porque los estudiantes debieron adquirir los equipos hardware necesarios, implementar sus propias rutinas en un simulador para luego montarlas en el circuito final de Arduino, para luego probar su funcionamiento real. Respecto a la segunda competencia (2. Plantea soluciones prototipo concurrentes tanto hardware como software en una organización), justamente los experimentos desarrollados corresponden a soluciones prototipo, que requerían el uso de concurrente propio del marco de trabajo de programación de Arduino. Se debe destacar el alto impacto práctico y aceptación de los estudiantes de Tinkercad como plataforma de simulación para el diseño, implementación y prueba de soluciones Arduino.

Con un dominio mayor tanto de programación y tecnologías informáticas, el futuro con Arduino es prometedor con restricciones de costo y conocimiento cada vez menores. Las nuevas versiones de Arduino incorporan nuevas tecnologías; por ejemplo, Arduino MKR Vidor 4000 (Arduino MKR, 2018) con una matriz de puertas programables (del inglés FPGA, Field-Programmable Gate Array). Así, Arduino permite desarrollar soluciones +electrónicas con bajas barreras de costo y aprendizaje.

## CONCLUSIONES

Este trabajo demuestra que Arduino facilita el desarrollo de soluciones con una simbiosis de electrónica y computación para con estudiantes de una baja o nula formación previa en electrónica y computación. Este trabajo demuestra así, desde un punto de vista pedagógico, la viabilidad para el desarrollo de competencias tanto de programación (Brock et al., 2009; Martín-Ramos et al., 2017), diseño de sistemas (Kojmane y Aboutajeddine, 2016) y electrónica aplicada (Byishimo y Garba, 2016) para el desarrollo de nuevas soluciones (Martín-Ramos et al., 2016). Este trabajo detalla los resultados de una experiencia académica para destacar la facilidad de acceso y dominio de estudiantes del área de computación e informática con el uso de Arduino en el diseño e implementación de soluciones donde son sólo necesarias nociones básicas de programación y electrónica para estar habilitado en el desarrollo de soluciones Arduino.

**REFERENCIAS**

- Ait-Ahmed, W., M. Aggour y M. Naciri, Biogas Control: Methane Production Monitoring Using Arduino, doi: 10.5281/zenodo.1339948, International Journal of Biotechnology and Bioengineering, World Academy of Science, Engineering and Technology, 11 (2), Enero (2017)
- Antoniou, A., Digital Signal Processing, isbn: 0071846034, Mcgraw-Hill, 2da edición (2016)
- ArduinoChile, Plataforma Open Source para el desarrollo de prototipos electrónicos, (en línea), <https://www.arduino.cl>. Acceso: 19 de Enero (2019)
- ArduinoHome, Arduino – Home, (en línea), <https://www.arduino.cc>. Acceso: 19 de Enero (2019)
- ArduinoMKR, Getting Started with the Arduino MKR Vidor 4000, (en línea), <https://www.arduino.cc/en/Guide/MKRVidor4000>. Acceso: 20 de Enero (2019)
- Brock, J. D., R. F. Bruce y S. L. Reiser, Using Arduino for introductory programming courses, issn: 1937-4771, Journal of Computing Sciences in Colleges, vol. 25, no. 2, 129–130, Diciembre (2009)
- Byishimo, A. y A. A. Garba, Designing a farmer interface for smart irrigation in developing countries, doi: 10.1145/3001913.3006639, en Actas de 7<sup>th</sup> Annual Symposium on Computing for Development, ACM DEV'16, New York, NY, USA, Noviembre (2016)
- Hertzog, P. y A. J. Swart, Arduino — Enabling engineering students to obtain academic success in a design-based module, doi: 10.1109/EDUCON.2016.7474533, IEEE Global Engineering Education Conference (EDUCON), Abu Dhabi, United Arab Emirates, Abril (2016)
- Jones, D., Arduino: Simple and Effective Strategies to Learn Arduino Programming (Volume 3), isbn: 197577762X, USA: CreateSpace Independent Publishing Platform, 3era edición (2017)
- Kojmane, J. y A. Aboutajeddine, Strengthening engineering design skills of first-year university students under resources constraints, doi: 10.1177/0306419016641006, International Journal of Mechanical Engineering Education, vol. 44, no. 2, 148 – 164 (2016)
- López-Rodríguez, F. y F. Cuesta, Andruino-A1: Low-Cost Educational Mobile Robot Based on Android and Arduino, doi: 10.1007/s10846-015-0227-x, Journal of Intelligent & Robotic Systems, 81(1), 63 – 76, Enero (2016)
- Lineros, M., B. Bastías, F. Muñoz, K. Aravena, M. Figueroa, L. Rodríguez, B. Villegas, F. Hinojosa, B. Gutiérrez, A. Guerra, O. Bernales, C. Román, S. Collantes, C. Vidal y R. Villarroel, Electronics for everybody: student practical experiences using Arduino, XIX Congreso Chileno de TICS para la Educación, TICXED 2018, Universidad Andrés Bello, Santiago, Chile, Noviembre (2018)
- Margolis, M., Arduino Cookbook, isbn: 1449313876, O'Reilly Media, Inc. (2011)
- Merkouris, A. y K. Chorianopoulos, Introducing computer programming to children through robotic and wearable devices, doi: 10.1145/2818314.2818342, en Actas de Workshop in Primary and Secondary Computing Education, WiPSCE '15, 69 – 72, New York, USA (2015)
- Nussey, J., Arduino For Dummies, isbn: 1118446372, For Dummies, 1era edición (2013)
- Martín-Ramos, P., M. M. Lima da Silva, M. João Lopes y M. Ramos Silva, Student2student: Arduino project-based learning, doi: 10.1145/3012430.3012500, TEEM '16 Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, España (2016)
- Martín-Ramos, P., M. João Lopes, M. M. Lima da Silva, P. E. B. Gómez, P. Pereira, J. Domínguez y M. Ramos Silva, First exposure to Arduino through peer-coaching: Impact on students' attitudes towards programming, doi: 10.1016/j.chb.2017.12.011, Computers in Human Behavior, Volume 76, 51 – 58, Noviembre (2017)
- Shiffman, D., Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction, isbn: 9780123947925, Morgan Kaufmann Publishers Inc., 2da edición, San Francisco, CA, USA (2016)
- Tan, W. L., S. Venema y R. González, Using Arduino to Teach Programming to First-Year Computer Science Students, en Actas de International Conference on Educational Technologies, International Association for Development of the Information Society (IADIS), Sydney, Australia, 11 – 13, Diciembre (2017)
- Tinkercad, Circuits has arrived: The easiest way to tinker with circuits!, (en línea), <https://www.tinkercad.com/circuits>. Acceso: 20 de Enero (2019)
- Vidal, C. y V. Gatica Rojas, Diseño e implementación de un sistema electrocardiográfico digital, Revista Facultad de Ingeniería, no. 55, pp. 99–107, Septiembre (2010)
- Vidal C. y L. Pavesi Farriol, Desarrollo de un Sistema de Adquisición y Tratamiento de Señales Electrocardiográficas, doi: 10.4067/S0718-13372005000100005. Revista Facultad de Ingeniería - Universidad de Tarapacá, vol. 13, pp. 39 –46, 04 (2005)
- Vidal, C., A. Philominraj y C. del Rio, A DSP Practical Application: Working on ECG Signal, doi: 10.5772/25499, Applications of Digital Signal Processing, IntechOpen, Ed. Rijeka: IntechOpen, 153 – 168 (2011)
- Wiring, What will you do with the W?, (en línea), <http://wiring.org.co/>. Acceso: 19 de Enero (2019)