Guía de ejercicios de la unidad 1

1. Determinar el valor final de la siguiente variable en los casos como se escriben a continuación teniendo en cuenta la jerarquía de operaciones:

-a. int c = 2+3*4-2*3^2-1;

-b. int c = 3; int b = c*3+1/2+3^1*2;

-c += c+b;

Solución:

Caso A

a. int c = 2+3*4-2*3^2-1;

$C = 2 + 3 * 4 – 2 * 3^2 – 1$

$C = 2 +12 - 2 \times 9 – 1$

$C = 14 – 18 – 1$

$C = 14 – 19$

$C = -5$

Caso B

b. int c = 3;

int b = c*3+1/2+3^1*2;

$b = C * 3 + 1/2 + 3^1 * 2$ (½ en programación es igual a 0)

$b = 3 * 3 + 1/2 + 3 * 2$

$b = 9 + 0 +6$

$b = 15$

C+ = c +b

$C = 3 + (3 +15)$

$C = 3 + 18$

$C = 21$

2. Investigar qué otras placas de Arduino están en la página Arduino.cc y cuáles son las que se encuentran en búsquedas de tiendas Argentinas.

La familia de placas de Arduino se ha expandido enormemente desde el clásico modelo UNO. Dependiendo de si buscas en el sitio oficial (**Arduino.cc**) o en el mercado local de **Argentina**, encontrarás opciones muy distintas que van desde hardware de última generación hasta versiones económicas para estudiantes.

1. Placas en Arduino.cc (Modelos Oficiales 2026)

El catálogo oficial se divide actualmente en familias según su potencia y conectividad. Estas son las más destacadas:

- **Familia UNO (El Estándar):**

- **Arduino UNO R4 (Mínima y Wifi):** La evolución moderna con procesador de 32 bits, matriz de LED (en la versión Wifi) y conector USB-C.

- **Arduino UNO Q (Novedad 2026):** Una placa híbrida que combina un microprocesador Qualcomm (capaz de correr Linux Debian) con un microcontrolador STM32 para tareas en tiempo real e Inteligencia Artificial.

3.Diagramar el siguiente problema: "Un supermercado decide que nos encarguemos de crear un sistema para el cobro en sus tiendas y nos da el siguiente proceso relevado: Los clientes pueden pagar con 3 modalidades: -Efectivo -Crédito -Debito En el caso del efectivo y del débito, el descuento es del 10% y 5% respectivamente. Con crédito, se hace un recargo de 2.5%. En el caso de que se lleven más de 100 unidades, además, se lleva un 2.5% de descuento. Calcular el importe final"

Si tuviéramos que escribir los pasos lógicos, el orden sería el siguiente:

1. **Inicio:** Ingresar el Subtotal y la Cantidad.
2. **Validación de Unidades:** * ¿Cantidad > 100?
    - Si **Sí**: Aplicar descuento del 2.5% ($Subtotal = Subtotal \times 0.975$).
3. **Selección de Modalidad de Pago:**
    - **Efectivo:** Descuento del 10% (ImporteFinal = Subtotal \times 0.90$).
    - **Débito:** Descuento del 5% (ImporteFinal = Subtotal \times 0.95$).
    - **Crédito:** Recargo del 2.5% (ImporteFinal = Subtotal \times 1.025$).
4. **Resultado:** Mostrar el Importe Final

4.Determinar cuáles de las siguientes líneas tiran error y por qué:

```
int var1 = 5
float var2 = 3;
char var3 = 48;
void setup()
{
        int var4 = 6;
}
void loop()
{
        double var5 = var2*var3;
        long float var6 = var3*0,7 + 1;
        int var7 = var5 + var6;
        int var8 = var4 - var3;
}
```

 nt var1 = 5: **ERROR**. Falta el punto y coma (;) al final de la línea. En C++, todas las declaraciones de variables deben terminar con ;.

 long float var6 = var3*0,7 + 1;: **DOS ERRORES**.

 En Arduino, no existe el tipo long float. Para mayor precisión se usa double, aunque en la mayoría de placas de 8 bits (como el Uno), double y float son lo mismo.

 Decimal incorrecto: Los decimales se marcan con **punto** (0.7), no con coma (0,7). La coma se usa para separar argumentos.

int var8 = var4 - var3;: **ERROR**. La variable var4 fue declarada dentro de la función setup().

Las variables declaradas dentro de una función son **locales**. Esto significa que var4 "muere" cuando termina el setup. Al intentar usarla en el loop(), el compilador dirá que var4 no ha sido declarada en ese ámbito.

5. Hacer un programa que secuencia 5 leds distintos y prenda por cada bucle del loop a un led distinto (cuando uno está prendido el resto están apagados).

```
// Definimos los pines en un arreglo para facilitar el manejo
int leds[] = {2, 3, 4, 5, 6};
int indice = 0; // Esta variable recordará qué LED sigue
void setup() {
 // Configuramos los 5 pines como salida usando un ciclo for
 for (int i = 0; i < 5; i++) {
  pinMode(leds[i], OUTPUT);
 }
}
void loop() {
 // 1. Apagamos todos los LEDs primero
 for (int i = 0; i < 5; i++) {
  digitalWrite(leds[i], LOW);
 }
 // 2. Encendemos solo el LED que corresponde a este bucle
 digitalWrite(leds[indice], HIGH);
```

```
  // 3. Esperamos un tiempo para ver el efecto
  delay(500);
  // 4. Incrementamos el índice para el siguiente LED
  indice = indice + 1;
  // 5. Si ya pasamos el último LED (índice 4), volvemos al primero (0)
  if (indice > 4) {
    indice = 0;
  }
}
```

6.Hacer un programa que prenda 6 leds de dos en dos incluyendo en ese par al último que estuvo prendido (primero 1 y 2 - después 2 y 3 - después 4 y 5, etc.)

```
// Definimos los 6 pines en un arreglo
int leds[] = {2, 3, 4, 5, 6, 7};
int totalLeds = 6;

void setup() {
  // Configuramos todos los pines como salida
  for (int i = 0; i < totalLeds; i++) {
    pinMode(leds[i], OUTPUT);
  }
}
void loop() {
  // El ciclo recorre hasta el penúltimo LED (índice 4)
  // para que siempre haya un "siguiente" (i+1) disponible.
  for (int i = 0; i < (totalLeds - 1); i++) {

    // Apagamos todos antes de empezar el par
    apagarTodos();
```

```
    // Encendemos el par: el actual y el que sigue
    digitalWrite(leds[i], HIGH);
    digitalWrite(leds[i + 1], HIGH);

    delay(500); // Velocidad de la secuencia
  }
}
// Función auxiliar para limpiar el estado de los pines
void apagarTodos() {
  for (int i = 0; i < totalLeds; i++) {
    digitalWrite(leds[i], LOW);
  }
}
```