**Universidad Autónoma de Tamaulipas**

**Facultad de Ingeniería Tampico**

A black background with a black square

Description automatically generated with medium confidenceA red and grey logo

Description automatically generated

**ASIGNATURA**

**Diseño Electrónico Basado en Sistemas Embebidos**

8vo. Semestre – Grupo “X”

2025 -1

**TRABAJO**

**Desarrollo de Tareas e Investigaciones**

**UNIDAD**

**2 – PROGRAMACION BASICAA**

**Docente:** Dr. García Ruiz Alejandro H.

|  |  |
| --- | --- |
| **Integrante del Equipo** | **Nivel de Participación** |
| ORTIZ DORIA EFRAIN ALEJANDRO | 25 |
| SANDOVAL VAZQUEZ CRISTHIAN MICHEL | 25 |
| CRUZ BONIFACIO LUIS FERNANDO | 25 |
| SEGURA ALONSO ADRIAN | 25 |
| Total: | 100% |

# Índice

[Índice 1](#_Toc192449158)

[Diseño de Circuitos utilizando Algún Software como Fritzing o Tinkercad 2](#_Toc192449159)

[Ejemplos de Proyectos Reales o Académicos en los que se haya Utilizado Arduino 2](#_Toc192449160)

[Arduino: Fundamentos y Características 3](#_Toc192449161)

[Escritura y Lectura de Señales Digitales en Arduino 4](#_Toc192449162)

[Escritura y Lectura de Señales Analógicas en Arduino 4](#_Toc192449163)

[Instrucciones del Paquete Serial 4](#_Toc192449164)

[Arreglos en Arduino 4](#_Toc192449165)

[Preprocesamiento y Limpieza de Datos 4](#_Toc192449166)

[Análisis de Datos y Estadísticas 5](#_Toc192449167)

[Optimización y Modelado 5](#_Toc192449168)

# Diseño de Circuitos utilizando Algún Software como Fritzing o Tinkercad

El diseño de circuitos es una etapa fundamental en el desarrollo de proyectos electrónicos. Herramientas como Fritzing y Tinkercad permiten crear esquemas y simulaciones de circuitos antes de su implementación física.

**Fritzing**

Fritzing es una herramienta de código abierto que facilita la creación de diagramas esquemáticos, diseños de PCB y prototipos virtuales. Es especialmente útil para principiantes debido a su interfaz intuitiva. Algunas de sus principales características incluyen:

* Biblioteca extensa de componentes electrónicos.
* Modos de visualización: esquema, protoboard y PCB.
* Exportación de diseños en formatos compatibles con la fabricación de placas de circuito impreso (Gerber).
* Simulación básica de circuitos.

**Tinkercad**

Tinkercad, desarrollado por Autodesk, es una plataforma en línea que permite diseñar circuitos y simular su funcionamiento con componentes como Arduino, sensores y actuadores. Sus ventajas incluyen:

* Accesibilidad en la web sin necesidad de instalar software adicional.
* Simulación interactiva de circuitos en tiempo real.
* Integración con herramientas de diseño 3D.
* Compatibilidad con microcontroladores como Arduino y Raspberry Pi.

# Ejemplos de Proyectos Reales o Académicos en los que se haya Utilizado Arduino

Arduino es ampliamente utilizado en proyectos académicos y profesionales debido a su versatilidad y facilidad de uso. Algunos ejemplos incluyen:

**Sistemas de Riego Automático**

* Uso de sensores de humedad para determinar la cantidad de agua en el suelo.
* Activación de bombas de agua mediante actuadores según los niveles de humedad.
* Implementación de lógica de control con Arduino para optimizar el riego.

**Robótica Educativa**

* Creación de robots autónomos o controlados remotamente.
* Implementación de motores, sensores de ultrasonido y servomotores.
* Enseñanza de principios de programación y electrónica a estudiantes.

**Monitoreo Ambiental**

* Medición de temperatura, humedad y calidad del aire con sensores como DHT11 o MQ135.
* Envío de datos a plataformas en la nube para análisis remoto.
* Desarrollo de aplicaciones para visualizar datos en tiempo real.

**Domótica**

* Control de luces, persianas y electrodomésticos mediante sensores y actuadores.
* Uso de protocolos de comunicación como Wi-Fi o Bluetooth para automatización.
* Integración con asistentes de voz como Alexa o Google Assistant.

# Arduino: Fundamentos y Características

Arduino se programa en un lenguaje basado en C/C++. Sus fundamentos incluyen:

**Variables**

Espacios de memoria para almacenar datos. Ejemplos:

* int contador = 10; (Entero)
* float temperatura = 23.5; (Decimal)
* char letra = 'A'; (Carácter)
* bool estado = true; (Booleano)

**Tipos de Datos**

Define el tipo de valor que puede almacenar una variable, tales como:

* int (enteros)
* float (decimales)
* char (caracteres)
* bool (booleanos)

**Estructuras de Control**

Permiten controlar el flujo del programa:

* Condiciones: if-else
* Bucles: for, while
* Estructuras de selección: switch-case

**Funciones**

Bloques de código reutilizables que realizan tareas específicas. Ejemplo:

void encenderLED() {

digitalWrite(13, HIGH);

}

# Escritura y Lectura de Señales Digitales en Arduino

Las señales digitales solo tienen dos estados: HIGH (5V) o LOW (0V). Para trabajar con ellas:

**Escritura**

digitalWrite(pin, HIGH); // Envía un valor HIGH o LOW a un pin

**Lectura**

int estado = digitalRead(pin); // Lee el estado de un pin digital

Ejemplo: Encender un LED conectado a un pin digital.

# Escritura y Lectura de Señales Analógicas en Arduino

Las señales analógicas varían en un rango de valores.

**Lectura**

int valor = analogRead(pin); // Lee un valor entre 0 y 1023

**Escritura**

analogWrite(pin, valor); // Envía un valor PWM (0-255)

Ejemplo: Controlar el brillo de un LED con un potenciómetro.

# Instrucciones del Paquete Serial

El puerto serial permite comunicación entre Arduino y una computadora. Algunas funciones clave:

Serial.begin(9600); // Iniciar la comunicación a 9600 baudios

Serial.println("Hola, mundo!"); // Enviar datos al puerto serial

String mensaje = Serial.readString(); // Leer una cadena de caracteres

# Arreglos en Arduino

Los arreglos almacenan múltiples valores del mismo tipo:

int valores[5] = {1, 2, 3, 4, 5};

Acceso a elementos:

int primer\_valor = valores[0];

# Preprocesamiento y Limpieza de Datos

El preprocesamiento prepara los datos para su análisis, incluyendo:

* **Normalización**: Ajustar valores a un rango común.
* **Filtrado**: Eliminar ruido o datos no deseados.
* **Corrección de valores atípicos**.

# Análisis de Datos y Estadísticas

**Interpolación Lineal**

Método para estimar valores desconocidos entre dos puntos conocidos:

y=y1+(x−x1)×(y2−y1)(x2−x1)y = y\_1 + (x - x\_1) \times \frac{(y\_2 - y\_1)}{(x\_2 - x\_1)}

**Graficación y Outliers**

Los BoxPlots visualizan la distribución de datos mediante cuartiles. Se identifican outliers mediante el rango intercuartílico (IQR):

Outlier<Q1−1.5×IQR o Outlier>Q3+1.5×IQROutlier < Q\_1 - 1.5 \times IQR \,\text{o}\, Outlier > Q\_3 + 1.5 \times IQR

# Optimización y Modelado

**Métricas de Error**

* **MAE**: Error absoluto medio.
* **MSE**: Error cuadrático medio.
* **RMSE**: Raíz del error cuadrático medio.

**Suavizamiento Exponencial**

Técnica para suavizar series de tiempo:

St=αXt+(1−α)St−1S\_t = \alpha X\_t + (1 - \alpha) S\_{t-1}