



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®

**INSTITUTO TECNOLÓGICO DE
DURANGO**

**INSTITUTO TECNOLÓGICO DEL
VALLE DEL GUADIANA**

**DIVISIÓN DE ESTUDIOS DE POSGRADO
E INVESTIGACIÓN**



**Implementación de modelo grey recursivo en sistema embebido
para predicción de variables meteorológicas.**

TESIS

Que como parte de los requisitos para obtener el grado de

Maestría en Ingeniería

Presenta:

Ing. Ricardo Zamudio Carbajal

Director de tesis:

M.C. José Antonio Martínez Rivera

Durango, Dgo., México, noviembre, 2022





Implementación de modelo grey recursivo en sistema embebido para predicción de variables meteorológicas

TESIS

Que como parte de los requisitos para obtener el grado de

Maestría en Ingeniería

Presenta:

Ing. Ricardo Zamudio Carbajal

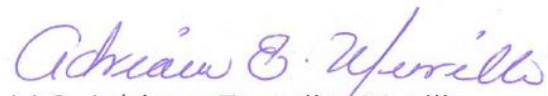
Dirigido Por

M.C. José Antonio Martínez Rivera

COMITÉ TUTORIAL

| | |
|--|---|
| <p><u>M.C. José Antonio Martínez Rivera</u> Director</p> |  <hr/> <p>Firma</p> |
| <p><u>Dr. Francisco Javier Godínez García</u> Codirector</p> |  <hr/> <p>Firma</p> |
| <p><u>M.I. Eduardo Gamero Inda</u> Asesor</p> |  <hr/> <p>Firma</p> |


M. C. Norma Alicia García Vidaña
 Nombre y firma
Coordinador de la Maestría en Ingeniería


M.C. Adriana Erendira Murillo
 Nombre y firma
Jefe(a) de la División de Estudios de Posgrado e Investigación

Durango, Dgo. México, noviembre, 2022



Victoria de Durango, Dgo., a 04 / Noviembre / 2022.

 DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
 DEPI / C / 455 / 22.

ASUNTO: Autorización de Tema de Tesis de Maestría.

C. RICARDO ZAMUDIO CARBAJAL
No. DE CONTROL G15041112
P R E S E N T E .

Con base en el Reglamento en vigor y teniendo en cuenta el dictamen emitido por el Jurado que le fue asignado, se le autoriza a desarrollar el tema de tesis para obtener el **Grado de Maestro en Ingeniería** cuyo título es:

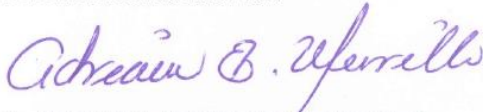
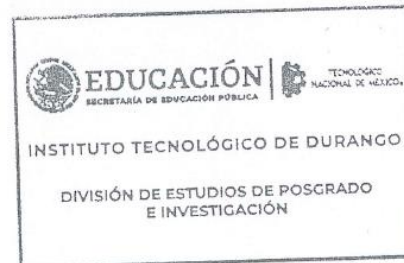
“Implementación de modelo grey discreto en sistema embebido para predicción de variables meteorológicas”

CONTENIDO:

| | |
|--------------|--------------------------------|
| | RESUMEN |
| CAPÍTULO I | INTRODUCCIÓN |
| CAPÍTULO II | MARCO TEÓRICO |
| CAPÍTULO III | DESARROLLO Y RESULTADOS |
| CAPÍTULO IV | CONCLUSIONES Y RECOMENDACIONES |
| | REFERENCIAS |
| | ANEXOS |

ATENTAMENTE.

Excelencia en Educación Tecnológica®
 "La Técnica al Servicio de la Patria"


C. ADRIANA ERÉNDIRA MURILLO
JEFA DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN


AEM'ammc.



Fecha de Emisión: 09/11/2022
 Fecha de Vencimiento: 09/11/2022
 Fecha de Emisión: 09/11/2022
 Fecha de Vencimiento: 09/11/2022
 BSCC 657

Av. Felipe Pescador #1830 Ote. Col. Nueva Vizcaya C.P. 24080 Durango, Durango.
 Tel. (618) 8290900 e-mail: dir_itdurango@tecnm.mx tecnm.mx | itdurango.edu.mx



2022 Flores
 Año de Magón
 PRECURSOR DE LA REVOLUCIÓN MEXICANA



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Durango
División de Estudios de Posgrado e Investigación

Victoria de Durango, Dgo., a **04 / Noviembre / 2022.**

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
DEPI / C / 456 / 22.

ASUNTO: Autorización de Impresión de Tesis de Maestría.

C. RICARDO ZAMUDIO CARBAJAL
No. DE CONTROL G15041112
PRESENTE.

De acuerdo al reglamento en vigor y tomando en cuenta el dictamen emitido por el jurado que le fue asignado para la revisión de su trabajo de tesis para obtener el **Grado de Maestro en Ingeniería**, esta División de Estudios de Posgrado e Investigación le autoriza la impresión del mismo, cuyo título es:

"Implementación de modelo grey discreto en sistema embebido para predicción de variables meteorológicas"

Sin otro particular de momento, quedo de Usted.

ATENTAMENTE.

Excelencia en Educación Tecnológica®
"La Técnica al Servicio de la Patria"

C. ADRIANA ERÉNDIRA MURILLO
JEFA DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN



AEM'ammc.



Fecha de Inicio: 2015.11.25
Fecha de Cierre Ciclo: 2016.11.25
Fecha de Cierre de Estudios: 2016.11.25
Fecha de Graduación: 2016.11.25
ASGC 937

Av. Felipe Pescador #1830 Ote. Col. Nueva Vizcaya C.P.34080 Durango, Durango.
Tel. (618) 8290900 e-mail: dir_itdurango@tecnm.mx tecnm.mx | itdurango.edu.mx



2022 Flores
Año de **Magón**
PRECURSOR DE LA REVOLUCIÓN MEXICANA

Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico otorgado para cursar los estudios de posgrado en forma de una beca, de acuerdo con el número 1085581. Al M.C. José Antonio Martínez, principal tutor que me asesoró durante el proceso, al Dr. Francisco Javier Godínez García por poner a disposición su amplio conocimiento y experiencia en el área de ingeniería, al Dr. Rubén Guerrero Rivera y al M.C. Eduardo Gamero Inda por las cátedras ofrecidas dentro del programa del posgrado.

Agradezco a mi familia por el apoyo brindado y la motivación para seguir con los estudios de posgrado, a mi madre Gabriela por siempre motivarme a seguir adelante, a la Ing. Nancy Vázquez Morales por su apoyo y valioso conocimiento en las áreas de tecnología.

Índice

| | |
|---|----|
| Capítulo I Introducción. | 1 |
| 1.1 Antecedentes | 1 |
| 1.2 Justificación. | 4 |
| 1.3 Objetivos. | 5 |
| 1.3.1 Objetivo general..... | 5 |
| 1.3.2 Objetivos específicos. | 5 |
| 1.4 Delimitación | 5 |
| 1.4.1 Alcances. | 5 |
| 1.4.2 Limitaciones. | 6 |
| Capítulo II Marco Teórico | 7 |
| 2.1 Teoría de sistemas grey..... | 7 |
| 2.1.1 Números grises..... | 8 |
| 2.1.2 Blanqueamiento de un número gris | 9 |
| 2.1.3 Operadores de secuencia..... | 10 |
| 2.1.4 Operadores acumulativos y operadores acumulativos inversos | 11 |
| 2.1.5 Operador serie Z..... | 12 |
| 2.2 Regresión por mínimos cuadrados | 13 |
| 2.2.1 Regresión lineal | 13 |
| 2.2.2 Criterio de mejor ajuste..... | 14 |
| 2.2.3 Ajuste de una línea recta por mínimos cuadrados..... | 14 |
| 2.2.4 Regresión polinomial | 16 |
| 2.2.5 Formulación de una matriz para regresión por mínimos cuadrados lineales. | 17 |

| | |
|--|----|
| 2.3 Modelos Grey..... | 19 |
| 2.3.1 Modelo Grey (1, 1)..... | 19 |
| 2.3.2 Modelo Grey multivariable (1, n)..... | 21 |
| 2.3.3 Modelo Grey multivariable con integral de convolución GMC (1, n) | 23 |
| 2.3.4 Modelo Grey discreto recursivo multivariable RDGM (1, n)..... | 25 |
| 2.4 Instrumentación | 29 |
| 2.4.1. SHT10 sensor humedad relativa y temperatura | 29 |
| 2.4.2 Sensor de luminosidad TSL2561 | 32 |
| 2.5 Procesamiento | 34 |
| 2.5.1 Microcontrolador ATmega4809..... | 34 |
| 2.6 Comunicaciones | 35 |
| 2.6.1 Protocolo LoRa | 35 |
| 2.6.2 Modulación por radio LoRa..... | 35 |
| 2.6.3 Link Budget..... | 36 |
| 2.6.4 Módulo transceptor LoRa E32-915D EBYTE..... | 37 |
| 2.6.5 Tarjeta de desarrollo SIM808 GSM/GPRS/GPS..... | 40 |
| 2.6.6 Antenas de comunicación..... | 43 |
| 2.7 Herramientas de software | 45 |
| 2.7.1 Editor de código..... | 45 |
| 2.7.2 PlatformIO..... | 46 |
| 2.7.3 Compilador AVR-GCC | 47 |
| Capítulo III Resultados y discusión | 48 |
| 3.1 Sistema IoT..... | 48 |
| 3.1.1 Nodo sensor | 49 |
| 3.1.2 Puerta de enlace..... | 52 |

| | |
|---|----|
| 3.1.3 Servidor IoT | 53 |
| 3.1.4 Implementación física del sistema | 54 |
| 3.1.5 Aplicación y funcionamiento de plataforma IoT | 57 |
| 3.2 Desarrollo de algoritmos predictivos | 59 |
| 3.2.1 Desarrollo de algoritmo de operadores Grey | 59 |
| 3.2.2 Operadores de matrices | 66 |
| 3.2.3 Algoritmo en código C del GM (1,1)..... | 67 |
| 3.2.4 Algoritmo en código C del RDGM (1, 3) | 68 |
| 3.3 Experimentos con datos reales | 70 |
| 3.3.1 Diseño del experimento | 71 |
| 3.3.2 Experimento 1..... | 75 |
| 3.3.3 Experimento 2..... | 78 |
| 3.3.4 Experimento 3..... | 81 |
| 3.3.5 Experimento 4..... | 83 |
| Capítulo IV Conclusiones y recomendaciones | 86 |
| Referencias | 88 |
| Anexos | 92 |
| Publicaciones y participaciones de congresos..... | 92 |

Índice de Figuras

| | |
|--|----|
| Figura 1 Sensor de Temperatura SHT10 | 31 |
| Figura 2 Diagrama sensor TSL2561 | 32 |
| Figura 3 Curva de reacción de respuesta sensor TSL2561 | 33 |
| Figura 4 Microcontrolador ATmega4809 | 34 |
| Figura 5 Alcance de varios protocolos inalámbricos. | 36 |
| Figura 6 Módulo Transceptor LoRa E32-915, | 37 |
| Figura 7 Diagrama de pines módulo E32.915 | 38 |
| Figura 8 Ejemplo de conexión módulo E32-915..... | 39 |
| Figura 9 Tarjeta de desarrollo modem SIM808 | 41 |
| Figura 10 Esquema de funcionamiento modem SIM808..... | 42 |
| Figura 11 Antena del transceptor LoRa..... | 44 |
| Figura 12 Software Visual Studio Code..... | 45 |
| Figura 13 Extensión PlatformIO, Fuente: | 46 |
| Figura 14 Esquema generar del sistema..... | 49 |
| Figura 15 Esquema nodo sensor | 50 |
| Figura 16 Diagrama de flujo nodo sensor | 51 |
| Figura 17 Esquema puerta de enlace | 53 |
| Figura 18 Implementación nodo sensor. | 54 |
| Figura 19 Nodo sensor desplegado | 55 |
| Figura 20 Sensor de luminosidad..... | 56 |
| Figura 21 Implementación puerta de enlace. | 56 |
| Figura 22 Puerta de enlace instalada..... | 57 |
| Figura 23 Despliegue de datos en plataforma IoT | 58 |
| Figura 24 Exportación de datos | 59 |
| Figura 25 Función operador AGO. | 61 |
| Figura 26 Monitor serie función 1-AGO..... | 62 |
| Figura 27 Función serie Z. | 63 |
| Figura 28 Monitor serie función serie Z..... | 64 |

| | |
|---|----|
| Figura 29 Función operador IAGO | 65 |
| Figura 30 Monitor serie operador IAGO | 65 |
| Figura 31 Diagrama de flujo del algoritmo RDGM (1, 3). | 69 |
| Figura 32 Ejemplo de datos exportados en extensión .csv | 70 |
| Figura 33 Tarjeta de desarrollo ATmega480..... | 72 |
| Figura 34 PCB Tarjeta de desarrollo ATmega4809. | 72 |
| Figura 35 Declaración de datos de entrada del algoritmo..... | 73 |
| Figura 36 Llamado de la función del modelo RDGM (1, 3). | 74 |
| Figura 37 Banco de pruebas del sistema embebido. | 75 |
| Figura 38 Resultados del experimento uno..... | 77 |
| Figura 39 Tiempo de ejecución del experimento uno..... | 78 |
| Figura 40 Resultados del experimento dos. | 79 |
| Figura 41 Tiempo de ejecución del experimento dos..... | 80 |
| Figura 42 Resultados del experimento tres..... | 82 |
| Figura 43 Tiempo de ejecución del experimento tres..... | 83 |
| Figura 44 Resultados del experimento cuatro..... | 84 |
| Figura 45 Tiempo de ejecución del experimento cuatro..... | 85 |

Índice de Tablas

| | |
|--|----|
| Tabla 1 Descripción de pines de conexión del módulo | 39 |
| Tabla 2 Descripción de modos de operación | 40 |
| Tabla 3 Antena para módulos LoRa TX915-XPL-100 | 44 |
| Tabla 4 Datos del día 03 de mayo del 2022..... | 75 |
| Tabla 5 Datos del día 02 de mayo del 2022..... | 76 |
| Tabla 6 Pronóstico del experimento uno | 77 |
| Tabla 7 Datos del día 01 de mayo del 2022..... | 78 |
| Tabla 8 Datos del día 30 de abril del 2022..... | 79 |
| Tabla 9 Pronóstico del experimento dos | 80 |
| Tabla 10 Datos del día 02 de mayo del 2022..... | 81 |
| Tabla 11 Datos del día 01 de mayo del 2022..... | 81 |
| Tabla 12 Pronóstico del experimento tres | 82 |
| Tabla 13 Datos de la madrugada del 30 de abril del 2022..... | 83 |
| Tabla 14 Datos de la madrugada del 29 de abril del 2022..... | 84 |
| Tabla 15 Pronóstico del experimento cuatro | 85 |

Resumen

La producción de alimentos bajo invernadero podría ser una actividad esencial hacia la futura estabilidad de nuestra sociedad, por lo que se evidencia la necesidad de mejores y más eficientes procesos de producción de alimentos, para lo cual la tecnificación de los procesos locales de producción de alimentos apunta a ser beneficiosa. La temperatura es una variable fundamental para monitorear dentro de un invernadero, conocer el estado real y el pronóstico de esta variable puede tener un impacto positivo en los procesos de toma de decisiones por parte del agricultor de invernadero para garantizar el buen estado de los cultivos. Para generar dicho pronóstico, la teoría de los sistemas grises ha demostrado ofrecer pronósticos aceptables con poca información de entrada y bajo sistemas inciertos en otras aplicaciones a costos computacionales asequibles, se presenta una metodología para ejecutar el modelo gris recursivo multivariable para obtener un pronóstico de la próxima hora de la temperatura interna del invernadero, combinando datos del día actual con las tendencias del día pasado para generar la solución del modelo. Las variables de entrada para el algoritmo son los valores de temperatura interna, humedad relativa y luminosidad. El algoritmo se ejecuta en un microcontrolador de 8 bits de bajo costo para probar su portabilidad. Los resultados muestran que el algoritmo ofrece un pronóstico de temperatura interna aceptable con tiempos de ejecución bajos utilizando datos de una estación de monitoreo IoT desarrollada para esta investigación e instalada dentro de un invernadero real en una planta de producción en la ciudad de Durango, México.

Capítulo I Introducción.

1.1 Antecedentes

La teoría de sistemas grey permite el pronóstico de la producción agrícola aplicado en la predicción de la producción de arroz, evidenciando que el método de pronóstico grey GM (1,1) genera una predicción con un error relativo absoluto de 5.74% o del 94.26% en la precisión de los pronósticos en comparación con el valor real obtenido. (Muqtadir, Suryono, & Vicensius Gunawan, 2016)

En (Karimi Khajehghiasi & Alinezhad, 2019) se compara el funcionamiento de un modelo grey GM y un modelo grey mejorado denominado de metabolismo, aplicado en la predicción de la temperatura máxima media en el mes de agosto en la provincia de Qazvin en Irán. Los resultados obtenidos muestran que mediante la aplicación del modelo GM mejorado llamado de metabolismo, cuando el tamaño de los datos es adecuado, el modelo puede no solo minimizar el error de evaluación, mejorando la precisión de la predicción de la tendencia de las fluctuaciones de temperatura y los cambios de la temperatura máxima.

El pronóstico de la producción agrícola, forestal, ganadera y pesquera en la república popular china se obtuvo mediante el empleo de un modelo grey estacional (GSM). Los resultados obtenidos indican la baja tasa de error porcentual absoluto medio del ajuste en los datos arrojados por el GSM, en comparación con otros métodos de predicción como el modelo Holt-Winters o el modelo SARIMA. El GSM se aplica en la predicción de otras series temporales tal como los valores de producción agrícola mensual y los precios de los productos agrícolas. (Chen, Nu, & Wu, 2020).

En (Darvishi Salookolaei, Liu, & Babaei, 2017), se estiman los factores agro climáticos mediante modelos grey, comparando los resultados obtenidos en los modelos GM(1,1), DGM(2,1) y el modelo Verhulst se concluye que la precisión del modelo de predicción GM(1,1) fue la mejor dentro de los tres evaluados, al dar como resultado

un nivel de error aceptable en la estimación de la cantidad de lluvia en el árido país de Irán.

En (Zeng, Luo, Liu, Bai, & Li, 2016) se propone una alternativa para mejorar el ajuste en la implementación de modelos grises multivariados GM (1, n) con la introducción de un término de corrección lineal y un término de cantidad de acción gris. El modelo obtenido tiene un proceso y una estructura más estable, que resuelve algunas de las carencias del GM (1, N).

La aplicación de la integral de convolución a los modelos grises multivariados permite una mejora en el ajuste de la predicción, en (Mishra, Chandra Sahoo, & Narayan Senapati, 2016) se aplica un modelo gris multivariable con la integral de convolución GMC (1, n) para la predicción de energía solar generada (kWh) en una granja fotovoltaica, esta variable es directamente afectada por las condiciones climatológicas del área geográfica donde se ubica el centro de generación. El pronóstico realizado con el modelo GMC (1, n) proporcionó una media del error absoluto a priori para el intervalo analizado de 1.7 % en comparativa con una evaluación de los mismos datos utilizando un modelo sin convolución GM(1, 3), que arrojó un valor de 4.19%.

Para los casos en los que la implementación de un modelo gris multivariable con integral de convolución GMC (1, n) no cumpla con una predicción adecuada (Ma & Liu, The GMC(1, n) model with optimized parameters and its application, 2017) propone un método para optimizar los parámetros del GMC (1, n), obteniendo considerables mejoras para el pronóstico de sistemas dinámicos lineales en comparación con el modelo original.

La aplicación de un modelo gris discreto DGM (1,1) combinado con un modelo lineal permite modelar la temperatura y humedad relativa interior en un invernadero, los resultados obtenidos en métricas de error relativo porcentual fueron de 2.94% para la temperatura y de 6.11% para la humedad relativa. (Yin, 2014)

En (Ma & Liu, 2015) se propone un modelo recursivo discreto RDGM (1, n) que evidencia un menor grado de error en comparación con el modelo GMC(1, N), aplicado en el pronóstico de la resistencia a la torsión de un material metálico, el valor bruto de

la producción industrial basado en indicadores económicos y el consumo per cápita de la república popular china basado en indicadores económicos.

El monitoreo en tiempo real de las condiciones meteorológicas de cultivos en invernaderos permite la toma de decisiones por parte del agricultor. Las nuevas tendencias tecnológicas como la computación en la nube y el IoT permiten la recopilación de información proporcionada por sensores de humedad relativa y temperatura con el fin de obtener una predicción del microclima de un invernadero (Oo & Phyu, 2019)

El protocolo de comunicación LoRa (Long Range), se considera una opción adecuada para el desarrollo de soluciones en sistemas de comunicación por radiofrecuencia para el envío de mensajes cortos, la ventaja principal de estos dispositivos es su alta eficiencia y la optimización en el consumo de potencia (Augustin et al., 2016).

Los parámetros de configuración de los módulos LoRa como, el tiempo de muestreo, la cantidad de bits de la cadena contenedora del mensaje a enviar, el factor de propagación y la tasa de codificación son parámetros que se pueden configurar para mejorar el rendimiento de las baterías de litio asociadas a la fuente de energía de los dispositivos LoRa. (Bouguera, et al., 2018)

En (Leelavinodhan et al., 2021) se muestra el desarrollo de un sistema embebido para el monitoreo de variables climatológicas, cuyo diseño incorpora módulos de transmisión con protocolo LoRa, habilitando la posibilidad de desarrollar soluciones con un consumo de corriente bajo incluso en estados de transmisión y recepción de datos a través del módulo LoRa.

La implementación de un canal de comunicación punto a punto por radiofrecuencia, de largo alcance y con desempeño confiable es viable a través de módulos transceptores basados en el protocolo LoRa® de Semtech®, siempre y cuando se tomen en cuenta las variaciones en la potencia de la señal provocadas por variaciones climáticas. (Ameloot et al., 2021)

El desarrollo de una solución a mediano plazo bajo el esquema de nodo sensor, puerta de enlace y servicio de computación en la nube se puede aplicar con hardware de bajo

costo en invernaderos y cultivos para el monitoreo de variables climáticas en tiempo real, obteniendo autonomía energética mediante celdas fotovoltaicas. (Codeluppi, et al., 2020)

Los módulos GSM habilitan la conexión a la red celular, son una alternativa para la creación de puertas de enlace en zonas con baja conectividad, la incorporación de un módulo GSM en el diseño de un sistema embebido dedicado a la recolección y envío de datos meteorológicos faculta al dispositivo para la comunicación de los datos en tiempo real. (Parvez et al., 2016)

1.2 Justificación.

La presente investigación se enfoca en el estudio e implementación de métodos avanzados en predicción de sistemas inciertos, aplicados a variables climáticas y otras variables trascendentes que ofrecen valor a los productores agrícolas, ya que debido a los recientes cambios medioambientales cada día son más frecuentes las condiciones climáticas adversas, afectando una de las principales actividades primarias del estado de Durango, actividad que en 2020 generó una producción agrícola con valor de 6,220 millones de pesos y contabilizó una cosecha de 7.7 millones de toneladas de productos agrícolas (SIAP, 2020) El resultado de esta investigación permitirá a los productores agrícolas contar con una herramienta para la supervisión local de las variables climáticas, permitiendo así, la toma de decisiones para el cuidado y desarrollo de los cultivos. Conjuntamente contribuirá a la implementación del método predictivo en un sistema embebido, potencial punto de partida para futuras investigaciones.

1.3 Objetivos.

1.3.1 Objetivo general.

Diseñar un sistema embebido que establezca técnicas predictivas basadas en teoría de sistemas grey de variables meteorológicas de interés para el sector de la agricultura protegida.

1.3.2 Objetivos específicos.

- Diseñar sistema de adquisición de datos para variables agrícolas con un tiempo de muestreo de 30 minutos.
- Filtrar y procesar información adquirida.
- Diseñar el algoritmo predictivo basado en teoría de sistemas grey en lenguaje de alto nivel considerando un horizonte de tiempo finito.
- Validar algoritmo predictivo mediante métricas de errores cuadráticos
- Implementar el algoritmo predictivo en sistema embebido
- Validar el funcionamiento del sistema embebido

1.4 Delimitación

1.4.1 Alcances.

Se desarrollará software y hardware alrededor de un dispositivo embebido profesional, designado a la recopilación y procesamiento de datos de variables climáticas notables específicamente para la agricultura protegida. Se determinará la viabilidad del uso de la teoría de sistemas grey en la predicción

de variables climáticas relevantes en la agricultura protegida según sea la tasa de error absoluto del ajuste en los datos.

1.4.2 Limitaciones.

Las condiciones ambientales y la ausencia de conectividad en las zonas agrícolas presentan un reto para el funcionamiento autónomo del dispositivo embebido a mediano y largo plazo, a su vez existe una falta de estudios previos en el área de investigación y limitada información de la implementación en sistemas embebidos de algoritmos de la teoría de sistemas grey. El desempeño del algoritmo de predicción basado en la teoría de sistemas grey debe cumplir con un criterio de error cuadrático medio mínimo, en caso de no optimizarse el algoritmo lo suficiente para brindar información de utilidad en beneficio de agricultores se estaría ante una considerable limitación.

Capítulo II Marco Teórico

2.1 Teoría de sistemas grey

Como se menciona en (Liu, Forrest, & Yang, 2012) la teoría de sistemas grey (TSG), establecida por Julong Deng en 1982, forma parte de una familia de soluciones y teorías modernas enfocadas a resolver sistemas con un alto índice de incertidumbre y ruido, dicho de otra manera, sistemas con falta de información, ya sea en sus parámetros, en la estructura del sistema, en las relaciones o en el comportamiento general del sistema, un segundo elemento que caracteriza un sistema con incertidumbre es la inexactitud que presentan sus datos.

La inclusión de la palabra grey en el nombre de la teoría, se debe a la caracterización de la información mediante colores, en este contexto se denomina negro para indicar información desconocida, blanco para representar la información completamente conocida y gris para representar la información parcialmente conocida y desconocida. En consecuencia, los sistemas con información completamente conocida se considerarán blancos, aquellos sistemas con información completamente desconocida negros y los sistemas con información parcialmente conocida y desconocida se representarán mediante el color gris.

La TSG plantea el análisis de los datos para que a partir de una muestra pequeña sea posible obtener una aproximación real de diversos objetivos como son el modelado, análisis, sistemas de monitoreo y de control, de esta manera el objetivo de TSG es obtener el mayor valor posible de los datos que sí son conocidos en un sistema gris y por consiguiente poder describir y entender los fenómenos que afectan el entorno del que proviene la información.

La teoría de sistemas grey se ha mantenido en desarrollo desde su primera aparición en 1982, diversas aportaciones han surgido al pasar de los años, por ende la teoría se ha diversificado según sea el objetivo buscado, dentro de las diversas aportaciones

dentro de la teoría destaca el desarrollo de los modelos grey como lo son: Modelos de análisis de incidencia gris (Grey incidence model), modelos de evaluación de agrupamiento gris (Grey cluster model), familia de modelos GM y modelos de predicción gris (Grey forecasting model) y el modelo de decisión gris (Grey decision model). (Liu, Yang, & Forrest, 2017)

2.1.1 Números grises

Un sistema gris es descrito con números grises, secuencias grises, ecuaciones grises o matrices, los números grises son elementos fundamentales para el desarrollo de la teoría, sus valores exactos son desconocidos. Un número gris se define como un número intermedio que toma posibles valores dentro de un intervalo o un conjunto general de números. Generalmente los números grises son representados con el símbolo “ \otimes ”. Existen diferentes tipos de números grises:

- ❖ Números grises con solo un límite inferior, este tipo de números \otimes son representados como $\otimes \in [a, \infty)$, o $\otimes (a)$ en donde a representa al límite inferior conocido del número gris.
- ❖ Números grises con solo un límite superior, es escrito como $\otimes \in (-\infty, \bar{a}]$, o $\otimes (\bar{a})$, donde \bar{a} representa el límite superior del número gris.
- ❖ Intervalos de números grises, se representa como $\otimes \in [\underline{a}, \bar{a}]$, estos tipos de números grises contienen un límite inferior y superior.
- ❖ Números grises discretos y continuos, esta clase de números grises toman solo un número finito o un número contable de

valores potenciales y es conocido como discreto. En cambio, si el número gris puede potencialmente tomar cualquier valor en un intervalo, es conocido como número gris continuo.

- ❖ Números negros o números grises, un número negro es el que puede tomar cualquier valor desde menos infinito hasta más infinito, representado por $\otimes \in (-\infty, +\infty)$, un número blanco puede tomar un valor dentro de un intervalo conocido representado como $\otimes \in [\underline{a}, \bar{a}]$.

(Liu, Yang, & Forrest, Grey Data Analysis, 2017, págs. 29-30)

2.1.2 Blanqueamiento de un número gris

Cuando el valor de un número gris oscila alrededor de un valor constante, el blanqueamiento de este tipo de números es relativamente sencillo, se puede simplemente utilizar el valor constante como su blanqueamiento, un número gris que oscila alrededor de a puede ser escrito como $\otimes(a) = a + \delta_a$ en donde δ_a es la oscilación. En este caso el valor de blanqueamiento es $\tilde{\otimes}(a) = a$.

Para un intervalo general de número gris $\otimes \in [a, b]$, se puede tomar su valor de blanqueamiento como $\tilde{\otimes} = \alpha a + (1 - \alpha)b, \alpha \in [0,1]$

Aquí α es llamado el coeficiente de posición de un intervalo de número gris $\otimes \in [a, b]$.

Un blanqueamiento por promedio sucede cuando $\alpha = \frac{1}{2}$. Cuando la distribución de un intervalo de número gris es desconocida, se utiliza comúnmente el blanqueamiento por promedio.

Cuando el valor de la información de un número gris es conocido hasta cierto punto, se puede utilizar una función de probabilidad que describe la posibilidad que tiene un número gris de tomar ciertos valores potenciales.

La función de probabilidad es diferente de la función de pertenencia de la matemática borrosa, la función de pertenencia describe en qué grado un objeto pertenece a un conjunto, sin embargo, la función de probabilidad describe la probabilidad de que un número gris tome un determinado valor, o la probabilidad que cierto valor sea el valor verdadero de un número gris. (Liu, Yang, & Forrest, Grey Data Analysis, 2017)

2.1.3 Operadores de secuencia

La teoría de los sistemas grises se desarrolla en el descubrimiento de la correlación matemática entre las diferentes variables de un sistema y las leyes de cambio de determinadas variables, a partir de los datos disponibles de los comportamientos característicos de los sistemas como por ejemplo pueden ser, los sociales, económicos, ecológicos, entre muchos otros. La teoría de los sistemas grises observa cada variable estocástica como una cantidad gris que varía en una región fija y en una determinada ventana de tiempo, y define cada proceso estocástico como un proceso gris.

Todos los sistemas tienen funciones y propiedades generales, incluso si el comportamiento de dicho sistema es complicado y sus datos caóticos, de acuerdo con lo anterior y en el marco de la teoría de los sistemas grises, entonces deben existir leyes mínimas que determinen la existencia del sistema, es fundamental elegir el método correcto para descubrir qué modelo determina el sistema y así comprenderlo.

Cuando se investigan las características de comportamiento de un sistema, lo que está disponible es una secuencia definida de números blancos. La teoría de los sistemas grises revela las leyes del cambio a través del descubrimiento y organización de los datos disponibles, utilizando los operadores de secuencia gris para encontrar datos a partir de otros datos.

2.1.4 Operadores acumulativos y operadores acumulativos inversos

El operador acumulativo es el método empleado para extraer las leyes implícitas en una secuencia de datos grises. Juega un papel extremadamente importante en el modelado de sistemas grises, mediante el método del operador acumulativo se puede potencialmente descubrir una tendencia de desarrollo existente en el proceso de cantidades grises acumuladas. Esto permite que las leyes y características de integración, escondidas en los datos caóticos originales sean suficientemente reveladas.

El operador acumulativo es generalmente empleado para obtener datos adicionales de una pequeña cantidad de información disponible.

Para una secuencia original: $X^{(0)} = (x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n))$, D es una secuencia de operador definido de la manera siguiente:

$X^{(0)}D = (x^{(0)}(1)d, x^{(0)}(2)d, \dots, x^{(0)}(n)d)$, en donde:

$$x^{(0)}(k)d = \sum_{i=1}^k x^{(0)}(i); k = 1, 2, \dots, n. \quad (1)$$

En este contexto, D es llamado operador acumulador de generación de $X^{(0)}$, denotado como 1-AGO. $X^{(0)}D$, la secuencia desarrollada mediante el operador acumulativo D, es denotada por $X^{(1)}$:

$$X^{(0)}D = X^{(1)} = (x^{(0)}(1)d, x^{(0)}(2)d, \dots, x^{(0)}(n)d) \quad (2)$$

Para una secuencia original: $X^{(0)} = (x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n))$, D es una secuencia de operador definido de la manera siguiente:

$X^{(0)}D = (x^{(0)}(1)d, x^{(0)}(2)d, \dots, x^{(0)}(n)d)$, en donde:

$$x^{(0)}(k) = x^{(1)}(k) - x^{(1)}(k - 1); k = 1, 2, \dots, n. \quad (3)$$

En este contexto, D es llamado operador acumulador inverso de generación de $X^{(0)}$, denotado como 1-IAGO. $X^{(0)}D$, la secuencia desarrollada mediante el operador acumulativo D, es denotada por $X^{(1)}$ (Liu, Yang, & Forrest, 2017)

2.1.5 Operador serie Z

Como se menciona en (Julong D., 1989) la serie Z, así como como los operadores 1-AGO y 1-IAGO es un operador de generación dentro de la teoría sistemas grey, es habitual encontrarlo en el proceso de modelado y obtención de un pronóstico dentro de los modelos grey, funciona mediante la obtención de un promedio para cada dos vecinos contiguos de la serie obtenida mediante el operador 1-AGO, obteniendo de esta manera un suavizado de los datos de la serie acumulativa. El operador de serie Z puede ser definido como:

$$Z^{(1)} = \{z^{(1)}(2), z^{(1)}(3), \dots, z^{(1)}(N)\} \quad (4)$$

En donde:

$$z^{(1)}(k) = \frac{x^{(1)}(k) + x^{(1)}(k + 1)}{2} \quad (5)$$

Siendo $x^{(1)}$ la serie obtenida mediante el operador acumulativo 1-AGO a partir de la serie primitiva.

2.2 Regresión por mínimos cuadrados

2.2.1 Regresión lineal

La aproximación más simple por mínimos cuadrados es ajustar una línea recta a un conjunto de observaciones definidas por puntos $(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)$. La expresión para la línea recta es:

$$y = a_0 + a_1x + e \quad (6)$$

Donde a_0 y a_1 , son coeficientes que representan la intersección entre el eje y y la pendiente, respectivamente, e es el error, o diferencia, entre el modelo y las observaciones, el cual se representa al redondear la ecuación como:

$$e = y - a_0 + a_1x \quad (7)$$

Así el error o residuo es la discrepancia entre el valor verdadero de y y el valor aproximado $a_0 + a_1x$. (Chapra & Canale, 2015, pág. 358)

2.2.2 Criterio de mejor ajuste

De acuerdo con (Chapra & Canale, 2015, pág. 358). Si se minimiza la suma de los cuadrados de los residuos entre la y medida y la y calculada con el modelo lineal, se obtiene una línea única que representa cierto conjunto de datos.

$$S_r = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_{i \text{ medida}} - y_{i \text{ modelo}})^2 = \sum_{i=1}^n (a_0 + a_1 x_i)^2 \quad (8)$$

2.2.3 Ajuste de una línea recta por mínimos cuadrados

Para determinar los valores de a_0 y a_1 , se deriva la ecuación anterior con respecto de cada uno de los coeficientes.

$$\frac{\partial S_r}{\partial a_0} = -2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i) \quad (9)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum_{i=1}^n [(y_i - a_0 - a_1 x_i)x_i] \quad (10)$$

Al igualar estas derivadas a 0, se dará como resultado un S_r mínimo

$$0 = \sum_{i=1}^n y_i - \sum_{i=1}^n a_0 - \sum_{i=1}^n a_1 x_i \quad (11)$$

$$0 = \sum_{i=1}^n y_i x_i - \sum_{i=1}^n a_0 x_i - \sum_{i=1}^n a_1 x_i^2 \quad (12)$$

Al resolver las ecuaciones normales simultáneamente obtenemos el parámetro a_1

$$a_1 = \frac{n \sum_{i=1}^n y_i x_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \quad (13)$$

En conjunto con las ecuaciones normales se obtiene:

$$a_0 = \bar{y} - a_1 \bar{x} \quad (14)$$

2.2.4 Regresión polinomial

El procedimiento de mínimos cuadrados se puede ampliar fácilmente al ajuste de datos con un polinomio de grado superior. Para un polinomio de segundo grado o cuadrático como:

$$y = a_0 + a_1x + a_2x^2 + e \quad (15)$$

Para este caso la suma de los cuadrados de los residuos se expresa como:

$$S_r = \sum_{i=1}^n (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2 \quad (16)$$

Se continúa como el ajuste de una recta, pero ahora se deriva parcialmente con respecto de cada uno de los coeficientes desconocidos del polinomio.

$$\frac{\partial S_r}{\partial a_0} = -2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i - a_2 x_i^2) \quad (17)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i - a_2 x_i^2) x_i \quad (18)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum_{i=1}^n (y_i - a_0 - a_1 x_i - a_2 x_i^2) x_i^2 \quad (19)$$

De las derivaciones parciales se obtiene un conjunto de ecuaciones normales, de las cuales se puede formar un sistema de tres ecuaciones y tres incógnitas

que puede ser resuelta por técnicas tradicionales, de esta manera obtenemos los parámetros de ajuste a_0, a_1, a_2 .

2.2.5 Formulación de una matriz para regresión por mínimos cuadrados lineales.

Se representa a continuación el modelo lineal general de mínimos cuadrados:

$$y = a_0\theta_0 + a_1\theta_1 + a_2\theta_2 + \dots + a_m\theta_m + e \quad (20)$$

En este contexto $\theta, \theta_1, \theta_2 \dots \theta_m$ son $m + 1$ funciones diferentes. La ecuación anterior puede ser representada de manera matricial de la forma siguiente:

$$\{Y\} = [\theta]\{A\} + \{E\} \quad (21)$$

$[\theta]$ Representa una matriz contenedora de los valores calculados de las funciones θ en los valores medidos de las variables independientes.

$$[\theta] = \begin{bmatrix} \theta_{01} & \theta_{11} & \dots & \theta_{m1} \\ \theta_{02} & \theta_{12} & \dots & \theta_{m2} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \theta_{0n} & \theta_{1n} & \dots & \theta_{mn} \end{bmatrix} \quad (22)$$

De esta manera m es el número de variables en el modelo y n es el número de datos.

El vector columna $\{Y\}$ contiene los valores observados de la variable dependiente:

$$\{Y\}^T = [y_1 \ y_2 \ \dots \ y_n] \quad (23)$$

El vector columna $\{A\}^T$ contiene los coeficientes desconocidos:

$$\{A\}^T = [a_0 \ a_1 \ \dots \ a_m] \quad (24)$$

El vector columna $\{E\}$ contiene los errores residuales

$$\{E\}^T = [e_1 \ e_2 \ \dots \ e_n] \quad (25)$$

La suma de los cuadrados de los residuos en el presente modelo se reduce como:

$$S_r = \sum_{i=1}^n (y_i - \sum_{j=0}^m (a_j \theta_{ji}))^2 \quad (26)$$

Se lleva esta cantidad a su mínimo haciendo uso de las derivadas parciales con respecto a cada uno de los coeficientes y se procede a igualar a cero, obteniendo como resultado del proceso las ecuaciones normales, organizadas de manera matricial como:

$$[[\theta]^T [\theta]] \{A\} = \{ [\theta]^T \{Y\} \} \quad (27)$$

De la ecuación anterior resolviendo para $\{A\}$ se obtiene

$$\{A\} = [[\theta]^T [\theta]]^{-1} [\theta]^T \{Y\} \quad (28)$$

2.3 Modelos Grey

Para el desarrollo de esta investigación se presentan algunos de los modelos grey enfocados a la predicción de sistemas, iniciando por la base de muchos modelos, el modelo GM (1, 1), continuando hasta el modelo grey discreto recursivo.

2.3.1 Modelo Grey (1, 1)

Esta clase de modelos se podrían considerar los modelos base para la predicción de sistemas dentro de TSG, el modelo base GM (1, 1), se ha utilizado ampliamente en diversas investigaciones, el uso de este modelo ha demostrado

aproximar el comportamiento en el tiempo de diversos sistemas, permitiendo predecir con tasas de error aceptables.

Sea $X^{(0)} = \{x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(N)\}$, $X^{(1)} = \{x^{(1)}(1), x^{(1)}(2), \dots, x^{(1)}(N)\}$,
 $Z^{(1)} = \{z^{(1)}(2), z^{(1)}(3), \dots, z^{(1)}(N)\}$

En donde $x^{(1)}(k) = \sum_{i=1}^k x^{(0)}(i)$ además de $z^{(1)}(k) = \frac{x^{(0)}(k) + x^{(0)}(k+1)}{2}$,

entonces $x^{(0)}(k) + az^{(1)}(k) = b$ es denominado como modelo gris (1,1).

El vector de parámetros $\hat{a} = [a, b]^T$ puede ser estimado haciendo uso del método de mínimos cuadrados, el cual satisface que:

$$\hat{a} = ([B]^T[B])^{-1} [B]^T\{Y\} \quad (29)$$

En donde los elementos de la matriz B están compuestos como

$$B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \vdots & \vdots \\ -z^{(1)}(n) & 1 \end{bmatrix}, Y = \begin{bmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \vdots \\ x^{(0)}(n) \end{bmatrix} \quad (30)$$

A continuación, se define la llamada “ecuación de blanqueamiento” representada por la siguiente ecuación diferencial:

$$\frac{dx^{(1)}}{dt} + ax^{(1)} = b \quad (31)$$

Para formar el modelo gris se da solución a la ecuación de blanqueamiento, obteniendo la siguiente formulación:

$$\hat{x}^{(1)}(k+1) = \frac{b}{a} + \left(x^{(0)}(1) - \frac{b}{a}\right) e^{-ak}, k = 1, 2, \dots, n \quad (32)$$

Obtenida una secuencia con el modelo gris generado, el último paso sería restablecer la secuencia al orden original llevándola de $\hat{x}^{(1)}$ a $\hat{x}^{(0)}$ mediante el operador acumulativo inverso o 1-IAGO ecuación 3 descrita en la sección 2.1.4 y definido de la siguiente manera:

$$x^{(0)}(k) = x^{(1)}(k) - x^{(1)}(k-1); k = 1, 2, \dots, n.$$

2.3.2 Modelo Grey multivariable (1, n)

Como se menciona en (Liu, Yang, & Forrest, Grey Data Analysis, 2017), este modelo tiene sustento similar al modelo grey (1,1), con la diferencia de la inclusión de n variables, ampliando así a la correlación de n variables para ajustar el modelo de la salida del sistema. Es junto con el modelo grey (1,1) los modelos esenciales para la estimación de pronósticos en la teoría de sistemas grey.

Sea $X_i^{(0)} = \{x_i^{(0)}(1), x_i^{(0)}(2), \dots, x_i^{(0)}(N)\}$, $X_i^{(1)} = \{x_i^{(1)}(1), x_i^{(1)}(2), \dots, x_i^{(1)}(N)\}$,
 $(i = 1, 2, \dots, n)$, $Z^{(1)} = \{z^{(1)}(2), z^{(1)}(3), \dots, z^{(1)}(N)\}$

En donde $x^{(1)}(k) = \sum_{i=1}^k x^{(0)}(i)$ además de $z^{(1)}(k) = \frac{x^{(0)}(k) + x^{(0)}(k+1)}{2}$

Entonces $X_i^{(1)}(k) + az_1^{(1)}(k) = \sum_{i=2}^n b_i x_i^{(1)}(k)$ es denominado como modelo gris (1, n).

El vector de parámetros $\hat{a} = [a, b]^T$ puede ser estimado haciendo uso del método de mínimos cuadrados, el cual satisface que:

$$\hat{a} = ([B]^T [B])^{-1} [B]^T \{Y\} \quad (33)$$

En donde los elementos de la matriz B están compuestos como

$$B = \begin{bmatrix} -z^{(1)}(2) & x_2^{(1)}(2) & \dots & x_i^{(1)}(2) \\ -z^{(1)}(3) & x_2^{(1)}(3) & \dots & x_i^{(1)}(3) \\ \vdots & \vdots & \ddots & \vdots \\ -z^{(1)}(n) & x_2^{(1)}(n) & \dots & x_i^{(1)}(n) \end{bmatrix}, Y = \begin{bmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \vdots \\ x^{(0)}(n) \end{bmatrix} \quad (34)$$

Obtenida una secuencia con el modelo gris generado, el último paso sería restablecer la secuencia al orden original llevándola de $\hat{x}^{(1)}$ a $\hat{x}^{(0)}$ mediante el operador acumulativo inverso o 1-IAGO ecuación 3 descrita en la sección 2.1.4 y definido de la siguiente manera:

$$x^{(0)}(k) = x^{(1)}(k) - x^{(1)}(k - 1); k = 1, 2, \dots, n.$$

2.3.3 Modelo Grey multivariable con integral de convolución GMC (1, n)

Este modelo incluye la integral de convolución dentro de su modelado, con el objetivo de incrementar la convergencia del pronóstico en comparación con el sistema real.

$$\text{Sea } X_i^{(0)} = \{x_i^{(0)}(1), x_i^{(0)}(2), \dots, x_i^{(0)}(N)\}, \quad X_i^{(1)} = \{x_i^{(1)}(1), x_i^{(1)}(2), \dots, x_i^{(1)}(N)\}, \\ (i = 1, 2, \dots, n), \quad Z^{(1)} = \{z^{(1)}(2), z^{(1)}(3), \dots, z^{(1)}(N)\}$$

En (Tzu-Li Tien, 2005) se define al modelo GMC (1, n), como una ecuación diferencial lineal, siendo esta:

$$\frac{dx_1^{(1)}(rp + t)}{dt} + b_1 x_1^{(1)}(rp + t) = \sum_{i=2}^n b_i x_i^{(1)}(t) + u \quad (35)$$

En donde $x_1^{(1)}(rp + t) = \sum_{k=rp+1}^{rp+t} x_1^{(0)}(k)$ $t = 1, 2, \dots, r + rf$ además de $x_1^{(1)}(t) = \sum_{k=1}^t x_1^{(0)}(k)$ $t = 1, 2, \dots, r + rf$

r es el número de entradas para construir el modelo, rp es el periodo de retardo, rf es el número de entradas que serán pronosticadas y $b_1, b_2 \dots b_n, u$, son los parámetros del modelo a ser estimados.

El vector de parámetros $\hat{a} = [b_1, b_2, \dots, b_n, u]^T$ puede ser estimado haciendo uso del método de mínimos cuadrados, el cual satisface que:

$$\hat{a} = ([B]^T [B])^{-1} [B]^T \{Y\} \quad (36)$$

En donde los elementos de la matriz B están compuestos como

$$B = \begin{bmatrix} -z_1^{(1)}(rp+2) & z_2^{(1)}(2) & \cdots & z_i^{(1)}(2) \\ -z_1^{(1)}(rp+3) & z_2^{(1)}(3) & \cdots & z_i^{(1)}(3) \\ \vdots & \vdots & \ddots & \vdots \\ -z_1^{(1)}(rp+r) & z_2^{(1)}(n) & \cdots & z_i^{(1)}(n) \end{bmatrix}, Y = \begin{bmatrix} x_1^{(0)}(2) \\ x_1^{(0)}(3) \\ \vdots \\ x_1^{(0)}(r) \end{bmatrix} \quad (37)$$

La función de respuesta para $\hat{x}_1^{(1)}(rp+t)$ puede ser obtenida por la ecuación

$$\hat{x}_1^{(1)}(rp+t) = x_1^{(0)}(rp+1) e^{-b_1(t-1)} + \int_1^t e^{-b_1(t-\tau)} f(\tau) dt. \quad (38)$$

En donde $f(\tau) dt = \sum_{i=2}^n b_i x_i^{(1)}(t) + u$

La integral de convolución puede ser discretizada utilizando la formula gaussiana y la serie 1-AGO puede ser evaluada utilizando la siguiente función discreta:

$$\hat{x}_1^{(1)}(rp + t) = x_1^{(0)}(rp + 1) e^{-b_1(t-1)} + u(t) \quad (39)$$

$$- 2) \times \sum_{\tau=2}^t \left\{ e^{-b_1(t-\tau+\frac{1}{2})} * \frac{1}{2} [f(\tau) + f(\tau - 1)] \right\}$$

La función de escalón unitario $u(t - 2)$ es definida como

$$u(t - 2) = \begin{cases} 0, & t < 2 \\ 1, & t \geq 2 \end{cases} \quad (40)$$

Obtenida una secuencia con el modelo gris generado, el último paso sería restablecer la secuencia al orden original llevándola de $\hat{x}^{(1)}$ a $\hat{x}^{(0)}$ mediante el operador acumulativo inverso o 1-IAGO ecuación 3 descrita en la sección 2.1.4 y definido de la siguiente manera:

$$x^{(0)}(k) = x^{(1)}(k) - x^{(1)}(k - 1); k = 1, 2, \dots, n.$$

2.3.4 Modelo Grey discreto recursivo multivariable RDGM (1, n)

En (Ma & Liu, 2015) se propone este modelo, es otra alternativa para un pronóstico por modelado gris de manera multivariable.

Sea $X_i^{(0)} = \{x_i^{(0)}(1), x_i^{(0)}(2), \dots, x_i^{(0)}(N)\}$, $X_i^{(1)} = \{x_i^{(1)}(1), x_i^{(1)}(2), \dots, x_i^{(1)}(N)\}$,
 $(i = 1, 2, \dots, n)$, $Z^{(1)} = \{z^{(1)}(2), z^{(1)}(3), \dots, z^{(1)}(N)\}$

El modelo RDGM (1, n) es basado en

$$\frac{dx_1^{(1)}(rp+t)}{dt} + b_1 x_1^{(1)}(rp+t) = \sum_{i=2}^n b_i x_i^{(1)}(t) + u(t) \quad (41)$$

Si se considera la integración en el intervalo $[k, k+1]$ tenemos:

$$\int_k^{k+1} dX_1^{(1)}(rp+t) + b_1 \int_k^{k+1} X_1^{(1)}(rp+t) dt = \sum_{i=2}^n b_i \int_k^{k+1} X_i^{(1)}(t) dt + \int_k^{k+1} u dt. \quad (42)$$

Entonces el modelo RDGM (1, n) puede ser representado como

$$X_1^{(1)}(rp+k+1) = \frac{1-0.5b_1}{1+0.5b_1} X_1^{(1)}(rp+k) + \sum_{i=2}^n \frac{b_i}{1+0.5b_1} Z_i^{(1)}(k+1) + \frac{u}{1+0.5b_1} \quad (43)$$

Ajustando a:

$$\beta_1 = \frac{1-0.5b_1}{1+0.5b_1}, \beta_i = \frac{b_i}{1+0.5b_1}, \mu = \frac{u}{1+0.5b_1}, (i = 2, \dots, n) \quad (44)$$

Sustituyendo se obtiene la siguiente ecuación:

$$X_1^{(1)}(rp + k + 1) = \beta_1 X_1^{(1)}(rp + k) + \sum_{i=2}^n \beta_i Z_i^{(1)}(k + 1) + u \quad (45)$$

El vector de parámetros $\hat{a} = [\beta_1, \beta_2, \dots, \beta_n, u]^T$ puede ser estimado haciendo uso del método de mínimos cuadrados, el cual satisface que:

$$\hat{a} = ([B]^T [B])^{-1} [B]^T \{Y\} \quad (46)$$

En donde los elementos de la matriz B están compuestos como

$$B = \begin{bmatrix} x_1^{(1)}(rp + 1) & z_2^{(1)}(2) & \dots & z_i^{(1)}(2) \\ x_1^{(1)}(rp + 2) & z_2^{(1)}(3) & \dots & z_i^{(1)}(3) \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(1)}(rp + r - 1) & z_2^{(1)}(r) & \dots & z_i^{(1)}(r) \end{bmatrix}, Y = \begin{bmatrix} x_1^{(1)}(rp + 2) \\ x_1^{(1)}(rp + 3) \\ \vdots \\ x_1^{(1)}(rp + r) \end{bmatrix} \quad (47)$$

Con el vector de parámetros estimados \hat{a} y las series $X_i^{(1)}$ y $Z_i^{(1)}$ se le puede dar solución al modelo grey discreto recursivo descrito por Ma, X., & Liu, Z. (2016) y representado en (47) considerando la condición inicial $\hat{X}_1^{(1)}(rp + 1) = \hat{X}_1^{(0)}(rp + 1) = X_1^{(0)}(rp + 1)$.

$$\hat{X}_1^{(1)}(rp + k + 1) = \beta_1^k X_1^{(0)}(rp + 1) + \sum_{m=2}^{k+1} \beta_1^{(k+1-m)} \left(\sum_{i=2}^n \beta_i Z_i^{(1)}(m) \right) + \frac{1 - \beta_1^k}{1 - \beta_1} \mu \quad (48)$$

Obtenida una secuencia con el modelo gris generado, el último paso sería restablecer la secuencia al orden original llevándola de $\hat{x}^{(1)}$ a $\hat{x}^{(0)}$ mediante el operador acumulativo inverso o 1-IAGO ecuación 3 descrita en la sección 2.1.4 y definido de la siguiente manera:

$$x^{(0)}(rp + k + 1) = x^{(1)}(rp + k + 1) - x^{(1)}(rp + k)$$

2.4 Instrumentación

2.4.1. SHT10 sensor humedad relativa y temperatura

Para el éxito de esta investigación es importante contar con un sensor de temperatura y humedad relativa confiable y con resultados constantes. En el mercado existen diversas alternativas que podrían cumplir con estos objetivos, pero debido a su disponibilidad y su relación calidad precio se opta por el sensor SHT10 de SENSIRION®. Este es un sensor de montaje superficial de humedad relativa y temperatura. El dispositivo integra los elementos sensores y el procesamiento de la señal en un mismo paquete entregando una salida totalmente calibrada de manera digital.

El efecto capacitivo es utilizado para medir la humedad relativa, mientras que para la temperatura se realiza la medición mediante un sensor de brecha de banda. Ambos sensores están acoplados a un convertidor analógico a digital de 14 bits. El dispositivo comunica la información ya procesada y digitalizada mediante una interfaz de dos terminales.

2.4.1.1 Funcionamiento del sensor de humedad relativa capacitivo

Consiste en un pequeño capacitor de material eléctrico higroscópico colocado al centro de dos electrodos. Cuando no hay humedad presente en el sensor la constante dieléctrica y la geometría del sensor determinan el valor de la capacitancia. A temperatura ambiente la constante dieléctrica del vapor de agua tiene un valor cercano de 80, un valor mucho más grande que la constante dieléctrica del material dieléctrico del sensor entonces, la absorción de humedad por el sensor resulta en un incremento en la capacitancia.

La humedad relativa es una función que relaciona la presión del vapor de agua y la temperatura ambiente, entonces esta relación existe también de manera directa entre la cantidad de humedad presente en el sensor, la humedad relativa y la capacitancia del sensor. Es así como los instrumentos de medición de la humedad relativa estiman el valor en el ambiente. Para determinar el desempeño del sensor hay que tomar en cuenta todos los elementos que lo componen como son el elemento capacitivo, el cable de transmisión de datos, el hardware de acondicionamiento de señal y el de entrega de la señal de salida.

2.4.1.2 Funcionamiento del sensor de temperatura

Los sensores de brecha de bandas implementados en silicio es un tipo de termómetro o sensor de temperatura utilizado de manera común en dispositivos electrónicos, tienen una buena estabilidad debido a la integridad de la estructura cristalina del silicio, es decir que el sensor está directamente añadido al dispositivo mediante métodos fotolitográficos. El principio de medición se obtiene mediante la dependencia de la temperatura en el voltaje que se aplica a un diodo de silicio en polarización directa, dicho voltaje es representado en la siguiente ecuación:

$$V_{BE} = V_{G0} \left(1 - \frac{T}{T_0}\right) + V_{BE0} \left(\frac{T}{T_0}\right) + \left(\frac{nKT}{q}\right) \ln\left(\frac{T_0}{T}\right) + \left(\frac{KT}{q}\right) \ln\left(\frac{I_c}{I_{c0}}\right) \quad (49)$$

Donde T es la temperatura en grados kelvin, T_0 la referencia de temperatura, V_{G0} el voltaje de la banda prohibida en el cero absoluto, V_{BE0} es el voltaje en la unión cuando la temperatura es T_0 y la corriente I_{c0} y K es la constante de Boltzmann. (Widlar, R.J., 1967).

Algunas de las ventajas de utilizar este tipo de sensores son:

Su bajo costo, su precisión y confiabilidad, sus mediciones constantes y su alto desempeño a través de su tiempo de vida útil.

2.4.1.3 Encapsulado del sensor SHT10

Si bien el sensor entregado directamente del fabricante cuenta con un encapsulado de bajo perfil y unas terminales de conexión diseñadas para el montaje superficial, para esta aplicación y debido a que éste deberá estar colocado en la intemperie de un invernadero se opta por conseguir un dispositivo que contenga al sensor y un encapsulado robusto. El encapsulado consiste en una aglomeración de esferas de acero inoxidable, éstas constituyen un contenedor en forma de prisma circular que permite el paso del aire en cantidades adecuadas para el muestreo, lo que ayuda a evitar una saturación en el elemento sensor. La carcasa funciona como un filtro de metal sinterizado para una gran permeabilidad al aire. La carcasa es resistente al agua y evitará que el agua se filtre en el cuerpo del sensor y lo dañe, pero permite que el aire pase a través de él para que pueda medir la humedad relativa dentro del invernadero como se observa en la Figura 1.



Figura 1 Sensor de Temperatura SHT10

Fuente: SHT-10 Mesh-protected Weather-proof Temperature/Humidity Sensor, Adafruit Industries©

2.4.2 Sensor de luminosidad TSL2561

Este es un sensor de luminosidad, métrica que nos permitirá conocer indirectamente la intensidad con la que incide la luz del Sol sobre el invernadero. El TSL2561 entrega los datos medidos ya procesados mediante un bus de datos serial I2C, lo que facilita la lectura de los datos además de un cálculo preciso de la cantidad de la intensidad luminosa determinada por la configuración interna de ganancias. El elemento sensor es la combinación de un fotodiodo sensible a la luz visible además de un segundo fotodiodo sensible a la luz infrarroja, ambos dispositivos integrados en un mismo circuito con tecnología CMOS. Dos convertidores analógico a digital embebidos en el encapsulado convierten las corrientes de los fotodiodos en una señal de salida digital que representa la radiación de luz medida por cada fotodiodo, esto es alimentado a un microprocesador en el cual se encuentra embebido el cálculo mediante una fórmula que permite aproximar el valor de esta intensidad en unidad Lux. De la hoja de datos del fabricante se obtiene un diagrama con los componentes principales del sensor además de sus terminales como se muestra en la figura 2.

Functional Block Diagram

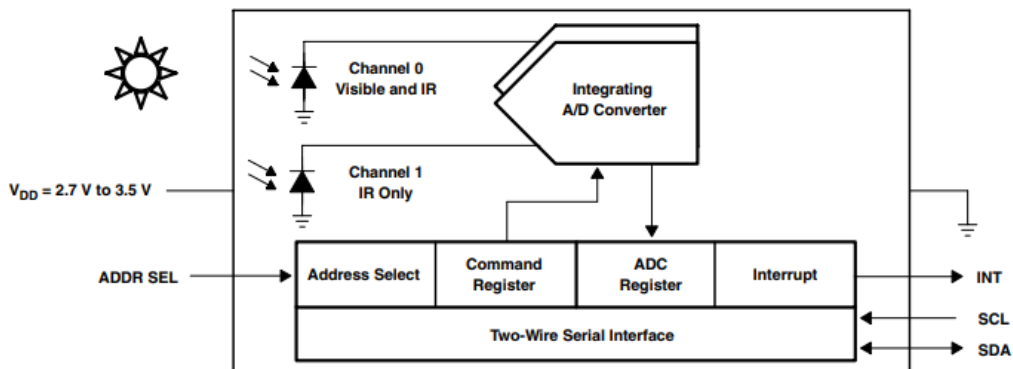


Figura 2 Diagrama sensor TSL2561.

Fuente: TSL2561 LIGHT-TO-DIGITAL CONVERTER, 2009, TAOS©

2.4.2.1 Principio de funcionamiento del fotodiodo

Como menciona (Boylestad & Nashelsky, 2003), el fotodiodo es un dispositivo semiconductor de unión P-N cuya región de operación se limita a la región de polarización inversa. En el cual, el número de electrones libres generado es proporcional a la intensidad de la luz incidente. La intensidad luminosa es una medida de cantidad de flujo luminoso que incide sobre un área de una superficie particular, el flujo luminoso por lo general se mide en lúmenes. La aplicación de luz a la unión entre los materiales semiconductores del diodo ocasionará una transferencia de energía de las ondas de luz incidentes en forma de fotones hacia la estructura atómica del diodo, lo que ocasionará un incremento en el número de portadores minoritarios y un nivel mayor de corriente inversa.

De la hoja de datos del fabricante se obtienen las curvas características de la respuesta de los dos fotodiodos incluidos en el sensor a los diferentes espectros de radiación de luz como se observa en la Figura 3.

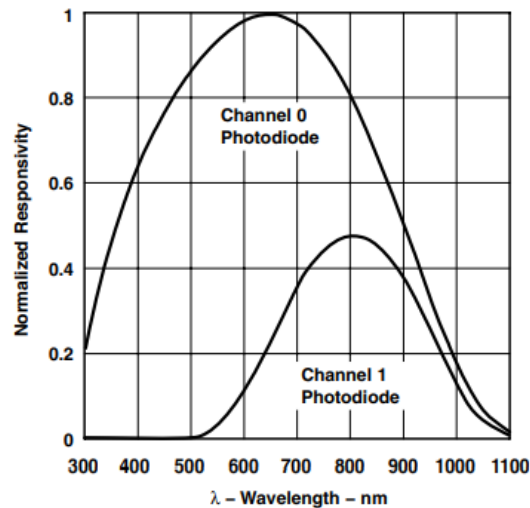


Figura 3 Curva de reacción de respuesta sensor TSL2561

Fuente: TSL2561 LIGHT-TO-DIGITAL CONVERTER, 2009, TAOS©

2.5 Procesamiento

2.5.1 Microcontrolador ATmega4809

Como elemento controlador y de procesamiento de señal se opta por el microcontrolador de 8 bits y con arquitectura AVR, el ATmega4809 de Microchip Technology®. Este microcontrolador pertenece a la familia megaAVR-0 y es capaz de funcionar a una frecuencia de hasta 20 mega Hertz en su reloj principal. Cuenta con 48 kB de memoria flash y 6 kB de memoria SRAM además de 256 bytes de memoria EEPROM, incorpora periféricos independientes al núcleo, varios modos de bajo consumo y un cristal oscilador incorporado directamente en el encapsulado.



Figura 4 Microcontrolador ATmega4809

Fuente: Minimal ATmega4809 on a Breadboard (<http://www.technoblogy.com/show?2QVZ>), 2020, David Johnson-Davies.

Las características importantes por las cuales se selecciona este microcontrolador son, su amplio rango de voltaje de operación que va desde 1.8 hasta 5.5 Volts lo que lo hace tolerante a variaciones en su fuente de alimentación. La segunda característica es la facilidad con la que esté puede ser programado utilizando herramientas Open Source además de la activa comunidad de usuarios que aportan bibliotecas de software abierto para diversas aplicaciones.

Los puertos serie son periféricos independientes al núcleo que escuchan mensajes entrantes y ejecutan acciones sin necesidad de la intervención de la unidad central de

procesamiento del microcontrolador, lo que permite una ejecución de ciertas tareas en paralelo.

2.6 Comunicaciones

2.6.1 Protocolo LoRa

LoRa es una tecnología de modulación por radiofrecuencia de bajo consumo y de largo alcance. El nombre de LoRa® es un acrónimo de “Long Range” o largo alcance traducido al español. Creada por Semtech® para estandarizar las redes de largo alcance, LoRa proporciona comunicaciones de hasta 5 km en áreas urbanas y de hasta 15 km en áreas rurales. Una característica clave de las soluciones basadas en LoRa son sus bajos requerimientos de consumo de energía, lo que permite la creación de dispositivos operados a baterías que pueden durar hasta 10 años. Si se utiliza un despliegue de tipo estrella en una red basada en el protocolo LoRa ésta será perfecta para aplicaciones que requieran comunicación de largo alcance o con mucha profundidad entre un gran número de dispositivos de bajo consumo y que recopilen pocos datos.

2.6.2 Modulación por radio LoRa

LoRa es la configuración del ancho de banda y tasa de transferencia de datos, de tal manera que se prioriza el rango de alcance, en la Figura 5 se observa la comparativa entre el ancho de banda y el rango de algunos protocolos populares:

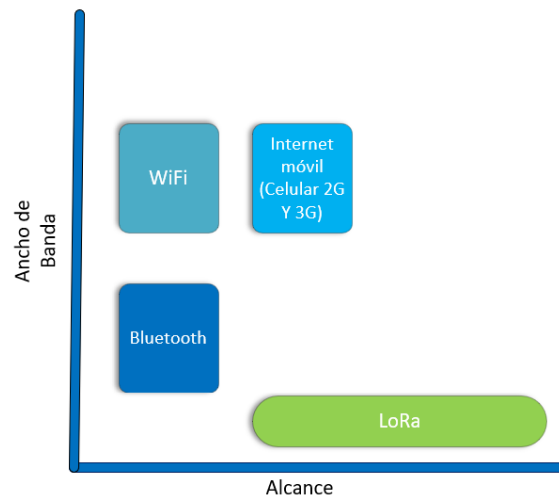


Figura 5 Alcance de varios protocolos inalámbricos.

Fuente: Adaptación traducida de LoRa® and LoRaWAN® A Technical Overview, 2018, Semtech Corporation®.

2.6.3 Link Budget

El *Link Budget*, también es llamado balance de potencia de transmisión indica la calidad del canal de transmisión por radiofrecuencia, utilizando un modelo simplificado el link Budget puede ser calculado mediante la suma de la potencia del transmisor, la sensibilidad del receptor, la ganancia de las antenas y el índice de pérdida en espacio libre.

LoRa es una técnica de modulación del espectro de esparcimiento propietaria, derivada de la existente tecnología de espectro de esparcimiento “chirp”, LoRa ofrece un balance entre recepción y tasa de envío de datos, mientras opera en un ancho de banda fijo de 125 KHz a 500 KHz. Adicionalmente LoRa utiliza factores de esparcimiento ortogonales, esto permite a la red de dispositivos preservar su cantidad de batería mediante la creación de optimizaciones especiales según los niveles de energía y tasas de datos de cada dispositivo. Por ejemplo, un dispositivo final cerca de una puerta de enlace debe transmitir a un factor de esparcimiento bajo; sin embargo, un dispositivo final localizado a varias millas de distancia de la puerta de

enlace va a necesitar transmitir con un factor de esparcimiento mucho mayor (Semtech Corporation, 2019).

De esta manera podemos sintetizar, LoRa® utiliza la metodología chirp que consta de modular la señal de datos por medio de una señal variante en frecuencia, además de implementar una tasa de transferencia de datos ajustable mediante la determinación de un factor de esparcimiento ortogonal, lo que permite a los desarrolladores configurar la modulación según las necesidades de transferencia de datos o de consumo energético.

2.6.4 Módulo transceptor LoRa E32-915D EBYTE

La tarjeta E3-915T20D es un módulo transceptor que implementa el protocolo LoRa® de largo alcance para la transmisión de los datos. Este modelo en particular utiliza el circuito integrado SX1276 de Semtech®, esta tarjeta está diseñada para transmitir los datos en radiofrecuencias de 868 MHz y 915 MHz, en México se permite la operación de transceptores de frecuencias de sub-Giga Hertz desde los 902 hasta los 928 MHz. La tarjeta interactúa con el microcontrolador mediante comunicación UART. (Chengdu Ebyte Electronic Technology, 2018)

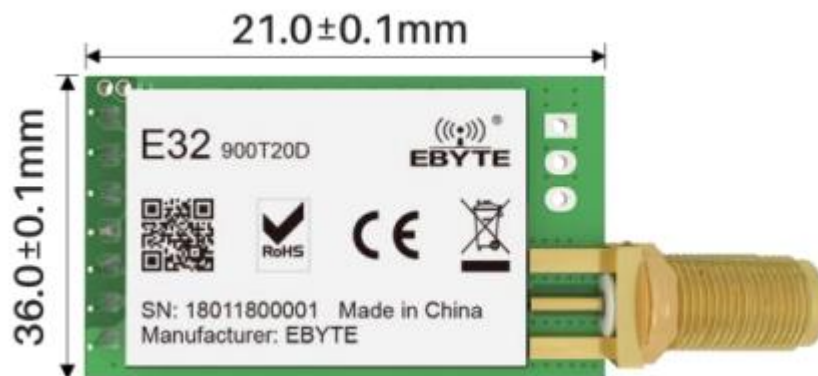


Figura 6 Módulo Transceptor LoRa E32-915,

Fuente: E32-915T20D User Manual, 2018, Chengdu Ebyte Electronic Technology.

Características:

- Modulación LoRa® avanzada con la ventaja de transmisión a larga distancia y con poca interferencia.
- En condiciones ideales puede alcanzar comunicaciones a distancias de hasta 3 km.
- “Air Wake up” permite un bajo consumo de energía, ideal para dispositivos que operan a baterías.
- Máxima ganancia de transmisión de 20dBm, ajustable por software.
- Tasa de transmisión de datos de 0.3kbps hasta 19.2kbps;
- Soporta un rango desde 2.3 ~ 5 V para la alimentación de la tarjeta.
- Diseño industrial para su uso a largo plazo en ambientes desde $-40C^{\circ}$ ~ $85C$.

Definición de pines:

PIN: 7 6 5 4 3 2 1

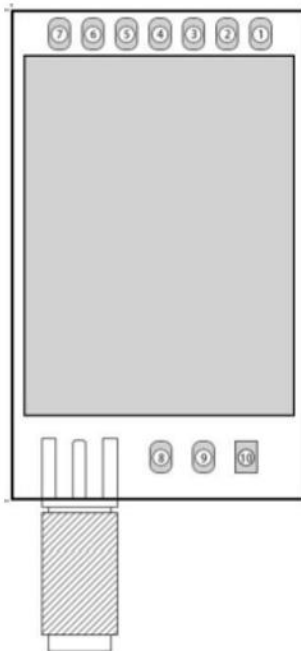


Figura 7 Diagrama de pines módulo E32-915

Fuente: E32-915T20D User Manual, 2018, Chengdu Ebyte Electronic Technology.

Tabla 1 Descripción de pines de conexión del módulo

| No. Pin | Nombre | Dirección | Función |
|---------|--------|-----------|--|
| 1 | M0 | Entrada | Los pines M0 y M1 sirven para configurar el modo de operación del módulo, (Normal, despertador, ahorro de energía, y modo de reposo) |
| 2 | M1 | Entrada | |
| 3 | RXD | Entrada | Entrada UART @3.3V, conectar a pin TXD del microcontrolador |
| 4 | TXD | Salida | Salida UART @3.3V, conectar a pin TXD del microcontrolador |
| 5 | AUX | Salida | Este pin se utiliza para comunicar el estado del módulo, por ejemplo, indicar cuando estamos recibiendo o transmitiendo datos |
| 6 | VCC | Entrada | Alimentación del módulo |
| 7 | GND | Entrada | Alimentación del módulo. |

Fuente: E32-915T20D User Manual, 2018, Chengdu Ebyte Electronic Technology.

2.6.4.1 Diagrama de conexión.

A continuación, se muestra un ejemplo de la implementación de comunicación del módulo con un microcontrolador STM8L, las conexiones varían según la implementación en software de la configuración y asignación de los pines.

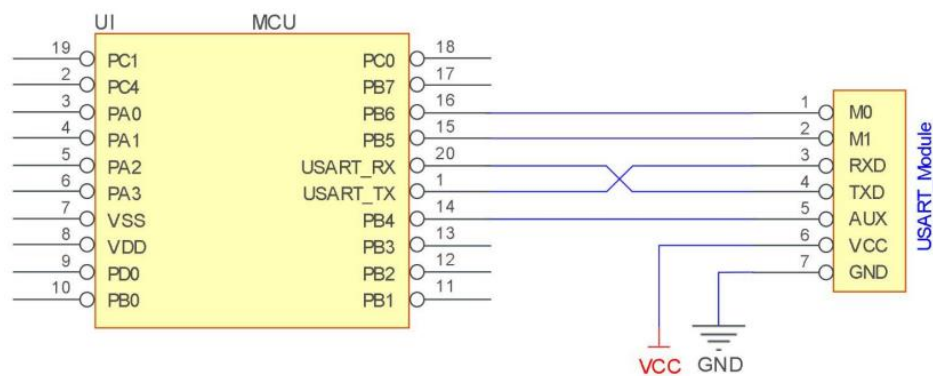


Figura 8 Ejemplo de conexión módulo E32-915.

Fuente: E32-915T20D User Manual, 2018, Chengdu Ebyte Electronic Technology.

2.6.4.2 Configuración del modo de operación del módulo.

El módulo puede operar en 4 configuraciones diferentes, se pueden usar GPIO del microcontrolador para determinar los estados lógicos que definirán el modo activo.

Tabla 2 Descripción de modos de operación

| Modo (0 – 3) | M0 | M1 | Descripción del modo | Nota |
|---------------------|----|----|--|--|
| 0 normal | 0 | 0 | Modo de transmisión transparente y continua, los canales UART e inalámbrico están activos | El receptor debe estar en modo 0 o 1 para funcionar. |
| 1 despertador | 1 | 0 | La única diferencia con el modo 0 es que antes de enviar los datos enviará una señal para despertar al módulo receptor. | El receptor puede estar en modos 0, 1 o 2. |
| 2 ahorro de energía | 0 | 1 | Canal UART cerrado, necesita de una señal para despertar. La señal la recibe el canal inalámbrico, una vez despierto abre el canal UART y transmite los datos recibidos. | El Transmisor debe estar en modo 1, en este modo no se puede transmitir. |
| 3 reposo | 1 | 1 | Modo de reposo, canales cerrados para el ahorro máximo de energía hasta salir del modo de reposo. | |

Fuente: E32-915T20D User Manual, 2018, Chengdu Ebyte Electronic Technology.

2.6.5 Tarjeta de desarrollo SIM808 GSM/GPRS/GPS.

Debido a la frecuencia con la que los cultivos en invernaderos se encuentran en la periferia de los centros urbanos o pueblos, su acceso a conectividad a la red es normalmente limitado. En orden de poder ofrecer una solución de monitoreo en tiempo real a través de la red se propone la inclusión de un modem de conectividad a la red celular GSM, para que, a través de ésta se puedan enviar los paquetes de datos contenedores de la información de monitoreo y predicción de las variables del invernadero, haciendo accesible esta información para ser desplegada en una solución para adquisición de datos y monitoreo como es por ejemplo Microsoft Azure IoT Central (Microsoft, 2022).

Para llegar a la solución propuesta, se opta por la tarjeta de desarrollo SIM808 EVB versión 1.3, cuyo elemento central es el SoC de SIMCOM®, el SIM808, pensado para tener todos los elementos necesarios para dar funcionamiento a un dispositivo de comunicación celular básico, entre sus capacidades se encuentran: el envío de llamadas, el envío de mensajes SMS, conexión a la red GSM para el acceso al internet a través de GPRS, sistema de rastreo GPS y conectividad Bluetooth.

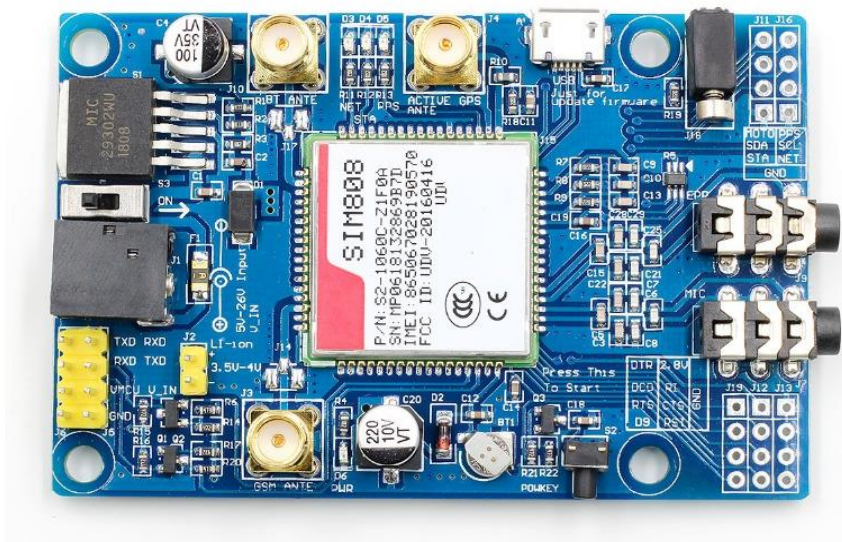


Figura 9 Tarjeta de desarrollo modem SIM808

Fuente: Módulo Sim808 GSM GPRS Con Antena GPS, (<https://uelectronics.com/producto/modulo-sim808-gsm-gprs-gps-con-antena-gps>), UNIT ELECTRONICS.

El módulo SIM808 es controlado de manera externa mediante un puerto serial universal asíncrono o por sus siglas en inglés UART, la tasa de velocidad de comunicación en baudios es variable y configurada por programación del módulo, siendo el estándar 115200 baudios, el módulo está programado para ejecutar ciertas rutinas según los comandos de atención o AT solicitados por un elemento maestro, a través de la comunicación serial. Estos comandos son ampliamente utilizados por fabricantes para el control de módems GSM, GPRS o de teléfonos móviles en general, y son usados para acceder a la información y los servicios del modem. Estos comandos pueden variar según el fabricante por lo que es necesario revisar

exhaustivamente la hoja de datos de la familia de módulo para conocer cuáles son los comandos disponibles y como ejecutarlos de manera correcta.

El módulo SIM808 necesita de un microcontrolador que funja como maestro, de esta manera el dispositivo se encarga de proporcionar las rutinas de códigos AT para que el modem ejecute ciertas operaciones, como es una solicitud HTTP mediante un método POST a una URL especifica. Las rutinas enviadas por el microcontrolador pueden estar condicionadas según la programación, esto ofrece la capacidad de ejecutar rutinas cada cierto intervalo de tiempo o a una hora especifica si se cuenta con un reloj en tiempo real (RTC).

La tarjeta de desarrollo con el módulo SIM808 requiere de una tarjeta SIM de un operador con cobertura en el área de aplicación de la solución, está tarjeta contiene los identificadores, como número telefónico, además de contener las cadenas que permitirán la identificación del dispositivo en la red celular, por lo que si no se cuenta con una tarjeta SIM con crédito ante el operador el dispositivo no funcionará.

El dispositivo maestro, la tarjeta de desarrollo con módulo SIM808 y tarjeta SIM válida tendrá la capacidad de conectarse a Internet mediante la conexión a una antena GSM próxima:

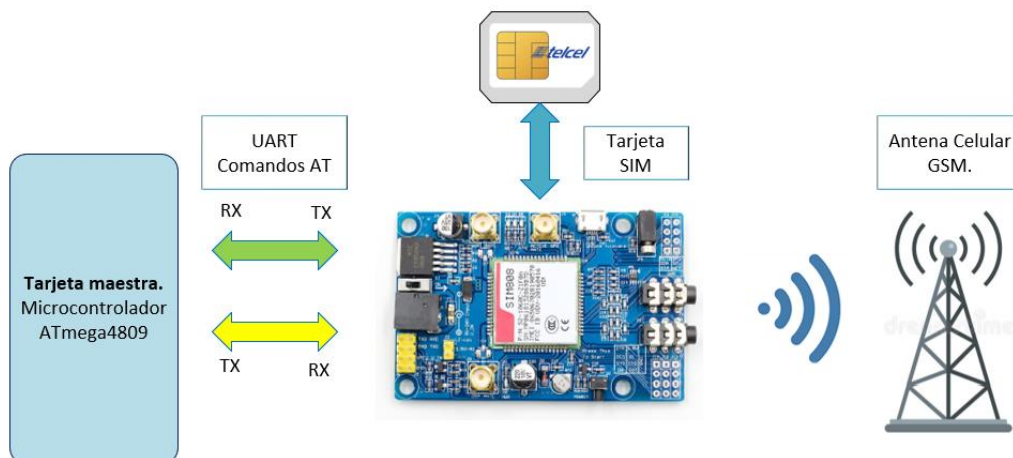


Figura 10 Esquema de funcionamiento modem SIM808.

Fuente: Elaboración propia.

El protocolo GSM el cual es un estándar que viene del acrónimo en inglés de sistema global para comunicaciones móviles, este estándar fue definido en 1991 y desde entonces se ha mantenido vigente. Comúnmente la manera en la que se brinda el acceso a esta red es a través de una antena que será la encargada de recibir las señales en un área determinada, El rango especificado para la red GSM es desde los 1850 a los 1990 MHz, la modulación que es el proceso de enviar la señal mediante la modificación de sus características en una señal portadora para la red GSM se utiliza GMSK (Gaussian minimum shift keying), la tasa de transmisión de datos en el aire es de 270kbps.

Este módulo cuenta con la capacidad de conexión a la red celular mediante el uso de la interfaz GSM y soporta las características para la transmisión de paquetes a través de radio conocido como GPRS (General Packet Radio System). La principal ventaja de GPRS es la implementación de las redes de datos con protocolos de Internet estandarizados cómo son TCP/IP. (Ghribi & Logrippo, 2000).

Esta capacidad de conexión le ofrece al módulo la posibilidad de conectarse a internet en locaciones sin acceso cableado a la red, ampliando las posibilidades de conectividad a servidores web para extender el monitoreo de los datos recibidos por la tarjeta de desarrollo.

2.6.6 Antenas de comunicación

Un elemento de alta importancia cuando se trabaja con transmisión de datos por radiofrecuencia es la antena acoplada al dispositivo transmisor pues, dependiendo de las características de ésta se pueden tener distintos resultados.

2.6.6.1 Antena para módulos LoRa TX915-XPL-100

Para obtener un buen rendimiento de los módulos LoRa E32-915D anteriormente mencionados es necesaria una antena adecuada. Para esto se seleccionada la antena TX915-XPL-100 cuyas características son las siguientes:

Tabla 3 Antena para módulos LoRa TX915-XPL-100

| Parámetro | Valor |
|---------------------------|-----------------|
| Rango de frecuencias | 900 – 915 MHz |
| Ganancia | 3.5dBi |
| SWR (Standing Wave Ratio) | < 1.5 |
| Capacidad de potencia | 10 watts |
| Impedancia de entrada | 50 Ω |
| Dirección de radiación | Omnidireccional |
| Conector | SMA-J |

Fuente: E32-915T20D User Manual, 2018, Chengdu Ebyte Electronic Technology.

Debido a que el dispositivo sensor debe ser relativamente compacto las dimensiones de la antena seleccionada son de un largo de 23 cm y de tres centímetros en su base.

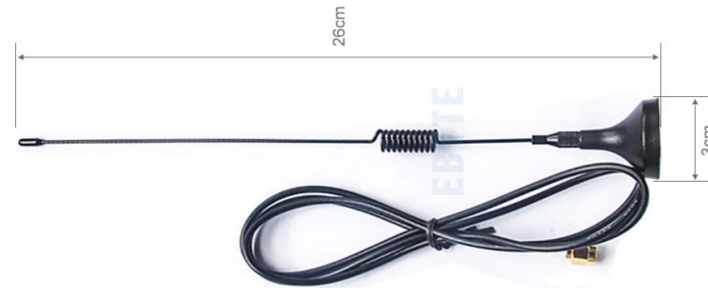


Figura 11 Antena del transceptor LoRa

Fuente: E32-915T20D User Manual, 2018, Chengdu Ebyte Electronic Technology.

2.7 Herramientas de software

Al desarrollar el firmware y algoritmo para el microcontrolador, es importante la selección de las herramientas para este propósito.

2.7.1 Editor de código

El punto de inicio es el editor de código, que es un punto medio entre un entorno de desarrollo integrado y un editor de texto normal. En el editor de código se escriben todas las sentencias, declaraciones y funciones. El editor seleccionado es Visual Studio Code de Microsoft, software gratuito y de fuente abierta. Lanzado en 2015, su principal atractivo es la cantidad de extensiones o “plugins” creados por la comunidad y por terceras empresas dedicadas el desarrollo. Cuenta con varias herramientas necesarias para un flujo de trabajo productivo.

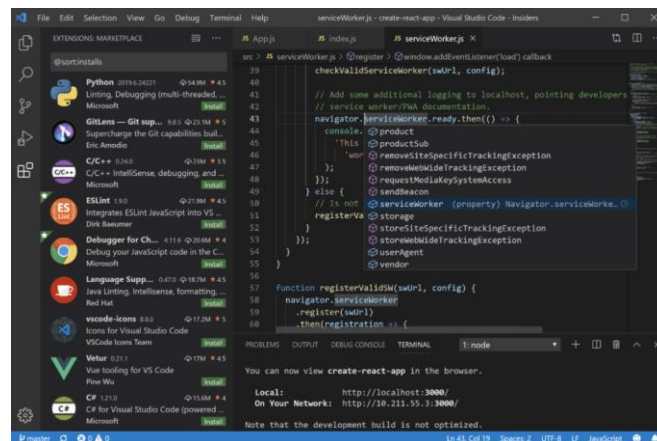


Figura 12 Software Visual Studio Code

Fuente: Visual Studio Code (<https://code.visualstudio.com/>), Microsoft Corporation.

2.7.2 PlatformIO

PlatformIO es una extensión instalable en Visual Studio Code, integra todas las herramientas necesarias para la programación de microcontroladores de distintos fabricantes y bajo una variedad de marcos de trabajo. Con esta herramienta podemos compilar el código en C y C++ a lenguaje de máquina en el formato requerido para el ATmega4809, en conjunto podemos programar el microcontrolador directamente desde esta herramienta mientras se cuente con un programador soportado como es el caso del MPLAB PICKIT 4.

La extensión se apoya en el compilador AVR-GCC y de la cadena de herramientas de avr-dude para la carga del programa al microcontrolador.

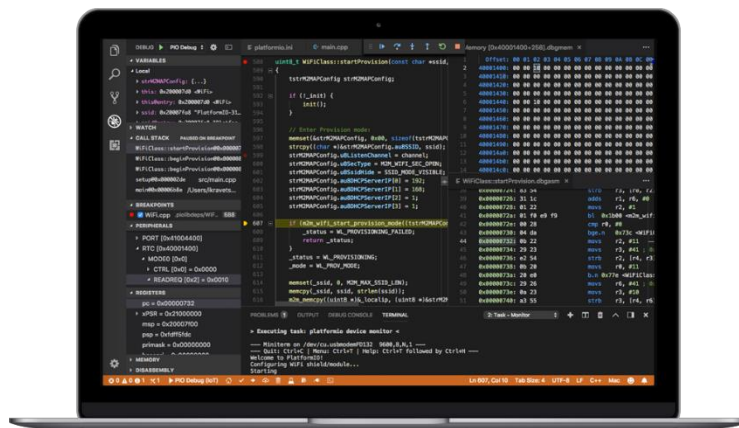


Figura 13 Extensión PlatformIO, Fuente:

Fuente: PlatformIO Visual Studio Code Extension, (<https://piolabs.com/>), PlatformIO Labs OÜ.

2.7.3 *Compilador AVR-GCC*

AVR-GCC es un compilador que toma lenguaje C de alto nivel y crea un archivo fuente en código máquina que puede ser programado en un microcontrolador AVR. Una vez que el código es escrito para un proyecto en particular, AVR-GCC convierte el programa en C en instrucciones en lenguaje ensamblador. Las bibliotecas de código AVR-libc incluidas contienen las direcciones de puertos y los nombres de los registros, la biblioteca para manejo de variables de punto flotante, macros específicos para las familias de microcontroladores AVR y el código para iniciar el microcontrolador. El compilador incluye documentación de las funciones que implementa. Este compilador es mantenido por una comunidad de código abierto, por lo que su uso es gratuito mientras se sigan las instrucciones de la licencia.

Capítulo III Resultados y discusión

Los resultados de la evaluación para los modelos grises en pronóstico de variables ambientales internas de un invernadero están directamente relacionados con la calidad y veracidad del valor numérico obtenido mediante los sensores designados, por lo tanto, es de gran importancia asegurar una correcta medición de las variables.

Acorde con los objetivos de la presente investigación, se propone la creación de un sistema embebido que pueda tomar muestra de los datos ambientales internos del microclima del invernadero, procesar estos datos mediante la ejecución del algoritmo predictivo seleccionado para su evaluación. Para poder obtener una conclusión adecuada sobre el rendimiento de los algoritmos predictivos es necesario contar con una base de datos confiable del sistema a predecir, en este sentido se propone como sistema de adquisición de datos una solución acorde con el concepto tecnológico de IoT o Internet de las Cosas. De esta manera se ofrece un monitoreo en tiempo real, la persistencia de los datos y el despliegue de datos de forma amigable para el usuario. Esta solución es por sí sola una herramienta para el monitoreo y obtención de datos internos del invernadero.

3.1 Sistema IoT

El monitoreo en tiempo real consta de un esquema de nodo sensor y puerta de enlace (Figura 14). Las funciones del nodo sensor están encaminadas a la recopilación de datos por sensores, su validación y posterior transmisión mediante radiofrecuencia en protocolo LoRa. La puerta de enlace tiene como tarea recibir las cadenas de datos de los nodos sensores, validar el buen estado de su información y enviarla mediante un dispositivo con acceso a la red global o Internet a un servidor en la nube para su

posterior despliegue, almacenamiento y análisis. Este modelo de solución hace referencia al concepto tecnológico de IoT aplicado a la agricultura.

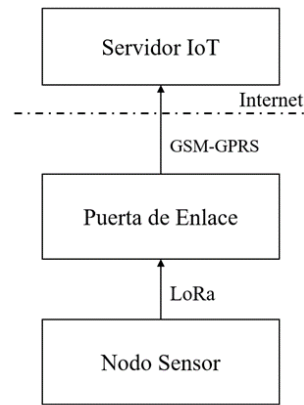


Figura 14 Esquema general del sistema

Fuente: Elaboración propia.

3.1.1 Nodo sensor

El sistema embebido del nodo sensor es alimentado por un sistema fotovoltaico de acumulador y regulador de carga, asegurando la operatividad del nodo sensor las 24 horas del día para un monitoreo continuo. Cuenta con el sensor SHT10 para medir la temperatura en aire T ($^{\circ}\text{C}$) y la humedad relativa HR (%), complementada con un encapsulado robusto, adecuado para las condiciones internas del invernadero e incluye transmisión de datos de manera digital. El siguiente elemento en la instrumentación es el sensor TSL2561 de luminosidad (Lux), variable que es de utilidad para tener una referencia sobre la incidencia lumínica en el área del invernadero.

Como unidad central de procesamiento se utiliza el microcontrolador de 8 bits y de núcleo AVR ATmega4809, destinado a realizar las consultas de los datos tomados por los sensores mediante un bus digital, validar los datos consultados y finalmente

proceder a llamar al módulo transceptor E32-915 coordinado mediante un bus UART para el envío de los datos. El módulo transceptor implementa el protocolo LoRa de largo alcance para la transmisión de los datos, este modelo en particular utiliza el circuito integrado SX1276 de Semtech®. El módulo está diseñado para transmitir los datos en radiofrecuencias de 868 MHz - 915 MHz.

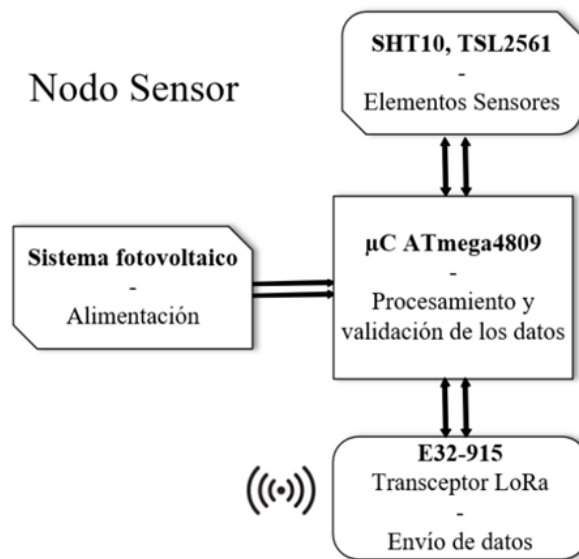


Figura 15 Esquema nodo sensor

Fuente: Elaboración propia.

Minimizar el consumo de potencia en dispositivos IoT es esencial, por esto el sistema tiene como una de sus prioridades el bajo consumo de energía, obteniendo el mejor rendimiento del sistema fotovoltaico, incluso en días sin mucha presencia de luz solar. Debido a lo anterior se implementa un RTC con cristal interno de 32.768 Khz incorporado por el fabricante en el microcontrolador, configurado en sus registros para funcionar como PIT (Periodic interrupt timer). El sistema es llevado inmediatamente al estado de reposo, para ser despertado únicamente por la rutina de servicio de interrupción detonada por el PIT. En dicha rutina se llaman las funciones con

sentencias y condiciones necesarias para leer los datos de los sensores a través de los buses digitales, llamar al módulo transceptor para el envío de los datos y finalmente regresar todos los componentes al estado de reposo. El tiempo configurado para detonar la rutina de toma de muestras y envío de datos mediante transceptor LoRa es de 60 segundos. El diagrama de flujo (figura 16) muestra una versión simplificada del proceso ejecutado por el nodo sensor.

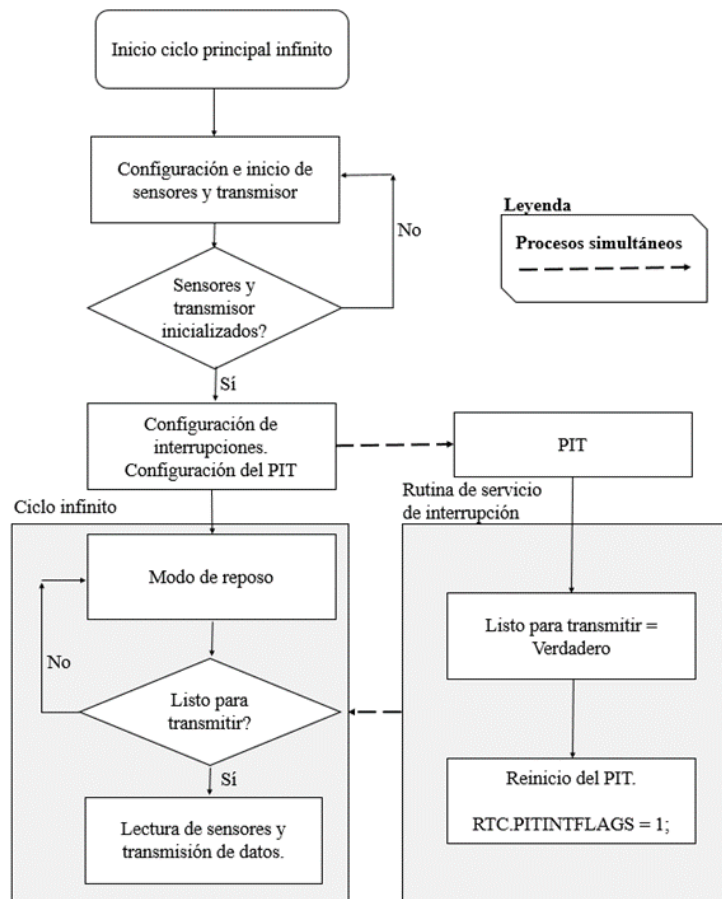


Figura 16 Diagrama de flujo nodo sensor

Fuente: Elaboración propia.

3.1.2 Puerta de enlace

El segundo sistema embebido está centrado en el módulo ESP-32-WROOM-32E, dispositivo que funciona como unidad de procesamiento. Conectado a un primer bus digital del módulo ESP-32, se encuentra un transceptor LoRa con una configuración de canal de transmisión y modulación igual a los del transceptor del nodo sensor, así la información transmitida por el nodo sensor será captada por el módulo transceptor en la puerta de enlace.

El módulo transceptor funciona de manera asíncrona por lo que puede recibir la información y almacenarla en un buffer interno en cuanto esta sea detectada, cuando el microcontrolador llame al módulo transceptor este indicará la presencia de datos en espera y procederá a enviarlos por medio del bus UART. Enseguida el microcontrolador procederá a verificar la integridad de la información.

Un segundo elemento coordinado por el microcontrolador mediante bus digital UART es la tarjeta de desarrollo con módulo GSM SIM808, cuya función es la de un modem para brindar de conectividad a internet al dispositivo mediante una torre celular de un proveedor de telefonía. Si los datos recibidos por el transceptor son correctos, el microcontrolador procede a ejecutar la rutina de coordinación del módulo SIM808, dicha rutina consiste en enviar una serie de comandos AT mediante UART, para indicar al módulo que debe conectarse a la red celular y solicitar la asignación de una dirección IP para posteriormente hacer una petición HTTP de tipo POST, para el envío de los datos a un servidor (Figura 17).

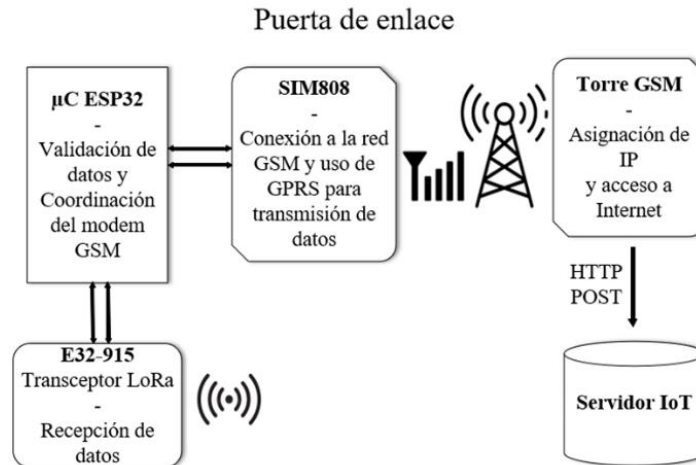


Figura 17 Esquema puerta de enlace

Fuente: Elaboración propia.

3.1.3 Servidor IoT

Para abordar una solución completa de monitoreo en tiempo real, almacenamiento y análisis de los datos, se opta por la alternativa del Cloud computing o computo en la nube, concepto que permite hacer uso de infraestructura informática robusta a través de Internet, en poco tiempo y sin necesidad de grandes recursos económicos. Se selecciona como proveedor de poder de cómputo en la nube a Azure de Microsoft, específicamente en su plataforma como servicio de Azure IoT Central (Microsoft, 2022), dedicado en acelerar el desarrollo de aplicaciones IoT, proporcionando una solución completa para monitoreo en tiempo real.

Se configura el servicio para recibir peticiones POST por medio de un enlace generado por la plataforma, específicamente para el dispositivo mediante la creación de una cadena de conexión única. Posteriormente se crea una plantilla de dispositivo en donde se le indica a la plataforma que datos se espera recibir y en qué formato se encuentran. La plataforma está configurada para almacenar cada dato recibido, la persistencia de los datos en la plataforma permite el futuro análisis de la dinámica térmica del invernadero.

Finalmente se configura la interfaz de despliegue de los datos recibidos, haciendo uso de las herramientas preconfiguradas del servicio para crear un Dashboard amigable con el usuario que exhiba las gráficas del comportamiento de las variables de temperatura, humedad relativa y luminosidad internas del invernadero.

3.1.4 Implementación física del sistema

La implementación física del nodo sensor (Figura 18) consta de una tarjeta fenólica perforada con todos los componentes electrónicos necesarios para su funcionamiento, además del módulo transceptor mencionado anteriormente.



Figura 18 Implementación nodo sensor.

Fuente: Elaboración propia.

El dispositivo es resguardado en una carcasa plástica con grado de protección IP66 (figura 18) de tal manera que se asegure su operatividad a mediano plazo en las

condiciones adversas del invernadero. Dentro de la carcasa se encuentra anexa una batería de gel y un dispositivo comercial de regulación de carga mediante panel solar. La carcasa cuenta con una glándula para el acceso externo de las conexiones de los sensores, la antena del módulo transceptor y el sistema fotovoltaico.

El dispositivo de nodo sensor es instalado dentro del centro geométrico del invernadero (figura 19), el panel solar es sostenido por un trípode de aluminio, el sensor de temperatura y humedad relativa es acoplado con su carcasa para evitar la saturación por radiación solar.



Figura 19 Nodo sensor desplegado

Fuente: Elaboración propia.

El sensor de luminosidad es expuesto aproximadamente a 50 cm del trípode (Figura 20) y es resguardado en una carcasa translúcida con protección IP66.



Figura 20 Sensor de luminosidad

Fuente: Elaboración propia.

La puerta de enlace se implementa en una tablilla fenólica perforada con el módulo de microcontrolador y módulo transceptor montado (Figura 21), incluye una línea de bornes para la segura conexión del modem GSM. Todo el dispositivo es resguardado en un contenedor rígido de plástico con incorporación de una glándula para el acceso exterior de la antena del transceptor y la alimentación.

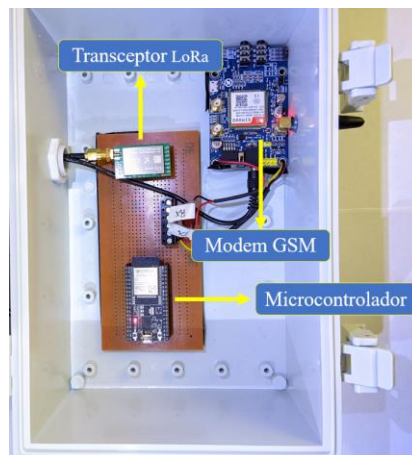


Figura 21 Implementación puerta de enlace.

Fuente: Elaboración propia.

La antena del modem GSM es expuesta al exterior mediante una perforación lateral del diámetro de la antena y es sellada respecto a la carcasa con silicón caliente.

La puerta de enlace es resguardada en una oficina dentro de las instalaciones de la empresa agrícola, cuenta con suministro eléctrico y se encuentra ubicada a 50 metros del invernadero.



Figura 22 Puerta de enlace instalada.

Fuente: Elaboración propia.

3.1.5 Aplicación y funcionamiento de plataforma IoT

Una vez que el nodo sensor es desplegado dentro del invernadero y la puerta de enlace se encuentra conectada a la red celular ésta enviará los datos validados a un servidor IoT mediante una URL generada por la plataforma. La plataforma recibe a

través de Internet una solicitud de la puerta de enlace con los datos de las variables ambientales cada 5 minutos aproximadamente, la plataforma está configurada para almacenar estos datos dentro de una base de datos no relacional, integrada de manera adicional dentro del servicio.

Cada entrada de datos cuenta con una estampa del tiempo en el que se recibió la solicitud, lo que permite filtrar y reorganizar la información por hora, día, mes y año. Dentro de la plataforma se cuenta con una herramienta para explorar y graficar los datos almacenados denominados como “Explorador de datos”, en la figura 23 se observa una captura de pantalla de la herramienta de exploración de datos dentro del portal WEB.

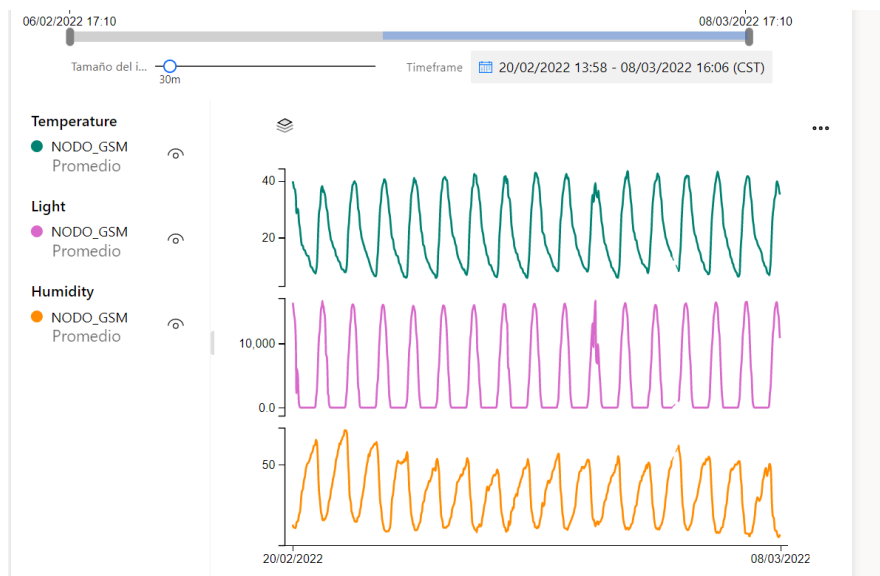


Figura 23 Despliegue de datos en plataforma IoT

Fuente: Elaboración propia.

Esta herramienta permite la exportación de la selección de datos a un archivo de tabla en formato de valores separados por comas o .csv como se observa en la figura 24, de esta manera es posible tener los datos a disposición de manera fuera de línea.

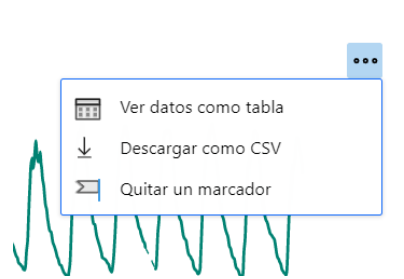


Figura 24 Exportación de datos

Fuente: Elaboración propia.

3.2 Desarrollo de algoritmos predictivos

Los desarrollos algorítmicos para los modelos grises están desarrollados en lenguaje C seleccionado con el objetivo de conseguir una mayor portabilidad entre sistemas y tener la capacidad futura de ejecutar el código en otra arquitectura o microcontrolador sin necesidad de muchas adecuaciones. Muchos fabricantes de microcontroladores modernos tienen excelentes herramientas y compiladores gratuitos en lenguaje C para sus productos. El estándar bajo el cual fue escrito el código es el de BARR group (BARR group, 2022), esto para reducir al máximo los posibles fallos en ejecución de código.

Mientras que parte del desarrollo destinado a la adquisición de datos, coordinación de comunicación y transferencia de datos está empleado en lenguaje C++ aprovechando los aportes de la comunidad de código abierto (Red Hat, 2019).

3.2.1 Desarrollo de algoritmo de operadores Grey.

La construcción del algoritmo parte de una propuesta modular en donde cada elemento u operación que compone el modelo grey tiene su propia función dentro del código, de esta manera a cada operador de generación grey como son el operador 1-AGO, 1-AGO inverso y la serie Z, les corresponderá una fracción de código en forma de función.

Para conseguir lo antes mencionado y llegar a un resultado estandarizado se crea el archivo de extensión .h o de cabecera según como se describe en (W. Kernighan & M. Ritchie, 1988, págs. 81-82), este archivo en conjunto con el archivo de extensión .c permiten exportar funciones de código y definiciones al proyecto compatible que lo llame mediante la directiva #include. De esta manera se llega a una solución de tipo “biblioteca” que puede ser compartida entre distintos proyectos.

3.2.1.1 Operador 1-AGO

Como se *observa* en la ecuación 1 el operador acumulativo es representado como:

$$x^{(1)}(k) = \sum_{i=1}^k x^{(0)}(i); k = 1, 2, \dots, n.$$

En donde $x^{(0)}$ es la serie de datos originales obtenidos de la medición numérica de las variables a evaluar dentro del desarrollo del modelo grey, $x^{(1)}$ es la serie de valores obtenidos mediante el resultado de operar 1-AGO sobre $x^{(0)}$.

Para iniciar la construcción de la biblioteca de código se declara la función *AGO_operator()* dedicada a contener el equivalente del operador de sumatoria $\sum_{i=1}^k x^{(0)}(i); k = 1, 2, \dots, n$ en código C mediante el empleo de un ciclo condicional de tipo *for*. Esta función recibe dos arreglos como parámetros, la serie de datos originales y la variable donde se almacenará el resultado de la operación 1-AGO.

Para validar la lógica creada dentro de la función se toman los datos del ejemplo 4.1 (Liu & Lin, Grey Systems Theory and Applications, 2010, págs. 113-116), en donde se propone la siguiente serie primitiva de datos:

$$X^{(0)} = (x^{(0)}(1), x^{(0)}(2), x^{(0)}(3), x^{(0)}(4), x^{(0)}(5), \dots) = (2.874, 3.278, 3.337, 3.390, 3.679)$$

La serie resultado después de aplicar el operador 1-AGO según el ejemplo se establece como:

$$X^{(1)} = (x^{(1)}(1), x^{(1)}(2), x^{(1)}(3), x^{(1)}(4), x^{(1)}(5), \dots) = (2.874, 6.152, 9.489, 12.879, 16.558)$$

Se utiliza la serie propuesta para evaluar la función *AGO_operator()* mediante el código del cual se aprecia un fragmento en la Figura 25, en donde *X0_DATA_SIZE* y *X1_DATA_SIZE* están definidos anteriormente por el preprocesador del compilador.

```

double x0_series[X0_DATA_SIZE] = {2.874, 3.278, 3.337, 3.390, 3.679};

double x1_series[X1_DATA_SIZE];

AGO_operator(x0_series, x1_series);

Serial.println("AGO OPERATOR RESULTS");
for (int i = 0; i < X1_DATA_SIZE; i++)
{
    Serial.println(x1_series[i], 6);
}

```

Figura 25 Función operador AGO.

Fuente: Elaboración propia.

La ejecución del código dentro del microcontrolador despliega por monitor serial el resultado de la operación como se puede observar en la figura 26. El resultado obtenido por el microcontrolador es idéntico al mencionado en la referencia del ejemplo, por lo que se asume que la función *AGO_operator()* cumple con lo esperado.

```

--- Miniterm on COM7 9600,8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
AGO OPERATOR RESULTS
2.874000
6.152000
9.489001
12.879001
16.558000

```

Figura 26 Monitor serie función 1-AGO

Fuente: Elaboración propia.

3.2.1.2 Operador de serie Z

Según la ecuación 5 antes definida, la serie Z tiene como entrada el arreglo acumulado mediante el operador 1-AGO y es definida como:

$$z^{(1)}(k) = \frac{x^{(1)}(k) + x^{(1)}(k + 1)}{2}$$

En continuación con la creación de la biblioteca de funciones de los operadores de generación grey, se propone la función `z_series()` que toma como argumentos el arreglo resultado del operador 1-AGO y un arreglo nuevo para almacenar los resultados.

Siguiendo con el ejemplo anteriormente visto en la sección 4.2.1.1, tenemos como resultado del operador acumulativo 1-AGO la siguiente serie de datos

$$X^{(1)} = (x^{(1)}(1), x^{(1)}(2), x^{(1)}(3), x^{(1)}(4), x^{(1)}(5), \dots) = (2.874, 6.152, 9.489, 12.879, 16.558)$$

De esta serie y como se muestra en el ejemplo 4.1 (Liu & Lin, Grey Systems Theory and Applications, 2010, págs. 113-116), el resultado de la serie Z será:

$$Z^{(1)} = (z^{(1)}(2), z^{(1)}(3), z^{(1)}(4), z^{(1)}(5),) = (4.513, 7.820, 11.184, 14.718)$$

Para evaluar el correcto funcionamiento de la función `z_series()` se utiliza el resultado de la función `AGO_operator()` y se imprimen los resultados de las operaciones mediante el monitor serie así como se observa en el fragmento de código de la figura 27, en donde `Z1_DATA_SIZE` está definido anteriormente por el preprocesador.

```

AGO_operator(x0_series, x1_series);
Serial.println("AGO OPERATOR RESULTS");
for (int i = 0; i < X1_DATA_SIZE; i++)
{
    Serial.println(x1_series[i], 6);
}
z_series(x1_series, z1_series);
Serial.println("Z SERIES RESULTS");
for (int i = 0; i < Z1_DATA_SIZE; i++)
{
    Serial.println(z1_series[i], 6);
}

```

Figura 27 Función serie Z.

Fuente: Elaboración propia.

La ejecución del código dentro del microcontrolador despliega por monitor serial el resultado de la operación como se puede observar en la figura 28. Con la información obtenida se puede validar el funcionamiento de la función `z_series()`.

```

--- Miniterm on COM6 9600,8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
AGO OPERATOR RESULTS
2.874000
6.152000
9.489001
12.879001
16.558000
Z SERIES RESULTS
4.513000
7.820500
11.184000
14.718501

```

Figura 28 Monitor serie función serie Z

Fuente: Elaboración propia.

3.2.1.3 Operador 1-AGO Inverso

En la ecuación 3 se observa que para restaurar una serie operada mediante el operador 1-AGO se utiliza el operador 1-AGO inverso definido como:

$$x^{(0)}(k) = x^{(1)}(k) - x^{(1)}(k - 1); k = 1, 2, \dots, n.$$

Para finalizar la creación de la biblioteca de código de los operadores de generación grey se propone la función *IAGO_operator()* que toma como argumentos la serie de datos resultado del operador 1-AGO y el arreglo donde se almacenará la serie restablecida.

Siguiendo con el ejemplo anteriormente visto en la sección 4.2.1.1, tenemos como resultado del operador acumulativo 1-AGO la siguiente serie de datos

$$X^{(1)} = (x^{(1)}(1), x^{(1)}(2), x^{(1)}(3), x^{(1)}(4), x^{(1)}(5),) = (2.874, 6.152, 9.489, 12.879, 16.558)$$

Como se muestra en el ejemplo 4.1 (Liu & Lin, Grey Systems Theory and Applications, 2010, págs. 113-116), el resultado de restablecer esta serie al orden de $X^{(0)}$ será:

$$X^{(0)} = (x^{(0)}(1), x^{(0)}(2), x^{(0)}(3), x^{(0)}(4), x^{(0)}(5),) = (2.874, 3.278, 3.337, 3.390, 3.679)$$

En este sentido, para evaluar la función *IAGO_operator()* se utiliza la serie $X^{(1)}$ como parámetro de entrada según como se observa en el fragmento de código en la figura 29 en donde *IAGO_DATA_SIZE* puede ser definido anteriormente por el preprocesador.

```
double x1_series[X1_DATA_SIZE] = {2.874, 6.152, 9.489, 12.879, 16.558};
double IAGO_series[IAGO_DATA_SIZE];

IAGO_operator(x1_series, IAGO_series);

Serial.println("IAGO SERIES RESULTS");
for (int i = 0; i < IAGO_DATA_SIZE; i++)
{
  Serial.println(IAGO_series[i], 6);
}
```

Figura 29 Función operador IAGO

Fuente: Elaboración propia.

La ejecución del código dentro del microcontrolador despliega por monitor serial la ejecución del código como se puede observar en la figura 30. Con la información arrojada se puede asumir el correcto funcionamiento de la función *IAGO_operator()*

```
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
IAGO SERIES RESULTS
2.874000
3.278000
3.337000
3.389999
3.679001
```

Figura 30 Monitor serie operador IAGO

3.2.2 Operadores de matrices

En el pronóstico de variables con modelos *grey* es habitual encontrarse que la estimación de los parámetros iniciales se realiza mediante una serie de operaciones matriciales que dan como resultado los parámetros de estimación por mínimos cuadrados lineales, así como se menciona anteriormente en la sección 2.2.5 en la ecuación número 27. Esta estimación implica la operación de matrices como la matriz transpuesta, el producto de dos matrices y la obtención de la matriz inversa, esto lo podemos observar a continuación en la ecuación número 27:

$$\{A\} = [[\theta]^T [\theta]]^{-1} [\theta]^T \{Y\}$$

Para complementar la biblioteca de código para la ejecución de modelos *grey* es necesario contar con funciones que puedan ejecutar dichas operaciones de matrices en tiempo real y desde el microcontrolador. Después de una búsqueda en los repositorios de código abierto se toma como referencia base el repositorio de Charlie Matlack con licencia GPL2.0, licencia que permite el uso comercial del código además de modificaciones (Matlack, 2018). El repositorio contiene una serie de funciones escritas en lenguaje C++ cuyo objetivo es satisfacer las operaciones de matrices del algebra lineal como son matriz transpuesta, el producto de dos matrices, la matriz inversa, suma y sustracción. Lo anterior se consigue mediante la utilización de variables de tipo puntero que realizan dentro de ciclos condicionales diferentes operaciones aritméticas en las matrices con las que se desea operar, estas se definen como arreglos de una o dos dimensiones dentro de la programación. Al evaluar el rendimiento de las funciones del repositorio se concluye que éstas cumplen con lo anticipado y se procede a adaptar el código para cumplir con los objetivos de esta investigación. Las funciones fueron llevadas a una biblioteca simplificada en código C para su implementación en el desarrollo de los modelos *grey*.

3.2.3 Algoritmo en código C del GM (1,1).

Una vez desarrolladas las bibliotecas de código para los operadores de generación grey además de los operadores de matrices, estos pueden ser utilizados como bloques de construcción para desarrollar los algoritmos en específico de los modelos de pronóstico grey.

Según como se describe anteriormente en la sección 2.3.1 el procedimiento para generar un pronóstico inicia con la serie de datos primitiva, dicha serie contiene los datos de las variables monitoreadas y sobre las cuales se genera el modelo grey.

El desarrollo simplificado del algoritmo procede de la siguiente manera:

1. Se procesa la serie primitiva mediante el llamado de la función *AGO_operator()*, obteniendo como resultado la serie acumulada o $X^{(1)}$.
2. Se procesa la serie acumulada mediante el llamado de la función *z_series()* obteniendo como resultado la serie de promedio de dos vecinos cercanos o $Z^{(1)}$.
3. Se utilizan las series de datos obtenidas en los pasos uno y dos para formar la matriz B mediante funciones especiales en el código.
4. Se estiman los parámetros de mínimos cuadrados lineales o \hat{a} mediante las operaciones de matrices mencionadas en la sección 2.2.5 utilizando la biblioteca de código mencionada en 4.2.2.
5. Se genera la fórmula de respuesta de tiempo (ecuación 32) mediante la sustitución de los parámetros a y b respectivamente.
6. Se obtienen los valores del pronóstico iterando la fórmula de respuesta en tiempo para k valores.
7. Se restaura la serie al orden original mediante la función *IAGO_operator()*.

3.2.4 Algoritmo en código C del RDGM (1, 3)

El segundo algoritmo de pronóstico mediante modelo grey y destinado a ser evaluado es el modelo recursivo discreto con entrada de dos variables, como se menciona anteriormente los bloques de código de operaciones grey también serán utilizados para la implementación en código de este algoritmo, este acercamiento modular permite generar el algoritmo RDGM (1, 3) añadiendo cambios puntuales como la modificación de la matriz B según la sección 2.3.4. Se desarrolla una función en código dedicada a generar la matriz apoyándose de las funciones para la generación de las series AGO y series Z denominada como *discrete_3var_MATRIX()*. La siguiente adición a la biblioteca de código es la generación del modelo del sistema según la ec. 45 mencionada anteriormente en el capítulo 2.3.4:

$$X_1^{(1)}(rp + k + 1) = \beta_1 X_1^{(1)}(rp + k) + \sum_{i=2}^n \beta_i X_i^{(1)}(k + 1) + u$$

Esta formulación es llevada a código C y ejecutada mediante el método *RDGM1_3()* tomando como parámetros de entrada la serie acumulada de la variable uno $X_1^{(1)}$, la series Z de la variable dos $Z_2^{(1)}$, la variable tres $Z_3^{(1)}$, y los parámetros estimados β_1 , β_2 y u . Con este contexto el desarrollo simplificado del algoritmo RDGM (1, 3) es de la siguiente manera:

1. Se procesan las serie primitivas de las dos variables mediante el llamado de la función *AGO_operator()*, obteniendo como resultado la series acumuladas $X_1^{(1)}$, $X_2^{(1)}$ y $X_3^{(1)}$
2. Se procesan la series acumuladas mediante el llamado de la función *z_series()* obteniendo como resultado la series de promedio de dos vecinos cercanos $Z_1^{(1)}$, $Z_2^{(1)}$ y $Z_3^{(1)}$

3. Se utilizan las series de datos obtenidas en los pasos uno y dos para formar la matriz B mediante funciones especiales en el código.
4. Se estiman los parámetros de mínimos cuadrados lineales o \hat{a} mediante las operaciones de matrices mencionadas en la sección 2.2.5 utilizando la biblioteca de código mencionada en 4.2.2.
5. Se implementa la función $RDGM(1, 3)$ con la ec. 44 utilizando los parámetros estimados en \hat{a} , la series $X_1^{(1)}$, $Z_2^{(1)}$ y $Z_3^{(1)}$
6. Se obtienen los valores del pronóstico iterando el modelo $RDGM(1, 3)$ para k valores.
7. Se restaura la serie al orden original mediante la función $IAGO_operator()$.

El anterior planteamiento se puede ver reflejado en el diagrama de flujo representado en figura 31.

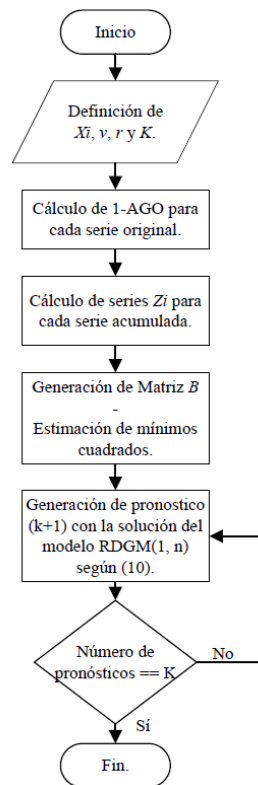


Figura 31 Diagrama de flujo del algoritmo $RDGM(1, 3)$.

Fuente: Elaboración propia.

3.3 Experimentos con datos reales

En esta sección se describen diferentes experimentos realizados para evaluar el ajuste de los pronósticos con respecto del valor real y la evaluación del tiempo de ejecución del algoritmo para ciertas condiciones definidas en el microcontrolador, utilizando los registros generados por el sistema de monitoreo IoT mencionado anteriormente. El sistema recopila los datos de las variables de temperatura interna, humedad relativa interna y luminosidad, el portal en la nube permite la exportación de la base de datos generada por el sistema IoT, estos datos son exportados en formato .csv y serán utilizados para los ejercicios descritos a continuación. Una muestra del formato de los datos exportados se observa en la figura 32.

| | A | B | C | D | E |
|----|------------------|------------------|--------------|-------------------------|----------------------------|
| 1 | Interval | Interval (UTC) | Temperature/ | Light/NODO_GSM.Promedio | Humidity/NODO_GSM.Promedio |
| 2 | 29/04/2022 02:00 | 29/04/2022 07:00 | 16.85 | 1 | 41.27 |
| 3 | 29/04/2022 02:30 | 29/04/2022 07:30 | 16.8236364 | 1 | 41.59727273 |
| 4 | 29/04/2022 03:00 | 29/04/2022 08:00 | 16.245 | 1 | 42.2 |
| 5 | 29/04/2022 03:30 | 29/04/2022 08:30 | 14.9290909 | 1 | 45.13 |
| 6 | 29/04/2022 04:00 | 29/04/2022 09:00 | 14.425 | 1 | 47.008 |
| 7 | 29/04/2022 04:30 | 29/04/2022 09:30 | 13.7809091 | 1 | 48.91090909 |
| 8 | 29/04/2022 05:00 | 29/04/2022 10:00 | 13.0127273 | 1 | 50.10272727 |
| 9 | 29/04/2022 05:30 | 29/04/2022 10:30 | 12.209 | 1 | 49.959 |
| 10 | 29/04/2022 06:00 | 29/04/2022 11:00 | 12.1209091 | 1 | 49.40181818 |
| 11 | 29/04/2022 06:30 | 29/04/2022 11:30 | 11.876 | 1 | 49.412 |
| 12 | 29/04/2022 07:00 | 29/04/2022 12:00 | 10.55 | 5.363636364 | 49.56181818 |
| 13 | 29/04/2022 07:30 | 29/04/2022 12:30 | 9.41818182 | 378.5454545 | 50.87454545 |
| 14 | 29/04/2022 08:00 | 29/04/2022 13:00 | 9.903 | 1216.2 | 49.417 |
| 15 | 29/04/2022 08:30 | 29/04/2022 13:30 | 12.3045455 | 2687.181818 | 44.41090909 |
| 16 | 29/04/2022 09:00 | 29/04/2022 14:00 | 17.139 | 4600 | 35.878 |
| 17 | 29/04/2022 09:30 | 29/04/2022 14:30 | 21.0272727 | 6439.454545 | 30.09727273 |
| 18 | 29/04/2022 10:00 | 29/04/2022 15:00 | 26.0654545 | 8566.636364 | 25.13363636 |
| 19 | 29/04/2022 10:30 | 29/04/2022 15:30 | 31.202 | 10860.5 | 20.662 |
| 20 | 29/04/2022 11:00 | 29/04/2022 16:00 | 33.6954545 | 12735.09091 | 16.25090909 |
| 21 | 29/04/2022 11:30 | 29/04/2022 16:30 | 33.6281818 | 14662.72727 | 14.3 |
| 22 | 29/04/2022 12:00 | 29/04/2022 17:00 | 35.8072727 | 16480.81818 | 13 |
| 23 | 29/04/2022 12:30 | 29/04/2022 17:30 | 38.5572727 | 18123.09091 | 11.06 |
| 24 | 29/04/2022 13:00 | 29/04/2022 18:00 | 39.3588889 | 19649.11111 | 9.91 |

Figura 32 Ejemplo de datos exportados en extensión .csv

Fuente: Elaboración propia.

3.3.1 Diseño del experimento

Este procedimiento en particular se propone para generar el pronóstico de la temperatura interna de un invernadero en la hora siguiente (60 minutos en el futuro) mediante la ejecución en un microcontrolador del software del RDGM (1, 3). La propuesta es construir el modelo utilizando los datos de tres variables, temperatura interna °C, humedad relativa interna en porcentaje RH % y luminosidad interna en Lux. Generando los conjuntos de datos originales con cinco muestras de cada variable tomadas en intervalos de 30 minutos. Dentro del contexto del RDGM (1, 3) los datos de la temperatura interna serán $X_1^{(0)}$, los datos de humedad relativa interna serán $X_2^{(0)}$ y los datos de luminosidad interna serán $X_3^{(0)}$. Para obtener el pronóstico de la temperatura interna del invernadero se completan los parámetros de $Z_2^{(1)}$ y $Z_3^{(1)}$ para alimentar la solución del modelo RDGM(1, 3) representado en (48) sustituyendo en la ecuación con información correspondiente del intervalo siguiente pero del día anterior al de la generación del pronóstico, esto significa incluir la información de las tendencias internas del invernadero del día anterior para generar un pronóstico del día actual, respetando la hora en la que se tomó la muestra pero retrocediendo un día, aprovechando así la estacionalidad que estos sistemas presentan.

Para procesar los datos obtenidos y generar un pronóstico se utiliza una tarjeta de desarrollo diseñada alrededor del microcontrolador ATmega4809 figura 33, este sistema embebido fue integrado para contener las conexiones de alimentación, de programación, módulo de radiofrecuencia LoRa, conexión de tipo USB con un convertidor serial a USB, además de incluir una bornera para la conexión directa de los sensores seleccionados. Este prototipo permitirá evaluar el algoritmo en un sistema objetivo y podrá ser utilizado en el campo de experimentación de ser necesario.

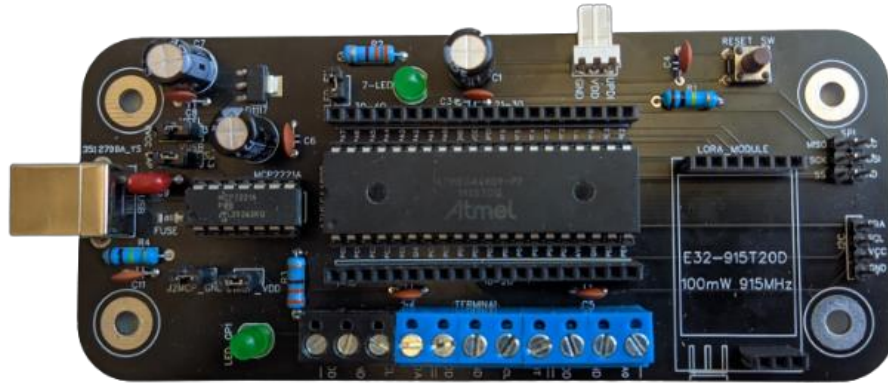


Figura 33 Tarjeta de desarrollo ATmega480.

Fuente: Elaboración propia.

El PCB de la tarjeta fue diseñado dentro de la unidad de investigación (UPIDET), para satisfacer los objetivos propuestos, utilizando software gratuito y manufacturándolo en un servicio automático de alta tecnología en la República Popular China.

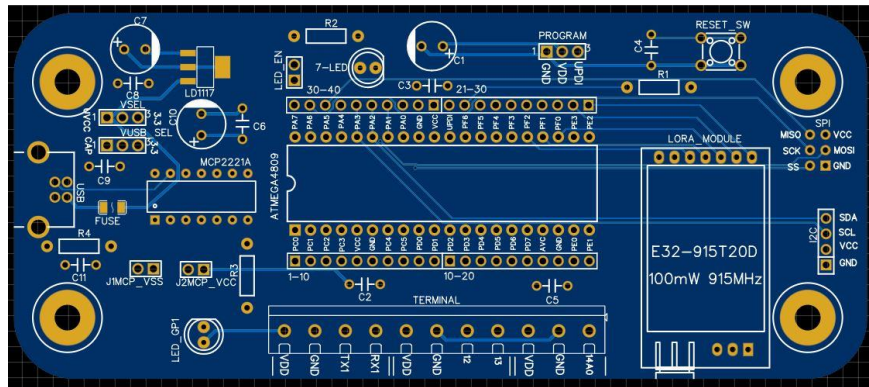


Figura 34 PCB Tarjeta de desarrollo ATmega4809.

Fuente: Elaboración propia.

La implementación del software para el microcontrolador se realiza mediante herramientas de código abierto y el código en C es compilado con el GCC AVR 8-Bit Toolchain en su versión 3.7.0. Para observar la información de los resultados se implementan sentencias para la impresión de la información a través de un puerto serial mediante de un convertidor UART a USB. La lectura de los resultados es realizada por medio de un software de terminal ejecutando en un computador personal con la tarjeta de desarrollo conectada mediante un puerto USB.

El código por ejecutar en el microcontrolador inicia definiendo los datos de entrada, el número de datos para construir el modelo y el número de pronósticos, según como se menciona en el desarrollo de los algoritmos, como se observa en la figura 37. El arreglo `x0_var1` se declara con los datos de temperatura, los arreglos de datos `x0_var2` y `x0_var3` de las variables de humedad relativa y luminosidad se construyen a partir de los cinco datos actuales concatenando con los dos datos de los intervalos correspondientes del día anterior.

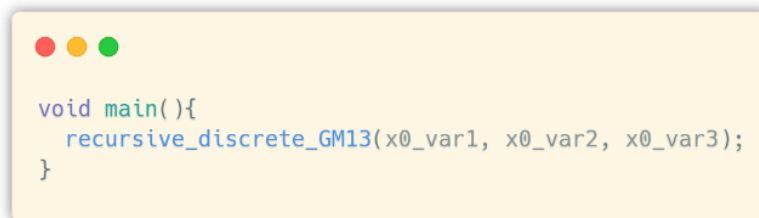
```
#define X0_DATA_SIZE 5
#define DISCRETE_FORECASTS 2

double x0_var1[X0_DATA_SIZE] = {
    43.341,
    40.876,
    38.505,
    36.973,
    34.976,
};
double x0_var2[(X0_DATA_SIZE + DISCRETE_FORECASTS)] = {
    5.780,
    5.875,
    7.633,
    8.083,
    9.814,
    8.905,
    10.520,
};
double x0_var3[(X0_DATA_SIZE + DISCRETE_FORECASTS)] = {
    11603.0,
    8446.0,
    6664.0,
    5067.0,
    3098.0,
    1368.0,
    337.0,
};
```

Figura 35 Declaración de datos de entrada del algoritmo.

Fuente: Elaboración propia.

Una vez declarados los elementos indicados se procede con el llamado de la función desarrollada en la biblioteca de código para la ejecución de modelo RDGM (1, 3) ejemplificada en la figura 36, Introduciendo como parámetros los arreglos de las variables anteriormente definidas.

A screenshot of a code editor window with a yellow background and three colored window control buttons (red, yellow, green) in the top left corner. The code is written in a light blue font and shows a C-style function call within a main function.

```
void main(){  
    recursive_discrete_GM13(x0_var1, x0_var2, x0_var3);  
}
```

Figura 36 Llamado de la función del modelo RDGM (1, 3).

Fuente: Elaboración propia.

Para determinar el tiempo de ejecución del modelo se suprimen las sentencias de impresión de datos por monitor serial del programa y se incluyen sentencias de manipulación de registros de entradas y salidas para cambiar de estado uno de los pines de uso general por cada ejecución del algoritmo ejemplificado en este experimento.

Enseguida se captura la salida de canal de un osciloscopio conectado a la salida del pin de uso general del microcontrolador ATmega4809 en la tarjeta de desarrollo figura 37, todos los experimentos son realizados con una configuración del oscilador interno del microcontrolador a 16 MHz y alimentándolo 3.3 Volts.



Figura 37 Banco de pruebas del sistema embebido.

Fuente: Elaboración propia.

3.3.2 Experimento 1

Del repositorio de datos obtenidos por el sistema IoT sobre las variables internas del invernadero se extrae el conjunto descrito en la tabla 4 y 5.

Tabla 4 Datos del día 03 de mayo del 2022.

| Fecha y hora | Temperatura °C | Humedad relativa %RH | Luminosidad Lux |
|------------------|----------------|----------------------|-----------------|
| 03/05/2022 17:00 | 43.341 | 5.780 | 11603 |
| 03/05/2022 17:30 | 40.876 | 5.875 | 8446 |
| 03/05/2022 18:00 | 38.505 | 7.633 | 6664 |
| 03/05/2022 18:30 | 36.973 | 8.083 | 5067 |
| 03/05/2022 19:00 | 34.976 | 9.814 | 3098 |

Fuente: Elaboración propia.

Tabla 5 Datos del día 02 de mayo del 2022.

| Fecha y hora | Humedad relativa %RH | Luminosidad Lux |
|------------------|-------------------------|--------------------|
| 02/05/2022 19:30 | 8.905 | 1368 |
| 02/05/2022 20:00 | 10.520 | 337 |

Fuente: Elaboración propia.

Con los datos mencionados anteriormente se declaran tres arreglos de variables y se ejecuta el algoritmo a evaluar. En la figura 38. se muestran los resultados de ejecutar el modelo grey RDGM (1, 3) resaltando la comparativa entre los valores reales de temperatura con los obtenidos mediante la ejecución del modelo, los valores representativos desde las 17:00 horas hasta las 19:00 horas fueron utilizados para construir el modelo, mientras que los valores de las 19:30 horas y las 20:00 horas son los pronósticos generados a partir del modelo.

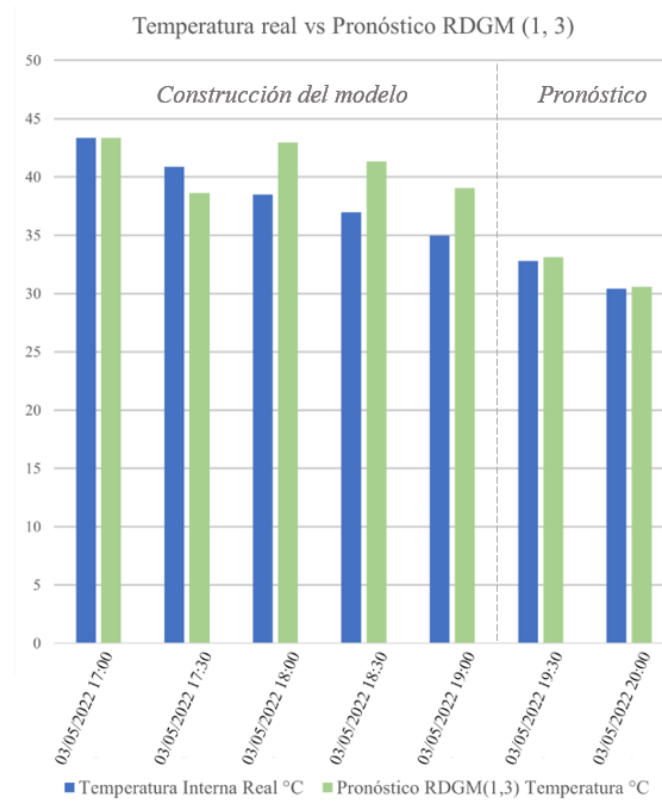


Figura 38 Resultados del experimento uno

Fuente: Elaboración propia.

En la Tabla 6 los pronósticos generados se contrastan con los datos de la estación de medición interna del invernadero.

Tabla 6 Pronóstico del experimento uno

| Fecha y hora | Temperatura Real ° C | Temperatura Pronostico ° C | Error relativo % |
|------------------|----------------------|----------------------------|------------------|
| 02/05/2022 19:30 | 32.802 | 33.136 | 1.02% |
| 02/05/2022 20:00 | 30.415 | 30.599 | 0.61% |

Fuente: Elaboración propia.

Para la prueba de tiempo de ejecución se realiza por medio de un monitoreo de una salida digital en el microcontrolador como se mencionó anteriormente. En la figura 39, el cambio de estado en la salida digital sugiere que el algoritmo toma 6.18 milisegundos por ejecución.

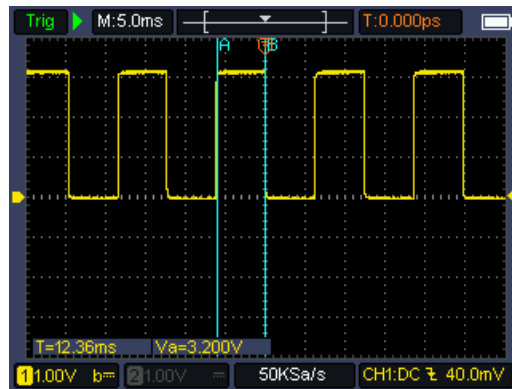


Figura 39 Tiempo de ejecución del experimento uno.

Fuente: Elaboración propia.

3.3.3 Experimento 2

Del repositorio de datos obtenidos por el sistema IoT sobre las variables internas del invernadero se extrae el conjunto descrito en la tabla 7 y 8.

Tabla 7 Datos del día 01 de mayo del 2022.

| Fecha y hora | Temperatura °C | Humedad relativa %RH | Luminosidad Lux |
|------------------|----------------|----------------------|-----------------|
| 01/05/2022 11:00 | 31.917 | 16.650 | 12296 |
| 01/05/2022 11:30 | 33.494 | 11.357 | 1419 |
| 01/05/2022 12:00 | 35.318 | 10.127 | 15947 |
| 01/05/2022 12:30 | 36.363 | 8.956 | 17448 |
| 01/05/2022 13:00 | 36.839 | 8.800 | 18650 |

Fuente: Elaboración propia.

Tabla 8 Datos del día 30 de abril del 2022.

| Fecha y hora | Humedad relativa %RH | Luminosidad Lux |
|------------------|-------------------------|--------------------|
| 30/04/2022 13:30 | 11.257 | 19606 |
| 30/04/2022 14:00 | 10.836 | 20154 |

Fuente: Elaboración propia.

Con los datos mencionados anteriormente se declaran tres arreglos de variables y se ejecuta el algoritmo a evaluar. En la figura 40. se muestran los resultados de ejecutar el modelo grey RDGM (1, 3) resaltando la comparativa entre los valores reales de temperatura con los obtenidos mediante la ejecución del modelo, los valores representativos desde las 11:00 horas hasta las 13:00 horas fueron utilizados para construir el modelo, mientras que los valores de las 13:30 horas y las 14:00 horas son los pronósticos generados a partir del modelo.

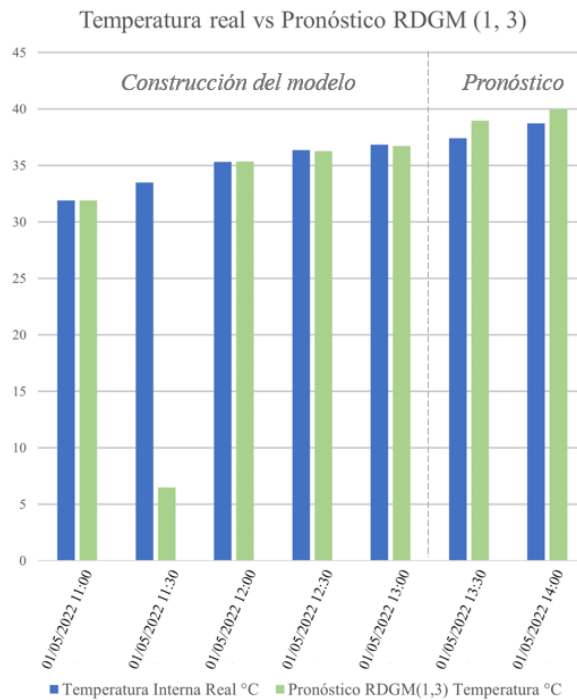


Figura 40 Resultados del experimento dos.

Fuente: Elaboración propia.

En la Tabla 9 los pronósticos generados se contrastan con los datos de la estación de medición interna del invernadero.

Tabla 9 Pronóstico del experimento dos

| Fecha y hora | Temperatura Real ° C | Temperatura Pronostico ° C | Error relativo % |
|------------------|-------------------------|-------------------------------|------------------|
| 01/05/2022 13:30 | 37.422 | 38.961 | 4.11% |
| 01/05/2022 14:00 | 38.740 | 39.960 | 3.14% |

Fuente: Elaboración propia.

Para la prueba de tiempo de ejecución se realiza por medio de un monitoreo de una salida digital en el microcontrolador como se mencionó anteriormente. En la figura 41, el cambio de estado en la salida digital sugiere que el algoritmo toma 10.30 milisegundos por ejecución.

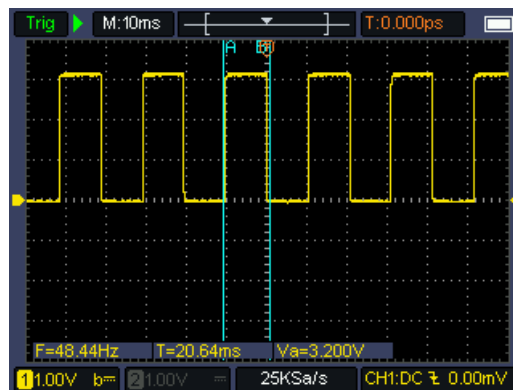


Figura 41 Tiempo de ejecución del experimento dos.

Fuente: Elaboración propia.

3.3.4 Experimento 3

Del repositorio de datos obtenidos por el sistema IoT sobre las variables internas del invernadero se extrae el conjunto descrito en la tabla 10 y 11.

Tabla 10 Datos del día 02 de mayo del 2022.

| Fecha y hora | Temperatura °C | Humedad relativa %RH | Luminosidad Lux |
|------------------|----------------|----------------------|-----------------|
| 02/05/2022 10:00 | 25.659 | 24.000 | 8310 |
| 02/05/2022 10:30 | 28.852 | 19.721 | 10296 |
| 02/05/2022 11:00 | 30.512 | 16.467 | 12089 |
| 02/05/2022 11:30 | 33.660 | 13.405 | 14032 |
| 02/05/2022 12:00 | 36.087 | 10.601 | 15840 |

Fuente: Elaboración propia.

Tabla 11 Datos del día 01 de mayo del 2022.

| Fecha y hora | Humedad relativa %RH | Luminosidad Lux |
|------------------|----------------------|-----------------|
| 01/05/2022 13:30 | 8.956 | 17448.090 |
| 01/05/2022 14:00 | 8.800 | 18650.909 |

Fuente: Elaboración propia.

Con los datos mencionados anteriormente se declaran tres arreglos de variables y se ejecuta el algoritmo a evaluar. En la Figura 42. se muestran los resultados de ejecutar el modelo grey RDGM (1, 3) resaltando la comparativa entre los valores reales de temperatura con los obtenidos mediante la ejecución del modelo, los valores representativos desde las 10:00 horas hasta las 12:00 horas fueron utilizados para construir el modelo, mientras que los valores de las 12:30 horas y las 13:00 horas son los pronósticos generados a partir del modelo.

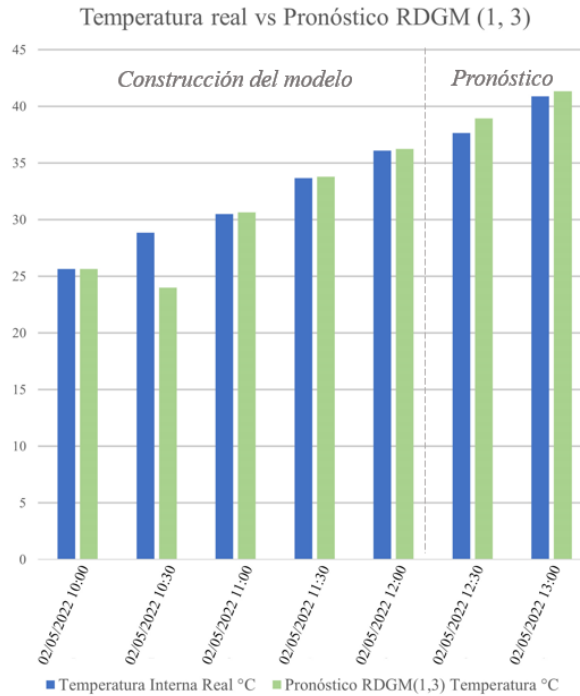


Figura 42 Resultados del experimento tres.

Fuente: Elaboración propia.

En la Tabla 12. los pronósticos generados se contrastan con los datos de la estación de medición interna del invernadero.

Tabla 12 Pronóstico del experimento tres

| Fecha y hora | Temperatura Real ° C | Temperatura Pronostico ° C | Error relativo % |
|------------------|----------------------|----------------------------|------------------|
| 02/05/2022 12:30 | 37.422 | 38.932 | 3.376 % |
| 02/05/2022 13:00 | 40.901 | 41.322 | 1.031 % |

Fuente: Elaboración propia.

Para la prueba de tiempo de ejecución se realiza por medio de un monitoreo de una salida digital en el microcontrolador como se mencionó anteriormente. En la Figura 43, el cambio de estado en la salida digital sugiere que el algoritmo toma 10.8 milisegundos por ejecución.

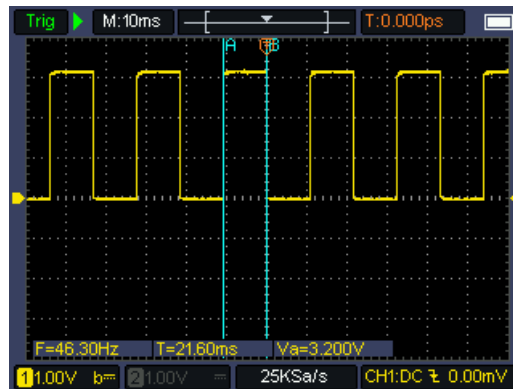


Figura 43 Tiempo de ejecución del experimento tres.

Fuente: Elaboración propia.

3.3.5 Experimento 4

Este experimento es desarrollado tomando los datos de la madrugada del día 30 de abril del 2022, del repositorio de datos obtenidos por el sistema IoT sobre las variables internas del invernadero se extrae el conjunto descrito en la tabla 13 y 14. Nótese que el sensor de luminosidad devuelve un valor unitario en la ausencia total de luz.

Tabla 13 Datos de la madrugada del 30 de abril del 2022

| Fecha y hora | Temperatura °C | Humedad relativa %RH | Luminosidad Lux |
|------------------|----------------|----------------------|-----------------|
| 30/04/2022 01:00 | 19.341 | 36.498 | 1 |
| 30/04/2022 01:30 | 18.517 | 38.173 | 1 |
| 30/04/2022 02:00 | 17.712 | 41.388 | 1 |
| 30/04/2022 02:30 | 17.104 | 43.252 | 1 |
| 30/04/2022 03:00 | 16.708 | 43.735 | 1 |

Fuente: Elaboración propia.

Tabla 14 Datos de la madrugada del 29 de abril del 2022.

| Fecha y hora | Humedad relativa %RH | Luminosidad Lux |
|------------------|-------------------------|--------------------|
| 29/04/2022 03:30 | 41.597 | 1 |
| 29/04/2022 04:00 | 42.200 | 1 |

Fuente: Elaboración propia.

Con los datos mencionados anteriormente se declaran tres arreglos de variables y se ejecuta el algoritmo a evaluar. En la figura 44. se muestran los resultados de ejecutar el modelo grey RDGM (1, 3) resaltando la comparativa entre los valores reales de temperatura con los obtenidos mediante la ejecución del modelo, los valores representativos desde las 01:00 horas hasta las 03:00 horas fueron utilizados para construir el modelo, mientras que los valores de las 03:30 horas y las 04:00 horas son los pronósticos generados a partir del modelo.

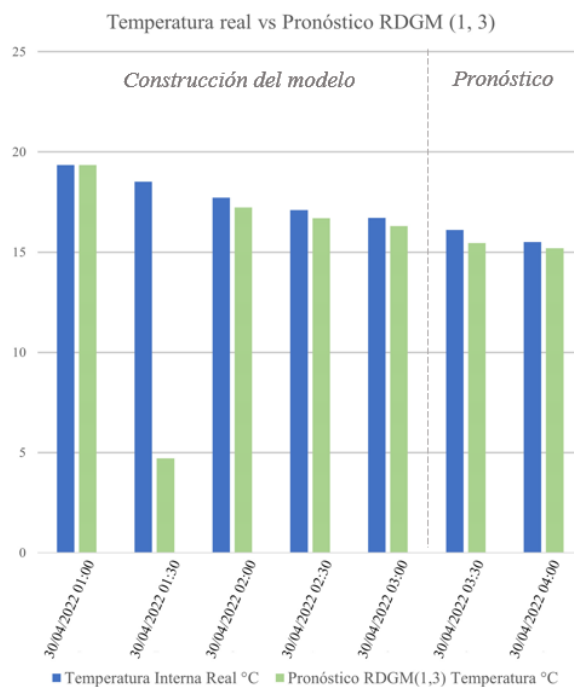


Figura 44 Resultados del experimento cuatro.

Fuente: Elaboración propia.

En la Tabla 15 los pronósticos generados se contrastan con los datos de la estación de medición interna del invernadero.

Tabla 15 Pronóstico del experimento cuatro

| Fecha y hora | Temperatura Real ° C | Temperatura Pronostico ° C | Error relativo % |
|------------------|-------------------------|-------------------------------|------------------|
| 29/04/2022 03:30 | 16.114 | 15.453 | 4.102 % |
| 29/04/2022 04:00 | 15.498 | 15.198 | 1.933 % |

Fuente: Elaboración propia.

Para la prueba de tiempo de ejecución se realiza por medio de un monitoreo de una salida digital en el microcontrolador como se mencionó anteriormente. En la figura 45, el cambio de estado en la salida digital sugiere que el algoritmo toma 10.65 milisegundos por ejecución.

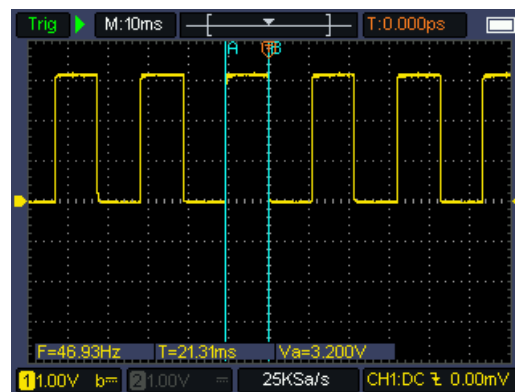


Figura 45 Tiempo de ejecución del experimento cuatro.

Fuente: Elaboración propia.

Capítulo IV Conclusiones y recomendaciones

La premisa de utilizar la teoría de sistemas grey en la predicción de fenómenos climáticos ha sido explorada con anterioridad como se observa en los antecedentes de la presente investigación, bajo estas observaciones se consideró como oportunidad de implementación el sistema térmico interno de un invernadero, dicho sistema presenta estacionalidad, el efecto invernadero en conjunto con la transmisión de calor ofrecida por el suelo en forma de radiación hacen que sea un sistema que cambia sus condiciones de manera relativamente lenta en comparación con otros sistemas dinámicos con una constante de tiempo relativamente pequeña. El modelado grey de la temperatura interna en el invernadero fue puesta a prueba según Yin, J. (2014) utilizando un modelo grey distinto al propuesto en esta investigación, siendo la principal diferencia la implementación multivariable que contempla más elementos que aporten al modelado del sistema.

En contraste la presente investigación puso a prueba el modelo recursivo discreto de una ecuación y tres variables, con cinco periodos de muestreo de 30 minutos para permitir que el modelo pueda seguir una tendencia definida, en esta ocasión se consideró la variable de temperatura para ser pronosticada a partir de los valores de temperatura interna, humedad relativa interna y luminosidad.

Para poner a prueba la metodología fue fundamental contar con datos de un sistema objetivo real, por lo que se desarrolló un sistema autónomo de adquisición de datos remoto bajo el concepto del internet de las cosas, que facilitó la adquisición remota de datos internos de un invernadero pasivo de producción de tomate, dicho sistema embebido fue diseñado e implementado dentro de la Unidad de Posgrado, Investigación y Desarrollo Tecnológico, (UPIDET).

La puesta a prueba de los datos demostró que el modelo RDGM (1, 3) se ajusta a la tendencia térmica del invernadero de manera relativamente aceptable y con un gasto computacional bajo al ser implementado en un sistema limitado como es un microcontrolador de 8 bits y con una respuesta menor a los 15 milisegundos, esto

permite crear nodos de bajo costo que generen su propio pronóstico sin depender de un servidor central o una unidad de cómputo con más capacidad de cálculo.

Sin embargo, existen casos en los cuales el modelo no consigue converger de manera adecuada, otorgando pronósticos con grados de error altos, por lo que quedan muchas pruebas por realizar, así como crear nuevos experimentos con diferentes variables relevantes internas del sistema del invernadero, probar con diferentes tiempos de muestreo y número de datos para la construcción del modelo. El principal aporte de esta investigación es el desarrollo del software portable a diferentes sistemas que permitirá explorar la utilización del algoritmo en diversas situaciones.

Obteniendo la combinación de variables, tiempo de muestreo y número de datos para construcción del modelo correctos, se conseguirá ofrecer información de valor para que el agricultor pueda adelantarse a tomar decisiones que cambien la tendencia térmica del invernadero, así mantener el producto en las condiciones óptimas reduciendo el porcentaje de merma.

Referencias

Mishra, S., Chandra Sahoo, N., & Narayan Senapati, R. (2016). Convolution Integral based Multivariable Grey Prediction . *IEEE International Conference on Power and Energy* , 663-667.

Ameloot, T., Van Torre, P., & Rogier, H. (2021). Variable Link Performance Due to Weather Effects in a Long-Range, Low-Power LoRa Sensor Network. *Sensors*, 3128-3149.

Augustin , A., Yi, J., Clausen, T., & Mark Townsley, W. (2016). A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. *Sensors*, 1466-1484.

BARR group. (02 de 03 de 2022). *BARR group*. Obtenido de <https://barrgroup.com/embedded-systems/books/embedded-c-coding-standard>

Bouguera, T., Diouris, J.-F., Chaillout, J.-J., & Andrieux, G. (2018). Energy consumption modeling for communicating sensors using LoRa technology. *IEEE Conference on Antenna Measurements & Applications (CAMA)*, (págs. 1-4).

Boylestad, R. L. (2004). *Introducción al análisis de circuitos*. Ciudad de México: PEARSON EDUCACIÓN.

Boylestad, R. L., & Nashelsky, L. (2003). *Electrónica: teoría de circuitos y dispositivos electrónicos*. Ciudad de México: PEARSON EDUCACIÓN.

Chapra, S., & Canale, R. (2015). *Métodos numéricos para ingenieros*. Ciudad de México: McGraw-Hill.

Chen, Y., Nu, L., & Wu, L. (2020). Forecasting the Agriculture Output Values in China Based on Grey Seasonal Model. *Mathematical Problems in Engineering*, 1-10.

Chengdu Ebyte Electronic Technology. (2018). E32-915T20D User Manual.

Codeluppi, G., Cilfone, A., Davoli, L., & Ferrari, G. (2020). LoRaFarM: A LoRaWAN-Based Smart Farming Modular IoT Architecture. *Sensors*, 2028-2048.

Darvishi Salookolaei, D., Liu, S., & Babaei, P. (2017). Application of Grey System Theory in Rainfall Estimation. *Control and Optimization in Applied Mathematics*, 15-32.

Ghribi, B., & Logrippo, L. (2000). Understanding GPRS: The GSM Packet Radio Service. *Computer Networks*, 763-779.

Hrisko, J. (2020). Capacitive Soil Moisture Sensor Theory, Calibration, and Testing. doi:10.13140/RG.2.2.36214.83522.

Julong, D. (1989). Introduction to Grey System Theory. *The Journal of Grey System*, 1-24.

Karimi Khajehghiasi, M., & Alinezhad, A. (2019). Forecast of Climatologically Events Using Improved Grey Model (Case Study: Qazvin Province Climatology). *Iranian Journal of Optimization*, 49-56.

Leelavinodhan, P., Vecchio, M., Antonelli, F., Maestrini, A., & Brunelli, D. (2021). Design and Implementation of an Energy-Efficient Weather Station for Wind Data Collection. *Sensors*, 3831-3849.

Liu, S., & Lin, Y. (2010). *Grey Systems Theory and Applications*. Berlin: Springer.

Liu, S., Forrest, J., & Yang, Y. (2012). A brief introduction to grey system theory. *Grey Systems: Theory and Application*, 89-104.

Liu, S., Yang, Y., & Forrest, J. (2017). *Grey Data Analysis*. Singapore: Springer Singapore.

Ma, X., & Liu, Z. (2017). The GMC(1, n) model with optimized parameters and its application. *Journal of Grey System*, 122-138.

Ma, X., & Liu, Z.-b. (2015). Research on the novel recursive discrete multivariate grey prediction model and its applications. *Applied Mathematical Modelling*, 4876-4890.

Matlack, C. (20 de Julio de 2018). *MatrixMath*. Obtenido de GitHub.com: <https://github.com/eecharlie/MatrixMath>

Microsoft. (01 de 03 de 2022). *Introducción a Azure IoT Central*. Obtenido de Introducción a Azure IoT Central: <https://docs.microsoft.com/es-es/azure/iot-central/core/overview-iot-central>

Muqtadir, A., Suryono, & Vicensius Gunawan. (2016). The Implementation of Grey Forecasting Model for Forecast Result's Food Crop Agricultural. *Scientific Journal of Informatics*, 159-166.

Oo, Z., & Phyu, S. (2019). Microclimate Prediction Using Cloud Centric Model Based on IoT Technology for Sustainable Agriculture. *International Conference on Computer and Communication Systems*, 660-663.

Parvez, S. &. (2016). A Novel Design and Implementation of Electronic Weather Station and Weather Data Transmission System Using GSM Network. *WSEAS Transactions on Circuits and Systems*, 21-34.

Red Hat. (24 de 10 de 2019). *Red Hat*. Obtenido de <https://www.redhat.com/es/topics/open-source/what-is-open-source>

Semtech Corporation. (Diciembre de 2019). LoRa® and LoRaWAN®: A Technical Overview. Camarillo, California , Estados Unidos.

Senapati, R. N., Sahoo, N. C., & Mishra, S. (2016). Convolution integral based multivariable grey prediction model for solar energy generation forecasting. 2016 IEEE International Conference on Power and Energy (PECon). doi:10.1109/pecon.2016.7951643

SIAP (Servicio de Información Agroalimentaria y Pesquera), Sistema de Información Agroalimentaria de Consulta, consultado el 19 de febrero de 2022, de <https://www.gob.mx/siap/documentos/siacon-ng-161430>.

Song, D., & Ruojin, L. (2020). A New Multivariable Grey Convolution Model Based on Simpson's Rule and Its Applications. *Complexity*, 14.

Tien, T.-L. (2005). The indirect measurement of tensile strength of material by the grey prediction model GMC(1, n). *Measurement Science and Technology*, 16, 1322–1328.

W. Kernighan, B., & M. Ritchie, D. (1988). *The C Programming Language*. New Jersey: PRENTICE HALL.

Yang, Y., & Wang, X. (2017). Fractional Accumulation GM(0,N) in the Ambient Air Quality Analysis. *Chinese Control Conference (CCC)*, 10424-10428.

Yin, J. (2014). Prediction of Temperature and Relative Humidity in Greenhouse Based on the combination of DGM(1,1) and linear model. *IEEE Workshop on Advanced Research and Technology in Industry Applications*, 1149-1152.

Ying-Yuan, C., Hao-Tien, L., & Hsiow-Ling, H. (2019). Time series interval forecast using GM(I,1) and NGBM(I, 1) models. *Soft Computing*, 1541-1555.

Zeng, B., Luo, C., Liu, S., Bai, Y., & Li, C. (2016). Development of an optimization method for the GM(1,N) model. *Engineering Applications of Artificial Intelligence*, 353-362.

Anexos

Publicaciones y participaciones de congresos

Zamudio Carbajal, R. et al (2022). Diseño de sistema embebido para monitoreo IoT del microclima interno de un invernadero con transceptores LoRa y modem GSM.



Universidad Autónoma de Baja California
Facultad de Ciencias Químicas e Ingeniería



VIII Congreso Internacional de Investigación Tijuana
Otorga la presente:

CONSTANCIA

a:

*Ricardo-Zamudio Carbajal, José Antonio-Martínez Rivera,
Francisco Javier-Godínez García, Eduardo-Gamero Inda,
Rubén-Guerrero Rivera*

Por su participación con el trabajo titulado: **"Diseño de sistema embebido para monitoreo IoT del microclima interno de un invernadero con transceptores LoRa y modem GSM"**, mediante la presentación oral en el 5th Conference on Electronics Engineering en el marco del Congreso Internacional de Investigación Tijuana que se llevó a cabo del 25 al 28 de abril del 2022, en la ciudad de Tijuana, B. C. México.


Alejandra Serrano Trujillo
Track Chair de COEE


Dr. Eduardo A. López Malconado
Coordinador General de C2T 2022





EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO



PARA TODOS
Dgo



**EL TECNOLÓGICO NACIONAL DE MÉXICO
A TRAVÉS DEL INSTITUTO TECNOLÓGICO DE DURANGO
Y EL DEPARTAMENTO DE METAL-MECÁNICA**

OTORGAN EL PRESENTE

RECONOCIMIENTO

A

ING. RICARDO ZAMUDIO

POR SU EXTRAORDINARIA IMPARTICIÓN DE LA
CONFERENCIA:

LA INDUSTRIA 4.0, A TRAVÉS DE LAS TECNOLOGÍAS QUE LA HABILITAN

DENTRO DEL **CONGRESO DE INGENIERÍAS
MECÁNICA Y MECATRÓNICA**, LLEVADO A CABO
DEL 19 AL 21 DE OCTUBRE DEL PRESENTE AÑO.

VICTORIA DE DURANGO, DGO. A 21 DE OCTUBRE DE 2021.

M.C. Isela Flores Montenegro
DIRECTORA



INSTITUTO TECNOLÓGICO SUPERIOR
de la Región de los Llanos

Otorga el presente

RECONOCIMIENTO

A: Ing. Ricardo Zamudio Carbajal

Por su destacada participación con la ponencia "*Predicción de Variables Ambientales del Microclima de un Invernadero Mediante modelos Grises*", en el marco del Foro Virtual de Ingeniería Mecatrónica 2021, el cual se llevó a cabo el día 16 de diciembre del presente año.

Cd. Guadalupe Victoria, Dgo., a 16 de diciembre de 2021.

Mtro. Juan Manuel García Hernández
Director General



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

DIRECCIÓN GENERAL



Dgo
SECRETARÍA DE EDUCACIÓN

SECRETARÍA DE
EDUCACIÓN

