

Ingeniería de Software para Mini-Fábrica de Plantas con Iluminación Artificial e Interconectividad IoT

Software Engineering for a Mini-PFAL (Plant Factory with Artificial Lighting) with IoT Interconnectivity



CU-ID: 2177/v31n3e04

✉ Luis Tonatiuh Castellanos Serrano^{1*}, ✉ María Victoria Gómez Águila¹, ✉ Fernando Salinas Paniagua¹,
✉ José Alfredo Castellanos Suárez¹, ✉ Arturo Martínez Rodríguez^{II}

¹Universidad Autónoma Chapingo, Texcoco de Mora, Edo. México, México

^{II}Universidad Agraria de La Habana, San José de las Lajas, Mayabeque, Cuba

RESUMEN: La ingeniería de software permite la planeación ordenada de los flujos de información y algoritmos para llegar al resultado deseado a través de la implementación de una metodología robusta y sistemática. Por otro lado, la tendencia *Indoor*, es un concepto tecnológico que ha evolucionado con la entrada del siglo XXI, con el objetivo de promover la agricultura urbana, periurbana y rural, fomentando agricultura en formato vertical en cualquier parte, desde un departamento hasta un invernadero. En el trabajo se exponen los resultados de investigación de ingeniería de software para una Mini-Fábrica de Plantas con Iluminación Artificial (*mini-PFAL: Plant Factory with Artificial Lighting*), tecnificada con múltiples capas de software para el control de parámetros que intervienen en el crecimiento de cultivos, como son la intensidad luminosa, temperatura, humedad, espectro luminoso, ciclos de riego/día y otros. Se exponen las estrategias de diseño para la creación de la arquitectura de software en sus diferentes niveles. Aplicando la metodología de diseño de software, se ordena la estructura del proceso de diseño, desde el análisis de requerimientos hasta el diseño de interfaz. El producto final resulta un sistema *PFAL IoT* que se interconecta vía *Bluetooth* y *WiFi* con el *SmartPhone*, a su vez vía puerto USB a una aplicación *GUI Visual Studio*, teniendo una data *WhereHosue* que administra la información de las variables en una base de datos local que gestiona la información en una página *SharePoint* que, a su vez, vincula con *Access* y un *Dashboard*, resultando un sistema *PFAL* inteligente.

Palabras clave: Fabrica de Plantas, *Indoor*, Internet de las cosas, Cultivos verticales, Diseño software.

ABSTRACT: Software engineering is a systematic approach that allows traceability of programming-oriented projects in the multiple paradigms that can be written, thus allowing the orderly planning of information flows and algorithms to reach the desired result through of the implementation of a robust and systematic methodology. On the other hand, the *Indoor* trend is a technological concept that has evolved with the entry of the 21st century, with the aim of promoting urban, peri-urban and rural agriculture. This concept creates agriculture anywhere from a department to a greenhouse. The present work presents the results of a research in software engineering referred to the design of a mini-PFAL, technified with multiple layers of software for the control of growth parameters in experimental lettuce crops, such as light intensity, temperature, humidity, light spectrum frequency, irrigation control, etc. The design strategies for the creation of the software architecture at its different levels are described, applying the software design methodology. The structure of the design process is ordered from the requirements analysis to the interface design, with them the final product reported is a PFAL IoT system that is interconnected via Bluetooth and WiFi with the SmartPhone, in turn via USB port to a GUI Visual Studio application, having a WareHouse data that administrates the information of the variables in a local database which manages the information on a SharePoint page that in turn, links with Access, having a Dashboard to directly control crop growth remotely, thus having an intelligent PFAL system for growing vegetable crops in vertical format.

Keywords: Plant Factory, Indoor, Internet of things, Vertical crops, Software design.

*Autor para correspondencia: Luis Tonatiuh Castellanos Serrano, e-mail: procesoslce@hotmail.com

Recibido: 16/10/2021

Aceptado: 24/06/2022

INTRODUCCIÓN

El modelo de agricultura vertical es propuesto en 1990 por el biólogo Dr. Dickson Despommier según [Chamberlain, \(2007\)](#), quien acuña este término a la estrategia de siembra realizando el aprovechamiento de espacios para maximizar las áreas de cultivos. Con la entrada de la era de la agricultura 4.0 y el acceso con menores costos a infraestructuras tales como los dispositivos LED's, sensores, procesadores, etcétera, la agricultura vertical se ha orientado a un término llamado Fábrica de Plantas con Iluminación Artificial (*Plant Factory with Artificial Lighting: PFAL*), definida por [Kozai et al. \(2015\)](#) como: "Estantería de múltiples cultivos con lámparas eléctricas en cada estante y posicionados verticalmente y que cuenta con otros equipos y dispositivos necesarios, tales como: acondicionadores y ventiladores de circulación de aire; unidades de suministro de CO₂ y soluciones de nutrientes; unidades de control del medioambiente". El mismo autor plantea que posicionar más cultivos verticalmente, incrementa la eficiencia del uso del suelo, así como que, aunque las lámparas fluorescentes han sido fundamentalmente usadas en las *PFAL*, en la actualidad las lámparas de diodos emisores de luz (*LED*), están atrayendo una gran atención para estos usos [Kozai et al. \(2016\)](#).

El diseño automatizado, las *TIC's*, *BigData*, Inteligencia Artificial y todos los nuevos fenómenos tecnológicos que están revolucionando la actualidad científico-técnica, se vinculan de forma correlacionada con el diseño de los *PFAL*, siendo más que nunca un factor de diseño sumamente importante el software, que les aporta un valor agregado tanto en el peso económico como de diseño de ingeniería en sistemas computacionales.

El Instituto de Ingenieros Electricistas y Electrónicos (IEEE) define la Ingeniería de Software como: 1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software; 2) El estudio de enfoques según el punto 1 ([IEEE, 2021](#)).

Por otro lado, [Pressman \(2010\)](#) desglosa en cuatro ejes rectores la ingeniería de software: Herramientas; Métodos; Procesos; Compromiso de Calidad, donde los procesos forman parte de la administración del proyecto para generar los reportes, informes, modelos, escritos, etc. Por otro lado, los métodos son las actividades prácticas de modelación, construcción, programación y afinamiento del proyecto, finalmente las herramientas son un vínculo entre los procesos y los métodos que permiten la asistencia de los programas que permitirán la integración del entorno del proyecto.

De este modo se genera una sinergia entre la importancia de la Ingeniería de Software y diversas aplicaciones en la agricultura, tales como:

Aplicaciones de Escritorio; App's Móviles para *Smartphone* en sistemas *Android* y *iOS*; *WebApp's*; *SpaceApp's*; Servicios de internet y gestores de bases de datos dedicados a la agricultura.

Todas estas plataformas están basadas en estructuras de software y soportadas en multilinguaje que les permiten programar sus funciones operativas. Asimismo, la correlación arquitectónica de su diseño se basa en la aplicación correcta de las metodologías de Ingeniería de Software según [Zielinski & Szmuc \(2005\)](#); [Pressman \(2010\)](#); [Serna \(2010\)](#), lo que hoy en día está tomando mayor terreno en el diseño de programas computacionales para aplicaciones en la agricultura en todo su contexto.

El desarrollo de la agricultura vertical según [Farminova Plant Factory \(2021\)](#), fábricas verdes o la tendencia *Indoor*, se encuentra concentrado en su mayoría en países líderes en este campo, tales como: Canadá, Japón, Taiwán, China, Estados Unidos, Países Bajos y algunos países europeos ([Tsukagoshi & Yutaka, 2016](#)). No obstante, las metodologías de *Ingenierías de Software*, en el ámbito de los *PFAL*, no son casi publicadas o la información es escasa. Estos países lideran el diseño, construcción e investigación de los *PFAL* mientras que otros países, como los latinoamericanos, se encuentran adentrándose en procesos de investigación según [Ocampo et al. \(2014\)](#) para analizar su viabilidad, lo cual muestra el atraso de la agricultura urbana, periurbana y popular, no aprovechándose las ventajas del uso de las *TIC's* y otras herramientas tecnológicas en la era de la agricultura 4.0, lo cual conlleva al ejercicio de la reflexión sobre la importancia que están tomando hoy en día las *AgroApp's* (por así nombrarlas) en el nicho de la agricultura moderna.

En este trabajo se exponen los resultados de una investigación que se ha trazado como **objetivo general**: Diseñar software para control y monitoreo de sistema de cultivos verticales Mini-*PFAL*, para la producción a baja escala de hortalizas agrícolas.

Dicho sistema cuenta con programación orientada, hardware basado en sistemas microcontrolados, que operan sensores y actuadores. De igual forma establecen interconexión de datos a una *PC*, Interfaz Gráfica de Usuario para monitoreo de los variables procesadas, interconexión a bases de datos (*BD*) local para el resguardo de información e interconexión inalámbrica vía *Bluetooth* y *WIFI*.

Asimismo, se trazó como objetivos específicos los siguientes:

- Diseñar algoritmos de control de lazo cerrado de iluminación y riego en sistemas micro-controlados;
- Diseñar Sistema de Gestión de Bases de Datos (*SGBD*) *Local-Host* con acceso a Internet;
- Diseñar algoritmos de retroalimentación de crecimiento de masa vegetal, temperatura, intensidad luminosa y humedad relativa;

- Interconectar *BD* local con la Interfaz Gráfica de Usuario (*GUI*), para control del sistema de cultivos verticales;
- Diseñar control inalámbrico *Bluetooth* con aplicación móvil basada en *Android Studio*;
- Diseñar conectividad *WIFI* para la gestión de datos en un Host en la nube.

MATERIALES Y MÉTODOS

Para el análisis de los requerimientos para el diseño de software para control y monitoreo de sistema de cultivos verticales *Mini-PFAL*, para la producción a baja escala de hortalizas agrícolas, se ha tenido en cuenta tanto las características operacionales, como el plan del proyecto y el modelo de análisis.

En cuanto a las **características operacionales** se tienen en cuenta las siguientes:

- Programación de Interfaz gráfica de Usuario de alto nivel en lenguaje *Visual Basic (VB)*;
- Programación de microcontroladores (*MCU*) para instrumentación del sistema, con panel *TFT 3.5"* con resolución de 320x480 RGB e interfaz *TouchPad*;
- Programación aplicación móvil en lenguaje *JAVA* para control de sistema con conectividad *Bluetooth*;
- Diseño de Base de Datos con gestor de *Access y SQL Server* e interconexión remota con aplicación de *SharePoint*;
- Implementación de *Dashboard* en *Power Apps* para interconexión con el servicio de *SharePoint*;

En el **plan del proyecto** se tiene en cuenta el trazado de requerimientos de sistemas bajo los siguientes diagramas:

- Aplicación de escritorio: Diagrama de flujo, diagrama *UML* y diseño de propuesta de formularios;
- Aplicación de *MCU's*: Diagrama de flujo y pseudocódigo;
- Aplicación Móvil: Diagrama *UML* y diseño de propuesta de formularios
- Bases de Datos: Diagramas de Entidad-Relación

Para configurar el **modelo de análisis** de la propuesta de sistema, se opta por implementar un diagrama de flujo de datos (DFD), recabando la información de las características operativas el resultado el DFD Nivel 0 como se muestra en la [figura 1](#).

La **metodología de diseño** propuesta para *Mini-PFAL* se muestra en el [siguiente diagrama](#).

RESULTADOS Y DISCUSIÓN

Diagrama de Flujo de Trabajo Operativo de Control del Sistema

Los diagramas de flujo de trabajo son herramientas para realizar el trazado del flujo de información, colocando los actores en canales divididos por bloques e interconectando los flujos de datos, para de ese modo representar el comportamiento del sistema. Según [Pressman \(2010\)](#), "...Un diagrama de actividades es similar a uno de flujo, y utiliza

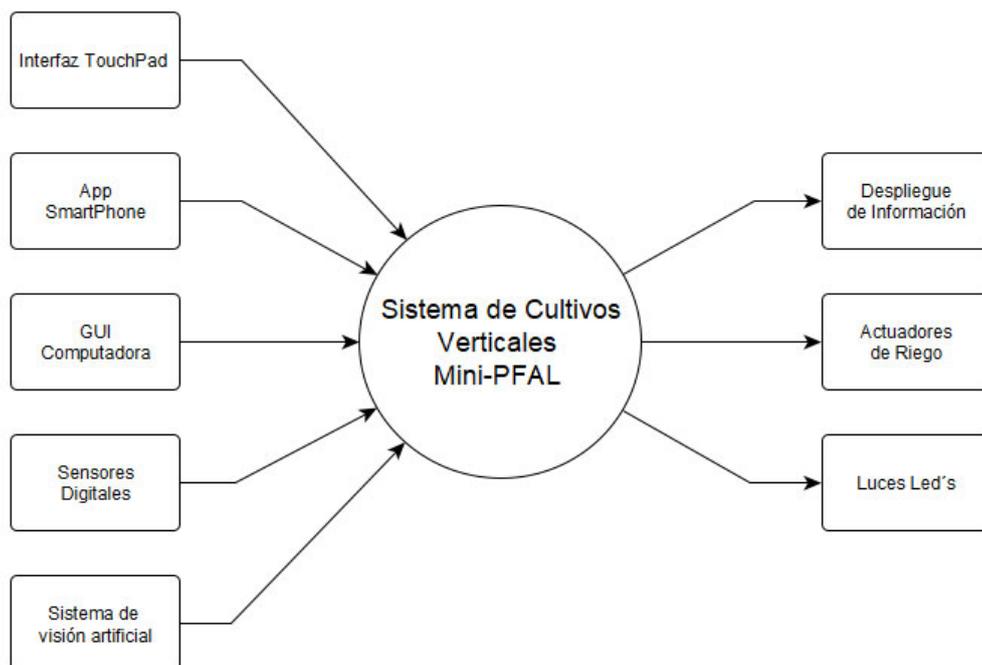
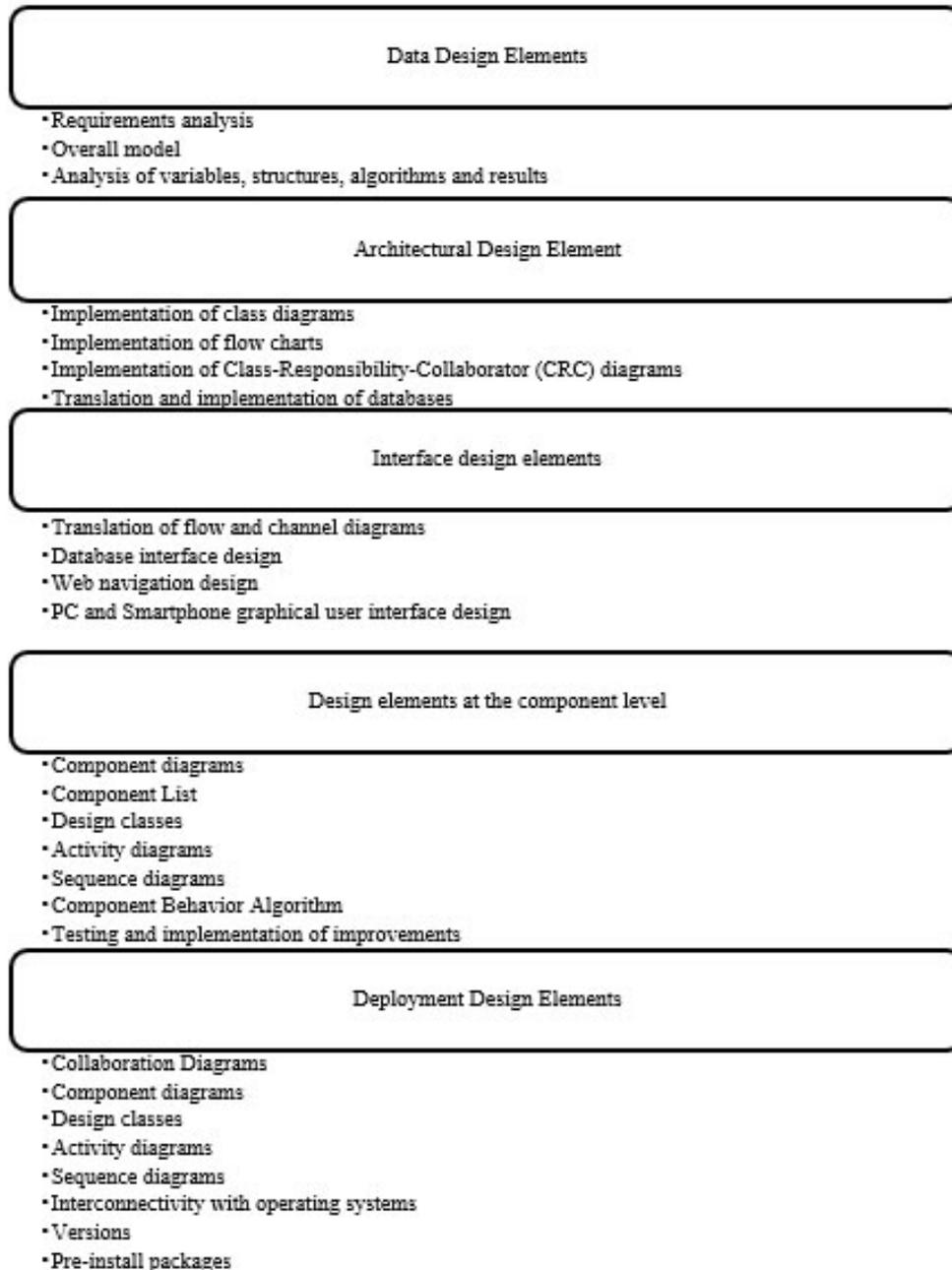


FIGURA 1. DFD Nivel 0 del Sistema de Cultivos Verticales Mini-PFAL (Autoría Propia).



rectángulos redondeados para denotar una función específica del sistema, flechas para representar flujo a través de éste, rombos de decisión para ilustrar una ramificación de las decisiones”. Este mismo autor plantea que: “Los casos de uso, junto con los diagramas de actividades y de canal, están orientados al procedimiento. Representan la manera en la que los distintos actores invocan funciones específicas (u otros pasos del procedimiento) para satisfacer los requerimientos del sistema.”

En el sistema *Mini-PFAL Networking* existen cuatro actores: Usuario, *PFAL*, *PC* y *SmartPhone*.

El usuario es el actor principal que se interconecta en primera instancia con el Sistema Embebido instalado en el *PFAL*, de allí la comunicación entre la

PC y el *SmartPhone* es seleccionada por el usuario. El siguiente diagrama ([Figura 2](#)) describe la forma operativa del comportamiento del flujo de información del sistema:

Modelo del Sistema de *Mini-PFAL*

Realizando la conjunción de modelado basado en clases y los diagramas de canal de modelo de caso de uso, se procedió a realizar el mapeo y la estructuración del diagrama representativo general de la propuesta del proyecto, el resultado se puede apreciar en la [Figura 3](#), reflejándose el diseño de la interfaz a través de un diagrama de lenguaje unificado de modelado (*UML*).

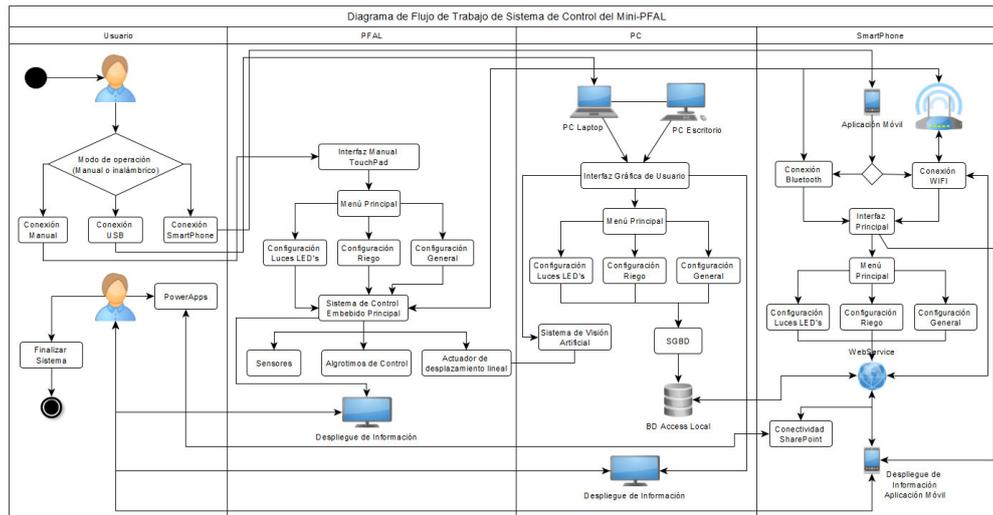


FIGURA 2. Diagrama de Flujo de Información del Sistema de Control del PFAL (Autoría Propia).

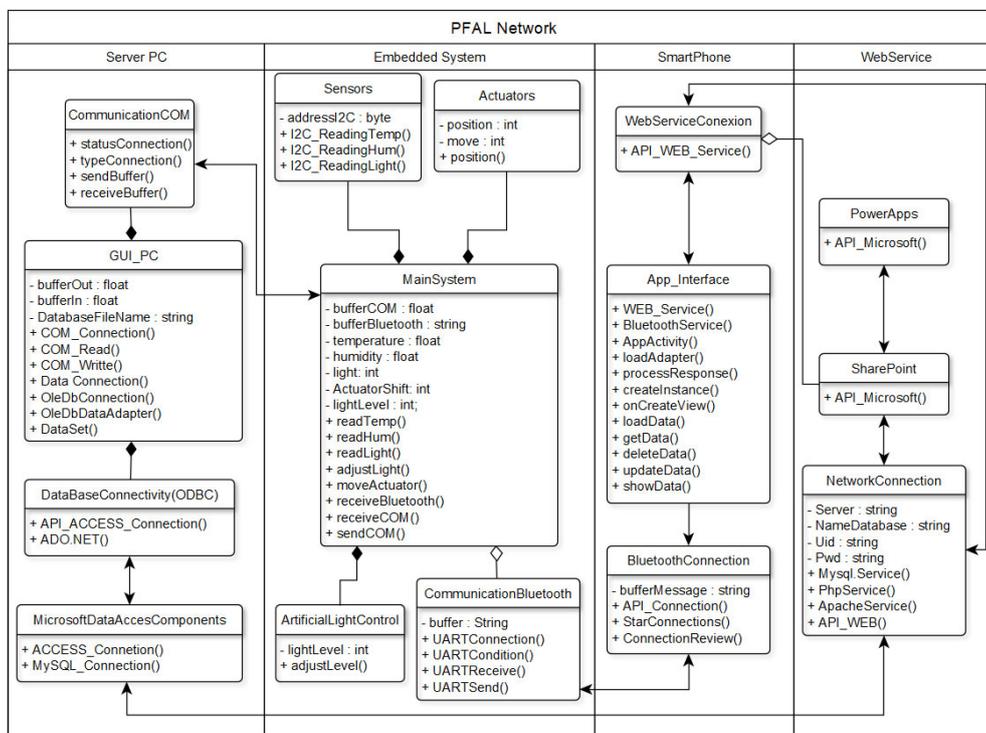


FIGURA 3. Modelo UML de diseño de la interfaz de Sistema de Cultivos Verticales Mini-PFAL (Autoría Propia)

Como se puede observar el diagrama de canal está dividido en cuatro secciones, lo que conjunta los análisis de requerimientos realizados en el anteproyecto, la [Tabla 1](#) explica de forma detallada los agentes que interviene en el Modelo UML propuesto para el diseño de la arquitectura de programación.

Diseño de Despliegue Formato de descriptor

En la metodología de Ingeniería de Software, según [Pressman \(2010\)](#), los elementos del diseño del despliegue indican la forma en la que se acomodarán

la funcionalidad del software y los subsistemas dentro del ambiente físico de la computación que lo apoyará.

La arquitectura de flujo de información, en el aspecto físico permite la trazabilidad del formato descriptor que se presenta en la [Figura 4](#):

Se puede observar que en su conjunto el sistema está integrado por cuatro nodos, donde se puede analizar la interconexión de los componentes-nodos en el flujo de información de las capas de procesamiento de la información alámbrica e inalámbrica.

TABLA 1. Descripción operativa de los bloques principales del Mini-PFAL

Bloque	Descripción Operativa de Flujo de Información
Server PC	<p>Este bloque tiene como eje central la clase <i>GUI_PC</i>, la cual es la Interfaz Gráfica de Usuario (GUI) basada en lenguaje Visual Basic, contiene una clase compuesta de <i>CommunicationCOM</i>, la cual gestiona una comunicación tipo <i>COM</i>, que por medio de un driver <i>COM-USB</i>, permite una comunicación dúplex vía puerto <i>USB</i>, con el Sistema Embebido (<i>Embedded System</i>), para obtener datos del <i>buffer</i> de entrada y salida que comandan el comportamiento operativo del <i>PFAL</i> y a su vez retroalimentan los datos para ser mostrados en la <i>GUI</i> por medio de herramientas gráficas. Por otro lado está compuesto de la clase <i>DataBaseConnectivity (ODBC)</i> y <i>MicrosoftDataAccessComponents</i>. Estas dos clases están relacionadas, la primera es la herramienta para la interconexión de una BD local de <i>Access</i> y la segunda es la BD local creada dentro del servidor local, lo cual permite almacenar los datos del <i>PFAL</i> en la <i>BD_Access</i>, ya sea por medio de la conexión <i>USB</i> o en su defecto por conexión a Internet con la clase <i>NetworkConnection</i></p> <p>Este bloque es el núcleo central de arranque, que congrega el inicio y fin de los 3 bloques restantes, donde se estructura la programación orientada a hardware, a través del uso de lenguaje C++ para la programación de un Atmega 2650 (Microchip, 2021). El bloque <i>MainSystem</i> representa el núcleo central del <i>MCU</i> donde se asignan las variables de control que operan los algoritmos de crecimiento del Mini-<i>PFAL</i>, como se puede observar en la figura 3, el sistema embebido este compuesto por tres elementos de acción (<i>Actuators-ArtificialLightControl</i>) y sensoramiento (<i>Sensors</i>). La clase <i>Sensors</i> se desglosa en la adquisición de tres variables:</p> <ul style="list-style-type: none"> • Temperatura: Variable a medir °C, implementando un integrado HTU21D por protocolo de comunicación I²C; • Humedad: Variable a medir %RH; integrado dentro del mismo integrado HTU21D; • Luminosidad: Variable a medir Lx, implementando un integrado BH1750 por protocolo de comunicación I²C.
Embedded System	<p>La clase <i>Actuators</i> se encarga del procesamiento de variables de 8 bits para la inyección de señal por modulación de ancho de pulsos (<i>PWM</i>) al <i>driver</i> de potencia, esto con el propósito de realizar el desplazamiento de un mecanismo lineal sinfin-corona, que es el encargado de desplazar el sistema de visión artificial para la evaluación de crecimiento de masa vegetal de los cultivos por medio de un algoritmo de procesamiento de video en tiempo real (Este sistema no es implementado en el diagrama <i>UML</i> porque es una mejora a largo plazo, pero se contempla para adición a futuro) El bloque de agregación de <i>CommunicationBluetooth</i>, es un periférico de salida por medio de un módulo HC-06 (Olimex, 2021), con la capacidad de comunicación dúplex maestro-esclavo y viceversa, esto permite la conexión al bloque <i>BluetoothConnection</i> la cual interactúa con la aplicación móvil para realizar la configuración de las variables del sistema embebido y a su vez retroalimenta a la <i>App</i> en un <i>Dashboard</i> que permita visualizar las variables en formas gráficas y etiquetas de despliegue de información.</p> <p>Como clase central en este bloque se encuentra <i>App_Interface</i>, basada en la arquitectura de Lenguaje de Marcas Extensible <i>XML</i>. Se realiza una <i>App</i> dinámica y visualmente atractiva en <i>Android Studio</i>, para la gestión remota del <i>PFAL</i> via <i>SmartPhone</i>. La gestión de datos se realiza en dos modalidades: la primera por medio de la clase <i>BluetoothConnection</i>, que permite configurar el <i>PFAL</i> y a la vez obtener datos todo por medio de la <i>App</i> móvil; por otro lado, también se cuenta con la clase <i>WebServiceConexion</i>, que permite una interconexión biunívoca con la BD local, pero a su vez también la conexión con un sitio en la nube por medio de la herramienta de <i>Sharepoint</i> de <i>Microsoft Teams</i>.</p>
SmartPhone	<p>El eje principal es la clase <i>NetworkConnection</i> donde es posible agregar una conexión con el componente <i>SharePoint</i> que a su vez se puede vincular con la aplicación de <i>PowerApps</i>, o simplemente vincular la conexión con <i>MicrosoftDataAccessComponents</i> del bloque <i>Server PC</i>, para vincular los datos de la BD local y así realizar indistintamente la consulta de los registros almacenados en la BD, o a su vez, actualizar los datos de la BD.</p>
WebService	<p><i>COM-USB</i> (Figura 6) por medio de un driver PL2303 (Prolific, 2022), intercomunicación bidireccional de comunicación <i>WIFI</i> por medio de un Esp8266 (ESPRESSIF, 2022) vía Serial, conectividad <i>Bluetooth</i> implementando un módulo HC-06 Serial Olimex (2021), y algunas salidas de control <i>PWM</i> y digitales para el control de la iluminación <i>LED</i>, actuadores de riego y de desplazamiento lineal del Mini-<i>PFAL</i>, lo cual se puede observar en el diagrama electrónico de la Figura 5, que se presenta a continuación:</p>

Diagrama electrónico del sistema embebido del Mini-PFAL

La capa de núcleo central ([Figura 5](#)) compuesta por la arquitectura electrónica, está diseñada por medio de la implementación de un microcontrolador Atmega 2560 [Microchip \(2021\)](#), el cual funge como administrador del control de tráfico de información, operando bajo un esquema de maestro para realizar la petición de las variables a los esclavos (módulos sensores, [Figura 7](#)), además de procesar y ordenar los datos de solicitud y gestionar la comunicación puerto

COM-USB ([Figura 6](#)) por medio de un driver PL2303 ([Prolific, 2022](#)), intercomunicación bidireccional de comunicación *WIFI* por medio de un Esp8266 ([ESPRESSIF, 2022](#)) vía Serial, conectividad *Bluetooth* implementando un módulo HC-06 Serial [Olimex \(2021\)](#), y algunas salidas de control *PWM* y digitales para el control de la iluminación *LED*, actuadores de riego y de desplazamiento lineal del Mini-*PFAL*, lo cual se puede observar en el diagrama electrónico de la [Figura 5](#), que se presenta a continuación:

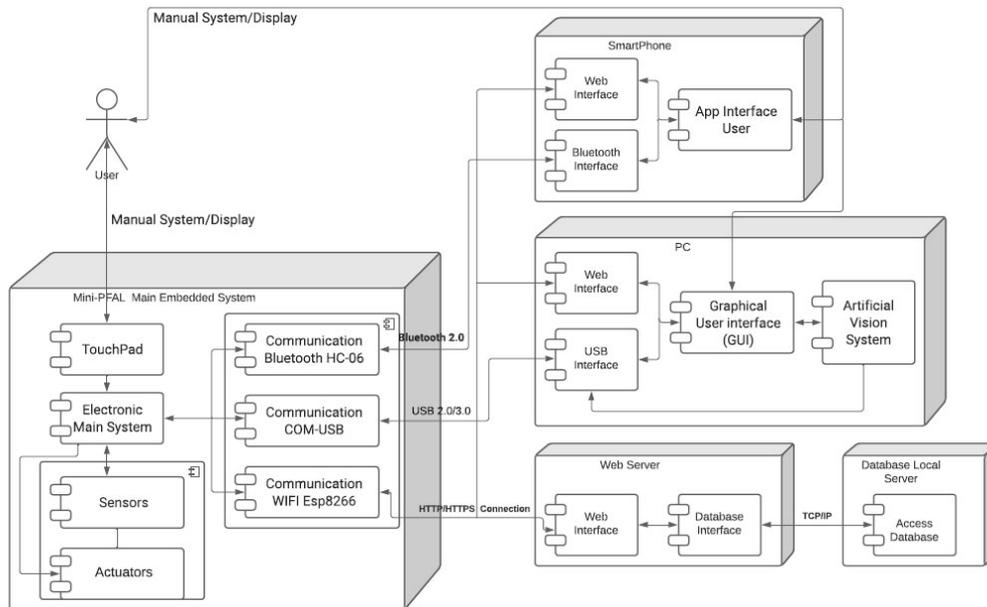


FIGURA 4. Diagrama de despliegue del formato descriptor del sistema de cultivos verticales Mini-PFAL (Autoría Propia).

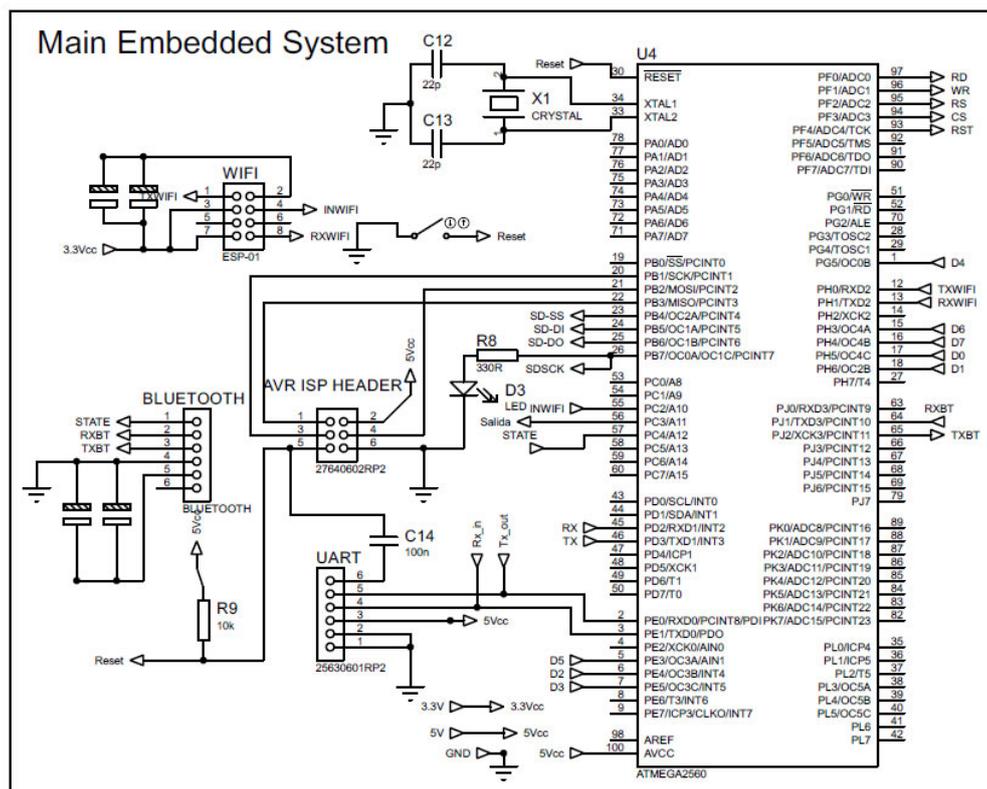


FIGURA 5. Esquema electrónico del sistema principal de control del Mini-PFAL (Autoría Propia)

Como se mencionó anteriormente, el chip PL2303 gestiona la comunicación *USB a RS232 Bridge Controller*, de este modo el PFAL logra conectarse vía puerto *COM* con la aplicación de escritorio para Windows de Visual Studio y con ello gestiona de forma bidireccional el tráfico de información de las variables de control ($^{\circ}\text{C}$, $\% \text{RH}$ y

Lux) para ser administradas con el Conector Abierto de Base de Datos *ODBC* de *VB* y administrar la información a la Base de Datos Local (Figura 6).

Los módulos sensores (esclavos), se intercomunican vía protocolo *RS-232* con el módulo principal (maestro). La petición se realiza bajo el algoritmo computacional que se muestra más adelante, donde el

maestro por medio de su algoritmo, gestiona el tráfico de información y lo administra a los dispositivos alámbricos o inalámbricos, dependiendo de la conexión en que se realiza la solicitud (Figura 7). Como núcleo principal se observa el Atmega 328p que recibe la solicitud del maestro vía Serial, posteriormente se interconecta vía I²C para dar lectura a los sensores HTU21D y BH1750, recolectando de este modo las variables de temperatura, humedad y luminosidad, para ser entregadas en formato de texto al maestro vía Serial.

Finalmente, el control del sistema embebido converge en una LCD TouchPad de 480x320 pixeles TFT de 3.5" (Figura 8), que ejerce la función de interfaz de salida y entrada para retroalimentar los datos bidireccionales del control y acuse de información del sistema, por medio de una interfaz gráfica visualmente atractiva y de fácil operación, lo que permite tener un menú con tres opciones: Sensores, Gráficas y Ajustes. La interfaz TouchPad permite configurar, efectuar la consulta en tiempo real de las variables operativas del invernadero vertical y

realizar los gráficos de temperatura, humedad y luminosidad con un periodo de 300 muestras por día.

Diseño de nivel de componentes del sistema embebido del Mini-PFAL

Implementando un proceso de lotificación al sistema embebido, se obtiene un diagrama estructurado que traza el comportamiento de flujo del programa como se muestra en la Figura 6.

La comunicación con los sensores se logra bajo protocolo RS-232, interconectados en una red de topología tipo bus en donde se interconectan 12 módulos en paralelo a las conexiones Vcc, GND, Tx y Rx. La petición de comunicación tiene como primera parte una cabecera de dirección para realizar la solicitud a los 12 módulos sensores interconectados en el sistema. La dirección de coincidencia permite realizar la solicitud de las variables de temperatura, humedad y luminosidad, para que, así, el esclavo las retorne al maestro.

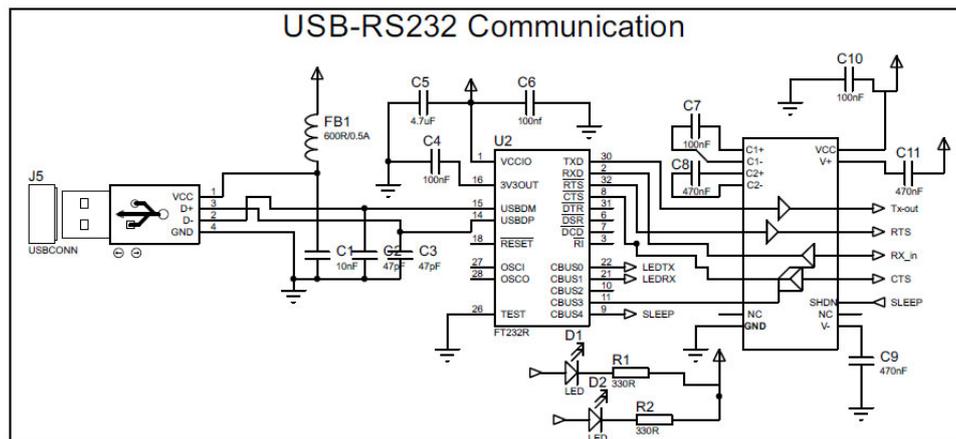


FIGURA 6. Diagrama electrónico del sistema de comunicación USB-RS232 para la vinculación con la PC y actualización de firmware (Autoría Propia).

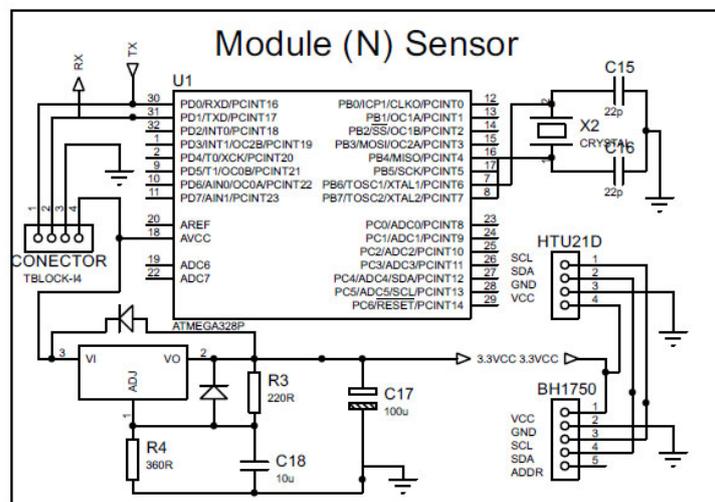


FIGURA 7. Esquema de módulos sensores (Autoría Propia).

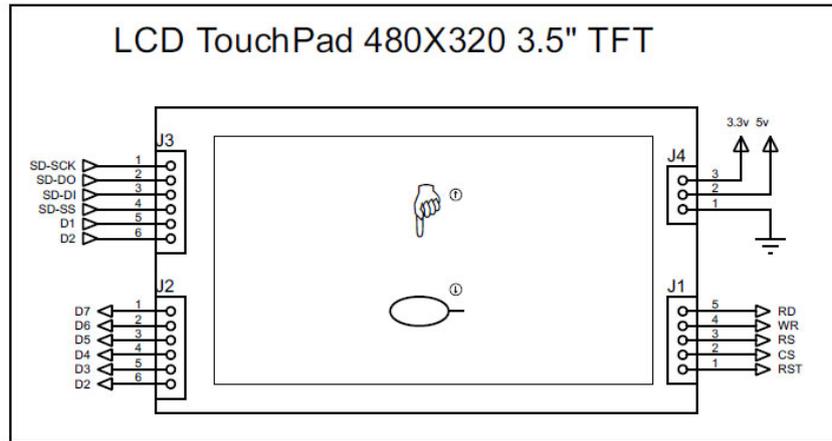


FIGURA 8. Interfaz LCD TouchPad (Autoría Propia).

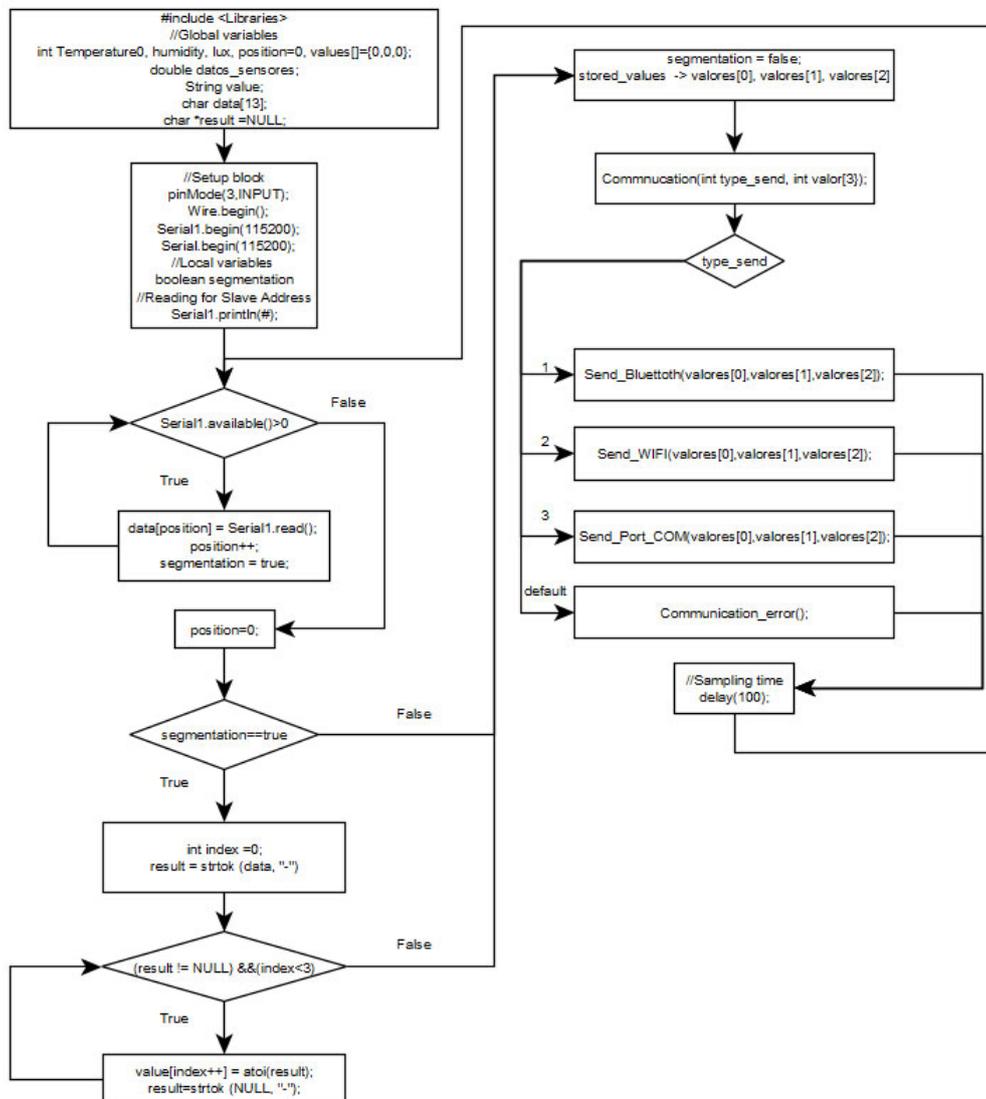


FIGURA 9. Diagrama de flujo de solicitud de datos de temperatura, humedad y luminosidad del Maestro-Eslavo (Autoría Propia).

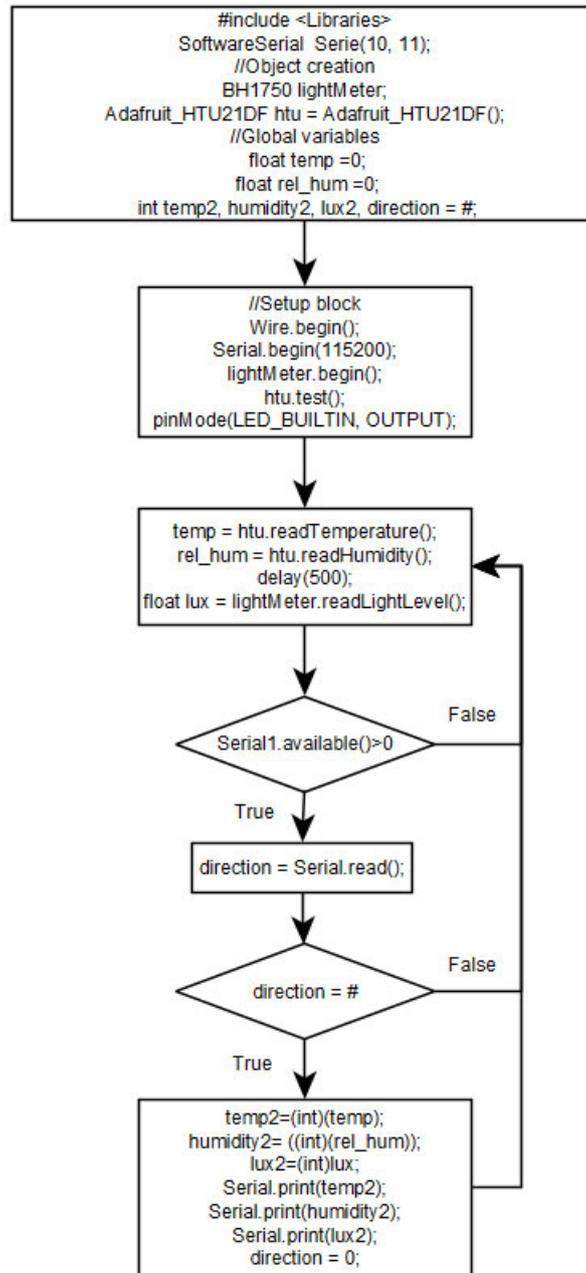


FIGURA 10. Diagrama de flujo de envío de datos de temperatura, humedad y luminosidad del Esclavo-Maestro (Autoría Propia).

Diseño de nivel de componentes de los módulos sensores esclavos

La solicitud descrita en diagrama de flujo por parte de los módulos esclavos se puede observar en el algoritmo de la programación de los esclavos que se muestra en la [Figura 10](#).

Como se puede observar, en el diagrama se revisa de forma iterativa, con el primer condicional, la presencia de paquetes en el buffer del puerto COM y, en ese mismo instante, todos los esclavos se ponen en alerta, por la naturaleza de la topología tipo bus, llegando la información a todos los esclavos, pero tan solo uno de ellos contiene la dirección de petición que

depura el segundo condicional. Posteriormente, los datos se envían al maestro para ser gestionado por el algoritmo del diagrama de la [figura 6](#), para enviar los datos vía *USB*, *Bluetooth* o *WIFI* respectivamente con la implementación de los bloques de clases correspondientes.

Diseño de la Interfaz de Sistema Embebido

La Interfaz Gráfica de Usuario para la operatividad principal de la máquina se basa en el control *LCD TouchPad* de 480x320, instalada en el panel frontal del módulo ([Figura 11](#)), el usuario puede navegar por tres menús principales de forma interactiva:

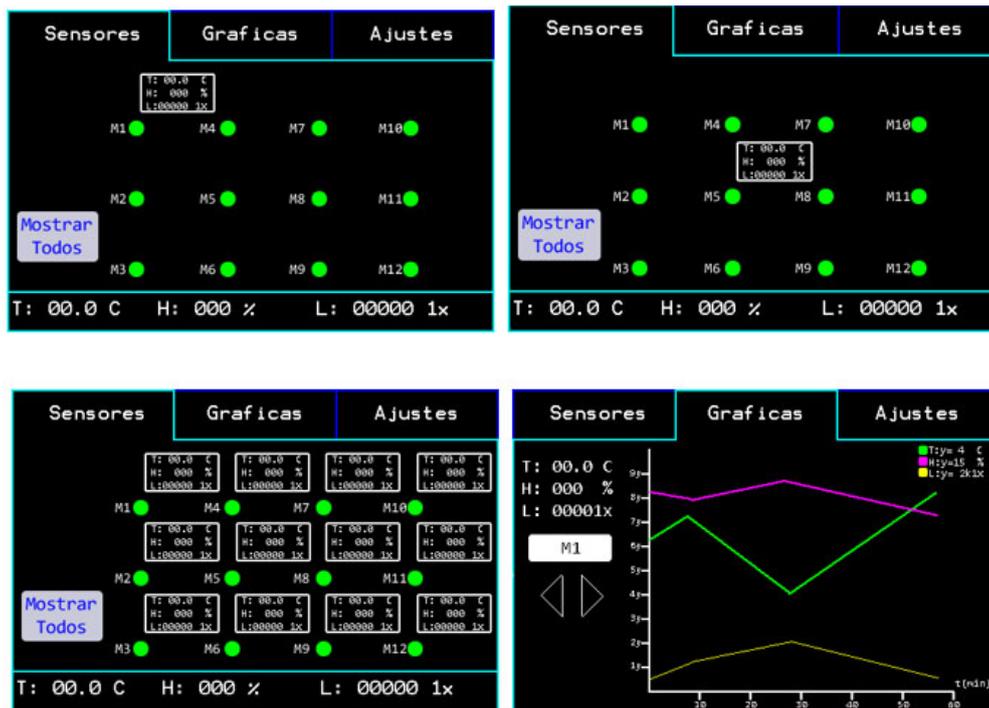


FIGURA 11. Interfaz Gráfica del Panel *TouchPad* (Autoría Propia).

1. *Sensores*: Permite realizar la consulta de los módulos sensores interconectados en el *PFAL*, por la petición multiplexada en protocolo RS-232. En la respuesta *Half-Duplex* de petición del *MCU* principal, el retorno de información contiene los valores cuantificados de las variables “Temperatura”, “Humedad” y “Luminosidad”, que se despliegan en la matriz de sensores mostrados en el panel. Basta con que el usuario de clic en algún punto verde de la matriz o seleccione la opción de “Mostrar Todo” para acusar toda la información
2. *Gráficas*: Muestra los valores promedio resguardados en un *ArrayList* en el transcurso de 24 horas, aproximadamente se grafican 300 muestras por día
3. *Ajustes*: Accede a los ajustes principales de la máquina, como es la frecuencia lumínica, activación de los ciclos de riego, test de la máquina, etc.

CONSIDERACIONES FINALES

Como ha podido apreciarse, el presente artículo se enfoca a la aplicación de la metodología de Ingeniería de Software, siguiendo las etapas de creación, a saber: el análisis de requerimientos y el modelo de diseño, que a su vez comprende [Mohapatra \(2010\)](#) el diseño de datos, el diseño de arquitectura, el diseño de interfaz y el diseño de nivel de componentes.

Las etapas en su conjunto conforman los módulos de software que interactúan en las capas de multilinguaje para implementar la solución, tanto

hardware (diseño electrónico aplicado), programación de *WebApps*, gestión de *Data WhereHouse*, interconectividad inalámbrica *IoT*, gestión de BD locales y el *Hosting*.

La explicación detallada de las capas de *Hosting*, [Data Warehouse IBM \(2021\)](#) y *GUI* no se implementa minuciosamente para términos de la exposición de resultados, dejando esta labor en la continuación de resultados de investigación a futuro. Los puntos importantes a desarrollar en la exhibición de la presente, se basa en la interconectividad de datos, el análisis de requerimientos, el modelo del diseño y, haciendo uso de los formatos de diagramas, se logró exponer las estrategias de programación empleadas para el diseño del sistema mini-PFAL.

La forma operativa de las capas de software en su conjunto de cohesión de funcionamiento dinámico se puede esbozar de mejor manera implementando la *Arquitectura de Capas del Sistema* como se muestra en el diagrama de la [Figura 12](#).

La sinergia interrelacionada de las capas de software que interactúan en el sistema, permite la gestión ordenada del tráfico de datos, desde la obtención de las variables hasta el despliegue de información en un “*Dashboard*”, con información dinámica, gráfica y depurada, con una interacción flexible e inteligente para el usuario, dando una solución técnica de la agricultura moderna, donde el nuevo agricultor del siglo XXI, se convierte en un sujeto con conocimientos técnicos en la operatividad *IoT* con dispositivos de agricultura inteligente, que conforman conocimientos y habilidades en la gestión y uso del *BigData*, *IA*, *RA*, *VA*, *Drones*, *IoT*, etcétera.

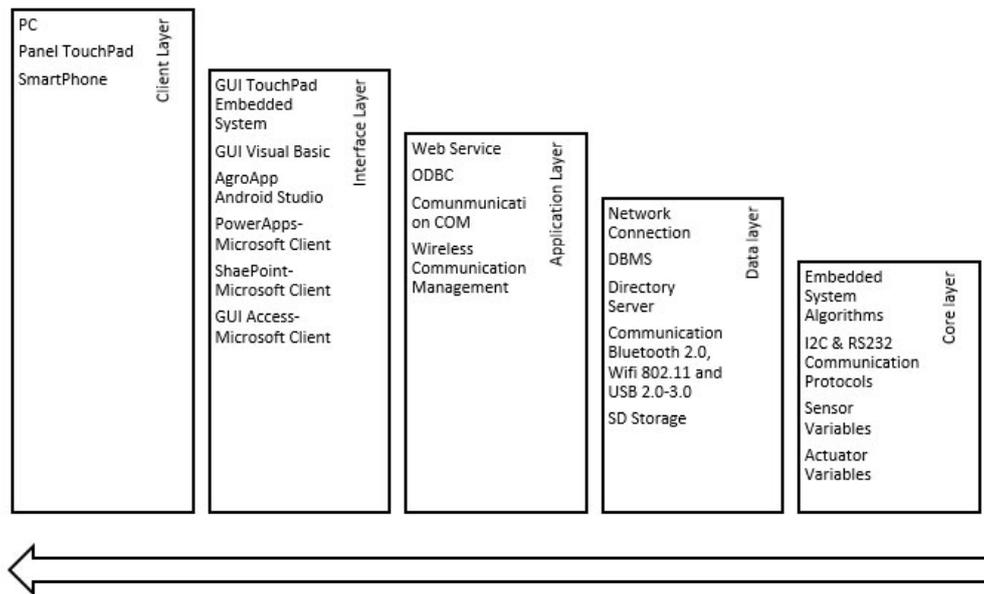


FIGURA 12. Diagrama de arquitectura de capas de sistema Mini *PFAL* con interconectividad *IoT* para gestión de datos inteligente (Autoría Propia).

El conjunto de todo el proyecto en su funcionalidad es una exposición del modelo de resultados bastante ambicioso, dado que esta subdividido en 7 partes importantes:

1. Análisis técnico del sistema
2. Análisis de diseño mecánicos aplicado
3. Diseño de electrónico aplicado
4. Sistema computacional multiplataforma
5. Telecomunicaciones y gestión de tráfico de información
6. Algoritmos de visión artificial
7. Modelado matemático del sistema *PFAL* en modo operativo
8. Estudio de crecimiento de cultivos, evaluación de rendimientos, nutrimental, etc.

El conjunto explicativo y detallado de todos estos aspectos requiere de una publicación más amplia, de manera que pueda desglosarse parte por parte cada uno de los componentes funcionales del sistema Mini *PFAL IoT* propuesto.

CONCLUSIONES

Como resultado de la investigación de ingeniería de software para una Mini-Fábrica de Plantas con Iluminación Artificial, se determinan y exponen los diferentes componentes del sistema, a saber:

- El diagrama de flujo de información del sistema de control de la Fábrica de Plantas con Iluminación artificial *PFAL*;
- El modelo *UML* de diseño de la interfaz de Sistema;

- El diagrama de despliegue del formato descriptor del sistema;
- El esquema electrónico del sistema principal de control;
- El diagrama electrónico del sistema de comunicación;
- El esquema de módulos sensores y su intercomunicación;
- La interfaz de salida y entrada para retroalimentar los datos bidireccionales del control y acuse de información del sistema;
- Los diagramas de flujo (algoritmos) de solicitud y envío de datos de temperatura, humedad y luminosidad del Maestro-Esclavo;
- El diseño de la interfaz gráfica del panel de control.

AGRADECIMIENTOS

Los resultados forman parte de proyecto de Desarrollo y Transferencia de Tecnológica (DTT), impulsados por los programas de investigación de la Dirección General de Investigación y Posgrado de la Universidad Autónoma Chapingo (UACH), así mismo a la línea de investigación No. 6 de “Ciencia y tecnología aplicada en la agricultura 4.0 para el desarrollo rural” perteneciente al centro de investigación No. 82 CIISCINACYC, del Departamento de Sociología Rural. Se agradece a la administración en turno del municipio de Nopaltepec, el cual promueve los convenios de investigación para poder gestionar los recursos en el pro de la vanguardia del campo y así llevar a cabo la transferencia tecnológica para la modernización del campo mexicano, y finalmente y no menos importante, a la

colaboración de la Universidad Agraria de La Habana de Cuba por la asesoría metodológica y técnica brindada para fortalecer el presente artículo de investigación.

REFERENCIAS BIBLIOGRÁFICAS

- CHAMBERLAIN, L.: "Skyfarming", *New York Magazine*, 24, 30 de marzo de 2007, Disponible en: <https://www.microchip.com/en-us/product/ATmega2560>.
- FARMINOVA PLANT FACTORY: *The Future of Agriculture*, [en línea], Cantex Group, 2021, Disponible en: <http://www.farmicova.com>, [Consulta: 20 de marzo de 2022].
- IBM: *¿Qué es un Data Warehouse?*, [en línea], IBM Db2 Warehouse, Edición Empresarial, 20 de mayo de 2021, Disponible en: <https://www.ibm.com/mx-es/analytics/data-warehouse>, [Consulta: 20 de junio de 2021].
- IEEE: *An Annotated Bibliography for ISO/IEC/IEEE 42010*, [en línea], 15 de octubre de 2021, Disponible en: <http://www.iso-architecture.org/ieee-1471/docs/bibliography-42010.pdf>, [Consulta: 29 de junio de 2021].
- KOZAI, T.; FUJIWARA, K.; RUNKLE, E.: "Why LED lighting for urban agriculture?", En: *LED lighting for urban agriculture*, Ed. Springer, Singapore, pp. 3-18, 2016.
- KOZAI, T.; NIU, G.; TAKAGAKI, M.: *Plant factory: An indoor vertical farming system for efficient quality food*, Ed. Springer, Academic ed., Singapore, 2015.
- MICROCHIP: *ATmega2560*, [en línea], Microchip, 2021, Disponible en: <https://www.microchip.com/en-us/product/ATmega2560>, [Consulta: 20 de junio de 2021].
- MOHAPATRA, P.K.J.: *Software Engineering*, Inst. New Age International, California, USA, 2010.
- OCAMPO, R.A.; HERNÁNDEZ, G.L.H.; FERNÁNDEZ, V.D.; FERNÁNDEZ, V.D.; CERVANTES, B.R.: "Análisis estático de los esfuerzos y deformaciones de la estructura de un Invernadero tipo Ventila Cenital ubicado en Veracruz", *Revista Ciencias Técnicas Agropecuarias*, 23(4): 10-16, 2014, ISSN: 1010-2760, e-ISSN: 2071-0054.
- OLIMEX: *HC Information Technology Co., Ltd. Guangzhou*, [en línea], Olimex, 20 de mayo de 2021, Disponible en: <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>, [Consulta: 23 de junio de 2021].
- PRESSMAN, R.: *Ingeniería del Software Un Enfoque Práctico* University of Connecticut, Ed. McGraw-Hill Interamericana, Madrid, publisher: SA, 2010.
- SERNA, M.E.: "Métodos formales e Ingeniería de Software", *Revista Virtual Universidad Católica del Norte*, (30): 1-26, 2010, ISSN: 0124-5821, Disponible en: <https://www.redalyc.org/articulo.oa?id=194214476008>, [Consulta: 21 de agosto de 2021].
- TSUKAGOSHI, S.; YUTAKA, S.: *Plant Factory. An Indoor Vertical Farming System for Efficient Quality Food Production*, Ed. Elsevier, Current Status and Perspectives ed., vol. Chapter 3 PFAL Business and R&D in the World, 35-68 p., publisher: Academic Press: London, UK, 2016.
- ZIELINSKI, K.; SZMUC, T.: *Software engineering: evolution and emerging technologies*, Ed. IOS Press, vol. 130, California, USA, 2005, ISBN: 1-58603-559-2.

Luis Tonatiuh Castellanos-Serrano, Profesor e Investigador, Universidad Autónoma Chapingo, Centro de Investigaciones Interdisciplinarias y de Servicio en Ciencia, Naturaleza, Sociedad y Cultura, Carretera Federal México-Texcoco km 38.5, C.P. 56230 Texcoco de Mora, Estado de México, México, estudiante de maestría programa pnp no escolarizado en sistemas computacionales universidad da Vinci (UDAVINCI), e-mail: procesoslce@hotmail.com

María Victoria Gómez-Águila, Profesora, Universidad Autónoma Chapingo, Departamento de Ingeniería Mecánica Agrícola, Carretera Federal México-Texcoco km 38.5, C.P. 56230 Texcoco de Mora, Estado de México, México, e-mail: mvaguila@hotmail.com

Ing. Fernando Salinas Paniagua, Profesor e Investigador, Universidad Autónoma Chapingo, Centro de Investigaciones Interdisciplinarias y de Servicio en Ciencia, Naturaleza, Sociedad y Cultura, Carretera Federal México-Texcoco km 38.5, C.P. 56230 Texcoco de Mora, Estado de México, México, e-mail: procesoslce@hotmail.com

José Alfredo Castellanos Suárez, Profesor e Investigador, Universidad Autónoma Chapingo, Centro de Investigaciones Interdisciplinarias y de Servicio en Ciencia, Naturaleza, Sociedad y Cultura, Carretera Federal México-Texcoco km 38.5, C.P. 56230 Texcoco de Mora, Estado de México, México, e-mail: procesoslce@hotmail.com

Arturo Martínez Rodríguez, Prof. Titular e Inv. Tiular, Prof. de Mérito. Universidad Agraria de La Habana (UNAH). Facultad de Ciencias Técnicas, Centro de Mecanización Agropecuaria (CEMA), San José de las Lajas, Mayabeque, Cuba, e-mail: armaro646@gmail.com

AUTHOR CONTRIBUTIONS: **Conceptualization:** L. T. Castellanos-Serrano, M. V. Gómez-Águila, F. Salinas-Paniagua. **Data curation:** L. T. Castellanos-Serrano, F. Salinas-Paniagua. **Formal analysis:** L. T. Castellanos-Serrano, M. V. Gómez-Águila, F. Salinas-Paniagua. **Funding acquisition:** M. V. Gómez-Águila. **Investigation:** L. T. Castellanos-Serrano, M. V. Gómez-Águila, F. Salinas-Paniagua. **Methodology:** L. T. Castellanos-Serrano, F. Salinas-Paniagua, A. Martínez-Rodríguez. **Project administration:** M. V. Gómez-Águila. **Resources:** Universidad Autónoma Chapingo. **Software:** L. T. Castellanos-Serrano, F. Salinas-Paniagua. **Supervision:** M. V. Gómez-Águila. **Validation:** M. V. Gómez-Águila, J. A. Castellanos-Suárez. **Visualization:** M. V. Gómez-Águila, J. A. Castellanos-Suárez. **Roles/Writing, original draft:** L. T. Castellanos-Serrano, M. V. Gómez-Águila, F. Salinas-Paniagua, J. A. Castellanos-Suárez, A. Martínez-Rodríguez. **Writing, review & editing:** L. T. Castellanos-Serrano, M. V. Gómez-Águila, J. A. Castellanos-Suárez, A. Martínez-Rodríguez.

Los autores de este trabajo declaran no presentar conflicto de intereses.

Este artículo se encuentra bajo licencia [Creative Commons Reconocimiento-NoComercial 4.0 Internacional \(CC BY-NC 4.0\)](https://creativecommons.org/licenses/by-nc/4.0/)

La mención de marcas comerciales de equipos, instrumentos o materiales específicos obedece a propósitos de identificación, no existiendo ningún compromiso promocional con relación a los mismos, ni por los autores ni por el editor.