Desarrollo de un sistema embebido para un compostero doméstico inteligente

Oscar Sepúlveda-Cisneros¹, Brenda Acevedo-Juárez², Claudia Estela Saldaña-Durán³, Wilson Castro⁴, Miguel De-la-Torre¹, Himer Avila-George¹

oscar.sepulveda9265@alumnos.udg.mx; brenda.acevedo@academicos.udg.mx;
cesduran@uan.edu.mx; wcastro@unf.edu.pe; miguel.dgomora@academicos.udg.mx;
himer.avila@academicos.udg.mx

¹ Departamento de Ciencias Computacionales e Ingenierías, Universidad de Guadalajara, Ameca, 46600, Jalisco, México.

² Departamento de Ciencias Naturales y Exactas, Universidad de Guadalajara, Ameca, 46600, Jalisco, México.

³ Universidad Autónoma de Nayarit, Tepic, 63000, Nayarit, México.

⁴ Facultad de Ingeniería en Industrias Alimentarias, Universidad Nacional de Frontera, Sullana, 20101, Piura, Perú.

DOI: 10.17013/risti.41.112–129

Resumen: Uno de los grandes problemas de la humanidad es la excesiva cantidad de residuos que se generan a nivel mundial; por lo tanto, es de vital importancia no sólo reducirlos sino también transformarlos en nuevos productos. Aproximadamente, el 50% de los residuos generados en las ciudades son orgánicos, para darles un valor agregado se pueden transformar en composta. En este artículo, se presenta el desarrollo de un software embebido para un compostero doméstico inteligente. El proceso de desarrollo del software propuesto está guiado por un método específico para sistemas embebidos el cual combina las técnicas de desarrollo en espiral y prototipos; consta de once etapas y tres iteraciones. En este trabajo se presentan los resultados de la primera iteración del proceso de desarrollo de software utilizado.

Palabras-clave: Compostero Doméstico Inteligente, Ingeniería de Software, Sistema Embebido, Metodología.

Development of an embedded system for a smart home composter system

Abstract: One of the great problems of humanity is the excessive amount of waste generated worldwide; therefore, it is of vital importance not only to reduce it but also to transform it into new products. Approximately 50% of the waste generated in cities is organic, and this waste can be given an added value when transformed into compost. In this paper, the development of embedded software for a smart home composter is presented. The development process of the proposed software is guided by a specific method for embedded systems which combines the

techniques of spiral development and prototyping; it consists of eleven stages and three iterations. This paper presents the results of the first iteration of the software development process used.

Keywords: Smart Home Composter, Software Engineering, Embedded System, Methodology.

1. Introducción

El proceso de compostaje es una técnica que permite la transformación residuos orgánicos en composta, con el propósito de obtener un producto con un valor agregado. El compostaje se lleva a cabo, mediante reacciones microbiológicas de mineralización y humidificación de la materia orgánica, para lograr un producto final biológicamente estabilizado, que no consume oxígeno, ni es capaz de generar metabolitos fitotóxicos (Cesaro, Belgiorno, & Guida, 2015; Das et al., 2011).

La composta se puede producir implementando el método aeróbico cerrado, este método requiere de un reactor cerrado y la oxigenación constante durante el proceso (Pratama, 2019). Por el estado de la temperatura son observadas tres etapas en el proceso de compostaje aerobio, éstas son: la mesófila (20 a 45° C), la termófila (45 a 70° C) y la de enfriamiento (70 a 20° C). Es importante que para llevar a cabo el proceso de compostaje sean controlados los factores fisicoquímicos, tales como: (1) temperatura, (2) humedad, (3) tamaño de la partícula, (4) pH, (5) aireación y (6) tipo de residuo orgánico. El producto será considerado de calidad, para ser usado como sustituto de tierra para maceta, si tiene las siguientes características: materia orgánica entre <20% en materia seca, humedad entre 25% y 35% y una relación carbono/nitrógeno < 15, según los parámetros establecidos en la norma ambiental para el distrito federal NADF-020-AMBT-2011 (Cesaro et al., 2015; Delgado Peralta, 2012; Ministerio de la Presidencia, 2013).

El proceso de compostaje podría automatizarse mediante el desarrollo de un sistema inteligente. De acuerdo con Sassone et al., los sistemas inteligentes incorporan funciones de detección, actuación y control con el fin de describir y analizar una situación, y tomar decisiones basadas en los datos disponibles de manera predictiva o adaptativa, realizando así acciones inteligentes (Sassone, Grosso, Poncino, & Macii, 2016). Por lo tanto, el desarrollo de un sistema inteligente que permita el monitoreo y control de las variables que intervienen en el proceso de compostaje, mediante la implementación de un conjunto de sensores y actuadores controlados por un software alojado en el microcontrolador, puede ser una solución eficiente y de bajo costo.

En este trabajo de investigación, se propone el desarrollo de un sistema de software embebido que tenga la capacidad de automatizar un compostero doméstico inteligente que mantenga las condiciones fisicoquímicas para obtener composta de calidad y que el desarrollo del sistema de software sea soportado por técnicas de ingeniería de software.

2. Antecedentes

El término de sistema embebido surge por primera vez en la década de los 70s, gracias a la aparición del chip 4004 por parte de una compañía Japonesa llamada *Busicom*,

siendo una calculadora la primera aplicación que se le dio para un sistema embebido. Esta implementación dio paso para el desarrollo de un circuito electrónico de propósito general que pudiera ser implementado en una línea completa de producción de calculadoras (Barr & Massa, 2006).

En la época de los 80s y 90s, el desarrollo de sistemas embebidos tuvo un gran incremento debido al desarrollo de una gran cantidad equipos electrónicos en diversas áreas que facilitaron las actividades cotidianas, entre las cuales se encuentran: la cocina (tostadoras, procesadores de comida, y hornos de microondas), en el hogar (televisión, control remoto y equipos de sonido) y áreas de trabajo (fax, impresoras, cajas registradoras y lectores de tarjetas de crédito).

En la actualidad, los sistemas embebidos los podemos encontrar en muchos aparatos y aplicaciones computacionales, resolviendo un gran número de actividades diarias. En (Malinowski & Yu, 2011), se mencionan algunas de las aplicaciones donde se pueden encontrar sistemas embebidos, entre ellas: (a) sistemas de automatización, (b) sistemas adaptativos de control, (c) compensación no-lineal, (d) procesamiento de audio e imágenes, (e) servicios de Internet, (f) sistemas de comunicación, (g) diagnóstico del sistema y análisis de ruido/fallas, (h) plataformas robóticas, (i) diseño de circuitos y (j) plataformas educativas.

En lo que respecta al proceso de transformación de los residuos orgánicos, hoy en día, es posible encontrar una gran variedad de sistemas. Pansari et al. presentaron un sistema de monitoreo remoto mediante sensores de temperatura, humedad y gas metano (DTH11, MQ) para mejorar el proceso aerobio en las pilas de compostaje; además, utilizaron una tarjeta Arduino y midieron las variables temperatura, humedad, gas metano (Pansari, Deosarkar, & Nandgaonkar, 2019). Wardhany et al. desarrollaron un compostero inteligente que consiste en activar un triturador de residuos orgánicos recolectados y depositados en una caja de fermentación; incorporan un sistema de monitoreo remoto de temperatura, humedad y altura de los residuos orgánicos triturados en la caja de fermentación utilizando los sensores LDR, DS18B20, Kalemaban, Ultrasónico y una placa Arduino MEGA (Wardhany et al., 2019). En 2019, Pratama et al. propusieron un prototipo para la elaboración de compostaje aerobio; el prototipo consiste en una máquina mezcladora de composta con un sistema de automatización del mezclado basado en lógica difusa en función de los valores de temperatura y humedad recibidos usando los sensores YL-39 YL-69 Soil Moisture, DS18B20, sensor pH y una tarjeta RaspBerry (Pratama, Ariyanto, & Karimah, 2019). Finalmente, Elalami et al. presentan el diseño y fabricación de una máquina electromecánica rotativa dedicada al tratamiento aerobio de residuos orgánicos y gestionada por un control remoto que permite controlar los parámetros de temperatura, pH, humedad y amonio en tiempo real, sin embargo, no mencionan que dispositivos electrónicos utilizaron (Elalami et al., 2019).

En lo que respecta a los modelos de desarrollo de software enfocados a sistemas embebidos, en la literatura se encuentran reportados varios modelos. En (Boehm, 1984) se abordan algunos enfoques basados en el *V-Model*; Brown analiza algunos modelos basados en el ciclo de vida reportados en el estándar IEC 61508 (Brown, 2000); Douglas introduce el método *Rapid Object-Oriented Process for Embedded Systems (ROOPES)* el cual se basa en el modelo en espiral (Douglass, 2004); Liggesmeyer y Trapp proponen

un método que combina el diseño en espiral y el método de prototipos (Liggesmeyer & Trapp, 2009); recientemente, Oshana y Kraeling abordan un modelo basado en metodologías ágiles (Oshana & Kraeling, 2019).

3. Materiales y Método propuesto

3.1. Materiales

Materiales electrónicos y mecánicos componen el compostero doméstico inteligente. En cuanto a los componentes electrónicos se emplean: un microcontrolador, leds, una pantalla LCD, una memoria micro SD, y sensores de humedad, temperatura, oxígeno y peso, cada uno de estos componentes fueron elegidos mediante un previo análisis y comparación de sus características. Estos componentes se observan en la Figura 1 y se describen a continuación.

Figura 1 – Componentes electrónicos que integran el sistema embebido. (a) placa Arduino; (b) sensor de humedad; (c) sensor de temperatura; (d) sensor de oxígeno; (e) sensor de peso; (f) pantalla LCD; (g) módulo SD; y (h) Leds.

- a. Una placa electrónica *Arduino Mega 2560* basada en el microcontrolador *ATmega2560*, cuenta con 54 entradas/salidas digitales de las cuales 14 pueden ser utilizadas como PWM (modulación por ancho de pulsos por sus siglas en inglés); asimismo, tiene 16 entradas analógicas y un oscilador de cristal de 16 MHz.
- b. El sensor *FC-28* está conformado por dos placas separadas recubiertas con un material conductor, estas placas permiten obtener la humedad del suelo cuando se detecta una variación en su conductividad. Cuenta con un módulo *YL-38* que contiene un amplificador operacional *LM393* que se encarga de transformar la señal de conductividad obtenida en un valor analógico. Este tipo de sensor permite obtener la humedad como un valor analógico (0-1023) o digital (*High* o *Low*).
- c. El sensor de temperatura utilizado es el *DS18B20* que se comunica sobre un bus denominado 1-Wire y requiere solo de una línea de datos para enviar la información al microcontrolador.

- d. El sensor *gas-Grove* permite obtener la concentración de oxígeno en el aire, basado en el principio de una célula electroquímica. Entre las ventajas de utilizar este sensor se tienen: (1) una alta precisión, (2) una gran fiabilidad, (3) una fuerte habilidad para evitar la interferencia y (4) una alta sensibilidad.
- e. Sensor de peso (*galga extensiométrica*), este sensor permite detectar variaciones de presión sobre una superficie. Las características de este sensor son las siguientes: opera con 5 V, bajo una de temperatura de -20 a 65° C, cuenta con 4 cables de conexión (1) GND, (2) VCC, (3) señal positiva y (4) señal negativa; además, requiere del módulo HX711 que se encarga de convertir la señal analógica obtenida en digital.
- f. Pantalla LCD, permite mostrar caracteres en sus 4 filas y 20 columnas, el voltaje de operación es de 3 a 5 V, con un consumo de corriente de 10 mA y cuenta con 16 pines de conexión. La pantalla LCD requiere de la librería del IDE de *Arduino LiquidCrystal* que permite acceder a las funciones para el manejo de la pantalla LCD. Este dispositivo contiene una memoria interna donde se encuentran almacenados los caracteres alfanuméricos; además, permite extender hasta 8 caracteres personalizados adicionales.
- g. Micro SD, este módulo permite al microcontrolador del Arduino almacenar y extraer información de la memoria micro SD, trabaja con 3.3 o 5 V, requiere la librería *SD.h* para establecer comunicación entre el microcontrolador del Arduino y la memoria SD. Las memorias SD son las más utilizadas por dispositivos portátiles gracias a su reducido tamaño y gran capacidad de almacenamiento.
- h. Leds, tiene la capacidad de emitir luz una vez que se energiza en polarización directa. Entre las ventajas de utilizar este tipo de componentes electrónicos se encuentran: el bajo consumo de energía requiere de una resistencia para limitar el paso de corriente, tiene un elevado tiempo de vida, consumen poca corriente de 10-20 mA dependiendo el color del LED y son muy utilizados en dispositivos que requieren iluminación debido a su bajo costo y gran variedad de colores.

En cuanto a los componentes mecánicos, el artefacto de compostaje consta de un biorreactor y los sistemas de mezclado, trituración, ventilación y aspersión.

3.2. Método propuesto

De acuerdo con Pressman, la ingeniería de software permite establecer un enfoque sistemático que garantice la calidad del sistema propuesto, siguiendo una metodología de desarrollo del sistema e implementado un conjunto de estándares (Pressman, 2010). El desarrollo de software para sistemas embebidos tiene la complejidad añadida de trabajar con hardware y software, puesto que, presenta el mismo nivel de importancia.

Para el desarrollo del presente trabajo de investigación, se decidió usar el método propuesto por Liggesmeyer y Trapp, este proceso permite a los desarrolladores producir implementaciones parciales después de cada iteración, siendo implementados primero los aspectos más críticos (Liggesmeyer & Trapp, 2009). Este modelo se seleccionó porque no se conocen inicialmente todos los requisitos, y los requisitos de funcionalidad y seguridad que se conocen son complejos. Por lo tanto, se busca ir mejorando el diseño mediante prototipos, ya que el tamaño del equipo de desarrollo es pequeño. La Figura

2, muestra el modelo de desarrollo de software seleccionado, en la parte superior de color verde se muestran las seis etapas asociadas a la parte del hardware y en la parte inferior de color azul se muestran las cinco etapas dedicadas al desarrollo de software. A continuación, se describe cada una de las etapas del proceso de desarrollo de software.

Figura 2 – Modelo de desarrollo de software para sistemas embebidos propuesto por Liggesmeyer y Trapp (Liggesmeyer, 2009).

Requerimientos: se llevó a cabo una encuesta dirigida al público en general con el objetivo de obtener la visión del cliente sobre los requerimientos del compostero inteligente. Posteriormente, se realizó una sesión grupal de lluvia de ideas con el fin de identificar requerimientos no detectados al analizar los datos recabados en la encuesta, se especificó la viabilidad del proyecto e identificaron los posibles riesgos. Para la documentación de los requerimientos se siguió el estándar IEEE 830-1998, al final se tuvo una lista de requerimientos clasificados como funcionales, no funcionales, seguridad e interfaz.

Diseño funcional: una vez que se definieron los requerimientos del sistema se crearon representaciones gráficas de los mismos usando el lenguaje de modelado unificado (UML), en específico los diagramas de casos de uso.

Arquitectura del sistema: se definió una arquitectura que soportara el sistema embebido del compostero doméstico inteligente. Para ello, se utilizó un diagrama de componentes, en el cual se ilustró la dependencia de los componentes electrónicos.

Arquitectura de software: se utilizaron los diagramas de vistas para agrupar en módulos las funciones que el sistema debe realizar.

Diseño de software: se modeló la comunicación entre los componentes del sistema de acuerdo con la secuencia en el tiempo de los eventos, para lo cual, se utilizaron los diagramas UML de secuencia.

Implementación: se realizó la implementación de cada uno de los componentes de software usando el Arduino IDE basado en lenguaje C y aplicando la Guía de sintaxis ANSI/ISO Estándar C++.

Pruebas de unidad: una vez que se codificó cada uno de los componentes de software, se realizaron por separado pruebas a cada uno de ellos, con el fin de identificar fallas en el código de cada componente de software, la técnica utilizada para comprobar el funcionamiento de los componentes es la clase de equivalencia.

Integración y pruebas de software: consistirá en integrar todos los componentes y se realizarán pruebas de integración de estos con el fin de detectar defectos en la comunicación e integración de los componentes, en esta etapa se utilizó la técnica de valor al límite.

Integración y pruebas del sistema: se realizará la integración de los componentes de hardware (electrónicos y mecánicos) y los componentes de software, de tal manera que el sistema completo quedará integrado aplicando la técnica de *botton-up* para poder realizar pruebas.

Pruebas funcionales: a partir de los requerimientos especificados se realizarán pruebas de cada una de las funciones del sistema.

Pruebas de validación: estas pruebas se realizarán una vez que el prototipo o sistema final está completo. Las pruebas las realizarán los usuarios del sistema con la finalidad de validar el producto.

4. Resultados

En este trabajo, se reportan los resultados de la primera iteración del método propuesto por Liggesmeyer y Trapp. Los resultados obtenidos son los siguientes: un documento que agrupa los requerimientos funcionales y no funcionales del sistema y un conjunto de diseños a diferentes niveles de abstracción, en los cuales se especifica el funcionamiento del sistema embebido. Los resultados obtenidos son los siguientes: un documento que agrupa los requerimientos funcionales y no funcionales del sistema, un conjunto de diseños a diferentes niveles de abstracción, en los cuales se especifica el funcionamiento del sistema embebido y un conjunto de pruebas diseñadas y aplicadas al sistema.

4.1. Etapa de requerimientos

Como resultado de la visión general del cliente y la lluvia de idea se identificaron y documentaron los requerimientos del sistema embebido usando el estándar *IEEE 830-1998*. En la Tabla 1 se listan los requerimientos especificados los cuales se muestran agrupados en las siguientes categorías: funcionales, no funcionales, seguridad e interfaz.

Funcionales	No funcionales	De seguridad	De interfaz
RF-01: Medir humedad	RNF-01: Sensor de temperatura ds18b20	RS-01: Acceso de seguridad a la trituradora	RI-01: Mostrar estado de las variables físicas con la utilización de LED
RF-02: Medir temperatura	RNF-02: Medir humedad higrómetro	RS-02: Activar paro de emergencia	RI-02: Mostrar datos del sistema en la pantalla LCD

Funcionales	No funcionales	De seguridad	De interfaz
RF-03: Medir oxígeno	RNF-03: Sensor de oxígeno	RS-03: Validar contraseña del usuario	RI-03: Mostrar el estado de los actuadores por medio de varios LED
RF-04: Medir volumen	RNF-04: Sensor de peso		RI-04: Implementación de teclado para realizar modificaciones al sistema
RF-05: Activar hélice	RNF-05: Lenguaje de programación C		
RF-06: Activar trituradora	RNF-06: Tipo de desechos orgánicos permitidos		
RF-07: Activar aspersores	RNF-07: Tiempo de almacenamiento de los datos en la memoria microSD		

Tabla 1 – Requerimientos del sistema propuesto

4.2. Diseño Funcional

A partir de los requerimientos obtenidos en la etapa 1, se generó el diagrama de casos de uso que se muestra en la Figura 3. En el diagrama se pueden apreciar los posibles escenarios que van a ser desarrollados por las funciones del compostero doméstico inteligente.

Figura 3 – Diagrama UML para la representación de casos de uso

Los escenarios se desarrollaron de acuerdo con las necesidades del proceso de compostaje. Dependiendo del valor obtenido por el sensor de temperatura, el sistema activará: el aspersor, el extractor, el ventilador y/o la hélice. Mientras que, dependiendo

de los valores obtenidos por el sensor de humedad, se activarán la hélice y el aspersor. La función del sensor de oxígeno es activar: el ventilador, la hélice y el extractor. El sensor volumétrico se encargará de activar una alarma visual. La función validar contraseña permitirá al usuario activar los actuadores del compostero doméstico inteligente. Por último, el usuario o el sensor de acceso a la trituradora pueden activar el paro de emergencia que consiste en desactivar la trituradora y la hélice.

4.3.Arquitectura del sistema

En la Figura 4, se muestra el diagrama de componentes del compostero doméstico inteligente diseñado de una manera abstracta. Este diagrama permite visualizar las dependencias entre cada uno de los componentes electrónicos. Los componentes que realizan la función de entrada de datos al sistema se muestran en la parte superior, estos son: (1) los sensores que se encargan de enviar datos a la placa electrónica y (2) los requerimientos de seguridad del sistema.

En la parte inferior de la imagen se observan los actuadores que son activados por el software embebido como respuesta a un evento, así como la memoria micro SD donde se guardará la información de los sensores. Por último, en la parte derecha del diagrama se muestran los componentes que corresponden a la interfaz, donde el usuario podrá monitorear el comportamiento del compostero doméstico en tiempo real.

Figura 4 – Diagrama para ilustrar los componentes que conforman el sistema embebido.

El diseño de los componentes del sistema embebido se realizó en un software especializado llamado *Proteus*. Este diseño permite visualizar la conexión entre los componentes de hardware y probar el funcionamiento. Se crearon módulos de cada una de las partes del sistema agrupando componentes dependiendo de la función que desempeñan, ver Figura 5.

4.4. Arquitectura de software

En la Figura 6, se muestra el diagrama de vistas que especifica los módulos de seguridad, interfaz, sensores, actuadores, almacenamiento y placa electrónica que conforman el

Figura 5 – Integración del hardware embebido para el compostero doméstico.

compostero doméstico inteligente. Cada uno de estos módulos está determinado por las funciones que realizan, así mismo, se especifican las relaciones existentes entre ellos.

Figura 6 – Diagrama para ilustrar los módulos que conforman el sistema embebido

Los sensores se encuentran dentro del compostero doméstico inteligente, estos obtienen los valores de las variables físicas y los envían a la placa electrónica donde se encuentra el software embebido almacenado en la memoria FLASH del microcontrolador. La placa electrónica contiene el software embebido del sistema que se encarga de obtener

los valores de los sensores y procesar la información; además, se encarga de enviar las señales de alerta y mensajes, así como también de activar los actuadores. El módulo de almacenamiento se encarga de registrar los valores en una memoria micro SD obtenidos por los sensores. La interfaz muestra los datos obtenidos por los sensores, así como el estado de cada una de las variables físicas que se están controlando. Además, muestra mensajes y alertas que pueden ser visualizados por el usuario. Los actuadores se activan dependiendo de los valores que se obtienen de los sensores; son activados cuando el software les envíe una señal de corriente a cada uno de los módulos donde se encuentran conectados. Finalmente, el módulo de seguridad permitió detener el funcionamiento de los actuadores durante un lapso de tiempo.

4.5. Diseño de software

En esta etapa, se obtuvo el diseño que especifica el comportamiento e interacción entre los componentes electrónicos del compostero doméstico inteligente de una manera ordenada y de acuerdo con la secuencia de tiempo en que ocurren los eventos. En la **Figura 7** se muestran los diagramas UML de secuencia, en los cuales se modelan las etapas (mesófila, termófila y de enfriamiento) del proceso de compostaje. En el interior del compostero doméstico inteligente, por medio de los sensores de temperatura, humedad, oxígeno y volumétrico se monitorea y controla de forma constante la transformación de los residuos sólidos orgánicos en composta, con la finalidad de mantener el sistema con las condiciones idóneas. Debido a esto, al inicio de cada una de las etapas se realiza la lectura de las variables por parte de los sensores y se determinan las acciones de los actuadores de cada etapa.

En la etapa 1, la mesófila, la temperatura debe permanecer en un rango de 25 a 45° C, la humedad en un 70%, la cantidad de oxígeno debe ser de 0.6 l/min*kg y la cámara de compostaje debe permanecer llena. En caso de detectar distintos valores a los establecidos se han determinados acciones que posibiliten conservar las condiciones ideales del compostero doméstico inteligente, estas son: (1) si la humedad es menor a 70%, se deben activar los aspersores, encender el indicador de nivel de humedad bajo (led azul), activar la hélice, y enviar un mensaje de alerta a la pantalla LCD; (2) en caso de detectar un nivel de oxígeno por debajo del establecido se deben activar la hélice y el ventilador, encender el indicador de nivel de oxígeno bajo (led azul) y enviar un mensaje de alerta a la pantalla LCD; y (3) si la cantidad de desechos orgánicos no cumple con el 70%, se debe enviar un mensaje a la pantalla LCD para que pueda ser visualizado por el usuario, véase Figura 7 (a).

En la etapa 2, la termófila, la temperatura tiende a aumentar debido a la actividad microbiana, por lo que el rango ideal para la temperatura está entre los 45 y los 70° C. La humedad se mantiene en un 70% y el oxígeno en 0.6 l/min*kg; asimismo, la cámara de compostaje debe permanecer con un nivel de medio a lleno. Las medidas correctivas en esta etapa son las mismas que las mencionadas en la etapa anterior (mesófila), con la única diferencia de que el volumen de la cámara de compostaje debe permanecer con un nivel de medio a lleno, véase Figura 7 (b).

En la etapa 3, la etapa de enfriamiento, la temperatura desciende de los 70° C, y la humedad disminuye. En caso de detectar valores anómalos, se enciende el indicador de humedad alta (led rojo), se activan el ventilador y extractor; asimismo, se envía un

Figura 7 – Diagramas de secuencia que ilustran las etapas del proceso de compostaje.
(a) Mesófila, (b) Termófila, (c) Enfriamiento.

mensaje de alerta a la pantalla LCD. El oxígeno debe mantenerse en 0.6 l/min*kg, la cámara de compostaje debe permanecer con un nivel de medio a lleno, véase Figura 7 (c).

4.6. Implementación

Las funciones de para obtener la temperatura, humedad y oxígeno se realizaron utilizando el lenguaje C. A continuación, se describe y presenta los algoritmos para cada una de esas funciones.

Para obtener la temperatura del compostero se utilizó el sensor *DS18B20* por medio de una función en el IDE de Arduino, esta función recibe un valor de tipo entero y determina si el sensor está dentro del rango correcto de funcionamiento. Si el valor no está dentro de los -55 a 125° C, se envía un mensaje al usuario para indicarle una falla del sensor; en cambio, si el sensor está dentro del rango permitido se devuelve el valor a la función, véase Figura 8 (a).

Figura 8 – Algoritmos para medir los parámetros físicos del compostero.
(a) Temperatura, (b) Oxígeno y (c) Humedad.

En la Figura 8 (b) se presenta el algoritmo para la función para obtener el oxígeno recibe un valor de tipo *float* y se verifica que se encuentre de 0 a 25% vol utilizando el sensor de oxígeno grove 2 conectado a una entrada analógica de la placa electrónica. Si se verifica un valor correcto se retorna a la función, en caso contrario se envía un mensaje que indica al usuario que el sensor está obteniendo datos incorrectos.

El sensor *FC-28* que permite obtener la humedad del compostero doméstico se encuentra conectado a una entrada analógica del Arduino, dicho valor es recibido por la función *obtenerHumedad()* y esta determina si el valor leído está dentro del rango de 0 a 1023; de ser así, el valor se retorna a la función, en caso contrario se manda una señal de alerta al usuario, véase Figura 8 (c).

Una vez retornados los valores de cada una de las funciones podrán ser utilizados para activar los demás componentes electrónicos que integran el hardware embebido del compostero doméstico.

4.7. Pruebas

Para la elaboración de la Estrategia de Pruebas se tomó como base los requerimientos obtenidos de la fase de análisis y el conjunto de esquemas que modelan el comportamiento del sistema en la fase de diseño. A partir de esta información se plantearon un conjunto de casos de prueba que permitan ingresar valores al sistema y analizar su comportamiento.

Se elaboraron un total de 40 casos de prueba repartidos de la siguiente manera: 10 pruebas unitarias, 12 pruebas de integración, 9 pruebas de sistema y 9 pruebas de validación.

En primer lugar, se llevaron a cabo las pruebas unitarias que consistió en comprobar el funcionamiento de cada uno de los componentes del sistema (hardware y software) de forma separada. Una vez concluidas las pruebas unitarias, se ejecutaron las pruebas de Integración, que consistieron en unir cada uno de los componentes del sistema de acuerdo con interacción que tienen entre ellos y que está estipulado en el diagrama de casos de uso. El resultado de cada una de las pruebas se registró en un documento que cuenta con los siguientes elementos: (1) una tabla para registrar la información de cada una de las pruebas y (2) una imagen representativa de cómo se conectaron los componentes electrónicos para poder ejecutar la prueba en turno.

En total se han ejecutado hasta el momento 37 pruebas, 12 de ellas corresponden a pruebas unitarias y 25 pruebas de integración, una vez aplicadas las técnicas de clase de equivalencia y la de valor al límite.

4.8. Comparación con otros sistemas de compostaje

En este artículo, se muestra la metodología para el desarrollo de un sistema embebido mediante el método de espiral y el método de prototipos para un compostero doméstico inteligente. Esta metodología ayuda a desarrollar la estructura del software con base a reglas preestablecidas a través de la experimentación, mejorando estas reglas en cada vuelta de la espiral hasta alcanzar todos los parámetros funcionales y de seguridad que requiere el equipo para la automatización del compostero doméstico inteligente que genere una composta de calidad, sea fácilmente manejado, y sea altamente seguro.

En la literatura se encuentran reportados varios equipos similares de compostaje (Pansari et al., 2019), estos también emplean hardware de Arduino y sensores de temperatura y humedad para el monitoreo del proceso de compostaje. En los casos de Wardhany et

al. y Elalami et al., los datos son transmitidos vía Web permitiendo el control remoto. El equipo de compostaje que presentan Pratama et al. procesa los datos de humedad y temperatura por medio de lógica difusa para la toma de decisiones en la operación del actuador del motor del equipo de compostaje (Elalami et al., 2019; Pratama et al., 2019; Wardhany et al., 2019).

A diferencia de los sistemas de composteo citados previamente, en esta investigación el uso de técnicas de ingeniería de software permite establecer un enfoque sistemático que ayuda a garantizar la calidad del sistema propuesto, siguiendo una metodología de desarrollo del sistema e implementado un conjunto de estándares. El desarrollo del software para el sistema embebido es muy importante, debido a la complejidad del equipo de compostaje, el cual además de incorporar el monitoreo y almacenamiento de datos de humedad, temperatura y oxígeno, realizará la toma de decisiones sobre los sistemas de: mezclado, trituración, ventilación y aspersión. Sin embargo, estas investigaciones proporcionan información relevante y proporcionaron ideas para implementar en esta investigación, tal como la monitorización vía remota empleando el internet de las cosas.

5. Conclusiones

Dada la problemática del manejo de los residuos sólidos, en el presente trabajo se propone la automatización de un compostero doméstico inteligente mediante un sistema embebido. El desarrollo del sistema embebido se basó en una metodología que combina el desarrollo de software en espiral con el método de prototipos, lo cual permite desarrollar de forma incremental el sistema propuesto.

Mediante la aplicación de una encuesta en línea y de sesiones de lluvia de ideas se lograron identificar los requerimientos del sistema. Estos requerimientos fueron especificados usando el estándar *IEEE 830-998* y se crearon modelos del diseño del software usando *UML*. Para el diseño de la integración de los diferentes componentes electrónicos y mecánicos del sistema embebido, se usó el software *Proteus*, y para la implementación de cada componente de software se usó el lenguaje *C*. La implementación de las pruebas permitió someter al sistema a fallos controlados para reafirmar el correcto funcionamiento del sistema, además de incorporar cada uno de los componentes de software y hardware, probando la comunicación entre ellos. Finalmente, se pudo modelar la integración de los componentes involucrados en el desarrollo del sistema embebido propuesto usando los diagramas de secuencia que se generaron con base a los requerimientos, para que se puedan llevar a cabo las tres etapas necesarias para el compostaje aerobio, i.e., mesófila, termófila y enfriamiento.

El uso de la metodología propuesta permite desarrollar el software embebido basado en reglas establecidas a través de la experimentación, mejorando estas reglas en cada vuelta de la espiral hasta alcanzar todos los parámetros funcionales y de seguridad que requiere el equipo para la automatización del compostero doméstico inteligente que genere una composta de calidad, sea fácilmente manejado, y sea altamente seguro.

Los resultados de este estudio proporcionan una contribución esencial para el entendimiento del campo en el desarrollo de sistemas embebidos como parte de una solución a la problemática de la transformación de los residuos sólidos orgánicos en

composta, utilizando un aparato de compostaje doméstico y basado en técnicas de ingeniería de software como parte del control de calidad del proceso.

Como trabajo futuro queda realizar las dos iteraciones restantes del proceso de desarrollo de software seleccionado y poder obtener el prototipo final.

Referencias

- Barr, M., & Massa, A. (2006). *Programming Embedded Systems* (2nd ed.). Sebastopol, CA: O'Reilly.
- Boehm, B. W. (1984). Verifying and Validating Software Requirements and Design Specifications. *IEEE Software*, 1(1), 75–88. <https://doi.org/10.1109/MS.1984.233702>
- Brown, S. (2000). Overview of IEC 61508. Design of electrical/electronic/programmable electronic safety-related systems. *Computing & Control Engineering Journal*, 11(1), 6–12. <https://doi.org/10.1049/cce:20000101>
- Cesaro, A., Belgiorno, V., & Guida, M. (2015). Compost from organic solid waste: Quality assessment and European regulations for its sustainable use. *Resources, Conservation and Recycling*, 94, 72–79. <https://doi.org/10.1016/j.resconrec.2014.11.003>
- Das, M., Uppal, H. S., Singh, R., Beri, S., Mohan, K. S., Gupta, V. C., & Adholeya, A. (2011). Co-composting of physic nut (*Jatropha curcas*) deoiled cake with rice straw and different animal dung. *Bioresource Technology*, 102(11), 6541–6546. <https://doi.org/10.1016/j.biortech.2011.03.058>
- Delgado Peralta, M. (2012). Norma ambiental para el Distrito Federal. *Gaceta Oficial del Distrito Federal*, NADF-020-AMBT-2011, MX.
- Douglass, B. P. (2004). *Real Time UML: Advances in the UML for Real-Time Systems*. Addison-Wesley.
- Elalami, M., Baskoun, Y., Zahra Beraich, F., Arouch, M., Taouzari, M., & Qanadli, S. D. (2019). Design and Test of the Smart Composter Controlled by Sensors. 2019 7th International Renewable and Sustainable Energy Conference (IRSEC), (4), 1–6. <https://doi.org/10.1109/IRSEC48032.2019.9078197>
- Liggesmeyer, P., & Trapp, M. (2009). Trends in Embedded Software Engineering. *IEEE Software*, 26(3), 19–25. <https://doi.org/10.1109/MS.2009.80>
- Malinowski, A., & Yu, H. (2011). Comparison of Embedded System Design for Industrial Applications. *IEEE Transactions on Industrial Informatics*, 7(2), 244–254. <https://doi.org/10.1109/TII.2011.2124466>
- Ministerio de la Presidencia (2013). Real Decreto 506/2013. *Boletín Oficial del Estado* No 164, Sec. I.
- Oshana, R., & Kraeling, M. (2019). *Software Engineering for Embedded Systems* (2nd ed.). Newnes.

- Pansari, N. B., Deosarkar, S. B., & Nandgaonkar, A. B. (2019). Smart Compost System. Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems, 597–600. <https://doi.org/10.1109/ICCONS.2018.8663206>
- Pratama, Y. F., Ariyanto, E., & Karimah, S. A. (2019). Prototype of Automation of Organic Fertilizer Manufacturing Systems Based on Internet of Things. 2019 7th International Conference on Information and Communication Technology (ICoICT), 1–6. <https://doi.org/10.1109/ICoICT.2019.8835299>
- Pressman, R. S. (2010). Ingenieria Del Software. Un Enfoque Practico (7ma ed.). Mc Graw Hill.
- Sassone, A., Grosso, M., Poncino, M., & Macii, E. (2016). Smart Electronic Systems: An Overview. In Smart Systems Integration and Simulation (pp. 5–21). https://doi.org/10.1007/978-3-319-27392-1_2
- Wardhany, V. A., Hidayat, A., A., M. D. S., Subono, Afandi, A., Muhammad Doni Sururin, A., & Afandi, A. (2019). Smart Chopper and Monitoring System for Composting Garbage. 2019 2nd International Conference of Computer and Informatics Engineering (IC2IE), 74–78. <https://doi.org/10.1109/IC2IE47452.2019.8940812>