

# Un enfoque desde la ingeniería de software a una solución tecnológica de IoT y aprendizaje automático que permita monitorear y controlar las variables ambientales en un cultivo de café.

Kevin Alexander Tipán Pérez

October 2023

## Resumen

En este artículo, se abordará el caso de una solución IoT y su enfoque de aprendizaje automático para supervisar y controlar las variables ambientales en un cultivo de café. Esto nos permite visualizar cómo interactúan ciertos componentes del sistema con los usuarios y cómo se comportan en relación con otros artefactos o dispositivos que conforman esta solución tecnológica.

Para lograr estos objetivos, hemos optado por la implementación de ingeniería de software con un enfoque en la conceptualización y el comportamiento del sistema. Hemos elegido utilizar el Lenguaje Unificado de Modelado (UML) para representar los elementos clave de la solución tecnológica tanto de manera estática a través de diagramas de casos de uso como de manera dinámica a través de diagramas de secuencia

## 1 Introducción

Se podría definir la ingeniería de software como la aplicación práctica del conocimiento científico al diseño y construcción de programas informáticos y la documentación asociada necesaria para desarrollarlos, operarlos y mantenerlos. Esto también se conoce como desarrollo de software o producción de software, es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, es decir, la aplicación de la ingeniería de software.

## 2 Materiales y métodos

### 2.1 Metodología de investigación

En la fase de revisión, se realizó una revisión de diferentes documentos relacionados con el tema propuesto, con el objetivo de establecer los artefactos o

componentes que representan las funcionalidades más destacadas e importantes de la solución tecnológica[1].

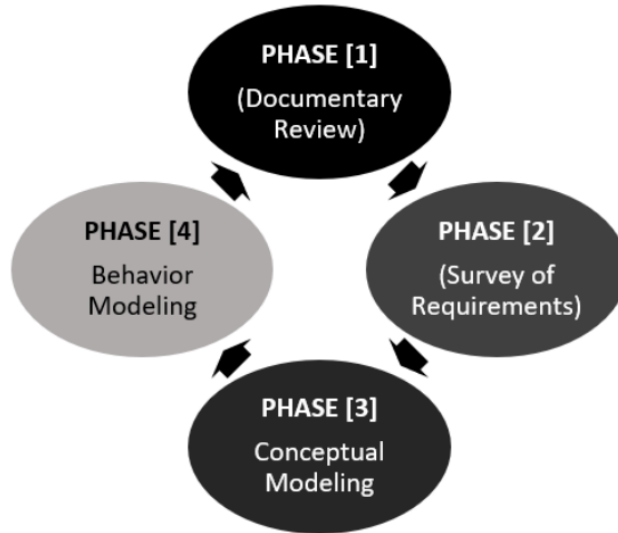


Figure 1: Fases de la investigación propuesta para el desarrollo del proyecto

De acuerdo con la información recopilada para determinar y documentar los requisitos se aplica la encuesta de requisitos

También se presenta un diagrama de casos de uso general, donde se visualizan los principales procesos de la solución tecnológica y las interacciones.[7]

Mediante la elaboración de los diagramas del comportamiento de la solución tecnológica, se da el modelado de comportamiento, para estos se da un diagrama de componentes y un diagrama de despliegue[5].

## 2.2 Descripción de la arquitectura.

Su diseño tiene un enfoque de la arquitectura de la red inalámbrica, tanto sus componentes y dispositivos, el cual trata de una red de sensores inalámbricos (WSN), que está formada por una serie de dispositivos distribuidos de forma autónoma en un área en crecimiento, la cual consta de tres nodos sensores, y cada uno de estos nodos consta de una tarjeta programable Lucy3, a la que se conectan un sensor de temperatura y humedad y un sensor de humedad del suelo.[8] En la comunicación entre los nodos sensores y el Gateway encontramos varios estándares que cumplen con las características y condiciones necesarias, pero se seleccionó el protocolo o estándar ZigBee 802.15.4, en su mayoría por factores económicos, ZigBee comprende una asociación de industrias que trabajan juntas para desarrollar estándares y productos, ya que se basa en el estándar IEEE 802.15.4 para redes de área personal inalámbricas o WPAN,

está tecnología integrada en una amplia gama de productos y aplicaciones para consumidores comerciales, industriales y gubernamentales[2].

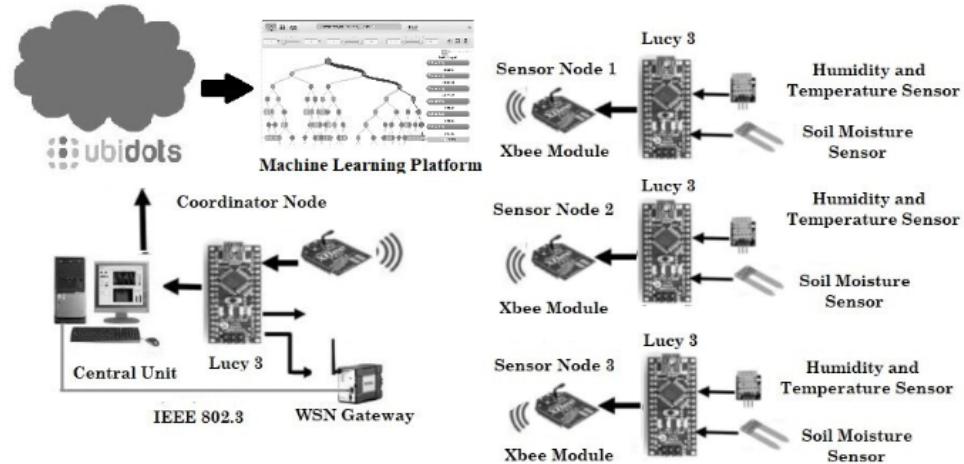


Figure 2: Arquitectura de la red de sensores inalámbricos y las plataformas en la nube

### 2.3 Requisitos del sistema.

Hace referencia a una condición o necesidad del usuario para resolver un problema y alcanzar una meta, la cual afirma que los requisitos son las capacidades y condiciones con las que debe identificarse el sistema propuesto, también definen como la rama de la ingeniería de software que se ocupa del establecimiento de los objetivos, funciones y restricciones de los sistemas de software, así como de la relación entre estos factores para establecer especificaciones precisas, podemos decir que es un proceso mediante el cual se intercambian diferentes puntos de vista entre usuarios, clientes, patrocinadores y miembros de un equipo de desarrollo, con el fin de recopilar y modelar qué y cómo funciona el sistema propuesto debe realizar.[6]

Este proceso utiliza un conjunto de métodos, herramientas y actores que generan un modelo a partir del cual se crea un documento de requisitos, por lo que se deben considerar los requisitos no funcionales, que pueden definirse como aquellos criterios o aspectos del sistema que no tienen una relación directa con su comportamiento funcional[3].

### 2.4 Modelado conceptual.

Aquí se utilizara Uml el cual es un estándar que ha sido adoptado internacionalmente por numerosas organizaciones y empresas para crear diagramas y docu-

mentación relacionada con el desarrollo de software (programas informáticos). Cabe aclarar que, en este punto, esto es también parece importante conocer, desde el punto de vista de diversos expertos en el tema, diferentes definiciones de lo que es un caso de uso, ya que es una técnica para capturar requisitos potenciales de un nuevo sistema o actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo el sistema debe interactuar con el usuario o con otro sistema para lograr un objetivo específico[4].

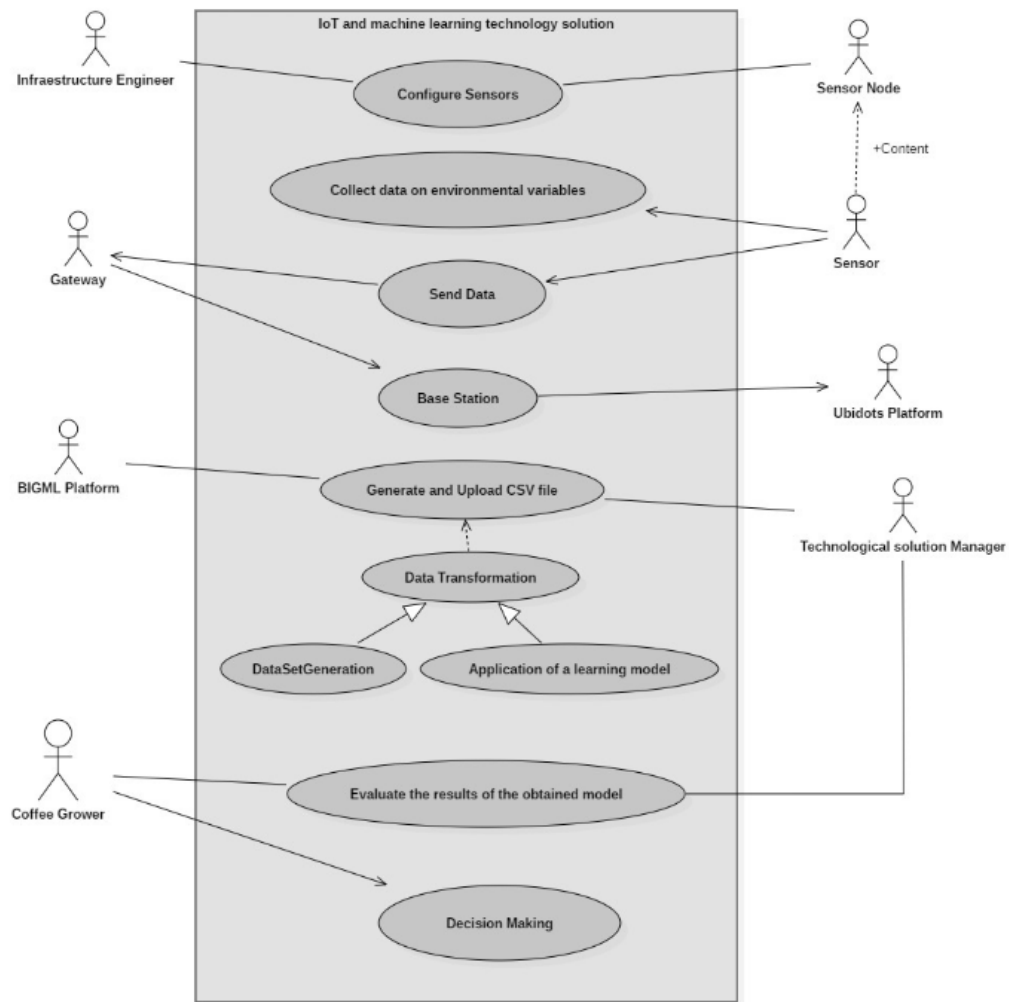


Figure 3: Diagrama de casos de uso principal de la solución tecnológica.

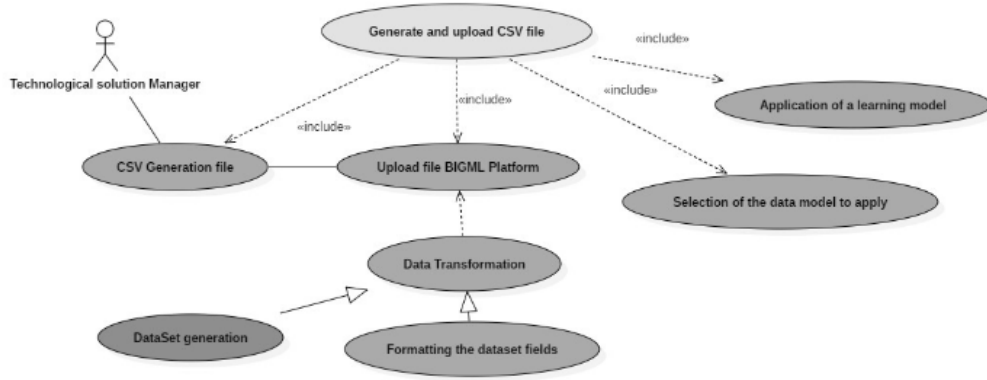


Figure 4: Caso de uso: generación y carga de archivos CSV.

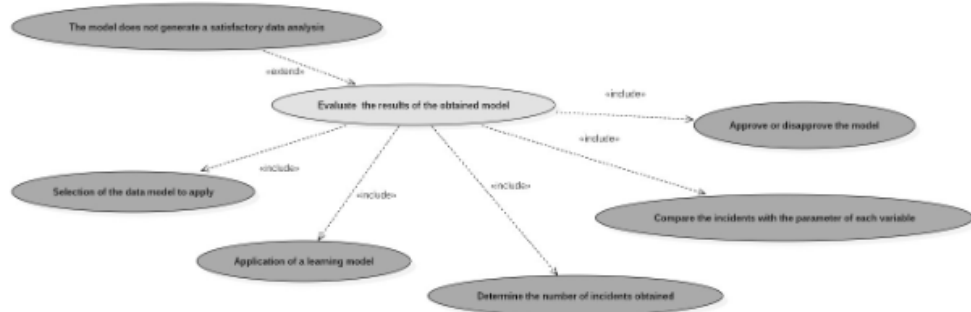


Figure 5: Evaluación de los resultados obtenidos del modelo.

## 2.5 Modelado de comportamiento.

El modelado de comportamiento de software nos permite determinar cómo se relacionan y se comportan ciertos objetos del sistema, los cuales son estáticos como en el modelado conceptual, sino que tienen un comportamiento dinámico, ya que se enfatiza que el objetivo de los modelos dinámicos es presentar o describir artefactos que exhiben el comportamiento del sistema a lo largo del tiempo[5].

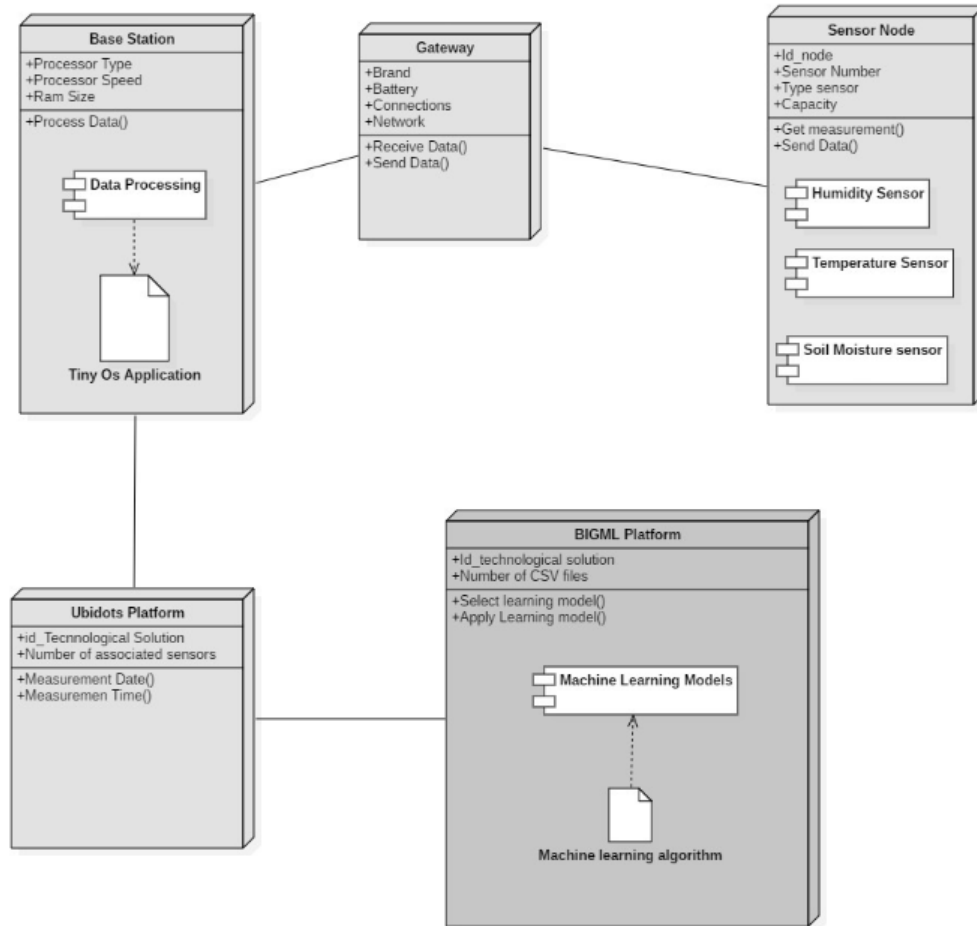


Figure 6: Diagrama de despliegue de soluciones tecnológicas.

### 3 Resultados

A través de diagramas de casos de uso general y subcasos de uso, se pueden identificar los principales procesos de la solución y los actores involucrados, lo que es crucial para garantizar que los requisitos funcionales del sistema se cumplan según las necesidades de las partes interesadas, destaca la relevancia del diagrama de despliegue para comprender en detalle cómo interactúan los componentes, tanto de hardware como de software, en la solución tecnológica. Se menciona el valor de utilizar el lenguaje de modelado UML, que permite exam-

inar procesos e interacciones desde diferentes perspectivas y facilita la identificación de relaciones entre usuarios y procesos. A través del modelado UML que en cifras o estadísticas de rendimiento que no son el objeto central de estudio, se logró demostrar que la aplicación de la ingeniería de software a través del modelado UML permite mostrar aspectos de gran importancia en la solución tecnológica, como lo son los diferentes actores involucrados en los procesos a través de casos de uso.

## References

- [1] B. Boehm. Una visión de la ingeniería de software de los siglos xx y xxi. In *28° Int. Conf. Suave. Ing. (CISE '06)*, page 466, mayo 2006.
- [2] ND Castro C, LE Chamorro F, and CA Viteri M. Una red de sensores inalámbricos para la automatización y control del riego localizado. *Revista de ciencias agrícolas*, 33(2):106–116, agosto 2016.
- [3] IEEE. *Guía IEEE para descripciones de diseño de software*. IEEE, 1993.
- [4] C. Larman. *UML y patrones*. Pearson, 2003.
- [5] L. Ramírez. *Diseño de una arquitectura para redes de sensores con soporte para aplicaciones de detección de eventos*. PhD thesis, Universidad Politécnica de Valencia, 2012.
- [6] M. Rossainz-López. *Diseño orientado a objetos*. Universidad Autónoma de Puebla, 2012.
- [7] E. Rozic and S. Herzovich. *UML y desarrollo de software orientado a objetos*. Universidad de San Andrés, 2016.
- [8] J. Rumbaugh, I. Jacobson, and G. Booch. *El lenguaje unificado de modelado: manual de referencia*. Addison Wesley, 2000.