

Problem-Based Learning as a Pedagogical Method for Enhancing Competencies Through a GPS-Based Internet of Things Project.

Davalos-Villarreal, Rafael Emilio ¹, Hernandez-Herrera, Diego ¹

Tecnologico de Monterrey, Escuela de Ingeniería y Ciencias, Ave. Eugenio Garza Sada 2501, Monterrey 64849, NL, Mexico
email: rafailemiliodavalosvillarreal@tec.mx, A01198786@tec.mx

Abstract – This study explores the implementation of Problem-Based Learning (PBL) as a pedagogical method for enhancing engineering, technical, and problem-solving competencies in students enrolled in an engineering course. A GPS (Global Positioning System)-based project is presented among several developed in the Internet of Things course. Through practical projects, students developed their competencies and problem-solving skills. The work presented in this paper refers to a GPS-based system, where real-world challenges allowed students to apply theoretical concepts to practical situations. The research demonstrates how PBL fosters autonomous learning while enabling students to develop essential technical and practical skills for their professional training. The results suggest that this approach significantly improves students' understanding of emerging technologies, while also enhancing their ability to solve complex problems in real-world contexts. This paper highlights the pedagogical implications of PBL in higher education and its potential to transform traditional teaching in technical disciplines.

Keywords—*Problem-Based Learning, Autonomous Learning, Higher Education, Educational Innovation.*

El Aprendizaje Basado en Problemas (ABP) como método pedagógico para el enriquecimiento de las competencias mediante un proyecto de Internet de las Cosas basado en GPS.

Davalos-Villarreal, Rafael Emilio ¹, Hernández-Herrera, Diego ¹

Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias, Ave. Eugenio Garza Sada 2501, Monterrey 64849, NL, Mexico
email: rafaelemiliodavalosvillarreal@tec.mx, A01198786@tec.mx

Resumen— Este estudio explora la implementación del Aprendizaje Basado en Problemas (ABP) como método pedagógico para el enriquecimiento de las competencias ingenieriles, técnicas y de resolución de problemas de estudiantes en un curso de ingeniería. Se presenta un proyecto basado en GPS (Global Positioning System) de varios que se desarrollaron en el curso de Internet de las Cosas. A través de proyectos prácticos los estudiantes desarrollaron sus competencias y resolución de problemas. El trabajo que se presenta en este documento se refiere a un sistema basado en GPS, en el cual hubo desafíos reales que permitieron aplicar los conceptos teóricos a situaciones prácticas. La investigación muestra cómo el ABP fomenta el aprendizaje autónomo, mientras les permite desarrollar habilidades técnicas y prácticas esenciales para su formación profesional. Los resultados obtenidos sugieren que este enfoque mejora significativamente la comprensión de los estudiantes sobre tecnologías emergentes, además de potenciar su capacidad para resolver problemas complejos en contextos reales. Este trabajo destaca las implicaciones pedagógicas del ABP en la educación superior y su potencial para transformar la enseñanza tradicional en disciplinas técnicas.

Palabras clave—Aprendizaje basado en problemas, Aprendizaje autónomo, innovación educativa, educación superior.

I. INTRODUCCIÓN

El objetivo del presente trabajo es mostrar de qué manera el Aprendizaje Basado en Problemas (ABP) permite enriquecer la enseñanza de la ingeniería dentro de las instituciones de educación superior mediante la descripción de uno de los proyectos que se basó en tecnología Global Positioning System (GPS) del curso de Internet de las Cosas.

El Aprendizaje Basado en Problemas (ABP) es un valioso método de enseñanza que utiliza problemáticas específicas para iniciar, motivar y promover la comprensión de los materiales de curso, la aplicación de conceptos teóricos y el desarrollo de habilidades profesionales [1]. Las exigencias de este método no se limitan simplemente a que el alumno sea capaz de “abordar un problema de manera objetiva, crítica y creativa” [2], sino que además se requieren habilidades de comunicación, negociación y toma de decisiones que permitan

determinar las expectativas y definir las metas de una solución de implicaciones reales.

II. MARCO TEÓRICO

El ABP además de servir como un marco metodológico, también sirvió como un impulsor para la integración efectiva de los contenidos de Internet de las Cosas (IoT, por sus siglas en inglés de Internet of Things). Cada fase del proyecto exigió a los estudiantes diagnosticar problemas reales, formular hipótesis de solución, investigar tecnologías pertinentes y aplicar soluciones técnicas bajo condiciones de incertidumbre, tomando áreas de la vida profesional en las que pueden aplicar las tecnologías IoT.

El curso TC1004B Implementación de Internet de las Cosas del Tecnológico de Monterrey es un bloque multidisciplinar que incluye módulos de circuitos digitales, bases de datos, administración de proyectos e internet de las cosas. Se realiza un reto que tenga que ver con la realidad de la sociedad, y puede ser de diversos ámbitos. Los estudiantes eligen los proyectos en los cuales van a trabajar durante el curso. el Aprendizaje Basado en Problemas (ABP) es una de la metodología activa y centrada en el estudiante. Aborda problemas auténticos o situaciones del mundo real, también fomenta el aprendizaje colaborativo e interdisciplinario. También entre sus múltiples beneficios ayuda a desarrollar el pensamiento crítico y en la búsqueda de soluciones.

Se ha aplicado el Aprendizaje Basado en Problemas en una serie de proyectos [1], [2], así como proyectos de internet de las cosas [3] que solucionan problemas del mundo real.

Se describirá y analizará un caso de éxito ocurrido dentro del curso TC1004B Implementación de Internet de las Cosas, impartido a los alumnos de tercer semestre de distintas ingenierías de computación, incluyendo la Ingeniería en Tecnologías Computacionales, del Tecnológico de Monterrey durante el semestre de otoño de 2024. A través de este caso, se explorarán las metodologías y enfoques pedagógicos empleados en la enseñanza del Internet de las Cosas (IoT), con un énfasis particular en el Aprendizaje Basado en Problemas

(ABP), su impacto en el desarrollo de competencias técnicas y la resolución de problemas en contextos reales.

El contenido temático del curso TC1004B incluye una serie de conceptos fundamentales tanto de la computación como de Internet de las Cosas (IoT, por sus siglas en inglés), tales como funciones booleanas, minitérminos y maxitérminos, mapas de Karnaugh, sumadores y restadores, codificadores y decodificadores, multiplexores, biestables (flip-flops), registros de memoria, circuitos secuenciales y combinacionales, arquitectura de IoT, bases de datos, lenguaje de consulta estructurado (SQL) y la creación de Interfaces de Programación de Aplicaciones (APIs), entre otros. Estos contenidos representaron los fundamentos teóricos esenciales que se impartieron durante las clases, proporcionando una base sólida para la comprensión de las competencias de la computación y su aplicación práctica en el ámbito del Internet de las Cosas. Además, los estudiantes adquirieron conocimientos adicionales, tanto técnicos como transversales, como resultado del reto de final del curso. El reto se formó por un conjunto de entregables. Dichos entregables incluyeron un documento de requerimientos, un prototipo de hardware de IoT destinado a la recolección de datos, y un software desarrollado de forma autónoma que permitiera la gestión del mismo.

Uno de los coautores, estudiante de la carrera de ITC (Ingeniero en Tecnologías Computacionales) adquirió la habilidad de realizar la codificación de un controlador de dispositivo (driver) mientras trabajaba con un módulo GPS (Global Positioning System) para ajustar las necesidades específicas del proyecto. Esto requirió que aprendiera de manera autodidacta el funcionamiento de la comunicación serial de tipo UART (Receptor-Transmisor Asíncrono Universal), así como el procesamiento de cadenas de texto a partir de un flujo continuo y no uniforme de caracteres ASCII. Además, desarrolló la capacidad de interpretar algorítmicamente los enunciados NMEA (especificación eléctrica y de datos para comunicación entre equipos electrónicos), fundamentales para la extracción y análisis de datos provenientes del módulo GPS [3]. Los enunciados NMEA son de distintos tipos, incluyendo: *Global Positioning System Fix Data* (GPGGA), *Recommended Minimum Course and speed* (GPRMC) y *Geographic Position, Latitude/Longitude* (GPGLL). Cada tipo provee los mismos datos básicos de posicionamiento, pero cada uno añade información particular.

Estos conocimientos, adquiridos fuera del marco curricular, son producto de imprevistos ocurridos durante el desarrollo del proyecto porque ninguno de los 3 módulos GPS que fueron adquiridos funcionaba adecuadamente. Además, las bibliotecas de código que implementaban los controladores de comunicación necesarios para conectarse con los módulos

tampoco ofrecían ninguna configuración para depurar (*debug*) el programa. Por esta razón, se vio en la necesidad de desarrollar su propio controlador con el propósito de diagnosticar y resolver el problema, lo que permitió garantizar la correcta integración del módulo GPS y cumplir con la entrega del proyecto final dentro del plazo establecido.

Es importante desarrollar la capacidad de resolver problemas que se presentan en la vida profesional y generar resultados adecuados y coherentes con el proyecto desarrollado. El reto de Internet de las cosas es ideal porque reúne diferentes áreas del conocimiento y problemas reales. El Aprendizaje Basado en Problemas (ABP) es una herramienta pedagógica con muy alto valor en este campo.

Se describirá en detalle el desarrollo del proyecto final del estudiante y, posteriormente, se analizarán las implicaciones pedagógicas derivadas de su experiencia. El objetivo de este análisis es demostrar cómo el Aprendizaje Basado en Problemas (ABP) amplía la enseñanza en el aula de ingeniería a nivel superior, trascendiendo el contenido teórico del programa académico y fomentando el desarrollo de habilidades prácticas y autónomas en la resolución de problemas reales.

III. DESARROLLO

A. Disposición y arquitectura del proyecto

El proyecto final del estudiante consistió en el desarrollo de un sistema de posicionamiento que, a través de la transmisión de datos GPS desde un microcontrolador, permite obtener la ubicación en tiempo real de los senderistas a lo largo de sus rutas, con el objetivo de registrar su progreso y aumentar su seguridad. En caso de requerir asistencia, las posiciones registradas a lo largo del tiempo proporcionan información valiosa, lo que facilita la elaboración de un plan de emergencia más preciso y mejora la probabilidad de éxito en la intervención. Adicionalmente, el sistema permite a los usuarios configurar una ruta predeterminada, de manera que, en caso de desviación, el microcontrolador emite una alerta a través de un indicador LED (diodo emisor de luz), advirtiendo al usuario sobre el desvío de la trayectoria.

Para la construcción de este sistema, se requiere la integración de diversos componentes, además del microcontrolador. En lo que respecta a los componentes físicos, es necesario disponer de un módulo receptor GPS, un *protoboard* con sus cables puente, y un LED. En cuanto al software, es necesario desarrollar el código de programación del microcontrolador, que se encargará de enviar y recibir los datos de posición mediante peticiones HTTP (HyperText Transfer Protocol). Además, se debe diseñar y codificar una API (Interfaz de Programación de Aplicaciones) que reciba los datos de ubicación, los almacene en una base de datos y envíe

las posiciones correspondientes a la ruta predeterminada. Finalmente, es necesario crear un programa de cliente de internet (web) que permita a los usuarios gestionar los datos e interactuar con el sistema de manera segura y eficiente, ofreciendo una interfaz atractiva y funcional.

En lo que respecta al microcontrolador, se seleccionó la placa de desarrollo programable ESP8266, debido a que es parte del material ofrecido de manera inmediata en el laboratorio del Tecnológico de Monterrey. Además, cumple con el requisito de capacidad inalámbrica [4]. Esta placa cumple con los estándares de telecomunicación IEEE 802.11 b/g/n [4], lo que le permite conectarse a Internet a través de redes WLAN (Redes de Área Local Inalámbrica) que operan en las bandas de 2.4 GHz y 5.0 GHz [5].

El receptor GPS adquirido fue el módulo GT-U7, seleccionado debido a su formato compacto y bajo costo, lo que facilita un desarrollo ágil del sistema. Este módulo opera con una tensión de alimentación que oscila entre 2.70 y 3.60 voltios, y soporta una salida de datos a través de la interfaz UART (Universal Asynchronous Receiver-Transmitter) utilizando el protocolo de comunicación GPS NMEA 0183 [6]. Estas características lo hacen altamente compatible con la placa ESP8266, garantizando una integración eficiente entre ambos componentes.

El *protoboard*, los cables puente, el LED y el cable USB (Universal Serial Bus) utilizados para conectar la placa con una laptop fueron proporcionados en el kit de desarrollo suministrado por la universidad. Aunque no se incluyeron especificaciones detalladas, estos componentes demostraron ser totalmente compatibles con los requerimientos del proyecto.

Para el desarrollo del software, se eligió el *framework* Astro [7], dado que permite implementar tanto la API como el cliente web dirigiendo al usuario a un único repositorio de código, optimizando el proceso de desarrollo. En cuanto a la gestión de datos, se optó por PostgreSQL debido a su reconocida estabilidad, robustez y facilidad de implementación [8]. En relación con el código destinado a la placa de desarrollo, se siguió una configuración estándar en el ámbito del Internet de las Cosas (IoT), utilizando el lenguaje de programación C++ [9] y el entorno de desarrollo integrado Arduino IDE.

B. Codificación

La codificación de los diversos componentes de software representó una de las tareas más exigentes del proyecto, tanto en términos de tiempo como de esfuerzo invertido. Esta actividad se estructuró en varias etapas clave, comenzando con la programación del microcontrolador, seguido por la

configuración de la base de datos y la API, y finalizando con el desarrollo de la interfaz visual destinada al usuario.

Fue en esta fase del proyecto donde surgieron los problemas imprevistos más significativos, los cuales requirieron ajustar el enfoque inicial de la solución. En particular, al iniciar el trabajo con el módulo GPS GT-U7, se presentó un inconveniente: la consola serial de la laptop conectada a la placa ESP8266, que debía mostrar los datos correspondientes a la latitud y longitud del dispositivo, permanecía en blanco.

Inicialmente, se supuso que la antena pasiva del módulo no poseía la sensibilidad suficiente para interceptar las transmisiones de los satélites GPS. Por este motivo, en lugar de depurar el problema dentro del aula, se intentó resolverlo en un entorno al aire libre, con la esperanza de mejorar la recepción de las señales. Sin embargo, el módulo no logró establecer conexión con ningún satélite.

Ante esta situación, se optó por adquirir dos módulos GT-U7 adicionales, idénticos en la especificación, pero de diferente fabricante. Se eligió un proveedor de mayor calidad, con la expectativa de que esto solucionara la falta de conexión. Este enfoque resultó ser parcialmente exitoso, ya que, eventualmente, se registraron dos momentos en los que los enunciados NMEA comenzaron a llenar la consola serial. No obstante, dichos eventos fueron breves y esporádicos.

En consecuencia, se decidió descartar las diversas bibliotecas de C++ que se habían empleado para la comunicación con el módulo, como TinyGPSPlus [10] y 107-Arduino-NMEA-Parser [11]. Esta decisión estuvo motivada por dos razones: en primer lugar, ambas bibliotecas —al igual que la mayoría de las bibliotecas de GPS para IoT— no son oficialmente compatibles con la placa ESP8266, lo que podría comprometer el funcionamiento correcto del prototipo. En segundo lugar, ninguna de las bibliotecas utilizadas o las disponibles en línea ofrecían la posibilidad de configurar el entorno de desarrollo de tal manera que fuera posible depurar o inspeccionar el flujo crudo de datos emitidos por el módulo GT-U7 a través de UART. Esta limitación dificultó la identificación del origen del problema, ya que resultaba imposible determinar si este se originaba en el propio módulo o en algún otro proceso interno de las librerías.

Con el objetivo de localizar la falla, se decidió desarrollar un controlador propio, también conocido como *driver*, cuya función es precisamente la de recibir, interpretar y abstraer la información cruda emitida por un dispositivo periférico (hardware) [12].

El primer paso consistió en sustituir el editor de código Arduino IDE, que formaba parte del planteamiento inicial de la solución, por el entorno de desarrollo Visual Studio Code

[13] (también conocido como VS Code). Esta nueva herramienta, a diferencia de la anterior, facilita la distribución del código en distintos archivos de C++, permitiendo una mejor modularización de la lógica del controlador. Además, incorpora una excelente plataforma de desarrollo para microcontroladores, denominada PlatformIO [14], que se encargó de compilar y depurar el código.

Se procedió a modularizar la lógica del controlador en tres clases distintas, cada una encargada de una función específica:

- **TrekkGPS:** procesa e interpreta la información serial enviada por el módulo GT-U7 a través de UART.
- **NetEngine:** gestiona la conexión a Internet, la encapsulación de los datos en un formato adecuado para la red y las solicitudes hacia la API.
- **SpatialProcessor:** realiza los cálculos matemáticos necesarios para determinar la distancia entre la posición actual del dispositivo y el punto más cercano de la ruta de excursión predeterminada por el usuario, con el fin de detectar y alertar cualquier desvío.

El archivo main.cpp es el encargado de importar estos módulos y coordinar su interacción. Al ejecutarse, la función main realiza una solicitud a la API para cargar en memoria los datos de la ruta predeterminada. Posteriormente, inicia la captura de la posición del dispositivo mediante el módulo GPS y procede a enviar los datos al servidor para su correspondiente almacenamiento.

Durante la codificación de este controlador, se presentaron varios problemas imprevistos que, aunque provocaron una demora en el desarrollo del proyecto final, contribuyeron significativamente al proceso de aprendizaje. Uno de los incidentes más relevantes ocurrió en la clase TrekkGPS.

Inicialmente, dado que el flujo de información del módulo GPS era nulo o escaso, se diseñó la clase TrekkGPS para procesar un único enunciado NMEA a la vez. Es decir, dado que cada enunciado NMEA comienza con un símbolo de dólar ('\$') y finaliza con un carácter nulo ('\n'), el algoritmo de TrekkGPS recibía los caracteres ASCII enviados por el módulo GPS uno por uno, procesando cada cadena de texto hasta encontrar su carácter nulo. Estos caracteres ASCII —que son simplemente caracteres de texto pertenecientes a un estándar para el intercambio digital de información [15]— se almacenaban en un arreglo de caracteres con una longitud fija de 128 elementos.

Esta disposición lógica funcionaba correctamente debido a que los enunciados NMEA producidos por el módulo aparecían de manera esporádica y ordenada. Sin embargo, durante el desarrollo del controlador, se logró resolver los

problemas de conexión del módulo, lo que resultó en un flujo continuo y desordenado de enunciados NMEA. La rapidez con la que se sucedían estos enunciados era tal que, en varias ocasiones, el carácter nulo no se emitía hasta que ya se habían transmitido múltiples enunciados. Esto provocó que la clase TrekkGPS ejecutara el algoritmo de procesamiento —sin éxito— sobre una serie de enunciados NMEA simultáneos, cuando estaba diseñada para interpretar solo uno a la vez.

Este problema imprevisto obligó al alumno a idear una solución tanto creativa como técnica: desarrolló un nuevo algoritmo capaz de detectar la omisión de caracteres nulos y almacenar los diferentes enunciados NMEA en una matriz bidimensional de caracteres. Esta matriz constaba de 3 filas y 128 columnas, donde cada fila almacenaría un enunciado. En otras palabras, el nuevo algoritmo permite procesar simultáneamente hasta 3 enunciados NMEA por ejecución, incluso cuando no se emita ningún carácter nulo para distinguirlos.

Este problema imprevisto condujo a idear una solución tanto creativa como técnica: desarrollar un nuevo algoritmo capaz de detectar la omisión de caracteres nulos y almacenar los diferentes enunciados NMEA en una matriz bidimensional de caracteres. Esta matriz consta de 3 filas y 128 columnas, donde cada fila almacena un enunciado. En otras palabras, el nuevo algoritmo permite procesar simultáneamente hasta 3 enunciados NMEA por ejecución, incluso cuando no se emita ningún carácter nulo para distinguirlos.

En la Tabla 1 se presentan los comandos correspondientes a la versión que procesa un solo enunciado NMEA por ejecución, así como a la versión que procesa múltiples enunciados NMEA en cada ejecución.

TABLA 1
IMPLEMENTACIONES DEL ALGORITMO DE INTERCEPTACIÓN Y ALMACENAMIENTO DE ENUNCIADOS NMEA EN LA CLASE TREKKGPS

Versión	Código escrito en C++
Versión que procesa un solo enunciado NMEA por ejecución.	<pre> GPSData TrekkGPS::get() { char startMarker = '\$'; char endMarker = '\n'; char currentChar; if (state == READY) { resetSentence(); state = WAITING; } while (gps.available() > 0 && state != READY) { currentChar = gps.read(); if (state == READING) { if (currentChar != endMarker) { sentence[index] = currentChar; index++; if (index >= nmea_length) { index = nmea_length - 1; } } else { </pre>

```

sentence[index] = '\0';
state = READY;
}
} else if (currentChar == startMarker) {
state = READING;
index = 0;
}
}
return GPGGA();
}

void TrekkGPS::readFromGPS() {
char startMarker = '$';
char endMarker = '\n';
char currentChar;

if (state == READY) {
resetSentences();
state = WAITING;
}

while (gps.available() > 0 && state != READY) {
currentChar = gps.read();

if (currentChar == startMarker && state !=
READING) {
state = READING;
sdx = 0; // El índice del enunciado
cdx = 0; // El índice del carácter
} else if (state == READING) {
if (currentChar == endMarker) {
sentences[sdx][cdx] = '\0';
state = READY;
} else if (currentChar == startMarker) {
sentences[sdx][cdx] = '\0';
sdx++;
cdx = 0;

if (sdx >= no_of_sentences)
sdx = no_of_sentences - 1;
} else {
sentences[sdx][cdx] = currentChar;
cdx++;

if (cdx >= sentence_buffer)
cdx = sentence_buffer - 1;
}
}
}
}
}

```

Versión que procesa varios enunciados NMEA por ejecución.

- b) *ruta*: Representa una ruta predeterminada que puede ser creada por un usuario, requiriendo únicamente un nombre para su identificación.
- c) *pts_ruta*: Representa un punto geográfico, definido por su longitud (lo) y latitud (la), que pertenece a una ruta específica.
- d) *dispositivo*: Almacena la información de cada dispositivo, permitiendo asociar los datos de posición enviados con el respectivo usuario y su ruta predeterminada.
- e) *gps_datum*: Representa un punto geográfico capturado por un dispositivo durante una excursión en una ruta determinada. Incluye la longitud (lo), latitud (la), tiempo de captura (ts) y el tipo de enunciado NMEA del que proviene la información (nt).

En la Fig. 1 se muestra un diagrama de las 5 tablas anteriores y las relaciones entre ellas.

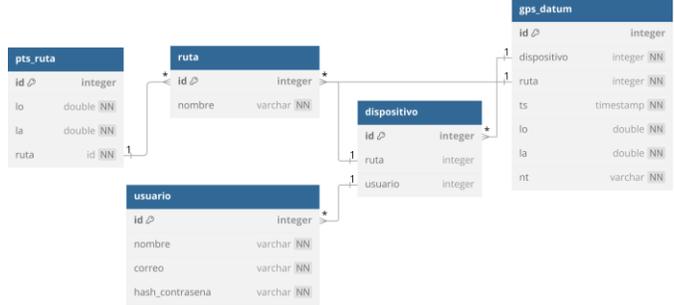


Fig. 1 Estructura y relación de las tablas en la base de datos.

La etiqueta "NN" en las propiedades de las tablas de la Fig. 1 indica que la propiedad correspondiente está sujeta a la restricción NOT NULL, lo que significa que dicha propiedad es obligatoria en cada registro y no puede contener valores nulos.

Este diseño de base de datos facilita la organización de la información de manera que se pueda asociar con el usuario correspondiente y se agrupe eficientemente según la ruta y/o el dispositivo, permitiendo un acceso más estructurado y rápido a los datos.

En un paso posterior, se desarrolló el código para una API que intercepta las peticiones enviadas por el microcontrolador y gestiona las interacciones con la base de datos. A continuación, se describen los puntos de acceso (endpoints) de la API en la Tabla 2.

TABLA 2
PUNTOS DE ACCESO DE LA API DEL PROYECTO FINAL DEL ALUMNO

Ruta URL	Método HTTP	Parámetros	Descripción
"/ruta"	GET	El identificador de ruta "id".	Retorna la información de la ruta en función del parámetro "id" provisto; o retorna un

“/gpsdatum”	POST	La longitud, latitud y su fecha de captura. Tipo de enunciado NMEA del cual se originan los datos, identificador del dispositivo e identificador de la ruta que se está Llevando en la excursión.	error HTTP 400. Inserta la información provista como un registro nuevo en la tabla correspondiente “gps_datum”; o retorna un error HTTP 500.
“/device”	GET	El identificador “idr” de un determinado dispositivo.	Retorna todos los puntos de las rutas predefinidas por el usuario, almacenadas en la tabla “pts_ruta”, asociados con el identificador de dispositivo pasado como parámetro.

Esta API permite a un dispositivo cargar los datos de la ruta predefinida asociada a su identificador y publicar los datos de posicionamiento recopilados por su módulo GPS. Cabe destacar que el cliente web desarrollado no utiliza esta API, ya que implementa sus propias consultas SQL para obtener y desplegar la información.

El desarrollo del cliente web fue llevado a cabo sin contratiempos, permitiendo que el usuario, tras iniciar sesión, accediera a la información relacionada con sus dispositivos, rutas y datos de posicionamiento.

IV. METODOLOGÍA

Aplicación del Aprendizaje Basado en Problemas (ABP)

Los retos planeados en el curso siguieron los principios fundamentales del ABP, se presentaron escenarios donde podían aplicar el internet de las cosas sin soluciones establecidas, fomentando en los estudiantes que definieran una situación a resolver, planearan diferentes soluciones, hicieran pruebas y realizaran ajustes. El rol del profesor en el caso del reto se limitó a ser facilitador, orientar y proporcionar retroalimentación cuando fue necesario. El proceso de evaluación consistió en la evaluación del proyecto, la presentación de éste y el resultado funcional del prototipo.

Propósito

Este proyecto corresponde al reto aplicado a los estudiantes del curso TC1004B Implementación de Internet de las Cosas en el Tecnológico de Monterrey durante el semestre agosto diciembre 2024, en el Grupo 503. Este curso forma parte de la currícula de las tres carreras del área de

Computación y son alumnos de ITC (Ingeniero en Tecnologías Computacionales), IRS (Ingeniero en Robótica y Sistemas Digitales) e ITD (Ingeniero en Transformación Digital de Negocios).

Contenido

Este reto fue desarrollado como parte del curso, el cual abarcó los temas establecidos en el temario de Circuitos Digitales Básicos y Avanzados, Bases de Datos, Administración de Proyectos, Sistemas Operativos, Interfaz de Usuario e Internet de las Cosas. El objetivo principal del reto es que el estudiante diseñe una aplicación real que integre los elementos clave del curso, tales como sensores, actuadores, controladores, transmisión de información mediante tecnologías como WiFi u otras inalámbricas y bases de datos. Este desafío promueve una integración multidisciplinaria dentro del ámbito de la computación, proporcionando un aprendizaje profundo al combinar diversos componentes tecnológicos en un solo proyecto. Además, enfrenta a los estudiantes a las dificultades inherentes a la estructuración de un proyecto tecnológico de alto nivel, lo que les permite adquirir un conocimiento integral y desarrollar habilidades cruciales para su futuro profesional.

Enfoque

El enfoque en este contexto se centra en la integración práctica de los diversos conocimientos adquiridos a lo largo del curso, aplicados al diseño y desarrollo de una aplicación real. La metodología seguida se basa en el Aprendizaje Basado en Problemas (ABP), donde los estudiantes se enfrentan a retos que requieren la resolución de problemas imprevistos, promoviendo una comprensión más profunda y el desarrollo de habilidades tanto técnicas como no técnicas. Este enfoque metodológico también promueve la experimentación y el pensamiento crítico, ya que los estudiantes no solo se limitan a aplicar lo aprendido en el temario, sino que deben buscar soluciones innovadoras a problemas técnicos que no siempre tienen respuestas predefinidas

V. RESULTADOS

Se recopilaron 85 enunciados NMEA durante la prueba de campo. Un proceso de depuración automática descartó 28 registros (sellos de tiempo truncados, coordenadas fuera de los rangos físicos de la Tierra o enunciados no reconocidos) y conservó 57 lecturas válidas (67 %) para el análisis. Todas las coordenadas se compararon con un punto de referencia: el centro del campus Monterrey del Tecnológico de Monterrey (25.6516 ° N, -100.2899 ° W). Se utilizó la fórmula del *haversine* para calcular la distancia sobre la superficie terrestre entre la posición de cada registro y el punto de referencia.

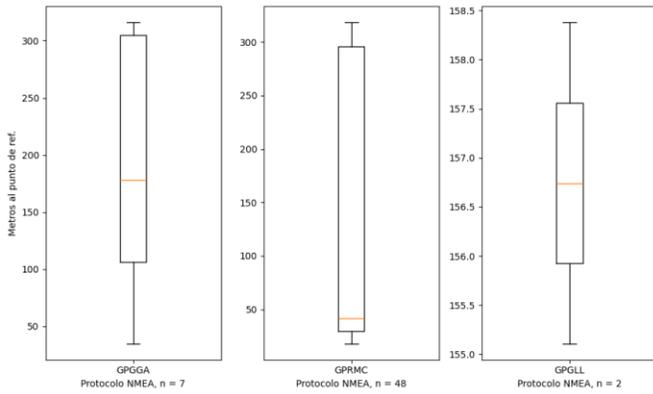


Fig. 2 Distancia al punto de referencia

El protocolo GPGGA agrupa 7 posiciones válidas. La mediana de la distancia al punto de referencia se ubica en torno a 180 metros y el rango entre el primer y el tercer cuartil se extiende unos 200 metros. Con un tamaño de muestra tan reducido, basta que dos valores se alejen considerablemente para ensanchar la caja y separar los cuartiles. Aún así, la amplitud refleja que estos enunciados suelen generarse en momentos de geometría deficiente (pérdida parcial de satélites o cambios rápidos de constelación), cuando el receptor ofrece resultados menos confiables.

El protocolo GPRMC reúne 48 posiciones válidas. La mediana se mantiene por debajo de los 50 metros y el rango intercuartílico alcanza aproximadamente 250 metros. Es decir, aunque el valor central está próximo al punto de referencia, la mitad de las lecturas se dispersa dentro de un corredor de un cuarto de kilómetro. Con una muestra amplia, esa amplitud denuncia lapsos intermitentes de señal débil—sombra urbana, vegetación densa o variaciones en la orientación de la antena—que desplazan temporalmente la posición antes de recuperar la precisión.

El protocolo GPGLL solo aporta 2 posiciones válidas, ambas situadas alrededor de 157 metros del punto de referencia. La muestra es insuficiente para estimar cuartiles, de modo que no se define una caja estadística. Estas dos lecturas, sin embargo, coinciden con las fases de mayor error observadas en los otros protocolos; se requerirá una recolección adicional para determinar si este formato puede estabilizarse o si mantiene un sesgo sistemático de varios centenares de metros.

A fin de detectar saltos espurios entre lecturas consecutivas se estimó la velocidad instantánea de cada posición válida. Para ello se ordenaron los 57 registros por su sello de tiempo y se calcularon: el intervalo temporal entre lecturas consecutivas ($\Delta t = t_i - t_{i-1}$), el desplazamiento lineal (Δs) como la diferencia entre las distancias *haversine*, y la velocidad instantánea ($v = \Delta s / \Delta t$).

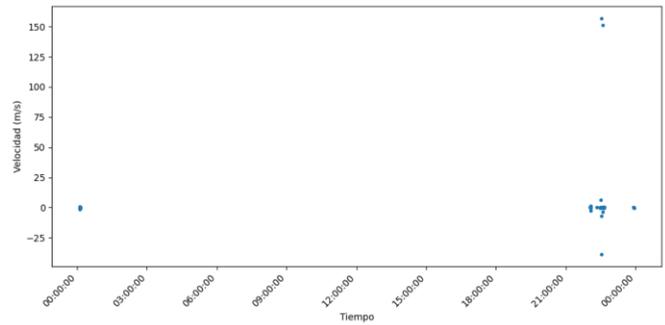


Fig. 3 Velocidad instantánea respecto al tiempo (m/s)

El 89 % de las velocidades (50 de 56) cae entre -1 m/s y $+1$ m/s. Estos valores corresponden a un recorrido muy lento o a un dispositivo en reposo con ligeros “temblores” numéricos propios del error del GPS. Seis valores superan ± 3 m/s; de ellos, dos son extremos ($+157$ m/s y $+151$ m/s) y uno alcanza -38.9 m/s. Ninguna de estas cifras es físicamente plausible para un peatón ni para un vehículo ligero en la zona de la prueba; se originan cuando la posición salta cientos de metros entre dos marcas de tiempo separadas apenas uno o dos segundos.

La Fig. 2 revela que los valores de distancia se agrupan en dos comportamientos muy claros: un núcleo cercano al punto de referencia (menores a 50 metros, dominado por los enunciados GPRMC), y episodios alejados que alcanzan 150 – 320 metros (presentes en GPGGA y, en menor medida, en GPRMC). Cuando estas mismas lecturas se observan en la Fig. 3, los puntos alejados se traducen en velocidades instantáneas físicamente imposibles (± 151 m/s, $+157$ m/s, -39 m/s). Por el contrario, el núcleo de posiciones cercanas produce velocidades comprendidas entre -1 m/s y $+1$ m/s, coherentes con el desplazamiento peatonal lento que efectuamos al realizar nuestras pruebas de campo.

La comparación entre los datos de la Fig. 2 y la Fig. 3 muestra que todo pico de velocidad coincide con una distancia anómalamente alta un par de segundos antes o después. Esto indica que los saltos podrían provenir de degradaciones momentáneas de la señal y no de un movimiento real.

VI. CONCLUSIÓN

Este trabajo confirma el valor del Aprendizaje Basado en Problemas (ABP) como un facilitador de la integración de conocimientos técnicos y del desarrollo de habilidades transversales, preparando a los estudiantes para enfrentar los desafíos profesionales mediante el autoaprendizaje, la autogestión y el pensamiento creativo.

El Aprendizaje Basado en Problemas (ABP) constituye una metodología pedagógica valiosa que enriquece la enseñanza de la ingeniería al permitir que los estudiantes

adquieran tanto conocimientos técnicos como no técnicos a través de la resolución de problemas imprevistos que surgen en el desarrollo de proyectos. Estos desafíos impulsan a los estudiantes a explorar más allá del currículo formal, adquiriendo el conocimiento necesario para implementar soluciones innovadoras.

A lo largo del desarrollo del proyecto final de un alumno en la materia TC1004B "Implementación de Internet de las Cosas" impartida en el Tecnológico de Monterrey, se evidencian diversos tipos de problemas imprevistos: desde fallos en un módulo GPS hasta limitaciones en la funcionalidad de las librerías de C++ o incluso errores en la implementación de algoritmos de procesamiento de enunciados NMEA. Los conocimientos requeridos para resolver estos imprevistos exceden los contenidos establecidos en el temario del curso, e incluyen habilidades como el manejo de caracteres ASCII en comunicación serial mediante UART, así como la capacidad para experimentar con diferentes enfoques para la resolución de problemas, tomar decisiones fundamentadas y gestionar proyectos dentro de un marco temporal.

Aunque los resultados del prototipo podrían no alcanzar los niveles de rendimiento y consistencia esperados, el verdadero valor del proyecto radica en el aprendizaje adquirido durante el proceso. La resolución de problemas en un entorno incierto fomenta el desarrollo de habilidades críticas para la ingeniería, como la adaptabilidad frente a limitaciones técnicas, la capacidad de evaluar soluciones bajo restricciones reales y la aplicación de conocimientos más allá del alcance del currículo formal. En este sentido, la experiencia vivida durante el proyecto no solo contribuye al desarrollo de habilidades técnicas, sino que también prepara a los estudiantes para enfrentar los desafíos futuros que surgirán en su carrera profesional en ingeniería y tecnología.

AGRADECIMIENTOS

Los autores desean agradecer el apoyo financiero y técnico de Writing Lab, Institute for the Future of Education, Tecnológico de Monterrey, México, en la producción de este trabajo.

REFERENCIAS

[1] L. R. De Camargo Ribeiro, «Electrical engineering students evaluate Problem-Based Learning (PBL)», *International Journal Of Electrical Engineering Education*, vol. 45, n.º 2, pp. 152-161, abr. 2008, doi: 10.7227/ijeee.45.2.7.

[2] M. C. Lozano-Ramírez, «El aprendizaje basado en problemas en estudiantes universitarios», *Tendencias Pedagógicas/Tendencias Pedagógicas*, vol. 37, pp. 90-103, dic. 2020, doi: 10.15366/tp2021.37.008.

[3] C. Steidley, R. Bachnak, R. S. Dannelly, R. Kulkarni, y C. Mayfield, «A multi-spectral imaging system for geo-spatial applications», *Journal Of Computational Methods In Sciences And Engineering*, vol. 5, n.º s1, pp. S93-S109, sep. 2005, doi: 10.3233/jcm-2005-5s108.

[4] D. R. P. Patnaikuni, «A comparative study of Arduino, Raspberry Pi and ESP8266 as IoT development board», *International Journal Of Advanced Research In Computer Science*, vol. 8, n.º 5, pp. 2350-2352, jun. 2017, doi: 10.26483/ijarcs.v8i5.3959.

[5] P. P. Ray, «Internet of things for smart agriculture: Technologies, practices and future direction», *Journal Of Ambient Intelligence And Smart Environments*, vol. 9, n.º 4, pp. 395-420, jun. 2017, doi: 10.3233/ais-170440.

[6] R. Franch Argullol, «Seguiment de vehicles via GPS: Desenvolupament d'un dispositiu localitzador de vehicles amb capacitats OBD», tesis de licenciatura, Universitat Politècnica de Catalunya, 2023. [En línea]. Disponible en: <http://hdl.handle.net/2117/396118>

[7] «Astro», *Astro*. <https://astro.build/>

[8] A. P. Garrido, Y. B. López, y G. G. Constante, «Rendimiento de MariaDB y PostgreSQL», *Revista Científica y Tecnológica UPSE*, vol. 7, n.º 2, pp. 09-16, dic. 2020, doi: 10.26423/rctu.v7i2.538.

[9] B. Stroustrup, «An overview of C++», *ACM SIGPLAN Notices*, vol. 21, n.º 10, pp. 7-18, jun. 1986, doi: 10.1145/323648.323736.

[10] M. Hart [mikahart], «TinyGPSPlus». 20 de mayo de 2024. [En línea]. Disponible en: <https://github.com/mikalhart/TinyGPSPlus.git>

[11] 107-Systems [107-systems], «107-Arduino-NMEA-Parser». 18 de marzo de 2024. [En línea]. Disponible en: <https://github.com/107-systems/107-Arduino-NMEA-Parser.git>

[12] D. Abbott, «Chapter 12 - Kernel Modules and Device Drivers», en *Linux for Embedded and Real-time Applications*, 3.ª ed., 2013, pp. 185-209. doi: 10.1016/b978-0-12-415996-9.00012-5.

[13] Microsoft, «Visual Studio Code». 16 de enero de 2026. [En línea]. Disponible en: <https://github.com/microsoft/vscode.git>

[14] PlatformIO Labs, «PlatformIO». 11 de enero de 2025. [En línea]. Disponible en: <https://platformio.org/>

[15] T. D. Lundgren, «The Case For ASCII», *ARMA Records Management Quarterly*, vol. 23, n.º 2, p. 48, abr. 1989, [En línea]. Disponible en: <https://www.proquest.com/scholarly-journals/case-ascii/docview/227751308/se-2>

[16] «Neon Serverless Postgres — Ship faster», *Neon*. <https://neon.tech/>

[17] J. Sequeda, S. Tirmizi, y D. Miranker, «SQL databases are a moving target», *Position Paper For W3C Workshop On RDF Access To Relational Databases*, Cambridge, MA, USA, 2007, [En línea]. Disponible en: <https://www.w3.org/2007/03/RdfRDB/papers/sequeda.pdf>