

# **SISTEMA DE POSICIONAMIENTO LOCAL BASADO EN UNA RED WI-FI Y UNA INTERFAZ GRÁFICA DESARROLLADA CON PYTHON PARA ANALIZAR EL DESEMPEÑO DE JUGADORES DE SOCCER**

***José Guadalupe Zavala Villalpando***

Tecnológico Nacional de México en Celaya

*jg.zavala@itcelaya.edu.mx*

***Víctor Manuel Sámano Ortega***

Tecnológico Nacional de México en Celaya

*M1703097@itcelaya.edu.mx*

***Nicolás Guerrero Chávez***

Tecnológico Nacional de México en Celaya

*M1703098@itcelaya.edu.mx*

## **Resumen**

Este artículo describe el desarrollo de un sistema de posicionamiento de jugadores de soccer en la cancha para analizar su ubicación, desplazamiento y velocidad en partidos y entrenamientos, con el fin de utilizar estos datos en la evaluación cuantitativa de su desempeño. El sistema está basado en la triangulación de la posición, mediante la medida de la intensidad de señal recibida (RSS) de las conexiones de una red Wi-Fi, dicha triangulación se realiza en un script de Python y los datos se despliegan en una interfaz gráfica desarrollada con Tkinter y Matplotlib.

**Palabras Clave:** Python, Sistema de posicionamiento local, Soccer, Wi-Fi.

## **Abstract**

*This article describes the development of a positioning system of soccer players on the field, in order of analysing their location, displacement and velocity in games*

*and training, with the final purpose of using this data for the quantitative evaluation of their performance. The system is based on the position triangulation, by the measurement of strong signal (RSSI) of Wi-Fi web connections, this triangulation works in a Python script and data is shown in a graphic interface developed using Tkinter and Matplotlib.*

**Keywords:** *Local position system, Python, Soccer, Wi-Fi.*

## **1. Introducción**

Evaluar el desempeño de un futbolista permite observar la efectividad del programa de entrenamiento realizado, evaluar los primeros signos de fatiga y sobre entrenamiento, identificar fortalezas y debilidades para poder enfocarlas a la consecución de un objetivo en particular, tomar decisiones sobre cambiar la forma que se trabaja con el atleta, etc. La evaluación de un atleta puede darse de forma cuantitativa o cualitativa, está última es la más utilizada en el fútbol soccer, mientras que, en otros deportes como el béisbol, basquetbol, o el fútbol americano, los datos estadísticos resultado de una evaluación cuantitativa están a la mano de prácticamente cualquier persona afín al deporte [1]. La evaluación cualitativa es subjetiva y no arroja resultados concretos respecto al desempeño de los atletas, por lo que es necesario desarrollar sistemas capaces de generar datos duros en etapas clave del desarrollo de los atletas como el entrenamiento y el juego.

Si bien el análisis estadístico de datos cuantitativos no es lo más común en el fútbol soccer, este si se utiliza valiéndose de sistemas que implican una infraestructura de cámaras y un grupo de personas especializadas que revisan cada momento del desarrollo de un juego, por ejemplo, el sistema AMISCO que utiliza 8 cámaras para grabar los juegos y después digitalizar la acción [2]. Sin embargo, sistemas como este requieren infraestructura y la presencia de expertos, lo que recae en elevados costos, por lo que no se los utiliza en la etapa de entrenamiento e inclusive no se los utiliza en todos los partidos.

El propósito del presente es proporcionar un sistema de posicionamiento de los jugadores en el campo en el transcurso del juego y entrenamiento, con el fin de

proporcionar datos de ubicación, desplazamiento y velocidad, posteriores al evento, que puedan utilizarse para el análisis del desempeño de los atletas.

Se logró implementar un sistema de posicionamiento basado en una red Wi-Fi que aproxima la posición de un jugador en el campo, con errores no significativos, y posteriormente almacena y despliega los datos obtenidos en una interfaz gráfica desarrollada en Python con Tkinter y Matplotlib, esto facilita su análisis y hace posible la evaluación cuantitativa del desempeño del atleta.

## 2. Métodos

Un sistema de posicionamiento local (SPL) también llamado sistema de posicionamiento en interiores se define como: “Una tecnología utilizada para identificar la posición de un elemento en interiores” [3]. Estos sistemas cuentan con una serie de terminales móviles (TM), de quienes puede obtenerse su posición; un determinado número de elementos que sensan parámetros asociados a la posición de las TM respecto a un sistema de referencia dado; un algoritmo que proporciona la posición en base a los parámetros obtenidos por los sensores y una interfaz que muestra al usuario tal posición [4]. En el sistema desarrollado se plantearon como TM servidores Wi-Fi (S) que se desplazan, llevados por los jugadores, en un área contenida un espacio definido por coordenadas (x,y), mientras que los sensores se plantearon como estaciones Wi-Fi (E) en torno al área (figura 1).

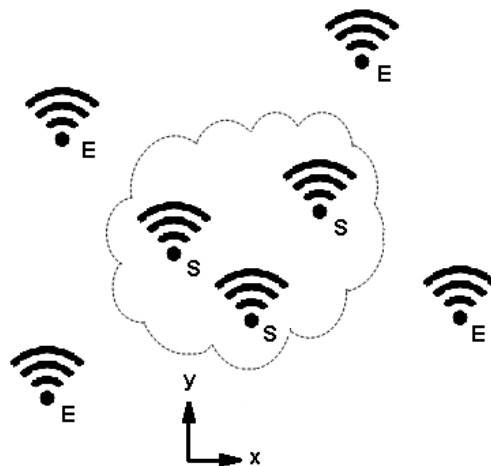


Figura 1 Esquema del sistema de posicionamiento planteado.

En el desarrollo del sistema se evaluó la posición de único un servidor en un área, de ancho  $d$ , que representa una cancha de fútbol a escala, sensado con dos estaciones ubicadas en esquinas del área y con el sistema de referencia fijo a una de ellas como muestra figura 2.

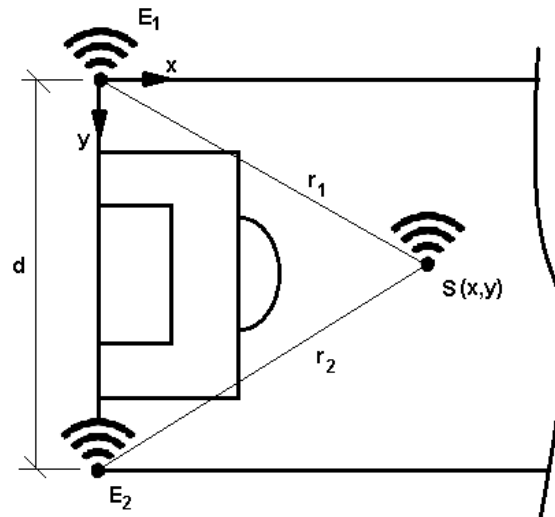


Figura 2 Elementos del sistema dispuestos en la cancha a escala.

Para determinar la posición de  $S$ ,  $(x,y)$ , se calcularon las distancias entre el servidor y las estaciones,  $r_1$  y  $r_2$ , en base a la medida de la potencia de la señal recibida (Received Signal Strength RSS) y después se aplicó un método de triangulación [5]. Para hacer esto posible el servidor, portado por el jugador, emite una señal Wi-Fi constantemente, mientras que las estaciones miden constantemente un parámetro que puede derivar en su posición relativa al servidor.

Para determinar  $r_1$  y  $r_2$  se utilizó el modelo de propagación (ecuación 1) que relaciona el indicador de la RSS (Received Signal Strength Indicator RSSI), dado en milivatios de decibeles (dBm), con la distancia entre la fuente y el receptor de la señal [6].

$$d = 10^{\left(\frac{P-RSSI}{10n}\right)} - c \quad (1)$$

En donde:  $d$  es la distancia en metros entre el receptor y el transmisor,  $P$  es la potencia entregada por el dispositivo fuente de la señal en dBm,  $n$  es denominada

la constante de propagación de la señal y al igual que  $c$  dependen de las condiciones del entorno de la red.  $P$  tiene un valor de +17dBm [7], mientras que experimentalmente se encontró para  $n$  un valor de 4 y para  $c$  un valor de 0.7.

Una vez encontrados  $r_1$  y  $r_2$  para triangular la posición de  $S$  se definió el ángulo  $\theta$  según ecuación 2.

$$\theta = \angle E_2 E_1 S = \cos^{-1} \left( \frac{d^2 + r_1^2 - r_2^2}{2dr_1} \right) \quad (2)$$

A partir del cual se definió la posición de  $S$  según la ecuación 3.

$$S(x, y) = (r_1 \sin \theta, r_1 \cos \theta) \quad (3)$$

Para cada par ordenado obtenido se tomó la hora del sistema, de modo que fuera posible determinar la velocidad del jugador, partiendo el desplazamiento, tanto en  $Y$  como en  $X$ , entre cada muestra y dividiendo tal cantidad entre el tiempo que pasa entre muestras.

Si  $P_i$  de coordenadas  $(x_i, y_i)$  es la posición del jugador en la  $i$ -ésima muestra tomada en la hora  $t_i$  y si además  $P_{i+1}$  de coordenadas  $(x_{i+1}, y_{i+1})$  es la posición de la muestra consecutiva tomada en la hora  $t_{i+1}$ ; entonces el desplazamiento del jugador en  $X$ ,  $d_x$ , está dado por la ecuación 4.

$$d_x = x_{i+1} - x_i \quad (4)$$

Mientras que el desplazamiento en  $Y$ ,  $d_y$ , está dado por la ecuación 5.

$$d_y = y_{i+1} - y_i \quad (5)$$

Entonces, el desplazamiento del jugador puede calcularse según la ecuación 6.

$$d = \sqrt{d_x^2 + d_y^2} \quad (6)$$

Y la velocidad se calcula según la ecuación 7.

$$v = \frac{d}{t_{i+1} - t_i} \quad (7)$$

El algoritmo de posicionamiento descrito se desarrolló en un script de Python por lo que los datos obtenidos de las estaciones se llevaron a una PC mediante conexiones seriales, mientras que el servidor es inalámbrico (figura 3).

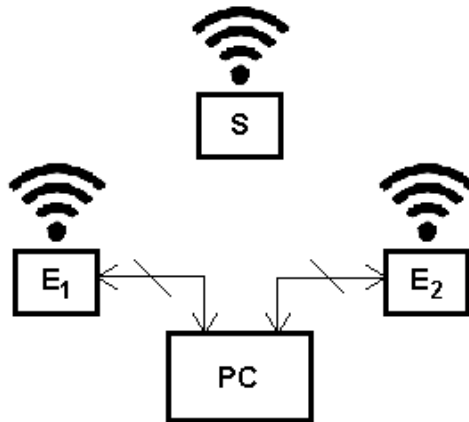


Figura 3 Arquitectura del sistema de posicionamiento.

Tanto las estaciones como el servidor se constituyeron teniendo como base un módulo Wi-Fi ESP8266 (ESP).

En el servidor el ESP se conectó a una batería de 3.3 V mediante un switch, S, que enciende y apaga el módulo, además de que ofrece la posibilidad de cargar la batería, cuando el módulo está apagado, a través del conector J (figura 4).

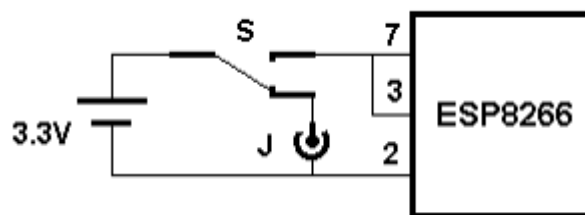


Figura 4 Diagrama de conexión del servidor.

Para que el ESP funcionara como servidor, se configuró usando comandos AT [8], mediante ellos se le asigna un modo de funcionamiento como AP (Access Point) y se le asigna el identificador SSID (Service Set Identifier), S1 para este proyecto. Después de introducir estos comandos, y una vez energizado, el ESP emite una red señal de red que se podrá identificar por otros dispositivos de red como S1.

En el caso de las estaciones el ESP se conectó a un Arduino UNO que hace de interfaz entre el módulo y la PC (figura 5).

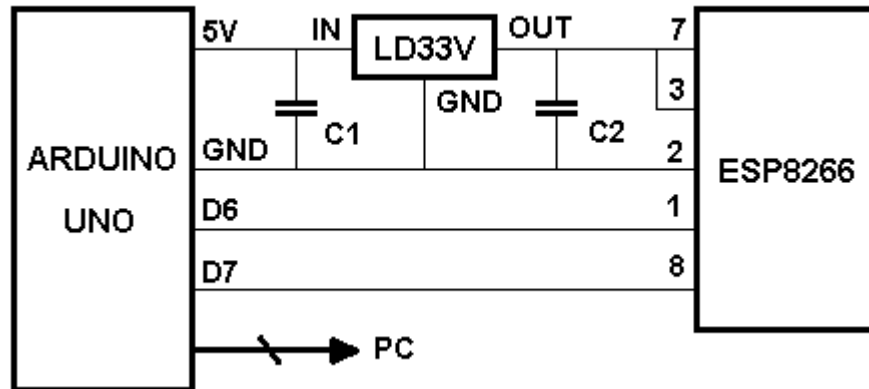


Figura 5 Diagrama de conexión de una estación.

En la estación el ESP obtiene el RSSI de la señal emitida por el servidor, S1, dato que envía al Arduino mediante un puerto serial emulado en los pines digitales 6 y 7 del controlador, para lograr establecer y validar ésta conexión fue necesario configurar ambos dispositivos; en el ESP se estableció su funcionamiento como estación y su protocolo de comunicación serial; los parámetros dados al protocolo fueron: una tasa de transferencia 9600 Baudios, 8 bits de datos, ninguno de paridad y un bit de parada (8N1), sin control de flujo.

Mientras que el Arduino se configuró haciendo uso de las librerías "WiFiEsp.h" [9] y "SoftwareSerial.h" [10].

En las estaciones, también fue necesario establecer comunicación entre el arduino y la PC, mediante el puerto serial, esto se consiguió con la inclusión de la librería "pySerial" en el script de Python [11] y el uso de los métodos de la clase Serial en Arduino.

La secuencia que sigue el script para establecer las comunicaciones y desarrollar el algoritmo de posicionamiento descrito se ilustra en figura 6.

En la primera parte de la secuencia del script, se establecen y validan las conexiones entre el ESP y el Arduino y entre el Arduino y Python, para cada una de las estaciones, esto se logró siguiendo los siguientes pasos:

- El servidor se energiza y genera la señal con el SDDI S1.
- Los scripts de los Arduinos, en las estaciones, entran en un ciclo en el que publican constantemente su identidad, 1 o 2, por el puerto serial.

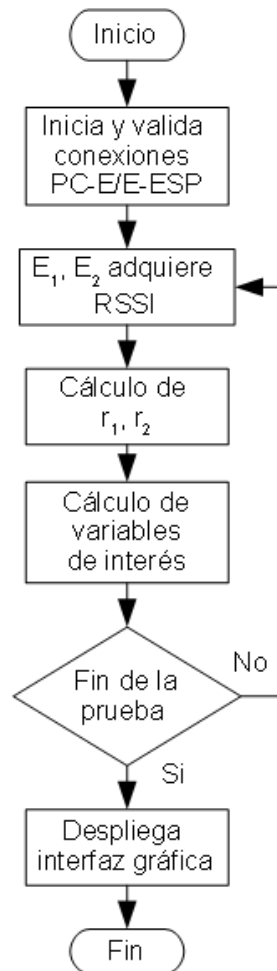


Figura 6 Diagrama de flujo del script de Python.

- El script de Python prueba a establecer conexión con los puertos COM0 a COM7, si la conexión en alguno de estos puertos es validada, lee el bus en busca de la identidad del emisor y asigna una clase a la conexión con un nombre que identifica a la estación conectada. Luego, envía un carácter para terminar el ciclo del Arduino. Una vez enviado el carácter, el Script de Python entra en un ciclo en espera de un carácter que enviará el Arduino una vez validada la conexión entre éste y el ESP.



- El Arduino termina el ciclo en el que publica su identidad y prueba a establecer conexión con el módulo, una vez que la conexión con el ESP se valida, envía un carácter al Script de Python y el Arduino entra en un nuevo ciclo en el que lee continuamente el RSSI de S1. Se envía un carácter al script de Python.
- El script de Python termina el ciclo de espera y continúa.

Una vez establecidas y validadas las conexiones el Arduino, de cada una de las estaciones, adquiere constantemente el RSSI de la señal y lo publica en el puerto serial. Por su parte el script en Python toma un número de muestras establecido y lleva a cabo el algoritmo de posicionamiento. Los datos se escalan, del valor dado para maqueta, a valores dados para la interfaz gráfica y a valores dados para una cancha real y son almacenados en una serie de arreglos, de una dimensión, para posteriormente pasarlos a la interfaz gráfica.

Finalmente, y una vez terminado el periodo de muestreo, los datos almacenados se despliegan en una interfaz desarrollada con las librerías “Matplotlib” [12] y “Tkinter” [13].

En la interfaz, se muestra un modelo de una cancha de fútbol en la que se representa la posición del jugador (figura 7), obtenida por el sistema, como una serie de puntos, esta función se ejecuta mediante un botón dispuesto al lado derecho de la cancha con la etiqueta “área jugada”, además de ese botón la interfaz cuenta con otros tres etiquetados como “Gráfica X”, “Gráfica Y” y “Gráfica V” (figura 8), la funcionalidad de estos botones es desplegar un gráfico dinámico que simula, en un tiempo menor al del periodo de muestreo, la posición del jugador en el eje X, su posición en el eje Y y su velocidad respectivamente, estos datos están dados en metros para las posiciones y en metros por segundo para la velocidad, y su marco referencial es el mismo que el expresado en la figura 2.

Con ésta herramienta del sistema, el interesado en los datos puede observar de manera clara la acción llevada a cabo por el jugador y la puede cuantificar, de manera que sirva como retroalimentación para buscar estrategias que lo mejoren.

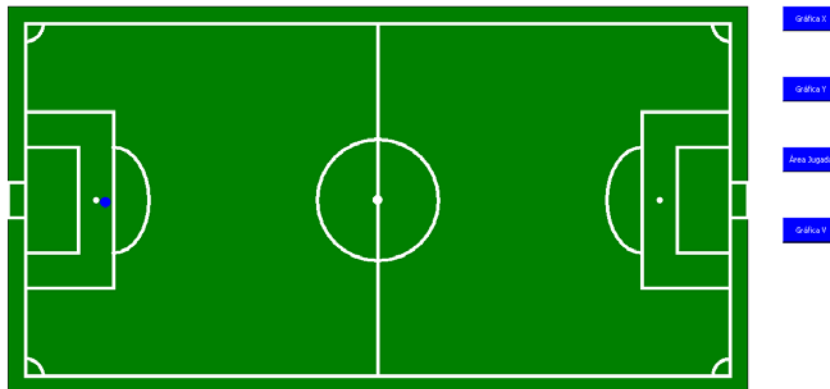


Figura 7 Interfaz gráfica del sistema.



Figura 8 Detalle de los botones de la interfaz gráfica.

### 3. Resultados

Para probar el sistema se desarrolló un experimento que consistió en desplazar el servidor sobre una trayectoria, en un área de 2 metros de ancho, desde y hasta los puntos que, en una cancha real de 90 x 120 m, corresponden a (24,18) y (96,72). Durante este desplazamiento se tomó una muestra cada segundo y el movimiento del servidor se ajustó de tal forma que el recorrido se cumpliera en 100 segundos; obteniendo entonces 100 muestras.

En la figura 9 se muestra el resultado de la interfaz gráfica para la posición del jugador. Además de los puntos generados por el algoritmo de posicionamiento, se incorporó una línea en color azul, que marca la trayectoria del servidor para tener una referencia y permitir un primer análisis visual de la exactitud del sistema.

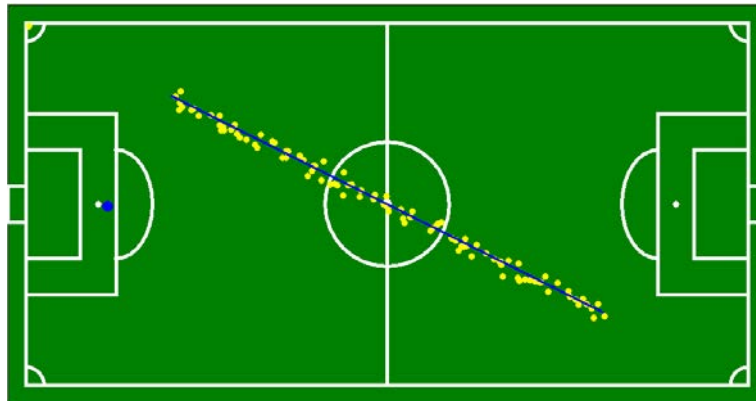


Figura 9 Resultado del sistema para el área jugada.

También se obtuvo el gráfico que ilustra la posición del jugador en el eje X, dada en metros, respecto a la hora en que se tomó la muestra; éste se ilustra en la figura 10.

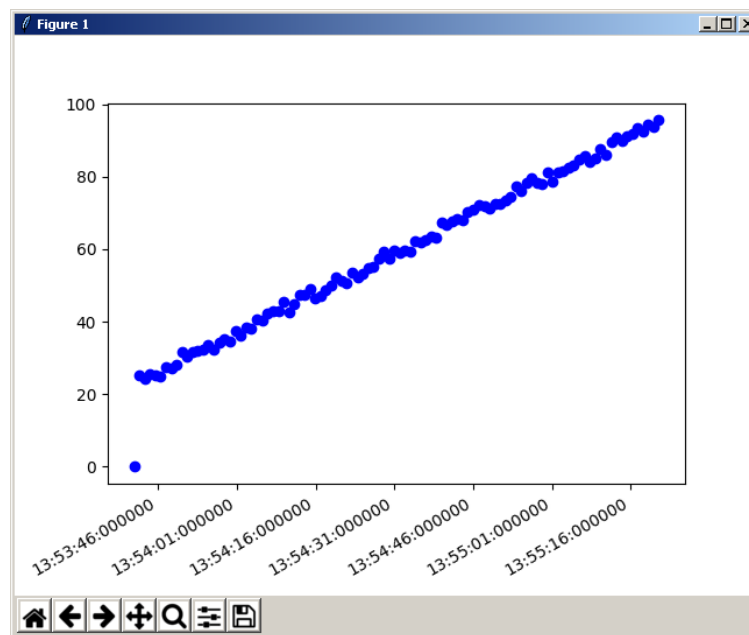


Figura 10 Desplazamiento el en eje X respecto a la hora en que se toma la muestra.

Para poner este resultado en perspectiva, los datos del posicionamiento se compararon con los valores reales por los que pasa el servidor. En la figura 11 se muestra una porción de la gráfica que ilustra esta comparación para la posición en eje X.

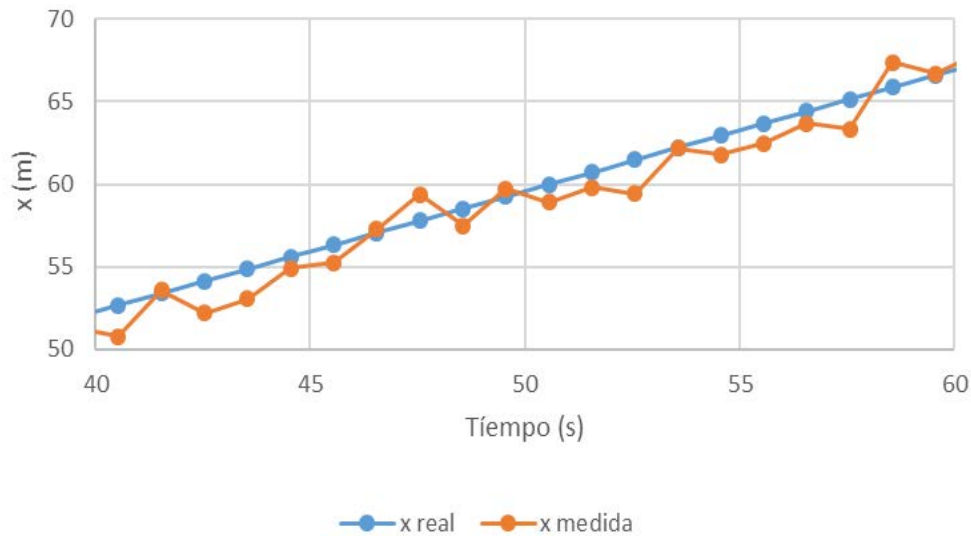


Figura 11 Relación entre X real y medida.

Mientras que los datos estadísticos del conjunto de muestras pueden observarse en la tabla 1.

Tabla 1 Datos estadísticos de la posición en X.

Error absoluto promedio (m)	Error absoluto máximo (m)	Error porcentual promedio (%)	Error porcentual máximo (%)
0.9587	2.5585	1.8362	7.2485

Otro de los gráficos arrojado por la interfaz es el que relaciona la posición del jugador en el eje Y, dada en metros, con la hora en que se tomó la muestra, este gráfico para el experimento realizado se observa en la figura 12.

Mientras que su comparación con los datos reales puede visualizarse en la figura 13. Los datos estadísticos de las muestras de posición en Y se resumen en la tabla 2. Finalmente, se obtuvo el gráfico que relaciona la velocidad del jugador y la

hora en que se tomó la muestra; este dato está expresado en metros por segundo y puede verse en la figura 14. La comparativa con los datos reales de la velocidad puede verse en la figura 15. Mientras que el resumen de sus datos estadísticos aparece en la tabla 3.

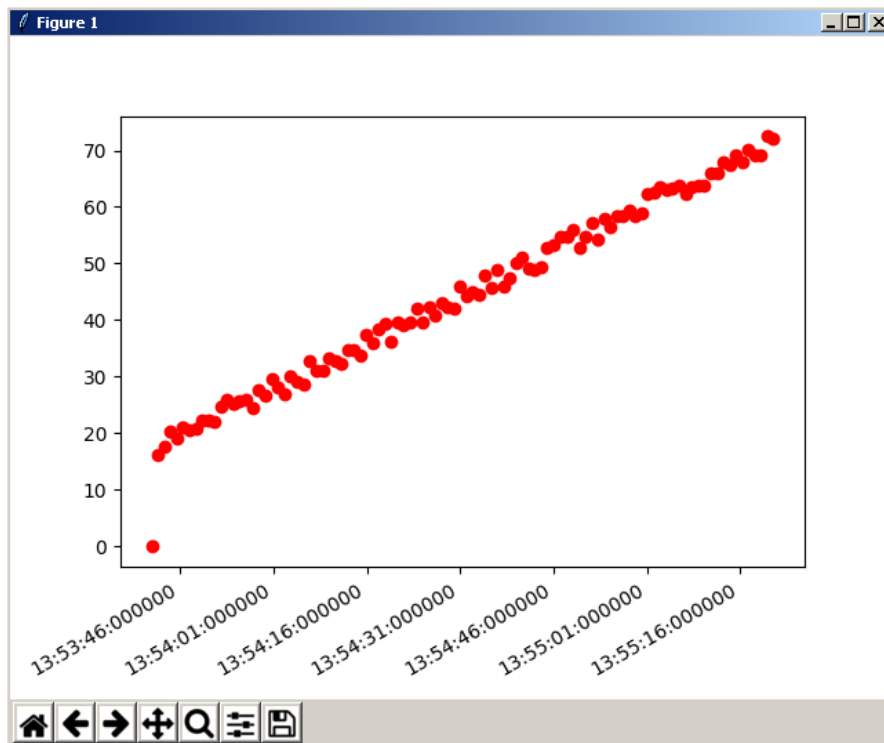


Figura 12 Desplazamiento el en eje Y respecto a la hora en que se toma la muestra.

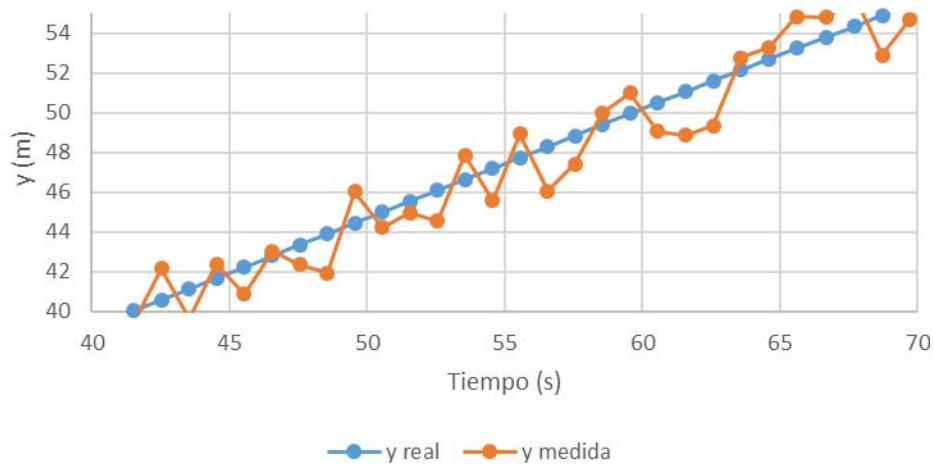


Figura 13 Comparación entre Y real y medida.

Tabla 2 Datos estadísticos de la posición en Y.

Error absoluto promedio (m)	Error absoluto máximo (m)	Error porcentual promedio (%)	Error porcentual máximo (%)
1.0571	2.5556	2.6466	9.5958

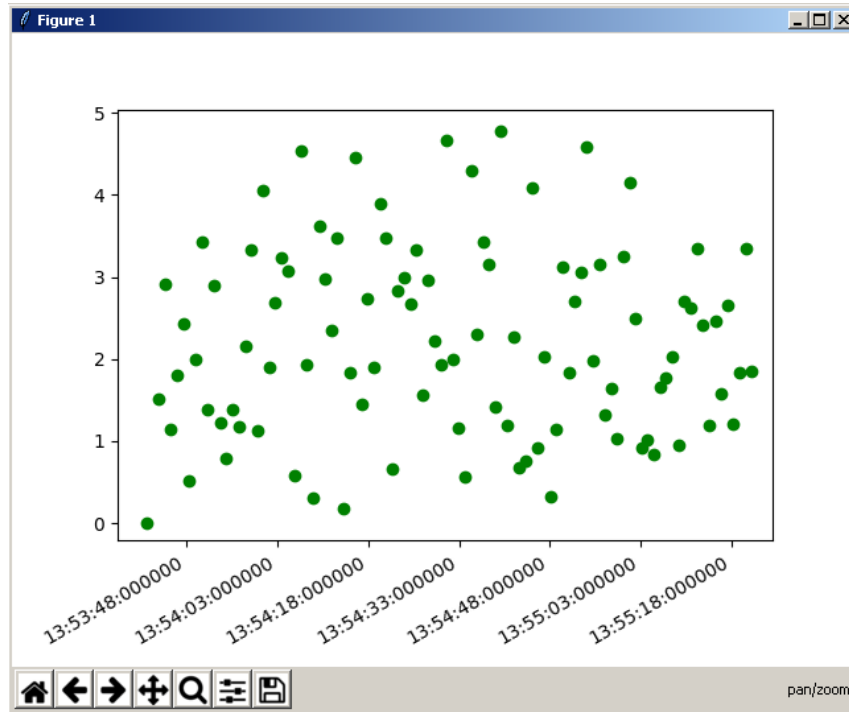


Figura 14 Velocidad respecto a la hora en que se toma la muestra.

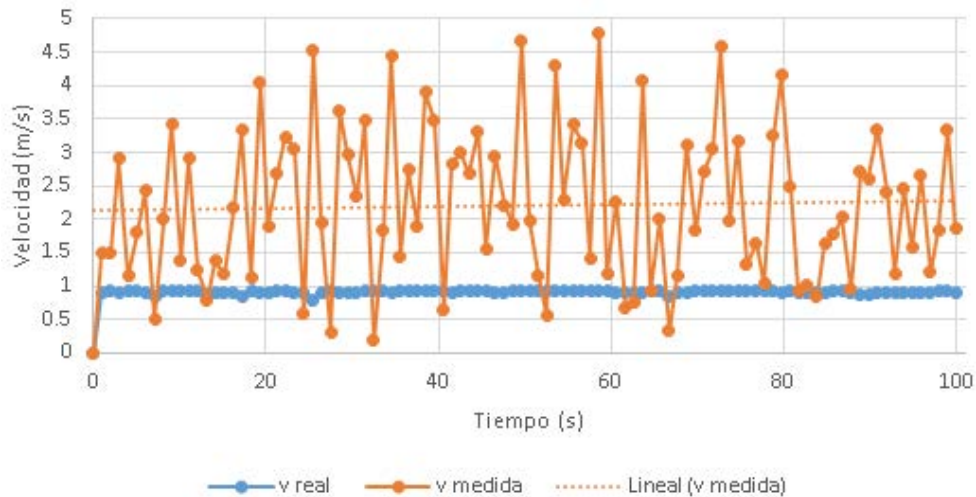


Figura 15 Relación entre velocidad real y medida.

Tabla 3 Datos estadísticos de la velocidad.

Error absoluto promedio (m/s)	Error absoluto máximo (m/s)	Error porcentual promedio (%)	Error porcentual máximo (%)	Promedio de las muestras (m/s)	Desviación estándar
1.3852	3.8652	152.4407	472.3567	2.2264	1.1341

#### 4. Discusión

El sistema genera una buena aproximación a la posición como se puede observar en los datos correspondientes; si bien para x se obtuvo un error de hasta 2.5585 m el promedio de todos los errores es menor a un metro y porcentualmente el error promedio es de apenas 1.8362% por lo que se puede decir que en promedio la posición en x es acertada hasta en un 98.1638%.

Mientras que, el error de la posición en y resultó ser mayor; teniendo un acierto de sólo 97.3534%.

El error de las muestras se tomó en relación a la posición del jugador en una cancha real, sin embargo, el experimento se realizó en una cancha escala, de modo que, al escalar las posiciones se escalan también los errores absolutos, por lo que, una vez implementado el sistema en una cancha real, se espera que los errores tiendan a reducirse.

Por otro lado, el error porcentual no se ve afectado por el escalamiento, sobre una base de confiabilidad aceptable, mayor a 95%, los resultados son muy acertados, sin embargo, una variación pequeña sobre los datos reales de posición, repercute en una variación mayor en la velocidad

Como se observó en la sección anterior, para datos confiables de posición se generó un error porcentual en la velocidad de hasta 472.3567%, además, acorde al error porcentual promedio esta medida no es nada confiable y fue una serie de datos muy dispersa; con una variación estándar de 1.1341.

La falta de fiabilidad en los datos de la velocidad los vuelve inutilizables, para lograr proporcionar este dato en desarrollos posteriores, se deberá implementar un filtro que suavice los errores en la posición, esto con el fin de buscar mayor exactitud en la medición de la velocidad.

En general, el sistema es capaz de proporcionar la posición de un jugador en la cancha y entregar tales datos de una forma fácil de estudiar mediante su interfaz gráfica, pero no es capaz de proporcionar la velocidad.

## 5. Bibliografía y Referencias

- [1] YOUNGCRACKS(2013). El Análisis Estadístico como Método de Valoración del Rendimiento del Futbolista. <http://www.futbolbaseenestadopuro.com/el-analisis-estadistico-y-cualitativo-como-metodo-de-valoracion-del-rendimiento-del-futbolista/>.
- [2] <http://entrenamientoactual.blogspot.mx/2013/08/amisco.html>. Entrenamiento Actual (2013). AMISCO.
- [3] Gartner IT: <http://www.gartner.com/it-glossary/?s=local+position+system>
- [4] Pahlavan K. Li X. and Makela J. (2002). Indoor geolocation science and technology. *IEEE Communications Magazine*. vol. 40. no. 2. pp. 112– 118.
- [5] Lim H. Kung L. Hou J. and Luo H. (2010). Zero-Configuration Indoor Localization over IEEE 802.11 Wireless Infrastructure. *Journal Wireless Networks*. Volume 16. Issue 2. Pages 405-420.
- [6] Conti M. (2017). Real Time Localización Using Bluetooth Low Energy. *Bioinformatics and Biomedical Engineering: 5th International Work-Conference, IWBBIO 2017, Granada, España, Abril 26–28, 2017, Proceedings, Parte 2*.
- [7] Espressif IOT Team. (2017). ESP8266EX Datasheet Version 5.4.
- [8] Espressif IOT Team. (2017). ESP8266 AT Instruction Set Version 2.1.0.
- [9] GitHub Repository: <https://github.com/bportaluri/WiFiEsp>
- [10] ARDUINO Reference: <https://www.arduino.cc/en/Reference/SoftwareSerial>
- [11] Liechti C. (2015). pySerial 3.0 documentation: <https://pythonhosted.org/pyserial/>.
- [12] Hunter D, Dale D, Firing E, Droettboom M and the Matplotlib development team: [https://matplotlib.org/api/pyplot\\_api.html](https://matplotlib.org/api/pyplot_api.html).
- [13] Python Software Foundation. 24.1. Tkinter — Python interface to Tcl/Tk: <https://docs.python.org/2/library/tkinter.html>.