



**COMILLAS**

UNIVERSIDAD PONTIFICIA

ICAI

# GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

## EVALUACIÓN DE TÉCNICAS PARA LA REDUCCIÓN DE ERROR EN LA ESTIMACIÓN DEL POSICIONAMIENTO EN INTERIORES CON BLE

Autor: Chávez Macías, Alejandro

Director: Contreras Bárcena, David

Madrid



Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título  
EVALUACIÓN DE TÉCNICAS PARA LA REDUCCIÓN DE ERROR EN LA  
ESTIMACIÓN DEL POSICIONAMIENTO EN INTERIORES CON BLE

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2020/21 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.



Fdo.: Chávez Macías, Alejandro

Fecha: 11/ 07/ 2021

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Contreras Bárcena, David

Fecha: 11/ 07/ 2021





**COMILLAS**

UNIVERSIDAD PONTIFICIA

ICAI

# GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

## EVALUACIÓN DE TÉCNICAS PARA LA REDUCCIÓN DE ERROR EN LA ESTIMACIÓN DEL POSICIONAMIENTO EN INTERIORES CON BLE

Autor: Chávez Macías, Alejandro

Director: Contreras Bárcena, David

Madrid



# **Agradecimientos**

A mi familia y amigos que me han acompañado durante estos años de universidad.



# **EVALUACIÓN DE TÉCNICAS PARA LA REDUCCIÓN DEL ERROR EN LA ESTIMACIÓN DEL POSICIONAMIENTO EN INTERIORES CON BLE**

**Autor: Chávez Macías, Alejandro.**

Director: Contreras Bárcena, David.

Entidad Colaboradora: ICAI - Universidad Pontificia Comillas

## **RESUMEN DEL PROYECTO**

En este proyecto, se analiza el funcionamiento de distintos algoritmos de predicción de distancia de balizas BLE con el fin de obtener aquel más eficiente para ser implementado en una aplicación móvil sobre Flutter. Se descubre y se demuestra que, a pesar de que numerosos papers recomienden métodos complejos, una simple regresión lineal puede llegar a ser suficiente en el caso de uso estudiado.

**Palabras clave:** Bluetooth, Mobile, Indoor, BLE, RRNN, Algoritmos predictivos

### **1. Introducción**

En un mundo cada vez más conectado por las redes de telecomunicaciones, se ha venido experimentando en los últimos años una alta evolución en la tecnología IoT (siglas de Internet of Things). El contexto pandémico actual hace evidente la necesidad de implementar dichas soluciones IoT a casos de uso en ámbitos sociales tales como las visitas guiadas, donde el contacto directo es necesario y a la vez puede suponer un riesgo para la salud. La implementación hoy en día sería muy sencilla, y es que tenemos la ventaja de que actualmente cualquier persona posee un Smartphone, capaz de procesar información y con el que el usuario tiene la posibilidad de interactuar. Esto, sumado a que cualquier Smartphone actual tiene implementada tecnología BLE, hace de este el dispositivo ideal para ser implementado en dichos casos de uso.

### **2. Definición del Proyecto**

El proyecto consistirá en el análisis de rendimiento de la tecnología BLE implementada en móviles haciendo uso de la plataforma Flutter, de forma que dicha aplicación pueda ser implementada tanto en teléfonos iOS como Android. Dicho análisis servirá para implementar un caso de uso como es el de un circuito con balizas, las cuales podrán conectarse al dispositivo en el cual se encuentre instalada la aplicación desarrollada en

Flutter. La aplicación procesará los datos recibidos por la baliza, en función de la distancia del usuario, y la cercanía o lejanía con respecto a unas u otras balizas.

Para que una aplicación dependiente de dispositivos externos sea funcional y eficiente, será necesario realizar un análisis de rendimiento para conocer cómo puede afectar a este la distancia entre balizas, la presencia de perturbaciones en el ambiente, o, la batería de los dispositivos, ya que la aplicación deberá ser lo más eficiente posible.

Por lo tanto, el fin del proyecto es realizar un estudio de diferentes algoritmos de predicción, aplicados al análisis de la señal BLE. A partir del valor de potencia de la señal, RSSI, recibida por el dispositivo, estos algoritmos deberán predecir la distancia a la que se encuentra el usuario de la baliza. Para el caso de uso final que se propone, el objetivo deberá ser, recibida la potencia de las señales en el medio, discriminar entre balizas más cercanas al usuario.

Por ello, se realizarán experimentos de medida de la señal y se compararán los resultados obtenidos con los algoritmos.

### **3. Descripción de los experimentos**

#### **Software utilizado:**

Para realizar el experimento, se ha desarrollado una aplicación móvil haciendo uso de la plataforma Flutter y de una librería externa que añade funcionalidades BLE. Dicha aplicación nos ha permitido realizar los escaneos necesarios para obtener los datos con los que realizar el experimento.

Para implementar los algoritmos, se hace uso de la herramienta RStudio, la cual nos facilita el análisis de la señal.

#### **Hardware:**

El hardware utilizado ha sido balizas CC2650, las cuales emiten la señal BLE, y un teléfono móvil el cual recibirá la señal de la baliza mediante la aplicación de Flutter.

#### **Experimentos:**

- **Escenario:** El escenario en el que se han realizado los experimentos es una habitación de 12 metros de largo. Colocándose la baliza en un extremo de la habitación se han realizado mediciones durante un minuto a distancias de 3, 5,

7, 9 y 11 metros. Se han tomado dos situaciones de medición, un entorno sin ruido y un entorno con ruido.

- **Entorno con ruido:** Para un entorno ruidoso se ha supuesto una situación habitual para nuestro caso de uso. El usuario tendrá el dispositivo móvil en la mano e irá realizando escaneos (la aplicación los realizaría en segundo plano) mientras otros usuarios pasarán alrededor suya.
- **Entorno sin ruido:** Para conseguir un entorno con el menor ruido posible, se han tomado las mediciones en la misma habitación, sin personas ni obstáculos de por medio, y colocando el dispositivo móvil en una mesa, a la misma altura que la baliza.

Tras tomar las mediciones necesarias, se procede al análisis mediante distintos algoritmos de filtrado y predicción de la señal, con el fin de discernir entre los algoritmos más capaces de realizar nuestro objetivo: la correcta predicción de distancia y la discriminación entre cercanía de balizas.

#### 4. Estado del arte

Numerosos autores han realizado estudios previos que relacionan el indicador de potencia de la señal BLE con la distancia, en la búsqueda de encontrar algoritmos capaces de explicar una relación entre ambas variables.

Para ello, estos autores hacen uso de algoritmos tradicionales de clasificación como regresiones lineales, estadísticos más o menos sofisticados o de algoritmos más complejos y utilizados hoy en día como redes neuronales.

Los investigadores, para realizar sus análisis de la señal, tienen en cuenta la alta variabilidad del indicador RSSI, y de que tomar una medida puntual no será representativo de la potencia de la señal en un punto. Es por ello por lo que los autores toman un set de varias mediciones por distancia para realizar análisis más acordes con la realidad.

Para el caso de utilización de estadísticos, por ejemplo, la mayoría de los autores hacen uso de la media, como indicador de la potencia en un punto del espacio, a la hora de implementar variables RSSI en el modelo. Si bien esta simplificación, aunque acelera el proceso de resolución de la distancia, supone una pérdida de información.

## **5. Técnicas de filtrado**

Para nuestro propósito de análisis y predicción de la señal, se emplean los siguientes algoritmos de filtrado y predicción:

### **5.1. Algoritmos de regresión lineal**

Se aplicará un modelo de regresión lineal a las medidas obtenidas analizando el coeficiente de regresión para medir la correlación entre el RSSI y la distancia y las variables dependientes e independientes de todos los experimentos realizados.

### **5.2. Inferencia bayesiana**

Para el algoritmo de inferencia bayesiana:

- Se realiza un histograma de densidad de cada distribución de datos RSSI por distancia, de forma que tengamos.
- Teniendo en cuenta que dichas distribuciones coinciden entre sí en ciertos valores de potencia con una cierta probabilidad, se calculará a partir de dichas probabilidades la distancia a la que se estima que estará al usuario. Si nos encontramos en un punto cuya probabilidad de que sea de la distribución de 5 metros es del 0.7 y de que sea de la de 7 metros, de 0.3, se hace una estimación de la distancia por Bayes.
- Para crear el modelo bayesiano, se realiza el procedimiento anterior para cada valor RSSI dentro del rango que ocupan las distribuciones introducidas.

### **5.3. RRNN**

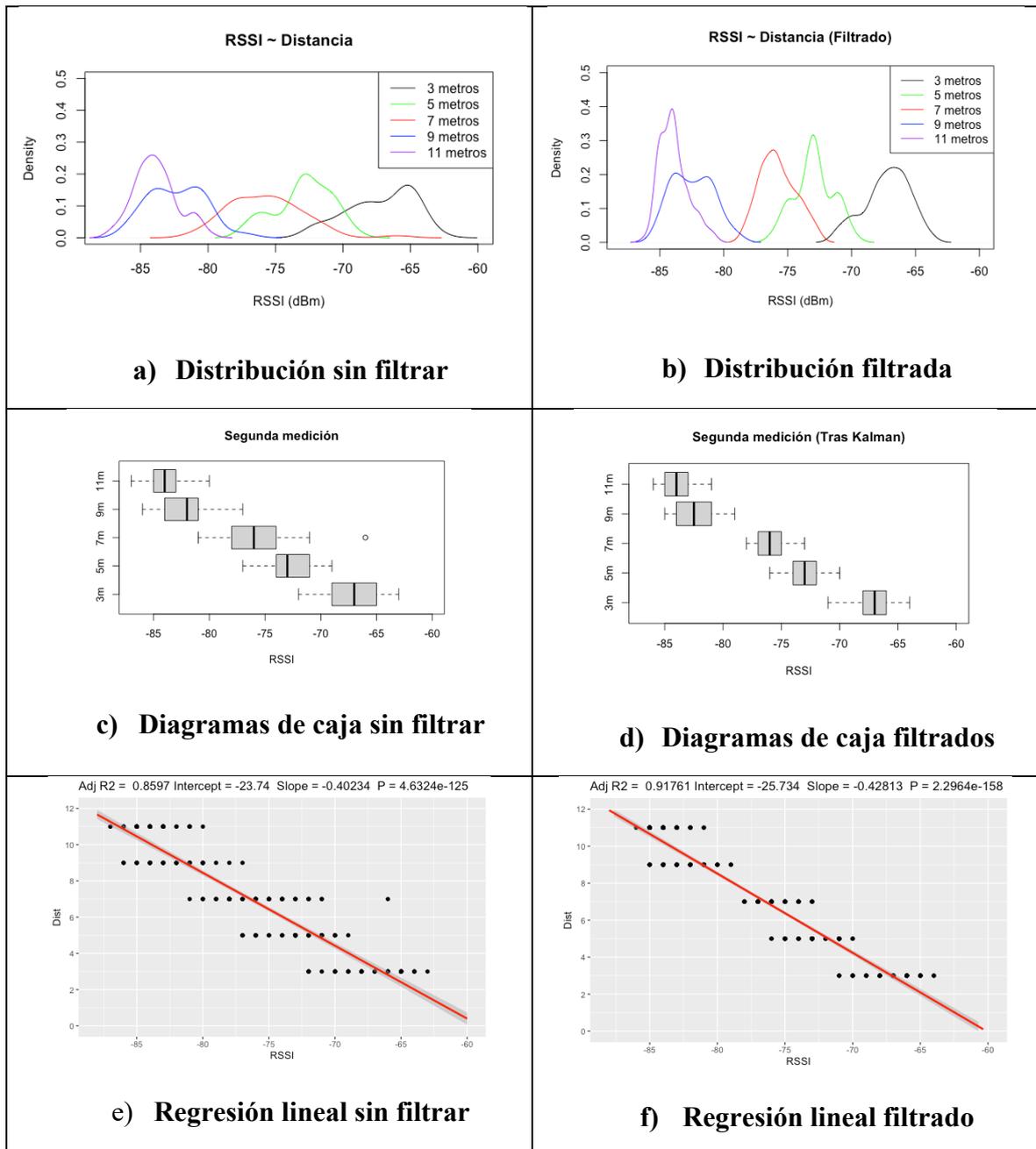
Se clasificarán los datos mediante un algoritmo de propagación trasera con redes neuronales, con el fin de predecir la distancia a la que pertenece un determinado RSSI. Se entrenará una red neuronal por cada distribución que sea capaz de determinar si un valor RSSI pertenece o no a la distribución de datos de distancia asociada.

### **5.4. Filtro de Kalman**

Se pasarán los datos iniciales por un filtro de Kalman, el cual reducirá el ruido en la señal, y se volverán a pasar dichos datos ya tratados por el filtro por los algoritmos anteriores, con el fin de comprobar el efecto del filtrado en la señal.

## 6. Resultados

Tras aplicar distintos algoritmos a los datos de medida, obtuvimos diferentes resultados. En la tabla siguiente, se muestran las distintas figuras obtenidas tras el paso de los datos por cada algoritmo. A la izquierda, los datos sin tratar, y a la derecha, con los datos tras ser pasados por un filtro Kalman.



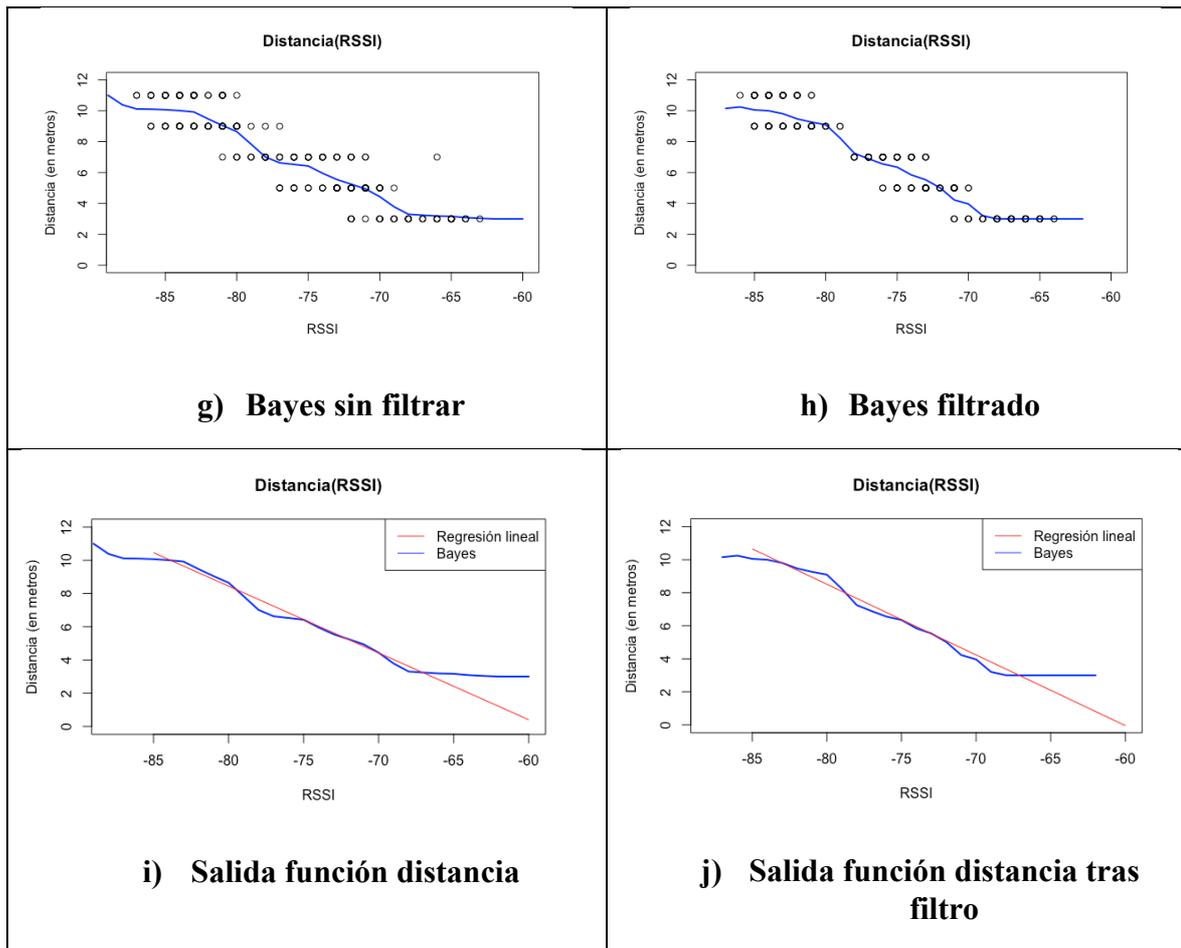


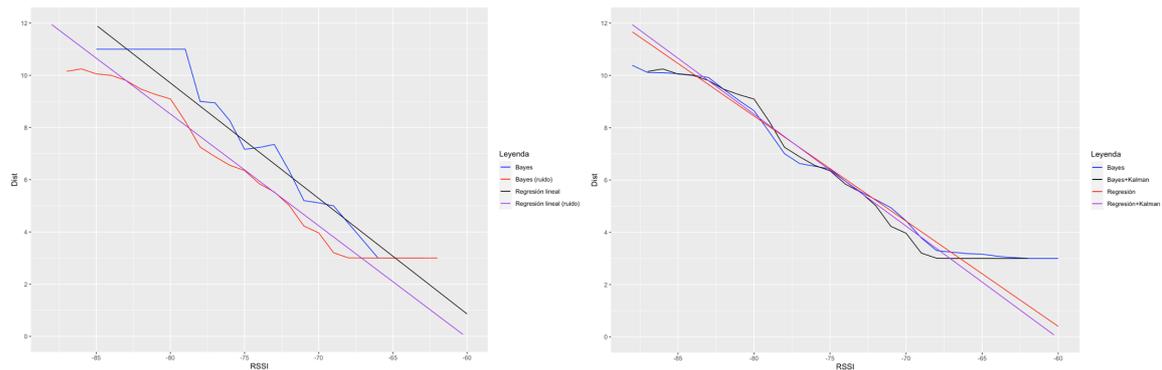
Figura 1. Resultados obtenidos tras aplicar los algoritmos a un set de datos en un entorno ruidoso. A la izquierda, datos previos al filtrado de Kalman, a la derecha, datos tras filtrado de Kalman.

La figura 1 muestra de forma gráfica el efecto de los algoritmos sobre el set de datos introducido. Puede comprobarse el efecto del filtro Kalman especialmente entre las figuras 1.a) y 1.b), donde las distribuciones se separan entre sí en la segunda tras haber aplicado el filtrado.

Además, es notable su efecto entre los diagramas de caja 1.c) y 1.d), donde vemos, en la figura 1.d) se elimina el outlier y se reduce el rango intercuartílico de los diagramas de caja, lo cual se relaciona con la disminución de la varianza del punto anterior.

Entre las figuras 1.e) y 1.f) Puede comprobarse el efecto del filtrado de Kalman sobre la reducción del ruido, el cual, en la figura de la derecha, incrementa el valor del R cuadrático un 6% aproximadamente. También puede verse, aunque de forma menos apreciable, cómo se reducen las bandas de error tras el filtrado.

Finalmente, en el entorno ruidoso no se consiguió que las redes neuronales convergieran correctamente, debido a la alta similitud entre los datos de entrada, solo conseguían funcionar para determinadas distribuciones, con mayor separación. Si bien, tras Kalman las redes neuronales funcionaban algo mejor, seguíamos sin obtener un resultado satisfactorio en el que convergieran para la clasificación de todas las distancias. Por ello en las figuras 1.i) y 1.j) no aparece reflejado el efecto del algoritmo de RRNN.



*Figura 2. Resultados obtenidos por algoritmos de Bayes y Kalman. A la izquierda, comparación entre algoritmos post-Kalman en entornos con ruido y sin ruido. A la derecha, algoritmos en entorno ruidoso, pre-Kalman y post-Kalman*

Por otro lado, se comparan en la figura 2 los resultados obtenidos por los algoritmos de Bayes y la regresión lineal en entornos con ruido y sin ruido tras un filtrado de Kalman, y a la derecha, en un entorno con ruido previo al filtrado y tras el filtrado.

Observamos en la figura de la izquierda una diferencia entre los algoritmos aplicados al entorno limpio y al entorno ruidoso de aproximadamente 2,5 RSSI de separación. Esta diferencia puede explicarse por la reflexión de la señal que, a una distancia de, por ejemplo, 12 metros, se verá mayormente atenuada en un entorno con obstáculos que en un entorno limpio. Esto también se corrobora si observamos las rectas de regresión a distancias menores, donde parece que tienden a acercarse. Esto tiene sentido, ya que, a una menor distancia, la señal llega al dispositivo, en un entorno con ruido, sin colisionar con tantos obstáculos.

La figura de la derecha muestra el resultado de comparar los algoritmos de regresión lineal y de Bayes previo al filtrado de Kalman y tras el filtrado. Como puede verse, Kalman, además de mejorar el R cuadrático, reduce la pendiente de la regresión lineal, en un mejor ajuste de los datos. En el caso del algoritmo bayesiano, la diferencia no es notable a simple

vista, pero vemos como, tras el filtro de Kalman, la inferencia bayesiana se ajusta más a las distancias desde las que se tomaron las mediciones de forma escalonada, fruto de la reducción del solape entre distribuciones.

Para una comparación más exhaustiva de ambos algoritmos, se calculó el error cuadrático medio:

*Tabla 1. Error cuadrático medio de todos los algoritmos.*

	<b>Entorno sin ruido</b>	<b>Entorno sin ruido + Kalman</b>	<b>Entorno con ruido</b>	<b>Entorno con ruido + Kalman</b>
<b>Regresión lineal</b>	<b>0.52m</b>	<b>0.35m</b>	1.12m	<b>0.66m</b>
<b>Bayes</b>	0.53m	0.42m	<b>0.88m</b>	0.80m

Como puede verse en la tabla, el error cuadrático obtenido es por lo general menor en la regresión lineal. Si bien es cierto que en un entorno con ruido gana el algoritmo bayesiano, tras la aplicación de un filtro de Kalman, la regresión lineal vuelve a superar a la inferencia bayesiana en ambos entornos. Además, se comprueba en la tabla el efecto que tiene el filtro Kalman sobre la reducción del error, siendo mucho más notable la diferencia en la regresión lineal que en el algoritmo de Bayes.

## 7. Conclusiones y trabajos futuros

Los resultados nos demuestran que un algoritmo tan simple como una regresión lineal, puede darnos resultados satisfactorios para el caso de uso.

Teniendo en cuenta que debemos utilizar el algoritmo que mejor opere bajo ruido, pues este va a ser el caso de uso, nos demuestra el error cuadrático medio que la combinación ganadora será la regresión lineal más el filtrado previo de los datos por Kalman, con un error de medición de **0.66m**, aunque no se aleja de su contraparte bayesiana, con una diferencia de tan solo 14cm.

Concluimos además que el indicador RSSI, a pesar de tener correlación con la variable distancia, es un parámetro poco fiable y de muy alta variabilidad, por lo que es necesario tomar un alto número de medidas para obtener resultados coherentes.

Como trabajos futuros, se propone una continuación del experimento, donde se realice un análisis de la señal mucho más exhaustivo. Si bien es cierto que para el caso de uso que nos ocupa en este proyecto no es necesaria mayor precisión, se propone cambiar el caso de uso de nuestro análisis para situaciones donde la precisión sí sea un factor clave.

Ampliar el rango de distancia en las mediciones, para obtener resultados más precisos. Ampliar el número de medidas a distintas distancias, con el fin de refinar el algoritmo de Bayes.

Se propone experimentar con un mayor número de algoritmos de predicción, y con nuevos métodos de filtrado y optimización, en busca de mejorar la precisión en la predicción de distancia.

# PERFORMANCE ANALYSIS OF BLUETOOTH LOW ENERGY THROUGH CROSS-PLATFORM TOOLS FOR MOBILE DEVELOPMENT

**Author: Chávez Macías, Alejandro.**

Supervisor: Contreras Bárcena, David.

Collaborating Entity: ICAI – Universidad Pontificia Comillas

## ABSTRACT

In this project, the operation of different BLE beacon distance prediction algorithms is analyzed in order to obtain the most efficient one to be implemented in a mobile application on Flutter. It is discovered and it is shown that, despite numerous papers recommending complex methods, a simple linear regression could be sufficient, for the case of study.

**Keywords:** Bluetooth, Mobile, Indoor, BLE, RRNN, Predictive algorithms

## 1. Introduction

In a world increasingly connected by telecommunications networks, in recent years there has been a high evolution in IoT technology (an acronym for Internet of Things). The current pandemic context makes clear the need to implement such IoT solutions to use cases in social settings such as guided tours, where direct contact is necessary and at the same time can pose a risk to health. Nowadays, the implementation would be very simple, and we have the advantage that currently anyone has a Smartphone, capable of processing information and with which the user has the possibility of interacting. This, added to the fact that any current Smartphone has BLE technology implemented, makes this the ideal device to be implemented in such use cases.

## 2. Project definition

The project will consist of the performance analysis of the BLE technology implemented on mobiles using the Flutter platform, so that said application can be implemented on both iOS and Android phones. Said analysis will serve to implement a use case such as a circuit with beacons, which can be connected to the device on which the application developed in Flutter is installed. The application will process the data received by the beacon, depending on the distance of the user, and the proximity or distance with respect to one or the other beacons.

For an application dependent on external devices to be functional and efficient, it will be necessary to perform a performance analysis to know how the distance between beacons, the presence of disturbances in the environment, or the battery of the devices can affect it, since the application should be as efficient as possible.

Therefore, the purpose of the project is to carry out a study of different prediction algorithms, applied to the analysis of the BLE signal. From the signal power value, RSSI, received by the device, these algorithms must predict the distance the user is from the beacon. For the proposed end-use case, the objective should be, receiving the power of the signals in the middle, to discriminate between the beacons closest to the user.

Therefore, signal measurement experiments will be carried out and the results obtained with the algorithms will be compared.

### **3. Experiments description**

#### **Software used:**

To carry out the experiment, a mobile application has been developed using the Flutter platform and an external library that adds BLE functionalities. This application has allowed us to carry out the necessary scans to obtain the data with which to carry out the experiment.

To implement the algorithms, the RStudio tool is used, which facilitates the analysis of the signal.

#### **Hardware:**

The hardware used has been CC2650 beacons, which emit the BLE signal, and a mobile phone which will receive the beacon signal through the Flutter application.

#### **Experiments:**

- **Stage:** The stage in which the experiments have been carried out is a room 12 meters long. Placing the beacon at one end of the room, measurements were made for one minute at distances of 3, 5, 7, 9 and 11 meters. Two measurement situations have been taken, a noise-free environment and a noisy environment.

- **Noisy environment:** For a noisy environment, a normal situation has been assumed for our use case. The user will have the mobile device in his hand and will carry out scans (the application would carry them out in the background) while other users will pass around him.

- **Noise-free environment:** To achieve an environment with the least possible noise, measurements have been taken in the same room, without people or obstacles in between, and placing the mobile device on a table, at the same height as the beacon.

After taking the necessary measurements, we proceed to the analysis using different signal filtering and prediction algorithms, in order to discern between the algorithms most capable of achieving our objective: the correct distance prediction and the discrimination between the proximity of beacons.

#### **4. State of art**

Numerous authors have carried out previous studies that relate the BLE signal power indicator with distance, in the search to find algorithms capable of explaining a relationship between both variables.

For this, these authors make use of traditional classification algorithms such as linear regressions, more or less sophisticated statistics or more complex algorithms used today as neural networks.

The researchers, to carry out their analysis of the signal, take into account the high variability of the RSSI indicator, and that taking a point measurement will not be representative of the signal strength at a point. That is why the authors take a set of several measurements per distance to perform analyzes more in line with reality.

In the case of using statistics, for example, most authors make use of the mean, as an indicator of the power at a point in space, when implementing RSSI variables in the model. Although this simplification, although it speeds up the distance resolution process, it supposes a loss of information.

## **5. Filtering techniques**

For our signal analysis and prediction purposes, the following prediction and filtering algorithms are employed:

### **5.1. Linear regression algorithms**

A linear regression model will be applied to the measurements obtained by analyzing the regression coefficient to measure the correlation between the RSSI and the distance and the dependent and independent variables of all the experiments carried out.

### **5.2. Bayesian inference**

For the Bayesian inference algorithm:

- A density histogram is made of each RSSI data distribution by distance, so that we have.
- Taking into account that said distributions coincide with each other in certain power values with a certain probability, the estimated distance from the user will be calculated from said probabilities. If we are at a point whose probability that it is in the 5-meter distribution is 0.7 and that it is in the 7-meter distribution of 0.3, an estimate of the distance is made by Bayes.
- To create the Bayesian model, the previous procedure is carried out for each RSSI value within the range occupied by the introduced distributions.

### **5.3. RRNN**

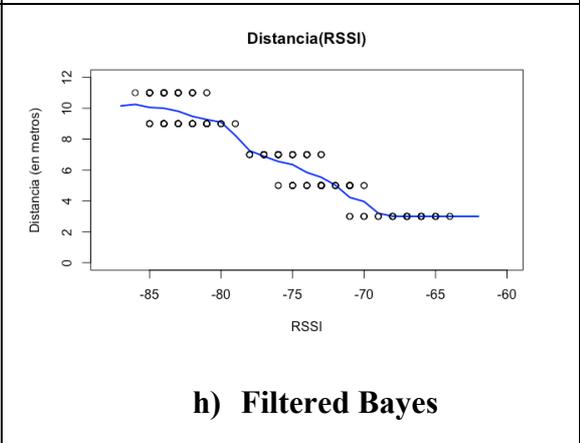
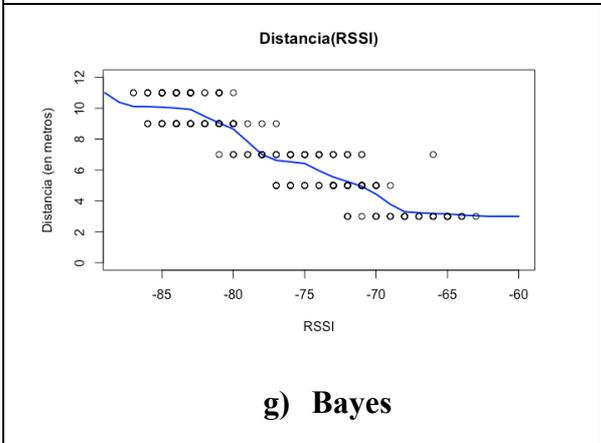
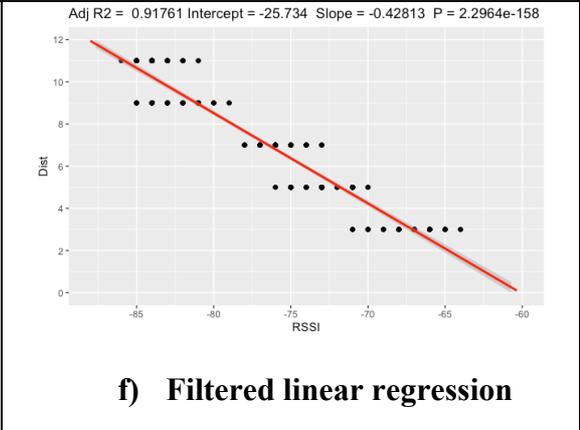
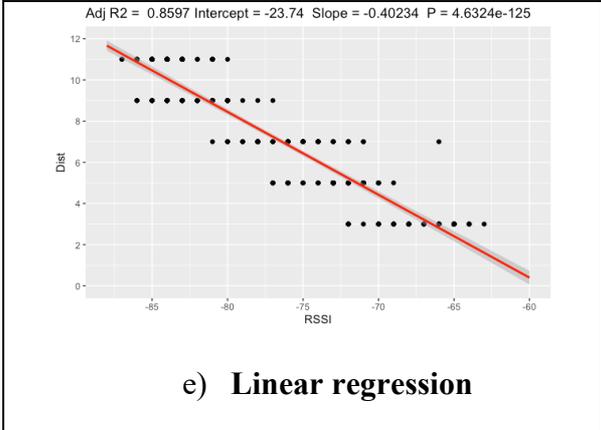
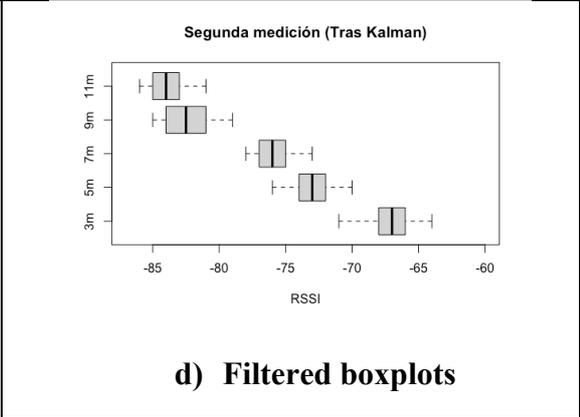
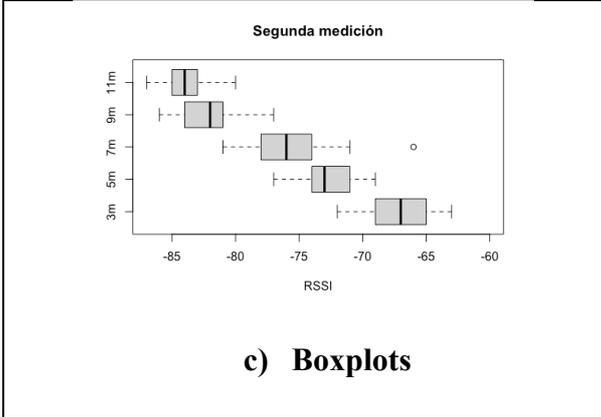
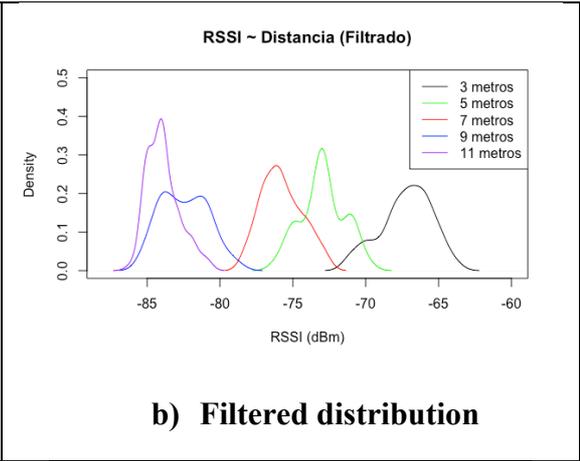
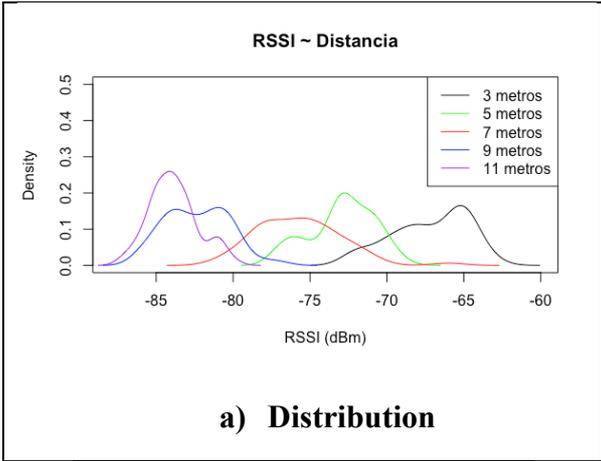
The data will be classified by means of a back propagation algorithm with neural networks, in order to predict the distance to which a certain RSSI belongs. A neural network will be trained for each distribution that is capable of determining whether or not an RSSI value belongs to the associated distance data distribution.

### **5.4. Kalman filter**

The initial data will be passed through a Kalman filter, which will reduce the noise in the signal, and said data already treated by the filter will be passed again by the previous algorithms, in order to check the effect of the filtering on the signal.

## **6. Results**

After applying different algorithms, we obtained different functions. The following table shows the different figures obtained after passing the data through each algorithm. On the left, the raw data, and on the right, with the data after being passed through a Kalman filter.



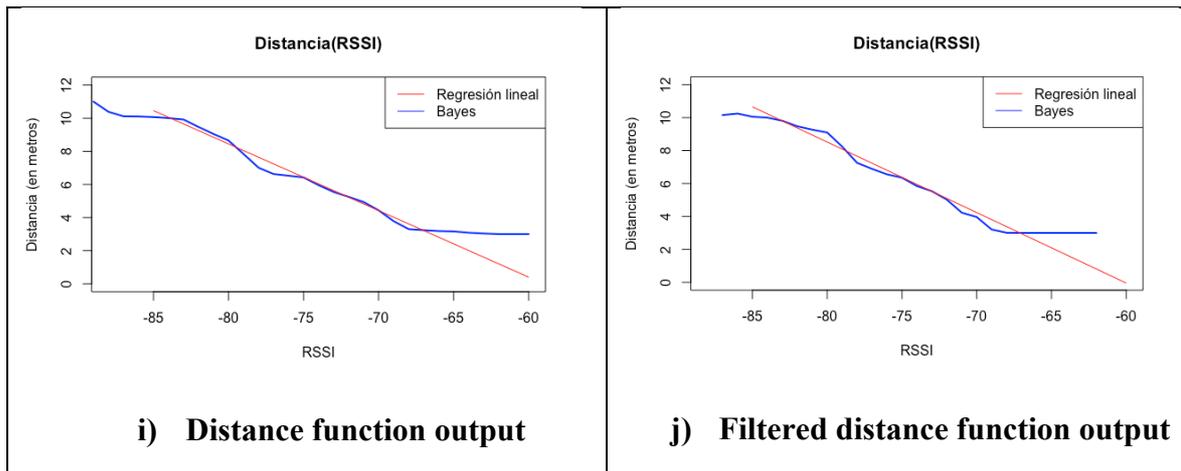


Figure 1. Results obtained after applying algorithms to data-set in noisy environment.

Figure 1 graphically shows the effect of the algorithms on the entered data set. The effect of the Kalman filter can be seen especially between Figures 1.a) and 1.b), where the distributions are separated from each other in the second after applying the filtering.

In addition, its effect is notable between box plots 1.c) and 1.d), where we see, in figure 1.d) the outlier is eliminated and the interquartile range of the box plots is reduced, which is related to the decrease in the variance of the previous point.

Between Figures 1.e) and 1.f) The effect of Kalman filtering on noise reduction can be seen, which, in the figure on the right, increases the value of the quadratic R by approximately 6%. It can also be seen, albeit less appreciably, how the error bands are reduced after filtering.

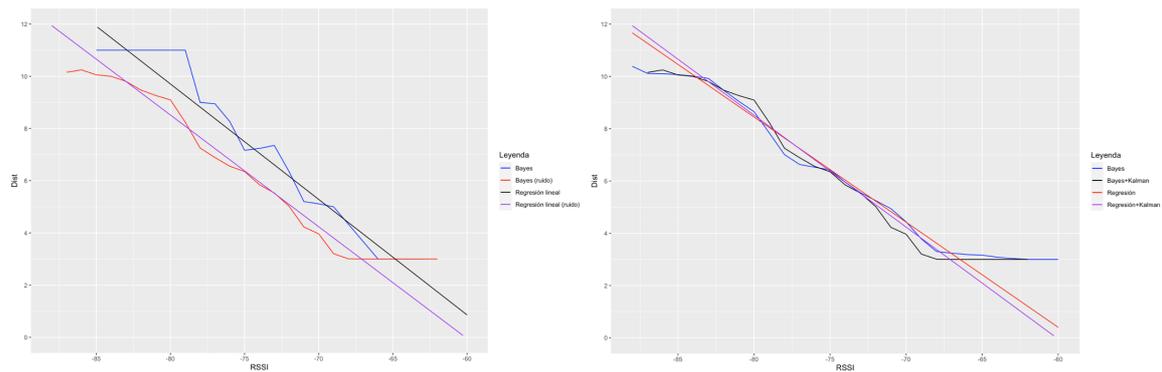


Figure 2. Results obtained by Bayes and Kalman algorithms. On the left, comparison between post-Kalman algorithms in noisy and noisy environments. On the right, algorithms in noisy environment, pre-Kalman and post-Kalman

On the other hand, the results obtained by Bayes algorithms and linear regression in noisy and noisy environments after Kalman filtering are compared in Figure 2, and on the right, in an environment with noise prior to filtering and after filtering. filtered out.

We observe in the figure on the left a difference between the algorithms applied to the clean environment and the noisy environment of approximately 2.5 RSSI of separation. This difference can be explained by the reflection of the signal which, at a distance of, for example, 12 meters, will be more attenuated in an environment with obstacles than in a clean environment. This is also corroborated if we observe the regression lines at shorter distances, where they seem to tend to approach each other. This makes sense, since, at a shorter distance, the signal reaches the device, in a noisy environment, without colliding with so many obstacles.

The figure on the right shows the result of comparing the linear and Bayes regression algorithms before the Kalman filtering and after the filtering. As can be seen, Kalman, in addition to improving the quadratic R, reduces the slope of the linear regression, in a better fit of the data. In the case of the Bayesian algorithm, the difference is not noticeable to the naked eye, but we see how, after the Kalman filter, the Bayesian inference is more adjusted to the distances from which the measurements were taken in a staggered way, as a result of the reduction of the overlap between distributions.

For a more exhaustive comparison of both algorithms, the mean square error is calculated:

	Noise-free environment	Noise-free environment + Kalman	Noisy environment	Noisy environment + Kalman
<b>Linear regression</b>	<b>0.5242516</b>	<b>0.3513124</b>	1.118551	<b>0.6568557</b>
<b>Bayes</b>	0.5329002	0.4226761	<b>0.8841576</b>	0.7985478

*Table 1. Mean square error comparison*

As can be seen from the table, the squared error obtained is generally smaller in linear regression. Although it is true that in a noisy environment the Bayesian algorithm wins, after applying a Kalman filter, linear regression once again outperforms Bayesian inference in both environments. In addition, the effect that the Kalman filter has on reducing the error is verified in the table, the difference being much more notable in the linear regression than in the Bayes algorithm.

## 7. Conclusions and future work

The results show us that an algorithm as simple as a linear regression can give us satisfactory results for the use case.

Taking into account that we must use the algorithm that best operates under noise, since this will be the use case, the mean square error shows us that the winning combination will be the linear regression plus the previous filtering of the data by Kalman, with a 0.66m measurement error, although it is not far from its Bayesian counterpart, with a difference of only 14cm.

We also conclude that the RSSI indicator, despite having a correlation with the distance variable, is an unreliable parameter with very high variability, so it is necessary to take a high number of measurements to obtain consistent results.

As future work, a continuation of the experiment is proposed, where a much more exhaustive signal analysis is carried out. Although it is true that for the use case that concerns us in this project, greater precision is not necessary, it is proposed to change the use case of our analysis for situations where precision is a key factor.

Expand the range of distance in measurements, to obtain more accurate results. Expand the number of measurements at different distances, in order to refine the Bayes algorithm.

It is proposed to experiment with a greater number of prediction algorithms, and with new filtering and optimization methods, in order to improve the precision of distance prediction.

## *Índice de la memoria*

<b>Capítulo 1. Introducción .....</b>	<b>8</b>
1.1 Motivación del proyecto .....	8
<b>Capítulo 2. Descripción de las Tecnologías.....</b>	<b>10</b>
2.1 Protocolo BLE .....	10
2.1.1 Pila del protocolo BLE. Capa del controlador.....	11
2.2 Flutter.....	16
2.2.1 Funcionalidades de Flutter.....	16
2.3 FlutterBleLib .....	17
<b>Capítulo 3. Estado de la Cuestión .....</b>	<b>18</b>
3.1 Modelos de atenuación de señal .....	18
3.2 Filtrado de Kalman .....	20
3.3 Implementación de algoritmos de Backpropagation .....	21
3.4 Otros métodos.....	23
<b>Capítulo 4. Definición del Trabajo .....</b>	<b>24</b>
4.1 Justificación .....	24
4.2 Objetivos.....	25
4.3 Metodología.....	25
<b>Capítulo 5. Sistema/Modelo Desarrollado.....</b>	<b>26</b>
5.1 Desarrollo de aplicación Medidor .....	26
5.2 Rendimiento de la batería del dispositivo móvil .....	29
5.2.1 Parametrización de FlutterBleLib.....	29
5.2.2 Experimento: Rendimiento de dispositivos Android en función de intervalos de escaneo de ventana.....	31
5.3 Descripción de algoritmos de filtrado.....	32
5.3.1 Filtro de Kalman.....	32
5.3.2 Boxplots .....	33
5.3.3 Regresión lineal.....	34
5.3.4 Bayes.....	35
5.3.5 RRNN .....	36

---

5.4 Propuesta de combinación de los algoritmos .....	37
<b>Capítulo 6. Análisis de Resultados.....</b>	<b>38</b>
6.1 Análisis de rendimiento de batería .....	38
6.1.1 Rendimiento de la batería del móvil sin realizar scans.....	38
6.1.2 Modo low-power.....	39
6.1.3 Modo balanced .....	40
6.1.4 Modo low-lattency .....	41
6.1.5 Modo opportunistic.....	41
6.2 Análisis de datos de distancia limpios .....	42
6.2.1 Filtrado de Kalman de datos limpios .....	54
6.3 Análisis de datos de distancia con ruido.....	66
6.3.1 Filtrado de Kalman de datos con ruido.....	73
6.4 Comparativas finales de datos de distancia .....	81
<b>Capítulo 7. Conclusiones y Trabajos Futuros.....</b>	<b>89</b>
<b>Capítulo 8. Bibliografía.....</b>	<b>95</b>
<b>ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS.....</b>	<b>98</b>

## Índice de figuras

Figura 1. Resultados obtenidos tras aplicar los algoritmos a un set de datos en un entorno ruidoso. A la izquierda, datos previos al filtrado de Kalman, a la derecha, datos tras filtrado de Kalman.....	14
Figura 2. Figuras de comparación de algoritmo bayesiano, Kalman y regresión lineal .....	<b>¡Error! Marcador no definido.</b>
Figura 3. Capa del controlador de la pila del protocolo BLE.....	11
Figura 4. Análisis en frecuencia de BLE. Extraído de [1].....	12
Figura 5. Ancho de banda de BLE frente a IEEE 802.11b. Recuperado de <a href="https://microchipdeveloper.com/wireless:ble-phy-layer">https://microchipdeveloper.com/wireless:ble-phy-layer</a> .....	13
Figura 6. Tiempos de ventana de advertising de baliza haciendo uso de protocolo BLE. Extraído de [1].....	13
Figura 7. Tiempos de ventana de escaneo de dispositivo. Extraído de [1].....	15
Figura 8. Logo de Flutter. Recuperado de <a href="http://flutter.dev">http://flutter.dev</a> .....	16
Figura 9. Ejemplo de aplicación de filtro Kalman sobre señales obtenidas de 4 balizas. Extraído de [13].....	20
Figura 10. Ejemplo de red neuronal aplicada a la predicción de distancia en función del valor RSSI. Extraído de [13].....	21
Figura 11. Resultado de estudio de aplicación de RRNN a la predicción de distancia en función de valores RSSI junto a mapa de balizas. Extraído de [14]. .....	22
Figura 12. Pantalla 1 de aplicación Medidor. Listado de balizas cercanas. ....	27
Figura 13. Pantalla 2 de aplicación Medidor. ....	28
Figura 14. Algoritmo de Kalman ilustrado. Extraído de [5]. ....	32
Figura 15. Descargue de batería de Samsung Galaxy S10 sin realizar scans.....	38
Figura 16. Descargue de batería de Samsung Galaxy S10 escaneando en modo low-power. ....	39

Figura 17. Descargue de batería de Samsung Galaxy S10 escaneando en modo balanced..	40
Figura 18. Descargue de batería de Samsung Galaxy S10 escaneando en modo low-lattency.	41
Figura 19. Distribución de valores RSSI para medidas en entorno controlado durante un minuto.	43
Figura 20. Diagrama de cajas de medidas en entorno controlado durante un minuto.	44
Figura 21. Medidas estadísticas de medidas en entorno controlado durante un minuto. ....	45
Figura 22. Modelo de regresión lineal para medidas en entorno controlado durante un minuto.	46
Figura 23. Relación Distancia-RSSI a partir de método bayesiano para medidas en entorno controlado durante un minuto, comparada con datos puntuales obtenidos de las distribuciones.	48
Figura 24. Red neuronal para clasificar si un dato pertenece o no a la distribución de 3 metros entrenada con medidas tomadas en entorno sin ruido durante un minuto.	50
Figura 25. Comparativa de resultados tras aplicar Regresión lineal, Bayes y RRNN sobre medidas en entorno controlado durante un minuto	51
Figura 26. Comparativa entre salida de función distancia e histograma inicial para medidas tomadas en entorno controlado durante un minuto	52
Figura 27. Filtrado de la señal limpia a través del filtro de Kalman. Se muestra en la figura el filtrado de cada distribución. 3 metros: arriba izquierda, 5 metros arriba derecha, 7 metros: centro izquierda, 9 metros: centro derecha, 11 metros: abajo izquierda.	54
Figura 28. Distribución de valores RSSI para medidas en entorno controlado durante un minuto tras filtro Kalman.	56
Figura 29. Diagrama de cajas de medidas en entorno controlado durante un minuto tras pasar por filtro Kalman.	57
Figura 30. Resultados estadísticos de los datos limpios obtenidos tras pasar por filtro Kalman.	57
Figura 31. Modelo de regresión lineal para medidas en entorno controlado durante un minuto tras pasar por filtro Kalman.	58

Figura 32. Relación Distancia-RSSI a partir de método bayesiano para medidas en entorno controlado durante un minuto tras pasar por filtro Kalman.....	59
Figura 33. Comparativa de resultados tras aplicar Regresión lineal, Bayes y RRNN sobre medidas en entorno controlado durante un minuto tras filtro Kalman. ....	61
Figura 34. Comparativa de algoritmos sobre medidas realizadas en entorno limpio durante un minuto. ....	63
Figura 35. Distribución de valores RSSI para medidas en entorno con ruido durante un minuto. ....	66
Figura 36. Diagrama de cajas para medidas en entorno con ruido durante un minuto. ....	67
Figura 37. Medidas estadísticas para medidas en entorno con ruido durante un minuto ....	67
Figura 38. Modelo de regresión lineal para medidas en entorno con ruido durante un minuto. ....	68
Figura 39. Relación Distancia-RSSI a partir de método bayesiano para medidas en entorno ruidoso durante un minuto, comparada con datos puntuales obtenidos de las distribuciones. ....	70
Figura 40. Salida de función distancia de datos con ruido tras filtrado de Kalman. ....	72
Figura 41. Filtrado de Kalman de datos con ruido. Se muestra en la figura el filtrado de cada distribución. 3 metros: arriba izquierda, 5 metros arriba derecha, 7 metros: centro izquierda, 9 metros: centro derecha, 11 metros: abajo izquierda. ....	73
Figura 42. Distribución de valores RSSI para medidas en entorno con ruido durante un minuto tras filtro de Kalman.....	74
Figura 43. Diagramas de caja de datos con ruido tras filtrar por Kalman.....	75
Figura 44. Medidas estadísticas de las mediciones con ruido tras el filtro Kalman.....	75
Figura 45. Modelo de regresión lineal para medidas en entorno con ruido durante un minuto tras pasar por filtro Kalman. ....	76
Figura 46. Relación Distancia-RSSI a partir de método bayesiano para medidas en entorno ruidoso durante un minuto, tras pasar por filtro Kalman, comparada con datos puntuales obtenidos de las distribuciones .....	78
Figura 47. Resultado de la función de distancia tras recibir datos con ruido pasados por el filtro Kalman.....	80

Figura 48. Comparativa entre algoritmo de Regresión lineal frente a Bayes tras el filtro de Kalman en entorno sin ruido y en entorno con ruido. ....	81
Figura 49. Comparativa entre algoritmo de Regresión lineal frente a Bayes, previo al filtro de Kalman y tras el filtro en entorno con ruido. ....	83

## *Índice de tablas*

Tabla 1. Error cuadrático medio de todos los algoritmos.....	16
Tabla 2. Comparativa de error cuadrático de algoritmo bayesiano y de regresión lineal en entorno sin ruido.....	64
Tabla 3. Comparación de error cuadrático para los distintos algoritmos.....	84
Tabla 4. Comparativa de figuras sin ruido.....	85
Tabla 5. Comparativa de figuras con ruido.....	87
Tabla 6. Alineación del proyecto con los Objetivos de Desarrollo Sostenible.....	98

## **Capítulo 1. INTRODUCCIÓN**

### ***1.1 MOTIVACIÓN DEL PROYECTO***

Vivimos en un momento único en la historia, el ser humano nunca había estado tan conectado como ahora. Gracias a las telecomunicaciones, y al cada vez más rápido avance de las tecnologías, podemos conocer la historia del mundo tal como sucede, podemos mantenernos en contacto con personas en cualquier parte del globo y recibir cualquier tipo de información en cuestión de segundos a través de la pantalla de nuestros ordenadores o, como bien estamos acostumbrados ahora, a través de nuestros dispositivos móviles, disponibles en cualquier momento y al alcance de nuestra mano, guardados en nuestros bolsillos. A esto hay que añadir la proliferación de las tecnologías IoT (Internet of Things) las cuales han fomentado que ya no solo las personas sean capaces de estar conectados, sino que cualquier dispositivo tan simple como una cafetera es capaz de recibir información de cualquier parte del mundo.

La imparable y cada vez más extendida comunicación global ha facilitado muchas facetas de nuestra vida y esto se ha hecho especialmente latente durante la pandemia del Covid-19. Esta pandemia, la cual comenzó en España con un confinamiento obligatorio para toda la población, nos hizo darnos cuenta de lo necesarias que son las telecomunicaciones, y de lo difícil que habría sido vivir una cuarentena sin el contacto con el exterior. Sin embargo, aún con la tecnología actual, no se han visto aplicaciones orientadas al control de una pandemia hasta ahora, que ha ocurrido. Por suerte, como ya disponemos de la técnica necesaria, es sencillo desarrollar dichas aplicaciones, y como tal, varios desarrolladores se han puesto manos a la obra.

Se hace evidente entonces buscar nuevas alternativas en la tecnología para volver a un estado de vida como el anterior, e incluso, mejorarlo. Hemos visto aplicaciones orientadas al control de la pandemia como el conocido Radar-Covid, impulsado por el gobierno, el cual

funciona a partir de la distancia interpersonal, pero aún así, existe todavía mucho campo de aplicación sin explorar.

Es por ello por lo que, se propone la creación de una aplicación que ayude a revitalizar uno de los sectores golpeados por el efecto de la pandemia, y este es el de las visitas guiadas a museos, exposiciones o universidades. En concreto, en este proyecto nos centraremos en una visita guiada a ICAI, la cual este año no ha podido organizar su habitual visita guiada, precisamente por el peligro del coronavirus.

La idea para el desarrollo de esta aplicación consiste en que el usuario portador del dispositivo móvil irá recorriendo distintas salas de exposición en las que se encontrarán situadas varias balizas. La aplicación será capaz de distinguir entre dichas balizas y mostrar al usuario la información en pantalla adecuada en función de la sala en la que este se encuentre.

Por otra parte, es bien sabido que desarrollar una aplicación multiplataforma es un proceso costoso, pues es necesario desarrollar una versión funcional para iOS y otra para Android. Se evitará dicho coste añadiendo una capa de abstracción al desarrollo del proyecto gracias a la herramienta Flutter, la cual facilita los tiempos de desarrollo al proveer de una compilación multiplataforma a raíz de un único código fuente.

Dicho proyecto se compondrá de dos partes diferenciadas, el desarrollo de la aplicación que llevará el usuario durante la visita, y el desarrollo de la funcionalidad BLE para esta, con un análisis documentado para garantizar el correcto funcionamiento de la red.

Este TFG se centrará en la parte del análisis del rendimiento de la funcionalidad BLE, necesaria para que la aplicación funcione correctamente, sea intuitiva para el usuario y ofrezca a este una experiencia equiparable a la de una visita guiada previa al estado de pandemia. La aplicación será desarrollada por mi compañera Teresa González García, quien desarrollará su parte en su respectivo TFG.

## **Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS**

Se describen aquí las tecnologías utilizadas para el desarrollo del presente proyecto, así como su funcionamiento. Se desarrolla en profundidad la estructura del protocolo BLE, explicándose cada parte de su pila. También se expone en este punto el entorno de desarrollo Flutter, junto a la librería externa FlutterBleLib, necesaria para implementar funcionalidad Bluetooth a la aplicación. Por último, se mostrarán las balizas utilizadas para la etapa de experimentación.

### **2.1 PROTOCOLO BLE**

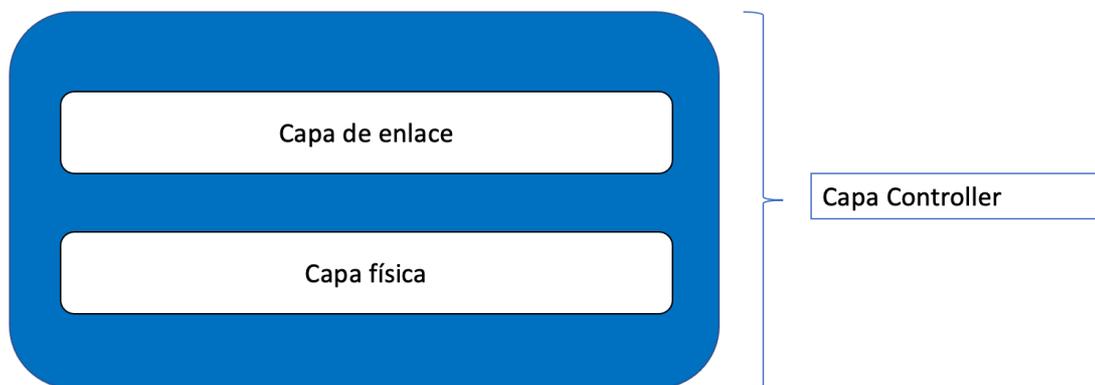
La tecnología Bluetooth Low Energy (por sus siglas, BLE), se implementa en la versión 4 del protocolo Bluetooth. Esta versión de Bluetooth fue desarrollada por el Bluetooth Special Interest Group (SIG), teniendo en cuenta mejorar dos elementos de anteriores versiones: Las fallas en seguridad, y el elevado consumo de energía. Es por ello por lo que BLE está enfocado en aplicaciones de bajo consumo energético, y aquellas en las que se realice una baja transmisión de datos.

Esta tecnología, debido a su bajo consumo y su buen rendimiento, es el candidato perfecto para aplicaciones en el ámbito de las IoT, pudiendo aplicarse a dispositivos pequeños que requieran estar operativos la mayor parte del tiempo.

### **2.1.1 PILA DEL PROTOCOLO BLE. CAPA DEL CONTROLADOR.**

Se desarrolla el funcionamiento de la capa del controlador, pues es la capa que nos concierne en la investigación.

Esta capa se encuentra subdividida en dos subcapas, la capa de enlace y la capa física, además de la interfaz host-controlador (HCI), la cual sirve para enlazar la capa Controller con la capa host [2].



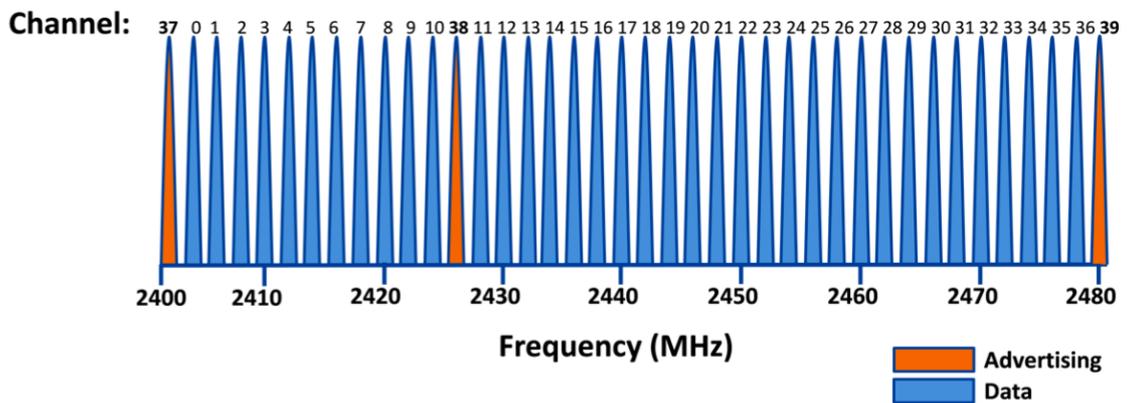
*Figura 3. Capa del controlador de la pila del protocolo BLE.*

#### **2.1.1.1 Capa física**

##### **Funcionamiento y rendimiento de BLE**

BLE permite una velocidad máxima de 1Mbit con un alcance teórico de hasta 50 metros. Al ser BLE un protocolo sin conexión, en el que no es necesario que 2 dispositivos estén conectados todo el tiempo, es capaz de llegar a necesitar tan solo el 10% de lo que consumiría un dispositivo utilizando Bluetooth tradicional. BLE opera en una banda de frecuencia ISM sin licencia a 2,4GHz, a lo largo de 40 canales, extendiéndose en una banda de 2402MHz a 2480MHz, con cada canal ocupando un ancho de banda de 1MHz [1].

El funcionamiento de BLE en la capa física puede verse en el siguiente análisis en frecuencia:



*Figura 4. Análisis en frecuencia de BLE. Extraído de [1]*

Los canales 37, 38 y 39 son los canales de advertising de los dispositivos, mientras el resto de los canales se utilizan para la transmisión de datos.

A estos canales se les han asignado frecuencias que minimizan el overlapping con los canales 1, 6 y 11 de IEEE 802.11b [2], como puede verse en la *Figura 3*.

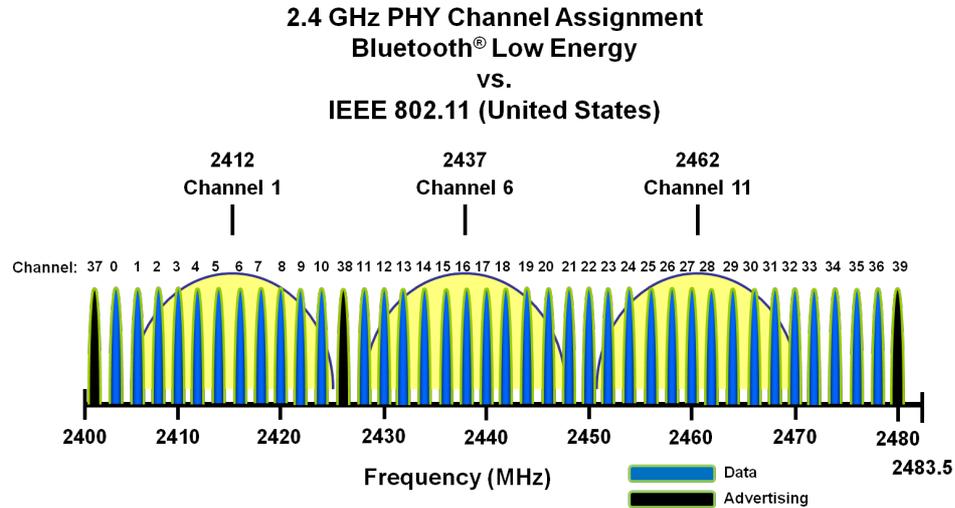


Figura 5. Ancho de banda de BLE frente a IEEE 802.11b. Recuperado de <https://microchipdeveloper.com/wireless:ble-phy-layer>

### 2.1.1.2 Capa de enlace

#### 2.1.1.2.1 Advertisers

Los dispositivos que emiten paquetes de advertising, a través de los canales de advertising, son conocidos como advertisers.

Existen dos tipos de eventos de *advertising*: *directed* y *undirected*.

Se muestran en la figura los tiempos de ventana de advertising:

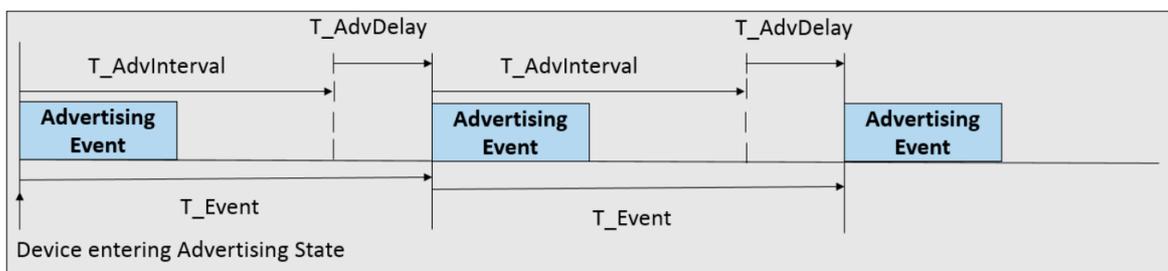


Figura 6. Tiempos de ventana de advertising de baliza haciendo uso de protocolo BLE. Extraído de [1]

El período  $T_{AdvInterval}$  debería tener un valor múltiplo de 0.625 milisegundos, y su valor estará comprendido entre 20 ms y 10.24 segundos [1].

### 2.1.1.2.2 Scanners

Un dispositivo puede considerarse scanner cuando está atento al medio en busca de señales de advertising. El scanner escucha a cada canal durante un periodo determinado por  $T\_ScanWindow$ .  $T\_ScanInterval$  por su parte, es el intervalo entre escaneos de ventana. Si ambos periodos son iguales, el escaneo que realiza el dispositivo puede considerarse continuo. Un escaneo continuo implica un consumo acelerado de la batería del dispositivo, por lo que no suele recomendarse aplicar periodos de ventana de escaneo muy elevados.

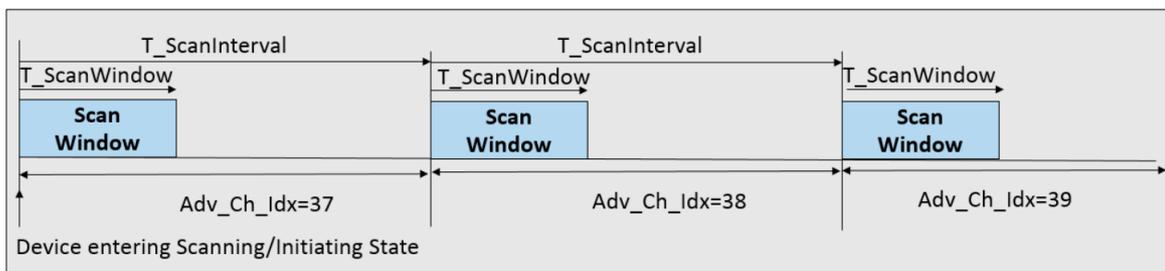


Figura 7. Tiempos de ventana de escaneo de dispositivo. Extraído de [1].

## 2.2 FLUTTER



*Figura 8. Logo de Flutter. Recuperado de <http://flutter.dev>*

Flutter es un toolkit open source de desarrollo de aplicaciones creado por Google, cuya característica principal es su versatilidad, de forma que una aplicación programada en el entorno de Flutter puede ser compilada para dispositivos iOS y Android indistintamente.

### 2.2.1 FUNCIONALIDADES DE FLUTTER

Por un lado, tenemos la calidad nativa que ofrece Flutter en cada dispositivo, pues el código de Flutter se traduce completamente para cada plataforma en la que se implemente. Las aplicaciones desarrolladas en Flutter son, por tanto, rápidas en su ejecución, con tiempos de carga inferiores al segundo.

Por último, cabe destacar la característica Hot-Reload, la cual nos permite realizar cambios en el código y ver los cambios realizados a la aplicación de forma inmediata en la pantalla del dispositivo, sin la necesidad de tener que reinstalar la aplicación como ocurriría en otros entornos de desarrollo.

## **2.3 FLUTTERBLELIB**

La funcionalidad BLE no viene implementada en Flutter de serie, al menos en la versión actual, por lo que es necesario hacer uso de herramientas de terceros para lograr nuestro objetivo final de conectividad con balizas.

Existen numerosas librerías gratuitas que pueden usarse para implementar la funcionalidad BLE en nuestro futuro aplicativo. Sin embargo, se ha decidido utilizar FlutterBleLib para este caso de estudio debido a su alto nivel de parametrización, el cual nos permitirá hacer un mayor número de pruebas que con otras alternativas más sencillas de utilizar, pero más simples en contenido, como flutterblue.

FlutterBleLib permitirá realizar un análisis de rendimiento en el caso de los dispositivos Android, por lo que descubriremos qué modo de rendimiento es más costoso energéticamente, y si merece la pena implementarlo. Estos modos de rendimiento están relacionados con la idea anterior de los periodos de ventana de escaneo, explicados en el punto: *Capa de enlace*.

## Capítulo 3. ESTADO DE LA CUESTIÓN

Existen ya numerosos estudios acerca de cómo lograr obtener el posicionamiento de una baliza conociendo el valor de la potencia que esta emite, tomando como punto de referencia la posición del receptor. Para ello, dichos estudios han hecho uso de distintos modelos matemáticos, algoritmos de predicción y, más recientemente, entrenamiento de redes neuronales.

### 3.1 *MODELOS DE ATENUACIÓN DE SEÑAL*

Para empezar, se muestran tres modelos de atenuación de señal en el canal utilizados en la propagación de señal a gran escala [11].

- **Modelo de propagación libre**

El modelo de propagación libre se usa para predecir la potencia de la señal recibida cuando no existe obstáculo entre el punto inicial y final, es decir, la señal alcanza su destino “en línea recta”. Este modelo no nos sirve, pues no es representativo de un caso de uso real.

- **Modelo de reflexión**

Supone grandes distancias (al menos un kilómetro) y que antenas emisora y receptora estén a una alta altura (50 metros o más). Por ello, este modelo tampoco es útil en nuestro caso.

- **Modelo log-distancia**

El modelo log-distancia sería más apropiado para nuestro caso, pues es un método aplicado en ámbitos tanto indoor como outdoor, sin importar la distancia y teniendo en cuenta obstáculos en el medio.

Supone que la relación entre potencia y distancia puede modelarse como una relación logarítmica.

$$RSSI = -10n \log\left(\frac{d}{d_0}\right) + A + X\sigma$$

Siendo  $d$ , distancia entre baliza emisora y receptor,  $n$ , parámetro relacionado con el medio de transmisión, de forma que  $n$  aumenta según el número de obstáculos en el medio.  $A$  es el RSSI que hay a una distancia  $d_0$  del transmisor (suele usarse distancia 1 metro para simplificar) y  $X$  una distribución gaussiana aleatoria de media 0 [13].

A pesar de que este modelo pudiera parecer solucionar el problema, lo cierto es que, en un caso real, no sería recomendable su implementación, pues, además de ser un modelo general y, por tanto, con posible amplio margen de error, sería necesario conocer los parámetros necesarios en tiempo real. En un entorno cambiante, como lo es una visita guiada, el número de obstáculos, esto es, personas, es variable, por lo que deberíamos saber cómo ajustar el parámetro  $n$  constantemente.

### 3.2 FILTRADO DE KALMAN

Un problema recurrente en la predicción del posicionamiento mediante el valor RSSI es la pureza en la señal. Tras la toma de mediciones de potencia en un punto, la señal puede aparecer con una alta oscilación de valores, aunque el usuario no haya cambiado su posición, debido a las perturbaciones en el ambiente. Una solución utilizada por investigadores es la implementación de un filtro Kalman para reducir outliers [7, 13, 15].

El filtro de Kalman es un método recurrente a la hora de predecir datos con poco conocimiento de las variables externas que actúan sobre estos [5-6].

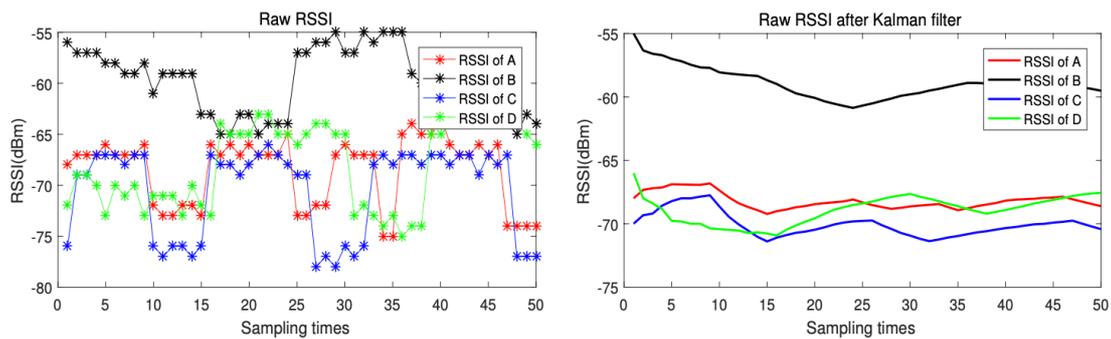


Figura 9. Ejemplo de aplicación de filtro Kalman sobre señales obtenidas de 4 balizas. Extraído de [13].

En la figura se muestra, en el lado derecho, el efecto correctivo que posee el filtro sobre 50 valores de potencia RSSI tomados de forma seguida, los cuales aparecen sin filtrar en el lado izquierdo.

### 3.3 IMPLEMENTACIÓN DE ALGORITMOS DE BACKPROPAGATION

Otros estudios implementan redes neuronales con el fin de clasificar valores de potencia recibida [13, 15, 20]. Las redes neuronales son algoritmos inspirados en el funcionamiento biológico de las neuronas, conectadas unas con otras mediante las dendritas (inputs) y el axón (output). En función de si la suma de los inputs es suficiente, la neurona decidirá si disparar o no a través de su axón. Dicho sistema, es el implementado, de forma computacional, en redes neuronales.

En algoritmos de propagación trasera, el error obtenido a la salida se propaga a los nodos traseros, con el fin de ajustar los pesos y reducir el error de la red. Esto se realiza tras numerosas iteraciones [14].

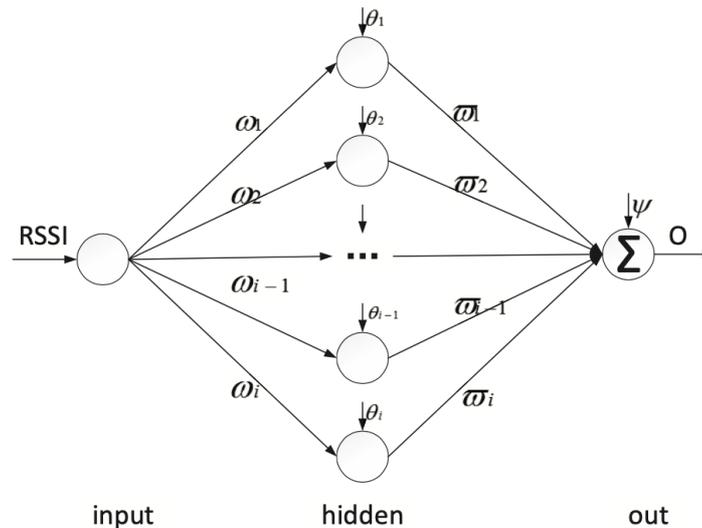
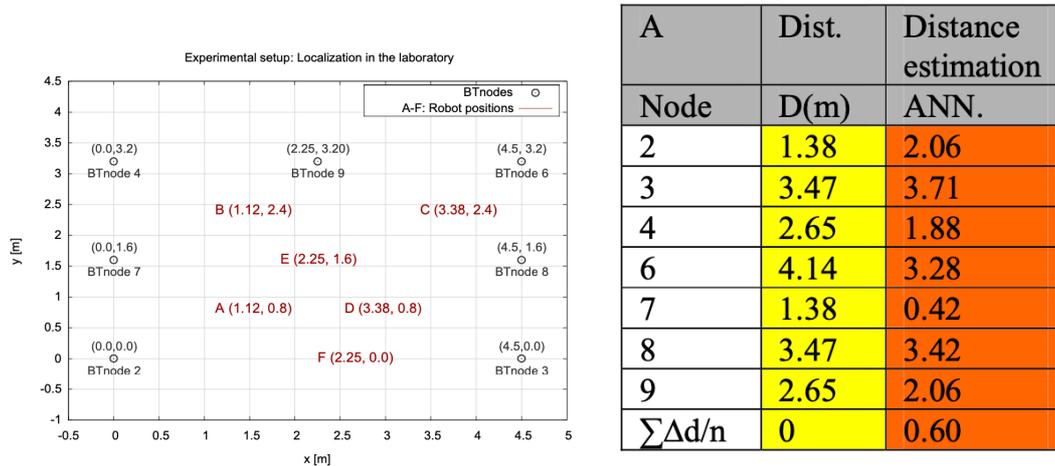


Figura 10. Ejemplo de red neuronal aplicada a la predicción de distancia en función del valor RSSI.

Extraído de [13].

Estudios muestran resultados satisfactorios a partir del uso de redes neuronales. Aunque existe error en las predicciones, este es bastante pequeño, con una media de 0.6m en el caso de la figura:



*Figura 11. Resultado de estudio de aplicación de RRNN a la predicción de distancia en función de valores RSSI junto a mapa de balizas. Extraído de [14].*

Como se dijo anteriormente, la finalidad de estos algoritmos tan complejos es la de la localización de balizas con una extrema precisión. Para nuestro aplicativo, la precisión no es un factor primordial, no es necesario conocer la ubicación exacta de la baliza, por lo que podremos permitirnos un pequeño margen de error, a favor de la funcionalidad y portabilidad, en nuestras predicciones de distancia.

### **3.4 OTROS MÉTODOS**

Otros estudios sugieren el uso de clasificadores Naive-Bayes [22], k-Nearest [18]. Mientras otros sugieren el uso de métodos más convencionales como regresiones lineales [23].

También hay estudios que proponen el filtrado mediante métodos gaussianos, combinados con filtro de Kalman para la limpieza del ruido de los datos [15].

Por último, existen estudios que proponen la combinación de algoritmos de optimización, como PSO (Particle Swarm Optimization) con redes neuronales [13] con el fin de obtener mejores resultados.

Teniendo en cuenta que existen numerosos métodos para el análisis de la señal, en nuestro experimento, trataremos de usar varios de ellos con el fin de obtener un resultado satisfactorio.

## **Capítulo 4. DEFINICIÓN DEL TRABAJO**

### **4.1 JUSTIFICACIÓN**

En el caso de uso de la aplicación será frecuente que esta tenga que discriminar entre señales de distintas balizas y aceptar aquella que esta perciba como más cercana. Este objetivo podría lograrse a partir del estudio del indicador de potencia de la señal de la baliza, el RSSI, el cual tendrá valores más altos según la cercanía del usuario a la baliza y valores más bajos en el caso contrario, debido a que la señal perderá mayor intensidad cuanto más se propague en el medio.

Sin embargo, la potencia de señal no depende única y linealmente de la distancia del receptor, sino que existen numerosas variables del entorno que no conocemos, entre ellas, las señales que puedan colisionar con la propia señal emitida por la baliza, y que, por tanto, podemos denominar ruido, que atenúan dicha potencia, y complican nuestro objetivo.

Es necesario estudiar el efecto de diversos algoritmos de predicción y filtrado con el fin de encontrar un método eficaz, capaz de determinar una distancia a partir de potencia recibida y que dicha predicción posea un escaso error.

## **4.2 OBJETIVOS**

El objetivo, finalmente, será descubrir qué algoritmo, o algoritmos, son capaces de determinar la distancia a una baliza con mayor precisión y, de manera más eficiente.

Por último, buscamos crear una función que reciba un valor RSSI y sea capaz de proporcionar un valor de distancia aproximado. Dicha función dependerá del análisis realizado en este trabajo.

## **4.3 METODOLOGÍA**

Se irán aplicando distintos algoritmos a un set de datos de mediciones RSSI tomados a distintas distancias, y se documentarán las funciones de predicción que habrán realizado los algoritmos sobre dichos datos.

Se documentarán las figuras de regresión y resultados, con el fin de comprender cómo se comportan de una manera gráfica.

Por último, se compararán las figuras obtenidas, con el fin de descubrir cuál es el algoritmo que mejor se comporta, las diferencias entre ellos, y cuánto error existe entre ellos.

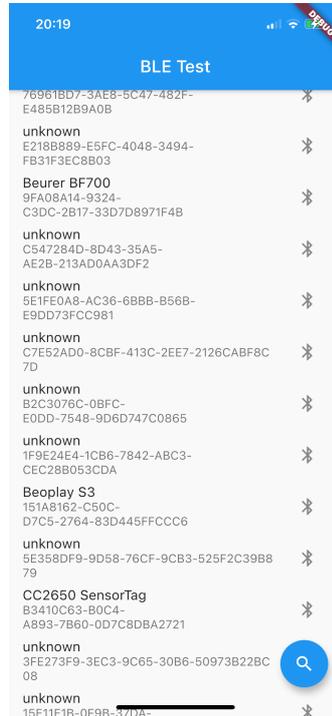
## **Capítulo 5. SISTEMA/MODELO DESARROLLADO**

En este proyecto, como se dijo en el apartado de Introducción, se realizará un análisis de la funcionalidad BLE en los dispositivos Android e iOS, para posteriormente aplicar la funcionalidad BLE en la aplicación.

### ***5.1 DESARROLLO DE APLICACIÓN MEDIDOR***

Se desarrolla una aplicación de escaneo y medición de potencia RSSI en Flutter, haciendo uso de la librería `Flutter_Ble_Lib`. Dicha aplicación será de utilidad en la obtención de los datos RSSI que se utilizarán para analizar distintos modelos predictivos de distancia.

La aplicación consta de una pantalla en la que se escuchan los mensajes de advertising en el medio y se muestran los dispositivos cercanos al usuario.



*Figura 12. Pantalla 1 de aplicación Medidor. Listado de balizas cercanas.*

Una vez listadas las balizas disponibles, puede seleccionarse el dispositivo al cual se quiera conectar, para nuestra investigación, nos conectaremos a la baliza CC2650 que será de la que recibiremos los valores RSSI a estudiar.



*Figura 13. Pantalla 2 de aplicación Medidor.*

Tras conectarnos a la baliza, podremos tomar mediciones de valores RSSI durante el tiempo que creamos conveniente, además de seleccionar la distancia a la que nos encontramos de la baliza. Estos parámetros nos servirán para tomar la medición y, una vez realizada, podremos mostrar los datos obtenidos en el terminal de Android Studio. Teniendo en cuenta que esta aplicación ha sido programada en exclusiva como apoyo para nuestra investigación, es más cómodo para nosotros que el móvil muestre los datos en consola inmediatamente.

## **5.2 RENDIMIENTO DE LA BATERÍA DEL DISPOSITIVO MÓVIL**

### **5.2.1 PARAMETRIZACIÓN DE FLUTTERBLELIB**

Para analizar el rendimiento de la batería del dispositivo, se parte de los parámetros disponibles en FlutterBleLib [29].

La función `startPeripheralScan` devolverá un `Stream` de todos los dispositivos cercanos descubiertos, con la información de advertising y de los dispositivos. Dicha información se encontrará en un objeto de la clase `ScanResult`.

```
Stream<ScanResult> startPeripheralScan({  
  int scanMode,  
  int callbackType,  
  List<String> uuids,  
  bool allowDuplicates,  
});
```

Los parámetros `scanMode` y `callbackType` son específicos de Android, mientras que `allowDuplicates` lo es de iOS

Distintos modos en los que opera un scan de BLE en Android, para el parámetro `scanMode`:

### **1. SCAN\_MODE\_BALANCED**

Los resultados del scan se devuelven a un ratio que provee de una buena compensación entre frecuencia de escaneo y consumo energético.

Valor de la constante: 1

### **2. SCAN\_MODE\_LOW\_LATENCY**

Modo de baja latencia. Escaneo usando el mayor ciclo de trabajo. Se recomienda usar este modo solo cuando la aplicación esté funcionando en primer plano.

Valor de la constante: 2

### **3. SCAN\_MODE\_LOW\_POWER**

Modo de escaneo por defecto; consume la mínima energía. Este modo tiene mayor utilidad si la aplicación de escaneo no está funcionando en primer plano.

Valor de la constante: 0

### **4. SCAN\_MODE\_OPPORTUNISTIC**

Modo especial de escaneo de BLE mediante el cual las aplicaciones escucharán pasivamente otros resultados de escaneo sin iniciar ellas mismas el scan.

Valor de la constante: -1

## **5.2.2 EXPERIMENTO: RENDIMIENTO DE DISPOSITIVOS ANDROID EN FUNCIÓN DE INTERVALOS DE ESCANEADO DE VENTANA.**

### **Consideraciones iniciales:**

Se usará la librería *battery* para automatizar la obtención de medidas. Se supondrá un caso ideal en el que el usuario no tiene otra aplicación funcionando en segundo plano.

### **Herramientas a utilizar:**

- Samsung Galaxy S10

### **Realización del experimento:**

Se medirá el rendimiento de la batería del dispositivo durante un período de 30 minutos, en el cual, el dispositivo Android estará escaneando sin descanso. Cada 30 segundos se recopilará el porcentaje de batería en un fichero .txt. El experimento se realizará cuatro veces, una para cada uno de los siguientes modos:

- Modo low-power
- Modo balanced
- Modo low-latency
- Modo opportunistic

### 5.3 DESCRIPCIÓN DE ALGORITMOS DE FILTRADO

Se definen aquí los algoritmos que se utilizarán para llevar a cabo el estudio del análisis de la distancia según el indicador RSSI de la baliza.

#### 5.3.1 FILTRO DE KALMAN

El filtro de Kalman es un algoritmo cuya finalidad es la de predecir sistemas. Es un conjunto de ecuaciones matemáticas que proveen una solución computacionalmente eficiente del método de mínimos cuadrados. Soporta estimaciones de estados pasado, presente y futuro, y puede hacerlo, a pesar de que la naturaleza precisa del sistema modelado sea desconocida [3, 20].

Se ilustra en la figura 12, cómo opera el algoritmo de Kalman.

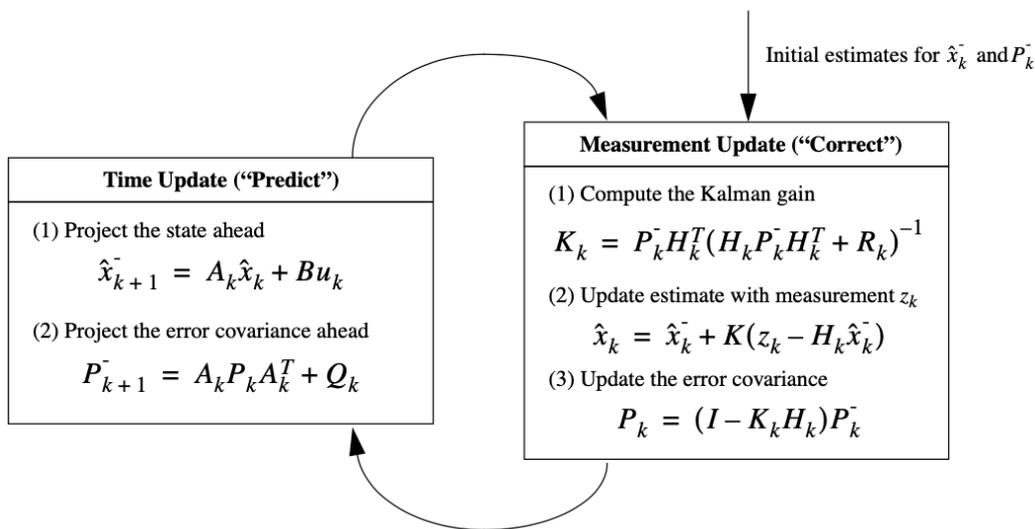


Figura 14. Algoritmo de Kalman ilustrado. Extraído de [5].

Para el experimento, se implementa un filtro de Kalman en R, por el que se procesan los datos obtenidos mediante las mediciones, con el fin de reducir el ruido de estas.

### 5.3.2 BOXPLOTS

Representar las medidas obtenidas mediante diagramas de caja nos proporciona una mejor visión sobre el problema. Estos boxplots se comparan además gráficamente con las densidades de las distribuciones de datos obtenidos mediante histograma.

Podemos entonces asumir mediante boxplots que aquel valor de RSSI obtenido que entre dentro del rango intercuartílico de una distribución, pertenecerá necesariamente a su distancia correspondiente.

Para un mayor conocimiento de nuestros datos, y para contribución a nuestro análisis, se obtienen además valores estadísticos, tales como media, moda y varianza, de todas las distribuciones.

### 5.3.3 REGRESIÓN LINEAL

El método de regresión lineal permite caracterizar el problema de forma lineal, es decir, nos permite, a partir de una variable conocida, predecir otra variable que dependa en parte de esta.

El algoritmo de regresión se define, siendo  $n$  un número entero positivo, como:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon$$

Donde  $Y$  será la variable exógena por predecir, suponiendo  $i$  como número entero en el rango  $\{1, \dots, n\}$ ,  $\beta_i$  serán los coeficientes de la función de regresión, las  $X_i$ , variables independientes y explicativas, y  $\varepsilon$  la variable ruido, la cual representa aquellas variables que influyen en el modelo pero que desconocemos [23].

Nuestro modelo, al ser únicamente dependiente de una variable (el indicador RSSI), se modelará de la siguiente forma, como un modelo de regresión lineal simple:

$$Distancia = \beta_0 + \beta_1 RSSI$$

Gráficamente, al depender de única variable, el modelo se representará como una recta de regresión de pendiente  $\beta_1$ .

Por otro lado, obtendremos el parámetro  $R^2$ , el cual se conoce como el coeficiente cuadrado de correlación de Pearson. Este se mide en tantos por ciento, tomando un valor entre 0 y 1. Un coeficiente de correlación alto implica que el modelo es capaz de explicar correctamente la variabilidad de sus datos de respuesta, es decir, a mayor  $R^2$ , mejor se ajusta el modelo utilizado a los datos.

### 5.3.4 BAYES

El algoritmo de Bayes implica obtener una función de distancia dependiente del valor RSSI teniendo en cuenta las probabilidades de que un valor pertenezca a una u otra distribución. Esto es, en última instancia, la realización de una inferencia bayesiana.

La obtención de dicho resultado tiene lugar en dos pasos:

En primer lugar, se diseña un histograma de densidad para todas las medidas obtenidas, en el cual, observaremos la curva que sigue cada una de las distribuciones, la densidad de estas a lo largo del eje RSSI, y cómo las distribuciones pueden llegar a solaparse entre sí.

Posteriormente, la función bayesiana se modela en función de la densidad que ocupa una determinada distribución en un punto en el eje de ordenadas, en comparación con otra que coincida en el mismo punto. De esta forma, una distribución de una distancia  $X$ , que coincida en un valor RSSI con una distribución de distancia  $Y$ .

Si una distribución en un punto no coincide con ninguna otra, suponemos que la probabilidad de que dicho valor RSSI coincida con la distancia esperada de dicha distribución es del 100%, tomando entonces su valor correspondiente de distancia.

Finalmente, el resultado obtenido se muestra en una gráfica.

Como para el experimento nuestras mediciones a mínima y máxima distancia se realizan a 3 y 11 metros respectivamente, la función bayesiana tomará dichos valores para medidas fuera del rango de medición RSSI.

### 5.3.5 RRNN

Por último, se utiliza un modelo de clasificación de redes neuronales con el fin de saber determinar la procedencia de un valor RSSI dentro de las distancias de los modelos con los que ha sido entrenado. De esta forma, el valor RSSI será introducido a través de distintas redes neuronales, las cuales serán capaces de determinar si dicho valor pertenece o no a la distribución, devolviendo un resultado binario. El uso del algoritmo de redes neuronales se ha realizado de esta manera debido a que solo tomaremos medidas desde 5 puntos de distancia para obtener el mayor número de datos posible desde cada punto, y por la sencillez que supone implementar datos de esta forma para el resto de los algoritmos.

Finalmente, obtendremos una figura en la que se verán clasificados cada uno de los valores de potencia según su distancia, tras pasar por el algoritmo de RRNN.

## **5.4 PROPUESTA DE COMBINACIÓN DE LOS ALGORITMOS**

En busca de una mayor precisión a la hora de predecir la distancia en función del indicador de potencia de las balizas, se propone combinar distintos algoritmos. Este acercamiento al problema nos permitirá comprobar de primera mano, la funcionalidad de estos algoritmos. Por ello, el procedimiento a realizar en el experimento es el siguiente.

En primer lugar, se toma un set de datos de mediciones, el cual contiene distintos valores RSSI de potencia de la baliza, asociados a la distancia en la que se han obtenido. A continuación, se pasan dichos datos a través de los métodos explicados en el punto anterior, comenzando con diagramas de caja, siguiendo con algoritmo de regresión lineal, Bayes, y, por último, redes neuronales.

Se realiza una comparativa de los resultados obtenidos y se procede a pasar los datos iniciales a partir de un filtro Kalman en búsqueda de reducir el ruido en la señal. Tras esto, se realiza de nuevo el procedimiento anterior, y se analizan los nuevos resultados.

Para finalizar, se realiza una comparativa de los resultados pre-filtrado y tras pasar por el filtro Kalman.

A partir de los resultados obtenidos, se tomarán conclusiones sobre qué algoritmo sea mejor, o si, finalmente, sea mejor combinar distintos algoritmos para obtener una mayor precisión en nuestras predicciones.

## Capítulo 6. ANÁLISIS DE RESULTADOS

### 6.1 ANÁLISIS DE RENDIMIENTO DE BATERÍA

Se muestra a continuación el rendimiento de la batería tras realizar escaneos con la aplicación Medidor durante media hora, en un móvil Android, en función de los distintos modos de parametrización

#### 6.1.1 RENDIMIENTO DE LA BATERÍA DEL MÓVIL SIN REALIZAR SCANS

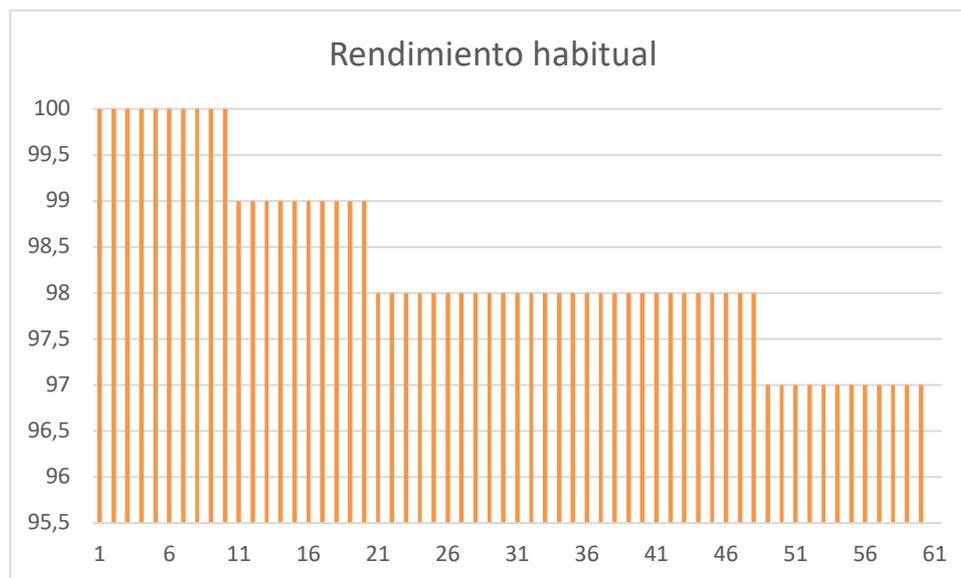


Figura 15. Descargue de batería de Samsung Galaxy S10 sin realizar scans.

Se muestra aquí el descargue de la batería de un Samsung Galaxy S10 con la aplicación Medidor abierta, sin realizar ningún scan, durante 30 minutos. Cada uno de los valores en el eje X, muestra cada medición tomada, las cuales se realizaron a razón de una cada 30 segundos.

Como puede verse, el descargue de la batería no fue muy alto como cabría esperar.

## 6.1.2 MODO LOW-POWER

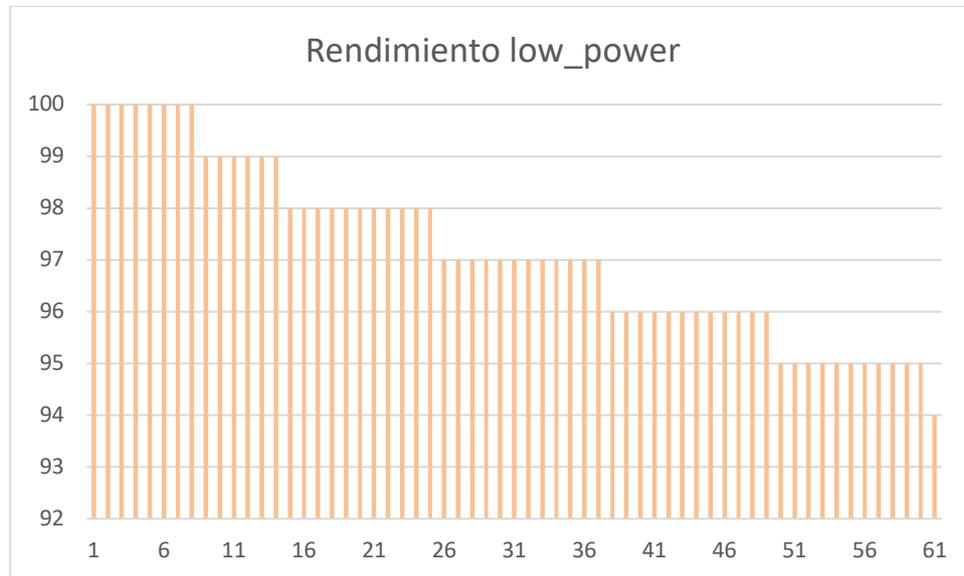


Figura 16. Descargue de batería de Samsung Galaxy S10 escaneando en modo low-power.

Se muestra el resultado tras activar el modo de escaneo low power. Como puede verse en la figura, el descargue de la batería es algo superior a cuando no se realizaba ningún escaneo, como cabría esperar, sin embargo, la diferencia no es muy alta, tan solo de un 2% de diferencia.

A simple vista, se obtuvieron resultados de forma rápida. Aparecieron bastantes dispositivos disponibles, por lo que la opción low power parece la más adecuada.

### 6.1.3 MODO BALANCED

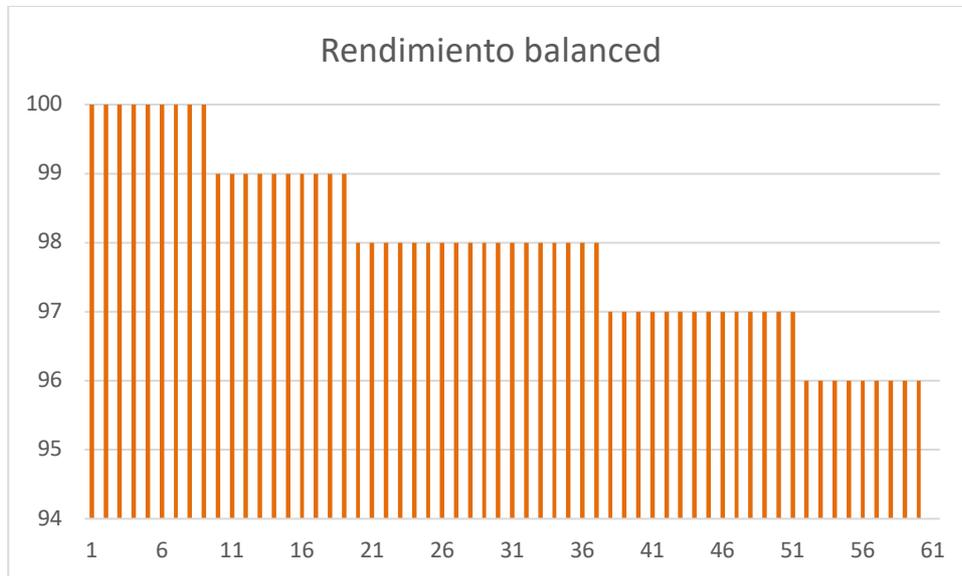


Figura 17. Descargue de batería de Samsung Galaxy S10 escaneando en modo balanced..

El modo balanced ofrece una ventana de escaneo mayor que la del modo low-power, sin embargo, el rendimiento del modo low-power es bastante similar, hasta tal punto que podría parecer que en este caso incluso, el modo balanced proporciona ligeramente un mayor ahorro de energía.

### 6.1.4 MODO LOW-LATENCY

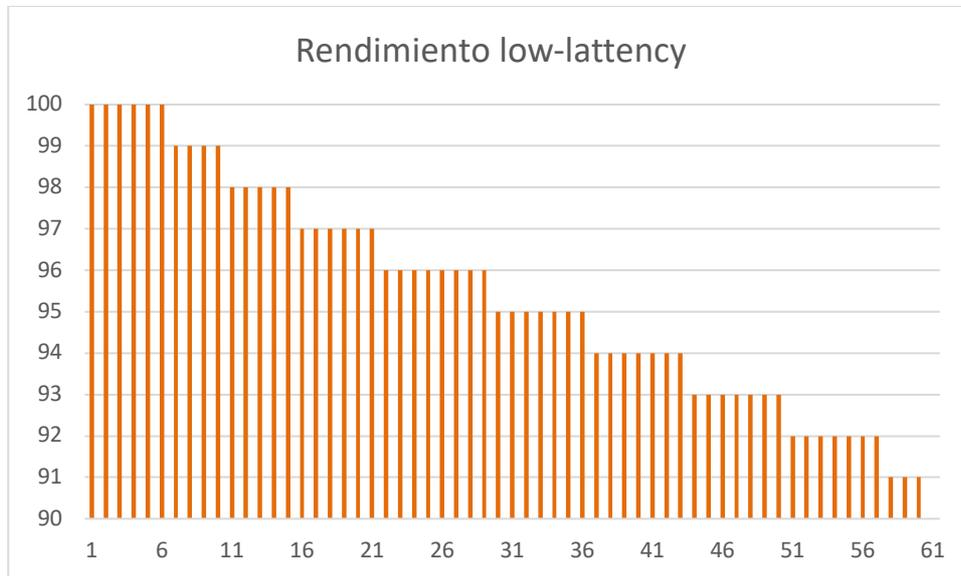


Figura 18. Descargue de batería de Samsung Galaxy S10 escaneando en modo low-lattency.

El modo low-lattency parece ser el más exigente con la batería del dispositivo, haciendo descender nivel de batería hasta un 91% en media hora. Realmente, no es necesario una ventana de escaneo tan amplia como la que nos brinda este modo, pues el usuario final no apreciará que una baliza se detecte a unos milisegundos más rápido que mediante el resto de los modos.

### 6.1.5 MODO OPPORTUNISTIC

No se ha realizado un análisis con este modo pues, como ha demostrado y como indica la documentación, la aplicación no iniciará el escaneo mediante este modo.

## **6.2 ANÁLISIS DE DATOS DE DISTANCIA LIMPIOS**

Se tomaron medidas de los valores del indicador de potencia RSSI de una baliza, desde distintas distancias, en una habitación de 12 metros de largo. La distancia más cercana a la baliza en dichas mediciones se encuentra a 3 metros, y cada medición se realizaría con una diferencia de 2 metros respecto a la anterior, siendo la más lejana a la baliza de 11 metros.

Dichas medidas se realizaron en intervalos de un minuto, obteniendo un vector de 58 medidas por cada distancia. Una vez recopilados los datos, se importaron en formato .csv a RStudio para su análisis.

## Histograma

Comenzamos observando la distribución de la potencia para cada una de las distancias en un diagrama de densidad:

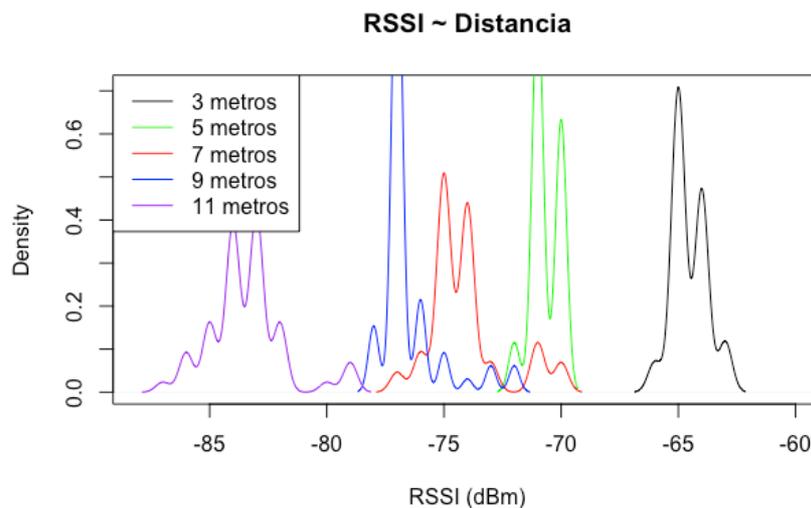


Figura 19. Distribución de valores RSSI para medidas en entorno controlado durante un minuto.

Vemos en la figura que, como cabría esperar, la potencia para distancias cercanas a la baliza será mayor que la que correspondería a las más lejanas. Puede verse también que, para la distancia de 3 metros, la distribución se encuentra bien diferenciada, de las demás distancias, sin embargo, la distribución de 5 metros se solapa ligeramente con la de 7 y 9 metros, y lo mismo ocurre con dichas distribuciones, que llegan a solaparse entre sí.

## Boxplots:

Comenzamos analizando el problema desde un punto de vista estadístico, representando las medidas como boxplots. Puede verse que las distribuciones de 3 y 11 metros poseen ambas, valores RSSI que no colisionan con las otras medidas. Sin embargo, los boxplots correspondientes a las medidas de 5, 7 y 9 metros colisionan entre sí. Esto, coincide con la Figura 13, donde son dichas distribuciones las que colisionan entre sí y coinciden en determinados valores del eje RSSI.

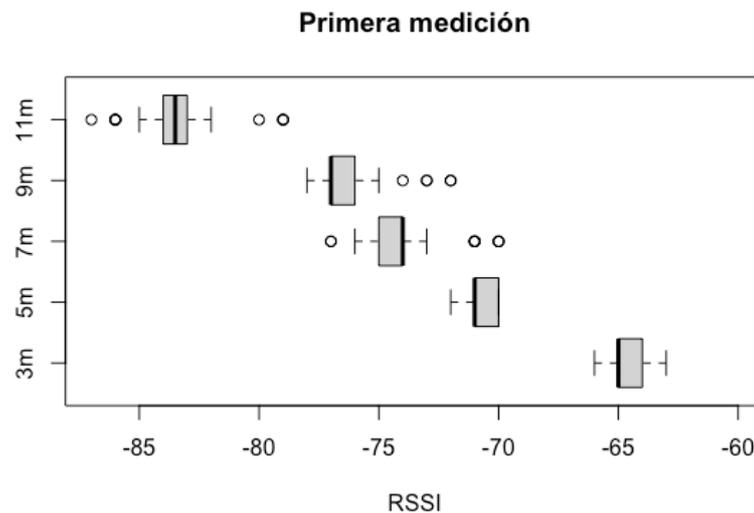


Figura 20. Diagrama de cajas de medidas en entorno controlado durante un minuto.

No podemos determinar la distancia correspondiente a un RSSI a partir de este modelo, sin embargo, obtenemos información interesante. Gracias a este modelo, sabemos que, si uniéramos cada boxplot con una recta por su valor medio obtendríamos una figura dependiente del RSSI, la cual, a menor valor RSSI menor valor de distancia tendrá asignado.

De cada distribución obtenemos las siguientes variables estadísticas:

---

```
> summary(X3m)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-66.00 -65.00  -65.00  -64.53 -64.00  -63.00
> summary(X5m)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-72.00 -71.00  -71.00  -70.69 -70.00  -70.00
> summary(X7m)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -77.0  -75.0   -74.0   -74.1  -74.0   -70.0
> summary(X9m)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-78.00 -77.00  -77.00  -76.50 -76.25  -72.00
> summary(X11m)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-87.00 -84.00  -83.50  -83.43 -83.00  -79.00
```

*Figura 21. Medidas estadísticas de medidas en entorno controlado durante un minuto.*

## Regresión lineal:

Se tratan las medidas bajo un modelo de regresión lineal:

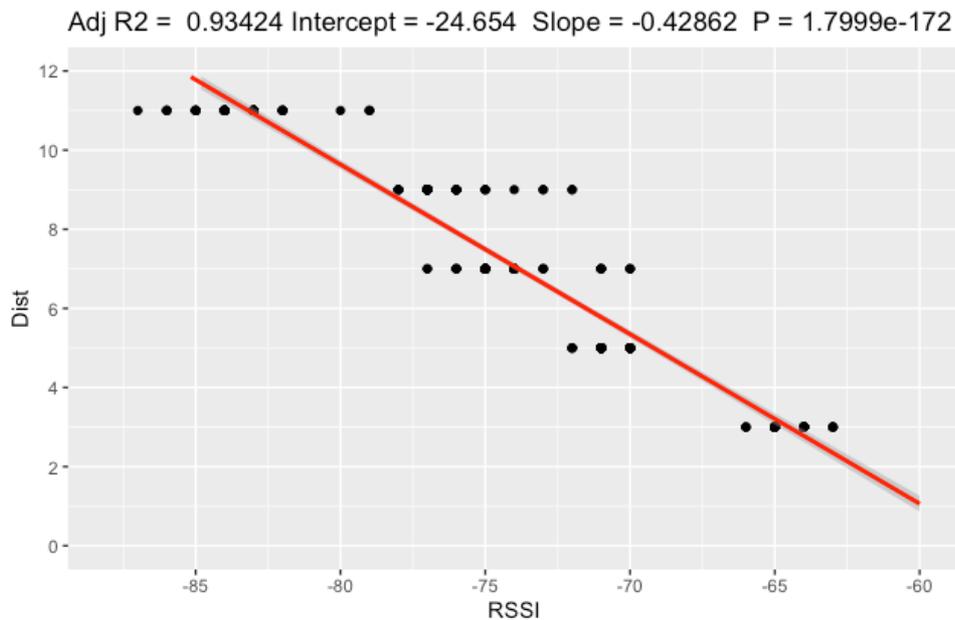


Figura 22. Modelo de regresión lineal para medidas en entorno controlado durante un minuto.

La fórmula de dicha regresión para obtener la distancia a partir de un RSSI:

$$Distancia(RSSI) = -24.6544 - 0.4286 * RSSI$$

El p-valor es prácticamente 0, por lo que podemos concluir que existe significación en la variable RSSI, por lo que es un indicador útil para predecir la distancia, como ya suponíamos de partida. Además, el indicador  $R^2$ , el cual nos indica el grado de correlación entre ambas variables, se muestra con un valor de 0.93, por lo que podemos concluir que el modelo se encuentra bien ajustado a los datos. Esto no sorprende en parte debido a que las medidas no tienen una muy alta variabilidad, ya que se han tomado en un entorno controlado, evitando el posible efecto del ruido.

Sin embargo, como ya sabemos gracias a la figura de la distribución de densidad, la distancia no depende linealmente de la potencia debido al ruido (bajo) existente en el medio, por lo que el problema no puede reducirse a una recta.

Si tomamos como indudable este modelo, podemos llegar al fallo que ocurre con la medida de  $-77\text{dBm}$  de la distribución de 7 metros. La función de regresión supone que una medida de esa potencia pertenecerá a una distancia de aproximadamente 9 metros, cuando realmente, se ha obtenido a una distancia de 7.

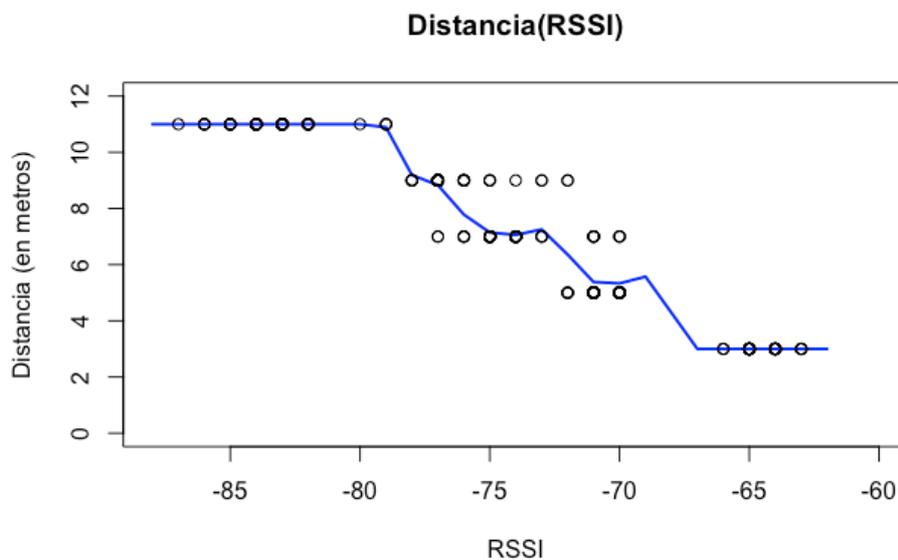
### **Cálculo del error**

Para calcular el error se emplea el método del error cuadrático medio, donde enfrentamos la medida esperada de distancia por cada valor RSSI de cada distribución con el algoritmo de regresión lineal

Se obtiene, a partir de los datos introducidos en la regresión lineal, y tras aplicar la fórmula del error cuadrático medio a las mediciones, un valor de error de 0.5242516 metros.

**Bayes:**

Utilizando el algoritmo de Bayes para las medidas sin ruido, se obtiene la siguiente figura:



*Figura 23. Relación Distancia-RSSI a partir de método bayesiano para medidas en entorno controlado durante un minuto, comparada con datos puntuales obtenidos de las distribuciones*

Como cabría esperar, las medidas obtenidas a 3 y 11 metros, al no solaparse con ninguna otra distribución, mantendrán el valor de la distancia en sus respectivos valores esperados.

Por otro lado, observamos la forma escalonada de la gráfica resultante del algoritmo bayesiano. Su forma se debe a la escasa variación y el poco solape de los datos introducidos, donde habrá una mayor probabilidad de encontrar datos exclusivos a una distribución.

### **Cálculo del error**

Aplicando el método de cálculo del error cuadrático medio de los datos medidos para el algoritmo bayesiano se obtiene un valor de: 0.5329002 metros, muy cercano al obtenido mediante la regresión lineal.

## RRNN:

Por último, se pasaron las medidas a través de redes neuronales. Se entrenó una red neuronal para cada distribución que fuera capaz de determinar si un RSSI determinado pertenecía o no a dicha distribución.

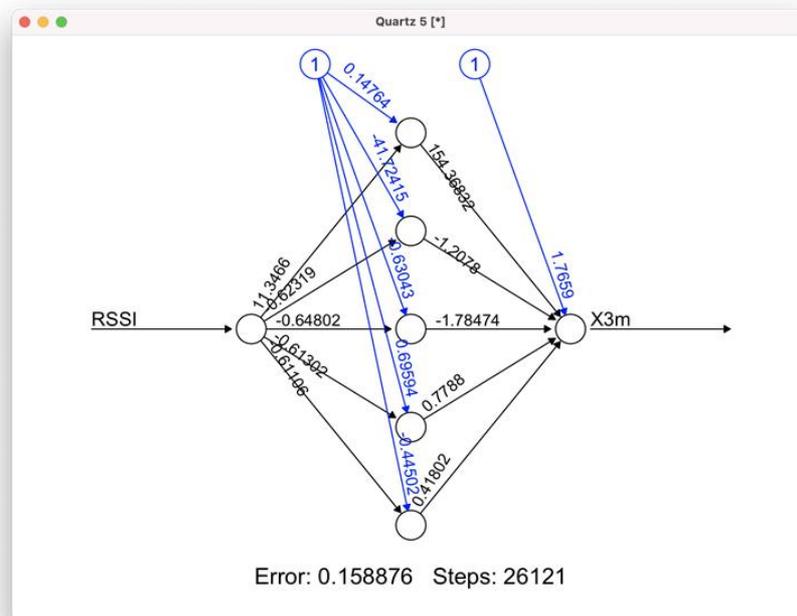


Figura 24. Red neuronal para clasificar si un dato pertenece o no a la distribución de 3 metros entrenada con medidas tomadas en entorno sin ruido durante un minuto.

A pesar de que, finalmente, los datos entregados, son satisfactorios, es decir, es capaz de clasificar. Este método demostró ser el más costoso computacionalmente de todos, llevando bastante más tiempo que el resto de los métodos para entrenar la red.

### Resultado final:

Uniando los resultados obtenidos mediante la regresión lineal, los diagramas de Bayes y las redes neuronales, se obtiene la siguiente figura:

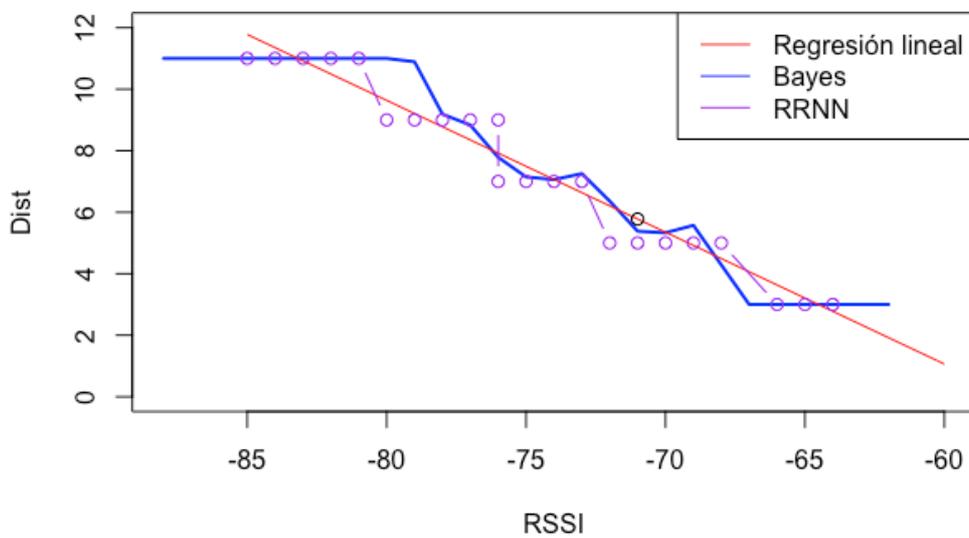


Figura 25. Comparativa de resultados tras aplicar Regresión lineal, Bayes y RRNN sobre medidas en entorno controlado durante un minuto

Como se demostró en el primer acercamiento a los algoritmos mediante los diagramas de caja, la función que relacionaría la potencia de la señal tendría una forma decreciente, y todos los algoritmos de predicción aplicados después, cumplen con dicha demostración.

## Comparativa de figuras de datos limpios

### Distribución vs. Resultado final

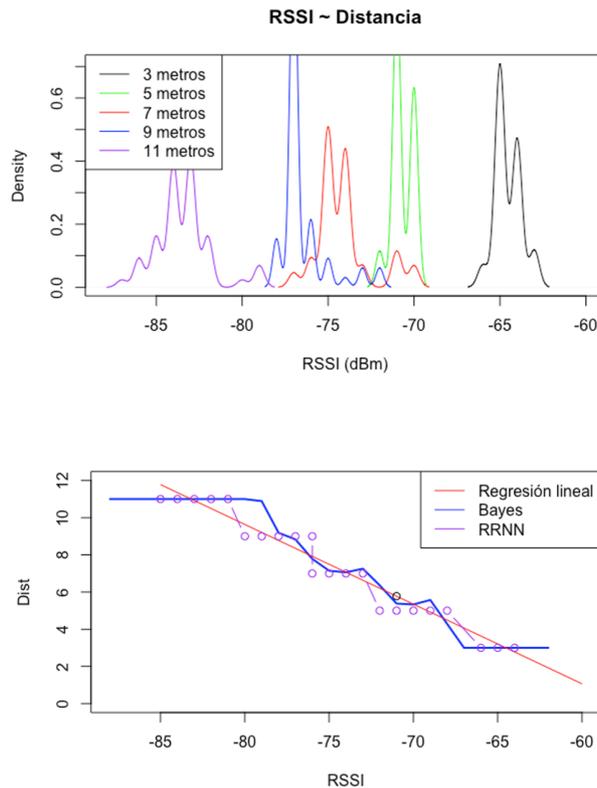


Figura 26. Comparativa entre salida de función distancia e histograma inicial para medidas tomadas en entorno controlado durante un minuto

Comparando la función de distancia con las distribuciones iniciales, puede comprobarse cómo se encuentran relacionadas, en especial, con la función de regresión de Bayes, cuya forma depende directamente de la densidad de la distribución. Así, puede verse que en los valores en los que la función de Bayes genera una recta paralela al eje X, en la figura superior, no existe ninguna distribución que colisione con otra, y que generase, por tanto, un valor dependiente de la densidad de ambas distribuciones en dicho punto.

La recta de regresión de Bayes parece ajustarse bien al modelo original, salvo en aquellos puntos en los que, por falta de medidas, existe un vacío en las distribuciones y la función de regresión deba ajustarse a los datos disponibles, dando lugar a, en el caso entre 3 y 5 metros, rectas muy pronunciadas a raíz de la escasez de información, o en el caso entre las distribuciones de 9 y 11 metros, a que la recta de regresión tome una pendiente 0 a pesar de la densidad de distribución en esa zona es muy baja, debido a que no existe en dicho punto otro dato disponible perteneciente a otra distribución.

## 6.2.1 FILTRADO DE KALMAN DE DATOS LIMPIOS

Se procederá ahora a pasar las medidas sin ruido mediante un filtro de Kalman para así suavizar datos atípicos provocados por el ruido, y se realizará el estudio previo sobre estos nuevos datos.

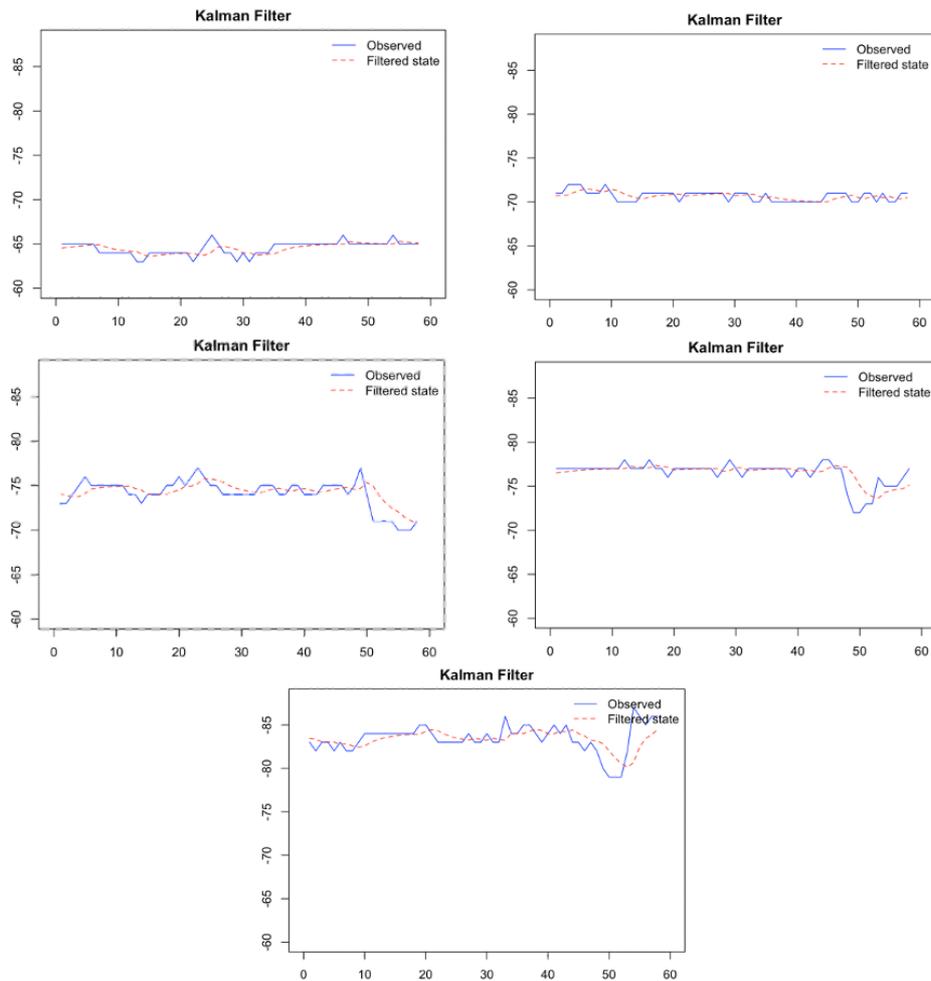


Figura 27. Filtrado de la señal limpia a través del filtro de Kalman. Se muestra en la figura el filtrado de cada distribución. 3 metros: arriba izquierda, 5 metros arriba derecha, 7 metros: centro izquierda, 9 metros: centro derecha, 11 metros: abajo izquierda.

Puede verse como, tras pasar por el filtro de Kalman, en las 5 mediciones, los datos son suavizados y no toman valores extremos fruto del ruido. Si bien es cierto que la diferencia no es muy apreciable, ya que las medidas obtenidas en un primer lugar se tomaron en un caso con pocas perturbaciones en el ambiente.

### Distribución tras pasar por filtro Kalman:

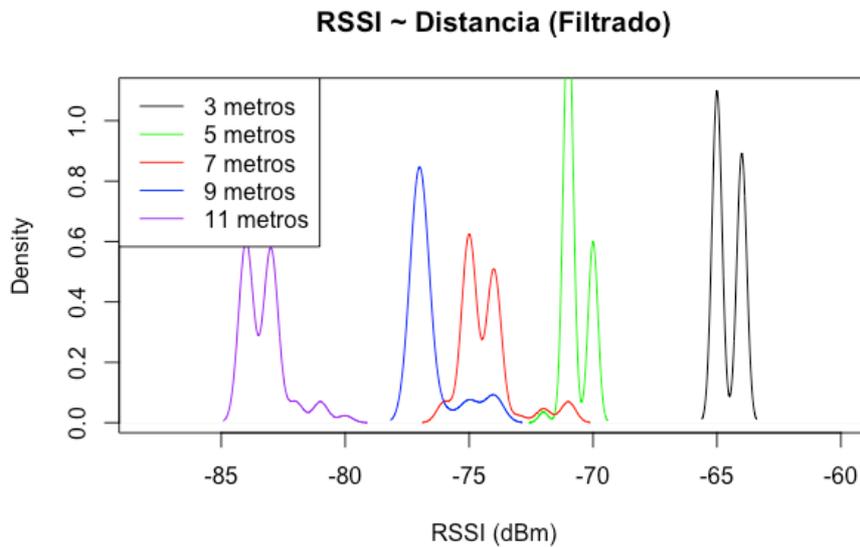


Figura 28. Distribución de valores RSSI para medidas en entorno controlado durante un minuto tras filtro Kalman.

En el histograma puede comprobarse el efecto del filtro Kalman, el cual separa las distribuciones, reduciendo la varianza de estas, y, en consecuencia, aumentando la densidad de datos en ciertos puntos. Sin embargo, elimina algunos rangos de valores RSSI en el proceso.

## Diagramas de caja tras filtro Kalman

Pasando las medidas tras el filtro a un diagrama de cajas se obtiene la siguiente figura:

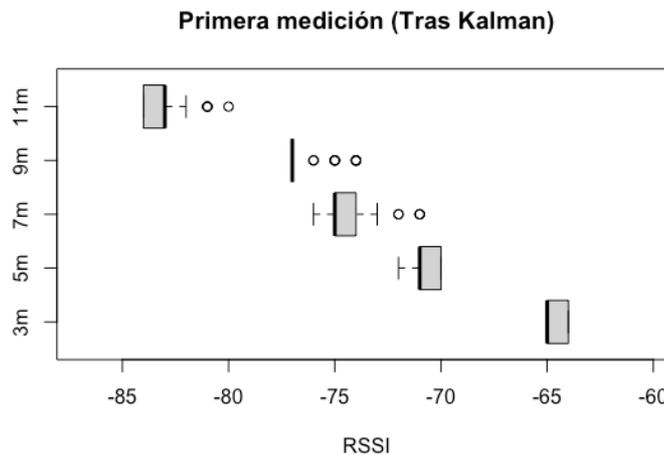


Figura 29. Diagrama de cajas de medidas en entorno controlado durante un minuto tras pasar por filtro Kalman.

Como podía verse en el histograma, los diagramas de caja reducen su varianza, y se muestran ligeramente más separados entre sí.

3m	5m	7m	9m	11m
Min. :-65.00	Min. :-72.00	Min. :-76.00	Min. :-77.00	Min. :-84.00
1st Qu. :-65.00	1st Qu. :-71.00	1st Qu. :-75.00	1st Qu. :-77.00	1st Qu. :-84.00
Median :-65.00	Median :-71.00	Median :-75.00	Median :-77.00	Median :-83.00
Mean :-64.55	Mean :-70.72	Mean :-74.33	Mean :-76.57	Mean :-83.24
3rd Qu. :-64.00	3rd Qu. :-70.00	3rd Qu. :-74.00	3rd Qu. :-77.00	3rd Qu. :-83.00
Max. :-64.00	Max. :-70.00	Max. :-71.00	Max. :-74.00	Max. :-80.00

Figura 30. Resultados estadísticos de los datos limpios obtenidos tras pasar por filtro Kalman.

La reducción de la varianza puede también comprobarse en la figura, en la que el rango entre el mínimo y el máximo para cada distribución se ve más acotado en comparación al momento previo al filtrado.

## Regresión lineal

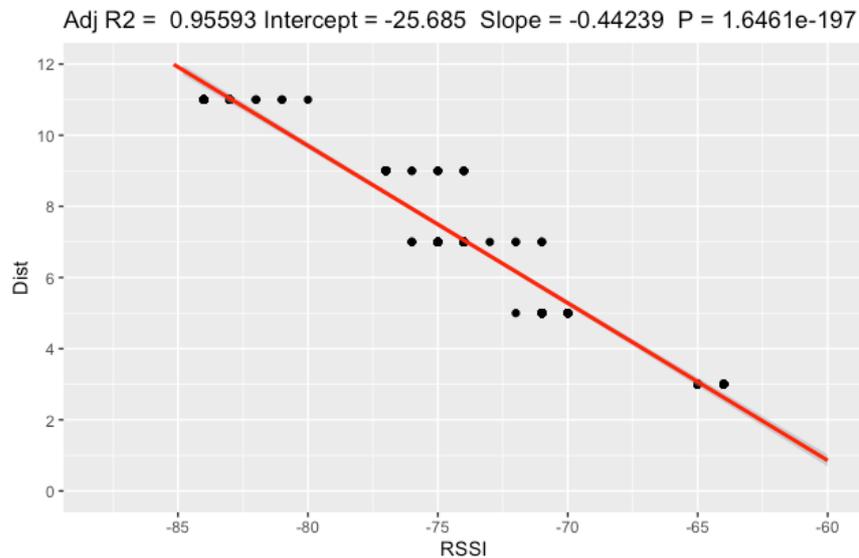


Figura 31. Modelo de regresión lineal para medidas en entorno controlado durante un minuto tras pasar por filtro Kalman.

La función de regresión lineal obtenida tras pasar por el filtro es la siguiente:

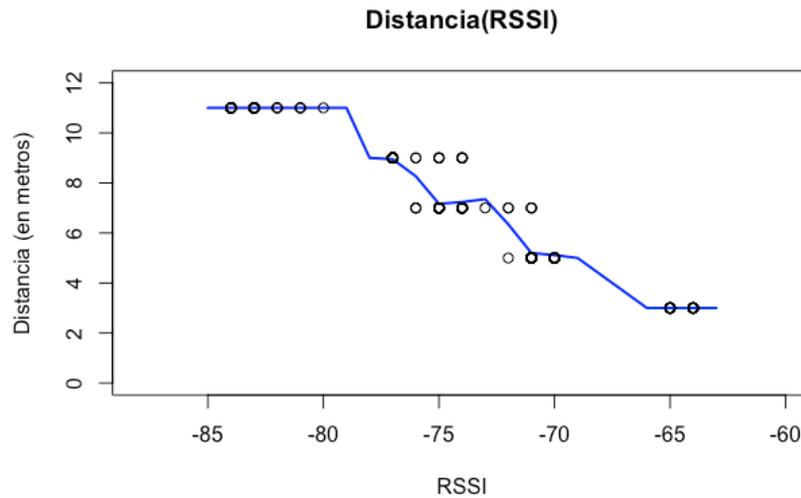
$$\text{Distancia}(RSSI) = -25.685 - 0.44239 * RSSI$$

El parámetro p-valor de la distribución es prácticamente 0, al igual que en el caso previo al paso por el filtro, por lo que el RSSI sigue siendo variable significativa. Puede apreciarse una mejoría en el valor del parámetro R2, siendo ahora de 0.95, cuando en el anterior caso era de 0.92, esto se debe a que el filtro elimina aquello que considera como ruido, resultando en que los parámetros RSSI se ajustarán mejor a la recta de regresión y, por tanto, nuestro modelo será capaz de explicarlos mejor.

Aplicando el error cuadrático medio al nuevo caso tras el filtrado de Kalman, se obtiene el valor de: 0.3513124 metros. Un valor inferior al obtenido previo filtrado.

## Bayes

De nuevo, se aplica el algoritmo de Bayes sobre los datos filtrados:



*Figura 32. Relación Distancia-RSSI a partir de método bayesiano para medidas en entorno controlado durante un minuto tras pasar por filtro Kalman.*

En el caso del algoritmo bayesiano tras su paso por el filtro de Kalman, se obtiene el siguiente valor para el error cuadrático medio: 0.4226761 metros

## **RRNN**

El entrenamiento de la red neuronal fue en este caso más costoso que en el caso anterior y dio resultados menos satisfactorios, pues, sobre todo para el caso de la distribución a 9 metros, se perdieron bastantes valores de entrenamiento, y puede comprobarse una relación en los diagramas de caja, donde el rango intercuartílico de la distribución a 9 metros consta de tan solo un valor, por lo que todos los datos parecen haberse concentrado únicamente en dicho punto.

**Función distancia tras pasar por filtro:**

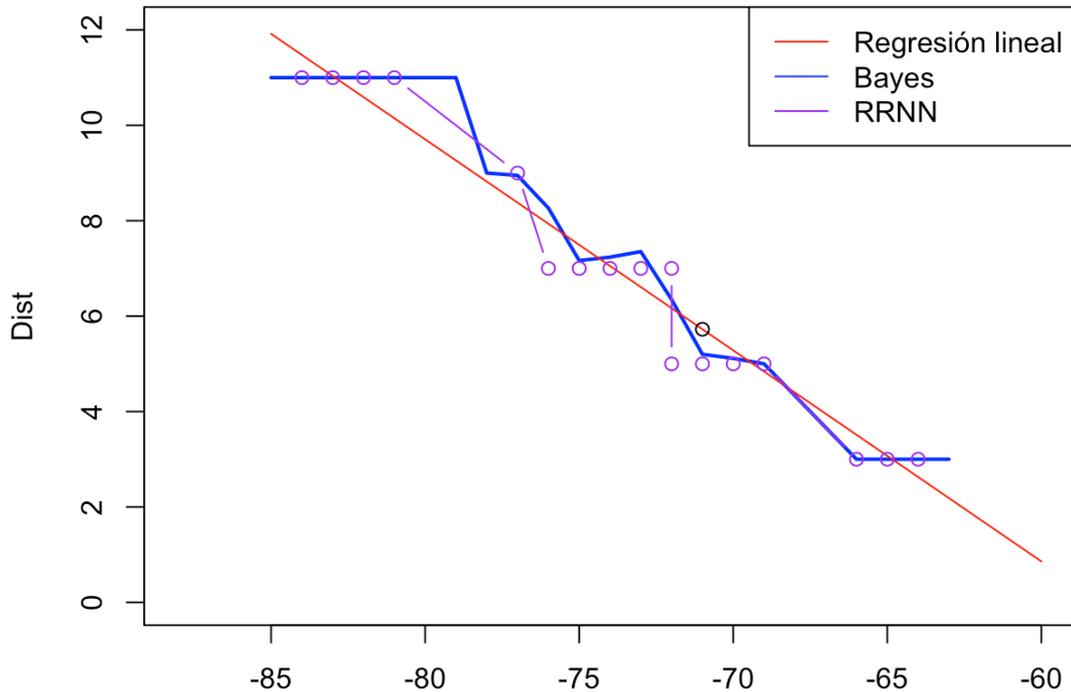


Figura 33. Comparativa de resultados tras aplicar Regresión lineal, Bayes y RRNN sobre medidas en entorno controlado durante un minuto tras filtro Kalman.

Se muestra en la figura el resultado tras aplicar todos los algoritmos sobre los datos filtrados.

Puede verse como al pasar por el filtro Kalman, la red neuronal pierde datos de entrenamiento para valores en el rango de 9 metros, quedando solo un punto para tal caso.

### **Error cuadrático medio**

Se muestra a continuación una tabla comparativa de rangos de error entre regresión lineal y Bayes:

	<b>Entorno sin ruido + Kalman</b>
<b>Regresión lineal</b>	<b>0.3513124</b>
<b>Bayes</b>	0.4226761

La regresión lineal tras el filtrado de Kalman demuestra ser el algoritmo que funciona mejor en este caso, frente al algoritmo de Bayes, con un error de 0.35 metros, menor que el obtenido por el algoritmo bayesiano.

### Comparativa pre filtro - tras filtro

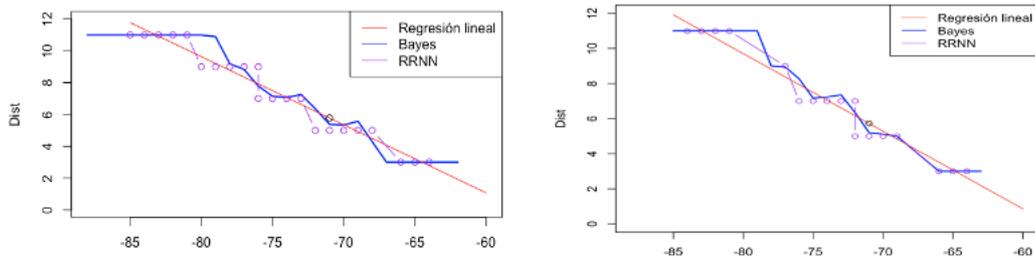


Figura 34. Comparativa de algoritmos sobre medidas realizadas en entorno limpio durante un minuto.

Ambas figuras poseen pendientes de regresión lineal muy similares y, a simple vista, es complicado determinar si existe alguna diferencia entre ambas.

La red neuronal funciona mejor en el caso previo al filtro, pues al tratar los datos mediante el filtro, perdemos valores RSSI en el proceso, resultando en que la red neuronal no podrá ser entrenada por esos valores, por lo que la función de cálculo de distancia no será capaz de proveer un valor para dichos valores.

## Comparativa de error

Se compara ahora el error cuadrático medio entre datos sin tratar y filtrados, tras pasar por los algoritmos de regresión lineal y el algoritmo de inferencia bayesiana.

Tabla 2. Comparativa de error cuadrático de algoritmo bayesiano y de regresión lineal en entorno sin ruido.

	<b>Entorno sin ruido</b>	<b>Entorno sin ruido + Kalman</b>
<b>Regresión lineal</b>	<b>0.5242516</b>	<b>0.3513124</b>
<b>Bayes</b>	0.5329002	0.4226761

Existe una ligera mejora en la predicción al utilizar el filtrado de Kalman. El error se reduce con respecto a la señal sin filtrar para ambos algoritmos. Parece que la diferencia tras aplicar el filtro de Kalman es mucho mayor para el algoritmo de regresión lineal que para el algoritmo de inferencia bayesiana. Para ambos casos, el caso pre-filtrado y post-filtrado, el algoritmo con menor cantidad de error ha demostrado ser el algoritmo de regresión lineal, ganando por bastante, un 0.07, al algoritmo bayesiano tras el filtro de Kalman

Por otro lado, se pierde precisión en la red neuronal al aplicar el filtro Kalman, ya que perdemos datos con los que entrenar a la red, sin embargo, para los datos disponibles, sigue prediciendo correctamente a qué distribución pertenece cada valor RSSI.

La gráfica de Bayes parece ajustarse mejor, gráficamente, a los valores de predicción de la red neuronal tras el filtrado.

## **Conclusiones**

Los datos al ser pasados por un filtro Kalman reducen su rango de valores de potencia RSSI, evitando valores extremos, y aumentando su amplitud como puede verse en el histograma.

La regresión lineal sigue siendo un método que induce a error, ya que como puede verse en la gráfica de regresión, datos introducidos correspondientes a una distribución, pueden perfectamente ser confundidos con la distribución de la distancia más próxima.

La gráfica de la función distancia la función de Bayes posee cambios más bruscos entre distancias, ya que ahora las distribuciones se solapan entre sí menos que con los datos sin filtrar.

La red neuronal tras el filtrado deja de clasificar en aquellos valores de RSSI eliminados por el filtro, puesto que la red no ha podido ser entrenada en ese rango, ya que tras el filtrado se han perdido aquellos valores de menor probabilidad.

### 6.3 ANÁLISIS DE DATOS DE DISTANCIA CON RUIDO

Se utilizan ahora datos más cercanos al de un caso real, en el que un usuario se encontrará en el momento del escaneo en una posición. Estos datos están muy influenciados por el ruido y, por lo tanto, su tratamiento tendrá más complicaciones. Se vuelve a realizar entonces el mismo análisis estadístico que en el caso anterior.

#### Distribución

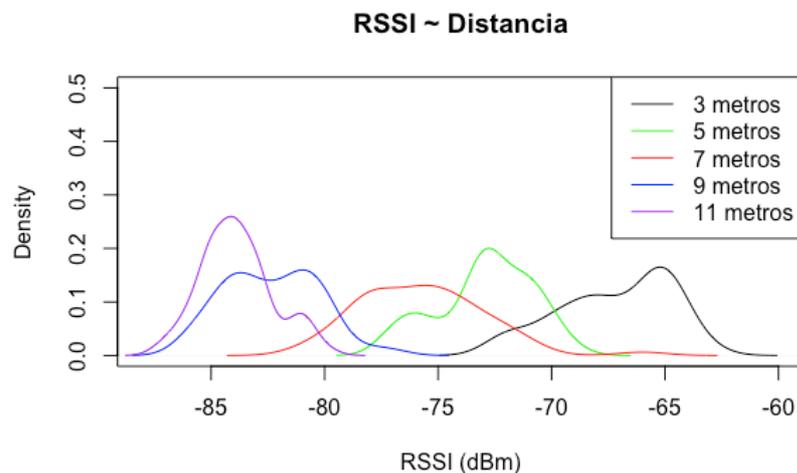


Figura 35. Distribución de valores RSSI para medidas en entorno con ruido durante un minuto.

En este caso particular, los valores RSSI se solapan mucho más entre sí, siendo más complicado diferenciar a simple vista a qué distribución pertenecería cada valor RSSI. Tal vez la única distribución que pudiera diferenciarse más de las demás es aquella a 3 metros, pues a tan baja distancia la señal está muy poco influenciada por el ruido del medio. A mayores distancias puede verse como la situación se complica, llegando a solaparse las demás distribuciones en mucha mayor medida.

## Diagramas de caja

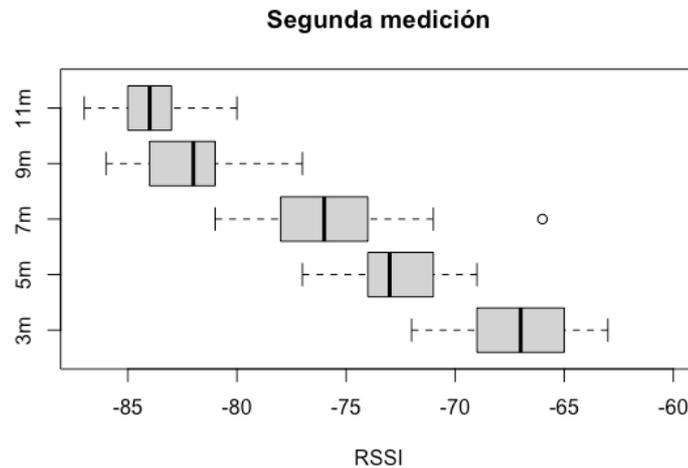


Figura 36. Diagrama de cajas para medidas en entorno con ruido durante un minuto.

Con los diagramas de caja ocurre lo mismo que con la figura del histograma, los rangos solapan entre sí, por lo que se complica determinar a simple vista a qué distancia pertenecería un valor RSSI.

	3m	5m	7m	9m	11m
Min.	:-72.00	Min. :-77.00	Min. :-81.00	Min. :-86.00	Min. :-87.00
1st Qu.	:-69.00	1st Qu.:-74.00	1st Qu.:-78.00	1st Qu.:-84.00	1st Qu.:-85.00
Median	:-67.00	Median :-73.00	Median :-76.00	Median :-82.00	Median :-84.00
Mean	:-67.19	Mean :-72.91	Mean :-75.72	Mean :-82.33	Mean :-83.86
3rd Qu.	:-65.00	3rd Qu.:-71.00	3rd Qu.:-74.00	3rd Qu.:-81.00	3rd Qu.:-83.00
Max.	:-63.00	Max. :-69.00	Max. :-66.00	Max. :-77.00	Max. :-80.00

Figura 37. Medidas estadísticas para medidas en entorno con ruido durante un minuto

Vemos mediante las medidas estadísticas de los diagramas de caja, que el rango de dichos diagramas ha ampliado mucho su varianza en comparación con las primeras medidas, culpa en gran parte del ruido del medio que ensucia la señal.

## Regresión lineal

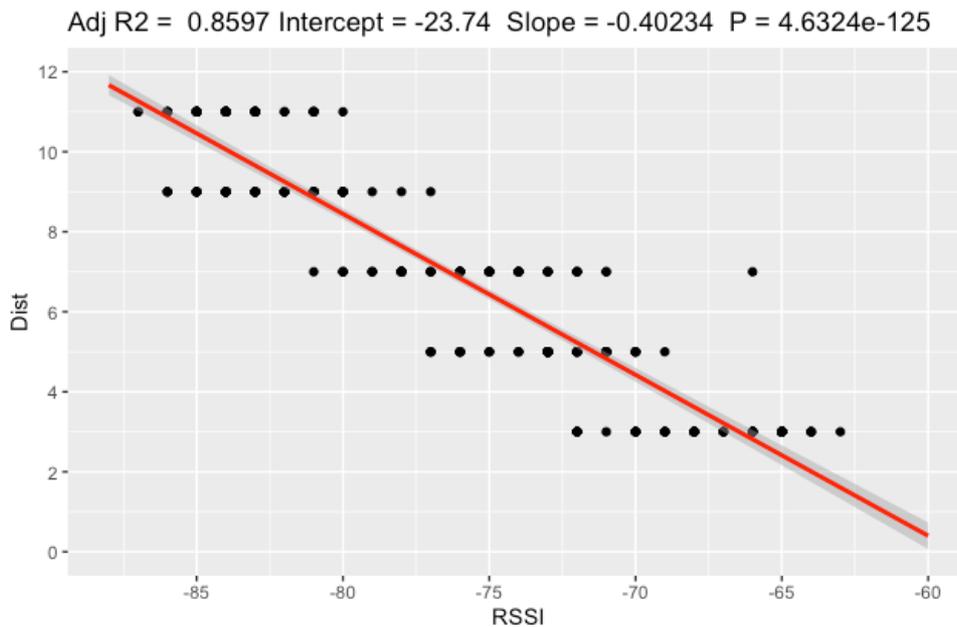


Figura 38. Modelo de regresión lineal para medidas en entorno con ruido durante un minuto.

La función de regresión lineal obtenida tras pasar por el filtro es la siguiente:

$$\text{Distancia}(RSSI) = -23.74 - 0.40234 * RSSI$$

El parámetro p-valor de la distribución es prácticamente 0, demostrando de nuevo que la variable RSSI es significativa en el modelo.

El  $R^2$  tiene un valor de aproximadamente 0.86, un valor un 7% menor que en el caso en el que teníamos datos limpios y un 10% menor en comparación con el caso en el que pasamos los datos limpios por un filtro de Kalman. Este resultado tiene sentido, pues el ruido existente en el modelo hace que le sea más complicado a este explicar la variabilidad de los resultados a partir de tan solo el indicador RSSI.

De nuevo, aplicamos el algoritmo del error cuadrático medio a la regresión lineal, esta vez con los datos en un entorno ruidoso. Obtenemos el valor: 1.118551 metros.

## Bayes

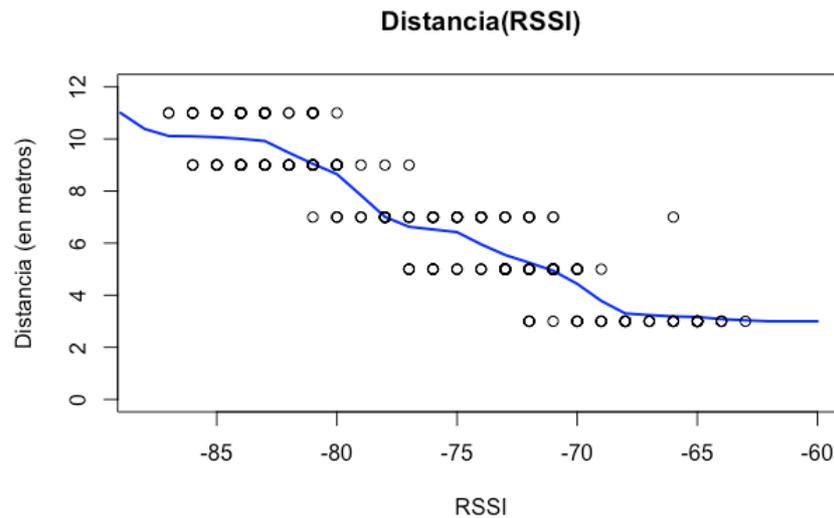


Figura 39. Relación Distancia-RSSI a partir de método bayesiano para medidas en entorno ruidoso durante un minuto, comparada con datos puntuales obtenidos de las distribuciones.

Por su parte, el resultado tras el algoritmo de Bayes, será más parecido a una recta que a una distribución escalonada, debido al solape de datos en los distintos valores RSSI. Como es de esperar, aunque exista ruido, la distancia es un factor importante a la hora de explicar la intensidad de la potencia de la señal recibida, por lo que, como vimos en el histograma, las distribuciones de potencia, aunque tengan alta variabilidad tendrán mayor densidad en ciertos puntos, que sí estarán diferenciados unos de otros de forma ordenada. Es por esto por lo que el algoritmo bayesiano nos proporciona al final una gráfica siempre decreciente, sin subidas repentinas en ningún punto.

Se aplica en el algoritmo bayesiano el cálculo del error cuadrático medio, obteniendo el valor de: 0.8841576 metros.

## **RRNN**

Al obtener mediciones tan similares entre sí, las redes neuronales no convergen de forma óptima cuando intentamos entrenarlas con dichos datos, por la similitud de los datos de entrada. No se ha logrado poder implementar este algoritmo en este caso.

## Combinación de todos los algoritmos

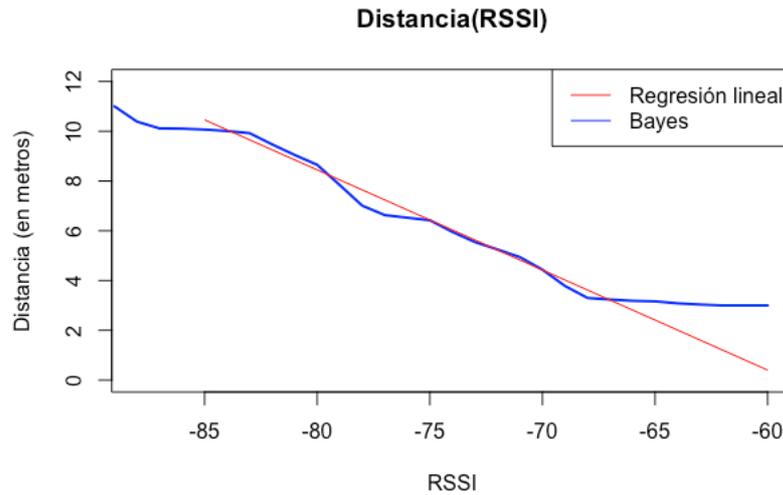


Figura 40. Salida de función distancia de datos con ruido tras filtrado de Kalman.

Finalmente, se obtiene la gráfica tras aplicar los distintos algoritmos al set de datos. En este caso, no tenemos el resultado de las redes neuronales pues los resultados obtenidos no fueron satisfactorios.

Comprobamos en el resultado obtenido, como el algoritmo de Bayes se asemeja bastante a la recta de la regresión lineal y, si no fuera por que no tenemos distancias inferiores a 3 metros y superiores a 11, en la gráfica probablemente la función bayesiana seguiría la forma de la regresión lineal con ligeras irregularidades.

### 6.3.1 FILTRADO DE KALMAN DE DATOS CON RUIDO

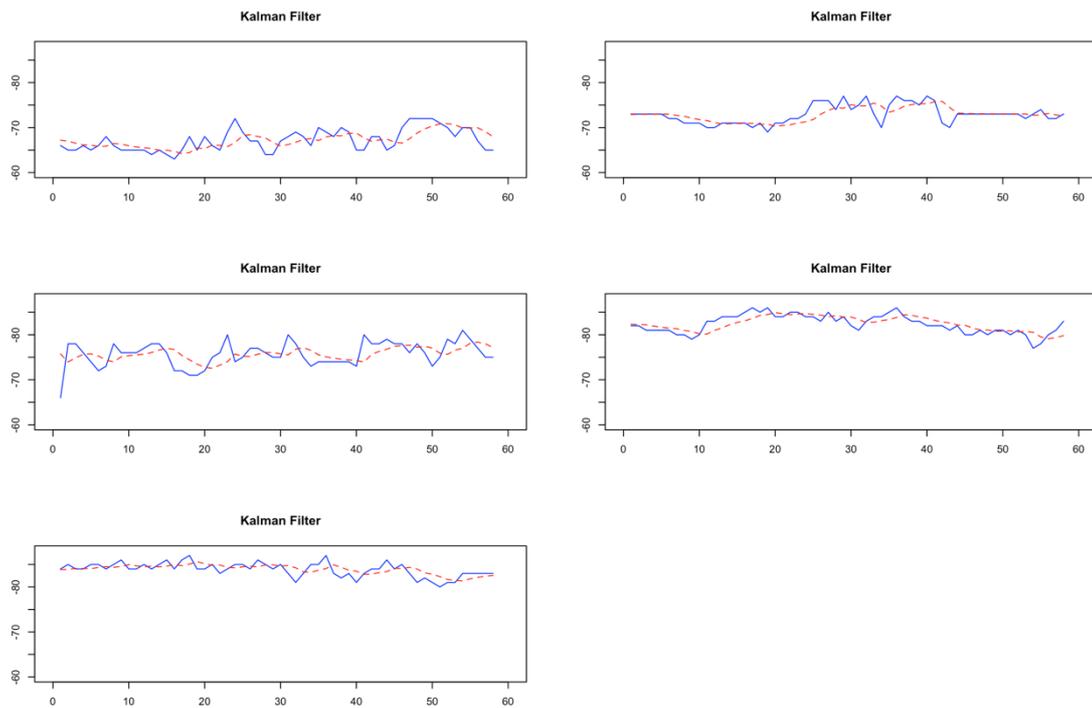


Figura 41. Filtrado de Kalman de datos con ruido. Se muestra en la figura el filtrado de cada distribución. 3 metros: arriba izquierda, 5 metros arriba derecha, 7 metros: centro izquierda, 9 metros: centro derecha, 11 metros: abajo izquierda.

En este caso, el efecto del filtro de Kalman es más notable debido a la alta presencia de ruido, y puede apreciarse mejor el suavizado de los datos en cada una de las distribuciones. Se procede ahora a aplicar los algoritmos de regresión lineal, Bayes y RRNN a los datos filtrados.

## Distribución

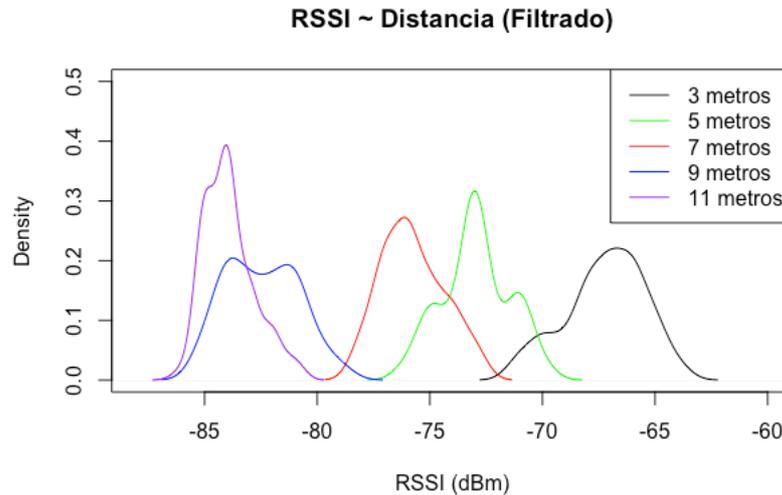


Figura 42. Distribución de valores RSSI para medidas en entorno con ruido durante un minuto tras filtro de Kalman.

La implementación del filtro Kalman produce en la distribución una reducción de la varianza en los datos y con ello, una mayor acumulación en los datos intermedios, provocando que las distribuciones se vean más separadas unas de otras. Una aproximación más analítica puede verse a partir del análisis de diagrama de cajas.

## Diagramas de caja

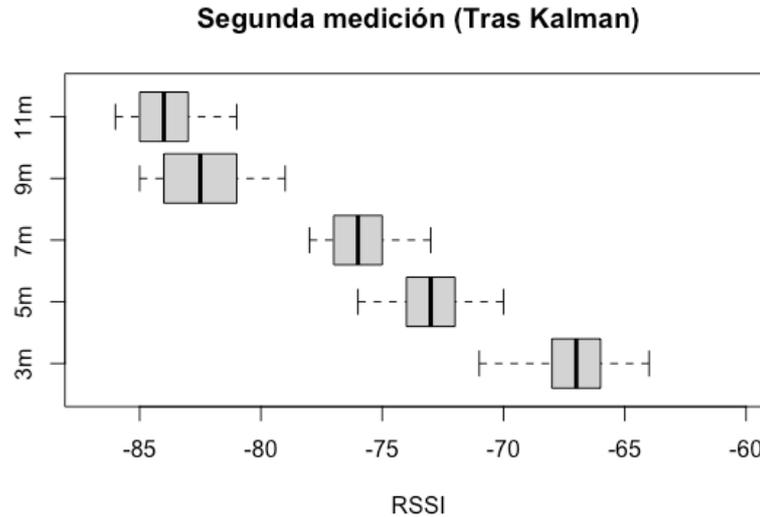


Figura 43. Diagramas de caja de datos con ruido tras filtrar por Kalman

Comprobamos aquí como la varianza se ve reducida tras el filtrado. Ahora son solo las distribuciones más alejadas las que coinciden en el rango intercuartílico.

	3m	5m	7m	9m	11m
Min.	:-71.00	:-76.00	:-78.00	:-85.00	:-86.00
1st Qu.	:-68.00	:-74.00	:-77.00	:-84.00	:-85.00
Median	:-67.00	:-73.00	:-76.00	:-82.50	:-84.00
Mean	:-67.22	:-72.95	:-75.76	:-82.45	:-83.91
3rd Qu.	:-66.00	:-72.00	:-75.00	:-81.00	:-83.00
Max.	:-64.00	:-70.00	:-73.00	:-79.00	:-81.00

Figura 44. Medidas estadísticas de las mediciones con ruido tras el filtro Kalman

No solo se reduce el rango intercuartílico como demuestran los datos, sino en parte se reduce un poco el máximo y mínimo de las distribuciones.

## Regresión lineal

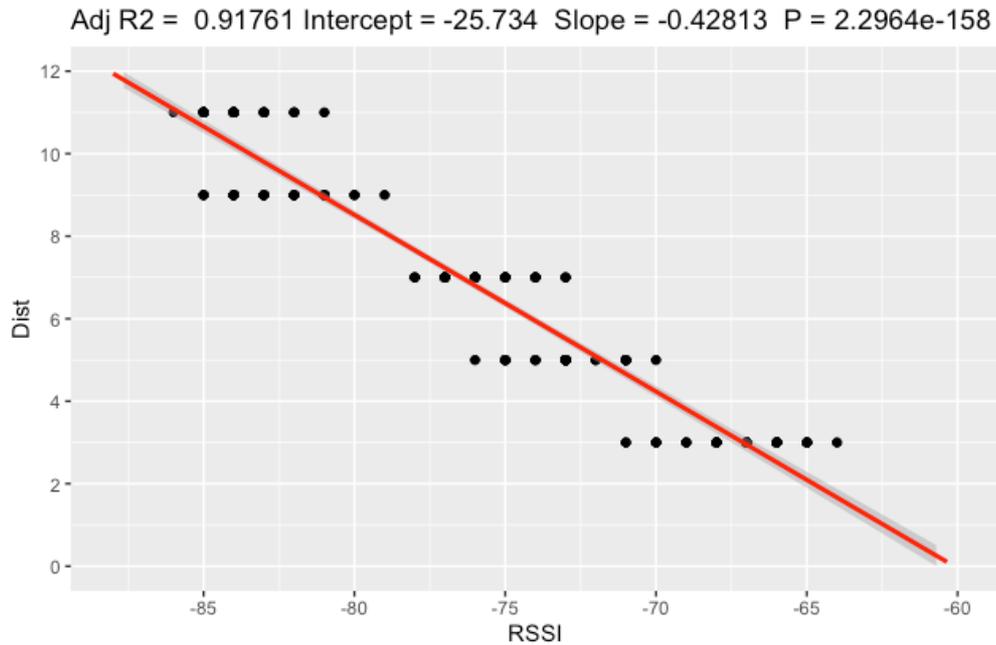


Figura 45. Modelo de regresión lineal para medidas en entorno con ruido durante un minuto tras pasar por filtro Kalman.

La función que sigue la recta de regresión es la siguiente:

$$\text{Distancia}(RSSI) = -25.734 - 0.42813 * RSSI$$

Fijándonos en los resultados de la regresión, vemos que, de nuevo, el p-valor tiene un valor prácticamente 0, lo que demuestra que la variable RSSI para medidas en entorno con ruido durante un minuto es significativa en el modelo.

El  $R^2$  tiene tras el filtrado de los datos con ruido, un valor de 0.91. Si tenemos en cuenta que el  $R^2$  del modelo generado con los datos previo paso por el filtro daba un valor 0.86, podemos concluir que, de nuevo, el filtro de Kalman, al reducir la variabilidad de los datos y con ello, el ruido, aumenta el coeficiente de correlación del modelo.

Tras el filtrado de Kalman, se obtiene el error cuadrático medio de la regresión lineal en un entorno ruidoso. El valor, en este caso es de: 0.6568557 metros

## Bayes

Se aplica a los datos ruidosos tras el filtro el algoritmo de Bayes, obteniendo la siguiente figura:

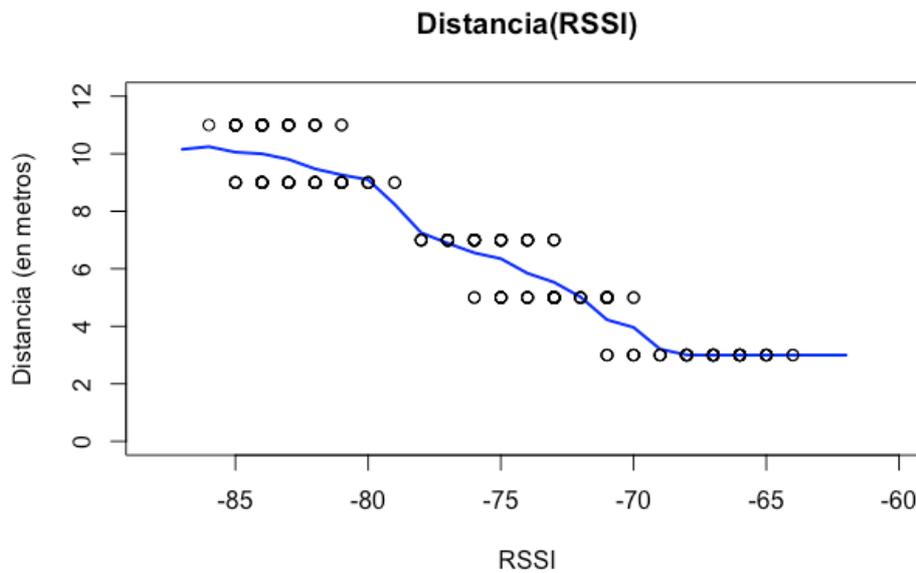


Figura 46. Relación Distancia-RSSI a partir de método bayesiano para medidas en entorno ruidoso durante un minuto, tras pasar por filtro Kalman, comparada con datos puntuales obtenidos de las distribuciones

Se obtiene el valor del error cuadrático medio para el caso post-filtro en un entorno ruidoso, para el algoritmo de inferencia bayesiana. En este caso, el valor del error es de: 0.7985478 metros.

## **RRNN**

A pesar de haber reducido el ruido de la señal a través del filtro, las redes neuronales siguen sin converger. Logra converger la red neuronal responsable de la clasificación a 3 metros, debido a que ha logrado diferenciarse en gran parte del resto de distribuciones. Sin embargo, solo es válido para nuestro estudio un resultado en el que todas las redes pudieran ser entrenadas, por lo que no usaremos redes neuronales en el caso de mediciones en un entorno ruidoso tras haber sido filtradas por Kalman.

## Todo junto

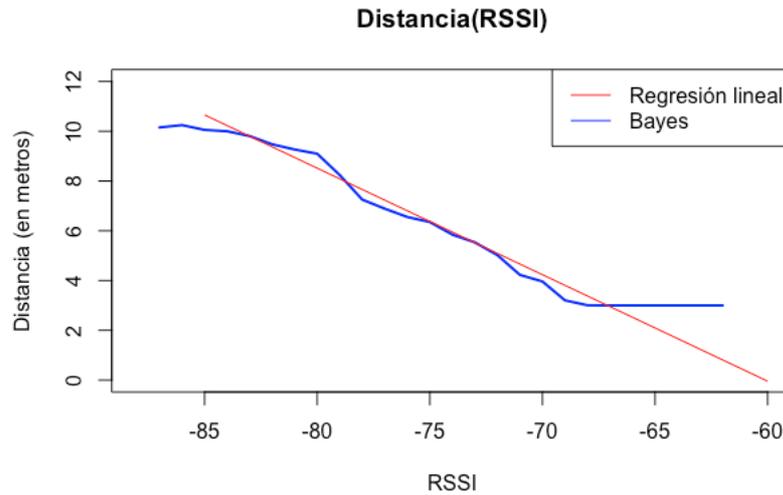


Figura 47. Resultado de la función de distancia tras recibir datos con ruido pasados por el filtro Kalman

Obtenemos un resultado con todos los algoritmos excepto redes neuronales, debido a que no se ha logrado entrenar a la red con los datos disponibles. La figura resultante resulta muy similar al caso previo al paso por el filtro Kalman. En próximos apartados se compararán ambos resultados.

## Comparativa del error cuadrático

Tabla 3. Comparativa error cuadrático para algoritmos bayesiano y de regresión lineal en entorno con ruido.

	Entorno con ruido	Entorno con ruido + Kalman
Regresión lineal	1.118551	<b>0.6568557</b>
Bayes	<b>0.8841576</b>	0.7985478

## 6.4 COMPARATIVAS FINALES DE DATOS DE DISTANCIA

### Comparación Bayes+Kalman

Finalmente, comparamos entre los métodos que mejor resultado nos han aportado, la regresión lineal, Bayes y Kalman.

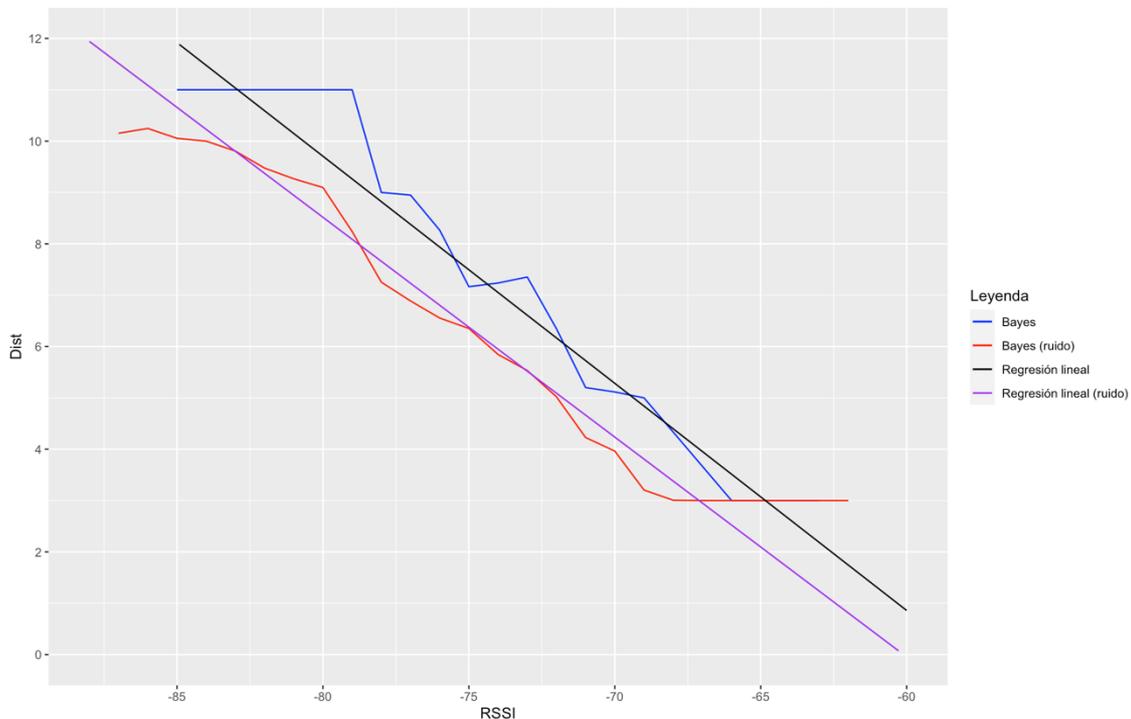


Figura 48. Comparativa entre algoritmo de Regresión lineal frente a Bayes tras el filtro de Kalman en entorno sin ruido y en entorno con ruido.

Se muestra en la figura el resultado de aplicar los algoritmos de regresión lineal y Bayes a un set de datos limpios y otro con ruido tras pasar por el filtro de Kalman.

En primer lugar, comparando las dos rectas de regresión lineal, puede verse que ambas poseen una pendiente muy similar, sin embargo, es el coeficiente  $b_0$  el que determina finalmente la distancia entre ambas rectas. Ambas rectas se acercan entre sí en valores de potencia RSSI mayores, dando a entender que la cercanía a la baliza no solo

aporta en ambos casos una mayor potencia, sino una menor oscilación de valores provocada por el ruido en el ambiente.

Por otro lado, comparando el resultado tras el algoritmo de Bayes de ambos sets de datos, puede verse como en el caso de los datos limpios, esta se encuentra más escalonada que en el caso de los datos con ruido, debido a que la probabilidad de encontrar datos pertenecientes a una distribución, sin colisionar con datos de otras, es mucho mayor en este caso, por lo que la regresión tenderá a acercarse a dichos valores de distancia. En el caso de los datos con ruido, la regresión de Bayes tiende a parecerse a la regresión lineal al final.

Podemos suponer, por tanto, en el caso de los datos limpios, que, si tomásemos infinitas mediciones de distancia, el resultado acabaría siendo una recta, muy semejante a la obtenida mediante la regresión lineal, como hemos observado con los datos con ruido.

## Comparación de algoritmos Bayes y Kalman aplicados a datos con ruido sin filtrar y tras filtrar

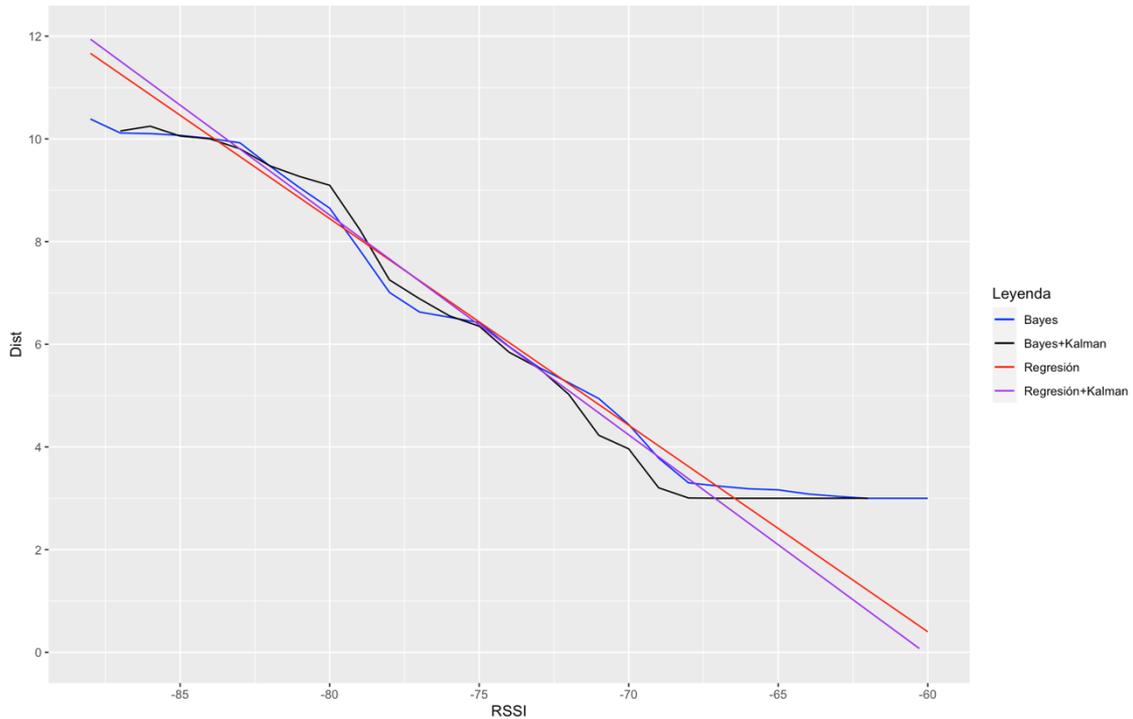


Figura 49. Comparativa entre algoritmo de Regresión lineal frente a Bayes, previo al filtro de Kalman y tras el filtro en entorno con ruido.

En la figura puede comprobarse cómo tras el filtro de Kalman, la pendiente de la regresión lineal se ve ligeramente alterada, aunque el cambio no es muy significativo. Por ejemplo, en un RSSI de -65 dBm, la diferencia entre una regresión con Kalman y sin Kalman es de menos de medio metro.

Por otro lado, la gráfica que modela el algoritmo de Bayes, en el caso de la regresión tras filtrado con Kalman parece alejarse más de la recta de regresión lineal que aquella con los datos sin filtrar. Esto es debido a que, tras el filtrado, la varianza de las distribuciones se reduce, dando lugar a que en aquellos lugares donde se tomaron las medidas, (a 3 metros, 5 metros, etc.) la función de Bayes tienda a acercarse a dichos valores, tomando una forma mayormente escalonada, como ocurría al ser comparada con los datos limpios.

### **Error cuadrático medio**

Para una comparación más exhaustiva de ambos algoritmos, se calcula el error cuadrático medio:

*Tabla 3. Comparación de error cuadrático para los distintos algoritmos.*

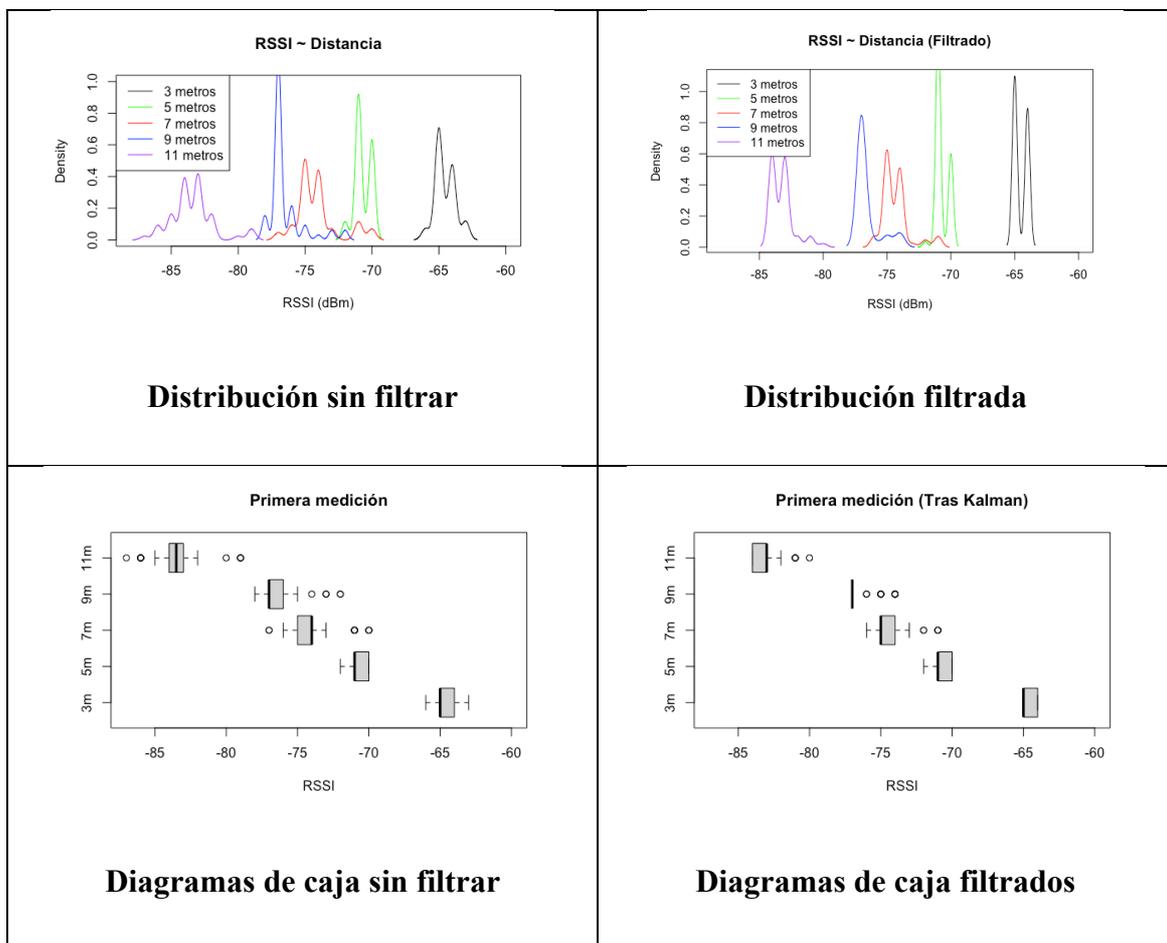
	<b>Entorno sin ruido</b>	<b>Entorno sin ruido + Kalman</b>	<b>Entorno con ruido</b>	<b>Entorno con ruido + Kalman</b>
<b>Regresión lineal</b>	<b>0.5242516</b>	<b>0.3513124</b>	1.118551	<b>0.6568557</b>
<b>Bayes</b>	0.5329002	0.4226761	<b>0.8841576</b>	0.7985478

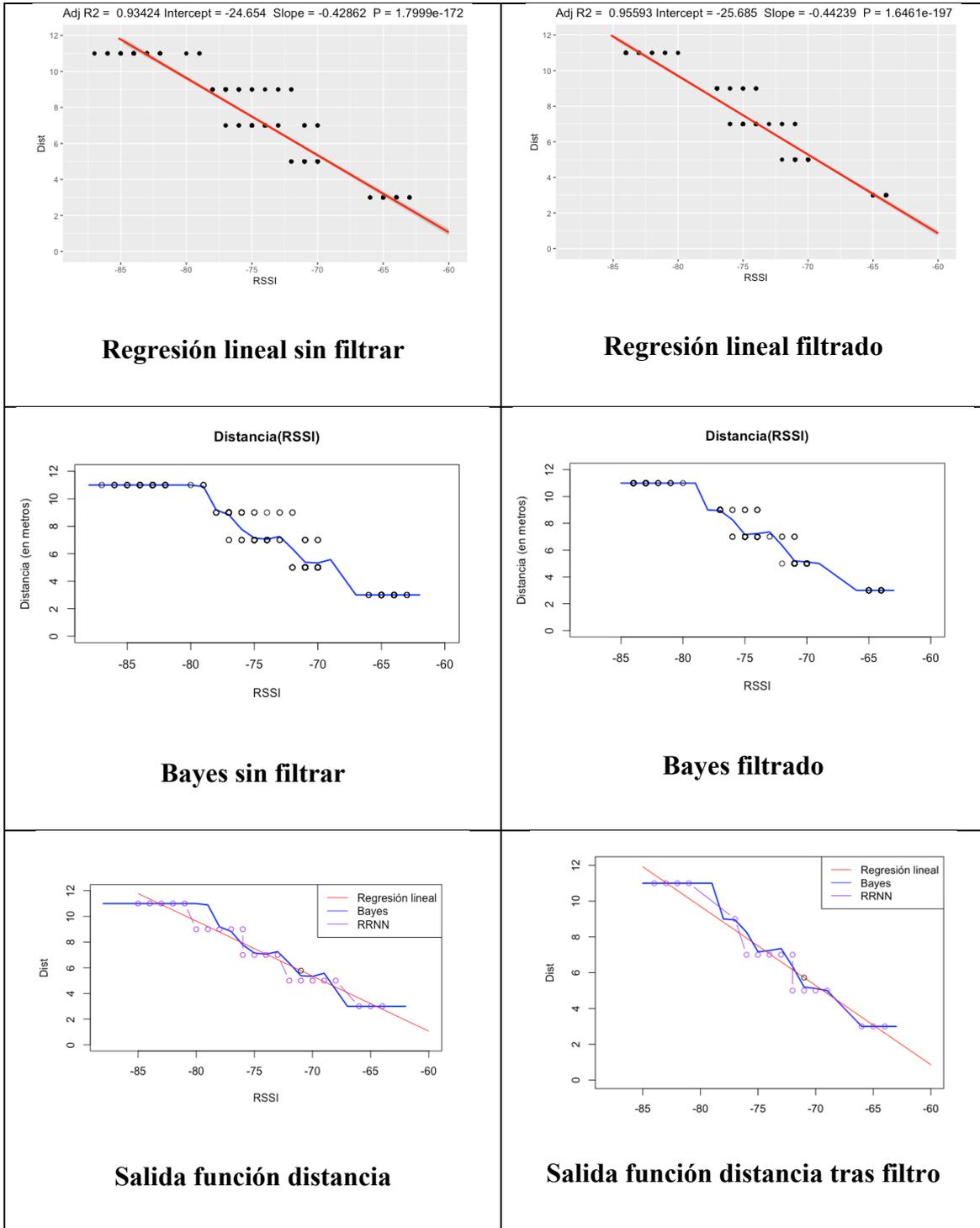
Como puede verse en la tabla, el error cuadrático obtenido es por lo general menor en la regresión lineal. Si bien es cierto que en un entorno con ruido gana el algoritmo bayesiano, tras la aplicación de un filtro de Kalman, la regresión lineal vuelve a superar a la inferencia bayesiana en ambos entornos. Además, se comprueba en la tabla el efecto que tiene el filtro Kalman sobre la reducción del error, siendo mucho más notable la diferencia en la regresión lineal que en el algoritmo de Bayes.

### COMPARATIVA DE FIGURAS SIN RUIDO

Se muestra a continuación una tabla con todas las figuras en entorno controlado libre de ruido, pre-filtrado y tras filtrado.

Tabla 4. Comparativa de figuras sin ruido.

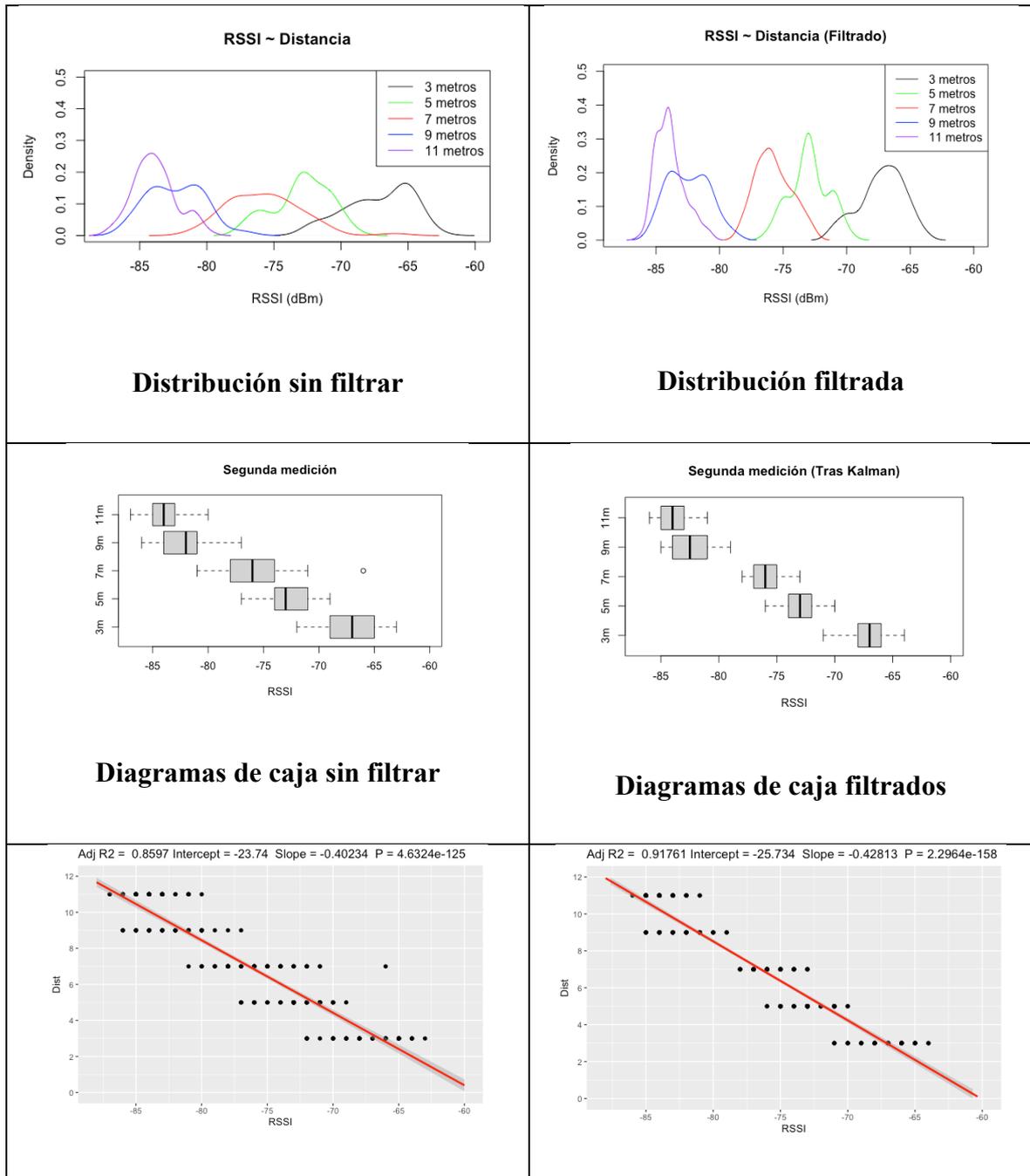


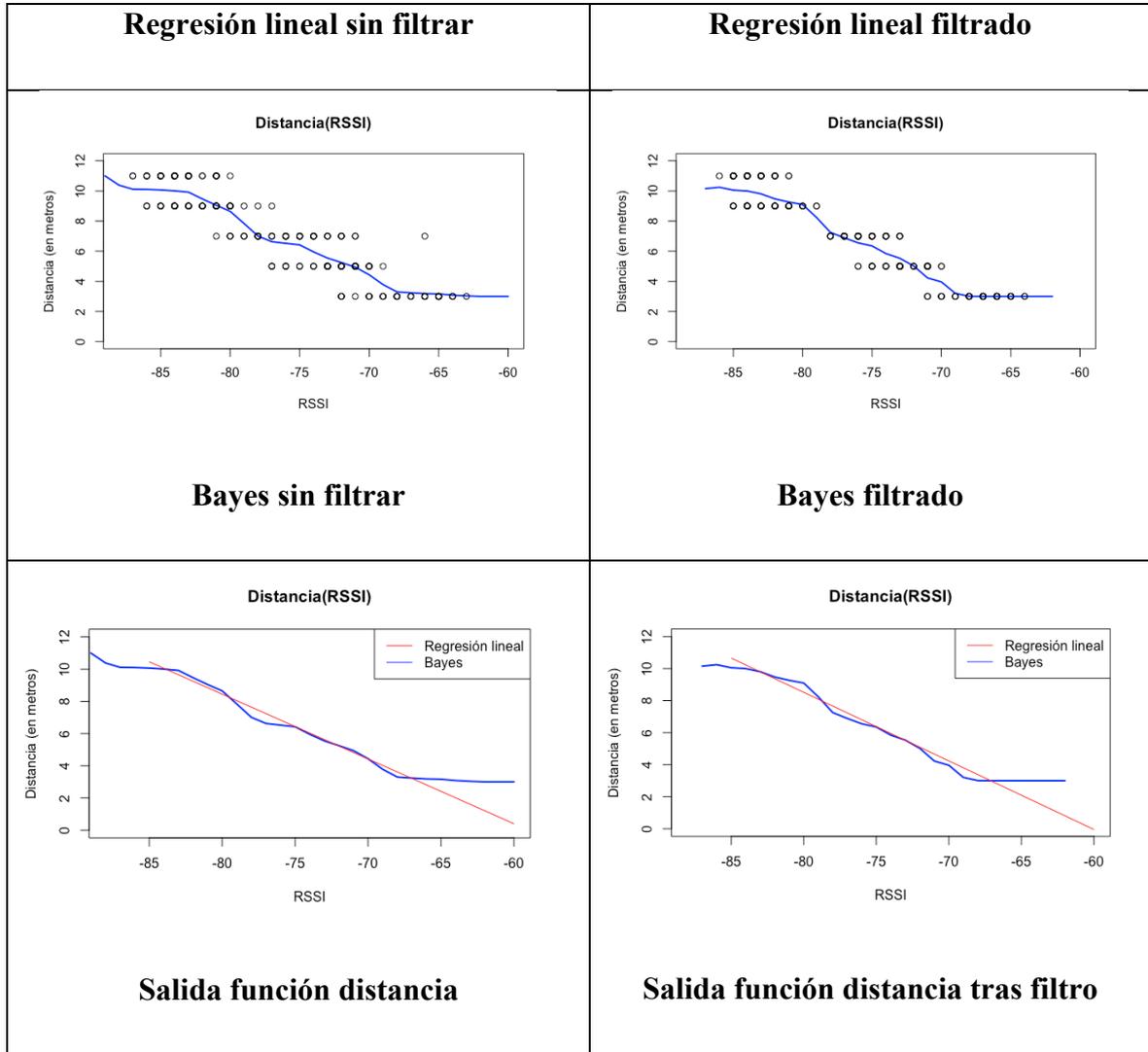


## COMPARATIVA DE FIGURAS CON RUIDO

Se muestra a continuación una tabla con todas las figuras con ruido, pre-filtrado y tras filtrado.

Tabla 5. Comparativa de figuras con ruido





## Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

Tras los resultados obtenidos en el experimento, y tras realizar varias comparaciones entre modelos, concluimos el experimento en los siguientes puntos:

### **Qué modelos se comportan mejor**

Tras realizar pruebas con todos los algoritmos disponibles, podemos llegar a las siguientes conclusiones.

En primer lugar, descartamos el entrenamiento de redes neuronales para la obtención de la distancia a partir de la clasificación de valores RSSI; esto se debe a que dichas redes neuronales dependen fundamentalmente de sus datos de entrenamiento, del ruido existente en el medio, y de la capacidad de computación.

Por otro lado, la asignación de un RSSI a una distancia a partir de los valores del rango de un diagrama de cajas es un método poco fiable, pues en el caso común de que 2 distribuciones coincidan entre sí, no podremos determinar la distancia real que le corresponda. Así que, este método queda descartado.

Vemos entonces que los modelos que mejor se comportan como algoritmos predictivos serán el bayesiano y el de regresión lineal, junto a la posible combinación con el filtrado de Kalman de los datos iniciales.

## **La regresión lineal es suficiente**

Finalmente, tras realizar comparativas entre distintos algoritmos de predicción tan avanzados y computacionalmente costosos como las redes neuronales, y otros tan simples como una regresión lineal, nos damos cuenta de que todos los algoritmos finalmente convergen en una función decreciente. Teniendo en cuenta que los algoritmos que mejor funcionan han demostrado ser el bayesiano y la regresión lineal, debemos discernir entre ambos. Tras realizar comparaciones de resultados, obtenemos que la gráfica desarrollada por el algoritmo bayesiano parece intentar imitar la forma de la recta de regresión, y esto se hace más notable en aquellas distribuciones que coinciden entre sí, donde existe probabilidad cruzada entre distribuciones. Esto podemos atribuirlo a que hemos tomado medidas discretas y bastante separadas entre sí en distancia, para un problema que estamos tratando de modelar de manera continua. Esto finalmente da un aspecto escalonado a la función (en especial en el caso en el que existe poco ruido en la señal).

Podemos entonces concluir que, si tomásemos infinitas mediciones en infinitos diferenciales de distancia, veríamos una función muy parecida a la recta que hemos obtenido con el algoritmo de regresión lineal.

Aún así, antes de tomar una decisión, debemos fijarnos en el error cuadrático medio obtenido finalmente por cada algoritmo, especialmente en el caso en el que exista ruido, pues será el caso de uso habitual para la implementación del algoritmo. Como se ve en los resultados, la combinación ganadora resulta ser el filtro de Kalman más la regresión, que es la que mejor reduce el efecto del error. Aún así la inferencia bayesiana sigue siendo un buen método de predicción, teniendo en su combinación con Kalman una diferencia de tan solo 15 cm de error con respecto al método elegido.

## **La importancia de estar en un entorno limpio**

Un entorno limpio nos permite generar un modelo mayormente dependiente del valor RSSI. Esta afirmación viene respaldada por el indicador  $R^2$  de la regresión lineal, donde, en el caso en el que las mediciones se realizan en entorno limpio, o tras haber limpiado la señal en entorno ruidoso con un filtro de Kalman, el porcentaje de correlación será mucho mayor que en un entorno con ruido.

Por el contrario, un entorno contaminado por el efecto de otras señales afectará a la relación entre RSSI y distancia provocando que los valores RSSI oscilen en mayor medida, y que los modelos de predicción tengan un margen de error mayor.

## **Relevancia del filtro Kalman**

Aplicar un filtro nos ayudará a reducir el ruido, como se ha demostrado en los resultados obtenidos, donde el  $R^2$  aumenta bastante porcentualmente tras el filtrado. Como se dijo anteriormente, a la hora de aplicar una regresión lineal, tener un  $R^2$  alto es importante, pues implica que el modelo es capaz de explicar mejor los datos introducidos para su entrenamiento, es decir, es influido en menor medida por variables externas.

Realizar los cálculos de un filtro Kalman no es un proceso excesivamente complejo, teniendo en cuenta que es un algoritmo matemático relativamente simple para un ordenador, por lo tanto, no se pierden realmente muchos recursos en su cómputo.

Como explicamos en el punto “la regresión lineal es suficiente”, el mejor método ante un entorno con ruido es la combinación de reg. Lineal junto al filtro de Kalman. La no aplicación del filtro supone una diferencia de error en el caso de la regresión lineal de casi medio metro, mientras que, en la inferencia bayesiana, esta diferencia es bastante inferior (tan solo 10 cm).

## **Aplicativo**

Para el caso de uso final cabe preguntar, ¿necesitamos tanta precisión en la estimación de la distancia?

La respuesta es no. La diferencia, finalmente, entre el método bayesiano y la regresión lineal es mínima, y en casos extremos podría ser de un metro, pero no mucho más. En el caso de un aplicativo móvil, en el que este tenga que discernir entre qué baliza escoger como más cercana, ese pequeño error no será determinante, pues suponemos que el circuito de balizas estará conformado de tal forma que las balizas estén colocadas a distancias superiores a un metro unas de otras.

Es por ello por lo que, concluimos finalmente, que una regresión lineal, con toda su simpleza, es el método más eficiente y funcional que podríamos emplear en una futura aplicación móvil donde debemos diferenciar entre la cercanía de balizas, teniendo en cuenta además que dicha aplicación no necesita una extrema precisión en la localización de balizas, pues el usuario solo necesitaría una aproximación de su cercanía a esta para el correcto funcionamiento de la aplicación.

## **Rendimiento de batería**

Por otro lado, en referencia al análisis de rendimiento de batería, como cabría esperar, el modo low-power que nos ofrece la librería FlutterBleLib parece el más adecuado, pues es el que mejor preserva la batería del dispositivo a lo largo del tiempo, tras un escaneo continuo. Sin embargo, el modo balanced, el cual muestra un intervalo algo superior demostró haber consumido un 1% menos que el modo low-power, aunque podemos achacar dicha variación a variables externas a la aplicación de escaneo, ya que no es una diferencia muy alta. Sí que es cierto que tampoco hubo diferencia entre los dispositivos mostrados en pantalla, pues los tiempos de escaneo, finalmente, no son apreciables para el usuario final, quien no notará la diferencia de milisegundos en la detección de balizas cercanas, entre un modo de baja latencia y un modo de baja energía. Donde sí notará diferencia, sin embargo, será en la batería de su dispositivo.

Finalmente, se recomienda el uso de *flutterblue* como librería para implementar el algoritmo de localización en una aplicación de Flutter, ya que, *flutterblue* implementa el modo low-power de serie, y, aunque no permite parametrización, para el desarrollador supone una librería mucho más sencilla de utilizar.

## **TRABAJOS FUTUROS**

El siguiente paso, tras determinar el método óptimo, es la implementación en una aplicación multiplataforma desarrollada en Flutter. Tras el análisis, concluimos que podemos implementar una función de regresión lineal como algoritmo para diferenciar entre distancias. Para ello tan solo habríamos de traducir el código utilizado para la regresión al lenguaje Dart, aplicar regresión lineal para balizas cercanas detectadas y discernir entre cuál es la que me da un valor de distancia de mayor cercanía.

Para próximos análisis de potencia de la señal se propone una continuación del experimento, donde se realice un análisis de la señal mucho mas exhaustivo. Si bien es cierto que para el caso de uso que nos ocupa en este proyecto no es necesaria mayor precisión, se propone cambiar el caso de uso de nuestro análisis para situaciones donde la precisión sí sea un factor clave.

Se sabe por el modelo log-distancia de la señal, que la potencia de una señal tiene una forma logarítmica con respecto a la distancia si tenemos en cuenta un rango de distancias que involucre distancias menores a 3 metros, que es la distancia mínima introducida en este proyecto. Por ello, una buena idea para análisis en futuros experimentos es ampliar el rango de distancia en las mediciones, para obtener resultados más precisos. Además, se propone ampliar el número de medidas a distintas distancias, con el fin de obtener resultados que no estén condicionados por la discretización de las mediciones.

Por otro lado, se propone experimentar con un mayor número de algoritmos de predicción, y con nuevos métodos de filtrado y optimización, en busca de mejorar la precisión en la predicción de distancia.

## Capítulo 8. BIBLIOGRAFÍA

- [1] Song, S. W., Lee, Y. S., Imdad, F., Niaz, M. T., & Kim, H. S. (2019). Efficient advertiser discovery in Bluetooth Low Energy devices. *Energies*, 12(9), 1707.
- [2] Gomez, C., Oller, J., & Paradells, J. (2012). Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12(9), 11734-11753.
- [3] Avila, L., & Reyes, C. (2012). Revisión estado del Arte de la tecnología Bluetooth. Investigación y desarrollo en TIC, 3(2), 9-16.
- [4] Bluetooth Core Specification Version 4.2 (2014). Bluetooth SIG. Recuperado de [https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc\\_id=441541](https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=441541) (visitado el 30 de enero 2021).
- [5] Welch, G., & Bishop, G. (1995). An introduction to the Kalman filter.
- [6] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.
- [7] Chai, S., An, R., & Du, Z. (2016, June). An indoor positioning algorithm using bluetooth low energy RSSI. In Proceedings of the 2016 International Conference on Advanced Materials Science and Environmental Engineering, Chiang Mai, Thailand (pp. 26-27).
- [8] Subhan, F., & Hasbullah, H. B. (2010, June). Minimizing discovery time in bluetooth networks using localization techniques. In *2010 International Symposium on Information Technology* (Vol. 2, pp. 648-653). IEEE.
- [9] Challa, N. S. R., Kesari, P., Ammana, S. R., Katukojwala, S., & Achanta, D. S. (2019, November). Design and implementation of Bluetooth-beacon based indoor positioning system. In *2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)* (pp. 1-4). IEEE.
- [10] Adewumi, O. G., Djouani, K., & Kurien, A. M. (2013, February). RSSI based indoor and outdoor distance estimation for localization in WSN. In *2013 IEEE international conference on Industrial technology (ICIT)* (pp. 1534-1539). IEEE.
- [11] Xu, J., Liu, W., Lang, F., Zhang, Y., & Wang, C. (2010). Distance measurement model based on RSSI in WSN. *Wireless Sensor Network*, 2(8), 606.
- [12] Benkic, K., Malajner, M., Planinsic, P., & Cucej, Z. (2008, June). Using RSSI value for distance estimation in wireless sensor networks based on ZigBee. In *2008 15th International Conference on Systems, Signals and Image Processing* (pp. 303-306). IEEE.
- [13] Li, G., Geng, E., Ye, Z., Xu, Y., Lin, J., & Pang, Y. (2018). Indoor positioning algorithm based on the improved RSSI distance model. *Sensors*, 18(9), 2820.

- [14] Awad, A., Frunzke, T., & Dressler, F. (2007, August). Adaptive distance estimation and localization in WSN using RSSI measures. In *10th Euromicro Conference on Digital System Design Architectures, Methods and Tools (DSD 2007)* (pp. 471-478). IEEE.
- [15] Zhang, K., Zhang, Y., & Wan, S. (2016, October). Research of rssi indoor ranging algorithm based on gaussian-kalman linear filtering. In *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)* (pp. 1628-1632). IEEE.
- [16] Parameswaran, A. T., Husain, M. I., & Upadhyaya, S. (2009, September). Is rssi a reliable parameter in sensor localization algorithms: An experimental study. In *Field failure data analysis workshop (F2DA09)* (Vol. 5). IEEE.
- [17] Jianyong, Z., Haiyong, L., Zili, C., & Zhaohui, L. (2014, October). RSSI based Bluetooth low energy indoor positioning. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)* (pp. 526-533). IEEE.
- [18] Bekkelien, A., Deriaz, M., & Marchand-Maillet, S. (2012). Bluetooth indoor positioning. *Master's thesis, University of Geneva*.
- [19] Altini, M., Brunelli, D., Farella, E., & Benini, L. (2010, May). Bluetooth indoor localization with multiple neural networks. In *IEEE 5th International Symposium on Wireless Pervasive Computing 2010* (pp. 295-300). IEEE.
- [20] Tosi, J., Taffoni, F., Santacatterina, M., Sannino, R., & Formica, D. (2017). Performance evaluation of bluetooth low energy: A systematic review. *Sensors*, 17(12), 2898.
- [21] Jung, J., Kang, D., & Bae, C. (2013, October). Distance estimation of smart device using bluetooth. In *The Eighth International Conference on Systems and Networks Communications* (pp. 13-18).
- [22] Wu, Z., Xu, Q., Li, J., Fu, C., Xuan, Q., & Xiang, Y. (2017). Passive indoor localization based on csi and naive bayes classification. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(9), 1566-1577.
- [23] Vanheel, F., Verhaevert, J., Laermans, E., Moerman, I., & Demeester, P. (2011). Automated linear regression tools improve rssi wsn localization in multipath indoor environment. *EURASIP Journal on Wireless Communications and Networking*, 2011(1), 1-27.
- [24] Abusara, A., & Hassan, M. (2015, May). Error reduction in distance estimation of rss propagation models using kalman filters. In *2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)* (pp. 1-5). IEEE.
- [25] Gamez, M. (2021). Objetivos y metas de desarrollo sostenible. Recuperado 9 de julio 2021, de <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- [26] Sector turístico en España. Recuperado 9 de julio 2021 de <https://mediterrani.com/blog/sector-turistico-espana/>
- [27] Objetivos de desarrollo sostenible. Recuperado el 9 de julio de 2021 de <https://www1.undp.org/content/undp/es/home/sustainable-development-goals.html>
- [28] Objetivos de desarrollo sostenible. Recuperado el 9 de julio de 2021 de <http://www.exteriores.gob.es/portal/es/politicaexteriorcooperacion/nacionesunidas/paginas/objetivosdedesarrollodelmilenio.aspx>

[29] FlutterBleLib Reference. Recuperado el 9 de julio de 2021 de <https://github.com/dotintent/FlutterBleLib/blob/master/REFERENCE.md>

# ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

Los ODS, Objetivos de Desarrollo Sostenible, son objetivos adoptados por todos los Estados Miembros de la ONU en 2015 como un llamado universal para poner fin a la pobreza, proteger el planeta y garantizar el bienestar de todas las personas para 2030 [27]. Se trata de 17 objetivos, con 169 metas de carácter integrado e indivisible que abarcan las esferas económica, social y ambiental [28]. Este proyecto se encuentra alineado con dichos objetivos, en la búsqueda de mejorar la calidad de vida de la población de cara al 2030.

La definición de cada uno de los objetivos con los que se alinea este proyecto se ha extraído de [25].

Se muestra a continuación una tabla con los ODS con los que se alinea este proyecto:

*Tabla 6. Alineación del proyecto con los Objetivos de Desarrollo Sostenible.*

<b>ODS</b>	<b>Dimensión</b>	<b>Rol</b>
<b>Objetivo 8:</b> Promover el crecimiento económico inclusivo y sostenible, el empleo y el trabajo decente para todos	Económica	Primario
<b>Objetivo 9:</b> Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación	Económica	Primario

<b>Objetivo 11:</b> Lograr que las ciudades sean más inclusivas, seguras, resilientes y sostenibles	Social	Secundario
---	--------	------------

### **Dimensión económica del proyecto**

Garantizamos el cumplimiento del **octavo objetivo**, el crecimiento económico y el mantenimiento del empleo de aquellos negocios que dependen de que sus clientes estén presentes, tales como lo son en principio la finalidad de una visita a un museo, visitas guiadas o, una jornada de puertas abiertas en ICAI.

El efecto de la pandemia ha sido notable especialmente en el sector servicios. Aquellos empleos basados en la interacción con el consumidor, y que requieren de que el cliente se encuentre presencialmente en el local, se han visto altamente perjudicados por el confinamiento y, aún hoy, las restricciones y, en parte, el miedo de la población a exponerse al contagio del virus, siguen afectando económicamente a dichos negocios. Este problema, si se mantiene en el tiempo, podría llegar al cese de la actividad para muchos negocios tales como museos, visitas y, en especial, para el sector del turismo, un sector, el cual representa el 11,2% del PIB en España, y que supone el 13% total del empleo [26]. Esto no solo implicaría un duro golpe para la economía, sino para el bienestar en el país, ya que muchas personas perderían su empleo si no se buscara una solución rápido al respecto.

Mediante nuestra propuesta de aplicar tecnologías BLE de localización a aplicaciones móviles, reducimos el contacto directo con los clientes, mientras mantenemos el funcionamiento de la actividad comercial, para el caso actual de pandemia en el que nos encontramos, y para estar preparado ante futuras enfermedades que puedan suponer un duro golpe para estos comercios.

El objetivo que cumple el proyecto es el siguiente: “Lograr niveles más elevados de productividad económica mediante la diversificación, la modernización tecnológica y la innovación, entre otras cosas centrándose en los sectores con gran valor añadido y un uso intensivo de la mano de obra”.

Por otro lado, el proyecto de investigación está vinculado al **objetivo número 9**, el fomento de la innovación tecnológica. La investigación de métodos de localización BLE, finalmente, no solo tiene aplicación en el sector del comercio y visitas guiadas, sino que es aplicable en entornos de manufacturación, donde podría ser de utilidad el conocimiento de la posición de elementos de ensamblado en una cadena de montaje. Además, el estudio de la señal BLE para su implementación en aplicaciones móviles permite un amplio campo de aplicación, pues no solo debe limitarse al caso de uso especificado en el proyecto, una visita guiada, sino a cualquier caso en el que pueda utilizarse la tecnología BLE. Un ejemplo podría ser, un sistema de check-in en la universidad o el trabajo, o incluso en el transporte público.

Se busca entonces, mediante este proyecto, cumplir la siguiente meta: “Aumentar la investigación científica y mejorar la capacidad tecnológica de los sectores industriales de todos los países, en particular los países en desarrollo, entre otras cosas fomentando la innovación y aumentando considerablemente, de aquí a 2030, el número de personas que trabajan en investigación y desarrollo por millón de habitantes y los gastos de los sectores público y privado en investigación y desarrollo”.

### **Dimensión social**

Por último, el proyecto está alineado, de forma secundaria, con el objetivo de desarrollo número 11: Lograr que las ciudades sean más inclusivas, seguras, resilientes y sostenibles.

La meta que este proyecto ayuda a cumplir es el siguiente: “Redoblar los esfuerzos para proteger y salvaguardar el patrimonio cultural y natural del mundo”.

El proyecto, mediante su implementación en aplicaciones móviles, puede ayudar a proteger el patrimonio cultural fomentando la actividad turística, la cual contribuye económicamente a la preservación del arte y la cultura.