



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escuela Técnica superior de Ingeniería Informática
Universidad Politécnica de Valencia

Dispositivo para la medición de la satisfacción del
alumno en la docencia

Trabajo Fin de Grado
Grado en Ingeniería Informática

Autor: Sergio Delgado Pozo

Tutor: Jose Luís Poza Luján

Cotutores:

Juan Luis Posadas Yagüe

Miguel Rebollo

2016-2017

	<p>DMSAD <i>Dispositivo para la medición de la satisfacción del alumno en la docencia.</i></p>	
---	---	---

Querría dedicar este trabajo a todas aquellas personas que conocí durante la carrera y con las cuales compartí este maravilloso trayecto, a los amigos que siempre han estado ahí, a los profesores que mantienen viva su pasión por la enseñanza, incluido mi tutor Jose Luís Poza y cotutores y a mis padres por su apoyo incondicional.

Índice de contenido

1. Introducción	8
1.1. Finalidad del documento objeto	8
1.2. Docencia, sociedad y emociones	8
1.3. Motivación	9
2. Objeto del proyecto.....	9
3. Antecedentes.....	10
4. Descripción de la situación actual.....	13
4.1. Descripción del entorno actual	13
4.2. Resumen de las principales deficiencias identificados	14
5. Definiciones y abreviaturas.....	14
6. Requisitos iniciales	18
7. Alcance.....	19
8. Hipótesis y restricciones	24
9. Estudio de alternativas y viabilidad.....	25
10. Descripción de la solución propuesta.....	26
11. Organización y gestión del proyecto	27
11.1.1. Roles y responsabilidades	27
12. Planificación temporal	28
12.1. Cronograma temporal.....	31
12.2. Evolución del plan de proyecto.....	34
13. Presupuesto.....	35
14. Especificaciones del sistema	36
14.1. Requisitos no funcionales.....	36
14.2. Requisitos funcionales.....	38
14.3. Diagramas BPMN.....	40
14.4. Diagrama de secuencia.....	42

15.	Marcos y diseño.....	43
15.1.	Marco Hardware	43
15.1.1.	Componentes del sistema	43
	Arduino UNO rev3.....	43
	Estructura.....	44
	Protoboard	46
	Potenciómetro	47
	Led Bar Grove.....	48
	Módulo Wifi ESP8266	49
15.2.	Marco Software	50
15.2.1.	Bibliotecas	50
15.2.2.	Implementación.....	52
15.3.	Versiones, diseños del sistema y evolución del prototipo.....	57
15.3.1.	Versión alfa 1.0	57
15.3.2.	Versión alfa 2.0	58
15.3.3.	Versión Beta 1.0.....	59
15.3.4.	Versión release	60
15.4.	Carcasa	61
15.4.1.	Bocetos.....	61
15.4.2.	Diseño final en 3D.....	63
16.	Prototipo final y ejemplo de funcionamiento.....	65
17.	Conclusión y valoración	68
18.	Bibliografía y herramientas	69
18.1.	Herramientas	69
18.2.	Bibliografía.....	70
18.3.	Fuentes de imágenes	71

Índice de tablas

Tabla 1: Alternativas al proyecto objeto.....	25
Tabla 2 : Matriz RACI Cliente-Desarrollador	27
Tabla 3 : Fechas e Hitos	31
Tabla 4 : Presupuesto del proyecto.....	35

Índice de imágenes

Figura 1: Esquema del proyecto Led Displayed Twitter Sentiment Analysis.....	10
Figura 2: Resultado de Crying Baby Detector.....	11
Figura 3: Resultado de World's Mod in a Box.....	12
Figura 4 : Contenido Sprint 1.....	19
Figura 5 : Contenido Sprint 2 y 3.....	19
Figura 6 : Contenido Sprint 4.....	20
Figura 7 : Contenido Sprint 5.....	20
Figura 8 : fechas Sprint 1.....	28
Figura 9 : fechas Sprint 2.....	29
Figura 10 : fechas Sprint 3.....	29
Figura 11: fechas Sprint 4 y 5.....	30
Figura 12 : Gantt Entregable 1.....	31
Figura 13 : Gantt Entregable 2.....	32
Figura 14 : Gantt Entregable 3.....	32
Figura 15 : Gantt Entregable 4.....	33
Figura 16 : Gantt Entregable 5.....	33
Figura 17 : Vúmetro y sus rangos.....	39
Figura 18 : Formato final de los Tweets.....	39
Figura 19 : Diagrama BPMN - Caso 1.....	40
Figura 20 : Diagrama BPMN - Caso 2.....	41
Figura 21 : Diagrama de secuencia del prototipo.....	42
Figura 22 : Placa Arduino UNO.....	43
Figura 23 : Placa Arduino UNO y componentes.....	44
Figura 24 : Protoboard.....	45
Figura 25 : Conexiones Protoboard.....	46
Figura 26 : Potenciómetro rotativo.....	47
Figura 27 : Led Bar en su placa.....	48
Figura 28 : Módulo ESP 8266.....	49
Figura 29 : Circuito prototipo 1.....	57
Figura 30 : Circuito prototipo 2.....	58
Figura 31 : Circuito prototipo 3.....	59
Figura 32 : Circuito prototipo 4.....	60
Figura 33 : Boceto del compartimento de Arduino y protoboard.....	61
Figura 34 : Boceto de la tapa.....	61
Figura 35 : Boceto del perfil.....	62
Figura 36 : Muestra del diseño final 1.....	63
Figura 37 : Muestra del diseño final 2.....	63

Figura 38 : Muestra del diseño final 3.....	64
Figura 39 : Muestra del diseño final 4.....	64
Figura 40 : Vúmetros en rango 0.	64
Figura 41 : Usuario moviendo potenciómetro.	65
Figura 42 : Vúmetros en rango 0 y 1 respectivamente	66
Figura 43 : Tweet enviado.....	66
Figura 44 : En espera.	66
Figura 45: (Izda) Usuario moviendo potenciómetro 1. (Centro) Usuario moviendo potenciómetro 2.. (Dcha) Vúmetros en rango 2 y 3 respectivamente	67
Figura 46 : Tweet enviado 2.....	67



DMSAD
*Dispositivo para la medición de
la satisfacción del alumno en la
docencia.*



1. Introducción

Para la elaboración de este documento se han seguido las pautas de la norma técnica para la realización de proyectos de sistemas de la información del Consejo de Colegios de Ingenieros en Informática, basada en el estándar UNE 157801.

1.1. Finalidad del documento objeto

En el presente documento, se realiza una visita guiada ante el nacimiento de la idea, los pasos preliminares, la metodología utilizada, el ciclo de vida, el diseño y el proceso de implementación del consecuente proyecto : “Dispositivo para la medición de la satisfacción del alumno en la docencia” (DMSAD).

Este documento supone un nexo de unión a todos los documentos básicos del proyecto objeto, conteniendo toda referencia necesaria para entenderlo, aportando también conocimiento sobre cómo se ha logrado su total desarrollo.

También se informará sobre el por qué de este, su motivación, las necesidades que han hecho que nazca el proyecto, sus antecedentes, la elección de esta opción frente a otras y el objetivo final a cubrir.

1.2. Docencia, sociedad y emociones

En el mundo de la docencia, tanto alumnos como profesores deben poner de su parte para lograr resultados satisfactorios y conseguir que uno de los pilares de la sociedad, la educación, de pie a un futuro con grandes profesionales dispuestos a cambiar el mundo y realizar nuevas soluciones ante los problemas de la vida moderna.

En este ámbito, no solo convergen los conocimientos, los apuntes, las pizarras y las largas noches de estudio, sino que también lo hacen las emociones, a menudo olvidadas, cuyo peso es más relevante que la importancia que se les dan.

Si el nivel de satisfacción de los alumnos con la docencia y los contenidos dados durante las clases son positivos y además su estado emocional durante el periodo de tiempo que ocupen también cumple con ello, los resultados serán notablemente mejores: un alumno satisfecho con las metodologías utilizadas y el desempeño del docente, es un alumno



contento con el sistema empleado, y un alumno contento es un alumno con mayor capacidad de triunfar en los retos que se le presenten durante el arduo camino del estudiante.

1.3. Motivación

La realización de este prototipo viene apoyada por el interés que he desarrollado durante la carrera por la relación entre las metodologías empleadas por el sistema educativo, el desempeño de los docentes y el aprendizaje relativo del alumnado dependiendo de dichos factores, ya que en un gran número de casos, estos son decisivos para el éxito del alumno.

Con el consecuente proyecto, se pretende rectificar las metodologías empleadas a partir del nivel de satisfacción y el estado emocional de los alumnos, ya que con este *feedback* se obtendrá una herramienta para la institución universitaria que puede dar lugar a un sistema de mayor calidad y ayudar a ambos bandos a disfrutar del maravilloso mundo que supone la enseñanza.

2. Objeto del proyecto

Diseñar y desarrollar un dispositivo portátil e independiente que permita a los alumnos evaluar a tiempo real y de forma subjetiva el nivel de aprendizaje que están obteniendo con la clase o una actividad concreta, así como su estado emocional en dicho momento. Los datos obtenidos podrán ser recogidos por el profesorado y/o otras instituciones universitarias para poder utilizarlo como *feedback* sobre el estado del alumnado en las aulas y/o para fines estadísticos.

Con la ya mencionada retroalimentación, se podrán mejorar las metodologías utilizadas en las asignaturas que sean objeto de estudio para dar como producto un sistema docente universitario de mejor calidad y con mayor cohesión entre los alumnos y los profesores.

3. Antecedentes

“LED Displayed Twitter Sentiment Analysis”

El proyecto se basa en la inferencia de los sentimientos de uno o varios usuarios de *Twitter* a través del análisis del lenguaje utilizado en los tweets de susodichos autores.

Mediante la introducción de un *hashtag* determinado, se realiza una búsqueda de todos los *tweets* en los que este ha sido citado, recopilándolos para su posterior análisis.

Tras la obtención de los *tweets*, estos se dividen en mensajes individuales, para así obtener las palabras que los componen, y mediante un algoritmo de comparación, asignar valores positivos o negativos a dichos mensajes en relación a las palabras o expresiones utilizadas en él.

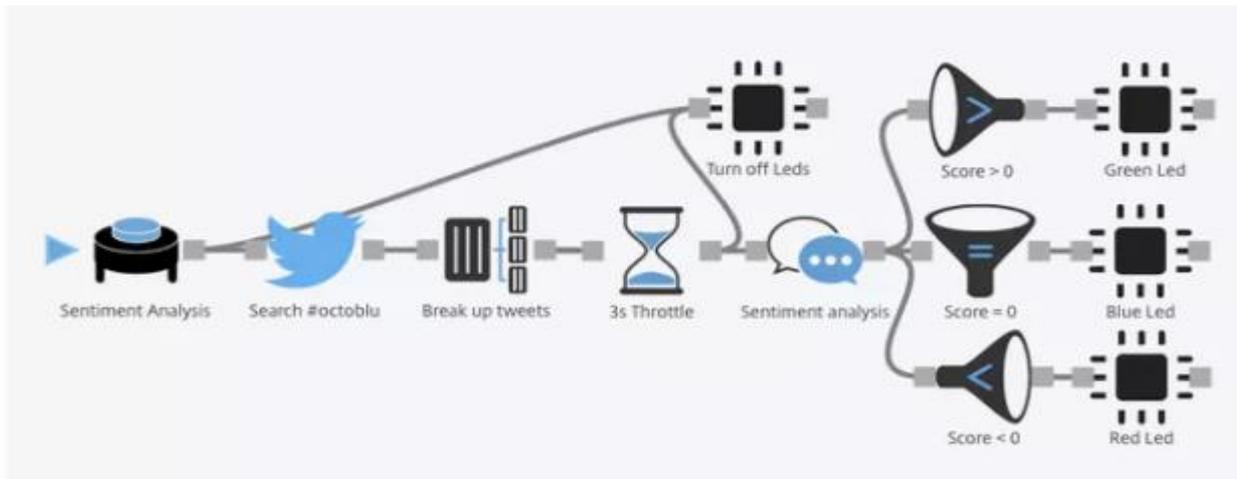


Figura 1: Esquema del proyecto Led Displayed Twitter Sentiment Analysis.

El valor obtenido en el producto de los valores de los *tweets* referentes al *hashtag*, será decisivo para que el sistema encienda el *LED* de un color concreto. Dependiendo del color del *LED* activado, el usuario podrá deducir el estado anímico conjunto de los autores que los redactaron:

Producto > 0 = verde (Feliz)

Producto == 0 = azul (Neutro)

Producto < 0 = rojo (disgustado)

Influencias al proyecto objeto:

En el proyecto “LED displayed sentiment analysis”, se hace uso de la herramienta *Twitter* para conocer el estado de ánimo respecto a un tema concreto o *hashtag* de un colectivo, es decir, la salida obtenida en ambos proyectos es similar, un *feedback* de datos numéricos referentes al estado anímico de los usuarios que es transformado en forma de luces de colores en el caso del proyecto que está siendo estudiado y un conjunto de datos numéricos que serán transformados en rangos de satisfacción y estado emocional en el proyecto a desarrollar.

Ambas salidas, aunque distintas en cuanto a naturaleza y precisión, sirven para realizar una estadística sobre un mismo dominio: el sentimental.

“Crying Baby Detector”

Proyecto que nació con la intención de “automatizar” el cuidado de los bebés en las horas nocturnas.

Similar al monitor habitual, pero con el aliciente de que dicho proyecto pretende calmar al bebé cuando llora y evitar que los padres trasnochen realizando ellos tal trabajo.

Monitorizando el sonido ambiente de la habitación, el dispositivo hará sonar música relajante o ruido blanco para intentar calmarlo cuando este llore.

El dispositivo está enlazado a una cuenta de *Twitter* y consta de un módulo *WiFi*. Gracias a la combinación de estos dos elementos, se enviarán mensajes privados con la cuenta enlazada a otra cuenta especificada por los usuarios del sistema, en los cuales vendrá incluida información a tiempo real del estado actual del bebé.



Figura 2: Resultado de Crying Baby Detector

Influencias al proyecto objeto:

“Crying baby detector” analiza el estado de ánimo del bebé a tiempo real mediante el uso de *Arduino* y *Twitter* e intenta automatizar las tareas repetitivas a las que pueden verse sometidos los padres.

La salida producida por el programa es similar a la del prototipo a desarrollar, ya que mediante el uso de ambas se permite recopilar información sobre el estado de ánimo de unos sujetos en concreto, independiente del uso que se le vaya a dar posteriormente a la misma.

Twitter mood light – The worlds mood in a box

Mood light está formado principalmente por un sistema *Arduino* con conexión *WiFi* y un led RGB metido dentro de un cubo de papel semitransparente.

El sistema se conecta a una red *WiFi* y realiza búsquedas dentro de la red social *Twitter* utilizando como palabras clave, frases, expresiones o palabras sueltas que indiquen contenido emocional.

El sistema, a tiempo real, hace cambiar el color del *LED RGB* dependiendo de la cantidad de *matchings* encontrados en la búsqueda.

Si hay más resultados con connotación negativa, el *LED* iluminará en rojo, si hay una gran cantidad de *Tweets* con contenido agradable, en amarillo, si ha ocurrido un suceso trágico, se iluminará de blanco...



Figura 3 : Resultado de World's Mod in a Box

Influencias al proyecto objeto:

El proyecto “World mood in a box” utiliza *twitter* y los sentimientos de sus usuarios para obtener datos sobre el estado del mundo y los sucesos que ocurren alrededor de él.

El proyecto objetivo, funciona de forma muy similar pero a pequeña escala, recogiendo los datos referentes al estado de ánimo de los usuarios y utilizándolos para estadísticas y para conocer el estado a tiempo real de un conjunto de alumnos.

	<p style="text-align: center;">DMSAD <i>Dispositivo para la medición de la satisfacción del alumno en la docencia.</i></p>	
---	--	---

4. Descripción de la situación actual

El proyecto se basa en una aplicación móvil existente utilizada con el mismo fin pero basada en el sistema *Android*.

El fin de volver a desarrollar el proyecto en una plataforma distinta (*Arduino*) es realizar un sistema físico desacoplado, único e independiente de cualquier sistema de evaluación emocional del alumno existente, pudiendo utilizar el proyecto objeto como sustituto o mejora de dichos métodos ya vigentes, que puedan generar distracciones en los sujetos a estudiar, sean más incómodos de utilizar, más caros, menos eficientes o menos ecológicos en recursos físicos.

4.1. Descripción del entorno actual

El proyecto, estando dentro del rango del estudio estadístico, repercute a la forma actual de toma de datos referentes al estado anímico de los alumnos y su opinión sobre el desempeño del profesorado, afectando de forma directa a los escasos medios para obtener determinados datos de los que consta la escuela.

En la actualidad, la evaluación de la satisfacción sobre el aprendizaje del alumno se realiza de forma muy parcial y casi inexistente mediante las encuestas repartidas por la **Escuela Técnica Superior De Ingeniería Informática**.

En cuanto al estudio de las emociones de los alumnos, no consta ninguna forma de la que se esté llevando a cabo en este momento.

Por lo que el entorno actual al que afectará el proyecto objeto es un entorno en el que no hay una competencia notable ni un gran número de sistemas externos afectados.

4.2. Resumen de las principales deficiencias identificados

DF#1: No existe un sistema de evaluación de las emociones de los alumnos en el entorno actual.

DF#2: La falta de un *feedback* fiable y rápido sobre el nivel de satisfacción en cuanto al aprendizaje del alumno.

DF#3: Ninguno de los métodos actuales permite obtener los datos objetivo a tiempo real.

DF#4: Obtener un *feedback* sobre los posibles problemas de metodología u organización de una asignatura concreta de forma continua y sencilla.

DF#5: La inexistencia de un sistema para evaluar el estado de ánimo a tiempo real de un alumno al encomendarle determinadas tareas.

DF#6: La falta de un sistema de mejora de la experiencia del alumno y asignaturas a partir de su estado de ánimo y nivel de aprendizaje. Esta deficiencia podría cubrirse de forma indirecta con el sistema desarrollado.

5. Definiciones y abreviaturas

Android : Sistema operativo para *smartphones* y *tablets* basado en el *kernel* de *Linux* que consta de gran popularidad.

Permite el uso de gran cantidad de aplicaciones gratuitas y de pago de diversos fines que pueden obtenerse a partir de la tienda online implementada en el mismo sistema operativo.

Arduino: Plataforma hardware de desarrollo en *open source* que permite realizar gran cantidad de sistemas de uso diverso.

Baudios: Unidad de medida referente al número de unidades de señal por segundo.

Biblioteca (Software): Conjunto de funcionalidades empaquetadas que pueden importarse a un proyecto para hacer uso de estas.

Bug : Defecto en el sistema que puede dar lugar a errores indeseados.

Case : Funda o carcasa.



Código fuente: Conjunto de *texto plano* que es “leído” por el ordenador para ejecutar un conjunto de funcionalidades.

Diagrama BPMN: Diagrama que permite comprender mejor el funcionamiento de un proceso funcional utilizando una notación sencilla de captar.

ETSINF: Siglas de “Escuela Técnica Superior de Ingeniería Informática”.

Feedback: Datos obtenidos como resultado de una tarea o proceso.

Hardware: Parte tangible de un sistema informático. Se trata de la electrónica del sistema.

Kanban: Tablero virtual o físico sobre el que apuntar las tareas pertenecientes a cada *sprint* del proyecto a desarrollar, sus correspondientes tareas y el estado de progreso en el que se encuentran.

Kernel : Software que constituye el núcleo de un sistema y que contiene las funcionalidades más básicas e importantes del mismo.

LCD: Sistema basado en el cristal líquido utilizado por determinadas pantallas para mostrar la información.

LED RGB : Diodo que consta de una bombilla de pequeño tamaño que se ilumina de distintos colores dependiendo de los parámetros que se le envían.

Link : Enlace que relaciona un recurso alojado en la red con otro de forma unidireccional.

Linux : Sistema operativo gratuito de *Software libre*, que puede ser modificado por cualquier desarrollador externo para adaptarlo a ámbitos concretos.

Log : Historial de acciones realizadas sobre un dato o conjunto de datos.

Matriz RACI: También conocida como **matriz de asignación de responsabilidades**.

Se utiliza para definir de que tareas se ocupará cada actor involucrado en el proyecto a desarrollar.

Popular en las metodologías ágiles.

Metadatos : Datos contenidos dentro de otros datos. Proporcionan información concreta sobre los datos a los que están enlazados, como la fecha de creación de estos, el peso, el autor...

Un símil sobre el que apoyarse, sería el de una canción (dato) y los metadatos referentes a esta: Autor, año de lanzamiento, duración, discográfica, género...

Microblogging: Servicio web que permite a los usuarios el envío y recepción de mensajes limitados por un número de caracteres predefinido.



Paradigma : Dominio que trata un determinado lenguaje de programación. Dependiendo del paradigma al que esté enfocado, un lenguaje será mejor para solucionar unos problemas determinados u otros.

Potenciómetro : Componente electrónico que actúa como una resistencia de valor variable. El valor de la resistencia puede ser modificado por el usuario al mover la rueda que lo compone.

Red social : Plataforma online que permite a los usuarios mantener contacto vía internet con gente que comparta los mismos intereses, familiares y amigos. Generalmente, permiten subir fotos y videos, compartir mensajes, crear eventos...

Refactoring : Técnica utilizada en la ingeniería del *Software* cuyo fin es mejorar la calidad y el entendimiento del código fuente sin modificar la funcionalidad original. Se logra mediante la eliminación de variables innecesarias, cambios de nombres en métodos o variables, eliminación de bloques de código o comentarios innecesarias...

Setup (Software/Hardware): Preparación del sistema para iniciar su funcionamiento.

Slider : Componente electrónico compuesto por un botón deslizante que actúa como un *potenciómetro*.

Software Libre: Un producto es de *Software libre* cuando los usuarios tienen la libertad de ejecutar, modificar, copiar, distribuir, estudiar y mejorar el software objeto.

Software: Parte intangible de un sistema informático formado por programas y rutinas que permiten al sistema realizar determinadas funcionalidades.

Sprint: Periodo de tiempo que contiene un determinado grupo de tareas a desarrollar referentes al proyecto objeto antes del fin de la fecha asociada. Utilizados en las metodologías de desarrollo ágil.

TCP: Protocolo de transmisión de datos a través de la red basado en la utilización de puertos para hacer distinción entre las aplicaciones que pueden formar parte de un sistema, que garantiza que los datos llegarán a su destino en el mismo orden que fueron enviados y sin errores.

Texto plano: Texto sin formato.

Timeline(Twitter): Página principal de la red social *Twitter* en la cual el usuario puede ver los *Tweets* enviados por la gente a la que tiene agregada a su cuenta.



Tweet : Mensaje de 140 caracteres que puede ser enviado a la web homónima o visualizado en la Timeline de los usuarios.

Puede contener imágenes, texto, video o links.

Twitter: Red social de *microblogging* basada en el envío y visualización de mensajes que pueden contener texto, imágenes o video. Los mensajes están limitados a 140 caracteres.

UDP: Protocolo de transmisión de datos a través de la red de carácter más simple que TCP que no garantiza ni la ausencia de errores ni que los datos lleguen a su destino en el mismo orden que fueron enviados

Versión Alpha: Primera versión de un producto en la que se incluyen funcionalidades básicas.

Versión Beta: Primera versión completa de un producto, que puede contener *bugs* o inestabilidad.

Versión Release: Versión de lanzamiento al público de un producto, en la que este está completo y estable.

Vúmetro : Componente electrónico formado por varios diodos *LED* situados en fila que sirve para obtener una representación visual de la variación de valores numéricos.

Wifi: Tecnología de conexión a redes de internet que prescinde de cables.

6. Requisitos iniciales

#RQI1: El prototipo se realizará sobre la plataforma *Arduino*.

#RQI2: Se imprimirá con una impresora 3D una *case* para albergar los componentes físicos del sistema.

#RQI3: Debe constar de dos *Sliders* que el usuario desplazará para obtener los valores deseados.

#RQI4: Cada *slider* representará un conjunto de valores distinto: Un *slider* el nivel de aprendizaje, el otro, el nivel de felicidad del usuario.

#RQI5: Se utilizará la red social *Twitter* para enviar los valores a la red.

#RQI6: El Tweet estará compuesto por un *hashtag* y los rangos obtenidos de los *Slider*.

#RQI7: Todos los datos se almacenarán de forma anónima en una misma cuenta de *Twitter*.

#RQI8: Los datos estarán divididos en 5 posibles valores.

#RQI9: Consta de dos *vúmetros* que servirán para dar una idea visual de los valores introducidos al usuario.

#RQI10: Cada *vúmetro* estará asociado a un *Slider* distinto.

#RQI11: Se dará una holgura de 60 segundos para dar la opción al usuario de cambiar los valores o no para cada envío.

#RQI12: Cada 60 segundos tras el envío del último *tweet*, se intentará realizar otro de forma automática solo si los rangos de los valores de uno o ambos *vúmetros* ha cambiado.

#RQI13: El lenguaje utilizado es C++.

#RQI14: Se utilizará una batería de 5 V para alimentar el sistema.

#RQI15: Se utilizará un módulo para convertir a 3.3V el voltaje para conectar al módulo *WiFi*.

7. Alcance

Para la realización del proyecto, se ha optado por hacer uso del modelo de metodología ágil, escogiendo el marco de desarrollo *SCRUM*.

Durante el ciclo de vida del proyecto, se realizaron reuniones semanales con el cliente en las que se fueron identificando las necesidades que el proyecto debía cubrir. En base a estas reuniones se fueron formando los distintos *Sprints*, cada uno referenciando a un entregable, que conforman la vida de este:

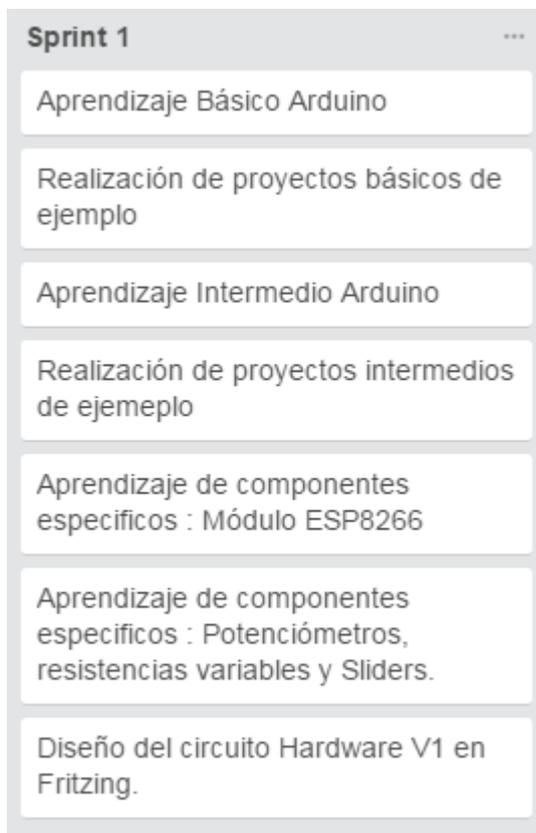


Figura 4 : Contenido Sprint 1



Figura 5 : Contenido Sprint 2 y 3



Figura 6 : Contenido Sprint 4

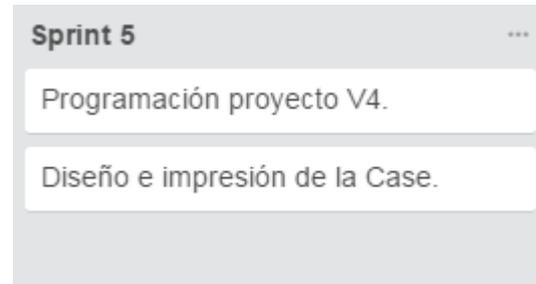


Figura 7 : Contenido Sprint 5

Entregable 1:

Primeros pasos del proyecto. En este entregable, en lo referente a *Hardware* se repasan los principios de la electrónica necesarios para poder montar circuitos con *Arduino* y componentes específicos que se utilizarán en *Sprints* futuros.

Por otra parte, se investigan las Bibliotecas y características propias del lenguaje de programación C++, en el cual se apoya la placa para la parte *Software*.

- **Aprendizaje básico Arduino:** Se realiza un aprendizaje general sobre las características de *Arduino UNO*, así como de los componentes básicos que se le pueden acoplar.
- **Realización de proyectos básicos de ejemplo.**
- **Aprendizaje intermedio Arduino:** Se realiza un aprendizaje general sobre características más específicas de *Arduino UNO*, como por ejemplo, los tipos de entradas y salidas, detalles de hardware, uso y descarga de nuevas Bibliotecas C++...

- **Realización de proyectos intermedios de ejemplo.**
- **Aprendizaje de componentes específicos. Módulo ESP8266 :** Se aprende a utilizar el componente ESP8266, un módulo *WiFi* para *Arduino*, realizando pequeños montajes con bloques de código que permiten al usuario conectarse a internet y enviar pequeños paquetes de datos de prueba.
- **Aprendizaje de componentes específicos : resistencias variables.**
Se aprende a utilizar el conjunto de elementos de las resistencias variables, enfocándonos en el *potenciómetro* rotativo, realizando pequeños montajes de *Hardware/Software* que permiten visualizar el cambio de comportamiento del circuito dependiendo del nivel de resistencia que se aplique.
Se realizan ejemplos con entradas analógicas que serán de utilidad en futuros *Sprints*.
- **Diseño del circuito Hardware V1 en Fritzing :**
Se aprende a utilizar la herramienta *Fritzing* y a diseñar circuitos con ella.
Se diseña la primera versión del circuito que será utilizada como referencia para el montaje físico en el siguiente entregable. Esta versión servirá de base e irá modificándose conforme se avance en el proyecto.

Entregable 2:

En este entregable se desarrolla una primera versión del proyecto que servirá como *Alpha*. Se pueden ir viendo las funcionalidades básicas del sistema de forma parcial, ya que serán modificadas en siguientes *Sprints* dependiendo de las necesidades del cliente.

- **Montaje de circuito Hardware:** Se monta el circuito de la versión 1 del proyecto.
Un circuito temprano que servirá para comprobar que los componentes están en armonía y trabajan juntos de forma correcta.
- **Programación proyecto V1:** Se programa la versión 1 del proyecto.
Este código realiza las funciones básicas del proyecto de forma parcial, las cuales además se verán modificadas en futuros *Sprints*.
- **Pruebas de integración:** Se realizan pruebas de funcionalidad conjunta sobre el proyecto para comprobar su correcto funcionamiento o detectar errores no depurados.

Entregable 3:

En este entregable se mejora el sistema implementado en el *Sprint* anterior. Se modifican tanto la parte *Hardware* como *Software* para disminuir la interacción necesaria entre el usuario y el sistema.

- **Modificación del circuito Hardware V1 en Fritzing: Circuito Hardware V2.**
Se modifica el circuito base diseñado en el *Sprint* 1.
Se eliminan elementos innecesarios para minimizar la interacción usuario-máquina.
- **Montaje de circuito Hardware V2:** Se monta el circuito acorde al nuevo diseño, realizado en *Fritzing* de este *Sprint*, aplicando las modificaciones y comprobando su correcto funcionamiento.
- **Programación Proyecto V2:** Se hacen modificaciones de código para que este sea acorde a los cambios en la parte *Hardware*.
- **Pruebas de integración:** Se realizan pruebas de funcionalidad conjunta sobre el proyecto para comprobar su correcto funcionamiento o detectar errores no depurados durante los entregables anteriores.

Entregable 4:

Se incorporan los últimos elementos *Hardware* al sistema y se modifica la parte *Software* para que sea acorde a este y funcione correctamente. En este punto, el proyecto está casi en su etapa final.

- **Aprendizaje de componentes específicos : Vúmetro.**
Se aprende a utilizar los *vúmetros* realizando montajes de *Hardware/Software* que permiten visualizar el cambio de comportamiento del componente dependiendo de la entrada que se le aplique.
Se realizan ejemplos utilizándolos en combinación con los *potenciómetros* rotativos.
- **Modificación del circuito Hardware V2 en Fritzing: Circuito Hardware V3.**
Se modifica el circuito resultante del *Sprint* 3, añadiendo los *vúmetros* al diseño total. Este circuito será la versión final del proyecto.

- **Montaje del circuito Hardware V3:** Se montan los vúmetros conforme al diseño especificado en *Fritzing* y se comprueba su correcto funcionamiento junto a los *potenciómetros*.
- **Programación proyecto V3:** Se hacen modificaciones de código para que este sea acorde a los cambios en la parte *Hardware*.
- **Pruebas de integración:** Se realizan pruebas de funcionalidad conjunta sobre el proyecto para comprobar su correcto funcionamiento o detectar errores no depurados durante los entregables anteriores.

Entregable 5:

En este entregable se procede a dar las últimas pinceladas al prototipo.

Se diseña una carcasa para el sistema, se mejora el código existente y se realizan pruebas de integración sobre el proyecto terminado.

- **Programación proyecto V4:** Se limpian los bloques de código innecesarios, se realizan *refactorings* sobre este y se revisan las funcionalidades existentes para minimizar los posibles errores y mejorar la experiencia del usuario.
- **Pruebas de integración:** Se realizan pruebas de funcionalidad conjunta sobre el proyecto para comprobar su correcto funcionamiento o detectar errores no depurados durante los entregables anteriores.
- **Diseño e impresión de la case:** Se diseña utilizando la herramienta online *Tinkercad* e imprime con la utilización de una impresora 3D un estuche para proteger y mantener ordenados los componentes físicos del sistema.

	<p style="text-align: center;">DMSAD <i>Dispositivo para la medición de la satisfacción del alumno en la docencia.</i></p>	
---	--	---

8. Hipótesis y restricciones

Hipótesis de partida :

La falta de un método fiable, eficiente e independiente para obtener información sobre el estado emocional de los alumnos en determinados momentos de una clase, partiendo de la existencia de un *Software* para *Android* que realiza dichas funciones, pero con el inconveniente de estar integrado en un sistema existente (*Smartphone*).

Restricciones :

Por plazo :

- Utilización de *potenciómetros* rotativos en lugar de Sliders.
- Reproducción del sistema para tener varios artículos y comprobar su funcionamiento en un entorno real.

Por Coste :

- Realizado con *Arduino UNO*, dado que es la opción más viable de desarrollar un sistema independiente y portátil.
- Utilización de *vímetros* para medir los valores introducidos en lugar de una pantalla *LCD*.
- Utilización de *potenciómetros* rotativos en lugar de Sliders.
- Reproducción del sistema para tener varios artículos y comprobar su funcionamiento en un entorno real.
- Circuitos diseñados con el apoyo de un Software Libre (*Fritzing*).

Por Calidad :

- Debe tratarse de un sistema independiente que cumpla una única función para evitar distracciones.

6/07/17	Página 24
---------	-------------

9. Estudio de alternativas y viabilidad

Alternativa	Causas de descarte
Aplicación Android	<p>-No es un sistema independiente.</p> <p>-Al estar acoplado en un dispositivo smartphone, puede causar distracciones indeseadas en el usuario (alumno).</p> <p>-No todos los usuarios disponen de un <i>Smartphone</i> con Sistema operativo <i>Android</i> o de el espacio suficiente en su dispositivo para instalarla.</p>
Aplicación Web	<p>-No es un sistema independiente.</p> <p>-Al estar alojado en la web, se debe acceder a el servicio con un dispositivo externo (móvil, portátil, <i>Tablet</i>...) lo que puede dar lugar a distracciones indeseadas en el usuario.</p> <p>-No todos los alumnos disponen de un dispositivo portátil con conexión a internet.</p>
Encuestas impresas	<p>-Gasto de papel.</p> <p>-Poco ecológico.</p> <p>-La eficiencia a la hora de recopilar los datos es menor en comparación al resto de alternativas posibles.</p>
Dispositivo independiente sin utilización de <i>Twitter</i> .	<p>-<i>Twitter</i> proporciona un <i>log</i> sobre el que obtener los <i>metadatos</i> de los datos enviados vía el sistema. En este caso, habría que crear una arquitectura de datos propia para soportar la recopilación de dicha información.</p> <p>-Menor eficiencia.</p>

Tabla 1: Alternativas al proyecto objeto.

	<p style="text-align: center;">DMSAD <i>Dispositivo para la medición de la satisfacción del alumno en la docencia.</i></p>	
---	--	---

10. Descripción de la solución propuesta

La propuesta realizada fue implementar el sistema en *Arduino*, ya que nos permite desarrollar de forma económica y eficiente un sistema independiente que permita realizar una funcionalidad determinada, evitando en medida de lo posible una interacción usuario-sistema excesiva que pueda dar pie a posibles distracciones.

Características de esta:

#CAR1: Propuesta de bajo coste económico para realizar sistemas independientes.

#CAR2: Eficiente, económico, ecológico, durable, portable, escalable.

#CAR3: Gran variedad de componentes acoplables, fácilmente obtenibles, que permiten realizar infinidad de proyectos y llevar a la realidad ideas que, sin esta opción, serían mucho más costosas económica y temporalmente de llevar a cabo.

#CAR4: El sistema está apoyado en un lenguaje de programación familiar para aquellos que ya han programado en el paradigma imperativo o en multiparadigma.

#CAR5: Solo se necesita el sistema *Arduino* con sus componentes y un ordenador para llevar a cabo el proyecto.
No son necesarias más herramientas.

#CAR6: Existe una gran cantidad de Bibliotecas, tutoriales y foros en internet sobre los que apoyarse a la hora de realizar el proyecto.

#CAR7: El uso de las redes sociales para la recopilación de datos facilita la obtención y tratamiento de estos.

#CAR8: Es sencillo hacer que la parte *Software* se comunique correctamente con los componentes *Hardware* y hacer que estos funcionen acorde a las funcionalidades definidas en el código gracias al uso de *Bibliotecas* propias de *Arduino*.

6/07/17	Página 26
---------	-------------

11. Organización y gestión del proyecto

11.1.1. Roles y responsabilidades

Para dar una idea sobre los roles y sus implicaciones en el proyecto, se ha optado por utilizar una **Matriz RACI**:

Actividad/Recurso	Cliente[Tutor]	Desarrollador [Alumno]
Análisis	R,I,C	R,A,C
Diseño	I	R,A,C
Desarrollo	I	R,A,C
Validación	R,I	R,A,C
Implementación	I	R,A,C
Pruebas	I	R,A,C

Tabla 2 : Matriz RACI Cliente-Desarrollador.

Dónde:

Responsable [R]: Actor o actores encargados de realizar la tarea asignada.

Accountable [A]: Encargado de reportar los resultados de la tarea realizada por los responsables de esta.

Consulted [C]: Actor con la información y capacidad necesarias para saber como realizar la tarea.

Informed [I]: Actor al que se le informa sobre el avance y los resultado de la tarea.

12. Planificación temporal

En este apartado se hace referencia al tiempo dedicado al proyecto y a los factores y condiciones que pueden hacer que este varíe.

El proyecto está estimado para tener una duración aproximada de un mes, aportando a continuación las **fechas** en las que serán realizadas las tareas de cada entregable:

Task name	Start date
[-] Proyecto	15/05/2017 00:
[-] Dispositivo para la medición de la satisfacción del alumno en la docencia	15/05/2017 00:
[-] Entregable1 [Sprint]	15/05/2017 00:
Reunión con el cliente	15/05/2017 21:
Aprendizaje Básico Arduino	15/05/2017 00:
Realización de proyectos básicos de ejemplo	15/05/2017 00:
Aprendizaje Intermedio Arduino	17/05/2017 00:
Realización de proyectos intermedios de ejemplo	17/05/2017 00:
Aqui se deben tener los conocimientos básicos para ir dando forma al proyecto.	19/05/2017 00:
AprendizajePotenciómetros	19/05/2017 00:
Aprendizaje ESP8266	19/05/2017 00:

Figura 8 : fechas Sprint 1.

Task name	Start date
<input type="checkbox"/> Entregable2 [Sprint]	22/05/2017 21:00
Reunión con cliente	22/05/2017 21:00
Montaje de circuito Hardware V1.	23/05/2017 00:00
Programación Proyecto V1.	24/05/2017 00:00
Pruebas	26/05/2017 00:00

Figura 9 : fechas Sprint 2.

Task name	Start date
<input type="checkbox"/> Entregable3 [Sprint]	29/05/2017 00:00
Reunión con el cliente	29/05/2017 00:00
Modificación del circuito Hardware V1 en Fritzing : Circuito Hardware V2.	29/05/2017 00:00
Montaje de circuito Hardware V2.	30/05/2017 01:00
Programación Proyecto V2.	31/05/2017 01:00
Pruebas	03/06/2017 01:00

Figura 10 : fechas Sprint 3.

Task name	Start date
<input type="checkbox"/> Entregable4 [Sprint]	06/06/2017 00:00
Reunión con el cliente	06/06/2017 00:00
Aprendizaje de componentes específicos : vúmetro	06/06/2017 00:00
Modificación del circuito Hardware V2 en Fritzing :Circuito Hardware V3.	06/06/2017 00:00
Montaje de circuito Hardware V3.	07/06/2017 00:00
Programación Proyecto V3.	08/06/2017 00:00
Pruebas	11/06/2017 03:00
+Add a task +Add a milestone	
<input type="checkbox"/> Entregable5 [Sprint]	13/06/2017 00:00
Reunión con el cliente	13/06/2017 00:00
Programación proyecto V4.	13/06/2017 00:00
Diseño e impresión de la Case.	15/06/2017 00:00
Pruebas	17/06/2017 00:00

Figura 11: fechas Sprint 4 y 5.

12.1. Cronograma temporal

Para mostrar la explicitación del plazo de ejecución del proyecto se ha optado por utilizar un cronograma Gantt, en el que podemos observar los hitos que forman parte del proyecto, el tiempo empleado en cada tarea y las dependencias entre estas :

Periodo del proyecto:	15/05/17 - 13/06/17
Hitos:	Reuniones y retos.

Tabla 3 : Fechas e Hitos.



Figura 12 : Gantt Entregable 1.

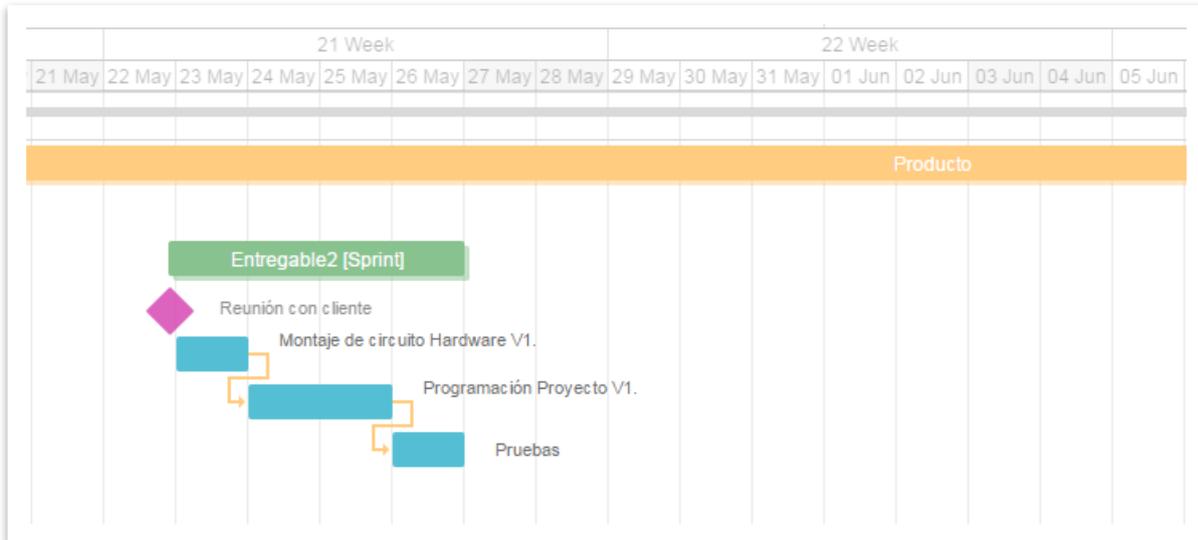


Figura 13 : Gantt Entregable 2.

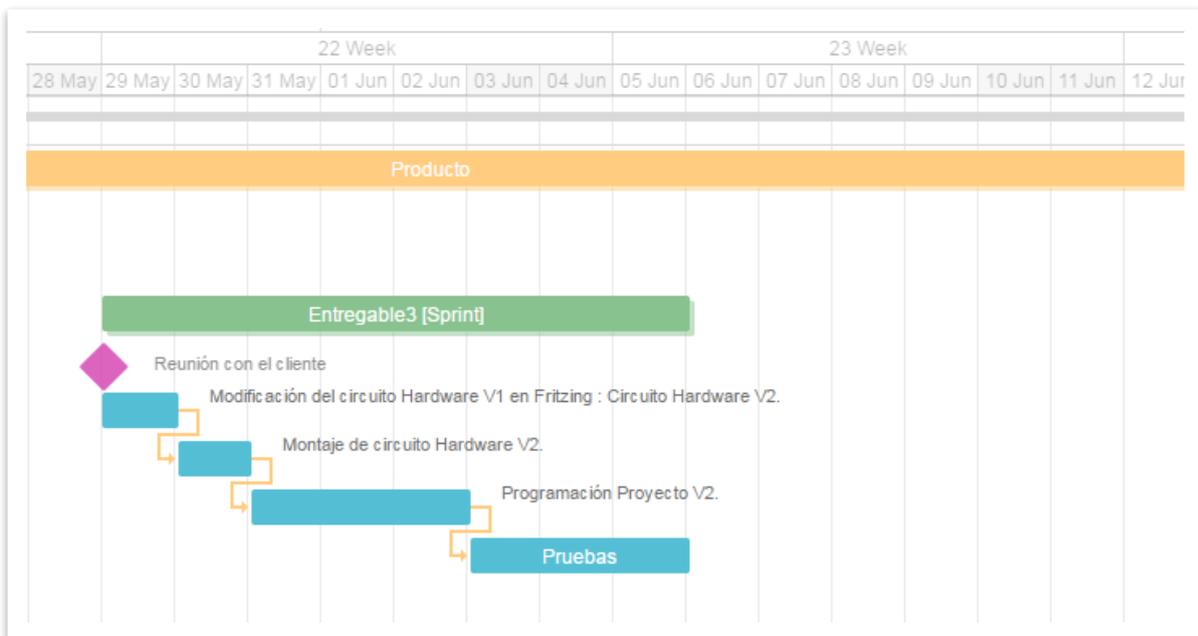


Figura 14 : Gantt Entregable 3.

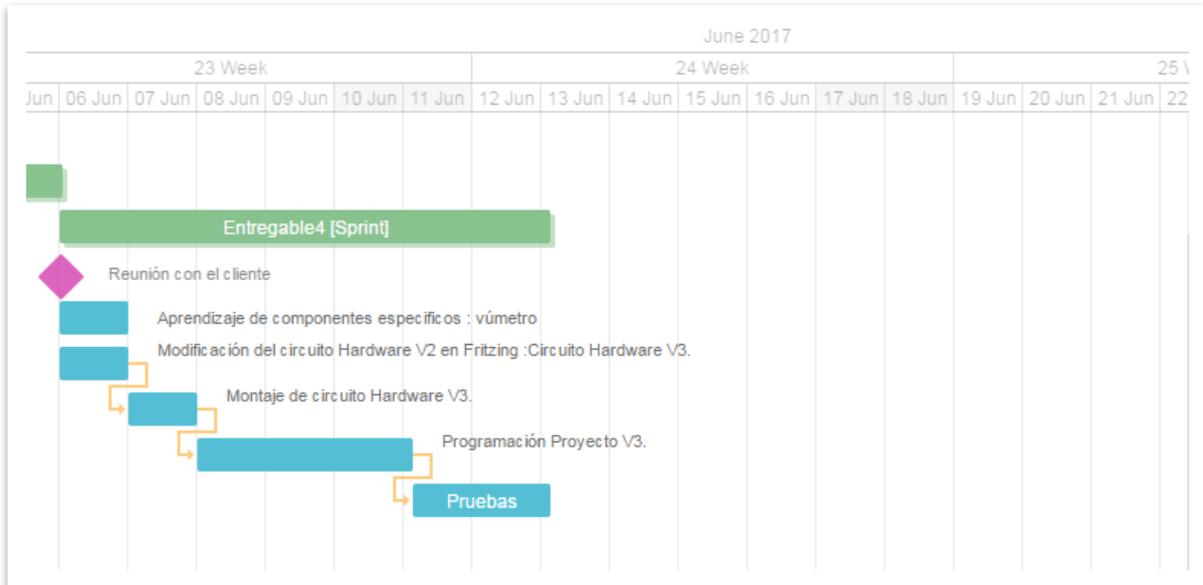


Figura 15 : Gantt Entregable 4.

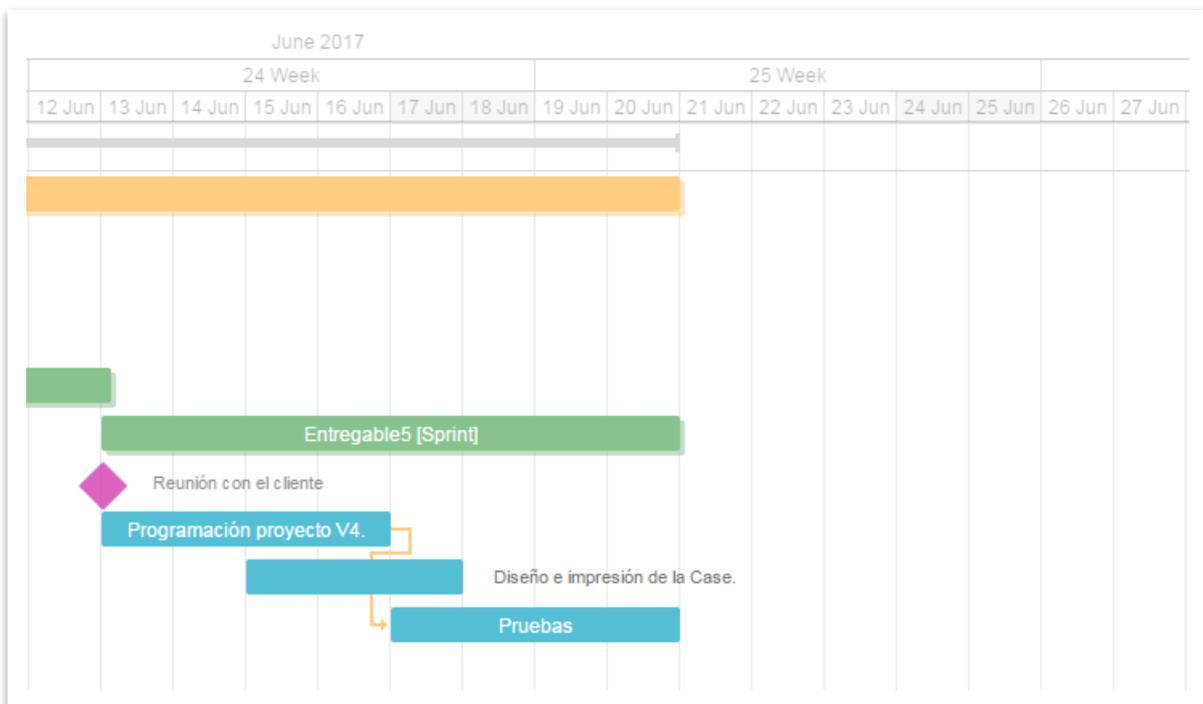


Figura 16 : Gantt Entregable 5.

12.2. Evolución del plan de proyecto

Dado que durante el ciclo de vida del proyecto pueden darse lugar a situaciones en las que la estimación inicial se vea afectada de forma negativa, se han sentado unas bases para la constatación de circunstancias de cambio en la planificación, **enumeradas a continuación:**

El criterio de actualización de la estimación temporal es diario, es decir, se harán las modificaciones temporales en el mismo día que se notifiquen los cambios, por lo que el diagrama de Gantt expuesto en el punto 13.1 es el diagrama final del ciclo de vida de nuestro prototipo.

#CRIT1: Adición de nuevas reglas que afecten al comportamiento de alguna funcionalidad del sistema.

#CRIT2: Modificación de reglas existentes que afecten al comportamiento de alguna funcionalidad del sistema.

#CRIT3: Mejoras en el sistema *Hardware*, como la introducción de un elemento que cumpla una misma función que el existente en el sistema pero que lo realice de forma más eficiente, sea más estético o se ajuste mejor al coste económico del proyecto.

#CRIT4: *Refactorings* y/o mejoras en la eficiencia del código.

#CRIT5: Cambios en el código para evitar ocupar un % dado del disponible en el *core* de Arduino Uno.

#CRIT6: Falta de elementos *Hardware* en el momento del montaje durante un determinado Sprint, lo que dará lugar a tener que esperar a la compra o recepción de estos.

#CRIT7: Ampliaciones de la funcionalidad del sistema vía *Software*.

#CRIT8: Cambios en la especificación.

#CRIT9: Errores en la fase de Análisis que desemboquen a una modificación de las bases de una determinada funcionalidad del prototipo.

13. Presupuesto

Los componentes del sistema han sido comprados a través de internet, principalmente desde proveedores como *Ebay* o proveedores privados de *Hardware* para *Arduino*.

Artículo	Descripción	Cantidad	Coste	Total por art.
Arduino UNO	Placa base sobre la que se realizará el proyecto.	1	7,42 €	7,42 €
Protoboard	Placa que permitirá conectar los elementos externos a la placa base.	1	2,55 €	2,55 €
LED bar Groove	Vúmetros para poder medir el nivel de entradas analógicas.	2	3,90 €	7,80 €
Potenciómetros	Resistencia variable para la entrada de valores analógicos.	2	1,89 €	3,78 €
Cables Arduino hembra macho x 40	Cables para conectar los distintos elementos del sistema entre el Arduino y la <i>protoboard</i> .	1	2,85 €	2,85 €
Cables arduino hembra hembra x 40	Cables para conectar los distintos elementos del sistema entre el Arduino y la <i>protoboard</i> .	1	2,37 €	2,37 €
Cables Arduino macho macho x 40	Cables para conectar los distintos elementos del sistema entre el Arduino y la <i>protoboard</i> .	1	2,25 €	2,25 €
Módulo WiFi ESP 8266	Módulo para la conexión a webs inalámbricas vía Arduino.	1	3,25 €	3,25 €
Carcasa	Impresa mediante una impresora 3D.Contendrá el sistema Hardware.	1	22,50 €	22,50 €

Total	54,77 €
--------------	----------------

Tabla 4 : Presupuesto del proyecto.

14. Especificaciones del sistema

El dispositivo para la medición de la satisfacción, cumple con una serie de requisitos funcionales y no funcionales que fueron enunciados de forma breve en el apartado 7.

En este apartado, los dividiremos en tipos y haremos uso de distintos modelos de diagrama que ayudarán a comprender el funcionamiento del sistema y cómo cubre dichos requisitos.

14.1. Requisitos no funcionales.

Eficiencia:

El dispositivo no debe tardar más tiempo del asignado por los ciclos de espera en realizar el intento de envío de un *Tweet*.

El dispositivo debe tardar menos de 5 segundos en realizar la puesta en marcha.

Los vúmetros deberán cambiar el nivel de sus *LED's* al instante tras el cambio de rango por parte del usuario.

Usabilidad:

El tiempo de aprendizaje del usuario para utilizar el sistema debe ser de menos de 10 minutos.

El sistema debe proporcionar un *feedback* continuo sobre los cambios introducidos por el usuario.

La interfaz física de usuario no debe dar lugar a confusiones, dando idea sobre que potenciómetro hace referencia a que valor.

Disponibilidad:

El dispositivo debe tener una disponibilidad del 99% cuando este esté en manos del usuario. Eso quiere decir que deberá estar preconfigurado por los responsables para estar conectado a una red *WiFi* estable siempre que los usuarios vayan a hacer uso de él.



Escalabilidad:

El sistema debe estar abierto a la inclusión de nuevas funcionalidades y componentes Hardware, es decir, el código existente no debe ocupar más del 40% de la memoria interna de Arduino Uno en ejecución y deben quedar pines de entrada y salida disponibles para un futuro.

Fiabilidad:

En caso de que el sistema falle, este debe ser capaz de poner reiniciarse manualmente por el usuario y poder continuar enviando los datos anteriores al reseteo.

Seguridad:

Los datos enviados serán anónimos, el usuario no verá comprometida ninguna información relativa a su persona.

14.2. Requisitos funcionales.

El sistema diseñado cumple una funcionalidad básica: enviar datos anónimos sobre varios usuarios a una cuenta de Twitter.

Esta funcionalidad está condicionada por varios factores vistos a continuación, los cuales forman el conjunto de los **requisitos funcionales del sistema**:

- Los datos de entrada se obtienen girando los ejes de los potenciómetros, enviando los datos a una de las entradas analógicas de Arduino Uno, conectada cada una a la patilla central de cada potenciómetro.
La patilla central es aquella que toma valores numéricos dependiendo de la variación que se haga de la resistencia interna del potenciómetro, por lo que es ideal para utilizarla como medio de entrada de datos.
- Los datos de entrada obtenidos de los potenciómetros pueden tomar un valor ajustado entre 0 y 1000. Estos valores no se verán reflejados en el estudio, solo se verá reflejado al rango que pertenecen, el cual será uno de los elementos principales del Tweet.

Los rangos vienen definidos de la siguiente forma:

$R_0 = \{0,200\}$

$R_1 = \{201,400\}$

$R_2 = \{401,600\}$

$R_3 = \{601,800\}$

$R_4 = \{801,1000\}$

- Los vúmetros incorporados en el sistema, se utilizarán para que el usuario pueda visualizar los rangos y tener una idea del valor de satisfacción y aprendizaje. Se manejarán vía software y su valor dependerá del rango de valores en el que se encuentre su potenciómetro asociado. La relación entre el rango del valor del potenciómetro y las luces activas en el vúmetro viene ilustrada en la imagen asociada.

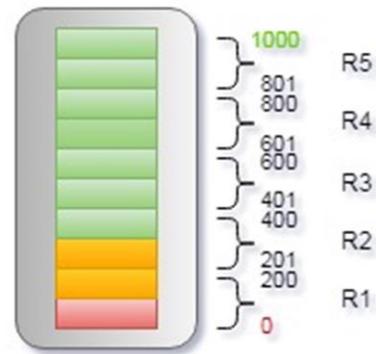


Figura 17 : Vúmetro y sus rangos.

- Los rangos obtenidos por el cambio de estado de los potenciómetros se envían a la red social Twitter para su recopilación mediante el módulo ESP8266. Estos envíos se intentan enviar en ciclos de 60 segundos.

Si el usuario no ha modificado el valor de ninguno de los dos rangos respecto al ciclo anterior, no se enviará ningún Tweet, en caso contrario, si uno de los dos rangos varia, el dato se enviará a la red, publicando un Tweet en la cuenta asociada al sistema con los datos obtenidos.

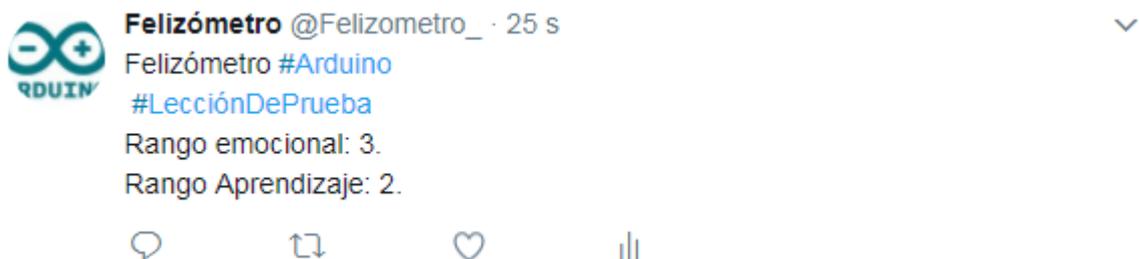


Figura 18 : Formato final de los Tweets.

14.3. Diagramas BPMN

Mediante los diagramas BPMN se pretende dar una visión simple sobre el funcionamiento del sistema y los casos de funcionamiento posibles a los que puede estar sometido. Hay un total de **dos casos** de funcionamiento posibles para el sistema:

Caso 1:

Se trata del primer envío de datos tras el encendido.

Se da una holgura de 60 segundos al usuario para poder introducir los datos deseados moviendo el eje de los potenciómetros.

En el caso 1, siempre se envían los datos a la red sin excepción alguna, ya que no ha habido un envío anterior a este y no hay un valor de rangos predecesores almacenado en el sistema.

Una vez el caso 1 llega a su fin, se da pie al caso 2, el cual se realizará de forma cíclica hasta que el sistema sea apagado o reseteado.

Este caso viene ilustrado mediante el siguiente diagrama:



Figura 19 : Diagrama BPMN - Caso 1.

Caso 2:

El caso 2 hace referencia a la naturaleza cíclica del sistema.

En este caso funcional, el sistema ya contiene en memoria los rangos que han sido enviados con el envío obligatorio del caso 1, por lo que hasta que estos no sean cambiados, no enviará datos a la red.

Si al menos uno de los dos potenciómetros ha sido movido lo suficiente como para cambiar el rango de valores en el que se encuentra, se enviará un Tweet al final del ciclo de holgura de 60 segundos que se le proporciona al usuario para el envío de datos.

Tras el intento de envío, sea este favorable o no, se dará lugar a un nuevo ciclo de espera.

En caso contrario, el sistema seguirá esperando a que los valores cambien, manteniéndose pasivo, hasta que al menos uno de los dos rangos vuelva a ser modificado para enviar datos al servidor en el siguiente ciclo.

Este caso viene ilustrado mediante el siguiente diagrama:

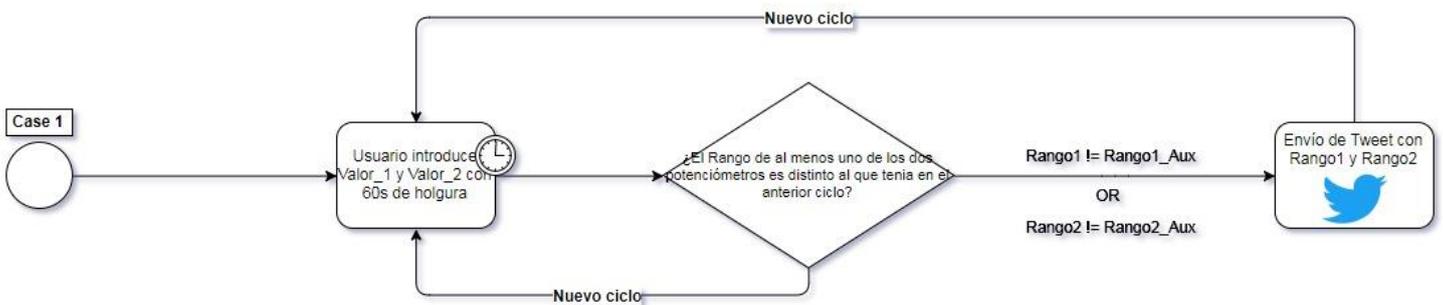


Figura 20 : Diagrama BPMN - Caso 2.

14.4. Diagrama de secuencia

En esta sección se presenta un diagrama de secuencia acorde a los requisitos funcionales especificados en secciones anteriores y al funcionamiento general del sistema.

En este diagrama, se muestra como interactúa el usuario con el sistema para dar lugar a las entradas de datos, y de forma similar, como actúan los distintos componentes del sistema entre sí y bajo que condiciones los intentos de envío de Tweet serán satisfactorios.

Todo esto ya ha sido explicado en apartados anteriores, no obstante, el diagrama de secuencia sirve como refuerzo para el entendimiento del funcionamiento interno del sistema sin entrar en demasía en detalles específicos.

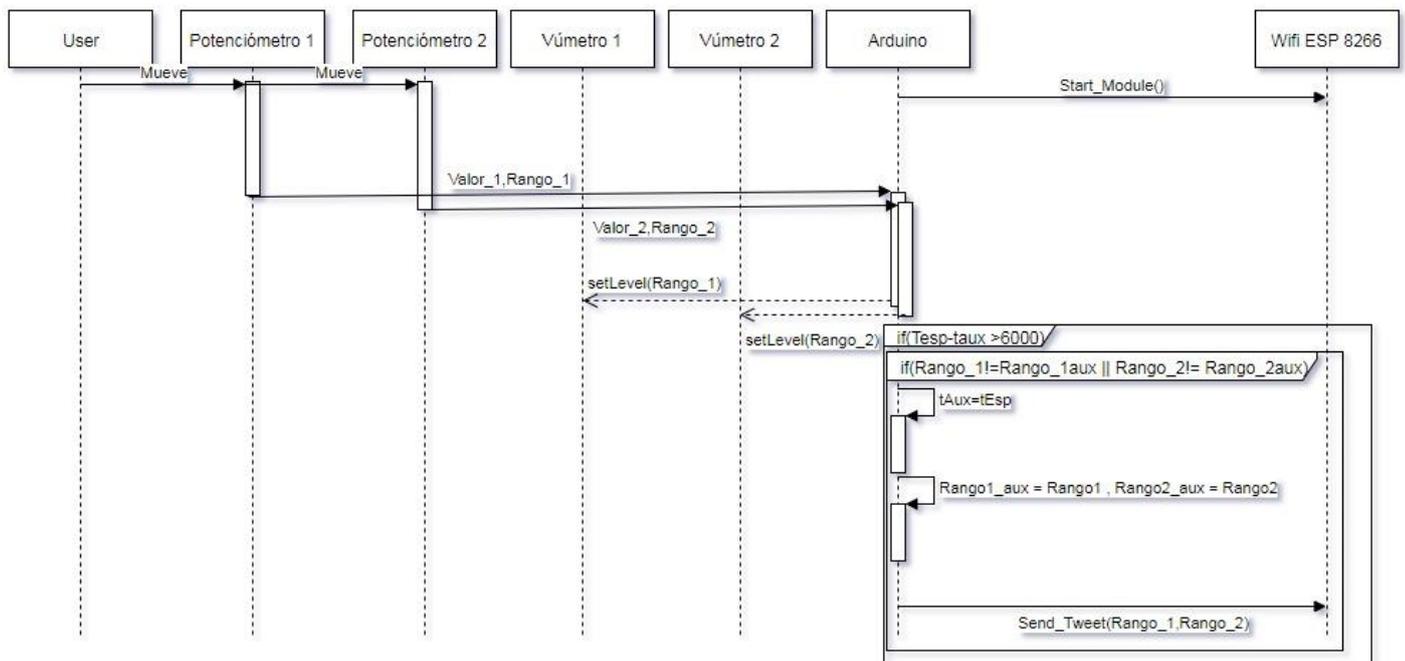


Figura 21 : Diagrama de secuencia del prototipo.

15. Marcos y diseño.

15.1. Marco Hardware

En este apartado haremos referencia a los componentes *Hardware* que conforman la totalidad del proyecto, sus especificaciones y su utilización en el prototipo, así como los diseños del circuito y la evolución de estos con el avance en los entregables.

15.1.1. Componentes del sistema

Arduino UNO rev3

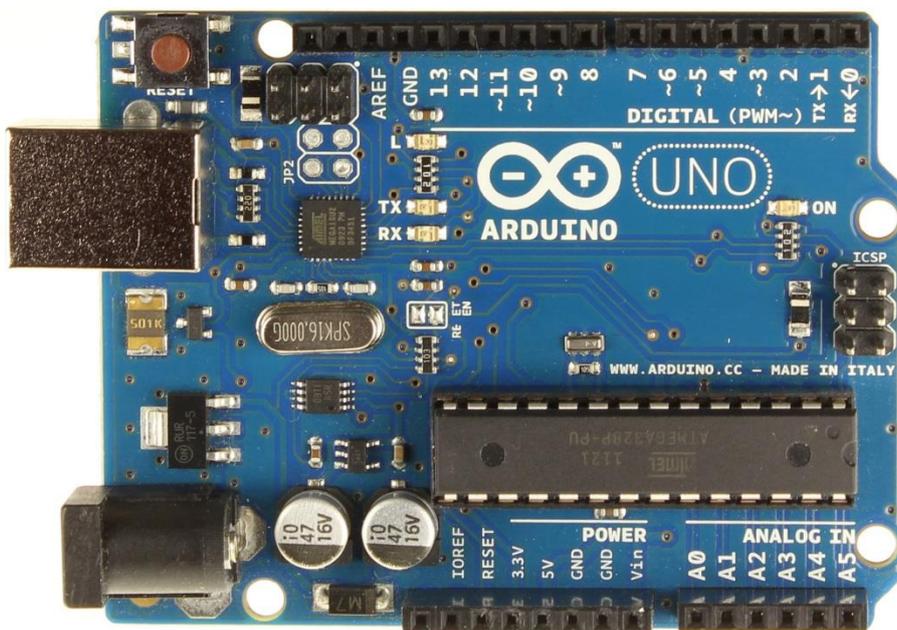


Figura 22 : Placa Arduino UNO

El Arduino UNO es el componente base del sistema. En él, se conectarán el resto de componentes *Hardware*, permitiéndonos además definir el comportamiento de estos asignándole rutinas *Software* a través de su IDE, siendo así el núcleo del proyecto.

Consta de un circuito impreso con su respectivo microcontrolador el cual permitirá ejecutar las ordenes que se le envíen al sistema, puertos de alimentación para componentes y sus puertos de entrada y salida tanto analógicos como digitales, que permiten conectar los distintos componentes que forman parte del proyecto e interactuar con ellos, definiendo su comportamiento a través del Software.

Estructura

Incluidos únicamente aquellos componentes que han sido utilizados en el proyecto.

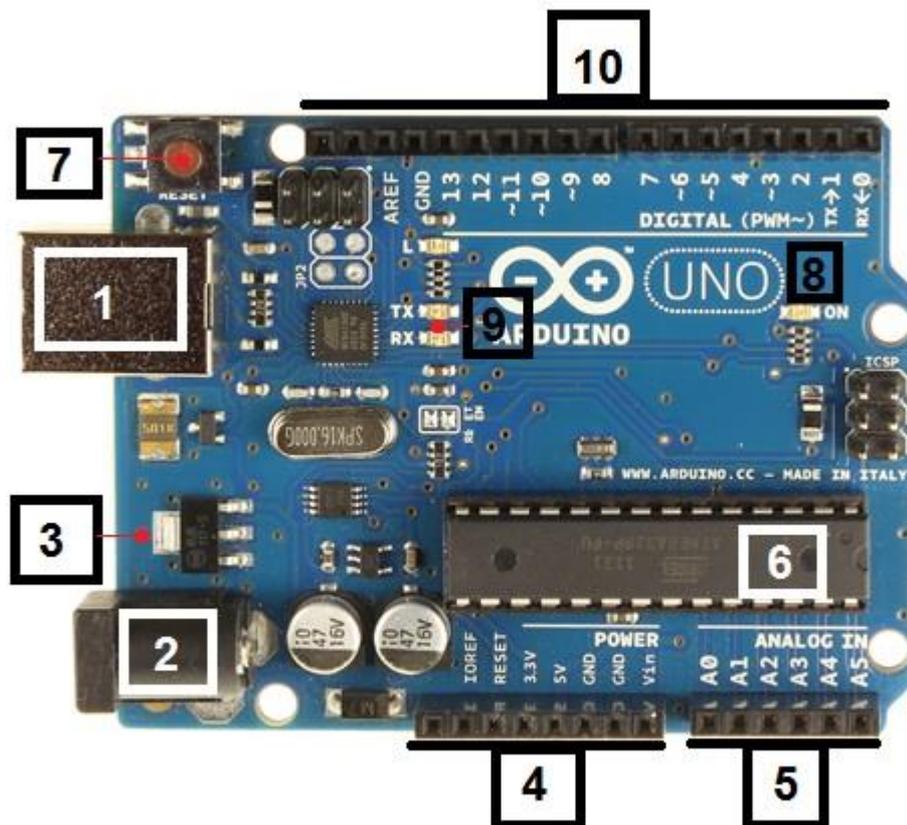


Figura 23 : Placa Arduino UNO y componentes.

1. Conector USB: Permite conectar la placa con el PC para acceder al modo de programación y depuración, además de servir como toma de alimentación, la cual admite una entrada de 5V, la proporcionada por cualquier toma USB de un PC.
2. Plug de conexión para fuentes externas.
3. Regulador de voltaje: Es el encargado de transformar el voltaje de entrada que llega desde el *plug* de conexión para fuentes externas (3), regulándolo a una entrada de 5V.
4. Conexiones Power: Consta de 6 pines de distinto cometido:
 - RESET: Permite resetear el microcontrolador cuando se le envía el valor lógico 0.
 - 3.3V : Para conectar componentes Hardware que no funcionen al voltaje estándar de Arduino UNO (5V), como por ejemplo el módulo *WiFi* ESP8266, del que se hablará más adelante.
 - 5V : Para conectar componentes Hardware que funcionen con el voltaje estándar de Arduino UNO.
 - GND (x 2): Tomas de tierra (0V).
 - Vin: Toma conectada al *plug* de conexión para fuentes externas (3). Utilizado para conectar la alimentación de la placa a la de una fuente externa.
5. Entradas analógicas: Puertos utilizados para recibir la salida de los componentes conectados a ellas. Admite un valor entre 0V y 5V.
6. Microcontrolador de Arduino UNO. Modelo Atmega 328: Se trata del núcleo de Arduino UNO, ya que contiene la CPU y la memoria del sistema, permitiendo ejecutar las ordenes que se le envíen vía *Software*.
7. Botón Reset: Permite resetear el sistema, inicializando de nuevo el programa que se está ejecutando.
8. LED de encendido.
9. LED's de recepción y envío: Se iluminan cuando la placa intercambia datos con el PC.
 - Tx: Transmisión de datos.
 - Rx: Recepción de datos.
10. Entradas y salidas digitales: Permite conectar componentes externos y utilizarlos como periféricos de entrada o salida de datos, dependiendo de lo que se defina vía *Software*.

Protoboard

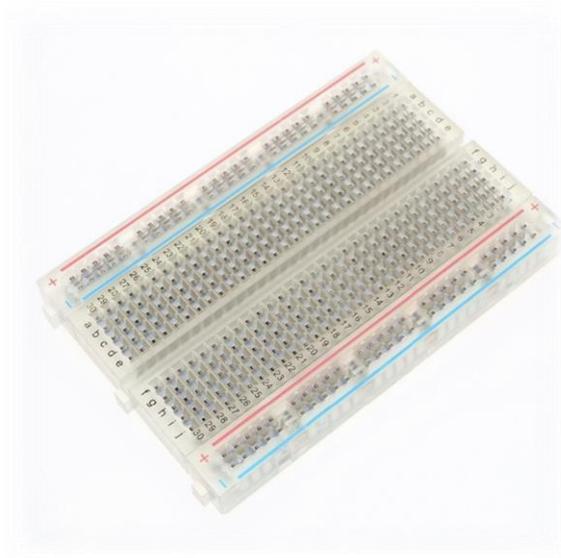


Figura 24 : Protoboard.

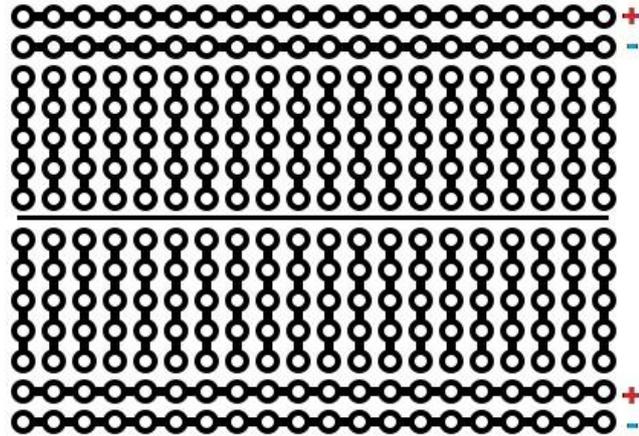


Figura 25 : Conexiones Protoboard.

La *Protoboard* nos permite conectar componentes electrónicos a ella para posteriormente, conectarlos a las salidas y entradas que la placa *Arduino UNO* nos ofrece.

Es de gran utilidad en proyectos medianos y grandes ya que facilita en gran medida la conexión de los *periféricos*, así como su alimentación, permitiéndonos además mantener más ordenados los componentes y cables.

Los orificios de la *Protoboard* están conectados entre sí para permitir la comunicación entre los distintos elementos que se le acoplen minimizando el uso de cables.

Generalmente, dicha interconexión sigue el patrón definido en la figura, dónde las líneas referentes a las tomas de alimentación y tierra están conectadas en horizontal y las líneas de los bloques de conexión de elementos *Hardware* en vertical.

Potenciómetro



Figura 26 : Potenciómetro rotativo.

Los potenciómetros se tratan de resistencias variables que vienen en diversos formatos. Los utilizados en el proyecto son los potenciómetros rotatorios, que permiten hacer variar el valor de su resistencia interna girando la rueda que lo compone.

En cuanto a la estructura y uso, las patillas A y B se utilizan para conectar a la alimentación y a masa, mientras que la patilla C es la utilizada para conectar el resto del circuito objetivo, ya que es la patilla de salida del valor de la resistencia variable.

En nuestro proyecto, estos se utilizan asociando la patilla de salida C a las entradas analógicas para obtener el valor de la variación del voltaje y ajustarlo a una entrada numérica representativa, ya que el fin con el que los vamos a utilizar no va a ser el de utilizarlos como resistencia variable, si no como componente de entrada de datos.

Led Bar Grove

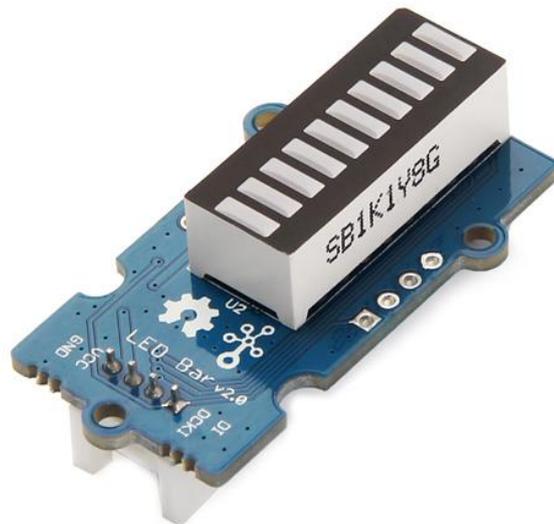


Figura 27 : Led Bar en su placa.

Para que el usuario pueda obtener un *feedback* sobre el valor que ha introducido al girar el eje de los potenciómetros y obtener así una idea de en que rangos se encuentran dichos valores, se utilizarán dos barras de luces *LED* a modo de *vúmetros*.

Las barras escogidas son las Grove LED Bar, proporcionadas por la marca Seed. Las *Grove LED Bar*, están divididas en 10 regiones independientes de *LEDs*, cuyo encendido puede configurarse de forma individual dependiendo de las necesidades que deban cubrir con la ayuda de una Biblioteca proporcionada por el proveedor del componente.

Las LED bar se puede alimentar con un voltaje que va desde los 3 hasta los 5.5 voltios, conectándose además de forma fácil a la placa Arduino, pudiendo utilizar un adaptador que el propio proveedor facilita o con cables macho – hembra.

Módulo Wifi ESP8266

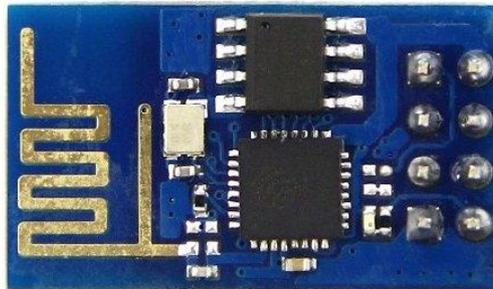


Figura 28 : Módulo ESP 8266

El módulo ESP8266 será el encargado de permitir que el sistema se conecte a internet, en este caso, a través de *Wifi*, junto a las comodidades que esto supone. Este componente, soporta los protocolos 802.11b,802.11g,802.11n y es capaz de conectarse a una red Wifi que esté a una distancia de hasta 91 metros.

Para operar con él, es necesario conectarlo a una toma de 3.3 Voltios como máximo, de la que *Arduino UNO* dispone.

En cuanto a las conexiones y configuraciones, el módulo permite utilizar los protocolos *TCP* y *UDP* para comunicarse, y podemos utilizarlo con configuración de usuario, de punto de acceso o ambas a la vez.

En nuestro proyecto, utilizaremos únicamente la conexión de cliente, ya que vamos a utilizar el módulo para enviar los datos obtenidos por el sistema a la red, en concreto a la red social Twitter.

15.2. Marco Software

En esta sección hablaremos sobre el apartado *Software* del proyecto, es decir, en cómo se ha implementado la funcionalidad para cubrir la especificación, en mostrar bloques de código de interés y en cómo se ha enlazado esta con el *Hardware*, mediante el uso de Bibliotecas específicas.

15.2.1. Bibliotecas

Para la implementación *Software* del proyecto se han utilizado dos Bibliotecas externas: <Grove_Led_Bar.h> y <SoftwareSerial.h>, cada una con un cometido distinto, que anunciaremos a continuación.

<SoftwareSerial.h>

La Biblioteca *Software Serial* se utiliza para replicar la funcionalidad de los puertos Rx y Tx de la placa Arduino.

Estos puertos son utilizados de forma específica para la Recepción (Rx) y envío (Tx) de datos a la placa.

La utilización de esta Biblioteca puede ser de gran utilidad a la hora de aumentar la escalabilidad del sistema, ya que permitirá más conexiones Rx/Tx de las originales.

En nuestro proyecto, la utilizamos por comodidad para conectar el módulo ESP8266 y definir que pines harán de Tx y Rx, dándonos así la ventaja de que si en un futuro se decidiera ampliar el proyecto, no estaremos limitados por el número de pines de este tipo que ofrece Arduino Uno (únicamente uno de cada tipo).

Simplemente con añadir en el *setup* del programa la instrucción (Aplicada a los pines Tx y Rx del módulo *Wifi*) :

```
SoftwareSerial esp8266(3, 2);
```

La placa entenderá que el puerto 3 hará de puerto de recepción Rx, y el puerto 2 de transmisión Tx.

<Grove_Led_Bar.h>

La Biblioteca Grove Led Bar se utiliza para poder inicializar el funcionamiento y definir el comportamiento del componente hardware homónimo.

Esta Biblioteca contiene, entre otros, los métodos necesarios para poder modificar el nivel de encendido de leds mostrado en el componente, funcionalidad principal de Biblioteca que utilizamos en nuestro proyecto.

-Asignación de los pines:

```
Grove_LED_Bar(unsigned char pinClock, unsigned char pinData, bool greenToRed);
```

-Inicialización del led bar:

```
void begin(){pinMode(__pinClock, OUTPUT); pinMode(__pinData, OUTPUT);}
```

-Definir la orientación de los colores del led bar:

```
void setGreenToRed(bool greenToRed);
```

-Definir en nivel en el que se sitúa el Led Bar (Método primordial en el Proyecto) :

```
void setLevel(float level);
```

-Definir el nivel de brillo de uno de los leds que componen la barra:

```
void setLed(unsigned char led, float brightness);
```

-Activar un Led específico:

```
void toggleLed(unsigned char led);
```

-Activar un conjunto de Leds acorde al valor que se le envíe:

```
void setBits(unsigned int bits);
```

-Obtener el estado actual del led bar:

```
unsigned int const getBits();
```

15.2.2. Implementación

En este apartado, haremos una visita a la implementación *Software* del sistema, veremos su relación con el marco *Hardware* y mostraremos los bloques de código de mayor interés para el facilitar la comprensión del prototipo.

Para mejor entendimiento de este cometido, nos apoyaremos en el diagrama de secuencia visto en el apartado 14.4 de este documento

Setup e inicialización del sistema:

En este bloque se incluyen las *Bibliotecas* mencionadas en el apartado anterior, se enlazan los pines a las entradas y salidas de los componentes *Hardware* y se definen variables necesarias para el control de los envíos de *Tweets* (Más adelante se verá con mayor profundidad).

```
#include <Grove_LED_Bar.h>
#include <SoftwareSerial.h>

static long lastUpdate=0; //Para ciclos de intento de envío de Tweet
int diferencia=0; //Para operaciones de ajuste de valores de entrada analogicos (A0).
int diferencia2=0; //Para operaciones de ajuste de valores de entrada analogicos (A1).

//Para medir el tiempo de envío
unsigned long tEnv;

//Rangos temporales para las secciones de los vómetros
int rangoT1;
int rangoT2;

//Rangos Finales para las secciones de los vómetros
int rangoF1;
int rangoF2;

SoftwareSerial esp8266(3, 2); //Rx y Tx para esp8266

Grove_LED_Bar bar(9,8,0); //Pin_Clock, Pin Data
Grove_LED_Bar bar2(11,10,0); //Pin_Clock, Pin Data
```

Constantes:

Este bloque se utiliza para definir aquellos valores que no cambiarán su valor durante la utilización del sistema, como el token *teamspeak* para conectarnos a la cuenta de *Twitter* y enviar los datos a esta, la dirección IP de *teamspeak*, el puerto de escucha para nuestro módulo *Wifi* y los datos de la Red a la que nos vamos a conectar.

```
#define APIKEY "HWHSUAYAJ3BY1VYS" //Token de la cuenta Twitter.

#define TARGET_IP "184.106.153.149" //Direccion IP thingspeak

#define TARGET_PORT "80" // Puerto escucha

#define ID "Red_Wifi_de_Ejemplo" //SSID Red
#define PASS "Contraseña_de_Ejemplo" // Pwd Red
```

Bloque setup:

En el bloque setup se inicializa la placa y se define a cuantos *baudios* funcionará el sistema, además, configuramos el módulo ESP8266 para que actúe en modo cliente y lo conectamos a la red wifi definida en las constantes.

```
void setup()
{

  Serial.begin(115200);
  esp8266.begin(115200);

  //Inicialización del módulo WiFi.

  sendData("AT+RST\r\n",1000,DEBUG); // Reseteo inicial.

  sendData("AT+CWMODE=1\r\n",1000,DEBUG); // Modo cliente (1)

  String cmd="AT+CWJAP=\""; // Comando de conexión a red WiFi
  cmd+=ID;
  cmd+="\", \"";
  cmd+=PASS;
  cmd+="\"";
  sendData( cmd+"\r\n",1000,DEBUG); // Envio la cadena completa para conexión
  sendData("AT+CWMUX=0\r\n",1000,DEBUG); // Habilitamos una única conexión
  delay(3000);
}
```

Bloque Loop:

Inicialización de ciclo:

Se asigna una cadena de texto con la plantilla para los tweets a la que posteriormente se le sumarán los valores de los rangos obtenidos por los potenciómetros.
 Por otra parte, se lee el valor de los potenciómetros y se reajusta su valor mínimo y máximo para que sean representativos.

```
void loop()

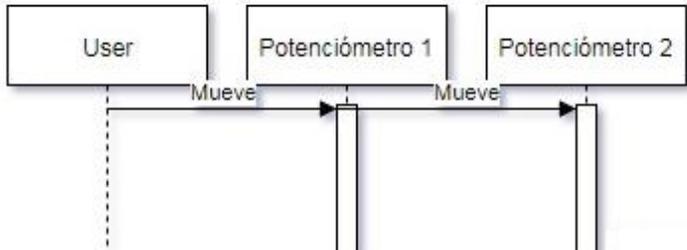
String updateTwitterStatus = ("Felizómetro #Arduino\n #LecciónDePrueba\n");
tEnv=millis(); //Se inicia el ciclo de holgura de 60 segundos para que el usuario tenga opción a mover los potenciómetros.
int valor_sensor = analogRead(A0); //lectura del valor analógico del potenciómetro1
int valor_sensor2 = analogRead(A1); //lectura del valor analógico del potenciómetro2

//Se reajusta el valor de los potenciómetros, ya que la entrada va comprendida entre el conjunto de valores [0,1023]
//como estos valores no son representativos, reajustamos a [0,1000]

if(valor_sensor>1000){
diferencia=valor_sensor-1000;
valor_sensor=valor_sensor-diferencia;
}

if(valor_sensor2>1000){
diferencia2=valor_sensor2-1000;
valor_sensor2=valor_sensor2-diferencia2;
}


```



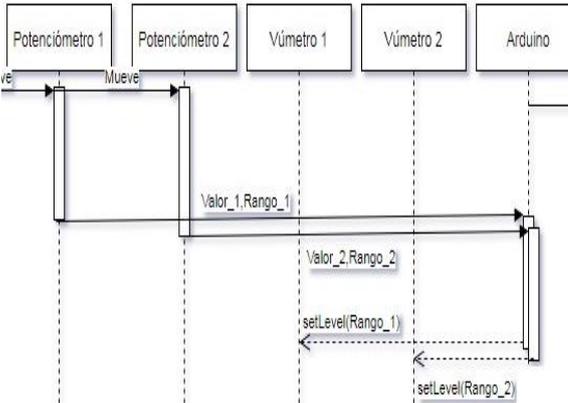
The diagram shows a User object sending 'Mueve' messages to Potenciometro 1 and Potenciometro 2 objects.

Definición de rangos :

Se definen los rangos en base al valor obtenido de los potenciómetros y se modifica el nivel en el que se encuentran los vúmetros en definición a estos :

```
if(valor_sensor>0 && valor_sensor<200){
bar.setLevel(2);
rangoI1 = 0;
}
if(valor_sensor>201 && valor_sensor<400){
bar.setLevel(4);
rangoI1 = 1;
}
if(valor_sensor>401 && valor_sensor<600){
bar.setLevel(6);
rangoI1 = 2;
}
if(valor_sensor>601 && valor_sensor<800){
bar.setLevel(8);
rangoI1 = 3;
}
if(valor_sensor>801 && valor_sensor<1000){
bar.setLevel(10);
rangoI1 = 4;
}


```



The diagram shows Potenciometro 1 and Potenciometro 2 sending 'Mueve' messages to Vúmetro 1 and Vúmetro 2. Vúmetro 1 sends 'Valor_1.Rango_1' and Vúmetro 2 sends 'Valor_2.Rango_2' to Arduino. Arduino then sends 'setLevel(Rango_1)' and 'setLevel(Rango_2)' back to Vúmetro 1 and Vúmetro 2 respectively.

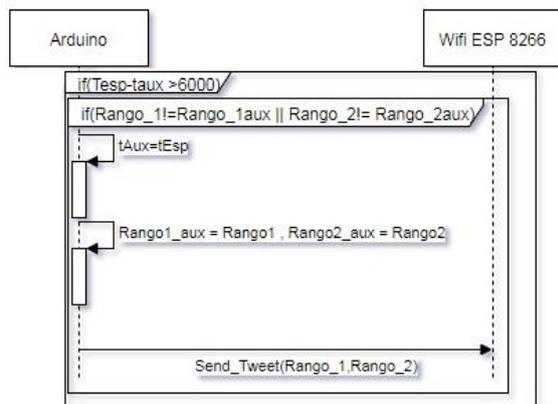
Final del ciclo de holgura:

Cuando se alcanza el final del ciclo en el que nos encontramos, el sistema reajusta el valor temporal para el inicio de un nuevo ciclo, restando el tiempo transcurrido en el último ciclo al tiempo total de ejecución y actualizando el tiempo transcurrido para poder restarlo al siguiente ciclo e iniciar uno nuevo de 60 segundos. Por otra parte, se comprueba si el rango de al menos uno de los dos potenciómetros ha cambiado durante la holgura.

```

if(tEnv - lastUpdate > 60000){
  lastUpdate = tEnv;
  if(rangoF1!=rangoT1 || rangoF2!=rangoT2){
    rangoF1=rangoT1;
    rangoF2=rangoT2;
  }
}

```



Tras el final del ciclo, si al menos el rango de uno de los dos potenciómetros ha cambiado, se da la orden de actualizar la cadena base del contenido del Tweet:

```

updateTwitterStatus+="Valor emocional: ";
updateTwitterStatus+=RangoF1;
updateTwitterStatus+=".\n";
updateTwitterStatus+="Valor Aprendizaje: ";
updateTwitterStatus+=RangoF2;
updateTwitterStatus+=".";

```

Y se procede a enviar los comandos correspondientes al módulo *WiFi* para que se realice la conexión a la red social y se envíen los paquetes de datos utilizando el protocolo *TCP* :

```
String webpage = "AT+CIPSTART=\\"TCP\\",\\"";
webpage += TARGET_IP;
webpage += "\",80\r\n";
sendData (webpage,1000,DEBUG);
```

//Cadena para conectar con thingspeak y obtener enviar datos con protocolo HTTP

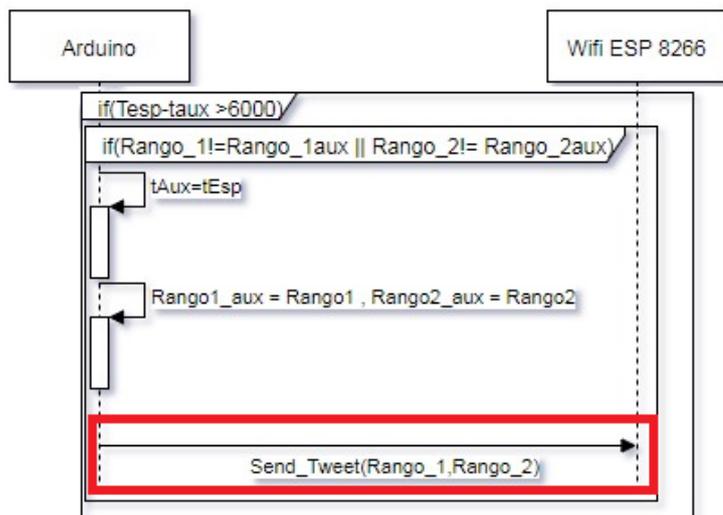
```
String tsData = "api_key=" APIKEY "&status=" + updateTwitterStatus ;
```

```
String webpage1 = "POST /apps/thingtweet/1/statuses/update HTTP/1.1\n";
webpage1+="Host: api.thingspeak.com\n";
webpage1+="Connection: close\n";
webpage1+="Content-Type: application/x-www-form-urlencoded\n";
webpage1+="Content-Length: ";
webpage1+=tsData.length();
webpage1+="\n\n";
webpage1+=tsData;
```

```
//Envío de la cadena POST
String cipsend = "AT+CIPSEND=";
cipsend+= webpage1.length();
cipsend+="\r\n";

sendData (cipsend,1000,DEBUG);
sendData (webpage1,1000,DEBUG);

sendData ("AT+CIPCLOSE=0\r\n",1500,DEBUG);
//cierre de conexión
```



15.3. Versiones, diseños del sistema y evolución del prototipo

Para dar una idea de la evolución del proyecto y los cambios en la funcionalidad y la usabilidad, haremos uso de los esquemas electrónicos diseñados con la herramienta *Fritzing*, adjuntado a cada uno una idea general del funcionamiento de cada uno.

Cada diseño está sujeto a uno o varios entregables del proyecto.

15.3.1. Versión alfa 1.0

Referente a los Entregables 1 y 2.

La versión más temprana del proyecto, realizada como alfa para realizar pruebas del funcionamiento conjunto de los componentes.

Esta versión necesitaba de mayor interacción usuario-sistema dado que los Tweets no se enviaban automáticamente, sino al pulsar sobre el botón que se ve en la imagen. Enviaba tweets simples que constaban del valor obtenido del potenciómetro, sin automatizar con ciclos temporales como en versiones posteriores del proyecto.

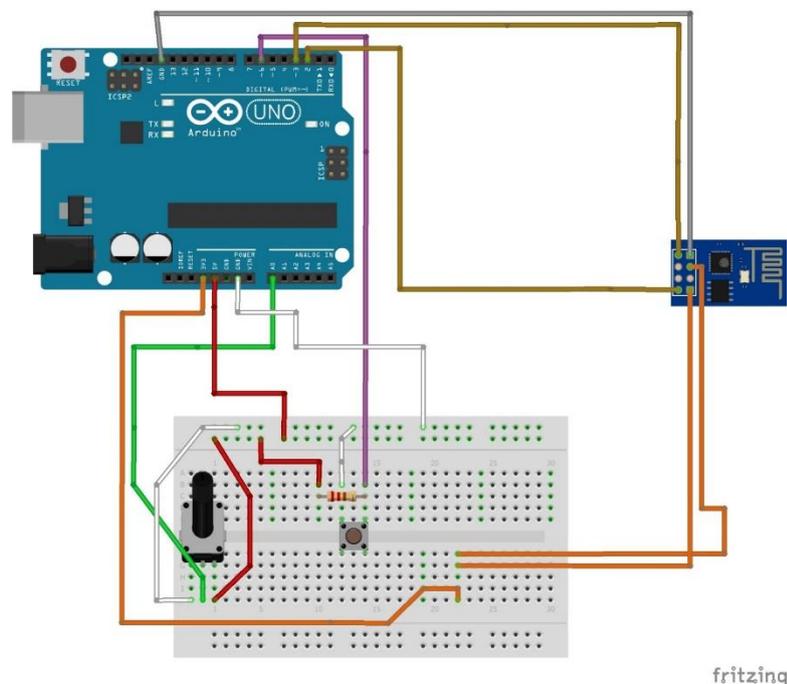


Figura 29 : Circuito prototipo 1.

15.3.2. Versión alfa 2.0

Referente al Entregable 3.

La versión más temprana del proyecto, realizada como alfa para realizar pruebas del funcionamiento conjunto de los componentes.

Esta versión necesitaba de mayor interacción usuario-sistema dado que los Tweets no se enviaban automáticamente, sino al pulsar sobre el botón que se ve en la imagen.

Enviaba tweets simples que constaban del valor obtenido de los potenciómetro, sin automatizar con ciclos temporales como en versiones posteriores del proyecto.

La única diferencia entre esta y la primera versión es que esta constaba de dos potenciómetros, además de que se pulió el diseño software, realizando *refactorings* sobre el código y eliminando bloques de prueba que quedaron como residuo del primer entregable.

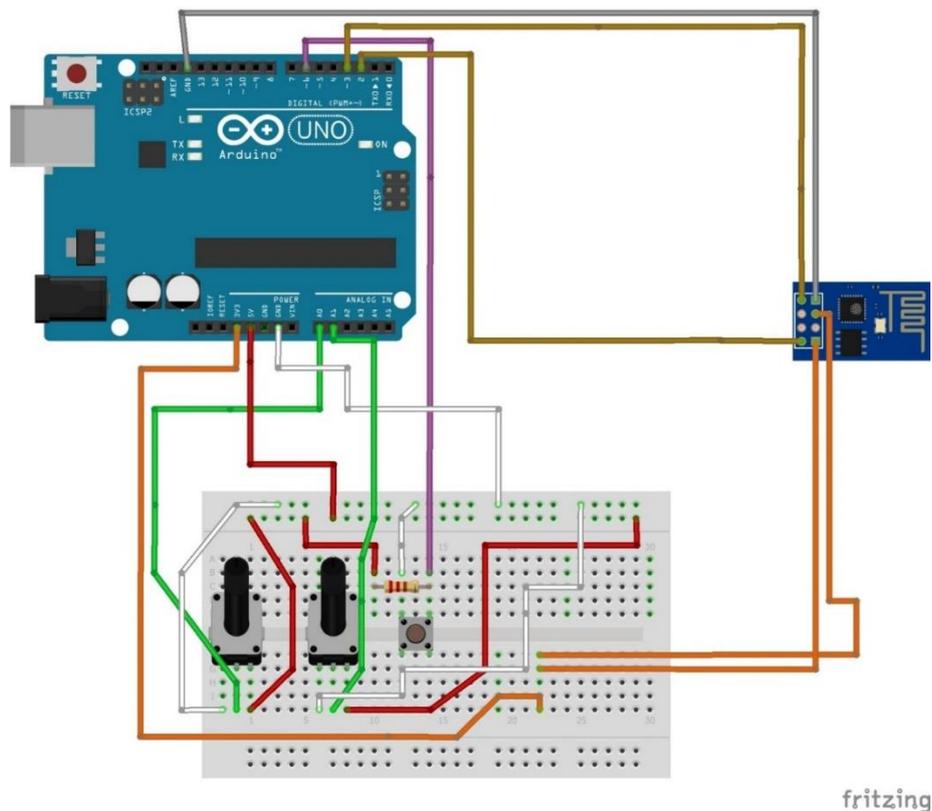


Figura 30 : Circuito prototipo 2.

15.3.3. Versión Beta 1.0

Referente al Entregable 4.

Versión intermedia del proyecto, que ya cumple con la funcionalidad final de este, pero con la ausencia de los vúmetros y sin pulir los nuevos cambios de software realizados (*refactorings*).

En esta versión ya se han implementado los rangos para poder realizar así el control de envío automático por ciclos.

Esta versión desacopla la interacción usuario-sistema dado que los Tweets se envían automáticamente, en ciclos de 60 segundos, enviándose si y solo si el usuario ha hecho cambiar el valor de al menos uno de los potenciómetros lo suficiente como para hacerlo variar de rango.

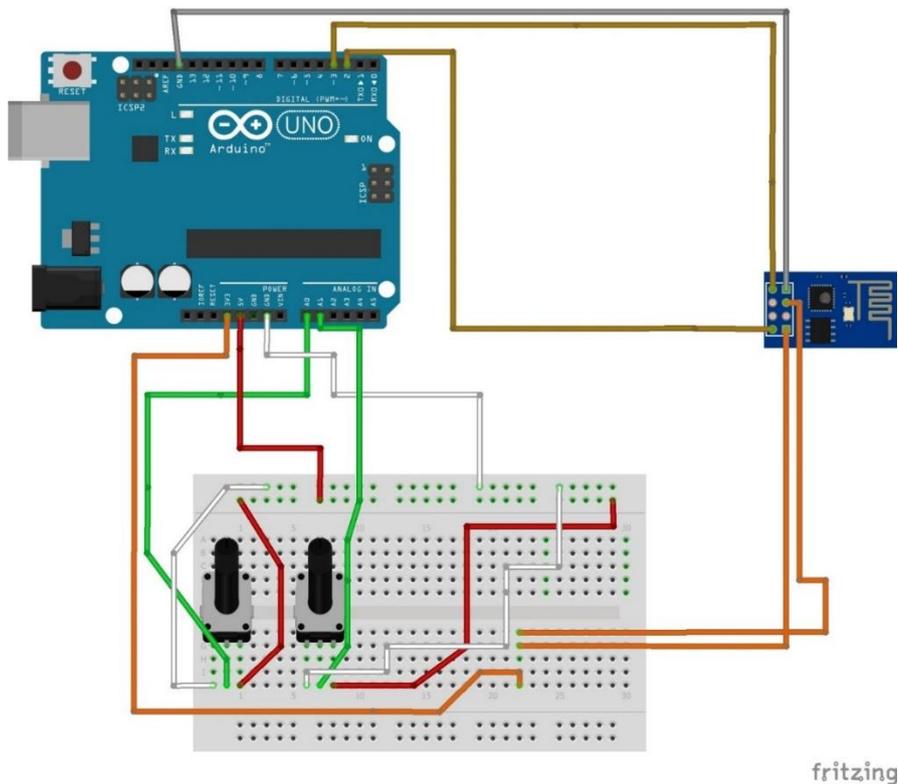


Figura 31 : Circuito prototipo 3.

15.3.4. Versión release

Referente al entregable 5.

Versión final del prototipo, que ya cumple con las especificaciones y requisitos objetivo.

En esta versión ya se han implementado los rangos para poder realizar así el control de envío automático por ciclos y desacopla la interacción usuario-sistema dado que los Tweets se envían automáticamente, en ciclos de 60 segundos, enviándose si y solo si el usuario ha hecho cambiar el valor de al menos uno de los potenciómetros lo suficiente como para hacerlo variar de rango.

Además, se han incluido dos vúmetros para dar *feedback* al usuario sobre los rangos de valores en los que se encuentra cada potenciómetro:

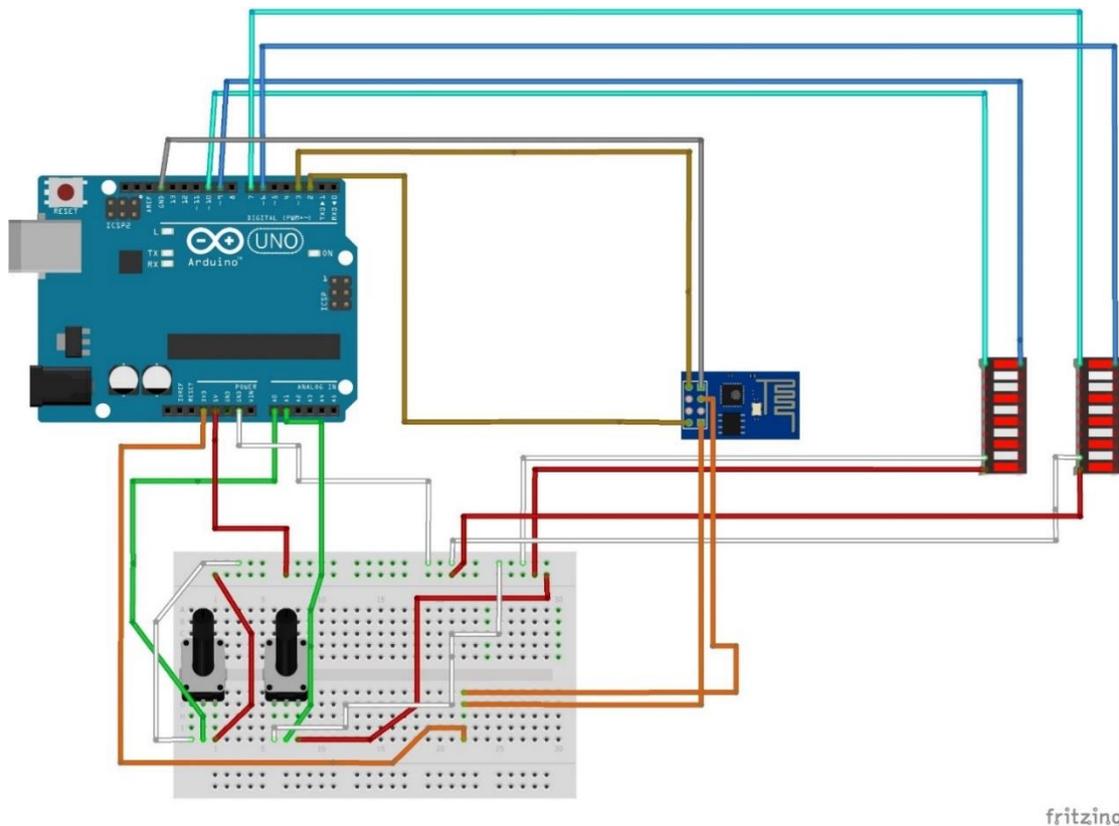


Figura 32 : Circuito prototipo 4.

**Para la realización de este esquema se han utilizado barras LED básicas en lugar de las Grove ya que Fritzing no disponía del modelo utilizado en el proyecto.*

15.4. Carcasa

Para contener la placa *Arduino*, la *Protoboard* y los cables y componentes ordenados, a parte de para dar una visión más estética al prototipo, se le ha diseñado una carcasa utilizando la herramienta online de diseño CAD *Tinkercad* que será impresa posteriormente.

A continuación se muestran los bocetos y el diseño final :

15.4.1. Bocetos

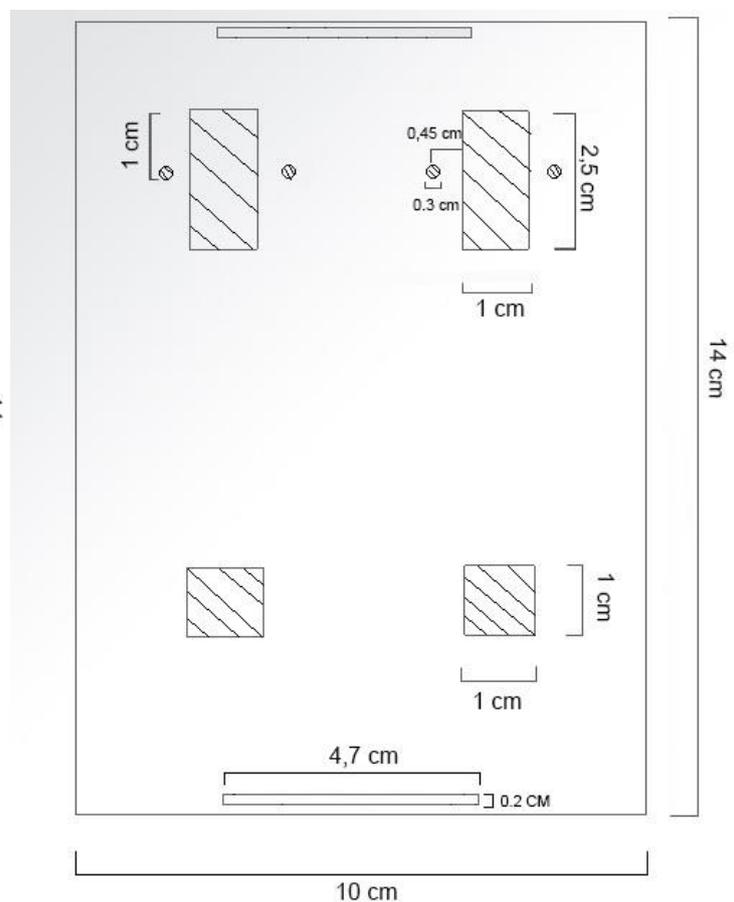
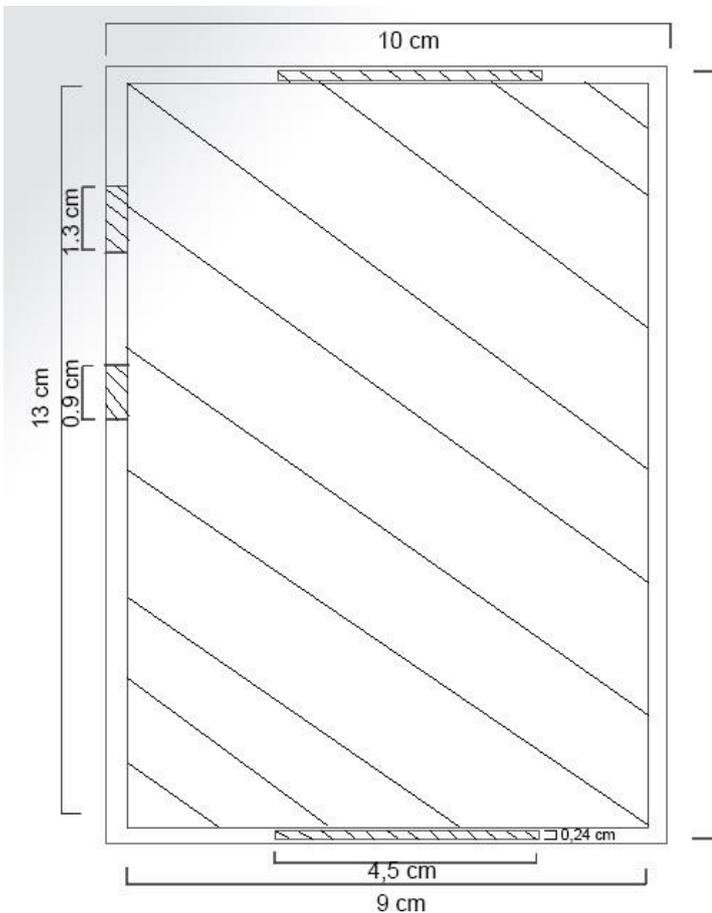


Figura 33 : Boceto del compartimento de *Arduino* y *protoboard*

Figura 34 : Boceto de la tapa

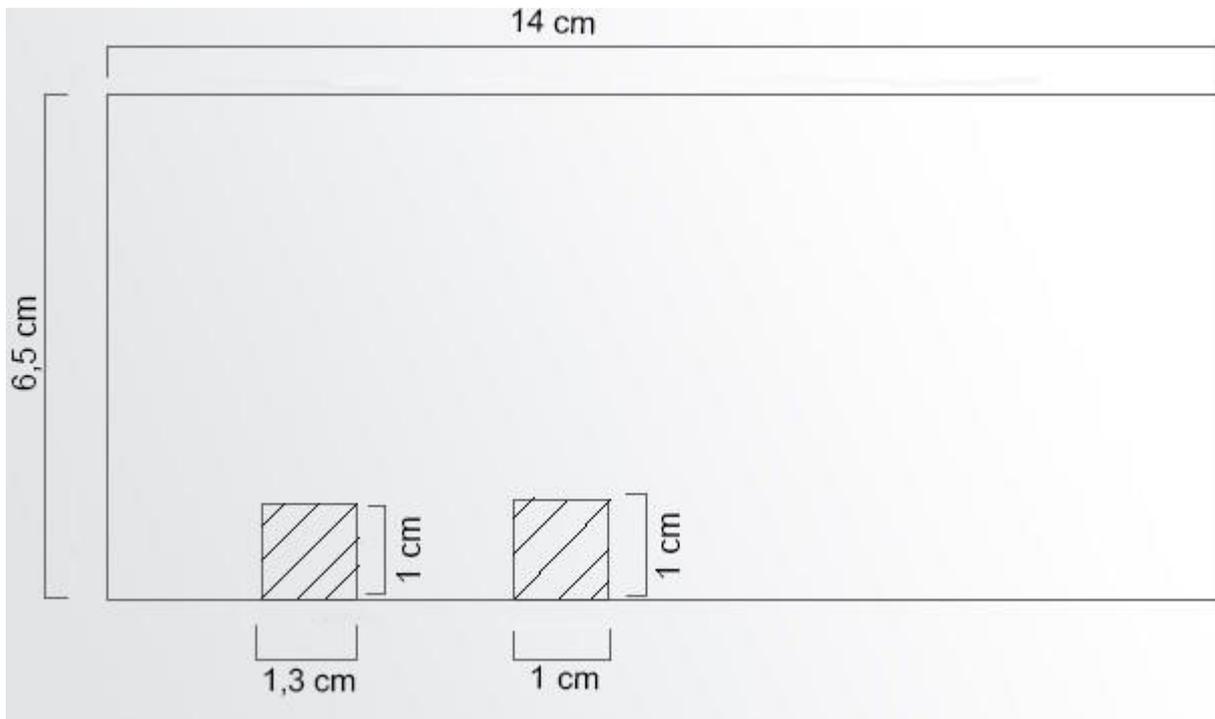


Figura 35 : boceto del perfil.

15.4.2. Diseño final en 3D

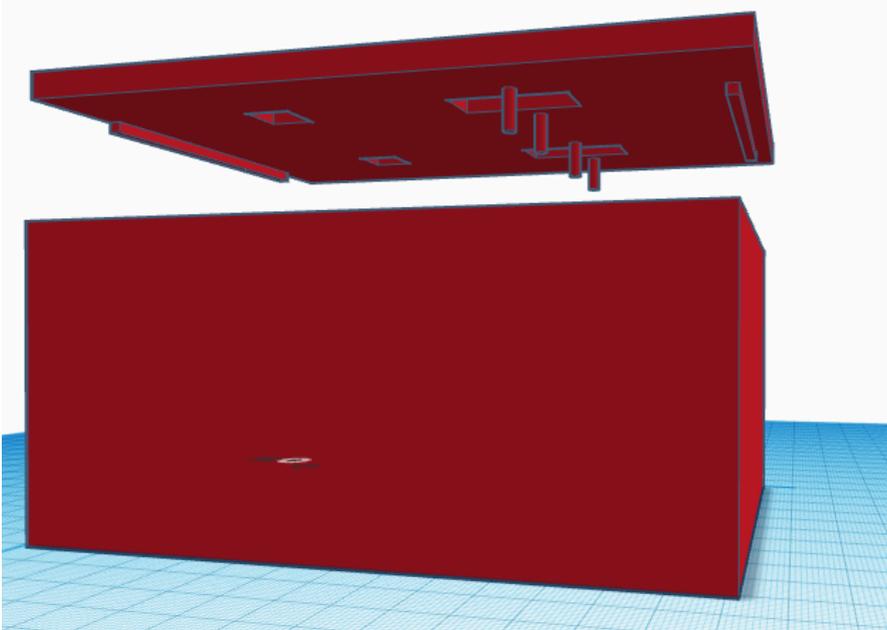


Figura 36 : Muestra del diseño final 1.

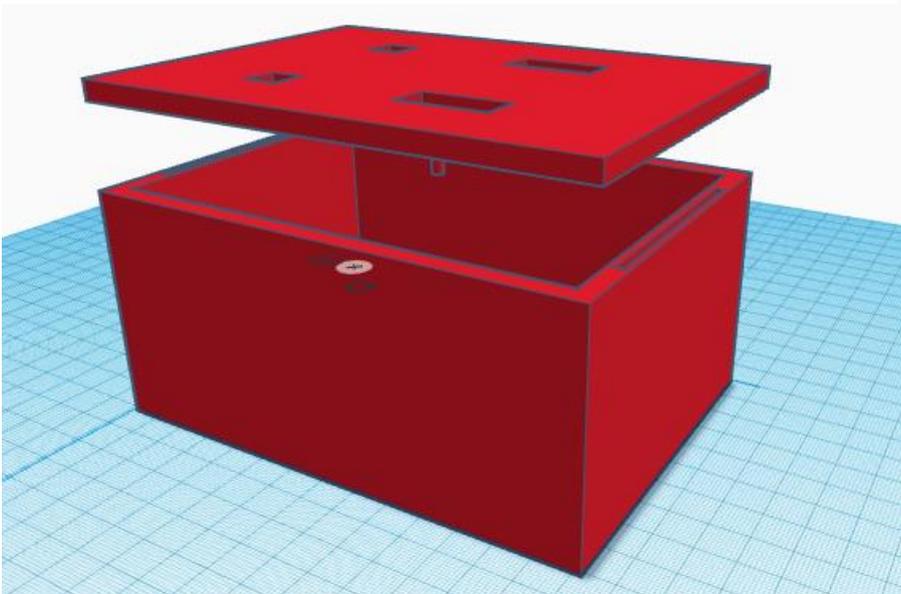


Figura 37 : Muestra del diseño final 2.

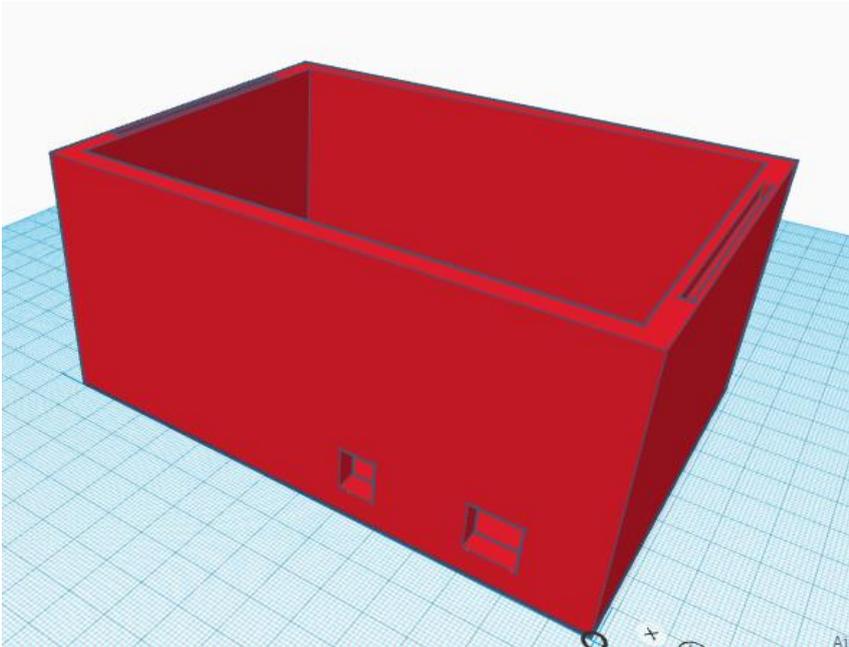


Figura 38 : Muestra del diseño final 3.

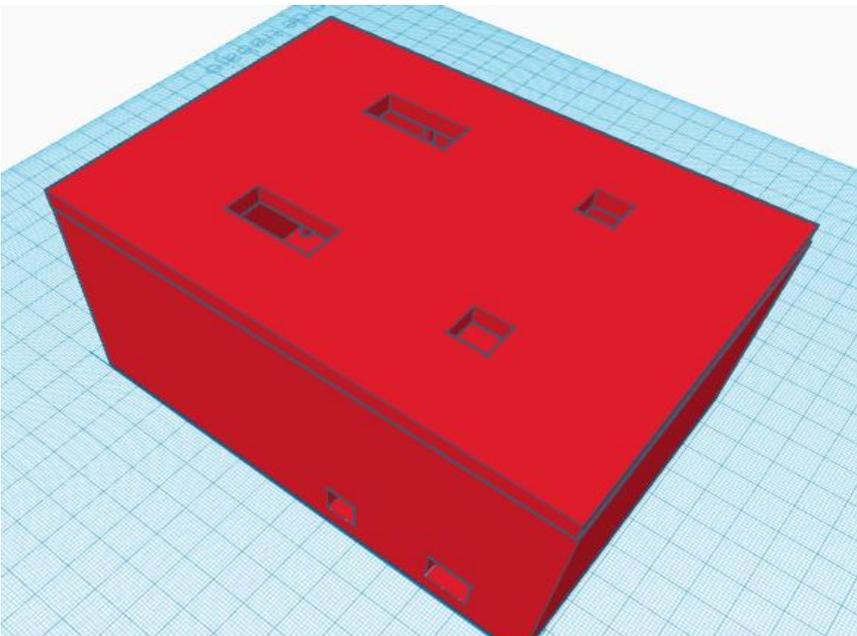


Figura 39 : Muestra del diseño final 4.

16. Prototipo final y ejemplo de funcionamiento

Para mostrar un ejemplo práctico del uso del sistema y además mostrar imágenes del sistema montado, en este apartado mostraremos un *Story Board* fotográfico exponiendo los posibles escenarios que pueden dar lugar durante la utilización del prototipo:

Caso de envío de Tweet por variación de uno de los rangos:

1.El usuario enciende el sistema.
En la imagen se pueden ver los vúmetros indicando que están en el rango 0 (Primeros dos leds encendidos).

2.El usuario decide mover uno de los potenciómetros cambiando el rango de valores en el que se encuentra originalmente.

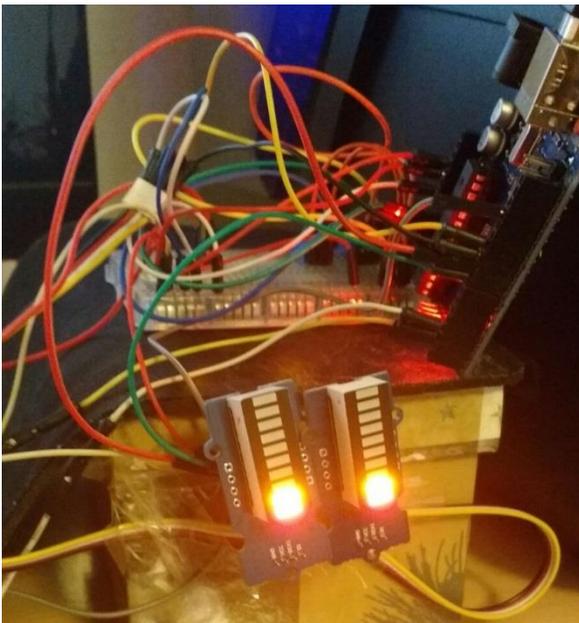


Figura 40 : Vúmetros en rango 0.

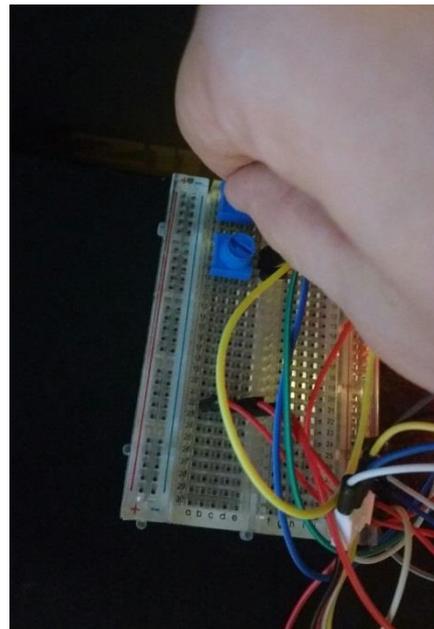


Figura 41 : Usuario moviendo potenciómetro.

3.El rango del vúmetro correspondiente cambia al mover el potenciómetro, siempre y cuando se supere el valor del umbral.

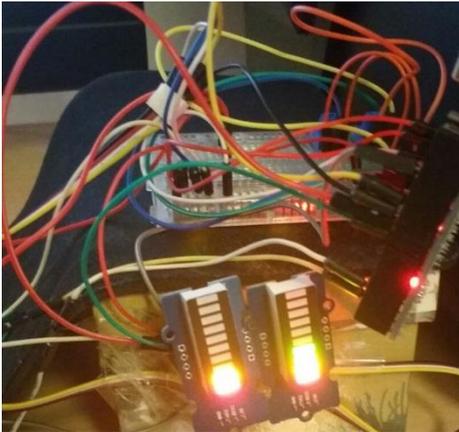


Figura 42 : Vúmetros en rango 0 y 1 respectivamente.

4.Al terminar el ciclo de holgura en el que Nos encontramos, se envía un tweet con los nuevos valores, ya que el rango ha cambiado.



Figura 43 : Tweet enviado.

Caso sin envío de tweet por que los rangos no han variado. Partiendo del caso anterior:

1.El usuario no mueve ninguno de los dos potenciómetros o no los mueve lo suficiente como para hacer variar el valor del rango, lo que hace que en el ciclo de holgura actual, no se envíe ningún *Tweet*, manteniendo el sistema en modo de espera. En la imagen puede observarse que los rangos siguen siendo los mismos que en el apartado anterior.

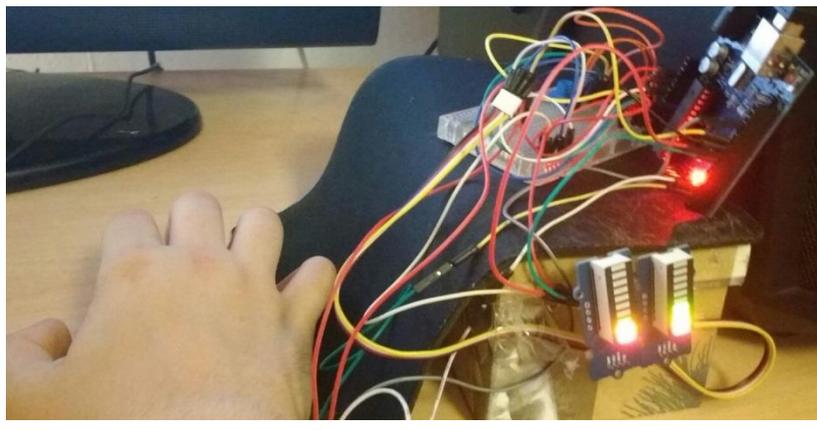


Figura 44 : En espera.

**Caso de envío de Tweet por variación de ambos rangos:
Partiendo del apartado anterior:**

1. El usuario decide mover ambos potenciómetros lo suficiente como para hacer variar el rango de estos, dando como resultado el envío de un Tweet con los nuevos datos una vez finalizado el ciclo de holgura actual.

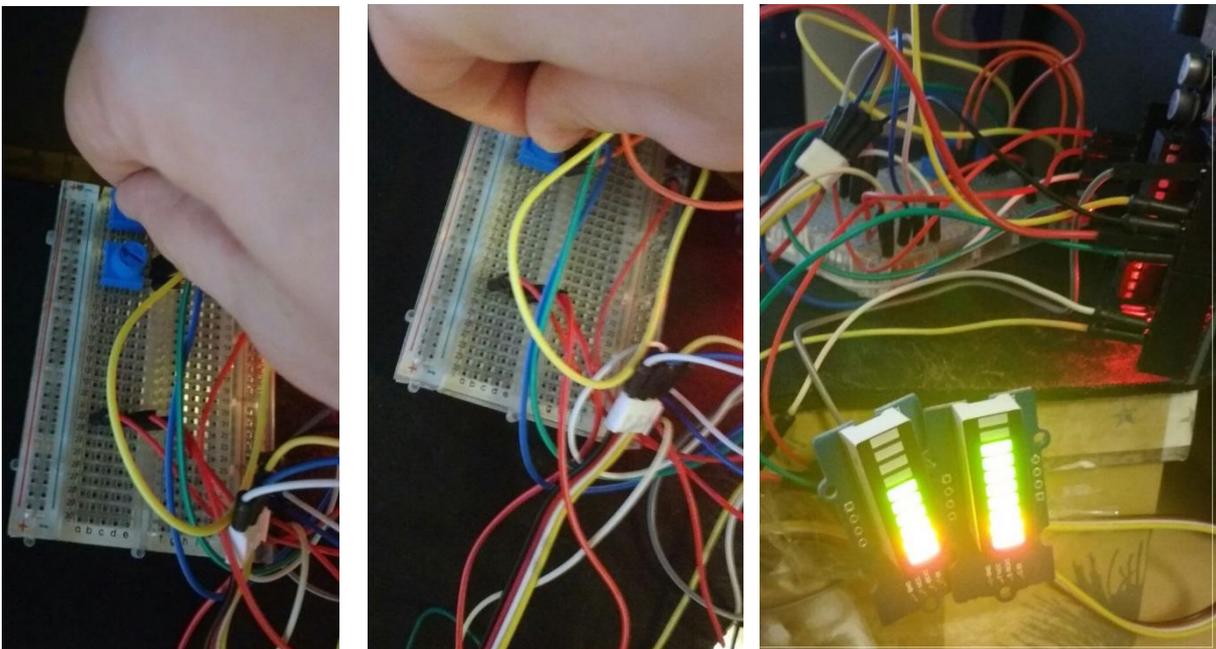


Figura 45 : (Izda) Usuario moviendo potenciómetro 1. (Centro) Usuario moviendo potenciómetro 2.. (Dcha) Vúmetros en rango 2 y 3 respectivamente.

2. Se envía el nuevo Tweet con los valores asociados.



Figura 46 : Tweet enviado 2.

17. Conclusión y valoración

Como persona interesada por la relación entre ingeniería y avance social, la realización de este proyecto me ha resultado muy satisfactoria.

Personalmente, opino que en gran parte de las ocasiones, hay falta de comunicación entre alumnos y profesores, sobre todo en los primeros cursos de estudio independientemente del grado de estos.

Con la realización de este prototipo, se podría ayudar a romper estas barreras y aportar mi grano de arena a que esa falta sea más leve o dar pie a nuevas ideas e iniciativas que puedan mejorar el campo de la enseñanza y la relación entre estudiante y maestro.

Por otra parte, siendo alumno de la rama de ingeniería Software, me ha resultado interesante realizar un proyecto centrado en el Hardware, ya que me permite expandir mis horizontes en cuanto al mundo de la informática y además, curiosear sobre proyectos de distinto enfoque respecto a los estudiados durante el curso.

En cuanto a las tecnologías utilizadas, considero que Arduino es una sistema que brinda una gran facilidad a los desarrolladores para realizar proyectos multidisciplinarios y que la popularidad de la que goza es justificada, además, la integración de este con el campo de las redes sociales es sencillo e interesante.

Finalmente, el resultado del proyecto fue gratificante, cumpliendo los requisitos expuestos por el cliente, dando lugar a un prototipo con visiones de futuro debido a su posible escalabilidad y la funcionalidad que desempeña.

18. Bibliografía y herramientas

18.1. Herramientas

Fritzing : Aplicación libre para el diseño de circuitos *Hardware*.

Utilizada para diseñar los circuitos de cada entregable para su posterior montaje.

Arduino IDE : Entorno de Arduino para la programación de este, pruebas de código y arranques del sistema.

Tinkercad : Servicio online de diseño de proyectos CAD. Utilizado para el diseño de la carcasa que será impresa en 3D para contener el sistema físico.

Thingspeak : Servicio online utilizado para enlazar la cuenta de Twitter con el sistema y obtener permisos para lanzar los Tweets.

Trello : Servicio online que permite crear tableros *Kanban* para la gestión de *Sprints* y tareas.

Ganttpro: Herramienta en línea para la creación de diagramas de Gantt.

GitHub: Herramienta online de trabajo colaborativo. Utilizada para obtener Bibliotecas de componentes hardware específicos.

Draw.io : Servicio online que proporciona herramientas para realizar una amplia gama de diagramas distintos.

Utilizado para dibujar los diagramas de secuencia y negocio.

Photoshop CS6 : Aplicación de diseño gráfico. Utilizada para vectorizar los escaneos de los bocetos de la carcasa, originalmente realizados a mano.

18.2. Bibliografía

Seed Studio Team. **Grove Led Bar** <<http://wiki.seeed.cc/>>

Seed Studio Team. **Biblioteca Grove Led Bar** <<https://github.com/>>

Jorragim. **Matriz RACI** <<http://www.codigo-binario.es/>>

Bayard Ocares S. **Diagrama BPMN** <<http://bpmn-bayard.blogspot.com.es/>>

Temas de diseño y manufactura. **Estructura Arduino**
<https://sites.google.com/site/temasdedisenoymanufactura/>

Steinheiser A. **LED Displayed Twitter Sentiment Analysis**
<https://create.arduino.cc/projecthub/>

RandomMatrix. **Twitter Mod Light – The world mod in a box**
<http://www.instructables.com/>

Arduino. **Biblioteca Software Serial** <https://www.arduino.cc/>

Manuel Infante J. **Módulo ESP 8266** <http://rambal.com/>

Free Software Foundation, Inc. **Software Libre**. Recuperado el 06 de abr de 2017.
<https://www.gnu.org/philosophy/free-sw.es.html>

Transmission Control Protocol. Recuperado el 31 may de 2017. En Wikipedia.
https://es.wikipedia.org/wiki/Transmission_Control_Protocol

Metadato. Recuperado el 24 abr de 2017. En Wikipedia.
<https://es.wikipedia.org/wiki/Metadato>

GNU/Linux. Recuperado el 14 jun de 2017. En Wikipedia.
<https://es.wikipedia.org/wiki/GNU/Linux>

Kernel. Recuperado el 5 jun de 2017. En Wikipedia.
[https://es.wikipedia.org/wiki/Nucleo_\(informatica\)](https://es.wikipedia.org/wiki/Nucleo_(informatica))

18.3. Fuentes de imágenes

Figura 1: <https://create.arduino.cc/projecthub/andrewstein/led-displayed-twitter-sentiment-analysis-715e0c>

Figura 2: <http://www.instructables.com/id/Crying-Baby-Detector-Using-1Sheeld/>

Figura 3: <http://www.instructables.com/id/Twitter-Mood-Light-The-Worlds-Mood-in-a-Box/>

Figura 21: <https://www.arduino.cc/en/Products/Counterfeit>

Figura 22: Modificación de la figura 21.

-Autor original : Arduino company.

-Modificada por Sergio Delgado Pozo el 29 de Jun de 2017

Figura 23: <https://es.aliexpress.com/w/wholesale-universal-solderless-breadboard.html>

Original Figura 24: https://es.wikipedia.org/wiki/Placa_de_pruebas/

Bajo una licencia Creative Commons que permite su modificación y difusión siempre que se acredite el autor original.

-Autor original Waveguy.

-Modificada por Sergio Delgado Pozo el 29 de Jun de 2017

Figura 26: http://wiki.seeed.cc/Grove-LED_Bar/

* Las imágenes sin fuente han sido creadas por el autor del documento.



DMSAD
*Dispositivo para la medición de
la satisfacción del alumno en la
docencia.*

