

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA DE SISTEMAS**

**DESARROLLO DE UN SISTEMA INTERFAZ CEREBRO-  
COMPUTADOR BASADO EN RESPUESTAS DE POTENCIALES  
VISUALMENTE EVOCADOS PARA CONTROL DE DISPOSITIVOS  
IOT**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**Johanna Carolina Cerezo Ramírez**  
johanna.cerezo@epn.edu.ec

**DIRECTOR: PhD. Sang Guun Yoo**  
sang.yoo@epn.edu.ec

**Quito, mayo 2021**

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por, **Johanna Carolina Cerezo Ramírez**, bajo mi supervisión.



---

**PhD. Sang Guun Yoo**  
**DIRECTOR DE PROYECTO**

## DECLARACIÓN

Yo Johanna Carolina Cerezo Ramírez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



---

**Johanna Carolina Cerezo Ramírez**

## **DEDICATORIA**

A mis padres, Jeaneth y Freddy, mi hermana Paula y a mi ñaña Ame, quienes me brindaron su apoyo incondicional y cariño durante toda esta etapa académica pese a todas las dificultades que se presentaron en el camino.

## **AGRADECIMIENTO**

A mis padres, por todos los esfuerzos que hicieron para apoyarme a cumplir mis metas, por siempre creer y confiar en mí y en mis capacidades. Por enseñarme a ser independiente y hacerme saber que obtendría todo aquello que me propusiera. A pesar de la distancia siempre sentí su cariño muy cerca, en cada mensaje de mi Pa y en cada cupcake que me enviaba mi Ma con todo su cariño. Cada vez que recibía sus consejos, cariño y palabras de aliento me daban fuerzas para no desistir y seguir adelante.

A mi hermana por su apoyo incondicional, cariño, compañía y por sacarme sonrisas siempre con sus ocurrencias, te quiero un mundo. A mi ñaña Ame que, como siempre, estuvo presente dándome aliento y cariño. A mi tía Katty y su esposo Bolívar, por recibirme en su hogar y hacerme sentir que siempre tendré un lugar en su mesa. A mi tía Elida, que siempre supo encontrar una forma de apoyarme.

A José Murillo, cuya colaboración y apoyo fue clave durante el arduo camino que significó el desarrollo de este trabajo de titulación.

A mis amigos, gracias por los buenos momentos, risas y apoyo en los tiempos más difíciles, ustedes hicieron mucho más llevadera mi vida aquí en Quito. Los llevo en el corazón.

A todas las personas que conocí en esta etapa, las que aún forman parte de mi vida y las que ya se han marchado. Porque todos, a su manera, me ayudaron a crecer y ser una mejor persona y profesional.

Al PhD. Sang Guun Yoo, quien contribuyó con su tiempo, consejos e invaluable conocimientos al desarrollo de este trabajo de titulación.

Gracias a todos, sin ustedes nada de esto sería posible.

# CONTENIDO

RESUMEN.....	1
ABSTRACT .....	2
1. INTRODUCCIÓN.....	3
1.2. Pregunta de Investigación .....	4
1.3. Objetivo general .....	4
1.4. Objetivos específicos.....	4
1.5. Alcance.....	4
1.6. Marco teórico.....	4
2. METODOLOGÍA .....	9
3. DESARROLLO DE LA SOLUCIÓN .....	11
3.1. Primer Incremento .....	12
3.1.1. Análisis .....	12
3.1.2. Diseño .....	12
3.1.3. Desarrollo.....	13
3.1.4. Pruebas .....	14
3.2. Segundo Incremento .....	14
3.2.1. Análisis .....	14
3.2.2. Diseño .....	14
3.2.3. Desarrollo.....	15
3.2.4. Pruebas .....	15
3.3. Tercer Incremento .....	15
3.3.1. Análisis .....	15
3.3.2. Diseño .....	16
3.3.3. Desarrollo.....	16
3.3.4. Pruebas .....	17
3.4. Cuarto Incremento.....	18
3.4.1. Análisis .....	18
3.4.2. Diseño .....	18
3.4.3. Desarrollo.....	19
3.4.4. Pruebas .....	20
3.5. Quinto Incremento .....	21
3.5.1. Análisis .....	21
3.5.2. Diseño .....	21
3.5.3. Desarrollo.....	21
3.5.4. Pruebas .....	22

3.6.	Pruebas del Prototipo Integrado .....	23
3.6.1.	Problema .....	23
3.6.2.	Planificación .....	23
3.6.3.	Experimentación .....	24
3.6.4.	Resultados .....	25
3.6.5.	Discusión de hallazgos .....	25
4.	DISCUSIÓN .....	26
5.	CONCLUSIONES .....	28
6.	BIBLIOGRAFÍA .....	29
7.	ANEXOS .....	I
7.1.	Anexo I. Artículo Científico.....	I
7.2.	Anexo II. Confirmación de recepción del artículo en revista científica .....	XIII
	.....	XIII
7.3.	Anexo III. Mock Up de la interfaz de usuario en colores claros .....	XIII
7.4.	Anexo IV. Mock Up de la interfaz de usuario en colores oscuros .....	XIII
7.5.	Anexo V. Pruebas de tiempos de encendido y apagado de los dispositivos a través de la interfaz .....	XIV
7.6.	Anexo VI. Pruebas de tiempos de la comunicación por MQTT .....	XV
7.7.	Anexo VII. Resultados de las pruebas de manipulación de los dispositivos con el prototipo final.....	XVI
7.7.1.	Tiempos Foco.....	XVI
7.7.2.	Tiempos Enchufe.....	XVI
7.7.3.	Clasificaciones Foco.....	XVII
7.7.4.	Clasificaciones Enchufe.....	XVII

## ÍNDICE DE FIGURAS

Imagen 1. Transmisión de la señal del estímulo a través del nervio óptico hacia el lóbulo occipital [12].	5
Imagen 2. Ubicación de los electrodos según el sistema 10/20 [15].	6
Imagen 3. Diagrama de la metodología incremental	9
Imagen 4. Arquitectura del Sistema	11
Imagen 5. Mockup de la interfaz de usuario en colores claros	12
Imagen 6. Mockup de la interfaz de usuario en colores oscuros.	13
Imagen 7. Captura de pantalla de la interfaz de usuario finalizada.	14
Imagen 8. Captura de pantalla de aplicación de configuración.	15
Imagen 9. Modelo de mensajes publicación/suscripción	16
Imagen 10. Comunicación con el servidor MQTT.	17
Imagen 11. Resultados de las grabaciones de los estímulos de cada estímulo.	20
Imagen 12. Captura de la página 1 del artículo.	I
Imagen 13. Captura de la página 2 del artículo.	II
Imagen 14. Captura de la página 3 del artículo.	III
Imagen 15. Captura de la página 4 del artículo.	IV
Imagen 16. Captura de la página 5 del artículo.	V
Imagen 17. Captura de la página 6 del artículo.	VI
Imagen 18. Captura de la página 7 del artículo.	VII
Imagen 19. Captura de la página 8 del artículo.	VIII
Imagen 20. Captura de la página 9 del artículo.	IX
Imagen 21. Captura de la página 10 del artículo.	X
Imagen 22. Captura de la página 11 del artículo.	XI
Imagen 23. Captura de la página 12 del artículo.	XII
Imagen 24. Confirmación de recepción del artículo en revista científica	XIII



## ÍNDICE DE TABLAS

Tabla 1. Promedio de encendido y apagado de los dispositivos a través de la interfaz....	17
Tabla 2. Resultados del tiempo promedio de la comunicación por MQTT.....	18
Tabla 3. Correspondencia entre acciones de la interfaz, frecuencias y su clasificación. ...	22
Tabla 4. Tiempos obtenidos por los usuarios al utilizar el prototipo.....	25
Tabla 5. Comparación de resultados con otros estudios de control de dispositivos IoT basados en SSVEP .....	27
Tabla 6. Resultados de las pruebas de delay de los dispositivos.....	XIV
Tabla 7. Resultados de las pruebas de delay de la comunicación por MQTT .....	XV
Tabla 8. Tiempos resultados de las pruebas de manipulación del foco por usuario. ....	XVI
Tabla 9. Tiempos resultados de las pruebas de manipulación del enchufe por usuario. ....	XVI
Tabla 10. Clasificaciones emitidas en las pruebas de manipulación del foco por usuario. ....	XVII
Tabla 11. Clasificaciones emitidas en las pruebas de manipulación del enchufe por usuario.....	XVII

## RESUMEN

Las respuestas de potenciales visualmente evocados (SSVEPs) son respuestas del cerebro a un estímulo visual que posea una señal de entre 3.5 y 75 Hz. Estas respuestas se pueden detectar a través de la clasificación de señales recogidas a través de un equipo de electroencefalografía que se encuentra conectado al usuario. Luego, las respuestas SSVEP que han sido clasificadas pueden ser convertidas en comandos que permitan controlar dispositivos electrónicos. Con estos antecedentes, en este estudio se realizó la construcción de un sistema Interfaz Cerebro-Computador (BCI) en línea para lograr el control de dispositivos IoT utilizando 4 estímulos SSVEP, generados a través de 4 leds que parpadeaban a diferentes frecuencias. Las ondas cerebrales requeridas para la generación de las respuestas SSEVEP fueron recolectadas a través del dispositivo de electroencefalografía llamado OpenBCI. Las ondas cerebrales de los usuarios que fueron generadas cuando observaban los estímulos visuales fueron clasificadas utilizando el algoritmo de clasificación Análisis de Correlación Canónica (CCA) y transformadas a comandos de control de la interfaz de usuario de la aplicación de control de los dispositivos IoT. Una vez obtenidos los comandos, éstos han sido enviados por socket a una interfaz de usuario, que permite el control de una lista de dispositivos IoT. El control de los dispositivos IoT desde la aplicación de interfaz de usuario ha sido realizado a través del protocolo MQTT. Durante las pruebas realizadas al prototipo final, con quince voluntarios, se obtuvo un tiempo promedio de 9,7 segundos para encender un foco inteligente y 16,68 segundos para apagarlo. Y el algoritmo de clasificación obtuvo una tasa de éxito del 97.61%.

### **Palabras Clave:**

**SSVEP, Interfaz Cerebro-Computador, BCI, Internet de las cosas, IoT**

## **ABSTRACT**

Steady-State Visually Evoked Potentials (SSVEPs) are natural responses of the brain to visual stimuli when staring at a signal between 3.5 y 75 Hz. These signals can be detected through the classification of brainwaves gathered through an electroencephalography device that is connected to the user. Then, the classified SSVEP responses can be transformed into commands to control electrical devices. With this background, in this study an online Brain-Computer Interface system to control IoT devices based in four SSVEP stimulus generated by LEDs blinking at distinct frequencies was built. The brainwaves required to generate SSVEP responses were gathered through an electroencephalography device called OpenBCI and were generated when the user stares to the visual stimulus. Then, were classified using Canonical Correlation Analysis (CCA) algorithm and transformed to commands to control the user interface of the application to control IoT devices. Once obtained the commands, were send through sockets to the user interface that allows to control a list of IoT devices. The IoT devices control from the user interface was made through the MQTT protocol. During the tests made to the final prototype, with fifteen volunteers, an average time of 9.7 seconds was obtained to turn on a smart bulb and 16,68 seconds to turn it off. And the classifier algorithm got a success rate of 97.61%.

### **Keywords:**

**SSVEP, Brain-Computer Interface, BCI, Internet of Things, IoT**

# 1. INTRODUCCIÓN

El internet de las cosas, IoT por sus siglas en inglés, es la interconexión entre objetos para crear un entorno inteligente, permitiendo a las personas controlar o acceder a dispositivos electrónicos de su entorno a través de comandos de voz, gestos o movimiento [1]. Otro de sus beneficios es que brinda más y mejor información para la toma de decisiones cotidianas. La popularidad de esta tecnología ha aumentado enormemente en los últimos años debido a una disminución de costos y simplicidad de los dispositivos inteligentes [2].

Aunque se podría pensar que IoT es una tecnología totalmente inclusiva, esta es una conclusión alejada de la realidad. Pues si bien facilita la vida de la mayoría, no considera a las personas que no poseen completa movilidad de su cuerpo. Personas que sufren de MND (*Motor Neuron Disease*), un conjunto de trastornos neurodegenerativos relacionados con la degeneración de la motoneurona superior e inferior [3]. Este grupo abarca a la población que no puede realizar movimientos de la parte superior o inferior de su cuerpo y que para 2016 sumaban 330 918 alrededor del mundo [3]. Y también a personas que se han visto involucradas en accidentes de tránsito quedando discapacitas temporal o definitivamente alcanzando una cifra de 20 a 50 millones [4]. Esta parte importante de la población no se beneficia de los avances del IoT debido a su falta de movilidad quedando aislados del resto del mundo.

Pero, el desarrollo de tecnologías como el BCI (*Brain-Computer Interface*) podría ayudar a esta población excluida. El BCI es un sistema de comunicación que traduce la actividad cerebral en comandos para un computador u otros dispositivos [5]. En otras palabras, un BCI permite a su usuario interactuar con los dispositivos electrónicos de su entorno a través de la interpretación de las señales de su actividad cerebral.

Existe una gran variedad de patrones cerebrales que pueden ser utilizados en un BCI entre los cuales tenemos; *P300*, *Motor Imagery*, *Error Potentials*, Parpadeos intencionales y los *Steady-State Visually Evoked Potentials* (SSVEPs). Estos últimos son generados naturalmente en el cerebro al recibir el estímulo visual de una luz parpadeando a ciertas frecuencias.

Algunas de las ventajas de los SSVEPS sobre los demás patrones cerebrales es que poseen una tasa alta de transferencia de información, son fácilmente generados y a bajo costo [6]. Además, este patrón cerebral se puede detectar de forma no invasiva y los

usuarios necesitan de poca capacitación para hacer uso de un sistema basado en este estímulo [7]. Es por todas estas razones que los SSVEPs fueron seleccionados como el patrón cerebral a detectar en esta investigación.

## **1.2. Pregunta de Investigación**

¿Se puede traducir ondas cerebrales recolectadas a través de un sistema BCI y generadas por estímulos SSVEP para controlar dispositivos inteligentes sin hacer uso de las capacidades motoras del usuario?

## **1.3. Objetivo general**

Desarrollar un sistema BCI basado en respuestas de potenciales visualmente evocados que al ser transformados en comandos permitan controlar dispositivos inteligentes.

## **1.4. Objetivos específicos**

- Traducir ondas cerebrales basadas en SSVEP a comandos operativos.
- Implementar un prototipo de un sistema BCI basado en SSVEPS para controlar dispositivos IoT.
- Verificar el funcionamiento del sistema BCI mediante pruebas al prototipo.

## **1.5. Alcance**

El presente proyecto de titulación tiene como meta desarrollar una solución que permita al usuario controlar dispositivos IoT a través de sus ondas cerebrales. Esto se logrará a través de un dispositivo BCI que recogerá SSVEP desde el lóbulo occipital del usuario utilizando una técnica de recolección de actividad cerebral no invasiva. Las señales SSVEP serán generadas por la observación de luces led que parpadean a distintas frecuencias y luego estas señales serán clasificadas para ser empleadas como comandos para accionar dispositivos electrónicos.

## **1.6. Marco teórico**

El sistema nervioso central (SNC) es el encargado de proporcionar una respuesta acorde con los estímulos del entorno, pero cuando el SNC falla, las respuestas del cuerpo humano se ven afectadas o son nulas. Un sistema BCI mide la actividad del SNC, lo convierte en una salida artificial que reemplaza, restaura, intensifica, suplementa o mejora la salida del

SNC [8], y con esto se modifica positivamente la interacción entre el SNC y el entorno. Es por esto que un BCI supone, para el ser humano, una forma artificial de interactuar con su entorno.

Para desarrollar un sistema BCI es fundamental capturar las ondas cerebrales. Para adquirirlas existen dos grandes grupos de técnicas; las invasivas, que usan implantes quirúrgicos en el cerebro, y las no invasivas que no requieren de estos implantes [9], siendo las más utilizadas, pues no representan ningún riesgo para la salud y son mucho más fáciles de implementar. Entre las técnicas no invasivas los más usados son los EEG (electroencefalografía) que producen un registro de actividad bioeléctrica cerebral [10], y que se recogen a través de electroencefalogramas.

A través de un EEG se pueden capturar diferentes tipos de patrones en la actividad cerebral como; las señales generadas por los parpadeos intencionales [11]. P300 en el cual el usuario espera a que suceda un evento, que desatará una señal detectable principalmente en el área parietal del cerebro [12]. *Motor Imagery*, en el cual el sujeto debe imaginar el movimiento de una parte de su cuerpo, generalmente de una las extremidades, lo cual generará un incremento de la actividad cerebral que podrá ser detectado por el EEG. O los SSVEP que son respuestas naturales del cerebro al estímulo visual causado por una señal de entre 3.5 y 75 Hz [11].

Cuando se presenta un estímulo visual repetitivo en una frecuencia en un rango mayor a 3.5 Hz y menor a 75 Hz, esta señal se transmite por el nervio óptico hacia la corteza visual, en donde se presentan una secuencia continua de cambios oscilatorios que se conocen como SSVEP [13]. Generalmente se los identifica como una onda que asemeja a una onda sinusoidal en la frecuencia en la cual se encuentra el estímulo que provocó la señal o en una de sus frecuencias armónicas.

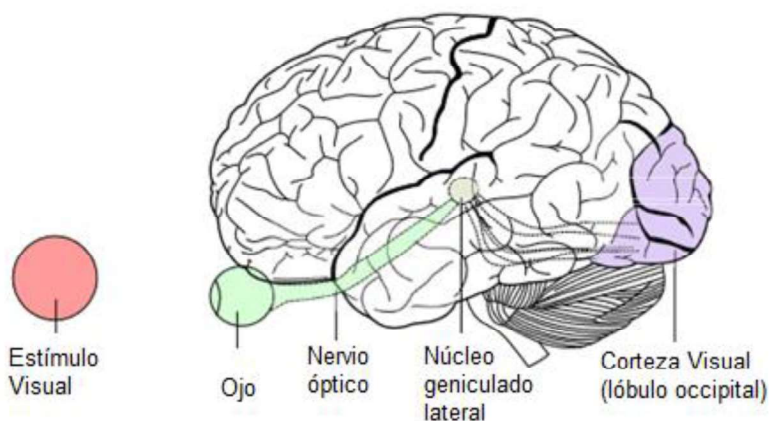


Imagen 1. Transmisión de la señal del estímulo a través del nervio óptico hacia el lóbulo occipital [12].

En palabras más simples, cuando el ojo humano es estimulado por una luz que parpadea a una señal en una frecuencia específica, esa señal es recreada en el lóbulo occipital del cerebro de donde puede ser recogida por el dispositivo BCI, clasificada y utilizada para enviar comandos a un dispositivo IoT. Los SSVEPs son uno de los patrones cerebrales más utilizados para los BCI, gracias a que no se necesita entrenar al usuario final y suelen presentar un valor predictivo positivo mayor al 78 % [14].

Para detectar la actividad cerebral o los cambios en la misma, a través de las técnicas no invasivas, se tiene que colocar varios electrodos sobre el cuero cabelludo, pero esto no se realiza arbitrariamente, sino que es habitual utilizar el sistema 10/20. Este un sistema reconocido internacionalmente y que sirve para describir la ubicación de los electrodos en el cráneo [15]. Este sistema se basa en la relación entre la ubicación del electrodo y el área de la corteza cerebral en la que se encuentra [16]. Los números 10 y 20 hacen referencia a que la distancia entre los electrodos adyacentes es del 10% o 20% de la distancia total de adelante hacia atrás o de derecha a izquierda del cráneo.

En este sistema cada lóbulo tiene una letra que lo identifica, excepto por los canales C que son utilizados únicamente para propósitos identificativos y no para referirse a ningún lóbulo, acompañado de un número que es par para los electrodos que se encuentran en el hemisferio derecho e impar para los que se encuentran en el izquierdo.

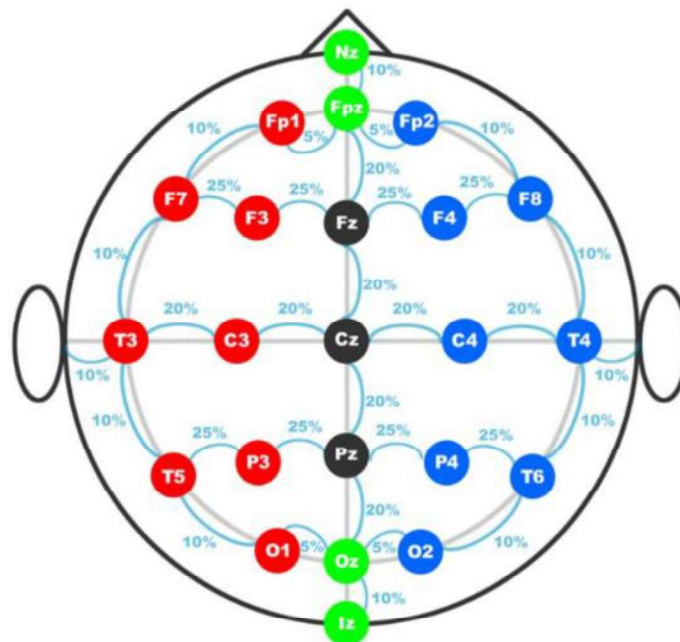


Imagen 2. Ubicación de los electrodos según el sistema 10/20 [15].

Como se explica en [17], el algoritmo más usado para clasificar las señales generados por SSVEP es CCA (Canonical Correlation Analysis). Otros métodos como FFT (Fast Fourier Transform) o LDA (Linear Discriminant Analysis) también tienen amplia acogida dentro de la comunidad científica. Últimamente se ha extendido el uso de algoritmos de inteligencia artificial como SVM (Support Vector Machine), CNN (Convolutional Neural Network) o AFNN (Adaptative Feedforward Neural Network) debido a que ofrecen mejor precisión, pero que en contra parte necesitan de mayor cantidad de datos y tiempo para ser entrenados.

CCA es un método clásico para determinar la relación entre, al menos, dos conjuntos de variables multidimensionales [18]. Este método estadístico extrae la relación entre pares de vectores  $x$  y  $y$ , y la resume en una matriz de covarianza. Luego, la matriz de covarianza se divide en cuadrantes utilizando las combinaciones de variables que pertenecen a ambos conjuntos. Después, estos cuadrantes se multiplican entre si haciendo uso de algebra matricial y la matriz resultante se somete a un análisis de componente principal para producir los resultados de CCA [19]. Y como se menciona en [20], CCA se ha utilizado ampliamente en el desarrollo de sistemas BCI basados en SSVEP en línea con alto rendimiento y estabilidad. Además, es menos dependiente del usuario y permite el uso de frecuencias armónicas [17].

Existe una gran variedad de sistemas BCI que utilizan SSVEP enfocados en brindarle una mejor calidad de vida a personas que sufren de discapacidades temporales o definitivas. Algunos ejemplos de aplicación de estos sistemas son: [21] en el que se implementa el control de una cama médica, [22] en donde se desarrolla un asistente para la alimentación de pacientes, en [23], [24] y [25] se controla brazos robóticos.

También existen ejemplos del uso de SSVEPs para controlar dispositivos IoT. En [26] se implementa con éxito el control una cortina eléctrica y un foco utilizando cuatro frecuencias de 8, 9, 11 y 12 Hz, este trabajo alcanzó una tasa de éxito del 87% y un tiempo promedio de 4 segundos. Este trabajo tiene una limitante importante ya que con 4 frecuencias controla únicamente dos dispositivos y realiza solo 2 acciones en cada uno de ellos debido a que, a cada estímulo le asigna una acción. Esto no es lo más eficiente a largo plazo, pues si se quiere controlar más dispositivos se tendría agregar más estímulos.

En [27] se desarrolló un sistema offline para controlar dispositivos IoT como un foco, un ventilador, una televisión y un reloj de alarma usando cuatro frecuencias (10, 11, 12 y 13 Hz). Este trabajo alcanzó una tasa de éxito promedio del 84,8%, sin embargo, el sistema fue probado únicamente con 2 personas. Y como en el trabajo anterior, se le asigna a cada



dispositivo una frecuencia, pero en este caso a cada frecuencia se la utiliza como un switch de encendido y apagado. Sin embargo, tiene el mismo problema anterior ya que si se deseara agregar más dispositivos se debería agregar más frecuencias y por cada dispositivo únicamente se podría realizar 2 acciones.

Otro ejemplo es [28], en donde se utiliza SSVEPs generados a través de realidad aumentada para controlar electrodomésticos. Este trabajo alcanzó una tasa de éxito del 92.8%. Sin embargo, no otorga los datos suficientes para replicar el trabajo en un escenario local, ya que no especifica ni el número de estímulos utilizados, ni sus frecuencias; y tampoco menciona que dispositivos se controlaron ni que acciones se podía realizar estos.

En [29] también se usa realidad aumentada combinada con SSVEPs para controlar un foco, un elevador y una máquina de café, alcanzado una tasa de éxito del 85.7% al ser probado con 7 usuarios. En este trabajo se utilizó un máximo de 4 estímulos a las frecuencias de 6, 7, 8 y 9 Hz y se utilizó un sistema de códigos QR para identificar el dispositivo al cual se quería acceder. Cuando un usuario quería activar el foco debía primero escanear el código QR del foco a través de unas gafas de realidad aumentada y luego en las gafas se le mostrarían dos estímulos correspondientes a encender o apagar el dispositivo. Dependiendo del dispositivo, se muestra en las gafas más o menor cantidad de estímulos. Aunque este trabajo tiene una implementación interesante, no está enfocado en personas con escasa movilidad. Debido a que los códigos QR de los dispositivos estaban repartidos por el edificio en el cual se realizaron las pruebas, los sujetos debían desplazarse por las distintas áreas para acceder al control de los distintos objetos.

En [30], se implementó el control en línea de un foco, un televisor y unas cortinas a través de una combinación entre SSVEPS y seguimiento de ojos con Hololens. Se utilizó 4 estímulos a las frecuencias de 4, 6, 10 y 15 Hz. A cada estímulo se le asignó una acción permitiendo realizar 4 acciones en cada uno de los dispositivos. El sistema fue probado con 11 participantes y obtuvo una tasa de éxito del 76.1% para los SSVEPs y 89.3% de éxito para la combinación entre SSVEPs y seguimiento de ojos. Sin embargo, en este sistema no existe navegación entre dispositivos, sino que se acceden por separado, por lo cual el usuario no podría desplazarse de las opciones del foco a las del televisor.

Por otro lado, en [31] se implementó un sistema BCI basado en SSVEPs para el control de un foco, un televisor, un aire acondicionado, una cama eléctrica, una silla de ruedas y un sistema de mensajes cortos (SMS). Los estímulos fueron generados en la pantalla de una Tablet en la cual se ejecuta una aplicación que muestra 6 estímulos, en la pantalla principal, que permiten acceder a las acciones de cada uno de los dispositivos antes mencionados.

Cuando el usuario accede a un dispositivo se le muestra un menú con varios estímulos, dependiendo de las acciones que tenga disponibles, más un estímulo adicional que permite regresar a la pantalla principal. Es por esto por lo que el número de estímulos mostrados en pantalla varía desde 3 como es el caso del foco, hasta un máximo de 6 en el caso del control de la televisión y el aire acondicionado. Las pruebas a este prototipo se realizaron con 7 hombres con un rango de edad entre 20 y 25 años y durante las mismas el sistema obtuvo un 100% de eficiencia para el control de luces, 98.94% de eficiencia para el control del televisor, 98.53% para el control del aire acondicionado y 96.82% para el sistema de mensajería. Sin embargo, no se muestran los resultados de eficiencia en las pruebas de la cama eléctrica o la silla de ruedas. Y tampoco se indica cuáles fueron las frecuencias utilizadas para cada estímulo.

Basado en las soluciones presentadas anteriormente la meta de este trabajo es desarrollar una solución BCI basado en el patrón cerebral SSVEP para controlar dispositivos IoT. Las señales serán clasificadas utilizando el algoritmo de clasificación CCA para luego ser convertidas a comandos de navegación. A diferencia de las soluciones expuestas anteriormente el número de acciones y dispositivos controlables disponibles no será dependiente del número de estímulos del sistema.

## 2. METODOLOGÍA

La metodología incremental es uno de los métodos de la ingeniería de software que se basa en la implementación de software paso a paso en forma de modelo en expansión [32]. En esta metodología el producto es diseñado, implementado y probado en varias etapas. En cada etapa o incremento se van añadiendo más características al producto, hasta llegar a la versión final.

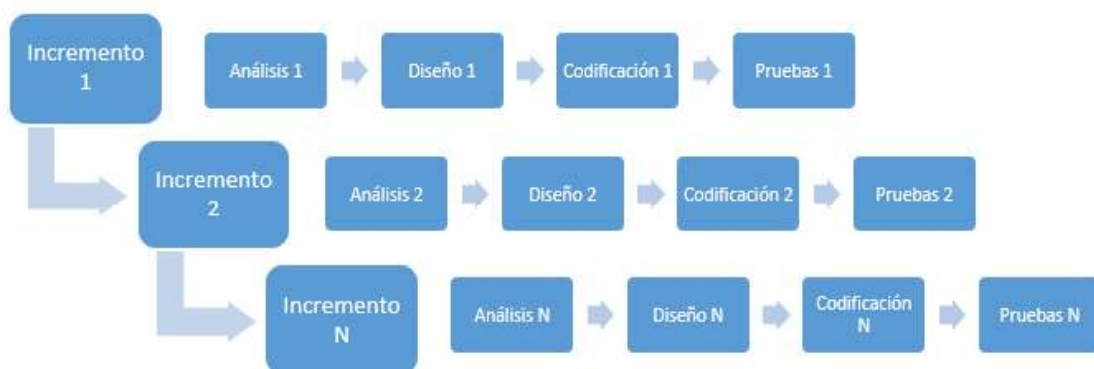


Imagen 3. Diagrama de la metodología incremental

Cada incremento consta de cuatro etapas [33]. La primera es análisis, en donde se recogen los requerimientos y especificaciones del software. Luego está la fase de diseño; una vez se han entendido los requerimientos, se planea las estrategias para implementar y lograr este requisito. Después sigue la etapa de codificación en la cual se desarrolla el código. Y finalmente viene la etapa de pruebas en la cual, como su nombre lo indica, se le realizan pruebas al prototipo desarrollado.

Para el desarrollo de este proyecto de titulación se utilizará una adaptación de la metodología incremental. Debido a que se desarrollará un prototipo que va más allá del desarrollo de software la etapa de codificación será reemplazada por una de desarrollo en la cual se implementarán las distintas características del prototipo. Las demás etapas de cada incremento serán las mismas.

Como último paso, después que el prototipo se haya completado se utilizará la metodología Action Research que se basa en la relación que existe entre la experimentación y la generación de conocimiento. Esta metodología se puede utilizar lineal o cíclicamente, dependiendo de las necesidades de la investigación [34]. Aunque esta metodología tuvo sus inicios en la rama de las ciencias sociales, últimamente ha ganado popularidad en otras ramas de la ciencia, especialmente en las ciencias de la computación [35]. Esto gracias a que prioriza la generación del conocimiento y se la considera como una metodología de investigación empírica general.

Si bien las fases de la metodología dependen del enfoque de la investigación es necesario que se incluyan cinco etapas: (1) Identificación del problema, (2) Planificación de las acciones a tomar para solucionar el problema, (3) selección de las acciones a tomar, (4) evaluar las consecuencias de las acciones seleccionadas y (5) identificar los hallazgos.

Esta metodología se ajusta a los requerimientos de la última fase de este proyecto de investigación ya que permite definir el mejor curso de acción a tomar al momento de realizar las pruebas finales del prototipo.

### 3. DESARROLLO DE LA SOLUCIÓN

La implementación de la solución constará de 5 incrementos que contribuirán a completar el prototipo más una etapa final de pruebas. En la primera etapa se codificará la interfaz de usuario que muestra las acciones y dispositivos disponibles para manipularse. La segunda etapa estará centrada en la elaboración de la interfaz de configuración de los dispositivos.

El tercer incremento consistirá en la implementación de la infraestructura de comunicación con los dispositivos de IoT a través de un broker MQTT. La implementación de los estímulos SSVEP a través de luces leds se desarrollará en el cuarto incremento. En el quinto incremento se desarrollará el algoritmo de clasificación, la recolección de señales y serán lo integradas con la interfaz de usuario. Y la última etapa consistirá en las pruebas completas del prototipo en base a la metodología Action Research.

Debido a la pandemia y al riesgo de contagio que conlleva el contacto con personas ajenas al círculo del hogar y la limitada disponibilidad de tiempo de los voluntarios, las pruebas de los 5 incrementos se realizarán con un número limitado de 2 a 4 personas. Sin embargo, para la última etapa en la cual el prototipo está completo y se puede corroborar la funcionalidad íntegra del prototipo, se realizarán las pruebas con 15 personas. La arquitectura del prototipo al final del desarrollo se puede observar en la imagen 4.

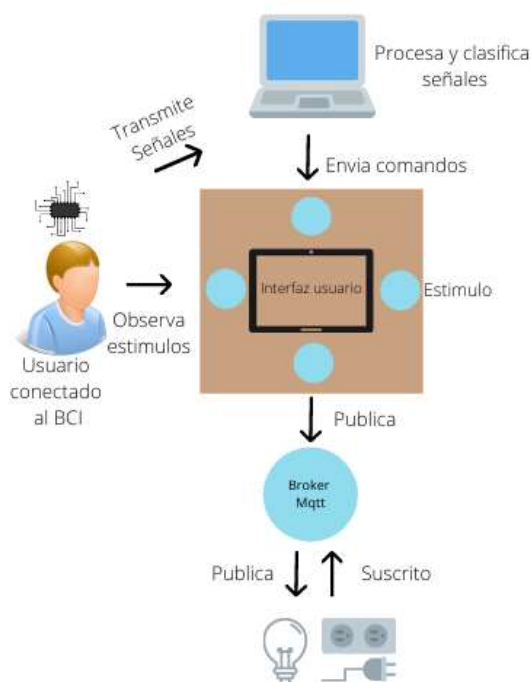


Imagen 4. Arquitectura del Sistema

## 3.1. Primer Incremento

En este incremento se realizó la implementación de la interfaz de usuario en la cual se podrán observar los dispositivos IoT disponibles para su control.

### 3.1.1. Análisis

Se determinó que la interfaz de usuario debería estar compuesta por un menú de atajos que contenga las acciones principales disponibles para los diferentes dispositivos IoT y la opción de acceder a la aplicación principal en la cual se incluirán los dispositivos agrupados por habitaciones para asemejar la distribución del hogar y que al hacer clic sobre cualquier dispositivo se desplieguen todas las acciones disponibles que posee.

### 3.1.2. Diseño

Se realizó mockups de la aplicación utilizando Invision Studio en los cuales se muestran a las acciones disponibles en el centro de la aplicación rodeada por 4 botones. En la parte superior el botón de aceptar que es el que realiza la acción. Para desplazarse entre las actividades disponibles se utilizarán 2 botones ubicados a la derecha e izquierda respectivamente y en la parte inferior se ubica el botón para salir de la aplicación. Se realizaron 2 estilos uno en colores claros y otro en oscuros, como puede observarse en las imágenes 5 y 6. La opción escogida fue la de colores oscuros porque es menos agresiva con la vista.



Imagen 5. Mockup de la interfaz de usuario en colores claros



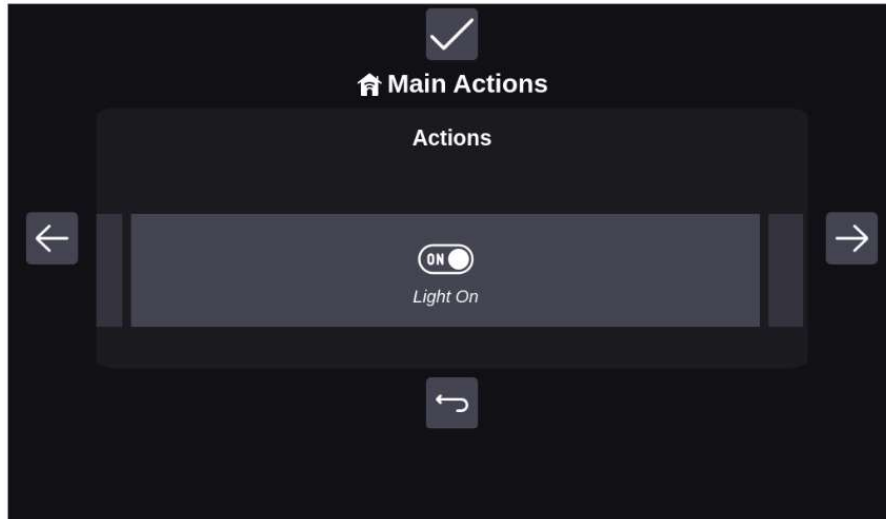
*Imagen 6. Mockup de la interfaz de usuario en colores oscuros.*

Se definió que la interfaz de usuario se realizaría con el framework Electron que permite crear aplicaciones de escritorio nativas a través de Javascript, Html y CSS. Las aplicaciones desarrolladas con electrón son multiplataformas, por lo cual pueden ejecutarse sobre cualquier sistema operativo [36]. Además, es un framework de código abierto y que al utilizar HTML y CSS permite desarrollar interfaces amigables con el usuario.

Para garantizar la persistencia de la información se diseñó un modelo entidad relación y se decidió utilizar SQLite pues un motor de base de datos ligero que no implementa servidor, por lo que es útil en esta aplicación ya que no se almacenara una gran cantidad de datos. Y para el control de versiones de la aplicación se eligió Git y Github para el almacenaje del repositorio.

### **3.1.3. Desarrollo**

Se realizó la codificación de la aplicación basados en los parámetros y frameworks establecidos en las etapas anteriores. En la imagen 7 se puede observar la interfaz finalizada.



*Imagen 7. Captura de pantalla de la interfaz de usuario finalizada.*

#### **3.1.4. Pruebas**

Las pruebas para esta etapa consistieron en verificar la navegación a través de la interfaz utilizando los botones. Se realizaron con 4 personas y obtuvieron un éxito del 100%.

### **3.2. Segundo Incremento**

En esta fase se realizó la implementación de una interfaz de configuración en la cual se podrán agregar y eliminar acciones y demás.

#### **3.2.1. Análisis**

Para incrementar la usabilidad del prototipo, se ideó una aplicación que permita configurar la aplicación de interfaz de usuario permitiendo agregar y eliminar dispositivos, habitaciones y acciones.

#### **3.2.2. Diseño**

Se definió que esta aplicación desplegaría la información de las habitaciones, acciones y dispositivos en tablas en las cuales estarán disponibles las opciones de eliminar y crear. Al igual que la interfaz de usuario esta aplicación sería realizada con Electrón y se conectaría a la misma base de datos de SQLite y tendría su propio repositorio en Github con Git para el control de versiones.

### 3.2.3. Desarrollo

Se realizó la codificación de la aplicación basados en los parámetros y frameworks establecidos en las etapas anteriores. El resultado de esta etapa se puede observar en la imagen 8.

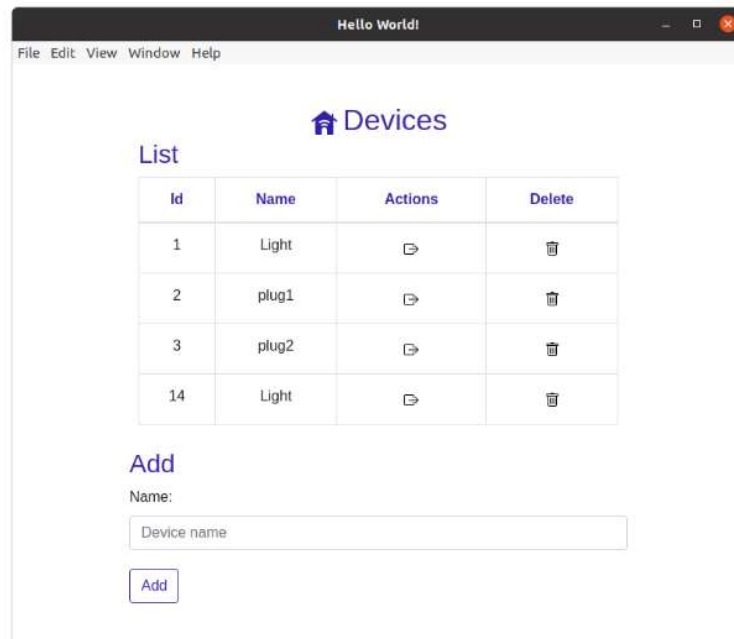


Imagen 8. Captura de pantalla de aplicación de configuración.

### 3.2.4. Pruebas

Las pruebas para esta etapa consistieron en crear una nueva habitación, con un dispositivos y dos acciones y luego eliminar dicha habitación. Al final de la prueba todo lo que creo el usuario debía haberse eliminado. Se realizaron con cuatro personas y obtuvieron un éxito del 100%.

## 3.3. Tercer Incremento

Este incremento abarcó la implementación de la infraestructura de los dispositivos IoT y su integración con la interfaz de usuario.

### 3.3.1. Análisis

En esta etapa surgió la necesidad de implementar una infraestructura que permita conectar la interfaz de usuario con los dispositivos IoT; un foco inteligente marca LIFX y una regleta marca Heyvalue que posee cuatro enchufes manipulables individualmente. Esta conexión se debe dar de una forma segura y rápida.



### 3.3.2. Diseño

Como ambos dispositivos poseían la capacidad de ser manipulados a través APIs para desarrolladores se decidió implementar la comunicación entre la interfaz y los dispositivos IoT a través de un broker MQTT que permita enviar la acción a realizarse al dispositivo correcto haciendo uso del modelo de cola de mensajes, publicación y suscripción, que se observa en la Imagen 9.



*Imagen 9. Modelo de mensajes publicación/suscripción*

El bróker escogido fue Beebotte MQTT que está basado en el modelo de canales y recursos, en el cual a canal se pueden asociar varios recursos. Opera sobre la versión 3.1.1 del protocolo MQTT y posee conexiones con SSL y sin SSL a través de los puertos 1883 y 8883 respectivamente [37]. Además, proporciona un puente transparente entre las comunicaciones utilizando Websockets, REST y MQTT.

También ofrece 3 esquemas de autenticación para cubrir distintas necesidades, acceder únicamente a un canal específico, acceder a varios canales y finalmente acceder a todos los canales de distintos usuarios. Otra de sus funcionalidades es que mantiene aisladas las conexiones de usuarios distintos. Es por estas razones que Beebotte ofrece los recursos necesarios para llevar a cabo este prototipo.

### 3.3.3. Desarrollo

En la plataforma Beebotte se creó un proyecto que contiene un canal y 5 recursos que corresponden al foco y a cada uno de los enchufes de una regleta. Se implementó un programa en Python que estará suscrito al bróker MQTT y que estará corriendo en un segundo plano y notificará a los dispositivos registrados de las acciones que deben ejecutar de acuerdo con lo recibido desde la interfaz de usuario, haciendo uso de sus respectivas librerías. Por otro lado, a la interfaz de usuario se le agregó la funcionalidad de publicar las acciones según los comandos que reciba desde la interfaz.

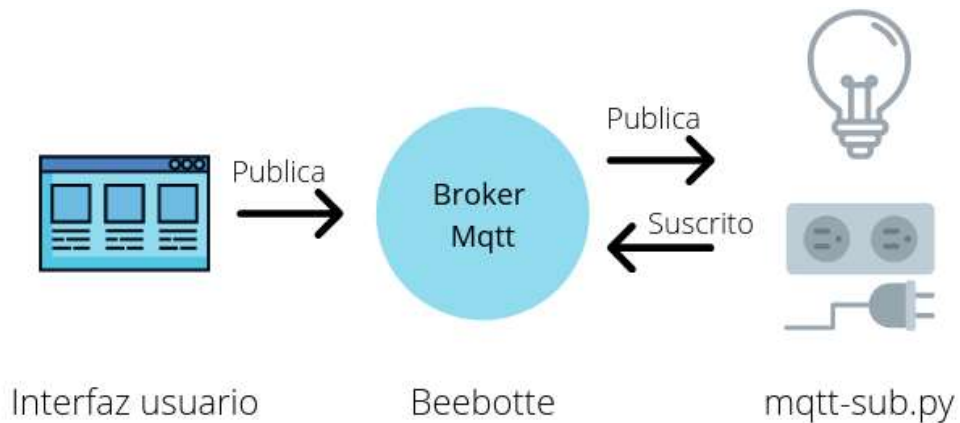


Imagen 10. Comunicación con el servidor MQTT.

### 3.3.4. Pruebas

Durante las pruebas de esta etapa se verificó que los dispositivos puedan ser accionados a través de la interfaz de usuario. Las pruebas se realizaron con 4 personas a las cuales se les pidió encender y apagar el foco y el primer enchufe de la regleta a través de la interfaz. Las pruebas fueron satisfactorias con un éxito del 100%.

También se realizaron pruebas de *delay* para verificar el tiempo que se demoraría en encender y apagar el foco y uno de los enchufes de la regleta desde la interfaz. Para esto se tomó el tiempo de encendido y apagado a través de un cronometro 35 veces y se realizó un promedio de estos resultados obteniendo los datos que se muestran en la tabla 1.

Dispositivo	Acción	Tiempo (s)
Foco	Encender	1.57
	Apagar	1.40
Enchufe	Encender	0.42
	Apagar	0.38

Tabla 1. Promedio de encendido y apagado de los dispositivos a través de la interfaz

Adicionalmente, se realizó un análisis extra para determinar cuál es el tiempo de *delay* que corresponde a la comunicación por MQTT. Para esto se tomó el *timestamp*, en segundos, al momento que se envía la publicación desde la interfaz y el *timestamp* del momento en que llega el mensaje al suscriptor. Se halló la diferencia de estos *timestamps* dando como resultado el tiempo que toma la comunicación. Igual que en la prueba anterior se realizó 35 veces el encendido y apagado del foco y un enchufe y se obtuvo un promedio de los valores obtenidos. El resultado de estas pruebas se puede observar en la tabla 2.

Dispositivo	Acción	Tiempo (s)
Foco	Encender	0,082
	Apagar	0,087
Enchufe	Encender	0,094
	Apagar	0,084

Tabla 2. Resultados del tiempo promedio de la comunicación por MQTT.

Comparando los resultados de las tablas 1 y 2 se evidencia que la comunicación por MQTT toma un tiempo ínfimo de 0,08 segundos en el caso del foco, por lo cual no tiene un gran impacto en el tiempo total de encendido del dispositivo. El resto del tiempo corresponde al tiempo que se demora el dispositivo en activarse.

### 3.4. Cuarto Incremento

En esta etapa se realizó la implementación los estímulos SSVEPS, haciendo uso de LEDs y se definió las frecuencias a utilizarse en los estímulos.

#### 3.4.1. Análisis

Ya que esta es una aplicación basada en SSVEPs, es necesario implementar un escenario que genere estos estímulos. Para navegar en la interfaz de usuario se necesita de 4 estímulos que simulen los 4 botones de la interfaz; aceptar, derecha, izquierda y regresar. Además, la interfaz debe ubicarse estratégicamente entre los estímulos para poder visibilizar la navegación en la interfaz mientras se observa a los mismos.

#### 3.4.2. Diseño

Para generar los estímulos, se consideró dos opciones LEDs y una pantalla LCD. Basado en un estudio de trabajos previos, la pantalla LCD fue descartado por las siguientes razones:

- Una pantalla LCD tiene dificultades para representar correctamente varios estímulos al mismo tiempo y a diferentes frecuencias por lo cual la respuesta SSVEP generada por un monitor puede ser débil [38],[39].
- En una pantalla, se puede representar únicamente frecuencias que sean obtenidas al dividir la tasa de refresco de la pantalla para un numero entero [40]. Es decir que en una pantalla con una tasa de refresco promedio de 60Hz y que se encuentre bien calibrada únicamente se podría usar las frecuencias: 6, 6.67, 7.5, 8.57, 10, 12, 15, y 20 Hz de las cuales se recomendarían únicamente las que se encuentran entre 7.5 y 15 Hz, por lo cual una pantalla limita las frecuencias que se pueden reflejar correctamente en la misma.

Debido a las limitaciones de las pantallas, se decidió utilizar LEDs ya que el número de estímulos o frecuencias que puede representarse es amplio y, a diferencia de los estímulos generados por pantallas, se puede representar cualquier frecuencia ya que no existe la limitante de la tasa de refresco [12].

Comúnmente los estímulos SSVEP suelen ser originados por frecuencias representadas por números enteros y que se encuentran en la banda media-baja de frecuencias. Sin embargo, la decisión de que frecuencias emplear suele ser tomada arbitrariamente por el autor. Para citar algunos ejemplos, en [26] se utiliza frecuencias entre 8 y 12 Hz, en [27] se utiliza frecuencias entre 10 y 13 Hz, en [29] se utiliza frecuencias entre 6 y 9 Hz y en otros trabajos [41] y [42] se utilizan frecuencias de 7, 8, 9 y 10 Hz con buenos resultados. Para este trabajo se escogió de igual forma frecuencias representadas por números enteros y que se encuentren en la banda media y fueron las frecuencias; 7, 8, 9 y 10 Hz.

En algunos estudios como [43], [44] y [45], utilizan únicamente los canales O2 y O1 del sistema internacional 10/20 para recoger las señales generadas por los SSVEP. En algunas ocasiones, se añade un electrodo adicional en el canal Oz pues, como se indica en [46], este canal tiene la ubicación más cercana a la corteza visual y por lo tanto un electrodo en este canal puede detectar la señal generada por los SSVEP con mayor éxito.

Aunque se puede colocar electrodos en canales adicionales (POz, P3, P4, etc.) como se realizó en [47] y [27]. Se ha definido que los canales OZ, O1 y O2 son ubicaciones críticas para recoger la actividad cerebral de los usuarios provocada por SSVEPs, ya que como se expone en [46], los SSVEP se detectan con más claridad y de manera consistente en la mayoría de los sujetos cuando se usa estos canales. Por estas razones se decidió utilizar los canales OZ, O1 y O2 ya que se simplifica el número de electrodos a utilizar, lo que lo vuelve más cómodo para el usuario, y no se pierde efectividad en la recolección de señales.

Además, se decidió colocar los LEDs en posición de rombo ya que esto permite colocar una pantalla LCD de 600x1024px en el medio de los leds, en la cual se mostrará la GUI, mientras con los Leds se imita la posición de los botones en la GUI. También se decidió utilizar leds de color blanco ya que se ha demostrado en varios estudios [48], [49] y [50] que este color tiene buenos resultados con estímulos SSVEPs.

### **3.4.3. Desarrollo**

El circuito de cada led se desarrolló en protoboards de 8 cm de largo por 5 cm de ancho, conectadas a un RaspberryPi, en donde corre un programa desarrollado en Python para

encender y apagar los leds a las frecuencias de 7Hz para el led superior, 8Hz para el led de la derecha, 9Hz para el led inferior y 10 Hz para el led de la izquierda. Los leds fueron ubicados en un panel cuadrado de 49cm x 49cm cubiertos por un difusor de acuerdo con la sugerencia de [50] ya que esto disminuye el cansancio en los ojos que se produce al ver al led directamente.

### 3.4.4. Pruebas

Las pruebas consistieron en verificar que los estímulos generados por cada frecuencia sean los correctos. Para esto se realizó una grabación de 20 segundos de una señal recogida desde el lóbulo occipital del usuario en los canales O1, O2 y Oz siguiendo el modelo 10-20. Y después, a través de la herramienta OpenBCI GUI, se graficó esta grabación para verificar los picos alcanzados en los diagramas de amplitud vs frecuencia correspondan a la frecuencia generada por el estímulo del LED.

Esta prueba se realizó con 4 personas y se evidenció que los estímulos se generaban correctamente para cada frecuencia. En la imagen 11 se puede observar los resultados. En la esquina superior izquierda, se encuentra el gráfico de la grabación de los estímulos de 7Hz. En la esquina superior derecha, se muestra la grabación de 8Hz. En la esquina inferior izquierda, se ubica la grabación de 9 Hz. Y finalmente en la esquina inferior derecha, se encuentra la grabación de la frecuencia de 10 Hz.

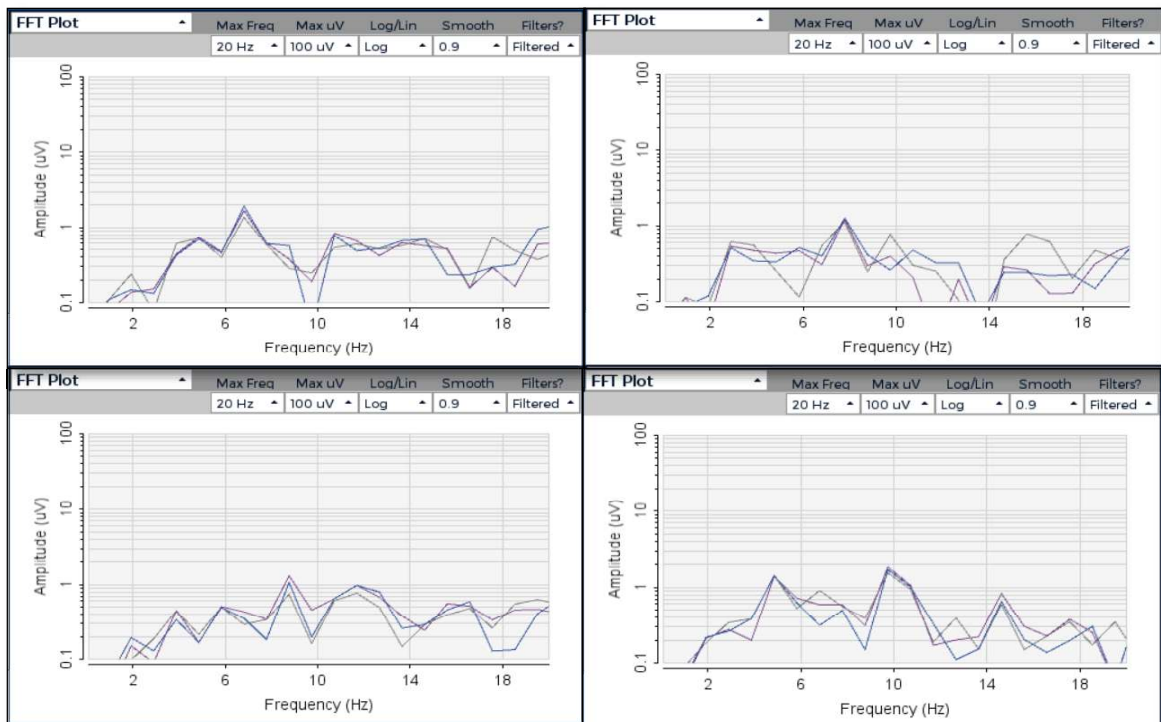


Imagen 11. Resultados de las grabaciones de los estímulos de cada estímulo.

## **3.5. Quinto Incremento**

En este incremento se implementó un escenario de recolección y filtrado de señales y el algoritmo de clasificación basado en CCA que permite identificar el estímulo que el usuario está observando y sus resultados fueron integrados con la interfaz de usuario para realizar la navegación en la misma.

### **3.5.1. Análisis**

Con la finalidad de identificar el estímulo que el usuario está observando de entre las 4 frecuencias que se estarán generando al mismo tiempo es necesario implementar un algoritmo de clasificación en una herramienta que sea compatible con el OpenBCI y que pueda comunicarse con la interfaz de usuario.

### **3.5.2. Diseño**

Para la recolección y clasificación de señales cerebrales, se decidió utilizar la plataforma OpenVibe ya que permite realizar la adquisición, filtrado, clasificación y visualización de señales cerebrales en tiempo real [51]. Además, OpenVibe es compatible con el OpenBCI, que es el dispositivo EEG utilizado. La clasificación de las señales se realizará utilizando el Canonical Correlation Analysis (CCA), un algoritmo de clasificación multiclase que no requiere entrenamiento y que posee una precisión promedio del 81.5% [17] cuando se utiliza para clasificar señales generadas por SSVEPs. La comunicación entre el algoritmo de clasificación y la interfaz de usuario se realizará mediante sockets.

### **3.5.3. Desarrollo**

En la herramienta OpenVibe, se modificó un escenario de clasificación de ondas cerebrales que hace uso de CCA. En este escenario se configuran varios pasos para el procesamiento de señales. Primero, se coloca un cliente de adquisición de señales en donde se especifica el nombre del servidor y el puerto por el cual se recibirán los datos puros. Luego se coloca un selector de canales, en donde se indica los canales a través de los cuales se va a recolectar la actividad cerebral, en este caso O1; O2 y Oz.

El siguiente paso fue el del filtrado de las señales. Aquí se configuró un filtro band pass con un rango entre 1 Hz y 40 Hz para atenuar las frecuencias que se encuentren fuera de este rango. Seguidamente, se configuró la longitud de la señal que se va a recoger para la clasificación y la ventana de tiempo, en este caso se tomará una señal de 3 segundos cada 0.5 segundos.

Después, se configuró la ruta hacia el script que contiene el algoritmo de clasificación y los parámetros que va a recibir. El algoritmo de clasificación (CCA) que utiliza este escenario se desarrolló en Python y se configuró para funcionar con cuatro frecuencias.

En este algoritmo se compara la señal de entrada con 4 señales de referencia, que corresponden a las frecuencias de 7, 8, 9 y 10 Hz y que son generadas haciendo uso de senos y cosenos multiplicados por el número de la frecuencia, para hallar su correlación y emitir una clasificación.

Al detectar la clase a la cual el algoritmo define que la señal recogida pertenece, emite el número de la clase que se le asignó según el orden en el cual se hayan ingresado las frecuencias en los parámetros del script. En este caso, para 7 Hz emite 1, para 8 Hz la salida es 2 y así sucesivamente. De tal forma que obtendremos la equivalencia entre acción en la interfaz, frecuencia y clasificación que se muestra en la tabla 3.

<b>Frecuencia (Hz)</b>	<b>Clasificación</b>	<b>Acción Interfaz</b>
7	1	Aceptar
8	2	Derecha
9	3	Salir
10	4	Izquierda

Tabla 3. Correspondencia entre acciones de la interfaz, frecuencias y su clasificación.

Basados en esta tabla de equivalencias, la navegación en la interfaz de usuario consiste en; observar al estímulo de la derecha o izquierda para desplazarse en dicha dirección entre las acciones, ver al estímulo superior para ejecutar la acción y ver al estímulo de abajo para salir del programa.

Una vez que se ha emitido una clasificación válida y con la finalidad de evitar clasificaciones descartables que ocurren mientras se cambia de un estímulo a otro o que se observa los cambios en la pantalla, se agregó también un tiempo de inactividad de 2.5 segundos durante los cuales no se tomarán en cuenta las clasificaciones que el algoritmo realice.

Las clasificaciones que emite el algoritmo se transmiten a través de un socket a la interfaz de usuario en donde realizaran la acción que les corresponde por equivalencia. También se modificó la GUI para cambiar de color el botón activado por la clasificación a verde durante un segundo, para así otorgarle al usuario retroalimentación extra sobre las acciones que está realizando al observar los estímulos.

#### **3.5.4. Pruebas**

Durante esta etapa de pruebas se comprobó que el prototipo funcione adecuadamente, es decir, que sea posible utilizar únicamente los estímulos SSVEP para desplazarse entre la

lista de acciones de la interfaz. Dos usuarios fueron conectados al OpenBCI y se les pidió que se desplacen de derecha a izquierda entre las acciones del menú y ejecuten alguna de estas acciones. El prototipo funcionó perfectamente y los usuarios fueron capaces de navegar y ejecutar las acciones a través de la interfaz sin errores.

### **3.6. Pruebas del Prototipo Integrado**

Durante esta etapa se utilizó la metodología Action Research, para definir la mejor ruta de acción al momento de evaluar el funcionamiento del prototipo.

#### **3.6.1. Problema**

Las pruebas realizadas en el quinto incremento fueron limitadas y aunque obtuvieron buenos resultados, podrían no ser confiables ya que la muestra de usuarios fue reducida y el prototipo podría ser dependiente del usuario y otorgar buenos resultados únicamente con esos usuarios. Además, no se definió retos puntuales que los usuarios debían cumplir.

#### **3.6.2. Planificación**

En esta etapa se definió las pruebas que iba a realizar el usuario, las cuales consistían en encender y apagar el foco, y encender y apagar uno de los enchufes de la regleta. Esto para abarcar desde las acciones más sencillas a las más complejas. La navegación que debe realizar el usuario en la interfaz para controlar el foco inteligente es la más simple, pues encender el foco es la primera acción que se despliega en el menú y únicamente debía observar el estímulo superior que simula el botón de aceptar para encenderlo. Y para apagar el foco el usuario tenía que realizar 2 acciones más, observar el estímulo de la derecha para navegar en esa dirección y así cambiar a la acción de apagar y luego observar el estímulo superior para realizar esta acción.

Para la segunda prueba en la cual se tenía que encender un enchufe, la navegación es un poco más compleja. Para acceder a la acción de encender el primer enchufe de la regleta, el usuario tendrá que observar dos veces el estímulo de la derecha para desplazarse dos veces en esa dirección y luego observar al estímulo superior para realizar la acción. Y luego, para apagarlo, el usuario tendrá que navegar una vez más hacia la derecha para acceder a la acción de apagar el enchufe y luego mirar el estímulo superior para ejecutar el apagado.

Y adicionalmente, se adoptó un método de colocación de los electrodos en los usuarios. El cual consiste en: (1) identificar la ubicación del canal Oz en el usuario palpando el cráneo con la yema de los dedos para encontrar el Inion, es decir la parte más prominente del



hueso occipital, y con ayuda de una cinta métrica buscar que el punto este centrado con la nariz del usuario. (2) Trazar una línea recta en el cuero cabelludo, siguiendo el punto del canal Oz, para separar el pelo, con ayuda de pines para el cabello, y dejar visible el cuero cabelludo. (3) Higienizar la superficie con alcohol y algodón. (4) Colocar crema en el punto identificado del canal Oz con ayuda de una paleta y colocar el electrodo sobre la crema. (5) Medir aproximadamente dos dedos hacia la derecha y hacia la izquierda del canal Oz para identificar los puntos O1 y O2. (6) Realizar el paso 4 para estos dos canales. (7) Pasar el cable de los electrodos sobre la cabeza del usuario hacia la parte delantera. (8) Higienizar los lóbulos del usuario con un algodón con alcohol y colocar crema conductiva sobre cada lóbulo para luego colocar los electrodos sobre esta crema. (9) Cubrir los electrodos de los lóbulos con esparadrapo para evitar que se muevan. Y (10) finalmente se realiza la conexión de los electrodos al OpenBCI de la siguiente forma; el O1 al primer pin, el Oz al segundo pin y el O2 al tercer pin; el electrodo del lóbulo derecho se conectó al pin de referencia y el del lóbulo izquierdo al pin de tierra.

### **3.6.3. Experimentación**

Durante esta etapa, se realizaron las pruebas en 15 personas con un rango entre de edad entre 20 a 25 años y que declararon no sufrir de epilepsia, debido a que observar los SSVEPs podría causar convulsiones en personas con este trastorno.

Antes de empezar con las pruebas, se familiarizó al usuario con los estímulos; cada usuario debía ver a cada estímulo por 20 segundos y se les explicó que pueden pestañear mientras realizan las pruebas, ya que esto no afecta los resultados. Luego, se situó al sujeto frente a la pantalla led con la GUI, se le conectó los electros de acuerdo con el método adoptado previamente y se le explicó cómo navegar por la misma, es decir, la equivalencia de los estímulos en la interfaz que se muestra en la Tabla 3. Además, se les indicó que cuando un botón se active cambiará a color verde inmediatamente y luego se activaría el dispositivo. Seguidamente, se procedió a realizar una prueba de familiarización con la navegación a través de la interfaz, utilizando únicamente los estímulos.

Finalmente, se pidió a los usuarios que realicen las 4 acciones definidas anteriormente, estos son; prender y apagar el foco, y prender y apagar un enchufe de la regleta. Y se registró el tiempo que se demoraba en realizar cada acción y los datos que emitía la clasificación. Los resultados de esta etapa se encuentran en la Tabla 4.

### 3.6.4. Resultados

El tiempo promedio que se obtuvo para encender el foco fue de 9,70 segundos y el tiempo promedio para apagarlo fue de 16,04 segundos. Y el tiempo promedio de encender el enchufe fue de 26,80 segundos y para apagarlo se sitúa en los 21,89 segundos. Esto se puede evidenciar en la Tabla 4.

	<b>Foco</b>		<b>Enchufe</b>	
	Encender (s)	Apagar (s)	Encender (s)	Apagar (s)
Promedio	9,70	16,68	26,80	21,89
Mínimo	7,02	14,07	21,76	13,6
Maximo	13,16	40,54	42,88	68,52

*Tabla 4. Tiempos obtenidos por los usuarios al utilizar el prototipo.*

Y, de las 126 clasificaciones realizadas en total para las navegaciones en la interfaz de los 15 sujetos, 3 fueron incorrectas, es decir, el modelo entregó una tasa de éxito del 97,61% al identificar el estímulo observado por el usuario.

### 3.6.5. Discusión de hallazgos

Los conocimientos generados en esta etapa de pruebas se han expuesto en la sección de Discusión.

## 4. DISCUSIÓN

El control de dispositivos IoT a través de un BCI ofrece toda una nueva gama de posibilidades para manipular objetos electrónicos con métodos poco comunes pero efectivos y que no requieren de acciones motoras sino de la actividad cerebral, por lo que personas con problemas de movilidad podrían ser capaces de accionar los dispositivos.

Se pueden utilizar distintos patrones cerebrales pero los SSVEPs tienen varias ventajas, se pueden detectar de forma no invasiva, colocando los electrodos con crema conductiva directamente sobre el cuero cabelludo. Se generan fácilmente y con un bajo costo, en este caso se necesitó únicamente 4 leds, 4 protoboards, cables macho hembra y una RaspberryPi para ejecutar el código que configura el parpadeo de los leds.

Además, el tiempo de capacitación a los usuarios para que puedan utilizar el sistema es corto, con una explicación de aproximadamente 4 minutos sobre cómo se realiza la navegación en la interfaz y lo que representa cada estímulo en ella, los usuarios ya son capaces de controlar los dispositivos por si mismos únicamente observando los estímulos.

El sistema construido tiene un grado de efectividad alto y tiempos de manipulación de los dispositivos aceptables. Y gracias a esto y a que no se necesita realizar ningún movimiento corporal para interactuar con los dispositivos electrónicos, podría ser implementado con éxito para mejorar la calidad de vida de personas que posean discapacidades del habla o motoras muy graves otorgándoles cierto grado de independencia.

Adicionalmente, el prototipo final es eficiente puesto que permite controlar 5 dispositivos, en tiempo real, y realizar dos acciones para cada uno de los enchufes de la regleta y seis para el foco inteligente, sumando un total de 14 acciones, haciendo uso únicamente de cuatro estímulos. En otros estudios de finalidad similar como [26] y [27], la cantidad de acciones disponibles se limita por el número de estímulos del sistema, ya que a cada estímulo le corresponde una acción. Estas propuestas son poco escalables a largo plazo puesto que más acciones requieren de más estímulos. Y es por esto que la solución propuesta aventaja a las anteriormente implementadas en esta perspectiva.

<b>Estudio</b>	<b>Precisión (%)</b>	<b>Número de Estímulos</b>	<b>Tipo de Estímulo</b>
[26]	87%	4	LCD
[27]	84.8%	4	LED
[28]	92.8%	Desconocido	Realidad Aumentada (Hololens)

[29]	85.7%	2 - 4	Realidad Aumentada (Lentes inteligentes)
Sistema Propuesto	97.61%	4	LED

Tabla 5. Comparación de resultados con otros estudios de control de dispositivos IoT basados en SSVEP

Sin embargo, la solución propuesta también presenta pequeñas limitaciones como: su tamaño, ya que la pantalla y los estímulos se encuentran asegurados sobre una placa de madera de 49x49 cm y que las personas deben enfocarse completamente en el estímulo que desean activar, ya que durante las pruebas se pudo observar que cuando el usuario es distraído por factores ambientales como el ruido, toma más tiempo obtener una clasificación exitosa y, por ende, la navegación en la interfaz se vuelve más lenta.

Otros factores que se deben tomar en cuenta son: (1) luego de largas jornadas de uso, los usuarios registran cansancio visual por las molestias que causa observar los leds durante un largo periodo de tiempo y (2) el uso de este sistema está contraindicado para personas que padezcan de epilepsia pues podría ocasionar convulsiones.

## 5. CONCLUSIONES

A través del presente trabajo, se logró transformar los estímulos SSVEPs generados por LEDs parpadeando a 4 distintas frecuencias (7, 8, 9 y 10 Hz) en comandos operativos, a través de un algoritmo de clasificación elaborado con CCA que compara la señal adquirida con señales de referencia sinusoidales para emitir la clase a la cual pertenece dicha señal.

Como producto final de este trabajo, se obtuvo un prototipo funcional que consta de varios componentes: (1) la recolección de señales del usuario a través del OpenBCI, (2) los estímulos SSVEP generados a través de LEDs parpadeando a frecuencias distintas que desencadenan la respuesta cerebral en el usuario, (3) la clasificación y filtrado de las señales acopiadas que se realiza a través de la herramienta OpenVibe, (4) la interfaz de usuario, que posee la lista de los dispositivos IoT disponibles para manipular, y que utiliza las clasificaciones que recibe desde el algoritmo de clasificación para desplazarse entre las acciones. Y (5) los dispositivos disponibles para manipular, que en este caso son un foco y una regleta con cuatro enchufes individuales, que reciben la acción seleccionada por el usuario a través de un bróker MQTT.

Gracias a que se tiene la libertad de conectar cualquier dispositivo eléctrico a cada uno de los cuatro enchufes de la regleta, la variedad de electrodomésticos que se podría controlar es amplia, y se adaptaría a las necesidades del usuario. Adicionalmente el prototipo es escalable pues se podrían agregar más dispositivos inteligentes que puedan ser manipulados a través de APIs para desarrolladores o dispositivos eléctricos a los cuales se les haya agregado algún controlador.

Finalmente, las pruebas realizadas al prototipo, con 15 usuarios, brindaron resultados prometedores pues se obtuvo una tasa de éxito del 97.61% y un tiempo promedio de 9,70 segundos para encender un foco inteligente y 16,04 para apagarlo y para uno de los enchufes de la regleta se obtuvo un tiempo de 26,80 segundos para encenderlo y para apagarlo 21,89 segundos.

## 6. BIBLIOGRAFÍA

- [1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.
- [2] M. Bhayani, M. Patel, and C. Bhatt, "Internet of Things (IoT): In a Way of Smart World," vol. 438, 2016, pp. 343–350.
- [3] G. Logroscino et al., "Global, regional, and national burden of motor neuron diseases 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016," *Lancet Neurol.*, vol. 17, no. 12, pp. 1083–1097, 2018, doi: 10.1016/S1474-4422(18)30404-6.
- [4] Global status report on road safety 2018. Geneva: World Health Organization; 2018. Licence: CC BYNC-SA 3.0 IGO.
- [5] U. Hoffmann, J.-M. Vesin, and T. Ebrahimi, "Recent Advances in Brain-Computer Interfaces," Nov. 2007, doi: 10.1109/MMSP.2007.4412807.
- [6] J. Chen, D. Zhang, A. K. Engel, Q. Gong, and A. Maye, "Application of a single-flicker online SSVEP BCI for spatial navigation," *PLoS One*, vol. 12, no. 5, pp. 1–13, 2017, doi: 10.1371/journal.pone.0178385.
- [7] L. Zhang, J. Liu, X. Wu, X. Guo, B. Zhou, and Z. Ye, "Implement an asynchronous online SSVEP-based brain computer interface," *Proc. - 2017 10th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI 2017*, vol. 2018-January, pp. 1–5, 2018, doi: 10.1109/CISP-BMEI.2017.8302169.
- [8] B. He, S. Gao, H. Yuan, and J. R. Wolpaw, *Neural Engineering*. 2013.
- [9] B. Graimann, B. Allison, and G. Pfurtscheller, "Brain-Computer Interfaces - Revolutionizing Human-Computer Interaction," *Brain-Computer Interfaces*, pp. 1–27, 2010, doi: 10.1007/978-3-642-02091-9.
- [10] W. Zhang, F. Sun, C. Liu, W. Su, C. Tan, and S. Liu, "A hybrid EEG-based BCI for robot grasp controlling," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 3278–3283, doi: 10.1109/SMC.2017.8123134.
- [11] F. Beverina, G. Palmas, S. Silvoni, F. Piccione, and S. Giove, "User adaptive BCIs: SSVEP and P300 based interfaces," *PsychNology J.*, vol. 1, pp. 331–354, 2003.
- [12] T. Zander and N. Roussel, "Optimizing the use of SSVEP-based brain-computer interfaces for human-computer interaction To cite this version: Andéol Évain Optimizing the Use of SSVEP-based Beaudouin-Lafon for Human-Computer Interaction Gernot Müller-Putz," 2017.
- [13] F. Di Russo et al., "Spatiotemporal analysis of the cortical sources of the steady-state visual evoked potential," *Hum. Brain Mapp.*, vol. 28, no. 4, pp. 323–334, 2007, doi: 10.1002/hbm.20276.
- [14] R. Ortner, B. Z. Allison, G. Korisek, H. Gaggl, and G. Pfurtscheller, "An SSVEP BCI to control a hand orthosis for persons with tetraplegia," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 1, pp. 1–5, 2011, doi: 10.1109/TNSRE.2010.2076364.
- [15] T. C. Technologies, "10 / 20 System Positioning," *Gloucester Road*, no. 2410, pp. 77–79, 2012.
- [16] M. Okamoto et al., "Three-dimensional probabilistic anatomical cranio-cerebral correlation via the international 10-20 system oriented for transcranial functional brain mapping," *Neuroimage*, vol. 21, no. 1, pp. 99–111, 2004, doi: 10.1016/j.neuroimage.2003.08.026.
- [17] S. Poveda Zavala, J. Luis Murillo López, K. Ortíz Chicaiza, and S. Guun Yoo, "Review of Steady State Visually Evoked Potential Brain-Computer Interface

- Applications: Technological Analysis and Classification,” *J. Eng. Appl. Sci.*, vol. 15, no. 2, pp. 659–678, 2019, doi: 10.36478/jeasci.2020.659.678.
- [18] D. M. Witten and R. J. Tibshirani, “Extensions of sparse canonical correlation analysis with applications to genomic data,” *Stat. Appl. Genet. Mol. Biol.*, vol. 8, no. 1, 2009, doi: 10.2202/1544-6115.1470.
- [19] P. Bajorski, “Canonical Correlation Analysis,” vol. 1, pp. 241–259, 2011, doi: 10.1002/9781118121955.ch8.
- [20] P. Zheng and X. Gao, “Fixed-point CCA algorithm applied to SSVEP based BCI system,” *Proc. 2013 IEEE Symp. Comput. Intell. Cogn. Algorithms, Mind, Brain, CCMB 2013 - 2013 IEEE Symp. Ser. Comput. Intell. SSCI 2013*, no. 1, pp. 107–114, 2013, doi: 10.1109/CCMB.2013.6609173.
- [21] N. Peng et al., “Control of a nursing bed based on a hybrid brain-computer interface,” in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 1556–1559, doi: 10.1109/EMBC.2016.7591008.
- [22] C. J. Perera, I. Naotunna, C. Sadaruwan, R. A. R. C. Gopura, and T. D. Lalitharatne, “SSVEP based BMI for a meal assistance robot,” in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 2295–2300, doi: 10.1109/SMC.2016.7844580.
- [23] C. Yang, H. Wu, Z. Li, W. He, N. Wang, and C. Su, “Mind Control of a Robotic Arm With Visual Fusion Technology,” *IEEE Trans. Ind. Informatics*, vol. 14, no. 9, pp. 3822–3830, 2018, doi: 10.1109/TII.2017.2785415.
- [24] C. Zhang, Y. Kimura, H. Higashi, and T. Tanaka, “A simple platform of brain-controlled mobile robot and its implementation by SSVEP,” in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 1–7, doi: 10.1109/IJCNN.2012.6252579.
- [25] P. Pelayo, H. Murthy, and K. George, “Brain-Computer Interface Controlled Robotic Arm to Improve Quality of Life,” in *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, 2018, pp. 398–399, doi: 10.1109/ICHI.2018.00072.
- [26] R. Zhao, Y. Qiao, Y. Zhu, and Y. Wang, “A SSVEP intelligent home service system based on CCA,” in *2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE)*, 2017, pp. 495–499, doi: 10.1109/CCSSE.2017.8087982.
- [27] P. Virdi, P. Syal, and P. Kumari, “Home automation control system implementation using SSVEP based brain computer interface,” in *2017 International Conference on Inventive Computing and Informatics (ICICI)*, 2017, pp. 1068–1073, doi: 10.1109/ICICI.2017.8365304.
- [28] S. Park, H. S. Cha, J. Kwon, H. Kim, and C. H. Im, “Development of an Online Home Appliance Control System Using Augmented Reality and an SSVEP-Based Brain-Computer Interface,” *8th Int. Winter Conf. Brain-Computer Interface, BCI 2020*, 2020, doi: 10.1109/BCI48061.2020.9061633.
- [29] A. Saboor et al., “SSVEP-based BCI in a smart home scenario,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10306 LNCS, pp. 474–485, 2017, doi: 10.1007/978-3-319-59147-6\_41.
- [30] F. Putze, D. Weib, L. M. Vortmann, and T. Schultz, “Augmented reality interface for smart home control using SSVEP-BCI and eye gaze,” *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, vol. 2019-October, pp. 2812–2817, 2019, doi: 10.1109/SMC.2019.8914390.

- [31] J. S. Lin and C. H. Hsieh, "A Wireless BCI-Controlled Integration System in Smart Living Space for Patients," *Wirel. Pers. Commun.*, vol. 88, no. 2, pp. 395–412, 2016, doi: 10.1007/s11277-015-3129-0.
- [32] S. Saeed, N. Z. Jhanjhi, M. Naqvi, and M. Humayun, "Analysis of software development methodologies," *Int. J. Comput. Digit. Syst.*, vol. 8, no. 5, pp. 445–460, 2019, doi: 10.12785/ijcds/080502.
- [33] P. S. Ganney, S. Pisharody, and E. Claridge, "Software engineering," in *Clinical Engineering*, Elsevier, 2020, pp. 131–168.
- [34] Susman, G.I. and Evered, R.D. "An Assessment of the Scientific Merits of Action Research," *Administrative Science Quarterly*, (23) 1978, pp. 582-603.
- [35] M. Staron, *Action Research in Software Engineering*. 2020.
- [36] Electronjs.org. 2021. *Electron | Build cross-platform desktop apps with JavaScript, HTML, and CSS*. [online] Available at: <<https://www.electronjs.org/>> [Accessed 1 May 2021].
- [37] Beebotte.com. 2021. *Beebotte MQTT Support*. [online] Available at: <<https://beebotte.com/docs/mqtt>> [Accessed 2 May 2021].
- [38] D. Zhu, J. Bieger, G. Garcia Molina, and R. M. Aarts, "A survey of stimulation methods used in SSVEP-based BCIs," *Comput. Intell. Neurosci.*, vol. 2010, 2010, doi: 10.1155/2010/702357.
- [39] Z. Wu, Y. Lai, Y. Xia, D. Wu, and D. Yao, "Stimulator selection in SSVEP-based BCI," *Med. Eng. Phys.*, vol. 30, no. 8, pp. 1079–1088, 2008, doi: 10.1016/j.medengphy.2008.01.004.
- [40] I. Sugiarto, B. Allison, and A. Gräser, "Optimization strategy for SSVEP-based BCI in spelling program application," *Proc. - 2009 Int. Conf. Comput. Eng. Technol. ICCET 2009*, vol. 1, pp. 223–226, 2009, doi: 10.1109/ICCET.2009.189.
- [41] I. Volosyak, F. Gemblar, and P. Stawicki, "Age-related differences in SSVEP-based BCI performance," *Neurocomputing*, vol. 250, pp. 57–64, 2017, doi: 10.1016/j.neucom.2016.08.121.
- [42] S. Mouli and R. Palaniappan, "Eliciting higher SSVEP response from LED visual stimulus with varying luminosity levels," *2016 Int. Conf. Students Appl. Eng. ICSAE 2016*, pp. 201–206, 2017, doi: 10.1109/ICSAE.2016.7810188.
- [43] X. Gao, D. Xu, M. Cheng, and S. Gao, "A BCI-based environmental controller for the motion-disabled," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 11, no. 2, pp. 137–140, 2003, doi: 10.1109/TNSRE.2003.814449.
- [44] P. L. Lee et al., "An SSVEP-actuated brain computer interface using phase-tagged flickering sequences: A cursor system," *Ann. Biomed. Eng.*, vol. 38, no. 7, pp. 2383–2397, 2010, doi: 10.1007/s10439-010-9964-y.
- [45] C. Brunner, B. Allison, C. Altstätter, and C. Neuper, "A comparison of three brain-computer interfaces based on event-related desynchronization, steady state visual evoked potentials, or a hybrid approach using both signals," *J. Neural Eng.*, vol. 8, p. 25010, Mar. 2011, doi: 10.1088/1741-2560/8/2/025010.
- [46] K. B. Ng, A. P. Bradley, and R. Cunnington, "Stimulus specificity of a steady-state visual-evoked potential-based brain-computer interface," *J. Neural Eng.*, vol. 9, no. 3, 2012, doi: 10.1088/1741-2560/9/3/036008.
- [47] X. Chen, Y. Wang, M. Nakanishi, T. P. Jung, and X. Gao, "Hybrid frequency and phase coding for a high-speed SSVEP-based BCI speller," *2014 36th Annu. Int.*



- Conf. IEEE Eng. Med. Biol. Soc. EMBC 2014*, pp. 3993–3996, 2014, doi: 10.1109/EMBC.2014.6944499.
- [48] T. Cao, F. Wan, P. Mak, P.-I. Mak, M. Vai, and Y. Hu, “Flashing color on the performance of SSVEP-based brain-computer interfaces,” *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 2012, pp. 1819–1822, Aug. 2012, doi: 10.1109/EMBC.2012.6346304.
- [49] M. Aljshamee, M. Mohammed, R.-U.-A. Choudhury, A. Malekpour, and P. Luksch, “Beyond Pure Frequency and Phases Exploiting: Color Influence in SSVEP Based on BCI,” *Comput. Technol. Appl.*, vol. 5, 2014, doi: 10.17265/1934-7332/2014.02.008.
- [50] J. Murillo et al., “Study of the Influences of Stimuli Characteristics in the Implementation of Steady State Visual Evoked Potentials based Brain Computer Interface Systems”, In Press.
- [51] OpenViBE. 2021. *Discover OpenViBE*. [online] Available at: <<http://openvibe.inria.fr/discover/>> [Accessed 2 May 2021].

## 7. ANEXOS

### 7.1. Anexo I. Artículo Científico

#### **An online BCI system based in SSVEPs to control IoT devices**

Johanna Carolina Cerezo Ramírez

Departamento de Informática y Ciencias de la Computación, Escuela Politécnica Nacional, Quito, Ecuador  
Smart Lab, Escuela Politécnica Nacional, Quito, Ecuador  
johanna.cerezo@epn.edu.ec

José Luis Murillo López

Departamento de Informática y Ciencias de la Computación, Escuela Politécnica Nacional, Quito, Ecuador  
Smart Lab, Escuela Politécnica Nacional, Quito, Ecuador  
jose.murillo01@epn.edu.ec

Sang Guun Yoo

Departamento de Informática y Ciencias de la Computación, Escuela Politécnica Nacional, Quito, Ecuador  
Smart Lab, Escuela Politécnica Nacional, Quito, Ecuador  
sang.yoo@epn.edu.ec

\* Corresponding author

#### **Abstract**

Internet of Things allows devices to interconnect and communicate each other and with users. This way, people can control and monitor these devices using applications, voice commands or gestures; it is a technology that makes people's life easier. However, people with severe physical disabilities cannot take advantage of this technology since the way of controlling smart devices were designed for people without disabilities. In this situation, it is necessary to find other means of communication with the devices such as Brain-Computer Interface which is a technology that seeks to connect the user's brain activity with any external application. In this work, a Brain-Computer Interface system based on SSVEP was built to control IoT devices. The brainwaves generated by the SSVEP stimuli were acquired using electroencephalography and then translated into operational commands to control an application to control different IoT devices using the MQTT protocol. The final prototype was tested on fifteen volunteers, achieving an average accuracy of 97.61%, requiring an average time of 9.7 seconds to turn on a smart light bulb and 16.68 seconds to turn it off.

**Keywords:** SSVEP, Brain-Computer Interface, BCI, Internet of Things, IoT

#### **1- Introduction**

Internet of Things (IoT) is the interconnection of different types of devices to create a smart environment, allowing people to control or access electronic devices in their environment through voice commands, gestures, or movements [1]. The popularity of this technology has increased greatly in recent years because of cost reduction and ease of use of smart devices [2].

Although one might think that IoT is an all-inclusive technology, this is a hasty conclusion. Even though this technology makes the life easier to most of the people, it does not consider people with different types of severe movement disabilities that are not capable of performing any movement to interact with any device or people with more severe disabilities who cannot even

speak e.g., people that suffer of Motor Neuron Disease (MND). MND is a set of neurodegenerative disorders related to the degeneration of the superior or inferior motoneuron that leads to a progressive loss of upper and lower body movement [3]. By 2016, patients with MND reached 330 918 people around the world. In addition, the number of people who have been permanently or temporally disabled due to traffic accidents ascends between 10 to 50 million [4]. In this situation, the development of technologies based on Brain Computer Interfaces (BCI) could help to the excluded population explained before.

BCI is a communication system that translates the brain activity signals into understandable commands for a computer or other smart devices [5]; in other words, a BCI allows users to interact with electronic devices using their brain signals.

*Imagen 12. Captura de la página 1 del artículo.*

When a BCI system is implemented, there are a variety of brain patterns that can be used. Some of them are: P300, Motor Imagery, Error Potentials, Intentional Blinking, and Steady-State Visually Evoked Potentials (SSVEPs). The latter are generated when the user is exposed to a visual stimulus flickering at frequencies between 3.5 and 75 Hz [6]. The advantages of SSVEP are that it has a high transfer rate, it can be easily generated at a low cost [7], it can be detected noninvasively through electroencephalography (EEG), and it requires just a little user training for getting accustomed to the system [8].

There are different algorithms that can be used to classify signals generated by SSVEPs, being the most popular the Canonical Correlation Analysis (CCA) [9]. Other methods such as Fast Fourier Transform (FFT) and Linear Discriminant Analysis (LDA) have also been used traditionally. On the other hand, the use of Artificial Intelligence algorithms has increased lately. Support Vector Machine (SVM), Convolutional Neural Network (CNN) and Adaptive Feedforward Neural Network (AFNN) are some of the most widely used artificial intelligence algorithms which offer more accuracy but has the limitation of requiring more time and data to be trained.

Among the different algorithms, CCA, which is a classical method for determining the relationship between at least two sets of multidimensional variables [10], has been identified as the most widely used algorithm in the development of SSVEPs based BCI systems because of its high stability and performance [11]. Additionally, its usage has become popular because it is less user-dependent than other algorithms and allows the usage of harmonic frequencies [9]. These reasons make CCA the perfect fit for the development of this solution.

Based on the aforementioned information, the objective of this work is to build a BCI solution that allows people with severe movement disabilities to control IoT devices through their brainwaves. This objective will be achieved through a BCI device that will collect SSVEP signals from the user's occipital lobe using a noninvasive acquisition technique. On the other hand, the visual stimuli for the SSVEP signals will be generated by using light emitting diodes (LEDs) which will be blinking at different frequencies.

## 2- State of Art

Several studies have proposed SSVEPs based BCI systems to improve the quality of life of people with temporary or permanent mobility disabilities. These systems enabled the control of a medical bed [12], feeding system for hospital patients [13], and robotic arms [14] [15] [16]. There are also studies that implement SSVEPs based BCI systems to control smart devices, e.g., blinds and a light are controlled using four stimuli (8, 9, 11 and 12 Hz) in [17]. This work reached a success rate of 87% and an average time of 4 seconds. Even though this work is very interesting, it has an important limitation. The system controls only two devices using four stimuli, that is, for each device, two stimuli are needed to perform on/off actions. This is not the most efficient approach because if the user requires to add more devices or actions, the system will require the addition of more stimuli.

In [18], an offline system to control different IoT devices such as a light, fan, television, and alarm clock using four stimuli (10, 11, 12 and 13 Hz) was built. This work achieved an average success rate of 84,8%, but it was tested with only two users. In this work, for each device, a stimulus is needed to turn on/off the device. However, for each device added to the system, the number of stimuli needs to be increased too and only two actions (on/off) could be performed per device.

In [19], SSVEP stimuli are generated using the Augmented Reality (AR) technology and they are used to control home appliances. This work achieved a success rate of 92.8%. However, this work does not give enough information to replicate the prototype i.e. the number and frequency of stimuli. Additionally, the work does not mention which devices they were able to control nor the action they were able to execute on them.

In [20], AR is also combined with SSVEPs to develop a system capable of controlling a light bulb, an elevator, and a coffee machine. The solution created in this work achieved an average success rate of 87.5% on seven users. This work used four stimuli at frequencies of 6, 7, 8 and 9 Hz, and a QR code system to identify the devices that the user wants to control. When the user decides to control the bulb, he/she must scan the QR code first, and two stimuli will be displayed



to turn on or off the light. Depending on the device to control, the glasses show the stimuli according to the actions that can be executed. Although this research has an interesting approach, it was not designed for people with physical disabilities since users had to move from one place to another to access the QR codes that were in different parts of the building in which the tests were carried out.

In [21], an online system was built to control a light bulb, a television and blinds using a combination of SSVEPs with eye tracking system implemented via HoloLens. This system used four stimuli at the frequencies of 4, 6, 19 and 15 Hz. An action was assigned for each stimulus allowing the prototype to implement four actions. The system was tested with eleven users and achieved an average success rate of 76.1% for SSVEPs and 89.3% for the combination of eye tracking with SSVEPs. Nevertheless, to test this prototype the participants had to walk around the room to switch between devices, which means that this solution could not be used by people with movement disabilities.

As we can see in the analyzed works, during the last years, SSVEP based BCI systems which control IoT devices have been developed. However, most of them use a simple non scalable approach like assign one stimulus to each device and allow only one on/off action per device. Therefore, the present work aims to develop a SSVEP based BCI system using CCA as its classification algorithm with a Graphic User Interface (GUI) that will list the available devices and actions to control them. The GUI will have the option to navigate through the list of devices and actions and thus the number of devices and actions available will not depend on the number of stimuli of the system.

### 3- Methodology

Incremental methodology is a method of software engineering that is based on step-by-step implementation as an expanding model [22]. In this methodology, the product is designed, implemented, and tested in several stages. More features are added to the product in each stage or increment until the final version is reached.

Each increment has four stages [23]. The first is analysis, where requirements and specifications are gathered. Then, the design phase is followed; once the requirements are understood, a plan is made to implement and fulfill these requirements. Next is the coding phase, where the code is created. And finally, the testing phase is performed, where the prototype is tested.

In this work, an adaptation of the incremental methodology will be used. Since the prototype to be developed goes beyond software development, the coding phase has been replaced by a development phase that will cover the implementation of the prototype's different features. The other stages will remain the same.

Once the prototype is completely built, the Action Research methodology will be used. This methodology is based on the relationship between experimentation and knowledge generation. It can be used linearly or as a cycle depending on the needs of the research [24]. Although this methodology was built to study social phenomena, it has recently gained popularity among other fields, especially in computer science [25]. This is because it prioritizes knowledge generation and because it has been considered a generic empirical research method. The phases of Action Research depend on the research approach, but it should include the following five steps: (1) Diagnosing, (2) Action Planning, (3) Action Taking, (4) Evaluating and, (5) Specify Learning. This methodology fits the last phase of this work's needs since it allows defining the best course of action to test the complete prototype.

### 4- Development of the Solution

The proposed solution will contain four increments that will contribute to build the final prototype. Once created the final prototype the test phase will be performed. In the first increment, the user interface that will show the actions and available IoT devices will be built. The second increment will focus on implementing the communication with IoT devices through a Message Queuing Telemetry Transport (MQTT) broker. Later, in the third increment, the SSVEPs stimuli generated by LED lights will be deployed. In the fourth increment, the classification algorithm and the

signals' acquisition strategy will be developed. And, in the last phase, the final prototype will be tested. The first four increments will be tested with two to four people, but in the last phase, the completed prototype will be tested with fifteen volunteers. The architecture of the final prototype is shown in Figure 1.

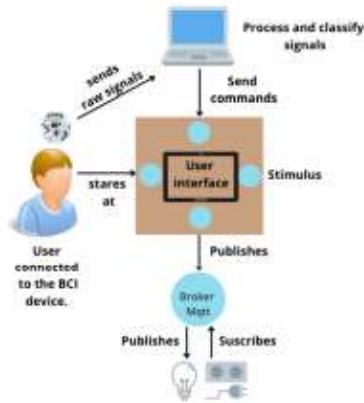


Fig. 1 System Architecture

### 3-1- First Increment

In this increment, the application with the user interface that shows the list of available actions and devices was built.

#### 3-1-1- Analysis

In order to provide an easy navigation to the user, the interface will be composed of a main menu listing all the actions. This list will also contain the option to access to other part of the application in which the devices will be grouped by rooms, and when a device is selected, the available actions for such device will be displayed.

#### 3-1-2- Design

Mockups of the user interface were created to show the available actions in the center of the application, which is surrounded by four buttons. At the top is the accept button, which is used to perform the action. To navigate between the actions, two buttons are used i.e., one arrow to go to the left and one to go to the right. And at the bottom of the screen, a button that allows to go back or close the application is generated. As shown in figures 2 and 3, two mockups were made with different styles. The dark style was chosen because it is more comfortable for users' eyes.



Fig. 2 Mockup of the user interfaces in light style.

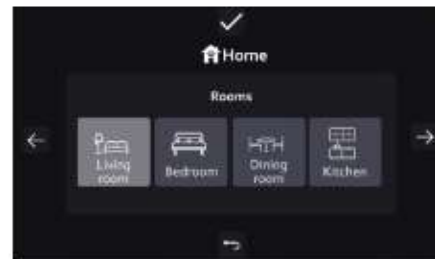


Fig. 3 Mockup of the user interface in dark style.

The user interface will be built using the Electron framework that allows building native desktop applications using HTML, CSS, and JavaScript. The applications developed with Electron are multiplatform and can be executed over different operating systems [26]. Additionally, it is open source and allows to create user friendly interfaces. To guarantee the persistence of the information, the SQLite database will be used, since it is a serverless and lightweight database engine that does not need a server. For the version control of the application, Git was chosen with GitHub for repository storage.

#### 3-1-3- Development

The coding of the application was done based on the parameters, technologies and frameworks listed in the previous subsections. Figure 4 shows the developed user interface.



Fig. 4 Screenshot of the developed user interface.



### 3-1-4 Test

The tests consisted in verifying the navigation of the interface with four people and the results were all correct.

### 3-2- Second increment

In this increment, an infrastructure for the IoT devices was built and it was integrated to the user interface application.

#### 3-2-1- Analysis

In this phase, it was necessary to implement an infrastructure that allows connecting the user interface with IoT devices (a smart bulb and a smart power strip with four plugs). This connection must be secure and fast.

#### 3-2-2- Design

The communication between the user interface application and the devices is going to be built using a Message Queuing Telemetry Transport (MQTT) broker that allows sending the user-selected action to the appropriate devices using the publish/subscribe messaging queue model.

The chosen MQTT broker is Beebotte MQTT which is based on a channel and resource model, where each channel can be related to several resources. It runs on version 3.1.1 of the MQTT protocol and provides SSL and non-SSL connections through the 1883 and 8883 ports, respectively [27]. In addition, it provides a reliable communication bridge using WebSockets, REST and MQTT.

#### 3-2-3- Development

A project was created in Beebotte platform containing one channel and five resources, corresponding to one smart bulb and four sockets of the smart power strip. A Python program was also developed to subscribe to the MQTT broker; it runs in the background to notify the actions the devices must perform when the user selects an action in the user interface. On the other hand, the ability to publish actions to the MQTT broker was added to the user interface.

#### 3-2-4 Test

The test phase was performed to verify the correct communication between the user interface and the IoT devices. These tests were conducted with four people who were asked to turn on/off the light and the first plug of the smart strip. The results were all satisfactory.

Delay tests were also performed to verify the time the prototype took for the bulb and smart power strip to turn on and off. To do this, the smart devices were turned on and off 35 times and the average time was measured. The results of these test are shown in Table 1.

Table 1. Average time on turning on/off smart devices.

Device	Action	Time (s)
Smart bulb	On	1.57
	Off	1.40
Smart Power Strip	On	0.42
	Off	0.38

Additionally, an extra analysis was performed to determine the delay corresponding to the communication through MQTT. For this experiment, two timestamps were recorded. The first one was recorded, when the message to activate the IoT device is sent from the user interface and the second one when it arrives to the subscribed device. These timestamps were used to obtain the time that the MQTT communication lasted. This test was also performed 35 times and the result is shown in Table 2.

Table 2. Average time of MQTT communication

Device	Action	Time(s)
Smart bulb	On	0,082
	Off	0,087
Smart Power Strip	On	0,094
	Off	0,084

Comparing the results shown in Tables 1 and 2, it is evident that the communication through the MQTT broker takes a trivial time of 0.08 seconds in the case of the smart bulb. Which means that it does not have a great impact on the performance of the device activation. The rest of the time corresponds to the time that the device takes for its activation.

### 3-3- Third Increment

In this phase, the SSVEPs stimulus were implemented using LEDs.

#### 3-3-1- Analysis:

Since the goal of this work is to implement a system based on SSVEPs, it is necessary to implement a device capable of generating the

visual stimuli. Four stimuli are needed to navigate the user interface: accept, left, right and back. In addition, the user interface must be placed around the stimuli, so that the user can observe it while looking at the stimuli.

### 3-3-2- Design

To generate the stimuli, two options were considered: LEDs and an LCD display. An LCD display can present problems to correctly represent several stimuli blinking at different frequencies at the same time, so the SSVEP response generated by a user staring at an LCD display may be weak [28][29]. Furthermore, only frequencies that are obtained by dividing the refresh rate of the display by an integer can be used [30]. Due to these limitations, it was decided to use LEDs, since the number of frequencies that can be represented is large, because there is no constraint such as the refresh rate that restricts the frequencies that can be used [31].

SSVEP stimuli usually originate from frequencies represented by integers in the mid-low frequency band. However, the decision of which frequencies to use is usually made arbitrarily by the researcher. Other works have used frequencies between 8 and 12 Hz [17], 10 and 13 Hz [18], 6 and 9 Hz [20], 7 and 10 Hz [32][33] with good results. In this work, frequencies located in the middle band were chosen i.e., 7, 8, 9 and 10 Hz.

In some studies, only the O2 and O1 channels of the international 10/20 system are used to acquire the signals generated by SSVEPs [34][35][36]. In other cases, an additional electrode is added in the Oz channel because this channel has the closest location to the visual cortex and therefore a good detection of the SSVEP response [37]. Additionally, it is possible to place electrodes in additional channels like POz, P3, P4 to gather SSVEP signals as made in [38] and [18]. However, the OZ, O1 and O2 channels have been defined as the main locations to acquire the user's brain activity elicited by SSVEPs since that activity is detected more clearly and consistently in most subjects when these channels are used [37]. For these reasons, in this work, it was decided to use channels OZ, O1 and O2.

In addition, it was decided to place the LEDs in a diamond position, as this allows a 600x1024 px LCD screen to be placed in the middle of the LEDs stimuli which will display the GUI, while the LEDs match the position of the GUI buttons. It was also decided to use white LEDs since several studies have shown that this color has good results with SSVEPs stimuli [39][40][41].

### 3-3-3- Development

The LEDs visual stimuli device were generated by using a Raspberry Pi with an application developed in Python which controls the frequency of LEDs. The frequencies were distributed as 7 Hz on top, 8 Hz on the right, 9 Hz on the bottom and 10 Hz on the left. The LEDs were placed in a 49cm square panel, each one covered by a diffuser filter to reduce the eye fatigue of users from looking directly at the LEDs [41].

### 3-3-4- Test

In this stage, tests were performed to verify that the frequency generated for each stimulus was correct. For this purpose, a 20-second signal was recorded from the occipital lobe (O1, O2 and Oz). The recorded data was then plotted to verify that the peaks reached in the amplitude vs. frequency plot matched the frequency generated by the stimulus.

This test was carried out with 4 people, and the stimuli were correctly generated for each frequency. The results are shown in the following figures (see Figures 5 to 8).

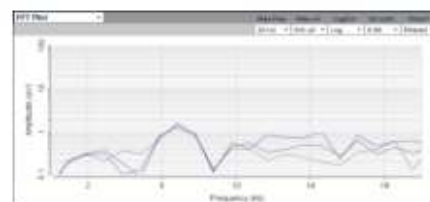


Fig. 5. The graph of the 7 Hz recording

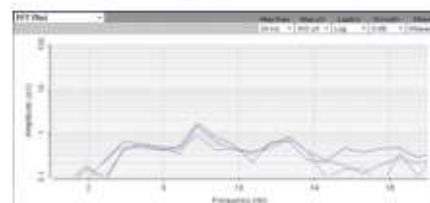


Fig. 6. The graph of the 8 Hz recording



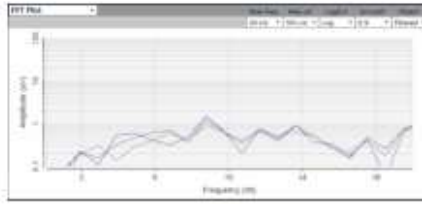


Fig. 7. The graph of the 9 Hz recording

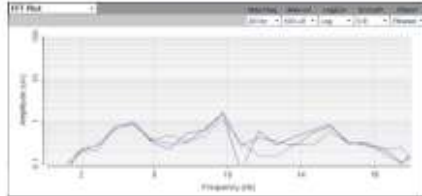


Fig. 8. The graph of the 10 Hz recording

### 3-4- Fourth Increment

In this increment, signal acquisition and filtering processes, and a CCA-based classification algorithm were implemented. These tasks were integrated with the user interface application.

#### 3-4-1- Analysis

To identify the stimulus that the user is looking at among the four frequencies, it is necessary to implement a classification algorithm of brain signals gathered from an EEG device.

#### 3-4-2- Design

For the collection of the brain signals, the Electroencephalography (EEG) device called OpenBCI was used. OpenBCI is an open source and low-cost platform that allows monitoring the electrical activity of the brain, muscles, and heart. On the other hand, for the processing and classification of the brain signals, a special software called OpenVibe was used, since it allows real time acquisition, filtering, classification, and visualization of brain signals [42]. Additionally, OpenVibe was selected since it supports OpenBCI. The classification of the signals will be carried out using Canonical Correlation Analysis (CCA), a multiclass classification algorithm that does not require training and has an average precision of 81.5% [9]. The communication between the classification algorithm and the user interface will be implemented through sockets.

#### 3-4-3- Development

In the OpenVibe tool, a brainwave classification scenario that makes use of the CCA was

modified. In this scenario, multiple steps are configured for acquisition, filtering, and processing of brain signals. The acquisition client was configured to receive data from the O1, O2 and Oz channels. A band pass filter was configured for the frequencies between 1 and 40 Hz. Additionally, the length of the signal to be collected for classification and the time window were configured i.e., a 3-second signal taken every 0.5 seconds.

The classification algorithm (CCA) that this scenario uses was developed in Python, and it was configured to work with four frequencies that will be received as parameters. In this algorithm, the input signal is compared with 4 reference signals, which correspond to the frequencies of 7, 8, 9 and 10 Hz, to find their correlation and emit a classification.

The algorithm output corresponds to the class number assigned to the frequencies following the order in which the frequencies were inserted in the scenario's parameters. In this case, for 7 Hz, the corresponding class is 1; for 8 Hz, the corresponding class is 2, and so on, in such a way that we will obtain the equivalence between action in the interface, frequency and classification shown in Table 3.

Table 3. Correspondence between interface actions, frequencies, and their classification

Frequency (Hz)	Classification	Interface Action
7	1	Accept
8	2	Right
9	3	Return
10	4	Left

Based on this equivalency table, the navigation in the user interface consists of stare at the stimulus on the right or left to move in the correspondent direction, stare at the top stimulus to execute the action, and stare at the bottom stimulus to exit the program.

The algorithm emits a classification every 0.5 seconds, but this is not suitable for the system because the interface's control becomes unstable, since too many commands are executed per second. For this reason, a second stage was implemented, in which the last 8 classifications



are stored in an array, of which 75% must belong to the same class to emit a valid classification.

Since the system can generate noise signal when the user switches his/her sight from one stimulus to another, or from one stimulus to the user interface, an inactivity time of 2.5 seconds was added when a valid classification has been sent. During this time the classifications made by the algorithm will not be considered to avoid unwanted selections in the interface.

The classifications recognized by the algorithm are transmitted to the user interface where the action that corresponds to them are performed. The GUI was also modified to change the color of the button (to green) for 2 seconds when it is activated, to give extra feedback to the user on which actions they are executing when observing the stimuli.

#### 3.4.4- Test

During this testing stage, the functionality of the prototype was verified (see Figure 9). It was possible to use the SSVEP stimuli to move between the list of actions of the interface. Two users were logged into OpenBCI and asked to navigate in the user interfaces and perform actions. The prototype worked perfectly, and users were able to navigate and execute actions through the interface without errors.

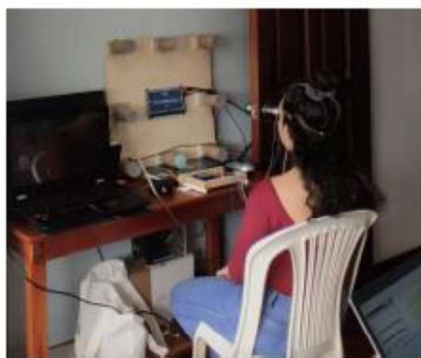


Fig. 9 Prototype in use

### 3-5- Test of the Integrated Prototype

At this stage, the Action Research methodology was used to define the best course of action to evaluate the functionality of the prototype.

#### 3-5-1- Diagnosing

The tests performed in the fourth increment were limited, which means that they could not be

considered reliable since the sample of users was too small.

#### 3-5-2- Action Planning and taking

At this stage, the test to be performed with the volunteers were defined. The test will consist in turning on and off the smart light bulb and one socket of the smart power strip. This process was performed to cover from the simplest to the most complex navigation steps in the user interface.

First, the user will have to turn the bulb on and off. Turning on is the first command of the interface, so it only requires accepting the action i.e., stare at the upper stimulus. To turn off the bulb, it is necessary to scroll to the right once and select the option. Secondly, the user will have to turn one socket of the smart power strip on and off. To turn it on, it requires scrolling to the right twice and selection the option; while turning it off, it requires a further scroll to the right and then the select the command.

The process for placing the electrodes on the user's scalp consisted of (1) finding the Inion, i.e., the most prominent part of the occipital bone; the Oz channel is located in the central part of this area. (2) Separate the hair with hairpins by drawing an imaginary horizontal straight line on the scalp over Oz. (3) Sanitize the scalp surface. (4) Apply conductive paste on Oz and then place the electrode over the paste. (5) Find the location of the O1 (left) and O2 (right) channels by measuring about half-inch from Oz. (6) Repeat step 4 for the channels O1 and O2. (7) Pass the electrodes leads over the user's head to the front. (8) Sanitize the earlobes, apply conductive paste, and then place the electrodes over the paste. (9) Cover the earlobe electrodes with tape to prevent them from falling off. (10) Connect the electrodes to the OpenBCI Cyton.

#### 3-5-3- Evaluating

During this stage, tests were carried out on fifteen volunteers, five women and ten men with an age range between 20 to 25 years, who declared not suffering from epilepsy. This verification was necessary since observing the SSVEPs could cause seizures in people with this disorder.

Before starting the tests, the user was familiarized with the stimuli. Each user had to see each stimulus for 20 seconds and it was explained that they can blink while performing the tests, since this activity does not affect the

results. Then, the subject was placed in front of the led screen with the GUI and he/she was explained how to navigate through it, that is, the equivalence between the stimuli and the interface shown in Table 3. In addition, they were told that when a button is activated, it will change to green color immediately and then the device will be activated. Next, a familiarization test with navigation was carried out through the interface, using only the stimuli.

Finally, the users were asked to carry out the four actions defined above, these are: turn the bulb on and off, and turn a socket of the smart power strip on and off. And the time required to perform each action and the data issued by the classification were recorded. The results of this stage are found in tables 4 and 5.

### 3-5-4- Results

The average time obtained to turn on the smart bulb was 9.70 seconds and the average time to turn it off was 16.04 seconds. And the average time to turn on the smart power strip was 26.80 seconds and to turn it off was 21.89 seconds. These results are shown in Tables 4 and 5.

Table 4. Tests results for the smart bulb

	On (s)	Off (s)
AVG	9,70	16,68
MIN	7,02	14,07
MAX	13,16	40,54

Table 5. Tests results for the smart power strip

	On (s)	Off (s)
AVG	26,80	21,89
MIN	21,76	13,6
MAX	42,88	68,52

From the 126 classifications performed in all of 15 subjects, only 3 were incorrect. This means that the model delivered a success rate of 97.61% when identifying the stimuli that were observed by the user.

### 3-5-5- Specify Learnings

The knowledge generated in this testing stage has been exposed in the discussion section.

## 4- Discussion

IoT device's control through a BCI system offers a whole new range of possibilities to manipulate electronic devices with unusual but effective methods, which could help people with physical disabilities. For this purpose, different brain patterns can be used, but the SSVEPs have several advantages. They can be detected in a non-invasively way placing the electrodes with conductive cream directly on the scalp; they are generated easily and at a low cost (in the present work, only four LEDs, four breadboards, some cables and a RaspberryPi were needed); in addition, the training time for users is short (a simple explanation of five minutes was enough).

It is also important to indicate that the final prototype can be considered efficient since it can control five devices, four sockets of the smart power strip and one smart bulb, in real time; and it can perform 14 actions (two actions for each socket of smart power strip and six for the smart bulb) using only four stimuli. In similar studies such as [17] and [18], the number of available devices/actions is limited by the number of stimuli configured in the system, since each stimulus corresponds to one action. Those approaches are not scalable in long term because to control more devices more stimuli must be added. Therefore, in this perspective, the proposed solution outperforms previously implemented ones (see Table 6).

Table 6. Result comparison with other SSVEP-based IoT

Reference	Precision (%)	# of stimuli	# of devices controlled	Stimulus type
[17]	87%	4	2	LCD
[18]	84,8 %	4	4	LED
[19]	92.8 %	Unkno wn	Unkno wn	Augment ed Reality
[20]	85.7 %	2 - 4	3	Augment ed Reality
Proposed System	97.61 %	4	5	LED

However, the proposed solution also presents some minor limitations. Firstly, the size of the



board where the screen and stimuli are placed is considerable (49x49 cm). Secondly, since the SSVEP requires the user to focus completely on the stimulus they wish to activate, the environmental noises can affect the activation speed of different devices of the system.

Other considerations to be aware of are: (1) after long periods of use, users register visual fatigue due to the discomfort caused by observing the LEDs for a long time and (2) the use of this system is contraindicated for people suffering from epilepsy as it could cause convulsions.

## 5- Conclusion

In the present work, SSVEP responses generated from stimuli delivered by LEDs flashing at four different frequencies (7, 8, 9 and 10 Hz) were transformed into operational commands to control different IoT devices using a CCA-based classification algorithm.

As final product of this work, a functional prototype consisting of several components was obtained. These components are (1) the user brain signals acquisition using the OpenBCI EEG, (2) the visual stimuli generated by LEDs flashing at different frequencies to trigger the SSVEP brain response in the user, (3) the brain signal processing and classification system using OpenViBE, (4) the user interface, which has the list of available IoT devices to control and its actions, and uses the output from the classification algorithm for navigation, and (5) the IoT devices (i.e., a light bulb and a power strip with four individual sockets) which receive the action selected by the user on the GUI through a MQTT broker.

Given the freedom to connect any electrical device to each of the sockets on the power strip, the variety of electrical appliances that could be controlled is wide, and therefore, the system could be adapted to the user's needs. Additionally, the prototype is scalable, since more smart devices can be added to the system if can be controlled through developer's API.

Finally, the tests conducted on 15 users provided promising results. An average precision of 97.61% was obtained through all the tests. The average time to turn on and off the smart bulb

was 9.70 and 16.04 seconds, respectively. As well as turning on and off one of the sockets of the power strip required an average time of 26.80 and 21.89 seconds, respectively. We believe that these results confirm the real possibilities of implementing the developed prototype in real life solutions for helping people with severe movement disabilities.

## 6- References

- [1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.
- [2] M. Bhayani, M. Patel, and C. Bhatt, "Internet of Things (IoT): In a Way of Smart World," vol. 438, 2016, pp. 343–350.
- [3] G. Logroscino et al., "Global, regional, and national burden of motor neuron diseases 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016," *Lancet Neurol.*, vol. 17, no. 12, pp. 1083–1097, 2018, doi: 10.1016/S1474-4422(18)30404-6.
- [4] Global status report on road safety 2018. Geneva: World Health Organization; 2018. Licence: CC BY-NC-SA 3.0 IGO.
- [5] U. Hoffmann, J.-M. Vesin, and T. Ebrahimi, "Recent Advances in Brain-Computer Interfaces," Nov. 2007, doi: 10.1109/MMSP.2007.4412807.
- [6] F. Beverina, G. Palmas, S. Silvoni, F. Piccione, and S. Giove, "User adaptive BCIs: SSVEP and P300 based interfaces," *PsychNology J.*, vol. 1, pp. 331–354, 2003.
- [7] J. Chen, D. Zhang, A. K. Engel, Q. Gong, and A. Maye, "Application of a single-flicker online SSVEP BCI for spatial navigation," *PLoS One*, vol. 12, no. 5, pp. 1–13, 2017, doi: 10.1371/journal.pone.0178385.
- [8] L. Zhang, J. Liu, X. Wu, X. Guo, B. Zhou, and Z. Ye, "Implement an asynchronous online SSVEP-based brain computer interface," *Proc. - 2017 10th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI 2017*, vol. 2018-January, pp. 1–5, 2018, doi: 10.1109/CISP-BMEI.2017.8302169.
- [9] D. M. Witten and R. J. Tibshirani, "Extensions of sparse canonical correlation analysis with applications to genomic data," *Stat. Appl. Genet. Mol. Biol.*, vol. 8, no. 1, 2009, doi: 10.2202/1544-6115.1470.
- [10] P. Zheng and X. Gao, "Fixed-point CCA algorithm applied to SSVEP based BCI system," *Proc. 2013 IEEE Symp.*

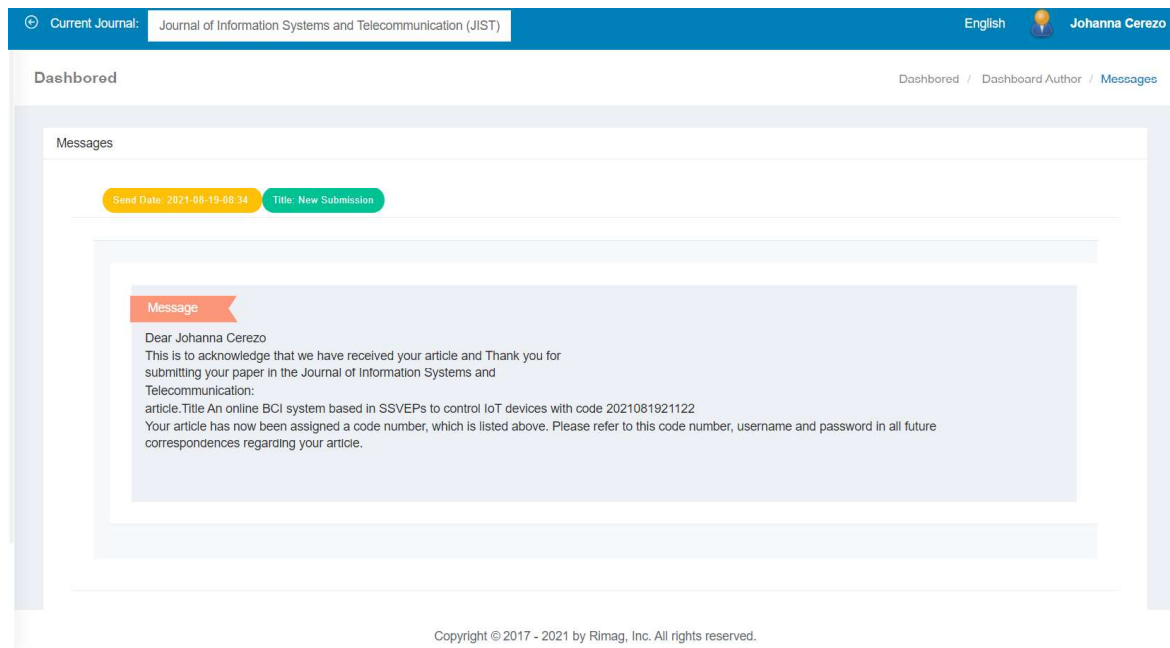
- Comput. Intell. Cogn. Algorithms, Mind, Brain, CCMB 2013 - 2013 IEEE Symp. Ser. Comput. Intell. SSCI 2013, no. 1, pp. 107–114, 2013, doi: 10.1109/CCMB.2013.6609173.
- [11] S. Poveda Zavala, J. Luis Murillo López, K. Ortíz Chicaiza, and S. Guun Yoo, "Review of Steady State Visually Evoked Potential Brain-Computer Interface Applications: Technological Analysis and Classification," *J. Eng. Appl. Sci.*, vol. 15, no. 2, pp. 659–678, 2019, doi: 10.36478/jeasci.2020.659.678.
- [12] N. Peng et al., "Control of a nursing bed based on a hybrid brain-computer interface," in 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2016, pp. 1556–1559, doi: 10.1109/EMBC.2016.7591008.
- [13] C. J. Perera, I. Naotunna, C. Sadaruwan, R. A. R. C. Gopura, and T. D. Lalitharatne, "SSVEP based BMI for a meal assistance robot," in 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2016, pp. 2295–2300, doi: 10.1109/SMC.2016.7844580.
- [14] C. Yang, H. Wu, Z. Li, W. He, N. Wang, and C. Su, "Mind Control of a Robotic Arm With Visual Fusion Technology," *IEEE Trans. Ind. Informatics*, vol. 14, no. 9, pp. 3822–3830, 2018, doi: 10.1109/TII.2017.2785415.
- [15] C. Zhang, Y. Kimura, H. Higashi, and T. Tanaka, "A simple platform of brain-controlled mobile robot and its implementation by SSVEP," in The 2012 International Joint Conference on Neural Networks (IJCNN), 2012, pp. 1–7, doi: 10.1109/IJCNN.2012.6252579.
- [16] P. Pelayo, H. Murthy, and K. George, "Brain-Computer Interface Controlled Robotic Arm to Improve Quality of Life," in 2018 IEEE International Conference on Healthcare Informatics (ICHI), 2018, pp. 398–399, doi: 10.1109/ICHI.2018.00072.
- [17] R. Zhao, Y. Qiao, Y. Zhu, and Y. Wang, "A SSVEP intelligent home service system based on CCA," in 2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE), 2017, pp. 495–499, doi: 10.1109/CCSSE.2017.8087982.
- [18] P. Viridi, P. Syal, and P. Kumari, "Home automation control system implementation using SSVEP based brain computer interface," in 2017 International Conference on Inventive Computing and Informatics (ICICI), 2017, pp. 1068–1073, doi: 10.1109/ICICI.2017.8365304.
- [19] S. Park, H. S. Cha, J. Kwon, H. Kim, and C. H. Im, "Development of an Online Home Appliance Control System Using Augmented Reality and an SSVEP-Based Brain-Computer Interface," 8th Int. Winter Conf. Brain-Computer Interface, BCI 2020, 2020, doi: 10.1109/BCI48061.2020.9061633.
- [20] A. Saboor et al., "SSVEP-based BCI in a smart home scenario," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10306 LNCS, pp. 474–485, 2017, doi: 10.1007/978-3-319-59147-6\_41.
- [21] F. Putze, D. Weib, L. M. Vortmann, and T. Schultz, "Augmented reality interface for smart home control using SSVEP-BCI and eye gaze," *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, vol. 2019-October, pp. 2812–2817, 2019, doi: 10.1109/SMC.2019.8914390.
- [22] S. Saeed, N. Z. Jhanjhi, M. Naqvi, and M. Humayun, "Analysis of software development methodologies," *Int. J. Comput. Digit. Syst.*, vol. 8, no. 5, pp. 445–460, 2019, doi: 10.12785/ijcds/080502.
- [23] P. S. Ganney, S. Pisharody, and E. Claridge, "Software engineering," in *Clinical Engineering*, Elsevier, 2020, pp. 131–168.
- [24] Susman, G.I. and Evered, R.D. "An Assessment of the Scientific Merits of Action Research," *Administrative Science Quarterly*, (23) 1978, pp. 582–603.
- [25] M. Staron, *Action Research in Software Engineering*. 2020.
- [26] Electronjs.org. 2021. Electron | Build cross-platform desktop apps with JavaScript, HTML, and CSS. [online] Available at: <<https://www.electronjs.org/>> [Accessed 1 May 2021].
- [27] Beebotte.com. 2021. Beebotte MQTT Support. [online] Available at: <<https://beebotte.com/docs/mqtt>> [Accessed 2 May 2021].
- [28] D. Zhu, J. Bieger, G. Garcia Molina, and R. M. Aarts, "A survey of stimulation methods used in SSVEP-based BCIs," *Comput. Intell. Neurosci.*, vol. 2010, 2010, doi: 10.1155/2010/702357.
- [29] Z. Wu, Y. Lai, Y. Xia, D. Wu, and D. Yao, "Stimulator selection in SSVEP-based BCI," *Med. Eng. Phys.*, vol. 30, no. 8, pp. 1079–1088, 2008, doi: 10.1016/j.medengphy.2008.01.004.
- [30] Sugiarto, B. Allison, and A. Gräser, "Optimization strategy for SSVEP-based BCI in spelling program application," *Proc. - 2009 Int. Conf. Comput. Eng. Technol.*



- IC CET 2009, vol. 1, pp. 223–226, 2009, doi: 10.1109/IC CET.2009.189.
- [31] T. Zander and N. Roussel, "Optimizing the use of SSVEP-based brain-computer interfaces for human-computer interaction To cite this version: Andéol Évain Optimizing the Use of SSVEP-based Beaudouin-Lafon for Human-Computer Interaction Gernot Müller-Putz," 2017.
- [32] Volosyak, F. Gembler, and P. Stawicki, "Age-related differences in SSVEP-based BCI performance," *Neurocomputing*, vol. 250, pp. 57–64, 2017, doi: 10.1016/j.neucom.2016.08.121.
- [33] S. Mouli and R. Palaniappan, "Eliciting higher SSVEP response from LED visual stimulus with varying luminosity levels," 2016 Int. Conf. Students Appl. Eng. ICSAE 2016, pp. 201–206, 2017, doi: 10.1109/ICSAE.2016.7810188.
- [34] X. Gao, D. Xu, M. Cheng, and S. Gao, "A BCI-based environmental controller for the motion-disabled," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 11, no. 2, pp. 137–140, 2003, doi: 10.1109/TNSRE.2003.814449.
- [35] P. L. Lee et al., "An SSVEP-actuated brain computer interface using phase-tagged flickering sequences: A cursor system," *Ann. Biomed. Eng.*, vol. 38, no. 7, pp. 2383–2397, 2010, doi: 10.1007/s10439-010-9964-y.
- [36] C. Brunner, B. Allison, C. Altstätter, and C. Neuper, "A comparison of three brain-computer interfaces based on event-related desynchronization, steady state visual evoked potentials, or a hybrid approach using both signals," *J. Neural Eng.*, vol. 8, p. 25010, Mar. 2011, doi: 10.1088/1741-2560/8/2/025010.
- [37] B. Ng, A. P. Bradley, and R. Cunnington, "Stimulus specificity of a steady-state visual-evoked potential-based brain-computer interface," *J. Neural Eng.*, vol. 9, no. 3, 2012, doi: 10.1088/1741-2560/9/3/036008.
- [38] X. Chen, Y. Wang, M. Nakanishi, T. P. Jung, and X. Gao, "Hybrid frequency and phase coding for a high-speed SSVEP-based BCI speller," 2014 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC 2014, pp. 3993–3996, 2014, doi: 10.1109/EMBC.2014.6944499.
- [39] T. Cao, F. Wan, P. Mak, P.-I. Mak, M. Vai, and Y. Hu, "Flashing color on the performance of SSVEP-based brain-computer interfaces," *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 2012, pp. 1819–1822, Aug. 2012, doi: 10.1109/EMBC.2012.6346304.
- [40] M. Aljshamee, M. Mohammed, R.-U.-A. Choudhury, A. Malekpour, and P. Luksch, "Beyond Pure Frequency and Phases Exploiting: Color Influence in SSVEP Based on BCI," *Comput. Technol. Appl.*, vol. 5, 2014, doi: 10.17265/1934-7332/2014.02.008.
- [41] Murillo et al., "Study of the Influences of Stimuli Characteristics in the Implementation of Steady State Visual Evoked Potentials based Brain Computer Interface Systems", In Press.
- [42] OpenViBE. 2021. Discover OpenViBE. [online] Available at: <<http://openvibe.inria.fr/discover/>> [Accessed 2 May 2021].

*Imagen 23. Captura de la página 12 del artículo.*

## 7.2. Anexo II. Confirmación de recepción del artículo en revista científica



*Imagen 24. Confirmación de recepción del artículo en revista científica*

## 7.3. Anexo III. Mock Up de la interfaz de usuario en colores claros

Enlace: <https://projects.invisionapp.com/prototype/CerezoJohanna-mockUpClaro-Titulacion-ckorucw8u007kem01m85vwkkr/play>

## 7.4. Anexo IV. Mock Up de la interfaz de usuario en colores oscuros

Enlace: <https://projects.invisionapp.com/prototype/ckorueujb00a4fa01zcmx6i11/play>

## 7.5. Anexo V. Pruebas de tiempos de encendido y apagado de los dispositivos a través de la interfaz

Foco		Enchufe	
Encender (s)	Apagar(s)	Endender(s)	Apagar(s)
1,62	1,61	0,46	0,72
1,77	1,27	0,73	0,46
1,75	1,75	0,51	0,62
1,24	1,32	0,47	0,47
1,96	1,41	0,63	0,43
1,78	1,27	0,49	0,72
1,56	1,40	0,46	0,52
1,4	1,21	0,49	0,51
1,84	1,43	0,74	0,54
1,49	0,74	0,53	0,44
1,37	1,38	0,47	0,58
1,62	1	0,78	0,45
1,43	1,4	0,49	0,41
1,74	1,39	0,42	0,24
2,04	1,49	0,42	0,29
1,38	1,22	0,27	0,33
1,54	1,49	0,43	0,27
1,34	1,38	0,35	0,23
1,4	1,63	0,39	0,28
1,35	1,4	0,29	0,34
1,74	1,34	0,33	0,27
1,41	1,43	0,35	0,36
1,42	1,37	0,35	0,29
1,13	1,4	0,27	0,4
1,66	1,68	0,46	0,27
1,9	1,46	0,35	0,29
1,54	1,4	0,31	0,31
2	1,63	0,37	0,32
1,37	1,67	0,38	0,26
1,46	1,48	0,34	0,35
1,55	1,48	0,26	0,3
1,53	1,42	0,32	0,26
1,73	1,59	0,36	0,35
1,67	1,23	0,35	0,33
1,46	1,38	0,36	0,43

Tabla 6. Resultados de las pruebas de delay de los dispositivos.

## 7.6. Anexo VI. Pruebas de tiempos de la comunicación por MQTT

Foco		Enchufe	
Encender (s)	Apagar (s)	Encender (s)	Apagar (s)
0,09827	0,088932	0,083651	0,088886
0,07987	0,080535	0,082265	0,082461
0,08137	0,080384	0,086606	0,087137
0,07981	0,083326	0,080904	0,082934
0,08075	0,080263	0,084993	0,082559
0,08079	0,091758	0,08931	0,083324
0,08201	0,164441	0,082906	0,081504
0,08874	0,082175	0,083407	0,080074
0,08321	0,082191	0,088391	0,081839
0,08215	0,081665	0,082581	0,08234
0,07972	0,080199	0,081697	0,080789
0,08305	0,080800	0,081237	0,081951
0,07894	0,084563	0,081885	0,081261
0,08078	0,081977	0,080958	0,08109
0,08176	0,081758	0,087859	0,08075
0,08012	0,080085	0,082619	0,080997
0,08036	0,081126	0,082713	0,082284
0,08494	0,080361	0,086261	0,082615
0,07959	0,079970	0,084454	0,082488
0,08042	0,080191	0,083067	0,081788
0,08145	0,152803	0,082395	0,081948
0,08064	0,079433	0,082344	0,082541
0,08146	0,080342	0,440336	0,081138
0,08006	0,079567	0,083974	0,083597
0,08150	0,081099	0,083531	0,101172
0,07976	0,079033	0,082637	0,082942
0,07991	0,081124	0,080822	0,120162
0,08101	0,081623	0,082375	0,080642
0,08071	0,082549	0,080288	0,081546
0,08534	0,081750	0,081725	0,080256
0,08176	0,083021	0,083202	0,082278
0,08200	0,103478	0,087357	0,082843
0,08228	0,084732	0,081263	0,081184
0,08203	0,085753	0,0934	0,086057
0,10045	0,084124	0,083303	0,081872

Tabla 7. Resultados de las pruebas de delay de la comunicacion por MQTT



## 7.7. Anexo VII. Resultados de las pruebas de manipulación de los dispositivos con el prototipo final.

### 7.7.1. Tiempos Foco

	Encender (s)	Apagar (s)
Sujeto 1	9,62	16,04
Sujeto 2	11,11	14,07
Sujeto 3	9,73	15,61
Sujeto 4	8,06	19,46
Sujeto 5	8,25	19,31
Sujeto 6	8,47	18,06
Sujeto 7	8,74	33,54
Sujeto 8	12,34	21,96
Sujeto 9	10,28	19,18
Sujeto 10	7,02	17,27
Sujeto 11	9,94	16,21
Sujeto 12	8,2	40,54
Sujeto 13	13,16	14,15
Sujeto 14	8,14	15,64
Sujeto 15	12,56	14,17

Tabla 8. Tiempos resultados de las pruebas de manipulación del foco por usuario.

### 7.7.2. Tiempos Enchufe

	Encender (s)	Apagar (s)
Sujeto 1	23,74	15,69
Sujeto 2	28,96	14,67
Sujeto 3	23,76	14,41
Sujeto 4	32,9	23,77
Sujeto 5	23,15	13,96
Sujeto 6	42,88	57,98
Sujeto 7	23,79	14,23
Sujeto 8	28,48	17,36
Sujeto 9	26,3	16,98
Sujeto 10	21,76	14,3
Sujeto 11	23,95	14,41
Sujeto 12	23,1	13,96
Sujeto 13	32,37	68,52
Sujeto 14	24,57	13,6
Sujeto 15	22,38	14,58

Tabla 9. Tiempos resultados de las pruebas de manipulación del enchufe por usuario.

### 7.7.3. Clasificaciones Foco

Clasificaciones Foco por Voluntario															
Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	
						4									
						1									

Tabla 10. Clasificaciones emitidas en las pruebas de manipulación del foco por usuario.

### 7.7.4. Clasificaciones Enchufe

Clasificaciones Enchufe por Voluntario															
Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto	Sujeto
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
2	2	2	2	2	4	2	2	2	2	2	2	2	2	2	
1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	1	1	1	1	4	1	
													2		
														1	

Tabla 11. Clasificaciones emitidas en las pruebas de manipulación del enchufe por usuario.