

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño e implementación de aplicación web orientada a la
medición de tiempo de reacción en el deporte.**

Trabajo de graduación presentado por Boggdan Josué Barrientos
Cifuentes para optar al grado académico de Licenciado en Ingeniería
Mecatrónica

Guatemala,

2019

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Diseño e implementación de aplicación web orientada a la
medición de tiempo de reacción en el deporte.**

Trabajo de graduación presentado por Boggdan Josué Barrientos
Cifuentes para optar al grado académico de Licenciado en Ingeniería
Mecatrónica

Guatemala,

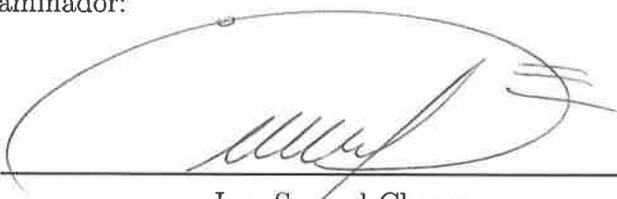
2019

Vo.Bo.:

(f) 

Ing. Samuel Chavez

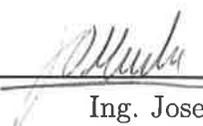
Tribunal Examinador:

(f) 

Ing. Samuel Chavez

(f) 

MSc. Miguel Zea

(f) 

Ing. Jose Morales

Fecha de aprobación: Guatemala, 8 de enero de 2019.

El tiempo y su forma de medirlo.

Un reloj, un cronómetro, un teléfono, todos nos permiten tener una referencia del tiempo en el que estamos o el tiempo que ha pasado desde un evento hasta otro. Desde que inicio el evento llamado: Ingeniería Mecatrónica, han transcurrido casi cinco años, que en tiempo es fácil de decir y medir pero que en eventos no. Han pasado tantos que no alcanzaría a detallarlos todos y cada uno me ha brindado algo para mí un aprendizaje, experiencia o lección.

Me he redescubierto a mí mismo. Conocí lugares nuevos, degusté nuevas comidas y como el buen conocedor de mi persona, seguí jugando al tenis de mesa.

Mientras escribo el proceso que realice para diseñar una aplicación web orientada a la medición de tiempo de reacción les quiero dejar un poco de mí. De tres pasiones que tengo: Tenis de mesa, viajar y comer, este último también lo relaciono con cocinar aunque no tengo certeza sobre la calidad de mis habilidades culinarias pero quién sabe, puede que sea algún día.

Que lo escrito anteriormente no te haga creer que solo hablo sobre mis pasiones, no es así. Hablo de los procesos que tuve que realizar uno a uno, casi desde cero en algunos para desarrollar la aplicación web.

Continúa con la lectura:

¡lo mejor está por llegar!

Prefacio	v
Lista de figuras	xii
Lista de cuadros	xiii
Resumen	xv
Abstract	xvii
1. Introducción	1
2. Antecedentes	3
2.1. Ciencia aplicada al deporte	3
2.2. Fitlight	4
3. Justificación	7
4. Objetivos	9
4.1. Objetivo general	9
4.2. Objetivos específicos	9
5. Alcance	11
6. Marco teórico	13
6.1. Aplicaciones móviles	13
6.1.1. Aplicación web	13
6.1.2. Aplicación web vs Aplicación nativa	14
6.2. Dispositivos móviles	15
6.2.1. Características generales	15
6.2.2. Tipos de dispositivos móviles	15
6.3. Redes de comunicación inalámbrica	17
6.3.1. Clasificación	17
6.4. Tecnologías para la web móvil	19

6.4.1.	Comunicación inalámbrica	19
6.4.2.	Exploradores web para móviles	21
6.5.	Tecnologías con capacidad de conexión inalámbrica	21
6.5.1.	ESP8266EX	22
6.5.2.	Raspberry Pi Zero W	22
6.5.3.	LinkIt Smart 7688 DUO	23
6.6.	Lenguajes de programación web	24
6.6.1.	Lenguajes del cliente	24
6.6.2.	Lenguajes del servidor	27
6.6.3.	Framework MVC	29
6.7.	Tiempo de reacción	31
6.7.1.	Conceptos básicos	31
6.7.2.	Medición y evaluación del tiempo de reacción	32
6.7.3.	Mejora del tiempo de reacción	33
7.	El arte de la selección	35
7.1.	Selección del tipo de aplicación	35
7.2.	Selección de tecnología que albergará la aplicación	37
7.2.1.	Sistema operativo	39
7.2.2.	Accesorios	39
7.3.	Selección del entorno de trabajo	40
7.4.	Selección de base de datos	41
8.	Aplicación Web	43
8.1.	Desarrollo	43
8.1.1.	Front End	43
8.1.2.	Back End	44
8.2.	Software de desarrollo	44
8.2.1.	Anaconda	45
8.2.2.	Visual Studio Code	45
8.2.3.	Atom	45
8.2.4.	GitHub	45
8.3.	Requerimientos de diseño	46
8.4.	Implementación	46
8.4.1.	Planteamiento del diseño	46
8.5.	Pruebas de funcionamiento	49
9.	Ejercicios de entrenamiento de tiempo de reacción	53
9.1.	Librerías de Node.js	53
9.1.1.	onoff	54
9.1.2.	socket.io	54
9.2.	Lógica de programación	55
9.2.1.	Tiempo de activación	56
9.2.2.	Tiempo de desactivación	57
9.2.3.	Generación de estímulos	57
9.2.4.	Captura del tiempo de reacción	58
9.2.5.	Resultados del entrenamiento	58
9.3.	Descripción de código	58

9.4. Interfaz entre Django y Node.js	60
10. Implementación de aplicación web en Raspberry Pi Zero W	63
10.1. Instalación de ambiente	63
10.1.1. Instalación del sistema operativo	63
10.1.2. Programas de comunicación	64
10.1.3. Configuración del cliente y comunicación SSH	64
10.2. Instalación e implementación de software	65
10.2.1. Instalación de Django	65
10.2.2. Instalación de PostgreSQL	66
10.3. Punto de acceso inalámbrico	67
10.3.1. De cliente a punto de acceso	67
10.3.2. De punto de acceso a cliente	68
10.4. Prueba de conexión	69
11. Conclusiones	71
12. Recomendaciones	73
13. Bibliografía	75
14. Anexos	79
14.1. Enlaces	79
14.2. Plantillas HTML	80
14.3. Entrenamientos predeterminados	85
14.3.1. Instrucciones	85
14.3.2. Diagramas de flujo	86
14.4. Circuito de prueba	89
14.4.1. Esquemático	89
14.4.2. Diagrama	90
14.4.3. Fotografía	91
15. Glosario	93

Lista de figuras

1.	Tablet como controlador y menú de aplicación de Fitlight	5
2.	Interfaz de aplicación Fitlight JR	5
3.	ESP8266EX	22
4.	Modelo Raspberry Pi Zero W	23
5.	LinkIt Smart 7688 DUO	23
6.	Regla de estilo CSS	25
7.	Flujo de información Modelo-Vista-Controlador	29
8.	Organización Modelo-Vista-Plantilla de Django	31
9.	Planteamiento general de funcionamiento del software	36
10.	Red WLAN para acceso de aplicación web	38
11.	UML de los modelos de descripción y sesión de entrenamiento	47
12.	UML de los modelos de descripción predeterminada y sesión de juego	48
13.	Mapa del sitio de la aplicación	50
14.	Plantilla de inicio de sesión	51
15.	Plantilla del menú principal	51
16.	Activación del ambiente de desarrollo	52
17.	Ejecución del servidor de la aplicación	52
18.	Ciclo de generación de tiempo de activación aleatorio	56
19.	Tiempos de activación	57
20.	Diagrama de flujo de captura de tiempo de reacción	59
21.	Diagrama de flujo de un entrenamiento general	61
22.	Interfaz de la aplicación concluido un entrenamiento	62
23.	Interfaz de Putty	65
24.	Interfaz de la aplicación accedida desde una tablet	70
25.	Plantilla de registro de usuario	80
26.	Plantilla de configuración de entrenamiento	80
27.	Plantilla de selección de juego	81
28.	Plantilla de instrucciones de juego	81

29.	Plantilla de configuración	82
30.	Plantilla de configuración de color de estímulo	82
31.	Plantilla de preguntas frecuentes	83
32.	Plantilla de resultados	83
33.	Plantilla de resultados detallados	84
34.	Diagrama de flujo:Juego 1	86
35.	Diagrama de flujo:Juego 2	87
36.	Diagrama de flujo: Juego 3	88
37.	Esquemático de conexión: Raspberry Pi Zero W, Leds, Pushbuttons	89
38.	Vista del esquemático y sus componentes	90
39.	Foto del circuito utilizado	91

Lista de cuadros

1.	Selectores CSS	26
2.	Comparativa entre aplicación nativa y aplicación web	37
3.	Comparación dispositivos inalámbricos	40
4.	Cuadro comparativo: Ambientes de desarrollo web	42
5.	Resultados de entrenamiento	60

El tiempo de reacción es el tiempo que transcurre entre un estímulo y el inicio de una respuesta. La aplicación web es un software que brinda las herramientas para generar y practicar ejercicios de entrenamiento orientado a la practica de tiempo de reacción. Puede ser utilizado para crear un entrenamiento bajo la elección de parámetros como lo son el tiempo total, período de activación del estímulo, cantidad de estímulos y dispositivos de captación de respuesta. Se toma registro del usuario y los resultados del entrenamiento se visualizan en tiempo real en la aplicación. Para la prueba de la aplicación se utilizaron estímulos visuales y botones como dispositivos de captación. Se utilizó un punto de acceso inalámbrico creado en una Raspberry Pi Zero W para acceder a la aplicación. Se provee el código fuente de la aplicación para poder adaptarlo a otras necesidades en la practica de tiempo de reacción, el potencial de nuevas adaptaciones o versiones es amplio. La última versión de la aplicación se encuentra disponible en GitHub [1].

The reaction time is the time that elapses between a stimulus and the start of a response. The web application is software that provides the tools to generate and practice training exercises aimed at the practice of reaction time. It can be used to create a training under the choice of parameters such as the total time, activation period of the stimulus, number of stimuli and response capture devices. The user's record is taken and the results of the training are displayed in real time in the application. For the test of the application, visual stimuli and buttons were used as capture devices. An access point was created in a Raspberry Pi Zero W to access the application. The source code of the application is provided to be able to adapt it to other needs in the practice of reaction time, the potential of new adaptations or versions is wide. The latest version of the application is available on GitHub [1].

El sistema de medición de tiempo de reacción en el deporte se inspira en la necesidad de entrenar, medir y llevar un control sobre importantes factores deportivos como la velocidad, concentración y reflejos. La Universidad del Valle de Guatemala, a través del departamento de Ingeniería Mecatrónica, diseñó e implementó un sistema de medición de tiempo de reacción. El sistema se dividió en dos módulos, un módulo de hardware que consiste en dispositivos de captación, capaces de detectar contacto o cercanía y otro módulo de software que consiste en una aplicación web descrita en el presente trabajo.

La aplicación web se desarrolló bajo la arquitectura Cliente-Servidor. El modelo de uso planteado sitúa a los clientes como dispositivos móviles y el servidor es provisto por una computadora Raspberry Pi Zero W donde este se ejecuta. Para el desarrollo del sistema se utilizaron lenguajes de programación como JavaScript, HTML, CSS, Python y como base de datos PostgreSQL. El desarrollo de la aplicación presentó un modelo secuencial en el cual se diferencia la fase de análisis de requerimientos, planteamiento del diseño, implementación y pruebas.

En el software se desarrollaron entrenamientos para el trabajo y práctica de ejercicios para la medición de tiempo de reacción. Un entrenamiento se define como una serie de estímulos visuales presentes en el dispositivo de captación, a los cuales la persona que realiza el entrenamiento debe reaccionar y desactivar estos estímulos mediante un gesto de contacto sobre el dispositivo de captación.

La aplicación web se utilizó para enviar y recibir información a los dispositivos de captación con la finalidad de activar estímulos y detectar respuestas ante ellos. Permitir al usuario configurar su tipo de entreno, realizar entrenamientos y consultar los resultados obtenidos, son otras de las funciones que posee la aplicación.

Para acceder a la aplicación se creó un punto de acceso inalámbrico al cual el dispositivo móvil se conecta y a través de un navegador web visualiza la aplicación web.

El tiempo de reacción ha sido estudiado desde mediados del siglo XIX, enfocado en el tiempo que toma realizar procesos mentales básicos.

El psicólogo F. C. Donders utilizó el *Hipp chronoscope*, un temporizador electromagnético de precisión que vibraba una frecuencia de 1000 Hz. Poseía un mecanismo de cuerda que se hacía girar continuamente por un motor accionado por un peso pesado. Al comienzo de una prueba de medición de tiempo de reacción, el mecanismo se puso en movimiento, pero se impidió mover las agujas indicadoras sobre su esfera mediante un embrague que se mantuvo en la posición desacoplada por medio de un solenoide eléctricamente activado. Cuando se interrumpió la corriente eléctrica a través del solenoide, el embrague se enganchó y el dial giró rápidamente. Cuando se reestableció la corriente, el embrague se desconectó y la esfera se detuvo en una lectura que mostraba el tiempo transcurrido en milésimas de segundo [2].

En la actualidad existen diversas plataformas y mecanismos que permiten la medición de tiempo de reacción y es importante conocer la interfaz con la cual interactúa el usuario para realizar su prueba.

2.1. Ciencia aplicada al deporte

El departamento de Ingeniería Mecatrónica en el año 2017 a través del megaproyecto de Ciencia aplicada al deporte se desarrolló un aparato inalámbrico para medir tiempo de reacción. Centralizado en el diseño mecánico y tipo de sensores menciona brevemente la interfaz que consistía en ingresar comandos como nombre y número de intentos a la consola de Python.

“no se creó un ejecutable ni una interfaz vistosa sino únicamente se creó un programa sencillo capaz de generar la base de datos” – extracto del trabajo de graduación de Ciencia Aplicada al Deporte 2017.

Utiliza una hoja de cálculo de Excel para almacenar los datos capturados, en esta hoja de cálculo se observan los parámetros de número de la prueba y el tiempo de cada una. Además, obtiene el valor mínimo, máximo y el promedio de los tiempos de reacción de las pruebas realizadas. El nombre de la hoja de cálculo es únicamente la hora a la cual se realizó el entrenamiento. El programa que generaba la hoja de cálculo se ejecutaba en una computadora.

No hay desarrollo de ejercicios de entrenamiento de tiempo de reacción, únicamente se presenta un estímulo en un tiempo no especificado, el documento no explica cómo se obtiene el tiempo de activación del estímulo, el programa espera hasta que el usuario reaccione ante él. Solamente deben existir aciertos en la respuesta del estímulo, no menciona un tiempo máximo para poder reaccionar por lo que hace notar que el estímulo podría estar activado minutos, horas y el programa seguiría contando este tiempo como un tiempo de reacción. El usuario no posee una interfaz gráfica a través de la cual pueda configurar su medición de tiempo de reacción, ni visualizar sus resultados inmediatamente al terminar sus pruebas, sino que debe ubicarse en el directorio donde se generan las hojas de cálculo con los tiempos resultantes para poder observarlos.

2.2. Fitlight

El FITLIGHT Trainer™ es un sistema de formación de tiempo de reacción, inalámbrico, compuesto de luces LED RGB controlados por una tableta. Las luces se utilizan como objetivos para que el usuario las desactive, pueden ser adaptados y configurados para todos los deportes y los regímenes de entrenamiento. Este sistema de entrenamiento está diseñado para capturar la reacción humana, orientado al tiempo de reacción, velocidad, agilidad, coordinación - todo medido para una retroalimentación del rendimiento inmediata. [3]

FITLIGHT CORP. comercializa dos productos principales, FITLIGHT y FITLIGHT JR, el primero es la opción más avanzada tanto en software como características de las luces, la segunda es una versión que concentra las características principales de su versión profesional.

Con la versión de FITLIGHT la interfaz del usuario que utiliza este sistema es una aplicación de Android preinstalada en una tablet, esta es parte del sistema de entrenamiento que comercializan. A través de la aplicación pueden generar un entrenamiento aleatorio, una secuencia programada, exportar resultados. La comunicación de controlador con las luces se realiza de forma inalámbrica.

La aplicación consta con cuatro botones en el menú de inicio como se observa en la Figura 1, los cuales permiten realizar un entrenamiento, juegos, verificar y establecer conexión con los módulos luminosos y acceder a la base de datos del atleta. En la opción para generar una rutina de entrenamiento permite cambiar parámetros como es la distancia a la cual detecta el sensor, la sensibilidad del contacto, la cantidad de intentos, tiempo de espera y tiempo de vencimiento para la activación del estímulo.



Figura 1: Tablet como controlador y menú de aplicación de Fitlight

El sistema FITLIGHT JR presenta una interfaz mucho más simplificada como se observa en la Figura 2, al igual que su versión profesional es una aplicación preinstalada en una tablet que funciona como controlador y al adquirir el sistema viene incluida.

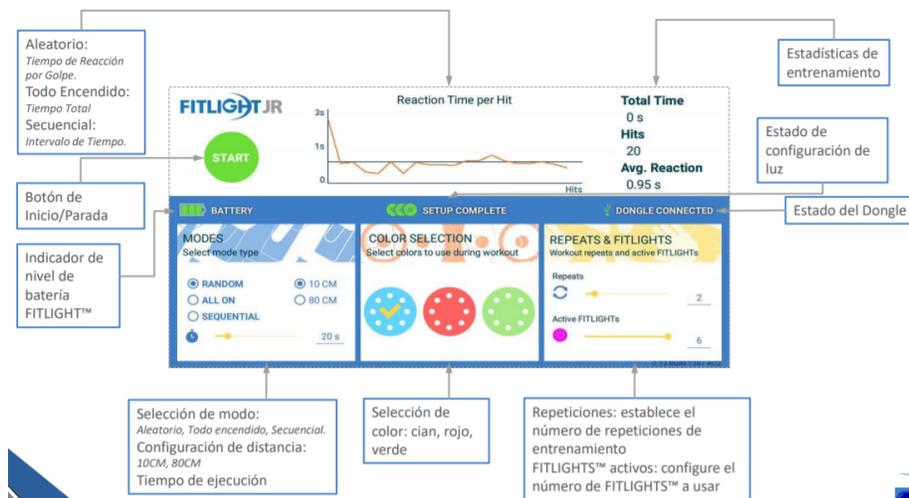


Figura 2: Interfaz de aplicación Fitlight JR

En la actualidad el desarrollo del rendimiento deportivo de las potencias mundiales y atletas olímpicos se apoya en dispositivos tecnológicos para la medición, mejora, entrenamiento, análisis de este. Cada vez vemos más dispositivos electrónicos que permiten monitorear aspectos de la salud y condición física.

En la última década el deporte guatemalteco ha incrementado su cosecha de medallas en los juegos del ciclo olímpico y competencias internacionales. Es un reflejo del apoyo e inversión del Comité Olímpico Guatemalteco (COG) y de la Confederación Deportiva Autónoma de Guatemala (CDAG) en infraestructura, campamentos de entrenamientos y herramientas para que los atletas de diversas disciplinas deportivas puedan desarrollar su máximo nivel. Los sistemas tecnológicos actuales con enfoque deportivo son diversos, pero con un costo que limita la adquisición de ellos a asociaciones departamentales, condicionando al deporte guatemalteco al uso de sistemas convencionales de entrenamiento.

La creación de herramientas y sistemas tecnológicos aplicables al deporte es parte de los compromisos de la Universidad del Valle de Guatemala y del departamento de Ingeniería Mecatrónica para contribuir con el desarrollo deportivo nacional. Es por ello por lo que se desarrollará un sistema de medición de tiempo de reacción para el deporte, dividido en módulos de hardware y software.

La aplicación de las competencias adquiridas y destrezas desarrolladas durante el estudio de la Licenciatura en Ingeniería Mecatrónica son un respaldo y motivación para el desarrollo e implementación del módulo de software del sistema de medición de tiempo de reacción, con ello poner a disposición del deporte guatemalteco una herramienta para el desarrollo y mejora del mismo.

4.1. Objetivo general

Diseñar aplicación web que permita la interacción entre el usuario final y los dispositivos de captación para la medición de tiempo de reacción.

4.2. Objetivos específicos

1. Diseñar la interfaz gráfica que contenga formularios para el registro de usuarios, inicio de sesión y configuración de entrenamientos.
2. Desarrollar código que permita la creación de entrenamientos para la medición de tiempo de reacción, predeterminados y configurables; aplicables en el deporte.
3. Verificar el funcionamiento de los ejercicios de medición de tiempo de reacción desarrollados en la aplicación web mediante pruebas de usuarios interactuando con los dispositivos de captación.
4. Realizar un análisis estadístico simple de los resultados obtenidos de tiempo de reacción de pruebas realizadas con los usuarios y desplegarlos a la aplicación web.
5. Establecer conexión inalámbrica entre un dispositivo inteligente y la Raspberry Pi Zero W donde se ejecuta la aplicación web.
6. Desarrollar un manual de usuario para el uso y configuración de la aplicación web.

La aplicación web debe permitir al usuario realizar un entrenamiento de tiempo de reacción. Un entrenamiento consiste en la activación de estímulos visuales y el usuario debe reaccionar ante el estímulo realizando la acción de presionar el dispositivo de captación.

Los dispositivos de captación de tiempo de reacción son implementados mediante un circuito que consta de un botón que fungirá como dispositivo de entrada o detección y un led unicolor que será la representación del estímulo visual.

La interfaz gráfica de la aplicación debe contener elementos sencillos y nombres adecuados para la navegación en la aplicación. El usuario debe poder parametrizar su entrenamiento a través de la aplicación. La aplicación debe tener un registro del usuario que realiza el entrenamiento y los parámetros del mismo.

Los resultados del entrenamiento se deben mostrar en tiempo real, esto implica que cuando el usuario reaccione ante el estímulo y el programa capture el tiempo de reacción este debe ser mostrado inmediatamente en la interfaz de la aplicación. Debe realizar un análisis estadístico simple de los resultados del entrenamiento, el valor mínimo, máximo y promedio.

La aplicación debe ser accedida a través de un dispositivo móvil, ya sea un smartphone, tablet o un computador, cuando este se conecte una red inalámbrica local designada para ello y acceda al navegador web e introduzca la dirección de la aplicación.

El tiempo de reacción a entrenar comprende únicamente del tipo sencillo, se deben dejar las bases para la implementación de un entrenamiento del tipo complejo bajo las condiciones del dispositivo de captación y estímulo utilizado.

6.1. Aplicaciones móviles

Es una aplicación con la finalidad de ser ejecutada en smartphones, tabletas y otros dispositivos móviles. Las aplicaciones permiten al usuario efectuar tareas específicas, aplicaciones dedicadas al entretenimiento y ocio, como aplicaciones con la finalidad de consultar su cuenta bancaria o calendarizar tareas. Se dividen en aplicaciones nativas y aplicaciones web.

6.1.1. Aplicación web

Una aplicación web es cualquier aplicación que es accedida vía web por una red como internet o una intranet. [4]

En esta categoría también se designan a los programas que se ejecutan en un navegador web y que utilizan el mismo para renderizar y ejecutar la aplicación.

Una de las ventajas de las aplicaciones web cargadas desde internet (u otra red) es la facilidad de mantener y actualizar dichas aplicaciones sin la necesidad de distribuir e instalar un software en el dispositivo. Con un uso potencial para miles de clientes. También la posibilidad de ser ejecutadas en múltiples plataformas.

La aplicación web se encuentra en otro equipo distante al nuestro que hace las veces de un servidor en la red. Este también puede ser un servidor local. De esta manera las aplicaciones web se encuentran instaladas en un servidor, teniendo a muchos clientes que se conectan a este servidor para utilizar las funciones de aplicación.

6.1.2. Aplicación web vs Aplicación nativa

Como se menciona, una aplicación web está optimizada para su visualización en dispositivos móviles, se accede a través de una URL, no está instalada en el dispositivo móvil y está se adapta al tamaño de la pantalla.

Una aplicación nativa es una aplicación desarrollada para un dispositivo celular particular, tiene acceso al hardware y es instalada directamente en este. Se descargan desde centros de aplicaciones como Apple App Store y Google Play Store.

En sentido de conocer que aplicación se adapta mejor a nuestras necesidades las comparamos en los aspectos de desarrollo y accesibilidad.

Proceso de desarrollo

Cada dispositivo móvil utiliza un lenguaje de programación distinto. IOS utiliza Objective-C, Android utiliza Java y los móviles de Windows C++. En cambio, las aplicaciones web utilizan lenguajes como HTML, CSS, Javascript y un Framework seleccionado por el diseñador. [5]

Cada plataforma móvil ofrece su propio estandarizado, herramientas de desarrollo y otros elementos que pueden ser utilizado para el proceso de desarrollo de la aplicación nativa. Para el desarrollo de aplicaciones web no hay una estandarización como tal, pero existen diversas herramientas y frameworks a disposición del desarrollador.

La visualización desde el punto de vista del usuario es muy similar entre ambos tipos de aplicación. La decisión se debe realizar basado en decidir si desarrollar una aplicación centrada en el usuario o una aplicación centrada en una función.

Accesibilidad

Una aplicación nativa es totalmente compatible con el hardware y sensores del dispositivo móvil, como lo puede ser el acelerómetro, GPS, cámara y demás herramientas. La aplicación web puede acceder a una cantidad limitada de herramientas del móvil. [6]

La aplicación nativa funciona por sí sola y sin conexión a internet, debe realizar actualizaciones para obtener el nuevo contenido o funciones. La aplicación web debe contar con acceso a internet u otra red para acceder a ella a través del navegador web del móvil. Las actualizaciones de la aplicación web se realizan en el servidor, el usuario no debe intervenir para acceder a ellas.

Las aplicaciones nativas son más costosas de desarrollar, son más veloces y eficientes porque trabaja en conjunto al móvil para el cual fue desarrollado. Tiene un grado de calidad garantizado debido a que solo se puede acceder a ella a través de la tienda en línea. El mantenimiento de una aplicación web puede resultar costoso según su escala de usuarios. Además, no existe ninguna regulación sobre su calidad.

6.2. Dispositivos móviles

Lo primero que se nos viene a la mente es un teléfono móvil, pero en la actualidad son varios dispositivos móviles: PC portátiles, tabletas, sistemas de navegación de automóviles, organizadores y asistentes personales, etc.

La diversidad de dispositivos compone una problemática para quien debe programarlos, cada uno cuenta con características específicas. Veremos las características generales para luego clasificarlos.

6.2.1. Características generales

Las características esenciales que presentan los dispositivos móviles:

- Tamaño pequeño
- Transportar en el bolsillo o una bolsa del usuario
- Capacidad de procesamiento.
- Conexión permanente o intermitente a una red.
- Poseen memoria.
- De uso personal.
- Capacidad de interacción a través de una pantalla o teclado.

Las anteriores características las podemos resumir en cuatro características que todo dispositivo móvil debe cumplir:

Movilidad: Describe la facilidad para ser transportado con frecuencia y facilidad de un lugar a otro.

Tamaño reducido: Característica física que permite el uso del dispositivo con una o dos manos sin la necesidad de un soporte externo. Facilita el transporte del dispositivo por parte del usuario.

Comunicación inalámbrica: Capacidad de enviar y recibir datos sin la necesidad de un enlace cableado.

Interacción con las personas: Proceso de uso que un usuario establece con un dispositivo.

6.2.2. Tipos de dispositivos móviles

El concepto de dispositivo móvil abarca un gran rango de dispositivos electrónicos. En esta sección únicamente se presentan los dispositivos relacionados al posible uso de la aplicación web a realizar.

Teléfono móvil

Se componen de micrófono, altavoz, pantalla de cristal líquido, teclado, antena, batería y su placa de circuitos. Posee un microprocesador encargado de las tareas del teclado y pantallas, control de las señales y gestión de comandos. Su principal función es la comunicación, a través de mensajes de texto o llamadas telefónicas.

Algunos modelos presentan capacidad para presentar información de vuelos, tráfico o noticias. La cantidad de datos para su funcionamiento es limitada por lo que esta información puede ser fácilmente mostrada. Las baterías con las que cuentan presentan una larga vida.

Smartphone

Los smartphones combinan las características de los teléfonos móviles con los de las computadoras. Permiten al usuario guardar información, instalar programas, navegar por la web además de utilizar un teléfono móvil, todo en un mismo dispositivo.

Un gran atractivo de estos dispositivos es la capacidad de expandir sus capacidades bajo la descarga de aplicaciones a través de tiendas de aplicaciones. Esto brinda un aspecto personalizable, cada usuario puede descargar las aplicaciones que le interesen.

Los smartphones se clasifican por las herramientas tecnológicas con las que cuentan, en tres categorías.

Smartphone de gama baja: Son los smartphones con menor costo en el mercado. Poseen limitada capacidad de almacenamiento, procesamiento y el tamaño de la pantalla. Los sistemas operativos que soportan son versiones no recientes, por lo que limita la cantidad y soporte de aplicaciones que pueden ser instaladas. Por su limitada capacidad de procesamiento, la duración de su batería es larga que es capaz de soportar días funcionando con una sola carga.

Smartphone de gama alta: Poseen características superiores a otros dispositivos disponibles en el mercado. Capacidad y velocidad de procesamiento, cámara de alta calidad, mayor memoria RAM y usualmente traen o son compatibles con la última versión de sistema operativo. La vida útil de la batería es limitada por los recursos que consume cada uno de los procesos de un smartphone de esta categoría, normalmente son acompañados por un cargador de batería de alta velocidad de carga. Los fabricantes de estos smartphones suelen implementar la tecnología más reciente en estos modelos y los presentan en ferias de tecnología. Poseen el costo más elevado en el mercado.

Existen otras clasificaciones intermedias para los smartphones, estas varían en precio usualmente por el tamaño de la pantalla, capacidad de la batería o especificaciones de la cámara. Poseen características muy similares a los de gama alta, pero sin llegar a igualarlos.

Tablet

Una tableta se caracteriza por tener como método de entrada una pantalla táctil. Se diferencia de los smartphones por tener dimensiones de pantalla mayores y ausencia para realizar llamadas telefónicas. Algunas de ellas permiten la introducción de un chip para realizar llamadas telefónicas.

Poseen cámara, micrófono, conexión WiFi, permiten la instalación de aplicaciones y navegación web.

6.3. Redes de comunicación inalámbrica

La comunicación inalámbrica es aquella entre dispositivos que utilizan el espectro electromagnético para intercambiar información. Por su mismo nombre se denota la ausencia de cables en este tipo de comunicación.

Esta comunicación es capaz de enviar y recibir información entre un emisor y un receptor, para ello utiliza un software que permite la transmisión y recepción, además que utiliza como hardware antenas o alambres conductores para transmitir esta información a través del espectro electromagnético.

6.3.1. Clasificación

Se clasificarán según su alcance y la forma de control de acceso a la red.

- Redes de área personal inalámbrica (WPAN: wireless personal area networks)
- Redes de área local inalámbrica (WLAN: wireless local area networks)
- Redes de área extendida inalámbrica (WWAN: wireless wide area networks)

Redes personales inalámbricas (WPAN)

La mayor limitación de estas redes es la distancia, aproximadamente soportan hasta unos 10 metros. Las tecnologías más utilizadas del WPAN son: Bluetooth, DECT, IrDA, NFC y Zigbee. Brevemente se describen cada uno de ellos.

Bluetooth Regulado por el grupo de trabajo IEEE 802.15.1, que permite la transmisión de voz y datos entre diferentes dispositivos mediante un enlace de radiofrecuencia en la banda ISM de 2,4 GHz. Utilizado en gran cantidad de dispositivos móviles. Define un alcance corto de 10 m y otro de hasta 100 m, sujeto a las paredes u obstáculos del ambiente de comunicación.

DECT Es un estándar que define un radioenlace entre dispositivos inalámbricos y estaciones fijas. Suele operar entre 1.880 a 1.900 MHz. Transfiere datos desde 552 Kbps hasta 2 Mbps.

IrDA Infrared Data Association es una asociación que integra más de sesenta compañías. Utiliza el espectro de frecuencia de infrarrojo para transmitir información. Se comunican a través del diodo LED, es necesario que el receptor y el emisor estén alineados uno con el otro con una desviación máxima de 30°. [7]

NFC Near field communication permite la transmisión de datos a una distancia muy cercana entre dispositivos a través de un enlace de radiofrecuencia en la banda ISM de 13.56 MHz. Suele ser utilizada para la identificación o validación de personas.

Zigbee Se trata de un conjunto estandarizado de soluciones para la comunicación inalámbrica que pueden ser implementadas por cualquier fabricante. Regulado por el grupo de trabajo IEEE 802.15.4 y promovido por ZigBee Alliance. [7]

Redes locales inalámbricas (WLAN)

Es una red de cobertura geográfica limitada, velocidad de transmisión alta, bajo nivel de errores y que se administra de forma privada. Presentan ventajas muy importantes como la movilidad, instalación simple, flexibilidad, bajo costo y escalabilidad.

La velocidad de transmisión puede alcanzar los 500 Mbps y el área de cobertura de una red WLAN puede ser de entre 10 y 100 m². Debido a que la información se transmite a través de ondas electromagnéticas que es un espacio abierto, se utilizan algoritmos de cifrado para proteger la información. [7]

Las tecnologías WLAN más utilizadas son variantes del IEEE 802.11.

IEEE 802.11

IEEE 802.11 es un estándar para redes inalámbricas definido por el Institute of Electrical and Electronics Engineers (IEEE). Continua en evolución, existen muchos grupos de investigación, trabajando en paralelo para mejorar el estándar, a partir de las especificaciones originales. [8]

El estándar IEEE 802.11 se divide en dos capas principales: la capa MAC (Media Access Control) y la capa física o PHY. Estas dos capas permiten hacer una separación funcional del estándar y, lo que es más importante, permite que un único protocolo de datos pueda usarse con distintos métodos de transmisión. [7]

El objetivo principal del estándar 802.11 es garantizar la funcionalidad de las aplicaciones. El estándar 802.11 es una familia de especificaciones, entre las cuales se destacan las siguientes:

- IEEE 802.11a: soporta velocidades de hasta 54 Mbps y utiliza la banda de frecuencias de 5 GHz. Este protocolo está orientado a la transmisión de paquetes.
- IEEE 802.11b: soporta velocidades de hasta 11 Mbps y utiliza la banda de frecuencias de 2.4 GHz. Este protocolo está orientado a la transmisión de paquetes, pero no soporta funciones de calidad de servicio.

- IEEE 802.11g: soporta velocidades de hasta 54 Mbps. Es una evolución del IEEE 802.11b y utiliza la misma banda de frecuencias de 2.4 GHz.
- IEEE 802.11n: soporta velocidades de hasta 600 Mbps y puede trabajar en dos bandas de frecuencia: 2.4 GHz y 5 GHz.
- IEEE 802.11ac: el más reciente protocolo y permite una transmisión de hasta 1.7 Gbps. Trabaja a una frecuencia de 5GHz.

Redes de gran alcance inalámbricas (WWAN)

Son conexiones de red que permiten el intercambio de información entre usuarios de zona geográficas distantes. Se pueden distinguir las fijas que utilizan radioenlace o satélite y las móviles que son las que utilizan los servicios públicos para la transmisión y recepción de señales.

Ejemplos de algunas de ellas son las redes satelitales de televisión, redes celulares como 3G y LTE.

6.4. Tecnologías para la web móvil

Los dispositivos móviles, específicamente los smartphones contienen entre sus características una amplia gama de elementos para crear, modificar y eliminar conexiones con otros dispositivos. Poseen una amplia flexibilidad y son personalizables a manera de que cada usuario puede añadir o eliminar aplicaciones para mejorar su experiencia de usuario. El mundo de los dispositivos móviles se encuentra en constante evolución, en esta sección se presentan tecnologías de comunicación inalámbrica y un resumen sobre navegadores web para móviles.

6.4.1. Comunicación inalámbrica

Un smartphone se define como un dispositivo celular con pantalla táctil que permite al usuario realizar conexiones, gestionar cuentas de correo, instalar aplicaciones y manejar recursos a pequeña escala de un computador. Los smartphones en general cuentan diversos protocolos de comunicación inalámbrica entre ellos se encuentran: Red Celular, WiFi, Bluetooth, NFC.

Red Celular

Podemos que existe 3G/4G/LTE para la transmisión de voz y datos. Puede llegar a una velocidad de descarga de 326 Mbps y de subida a 86.5 Mbps. El LTE permite conectividad a los dispositivos en movimiento hasta a una velocidad de 500 km/h. [9] La comunicación por red celular depende de la cobertura que a la vez depende de la compañía telefónica y la cantidad de celdas transmisoras tenga en una zona determinada.

WiFi

Es un protocolo de comunicación que utiliza ondas de radio para realizar una conexión de red. Una conexión WiFi se establece mediante un adaptador creando un hotspot. Los dispositivos cercanos al Router y conectados a la red pueden acceder a los servicios de internet. El termino hotspot se utiliza para definir un área donde el acceso a WiFi está disponible. Se pueden establecer conexiones a frecuencias de 2.4 GHz o 5 GHz basado en la compatibilidad, velocidad y cantidad de datos a transmitir. [10]

El dispositivo debe tener un adaptador inalámbrico que transforma la información en señales de radio, transmitida mediante una antena hacia un decodificador llamado Router. Una vez decodificada la información este la envía a Internet a través de una conexión cableada de Ethernet. La conexión inalámbrica es de dos vías así que para recibir datos estos pasar a través del Router y son decodificados por el adaptador inalámbrico del dispositivo.

El estándar de conexión 802.11 varía según las necesidades del usuario. Se dividen según la velocidad de transmisión de datos encontramos 802.11/a/b/g/n/ac desde 54 Mbit/s a 5GHz pasando por 11 Mbit/s a 2.4 GHz hasta el último estándar con una velocidad de 433 Mbit/s

Bluetooth

Protocolo de comunicación de bajo consumo, corto alcance de emisión que utiliza ondas de radios para transmitir y recibir información. El alcance puede llegar hasta unos 25 metros lineales sin obstáculos o elementos que interfieran con la señal. Para realizar la conexión mediante Bluetooth se debe realizar un proceso de emparejamiento.

Bluetooth cuenta con estándares de comunicación desde 1.0 hasta la versión más reciente 5.0, las velocidades van de 1 Mbit/s hasta los 50 Mbit/s en su versión más moderna.

NFC

Siglas del sistema de comunicación Near Field Communication. Comunicación limitada a una distancia entre 10 o 15 cm entre dispositivos. Su funcionamiento se basa en la creación de un campo electromagnético para el intercambio de información. Existen dos modos de funcionamiento: Activo y Pasivo.

En el modo activo los dispositivos para intercambiar información son capaces de generar su propio campo electromagnético debido a que cuentan con su propia fuente de energía. En el modo pasivo solo uno de los dispositivos posee fuente propia y al generar el campo electromagnético se intercambia información mediante la modulación de carga. En conclusión, el modo activo permite la comunicación bidireccional y el modo pasivo únicamente en un sentido.

Esta tecnología está teniendo un auge a las modalidades de pago Apple Pay, Samsung Pay, Google Pay además de empresas como Sony estén utilizando esta tecnología para realizar emparejamiento entre dispositivos y guardar configuraciones en pequeñas etiquetas con

este sistema. Además de las anteriores tecnologías con las que cuentan la mayoría de Smartphones de Gama Media-Alta. La mayoría de teléfonos poseen una tecnología de comunicación en común: USB. Esta utiliza cables, conectores y protocolos para comunicar entre dispositivos electrónicos. Existen diversos protocolos, el más reciente el USB 3.0 permite hasta 600 Mbps. [11]

6.4.2. Exploradores web para móviles

Los navegadores web para móviles están diseñados para su uso en teléfonos y smartphones. Están optimizados para mostrar el contenido en las pantallas de estos dispositivos.

La mayoría de estos navegadores se pueden instalar desde una tienda de aplicaciones. Se realizará un comparativa entre cuatro navegadores web, Chrome, Firefox, Opera Browser y UC Browser. Existen muchos otros, pero únicamente se describen los mencionados.

Chrome Es un navegador gratuito. Posee función de navegación segura y se puede buscar directamente desde la barra de ingreso de la página web. Permite la descarga de contenido y acceder a él sin conexión a internet. Permite la sincronización entre dispositivos y está disponible para todas las versiones de Android y también disponible para IOS.

Mozilla Firefox Disponible desde la versión 4.3 de Android y es una aplicación gratuita. Cuenta con accesos directos a buscadores de páginas y se pueden añadir cientos de complementos para expandir las capacidades del navegador.

Opera Browser Gratuito, posee un bloqueador de anuncios. Permite seleccionar el motor de búsqueda predeterminado. Permite descargar archivos en segundo plano y habilita descargas de mayor tamaño únicamente con conexión WiFi.

UC Browser Navegador gratuito con enfoque en la reproducción multimedia de videos y música. Poco consumo de recursos y cuenta con bloqueador de anuncios. [12]

En un estudio llevado por el blog *uptodown* se comparó el rendimiento de los cuatro navegadores con tecnologías como JetStream y Kraken para evaluar el funcionamiento de JavaScript y HTML5Test. El resultado de este estudio destaca Opera y Chrome como los navegadores con mayor compatibilidad. [13]

6.5. Tecnologías con capacidad de conexión inalámbrica

La mayoría de los dispositivos electrónicos y circuitos electrónicos están evolucionando a un mundo sin cables y con conectividad a Internet. Un mundo donde todo esté conectado a Internet, por ello el termino IoT (Internet of Things). En la actualidad existen muchos microcontroladores, microprocesadores o combinación de ambos que poseen la funcionalidad para comunicarse con otros dispositivos a través de Internet. En esta sección se presentan dispositivos con esta capacidad de conectividad.

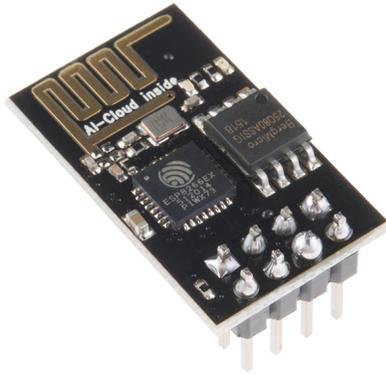


Figura 3: ESP8266EX

6.5.1. ESP8266EX

Es un módulo WiFi SoC (System on Chip) altamente utilizado por su bajo consumo de energía, diseño compacto y su rendimiento en las aplicaciones de Internet of Things. Integra un procesador RISC Tensilica L 106 32 – bit. Algunas de sus especificaciones:

WiFi: Protocolos: 802.11 b/g/n, Rango de frecuencia: 2.4 – 2.5 G.

Hardware: CPU: Tensilica L 306 32 – bit, Puertos periféricos: UART/SDIO/SPI/I2C/IR Remote Control/GPIO/ADC/PWM/Led Light-Button, Voltaje de operación: 2.5 – 3.6 V

Software: Modo WiFi: Station/SoftAP/SoftAP + Station, Seguridad: WPA/WPA2, Encriptación: WEP/TKIP/AES, Protocolos de red: IPv4, TCP/UDP/HTTP/FTP

Este chip suele utilizarse para aplicaciones de domótica, cámaras IP, red de sensores, encendido de luces, electrónicos incorporados en prendas de vestir, control industrial WiFi, localización de dispositivos WiFi, monitores de bebe, posición de señalizaciones. En general en muchas aplicaciones de Internet of Things. [14]

6.5.2. Raspberry Pi Zero W

Raspberry Pi es una pequeña computadora programable. De tamaño reducido pero de gran potencial, desde realizar hojas de cálculo, procesadores de texto, juegos, incluso reproducción de vídeos en alta definición. Este modelo es el más pequeño de Raspberry Pi hasta la fecha, combina las características de las versiones anteriores en un modelo más compacto. Trae consigo un compartimiento para micro SD, un puerto mini HDMI, dos puertos micro USB, 512MB de RAM y un procesador de 1 GHz. [15] Algo que hace este pequeño computador tan potente son sus características de conectividad: 802.11 b/g/n Wireless LAN, Bluetooth 4.1 y Bluetooth Low Energy.

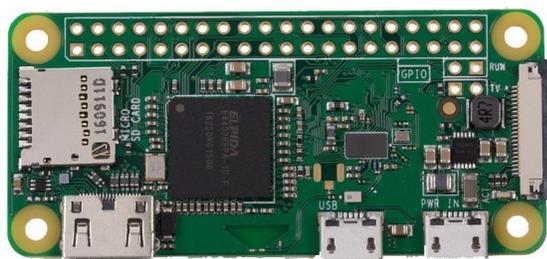


Figura 4: Modelo Raspberry Pi Zero W



Figura 5: LinkIt Smart 7688 DUO

6.5.3. LinkIt Smart 7688 DUO

Combina un microprocesador con un microcontrolador, potenciado por Media Tek MT7688 y ATmega 32U4. Es una placa de desarrollo basado en Arduino Yun, especializada para aplicaciones de IoT para hogar u oficina. Por su compatibilidad con Arduino, permite desarrollar aplicaciones basada en programas de Arduino. Además, por su memoria y su capacidad de almacenamiento permite procesamiento de vídeo. También permite la creación de aplicaciones en lenguaje Python, Node.js y C.

Es una placa desarrollada en conjunto por Seeed Studio y MediaTek. Por lo que combina el conocimiento de hardware y dispositivos IoT para la creación de esta placa. Algunas de sus especificaciones son WiFi 802.11 b/g/n (2.4G), puertos de I/O para GPIO, I2C, SPI, SPIS, UART, PWM y puerto Ethernet. Requiere una alimentación de 5V. [16]

6.6. Lenguajes de programación web

El desarrollo web combina funciones y características de la programación de un sitio web que incluyen el diseño, el registro, la gestión de contenidos, base de datos y la interacción.

6.6.1. Lenguajes del cliente

Lenguajes utilizados para construir cualquier sitio de internet desde el lado del cliente. Incluye todo el código para dar forma, estilo y análisis a una página web. Se ejecuta por un explorador web y puede ser observado, incluso modificado; por el cliente.

HTML

Es un lenguaje estándar de estructura para los contenidos de un sitio web. HTML que significa Lenguaje de Marcado para Hipertextos (HyperText Markup Language) es el elemento más básico de una página web.

Su finalidad es la representación de estructuras y contenidos dentro de la página web, pero no de su funcionalidad o apariencia. “HyperText” se refiere a los enlaces que conectan las páginas web unas con otras. Los enlaces son fundamentales en la Web. HTML utiliza marcadores “Markup” para anotar texto, imágenes y demás contenido a desplegar en el navegador. Incluye marcadores como los siguientes `<head>`, `<title>`, `<body>`, `<header>`, `<footer>`, `<article>`, `<section>`, `<p>`, `<div>`, ``, ``, `<aside>`, `<audio>`, `<canvas>`, `<datatable>`, `<details>`, `<embed>`, `<nav>`, `<output>`, `<progress>`, `<video>` y muchos otros. [17]

Los marcadores en HTML son sensibles a la diferencia entre minúsculas y mayúsculas, solo deben ser escritos en letras minúsculas.

CSS

Son hoja de Estilo en Cascada de su nombre en inglés Cascading Style Sheets, se utiliza para la presentación y apariencia de documentos HTML. Desarrollado en niveles CSS1 es obsoleto, CSS2.1 es un recomendable y CSS3 es última versión, incluye sombras, redondeados, transiciones o animaciones. [18]

Es un lenguaje de hojas de estilo que permite aplicar estilos de manera selectiva a diversos elementos de documentos HTML. Para la escritura de una regla de estilo se debe cumplir con los elementos de la “regla predeterminada” que se observa en la Figura 6.

El selector es el elemento HTML el cual se estilizará. La declaración especifica cual de todas las propiedades del elemento será estilizada. Las propiedades son a las cuales se le puede dar estilo y el valor de la propiedad es la elección, hay posibles apariencias para una propiedad determinada.



Figura 6: Regla de estilo CSS

La sintaxis de CSS, cada regla debe estar contenido dentro de corchetes `{}`. Entre cada declaración debe haber `:` entre la propiedad y su valor. Las declaraciones se separan con `;`.

Tipos de selectores

Se pueden realizar selecciones específicas con distintos de selectores, en el cuadro (1) se observan los más comunes. Existen otros selectores, combinación de ellos. Los listados en la tabla son los más comunes. [19]

JavaScript

Es un lenguaje de programación que permite realizar operaciones complejas en una página web. Permite animaciones, juegos, aplicaciones, interactividad y otras dinámicas. El código se ejecuta directamente en el explorador por lo que puede funcionar con o sin conexión a Internet. [20]

¿Qué podemos hacer?

- Almacenar valores dentro de variables.
- Operaciones con texto, números, arreglos de datos.
- Crear eventos y adjudicarlos a elementos de la página web.

La mayor funcionalidad construida sobre las bases del lenguaje de JavaScript se llama Interfaz de programación en aplicaciones. Application Programming Interfaces (API's). [20]

Las APIs son bloques de código que permiten a un desarrollador implementarlo a un programa. Son herramientas que permiten que el trabajo de diseño y funcionamiento sea mucho más sencillo, de cualquier otra forma podría ser muy difícil poder terminar el programa. Generalmente se dividen en dos categorías:

Nombre del selector	Selección	Ejemplo
Selector de elemento o tipo	Todos los elementos HTML con el tipo especificado	p
Selector de identificación (ID)	Elemento de la página con al ID especificado. (Solo se permite un único elemento por ID)	#my-id
Selector de clase	Los elementos en la página con la clase especificada (La clase puede aparecer varias veces en una página)	.my-class
Selector de atributo	Los elementos en una página con el atributo especificado	img[src]
Selector de pseudo-clase	Los elementos especificados, pero solo cuando se presente el estado específico	a:hover

Cuadro 1: Selectores CSS

Los **navegadores APIs (Browser APIs)** son construidos dentro del buscador web y poseen la capacidad de exponer información del navegador y otras operaciones complejas. Algunas de ellas son:

- DOM (Modelo de Objeto de Documento) API, permite manipular, crear, remover y cambiar códigos escritos en HTML y CSS. Agrega dinamismo a la página web. La Localización-Geo API establece la información geográfica.
- Las Canvas y el WebGL API permiten crear animaciones y gráficos 2D y 3D.

APIs de terceras personas, son código generados que obtienen su información de alguna parte del web. Algunas de ellas son: El Twitter API que permite mostrar los últimos tweets de un usuario en específico en un sitio web. El Google Maps API que permite incrustar mapas personalizados en un sitio web. [20]

¿Qué hace JavaScript en la página web?

El lenguaje de JavaScript es ejecutado por motor del navegador luego que el HTML y CSS han sido ejecutados y mostrados en la página Web. Este orden asegura que el estilo y la estructura de la página estén en la página web al momento en que JavaScript comienza a ejecutarse.

JavaScript funciona con un orden, generalmente de arriba hacia abajo. Por lo que se debe ser cuidadoso en que orden se escribe el código. Esto se debe a que JavaScript es un lenguaje de interpretación.

Existen dos formas de agregar JavaScript a una página Web. Interno en el HTML para lo cual se utiliza los selectores `<script>` `</script>` dentro de ellos se agrega el código. Externo para lo cual se debe crear un archivo con terminación `.js` y en HTML se debe agregar la siguiente expresión `<script src="archivo.js"></script>`. En la sección de `src` se agrega la ubicación del archivo con el código de JavaScript. [20]

6.6.2. Lenguajes del servidor

Los sitios deben ser guardados en un servidor web. Este lenguaje resume la transferencia de datos desde el servidor web hacia el explorador web. Cualquier código utilizado dentro del servidor para la creación o manejo de datos. El lenguaje de servidor se ejecuta dentro del mismo servidor por lo que aliviana la carga de recursos para el explorador web. Es más seguro debido que este código permanece en el servidor web.

PHP

Es un lenguaje de código abierto para desarrollo web que puede ser incrustado en el HTML. Su acrónimo viene de Hypertext Preprocessor . Utiliza etiquetas `<?php>` `<?>` dentro de un documento HTML que permiten entrar y salir del modo PHP. [21]

Con PHP se puede configurar que el servidor web procese todos los ficheros de HTML, el usuario no tiene forma de saber que fichero es el que se está ejecutando. Se puede emplear en la mayoría de los sistemas operativos, Linux, muchas de sus variantes; Windows, Mac OS entre otros muchos. Para su funcionalidad se necesita de un servidor web, Apache por ejemplo y colocar todos los archivos `.php` en el directorio y el servidor automáticamente los analizará. [22]

El código PHP es invisible al navegador web y al cliente, ya que es el servidor web quien se encarga de ejecutar el código y enviar su resultado al HTML. Posee la capacidad de conexión a la mayoría de las bases de datos, destaca su conexión con MySQL y PostgreSQL. Es una mezcla de programación por procedimientos y programación orientada a objetos. [23]

La forma de ejecución de PHP ante la petición de un documento se ve refleja en el siguiente ejemplo [24] :

1. Envía la tarea de solicitud de documento al sistema.
2. Espera que el sistema abra y lea el documento
3. Regresa el contenido al cliente
4. Listo para la siguiente petición.

Node.js

Node.js[®] es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8. Es una máquina virtual que ejecuta el código de JavaScript.

Node.js orientado a eventos asíncronos, diseñado para la construcción de aplicaciones en red escalables. [25]

Node.js es un lenguaje de código abierto que puede ser ejecutado en Mac OS X, Windows y Linux.

La forma en que Node.js maneja una petición se puede apreciar en el siguiente ejemplo: Manejo de Node.js a la petición de un documento.

1. Envía la tarea de solicitud de documento al sistema.
2. Listo para la siguiente petición.
3. Cuando el documento ha sido abierto y leído, el servidor regresa el contenido al cliente.

Node.js elimina la espera y siempre continua con la siguiente petición. A través de Node.js se puede generar contenido dinámico en una página, crear, abrir leer, escribir, borrar y cerrar archivos en el servidor, recolectar información, agregar, borrar y modificar información de una base de datos. [24]

Algunas de las ventajas de Node.js para el desarrollo de servidor web son:

- El código está escrito en “simple JavaScript”.
- Posee un gesto de paquetes NPM (Node Packet Manager) que proporciona acceso a miles de paquetes reutilizables. [26]
- Posee una comunidad de desarrolladores muy amplia, activa y en crecimiento.

Python

Lenguaje de programación desarrollado como proyecto de código abierto que es administrado por Python software Foundation. [27] Python es un lenguaje de scripts independiente de plataforma y orientado a objetos con la capacidad de realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web.

Es un lenguaje interpretado por lo que ofrece ventajas en la rapidez del desarrollo, ya que se ejecuta línea por línea. Es muy popular debido a que cuenta con una amplia gama de librerías, sencillez y velocidad de creación de los programas, cantidad de plataformas en las que se puede desarrollar Unix, Windows, OS y otros. [28]

Es un lenguaje de propósito general, entre sus posibilidades se encuentra el desarrollo web. Es interactivo ya que dispone de un intérprete de línea de comando en el cual se pueden introducir líneas de código y está produce un resultado visible. Es orientado a objetos por

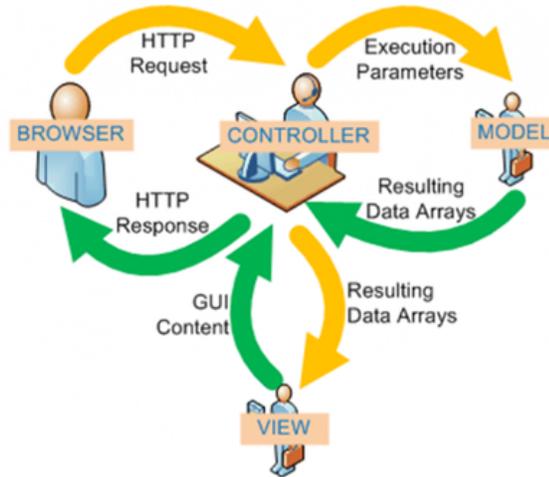


Figura 7: Flujo de información Modelo-Vista-Controlador [30]

lo que fácilmente se pueden crear programas con componentes reutilizables. Su sintaxis es muy clara, utiliza una notación indentada.

6.6.3. Framework MVC

El Modelo-Vista-Controlador es una arquitectura que separa una aplicación en tres componentes principales. Es uno de los estándares más utilizados para la estructura de un proyecto de desarrollo web para la creación de proyectos escalables y extensibles.

Modelo (Model)

Es el componente que corresponde a toda la información y su lógica con la que el usuario trabaja. Representa la información que es transferida a la Vista y al Controlador.

Vista (View)

Son todos los componentes para la interfaz gráfica de la aplicación. Es con la cual el usuario final interactúa.

Controlador (Controller)

Es el que actúa entre el modelo y la vista, es quien procesa toda la lógica y las peticiones, manipula la información utilizando el modelo y con la vista muestra el resultado final. [29]

Flujo de información

En la Figura 7 se observa la interacción del usuario con la arquitectura MVC.

El ciclo inicia con la interacción del usuario con el navegador web (browser), a través de un componente de la vista el usuario genera una acción que crea una petición. El controlador recibe la petición realizada por el usuario y este accede al modelo, actualizándolo, modificándolo u obteniendo información de este. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz adecuada, reflejando así los cambios en el modelo. La interfaz del usuario siempre está a la espera de nuevas peticiones del usuario para comenzar un nuevo ciclo de interacción.

Django

Es un entorno de aplicación ‘open source’ escrito en Python. Diseñado para un desarrollo veloz, muy completo ya que incluye decenas de aplicaciones extras que utilizar en tareas comunes en la Web. Con una librería de paquetes muy completa expandir las capacidades del framework [31].

Django puede ser ejecutado desde cualquier sistema operativo que contenga Python. Además, Python incluye una base de datos ligera, SQLite. Inicialmente no es necesario agregar una nueva.

Una aplicación web de Django agrupa el código con el que se gestiona cada uno de los siguientes ficheros:

URLs: procesan las peticiones de cada URL. Se utilizan para redirigir las peticiones HTTP a la vista apropiada basado en la URL de la petición.

Vista (View): es una función de gestión de peticiones que recibe peticiones HTTP y devuelve respuestas HTTP. Acceden a los datos que necesitan a través de los modelos y delegan el formato de la respuesta a las plantillas.

Modelos (Models): son objetos de Python que definen la estructura de los datos de una aplicación y proporcionan mecanismos para gestionar y consultar registros en la base de datos.

Plantillas (Templates): es un fichero de texto que define la estructura, con marcadores de posición que utilizan para representar el contenido real.

Este tipo de organización se define como Modelo-Vista-Plantilla como se observa en la Figura 8. Posee muchas similitudes con la arquitectura MVC.

Además, Django posee algunas otras características que incluyen lo siguiente:

- **Formularios:** Son formularios HTML que se utilizan para recolectar datos de los usuarios para su procesamiento por en el servidor.
- **Autenticación y permisos de los usuarios:** Django incluye un sistema robusto de

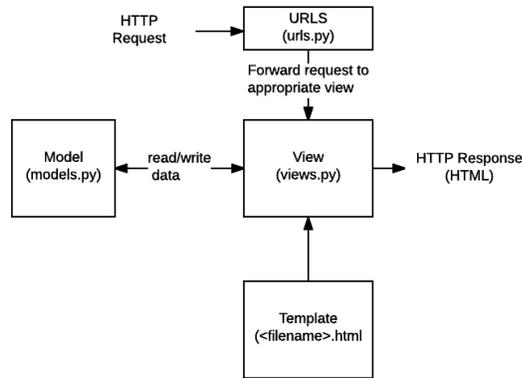


Figura 8: Organización Modelo-Vista-Plantilla de Django [32]

autenticación y permisos.

- **Sitio de Administración:** Includido por defecto, permite a los administradores crear, editar y visualizar cualquiera de los modelos de datos del sitio. [32]

6.7. Tiempo de reacción

6.7.1. Conceptos básicos

El tiempo de reacción es la cantidad de tiempo que transcurre desde que percibimos un estímulo hasta que damos una respuesta en consecuencia. Es la capacidad de detectar, procesar y dar respuesta al estímulo. [33] El tiempo de reacción depende de los siguientes factores:

- Percepción: Ver, oír o sentir el estímulo.
- Procesamiento: Concentrarse y entender la información.
- Respuesta: Agilidad motora para actuar ante el estímulo.

Si alguno de los procesos anteriores se ve afectado, el tiempo de acción se verá afectado. El tiempo de reacción está relacionado con los reflejos. El tiempo de reacción puede variar dependiendo la variación de cada uno de los factores:

- Complejidad del estímulo: Mientras más complejo sea, mayor cantidad de información habrá que procesar y por tanto se empleará mayor tiempo en ello.
- Familiaridad, preparación y expectativa: Mientras menos información nueva haya que procesar, el tiempo de procesamiento será menor.
- Estado: Las condiciones físicas pueden afectar el tiempo de reacción, la fatiga, atención, temperatura, edad o el consumo de alguna sustancia.

Los estímulos auditivos requieren un menor tiempo de procesamiento que los visuales. Cada estímulo conlleva un tiempo de respuesta distinto. El tipo de estímulo también se puede clasificar en:

1. Simple: única respuesta a un estímulo único.
2. De elección: Distintas respuestas a distintos estímulos.
3. De selección: Distintos estímulos, pero solo se debe reaccionar a uno.

El tiempo de reacción se aplica en diversas situaciones de la vida, como en una conversación, al conducir, realizar un deporte. Un tiempo de reacción adecuado nos dará beneficios, pero debemos procesar y responder de la forma correcta. Existen situaciones donde debemos reaccionar rápido y bien. Convenientemente el tiempo de reacción es algo que se puede entrenar y mejorar.

El tiempo de respuesta una habilidad cognitiva que es muy sensible a las alteraciones. Problemas de ceguera o sordera afectarán el tiempo de reacción. Enfermedades como el Alzheimer reducen la velocidad de procesamiento por lo que afectan el tiempo de respuesta.

6.7.2. Medición y evaluación del tiempo de reacción

Para la medición de tiempo de reacción existen diversos experimentos o pruebas que se pueden realizar a través de mecanismos e instrumentos de medición o únicamente a través de software.

Entre algunos de ellos, Cometti menciona un dispositivo que consta de dos aparatos, un aparato con una señal luminosa y un botón destinado al sujeto y otro al experimentador que activa la señal. Se le indica al sujeto que presione el botón y que levante cuando se encienda la luz, el cronómetro mide el tiempo que transcurre entre que se encendió la luz y el acto de levantar el botón. Para la medición de tiempo de reacción complejo menciona el dispositivo de Dal Monte, que es muy similar, pero implementa luces de distintos colores [34].

Existe otro experimento en el cual se utiliza una regla de 30 o 50 centímetros, se pide una persona que sostenga la regla desde la parte superior. El sujeto quien mide su tiempo de reacción debe situar sus dedos sobre el 0 de la regla sin tocarla. La persona que sujeta la regla la soltará sin previo aviso y el sujeto cerrará los dedos en que detecte que la regla comienza a caer. Se anotará la distancia que la regla haya caído. Cuanto mayor sea el tiempo de reacción mayor será la distancia, para encontrar el tiempo de reacción se utiliza la ecuación (1), donde g es la constante de la gravedad, d la distancia. [35]

$$t = \sqrt{\frac{2d}{g}} \quad (1)$$

Además, existen plataformas como *Cognifit* que ofrecen distintas herramientas para medir el tiempo de reacción. Estas pruebas además de medir el tiempo de reacción miden memoria de trabajo, escaneo visual, coordinación ojo-mano, inhibición, monitorización, denominación, percepción visual, memoria contextual, reconocimiento, atención focalizada y percepción espacial.

- Test de Indagación REST-COM: Aparecen objetos durante poco tiempo. Después se debe seleccionar la palabra que corresponda con las imágenes presentadas, lo más rápidamente posible.
- Test de Decodificación VIPER-NAM: Aparecen imágenes de objetos en la pantalla durante un periodo corto de tiempo y desaparece. Acto seguido aparecen cuatro letras, y sólo una corresponderá con la primera letra del nombre del objeto, siendo esa la letra objetivo. Hay que llevarlo a cabo tan rápido como sea posible.
- Test de Reconocimiento WOM-REST: Aparecen tres objetos comunes en la pantalla. Primero habrá que recordar el orden de presentación de los tres objetos tan rápido como sea posible. Posteriormente, aparecerán cuatro series de tres objetos diferentes a los presentados y habrá que detectar la secuencia inicial.
- Test de Resolución REST-SPER: Aparecen en la pantalla numerosos estímulos en movimiento. Habrá que pinchar en los estímulos objetivo tan rápido como sea posible, pero evitando pinchar en los estímulos intrusos.
- Test de Celeridad REST-HECOOR: Aparece en la pantalla un cuadrado azul. Habrá que pulsar tan rápido como sea posible el botón situándose dentro del cuadrado. Cuantas más veces se pulse el botón en el tiempo disponible, mejor resultado se obtendrá.
- Test de Procesado REST-INH: En esta tarea, irán apareciendo en la pantalla dos bloques con números y formas diferentes. Inicialmente habrá que atender al tamaño de la forma e indicar el más alto. Después, habrá que atender bloque que contenga la numeración más alta. [33]

Las anteriores pruebas se encuentran en el sitio de cognifit.com y están basadas en pruebas tradicionales como NEPSY, Variables of Attention (TOVA), test de ejecución continua (CPT), Test of Memory Malingering (TOMM) y en Visual Organisation Tak (VOT).

Existen otras pruebas en páginas web que miden el tiempo de reacción a través de experimentos sencillos, que únicamente involucran realizar un clic para capturar el tiempo de reacción. Los sitios encontrados en la web únicamente evalúan el tiempo de reacción simple. Algunos de ellos son RED LIGHT – GREEN LIGHT Reaction Time Test [36] y de la misma modalidad podemos encontrar varios sitios. [37] Podemos encontrar infinidad de pruebas para la medición de tiempo de reacción en muchos sitios de internet y aplicaciones móviles.

6.7.3. Mejora del tiempo de reacción

Las habilidades cognitivas pueden ser entrenadas y mejoradas mediante la práctica de ejercicios especializados. Como buen músculo, el cerebro se fortalece al trabajar frecuente-

mente ejercicios focalizados en una habilidad en específico. Al entrenar mejoraremos nuestras conexiones, serán más rápidas y eficientes, mejorando nuestra capacidad.

El entrenamiento aumenta la experiencia por lo que permite reducir el número de elementos a observar para poder predecir la respuesta. Al reducir el número de elementos a analizar, se reduce el tiempo de respuesta.

El arte de la selección

Antes de dar vida a la aplicación que debe medir tiempos de reacción y crear ejercicios que permitan el entrenamiento y medición de este, se debe conocer o seleccionar las herramientas con las cuales se contará y aquellas que debemos crear para el desarrollo del programa. En este capítulo se describe el procedimiento realizado para la selección de tecnologías de software y hardware para el desarrollo de la aplicación orientada a la medición de tiempo de reacción en el deporte.

Un principio sólido de este capítulo es la investigación previa realizada y plasmada en el marco teórico sobre tecnologías con características que permiten el cumplimiento de cada uno de los objetivos de la aplicación.

Con este capítulo debe poder comprender el arte de diseñar antes de construir, de desarrollar. Se explica cómo se llegó a escoger cada una de las tecnologías y a partir de ellas se creó la idea general de funcionamiento plasmada en la Figura 9 como resultado de todas las selecciones realizadas. Este diagrama será posteriormente explicado en con mayor detalle en otro capítulo, por ahora solo debe comprender la idea general de funcionamiento.

7.1. Selección del tipo de aplicación

El primer paso es la selección del tipo de aplicación a desarrollar y que esta pueda ser accedida a través de un dispositivo móvil, como un smartphone, una tablet o una computadora portátil. Para la selección del tipo de aplicación a desarrolló un cuadro comparativo entre aplicación nativas y aplicaciones web. En el Cuadro 2 se observan los distintos aspectos evaluados.

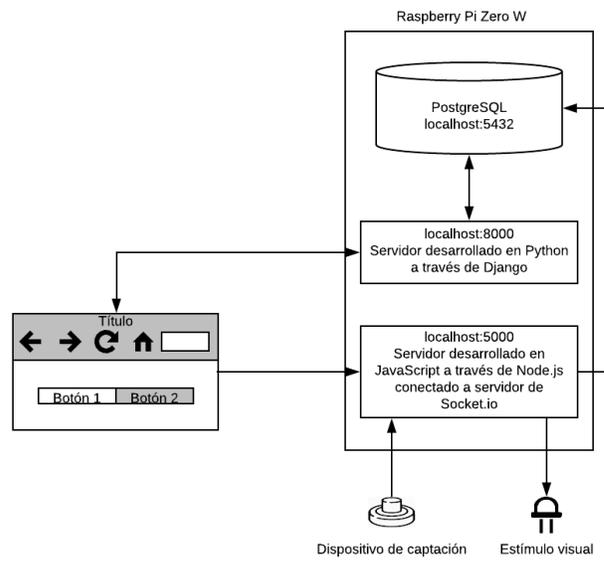


Figura 9: Planteamiento general de funcionamiento del software

Uno de los aspectos importantes tomados en cuenta para la decisión es la opción que la aplicación sea multiplataforma, esto significa que pueda ser ejecutada en diversos sistemas operativos sin alterar o tener que cambiar la aplicación. La aplicación web presenta mejores características en esta categoría, únicamente es necesario un navegador web instalado, que el dispositivo se encuentre conectado a la red local del servidor y escribir la dirección URL de la aplicación en el navegador web para poder acceder a ella.

En el criterio de necesidad de conexión a una red, la aplicación nativa posee mejores características debido a que no necesariamente necesita estar conectada a alguna red para ser ejecutada, por lo que se definió en el alcance, la aplicación a desarrollar cumplirá una función específica por lo que se espera que el usuario acceda a ella únicamente cuando la necesite y para ello necesitará conectarse a una red.

La aplicación nativa posee la desventaja de que cada sistema operativo cuenta con su propio lenguaje de programación para el desarrollo de aplicaciones. Algo que limita a que la aplicación se desarrolle para un usuario con un sistema operativo en específico.

La aplicación web será utilizada por el usuario únicamente cuando este lo desee y necesite, sin necesidad de descarga, por lo que la vida útil de aplicación será continua mientras el usuario se encuentre utilizándola. En cambio, si la aplicación nativa no cumple con las exigencias del usuario, esta será desinstalada. La actualización de una aplicación web sucede únicamente del lado del servidor por lo que para el usuario puede pasar por inadvertido.

Bajo los criterios mostrados en la comparativa de los tipos de aplicación y la discusión de estos se tomó la decisión de desarrollar una aplicación web por su característica que lo hace multiplataforma, falta de necesidad de descarga para su uso, facilidad de actualización o mejora y escalabilidad.

7.2. Selección de tecnología que albergará la aplicación

Debido a que se trabajó con una aplicación web, se creó una red WLAN sin conexión a Internet para acceder a la aplicación a través de esta. Como se muestra en la Figura 10 el acceso de la aplicación es a través de un punto de acceso inalámbrico (AP por su nombre en inglés *access point*) y este apunta hacia donde se ejecuta la aplicación, todos los dispositivos que se conecten al AP podrán ingresar a la aplicación, aunque por restricciones de diseño solo un cliente podrá utilizar el sistema de medición de tiempo de reacción a la vez.

La elección de la tecnología inalámbrica que albergará la aplicación web se realizó basado entre tres dispositivos con capacidad de conexión WiFi que puedan intercambiar entre modo cliente y punto de acceso. Este listado está conformado por el ESP8266EX, Raspberry Pi Zero W y Linkit Smart 7688 Duo. En el Cuadro 3 se observa la comparación realizada entre ellos.

		Tipo de Aplicación	
		Aplicación Nativa	Aplicación Web
Aspecto evaluador	Inmediatez	Descargar de una tienda para su instalación.	Instantánea, sin necesidad de descargas o programas especiales.
	Compatibilidad	Aplicaciones categorizadas por el sistema operativo del móvil	Cualquier sistema operativo, solo requiere de un navegador web.
	Facilidad de actualización	Requiere de descarga para obtener la última versión	La actualización se realiza únicamente del lado del servidor, visualización inmediata
	Facilidad de búsqueda	La visibilidad está restringida por las tiendas de aplicaciones	Aparecen como resultado de los motores búsqueda
	Ciclo de vida	Tiempo que permanezca instalada en el dispositivo	Siempre disponibles cuando el usuario desee acceder
	Alcance	Limitado por el sistema operativo. Puede utilizarse sin una conexión a internet.	Accesible desde cualquier dispositivo conectado a la red y que posea un navegador web. Se requiere de una conexión a internet
	Tiempo y costo	Mayor tiempo de desarrollo y por ende más costoso	Más sencillos y menos costosos
	Soporte y mantenimiento	Más costoso mantener una aplicación	Facilidad para realizar un mantenimiento

Cuadro 2: Comparativa entre aplicación nativa y aplicación web [5], [6]

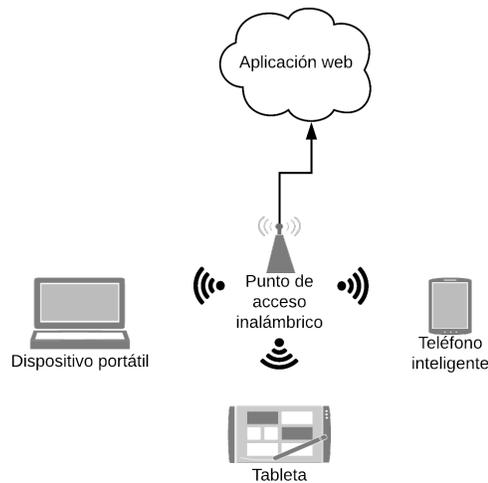


Figura 10: Red WLAN para acceso de aplicación web

El precio de cada uno de los tres dispositivos comparte un rango muy similar, todos estos dispositivos tienen un costo menor de 20 dólares americanos. Este criterio no fue decisivo en la elección final.

Todos presentan conectividad WiFi y soportan lenguajes de programación de gran alcance, con sistemas operativos especializados. Se destaca el ESP8266EX fue el único microcontrolador en entrar este listado por sus amplias aplicaciones en IoT, por su funcionalidad WiFi y capacidad de procesamiento similar a la de un Arduino.

Por la función que cumplirá el dispositivo, la capacidad de ejecutar un servidor web es crucial. Todos soportan la creación de un servidor web, pero con alcances distintos, por eso se clasificó en limitado, medio y complejo. El que en teoría posee mejor desempeño en este aspecto es la Raspberry Pi Zero W, dispone del potencial de una computadora en una placa que ocupa no más de la palma de la mano. El Linkit Smart 7688 Duo soporta Node.js por lo que lo hace una opción llamativa y el MCU ESP8266EX no soporta ningún lenguaje externo para el desarrollo del servidor web, aunque se hallaron algunos ejemplos de un servidor web sencillo a través de librerías para activar y desactivar una salida.

El criterio que permitió tomar la decisión final acerca de la selección del dispositivo inalámbrico es la capacidad de ejecutar un servidor de base de datos dentro del mismo. La Raspberry Pi Zero W es la única con esta capacidad, por ello y que en el resto de los aspectos evaluados presenta un margen de superioridad; es la elección para el dispositivo que albergará la aplicación.

La fuente de alimentación se evaluó entre los tres dispositivos en una escala de 0 a 10. La mayor puntuación hace referencia a un dispositivo eficiente energéticamente, quiere decir que puede cumplir con todas sus capacidades con el menor voltaje y corriente. El mejor calificado es el MCU que requiere de 3.3 V y 0.3 A. La Raspberry recibe el puntaje más bajo debido a su alta demanda de corriente, capaz de funcionar desde un amperaje de 1.2 A, su funcionamiento óptimo se recomienda 5V y 2 A.

La elección de la Raspberry Pi Zero W como tecnología que albergará la aplicación web nos abre un mundo de posibilidades para el desarrollo. La primer gran ventaja de utilizar esta tecnología es que permite desarrollar la plataforma en otro computador y no directamente trabajando con la Raspberry y luego dedicarse a la implementación de la aplicación en ella. La posibilidad de ampliar el software para mejorar el rendimiento y funcionamiento de la aplicación: bases de datos, instaladores y manejadores de paquetes, librerías para el uso de pines de la Raspberry, entornos de trabajo, entre otros. La mayoría de estos paquetes de software poseen una versión dedicada para trabajar con el sistema operativo Linux en el cual se basa el sistema operativo que sugiere Raspberry Pi para todos sus modelos.

7.2.1. Sistema operativo

En el sitio de Raspberry se encuentran dos opciones principales para sistema operativo de sus diversos de computadora. Se encuentra NOOBS y Raspbian, el primero es un sistema mucho más sencillo, simplificado y es recomendado para principiantes, el segundo sistema operativo oficial de Raspberry Pi, basado en Debian.

Raspbian fue el escogido para ser el sistema operativo, el cual se le instaló al modelo Zero W. Está optimizado para todos los dispositivos Raspberry Pi. Un sistema operativo es el conjunto de programas básicos y herramientas para poder ejecutar la Raspberry Pi. Aunque Raspbian va más allá de un simple sistema operativo, trae más de 35,000 paquetes, precompilados para su fácil uso e instalación.

Raspbian, creado por un equipo de desarrollados apasionados por la Raspberry Pi, su función educativa y el proyecto Debian. La fundación de Raspberry Pi produce y distribuye las imágenes de este sistema operativo, serán utilizadas por más de 100,000 usuarios de Raspberry y eso lo consideran el mejor soporte. La última versión que distribuyen al tiempo de redacción es Raspbian Stretch with Desktop de versión junio 2018. Se descargó la versión de marzo 2018, se recomienda actualizar cada vez que se añade un nuevo paquete de software.

7.2.2. Accesorios

Existen dos accesorios esenciales para el funcionamiento de la Raspberry Pi, uno de software que consiste en un dispositivo de almacenamiento con el sistema operativo y uno de hardware que funcione como fuente de alimentación para la placa del computador. Una memoria microSD es la tecnología adecuada para instalar el sistema operativo. En este trabajo se escogió una memoria de 16 GB, por facilidad y disponibilidad la misma. Un adaptador de corriente alterna también es necesario, debe otorgar un voltaje de alimentación de 5 V y una corriente mínima de 1 A, se trabajó con un convertidor de 5V y 2A, aunque también se realizaron algunas pruebas con un convertidor que solamente otorga 1A. La Raspberry Pi funcionó correctamente.

Para el proceso de desarrollo, se puede acceder a la Raspberry Pi a través de la interfaz gráfica del sistema operativo para realizar la configuración inicial de algunos procesos. Aunque existen otros métodos en los cuales esta configuración se puede realizar a través de la consola, modificación de archivos en directorio del sistema operativo y comunicación SSH.

Se eligió acceder a la interfaz gráfica y realizar la configuración inicial desde esta. Para ello requerimos de un teclado y monitor. Además, se utilizaron adaptadores de miniHDMI a HDMI, luego uno de HDMI a VGA ya que el monitor con el que se contaba solo tenía este tipo de conexión y un adaptador de microUSB a USB hembra para poder conectar un teclado.

La configuración inicial realizada consistió en la conexión a una red WiFi y la habilitación del protocolo de comunicación SSH para luego poder acceder a ella a través de este protocolo sin la necesidad de conectar nuevamente un teclado o monitor.

		Dispositivos		
		ESP8266EX	Raspberry Pi Zero W	LinkIt Smart 7688 Duo
Concepto	Chip	MCU	CPU	MCU/CPU
	Wi-Fi	Sí	Sí	Sí
	Sistema operativo	Mongoose OS	NOOBS/Raspbian	OpenWrt
	Lenguaje de programación	Lua/C/Arduino/JavaScript	C/C++/Python/Java/JavaScript	Arduino/Python
	Puertos	UART/SDIO/SPI/I2C/IR Remote Control/GPIO/ADC/PWM	USB/Mini HDMI/GPIO/I2C/SPI/UART	GPIO/I2C/SPI/SPIS/UART/PWM
	Servidor Web	Limitado	Complejo	Medio
	Soporta PHP o Node.js	Ninguno	Ambos	Node.js
	Soporta base de datos en el chip	No	Sí	No
	Fuente de alimentación	10	7	8

Cuadro 3: Comparación dispositivos inalámbricos

7.3. Selección del entorno de trabajo

Definida la tecnología que ejecutará el servidor de la aplicación web, Raspberry Pi W Zero; las opciones para la selección del entorno del trabajo para la aplicación son amplias. Para esta elección se compararon los ambientes de desarrollo basados en diversos lenguajes de programación web mostrados en el Cuadro 4.

A la escala de funcionalidad a la que trabaja la aplicación la velocidad de procesamiento no es un tema de complejidad o de diferencia significativa para cualquiera de los tres lenguajes de programación.

La cantidad de paquetes adicionales de software a disposición del desarrollador es mucho mayor en Node.js a comparación a los otros dos ambientes, por lo que su capacidad se ve expandida debido a estos *add-ons*. Todos presentan compatibilidad para el uso de bases de datos SQL, por lo que es un aspecto equitativo entre ellos.

Un factor determinante para la elección del tipo de ambiente de desarrollo es la capacidad de integrar aplicaciones con comunicación en *Real-Time*. Para establecer este tipo de comunicación se utiliza Socket.io que permite comunicación bidireccional en tiempo real, basado en eventos, trabaja en cualquier plataforma, navegador o dispositivo, enfocado en fiabilidad y velocidad.

Node.js permite una combinación sencilla y adecuada para las aplicaciones en tiempo real, la facilidad para implementar Socket.io en Node consiste únicamente en añadir un paquete dentro de la aplicación y agregar unas cuantas líneas de código como se observa su sitio [38] para ejecutar un servidor con comunicación en tiempo real. Por la versatilidad y capacidad de expansión, Node.js es el entorno de trabajo ideal para el desarrollo de la programación de los ejercicios de entrenamiento de tiempo de reacción por su facilidad para trabajar con Socket.io que a su vez permite establecer comunicación en tiempo real.

Para la visualización y desarrollo de los modelos, vistas y plantillas que es con lo que el usuario interactúa directamente se escogió trabajar con Python a través de su framework Django. Posee muchas características positivas para dedicarse únicamente en el desarrollo de la aplicación, aprovechando sus facilidades para crear autenticación y permiso de los usuarios. Django trae por defecto como base de datos mySQLite, para el desarrollo inicial no se realizó ningún cambio sobre esto, en la sección de selección de base de datos se explica la escogencia final y la razones de esta.

La selección del entorno de trabajo queda definida de la siguiente manera: Python, a través de Django para el manejo y desarrollo de los modelos, vistas y plantillas creadas. JavaScript, a través de su entorno de trabajo Node.js por su facilidad de trabajo con aplicaciones en tiempo real como lo es Socket.io. Se creará una conexión de enlace entre ambos ambientes de desarrollo a través del servidor de Socket.io para la comunicación entre ambos ambientes.

7.4. Selección de base de datos

Cronológicamente la selección de la base de datos se realizó cuando se desarrollaban los modelos de la aplicación. Inicialmente todo lo referente a la base de datos se trabajó con mySQLite que viene por defecto en Django.

Planteando los modelos de la aplicación, específicamente para el modelo de sesión de entrenamiento y de juego. Para guardar los resultados se observó la oportunidad de agregar un tipo de campo JSON, en el cual el objeto de resultados contendrá los intentos, tiempos y acierto del entrenamiento.

mySQLite no soporta el manejo de un tipo de campo JSON. No puede crear un esquema de base de datos que contenga este tipo de dato. En cambio, postgresQL posee soporte para crear esquemas con campos del tipo JSON. La elección de la base de datos fue consecuencia del planteamiento de un modelo de la aplicación.

		Ambientes de Desarrollo Web		
		PHP	Node.js	Django
Concepto	Modo de compilación	Utiliza compilación por interprete, lo que hace que sea menos optimizado que una ejecución <i>JIT</i>	Compila utilizando <i>JIT</i> con el V8 JavaScript Engine. Es eficiente aunque aumenta el tiempo de arranque.	Es un lenguaje interpretado no compilado, por lo que su ejecución es más lenta.
	Paquetes	155,000 en packgist	450,000 en npm	3,000 en django packages
	Proyectos Open Source	300,000	115,000	140
	Consultas por segundo	1,800	3,600	750
	Rapidez de procesamiento	Herramientas y extensiones para mejorar el rendimiento como, HHVM, Opcahcing, JIT Compilation	V8 JavaScript, motor de código abierto de alto rendimiento de Google. Se puede ejecutar en Windows, macOS y Linux.	Con un mayor de tiempo de ejecución. Ya que es un lenguaje interpretado
		<i>Bubble sort</i> , se degrada luego de 250 procesos en el servidor	<i>Bubble sort</i> , constante hasta con 1,000 elementos. Incluso con 10,000 es doble de veloz que HHVM.	Mayor tiempo ejecución en comparación a sus homologos.
	<i>Requests</i>	Crea múltiples hilos para responder a múltiples peticiones	Eficiente en el manejo de múltiples y concurrentes peticiones	Creación de múltiples hilos para atender a cada una de las peticiones
	Base de datos	Coexiste con MySQL y sus variantes.	Posee librerías para acceder a bases de datos SQL. Además comprende JSON que el lenguaje para comunicarse con bases de datos NoSQL.	Múltiples bases de datos compatibles, trabaja por defecto con MySQLite
<i>Real Time</i>	Cuenta con librerías para <i>Real-Time</i> como Ratchet, Wrench, Elephant.io, Socket.io	Librerías <i>Real-Time</i> como ws, Sockjs, Socket.io	No contiene librerías para <i>Real-Time</i> , debe correr el servidor Socket.io en paralelo	

Cuadro 4: Cuadro comparativo: Ambientes de desarrollo web [39], [40]

Este capítulo describe el proceso de diseño de la aplicación web, los conocimientos previos requeridos para la escritura del código, los editores de código utilizados, pruebas de funcionamiento y navegación de la aplicación.

Como todo buen proceso de diseño, hubo una investigación previa, se enlistaron los requerimientos generales de funcionamiento, se planteó un diseño, bajo pruebas iterativas se realizó la implementación y verificación. El resultado de la investigación previa reflejó conocimientos muy específicos necesarios para iniciar con el desarrollo de la aplicación entre ellos los lenguajes de programación y su sintaxis, arquitecturas de diseño web, entre otros.

8.1. Desarrollo

El desarrollo de la aplicación se dividió en dos frentes, todo lo referente a lo visual y con lo que el usuario interactúa *Front end*. Toda la lógica detrás de lo visual, para el funcionamiento y manejo de datos de la aplicación llamado *Back end*. La aplicación fue una combinación en el uso de lenguajes y herramientas de desarrollo web.

8.1.1. Front End

Se inició adquiriendo los conocimientos básicos acerca del contenido de una página web. Incluyendo lenguaje de estructura HTML, lenguaje de estilo CSS y para agregar dinamismo se utilizó el lenguaje JavaScript.

Se tomo el curso de “Intro to HTML and CSS” en la plataforma Udacity. [41] El curso proporciono el contenido adecuado acerca de la sintaxis HTML y CSS, además cuenta con

lecciones específicas para poner a prueba los conocimientos adquiridos por medio de problemas y preguntas sobre el contenido. A través del curso se adquirió el conocimiento básico para poder dar estructura y formato a una página web.

En la misma plataforma se realizó el curso “Intro to JavaScript”, el curso abarca los siguientes temas: definición, tipos de dato y variables, condicionales, ciclos, funciones, arreglos de datos y objetos. Finalizado el curso, brinda las herramientas básicas para implementarlo a una página web así mejorar la interacción y hacer del sitio dinámico. Complementando los temas abarcados en los cursos realizados, se utilizó la plataforma W3Schools [42] que cuenta con diversas guías sobre HTML, CSS y JavaScript. Se utilizaron herramientas como un selector de colores para HTML [43], generador de botones CSS [44] y generador de formato de texto. [45]

Se solucionaron bugs durante el diseño con consultas a través un motor de búsqueda que daba como resultado formas específicas de solucionarlo, en foros de desarrollo como lo son Stackoverflow [46], Quora [47] y algunos otros. No se llevó un registro completo de la cantidad errores y sus soluciones durante el proceso de desarrollo, es un proceso iterativo.

8.1.2. Back End

Para el desarrollo del *Back end* se utilizó Python. El conocimiento sobre este se encontraba a un nivel avanzado para el diseño de programas generales con arquitectura modular y enfocado en sus paquetes de análisis de datos y paquetes científicos. No se había trabajado Python para desarrollo web a través de sus framework Django.

El proceso de aprendizaje se enfocó en el trabajo y aplicación de Django. Para ello se realizó el tutorial que se encuentra en el sitio oficial de Django. [48] El tutorial consiste en la creación de una aplicación de encuestas, de forma muy detallada describe cada uno de los pasos para diseñarla.

Los conocimientos adquiridos de realizar el tutorial fueron la base para iniciar con el desarrollo de la aplicación web del proyecto. Al igual que sucedió con el Front End en el proceso existieron errores o dudas sobre cómo realizar cierta acción dentro del ambiente de desarrollo para la resolución de ellos se utilizó la documentación muy detallada que ofrece Django, consulta de foros y asesoramiento de expertos en el tema.

8.2. Software de desarrollo

Esta sección presenta los programas utilizados para facilitar el desarrollo web. No se detalla el proceso de la selección de estos, es un aspecto muy personal del desarrollador. Posiblemente otros desarrolladores hubiesen escogido otros programas. Los siguientes programas facilitaron el desarrollo de aplicación.

8.2.1. Anaconda

En su pagina web [49] se describe así mismo como un programa de código libre distribuido por Anaconda Distribution, es la forma más sencilla de hacer Python para análisis de datos y *machine learning* en plataformas con sistemas Linux, Windows y Mac OS. Es un estándar para la industria para el desarrollo y prueba en un computador.

Anaconda permite la creación de diversos ambientes de desarrollo, cada uno puede contener versiones distintas del mismo software. En la documentación [50] se encuentran detallados los pasos a seguir para crear un ambiente.

El programa se utilizó para crear un ambiente de desarrollo con la versión de Python 3.5.2. La documentación es muy completa y detalla algunas acciones importantes como activar un ambiente, determinar el ambiente que actualmente se encuentra activado y listar los ambientes creados.

8.2.2. Visual Studio Code

Es un editor de código fuente gratuito, de código libre desarrollado por Microsoft para Windows, Linux y Mac OS. Utiliza IntelliSense una herramienta para autocomplementar, permite completar líneas de código basadas en el tipo de variable, definición de funciones y al importar módulos. Es completamente expandible y personalizable a través de la instalación de extensiones de software para agregar nuevos lenguajes, temas, debuggers o para conectar con servicios adicionales como GitHub [51].

Visual Studio Code se utilizó para el desarrollo de código de la aplicación en Python, a su vez se fueron agregando los archivos estáticos de otros lenguajes como HTML, CSS y JavaScript. Se utilizó una extensión Django para autocomplementar líneas específicas de sintaxis y una extensión de GitHub para el control de versiones.

8.2.3. Atom

Es un editor de código fuente, multiplataforma, expandible, personalizable y con funciones de para autocomplementar [52]. Trae una integración para control de versiones con GitHub. Este editor se utilizó inicialmente para el desarrollo de las plantillas HTML, con su integración de CSS y JavaScript. Se utilizó para escribir el código de los ejercicios de entrenamiento.

8.2.4. GitHub

Es un programa basado en la web para el control de versiones. Permite llevar un registro acerca de los cambios en el código. [53] Poseen dos tipos de repositorio, uno gratuito que cualquier persona puede acceder a él y descargar el código desarrollado, otro privado por el cual se debe pagar para mantener el código alejado del alcance público.

Se utilizaron extensiones en los editores de código para cargar las versiones al repositorio. La creación de versiones durante el proyecto se realizó cada vez que se obtenían avances significativos.

8.3. Requerimientos de diseño

La aplicación debe cumplir con algunas características para su funcionamiento. El alcance nos permitió delimitar aquellas tareas específicas que debe realizar la aplicación:

- Inicio de sesión con un usuario y contraseña
- Llevar un registro de usuarios
- Interfaz sencilla y de fácil navegación
- El usuario puede crear de ejercicios de entrenamiento
- Registrar los entrenamientos creados por el usuario
- El usuario puede practicar ejercicios de entrenamiento preprogramados
- Activar un estímulo visual
- Detectar la respuesta ante el estímulo
- Calcular el tiempo de reacción
- Captar los resultados en tiempo real
- Realizar análisis estadístico simple de los resultados

8.4. Implementación

Empezamos a diseñar, analizar, crear y observar en Python, HTML, CSS y JavaScript. Hasta este punto solo nos enfocaremos en el aspecto visual y funcionamiento de la aplicación con la que el usuario interactúa.

8.4.1. Planteamiento del diseño

Para cumplir con los requerimientos de diseño se crearon los siguientes modelos, formularios y vistas bajo un diseño iterativo, combinando la implementación de ellos mediante la escritura de código y verificando si su funcionalidad cumplía con lo deseado de la aplicación.

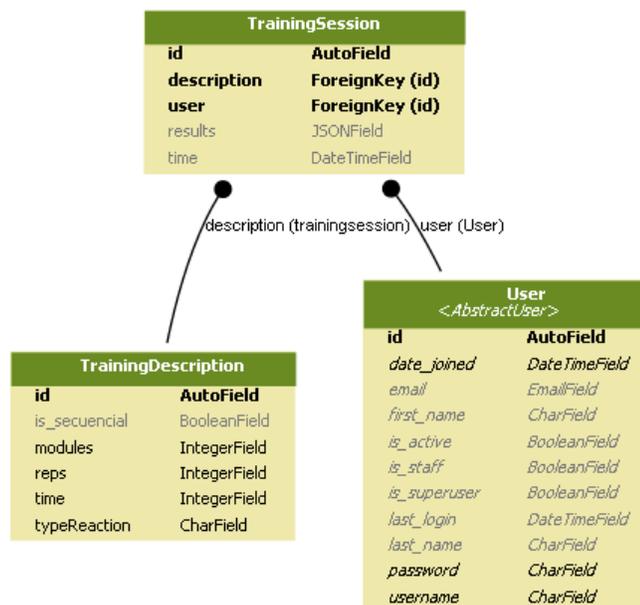


Figura 11: UML de los modelos de descripción y sesión de entrenamiento

Modelos

Como se explica en el marco teórico los modelos son estructuras de datos. Django facilita un modelo de Usuario. El cual se tomó y se le agregaron algunos campos como correo, nombre, apellido para completar los datos de usuario de usuario y contraseña. La idea de agregar estos campos es permitir una interacción con la aplicación más personalizada. Conforme a la funcionalidad que debe cumplir la aplicación se crearon los siguientes modelos:

Descripción de entrenamiento: en este se define el tiempo, cantidad de módulos, cantidad de repeticiones, el tipo de tiempo de reacción a entrenar y si es un entrenamiento con tiempos de activación periódico o aleatorio.

Se parametrizo el tiempo para que el usuario pueda escoger entre 30, 45, 60, 75, 90, 105 y 120 segundos de duración del entrenamiento. Otra opción parametrizada es la cantidad de repeticiones, esta consiste en la cantidad de veces que desea se presente un estímulo, que va desde 10 a 20.

Sesión de entrenamiento: este modelo se genera luego que el usuario ha seleccionado sus parámetros para el entrenamiento y asegura estar listo para iniciar un entrenamiento. Registra el tiempo en el que usuario inicia el entrenamiento, el usuario que entrena, toma una descripción de entrenamiento la cual es la que el usuario ha creado recientemente y posee un campo para la escritura de resultados de la sesión. En la Figura 11 se observan los campos detallados.

Descripción de entrenamiento preprogramado: es una descripción de entrenamiento previamente configurada. Tiene dos campos adicionales, uno es una etiqueta de nombre y el otro son instrucciones. El nombre se utilizó para identificar el entrenamiento preprogra-

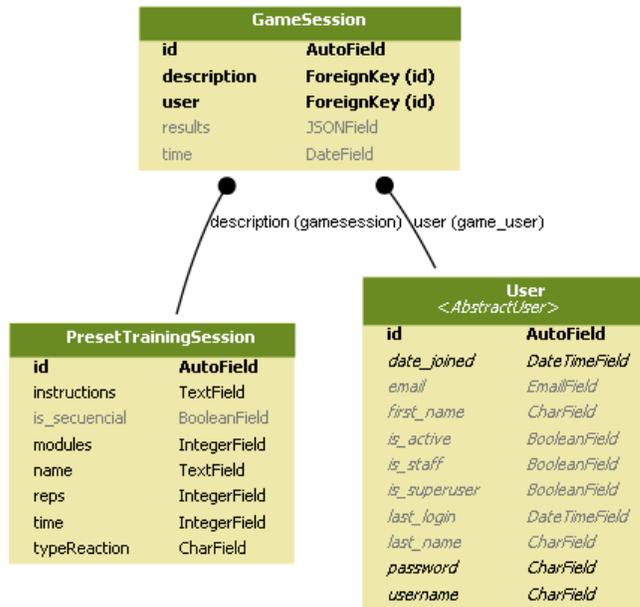


Figura 12: UML de los modelos de descripción predeterminedada y sesión de juego

mado que el usuario seleccionó y las instrucciones para mostrar al usuario en lo que consiste el entrenamiento, por medio de texto se relata de descripción y lo que se espera el usuario realice.

Sesión de juego: contiene los mismos campos que una sesión de entrenamiento, solo que su uso es específico para sesiones de entrenamiento con descripciones creadas por desde el inicio de la aplicación. Haciendo una diferencia entre lo que será un entrenamiento que es algo que el usuario configura y un juego que trae instrucciones ya dadas. La relación entre ambos modelos se observa en la Figura 12.

Formularios

Son la forma en que la aplicación recolecta información que introduce el usuario para crear, modificar o eliminar los campos del modelo. El formulario facilita la creación de los componentes para selección o introducción de la información del modelo por una de las grandes ventajas de Django los template tags. Los template tags nos permiten transferir información de Python a cosas en HTML. No todos los modelos requieren de un formulario para instanciarse, solo se crearon dos formularios. [54]

Registro de usuario: para el registro de un usuario se solicita el nombre de usuario, nombre, apellido, correo, contraseña y verificación de esta. Esta información se almacena en el modelo de usuario.

Creación de descripción de entrenamiento: relacionado con el modelo de descripción de entrenamiento, despliega las opciones del modelo a través de menús despegables. Al usuario seleccionar el tipo, tiempo, cantidad de repeticiones, de módulos y periodicidad del

entrenamiento. Enviar el formulario crea un modelo de descripción de entrenamiento.

Vistas

Cada vista tiene relacionada una URL y una plantilla. Se crearon tantas vistas como plantillas se diseñan. Se tienen vistas para el inicio de sesión, plantilla de página de inicio, plantilla de entrenamiento, plantilla de juegos, plantilla de instrucciones entre otras. También se crearon vistas que permiten obtener los valores del formulario que el usuario selecciono para crear las instancias de los modelos de sesión de entrenamiento y sesión de juego. En las Figuras 14 y 15 se observan dos de las plantillas creadas.

En la vista se ve reflejado en gran parte el código implementado en HTML y CSS para dar estructura y formato a cada una de las plantillas. Las plantillas se adaptaron y se le agrego template tags que facilitan la separación por bloques y que permiten utilizar datos que provienen como parámetros al renderizar la plantilla. Para agregar un template tag se utilizan llaves "{ }", dentro de ellos se escribe el dato proveniente de Python. Esto nos permite tener dentro de nuestro código HTML lógica de Python, por ejemplo, ejecutar un ciclo sobre un arreglo de datos y mostrar cada valor del arreglo en botones, como se realiza en la plantilla de juegos.

Se creo un documento HTML base con toda la estructura que se repetía en cada una de las plantillas, este se extiende mediante template tags al inicio de cada plantilla. Reduciendo así la cantidad de código en cada plantilla.

El mapa del sitio con las plantillas diseñadas se observa en la Figura 13. A través del mapa analizamos la navegación del usuario dentro de la aplicación, la misma aplicación lo guía por lo que la navegación suele ser muy predictiva para el usuario. Casi todas las plantillas presentan la opción de regresar a la página anterior, y también en casi todas podemos dirigirnos directamente al inicio o a cualquier de las dos plantillas de información como lo son contáctanos y preguntas frecuentes.

8.5. Pruebas de funcionamiento

Esta sección detalla los pasos para utilizados para verificar el funcionamiento de la aplicación. Se explican algunas acciones recurrentes realizadas para poner en marcha el servidor de la aplicación.

El desarrollo de la aplicación se realizó en una computadora con sistema operativo Windows 10 Home de 64 bit. Previamente se explicó que se creó un ambiente de desarrollo para la aplicación en Anaconda. Para la activación del ambiente de desarrollo se utiliza el comando `activate py35`. El nombre del ambiente se define cuando se crea, en este caso se nombró así por la versión de Python que utiliza. En la Figura 16 se observa que cuando el ambiente es activado este aparece con paréntesis al lado izquierdo del directorio.

En el ambiente se instaló la última versión de Django disponible, la versión instalada fue la 2.0.6. Utilizando los siguientes comandos: `pip install django`

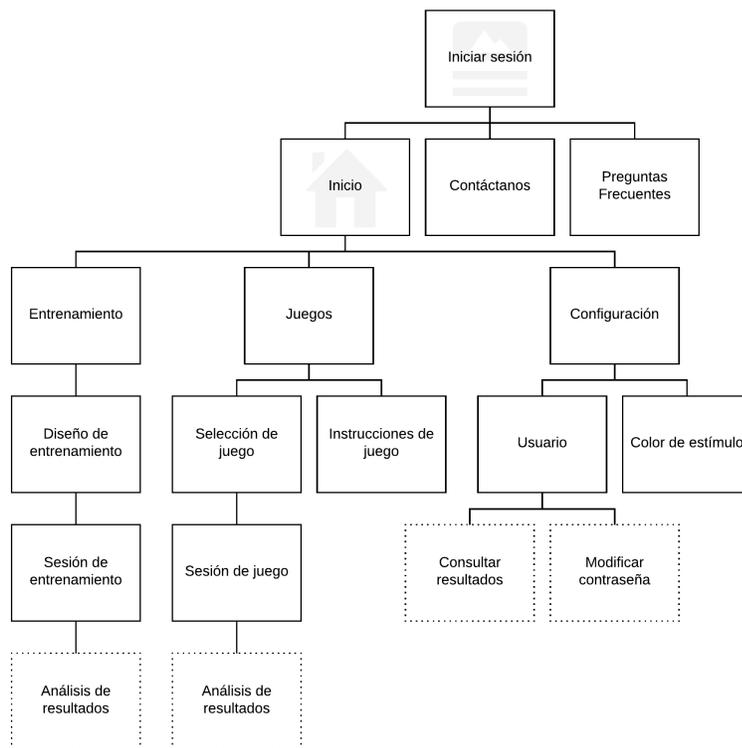


Figura 13: Mapa del sitio de la aplicación

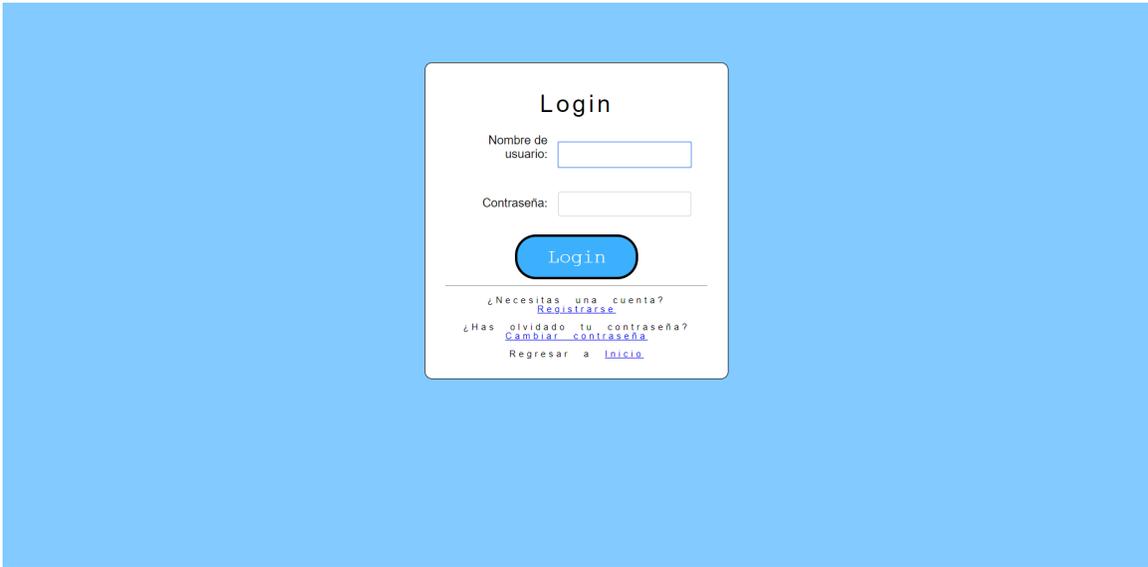


Figura 14: Plantilla de inicio de sesión

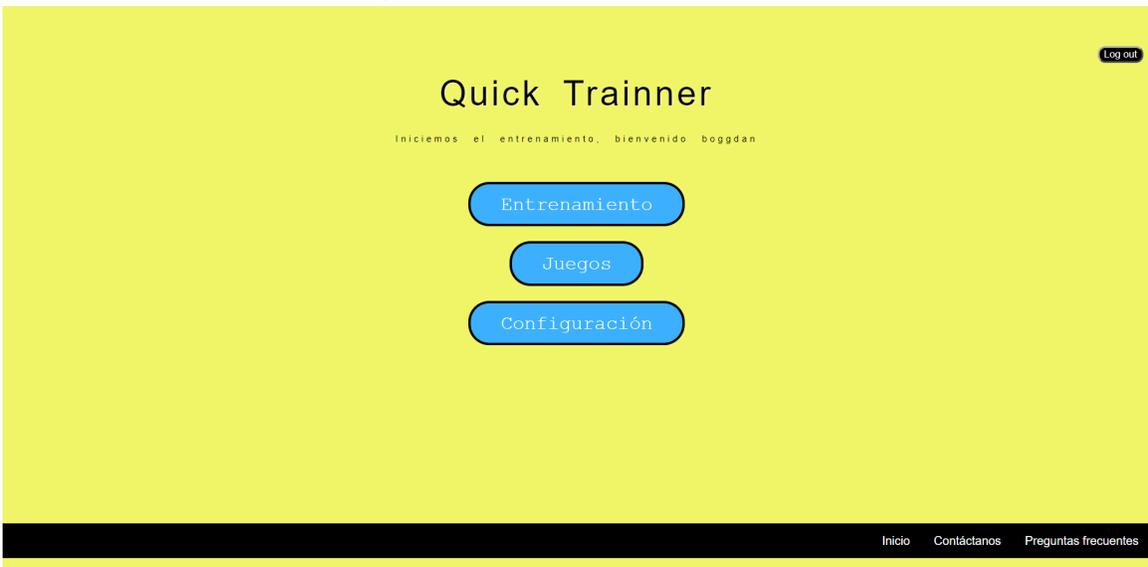
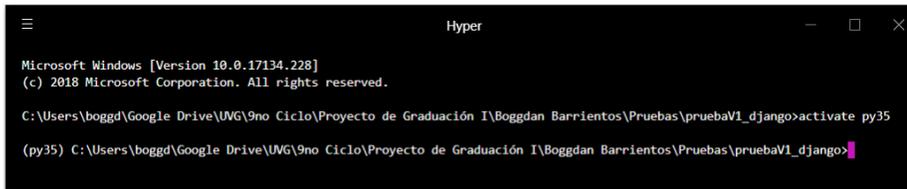


Figura 15: Plantilla del menú principal

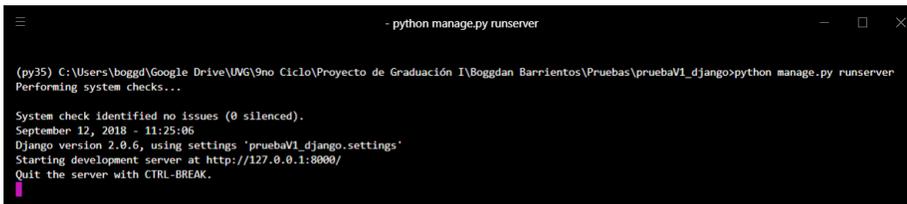


```
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\boggd\Google Drive\WG\9no Ciclo\Proyecto de Graduación I\Boggdan Barrientos\Pruebas\pruebaV1_django>activate py35

(py35) C:\Users\boggd\Google Drive\WG\9no Ciclo\Proyecto de Graduación I\Boggdan Barrientos\Pruebas\pruebaV1_django>
```

Figura 16: Activación del ambiente de desarrollo



```
- python manage.py runserver

(py35) C:\Users\boggd\Google Drive\WG\9no Ciclo\Proyecto de Graduación I\Boggdan Barrientos\Pruebas\pruebaV1_django>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
September 12, 2018 - 11:25:06
Django version 2.0.6, using settings 'pruebaV1_django.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Figura 17: Ejecución del servidor de la aplicación

Para ejecutar el servidor de la aplicación en la computadora se utiliza la consola de comandos y desde el directorio donde se desarrolla el proyecto se realiza la escritura del comando:

```
python manage.py runserver
```

Este inicia la ejecución del servidor. En la Figura 17 se observa el mensaje que se muestra en la consola si la ejecución se ha realizado con éxito. Se utilizó un navegador web, durante el desarrollo se realizaron pruebas con Google Chrome y Mozilla Firefox; para acceder a la dirección y puerto donde se ejecuta el servidor. La dirección `127.0.0.1:8000` se escribe en la barra de navegación para acceder a la aplicación.

Los detalles de los que se hablaba al inicio de la sección se añadieron o corrigieron durante la ejecución del servidor. Este recarga los archivos estáticos como lo son HTML, CSS y JavaScript si se presenta algún cambio en ellos. Se realizaron cambios en la estructura o estilo para dejar la mejor experiencia visual al usuario. Para realizar las pruebas sobre la información de los resultados se utilizó la función `print`. Cuando la información era enviada al servidor a través de un método POST este se imprimía en la consola. Si coincidía con la información seleccionada la operación había exitosa. Al igual para el desarrollo nos apoyamos en el sitio administrativo que provee Django dónde se pueden verificar las instancias creadas de cada modelo. Durante el proceso de prueba de la aplicación hubo errores que se corrigieron buscando en la documentación, buscando el nombre del error en foros o intentando una nueva forma de la cual se creía este se solucionaría, fue un proceso iterativo hasta el punto de encontrar una solución.

Hasta este punto hemos alcanzado a reunir modelos, formularios y plantillas en un solo sitio. La navegación dentro del sitio no será explicada, la aplicación y el mapa del sitio 13 brindan la información suficiente para que un usuario pueda acceder a cada página de la aplicación sin algún problema.

Ejercicios de entrenamiento de tiempo de reacción

La creación de ejercicios de tiempo de reacción puede consistir en instrucciones sencillas para generar un estímulo, al cual se debe reaccionar. Como puede ser algo tan complejo al cual usuario, varios estímulos y solo deba reaccionar a alguno. Entrenar precisión o solo velocidad de respuesta.

Lo más importante para iniciar el proceso de diseño fue definir el tipo de estímulo que se utilizaría, visual. A partir de ello queremos conocer el tiempo que toma el usuario en realizar una acción de respuesta a dicho estímulo. El enfoque de los ejercicios será la velocidad con la que reacciona ante el estímulo, por ello se toma cuidado del cálculo del tiempo de reacción.

Se definió utilizar LED y botones (Push Button) para el estímulo y el dispositivo de captación respectivamente. El desarrollo del código que genera la activación de los estímulos, detección de respuesta y cálculo del tiempo de reacción se realizó en Node.js, una implementación de JavaScript. Se explica brevemente por secciones las acciones realizadas para el diseño de los ejercicios de entrenamiento en analogía a los pasos que se realizan cuando se realiza un viaje.

9.1. Librerías de Node.js

El objetivo del ejercicio de entrenamiento de tiempo de reacción se puede resumir en: activar, detectar y calcular.

Para este desarrollo se utilizó Node.js. La versatilidad de este ambiente y la gran cantidad de paquetes de software que se le pueden añadir fue clave para el diseño de los ejercicios. La conexión entre los LEDs, Push Button y Raspberry Pi Zero W se realizó a través de los puertos GPIO. Algunos de los paquetes de software a utilizados fueron los siguientes:

9.1.1. onoff

Es un paquete de software que permite acceder a los puertos GPIO y detectar interrupciones en placas como Raspberry Pi, C.H.I.P o BeagleOne [55].

Su instalación se puede realizar con la siguiente línea de código: `npm install onoff`

Las siguientes líneas de código muestran de forma general como se importa la librería y como se define cada pin como entrada o salida. El código desarrollado contiene el resto de las definiciones utilizadas. Se destaca la configuración en la definición de una entrada que permite la librería, permite escoger el tipo de flanco que desea detectar y agregar un tiempo en milisegundos de antirrebote para evitar que el software detecte que el botón se presionó varias ocasiones cuando fue solo una.

```
//Incluir la libreria onoff para interactuar con los puertos GPIO
var Gpio = require('onoff').Gpio;
//Definir el pin 4 como salida
var led1 = new Gpio(4, 'out ');
//Definir el pin 18 como entrada
var pushButton1 = new Gpio(18, 'in ', 'falling ', {debounceTimeout: 3});
```

De esta librería se utilizaron varias funciones, tres fueron trascendentales para el desarrollo de los ejercicios de entrenamiento.

- `readSync()`, permite leer el estado del puerto GPIO de manera síncrona. Retorna un valor de 0 o 1 que representa dicho estado.
- `writeSync(value)`, el valor debe ser 0 o 1 para escribir que el puerto se apague o encienda respectivamente.
- `watch(callback)`, permite verificar si se presentan interrupciones de hardware en los puertos GPIO. El parámetro del tipo de flanco determina el tipo de interrupción que se espera. La función de *callback* toma dos argumentos uno para

El paquete de software posee otras funciones, las cuales se encuentran detalladas en su documentación.

9.1.2. socket.io

Permite la comunicación bidireccional en tiempo real basado en eventos. Consiste en un servidor generado por este paquete y un cliente en una librería de JavaScript para el navegador web.

Su instalación se puede realizar con la siguiente línea de código: `npm install socket.io`

También utilizan el paquete express, un framework de desarrollo web para Node.js. Express se utilizó únicamente para la implementación de socket.io y se puede descargar de la misma forma que los anteriores paquetes. Para utilizarlo en su documentación lo explican a través de las siguientes líneas de código.

```
var app = require('express')();
var server = require('http').createServer(app);
var io = require('socket.io')(server);
io.on('connection', function(socket){ /* */ });
server.listen(5000);
```

Luego haber escrito esa configuración, dentro de la función se pueden escribir todos los eventos que se desean emitir o recibir.

Para crear un evento utilizamos la siguiente sintaxis:

```
io.on('nombre del evento', datos a enviar)
```

Este evento debe ser esperado por alguno de los clientes con el mismo nombre del evento para crear el canal y recibir los datos enviados a través de este.

Para recibir un evento la sintaxis es muy similar:

```
io.emit('nombre del evento', function(datos a recibir){ /* */}).
```

Dentro de la función se realiza la manipulación de los datos recibidos, a partir de un evento recibido se puede crear otro evento y así con cada uno. Por ello el gran potencial de socket.io para aplicación fue obtener un resultado de tiempo de reacción y enviarlo a través de un nuevo evento al cliente para que este lo mostrará en la plantilla.

9.2. Lógica de programación

Un entrenamiento tiene una descripción, este viene del modelo. El entrenamiento debe tener como parámetro cada uno de los elementos de la descripción y partir de ellos generar toda la información restante para crear un entreno.

Los parámetros que se emitirán a través del canal de socket.io desde el cliente cuando se desee realizar un entreno, son los siguientes:

- Tipo de tiempo de reacción a entrenar
- Tiempo total del entrenamiento (totalTime)
- Cantidad de activaciones del estímulo (noTry)
- Cantidad de dispositivos de captación a utilizar (modules)
- Periodicidad de la activación del estímulo (secuencial)

De aquí en adelante se tratarán conceptos utilizados en el desarrollo del código como: tiempo de activación, tiempo de desactivación, módulo a activar, tiempo de reacción, estado, resultado de estímulo.

9.2.1. Tiempo de activación

El servidor toma los datos de la descripción para generar el entrenamiento y partir de ellos el tiempo de activación de estímulo, definido en la programación como `stepTime`. El `stepTime` es el tiempo máximo que el estímulo puede permanecer activo. Los tiempos de activación se generan y se almacenan en un arreglo de datos. El tiempo de activación se calcula de distinta forma según la periodicidad de activación del estímulo.

Si el entrenamiento es periódico el tiempo de activación se calcula con la ecuación 2. Solo involucra el `stepTime` y este se multiplica dentro de un ciclo para generar tantos tiempos como cantidad de activaciones existan. Por ejemplo, si el `stepTime` es igual a 3000 milisegundos, los estímulos se activarán cuando hayan transcurrido 3000,6000,9000,12000, ... milisegundos, así hasta completar la cantidad de activaciones.

$$stepTime = \frac{totalTime}{noTry} \quad (2)$$

En cambio, si el entrenamiento no posee periodicidad en la activación de estímulos este tiempo se calcula con la siguiente línea de código. Esta utiliza una función `getRandomInt` que números aleatorios entre ambos parámetros de la función. El primer parámetro es 0 y el segundo es el valor de `stepTime` menos un factor de este. El resultado aleatorio de la función se le suma al `stepTime` para conformar así el tiempo de activación.

```
activationTime = getRandomInt(0,(stepTime-(stepTime/K)))+(i*stepTime))
```

En el código el valor de `K` es igual a 5, se definió así al calcular este valor de `stepTime` máximo y mínimo que las combinaciones de parámetros de entrenamiento permiten. El valor de `i` es el iterador dentro del ciclo `for` utilizado para generar los tiempos de activación. En la Figura 18 se observa un ejemplo de generación de tiempos aleatorio con tiempo total de 45000 milisegundos y con una cantidad de activaciones de 10.

```
> stepTime
< 4500
> activationTime = []
< ▶ []
>   for (var i = 0; i < noTry; i++) {
       activationTime.push(getRandomInt(0,(stepTime-(stepTime/5)))+(i*stepTime));
   }
< 10
> activationTime
< ▶ (10) [3034, 7926, 9041, 13701, 18262, 22553, 29460, 33457, 37085, 43259]
```

Figura 18: Ciclo de generación de tiempo de activación aleatorio

9.2.2. Tiempo de desactivación

El tiempo en cual se desactiva el estímulo si el usuario no ha reaccionado ante él. Al igual que el tiempo de activación, se calcula de dos formas distintas dependiendo la periodicidad.

Si es periódico se define un tiempo muerto que será el que transcurra entre activación como `deadTime`. Por lo tanto, se desactivará el estímulo si luego de la activación a transcurrido este `deadTime`. La forma en que se calcula el `deadTime` se realiza con la ecuación 3. Se definió un tiempo de 500 milisegundos como el tiempo adecuado para que el usuario pueda percibir que el estímulo se desactivo por no haber realizado una acción de respuesta y pueda percibir la siguiente activación.

$$deadTime = stepTime - 500 \quad (3)$$

Si el entrenamiento es de activaciones aleatorias, el tiempo de desactivación será el mismo `stepTime`. Para generar los tiempos de activación y desactivación se realizan dentro de un mismo ciclo `for`. En la Figura 19 se observan como los tiempos de activación y desactivación periódicos se intercalan, primero se activa a los 4500 milisegundos, si el usuario no lo desactiva este estímulo se apagará a los 8500 milisegundos.

```
> activationTime
< ▶ (10) [4500, 9000, 13500, 18000, 22500, 27000, 31500, 36000, 40500, 45000]
> desactivationTime
< ▶ (10) [8500, 13000, 17500, 22000, 26500, 31000, 35500, 40000, 44500, 49000]
```

Figura 19: Tiempos de activación

9.2.3. Generación de estímulos

La cantidad de estímulos visuales está relacionado con la cantidad de módulos que se utilizan durante un entrenamiento. Para el desarrollo de la aplicación se trabajó un número máximo de 4, por lo tanto, se utilizó la misma cantidad de LEDs.

Se desarrollaron dos formas de trabajo cuando se utiliza un único módulo o cuando la cantidad módulos es mayor. Si se seleccionó un único módulo, la aplicación tiene definido por defecto el módulo a activarse y solamente estará pendiente de las interrupciones al Push Button asociado a este. En cambio, si la cantidad de módulos seleccionados es mayor a 1, se genera un orden aleatorio de activación entre los módulos seleccionados y todos los Push Button estarán pendientes de alguna interrupción.

Se desarrollo la función `changeState()` que permite cambiar el estado de un puerto GPIO configurado como salida. Toma como parámetros el puerto y el estado a escribir en este. Esta función se utiliza para encender o apagar el estímulo escribiendo como estado 1 que corresponde a encendido y 0 a apagado. Esta retorna el valor que del estado que escribe en el puerto.

Para realizar la activación y desactivación del estímulo se utilizó la función `setTimeout()` que toma parámetros una función y un tiempo en milisegundos. Lo que hace es evaluar la función que tiene como parámetro luego que ha transcurrido el tiempo especificado. En la función que toma como parámetro definimos la acción de activar o desactivar el estímulo y el tiempo se especificó con cada uno de los valores generados en los pasos anteriores. Creamos un arreglo de métodos de `setTimeout()` tanto para activar y desactivar.

La función `clearTimeout()` permite evitar la ejecución de un `setTimeout()` previamente generado. Esta acción se utiliza cuando el usuario reacciona en tiempo al estímulo, por lo tanto, se desactiva el apagado del estímulo por software.

Las funciones provistas por JavaScript forman parte de la lógica realizada dentro de funciones desarrolladas `onSetTimeout()` y `offSetTimeout()`. Estas se utilizan específicamente cuando el entrenamiento utiliza más de un módulo. Estas funciones toman como parámetro el valor de iteración de un ciclo `for`. Este permite acceder a los arreglos de datos y al dato que contiene el tiempo y módulo a activar según el orden generado.

9.2.4. Captura del tiempo de reacción

Para medir el tiempo se utilizó el constructor `Date` que es un objeto global de JavaScript. Este mide la fecha en milisegundos desde la media noche exacta del 01 de enero de 1970 en formato UTC. Presenta un comportamiento uniforme entre plataformas. El método utilizado fue `now()` el cual retorna el valor numérico correspondiente a la fecha actual [56].

Este método utiliza el tiempo del sistema como referencia, por lo que, aunque el tiempo del sistema se encuentre configurado en alguna otra región que no coincida, el cálculo del tiempo de reacción se realiza bajo la misma referencia.

El diagrama de la Figura 20 explica cómo se realiza el proceso de captura del tiempo de reacción en la aplicación, se guarda el tiempo cuando se activa el estímulo y cuando este se desactiva, luego se realiza una resta entre ambos. Si el estímulo no es desactivado mediante una respuesta del usuario, se desactiva por software y no se calcula ningún de tiempo ya que la reacción fue inexistente.

9.2.5. Resultados del entrenamiento

El entrenamiento puede tener tres resultados posibles: acierto, desacierto o error. El Cuadro 5 muestra cada posible resultado con su descripción, valor y expresión el cual se da en la aplicación.

9.3. Descripción de código

El diagrama de flujo de la Figura 21 explica de forma general cuales son los procesos que se realizan durante la generación de un entrenamiento. Se utilizan los procesos explicados anteriormente como lo es la generación de los tiempos y del orden de los módulos si es que

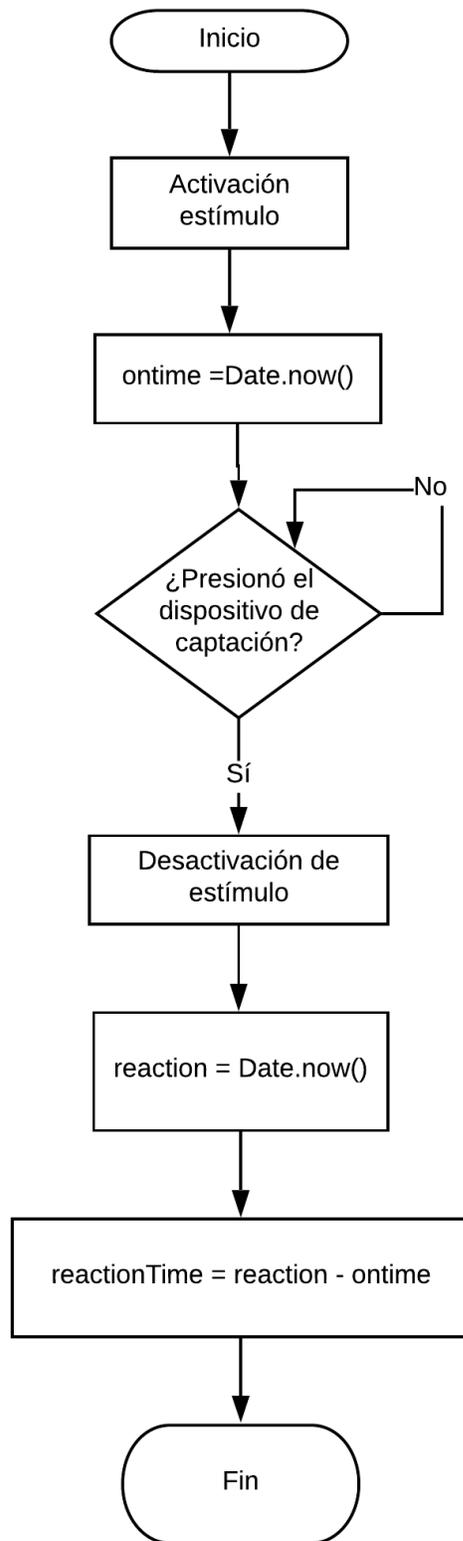


Figura 20: Diagrama de flujo de captura de tiempo de reacción

Resultado	Descripción	Valor	Expresión
Acierto	El usuario da respuesta al dispositivo de captación asociado al estímulo activado. Genera un tiempo de reacción	Valor numérico del tiempo de reacción en milisegundos, ej: 528	Hit
Desacierto	El usuario no da una respuesta o da una respuesta muy tardía que puede incurrir en un error	--	Miss
Error	El usuario da una respuesta a un dispositivo de captación el cual no esta asociado al estímulo activado	--	Error

Cuadro 5: Resultados de entrenamiento

usa más de uno. El proceso de tiempo de captura generalizado se utiliza de la misma forma en todos los entrenamientos.

Cada vez que se obtiene un resultado este mismo se emite a través de un evento por socket.io que el cliente espera para mostrarlo en la plantilla. Al final del entrenamiento se guardan los resultados a la base de datos.

9.4. Interfaz entre Django y Node.js

Luego de toda la lógica para la creación del entrenamiento, la activación de estímulos y detección de respuestas ante ellos. Las pruebas fueron diversas y continuas. La combinación entre el desarrollo en Node.js con el Django. Como primer paso se estableció la conexión entre ambos ambientes, lo cual se desarrolló en las plantillas y scripts.

En la aplicación se hizo referencia al servidor de socket.io desde la plantilla HTML. Esto permitió la comunicación entre la plantilla mostrada por Django y el servidor de socket.io en Node.js. En el archivo de Javascript asociado a la plantilla también se establece la conexión como cliente al servidor de socket.io lo que permití emitir y recibir mensajes, a través del cual se actualizan los eventos provenientes del servidor a la plantilla y permite una visualización en tiempo real.

Luego de algunos pasos, como lo es configurar su entrenamiento e indicar que se encuentra listo para iniciar el entrenamiento. Todo comienza a funcionar, se activan estímulos visuales, la aplicación se queda a espera de una respuesta del usuario, si esta se da se genera un primer tiempo de reacción, sino se marca como un desacierto.

Los resultados se reflejan en tiempo real en la interfaz. La Figura 22 es el resultado que se observa en la interfaz de la aplicación de un entrenamiento finalizado.

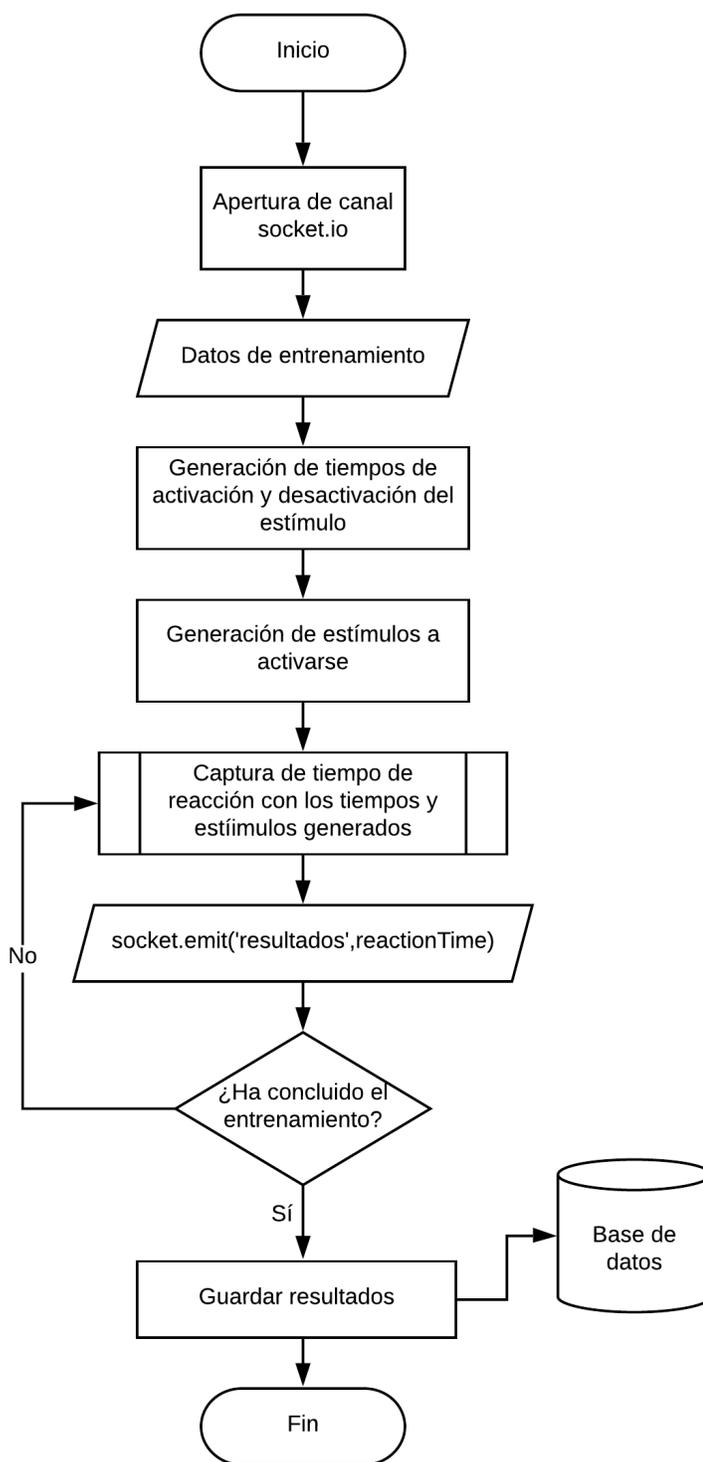


Figura 21: Diagrama de flujo de un entrenamiento general

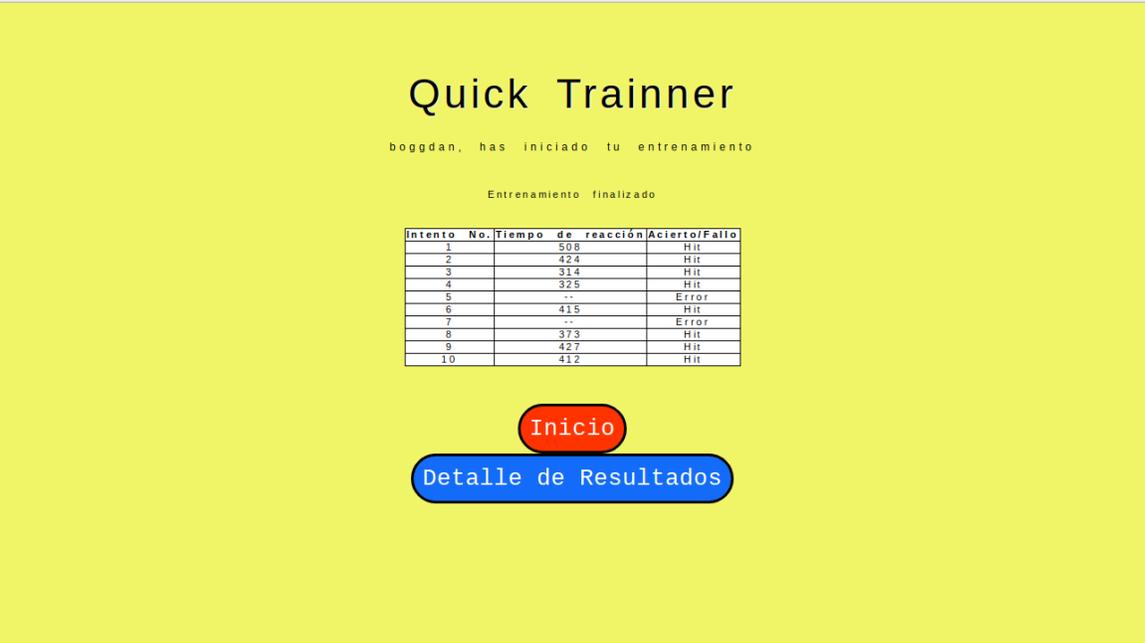


Figura 22: Interfaz de la aplicación concluido un entrenamiento

Implementación de aplicación web en Raspberry Pi Zero W

En los anteriores capítulos se ha desarrollado la aplicación. En sí son dos partes que se unen dentro de una computadora Raspberry Pi Zero W. Dentro de esta pequeña placa es donde sucede todo y a través de ella se puede acceder a la aplicación web con su función de punto de acceso inalámbrico.

10.1. Instalación de ambiente

Para iniciar el trabajo de implementación en la Raspberry Pi se necesita de herramientas de software para facilitar las pruebas y la implementación final.

10.1.1. Instalación del sistema operativo

Este proceso se resume en dos herramientas para el proceso, una computadora con acceso a internet y lector SD, y una tarjeta microSD.

Desde la computadora se debe acceder al sitio de Raspberry Pi y descargar la versión de RASPBIAN STRECH WITH DESKTOP. Esta la versión utilizada para el desarrollo de la aplicación. Se pueden conseguir versiones anteriores, pero no se puede asegurar el correcto funcionamiento de la aplicación. Se realizó la descarga de un archivo .zip.

Adicionalmente se descargó el software Etcher desde su página web <https://etcher.io/>. Este requiere de instalación en el sistema operativo que utilice la computadora.

Cuando se ha completado la descarga de la imagen de Raspbian, se introduce la tarjeta microSD a la computadora, usualmente a través de un adaptador SD. Se abrió el software

Etcher, se selecciona el .zip de la imagen y la memoria SD. Luego presiona ‘Flash!’ y cuando el proceso acaba se tiene una memoria SD con el sistema operativo Raspbian Stretch en ella. Esta memoria ya puede ser introducida a la Raspberry Pi y comenzar a transferir los archivos.

10.1.2. Programas de comunicación

Para comunicarse con la Raspberry Pi desde la computadora se utilizó SSH, para lo cual en la computadora se instalaron dos programas que facilitarían el proceso de comunicación.

Putty, que es un software que permite ser cliente SSH. Se puede descargar desde su sitio <https://www.putty.org/>.

A través del protocolo SSH también se pueden traspasar archivos, para lo cual se utilizó el programa FileZilla. Se descarga desde su sitio <https://filezilla-project.org/>.

Luego de la instalación de ambos programas se puede iniciar el proceso de adecuar todo lo desarrollado en la computadora a la Raspberry Pi.

Como se explica en la sección de Accesorios en el capítulo El Arte de la selección, la primera configuración realizada fue la conexión a una red WiFi y habilitar el protocolo SSH, esto se realizó desde la interfaz gráfica del sistema operativo conectado un monitor y un teclado a la Raspberry. También se sugiere cambiar la contraseña de la Raspberry. Para realizar ambos procesos existe suficiente documentación en la web.

10.1.3. Configuración del cliente y comunicación SSH

La Raspberry Pi se encuentra conectada a la misma red WiFi a la que se encuentra conectado el computador desde el cual accederemos. Para ello debemos conocer la dirección IP que el Router le asigna a la Raspberry.

Hay varias opciones para conocer dicha dirección: Puede ser desde la consola de la Raspberry ejecutar el comando `sudo ifconfig` y en el resultado que despliega ubicar la red wlan0. Dentro del mismo apartado encontraremos la dirección IP.

Otra forma es acceder a la configuración del Router y anotar la dirección que este asigna al dispositivo. Otra opción muy similar y que durante el desarrollo de la implementación se utilizó fue la opción que posee Windows 10 de activar la computadora como Mobile Hotspot o Punto de Acceso y conectar la Raspberry a esta red. Con ello es mucho más simple conocer la dirección IP que este le asigna a la Raspberry.

Para utilizar Putty, en la barra de dirección del Host Name o dirección IP escribimos la dirección IP como se muestra en la Figura 23. Cuando solicite el usuario accedemos con el usuario “pi” y la contraseña que se definió, ingresar con la contraseña “raspberrypi”. Este proceso nos debería permitir acceder a la consola de la Raspberry y a partir de ella realizar cambios, actualizaciones, crear configuraciones.

El proceso para acceder con FileZilla es muy similar, solo que en esta se escriben todos

los parámetros antes de establecer la conexión se debe escribir la dirección IP, nombre de usuario, contraseña y puerto, este debe ser el 22.

Estas herramientas fueron utilizadas repetidamente para realizar las pruebas de funcionamiento de los entrenamientos programados. A través de FileZilla se traspasaron los archivos desde la computadora a la Raspberry y luego con Putty desde la consola eran ejecutados. Ambos programas estaban en comunicación con la Raspberry Pi al mismo tiempo.

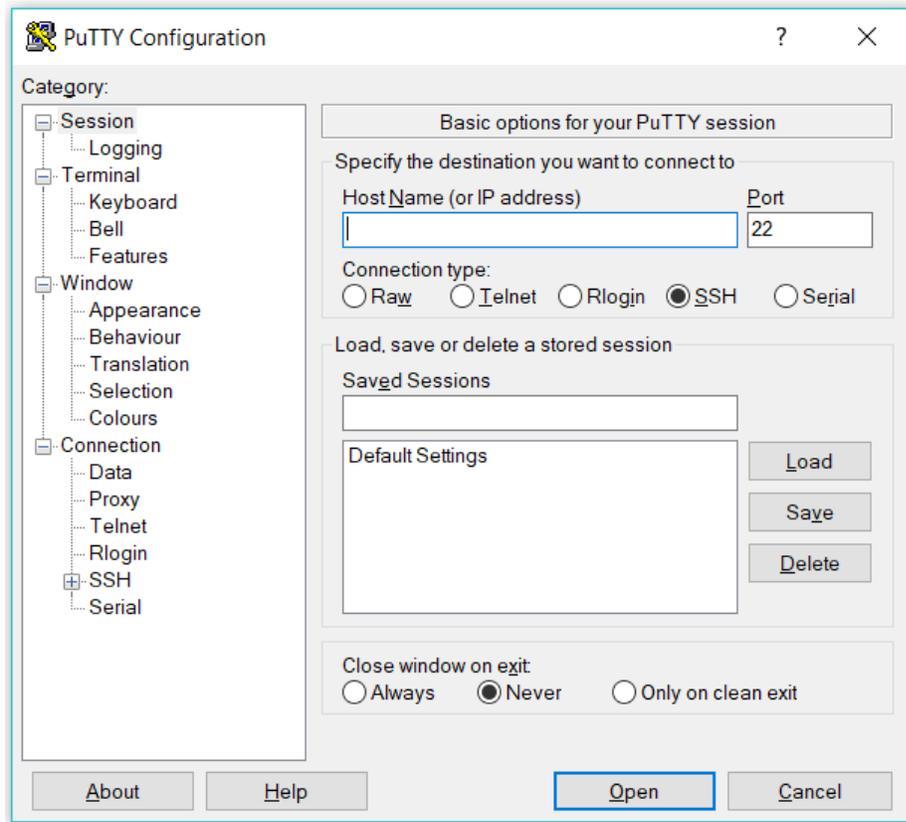


Figura 23: Interfaz de Putty

10.2. Instalación e implementación de software

El trabajo en la Raspberry fue enfocado en buscar la compatibilidad de las acciones que se realizan desde una computadora con un sistema operativo como Windows a un sistema operativo como lo es Raspbian. El primer paso fue instalar muchos de los mismos paquetes de software que se utilizaron para el desarrollo de aplicación ahora en otro ambiente.

10.2.1. Instalación de Django

Raspbian trae dos versiones de Python instalada por defecto: Python 2.7.13 y Python 3.5.3.

Para utilizar la versión 3.5.3 se utilizan los mismos comandos de Python añadiendo un 3 al final. Para ejecutar el Python Shell con la versión 3.5.3 se utiliza el comando `python3`. Para instalar paquetes de software se utiliza el mismo sistema de gestión de paquetes que posee Python, `pip`. Con la variable `pip3` todos los paquetes se instalarán en el ambiente con la versión de Python con la que se desarrolló el proyecto.

No se recomienda actualizar la versión de `pip3` ya que este cambia el nombre de la variable de entorno en el sistema a `pip`, sobrescribiendo la que existe para la versión de Python 2.7. Posiblemente con mayor cuidado se puede realizar la actualización sin afectarla variable de entorno.

Para instalar Django utilizamos los siguientes comandos:

```
$ pip3 install django
$ pip3 install psycopg2
```

Si alguno de los comandos anteriores falla, siempre se recomienda utilizar los siguientes comandos y luego reintentar con el comando anterior.

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Se puede volver un proceso tardado si alguno falla, seguir iterando hasta que se tenga éxito.

10.2.2. Instalación de PostgreSQL

Para instalar la base de datos PostgreSQL debemos seguir los siguientes comandos:

```
$ pyenv deactivate
$ sudo apt-get install libpq-dev python-dev
$ sudo apt-get install postgresql postgresql-9.5
$ sudo -u postgres psql
```

Para configurar la base de datos y crearla en la aplicación se utilizan los siguientes comandos: Accediendo al usuario de Postgres

```
$ sudo su postgres
```

Creamos un usuario para la base de datos, utilizaremos el usuario por defecto `pi`.

```
$ createuser pi -P --interactive
```

Nos pedirá una contraseña la cual debemos apuntar para no olvidar. A las siguientes preguntas responder **n**, **y**, **y**. Ahora nos conectaremos a PostgreSQL con el comando `psql` como se observa en los siguientes comandos:

```
$ psql
> CREATE DATABASE training_app
```

`training_app` es el nombre de la base de datos que se configuro en la aplicación, por eso se crea una base de datos en la Raspberry Pi con el mismo nombre. Luego de realizar esta configuración, la utilización del programa de desarrollado en la computadora es sencillo y muy similar. Quizás falta crear las migraciones de los modelos, Django cuando desee poner en marcha el servidor le recordará y no será nada distinto a lo trabajado en la computadora, será un `python3 manage.py makemigrations` y luego utilizar un `python3 manage.py migrate` para crear los modelos en la nueva base de datos.

10.3. Punto de acceso inalámbrico

La aplicación utiliza la Raspberry Pi como un punto de acceso. Parte fundamental del modelo de funcionamiento es configurar a la Zero W como punto de acceso y que otros dispositivos se puedan conectar a la red creada. También es importante poder configurar los comandos para que esta funcione como cliente de una red nuevamente.

Este fue uno de los procesos más tediosos y frustrantes, consultar foros que mencionaban cómo realizarlo, ninguno funcionó con claridad. Tomando pequeñas partes de unos y combinadoló con la documentación oficial se logró definir una serie de pasos estipulados para poder cambiar entre modo punto de acceso y cliente de una forma relativamente más sencilla a como lo presentaban algunos autores.

Se deben instalar algunos paquetes, las siguientes líneas guían el proceso de instalación.

```
$ sudo apt-get install dnsmasq
$ sudo apt-get hostapd
```

Si alguno de ellos falla durante el proceso de instalación o se muestra algún error similar como lo puede ser `E: unable to locate package`. Se deben ejecutar los siguientes comandos e intentar de nuevo hasta tener éxito.

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

10.3.1. De cliente a punto de acceso

La configuración inicial se puede realizar con la documentación que presenta Raspberry Pi [57]. El tutorial está muy bien explicado, se debe seguir hasta el punto de iniciar los servicios "Start it up", con eso será suficiente para habilitar a la Zero W como punto de acceso. Cuando se realizó esta configuración el nombre de la red se definió como: `RPI_network` y la contraseña como: `boggdanB`.

Luego de realizada la configuración inicial para habilitar a la placa como punto de acceso, los pasos para activarlo cuando se encuentre en modo cliente son los siguientes:

Abrir el archivo de la configuración

```
$ sudo nano /etc/dhcpd.conf
```

La línea donde se asigna dirección ip debe estar descomentada.

```
interface wlan0
static ip_address=192.168.4.1/24
nohook wpa_supplicant
```

Se deben comentar líneas del archivo, si estas no existen se pueden escribir ya comentadas.

```
$ sudo nano /etc/network/interfaces
```

El archivo debería lucir de la siguiente manera:

```
#auto lo
#iface lo inet loopback

#allow-hotplug wlan0
#iface wlan0 inet manual
#wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Se habilita el servicio de punto de acceso desde el arranque

```
$ sudo update-rc.d hostapd enable
```

Se procede a reiniciar el sistema

```
$ sudo reboot
```

Cuando este inicie nuevamente, la raspberry será un punto de acceso con las credenciales configuradas.

10.3.2. De punto de acceso a cliente

La Raspberry Pi tendrá acceso nuevamente a una red WiFi para acceder a internet. Para realizar este proceso es necesario configurarlo desde la Raspberry, utilizando un teclado y monitor ya que desde el modo de punto de acceso no se puede acceder a través de SSH.

Los pasos a escribir y modificar desde la consola son los siguientes:

Detener los servicios de punto de acceso

```
$ sudo systemctl stop hostapd
$ sudo update-rc.d hostapd disable
```

Abrir el archivo y modificarlo

```
$ sudo nano /etc/network/interfaces
```

Consiste en descomentar y el resultado debería lucir de la siguiente forma

```
auto lo
iface lo inet loopback

allow-hotplug wlan0
iface wlan0 inet manual
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Activaremos el servicio de wlan0, esto debería cambiar el ícono a cliente WiFi en pantalla

```
$ sudo ifup wlan0
```

Accedemos al siguiente archivo y lo modificamos

```
$ sudo nano /etc/dhcpd.conf
```

Se comenta la línea donde se asigna un dirección estática, se debe ver de la siguiente manera

```
interface wlan0
#static ip_address=192.168.4.1/24
nohook wpa_supplicant
```

Reiniciamos `$ sudo reboot`. Cuando la Raspberry Pi inicie nuevamente, está se conectará a una red WiFi previamente utilizada y se podrá acceder a ella nuevamente a través de SSH.

10.4. Prueba de conexión

Se han definido todos los procesos para ejecutar la aplicación web dentro de la Raspberry Pi, luego de transferir los archivos de la aplicación desarrollada en Django y el servidor de socket.io de Node.js. Se procede a ejecutarlos desde el directorio de cada uno de ellos:

```
$ python3 manage.py runserver 192.168.4.1:8000
$ sudo node webserverRPI.js
```

Nos conectamos a la red WiFi que ofrece la Raspberry Pi, abrimos el navegador, escribimos la dirección `192.168.4.1:8000` y habremos accedido a la aplicación web como se observa en la figura (24).

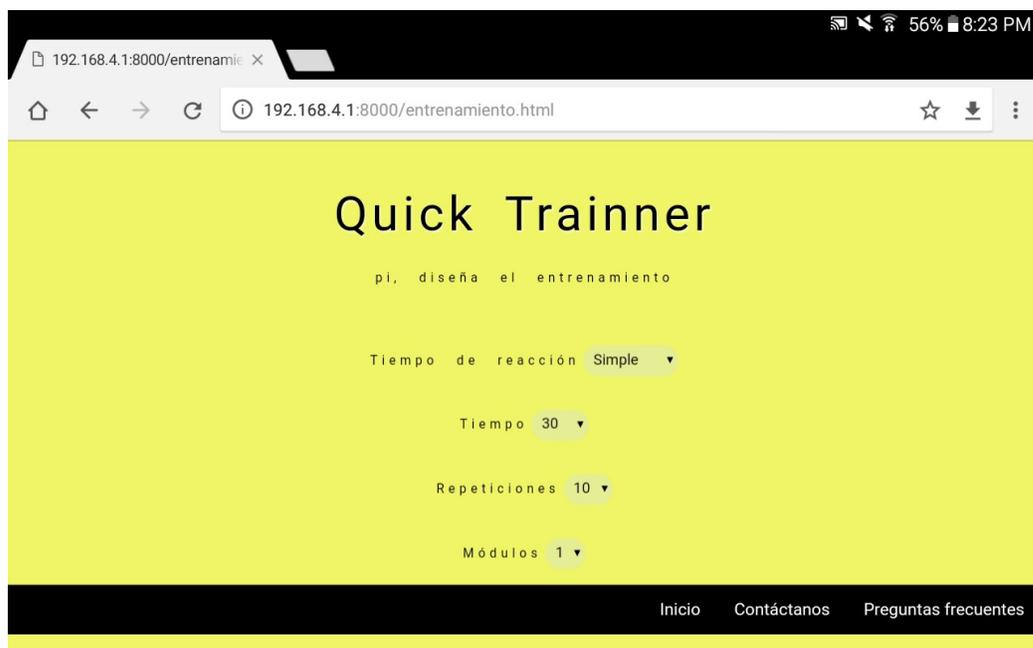


Figura 24: Interfaz de la aplicación accedida desde una tablet

1. La aplicación web se utiliza como interfaz entre el usuario y el hardware que ejecuta el programa que permite la activación de estímulos y detección de respuesta a estos.
2. La aplicación web permite al usuario crear un entrenamiento orientado a la medición de tiempo de reacción, seleccionando parámetros definidos como el tiempo de entreno, cantidad de activaciones del estímulo, cantidad de estímulos a activar y periodicidad de activación del estímulo.
3. El acceso a la aplicación web se realiza a través de un punto de acceso inalámbrico creado desde la Raspberry Pi Zero W. Smartphones, tabletas y computadoras se pueden conectar a esta red y acceder a la aplicación desde un navegador web.
4. Los resultados de un ejercicio de entrenamiento se muestran en tiempo real en la interfaz de la aplicación.
5. La aplicación despliega los tiempos de reacción del entrenamiento si el usuario consulta el detalle de su entrenamiento. Se muestran el valor mínimo, máximo y promedio de tiempo de reacción.
6. La aplicación web posee una navegación predictiva y fácil de realizar. Cuenta con páginas específicas para aclarar conceptos de la aplicación o contactar al desarrollador.

- Agregar modalidades de entrenamiento adicionales al tiempo de reacción, que pueda funcionar como un temporizador de contacto para ampliar su funcionalidad en el deporte.
- Implementar que la Raspberry Pi Zero W pueda funcionar como cliente y punto de acceso a la vez para que los resultados se puedan compartir a través de internet.
- Diseñar alguna otra manera de crear un entrenamiento la medición de tiempo de reacción que permita al usuario tener nuevas capacidades sobre la configuración del entrenamiento.
- Adaptar otro tipo de estímulos como lo puede ser auditivo dentro de la misma lógica de la aplicación y comparar el rendimiento del usuario con esta variante.
- Implementar código fuente con lógica para la activación de estímulos y lectura de tiempo de dispositivos de captación diseñados específicamente para esa función.

- [1] boggdan95, *Contribute to boggdan95/aplicacion-web_reaction-time development by creating an account on GitHub*, original-date: 2018-09-17T07:35:39Z, sep. de 2018. dirección: https://github.com/boggdan95/aplicacion-web_reaction-time (visitado 02-10-2018).
- [2] *MUSEUM OF THE HISTORY OF REACTION TIME RESEARCH*. dirección: http://tomperera.com/psychology_museum/mrt.htm (visitado 14-09-2018).
- [3] *The FITLIGHT Trainer | Speed Training System For Athletes*, en-US. dirección: <https://www.fitlighttraining.com/the-fitlight-trainer/> (visitado 02-10-2018).
- [4] *Definicion de aplicación web*. dirección: http://www.alegsa.com.ar/Dic/aplicacion_web.php (visitado 02-06-2018).
- [5] P. Viswanathan, *Should You Develop a Native App or a Web App?*, en. dirección: <https://www.lifewire.com/native-apps-vs-web-apps-2373133> (visitado 03-06-2018).
- [6] J. Summerfield, *Mobile Website vs. Mobile App (Application) – Which is Best for Your Organization?* Dirección: <https://www.hswsolutions.com/services/mobile-web-development/mobile-website-vs-apps/> (visitado 03-06-2018).
- [7] J. P. Blázquez, R. R. Vique, J. D. M. Pozo y M. D. Prieto, “Tecnología y desarrollo en dispositivos móviles”, es, págs. 10-20,
- [8] *IEEE 802.11, The Working Group Setting the Standards for Wireless LANs*. dirección: <http://www.ieee802.org/11/> (visitado 11-09-2018).
- [9] *Tipos de Conectividad de los Celulares Android: Red, WiFi, Bluetooth y Roaming – Lanix Colombia Atención a Usuarios*. dirección: <https://soporte.lanix.co/hc/es/articles/204015959-Tipos-de-Conectividad-de-los-Celulares-Android-Red-WiFi-Bluetooth-y-Roaming-> (visitado 07-09-2018).
- [10] *What is WiFi and How Does it Work?*, en. dirección: <https://ccm.net/faq/298-what-is-wifi-and-how-does-it-work> (visitado 07-09-2018).
- [11] *Cómo pasar archivos con NFC*, es. dirección: <https://www.androidpit.es/como-pasar-archivos-con-nfc> (visitado 07-09-2018).

- [12] *Mejores navegadores de internet gratis para Android 2018*, es. dirección: <https://www.solvetic.com/page/recopilaciones/s/recopilacion/mejores-navegadores-internet-gratis-android> (visitado 11-09-2018).
- [13] R. Rosso, *Comparativa de navegadores para Android (marzo 2017)*, es-ES, mar. de 2017. dirección: <https://blog.uptodown.com/comparativa-navegadores-android-2017/> (visitado 11-09-2018).
- [14] *ESP8266 Overview | Espressif Systems*. dirección: <https://www.espressif.com/en/products/hardware/esp8266ex/overview> (visitado 21-02-2018).
- [15] R. P. Foundation, *Raspberry Pi — Teach, Learn, and Make with Raspberry Pi*. dirección: <https://www.raspberrypi.org> (visitado 19-09-2018).
- [16] *LinkIt Smart 7688 Duo*, en. dirección: <https://www.seeedstudio.com/LinkIt-Smart-7688-Duo-p-2574.html> (visitado 04-03-2018).
- [17] *HTML*, es. dirección: <https://developer.mozilla.org/es/docs/Web/HTML> (visitado 07-09-2018).
- [18] *CSS*, es. dirección: <https://developer.mozilla.org/es/docs/Web/CSS> (visitado 07-09-2018).
- [19] *CSS básico*, es. dirección: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/CSS_basics (visitado 07-09-2018).
- [20] *¿Qué es JavaScript?*, es. dirección: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qu%C3%A9_es_JavaScript (visitado 10-09-2018).
- [21] *PHP: ¿Qué es PHP? - Manual*. dirección: <http://www.php.net/manual/es/intro-what-is.php> (visitado 10-09-2018).
- [22] *PHP: What do I need? - Manual*. dirección: <http://php.net/manual/en/tutorial-requirements.php> (visitado 10-09-2018).
- [23] *PHP: ¿Qué puede hacer PHP? - Manual*. dirección: <http://php.net/manual/es/intro-whatcando.php> (visitado 04-03-2018).
- [24] *Node.js Introduction*. dirección: https://www.w3schools.com/nodejs/nodejs_intro.asp (visitado 10-09-2018).
- [25] F. d. Node.js, *Acerca*, es. dirección: <https://nodejs.org/es/about/> (visitado 10-09-2018).
- [26] *Beginner's Guide to Node.js (Server-side JavaScript)*, en-US, nov. de 2011. dirección: <https://www.hongkiat.com/blog/node-js-server-side-javascript/> (visitado 10-09-2018).
- [27] DesarrolloWeb.com, *Qué es Python*, en. dirección: <http://www.desarrolloweb.com/articulos/1325.php> (visitado 10-09-2018).
- [28] *Definición de Python (lenguaje de programación)*. dirección: <http://www.alegsa.com.ar/Dic/python.php> (visitado 10-09-2018).
- [29] tutorialspoint.com, *MVC Framework Introduction*. dirección: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm (visitado 10-09-2018).
- [30] *Modelo vista controlador (MVC)*, es. dirección: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html> (visitado 10-09-2018).

- [31] *Django overview | Django*. dirección: <https://www.djangoproject.com/start/overview/> (visitado 27-05-2018).
- [32] *Introducción a Django*, es. dirección: <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introducci%C3%B3n> (visitado 10-09-2018).
- [33] *CogniFit*, es. dirección: <https://www.cognifit.com/es/habilidad-cognitiva/tiempo-de-respuesta> (visitado 10-09-2018).
- [34] G. Cometti, *El entrenamiento de la velocidad*. Editorial Paidotribo, 2007, vol. 24.
- [35] Javier, *Cálculo de tu tiempo de reacción – Instinto Lógico*, es-ES. dirección: <http://instintologico.com/calculo-de-tu-tiempo-de-reaccion/> (visitado 11-09-2018).
- [36] *Online Reaction Time Test*. dirección: <https://faculty.washington.edu/chudler/java/redgreen.html> (visitado 02-10-2018).
- [37] *Human Benchmark - Reaction Time Test*. dirección: <https://www.humanbenchmark.com/tests/reactiontime/> (visitado 02-10-2018).
- [38] *Socket.IO*. dirección: <https://socket.io/index.html> (visitado 01-10-2018).
- [39] *Node.js vs. PHP: Which Environment To Choose For Your Next Project?*, en. dirección: <https://www.netguru.co/blog/nodejs-vs-php> (visitado 04-03-2018).
- [40] *Django VS NodeJs - Framework Technologies Market Share Comparison*, en. dirección: <https://www.similartech.com/compare/django-vs-nodejs> (visitado 27-05-2018).
- [41] *Udacity - Free Online Classes & Nanodegrees | Udacity*. dirección: <https://www.udacity.com/> (visitado 01-10-2018).
- [42] *W3Schools Online Web Tutorials*. dirección: <https://www.w3schools.com/> (visitado 01-10-2018).
- [43] *HTML Color Picker*. dirección: https://www.w3schools.com/colors/colors_picker.asp (visitado 14-09-2018).
- [44] *A useful tool for designing css buttons*. dirección: <http://www.bestcssbuttongenerator.com/> (visitado 14-09-2018).
- [45] *Css code generator for text - CSSTXT*. dirección: <http://csstxt.com/> (visitado 14-09-2018).
- [46] *Stack Overflow - Where Developers Learn, Share, & Build Careers*. dirección: <https://stackoverflow.com/> (visitado 01-10-2018).
- [47] *Quora - Un lugar para compartir conocimiento y entender mejor el mundo*. dirección: <https://es.quora.com/> (visitado 01-10-2018).
- [48] *Writing your first Django app, part 1 | Django documentation | Django*. dirección: <https://docs.djangoproject.com/en/2.1/intro/tutorial01/> (visitado 01-10-2018).
- [49] *Home*, en-US. dirección: <https://www.anaconda.com/> (visitado 01-10-2018).
- [50] *Managing environments — Conda documentation*. dirección: <https://conda.io/docs/user-guide/tasks/manage-environments.html#creating-an-environment-with-commands> (visitado 14-09-2018).
- [51] *Visual Studio Code - Code Editing. Redefined*, en. dirección: <http://code.visualstudio.com/> (visitado 14-09-2018).

- [52] *A hackable text editor for the 21st Century*, en. dirección: <https://atom.io/> (visitado 15-09-2018).
- [53] *What is GitHub? - Definition from WhatIs.com*, en. dirección: <https://searchitoperations.techtarget.com/definition/GitHub> (visitado 15-09-2018).
- [54] *Plantillas de Django · Django Girls Tutorial*. dirección: https://tutorial.djangogirls.org/es/django_templates/ (visitado 15-09-2018).
- [55] *onoff*. dirección: <https://www.npmjs.com/package/onoff> (visitado 26-04-2018).
- [56] *Date*, es. dirección: https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/Date (visitado 16-09-2018).
- [57] *Setting up a Raspberry Pi as an access point in a standalone network (NAT) - Raspberry Pi Documentation*. dirección: <https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md> (visitado 01-10-2018).

14.1. Enlaces

Aplicación web orientada al tiempo de reacción

Repositorio que integra Django y Node.js como una aplicación final.

https://github.com/boggdan95/aplicacion-web_reaction-time.git

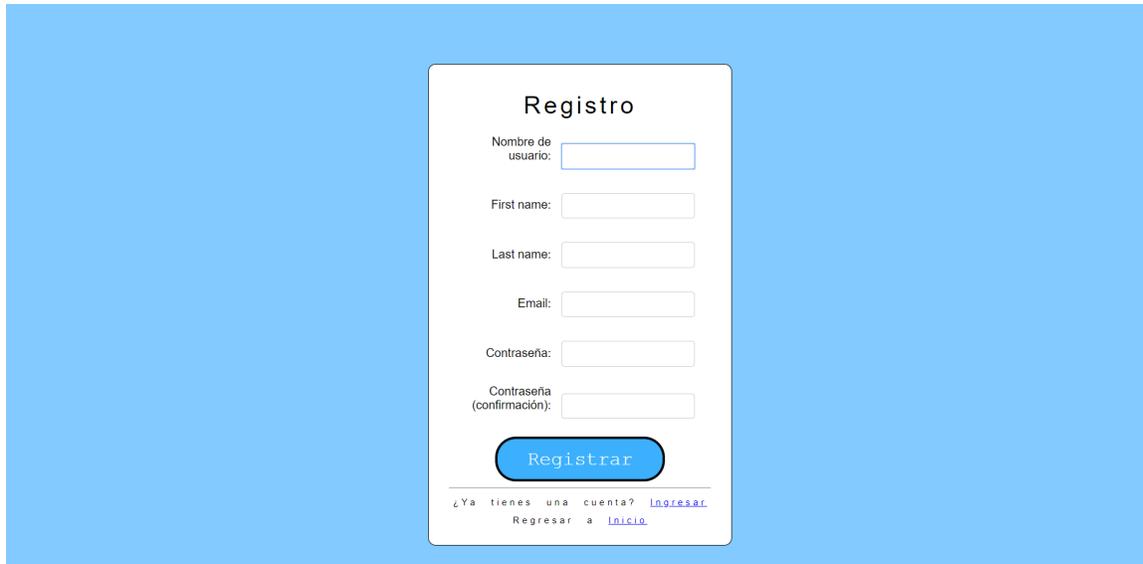
Interfaz gráfica aplicación web - Django

https://github.com/boggdan95/raspberry_entrenamiento.git

Ejercicios de entrenamiento - Node.js

https://github.com/boggdan95/quick_trainer.git

14.2. Plantillas HTML



Registro

Nombre de usuario:

First name:

Last name:

Email:

Contraseña:

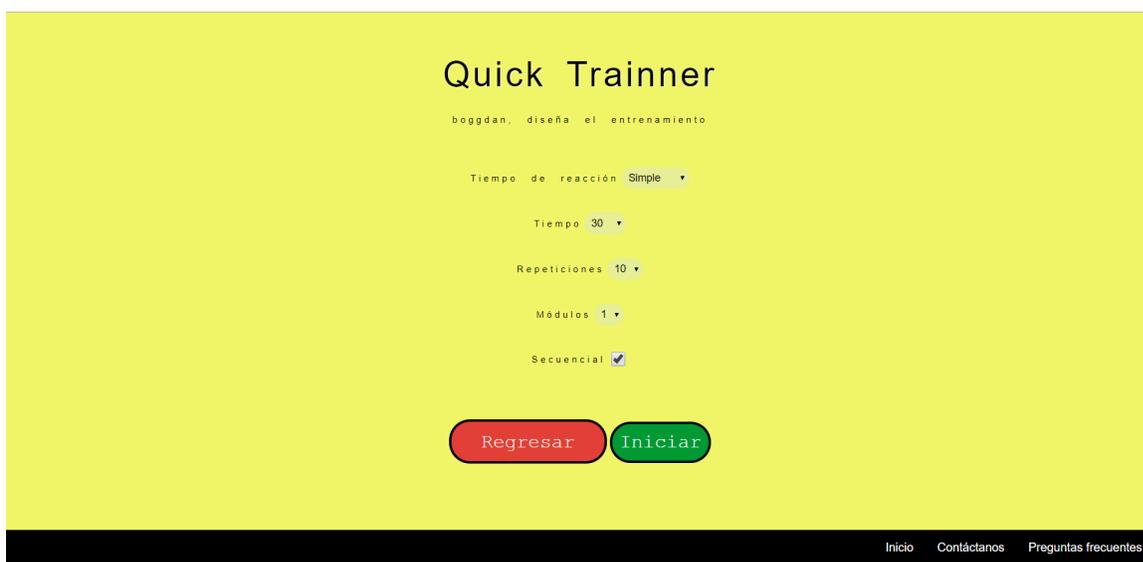
Contraseña (confirmación):

[Registrar](#)

¿Ya tienes una cuenta? [Ingresar.](#)

Regresar a [Inicio](#)

Figura 25: Plantilla de registro de usuario



Quick Trainer

boggdan, diseña el entrenamiento

Tiempo de reacción Simple ▾

Tiempo 30 ▾

Repeticiones 10 ▾

Módulos 1 ▾

Secuencial

[Regresar](#) [Iniciar](#)

[Inicio](#) [Contáctanos](#) [Preguntas frecuentes](#)

Figura 26: Plantilla de configuración de entrenamiento

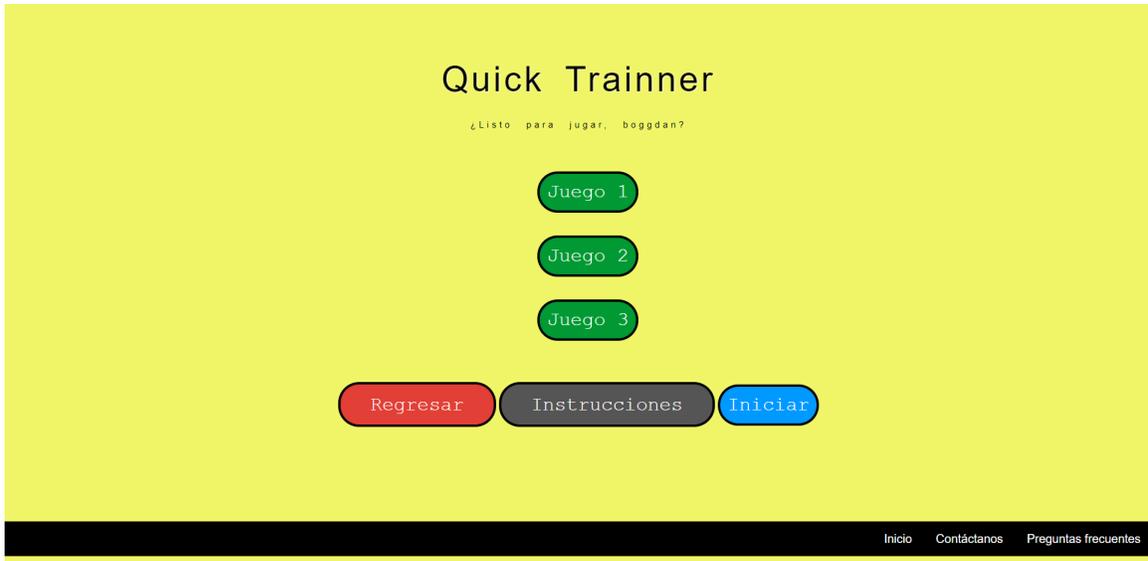


Figura 27: Plantilla de selección de juego

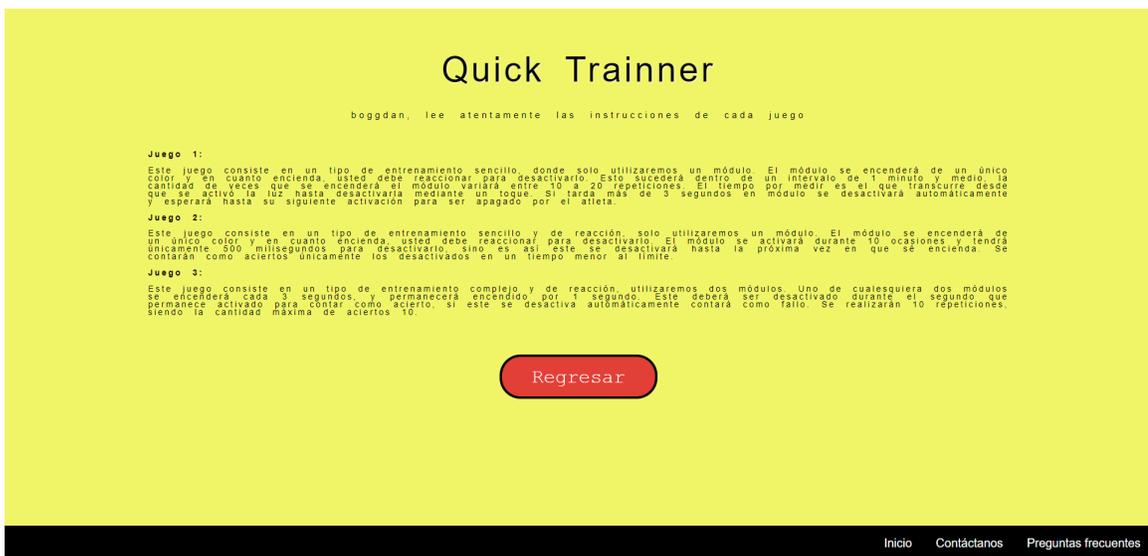


Figura 28: Plantilla de instrucciones de juego

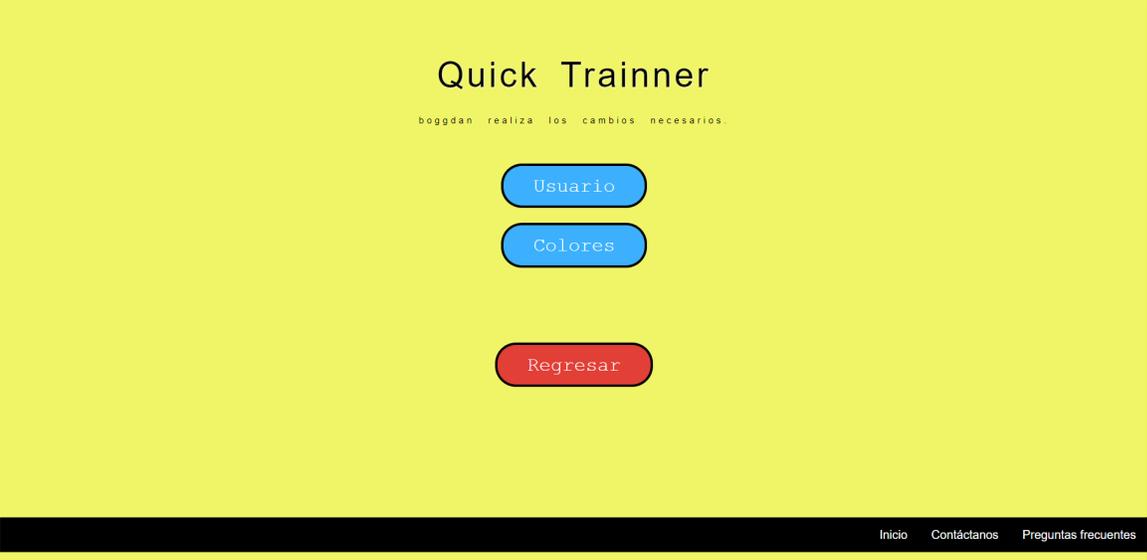


Figura 29: Plantilla de configuración

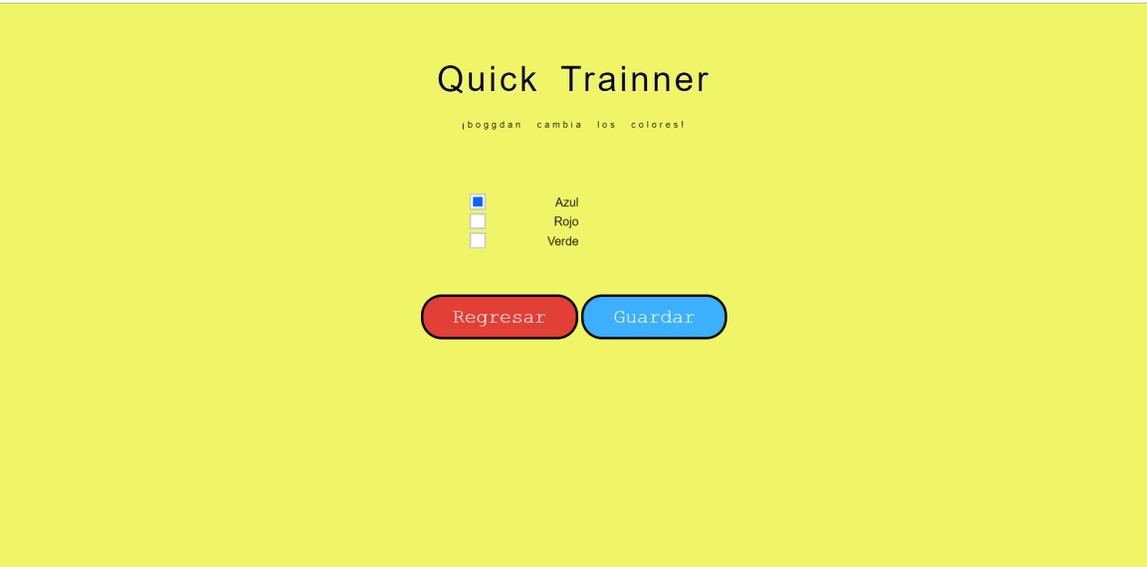


Figura 30: Plantilla de configuración de color de estímulo

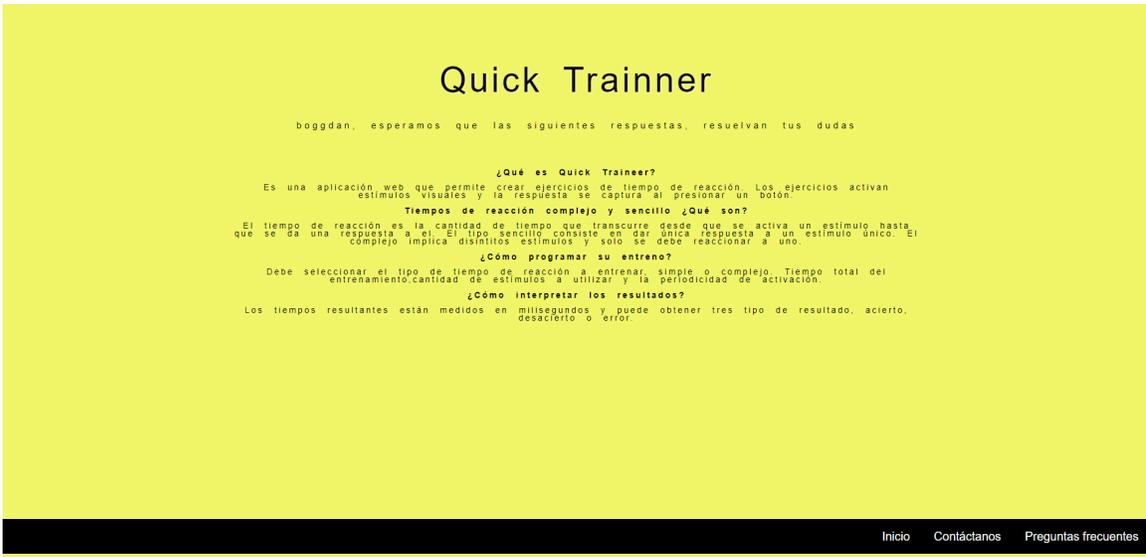


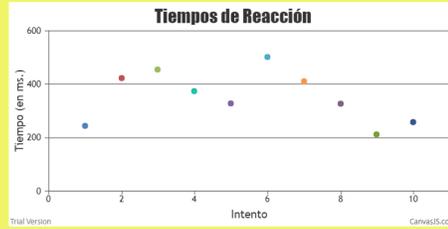
Figura 31: Plantilla de preguntas frecuentes



Figura 32: Plantilla de resultados

Quick Trainer

bogdan, estos son los resultados de tu entrenamiento



[Página de Inicio](#)

Figura 33: Plantilla de resultados detallados

14.3. Entrenamientos predeterminados

14.3.1. Instrucciones

Juego 1

Este juego consiste en un tipo de entrenamiento sencillo, donde solo utilizaremos un módulo. El módulo se encenderá de un **único color** y en cuanto encienda, usted debe reaccionar para desactivarlo. Esto sucederá dentro de un intervalo de **1 minuto y medio**, la cantidad de veces que se encenderá el módulo variará entre **10 a 20** repeticiones. El tiempo por medir es el que transcurre desde que se activó la luz hasta desactivarla mediante un toque. Si tarda más de 3 segundos en módulo se desactivará automáticamente y esperará hasta su siguiente activación para ser apagado por el atleta.

Juego 2

Este juego consiste en un tipo de entrenamiento sencillo y de reacción, solo utilizaremos un módulo. El módulo se encenderá de un **único color** y en cuanto encienda, usted debe reaccionar para desactivarlo. El módulo se activará durante **10 ocasiones** y tendrá únicamente **500 milisegundos** para desactivarlo, sino es así este se desactivará hasta la próxima vez en sé encienda. Se contarán como aciertos únicamente los desactivados en un tiempo menor al límite.

Juego 3

Este juego consiste en un tipo de entrenamiento complejo y de reacción, utilizaremos dos módulos. Uno de cualesquiera dos módulos se encenderá cada **3 segundos**, y permanecerá encendido por **1 segundo**. Este deberá ser desactivado durante el segundo que permanece activado para contar como acierto, si este se desactiva automáticamente contará como fallo. Se realizarán 10 repeticiones, siendo la cantidad máxima de aciertos 10.

14.3.2. Diagramas de flujo

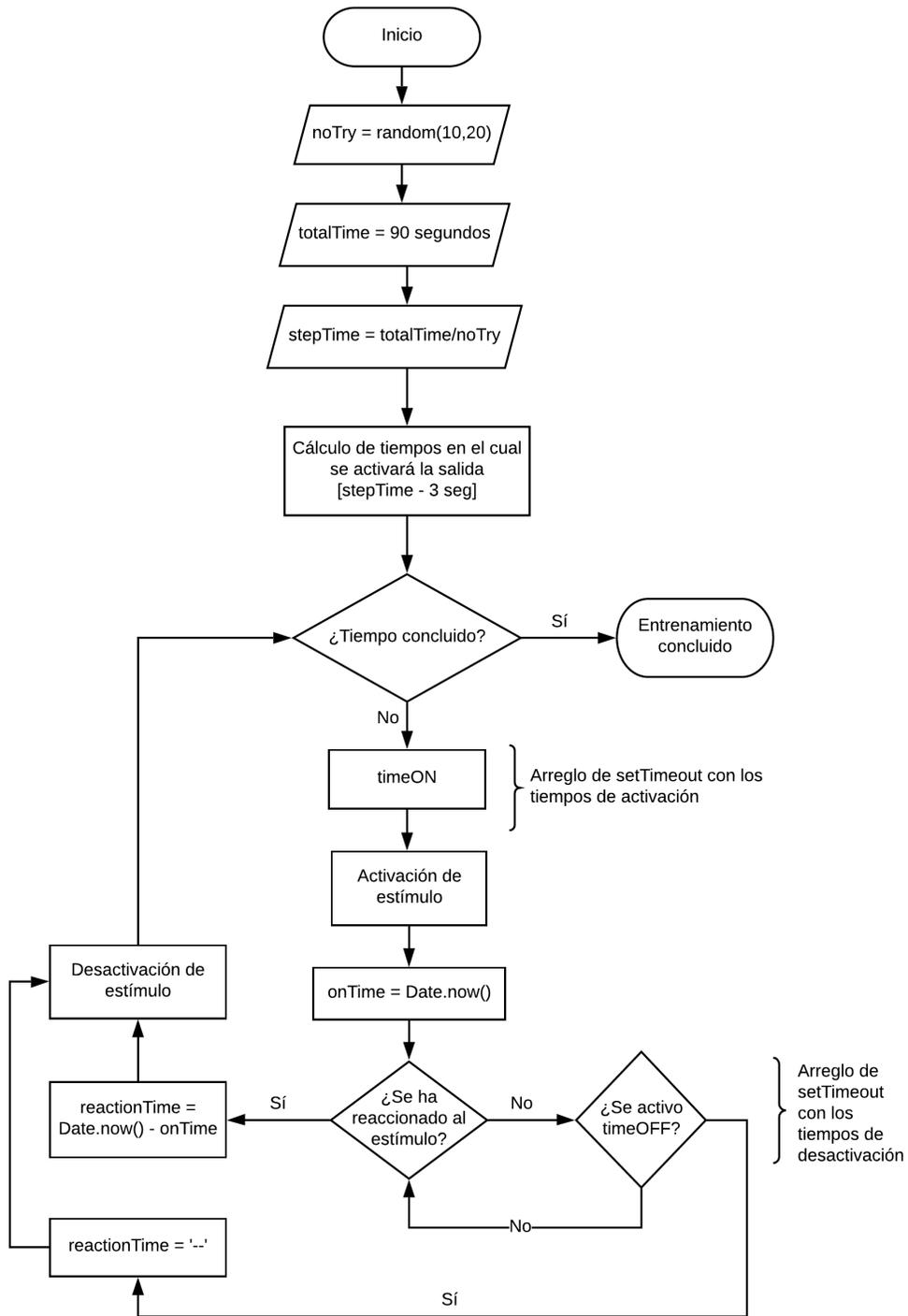


Figura 34: Diagrama de flujo:Juego 1

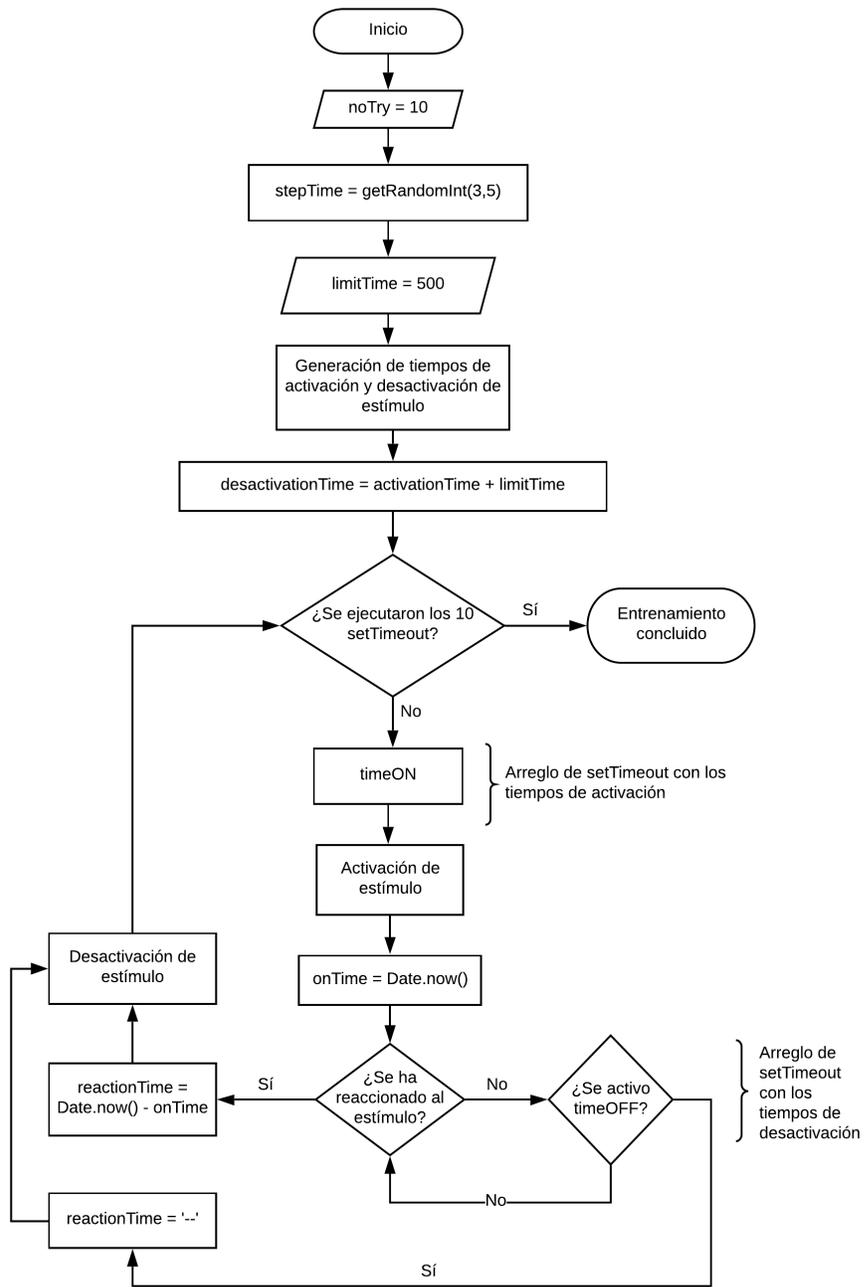


Figura 35: Diagrama de flujo:Juego 2

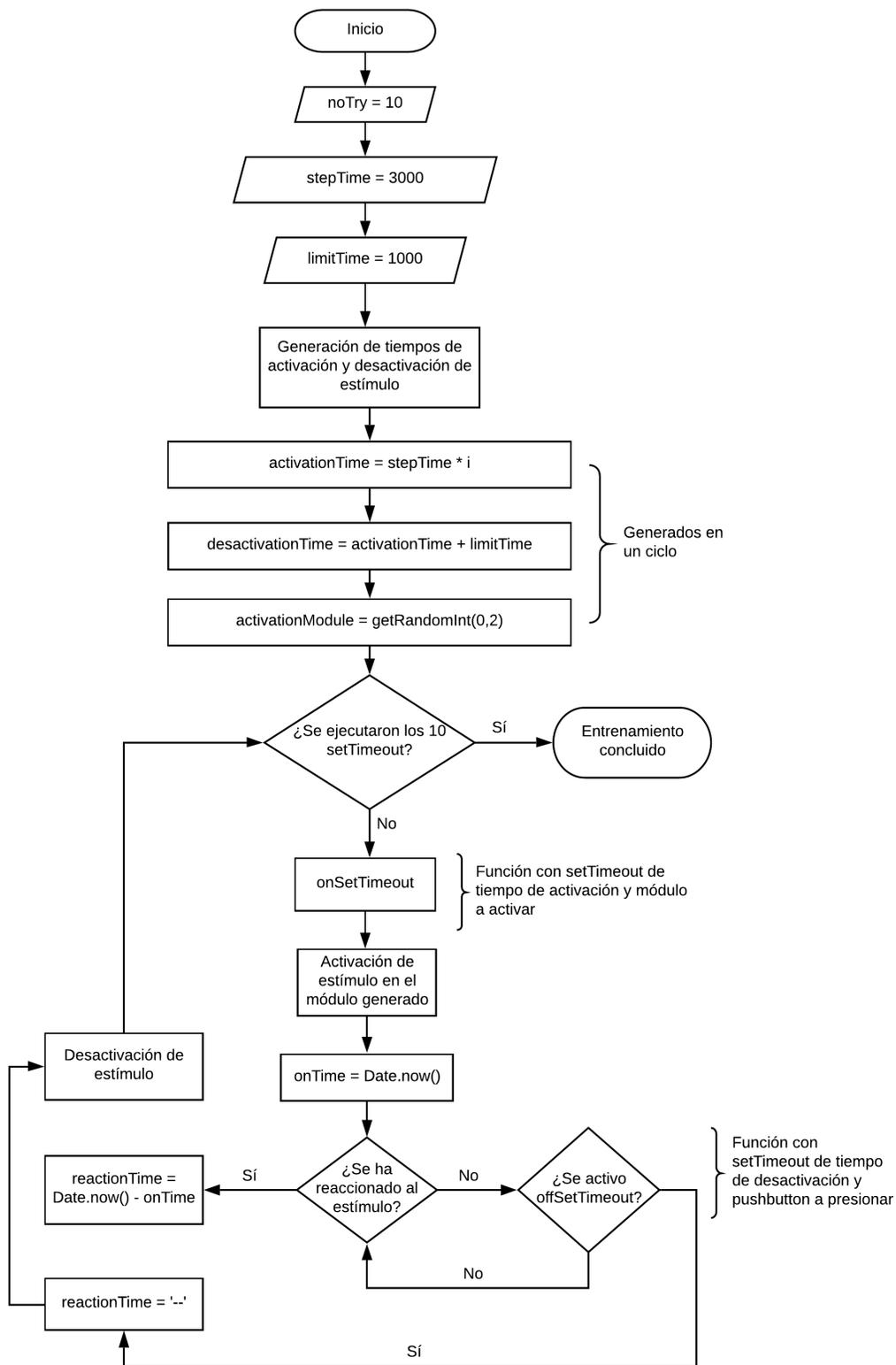


Figura 36: Diagrama de flujo: Juego 3

14.4. Circuito de prueba

14.4.1. Esquemático

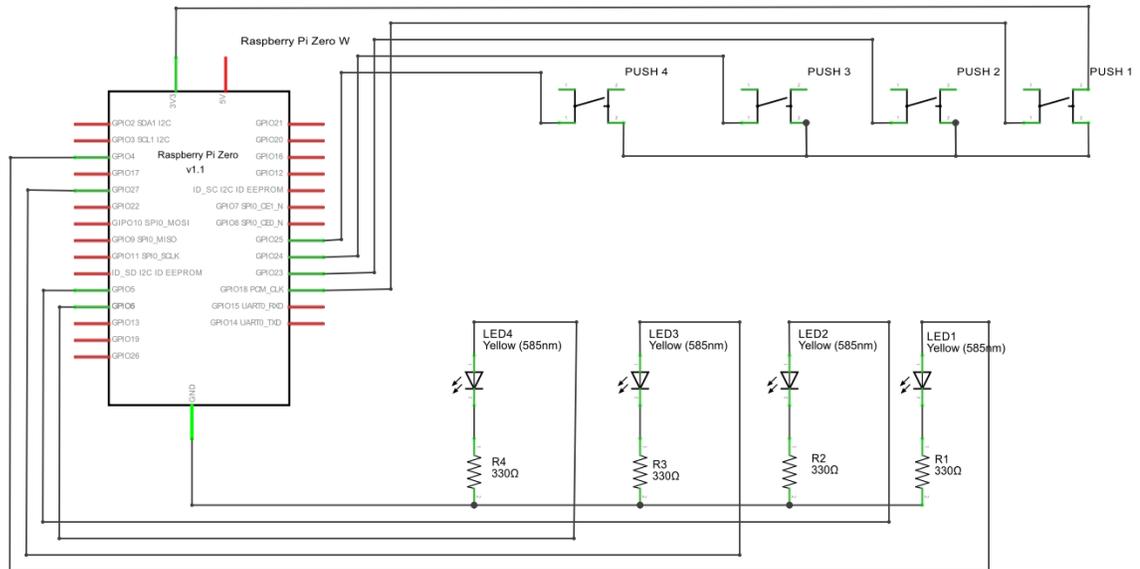


Figura 37: Esquemático de conexión: Raspberry Pi Zero W, Leds, Pushbuttons

14.4.2. Diagrama

Ilustración de conexiones entre Raspberry Pi Zero W, LED's y Pushbuttons.

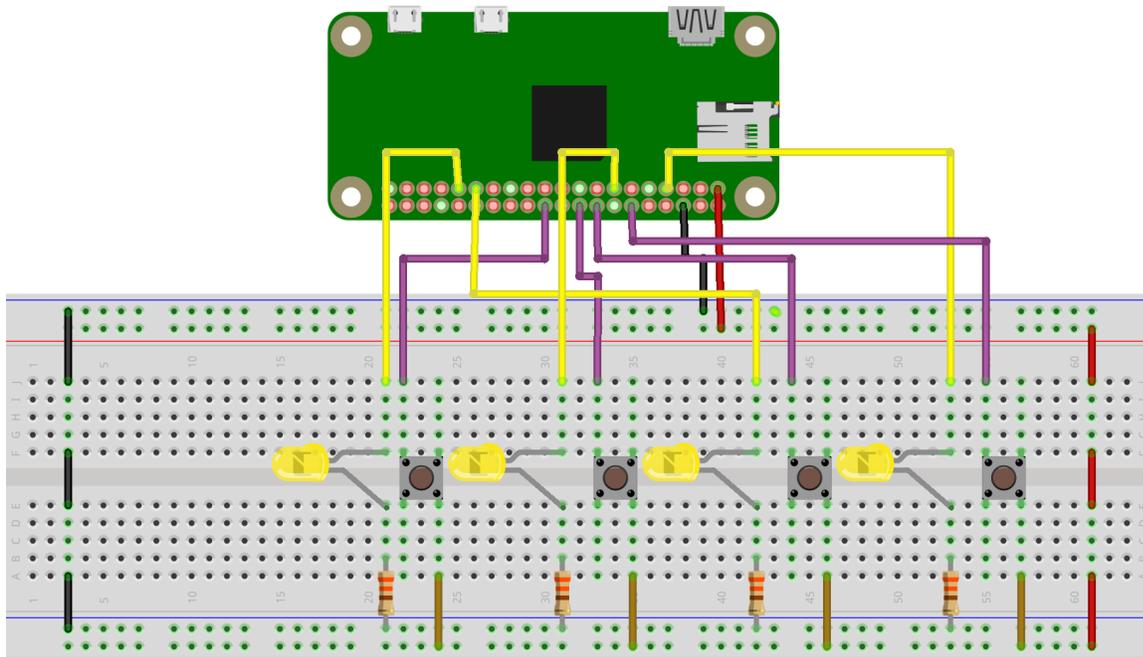


Figura 38: Vista del esquemático y sus componentes

14.4.3. Fotografía

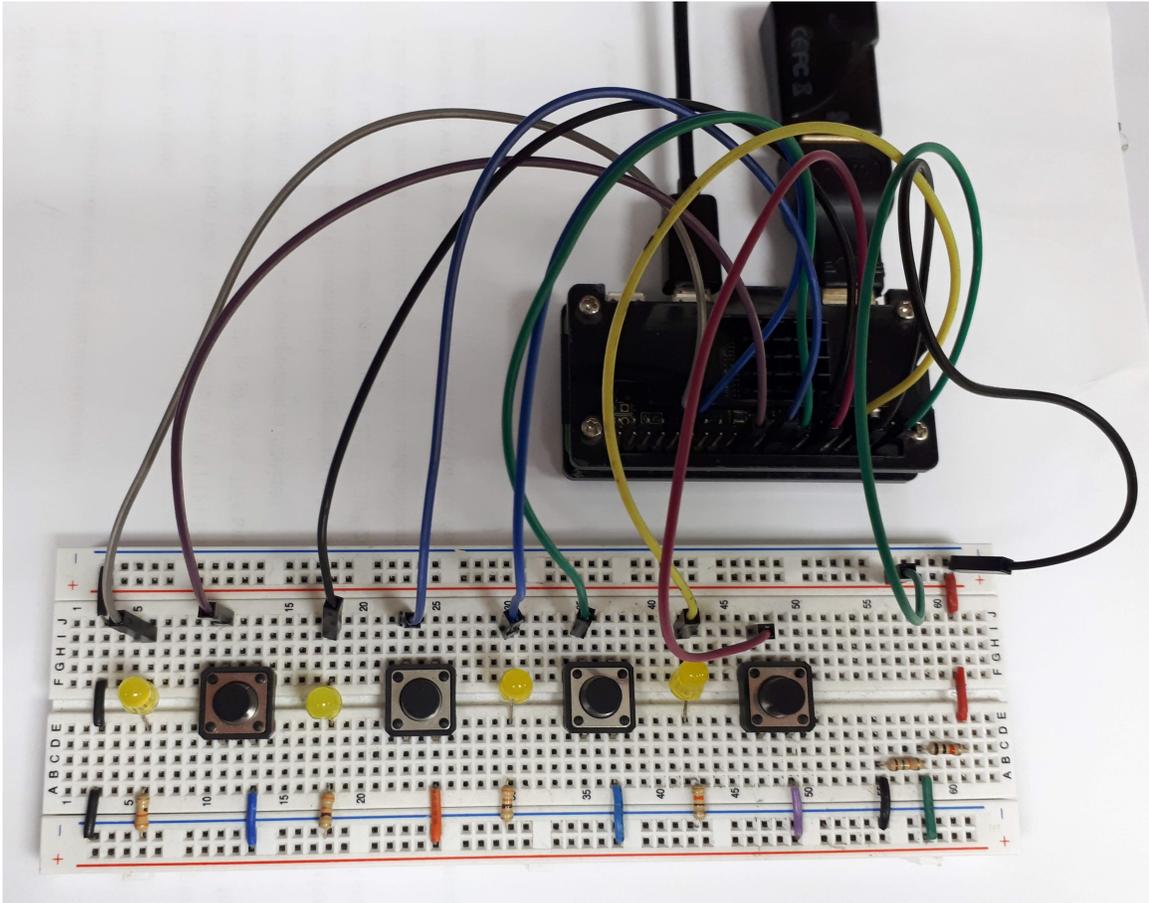


Figura 39: Foto del circuito utilizado

bug es un error o un defecto en el software o hardware que hace que un programa funcione incorrectamente. 44

POST El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor. 52

SSH SSH (Secure Shell o intérprete de órdenes seguro) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. 64

URL se trata de una serie de caracteres que responden a un formato estándar y que permiten clasificar recursos en páginas web. 36, 49

V8 es el motor de código abierto JavaScript de alto rendimiento de Google, escrito en C ++. Se usa en Google Chrome, el navegador de código abierto de Google y en Node.js, entre otros. Implementa ECMAScript como se especifica en ECMA-262, y se ejecuta en Windows 7 o posterior, macOS 10.12+ y sistemas Linux que usan procesadores IA-32, ARM o MIPS. V8 puede ejecutarse de forma independiente o puede integrarse en cualquier aplicación de C ++ . 28