

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA



TRABAJO DE GRADO PREVIO LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN MECATRÓNICA

TITLE:

“COLLECTION SYSTEM OF ELECTRONIC VOTES THROUGH
NATURAL USER INTERFACE”

TEMA:

“SISTEMA DE RECOLECCIÓN DE VOTOS ELECTRÓNICOS A
TRAVÉS DE INTERFAZ NATURAL DE USUARIO”

AUTOR:

JORGE DAVID SÁNCHEZ CUAICAL

DIRECTOR:

CARLOS XAVIER ROSERO CHANDI

IBARRA, Julio de 2019



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA
AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DEL AUTOR			
CEDULA DE IDENTIDAD:	1003927553		
APELLIDOS Y NOMBRES:	SÁNCHEZ CUAICAL JORGE DAVID		
DIRECCIÓN:	Los Incas y Lucila Benalcázar -Ibarra		
EMAIL:	jdsanchezc@utn.edu.ec – deyvid669@gmail.com		
TELÉFONO FIJO:	062510769	TELÉFONO MÓVIL:	0988277583
DATOS DE LA OBRA			
TÍTULO:	“SISTEMA DE RECOLECCIÓN DE VOTOS ELECTRÓNICOS A TRAVÉS DE INTERFAZ NATURAL DE USUARIO”		
AUTOR:	JORGE DAVID SÁNCHEZ CUAICAL		
FECHA (AAAA-MM-DD):	2019-07-18		
SÓLO PARA TRABAJOS DE GRADO			
PROGRAMA:	PREGRADO		
TÍTULO POR EL QUE OPTA:	INGENIERO EN MECATRÓNICA		
ASESOR/DIRECTOR:	CARLOS XAVIER ROSERO CHANDI		

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló sin violar derechos de autor de terceros, por lo tanto, la obra es original, y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 18 días del mes de julio de 2019.



Jorge David Sánchez Cuical
C.I: 1003927553



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CERTIFICACIÓN

En calidad de director del trabajo de grado "SISTEMA DE RECOLECCIÓN DE VOTOS ELECTRÓNICOS A TRAVÉS DE INTERFAZ NATURAL DE USUARIO", presentado por el egresado JORGE DAVID SÁNCHEZ CUAICAL, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, a los 18 días del mes de julio de 2019.



Carlos Xavier Rosero Chandi
DIRECTOR DE TESIS

Agradecimiento

Agradezco a Dios por guiar mis pasos, darme sabiduría y fuerza para sobrellevar los momentos difíciles y llegar a estas instancias de mi vida.

Agradezco a mis padres por el apoyo, el amor y el esfuerzo que han hecho para que pueda culminar mi formación profesional.

A mis hermanos por ser fuente de motivación que me han ayudado a no rendirme y lograr culminar mi carrera universitaria.

A mi director de trabajo de grado Xavier Rosero por brindarme toda su sabiduría, experiencia y guiarme en el camino para culminar este proceso.

A mis compañeros de equipo que han dado todo su esfuerzo proyecto tras proyecto para que juntos podamos alcanzar el objetivo en común que es obtener el título de Ingeniero en Mecatrónica.

David Sánchez

Dedicatoria

Dedico este trabajo a mis padres y a mis hermanos porque fueron los responsables de que este sueño se vuelva realidad, día tras día me brindaron su apoyo incondicional y fueron mi motivación para culminar mi formación profesional.

David Sánchez

Resumen

Las encuestas son una herramienta muy utilizada para obtener de forma rápida la opinión de las personas respecto a un tema específico; están conformadas de preguntas y respuestas y la opinión dada por las personas se define como voto. El presente trabajo ofrece un sistema de recolección de votos que obtiene la información de forma interactiva y rápida, propiciando que las personas disfruten de colaborar con la realización de encuestas. El desarrollo de la recolección de votos electrónicos se realiza en tres etapas que son: adquisición de la información, acondicionamiento y almacenamiento de la información, presentación de datos obtenidos. El conteo de los votos es instantáneo y con un bajo margen de error. Este sistema consta de una interfaz natural de usuario que interactúa de forma directa con las personas y sus votos son receptados por sensores. Estos votos son almacenados en una base de datos para posteriormente mostrar los resultados de las encuestas en una interfaz gráfica. El sistema está construido para poder ser trasladado de un lugar a otro sin ningún problema y su funcionamiento es el mismo tanto en ambientes muy iluminados como en ambientes donde hay baja iluminación. El sistema de recolección de votos electrónicos puede ser usado y mejorado con el tiempo debido a que las encuestas seguirán teniendo un papel fundamental en la recolección de información

Abstract

Surveys are a widely used tool to quickly obtain people's opinions on a specific topic; they are made up of questions and answers and the opinion given by the people is defined as vote. The present work offers a collection system of votes that obtains information in an interactive and fast way, enabling people to enjoy collaborating with surveys. The development of electronic votes collection is carried out in three stages: information acquisition, information storage and conditioning, presentation of obtained data. The counting of the votes is instantaneous and with a low margin of error. This system consists of a natural user interface that interacts directly with people and their votes are received by sensors. These votes are stored in a database to later show the results of the surveys in a graphical interface. The system is built to be able to be moved from one place to another without any problem and its operation is the same both in very bright environments and in environments where there is low lighting. The electronic vote collection system can be used and improved over time because the surveys will continue to have a fundamental role in the collection of information.

Índice general

Índice general.....	IX
Índice de figuras.....	XII
Índice de tablas	XIV
Introducción	1
1.1 Problema.....	1
1.2 Objetivos	2
1.2.1 Objetivo general.....	2
1.2.2 Objetivos específicos	3
1.3 Justificación.....	3
1.4 Antecedentes	4
1.5 Alcance.....	4
Revisión Literaria.....	6
2.1 Internet de las cosas (IoT)	6
2.2 Sitio o portal web	7
2.3 Funcionalidades de un sitio o portal web.....	7
2.3.1 Información en tiempo real.....	7
2.3.2 Clasificación automática de los contenidos	7
2.4 Base de datos	8
2.4.1 Características de la base de datos	8
2.4.2 Tipos de base de datos	9
2.5 Interfaces naturales de usuario	9
Metodología	10

3.1	Descripción general del sistema.....	10
3.2	Requerimientos del sistema.....	10
3.3	Diagrama y descomposición funcional	11
3.4	Aproximación en bloques.....	12
3.4.1	Adquisición de información.....	12
3.4.2	Acondicionamiento de la información.....	12
3.4.3	Presentación de la información.....	13
3.5	Caracterización del hardware	13
3.5.1	Interfaz natural de usuario	13
3.5.2	Visualizador LED	15
3.5.3	Sensado	15
3.5.4	Cableado	16
3.5.5	Procesador.....	17
3.5.6	Diagrama de conexión	18
3.6	Descripción del Software	19
3.6.1	Diagrama de flujo de la adquisición y acondicionamiento de información.....	19
3.6.2	Diagrama de flujo de la presentación de información	20
3.6.3	Módulos de Python	21
3.6.3.1	Módulo time	22
3.6.3.2	Módulo datetime	22
3.6.3.3	Módulo sys	22
3.6.3.4	Módulo sqlite3.....	22
3.6.3.5	Módulo functools	23
3.6.3.6	Módulo Tkinter	23
3.6.3.7	Módulo RPi.GPIO.....	23
	Implementación y pruebas	24
4.2	Hardware	24
4.2.1	Implementación de la interfaz natural de usuario.....	24
4.2.2	Configuraciones adicionales	26
4.3	Software	28
4.3.1	Implementación de la adquisición y acondicionamiento de la información.....	28

4.3.1.1	Configuraciones adicionales	29
4.3.2	Implementación de la presentación de la información	29
4.3.2.1	Configuraciones adicionales	30
4.4	Pruebas de funcionamiento	31
4.4.1	Prueba de los sensores a diferentes condiciones de luz	31
4.4.2	Prueba de los sensores a diferente velocidad del paso de los usuarios	33
4.4.3	Prueba de alcance de los sensores.....	33
4.4.4	Prueba de los sensores al paso de las personas	33
4.4.5	Prueba de visualización del letrero led a diferentes condiciones de luz	33
4.4.6	Prueba de visualización del letrero led a diferentes distancias	35
4.5	Análisis de resultados.....	37
Conclusiones y trabajo futuro		40
5.1	Conclusiones	40
5.2	Trabajo futuro	41
Bibliografía		42
Apéndice		45
A.	Método de criterios ponderados	45
A.1	Matriz morfológica.....	45
A.2	Método de criterios ponderados	46
A.2.1	Tablas de ponderación de criterios.....	46
A.2.2	Tablas de conclusión del método de criterios ponderados	51
B.	Software.....	53
B1.	Adquisición y acondicionamiento de la información	53
B2.	Representación de la información.....	56

Índice de figuras

Figura 2.1 Tipo de base datos.	9
Figura 3.1 Descomposición funcional.....	11
Figura 3.2 Diagrama de bloques del sistema	12
Figura 3.3 Soportes de Aluminio	14
Figura 3.4 Columnas de madera y aluminio	14
Figura 3.5 sensores de proximidad E18-D80NK	15
Figura 3.6 Visualizador LED wifi.....	15
Figura 3.7 Funcionamiento del sensor infrarrojo E18-D80NK	16
Figura 3.8 Componentes de la Raspberry Pi.....	18
Figura 3.9 Diagrama de conexión	18
Figura 3.10 Flujograma básico del sistema.....	20
Figura 3.11 Flujograma de la presentación de información.....	21
Figura 4.1 Columna de aluminio y madera(MDF).....	25
Figura 4.2 Estructura final.....	25
Figura 4.3 Base de la estructura	26
Figura 4.4 Tuercas mariposa	26
Figura 4.5 Cable de poder cubierto con la manguera corrugada.....	27
Figura 4.6 Protección de los cables de conexión de los sensores	27
Figura 4.7 Raspberry Pi 3b+	28
Figura 4.8 Interfaz de visualización del resultado de las encuestas	30
Figura 4.9 Interfaz interactiva (Encuesta actual)	30

Figura 4.10 Interfaz interactiva (Historial encuestas).....	31
Figura 4.11 Prueba de los sensores a alta luminosidad	32
Figura 4.12 Prueba de los sensores a baja luminosidad	32
Figura 4.13 Visualización del mensaje con luminosidad baja	34
Figura 4.14 Visualización del mensaje con luminosidad alta	34
Figura 4.15 Visualización del mensaje a 1 metro de distancia	35
Figura 4.16 Visualización del mensaje a 4 metros de distancia.....	36
Figura 4.17 Visualización del mensaje a 10 metros de distancia.....	36

Índice de tablas

Tabla 4.1 Resultados de la prueba de los sensores a diferentes condiciones de luz	38
Tabla 4.2 Resultados de la prueba de los sensores a diferente velocidad del paso de los usuarios	38
Tabla 4.3 Resultados de la prueba de los sensores al paso de las personas	38
Tabla 4.4 Resultados de la prueba de visualización del letrero LED a diferentes condiciones del luz	39
Tabla 4.5 Resultados de la prueba de visualización del letrero LED a diferentes distancias ..	39
Tabla A.1 Matriz morfológica.....	45
Tabla A.2 Tabla de criterios	46
Tabla A.3 Ponderación de criterios de los soportes	46
Tabla A.4 Ponderación de criterios del material de los soportes	47
Tabla A.5 Ponderación de criterios del material de las columnas	47
Tabla A.6 Ponderación de criterios del procesador	48
Tabla A.7 Ponderación de criterios de los sensores	48
Tabla A.8 Ponderación de criterios del visualizador	49
Tabla A.9 Ponderación de criterios de los cables de conexión	49
Tabla A.10 Ponderación de criterios de la base de datos	50
Tabla A.11 Ponderación de criterios del portal web	50
Tabla A.12 Selección de los soportes	51
Tabla A.13 Selección del material de los soportes	51
Tabla A.14 Selección del material de las columnas.....	51

Tabla A.15 Selección del procesador	52
Tabla A.16 Selección de los sensores	52
Tabla A.17 Selección del visualizador.....	52
Tabla A.18 Selección de los cables de conexión	52
Tabla A.19 Selección de la base de datos	52
Tabla A.20 Tabla de selección del portal web	53

Capítulo 1

Introducción

Este trabajo de grado ha sido realizado con el Grupo de Investigación en Sistemas Inteligentes de la Universidad Técnica del Norte (GISI-UTN).

1.1 Problema

El voto tradicional donde se utilizan boletas de papel puede presentar errores de cómputo ya que los votos son contados por humanos. Debido a esto, si se lleva un recuento total existe una probabilidad alta de que se encuentren diferencias entre el conteo original y el recuento. El voto tradicional además posee inconvenientes tales como baja motivación de las personas para opinar, el tiempo invertido en responder una encuesta y el desperdicio de materiales no renovables como el papel [1].

El uso del correo postal como medio de votación precisa moderada aceptación a pesar de que las comunicaciones a través de la red están reduciendo al mínimo su tradicional uso. Existe menor tasa de colaboración de la gente a la encuesta a través de Internet en comparación con la encuesta postal, algo que ha sido constatado en numerosas investigaciones [2].

El desarrollo de nuevas metodologías deja obsoletas a las formas tradicionales de recolección de votos. Sin embargo, las nuevas metodologías no cumplen con las características esenciales que

poseen los métodos tradicionales, por lo que es necesario un sistema que integre la nueva tecnología con los métodos tradicionales.

Las encuestas seguirán teniendo un papel fundamental en la recolección de información. Si no se cambia la modalidad de realizar las encuestas continuará existiendo un alto margen de error en el conteo de votos, desperdicio de papel y baja tasa de colaboración de las personas en la encuesta online, que tiene como consecuencia votos alejados de la realidad

Por otro lado, cuando se realizan eventos científicos o tecnológicos es necesario conocer acerca de la experiencia que tuvo el usuario; esta información puede ser tomada en cuenta para la realización de eventos posteriores. Así, se requiere un sistema para obtener las opiniones de las personas de una forma rápida a través de una votación electrónica.

El sistema de recolección de votos electrónicos a través de interfaz natural de usuario brindará una forma diferente de realizar una encuesta. Las personas se verán interesadas en utilizar el sistema debido a su interactividad. Se responderán las preguntas planteadas de forma rápida y divertida sin uso de materiales no renovables. Además, el conteo de votos será inmediato con el mínimo margen de error y los usuarios podrán visualizar los resultados de forma online en una página web.

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar un sistema para recolección de votos electrónicos utilizando una interfaz natural de usuario.

1.2.2 Objetivos específicos

- Determinar la funcionalidad del sistema considerando interacción humano-computador a nivel de adquisición de datos y a nivel de visualización de resultados.
- Desarrollar el sistema de recolección de votos electrónicos considerando hardware, software y estructura física.
- Validar el método propuesto a través de pruebas de campo.

1.3 Justificación

El voto electrónico se puede definir como una forma de votación basada en medios electrónicos que se diferencia del método tradicional por la utilización de componentes de hardware y software que permiten automatizar los procesos de emisión del voto, conteo (escrutinio), emisión de reportes de resultados, entre otros; así como de una red de comunicaciones para la transmisión y presentación de resultados [3].

La interacción hombre-máquina es un tema evaluado desde el inicio del desarrollo de la tecnología, motivo de estudio que pretende facilitar el uso de los dispositivos y mejorar la comunicación que se da con ellos. El desarrollo y la integración de diferentes tecnologías han permitido que se genere un cambio en la interacción con estos dispositivos creando una interacción diferente [4].

La forma tradicional de realizar votaciones es a través de encuestas impresas en hojas de papel, lo cual ocasiona gasto de dinero, pero principalmente provoca un impacto negativo en la naturaleza. Por el contrario, al realizar una recolección de votos electrónicos a través de una

interfaz natural de usuario se disminuye radicalmente el impacto ambiental ocasionado por el desperdicio de hojas de papel.

1.4 Antecedentes

El punto más importante para el desarrollo del presente proyecto está vinculado a internet de las cosas (internet of things, IoT). Existen varios proyectos planteados como es el caso del desarrollo de una herramienta que proporciona una interfaz de comunicación entre humanos y dispositivos en internet de las cosas (IoT) mediante diálogo en lenguaje natural escrito a través de servicios de mensajería instantánea [5]. El desarrollo e implementación de una aplicación para teléfonos inteligentes que supervisa y controla un proceso industrial a través del sistema embebido Raspberry Pi, conectado a internet mediante el protocolo de intérprete de órdenes seguras se encuentra en [6]. La arquitectura cliente-servidor para el control vía web de la iluminación de una casa habitación a través de una aplicación para dispositivos móviles con sistema operativo Android y el empleo de una tarjeta de desarrollo Raspberry-Pi se puede observar en [7]. Otro proyecto se encuentra enfocado al desarrollo de una aplicación para IoT que realiza la comunicación entre la plataforma Raspberry Pi y Arduino para obtener y enviar información a una base de datos que está almacenada en la Raspberry Pi [8]. El diseño de un sistema domótico mediante una aplicación Android instalada en un dispositivo móvil, conjuntamente con una base de datos en una tarjeta Raspberry Pi se puede observar en [9]. También se ha creado una aplicación para Android para ayudar a controlar la humedad del suelo [10]. Esta aplicación obtiene datos del sensor que está conectado a la Raspberry Pi 3. Estos datos se envían al servidor y se muestran en una aplicación de Android.

1.5 Alcance

El presente proyecto constará de una estructura semejante a dos portales donde se colocará un visualizador LED que proyectará cada pregunta con sus respectivas opciones de respuesta. Además, se colocará un sensor de presencia en cada umbral para detectar a las personas que los

atraviesen. Se elaborará una página web con ayuda de una base de datos y un microcomputador. En dicha página se observará detalladamente las preguntas y respuestas de la encuesta realizada además el número de votos electrónicos obtenidos.

Capítulo 2

Revisión Literaria

Para la realización del sistema de recolección de votos electrónicos se tiene como puntos fundamentales la adquisición, almacenamiento y presentación de la información en este caso los votos de las personas. En este capítulo se describen conceptos referentes al funcionamiento de una interfaz natural de usuario, la interacción con una base de datos y el uso de un portal web para mostrar la información recolectada, todo esto unido a través del internet de las cosas.

2.1 Internet de las cosas (IoT)

Internet de las cosas, revolución tecnológica que posibilita que Internet alcance el mundo real de los objetos físicos, convirtiendo objetos comunes en “cosas inteligentes” conectadas a Internet. Se está convirtiendo en una realidad capaz de generar gran cantidad de datos del mundo físico, los que, luego de ser analizados con herramientas informáticas, pueden ser útiles en nuestra toma de decisiones al contar con información valiosa en tiempo real y así mejorar nuestras actividades cotidianas de manera inmediata. Elemento clave de Internet de las Cosas para convertir toda clase de objetos comunes en una interfaz de Internet con el mundo real, es el microcontrolador, que junto a una amplia variedad de tecnologías modernas, tales como las redes de sensores y actuadores, GPS, RFID, comunicaciones inalámbricas, localización en tiempo real, entre otros, y por supuesto, Internet, formen esta enorme red, con el objetivo de que todas las cosas conectadas con esta red puedan servir dentro de un sistema automatizado

para, por ejemplo, la identificación de objetos en tiempo real, la localización, seguimiento, monitoreo y activación de eventos de diversa índole, incluso yendo hacia lo que se ha denominado como Planeta Inteligente donde la inteligencia se apodera de cada aspecto de nuestra vida [11].

2.2 Sitio o portal web

Un Sitio o Portal Web es una aplicación que, por medio de una interfaz simple, provee a sus usuarios (consumidores de datos) acceso a diferentes fuentes de datos o proveedores. Para el caso de este trabajo, en particular, la aplicación web proveerá a los usuarios información correspondiente a los resultados de las encuestas realizadas.

El sitio o portal web, no es otra cosa que una puerta de entrada de datos, en una interfaz capaz de modificar la información según los requerimientos de un determinado usuario, en el caso de este proyecto, el portal web buscará actualizar la información que recibe de la interfaz y mostrarla de forma gráfica y ordenada.

2.3 Funcionalidades de un sitio o portal web

2.3.1 Información en tiempo real

Un sitio o aplicación web, deberá permitir al usuario requerir de una determinada información externa, antes publicada, permitiendo que éste usuario, actualice dicha información en tiempo real, de tal forma que, esa información, se modifique de forma interna.

2.3.2 Clasificación automática de los contenidos

Hay que entender que un aplicativo web, no es un ente estático, ya que éste debe tener la capacidad de gestionar, de forma dinámica, el flujo de información, de tal forma que los directorios (de la FireBase) existentes se modifiquen en tiempo real. Así pues, la información

mostrada en la interfaz quedará actualizada.

Un sitio o portal web, deberá tener muchas más funcionalidades, como un directorio de navegación o el enlace a sitios web internos, sin embargo, para el caso de estudio de este trabajo sólo se han presentado dos funcionalidades básicas, pues el aplicativo web desarrollado es una interfaz de actualización de contenido del estado de la máquina Raspberry Pi [12].

2.4 Base de datos

Una base de datos, es un conjunto de datos relacionados entre sí, organizados y estructurados con información referente a algo. Podremos utilizar una base de datos para cosas tan sencillas como mantener un registro de nuestra agenda personal de teléfonos, o tan complicadas como llevar toda la gestión de una gran empresa u organización.

Así entonces, se puede entender una base de datos como un fondo común de información almacenada en una computadora para que cualquier persona o programa autorizado pueda acceder a ella, independientemente de su procedencia y del uso que haga.

2.4.1 Características de la base de datos

Entre las principales características de los sistemas de base de datos se puede mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

2.4.2 Tipos de base de datos

Existen dos tipos principales de base de datos: base de datos estáticas y base de datos dinámicas, en la Figura 2.1 se puede observar una breve descripción de cada una de ellas.

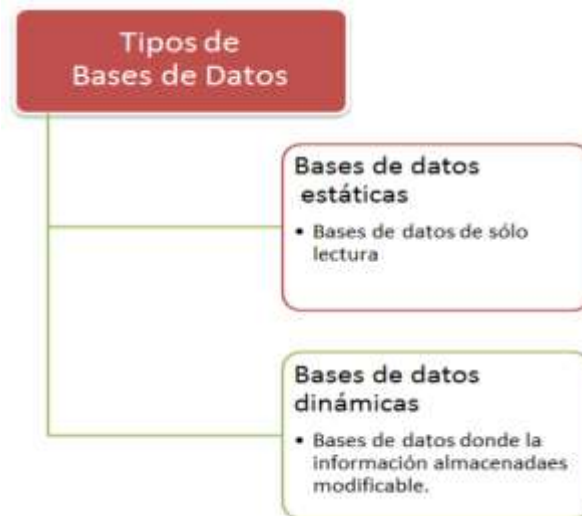


Figura 2.1 Tipo de base datos [12].

2.5 Interfaces naturales de usuario

Las interfaces naturales de usuario (NUI) son aquellas en las que se interactúa con un sistema sin utilizar sistemas de mando o dispositivos de entrada de las GUI (Interfaces de Usuario Gráficas) como sería el ratón, teclado alfanumérico, joystick y en su lugar, se hace uso de movimientos gestuales tales como las manos o el cuerpo que funcionan como el mismo mando de control, algunos ejemplos son el reconocimiento a través de sensores de voz, de mirada, de movimiento y multitáctiles [13].

Capítulo 3

Metodología

En este capítulo se realiza un análisis del funcionamiento del sistema de recolección de votos electrónicos para determinar el diseño más óptimo y posteriormente elegir los componentes que se utilizarán para construirlo.

3.1 Descripción general del sistema

El sistema implementado constará de una estructura física semejante a dos portales donde se ubicarán los sensores para adquirir la información de las personas que atraviesan los portales además poseerá un letrero led para la visualización de las preguntas y respuestas de las encuestas por último se implementa un microcomputador en este caso una Raspberry Pi 3 que será útil para el acondicionamiento, almacenamiento y presentación de la información.

3.2 Requerimientos del sistema

Los requerimientos del sistema de recolección de votos electrónicos se especifican teniendo en cuenta las necesidades y considerando los factores que van a influenciar sobre el sistema para obtener un funcionamiento óptimo, detallados a continuación:

- Todos los componentes del sistema deben tener un bajo costo.
- Los elementos que componen el sistema deben ser accesibles en el mercado nacional.

- El peso de la estructura del sistema debe ser bajo para facilitar su movilidad.
- El sistema debe tener un diseño con el cual sea fácil su ensamblaje.

3.3 Diagrama y descomposición funcional

El diagrama funcional ayuda a tener una comprensión más clara del funcionamiento del sistema de recolección de votos electrónicos, el sistema se descompone en varias subfunciones para así poder trabajar con una subfunción a la vez y de esta forma facilitar el diseño y posteriormente construir todo el sistema planteado. Luego de haber realizado el diagrama funcional se procede con la descomposición funcional que consiste en colocar los diferentes módulos que conforman cada subfunción para así tener al sistema de recolección de votos electrónicos de forma desglosada y clara como se muestra en la Figura 3.1.

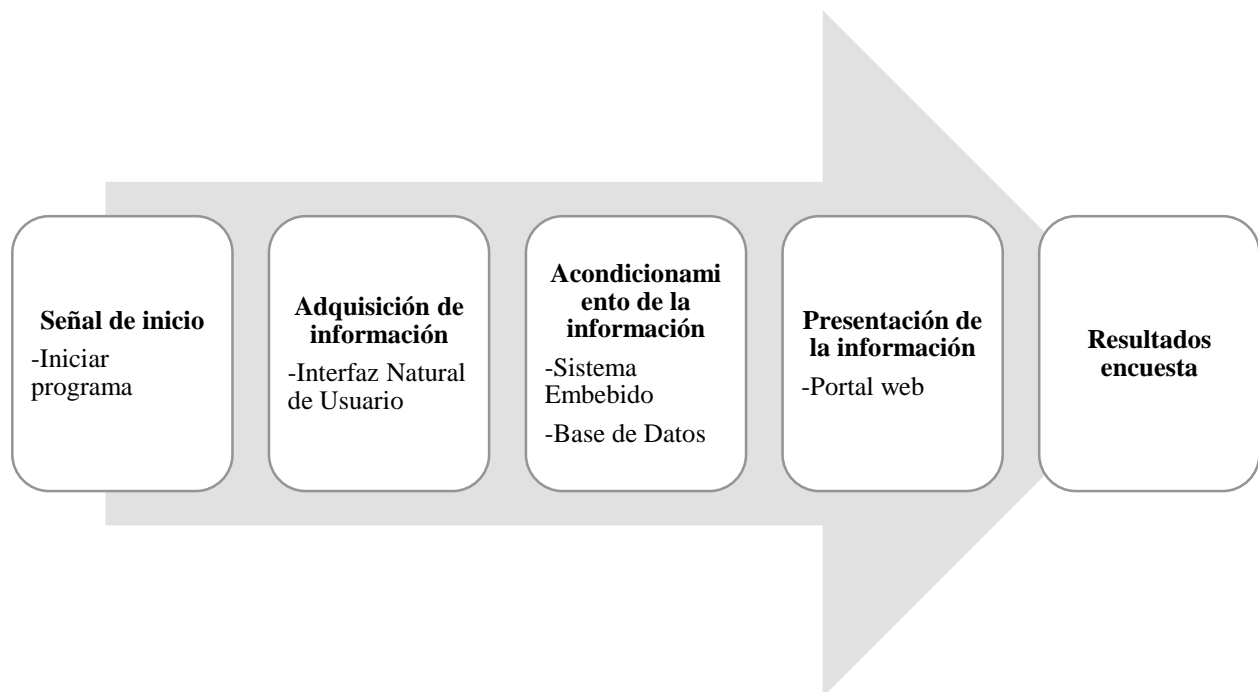


Figura 3.1 Descomposición funcional

3.4 Aproximación en bloques

La concepción básica del sistema de recolección de votos electrónicos a través de interfaz natural de usuario se comprende a través de subsistemas con funciones específicas y conformadas por bloques de acuerdo a sus funciones como se indica en la Figura 3.2.

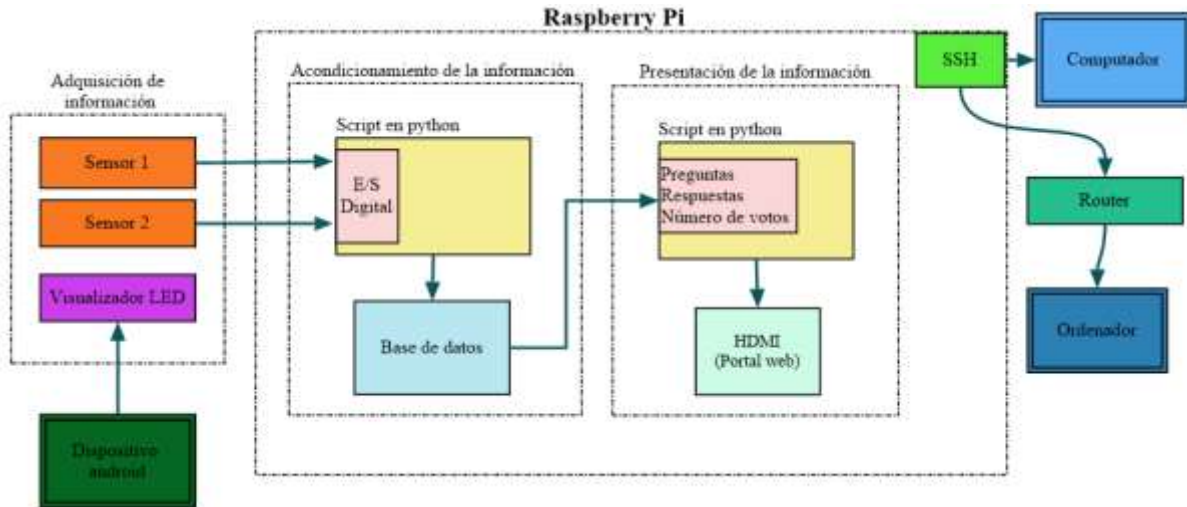


Figura 3.2 Diagrama de bloques del sistema

3.4.1 Adquisición de información

Permite adquirir en tiempo real las opiniones de las personas referentes a las preguntas y respuestas planteadas en la encuesta de una forma rápida e interactiva con el usuario.

3.4.2 Acondicionamiento de la información

Se basa fundamentalmente en un sistema embebido que recibe la información adquirida la ordena y la almacena de la forma más conveniente para posteriormente guardarla en una base de datos donde se pueda interpretar la información de forma fácil y rápida.

3.4.3 Presentación de la información

Su funcionamiento principal consiste en conectarse e ingresar en la base de datos para extraer la información almacenada para posteriormente mostrarla en un portal web, los usuarios podrán acceder para observar el progreso de la encuesta en tiempo real

3.5 Caracterización del hardware

Luego de haber realizado todo el análisis del funcionamiento del sistema de recolección de votos electrónicos se aplica el método de criterios ponderados para seleccionar los elementos más idóneos para la realización del sistema. El método de criterios ponderados se encuentra en el Apéndice A.1. Como resultado del método se obtiene la siguiente conclusión: para iniciar el funcionamiento del sistema se realizará a través de la consola del compilador de Python, la interfaz natural de usuario estará construida por soportes cortos de aluminio y columnas de madera (MDF) las cuales unirán los soportes entre sí con la ayuda de pernos y tuercas, las preguntas y respuestas de la encuesta se visualizarán en un letrero LED ubicado en la parte superior de la estructura, se utilizará sensores infrarrojos situados en las columnas conectados al procesador mediante cable UTP, la información que recolecten los sensores será procesada por una Raspberry Pi modelo 3b+ con un compilador de lenguaje Python, la información procesada se guardará en una base de datos creada con la herramienta Sqlite3 y la información se visualizará en un portal web diseñado con el módulo Tkinter de Python.

3.5.1 Interfaz natural de usuario

La interfaz natural de usuario se asemeja a dos portales que están conformados por 3 columnas cada columna está construida con 3 soportes que fueron hechos de perfiles de aluminio rectangulares (5 x 6 cm), la columna contiene un soporte inferior, un soporte intermedio y un soporte superior como se indica en la Figura 3.3.



Figura 3.3 Soportes de Aluminio

El soporte inferior tiene 60 cm de largo, en un extremo consta de una base que ayudará a sujetarse con la base de la estructura, el soporte intermedio tiene 10 cm de largo, el soporte superior tiene 10 cm de largo y en un extremo consta de una base que ayudará a sujetarse con la parte superior de la estructura, dichos soportes poseen agujeros de 1/8 pulg y son unidos mediante piezas largas de madera(MDF), estas piezas tienen las siguientes dimensiones: 100 cm de largo, 6cm de ancho y 0.5 cm de espesor. Al unir todos los componentes se obtiene una columna como se indica en la Figura 3.4. Tres de estas columnas conforman la estructura de los portales que además es capaz de soportar el peso del visualizador LED.



Figura 3.4 Columnas de madera y aluminio

Además, la interfaz natural de usuario consta de dos sensores de proximidad E18-D80NK como se observa en la Figura 3.5

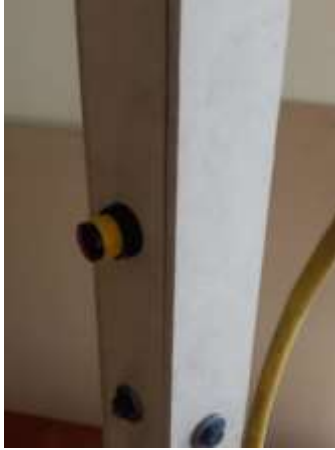


Figura 3.5 sensores de proximidad E18-D80NK

3.5.2 Visualizador LED

El visualizador led posee dimensiones de 200 cm x 20 cm y un peso de 5.6 Kg, las luces LED son de color blanco su manipulación es a través de una aplicación llamada HW LEDShow que es compatible con dispositivos Android, se comunica de forma inalámbrica a través de wifi con el controlador del visualizador. En la Figura 3.6 se puede observar el visualizador LED que se colocará en la parte superior de la estructura.



Figura 3.6 Visualizador LED wifi

3.5.3 Sensado

Para realizar el sistema de recolección de votos electrónicos se usa dos sensores de proximidad E18-D80NK como se puede apreciar en la Figura 3.7, es un transmisor infrarrojo y un receptor

que en conjunto conforman un sensor fotoeléctrico, utiliza luz infrarroja modulada. Este sensor tiene un ajuste por tornillo para regular la distancia, dispone de una salida digital que se activa cuando detecta un obstáculo en el rango especificado; presenta gran funcionalidad, fácil manejo, tamaño y precio aceptables.

Sus características son: Corriente de salida: 100mA/5V

- Consumo de corriente: <25mA DC
- Tiempo de respuesta: 2ms
- Angulo de detección: 15 grados.
- Distancia Ajustable 3cm – 80cm
- Diámetro: 18mm
- Longitud del sensor: 45mm

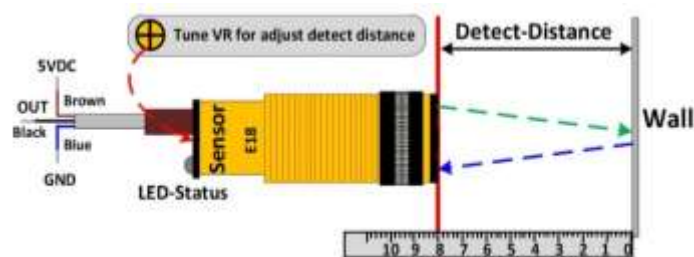


Figura 3.7 Funcionamiento del sensor infrarrojo E18-D80NK

Se eligió estos sensores debido a la gran funcionalidad que tienen frente al proyecto en desarrollo ya que permite detectar objetos a distancia sin necesidad de contacto, posee un buen rango de detección que puede ser regulado, es utilizado ampliamente en 2 campos: la robótica móvil y la automatización industrial [14].

3.5.4 Cableado

Para realizar la comunicación entre los sensores y el procesador se utiliza el cable par trenzado no blindado, es un tipo de cableado de cobre que se utiliza para las comunicaciones telefónicas y la mayoría de las redes Ethernet. Un par de hilos forma un circuito que transmite datos. El par está

trenzado para proporcionar protección contra crosstalk que es el ruido generado por pares de hilos adyacentes en el cable. Los pares de hilos de cobre están envueltos en un aislamiento de plástico con codificación de color y trenzados entre sí. Un revestimiento exterior protege los paquetes de pares trenzados. Los cables UTP tienen un alcance de 100 m [15].

3.5.5 Procesador

El procesador que se utiliza es una Raspberry Pi, es un dispositivo extraordinario, una computadora completamente funcional en un tamaño pequeño y de bajo costo que se puede usar para navegar por juegos de red o escribir sus propios programas, realizar circuitos y dispositivos físicos.

La Raspberry Pi es conocida como una computadora de una sola placa, es igual que una computadora de escritorio, computadora portátil o teléfono inteligente, pero construida en una sola placa de circuito.

El software es de código abierto, siendo su sistema operativo oficial una versión adaptada de Debian, denominada Raspbian, aunque permite usar otros sistemas operativos, incluido una versión de Windows 10 [16].

En todas sus versiones incluye un procesador Broadcom, una memoria RAM, una GPU, puertos USB, HDMI, Ethernet, 40 pines GPIO y un conector para cámara como se muestra en la Figura 3.8.

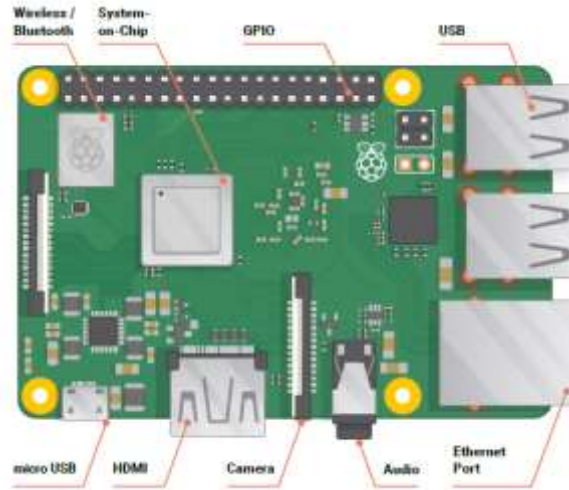


Figura 3.8 Componentes de la Raspberry Pi

3.5.6 Diagrama de conexión

Se usa cable de par trenzado (UTP) para transmitir las señales desde el sensor al procesador. El diagrama de conexión del sistema embebido se indica en la Figura 3.9. La visualización de las preguntas y respuestas se realiza con la ayuda de un visualizador LED que es controlado mediante una aplicación propia del visualizador que se instala en un teléfono celular con un sistema operativo Android.

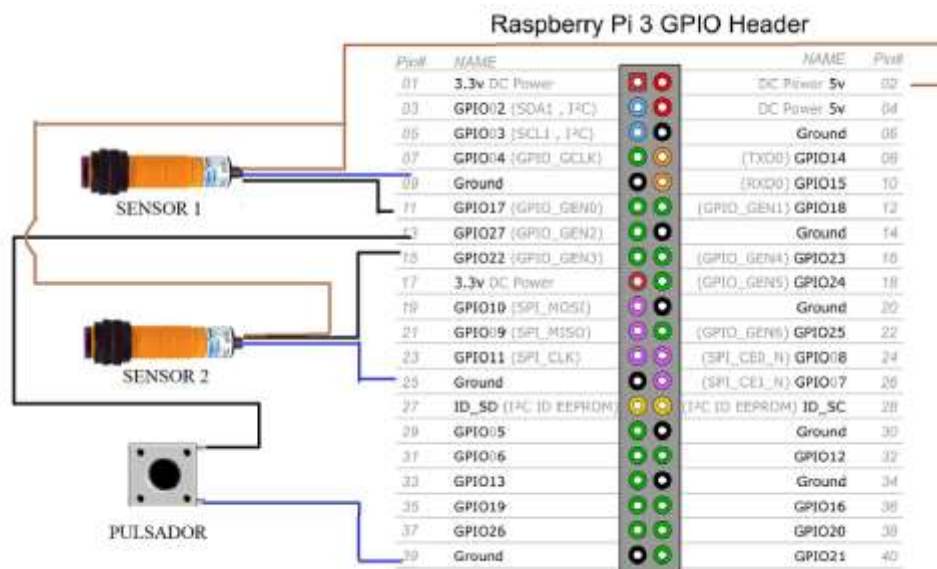


Figura 3.9 Diagrama de conexión

3.6 Descripción del Software

El sistema de recolección de votos electrónicos se divide en dos procesos fundamentales que son: la adquisición y la visualización de información, se realiza un diagrama de flujo para cada proceso para posteriormente realizar su implementación en el procesador a través de scripts en Python.

3.6.1 Diagrama de flujo de la adquisición y acondicionamiento de información

En el diagrama de flujo de la Figura 3.10 se muestra el proceso de adquisición de información, inicia con la importación de librerías e inicialización de variables para ingresar las diferentes preguntas y respuestas de la encuesta, luego se procede a recolectar los votos de los diferentes usuarios para poderlos guardar en una base de datos, existe una señal de entrada proporcionada por un pulsador que tiene como función cambiar de pregunta y respuestas, el proceso se repite hasta que todas las preguntas hayan sido realizadas.

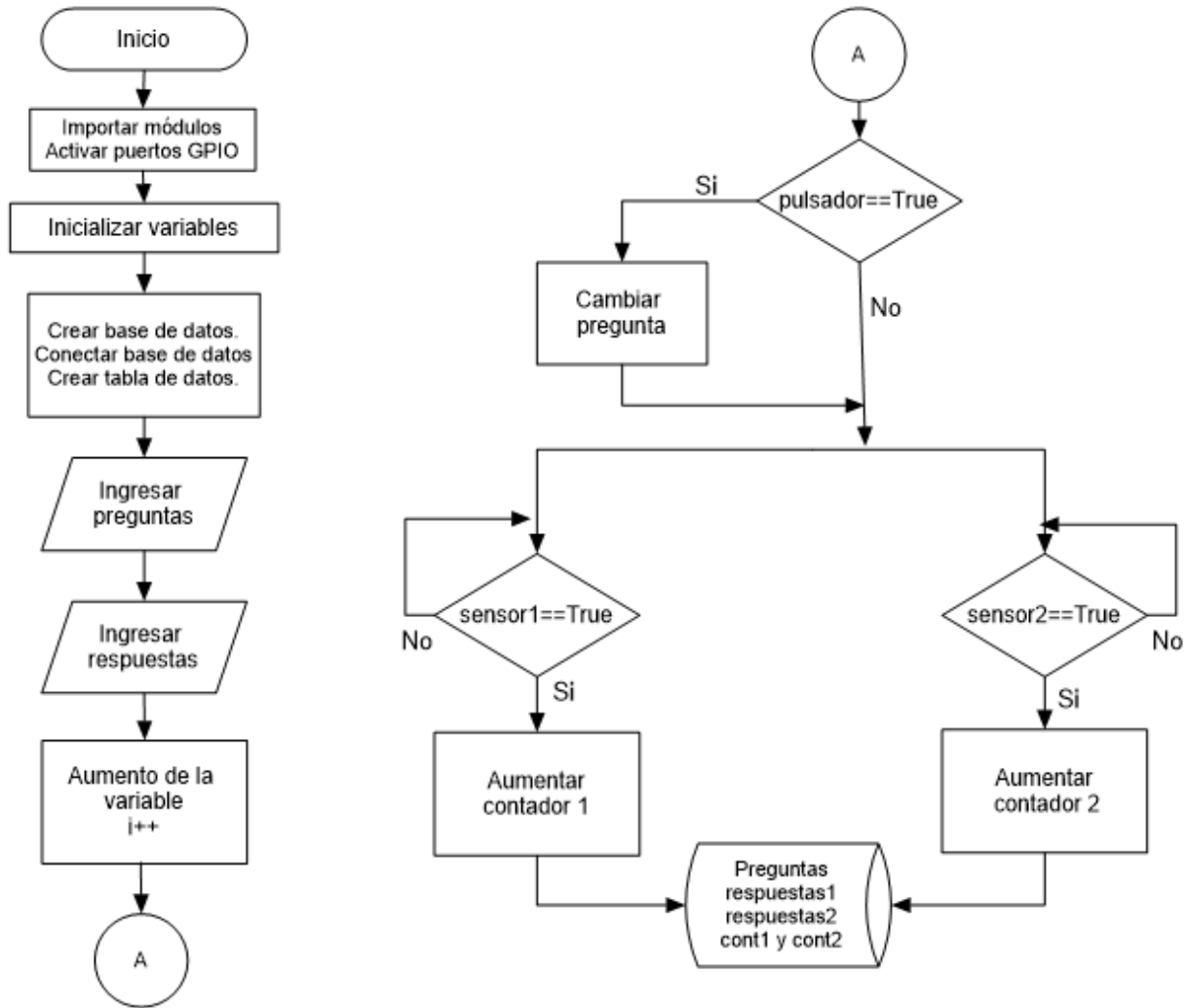


Figura 3.10 Flujograma básico del sistema

3.6.2 Diagrama de flujo de la presentación de información

En la Figura 3.11 se muestra el proceso de presentación de información que consiste en crear un portal web para mostrar la información que existe en una base de datos. Este proceso comienza con la conexión a la base de datos creada por el proceso anterior para luego extraer la información proporcionada por los usuarios tales como: las preguntas, respuestas y los votos obtenidos, para posteriormente mostrarla en una interfaz gráfica que indique los resultados de la encuesta de tal forma que sea fácil de entender.

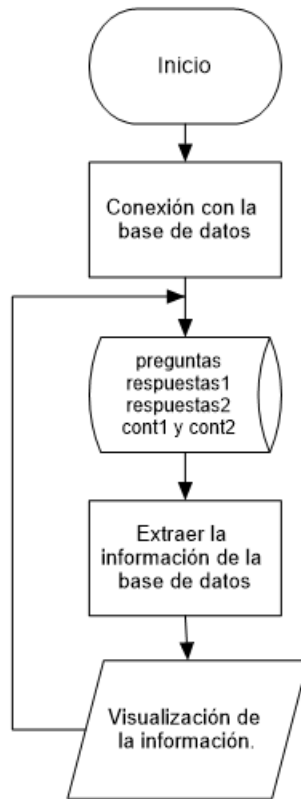


Figura 3.11 Flujograma de la presentación de información

3.6.3 Módulos de Python

Python es un lenguaje que todo el mundo debería conocer. Su sintaxis simple, clara y sencilla, el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo, muy rápido y, lo que es más importante, divertido. Algunos casos de éxito en el uso de Python son Google, Yahoo, la NASA, Industrias Light & Magic, y todas las distribuciones Linux, en las que Python cada vez representa un tanto por ciento mayor de los programas disponibles [17].

Para facilitar el mantenimiento y la lectura los programas demasiado largos pueden dividirse en módulos, agrupando elementos relacionados. Los módulos son entidades que permiten una organización y división lógica del código principal.

Para importar un módulo se utiliza la palabra clave `import` seguida del nombre del módulo, que consiste en el nombre del archivo menos la extensión. El `import` no solo hace que se tenga disponible todo lo definido dentro del módulo, sino que también ejecuta el código del módulo.

3.6.3.1 Módulo `time`

El módulo `time` [18], Este módulo proporciona varias funciones relacionadas con el tiempo. Aunque este módulo siempre está disponible, no todas las funciones están disponibles en todas las plataformas. Las funciones en este módulo pueden no manejar fechas y horas anteriores a la época o en el futuro. El punto de corte en el futuro está determinado por la biblioteca C; para sistemas de 32 bits, es típicamente en 2038.

3.6.3.2 Módulo `datetime`

El módulo `datetime` [19], El módulo proporciona clases para manipular fechas y horas de manera simple y compleja. Si bien se admite la aritmética de fecha y hora, el enfoque de la implementación es la extracción eficiente de atributos para el formato y la manipulación de salida.

3.6.3.3 Módulo `sys`

Es módulo `sys` [20], proporciona acceso a algunas variables utilizadas o mantenidas por el intérprete de Python y a funciones que interactúan fuertemente con el intérprete. Siempre está disponible.

3.6.3.4 Módulo `sqlite3`

El módulo `sqlite3` [21], es una biblioteca que proporciona una base de datos liviana basada que no requiere un proceso de servidor independiente y permite acceder a la base de datos utilizando una variante no estándar del lenguaje de consulta SQL. Algunas aplicaciones pueden usar SQLite para el almacenamiento interno de datos. También es posible crear un prototipo de

una aplicación utilizando SQLite y luego transferir el código a una base de datos más grande como PostgreSQL u Oracle.

3.6.3.5 Módulo functools

El módulo functools [22], es para funciones de orden superior: funciones que actúan sobre o devuelven otras funciones. En general, cualquier objeto exigible puede tratarse como una función para los fines de este módulo.

3.6.3.6 Módulo Tkinter

El módulo Tkinter (Tk interfaz) [23], es la interfaz estándar de Python para el kit de herramientas GUI de Tk que son utilizadas para crear ventanas donde se visualiza diferentes tipos de información. Ambas Tkinter y Tk están disponibles en la mayoría de las plataformas Linux, así como en los sistemas Windows.

3.6.3.7 Módulo RPi.GPIO

El módulo RPi.GPIO de Python [24], permite leer y escribir fácilmente señales de entrada / salida en los pines GPIO de la Raspberry Pi dentro de una secuencia de comandos de Python. La librería está ahora incluida en la imagen estándar de Raspbian.

Capítulo 4

Implementación y pruebas

En este capítulo, la teoría desarrollada hasta ahora es comprobada a través de la implementación de la estructura que corresponde a la interfaz natural de usuario y la implementación de los criterios de los diagramas de flujo sobre el procesador Raspberry Pi. Además, se explican y analizan las pruebas para corroborar su funcionamiento adecuado.

4.2 Hardware

Teniendo definido los materiales que se va a utilizar para construir el sistema de recolección de votos electrónicos, se procede a la implementación de todos los componentes para así obtener un resultado final al que se le puedan realizar pruebas de funcionamiento.

4.2.1 Implementación de la interfaz natural de usuario

Para la implementación de la interfaz natural de usuario se utilizó todos los elementos mencionados en la sección 3.5. Se inicia la implementación con el ensamblaje de las columnas como se muestra en la Figura 4.1, en la columna central se implementan los sensores infrarrojos E18-D80NK, el cableado de los sensores se colocan en la parte interna de la columna y salen por un orificio en la parte superior de la misma para conectarse con la Raspberry Pi, en la parte interna de la columna lateral derecha se ubica un cable que servirá para alimentar a la Raspberry Pi y al visualizador LED que se ubicaran en la parte superior de la estructura, posteriormente se procede

a empotrar las columnas en la base de madera y a la parte superior de la estructura con la ayuda de pernos y tuercas, como resultado se obtiene la estructura que se muestra en la Figura 4.2.

La altura de la estructura está basada en la Norma Técnica Ecuatoriana INEN 2309 accesibilidad de las personas al medio físico. Puertas. Requisitos. [25]



Figura 4.1 Columna de aluminio y madera(MDF)



Figura 4.2 Estructura final

4.2.2 Configuraciones adicionales

Con la implementación de la estructura completa surgieron algunas observaciones las cuales fueron tomadas en cuenta para mejorar el funcionamiento del sistema, la principal configuración se la realizó en la base de la estructura, se vio la necesidad de realizar una pequeña rampa como se muestra en la Figura 4.3, para evitar que los usuarios tropiecen al momento de atravesar los portales.



Figura 4.3 Base de la estructura

La segunda configuración adicional corresponde a la unión de la estructura, se cambió las tuercas convencionales de los pernos por tuercas mariposa como se puede observar en la Figura 4.4, estas tuercas facilitan el montaje y desmontaje de la estructura.



Figura 4.4 Tuercas mariposa

Finalmente se tiene una configuración adicional que corresponde al cableado de la interfaz natural de usuario, este cableado fue protegido con manguera corrugada para evitar daños en los cables ya que estaban expuestos a la intemperie. En la Figura 4.5 se muestra la manguera corrugada en el cable de poder de la interfaz natural de usuario, mientras que en la Figura 4.6 se observa la protección de los cables de los sensores que van conectados a la Raspberry Pi.



Figura 4.5 Cable de poder cubierto con la manguera corrugada



Figura 4.6 Protección de los cables de conexión de los sensores

4.3 Software

Luego de comprender el funcionamiento del sistema a través del diagrama de flujo se procede a implementarlo en el procesador Raspberry Pi 3b+ que se muestra en la Figura 4.7. Este procesador contiene un sistema operativo Raspbian, además para la programación se usa el lenguaje Python el cual es muy sencillo de entender, pero a la vez ofrece mucha robustez en sus programas.



Figura 4.7 Raspberry Pi 3b+

4.3.1 Implementación de la adquisición y acondicionamiento de la información

Se realizó un script donde se encuentra todo el código de programación del proceso de adquisición y acondicionamiento de la información, dicho código empieza con la importación de algunos módulos propios de Python y declaración de los pines GPIO que se van a utilizar para la lectura de las señales de los sensores y el pulsador, luego se procede a crear y conectarse a una base de datos, se creará una tabla y se guardará la información suministrada por el sistema, las preguntas y respuestas de la encuesta se guardan en variables tipo arreglos para tener la información de una forma más ordenada para posteriormente poderla enviar a la base de datos y usarla en el resto del programa, luego de verificar que las preguntas están correctamente ingresadas el programa ingresa en un ciclo de lectura de los sensores que al detectar una señal de alto y bajo aumenta una unidad a un contador, posteriormente se guarda esta información en la base de datos, cada sensor tiene su propio contador, ese ciclo se repite hasta recibir la señal de cambio de pregunta proporcionada por el pulsador, si existe otra pregunta en fila el programa procederá repetir la misma dinámica descrita anteriormente, caso contrario el programa pedirá ingresar más preguntas,

si no fuese este el caso el programa dará la opción de salir del programa y así finalizar el proceso. El programa completo se encuentra en el Apéndice B.1.

4.3.1.1 Configuraciones adicionales

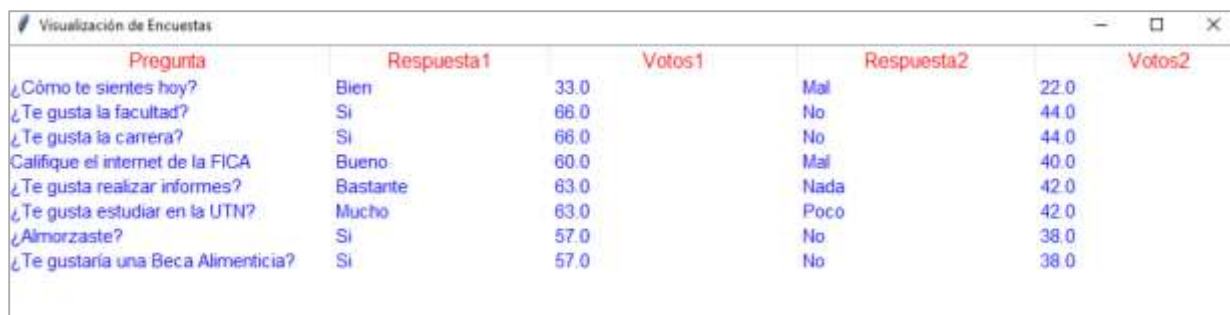
Se puede destacar una configuración luego de haber terminado la programación del proceso de adquisición y acondicionamiento de la información, consiste en el conteo de los usuarios al pasar por los portales, este conteo se mejoró implementado una función que permite aumentar el contador siempre y cuando el procesador reciba una señal de alto y bajo proporcionada por los sensores, es decir se realizará el conteo cuando el usuario pase completamente por el portal.

4.3.2 Implementación de la presentación de la información

Teniendo en claro la dinámica del flujograma de la presentación de la información, se procede a realizar la implementación en el procesador de igual forma que el proceso anterior, se lo realizó en lenguaje Python, la única diferencia que en este caso se utilizó programación orientada a objetos, dando uso a ventanas hechas con el complemento Tkinter de Python para poder visualizar de forma gráfica los resultados de las encuestas realizadas. Se realizó un nuevo script donde se encuentra todo el código referente al proceso de presentación de la información que comienza con la importación de algunos módulos de Python que facilitarán la programación, posteriormente se procede a conectarse con la base de datos creada por el anterior proceso para luego extraer la información contenida, con la ayuda de del complemento Tkinter se crea una ventana con dos pestañas donde se visualizará el historial de encuestas y la encuesta actual, con ayuda de los widgets de Tkinter principalmente Labels, Canvas y Graphs se puede exportar la información extraída de la base de datos a la ventana creada que tendrá la función de un portal web, todo este proceso está dentro de un ciclo para que se actualice en un tiempo determinado. El programa completo se encuentra en el Apéndice B.2.

4.3.2.1 Configuraciones adicionales

Después de la implementación del proceso de presentación de la información se obtuvo la interfaz que se muestra la Figura 4.8, se pudo observar que no se comprendía de forma rápida y clara la información correspondiente a los resultados de las encuestas, se procedió a realizar una nueva interfaz para que se comprenda la información de forma más clara y además interactiva con el usuario. En la Figura 4.9 y Figura 4.10 se observa el resultado obtenido luego de realizar las configuraciones respectivas.



Pregunta	Respuesta1	Votos1	Respuesta2	Votos2
¿Cómo te sientes hoy?	Bien	33.0	Mal	22.0
¿Te gusta la facultad?	Si	66.0	No	44.0
¿Te gusta la carrera?	Si	66.0	No	44.0
Califique el internet de la FICA	Bueno	60.0	Mal	40.0
¿Te gusta realizar informes?	Bastante	63.0	Nada	42.0
¿Te gusta estudiar en la UTN?	Mucho	63.0	Poco	42.0
¿Almorzaste?	Si	57.0	No	38.0
¿Te gustaría una Beca Alimenticia?	Si	57.0	No	38.0

Figura 4.8 Interfaz de visualización del resultado de las encuestas



Figura 4.9 Interfaz interactiva (Encuesta actual)

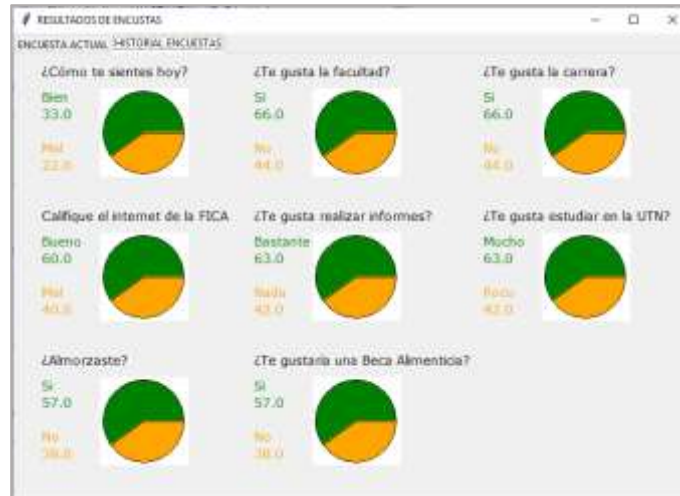


Figura 4.10 Interfaz interactiva (Historial encuestas)

4.4 Pruebas de funcionamiento

Con el sistema de recolección de votos electrónicos implementado completamente tanto Hardware como software se procede a realizar las pruebas de campo para validar su funcionamiento.

4.4.1 Prueba de los sensores a diferentes condiciones de luz

En esta prueba se demuestra el funcionamiento de los sensores expuestos a diferentes condiciones de luminosidad, la primera prueba se la realizó en una habitación donde exista alta cantidad de luz solar como se muestra en la Figura 4.11. La segunda prueba se la realizó en una habitación donde casi no existe la luz solar como se indica en la Figura 4.12, en las diferentes condiciones se revisó la lectura de los sensores, para esto se realizó 20 tomas de datos (10 para cada sensor) para cada condición.



Figura 4.11 Prueba de los sensores a alta luminosidad



Figura 4.12 Prueba de los sensores a baja luminosidad

4.4.2 Prueba de los sensores a diferente velocidad del paso de los usuarios

Las pruebas fueron realizadas tomando en cuenta las diferentes velocidades con la que las personas atraviesan los diferentes portales. Para esta prueba se eligió tres condiciones de velocidad que son: velocidad alta, velocidad normal, velocidad lenta. En las diferentes condiciones se revisó la lectura de los sensores, se realizó 10 tomas de datos para cada condición.

4.4.3 Prueba de alcance de los sensores

En esta prueba se demuestra el funcionamiento de los sensores a diferentes distancias, es decir cuando el usuario pasa por el centro del portal o si su ubicación está más posicionada a la derecha o a la izquierda del portal, cabe destacar que el sensor infrarrojo utilizado tiene un alcance que va desde los 3 cm hasta los 80 cm. En las diferentes condiciones se revisó la lectura de los sensores realizando 10 tomas de datos para cada condición.

4.4.4 Prueba de los sensores al paso de las personas

Esta prueba de funcionamiento de los sensores consiste en realizar la lectura de las personas cuando pasa una a la vez y cuando pasan en grupo. Adicionalmente en esta prueba se realizó la lectura de los usuarios cuando pasan con mochila. En las diferentes condiciones se realizó la lectura de los sensores realizando 10 tomas de datos para cada condición.

4.4.5 Prueba de visualización del letrero led a diferentes condiciones de luz

En esta prueba tiene como objetivo verificar la visualización del mensaje mostrado por el letrero LED a diferentes condiciones de luz, la prueba con luminosidad baja se la realizó en una habitación con poca luminosidad como se muestra en la Figura 4.13. La prueba de funcionamiento con luminosidad alta se la realizó en una habitación donde entra gran cantidad de luz solar como se indica en la Figura 4.14, en las diferentes condiciones se observó el letrero LED para determinar

si el mensaje mostrado es o no visible, para un mejor análisis la visibilidad se ponderó del 1 al 5 donde 1 corresponde a no visible y 5 corresponde a muy visible.



Figura 4.13 Visualización del mensaje con luminosidad baja



Figura 4.14 Visualización del mensaje con luminosidad alta

4.4.6 Prueba de visualización del letrero led a diferentes distancias

De igual forma como la prueba anterior el objetivo es la visualización del mensaje mostrado, pero en este caso en condiciones de distancia, la prueba se la realizó en tres condiciones diferentes que son: 1 metro de distancia como se muestra en la Figura 4.15, 4 metros de distancia como se muestra en la Figura 4.16 y mayor a 10 metros de distancia como se observa en la Figura 4.17. En las diferentes condiciones se observó el letrero LED para determinar si el mensaje mostrado es o no visible, para un mejor análisis la visibilidad se la ponderó del 1 al 5 donde 1 corresponde a no visible y 5 corresponde a muy visible.



Figura 4.15 Visualización del mensaje a 1 metro de distancia



Figura 4.16 Visualización del mensaje a 4 metros de distancia



Figura 4.17 Visualización del mensaje a 10 metros de distancia

4.5 Análisis de resultados

Con todas las pruebas realizadas correctamente se obtienen varias tablas que resumen los resultados obtenidos que posteriormente se analizan para poder llegar a un resultado final de cada prueba.

En la Tabla 4.1 se muestra las lecturas realizadas por los sensores expuestos a diferentes condiciones de luz. Se puede ver claramente que la variación de la luminosidad no afectó en nada la lectura de los sensores ya que no hubo errores en sus lecturas, esto se debe a que se utilizó sensores infrarrojos de grado industrial los cuales trabajan en ambientes abiertos y cerrados sin ningún problema.

La Tabla 4.2 muestra las lecturas realizadas por los sensores a diferentes velocidades del paso de los usuarios. De la tabla se puede destacar que existió errores en la lectura de los sensores cuando los usuarios pasan por los portales a una velocidad alta, esto se debe a que existe un tiempo de espera en la lectura del sensor, este tiempo se lo estableció tomando en cuenta que los usuarios atravesaran los portales con una caminata normal, es por eso que en velocidad normal y velocidad baja no existió ningún error en la lectura de datos.

En la Tabla 4.3 se puede observar los resultados de las lecturas realizadas por los sensores con respecto al diferente posicionamiento de las personas al travesar los portales. En los resultados se observa que no existe ningún error en la lectura de los sensores a pesar de los diferentes posicionamientos de los usuarios, esto se debe a que los sensores tienen un alcance máximo de 80 cm que cubre todo el ancho de los portales que es de 75 cm.

La Tabla 4.4 contiene información que corresponde a la visualización del mensaje mostrado por el letrero LED a diferentes condiciones de iluminación. De la tabla se puede destacar que la visibilidad del mensaje es más clara en un ambiente oscuro que en un ambiente iluminado, esto se debe a la intensidad de la luz natural, a pesar de esto el mensaje si es legible por lo que el visualizador LED puede trabajar en ambientes abierto sin problema.

En la Tabla 4.5 se muestra los resultados de la visibilidad del mensaje mostrado por el letrero LED a diferentes condiciones de distancia. La visibilidad del mensaje se va distorsionando conforme la distancia entre el letrero y el usuario aumenta. Esto también depende de la visión de cada usuario ya que si posee algún problema de vista no visualizara claramente el mensaje a una larga distancia. Cabe destacar que el mensaje mostrado por el letrero sigue siendo claro a los 10 metros de distancia.

Tabla 4.1 Resultados de la prueba de los sensores a diferentes condiciones de luz

Luminosidad	Número de lecturas realizadas	
	Sensor 1	Sensor 2
Alta	10	10
Baja	10	10

Tabla 4.2 Resultados de la prueba de los sensores a diferente velocidad del paso de los usuarios

Velocidad	Número de lecturas realizadas	
	Sensor 1	Sensor 2
Alta	8	7
Normal	10	10
Baja	10	10

Tabla 4.3 Resultados de la prueba de los sensores al paso de las personas

Posicionamiento	Número de lecturas realizadas	
	Sensor 1	Sensor 2
Izquierda	10	10
Normal	10	10
Derecha	10	10

Tabla 4.4 Resultados de la prueba de visualización del letrero LED a diferentes condiciones de luz

Luminosidad	Visibilidad
Alta	4
Baja	5

Tabla 4.5 Resultados de la prueba de visualización del letrero LED a diferentes distancias

Luminosidad	Visibilidad
1 metro	5
4 metros	5
Más de 10 metros	4

Capítulo 5

Conclusiones y trabajo futuro

En este capítulo se describe las conclusiones obtenidas del diseño, implementación y pruebas del sistema de recolección de votos, además las recomendaciones que se pueden aplicar para un trabajo futuro.

5.1 Conclusiones

- El desarrollo de la recolección de votos electrónicos utilizando una interfaz natural de usuario se realiza en las siguientes etapas: adquisición de la información, acondicionamiento y almacenamiento de la información, presentación de los datos obtenidos.
- El sistema de recolección de votos electrónicos consta de tres subfunciones, la primera subfunción corresponde a la interfaz natural de usuario que interactúa de forma directa con las personas para adquirir sus opiniones respecto a la encuesta realizada, la segunda subfunción tiene como propósito ordenar y guardar la información en una base de datos, la tercera subfunción corresponde a la presentación de los votos obtenidos en un portal web.
- Se obtuvo un sistema de recolección de votos robusto tanto en hardware como en software. El hardware corresponde a la estructura que interactúa con los usuarios para adquirir sus

opiniones sobre las encuestas planteadas. El software corresponde a los programas realizados para adquirir, acondicionar, guardar y presentar la información.

- El sistema de recolección de votos electrónicos funciona correctamente en ambientes donde existe iluminación alta y baja. El sensado no se ve afectado por las diferentes posiciones de los usuarios al atravesar los portales, pero existe una breve variación en la lectura cuando las personas pasan a alta velocidad por los portales.

5.2 Trabajo futuro

En el presente proyecto se desarrolla el sistema de recolección de votos electrónicos para realizar encuestas de respuestas rápidas. Para ampliar y mejorar el funcionamiento del sistema se propone como trabajo a futuro lo siguiente:

- Realizar la comunicación directa entre la Raspberry Pi y el visualizador LED para que la dinámica del proceso tenga un funcionamiento más óptimo.
- Implementar un visualizador LED adicional que tenga como objetivo mostrar los mensajes en la vista posterior de la de la interfaz natural de usuario.
- Crear una página web la cual interactúe con la base de datos del sistema, de esta forma poder visualizar el resultado de las encuestas vía internet.
- Desarrollar un sistema de recolección de votos electrónicos a través de interfaz natural de usuario que permita realizar encuestas con opciones de respuesta mayores a dos.

Bibliografía

- [1] V. Morales Rocha, Ó. Ruiz Hernández y L. F. Fernández Martínez, «Mecanismo de auditoría para detección de manipulación de votos en sistemas de votación electrónica,» *Paakat: Revista de Tecnología y Sociedad*, nº 14, 19 Febrero 2018.
- [2] V. Díaz de Rada y J. A. Domínguez Álvarez, «Comparación de métodos de campo en la encuesta,» *Reis. Revista Española de*, nº 158, pp. 137-147, 2017.
- [3] J. T. Places Chungata, E. R. Portilla López, O. D. León Granizo y M. Botto-Tobar, «Confiability and considerations of the electronic vote,» *Revista Ciencia e investigación*, vol. 2, nº 5, pp. 26-38, 2017.
- [4] O. J. G. Echeverri, «Considerations on the Development of Natural Interfaces,» *CINTEX*, vol. 19, pp. 189-193, 2014.
- [5] J. Á. Noguera Arnaldos, M. A. Paredes Valverde, R. Valencia García y M. Á. Rodríguez García, «Sistema de diálogo basado en mensajería instantánea para el control de dispositivos en el internet de las cosas,» *Procesamiento del Lenguaje Natural*, nº 55, pp. 173-176, 2015.
- [6] S. A. Castro, B. Medina y L. L. Camargo, «Supervisión y Control Industrial a través de Teléfonos Inteligentes usando un Computador de Placa Única Raspberry Pi,» *Información Tecnológica*, vol. 27, nº 2, 2016.
- [7] O. d. J. May Tzuc, R. G. Quijano Cetina y F. Fernández Castillo, «Sistema domótico de control de luces mediante una aplicación Android basado en una tarjeta Raspberry-Pi,» *Tendencias de Investigación y Desarrollo en la Ingeniería*, pp. 45-49, 2015.
- [8] C. R. Cusi Vaca y D. F. Sánchez Morocho, «Estudio y diseño de un prototipo para el monitoreo de acuarios utilizando tecnología Wifi (IEEE 802.11b/g/n) enfocada al IoT (Internet of things) mediante una plataforma Raspberry Pi y el sistema operativo Android,» Quito, 2016.

- [9] L. M. Chicaiza Satan, «Diseño e implementación de un sistema domótico mediante una tarjeta raspberry Pi y controlado con una aplicación android para que personas con discapacidad física puedan cumplir tareas básicas del hogar,» Guayaquil, 2017.
- [10] L. Puspa Dewi, J. Andjarwirawan y R. Putra Wardoyo, «Android Application for Monitoring Soil Moisture Using Raspberry Pi,» *2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIIT)*, pp. 178-184, 18 Enero 2018.
- [11] F. Pérez, A. Félix y J. L. Guerra Guerra, «Internet de las cosas,» *Perspectiv@s*, vol. 10, nº 11, pp. 45-49, 2017.
- [12] J. Llerena, B. Velasco y F. Edmundo, «Implementación de un control domótico utilizando raspberry pi y una base de datos NoSQL,» Sangolquí, 2017.
- [13] A. P. Cervantes Márquez, C. Cerón Garnica y E. Archundia Sierra, «La usabilidad en interfaces naturales como apoyo en el proceso de enseñanza-aprendizaje: estudio de caso para un sistema interactivo de estructuras de datos,» *Universidad&Ciencia*, vol. 6, pp. 206-220, 2017.
- [14] A. M. Alta Andrango y J. K. Guijarro Rebelo, «DISEÑO E IMPLEMENTACIÓN DE UN CONTROL POR COMANDOS DE VOZ A UNA SILLA DE RUEDAS,» Ibarra, 2016.
- [15] José Ricardo Núñez Álvarez, Israel Francisco Benítez Pina y Danilo Mariano Carbonell Mestre, «Cables de red,» *Journal of Engineering and Technology for Industrial Applications*, vol. IV, nº 13, pp. 166-175, 2018.
- [16] G. Halfacree, «How to use your new computer,» *The Official Beginners'Guide*, vol. 1, nº 1, p. 244, 2018.
- [17] R. González Duque, *Python para todos*, Madrid, 2014, p. 160.
- [18] P. S. Foundation, «Python,» Python Software Foundation, 2001-2019. [En línea]. Available: <https://docs.python.org/2/library/time.html?highlight=time#module-time>. [Último acceso: 8 Julio 2019].
- [19] P. S. Foundation, «Python Software Foundation,» 2001-2019. [En línea]. Available: <https://docs.python.org/2/library/datetime.html?highlight=time#module-datetime>. [Último acceso: 8 Julio 2019].
- [20] P. S. Foundation, «Python Software Foundation,» 2001-2019. [En línea]. Available: <https://docs.python.org/2/library/sys.html?highlight=sys#module-sys>. [Último acceso: 8 Julio 2019].

- [21] P. S. Foundation, «Python Software Foundation,» 2001-2019. [En línea]. Available: <https://docs.python.org/2/library/sqlite3.html?highlight=sqlite#module-sqlite3>. [Último acceso: 8 Julio 2019].
- [22] P. S. Foundation, «Python Software Foundation,» 2001-2019. [En línea]. Available: <https://docs.python.org/2/library/functools.html?highlight=functools#module-functools>. [Último acceso: 8 Julio 2019].
- [23] P. S. Foundation, «Python Software Foundation,» 2001-2019. [En línea]. Available: <https://docs.python.org/2/library/tkinter.html?highlight=tkinter>. [Último acceso: 8 Julio 2019].
- [24] R. P. foundation, «Raspberry Pi,» [En línea]. Available: <https://www.raspberrypi.org/blog/gpio-zero-a-friendly-python-api-for-physical-computing/>. [Último acceso: 8 Julio 2019].
- [25] S. E. d. Normalización, «ACCESIBILIDAD DE LAS PERSONAS AL MEDIO FÍSICO. PUERTAS. REQUISITOS,» de *Norma Tecnica Ecuatoriana*, Quito, Pichincha, 2018, pp. 1-10.

Apéndice

A. Método de criterios ponderados

A.1 Matriz morfológica

Tabla A.1 Matriz morfológica

Señal de inicio	
Iniciar programa	Opción A: Desde consola
Adquisición de información	
Interfaz Natural de Usuario	
Soportes	Opción A: Largos (una sola pieza)
	Opción B: Cortos (3 piezas)
Material de soportes	Opción A: Aluminio
	Opción B: Acero
Material de las columnas	Opción A: Acrílico
	Opción B: Madera (MDF)
	Opción C: Madera
Union de soportes y columnas	Opción A: Tuercas Tornillos
Sensores	Opción A: Ultrasónico
	Opción B: Infrarrojo
Visualizador	Opción A: FLIO DOT display
	Opción B: Letrero Led
	Opción C: Matrices Led
Cables	Opción A: UTP
	Opción B: STP
Acondicionamiento de información	
Sistema Embebido	
Procesador	Opción A: Arduino
	Opción B: Raspberry Pi-Python

	Opción C: PIC
Base de datos	Opción A: CSV File
	Opción B: SQLite
	Opción C: MySQL
Presentación de información	
Portal web	Opción A: Django
	Opción B: Python Tkinter

A.2 Método de criterios ponderados

Criterios de Diseño

- Precio (Bajo Costo)
- Accesibilidad en el Mercado
- Peso (Liviano)
- Fácil ensamblaje

A.2.1 Tablas de ponderación de criterios

Tabla A.2 Tabla de criterios

CRITERIO	Precio	Accesibilidad	Peso	Ensamble	E+1	Ponderación
Precio	x	1	1	1	4	0,4
Accesibilidad	0	x	1	1	3	0,3
Peso	0	0	x	0,5	1,5	0,15
Ensamble	0	0	0,5	x	1,5	0,15
				Suma	10	1

Tabla A.3 Ponderación de criterios de los soportes

Soportes				
Precio	Solución A	Solución B	E+1	Pondera
Solución A	x	0	1	0,333333333
Solución B	1	x	2	0,666666667
		suma	3	1
Accesibilidad	Solución A	Solución B	E+1	Pondera
Solución A	x	0,5	1,5	0,5
Solución B	0,5	x	1,5	0,5
		suma	3	1

	Solución A	Solución B	E+1	Pondera
Peso	x	0	1	0,333333333
Solución A	x	0	1	0,333333333
Solución B	1	x	2	0,666666667
		suma	3	1
Ensamble	Solución A	Solución B	E+1	Pondera
Solución A	x	0	1	0,333333333
Solución B	1	x	2	0,666666667
		suma	3	1

Tabla A.4 Ponderación de criterios del material de los soportes

Material soportes				
Precio	Solución A	Solución B	E+1	Pondera
Solución A	x	1	2	0,666666667
Solución B	0	x	1	0,333333333
		suma	3	1
Accesabilidad	Solución A	Solución B	E+1	Pondera
Solución A	x	0,5	1,5	0,5
Solución B	0,5	x	1,5	0,5
		suma	3	1
Peso	Solución A	Solución B	E+1	Pondera
Solución A	x	1	2	0,666666667
Solución B	0	x	1	0,333333333
		suma	3	1
Ensamble	Solución A	Solución B	E+1	Pondera
Solución A	x	1	2	0,666666667
Solución B	0	x	1	0,333333333
		suma	3	1

Tabla A.5 Ponderación de criterios del material de las columnas

Material Columnas					
Precio	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0	0	1	0,166666667
Solución B	1	x	0,5	2,5	0,416666667
Solución C	1	0,5	x	2,5	0,416666667
			suma	6	1
Accesabilidad	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0,5	0,5	2	0,333333333
Solución B	0,5	x	0,5	2	0,333333333
Solución C	0,5	0,5	x	2	0,333333333
			suma	6	1
Peso	Solución A	Solución B	Solución C	E+1	Pondera

Solución A	x	0,5	1	2,5	0,416666667
Solución B	0,5	x	1	2,5	0,416666667
Solución C	0	0	x	1	0,166666667
			suma	6	1
Ensamble	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0,5	0,5	2	0,333333333
Solución B	0,5	x	0,5	2	0,333333333
Solución C	0,5	0,5	x	2	0,333333333
			suma	6	1

Tabla A.6 Ponderación de criterios del procesador

Procesador					
Precio	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0,5	0,5	2	0,333333333
Solución B	0,5	x	0,5	2	0,333333333
Solución C	0,5	0,5	x	2	0,333333333
			suma	6	1
Accesabilidad	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0,5	1	2,5	0,416666667
Solución B	0,5	x	1	2,5	0,416666667
Solución C	0	0	x	1	0,166666667
			suma	6	1
Peso	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0,5	0,5	2	0,333333333
Solución B	0,5	x	0,5	2	0,333333333
Solución C	0,5	0,5	x	2	0,333333333
			suma	6	1
Ensamble	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0	1	2	0,333333333
Solución B	1	x	1	3	0,5
Solución C	0	0	x	1	0,166666667
			suma	6	1

Tabla A.7 Ponderación de criterios de los sensores

Sensores				
Precio	Solución A	Solución B	E+1	Ponderación
Solución A	x	0	1	0,333333333
Solución B	1	x	2	0,666666667
		suma	3	1
Accesabilidad	Solución A	Solución B	E+1	Pondera
Solución A	x	0,5	1,5	0,5
Solución B	0,5	x	1,5	0,5
		suma	3	1
Peso	Solución A	Solución B	E+1	Pondera

Solución A	x	0,5	1,5	0,5
Solución B	0,5	x	1,5	0,5
		suma	3	1
Ensamble	Solución A	Solución B	E+1	Pondera
Solución A	x	0	1	0,333333333
Solución B	1	x	2	0,666666667
		suma	3	1

Tabla A.8 Ponderación de criterios del visualizador

Visualizador					
Precio	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0	0,5	1,5	0,25
Solución B	1	x	1	3	0,5
Solución C	0,5	0	x	1,5	0,25
			suma	6	1
Accesabilidad	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0	0	1	0,166666667
Solución B	1	x	1	3	0,5
Solución C	1	0	x	2	0,333333333
			suma	6	1
Peso	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0	0	1	0,166666667
Solución B	1	x	0,5	2,5	0,416666667
Solución C	1	0,5	x	2,5	0,416666667
			suma	6	1
Ensamble	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0	0,5	1,5	0,25
Solución B	1	x	1	3	0,5
Solución C	0,5	0	x	1,5	0,25
			suma	6	1

Tabla A.9 Ponderación de criterios de los cables de conexión

Cables				
Precio	Solución A	Solución B	E+1	Pondera
Solución A	x	1	2	0,666666667
Solución B	0	x	1	0,333333333
		suma	3	1
Accesabilidad	Solución A	Solución B	E+1	Pondera
Solución A	x	0,5	1,5	0,5
Solución B	0,5	x	1,5	0,5

		suma	3	1
Peso	Solución A	Solución B	E+1	Pondera
Solución A	x	0,5	1,5	0,5
Solución B	0,5	x	1,5	0,5
		suma	3	1
Ensamble	Solucion A	Solución B	E+1	Pondera
Solucion A	x	0,5	1,5	0,5
Solución B	0,5	x	1,5	0,5
		suma	3	1

Tabla A.10 Ponderación de criterios de la base de datos

Base de datos					
Precio	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0,5	0,5	2	0,333333333
Solución B	0,5	x	0,5	2	0,333333333
Solución C	0,5	0,5	x	2	0,333333333
			suma	6	1
Accesabilidad	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0,5	0,5	2	0,333333333
Solución B	0,5	x	0,5	2	0,333333333
Solución C	0,5	0,5	x	2	0,333333333
			suma	6	1
Peso	Solucion A	Solución B	Solución C	E+1	Pondera
Solucion A	x	0,5	0,5	2	0,333333333
Solución B	0,5	x	0,5	2	0,333333333
Solución C	0,5	0,5	x	2	0,333333333
			suma	6	1
Ensamble	Solución A	Solución B	Solución C	E+1	Pondera
Solución A	x	0,5	0,5	2	0,333333333
Solución B	0,5	x	0,5	2	0,333333333
Solución C	0,5	0,5	x	2	0,333333333
			suma	6	1

Tabla A.11 Ponderación de criterios del portal web

Portal web				
Precio	Solución A	Solución B	E+1	Pondera
Solución A	x	0,5	1,5	0,5
Solución B	0,5	x	1,5	0,5

		suma	3	1
Accesibilidad	Solución A	Solución B	E+1	Pondera
Solución A	x	0,5	1,5	0,5
Solución B	0,5	x	1,5	0,5
		suma	3	1
Peso	Solución A	Solución B	E+1	Pondera
Solución A	x	0,5	1,5	0,5
Solución B	0,5	x	1,5	0,5
		suma	3	1
Ensamble	Solución A	Solución B	E+1	Pondera
Solución A	x	0,5	1,5	0,5
Solución B	0,5	x	1,5	0,5
		suma	3	1

A.2.2 Tablas de conclusión del método de criterios ponderados

Tabla A.12 Selección de los soportes

Soportes						
Conclusión	Precio	Accesibilidad	Peso	Ensamble	Sumatoria	Prioridad
Solución A	0,133333333	0,15	0,05	0,05	0,383333333	2
Solución B	0,266666667	0,15	0,1	0,1	0,616666667	1

Tabla A.13 Selección del material de los soportes

Material soportes						
Conclusión	Precio	Accesibilidad	Peso	Ensamble	Sumatoria	Prioridad
Solución A	0,266666667	0,15	0,1	0,1	0,616666667	1
Solución B	0,133333333	0,15	0,05	0,05	0,383333333	2

Tabla A.14 Selección del material de las columnas

Material Columnas						
Conclusión	Precio	Accesibilidad	Peso	Ensamble	Sumatoria	Prioridad
Solución A	0,066666667	0,1	0,0625	0,05	0,279166667	3
Solución B	0,166666667	0,1	0,0625	0,05	0,379166667	1
Solución C	0,166666667	0,1	0,025	0,05	0,341666667	2

Tabla A.15 Selección del procesador

Procesador						
Conclusión	Precio	Accesibilidad	Peso	Ensamble	Sumatoria	Prioridad
Solución A	0,133333333	0,125	0,05	0,05	0,358333333	2
Solución B	0,133333333	0,125	0,05	0,075	0,383333333	1
Solución C	0,133333333	0,05	0,05	0,025	0,258333333	3

Tabla A.16 Selección de los sensores

Sensores						
Conclusión	Precio	Accesibilidad	Peso	Ensamble	Sumatoria	Prioridad
Solución A	0,133333333	0,15	0,075	0,05	0,408333333	2
Solución B	0,266666667	0,15	0,075	0,1	0,591666667	1

Tabla A.17 Selección del visualizador

Visualizador						
Conclusión	Precio	Accesibilidad	Peso	Ensamble	Sumatoria	Prioridad
Solución A	0,1	0,05	0,025	0,0375	0,2125	3
Solución B	0,2	0,15	0,0625	0,075	0,4875	1
Solución C	0,1	0,1	0,0625	0,0375	0,3	2

Tabla A.18 Selección de los cables de conexión

Cables						
Conclusión	Precio	Accesibilidad	Peso	Ensamble	Sumatoria	Prioridad
Solución A	0,266666667	0,15	0,075	0,075	0,566666667	1
Solución B	0,133333333	0,15	0,075	0,075	0,433333333	2

Tabla A.19 Selección de la base de datos

Base de Datos						
Conclusión	Precio	Accesibilidad	Peso	Ensamble	Sumatoria	Prioridad
Solución A	0,133333333	0,1	0,05	0,05	0,333333333	1=2=3
Solución B	0,133333333	0,1	0,05	0,05	0,333333333	1=2=3
Solución C	0,133333333	0,1	0,05	0,05	0,333333333	1=2=3

Tabla A.20 Tabla de selección del portal web

Portal web						
Conclusión	Precio	Accesibilidad	Peso	Ensamble	Sumatoria	Prioridad
Solución A	0,2	0,15	0,075	0,075	0,5	1=2
Solución B	0,2	0,15	0,075	0,075	0,5	1=2

B. Software

B1. Adquisición y acondicionamiento de la información

Programa 1: Conteo y almacenamiento de los votos electrónicos

```
import RPi.GPIO as GPIO # Librería controlar gpio
import time                # Librería tiempo
import sqlite3             # Librería base de datos
import datetime           # Librería para usar tiempo
import sys                # Librería de acceso a variables del interprete
from functools import partial # Librería de funciones
GPIO.setmode(GPIO.BCM)   # Sirve para usar número de pines de la placa
GPIO.setwarnings(False)  # Se impide que aparezcan warnings
sen=17                    # Número de Pin GPIO
s2=27                    # Número de Pin GPIO
sen2=22                  # Número de Pin GPIO
GPIO.setup(s2,GPIO.IN, GPIO.PUD_UP) #PIN ENTRADA
GPIO.setup(sen,GPIO.IN, GPIO.PUD_UP) #PIN ENTRADA
GPIO.setup(sen2,GPIO.IN, GPIO.PUD_UP) #PIN ENTRADA
tiempo=0.35;
sensor1= partial(GPIO.input, 17) #Variable de congelamiento
sensor2 = partial(GPIO.input, 22) #Variable de congelamiento
```

```

#Crear y conectar con la base de datos

#           Nombre Base de datos
conn = sqlite3.connect('basetesis2.db') #Conexión con la base de datos

c = conn.cursor()                #Acceso a la base de datos
def create_table():                # Creación de la Tabla
    c.execute('CREATE TABLE IF NOT EXISTS stuffToPlot(Pregunta TEXT, Respuesta1 TEXT, Votos1 INTEGER, Respuesta2
TEXT, Votos2 INTEGER, Fecha TEXT)')
while True:
    #Inicializar contadores
    cont1=int(0)
    cont2=int(0)
    #Crear lista de preguntas
    preguntas=[]
    respuestas1=[]
    respuestas2=[]
    y='n'
    while y=='n':
        n=int(input("¿Cuántas preguntas va a ingresar? \n"))
        if n==0:
            print("¿Desea finalizar el programa? Si/No")
            r=input()
            r.lower()
            if r=='si':
                sys.exit()
        for i in range(n):                #Ingreso de preguntas y respuestas
            m=i+1
            pregunta=input("Pregunta \n")
            preguntas.append(pregunta)
            respuesta1=input("Respuesta 1 \n")
            respuestas1.append(respuesta1)
            respuesta2=input("Respuesta 2 \n")
            respuestas2.append(respuesta2)
        print("La lista de preguntas y respuestas es la siguiente\n")

```

```

for k in range(n):
    #Almacenamiento de las preguntas y respuestas
    print(preguntas[k])
    print(respuestas1[k])
    print(respuestas2[k], "\n")
print("Desea Continuar")
y=input("Si/No \n")
y.lower()
if y=='no':
    break
fin=int(time.time()+20)
## print(fin)
## Lectura de las señales de los sensores y almacenamiento de la información en la base de datos
for j in range(n):
    create_table()
    unix = time.time()
    date = str(datetime.datetime.fromtimestamp(unix).strftime('%Y-%m-%d %H:%M:%S'))
    c.execute("INSERT INTO stuffToPlot (Pregunta, Respuesta1, Votos1, Respuesta2, Votos2, Fecha)
              VALUES (?, ?, ?, ?, ?)",(preguntas[j], respuestas1[j], cont1, respuestas2[j], cont2, date))
    time.sleep(0.5)
    conn.commit()
    while GPIO.input(s2):
        print (preguntas[j])
        if sensor1()==0:
            cont1 += 1
            tiempo_st= time.localtime()
            print (respuestas1[j],cont1,time.asctime(tiempo_st))
            c.execute ("UPDATE stuffToPlot SET Votos1 = ? WHERE Pregunta = ?",(cont1,preguntas[j]) )
            conn.commit()
            while sensor1() != 1:
                pass
            time.sleep(0.2)
        if sensor2() == 0:
            cont2 += 1
            tiempo_st2= time.localtime()

```

```

print (respuestas2[j],cont2,time.asctime(tiempo_st2))

c.execute ("UPDATE stuffToPlot SET Votos2 = ? WHERE Pregunta = ?",(cont2,preguntas[j]) )

conn.commit()

while sensor2() != 1:

    pass

    time.sleep(tiempo)

#c.close()

#conn.close()

cont1=0

cont2=0

fin=fin+20

```

B2. Representación de la información

Programa 2: Lectura y presentación en una ventana los resultados de la encuesta

```

import sqlite3                #Librería de base de datos
import time                   #Librería para usar el tiempo
import tkinter                #Librería para usar objetos
from tkinter import ttk       #Librería para usar objetos
from tkinter import *         # Librería para usar objetos
##      Nombre Base de Datos
con=sqlite3.connect('basetesis7.db') #Conexión con la Base de Datos
cur = con.cursor()           #Acceso a la base de datos
print ("La base de datos se abrió correctamente")
def CenterWindow(win, width, height): #Función para centrar la ventana
    win.update_idletasks()
    x=(win.winfo_screenwidth() // 2) - (width // 2)
    y=(win.winfo_screenheight() // 2) - (height // 2)
    return [width, height, x, y]

```

```

def MakeGraph():
    #Función para graficar pasteles
    cur = con.cursor()
    #Acceso a la base de datos
    cur.execute("SELECT * FROM stuffToPlot") #Ejecutar Base de Datos
    for i in cur:
        #Extraer valores de la Base de DATos
        Value= float(i[2])
        Value2= float(i[4])
        Total= Value + Value2
        if Total==0:
            Total=1
        # Canvas Espacio de gráfico
        Graph=Canvas(frame1, width=330, height=315, bg="white")
        Graph.place(x=380, y=120)
        # Graficar
        ValueDeg=360*(Value/Total)
        Arc=Graph.create_arc(25, 15, 305, 300, start=0, extent=ValueDeg, fill="green")
        Rest=Graph.create_arc(25, 15, 305, 300, start=ValueDeg, extent=(360-ValueDeg), fill="orange")
    # Labels
    s=' ' #Variable en blanco para los Labels
    m=' ' #Variable en blanco para los Labels
    TexPreg=Label(frame1, text= "Pregunta:", font=("Verdana",16)).place(x=60, y=50)
    Preg=tkinter.Label(frame1, text=s, font=("Verdana",16)).place(x=200, y=50)
    Preg=tkinter.Label(frame1, text=i[0], font=("Verdana",16)).place(x=200, y=50)
    TexVotos=Label(frame1, text="Resultado de los votos", font=("Verdana",16)).place(x=60, y=140)
    Resp1=Label(frame1, text=m, foreground="green", font=("Verdana",16)).place(x=90, y=190)
    Resp1=Label(frame1, text=i[1], foreground="green", font=("Verdana",16)).place(x=90, y=190)
    Resp2=Label(frame1, text=m, foreground="orange", font=("Verdana",16)).place(x=90, y=255)
    Resp2=Label(frame1, text=i[3], foreground="orange", font=("Verdana",16)).place(x=90, y=255)
    Vot1=Label(frame1, text=m, foreground="green", font=("Verdana",16)).place(x=210, y=190)
    Vot1=Label(frame1, text=i[2], foreground="green", font=("Verdana",16)).place(x=210, y=190)
    Vot2=Label(frame1, text=m, foreground="orange", font=("Verdana",16)).place(x=210, y=255)
    Vot2=Label(frame1, text=i[4], foreground="orange", font=("Verdana",16)).place(x=210, y=255)
    time.sleep(0.7)
    ## Segunda pestaña
def MakeGraph2():

```



```

# Canvas
Graph2=Canvas(frame2, width=104, height=104, bg="white")
Graph2.place(x=100, y=40)

# Labels
InfoLabel=Label(frame2, text=i[0], font=("Verdana",11)).place(x=30, y=10)
ValueLabel=Label(frame2, text=i[1],foreground="green",font=("Verdana",11)).place(x=30, y=40)
TotalLabel=Label(frame2, text=i[3],foreground="orange",font=("Verdana",11)).place(x=30, y=100)
VaUnit=Label(frame2, text=i[2],foreground="green",font=("Verdana",11)).place(x=30, y=60)
ToUnit=Label(frame2, text=i[4],foreground="orange",font=("Verdana",11)).place(x=30, y=120)

Value= float(i[2])
Value2= float(i[4])
Total= Value + Value2
ValueDeg=360*(Value/Total)
Arc=Graph2.create_arc(5, 5, 100, 102, start=0, extent=ValueDeg, fill="green")
Rest=Graph2.create_arc(5, 5, 100, 102, start=ValueDeg, extent=(360-ValueDeg), fill="orange")

def MakeGraph3():
    # Canvas
    Graph3=Canvas(frame2, width=104, height=104, bg="white")
    Graph3.place(x=350, y=40)

    # Labels
    InfoLabel=Label(frame2, text=i[0], font=("Verdana",11)).place(x=280, y=10)
    ValueLabel=Label(frame2, text=i[1],foreground="green",font=("Verdana",11)).place(x=280, y=40)
    TotalLabel=Label(frame2, text=i[3],foreground="orange",font=("Verdana",11)).place(x=280, y=100)
    VaUnit=Label(frame2, text=i[2],foreground="green",font=("Verdana",11)).place(x=280, y=60)
    ToUnit=Label(frame2, text=i[4],foreground="orange",font=("Verdana",11)).place(x=280, y=120)

    Value= float(i[2])
    Value2= float(i[4])
    Total= Value + Value2
    ValueDeg=360*(Value/Total)
    Arc=Graph3.create_arc(5, 5, 100, 102, start=0, extent=ValueDeg, fill="green")
    Rest=Graph3.create_arc(5, 5, 100, 102, start=ValueDeg, extent=(360-ValueDeg), fill="orange")

def MakeGraph4():
    # Canvas

```

```

Graph4=Canvas(frame2, width=104, height=104, bg="white")
Graph4.place(x=620, y=40)
# Labels
InfoLabel=Label(frame2, text=i[0], font=("Verdana",11)).place(x=550, y=10)
ValueLabel=Label(frame2, text=i[1],foreground="green",font=("Verdana",11)).place(x=550, y=40)
TotalLabel=Label(frame2, text=i[3],foreground="orange",font=("Verdana",11)).place(x=550, y=100)
VaUnit=Label(frame2, text=i[2],foreground="green",font=("Verdana",11)).place(x=550, y=60)
ToUnit=Label(frame2, text=i[4],foreground="orange",font=("Verdana",11)).place(x=550, y=120)
Value= float(i[2])
Value2= float(i[4])
Total= Value + Value2
ValueDeg=360*(Value/Total)
Arc=Graph4.create_arc(5, 5, 100, 102, start=0, extent=ValueDeg, fill="green")
Rest=Graph4.create_arc(5, 5, 100, 102, start=ValueDeg, extent=(360-ValueDeg), fill="orange")
def MakeGraph5():

# Canvas
Graph5=Canvas(frame2, width=104, height=104, bg="white")
Graph5.place(x=100, y=210)
# Labels
InfoLabel=Label(frame2, text=i[0], font=("Verdana",11)).place(x=30, y=180)
ValueLabel=Label(frame2, text=i[1],foreground="green",font=("Verdana",11)).place(x=30, y=210)
TotalLabel=Label(frame2, text=i[3],foreground="orange",font=("Verdana",11)).place(x=30, y=270)
VaUnit=Label(frame2, text=i[2],foreground="green",font=("Verdana",11)).place(x=30, y=230)
ToUnit=Label(frame2, text=i[4],foreground="orange",font=("Verdana",11)).place(x=30, y=290)
Value= float(i[2])
Value2= float(i[4])
Total= Value + Value2
ValueDeg=360*(Value/Total)
Arc=Graph5.create_arc(5, 5, 100, 102, start=0, extent=ValueDeg, fill="green")
Rest=Graph5.create_arc(5, 5, 100, 102, start=ValueDeg, extent=(360-ValueDeg), fill="orange")
def MakeGraph6():

# Canvas
Graph6=Canvas(frame2, width=104, height=104, bg="white")

```

```

Graph6.place(x=350, y=210)

# Labels
InfoLabel=Label(frame2, text=i[0], font=("Verdana",11)).place(x=280, y=180)
ValueLabel=Label(frame2, text=i[1],foreground="green",font=("Verdana",11)).place(x=280, y=210)
TotalLabel=Label(frame2, text=i[3],foreground="orange",font=("Verdana",11)).place(x=280, y=270)
VaUnit=Label(frame2, text=i[2],foreground="green",font=("Verdana",11)).place(x=280, y=230)
ToUnit=Label(frame2, text=i[4],foreground="orange",font=("Verdana",11)).place(x=280, y=290)

Value= float(i[2])
Value2= float(i[4])
Total= Value + Value2
ValueDeg=360*(Value/Total)
Arc=Graph6.create_arc(5, 5, 100, 102, start=0, extent=ValueDeg, fill="green")
Rest=Graph6.create_arc(5, 5, 100, 102, start=ValueDeg, extent=(360-ValueDeg), fill="orange")
def MakeGraph7():
    # Canvas
    Graph7=Canvas(frame2, width=104, height=104, bg="white")
    Graph7.place(x=620, y=210)
    # Labels
    InfoLabel=Label(frame2, text=i[0], font=("Verdana",11)).place(x=550, y=180)
    ValueLabel=Label(frame2, text=i[1],foreground="green",font=("Verdana",11)).place(x=550, y=210)
    TotalLabel=Label(frame2, text=i[3],foreground="orange",font=("Verdana",11)).place(x=550, y=270)
    VaUnit=Label(frame2, text=i[2],foreground="green",font=("Verdana",11)).place(x=550, y=230)
    ToUnit=Label(frame2, text=i[4],foreground="orange",font=("Verdana",11)).place(x=550, y=290)
    Value= float(i[2])
    Value2= float(i[4])
    Total= Value + Value2
    ValueDeg=360*(Value/Total)
    Arc=Graph7.create_arc(5, 5, 100, 102, start=0, extent=ValueDeg, fill="green")
    Rest=Graph7.create_arc(5, 5, 100, 102, start=ValueDeg, extent=(360-ValueDeg), fill="orange")
def MakeGraph8():
    # Canvas
    Graph8=Canvas(frame2, width=104, height=104, bg="white")
    Graph8.place(x=100, y=380)
    # Labels

```

```

InfoLabel=Label(frame2, text=i[0], font=("Verdana",11)).place(x=30, y=350)
ValueLabel=Label(frame2, text=i[1],foreground="green",font=("Verdana",11)).place(x=30, y=380)
TotalLabel=Label(frame2, text=i[3],foreground="orange",font=("Verdana",11)).place(x=30, y=440)
VaUnit=Label(frame2, text=i[2],foreground="green",font=("Verdana",11)).place(x=30, y=400)
ToUnit=Label(frame2, text=i[4],foreground="orange",font=("Verdana",11)).place(x=30, y=460)
Value= float(i[2])
Value2= float(i[4])
Total= Value + Value2
ValueDeg=360*(Value/Total)
Arc=Graph8.create_arc(5, 5, 100, 102, start=0, extent=ValueDeg, fill="green")
Rest=Graph8.create_arc(5, 5, 100, 102, start=ValueDeg, extent=(360-ValueDeg), fill="orange")
def MakeGraph9():
    # Canvas
    Graph9=Canvas(frame2, width=104, height=104, bg="white")
    Graph9.place(x=350, y=380)
    # Labels
    InfoLabel=Label(frame2, text=i[0], font=("Verdana",11)).place(x=280, y=350)
    ValueLabel=Label(frame2, text=i[1],foreground="green",font=("Verdana",11)).place(x=280, y=380)
    TotalLabel=Label(frame2, text=i[3],foreground="orange",font=("Verdana",11)).place(x=280, y=440)
    VaUnit=Label(frame2, text=i[2],foreground="green",font=("Verdana",11)).place(x=280, y=400)
    ToUnit=Label(frame2, text=i[4],foreground="orange",font=("Verdana",11)).place(x=280, y=460)
    Value= float(i[2])
    Value2= float(i[4])
    Total= Value + Value2
    ValueDeg=360*(Value/Total)
    Arc=Graph9.create_arc(5, 5, 100, 102, start=0, extent=ValueDeg, fill="green")
    Rest=Graph9.create_arc(5, 5, 100, 102, start=ValueDeg, extent=(360-ValueDeg), fill="orange")
def MakeGraph10():
    # Canvas
    Graph10=Canvas(frame2, width=104, height=104, bg="white")
    Graph10.place(x=620, y=380)
    # Labels
    InfoLabel=Label(frame2, text=i[0], font=("Verdana",11)).place(x=550, y=350)

```

```

ValueLabel=Label(frame2, text=i[1],foreground="green",font=("Verdana",11)).place(x=550, y=380)
TotalLabel=Label(frame2, text=i[3],foreground="orange",font=("Verdana",11)).place(x=550, y=440)
VaUnit=Label(frame2, text=i[2],foreground="green",font=("Verdana",11)).place(x=550, y=400)
ToUnit=Label(frame2, text=i[4],foreground="orange",font=("Verdana",11)).place(x=550, y=460)
Value= float(i[2])
Value2= float(i[4])
Total= Value + Value2
ValueDeg=360*(Value/Total)
Arc=Graph10.create_arc(5, 5, 100, 102, start=0, extent=ValueDeg, fill="green")
Rest=Graph10.create_arc(5, 5, 100, 102, start=ValueDeg, extent=(360-ValueDeg), fill="orange")
## Crear ventana con pestañas
mGui=tkinter.Tk()
mGui2=ttk.Notebook(mGui)
mGui2.pack(fill='both', expand='yes')
frame1 = ttk.Frame(mGui2)
frame2 = ttk.Frame(mGui2)
mGui2.add(frame1, text='ENCUESTA ACTUAL')
mGui2.add(frame2, text='HISTORIAL ENCUESTAS')
ValueObj=IntVar()
TotalObj=IntVar()
WinWidth=800
WinHeight=550
mGui.geometry("{0[0]}x{0[1]}+{0[2]}+{0[3]}".format(CenterWindow(mGui, WinWidth, WinHeight)))
mGui.deiconify()
mGui.title("RESULTADOS DE ENCUESTAS")
while True:
    #Ciclo para graficar todos los pasteles y actualizarlo cada cierto tiempo
    x= -1
    i=0
    cur.execute("SELECT * FROM stuffToPlot")
    MakeGraph()
    for i in cur:
        x+=1
        if x==0:
            MakeGraph2()

```

```
mGui.update_idletasks()
mGui.update()
if x==1:
    MakeGraph3()
    mGui.update_idletasks()
    mGui.update()
if x==2:
    MakeGraph4()
    mGui.update_idletasks()
    mGui.update()
if x==3:
    MakeGraph5()
    mGui.update_idletasks()
    mGui.update()
if x==4:
    MakeGraph6()
    mGui.update_idletasks()
    mGui.update()
if x==5:
    MakeGraph7()
    mGui.update_idletasks()
    mGui.update()
if x==6:
    MakeGraph8()
    mGui.update_idletasks()
    mGui.update()
if x==7:
    MakeGraph9()
    mGui.update_idletasks()
    mGui.update()
if x==8:
    MakeGraph10()
    mGui.update_idletasks()
    mGui.update()
```

```
#time.sleep(0.3)
x=0
mGui.update_idletasks()
mGui.update()
time.sleep(2)
```