

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“Diseñar e Implementar un dispositivo para mostrar anuncios recibidos mediante correo electrónico, utilizando hardware de bajo costo y software libre”

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Nombre de la titulación

Licenciados en Redes y Sistemas Operativos

Presentado por:

Julissa Andrea Alfonzo Figueroa

Alfonso Javier Bayona García

GUAYAQUIL - ECUADOR

Año: 2019

DEDICATORIA

El presente proyecto lo dedico a mis padres, quienes han sido el pilar fundamental tanto para mi crecimiento profesional y personal, a mis profesores, quienes fueron los guías de mi desarrollo profesional durante estos años y a mi compañero de tesis, que sin su apoyo no hubiera sido posible lograr esta meta.

Julissa Andrea Alfonzo Figueroa.

El presente proyecto lo dedico a mi familia, quienes han sido el pilar fundamental para poder salir adelante con mis estudios; a mis profesores, por haber sido los guías de mi desarrollo profesional durante todos estos años y a mi compañera de tesis, que sin su apoyo no hubiera sido posible lograr esta meta.

Alfonso Javier Bayona García.

AGRADECIMIENTOS

Mi más sincero agradecimiento a Dios que me dio sabiduría para cumplir con éxitos mi meta, a mis padres, que siempre estuvieron presentes para brindarme su apoyo durante el proceso de mi formación académica, a mis primas y amigas más cercanas, quienes mediante su apoyo incondicional me dieron aliento para persistir y nunca desistir.

Julissa Andrea Alfonzo Figueroa.

Mi más sincero agradecimiento a Dios por permitirme cumplir esta meta, a mis padres por brindarme su apoyo incondicional durante toda mi formación académica, a mis tíos, quienes siempre estuvieron pendientes de ayudarnos a mí y a mis hermanos a salir adelante con nuestros estudios, a mis amigos y compañeros.

Alfonso Javier Bayona García.

DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Julissa Alfonzo y Alfonso Bayona* damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Julissa Andrea Alfonzo Figueroa

Alfonso Javier Bayona García

EVALUADORES

Robert Stalin Andrade Troya

PROFESOR DE LA MATERIA

Albert Giovanny Espinal Santana

PROFESOR TUTOR

RESUMEN

El Departamento de Soporte Técnico pertenece a la Facultad de Ingeniería en Electricidad y Computación en la Escuela Superior Politécnica del Litoral, se encarga de la creación y difusión de anuncios de eventos mediante correo electrónico, con la finalidad de mantener informado a la comunidad politécnica sobre las actividades que se realizarán en la facultad.

En el presente documento se detallan los resultados obtenidos gracias al uso de las herramientas que posee la metodología de Design Thinking. De esta manera, se adquirió toda la información necesaria, para definir la problemática existente. Se analizó y comprendió que los estudiantes no acostumbran a revisar su correo constantemente, perdiendo oportunidades en el desarrollo de su formación académica. Por ende, se diseñó un sistema con hardware de bajo costo y software libre, que permita publicar los anuncios de eventos que son enviado por correo electrónico, usando las redes sociales como un medio de difusión.

Se elaboró un prototipo de bajo nivel, que cumplía con los requerimientos del cliente, y sirvió para evaluar la viabilidad del sistema, de esta manera se obtuvo una retroalimentación para llegar al resultado final de la propuesta planteada.

El presente proyecto no solo ayuda a la automatización de un proceso de publicaciones de eventos, sino también a los estudiantes, quienes mediante el uso frecuente que les dan a las redes sociales, podrán mantenerse informados de las actividades que se realizarán en la facultad.

Palabras Clave: correo electrónico, Design Thinking, hardware bajo costo, software libre, automatización.

ABSTRACT

The Technical Support Department belongs to the Faculty of Electrical and Computer Engineering at the Polytechnic School of the Coast, is responsible for creating and disseminating event announcements by email, in order to keep the polytechnic community informed about activities to be carried out in the faculty.

This document details the results obtained thanks to the use of the tools that the Design Thinking methodology has. In this way, all the necessary information was acquired to define the existing problem. It was analyzed and understood that students do not usually check their mail constantly, losing opportunities in the development of their academic training. Therefore, a system with low-cost hardware and free software was designed, which allows advertising of events that are sent by email to be published, using social networks as a means of dissemination.

A low-level prototype was developed, which complied with the client's requirements, and served to assess the viability of the system, in this way a feedback was obtained to reach the result of the proposed proposal.

This project not only helps the automation of an event publishing process, but also the students, who through frequent use of social networks, can keep informed of the activities that will be carried out in the faculty.

Keywords: *email, Design Thinking, low cost hardware, free software, automation.*

ÍNDICE GENERAL

RESUMEN	I
ABSTRACT	II
ÍNDICE GENERAL	III
ABREVIATURAS.....	V
SIMBOLOGÍA.....	VI
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS.....	VIII
CAPÍTULO 1	1
1. INTRODUCCIÓN.....	1
1.1. Descripción del problema	2
1.2. Objetivos	2
1.2.1 Objetivo General	2
1.2.2 Objetivos Específicos.....	3
CAPÍTULO 2	6
2.1 METODOLOGÍA.....	6
2.1.1 Fase Empatizar	6
2.1.2 Fase Definir.....	7
2.1.3 Fase Idear	8
2.1.4 Fase Prototipar.....	9
2.1.5 Fase Testear	10
2.2 DISEÑO Y DESARROLLO DE LA SOLUCIÓN.....	12
2.2.1 Diseño de la solución	12
2.2.1.1 Componentes del dispositivo de la solución.....	14
2.2.2 Desarrollo de la solución	16
2.2.2.2 Proceso de Búsqueda de correo	22
2.2.2.3 Proceso de Aplicación de filtros	22

2.2.2.4 Proceso Almacenamiento de correo	23
2.2.2.5 Proceso de Publicación de contenido	24
2.2.2.6 Proceso de Creación de registro	27
CAPÍTULO 3	28
3. PLAN DE IMPLEMENTACIÓN Y PRESUPUESTO.....	28
CAPÍTULO 4	30
4. CONCLUSIONES Y RECOMENDACIONES	30
Conclusiones.....	30
Recomendaciones.....	31
BIBLIOGRAFÍA	32
APÉNDICES.....	34

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
CDP	Centro de Difusión y Publicaciones
FIEC	Facultad de Ingeniería en Electricidad y Computación
FIMCP	Facultad de Ingeniería Mecánica y Ciencias de la Producción
FCSH	Facultad de Ciencias Sociales y Humanística
DST	Departamento de Soporte Técnico
USB	Universal Serial Bus
HDMI	High Definition Multimedia Interface
MicroSD	Micro Secure Digital
MicroUSB	Micro Universal Serial Bus
NOOBS	New Out Of Box Software
IVA	Impuesto al Valor Agregado

SIMBOLOGÍA

%	Por ciento
\$	Dólar Americano
V	Voltios
A	Amperios
GHz	Gigahercios
GB	Gigabytes

ÍNDICE DE FIGURAS

Figura 2.1 Mapa de empatía de la Ing. Katherine Campos, miembro del DST.	7
Figura 2.2 Prototipo de bajo nivel del sistema de categorización de anuncios.....	10
Figura 2.3 Esquema de la solución planteada.....	13
Figura 2.4 Publicación de los anuncios en las redes sociales.....	13
Figura 2.5 Comunidad politécnica y las redes sociales.....	14
Figura 2.6 Partes de la Raspberry Pi Modelo B.	15
Figura 2.7 Estuche protector del dispositivo.....	15
Figura 2.8 Versión y detalles del Sistema Operativo Raspbian.....	16
Figura 2.9 Diagrama de flujo del sistema.	17
Figura 2.10 Interfaz Gráfica de inicio del sistema.....	18
Figura 2.11 Menú principal del sistema.....	18
Figura 2.12 Configuración de cuenta de correo del dispositivo y usuario.....	19
Figura 2.13 Configuración de un nuevo usuario de Facebook.	20
Figura 2.14 Manejo de direcciones de correos permitidos.	21
Figura 2.15 Visualización del registro de publicaciones realizadas.....	21
Figura 2.16 Uso del módulo Imaplib para buscar un nuevo correo.	22
Figura 2.17 Formato para realizar una publicación.	23
Figura 2.18 Almacenamiento de la información en el dispositivo.	23
Figura 2.19 Notificación de error a la hora de almacenar un correo.....	24
Figura 2.20 Uso de Selenium y PhantomJs para acceder a Facebook.	25
Figura 2.21 Uso de la librería InstagramAPI para realizar la publicación.	25
Figura 2.22 Diagrama de flujo para la publicación de contenido.....	26
Figura 2.23 Código usado para la creación de registros.	27
Figura 3.1 Plan de implementación y presupuesto de la solución desarrollada.	28

ÍNDICE DE TABLAS

Tabla 1.1 Referencias de trabajos relacionados a la problemática planteada.....	3
Tabla 2.1 Ideas seleccionadas de la ‘Lluvia de Ideas’.....	8
Tabla 2.2 Matriz de Decisión del problema a solucionar.	9
Tabla 3.1 Costo de componentes para la creación del dispositivo.....	29
Tabla 3.2 Costo de servicios para la implementación de la solución.	29

CAPÍTULO 1

1. INTRODUCCIÓN

La Escuela Superior Politécnica del Litoral (ESPOL) es una institución pública de educación superior, tiene el prestigio de formar profesionales de calidad, con valores éticos y habilidades para desenvolverse con eficiencia en el mercado laboral.

La ESPOL cuenta con diversos campos de conocimiento, como: Ciencias e Ingeniería, Administración, Economía, Arte y Diseño y Humanidades.

El Centro de Difusión y Publicaciones (CDP) de la institución, se encarga de mantener informados a la comunidad politécnica sobre las distintas actividades que se realizan. Entre ellos, promocionar proyectos de investigación, maestrías que se dictan según la unidad académica, entre otros.

La Facultad de Ingeniería en Electricidad y Computación (FIEC) promueve el uso diario de las tecnologías, para la investigación e innovación de nuevas y mejores soluciones a problemas existentes o necesidades futuras.

Según el informe ‘Rendición de Cuentas’ del 2018, la ESPOL contó con 10.260 estudiantes matriculados, de los cuales, el 23% son de la FIEC, seguido de la Facultad de Ingeniería Mecánica y Ciencia de la Producción (FIMCP) y la Facultad de Ciencias Sociales y Humanísticas (FCSH), ambas con el 18% [1].

El correo como la página web son herramientas tecnológicas, usadas diariamente para dar a conocer las actividades que se realizarán en la universidad a nivel general o en cada facultad, y que pueden ser de interés para los estudiantes.

Tanto la ESPOL como las demás unidades académicas, llevan a cabo continuos programas semanales, sean charlas o eventos, que le permiten a los estudiantes de las distintas carreras, fortalecer sus conocimientos académicos, ofrecerles oportunidades de pasantías profesionales y ofertas de trabajo.

Cuenta también con programas institucionales de apoyo a los clubes y a las asociaciones estudiantiles, para poder inducir a los estudiantes a que realicen actividades extracurriculares, y que ellos puedan desarrollar sus habilidades de trabajo en equipo, inclusive de liderazgo.

1.1. Descripción del problema

El Departamento de soporte técnico (DST), formado por asistentes de la FIEC, es la encargada de ofrecer apoyo tecnológico a profesores y estudiantes, con el fin de aprovechar al máximo los recursos con los que cuenta la facultad.

El DST lleva a cabo múltiples actividades diarias, como el préstamo de equipos tecnológicos, soporte técnico a docentes o personal administrativo, la administración del correo de la FIEC, entre muchas otras.

El departamento se encarga de crear y enviar los anuncios, como: charlas, conferencias, oportunidades de pasantías profesionales y ofertas laborales, mediante correo electrónico y de esta manera, informar a los estudiantes de la facultad sobre las actividades que se realizarán durante el día o la semana.

A menudo, los estudiantes no se llegan a informar de las actividades que se realizan, la mayoría no leen los correos que son enviados a diario y muchos otros no tienen configurado la cuenta de correo en su dispositivo móvil, por ende, es poca la asistencia a charlas o eventos que se realizan en la facultad, y que pueden ser de beneficio para el desarrollo académico del estudiante.

El envío masivo y repetitivo de correos, que en muchas ocasiones no son afín al estudiante o su carrera, es una de las causas por la que pierden el interés en leer los correos, desaprovechando así, oportunidades de pasantías profesionales y ofertas laborales.

1.2. Objetivos

1.2.1 Objetivo General

Crear un sistema para mostrar los anuncios de eventos enviados por correo electrónico, usando las redes sociales como medio de difusión, con la finalidad de dar a conocer a los estudiantes las actividades que se realizan durante la semana, en las distintas facultades de la institución de educación superior.

1.2.2 Objetivos Específicos

- Considerar la mejor opción para proporcionar la solución a la problemática planteada.
- Mostrar los anuncios de una manera dinámica para que los estudiantes se informen de las actividades diarias que ocurren en las facultades.
- Obtener un aumento en la asistencia a charlas y eventos que se realizan en las facultades.
- Desarrollar un sistema que despierte el interés para buscar eventos por parte de los estudiantes.

1.3 Marco Teórico

Tabla 1.1 Referencias de trabajos relacionados a la problemática planteada.

#	Título	Autor	Institución	Año
1	Diseño e Implementación de un Dispositivo de Señalización Digital utilizando hardware de bajo costo y software libre. [2]	Gabriela Ninoska Beltrán Triviño Lenin Eliezer Espinal Ordoñez	ESPOL	2018
2	Análisis, diseño e implementación de una red social para ESPOL Digital utilizando herramientas 2.0. [3]	Lissette E. Espín Noboa	ESPOL	2010
3	Estudio de la utilidad de Redes Sociales para difundir el Potencial Turístico De las comunas de Ayangue, Cadeate y Libertador Bolívar. [4]	Suarez Ricardo Julissa Elizabeth	ESPOL	2015

Caso #1: Se efectuó un estudio para implementar una solución innovadora, con el propósito de difundir la información de los cursos ofrecidos por la Academia Cisco ESPOL. El resultado fue la creación de un dispositivo de señalización digital, en el cual fue usado una placa Raspberry Pi como hardware y el sistema operativo Raspbian, que es una

distribución de GNU/Linux basado en Debian, como software. Mediante un aplicativo se administra el contenido multimedia almacenado en una memoria USB. El contenido es mostrado en una pantalla o monitor conectado al dispositivo por el puerto HDMI.

De esta manera fue posible que los interesados puedan visualizar la información de los cursos ofrecidos, optimizando el tiempo de publicación y reduciendo el costo de imprimir volantes.

Caso #2: El trabajo realizado, se basó en el análisis y diseño de un portal web llamada ‘miESPOL’ de código abierto, para ofrecer los servicios de la ESPOL como el SIDWeb, el CENACAD, el Académico, etc.

Les permitía a los estudiantes crear asociaciones o interactuar con otros usuarios dentro del portal.

Se hicieron análisis para establecer la mejor opción para el desarrollo del portal web, creando una arquitectura que integraba los servicios más usados por los estudiantes y que ellos puedan acceder desde cualquier parte usando su dispositivo móvil.

Caso #3: La investigación realizada, abarca el uso de las redes sociales, como medio para dar a conocer los atractivos turísticos de algunas comunidades de la provincia de Santa Elena, y así llegar a un mayor número de turistas.

La estrategia mencionada en el documento es la capacitación sobre el manejo de las redes sociales, para poder difundir el producto o servicio que posea la persona dueña del establecimiento.

En las encuestas mencionadas en la investigación, dan como resultado que, las personas desconocen las ventajas del uso de las redes sociales dentro de un negocio, ni cómo aprovechar para incrementar la clientela.

Luego de analizar los casos planteados anteriormente, concluimos que:

- La señalización digital nos permite promocionar un producto en específico, pero es necesario que el interesado se encuentre frente al dispositivo donde se proyecta el anuncio.
- Los servicios que ofrece la Espol, como el SIDWeb o el Académico, fueron diseñados para ser usados en un ambiente laboral o de oficina, por esta razón, a los estudiantes no les llama la atención revisarlos, sin embargo, lo hacen por cuestiones de subir tareas y mantenerse informados de sus calificaciones o su avance académico.
- Las redes sociales son herramientas tecnológicas que nos permite difundir información de interés, y la ventaja de poder llegar a un mayor número de personas.

De esta forma se plantea, usar las redes sociales como medio de difusión para propagar los anuncios que son enviados por correo electrónico, y así informar a los estudiantes sobre las actividades que se realizarán en la institución.

CAPÍTULO 2

2.1 METODOLOGÍA

Para llevar a cabo el presente proyecto empleamos las distintas herramientas de la metodología de Design Thinking, de esta manera pudimos analizar la problemática planteada y hallar las posibles soluciones para satisfacer las necesidades del cliente. Detallaremos a continuación lo realizado en cada fase.

2.1.1 Fase Empatizar

En la primera fase, se realizó una “Investigación de escritorio” con la finalidad de conocer un poco más sobre la facultad, y a lo que se dedica el DST. Con esta herramienta obtuvimos información relevante y profunda sobre las actividades que realizan, y los servicios que aportan a la facultad.

Esto nos permitió realizar un “Mapa de Actores” (Apéndice A), así logramos definir los grupos de usuarios relacionados a nuestro proyecto.

Después de haber obtenido toda la información necesaria, procedimos a realizar una “Lluvia de Ideas”, obteniendo así, un banco preguntas (Apéndice B) para la realización de entrevistas, tanto para los estudiantes como para el personal del DST. Se elaboró también una encuesta (Apéndice C), la cual se distribuyó a más estudiantes para ratificar la información obtenida en las entrevistas.

Luego de haber culminado con la recopilación de datos, se elaboró un “Mapa de empatía” de los principales actores. En la figura 2.1 podemos observar el mapa de empatía de la ingeniera Katherine Campos, encargada de la administración de red y servidores de la FIEC, en la cual se resume lo que piensa y siente frente a la problemática. El mapa de empatía de los demás actores los podemos encontrar en el Apéndice D.

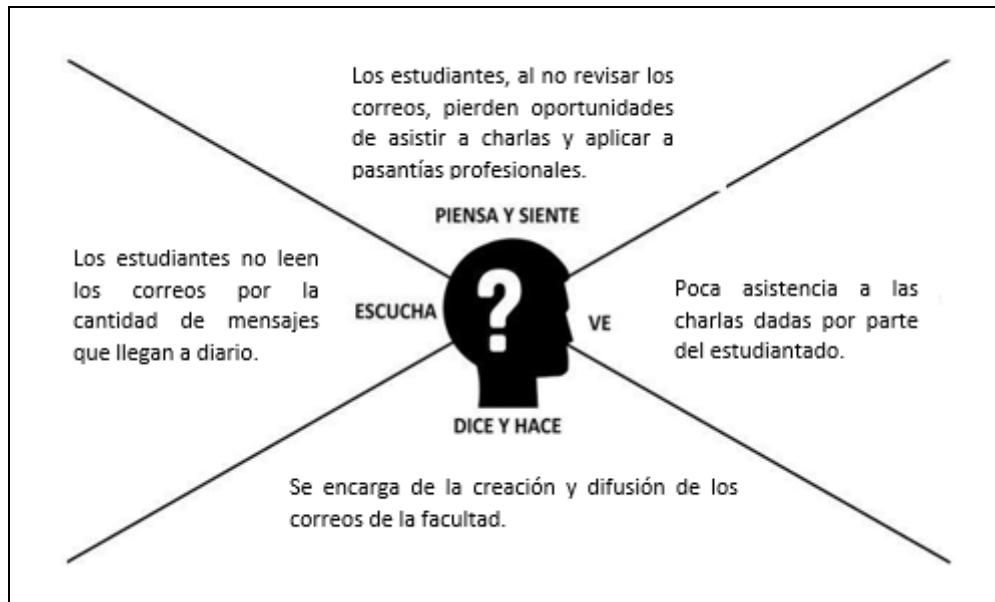


Figura 2.1 Mapa de empatía de la Ing. Katherine Campos, miembro del DST.

2.1.2 Fase Definir

En la segunda fase, luego de analizar la información obtenida, y con el uso de la herramienta “Árbol de problemas” (Apéndice E), logramos precisar las causas y consecuencias de la problemática existente dentro de la comunidad politécnica.

Mediante esta herramienta, llegamos a comprender las causas por las que se produce la problemática, entre ellas:

- El envío masivo, repetitivo de anuncios por correo y que muchas veces no son del interés del estudiante.
- No todas las facultades cuentan con un correo dedicado.
- El no tener configurado la cuenta de correo en el dispositivo móvil.
- Falta de un mejor sistema para publicar anuncios.

2.1.3 Fase Idear

Después de haber definido las causas y consecuencias, se realizó una “Lluvia de ideas” en base a la problemática previamente planteada, con el objetivo de considerar las posibles soluciones que satisfagan las necesidades y poder solucionarlas. A continuación, en la tabla 2.1 mostramos las ideas de posibles soluciones a la problemática.

Tabla 2.1 Ideas seleccionadas de la ‘Lluvia de Ideas’.

Ideas	Alternativa
Sistema que filtre los anuncios que son enviados por correo electrónico.	A
Sistema de comunicación visual para mostrar eventos de interés para el estudiante.	B
Establecer un servidor de correo único para la creación y difusión de anuncios por cada facultad.	C
Sistema que categorice los anuncios que son enviados por correo en base a los gustos del estudiante.	D
Sistema que categorice los anuncios que son enviados por correo electrónico, usando un portal web como medio para difundirlos.	E

La tabla 2.2 muestra la “Matriz de Decisión” del problema a solucionar, en la cual se evaluaron las ideas establecidas anteriormente como alternativa, en base a un puntaje.

Tabla 2.2 Matriz de Decisión del problema a solucionar.

Necesidades	Alternativas				
	A	B	C	D	E
Mostrar los últimos anuncios de eventos.		1			1
Uso del correo electrónico.	1		1	1	1
Poder leer las publicaciones en cualquier parte.			1	1	1
Mostrar una descripción del evento.		1	1	1	1
Facilidad de implementación.	1	1		1	1
Difundir anuncios mediante categorías o etiquetas.			1	1	1
Automatizar la publicación de anuncios.					1
Total	2	3	4	5	7

2.1.4 Fase Prototipar

Se procedió a desarrollar un prototipo de bajo nivel para ser mostrado a los encargados del DST, siendo lo más práctico para cubrir las necesidades del usuario. Para poder resolver el problema principal, se pensó en el uso de un dispositivo de bajo costo y el diseño de un sistema que nos permita almacenar y mostrar los anuncios que son enviados al correo electrónico, como se muestra en la figura 2.2.

El sistema realizaba una extracción de la información, tanto de la imagen como texto adjunto de cada mensaje enviado a la cuenta de correo establecida en el dispositivo, luego se almacenaba y categorizaba en base al asunto y hora de envío del mensaje. El dispositivo mostraba, en una pantalla externa, ubicada en un pasillo de la facultad, cada anuncio

almacenado previamente y así los estudiantes podían observar e informarse de los últimos eventos o actividades mientras se dirigían a su aula de clases. Cada anuncio era mostrado en un orden secuencial y por categorías. El contenido al mismo tiempo era publicado en un portal web, de esta manera cualquier estudiante mediante un dispositivo con acceso a internet tenía la opción de acceder a la página creada y buscar los anuncios que más le llamaran la atención. Dependiendo del dispositivo de donde se accediera a la página, era posible mostrar o quitar opciones como permitir descargar el anuncio o crear un recordatorio del evento.



Figura 2.2 Prototipo de bajo nivel del sistema de categorización de anuncios.

2.1.5 Fase Testear

El prototipo explicado en la fase anterior fue presentado a los integrantes que conforman el DST, los mismo que evaluaron la propuesta presentada, con la finalidad de registrar sus impresiones acerca de su nivel satisfacción respecto a la eficacia de la solución, resolver sus inquietudes sobre el uso del prototipo y recolectar sus ideas de mejoras.

Gracias a esta retroalimentación, se logró establecer que, en lugar de mostrar los anuncios mediante una pantalla en la facultad o por una página web, se use las redes sociales como medio para difundir esta información.

El motivo del cambio se debía a que el estudiante tenía que permanecer observando una pantalla hasta que cada anuncio fuera mostrado, y al igual que el correo electrónico, no se genera un interés en acceder a la página web de los anuncios.

Es por lo que se consideró usar las redes sociales para mostrar los anuncios enviados al correo electrónico, en el momento que el estudiante se encuentre revisando su perfil, pueda enterarse de las actividades que se realizan o se realizaran en la universidad.

Mediante el informe ‘Ecuador Estado Digital’ realizado en el 2019, encontramos que en el Ecuador existen 13,8 millones de usuarios de internet que hacen uso de las redes sociales, siendo Facebook la red social que predomina dentro del área, con 12 millones del total de usuarios conectados, seguido de Twitter con 4 millones del total, e Instagram con 3.9 millones de usuarios conectados [5].

En base a la información obtenida, nuestra solución se enfocará en las publicaciones automáticas de eventos en estas tres redes sociales.

2.2 DISEÑO Y DESARROLLO DE LA SOLUCIÓN

2.2.1 Diseño de la solución

La solución propuesta va a ser diseñada con recursos de bajo costo para probar la eficiencia de la solución. Una de las principales ventajas del dispositivo, es automatizar el proceso de publicación de los eventos, usando las distintas redes sociales como medio de difusión.

En esta sección se describe los detalles de la solución planteada, el diseño del dispositivo para mostrar anuncios enviados por correo electrónico, que está basado en el uso de un hardware de bajo costo y software libre. En la figura 2.3 se detalla por partes el esquema de la solución.

El DST diariamente realiza la creación y difusión de anuncios que luego son envían por correo electrónico a la comunidad politécnica. Los encargados del departamento reciben, de las demás unidades, correos con información en texto de las actividades que quieren promocionar, así en la primera estación de trabajo (1) se crea el arte o imagen del anuncio, luego la segunda estación de trabajo (2) se encargará de enviar un mensaje a la dirección correo usada por el dispositivo, en el que se debe especificar el título de la publicación, una pequeña descripción, etiquetas relacionadas al evento, las redes sociales a las que se desea publicar y la imagen o arte de la publicación. El mensaje llegará al servidor de correo (3) desde donde el dispositivo (4), que se encuentra en el DST, consulta si existe algún nuevo mensaje en el buzón, y de encontrarlo, realiza la extracción de la información, si y solo si, el mensaje cuenta tanto con una imagen como la descripción detallada de la publicación que se desea realizar.

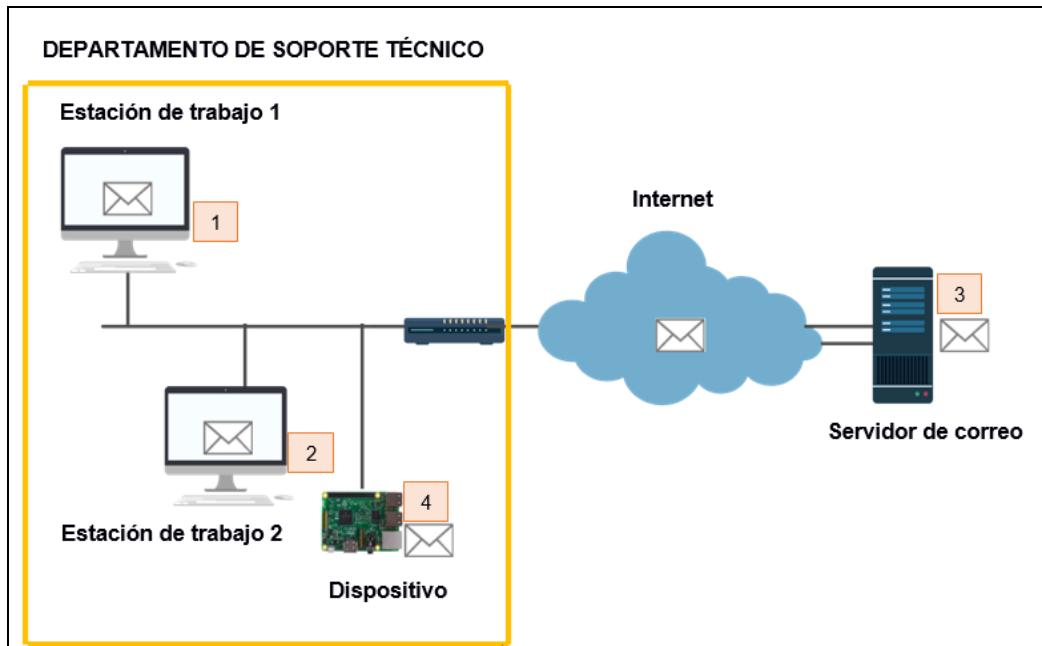


Figura 2.3 Esquema de la solución planteada.

Como se muestra en la figura 2.4, luego de almacenar la información del correo desde el servidor, el dispositivo (4) realizará la publicación automática del anuncio almacenado, a las distintas cuentas de redes sociales (5) que previamente tenga configurado.

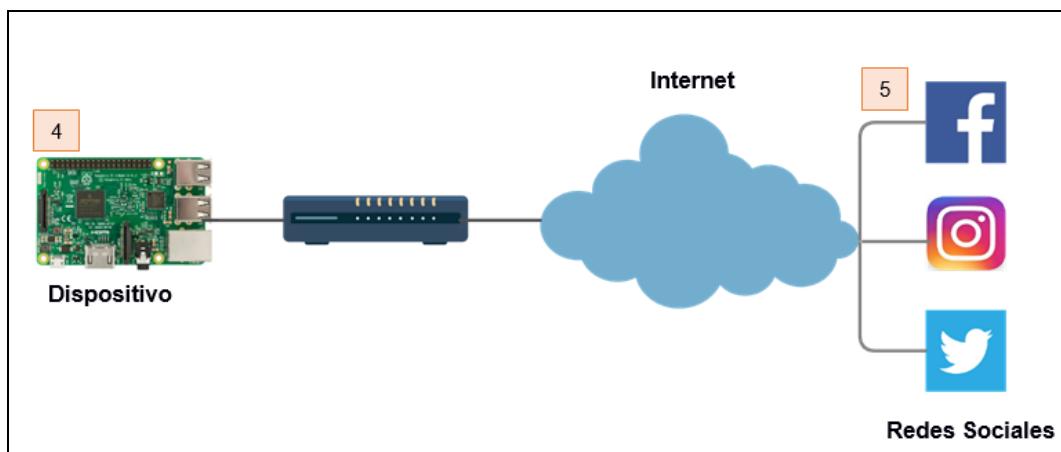


Figura 2.4 Publicación de los anuncios en las redes sociales.

Los estudiantes y demás miembros que conforman la comunidad politécnica, en el momento que accedan desde cualquier dispositivo a sus cuentas de redes sociales, se informarán de las publicaciones hechas por el dispositivo, como se muestra en la figura 2.5.

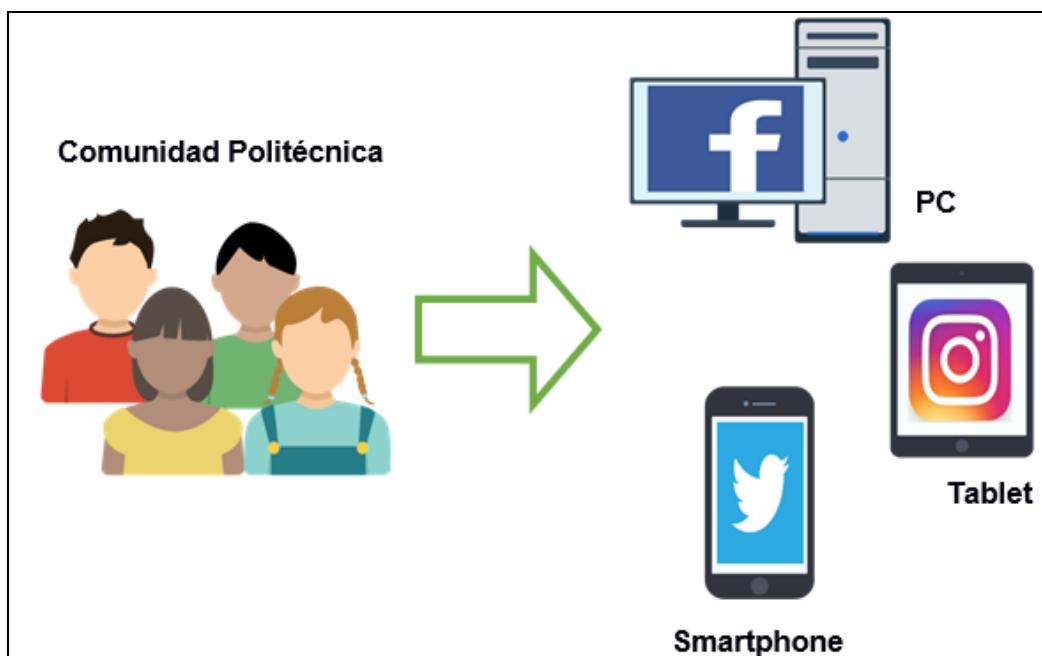


Figura 2.5 Comunidad politécnica y las redes sociales.

2.2.1.1 Componentes del dispositivo de la solución

Como se muestra en la figura 2.6, el dispositivo usado es una placa Raspberry Pi 3 Modelo B, trae un procesador Broadcom BCM2387, un clúster ARM Cortex-A53 (64 bits) (1) que funcionan a una velocidad de ejecución de 1.2 GHz y posee 1 GB de memoria RAM. La placa cuenta con 4 puertos USB 2.0 (2) para conectar el teclado, ratón y otros periféricos que la Raspberry Pi pueda necesitar, además de un puerto 10/100 Ethernet (3) y un conector HDMI (4).

Adicional a esto, la placa también posee una ranura para tarjetas microSD (5), en la tarjeta de memoria se instalará el sistema operativo con el que estará trabajando el dispositivo. Por el puerto microUSB (6) el dispositivo se conectará a un tomacorriente que le provea de energía eléctrica mediante un cable de alimentación, el cual se recomienda usar un adaptador eléctrico de 5 Voltios y 3 Amperios [6].

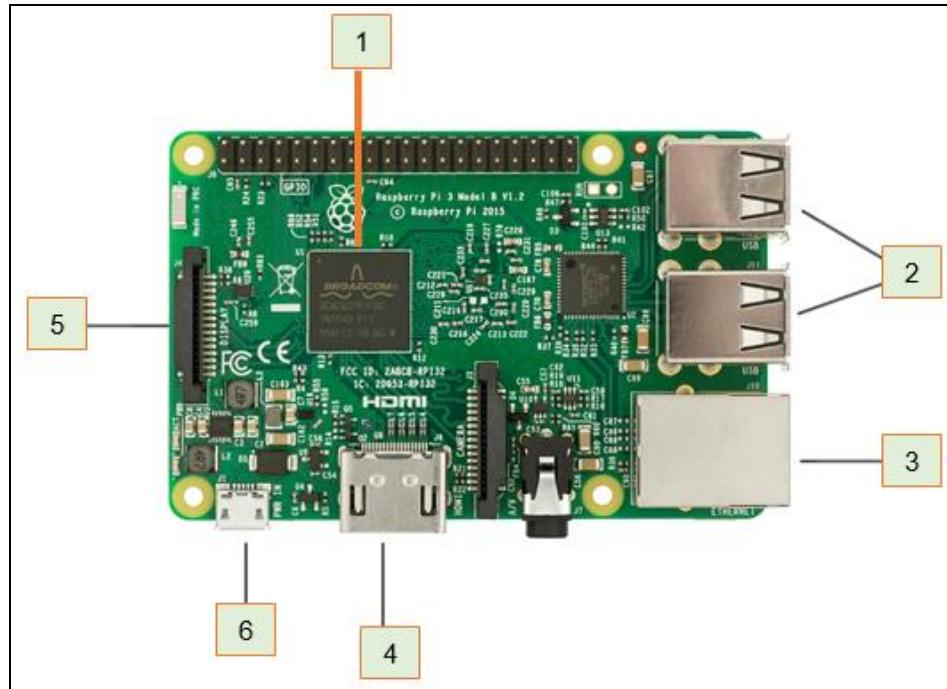


Figura 2.6 Partes de la Raspberry Pi Modelo B.

También se tomó en cuenta utilizar un estuche, para proteger el dispositivo contra golpes, evitar que los componentes sensibles que están adheridos a la placa tengan algún contacto con la suciedad que pueda existir en el entorno, como se ve en la figura 2.7.



Figura 2.7 Estuche protector del dispositivo.

En la tarjeta microSD se instaló el sistema operativo Raspbian con interfaz gráfica, la cual es una distribución GNU/Linux, siendo el sistema considerado oficial para la Raspberry Pi, la versión instalada es Raspbian Linux 10 (Buster), tal como se puede observar en la figura 2.8. Esta versión puede ser descargada gratuitamente desde la página oficial de Raspberry Pi [7].

```
pi@raspberrypi:~ $ cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 10 (buster)"
NAME="Raspbian GNU/Linux"
VERSION_ID="10"
VERSION="10 (buster)"
VERSION_CODENAME=buster
ID=raspbian
ID_LIKE=debian
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
pi@raspberrypi:~ $
```

Figura 2.8 Versión y detalles del Sistema Operativo Raspbian.

El proceso de instalación se lo realizó mediante NOOBS, un administrador de instalación que nos permite seleccionar una de las diferentes distribuciones para la Raspberry Pi [8]. Además, se realizó la instalación de Python versión 3.7.3, un lenguaje de programación de alto nivel con sintaxis fácil de aprender y usar [9]. Con el uso de los distintos módulos y librerías se desarrolló el sistema para el funcionamiento del dispositivo de publicación de eventos.

2.2.2 Desarrollo de la solución

Para explicar el desarrollo de la solución se usó un diagrama de flujo, figura 2.9, en la que se detalla paso a paso los procesos que el sistema realiza para llegar a la solución esperada. A continuación, se detalla el flujo del sistema de la solución propuesta.

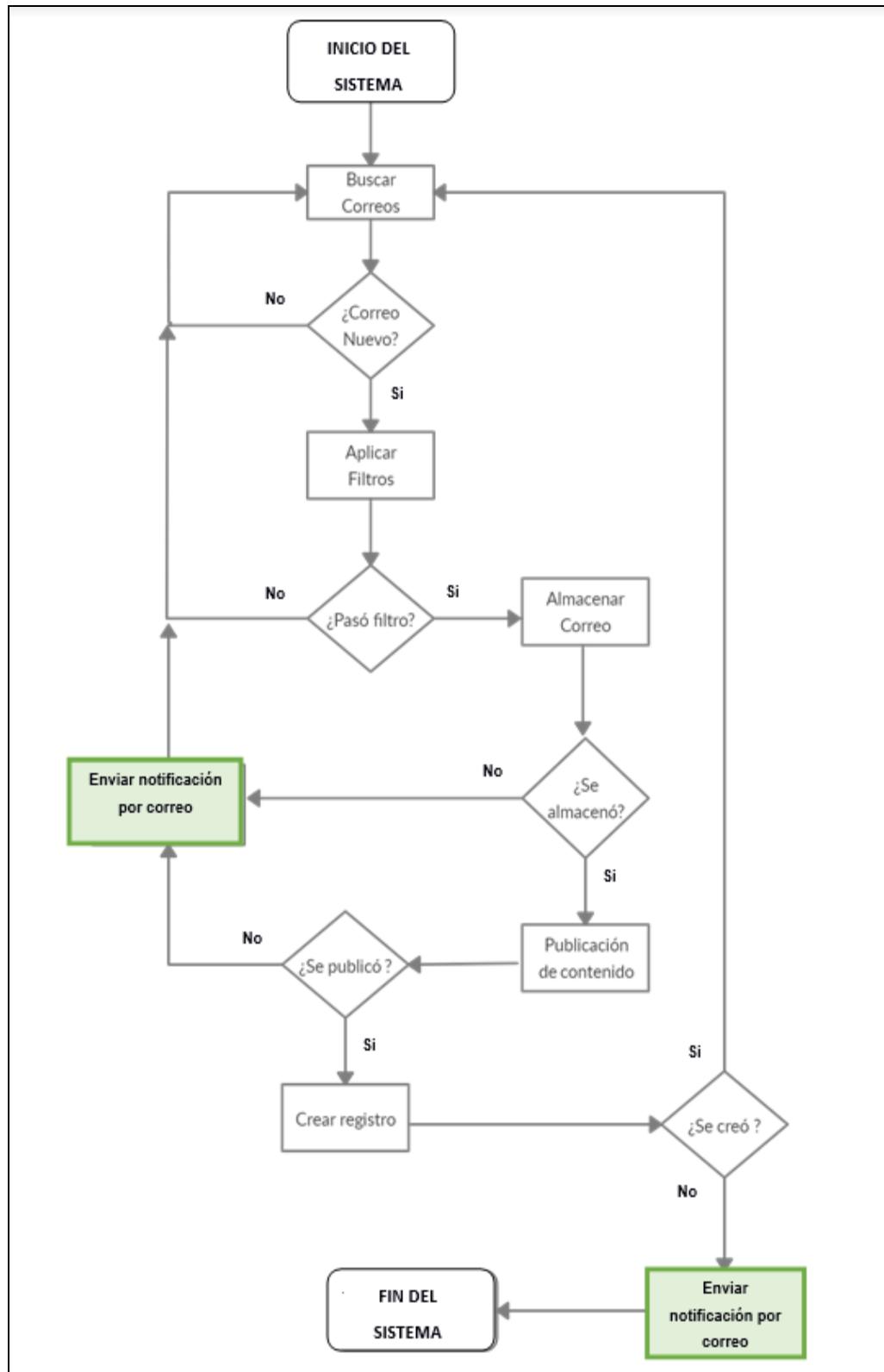


Figura 2.9 Diagrama de flujo del sistema.

2.2.2.1 Proceso de Inicio del sistema

Al iniciar por primera vez el dispositivo y estando conectado a una pantalla por el puerto HDMI, se mostrará una interfaz gráfica de usuario (GUI, por sus siglas en inglés). El diseño de la GUI fue elaborado usando el paquete Tkinter del lenguaje de programación Python, que permite la creación de ventanas y formularios [10]. En la figura 2.10 muestra la pantalla de inicio del dispositivo con el nombre de “Raspberry Ads”. Esta ventana permite al cliente iniciar el funcionamiento del sistema y mostrar el menú principal del aplicativo.

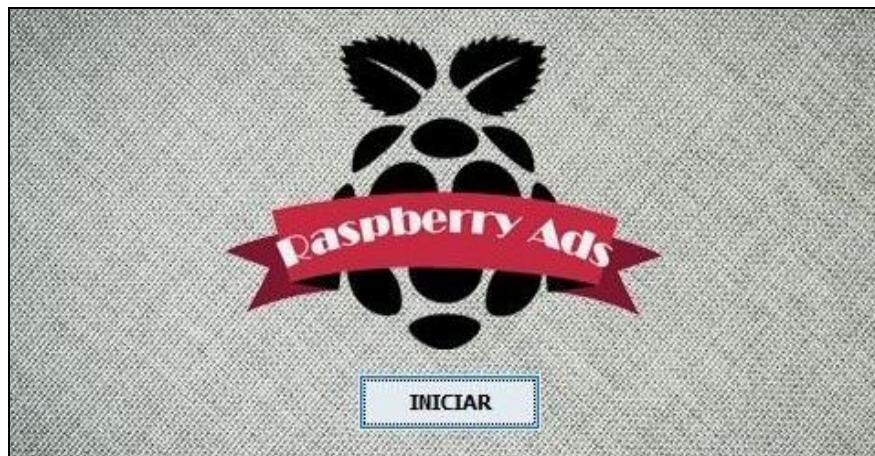


Figura 2.10 Interfaz Gráfica de inicio del sistema.

Para realizar la configuración y administración básica del dispositivo, se diseñó un menú gráfico con cinco opciones, como se observa en la figura 2.11.

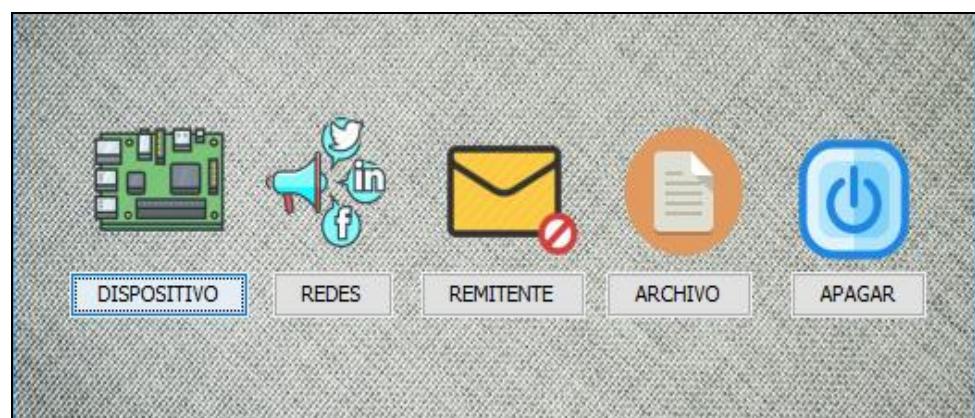


Figura 2.11 Menú principal del sistema.

A continuación, se detalla cada opción seleccionable.

Dispositivo: para especificar la cuenta de correo electrónico del dispositivo como también la cuenta de correo del usuario, con el fin de enviar una notificación si un proceso no se ejecuta correctamente, tal como se ve en la figura 2.12.



Figura 2.12 Configuración de cuenta de correo del dispositivo y usuario.

Redes: para establecer las cuentas de usuario de las redes sociales, ya sea *Facebook*, *Twitter* e *Instagram*, a las cuales se desea hacer la publicación automática de anuncios. En el caso de *Facebook*, es necesario especificar la página donde se desea publicar los eventos, porque de las tres redes sociales, es la única en donde una misma cuenta puede manejar múltiples páginas, en la figura 2.13 vemos la ventana principal para agregar un nuevo usuario de *Facebook*, en el Apéndice F encontramos las demás opciones para agregar un nuevo usuario y pagina de *Facebook*.



Figura 2.13 Configuración de un nuevo usuario de Facebook.

Se uso el módulo cryptography y la clase Fernet para cifrar la información ingresada al dispositivo. Se genera una llave que permite cifrar simétricamente la información [11], de esta manera evitamos que un tercero pueda ver los usuarios y contraseñas de las diferentes cuentas. El código usado se encuentra en el Apéndice G.

Remitente: para indicar las direcciones de correo electrónicos de remitentes permitidos, así solo se publicarán los mensajes enviados desde estas direcciones y evitar que un tercero haga publicaciones no autorizadas en las cuentas de redes sociales establecidas en el paso anterior. En la figura 2.14 se detalla las cuentas de remitentes admitidos por el dispositivo, en la que se permitirá editar o eliminar cada entrada en la tabla.

TABLA DE CORREOS PERMITIDOS

INGRESA CORREO:	<input type="text"/>	OK
CORREOS correos@fiec.espol.edu.ec postgrados@fiec.espol.edu.ec pasantias-comp@fiec.espol.edu.ec pasantias-teleco@fiec.espol.edu.ec pasantias-telematica@fiec.espol.edu.ec pasantias-indus@fiec.espol.edu.ec		
<input type="button" value="EDITAR"/> <input type="button" value="ELIMINAR"/>		

Figura 2.14 Manejo de direcciones de correos permitidos.

Archivo: para visualizar los registros de las publicaciones realizadas por el dispositivo, cada nueva entrada guarda el nombre, fecha, hora y de cada publicación, como se aprecia en la figura 2.15 es posible editar o eliminar cada registro por parte del usuario.

REGISTRO DE PUBLICACIONES

PUBLICACION	FECHA	HORA
Conferencia Administració...	02/08/2019	12:25
Charla FIEC	01/08/2019	11:57
Visita Técnica Automatizac...	01/08/2019	11:30
Feria Gastronómica ESPOL	31/07/2019	09:10
Cursos de AutoCad	30/07/2019	13:55
Talentos ESPOL	30/07/2019	10:50
Taller Python	30/07/2019	10:36
Pasantia Profesional Elec.	29/07/2019	14:25
Oferta Laboral Computación	29/07/2019	12:40

Figura 2.15 Visualización del registro de publicaciones realizadas.

Apagar: para permitir al usuario apagar o reiniciar el dispositivo, de esta manera se evita cualquier daño que se produzca al momento de desconectar el dispositivo del tomacorriente.

2.2.2.2 Proceso de Búsqueda de correo

En este proceso se realiza una búsqueda de nuevos correos, no vistos, en el buzón de mensajes del servidor de correo. En la figura 2.16 observamos el uso del módulo `Imaplib` y su clase `IMAP4_SSL`, que nos permite establecer una conexión segura con el servidor de correo [12]. El sistema buscará si existe o no un nuevo correo, accediendo con el usuario y contraseña de la cuenta de correo previamente configurada en el dispositivo.

```
1 import imaplib
2 def buscar_Emails(user,password):
3     #Funcion que nos permite buscar nuevos correos en el servidor de correo.
4     mail = imaplib.IMAP4_SSL("imap.gmail.com")
5     mail.login(user, password)
6     print ("INGRESO EXITOSO")
7     mail.select("INBOX") #seleccionar el buzon donde buscar los correos.
8     data = mail.search(None, "UNSEEN") # ALL para todos los correos, UNSEEN para los no vistos.
```

Figura 2.16 Uso del módulo `Imaplib` para buscar un nuevo correo.

2.2.2.3 Proceso de Aplicación de filtros

En este proceso, el nuevo correo debe pasar por tres filtros para poder continuar con la extracción y almacenamiento de la información del mensaje. El filtro de "Remitentes, como vimos en la figura 2.14, solo permitirá que el sistema extraiga la información de los mensajes enviados desde las direcciones de correos permitidos. El segundo filtro es de los "Registros de Publicación", como se puede observar en la figura 2.15, se realizará una comparativa entre el asunto del nuevo correo con la información del registro, si existe una similitud, el correo no pasará el filtro. El último filtro realizará un análisis del mensaje y solo aceptará los correos que cumplan con los elementos necesarios, como la imagen e información de la publicación. Se recomienda el uso de una plantilla, como podemos ver en la figura 2.17, para poder realizar una publicación exitosa. En el caso de no pasar al menos uno de los filtros, el correo será descartado y el sistema procederá a buscar

un nuevo correo en el buzón. En el Apéndice H encontramos el código de los filtros aplicados.

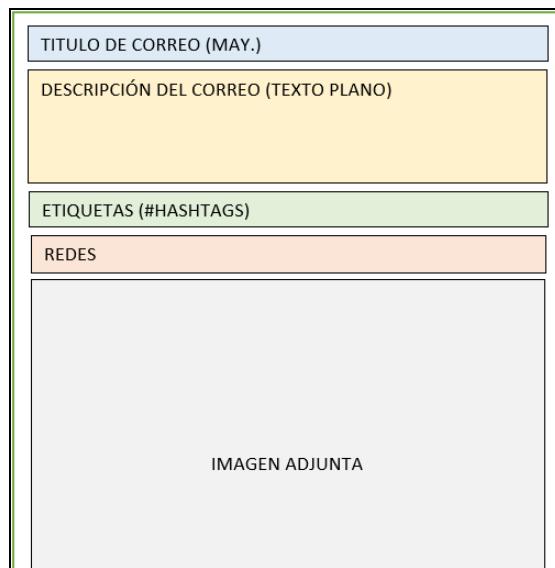


Figura 2.17 Formato para realizar una publicación.

2.2.2.4 Proceso Almacenamiento de correo

Si el nuevo correo pasa los filtros anteriormente mencionados, se procede a la extracción y almacenamiento de la información en el dispositivo. El sistema creará una carpeta con el asunto y fecha del correo, dentro se guardará la imagen y un archivo en texto con los datos enviados en base al formato de la figura 2.17.

En la figura 2.18 podemos observar las carpetas creadas por el sistema. El código de este proceso se encuentra en el Apéndice I.

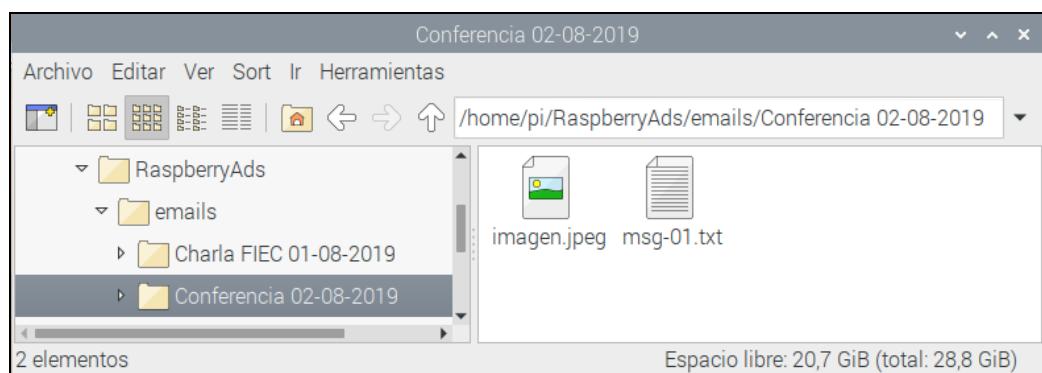


Figura 2.18 Almacenamiento de la información en el dispositivo.

En el caso de que el proceso de almacenamiento de los correos falle, ya sea por falta de espacio para guardar la información o algún error en el proceso, se enviará una notificación al correo electrónico del usuario previamente establecido, en la figura 2.19 podemos ver una notificación enviada por correo, con información sobre el mensaje y la hora en que ocurrió el error. Luego de enviar el mensaje, el sistema regresará al proceso de buscar un nuevo correo para procesarlo.

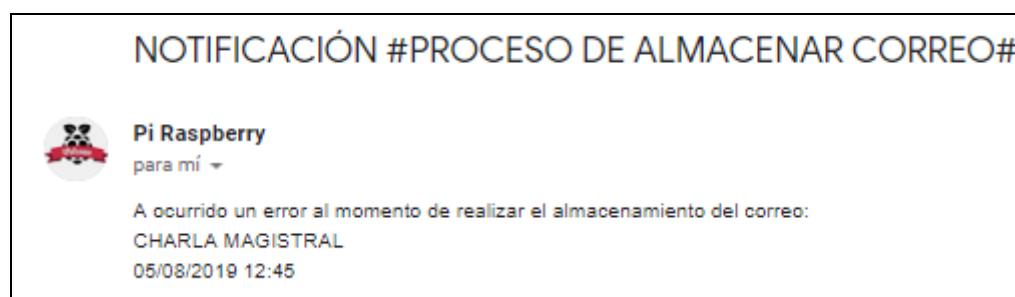


Figura 2.19 Notificación de error a la hora de almacenar un correo.

2.2.2.5 Proceso de Publicación de contenido

El quinto proceso se encarga de la publicación de la información almacenada en el dispositivo a las diferentes redes sociales. En este proceso se hizo uso de la librería Selenium que permite el manejo de un webdriver, el cual interactúa con todos los elementos que forman las páginas web, así se accede a cada red social y se realiza las publicaciones [13]. El webdriver usado fue PhantomJS, que realiza todo el proceso en segundo plano sin la necesidad de tener un navegador abierto [14]. En la figura 2.20, detallamos una parte del código de acceso a *Facebook*, usando Selenium y el webdriver PhantomJS. En el Apéndice J se encuentra el código completo para *Facebook* y *Twitter*.

```

1  from selenium import webdriver
2  from selenium.webdriver.common.keys import Keys
3  import os
4  #FUNCION PARA PUBLICAR EN FACEBOOK
5  def post_Facebook(titulo_, descripcion_, etiquetas_, path_Img, ext_Img):
6      #obtener usuario y contrasena
7      with open ("home/pi/RaspberryAds/facebook/cuenta.txt", "r") as f:
8          userfa = f.readline()
9          passfa = f.readline()
10         f.close()
11     #obtener pagina para publicar
12     with open ("home/pi/RaspberryAds/facebook/paginas.txt", "r") as f:
13         pagepost = f.readline()
14         f.close()
15     #accedemos al webdriver PhantomJS
16     driver = webdriver.PhantomJS()
17     driver.get('https://mbasic.facebook.com')
18     email = driver.find_element_by_name("email")
19     email.send_keys(userfa)
20     psw = driver.find_element_by_name("pass")
21     psw.send_keys(passfa)
22     psw = driver.find_element_by_name("pass").send_keys(Keys.ENTER)

```

Figura 2.20 Uso de Selenium y PhantomJs para acceder a Facebook.

Para *Instagram* se usó la librería InstagramAPI, que permite acceder a todas las funciones que la red social posee [15], véase figura 2.21. De esta manera se ingresa al aplicativo web para realizar las publicaciones correspondientes.

```

1  from InstagramAPI import InstagramAPI
2  import os
3  #FUNCION PARA PUBLICAR EN INSTAGRAM
4  def post_Instagram(path_Img, ext_Img, titulo_, descripcion_, etiquetas_):
5      #obtener usuario y contrasena
6      with open ("home/pi/RaspberryAds/instagram/cuenta.txt", "r") as f:
7          userig = f.readline()
8          passig = f.readline()
9          f.close()
10     #direccion de la imagen
11     img = os.path.abspath(path_Img+"/"+ext_Img)
12     #texto de la publicacion
13     caption = titulo_+" "+descripcion_+" "+etiquetas_
14     #iniciar Instagram
15     igapi = InstagramAPI(userig, passig)
16     #login
17     igapi.login()
18     igapi.uploadPhoto(img, caption=caption)
19     return

```

Figura 2.21 Uso de la librería InstagramAPI para realizar la publicación.

En el caso de que no se pueda realizar una publicación en una o varias redes sociales como se muestra en la figura 2.22, se realizará un reintento (hasta 3 veces) de publicación por red social, si no se llega a completar el proceso, se enviará una notificación por correo informando sobre la existencia de un error o falla en el proceso por cada red social. Luego de enviar la notificación por correo, el sistema ejecutará el proceso de búsqueda de un nuevo correo.

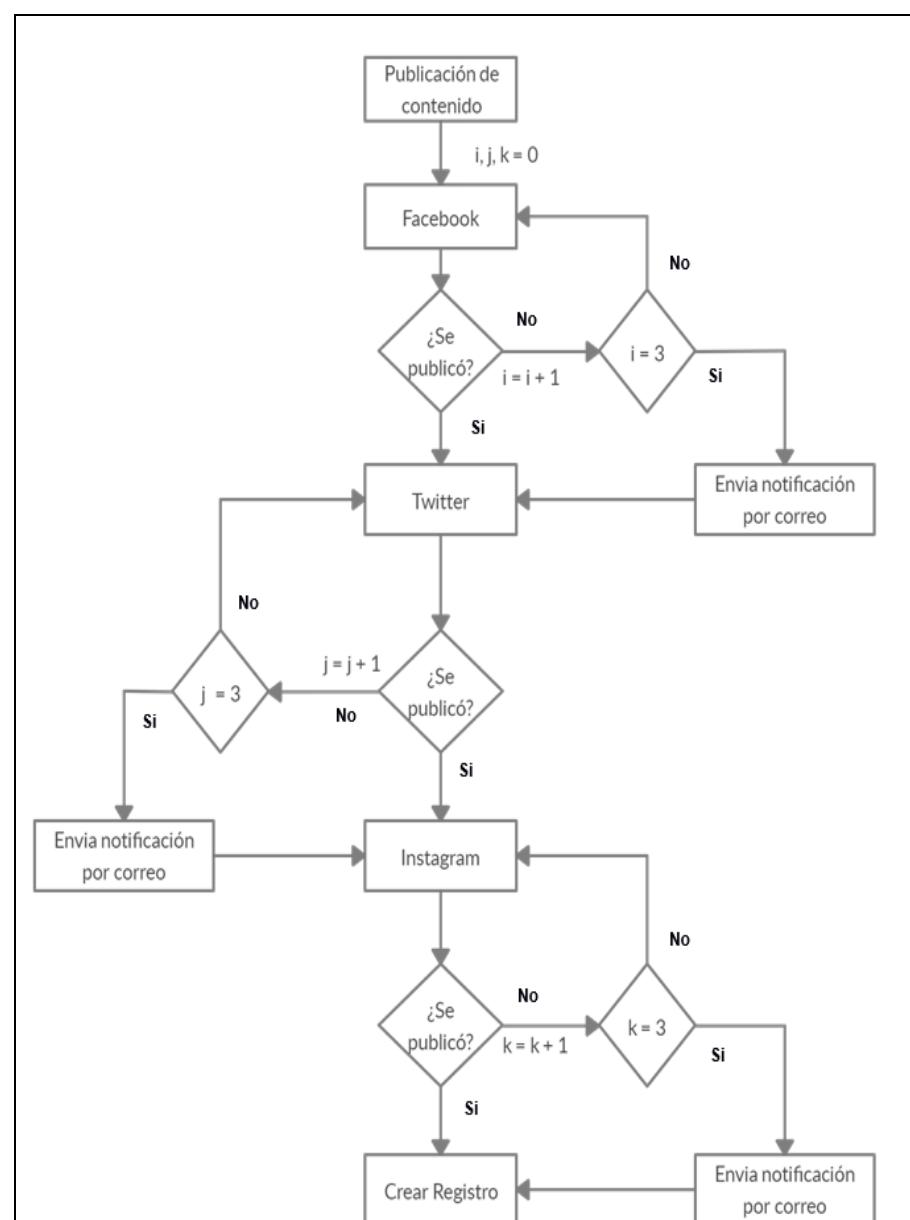


Figura 2.22 Diagrama de flujo para la publicación de contenido.

2.2.2.6 Proceso de Creación de registro

Si se logró realizar con éxito la publicación del contenido, el sistema creara un registro en texto de la publicación realizada, con hora y fecha. Este registro de datos permite usar un filtro para evitar que se realicen múltiples publicaciones. Si el proceso llega a fallar, se enviará una notificación por correo al usuario, pero en lugar de proceder con buscar un nuevo correo, el sistema termina su ejecución hasta que se resuelva el error. En la figura 2.23 podemos ver el código usado para la creación de los registros de las publicaciones realizadas.

```
1  from time import localtime, asctime
2  #CREA UN REGISTRO DE LA PUBLICACION PARA EVITAR EMAILS DUPLICADOS
3  def guardar_Registro(subject_):
4      with open ("home/pi/RaspberryAds/registro/publicaciones.txt", "a") as f:
5          date = asctime(localtime())
6          regis = subject_
7          regis += " "+date
8          print("-----")
9          print("CREANDO REGISTRO")
10         print("-----")
11         print (regis)
12         print("-----")
13         f.write(regis+"\n")
14         f.close()
15     return
```

Figura 2.23 Código usado para la creación de registros.

CAPÍTULO 3

3. PLAN DE IMPLEMENTACIÓN Y PRESUPUESTO

A continuación, en la figura 3.1, se detalla el plan de implementación y presupuesto de la solución planteada, en donde se especifica la duración del desarrollo de la solución, en lapso de 2 meses y 4 días. Además de los costos relacionados a la adquisición de equipos y elaboración del sistema.

Nombre de tarea	Duración	Comienzo	Fin	Costo
« Diseñar e implementar un dispositivo de almacenamiento de correo.	64 días	lun 17/06/19	mar 10/09/19	\$2.421,14
« Fase 1: Análisis	17 días	lun 17/06/19	vie 05/07/19	\$121,14
Entrevistas y encuestas con los actores principales.	5 días	lun 17/06/19	vie 21/06/19	\$0,00
Análisis de los resultados obtenidos.	4 días	sáb 22/06/19	mié 26/06/19	\$0,00
Cotización de equipos y componentes a usar.	3 días	jue 27/06/19	sáb 29/06/19	\$0,00
Adquisición de los equipos y componentes a usar.	5 días	lun 01/07/19	vie 05/07/19	\$121,14
« Fase 2: Desarrollo del sistema de almacenamiento	29 días	lun 08/07/19	jue 15/08/19	\$1.300,00
Instalación y configuración del sistema operativo Raspbian	3 días	lun 08/07/19	mié 10/07/19	\$300,00
Instalación del lenguaje de programación Python	1 día	jue 11/07/19	jue 11/07/19	\$0,00
Análisis del diseño de módulos del sistema	5 días	vie 12/07/19	jue 18/07/19	\$0,00
Configuración de módulos del sistema en base a las necesidades del cliente	20 días	vie 19/07/19	jue 15/08/19	\$1.000,00
« Fase 3: Desarrollo GUI	11 días	vie 16/08/19	vie 30/08/19	\$500,00
Análisis del diseño de la Interfaz gráfica	3 días	vie 16/08/19	mar 20/08/19	\$0,00
Configuración de la GUI del sistema	7 días	mié 21/08/19	jue 29/08/19	\$500,00
« Fase 4: Prueba y corrección de errores	7 días	vie 30/08/19	lun 09/09/19	\$500,00
Prueba y corrección de errores	5 días	vie 30/08/19	jue 05/09/19	\$0,00
Implementación del sistema de almacenamiento al cliente	1 día	vie 06/09/19	vie 06/09/19	\$500,00

Figura 3.0.1 Plan de implementación y presupuesto de la solución desarrollada.

En las fechas establecidas, hemos considerado fines de semana como días laborales, incluyendo los feriados.

En la tabla 3.1, detallamos el precio por unidad de todos los componentes que fueron usados para la creación del dispositivo. Estos valores incluyen el impuesto al valor agregado (IVA).

Tabla 3.1 Costo de componentes para la creación del dispositivo.

CANTIDAD	EQUIPOS PARA USAR	Precio (USD)
1	Modulo Raspberry Pi 3 Modelo B	\$70,54
1	Ventilador Cooler 5V 0.2A	\$2,86
1	Disipador Aluminio CPU	\$0,89
1	Cable HDMI 2mt	\$5,22
1	MicroSD Sandisk 32GB Clase 10	\$18,00
1	Adaptador Eléctrico 5V 3A	\$6,00
1	Case Enclosure	\$17,63
Costo de hardware		\$121,14

En la tabla 3.2, se detalla el costo del diseño y configuración para el funcionamiento del sistema, además del costo final de la implementación de la solución.

Tabla 3.2 Costo de servicios para la implementación de la solución.

ÍTEM	TOTAL
Configuración del dispositivo Raspberry.	\$300,00
Configuración de la GUI del sistema.	\$500,00
Configuración de módulos del sistema de almacenamiento.	\$1000,00
Instalación del sistema al cliente.	\$500,00
Costo de Diseño y Configuración	\$2500,00
Costo de HARDWARE	\$121,14
Costo de Diseño y Configuración	\$2500,00
TOTAL, COSTO IMPLEMENTACIÓN	\$2621,14

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

Conclusiones

El diseño y la implementación de un sistema que permite almacenar el contenido de los correos, que son enviados por la facultad, y publicarlos en las diferentes redes sociales, es una de las maneras más eficiente que logramos desarrollar para captar el interés del estudiante, debido a que, mediante un estudio ya realizado se comprobó que las redes sociales tanto como Facebook (siendo la red con mayor cantidad de usuarios conectados), Instagram y Twitter siguen en auge dentro de la sociedad.

Con la automatización del proceso de publicación de contenido a las redes sociales, se le brinda al encargado de difusión, optimizar y agilizar el tiempo que invierte en hacer manualmente el proceso, como el de buscar la imagen del anuncio en el ordenador, agregarle un encabezado, añadirle los hashtags, y repetir la misma operación con las diferentes cuentas que maneja.

Fue logro el desarrollo del sistema para almacenar los correos usando una placa Raspberry Pi, ya que, a pesar de ser un dispositivo de bajo costo, pudimos trabajar con el sistema operativo Raspbian, que es una de las diferentes distribuciones basada en GNU/Linux.

Esta solución puede ser desarrollada como un aplicativo, pero con la implementación de un dispositivo de bajo costo, se demuestra que no se necesita de muchos recursos para lograr que el sistema funcione.

Recomendaciones

Aunque el dispositivo pueda ser trasladado de un lugar a otro, es recomendable mantenerlo en un departamento u oficina, conectado siempre a un punto de datos, mediante el puerto Ethernet, de esta manera no existiría interferencia u obstáculos cuando se conecte a Internet, a diferencia de una conexión inalámbrica.

Se recomienda tener una fuente de alimentación ininterrumpida, así en caso de cortes de luz, el dispositivo pueda seguir con su proceso de clasificar y publicar los eventos que son enviados mediante correo electrónico.

Realizar una prueba de rendimiento para conocer el límite en que el dispositivo puede permanecer encendido.

BIBLIOGRAFÍA

- [1] Gerencia de Planificación Estratégica. (2019). Informe De Rendición De Cuentas ESPOL 2018 [En Línea]. Disponible en:
<http://planificacion.espol.edu.ec/sites/default/files/InformeIRC2018-web.pdf>
- [2] Beltran Triviño, G. N., & Espinal Ordoñez, L. E. (2018). Diseño e Implementación de un Dispositivo de Señalización Digital utilizando hardware de bajo costo y software libre. Guayaquil: Espol.
- [3] Noboa, L. E. (2010). Análisis, diseño e implementación de una red social para Espol digital utilizando herramientas 2.0. Guayaquil: Espol.
- [4] Elizabeth, S. R. (2015). Estudio de la utilidad de Redes Sociales para difundir el Potencial Turístico De las comunas de Ayangue, Cadeate y Libertador Bolívar. Guayaquil: Espol.
- [5] Alcazar Ponce, J. P. (2019). Estado Digital Ecuador 2019. Obtenido de
<https://blog.formaciongerencial.com/estadodigitalecuador/>
- [6] Raspberry Pi. (2019). Raspberry Pi 3 Model B [En Línea]. Disponible en:
<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [7] Raspberry Pi. (2019). Download Raspbian for Raspberry Pi [En Línea]. Disponible en: <https://www.raspberrypi.org/downloads/raspbian/>
- [8] Raspberry Pi. (2019). NOOBS - Raspberry Pi Documentation [En Línea]. Disponible en: <https://www.raspberrypi.org/documentation/installation/noobs.md>
- [9] Python. (2019). What is Python? Executive Summary [En Línea]. Disponible en:
<https://www.python.org/doc/essays/blurb/>
- [10] Python. (2019). tkinter - Python interface to Tcl/Tk [En Línea]. Disponible en:
<https://www.python.org/doc/essays/blurb/>
- [11] Vollebregt, B. (2019). Encryption and Decryption in Python [En Línea]. Disponible en: <https://nitratine.net/blog/post/encryption-and-decryption-in-python/>
- [12] Python. (2019). imaplib - IMAP4 protocol client [En Línea]. Disponible en:
<https://docs.python.org/3/library/imaplib.html>

- [13] Selenium. (2019). Installation - Selenium Python Bindings [En Línea]. Disponible en: <https://selenium-python.readthedocs.io/installation.html#introduction>
- [14] Oloruntoba. S. (2018). What is PhantomJS and How is it Used? [En Línea]. Disponible en: <https://scotch.io/tutorials/what-is-phantomjs-and-how-is-it-used>
- [15] LevPasha. (2019). Instagram-API-python [En Línea]. Disponible en: <https://github.com/LevPasha/Instagram-API-python/>

APÉNDICES

APÉNDICE A: ‘Mapa de Actores’ de los involucrado a la problemática general dentro de la institución.



Figura A.1 Mapa de actores

APÉNDICE B: Banco de preguntas

LLUVIA DE IDEAS

Preguntas tentativas de entrevistas (estudiantes)

- ¿Cuál es tu nombre?
- ¿A qué facultad perteneces?
- ¿Como te informas de las charlas o eventos que se realizan en tu facultad?
- ¿Qué acción tomas al momento de que te llega un correo?
- ¿Tu facultad cuenta con un correo exclusivo para tu facultad?
- ¿Qué tan a menudo lees los correos que la Espol te envía?
- ¿Crees que al no leer puedes estar desinformado de ciertas actividades? ¿Si te enteraras de eventos que puedan ser de tu beneficio, asistirías?
- ¿Tienes configurado el correo de la Espol en tu móvil?

Preguntas de encuesta (online estudiantes)

¿Tienes configurado el correo de la universidad en tu smartphone?

Si

No

¿Qué tan a menudo lees los correos recibidos?

Muy frecuentemente

Regular

Casi nada

Los abro, pero no los leo Cuando te llega un correo...

Lo lees al instante

Lo descartas

Si lo descartas, ¿Cuál es la razón?

No son de mi interés/beneficio/importancia

No tengo tiempo de leer porque son muchos correos

Me olvido de leerlos

¿Crees que, al no leer los correo, puedes estar desinformados de ciertas actividades a realizarse en tu facultad?

Si

No

Tal vez

¿Cómo te informas de los eventos o charlas que se realizan en tu facultad?

Grupo de amigos

Redes Sociales

Correo

¿Si te enteras de algún evento que podría ser de tu beneficio, asistirías?

Si

No

Tal vez

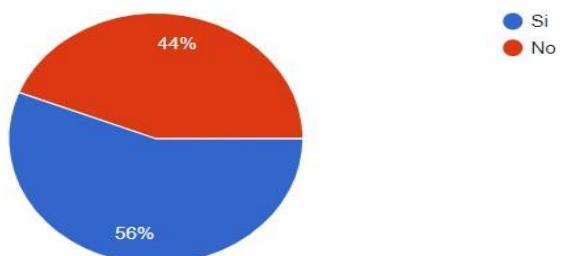
Preguntas de entrevistas (Personal DST)

- ¿Cuánto tiempo lleva ejerciendo su puesto?
- ¿Qué actividades realiza?
- ¿Usted se encarga del envío o también de la creación de los correos?
- ¿Existe algún formato para la creación de los correos?
- ¿Existe una hora específica para el envío de correos?
- ¿Cuántas veces son enviados los correos?
- ¿Considera que es un problema que los estudiantes no revisen el correo?
- ¿Considera que la poca asistencia a las charlas se debe a que los estudiantes no se informan de las mismas?

APÉNDICE C: Diagrama de respuestas a la encuesta online a un grupo de 25 estudiantes de las diferentes facultades.

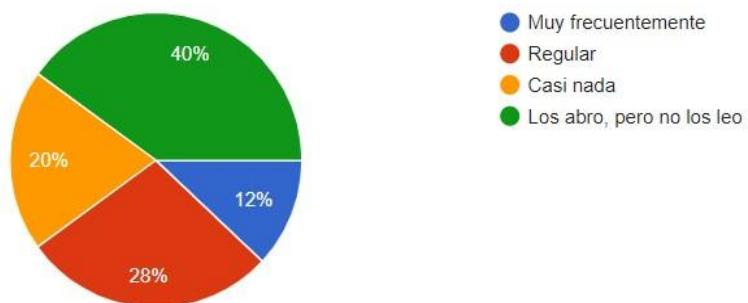
¿Tienes configurado el correo en tu smartphone?

25 respuestas



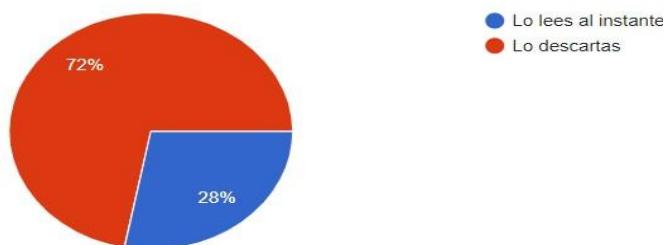
¿Qué tan a menudo lees los correos enviados?

25 respuestas



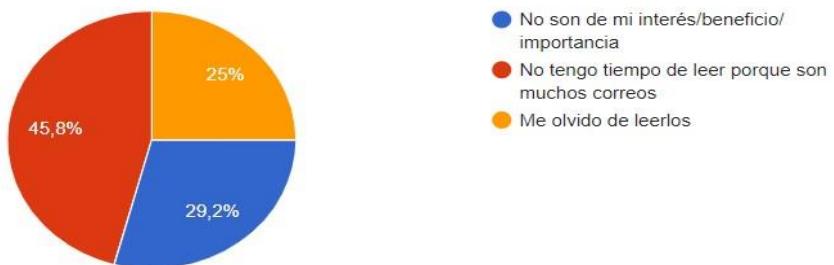
Cuando te llega un correo..

25 respuestas



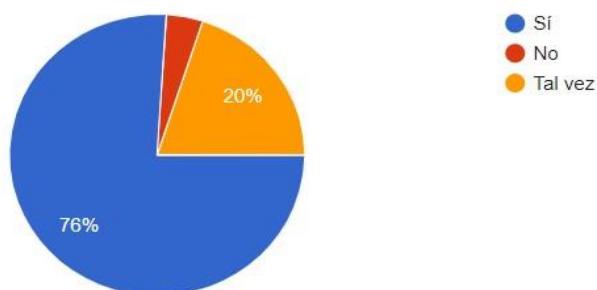
Si lo descartas, ¿Cuál es la razón?

24 respuestas



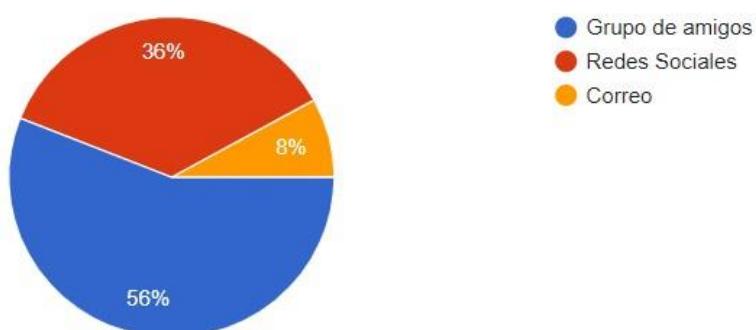
¿Crees que al no leer los correo, puedes estar desinformados de ciertas actividades a realizarse en tu facultad?

25 respuestas



¿Cómo te informas de los eventos o charlas que se realizan en tu facultad?

25 respuestas



Si te enteras de algún evento que podría ser de tu beneficio, asistirías?

25 respuestas

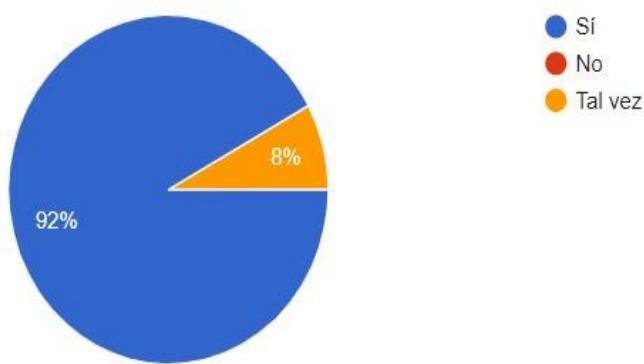


Figura C. Encuestas online

APÉNDICE D: 'Mapa de empatía' de los estudiantes entrevistados.

Estudiante: Domingo Villegas

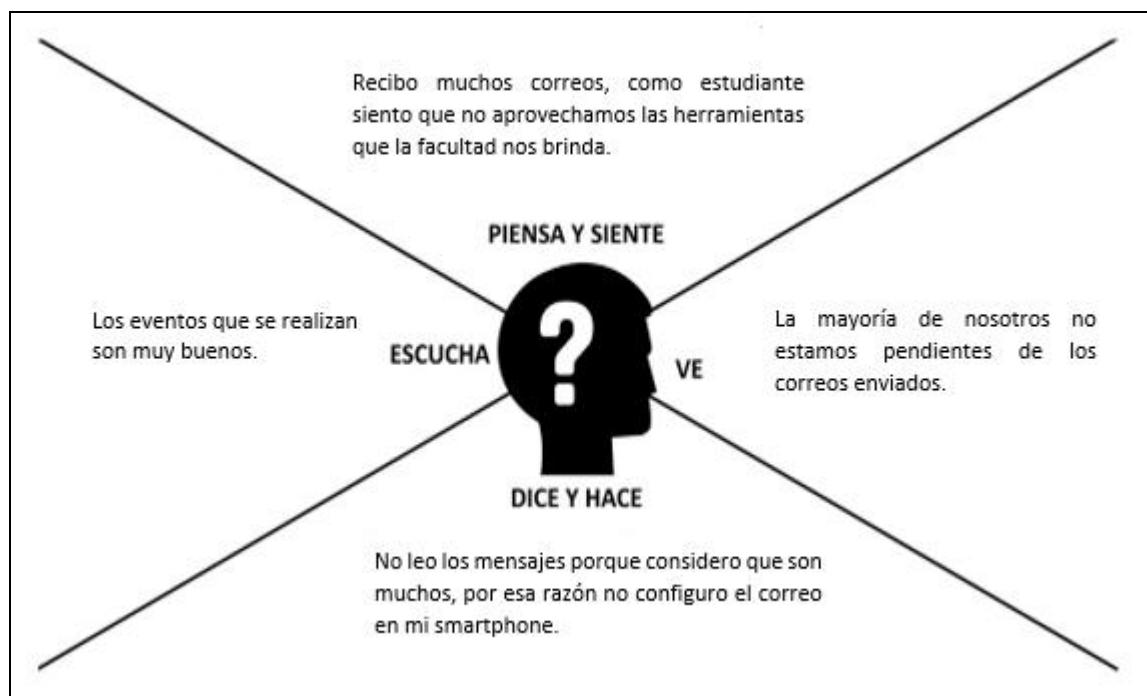


Figura D.1 Mapa de empatía, estudiante: Domingo

Estudiante: Carlos Cañarte

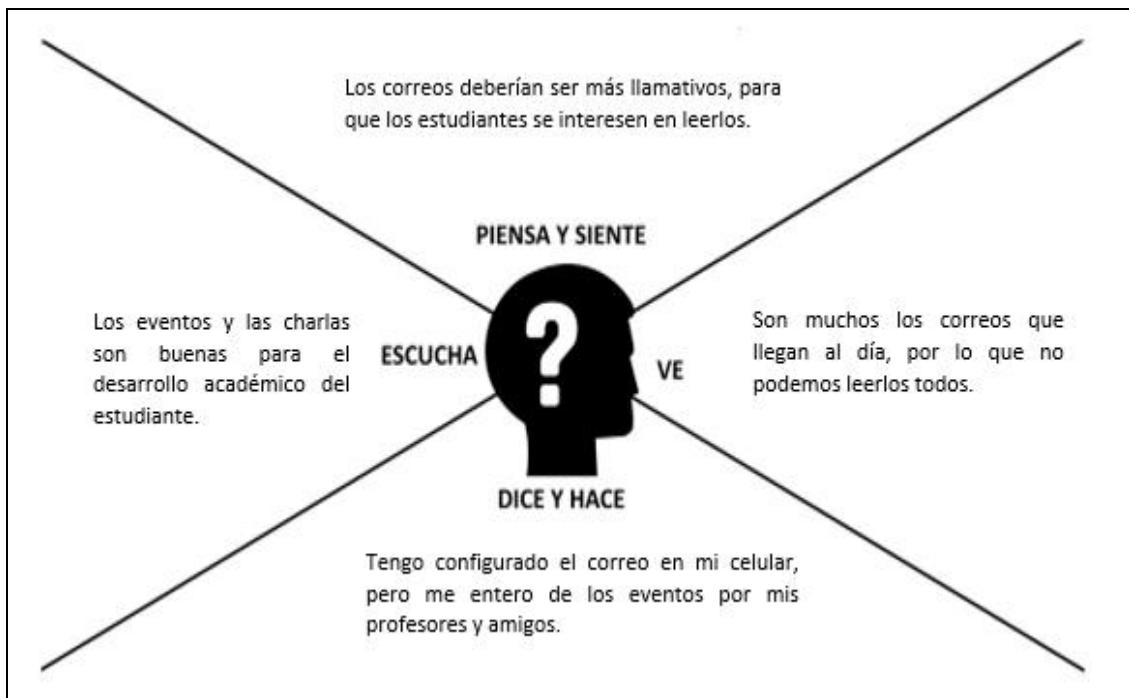


Figura D.2 Mapa de empatía, estudiante: Carlos

APÉNDICE E: ‘Árbol de Problema’, encontramos las causas y los efectos de la problemática principal dentro de la institución.

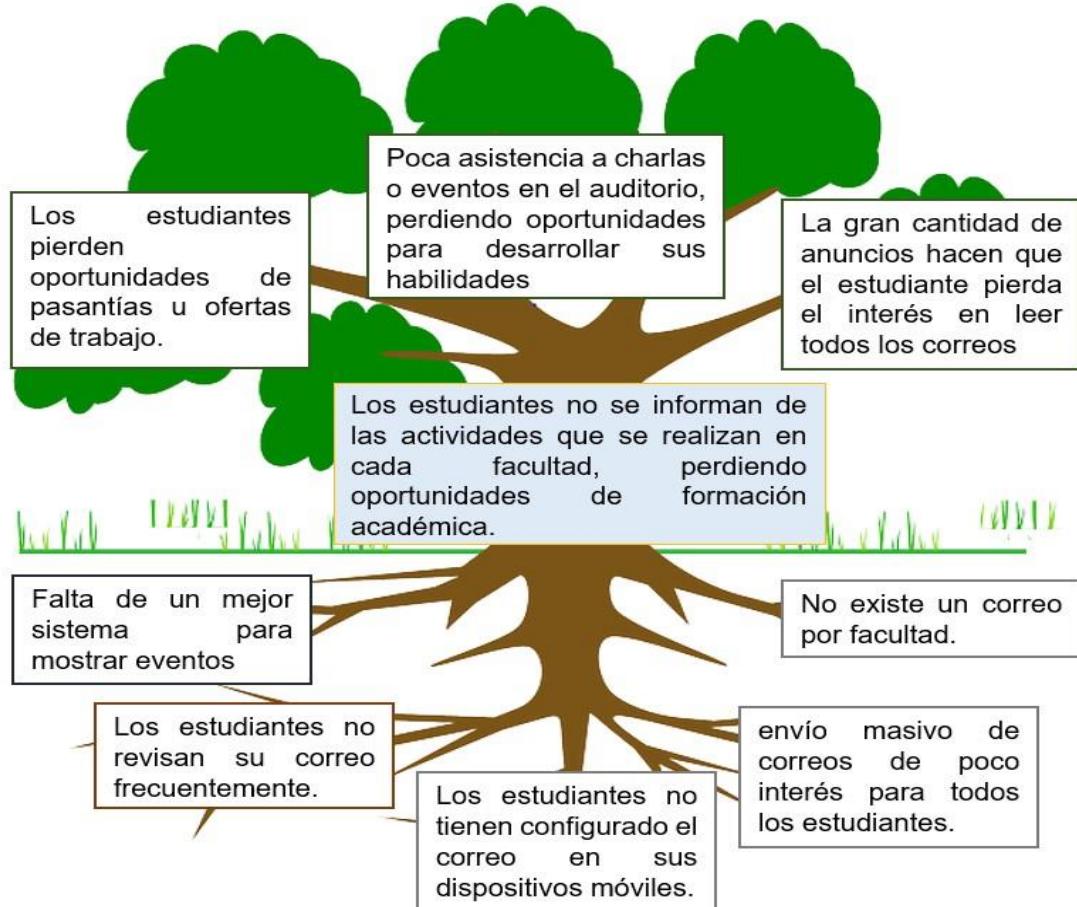


Figura E.1 Árbol de problema.

APÉNDICE F: Ventanas de opciones para crear un establecer un nuevo usuario y pagina de *Facebook*.



Figura F.1 Ventana para visualizar y editar los usuarios creados.



Figura F.2 Ventana para agregar paginas a los usuarios creados.



Figura F.3 Ventana para agregar visualizar y borrar páginas.

APÉNDICE G: Creación de llave para el cifrado y descifrado de la información.

```
1 import cryptography
2 from cryptography.fernet import Fernet
3 key = Fernet.generate_key() #genero una llave
4 def cifrar_info (key):
5     #Archivo para cifrar
6     with open ('usuario/acceso.txt', 'rb') as f:
7         data = f.read()
8         f.close()
9     fernet = Fernet (key)
10    infodatos = fernet.encrypt(data) #Se cifra la informacion
11    #Guardar archivo cifrado
12    with open ('usuario/acceso.txt', 'wb') as f:
13        f.write(infodatos)
14        f.close()
15    return
```

Figura G.1 Cifrado de la información.

APÉNDICE H: Código para aplicar el filtro a los nuevos correos.

```
1 def aplicar_filtro(from_, subject_):
2     #esta funcion nos permite aplicar distintos filtros al correo
3     fil = from_.lower()
4     with open ("home/pi/RaspberryAds/remitente/filtro.txt", "r") as f:
5         lista = f.readlines()
6         f.close()
7     fil2 = subject_.title()
8     with open ("home/pi/RaspberryAds/registro/publicaciones.txt", "r") as f:
9         lista2 = f.readlines()
10        f.close()
11    # Recorre todas los elementos y busca coincidencia con filtro.txt
12    for linea in lista:
13        result = linea.find(fil) #retorna un valor, -1 si no se encuentra
14        if result < 0:
15            ban = 0
16        else:
17            print ('Remitente Autorizado')
18            ban = 1
19            break
20    #Recorre todos los elementos y busca conincidencia con registro.txt
21    for linea2 in lista2:
22        result2 = linea2.find(fil2) #retorna un valor, -1 si no se encuentra
23        if result2 >= 0:
24            print('Duplicado')
25            ban2 = 0
26            break
27        else:
28            ban2 = 1
29    if ban and ban2 == 1:
30        print ("PASO FILTRO")
31    else:
32        return False
```

Figura H.1 Función para aplicar el filtro a los correos.

APÉNDICE I: Uso de las librerías *email*, *minetypes* y *os* para la extracción y almacenamiento de la información.

```
1 import email
2 import mimetypes
3 import os
4 def extraer_Info (data):
5     #esta funcion permite obtener datos basicos del correo
6     raw_email = data[0][1]
7     raw_email_string = raw_email.decode('UTF-8')
8     email_message = email.message_from_string(raw_email_string)
9     from_=email_message['From'] #remitente
10    subject_=email_message['Subject'] #asunto
11    date_ = email_message['date'] #fecha
12    #recorrer el correo para obtener la informacion
13    for part in email_message.walk():
14        if part.get_content_maintype() == "multipart":
15            continue
16        filename = part.get_filename()
17        content_type = part.get_content_type()
18        if not filename:
19            ext = mimetypes.guess_extension(content_type)
20            if not ext:
21                ext = '.bin'
22            if 'text' in content_type:
23                ext = '.txt'
24            elif 'html' in content_type:
25                ext = '.html'
26            filename = 'msg-%02d%s' %(ext)
27        #almacenar los archivos en el dispositivo
28        save_path1 = os.path.join(os.getcwd(), 'emails', subject_, date_)
29        if not os.path.exists(save_path1):
30            os.makedirs(save_path1)
31        with open(os.path.join(save_path1,filename),'wb') as fp:
32            fp.write(part.get_payload(decode=True))
```

Figura I.1 Extracción y almacenamiento de la información.

APÉNDICE J: Código usado para realizar las publicaciones tanto en *Facebook* como el *Twitter*.

```
1  from selenium import webdriver
2  from selenium.webdriver.common.keys import Keys
3  import os
4  #FUNCION PARA PUBLICAR EN FACEBOOK
5  def post_Facebook(titulo_, descripcion_, etiquetas_, path_Img, ext_Img):
6      #obtener usuario y contrasena
7      with open ("home/pi/RaspberryAds/facebook/cuenta.txt", "r") as f:
8          userfa = f.readline()
9          passfa = f.readline()
10         f.close()
11     #obtener pagina para publicar
12     with open ("home/pi/RaspberryAds/facebook/paginas.txt", "r") as f:
13         pagepost = f.readline()
14         f.close()
15     #accedemos al webdriver PhantomJS
16     driver = webdriver.PhantomJS()
17     driver.get('https://mbasic.facebook.com')
18     email = driver.find_element_by_name("email")
19     email.send_keys(userfa)
20     psw = driver.find_element_by_name("pass")
21     psw.send_keys(passfa)
22     psw = driver.find_element_by_name("pass").send_keys(Keys.ENTER)
23     img = os.path.abspath(path_Img+"/"+ext_Img)
24     mssg = titulo_+"_"+descripcion_+"_"+etiquetas_
25     driver.get(pagepost)
26     campo_texto = driver.find_element_by_name("xc_message")
27     campo_texto.send_keys(mssg)
28     driver.find_element_by_name("view_photo").click()
29     driver.find_element_by_name("file1").send_keys(img)
30     driver.find_element_by_name("add_photo_done").click()
31     publicar = driver.find_element_by_name("view_post").click()
32     driver.quit()
```

Figura J.1 Código para realizar publicaciones en Facebook.

```

1  from selenium import webdriver
2  from selenium.webdriver.common.keys import Keys
3  import os
4  #FUNCION PARA PUBLICAR EN TWITTER
5  def post_Twitter(path_Img, ext_Img, titulo_, descripcion_, etiquetas_):
6      #obtener usuario y contrasena
7      with open ("home/pi/RaspberryAds/twitter/cuenta.txt", "r") as f:
8          usertw = f.readline()
9          passtw = f.readline()
10         f.close()
11     ##accedemos al webdriver PhantomJS
12     driver = webdriver.PhantomJS()
13     driver.get('https://twitter.com/login')
14     usr_box = driver.find_element_by_class_name('js-username-field')
15     usr_box.send_keys(usertw)
16     pwd_box = driver.find_element_by_class_name('js-password-field')
17     pwd_box.send_keys(passtw)
18     login_button = driver.find_element_by_css_selector('button.submit\
19 EdgeButton.EdgeButton--primary.EdgeButton--medium')
20     login_button.submit()
21     img = os.path.abspath(path_Img+"/"+ext_Img)
22     mssg = titulo_+"_"+descripcion_+"_"+etiquetas_
23     driver.get('https://twitter.com/compose/tweet')
24     box_text = driver.find_element_by_id('tweet-box-home-timeline')
25     box_text.send_keys(mssg)
26     driver.find_element_by_css_selector('input.file-input').send_keys(img)
27     driver.quit()
28     return

```

Figura J.2 Código para realizar publicaciones en Twitter.