

UNA APLICACIÓN COLABORATIVA PARA LA GESTIÓN DE PROTOCOLOS DE TESIS BASADA EN UNA RASPBERRY PI

A COLLABORATIVE APPLICATION FOR THE MANAGEMENT OF THESIS PROTOCOLS BASED ON RASPBERRY PI

Jesús Omar Sánchez Tobón

Benemérita Universidad Autónoma de Puebla, México
sanchez.galvez@correo.buap.mx

Luz A. Sánchez Gálvez

Benemérita Universidad Autónoma de Puebla, México
sanchez.galvez@correo.buap.mx

Mario Anzures García

Benemérita Universidad Autónoma de Puebla, México
mario.anzures@correo.buap.mx

Mariano Larios Gómez

Benemérita Universidad Autónoma de Puebla, México
mlarios77@gmail.com

Sully Sánchez Gálvez

Benemérita Universidad Autónoma de Puebla, México
ssanchez@cs.buap.mx

Recepción: 7/noviembre/2021

Aceptación: 21/diciembre/2021

Resumen

Una aplicación colaborativa apoya a un grupo de personas alcanzar conjuntamente un objetivo común a través de una interfaz de usuario compartida; siendo utilizadas estas aplicaciones en el ámbito educativo (e-learning), de comunicación (videoconferencias) de entretenimiento (redes sociales), de gestión (compartir archivos) y en muchos otros. Esta interacción entre los miembros del grupo implica una gran cantidad de peticiones (incluyendo concurrentes) de cada uno a la aplicación y entre ellos. Lo que implica realizar un manejo adecuado de las mismas para una pronta y adecuada respuesta. Por tanto, en este artículo se plantea desarrollar una aplicación colaborativa para la gestión de protocolos de tesis, funcionando como una API colocada en una Raspberry PI, que será el

servidor; para gestionar el almacenamiento, peticiones y respuestas de información de una manera simple y económica tanto para dispositivos móviles como aplicaciones en la Web.

Palabras Clave: API, aplicación colaborativa, desarrollo móvil y Web, protocolos de tesis, Raspberry Pi.

Abstract

A collaborative application supports a group of people to jointly achieve a common goal through a shared user interface; being used these applications in the educational (e-learning), of communication (videoconferences) of entertainment (social networks), of management (sharing files) environment, and in many others. This interaction between group members implies a large number of requests (including concurrent ones) from each to the application and between them. Which implies to carry out an adequate handling of them for a prompt and adequate response. Therefore, in this paper a collaborative application for the management of thesis protocols, is proposed; by functioning as an API placed in a Raspberry Pi. This will be the server; to manage the storage, requests and responses of information in a simple and economical way for both mobile devices and applications on the Web.

Keywords: *API, collaborative application, mobile and Web development, Raspberry Pi, thesis protocols.*

1. Introducción

La Raspberry Pi es una SBC (*Single Board Computer*) o computadora en una sola placa, hay muchas otras versiones; las primeras fueron SBC para la evaluación de procesadores como el 8080 de Intel o Z80 de Zilog. Este concepto de las SBC resurgió con la Raspberry. La cual fue desarrollada por la *Raspberry Pi Foundation*, cuyo principal objetivo fue facilitar la enseñanza de la programación para escuelas y personas que no pudieran invertir grandes cantidades en equipos de cómputo, pero al ser un dispositivo de gran versatilidad, buena capacidad de cómputo y un bajo precio, rápidamente ganó popularidad en diversos campos, siendo utilizada en

múltiples y variados campos, como pueden ser la robótica, automatización o inteligencia artificial. Conociendo estas características se puede crear cualquier aplicación colaborativa usando una Raspberry como servidor, para gestionar el almacenamiento, peticiones y retorno de información de una manera simple y sobre todo económica.

Las aplicaciones colaborativas asisten a grupos implicados en un objetivo común, mediante una interfaz para un entorno compartido [Anzures, 2018], [Anzures, 2021], [Grudin, 1997]. De tal manera, que el trabajo en grupo se realiza en un espacio de trabajo compartido (*entorno*) donde un conjunto de individuos, geográficamente distribuidos que comparten un interés común, se organizan, comunican, colaboran y coordinan para realizar tareas y alcanzar un objetivo común [Anzures, 2020], [Grudin, 1994].

En estos tiempos de emergencia sanitaria, las aplicaciones colaborativas se volvieron indispensables en empresas, organizaciones e instituciones de cualquier ámbito, obviamente, se encuentra incluido, el educativo; ya que al estar cada uno en su casa resulta más sencillo, utilizar y/o implementar una aplicación colaborativa. Existen diversas aplicaciones colaborativas, como: *Google Drive* [Google Drive, 2021] ofrece el servicio de almacenamiento de archivos y sincronización de estos con otros dispositivos; *Dropbox* [Dropbox, 2021] es un servicio multiplataforma de alojamiento de archivos en la nube, para almacenar, compartir y sincronizar tanto archivos como carpetas en línea entre computadoras, tabletas y móviles; *Microsoft Kaizala* [Microsoft Kaizala, 2021] es una aplicación de software de gestión de trabajo y mensajería segura para colaborar con otros en una organización, permitiendo enviar y recibir mensajes instantáneos, coordinar tareas y usar herramientas especiales para interactuar con su equipo; *Microsoft Teams* [Microsoft Teams, 2021] es un software cuyo objetivo es la colaboración en equipo, siendo su primordial función la mensajería empresarial para comunicarse y colaborar con miembros de la propia organización; *Trello* [Trello, 2021] es una plataforma para el desarrollo de proyectos que simplifica la colaboración y sirve para lo que uno quiera desarrollar, organizar o coordinar; y, *Ownccloud* [Ownccloud, 2021] es una aplicación web que permite acceder, compartir y sincronizar datos.

En la Facultad de Ciencias de la Computación (FCC) de la Benemérita Universidad Autónoma de Puebla (BUAP), existen diversas formas de titulación entre las que se encuentran: Tesis, Diplomado, CENEVAL, Créditos de Maestría, Experiencia Profesional y Experiencia Profesional Docente. Cada una de estas tiene sus propios requisitos y resulta bastante complejo gestionar todo el proceso involucrado con ellas. Este trabajo se centra en la titulación por tesis, en particular, en sistematizar los procesos relacionados con el protocolo de tesis para simplificarlo, agilizarlo, reducirlo en tiempo y costo. Tal protocolo es un documento, que permite plantear una idea o tema de interés y de impacto para considerarse un proyecto válido, que además contiene antecedentes, objetivos, metodologías, aportaciones y demás cuestiones que reflejen que el tema propuesto es acreedor a desarrollarse como tesis. Estableciendo las preguntas ¿Qué?, ¿Cómo?, ¿Cuándo? y ¿Por qué? se quiere desarrollar la tesis. El proceso relacionado con el protocolo de tesis es el siguiente:

- El estudiante elabora un protocolo de tesis, que entrega a secretaría académica.
- Secretaría académica asigna el protocolo a miembros de la comisión de tesis para su revisión.
- Cada revisor del protocolo, entrega su evaluación a secretaría académica que se lo devuelve al estudiante. Dicha evaluación, puede ser Aceptado, En revisión o Rechazado.
- Una vez que el protocolo es aceptado, el estudiante empieza a trabajar su sistema y documento de tesis junto con su(s) asesor(es).

Actualmente, el interesado utiliza los formatos que contienen todas las cláusulas que debe cumplir un protocolo y se centra a escribirlo de manera clara, delimitando todo lo que propondrá y realizará. Al finalizar este documento, un jurado dictaminara si el tema propuesto es aprobado, modificado para una nueva revisión o rechazado. Sin embargo, en el tiempo en el que vivimos la mayoría de los trámites se hace de manera digital, usando las herramientas que internet ofrece, conllevando a que sea más fácil y mejor la comunicación para solicitudes de cualquier índole.

En consecuencia, en este trabajo se propone desarrollar una aplicación colaborativa para la recepción, evaluación y validación del protocolo de tesis basada en una Raspberry Pi, que funge el rol de un servidor Web. Por tanto, en ella estarán alojadas tanto la base de datos como el dominio donde se dirigirán todas las peticiones (móviles o Web) e intercambio de información de la aplicación a desarrollar. De tal manera, que se simplifique y agilice el proceso, así como para permitir que el protocolo sea evaluado por más de un revisor y el estudiante conozca el estado de su protocolo: *Recibido* (lo tiene secretaría académica); *Asignado* (lo tienen los revisores); *En Evaluación* (lo están evaluando) y *Evaluado* (se establece el protocolo, como aceptado, en revisión o rechazado).

El alumno se podrá registrar con sus datos, teniendo acceso al formulario de protocolo de tesis, se validará toda la información para que cumpla con lo requerido, por medio de técnicas de expresiones regulares. Una vez enviada la información podrá revisar el estado de su protocolo. La aplicación implementará comunicación asíncrona y síncrona, con el fin de permitir a los revisores evaluar en tiempo real y que no suponga un problema, cuando varios lo hagan a la vez, además de poder reconocerse y diferenciarse entre sí, al momento de pedir un cambio en el protocolo se podrán dejar notas u observaciones con el nombre del revisor.

El documento se encuentra organizado de la siguiente forma: La Sección 2 presenta el método utilizado para el desarrollo de la Aplicación Colaborativa para la Gestión de Protocolos de Tesis basada en una Raspberry Pi. La Sección 3 muestra los resultados del desarrollo de la Aplicación Colaborativa para la Gestión de Protocolos de Tesis. La Sección 4 expone una breve discusión al respecto del desarrollo de la Aplicación Colaborativa para la Gestión de Protocolos de Tesis el desarrollo de la aplicación. Finalmente, en la Sección 5 se proporcionan las conclusiones y el trabajo futuro derivados de la Aplicación Colaborativa para la Gestión de Protocolos de Tesis.

2. Métodos

La elaboración del presente proyecto se basa en una metodología ágil [Beck, 2021], [Sommerville, 2010], ya que delimita, guía y concluye de manera eficiente tal

proyecto. Existen muchas metodologías que proporcionan diferentes herramientas y, por ende, diferentes calidades en los resultados. Como se quiere dar un buen resultado en poco tiempo, se ha elegido la metodología ágil XP, que ofrece significativas ventajas al centrarse directamente al desarrollo, pero sin dejar de lado los requisitos y alcances que se propongan. Para resumir la metodología ágil XP [Wells, 2021] dictamina 4 hitos importantes para el buen desarrollo de un proyecto y que son: *Planeación, Diseño, Desarrollo y Pruebas*.

Planeación

Se plantea el desarrollo de una aplicación para la recepción y evaluación de protocolos de Tesis, en dicha aplicación participan *alumno* (elaboran y envían dicho protocolo para su evaluación), revisor (recibe el protocolo a evaluar y regresan su dictamen con observaciones) y un administrador (recibe, asigna y envía la evaluación). Cada usuario tiene tareas y funciones diferentes, por ello, se idea una aplicación colaborativa que suministre un ambiente virtual para interactuar en procesos, como la creación, registro, evaluación y resultados del protocolo de tesis. De esta manera, se busca transformar un servicio que siempre necesitó que los involucrados se vieran en persona, a uno no presencial. En la figura 1 se presenta el modelo arquitectónico de la aplicación colaborativa para la gestión del servicio de protocolos de tesis, que está conformada de:

- **Servidor.** La Raspberry Pi funciona como servidor, es decir, es el contenedor de la API, base de datos, scripts y su principal tarea es proveer un ambiente seguro entre la aplicación y API. El servidor redirecciona las consultas desde la aplicación hacia la API y ejecuta los archivos y scripts alojados en éste. Además, contiene la dirección o IP única para establecer la comunicación y acceder desde cualquier parte. Por tanto, la Raspberry se configura instalándole un servidor (Linux), un gestor de base de datos (MySQL y MongoDB), PHP, estableciendo el puerto 80 y un dominio.
- **API.** Constituida por una serie de scripts en PHP con una lógica programada que gestiona todas las peticiones de los clientes (que en este caso son la aplicación móvil, Web y base de datos) a la aplicación, y dependiendo de la

información obtenida la API siempre devuelve una respuesta y sabe con precisión qué tipo de información regreso.

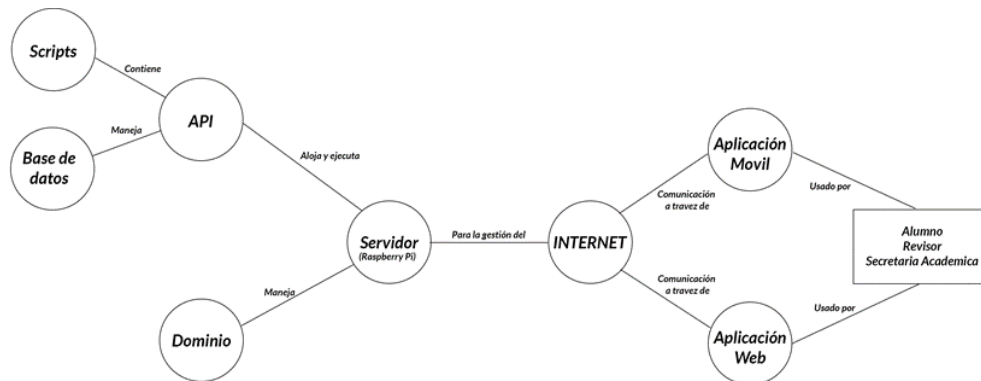


Figura 1 Arquitectura de la aplicación colaborativa de gestión de protocolos de tesis.

- **Base de Datos.** Almacena la información por medio de la API, que es el intermediario entre la aplicación y la base de datos. La cual contiene una serie de tablas que especifican una manera de guardar los datos, cada tabla tiene una estructura, comportamiento y uso. Se creó una base de datos híbrida, con Mysql y MongoDB para mayor seguridad, escalabilidad y contar con copias de seguridad.
- **Dominio.** Host virtual que presenta una dirección IP y puerto enmascarados para acceder a los recursos desde el Internet.
- **Scripts.** Archivos de configuración que permiten ofrecer varios servicios para promover el buen funcionamiento de la aplicación.
- **Aplicación Móvil.** Permite acceder a la aplicación a través de dispositivos móviles, utilizándola en cualquier momento y lugar. La aplicación requiere información al servidor, especificando tipo y forma, la petición se envía y recibe en formato JSON, transformado en un objeto para su manipulación.
- **Aplicación Web.** La aplicación colaborativa es accedida a través de computadoras personales, laptops o dispositivos que cuenten con un navegador web, utilizada en cualquier momento y lugar.
- **Usuarios.** Son las personas que están involucradas en el proceso de gestión de un protocolo de tesis.

- ✓ **Usuario alumno:** Complementa el formato de protocolo de tesis, entrega a secretaria académica y espera el resultado de su evaluación.
- ✓ **Administrador:** Recibe los protocolos y asigna a uno o varios revisores, al recibir el resultado, lo comunica al alumno.
- ✓ **Revisores:** Recibe y evalúa protocolos, mandan su respuesta a secretaria académica.

De acuerdo con esta arquitectura, la aplicación colaborativa se centra en tres vistas: alumno, revisor y administrador; que contienen funciones similares, véase las figuras 2, 3 y 4, tal como:

- **Pantalla de carga.** El usuario ve una pantalla de carga o splashscreen con el logo y nombre de la aplicación, esta pantalla durará 10 segundos. En segundo plano, sirve para comprobar los servicios y recursos esenciales que necesitara la aplicación para trabajar correctamente.
- **Registro.** Esta opción presenta un banner, seguido del texto “Regístrate” en mayúsculas. En la parte sobrante, se mostrarán una serie de campos que se solicitan para su registro al usuario correspondiente.

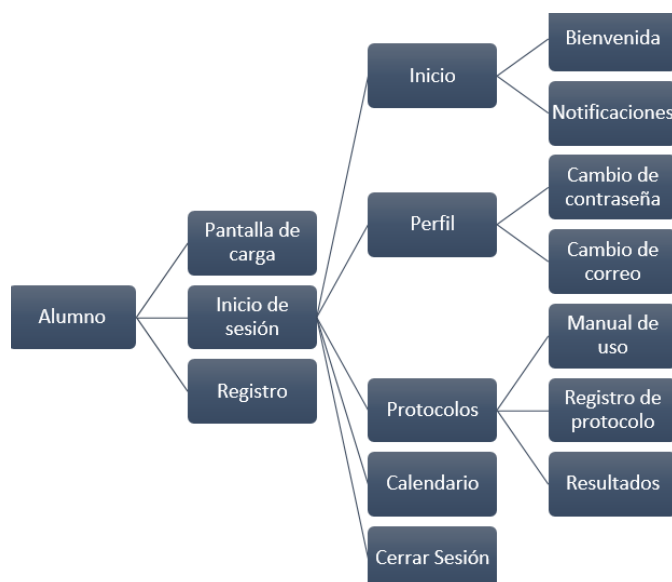


Figura 2 Esquema general de las funciones del alumno.

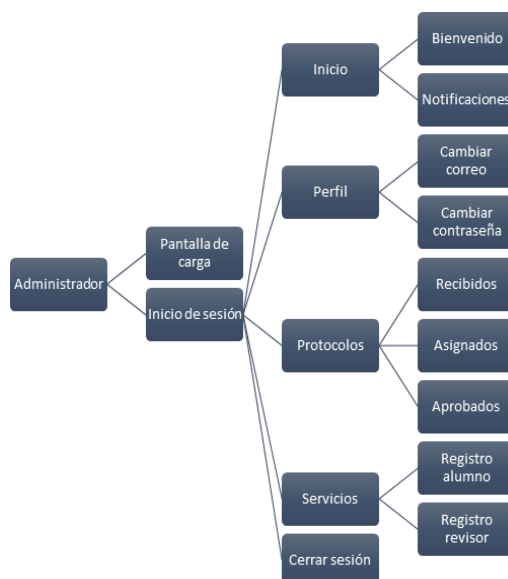


Figura 3 Esquema general de las funciones del administrador.

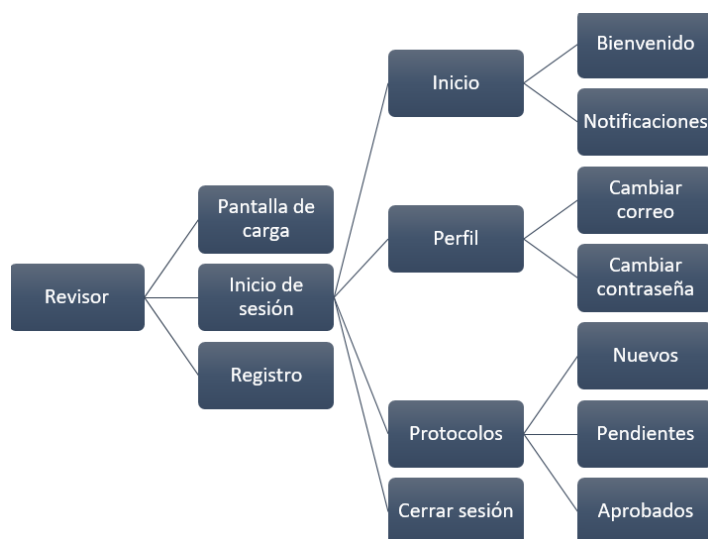


Figura 4 Esquema general de las funciones del administrador.

- **Inicio de sesión.** Se muestra cuando el tiempo asignado a Pantalla de carga expire. En esta opción se verá un banner con: Matrícula, Contraseña; Recordar (guarda matrícula y contraseña, para que cada vez que abra la aplicación no se soliciten dichos datos), Iniciar sesión, Registrarse (hipervínculo que lleva al usuario al Registro) y Recuperar contraseña (hipervínculo que despliega una ventana emergente solicitando el correo de la cuenta). Cuando se inicia sesión correctamente, el alumno visualiza:

- ✓ **Inicio.** Es la primera pantalla que el usuario ve al iniciar sesión correctamente o al registrarse. Esta sección contendrá dos pestañas a mostrar, quedando por defecto la primera.
 - **Bienvenido.** Contiene un mensaje de bienvenida para el usuario e información de interés.
 - **Notificaciones.** Cuando el usuario mande su protocolo de tesis, le llegan mensajes al usuario en cada etapa en la que se encuentre su documento, como: Protocolo enviado, aceptado, asignado, etc.
- ✓ **Perfil.** El usuario ve todos los datos que se le solicitaron cuando se registró. Siendo de mayor interés:
 - **Cambiar correo.** Solicita el correo electrónico anterior, el nuevo y la contraseña de la cuenta para confirmar. Se presiona el botón “Cambiar”, para actualizarlos, si es correcta la información; sino se muestran mensajes en rojo en los campos con errores.
 - **Cambiar contraseña.** Requiere la nueva contraseña, repetirla y la contraseña actual de la cuenta para confirmar. Se presiona el botón “Cambiar”, para actualizarlos, si es correcta la información; sino se muestran mensajes en rojo en los campos con errores.
- ✓ **Cerrar sesión.** Cuando el usuario presione dicha opción se muestran:
 - **Olvidar usuario.** Cuando el usuario verifica el checkbox “Iniciar sesión” se eliminan los datos almacenados y se redirige al usuario aquí. Sino de igual manera lo conduce a la opción mencionada.
 - **Salir.** Cierra la sesión del usuario y sale de la aplicación.

La vista del alumno, además, presenta las opciones, véase la figura 2:

- **Protocolos.** Contiene información y herramientas para la elaboración del protocolo de tesis. Se compone de tres subpestañas:
 - ✓ **Manual.** Presenta información de cómo realizar el protocolo de tesis y explica el uso de las herramientas que se ofrecen. Se guarda esta información en un archivo PDF, que se muestra cuando se requiera.
 - ✓ **Registro de protocolo.** Muestra el botón “Empezar”, al presionarlo se habilita el formulario que requiere toda la información necesaria para

crear este documento. El usuario también cuenta con un botón emergente que al presionarlo despliega las siguientes opciones:

- **Eliminar todo.** Se elimina toda la información del formulario.
- **Editar.** Se activa esta opción para editar el formulario, que por defecto está bloqueado.
- **Guardar.** Guarda la información contenida en el formulario.
- **Enviar.** Envía el formulario si la información es correcta, sino presenta un mensaje emergente con los errores existentes.
- ✓ **Resultados.** Se visualizan los diferentes estados al enviar el protocolo de tesis y los resultados obtenidos al evaluar el protocolo:
 - **Protocolo no enviado.** Aún no se ha enviado o está en elaboración.
 - **Protocolo enviado.** Se envía y está a la espera de ser aceptado.
 - **Protocolo rechazado.** Cuando se considera que el protocolo no cumple con el formato especificado, contiene o falta información.
 - **Protocolo a revisión.** Cuando fue aceptado y se requieren cambios.
 - **Protocolo aceptado.** Cuando tiene una evaluación positiva.
- **Calendario.** Presenta las fechas en que se enviará el protocolo de tesis, así como aquellas en las que recibirá su resultado de evaluación.

La vista del administrador que además de las funciones comunes, tiene las opciones, véase la figura 3.

- **Opción de protocolos.** Concentra las funciones principales, realizando la recepción de los protocolos enviados, revisión preliminar, asignación de protocolos a los revisores y evaluación realizada. Tiene tres subpestañas:
 - ✓ **Recibido.** Se listan los protocolos enviados por los alumnos, identificándose cada uno por el nombre del alumno, el título del protocolo y la fecha en que lo envió. Al darle click a un elemento de la lista, dirige al usuario al protocolo completo seleccionado, se visualiza toda la información del formulario, en la parte final de protocolo el

usuario podrá ver adjunto las firmas del alumno y asesores, así como un botón que al darle clic descarga las identificaciones del alumno. En la parte inferior derecha se encuentra un botón que contiene:

- **Aceptar.** El protocolo se muestra en una ventana emergente listando los revisores disponibles para evaluarlo, el usuario podrá escoger entre 1 y 2 revisores, si selecciona más de 2 o ninguno obtendrá un mensaje de error. Si la selección es correcta, el protocolo ha sido asignado exitosamente y se moverá a la subpestaña asignados.
- **Rechazar.** Se abre una ventana emergente para confirmar la decisión y un espacio para escribir el motivo por el cual fue rechazado el protocolo, manteniéndolo en la subpestaña recibidos, pero en color rojo para distinguirlo de los demás.
- **Descargar documento.** Al pulsar esta opción el usuario obtendrá una copia en formato Word del protocolo seleccionado.
- ✓ **Asignado.** Presenta los protocolos asignados por el administrador a los revisores, esta sección servirá para verificar el progreso y evaluación que estén realizando los revisores. El usuario podrá ver si ya se evaluó el protocolo o se agregaron notas para solicitar cambios. Cuando los revisores hayan concluido sus evaluaciones, el administrador envía el resultado al alumno, cuando es positivo el protocolo se mueve a la subpestaña aprobados, si se necesitan cambios se mantiene en esta subpestaña cambiando su color a naranja.
- ✓ **Aprobados.** Lista los protocolos aprobados, que tienen color verde, si el usuario le da clic se abrirá y visualiza su información, también puede descargarlo en formato *word*.
- **Servicios.** En esta opción se concentran funciones para habilitar o deshabilitar los registros de las vistas del alumno y revisor. Existen 2 botones, al presionar uno de ellos se le pedirá al usuario elegir, si aprueba o deniega que la opción para registrarse este activa en la aplicación.

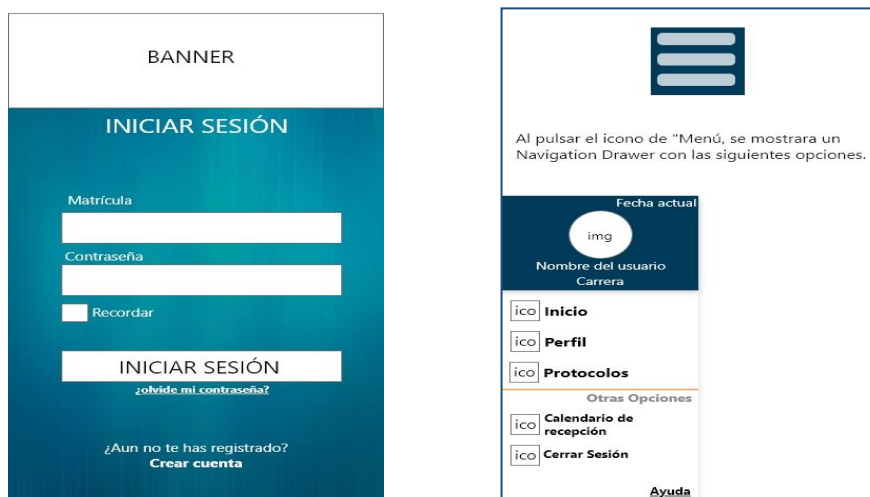
La vista del revisor que se presenta en la figura 4 presenta las siguientes funciones, además, de las comunes.

- **Nuevos.** Lista los protocolos que tiene asignados, cada elemento despliega el nombre del alumno, título del protocolo y fecha de envió. Al darle clic a cualquier elemento se muestra el protocolo. En la esquina inferior derecha se visualiza un botón que presenta las opciones:
 - ✓ **Evaluar.** Abre una ventana emergente para seleccionar una opción:
 - Aceptar. Se aprueba el protocolo.
 - Revisar. Se necesitan algunos cambios, que se indican mediante notas en los campos que requieren corrección o cambios, sino deja notas, no se envía la evaluación.
 - Rechazar. Se rechaza el protocolo, se dejan notas indicando los motivos, sino hay notas no se puede enviar la evaluación
 - Agregar notas. Habilita botones en distintas partes del protocolo, al pulsar cualquier botón despliega una ventana emergente y un campo para escribir la información que se requiera en esa parte.
 - Descargar documento. Se obtiene el protocolo en formato Word.
 - ✓ **Pendiente.** Contiene los protocolos enviados a revisión o aún sin revisar, listándolos y al darle clic se muestra un menú con dos opciones:
 - **Ver Evaluación.** Presenta el nombre de los revisores asignados y el resultado de la evaluación.
 - **Ver Protocolo.** Despliega el protocolo, las notas que presentan un color diferente, si hay más de un revisor, para identificar a cada uno.
 - ✓ **Aprobados.** Visualiza los protocolos aprobados en forma de lista en color verde claro, al seleccionar uno, despliega el protocolo, también puede “Descargar archivo” en formato Word.

Diseño

Considera todo lo especificado en la planeación, para determinar las vistas: alumno, revisor y administrador; así como estableciendo su estética y apariencia

visual, el orden de sus elementos, paleta de colores, tipo y tamaño de fuentes, gráficos e imágenes a utilizar. Este proyecto usa los estándares de *Material Design* para ello. A partir de lo anterior, se elaboran los *mockups* de cada vista, por cuestiones de espacio sólo se presentan dos: inicio de sesión figura 5a y vista general del alumno figura 5b.



a) Mockup de inicio de sesión

b) Mockup de alumno

Figura 5 Mockup.

Desarrollo

En esta sección, se hace la codificación especificando los paquetes, clases, objetos, métodos y funciones de la aplicación colaborativa para gestionar el protocolo de tesis, de acuerdo con lo mencionado en la planeación y diseño. Uno de los aspectos esenciales en la programación, es la plantilla del protocolo de tesis (<https://secreacademica.cs.buap.mx/Servicios-Ecolares.html>), ya que se utiliza en las tres vistas que comprende la aplicación. Por tanto, la plantilla se divide por secciones y, posteriormente, se almacena en tablas en la base de datos, cuando la aplicación requiera estos datos, se le devuelven en formato JSON; así que se crea una estructura JSON de la plantilla y las respuestas que se esperan. Logrando así reducir el tamaño en bytes del archivo y agilizando las peticiones de la plantilla entre el API y las aplicaciones Web y móvil. El desarrollo de la aplicación es el principal resultado y del cual se derivan los demás.

Pruebas

Se probó cada una de las interfaces de usuario de las tres vistas: *alumno*, *revisor*, *administrador*; detectando y corrigiendo los errores. Una vez que cada interfaz fue aceptada, se integraron y se realizaron las pruebas correspondientes, obteniendo los resultados especificados en la planeación y diseño. También, se evaluó el rendimiento de la aplicación para proceder a su optimización, corrección, mejora y refactorización final del sistema. Finalmente, se llevaron a cabo pruebas SandBox, que se hacen cuando el sistema ya está terminado, éstas las ejecutó el programador; analizando el comportamiento de la aplicación y creando versiones que corrigieron y mejoraron los resultados de estas.

3. Resultados

Los principales resultados son: una Aplicación colaborativa para gestionar el servicio de Protocolo de Tesis; la Raspberry Pi funcionando como un servidor; una API; un Cliente, aplicación móvil; y un Cliente, aplicación Web.

Aplicación colaborativa

El resultado más importante de este artículo es la aplicación colaborativa de la gestión del protocolo de tesis, que permite la comunicación, colaboración y coordinación del alumno, revisor y administrador en un espacio virtual; para crear, evaluar y dictaminar tales protocolos.

En la figura 6 se establece el funcionamiento general de esta aplicación de acuerdo con tres usuarios; *alumno*, *revisor* y *administrador*; que fueron descritos en la sección: Métodos. Por tanto, se presenta una interfaz por cada usuario y otra general, véase las figuras 7, 8, 9 y 10.

Servidor: Raspberry Pi

Otro resultado esencial, que se deriva de la aplicación y que contribuye con su bajo costo, alto rendimiento y seguridad, es programar la Raspberry Pi Zero w como un servidor. Con esto, el proceso de elaboración, evaluación y dictaminación de un protocolo se simplifica y agiliza, de tal manera que, puede haber varios estudiantes

creando su protocolo, más de un revisor evaluando cada uno de éstos, y siendo ejecutada la aplicación en varias Raspberry Pi Zero w.

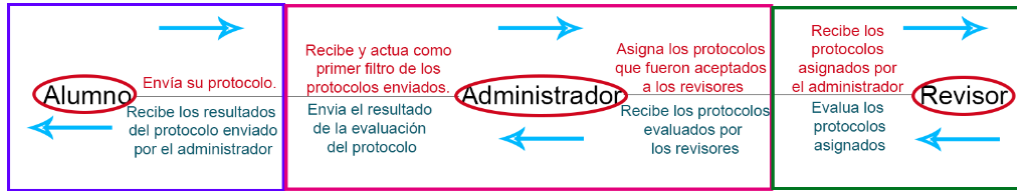


Figura 6 Funcionamiento general de las vistas.



Figura 7 Inicio de sesión de la aplicación.



Figura 8 Alumno: registro de protocolo.



Figura 9 Revisión del protocolo.

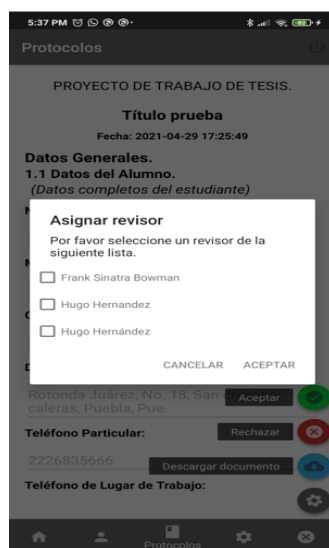


Figura 10 Asignar protocolo.

Servidor: Raspberry Pi

Otro resultado esencial, que se deriva de la aplicación y que contribuye con su bajo costo, alto rendimiento y seguridad, es programar la Raspberry Pi Zero w como un servidor. Con esto, el proceso de elaboración, evaluación y dictaminación de un protocolo se simplifica y agiliza, de tal manera que, puede haber varios estudiantes creando su protocolo, más de un revisor evaluando cada uno de éstos, y siendo ejecutada la aplicación en varias Raspberry Pi Zero w.

Servidor: Raspberry Pi

Otro resultado esencial, que se deriva de la aplicación y que contribuye con su bajo costo, alto rendimiento y seguridad, es programar la Raspberry Pi Zero w como un servidor.

Con esto, el proceso de elaboración, evaluación y dictaminación de un protocolo se simplifica y agiliza, de tal manera que, puede haber varios estudiantes creando su protocolo, más de un revisor evaluando cada uno de éstos, y siendo ejecutada la aplicación en varias Raspberry Pi Zero w.

API

Un resultado, también de gran importancia, es el desarrollo de la API que se aloja en el servidor, Raspberry Pi, haciendo que las aplicaciones móviles, Web y base de datos actúen como un servicio Web, que simplemente hace las peticiones necesarias al servidor y reciben el resultado esperado en el formato apropiado. Así que esto permite que la comunicación sea simple, ágil, ligera (se comunican documentos JSON) y segura.

Cliente: aplicación móvil

El desarrollo de la aplicación móvil es un resultado que se adecua al contexto actual y sobre todo al de los alumnos, que en su mayoría son jóvenes, quienes utilizan con gran frecuencia y familiaridad los dispositivos móviles, en particular, su celular. Lo que hará que esta aplicación colaborativa sea empleada por ellos con mayor periodicidad, además de que lo podrán hacer cuando y donde quieran.

Cliente: aplicación Web

El administrador y revisores son personas mayores que están más adaptadas al uso de computadoras personales y laptops, con relación a los jóvenes, Por ello, un resultado importante para ellos es la creación de una aplicación Web véanse las figuras 11 y 12, de tal manera, que tanto que éstos dos usuarios puedan utilizar la aplicación colaborativa desde su oficina, casa o salón de clases, con tan sólo acceder a la dirección Web donde se encuentra alojada la aplicación colaborativa.

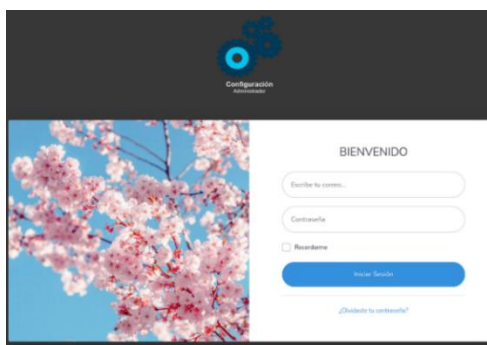


Figura 11 Página web de inicio.

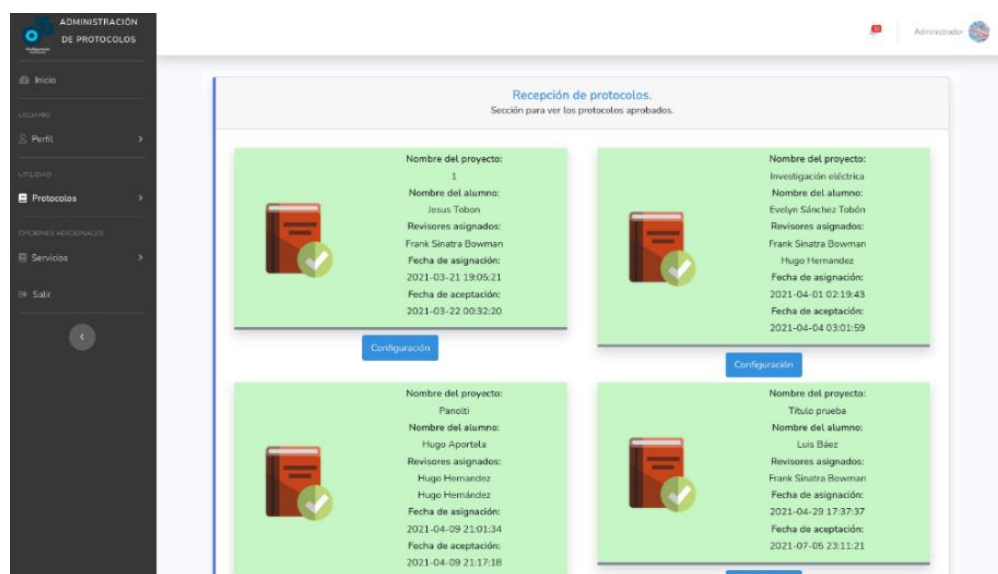


Figura 12 Página web de recepción de protocolos.

4. Discusión

Promover el uso de la Raspberry como un servidor web, a pesar que, se ocupa principalmente para proyectos orientados a la mecatrónica y electrónica, ha sido un gran acierto. Debido a su bajo costo, alto desempeño y extensa variedad de herramientas que permiten desarrollar aplicaciones complejas que proporcionan un gran rendimiento, seguridad y rápida respuesta. Este hecho condujo a crear una API, que simplifica y agiliza las peticiones, más aún, cuando se han desarrollado como *servicios Web*, una aplicación móvil y una Web que hacen las peticiones necesarias, para el buen funcionamiento de la aplicación colaborativa que gestiona el protocolo de tesis, donde participan alumno, revisor y administrador.

5. Conclusiones

Se ha desarrollado una aplicación colaborativa para la gestión del protocolo de tesis, requisito en algunas instituciones de educación superior para realizar una tesis de grado. Esta aplicación permite la comunicación, colaboración y coordinación entre aplicaciones (API con base de datos, aplicación Web o móvil) y usuarios (*alumno, revisor y administrador*). Además, se ha creado una API alojada en un servidor, una Raspberry Pi, que puede ser configurada para diferentes unidades académicas de la BUAP o de otras universidades; sólo se tendría que configurar la plantilla del protocolo para adaptarse a cada dependencia o institución. El hecho de usar la Raspberry ha reducido costo, tiempo y esfuerzo en el desarrollo y resultado. El trabajo futuro sería portar la aplicación a un IDE de desarrollo híbrido, que proporcione la creación de la aplicación Android e iOS para que pudiera ser usada en cualquier dispositivo móvil.

6. Bibliografía y Referencias

- [1] Anzures García, M., Sánchez Gálvez, L. A., Hornos, M., & Paderewski, P., Tutorial Function Groupware Based on a Workflow Ontology and a Directed Acyclic Graph. *IEEE Latin American Transactions* 16-1, 294-300, 2018.
- [2] Anzures García, M., & Sánchez Gálvez, L. A., PRoposing an Ontological Model for developing collaboratlve SystEms. *Journal Intelligent & Fuzzy Systems* 39-2, 2545 – 2557, 2020.
- [3] Anzures García, M., Sánchez Gálvez, L. A. & Campos Gregorio, A., Weighted Bidirectional Graph-based Academic Curricula Model to support the Tutorial Competence *Journal CyS*, 24-2, 619–631, 2020,
- [4] Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Thomas, D., (2018). *Manifesto for Software Agile Development*, <http://agilemanifesto.org/>.
- [5] Dropbox, (2021). <https://www.dropbox.com/>.
- [6] Google Drive, (2018). <https://www.google.com/intl/es/drive/>.
- [7] Grudin, J., *Computer-Supported Cooperative Work: Its History and Participation*. *IEEE Computer*, 27-5, 19-26, 1994.

- [8] Grudin, J. & Poltrock, S. E. Computer-Supported Cooperative Work and groupware. In: *Advances in Computers*, 45, 269–320, 1997.
- [9] Microsoft Kaizala, (2019). <https://www.microsoft.com/es-mx/microsoft-365/business/microsoft-kaizala>.
- [10] Microsoft TEAMS, (2020). <https://www.microsoft.com/es-mx/microsoft-teams/group-chat-software>.
- [11] Owncloud, (2020). <https://owncloud.com/>.
- [12] Sommerville, I., *Software engineering*. 9ª ed. Addison Wesley, Boston, 2010.
- [13] Trello, (2021). <https://trello.com/>.
- [14] Wells, D., (1999). *XP, Lessons learned: pair programming* [1999], <http://www.extremeprogramming.org/rules/pair.html>.