BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

CREACIÓN DE UNA APLICACIÓN MÓVIL INFORMATIVA PARA LA ASOCIACIÓN SINDICAL DEL PERSONAL ACADÉMICO DE LA BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA (ASPABUAP), UTILIZANDO UN FRAMEWORK

T E S I S

PARA OBTENER EL TÍTULO DE:

INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A:

JOSÉ ANTONIO ÁLVAREZ GÓMEZ

DIRECTOR DE TESIS:
M.C. CARLOS ARMANDO RIOS ACEVEDO
Noviembre del 2019

Agradecimientos

No tengo ninguna forma de mostrar cuan agradecido estoy con mi familia, mi padre, mi madre y mi hermana por todo el apoyo que me dieron durante toda mi carrera y en general por todo el apoyo que he recibido de ellos durante toda mi vida, estoy eternamente agradecido con ellos por todo su esfuerzo y apoyo incondicional para que yo concluyera mi carrera universitaria, por ello les dedico totalmente este gran logro esperando que se sientan orgullosos no solo de este logro y de mí, si no de ellos mismos por ser unos padres ejemplares, agradezco infinitamente también a los profesores que desde la preparatoria me inspiraron a seguir adelante en especial al Maestro Carlos Ríos por ser un ejemplo a seguir además de un excelente académico, a la Maestra María Pérez de Celis por su gran impacto innovador en mí y su lucha para que los alumnos se vuelvan emprendedores. A la Ingeniera Mariana Romero Tafoya por su apoyo, compañía, paciencia y tolerancia durante estos 4 años de carrera a mi lado. Quiero agradecer también al diseñador Daniel Terés por su apoyo en el diseño UX/UI de la aplicación.

Por último quiero agradezco de igual forma a mis amigos más cercanos (Erick, Lalo, Yael, Beto y Toxqui) por siempre quedarse a mi lado en esas noches de estudio a las que no le veíamos final, por compartir la misma pasión hacia la computación como yo y por siempre mantener la amistad que nos caracteriza.

Tabla de contenido

Capítulo I Introducción _____ *4*

1.1 Resumen _____ *5*

1.2 Justificación _____ *6*

1.3 Antecedentes del Proyecto. _____ *7*

1.4 Objetivos Generales y Específicos del Proyecto. _____ *7*

1.5 Estado del Campo o del Arte _____ *8*

1.6 Metodología. _____ *8*

Capítulo II Marco Teórico _____ *9*

2.1 Definición de Aplicaciones móviles. _____ *10*

2.2 Importancia y objetivos de las Aplicaciones Móviles. _____ *10*

2.3 Tipos de Aplicaciones Móviles. _____ *11*

2.4 Componentes de las Aplicaciones Móviles. _____ *11*

2.5 Criterios para Publicación de las Aplicaciones Móviles. _____ *12*

2.6 Metodología Scrum. _____ *12*

2.7 Modelado UML. _____ *15*

2.8 Algoritmo de Geolocalización. _____ *16*

Capítulo III Análisis del sistema y Diseño del Sistema _____ *17*

3.1 Diagramas de Casos de Usos. _____ *18*

3.2 Diagramas de Casos de Usos Específicos. _____ *19*

3.3 Diagrama de Clases del Sistema _____ *24*

3.4 Diagramas de Secuencia _____ *25*

3.5 Diagrama de la Base de Datos. _____ *28*

3.6 Pilas de Productos. _____ *29*

3.7 Conclusiones del Sistema. _____ *32*

Capítulo IV Puesta en marcha del Sistema. _____ *33*

4.1 Pantallas del sistema. _____ *34*

4.2 Metodos principales del sistema. _____ *43*

Bibliografía _____ *49*

Capítulo I Introducción

1.1 Resumen

Actualmente, la tecnología juega un papel importante en la vida cotidiana, la digitalización permite el flujo de información de manera más eficaz, por lo cual, según Jiménez, A., en su artículo “La digitalización de información es en realidad sinónimo de preservación y difusión”, las formas tradicionales de publicación de información serán en unos años obsoletas. (2006)

ASPABUAP ocupa un método tradicional para hacer llegar información por medio de carteles y lonas en sitios. Basándose en Duran O. y Carranza E. ha aumentado la dependencia de la información en formato digital en organismos gubernamentales, universidades y empresas; en cualquier tipo de institución, las cuales actualmente requieren para sus procesos documentos digitales los cuales no se encuentran disponibles en otros formatos (2016).

Por su parte los sindicatos dentro de la república mexicana se encuentran renovando sus tecnologías para digitalizar el acceso a la información, tal es el caso de la UNAM (Schwab, 2016) que se encuentra desarrollando su aplicación. Las principales universidades del país que cuentan con un sindicato aun no han desarrollado las herramientas tecnológicamente necesarias, esto es resultado de no mantener sus tecnologías actualizadas. Los principales sindicatos del país cuentan con una página web de carácter informativo que, aunque cumple su función no está acorde a la vanguardia tecnológica, lo cual impide al usuario observar e investigar desde un dispositivo móvil accediendo a internet.

Se analizó a detalle los 3 principales sindicatos asociados a universidades en México; UNAM, UAM, y Politécnico Nacional. De ellos solo uno posee una página funcional y actualizada a las nuevas tecnologías, por lo cual los dos restantes sindicatos cuentan con un punto de información obsoleto. Se pretende con el siguiente proyecto la creación de la APP ASPABUAP para crear un punto de arranque en innovación y competencia tecnológica entre sindicatos con la finalidad de favorecer a los docentes como usuarios finales de la misma.

1.2 Justificación

ASPABUAP es un sindicato que agrupa alrededor de 3800 trabajadores académicos de la BUAP, ofreciendo servicios de: manejo de contrato colectivo de trabajo, interpretación de conceptos en recibos de nómina, asesoría y requerimiento en trámites del hospital universitario de Puebla, asesoría y seguimientos a conflictos laborales, divulgación de convenios con hoteles, restaurantes y clubes deportivos, y la tarea más importante es gestionar y asesorar a todos los agremiados en beneficio de los derechos laborales. Las oficinas sindicales es donde los afiliados pueden recibir atención de parte del secretario general y del comité ejecutivo general dividido en secretario del interior y relaciones exteriores, secretario de organización y prensa, secretario de conflictos laborales, secretario de instrucción académica y secretario de cultura y eventos deportivos.

Sin embargo, la nueva tendencia tecnológica marca la pauta para ingresar en una actualización tecnológica que permita vincular a los afiliados con el sindicato y obtener beneficios en todas las áreas que contempla la función orgánica del mismo. Por lo anterior la APP denominada APP ASPABUAP será de gran utilidad para vincular, coordinar, difundir y concientizar el uso de la TIC'S hacia un sindicato con una visión de industria 4.0. El desarrollo de la APP será utilizando una metodología ágil que potencialice los procesos dinámicos en las etapas de análisis, diseño, implementación y pruebas del ciclo del software, logrando con ello un proceso ágil y eficaz que en 6 Springs (ciclos de desarrollo), hagan tangible la creación e implementación de APP ASPABUAP.

El 42% de la población mundial esta conectada a internet, se prevé que para el 2020 los dispositivos móviles sean la principal forma en que las personas puedan acceder a internet (Canavilhas, J. 2015).

Esta APP simplificará la manera en la cual los afiliados se enteren de los procesos, convocatorias y convenios que existen dentro del sindicato, agilizando dichos procesos y facilitando la forma en que el docente pueda acceder a todos los beneficios.

1.3 Antecedentes del Proyecto.

Hoy en día el ASPABUAP cuenta con diferentes mecanismos de divulgación de sus eventos y actividades sindicales; los cuales son: periódico opinión universitaria en donde cada semana se notifica de las actividades del sindicato, periódico digital "soy universitario" donde se coloca también una nota diario de eventos y acontecimientos sindicales, carteles impresos que se colocan en cada una de las unidades académicas de la BUAP y finalmente la página www.aspabuap.com.mx donde se divulgan actividades y eventos.
Sin embargo, actualmente el usuario demanda el uso de la tecnología para facilitar la divulgación de información, cada docente activo actualmente dentro de la universidad que se encuentre afiliado al sindicato o que en su caso quiera afiliarse se encuentra con que para poder realizar un trámite tiene que acudir a las oficinas centrales de ASPABUAP además tiene que enterarse de las convocatorias por carteles pegados en cada dependencia universitaria, por ello y teniendo una visión a una industria 4.0 no solo como un paso para ingresar el ASPABUAP a una era digital si no como un requerimiento actual del uso de las nuevas tecnologías se busca actualizar por completo la forma en que el usuario en este caso el docente reciba la información y se mantenga actualizado de los sucesos más importantes inmediatamente después de su publicación ahorrando costos y tiempo.

1.4 Objetivos Generales y Específicos del Proyecto.

Objetivo general

Desarrollar una APP denominada APP ASPABUAP que permita a los afiliados conocer convenios, actividades, eventos y la vida sindical del ASPABUAP.

Objetivos específicos

1. Analizar mediante una API de Google todos los convenios cercanos utilizando GPS para recomendar un mejor servicio a los afiliados.
2. Implementar una interfaz móvil agradable para cualquier nivel de usuario.
3. Hacer uso de los Webservices para que las noticias, actividades y eventos sean publicados y visualizados en tiempo real.
4. Implementar una estructura de convenios que sea validada en cada establecimiento afiliado.
5. Generar una base de datos general para que los afiliados observen su estado actual dentro del sindicato, así como sus trámites.

1.5 Estado del Campo o del Arte

Actualmente, los sindicatos dentro de la república mexicana se encuentran renovando sus tecnologías para ingresar a la industria 4.0 tal es el caso de la UNAM (Schwab, 2016) que se encuentra desarrollando su aplicación y aunque las principales universidades del país cuenten con un sindicato están lejos de entrar directamente a la industria 4.0 ya que no mantienen sus tecnologías actualizadas o hacen uso del IOT, los principales sindicatos del país cuentan con una página web de carácter informativo que aunque cumple su función se aleja bastante de ir a la vanguardia tecnológica, esto porque la nueva tendencia tecnológica es observar e investigar desde un dispositivo móvil accediendo a internet.

No solo los sindicatos no quieren invertir en una infraestructura adecuada para los medios de difusión si no que los que tienen uno se encuentra obsoleto. Esto es un gran punto para la APP ASPABUAP ya que despertaría ese instinto de innovación y competencia tecnológica entre sindicatos apoyando sobre todo a los docentes como usuarios finales de la misma.

1.6 Metodología.

Al tratarse de una aplicación móvil se utilizará la metodología ágil scrum con xp designando entregas parciales previamente planeadas y aprobadas, creando una lista de entregas programadas contra tiempo (springs), cada tarea designada no podrá ser adelantada bajo ninguna circunstancia ya que primero deberá completarse la tarea actual esto en forma de control para realizar una aplicación organizada y apegada a los tiempos establecidos.

Como se puede visualizar en el siguiente apartado el proyecto se dividirá en 16 semanas (4 meses) con un total de 6 iteraciones y cada iteración tomará 2 semanas como fecha límite de entrega, cada iteración esta diseñada para que se lleve a cabo en una manera organizada y en cascada, es decir, si la iteración de la semana 1-2 no se cumple a tiempo la iteración siguiente no podrá ser iniciada.

Como punto importante cada iteración tiene actividades específicas que deben cumplirse de manera ordenada para que exista un correcto flujo de trabajo y con ello se aumente la eficacia y se reduzcan los tiempos. En la siguiente tabla se muestra las actividades establecidas con sus tiempos en semanas dando el total de 16 semanas y 6 iteraciones (Kniberg, 2007).

Capítulo II Marco Teórico

2.1 Definición de Aplicaciones móviles.

Según la “Comisión federal del comercio e información para los consumidores”, una aplicación móvil es aquel programa o software que puede ser descargado directamente a su dispositivo inteligente (Smartphone) cuyo objetivo principal es brindar un servicio o entretenimiento al usuario. “Las app’s” tienen su origen al final de los años 90’s con la llegada de los dispositivos móviles que no solo se limitaban a realizar llamadas, en un caso particular la marca “Nokia” implemento una aplicación de entretenimiento llamada “Snake” la cual marcaria el inicio de las llamadas “app’s” contando ahora no solo con juegos como snake, si no también con agendas, calendarios, tonos de llamada etc. Debemos de recordar que las “app’s” tienen su origen mucho antes de que existieran los teléfonos inteligentes, puesto que cada utilidad que desempeñe el dispositivo que no sea la de llamar es considerada como una aplicación móvil.

Las Aplicaciones Móviles se han convertido hoy en día en un factor vital en el uso de dispositivos móviles ayudando a implicar las tareas, la comunicación y el entretenimiento de una manera jamás vista, gracias a las “app’s” podemos realizar un sin fin de tareas que antes resultaban laboriosas o complicadas.

2.2 Importancia y objetivos de las Aplicaciones Móviles.

Las aplicaciones móviles tienen un papel de suma importancia en la vida cotidiana de cada persona que es poseedora de un dispositivo móvil, según INEGI, el 73% de la población de 6 años o más utiliza o utilizo un dispositivo móvil, es decir, la población mexicana que usan el dispositivo móvil paso de 64.7 millones de personas en el 2017 a 69.6 millones en el 2018. Ahora, del total de usuarios de los “Smartphones” 45.5 millones instalaron aplicaciones en sus dispositivos, tales como aplicaciones de mensajería, herramientas de redes sociales y aplicaciones de audio y video (INEGI, COMUNICADO DE PRENSA NÚM. 179/19, 2 ABRIL DE 2019). Gracias a estos datos podemos asumir que el mercado de las aplicaciones móviles seguirá creciendo en los próximos años es por ello que la tecnología y los avances tecnológicos demandan el uso de aplicaciones móviles.

Las aplicaciones móviles basan su desarrollo en 3 objetivos principales;

1. Interacción: Lograr que un usuario final sin conocimientos previos en tecnología, interactúe con su dispositivo de una manera fácil, amigable y confiable.
2. Eficiencia: Mejorar el trabajo y las tareas que el usuario desempeña día con día, ahorrando tiempo y costes.

3. Conectividad: Permite al usuario estar siempre comunicado con otros usuarios, a diferencia de las llamadas aquí se hace uso de las trasferencias de información y datos (documentos, fotos, etc.).

2.3 Tipos de Aplicaciones Móviles.

Publicas: Las aplicaciones públicas son aquellas que son publicadas en tiendas digitales tales como la “Play Store o App Store” se dice que son públicas ya que cualquier usuario de un Smartphone puede descargarlas. En esta categoría se encuentran una gran variedad de subcategorías de aplicaciones tales como: Salud, Belleza, finanzas, estilo de vida, Juegos, etc.

Privadas: Las aplicaciones privadas son aquellas que fueron desarrolladas para un ambiente especifico ya sea una empresa o una institución y no son accesibles a todo el público, dentro de estas aplicaciones contamos con APP BUAP y es aquí donde se clasificaría nuestra aplicación.

Nativas: Las aplicaciones nativas son aquellas que fueron desarrolladas en su lenguaje nativo es decir para un Smartphone con un sistema operativo “Android” se requiere un lenguaje de desarrollo llamado JAVA y para un Smartphone con un sistema “IOS” se requiere de un lenguaje llamado “SWIFT”. cuando una aplicación es desarrollada en sus lenguajes se dice que es una aplicación nativa.

Hibridas: Las aplicaciones hibridas son aquellas se desarrollan en un lenguaje de programación ajeno como HTML y CSS. Estas aplicaciones son ejecutadas en ambas plataformas sin la necesidad de crear la misma aplicación dos veces.

2.4 Componentes de las Aplicaciones Móviles.

Según el tipo de aplicación y su clasificación se desglosan los componentes de la misma en este caso colocaremos los componentes generales que se requiere para una aplicación móvil.

Interfaz gráfica: Diseño y estructuración de la aplicación, este es el componente donde el usuario final interactúa.

Contenido: Se refiere a todo el contenido estructurado que la aplicación contiene.

Usabilidad: características de la aplicación para que se le permita al usuario final la interacción fácil y amigable con la aplicación.

Desempeño/optimización: Componente que le permite a la aplicación usar correctamente los recursos del sistema tales como almacenamiento y memoria RAM.

Seguridad: Componente que permite resguardar y garantizar la confidencialidad del usuario.

2.5 Criterios para Publicación de las Aplicaciones Móviles.

Para publicar una aplicación móvil se requiere de ciertos criterios o lineamientos, esto se maneja más como un estándar al momento de lograr que la aplicación sea presentada en las tiendas (App Store y Play Store).

Logotipo: Logotipo de la aplicación creada que deberá ser mostrado en la información de la tienda donde se oferte.

Pantallas de ejemplo: Pantalla que muestren el funcionamiento de la aplicación y que den una idea al usuario de cómo se comporta la aplicación

Versión de la aplicación: Versión compilada de la aplicación al momento de su publicación.

Información general de la aplicación: Esta información debe contener los siguientes datos: Dueño de la aplicación, correo de contacto, desarrollador, derechos de autor, política de privacidad, términos y condiciones, instrucciones de uso, nombre de la aplicación, descripción de funcionalidad, lenguaje ciudadano, tamaño de la aplicación, requerimientos técnicos, servicios api utilizados, recursos y permisos usados, reporte de validación.

(Gobiernos locales, 2017)

2.6 Metodología Scrum.

La metodología SCRUM es definida como un conjunto de procesos organizados que llevan al cumplimiento de un objetivo propuesto, principalmente está diseñado

para trabajar en equipo y asignar una serie de tareas que tienen que ser completadas y mostradas en un tiempo establecido, esta metodología se caracteriza por entregas rápidas y cuenta con la capacidad de realizar cambios al producto final de forma rápida y sin que afecte o genere un impacto mayor en el tiempo total del desarrollo. Para aplicar SCRUM debemos tener una serie de conocimientos previos los cuales serán redactados en esta sección del documento.

Product backlog: El product backlog está definido como la lista organizada y detallada de todas las tareas que deben completarse para llegar a la conclusión del producto entre dicha lista se debe contemplar los requerimientos, las historias de usuario y los tiempos.

El Product Backlog tiene su fuerza en las historias de usuario las cuales se conforman de los siguientes datos:

ID: Un identificador único, que se incrementará de uno en uno.
Nombre: Es un nombre corto y descriptivo por historia de usuario, debe ser muy claro para los desarrolladores y clientes.
Prioridad: Es la cantidad de importancia que un cliente asigna a cada historia. Puede ir de menor a mayor o viceversa.
Tiempo Estimado: Después de comparar cada historia, se asigna una puntuación llamada story point y se evalúa con todo el equipo involucrado (desarrolladores, tester, diseñadores, líder y clientes).
Ejecución de Prueba: Es una descripción de alto nivel se indicará la forma de ejecución del demo.
Notas: Referencias, información o aclaraciones.

Ejemplo de un Product Backlog (Ilustración I,) (The scrum institute)

ToDo List

ID	Story	Estimation	Priority
7	As an unauthorized User I want to create a new account	3	1
1	As an unauthorized User I want to login	1	2
10	As an authorized User I want to logout	1	3
9	Create script to purge database	1	4
2	As an authorized User I want to see the list of items so that I can select one	2	5
4	As an authorized User I want to add a new item so that it appears in the list	5	6
3	As an authorized User I want to delete the selected item	2	7
5	As an authorized User I want to edit the selected item	5	8
6	As an authorized User I want to set a reminder for a selected item so that I am reminded when item is due	8	9
8	As an administrator I want to see the list of accounts on login	2	10
Total		**30**	

Ilustración 1 Ejemplo de Product Backlog

Spring: Una vez que las tareas de usuario están definidas se deben de llevar a cabo y para ello utilizamos los Springs, estos pueden considerarse mini proyectos o tareas específicas las cuales juntas conforman el proyecto final, los Springs tienen 5 fases:

Planificación del Spring: En esta fase todo el equipo que conforma el scrum es reunido y en él se establecen los puntos a entregar, el tiempo, los involucrados los equipos y la asignación de tareas.
Scrum Diario: son pequeñas juntas de máximo 15 minutos donde el equipo de desarrollo sincroniza sus actividades y crea un plan para las próximas 24 horas.
Proceso del Spring: Tareas diarias en proceso que ya fueron definidas, en esta ase nos debemos de asegurar de no cambiar los requerimientos, no disminuir las tareas ni la calidad, no iniciar un proceso sin finalizar el anterior.
Revisión del Spring: Esta fase se lleva a cabo con el equipo SCRUM donde se muestra los avances concluidos al cliente y esta muestra su punto de vista, en esta fase aún se pueden realizar cambios ya que es más fácil modificar un producto cuando aún no está concluido.
Retrospectiva del Spring: En esta fase el equipo SCRUM se reúne para hacer comentarios internos cuyos objetivos son los de mejorar durante el siguiente Spring.

Un Spring puede adaptarse a diferentes tipos de organización, pero todos tienen un formato como el siguiente. (Ilustración 2) (Scrum.org)

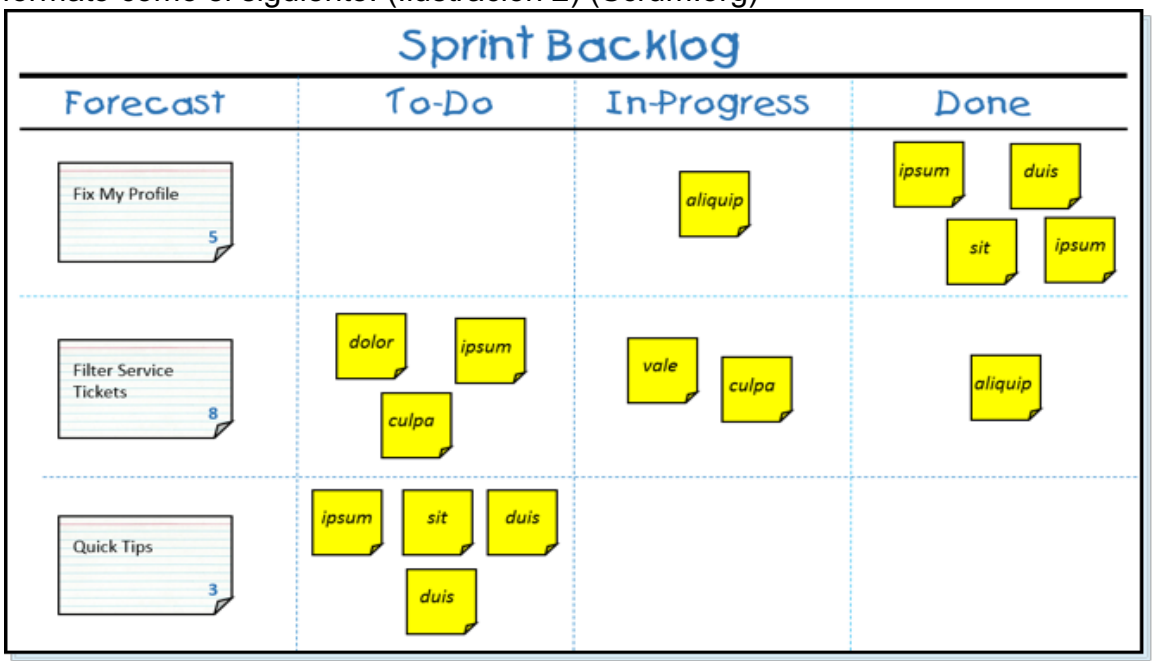


Ilustración 2 Ejemplo de un Spring

2.7 Modelado UML.

El Lenguaje de modelado Unificado (UML) es un lenguaje gráfico que nos ayuda a visualizar, construir y documentar un sistema. De igual manera el UML se puede traducir como un estándar internacional que representa y modela las fases de diseño y análisis.

Los objetivos del UML son bastantes y estos van de acuerdo a su aplicación ya que no está definido estrictamente para el desarrollo de software sin embargo podemos plasmar los generales que se adaptan a cualquier tipo de proyecto.

Visualizar: Expresar de forma gráfica un sistema y con ello se garantiza que otra persona lo entienda.

Especificar: Especifica las características de un sistema antes de construirlo.

Construir: A partir de los modelos especificados se puede construir el sistema.

Documentar: Los elementos UML nos sirven para documentar toda la información correspondiente de nuestro sistema.

Para este sistema vamos a utilizar el UML con diagramas de clases para mostrar de forma gráfica las clases de las que se compone el producto, esto nos ayudara bastante ya que al observar la base de datos en un diagrama UML nos ahorrara tiempo de construcción y validación además de que hacemos uso de buenas prácticas de programación y desarrollo.

Un diagrama de clases se compone de un identificador único, atributos y métodos (Ilustración 3), el atributo único debe tener una referencia al paquete de la aplicación para que en caso de modificarse se encuentre rápido, los atributos son las propiedades de la clase y toda clase debe contar con ellos y un método son las acciones que realiza esa clase ya sea con sus a tributos o con atributos de otras clases.

Ilustración 3 Ejemplo de diagrama de clases.

2.8 Algoritmo de Geolocalización.

Geolocalización se define como la capacidad para obtener la ubicación geográfica real de un objeto, con la ayuda de un medio electrónico como un radar o un teléfono móvil por medio del internet. Principalmente la Geolocalización se divide en dos modos:

Activo: Este modo se basa en el dispositivo del usuario y el método de recolección está basado completamente en el chip del GPS triangulando la información de las torres celulares, las tecnologías que se ven involucradas en este modo son las siguientes:

- **GPS**
- **WI-FI**
- **3G / 4G**

Actualmente las aplicaciones móviles, los dispositivos móviles y los dispositivos de rastreo utilizan esta tecnología.

Pasivo: Este modo involucra el uso de proveedores de servicios de geolocalización de terceros, es decir para obtener la ubicación del usuario no es necesario adquirirla del dispositivo del usuario, sino que se toma directamente de su ir, dns o búsqueda whois.

Para casos de usos prácticos utilizaremos el modo activo, obteniendo directamente la ubicación del usuario por medio de su dispositivo con la ayuda del GPS de alta precisión. Para ayudarnos a lograr nuestro objetivo nos apoyaremos en las tecnologías de Google mediante el uso de su API de Google Maps para obtener la ubicación del usuario, sin embargo, debemos de procesar esa información para ello seguiremos el siguiente algoritmo propuesto.

- Obtener la ubicación del usuario mediante la Api de Google Maps
- Almacenar dichas coordenadas en dos variables separadas
- Enviar una consulta a la base de datos con la ubicación para que el software determine mediante una operación matemática que convenios se encuentran cercanos a la ubicación del usuario.
- Validar en la aplicación los convenios y mostrarlos en pantalla con su respectiva información.

Capítulo III Análisis del sistema y Diseño del Sistema

3.1 Diagramas de Casos de Usos.

Los diagramas de casos de uso se definen como la descripción de las acciones que un sistema realiza desde la perspectiva del usuario, por otra parte, define también el comportamiento deseado de nuestro sistema sin la necesidad de redactar o especificar dicho comportamiento, para este sistema o aplicación contamos con el siguiente modelo de caso de uso. (Ilustración 4)

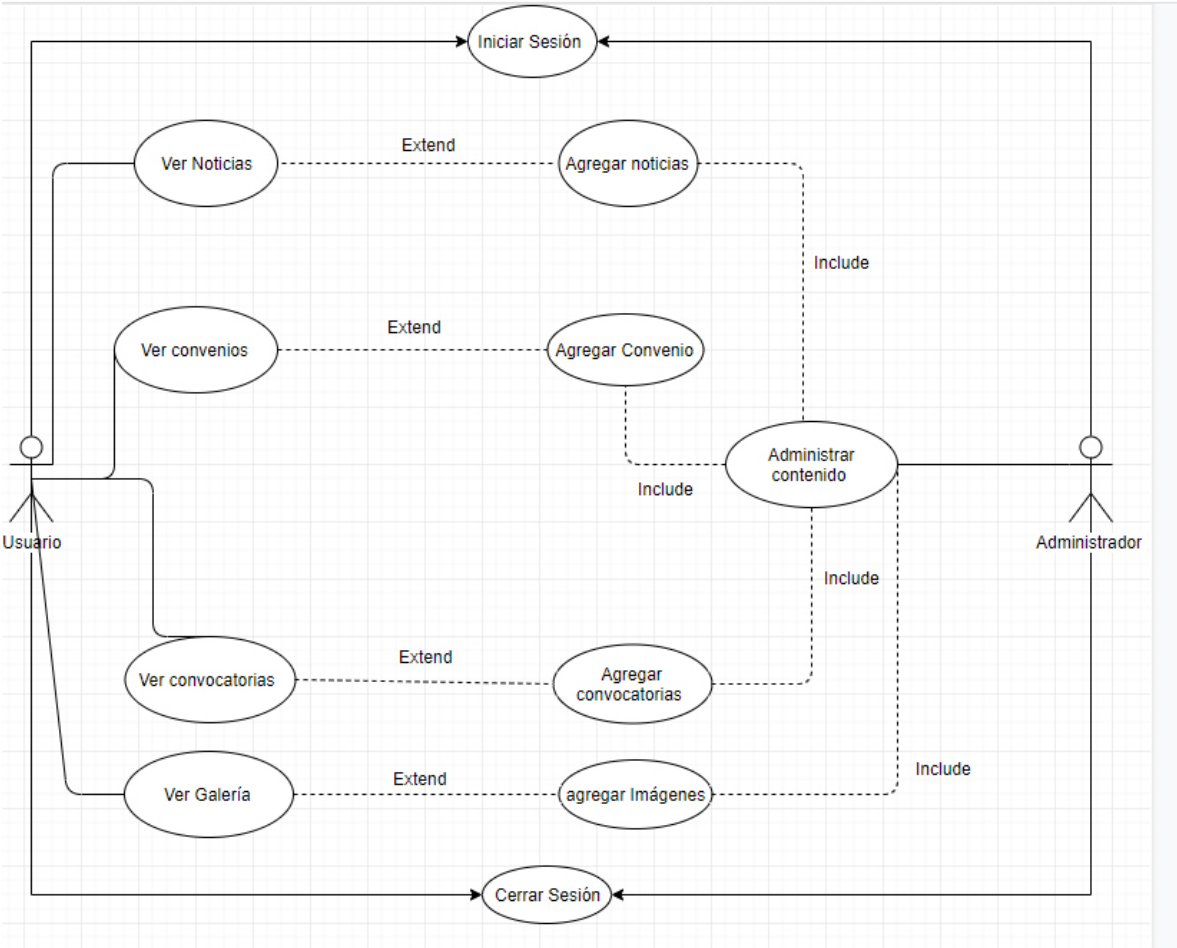
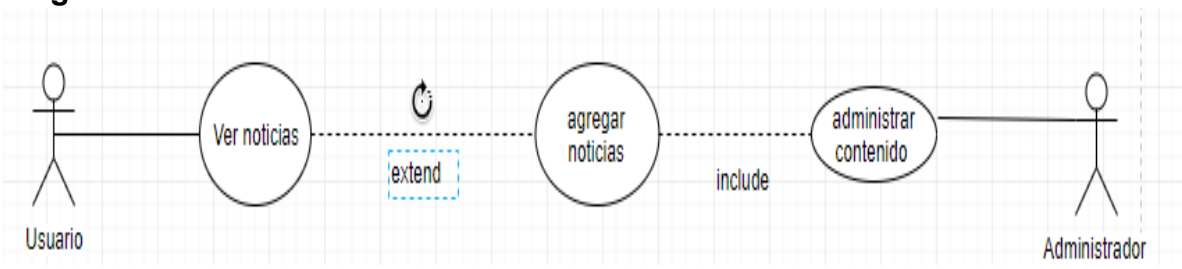


Ilustración 4 Modelo de casos de uso de aplicación.

3.2 Diagramas de Casos de Usos Específicos.

A continuación, describiremos cada caso de uso siguiendo la correcta ingeniería de software especificando a detalle cada elemento del mismo, cada caso de uso será explicado de manera individual con su respectivo diagrama.

a. **Caso de Uso:** Ver Noticias.
Diagrama:



Actor: Usuario y administrador.

Descripción: Un administrador se encarga de subir las noticias mediante una aplicación web y por medio de web servicies se muestra dentro de la aplicación todas las noticias listadas desde la más reciente hasta la más antigua. El usuario solo podrá observar dichas noticias.

Precondiciones:

1. El administrador debe estar “logeado” y debe contar con permisos suficientes para administrar el sistema.
2. El usuario debe ser miembro de la comunidad de docentes y del sindicato de trabajadores de la BUAP para poder ingresar al sistema.

Flujo de eventos normales:

Paso	Acción	Paso	Acción	Excepción
1	El administrador da clic en el apartado de noticias	2	El sistema muestra una serie de campos que el administrador debe llenar obligatoriamente	E1
3	El administrador observa que los datos estén completos y da clic en enviar	4	El sistema valida los datos y los almacena en la base de datos	E1
5	El usuario da clic en el apartado de noticias	6	El sistema realiza la consulta necesaria y	E1, E2

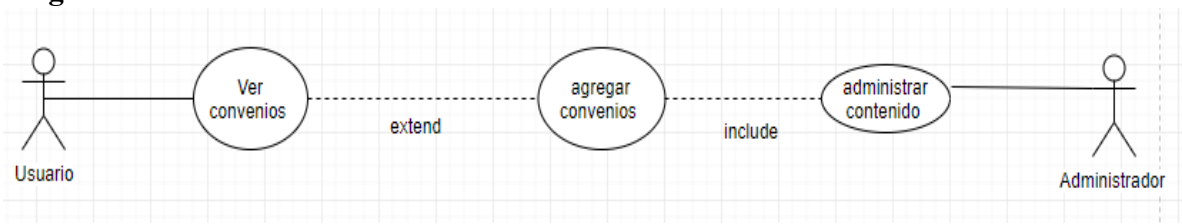
			muestra todas las noticias actuales	
7	El usuario selecciona la noticia que desea leer y da clic en ella	8	El sistema muestra todos los datos de la actual noticia al usuario	E1,E2

Flujo de eventos alternos:

ID	Nombre	Acción
E1	Error en la conexión a la base de datos	El sistema muestra un mensaje de error
E2	El sistema no cuenta con acceso a internet	El sistema muestra un mensaje de error y pide validar la conexión

b. **Caso de Uso:** Ver convenios.

Diagrama:



Actor: Usuario y Administrador.

Descripción: Un administrador se encarga de colocar el convenio junto con su dirección proporcionada por google maps, cuando el usuario accede a la sección de convenios el mismo puede visualizar dos “tabs” la primera mostrara el convenio por imagen y la segunda mostrara los convenios por posición según su ubicación.

Precondiciones:

1. El usuario debía aceptar el uso de GPS al instalar la aplicación
2. El administrador debe estar “logeado” y debe contar con permisos suficientes para administrar el sistema.
3. El usuario debe ser miembro de la comunidad de docentes y del sindicato de trabajadores de la BUAP para poder ingresar al sistema.

Flujo de datos normal:

Paso	Acción	Paso	Acción	Excepción
1	El administrador accede al panel y da clic en el apartado de dar de alta nuevo convenio	2	El administrador llena los campos solicitados como imagen, descripción, latitud y longitud.	E1
3	El administrador valida los datos y da clic en guardar	4	El sistema valida los datos y los almacena en la base de datos	E1
5	El usuario da clic en el apartado de convenios dentro del menú principal de la aplicación	6	El sistema muestra una pantalla con 2 tabs y por default muestra el tab 1 donde muestra un listado de los convenios con imagen y título del convenio	E1
7	El usuario puede seleccionar el convenio o cambiar de tab en caso de que seleccione un convenio este muestra la descripción del mismo	8	El usuario cambia de tab	E1,E2,E3
9	El sistema muestra la ubicación del usuario y coloca un marcador en la posición donde se encuentra y pide al usuario buscar convenios en la zona	10	El sistema muestra una serie de marcadores con los convenios que se encuentren a menos de 500 metros de la ubicación del usuario	E1, E2, E3
11	El usuario selecciona un marcador	12	El sistema despliega la información del convenio seleccionado y da la opción de mostrar la ruta más rápida a la ubicación del convenio	E1, E2, E3

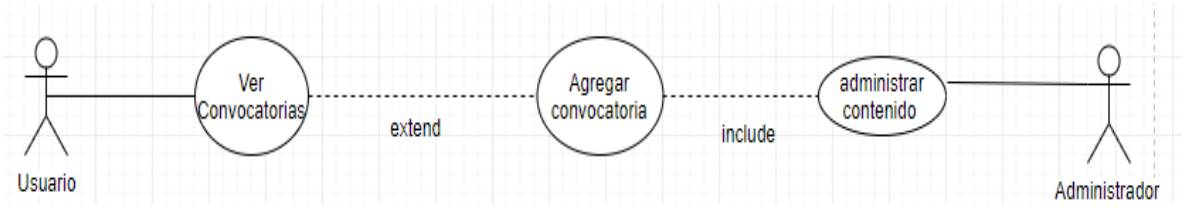
Flujo de datos alternativo:

ID	Nombre	Acción
E1	Error en la conexión a la base de datos	El sistema muestra un mensaje de error
E2	El sistema no cuenta con acceso a internet	El sistema muestra un mensaje de error y pide validar la conexión

E3	El GPS del dispositivo se encuentra apagado	El sistema pedirá al usuario que encienda el GPS y recargue la aplicación

c. **Caso de Uso:** Ver Convocatorias.

Diagrama:



Actor: Usuario y Administrador.

Descripción: Un administrador se encarga de dar de alta la convocatoria subiendo el PDF y el usuario puede descargarlo desde su dispositivo.

Precondiciones:

1. El administrador debe estar "logeado" y debe contar con permisos suficientes para administrar el sistema.
2. El usuario debe ser miembro de la comunidad de docentes y del sindicato de trabajadores de la BUAP para poder ingresar al sistema.

Flujo de datos normal:

Paso	Acción	Paso	Acción	Excepción
1	El administrador da clic en el apartado de convocatorias	2	El sistema muestra una serie de campos que el administrador debe llenar obligatoriamente	E1
3	El administrador observa que los datos estén completos y da clic en enviar	4	El sistema valida los datos y los almacena en la base de datos	E1
5	El usuario da clic en el apartado de convocatorias	6	El sistema realiza la consulta necesaria y muestra todas las convocatorias actuales	E1, E2

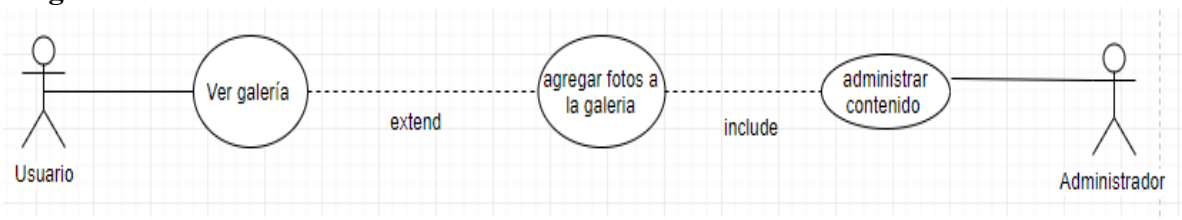
7	El usuario selecciona la convocatoria que desea leer y da clic en ella	8	El sistema descarga al dispositivo el PDF de la convocatoria seleccionada	E1, E2

Flujo de datos alternativo:

ID	Nombre	Acción
E1	Error en la conexión a la base de datos	El sistema muestra un mensaje de error
E2	El sistema no cuenta con acceso a internet	El sistema muestra un mensaje de error y pide validar la conexión

d. **Caso de Uso:** Ver Galería.

Diagrama:



Actor: Usuario y Administrador.

Descripción: El administrador sube una serie de fotos a la galería y el usuario puede acceder a ellas para su correcta visualización.

Precondiciones:

3. El administrador debe estar “logeado” y debe contar con permisos suficientes para administrar el sistema.
4. El usuario debe ser miembro de la comunidad de docentes y del sindicato de trabajadores de la BUAP para poder ingresar al sistema.

Flujo de datos normal:

Paso	Acción	Paso	Acción	Excepción
1	El administrador da clic en el apartado de Galería	2	El sistema muestra una serie de campos que el administrador debe llenar obligatoriamente	E1

3	El administrador observa que los datos estén completos y da clic en enviar	4	El sistema valida los datos y los almacena en la base de datos	E1
5	El usuario da clic en el apartado de Galería	6	El sistema realiza la consulta necesaria y muestra todas las fotos	E1, E2
7	El usuario puede visualizar la galería.	E1, E2		

Flujo de datos alternativo:

ID	Nombre	Acción
E1	Error en la conexión a la base de datos	El sistema muestra un mensaje de error
E2	El sistema no cuenta con acceso a internet	El sistema muestra un mensaje de error y pide validar la conexión

3.3 Diagrama de Clases del Sistema

El diagrama de clases de un sistema se utiliza para mostrar las clases estructuradas con sus atributos y sus operaciones y la relación entre ellas. en este caso existen 6 clases principales: administradores, convenios, convocatorias, noticias y usuarios, ubicaciones. El administrador tiene control sobre convenio, convocatoria y noticias mientras que el usuario solo puede ver los datos ya almacenados en la base de datos como se muestra en ilustración 5.

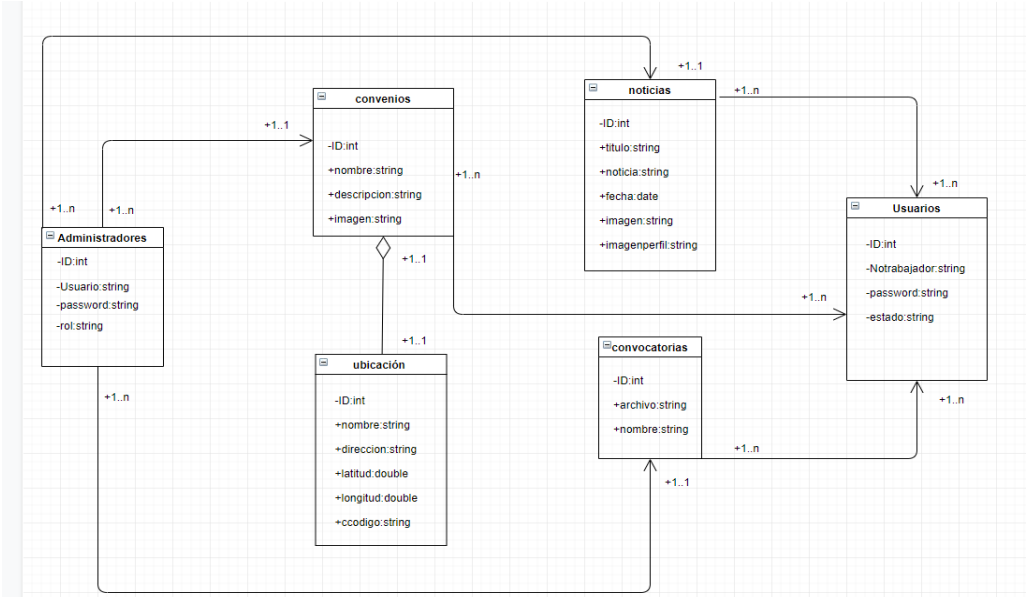


Ilustración 5 Diagrama de clases del sistema.

3.4 Diagramas de Secuencia

Los diagramas de flujo nos indican el proceso que sigue el sistema para completar una acción, su principal función es la documentar, planificar y mejorar los procesos que el sistema realiza. A continuación, se mostrarán los diagramas de flujo de los principales procesos.

Proceso 1: Inicio de sesión al sistema. Ilustración 6

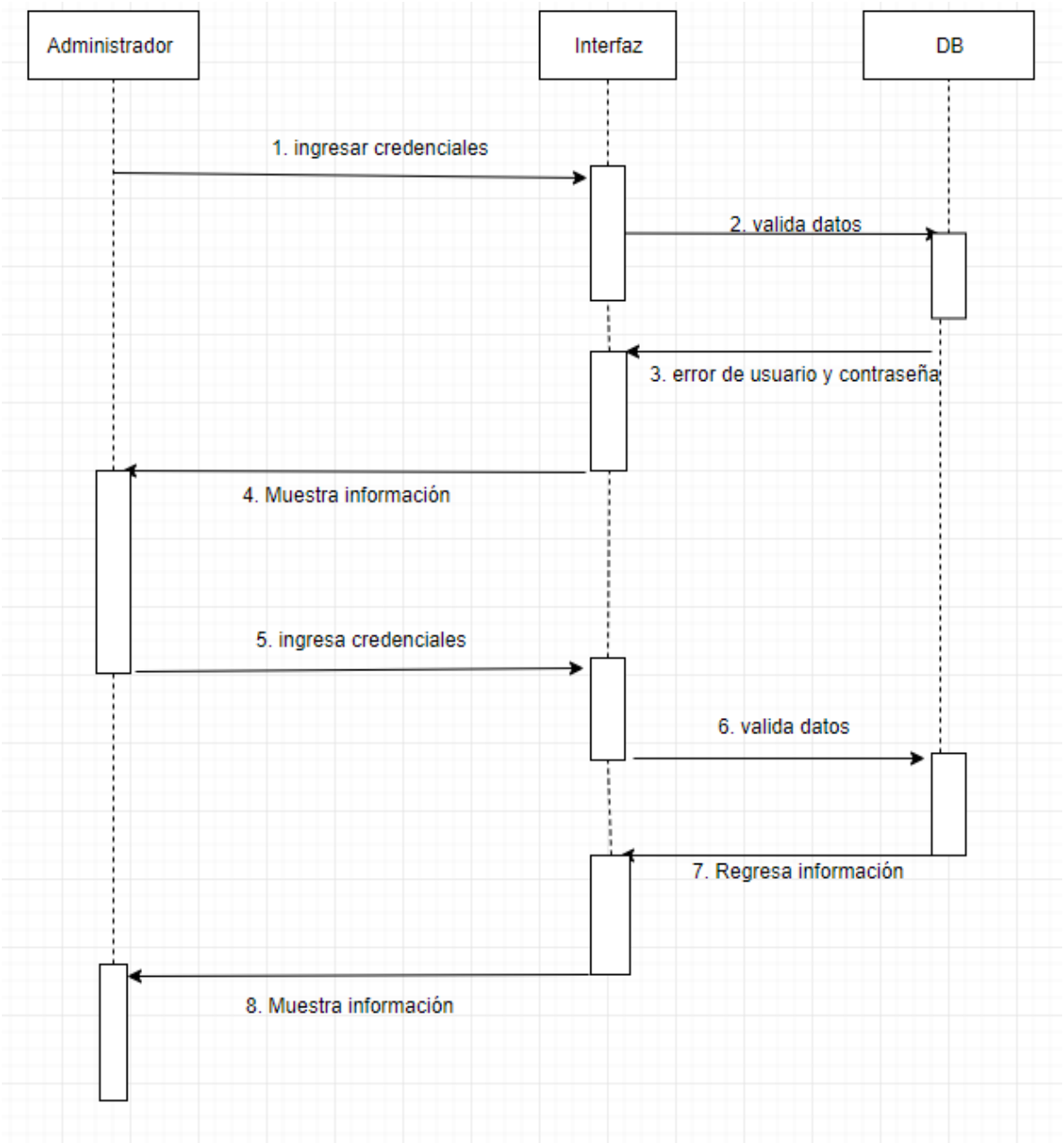


Ilustración 6 Diagrama de secuencia del proceso 1.

Descripción del flujo: el administrador ingresa sus credenciales al sistema este los valida en caso de ser erróneas retorna un error, el administrador vuelve a introducir las credenciales en caso de ser correctas se da acceso al sistema y se muestra la información.

Proceso 2: dar de alta una noticia. (Ilustración 7)

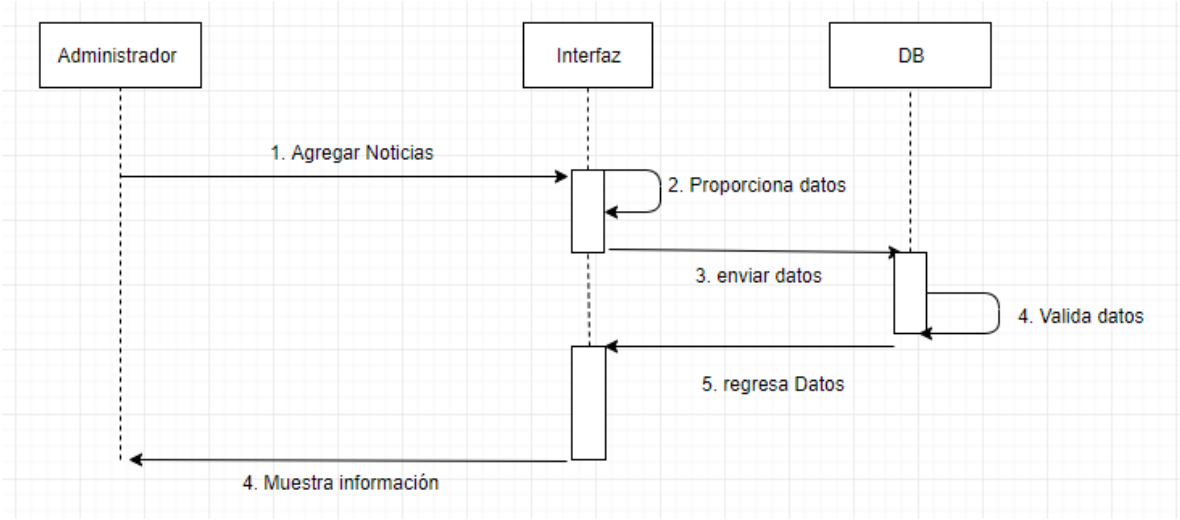


Ilustración 7 Diagrama de secuencia del proceso 2.

Descripción: El administrador ingresa al sistema y proporciona los datos establecidos estos son validados, enviados y almacenados.

Proceso 3: Agregar un convenio **(**Ilustración 8**)**

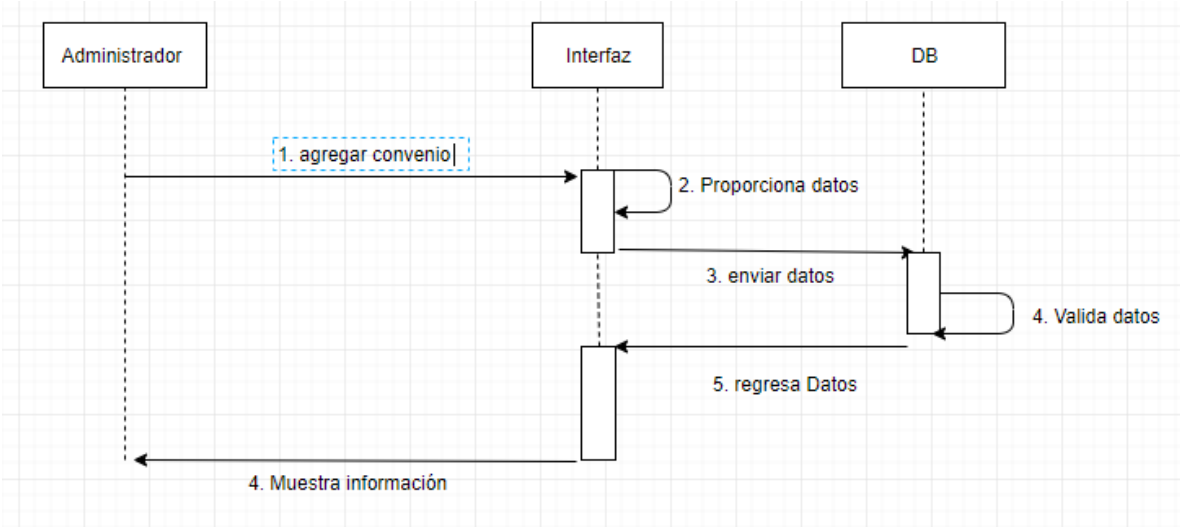


Ilustración 8 Diagrama de secuancia del proceso 3.

Descripción: El administrador ingresa al sistema, proporciona los datos establecidos y los manda para su validación si son aprobados serán almacenados para después ser mostrador por medio de web Services.

Proceso 4: Agregar una nueva convocatoria (Ilustración 9)

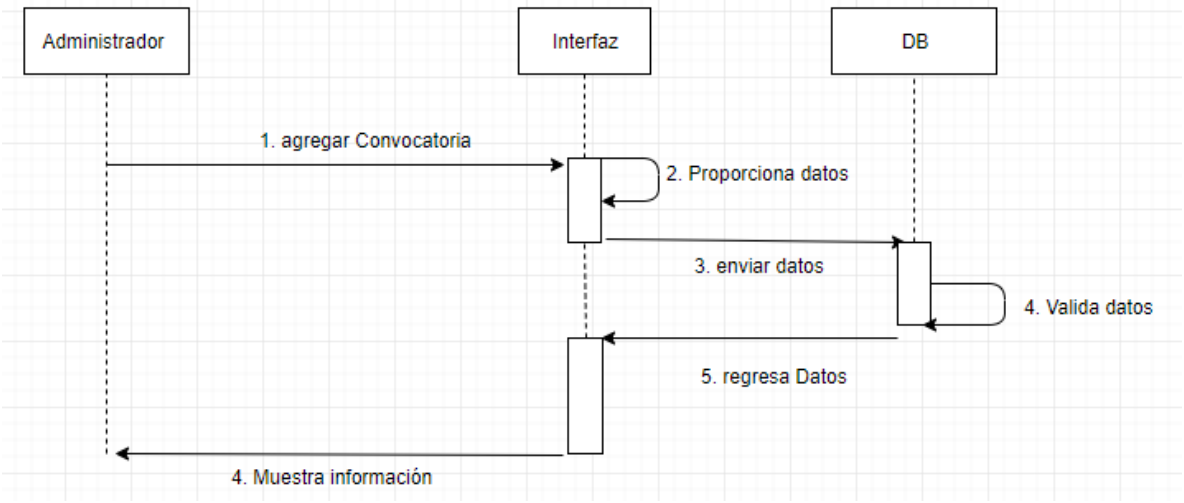


Ilustración 9 Diagrama de secuencia del proceso 4.

Descripción: El administrador ingresa al sistema, proporciona los datos establecidos y los manda para su validación si son aprobados serán almacenados para después ser descargados.

Proceso 5: Agregar galería de imágenes (Ilustración 10)

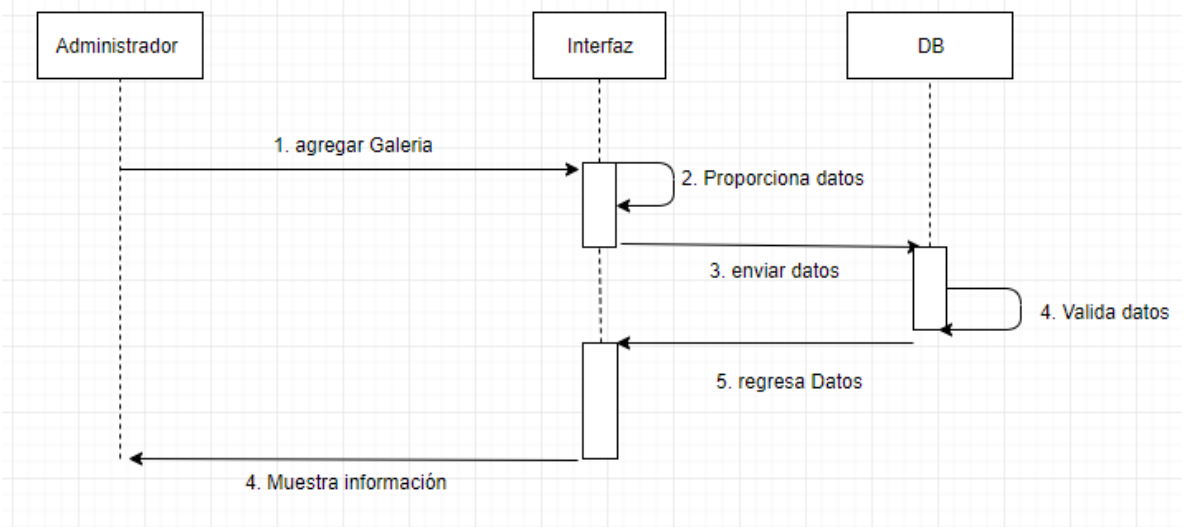


Ilustración 10 Diagrama de secuencia del proceso 5.

Descripción: El administrador ingresa al sistema, proporciona los datos establecidos y los manda para su validación si son aprobados serán almacenados para después ser visualizados.

3.5 Diagrama de la Base de Datos.

El diagrama de la base de datos nos ayuda a observar de forma gráfica la estructura de la base, además de permitirnos optimizar las consultas, agilizar el proceso de descarga y carga de información y sobre todo documentar de una forma precisa nuestra base de datos.

La base de datos creada consta de 6 tablas las cuales observaremos en la ilustración 11

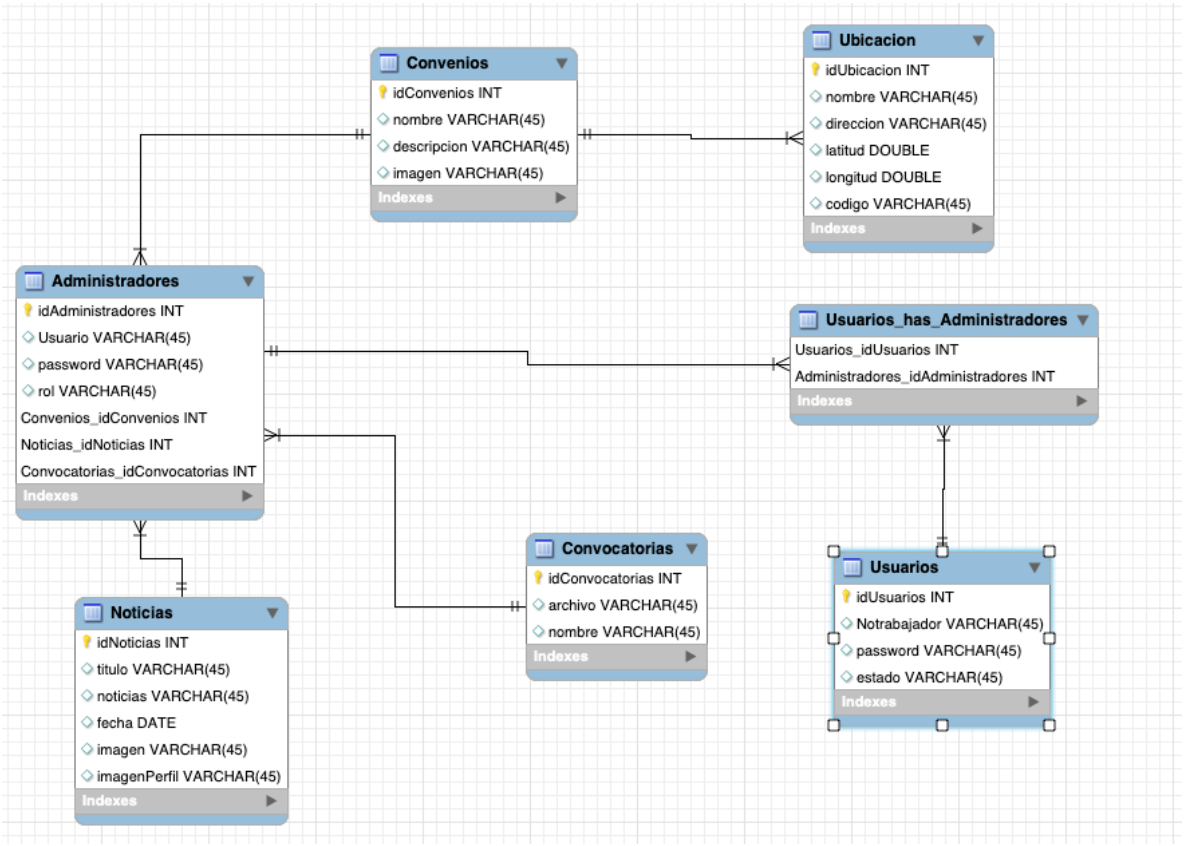


Ilustración 11 Diagrama de la base de datos.

3.6 Pilas de Productos.

Como se mencionó en el capítulo II en la sección 2.6, cuando se trabaja con Scrum es de suma importancia generar el Product Backlog del proyecto siendo este el núcleo de SCRUM, es aquí donde se comienza a desarrollar el proyecto, comenzando con la toma de requerimientos y su análisis. Como pequeña introducción, un product backlog se compone de 5 apartados muy importantes:

ID: Es un identificador único, auto-incremental.

Nombre: Es una descripción corta de la historia

Importancia: Es la ponderación del cliente a cierta historia y debe ser justificada por él, para que todo el equipo haga conciencia de la priorización de las tareas que conformarán la historia de usuario. Generalmente el rango de la importancia va de mayor a menor en una escala del 1 al 100, donde 100 es el valor máximo.

Estimación Inicial: Es la valoración inicial del equipo, donde se indicará cuanto trabajo es necesario para implementar la historia. Las unidades son "puntos de historia" y usualmente corresponde a "días-persona ideales". Es decir ¿Cuántas horas por día se llevaría una persona en desarrollar dicha historia, incluyendo todas las tareas necesarias para culminarla?

Pruebas: Una descripción de alto nivel de cómo se demostrará la historia al finalizar el sprint. Por ejemplo, se puede utilizar pruebas de caja negra, blanca, desarrollo basado en pruebas, etc.

Notas: Es una descripción para clarificar las fuentes de información y/o material adicional a utilizarse en el desarrollo de la historia y los componentes de cada una.

Spring 1 y 2					
ID	Nombre	Importancia	Estimado	Pruebas	Notas
1	Diseño e implementación de la base de datos	100	16 días	Se diseñará la base con base en los requerimientos establecidos, se utilizará el lenguaje SQL para realizar consultas de prueba. El sistema contara con un super administrador.	El sistema contara con la interfaz CRUD del framwork Codeignaiter pero con la modificación de API Rest para un manejo de Web Services contando así con dos

					métodos de entrada y salida.
2	Diseño de la interfaz gráfica con apoyo de software de diseño.	95	3 días	Se hará uso del software Photoshop para diseñar la aplicación y así generar y centrar la idea principal del flujo de pantallas, colores, tipografía, tamaños. Etc.	El diseño es creado específicamente bajo los lineamientos de las aplicaciones web, contando con una retícula adaptada al tamaño estándar de los dispositivos móviles.
3	Implementación del diseño a la aplicación web (creación de la interfaz en HTML con CSS y JS)	90	16 días	Creación de la interfaz contemplando el css, js y html, en este spring, se hará uso de angular, ionic, codeiganter y dejaremos preparado el back-end para que sea incorporado en la siguiente etapa.	Solo se creará la estructura vacía ya que aún no podrá cargar ni visualizar ningún dato.
4	Optimización de retícula y pruebas de rendimiento en dispositivos móviles.	85	2d	Se realizará las pruebas correspondientes para observar sí la interfaz cumple con los estándares marcados anteriormente además de visualizar el rendimiento del dispositivo,	Se cargarán datos ficticios para observar la trasferencia de datos y la optimización de recursos en el dispositivo.

Spring 3 y 4					
ID	Nombre	Importancia	Estimado	Pruebas	Notas
5	Integración del back-end en los archivos creados previamente.	80	16 días	Se generarán los webservices, los métodos y las llamadas a la base, enseguida se integrará	Con la ayuda del software postman se realizarán pruebas de

				en el código final para comenzar con la estructura final del proyecto.	conexión a la base para probar las consultas y los webservices sin la necesidad de correo el proyecto en un servidor.
6	Pruebas de trasferencia masiva (altas, bajas y modificaciones)	75	8 días	Se someterá el software a la carga continua de datos y transferencia de los mismo, agregando, eliminando y modificando datos directos a la base para medir el rendimiento y la carga generada.	Para estas pruebas ya se encuentra el proyecto montado en un servidor y los webservices ya se encuentran funcionando en vivo, es requerido medir la taza de trasferencia para evitar posibles errores de consumo masivo de datos
7	Pruebas finales del sistema aun en un ambiente controlado.	70	16 días	Pruebas de ruptura en un ambiente controlado, principalmente se busca romper el sistema colocando datos inválidos para observar la reacción del sistema y las validaciones que realiza el mismo.	Cada error será documentado para tener una bitácora de fallos ya que en caso que se repita estar al tanto de que pudo generarlo

8	Pruebas en un ambiente real con el cliente	65	1d	El usuario final navegará, realizará consultas y peticiones, el objetivo es ver la accesibilidad y la facilidad de uso de la app.	Se tomará nota sobre los posibles cambios que el cliente pueda pedir sin llegar a realizar cambios mayores.

Spring 5					
ID	Nombre	Importancia	Estimado	Pruebas	Notas
9	Últimas modificaciones a la interfaz	60	2 días	Últimos cambios a la interfaz de usuario agregando las notas finales del usuario	Sin notas.

3.7 Conclusiones del Sistema.

El sistema creado depende de una plataforma web creada bajo el framework codeignaiter con el motor de desarrollo PHP y una aplicación móvil creada bajo el framework Ionic con el motor de desarrollo angular, en este caso la aplicación depende totalmente de la plataforma ya que es la que provee los servicios web, los métodos, las llamadas al sistema, las consultas y peticiones, sobre todo, la que administra todo el contenido. La aplicación web solo se encarga de tomar la información ya procesada y mostrarla en la interfaz previamente diseñada. Un punto clave de la aplicación es el algoritmo de Geolocalización el cual se encarga de determinar la ubicación del usuario, realizar una operación y determinar qué convenio se encuentra más cercano a la ubicación del usuario. El punto fuerte de la aplicación es la parte en que para el usuario determinar el convenio más cercano con el simple hecho de encender su GPS cumple la función de toda aplicación la cual sería facilitar el día a día del usuario.

Capítulo IV Puesta en marcha del Sistema.

4.1 Pantallas del sistema.

Pantalla 1

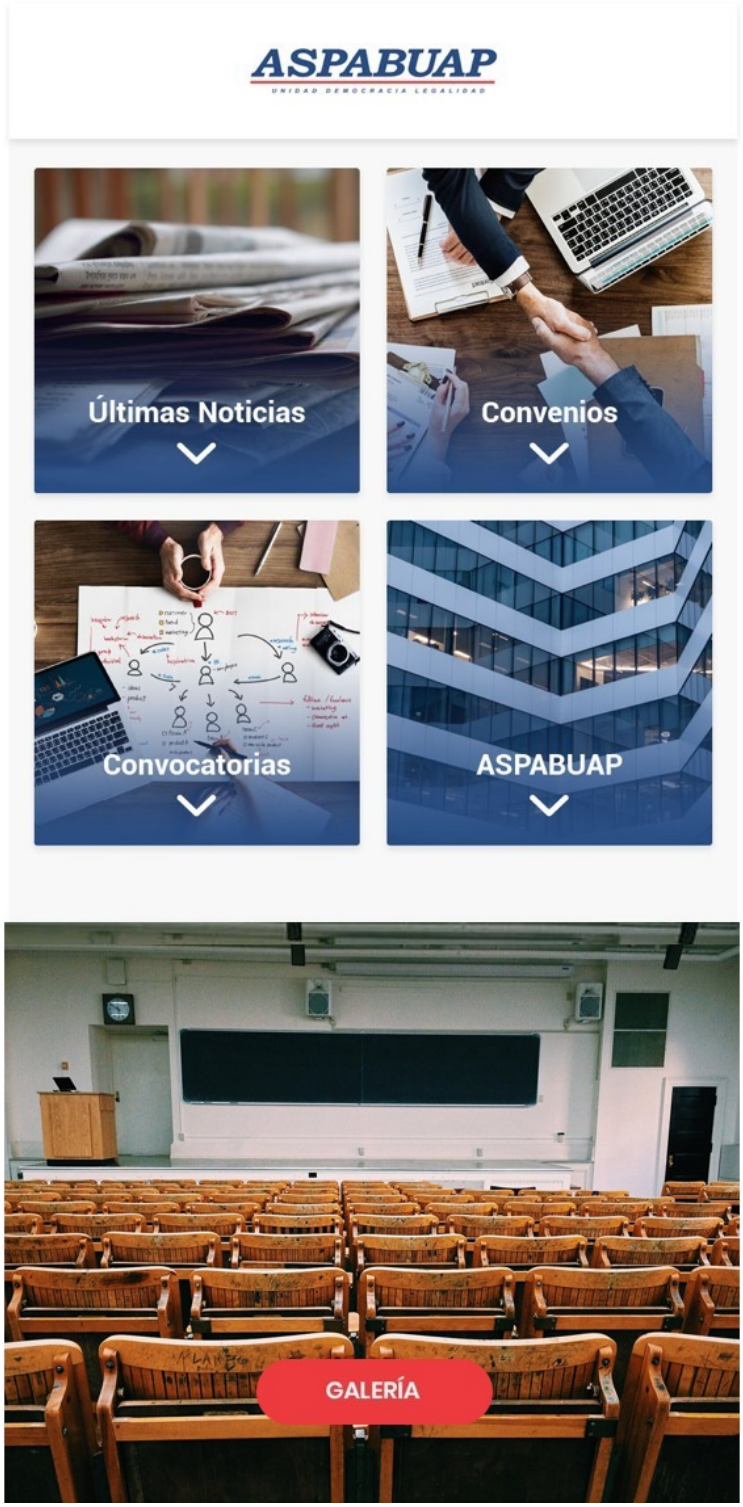
Descripcion de la pantalla: “Splash Screen” Esta pantalla muestra el logo de ASPABUAP en el Front-End y en el Back-End espera la respuesta del servidor y del local storage para validar la sesión y tener los datos cargador previamente.

Pantalla 2

Descripción de la pantalla: Ingreso de credenciales al sistema en caso de que se instale la aplicación por primera vez, en esta vista se ingresa la contraseña y el usuario los cuales ya estan dados de alta como trabajadores de la BUAP.

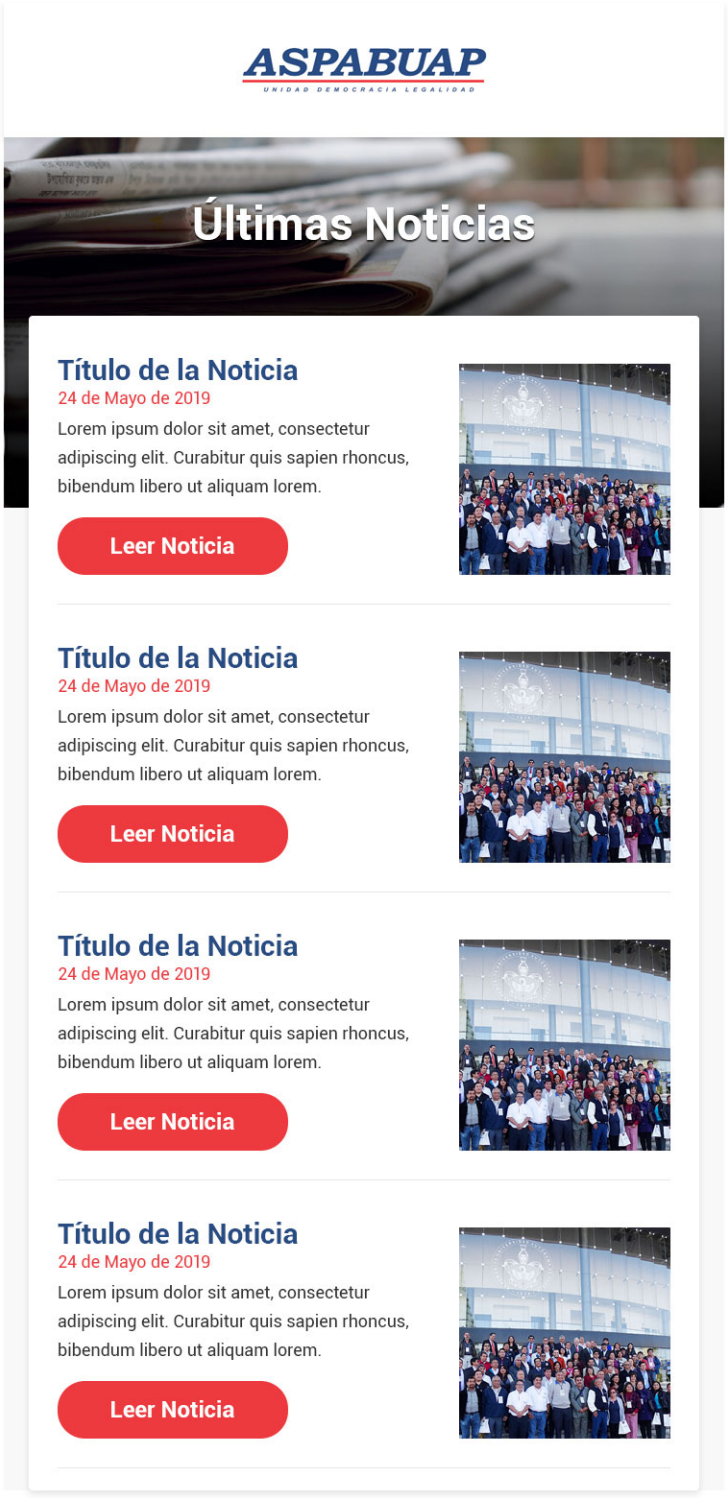
Pantalla 3

Descripción de la pantalla: Menu principal de la aplicación, todos los accesos se encuetran en este menu el cual tiene una estructura amigable y facil de utlizar para el usuario.



Pantalla 4

Descripción de la pantalla: Vista de Últimas Noticias, es esta vista se muestran las noticias mas recientes el adminitrador va ingresando en el sistema.

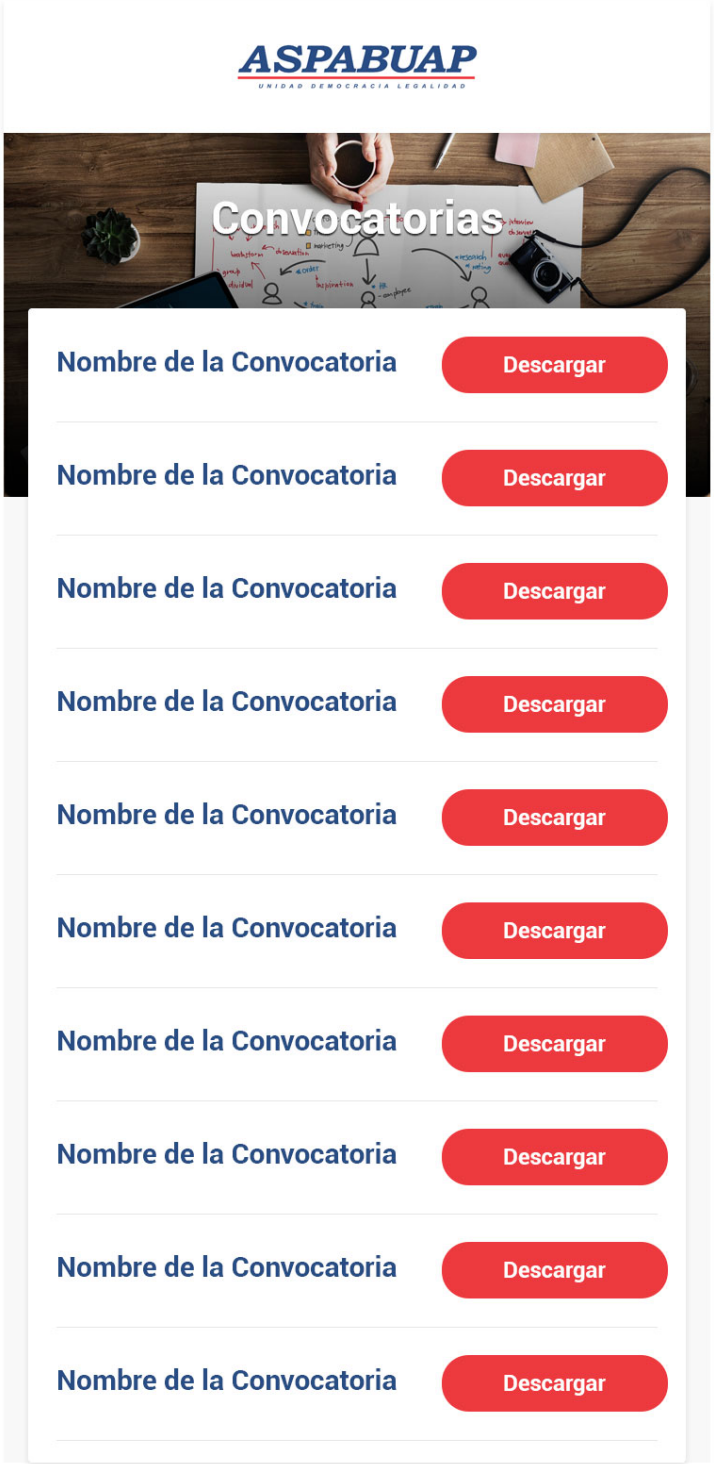


Pantalla 5

Descripción de la pantalla:Vista “Single de Noticias” una vez que el usuario selecciono la noticia a leer esta es mostrada en una nueva vista con sus respectivos datos.

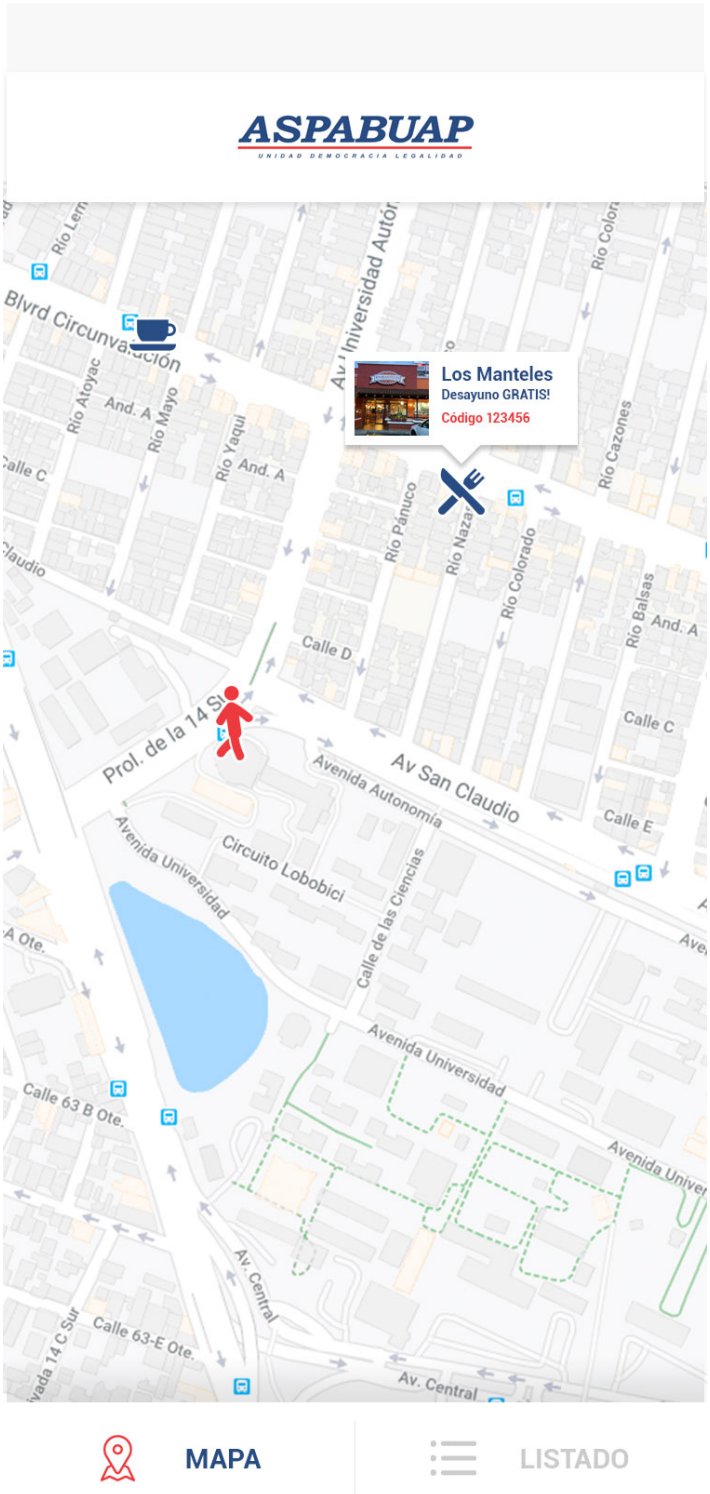
Pantalla 6

Descripción de la pantalla:Vista “Convocatorias” Se muestra un listado con los archivos descargables de todas las convocatorias vigentes.



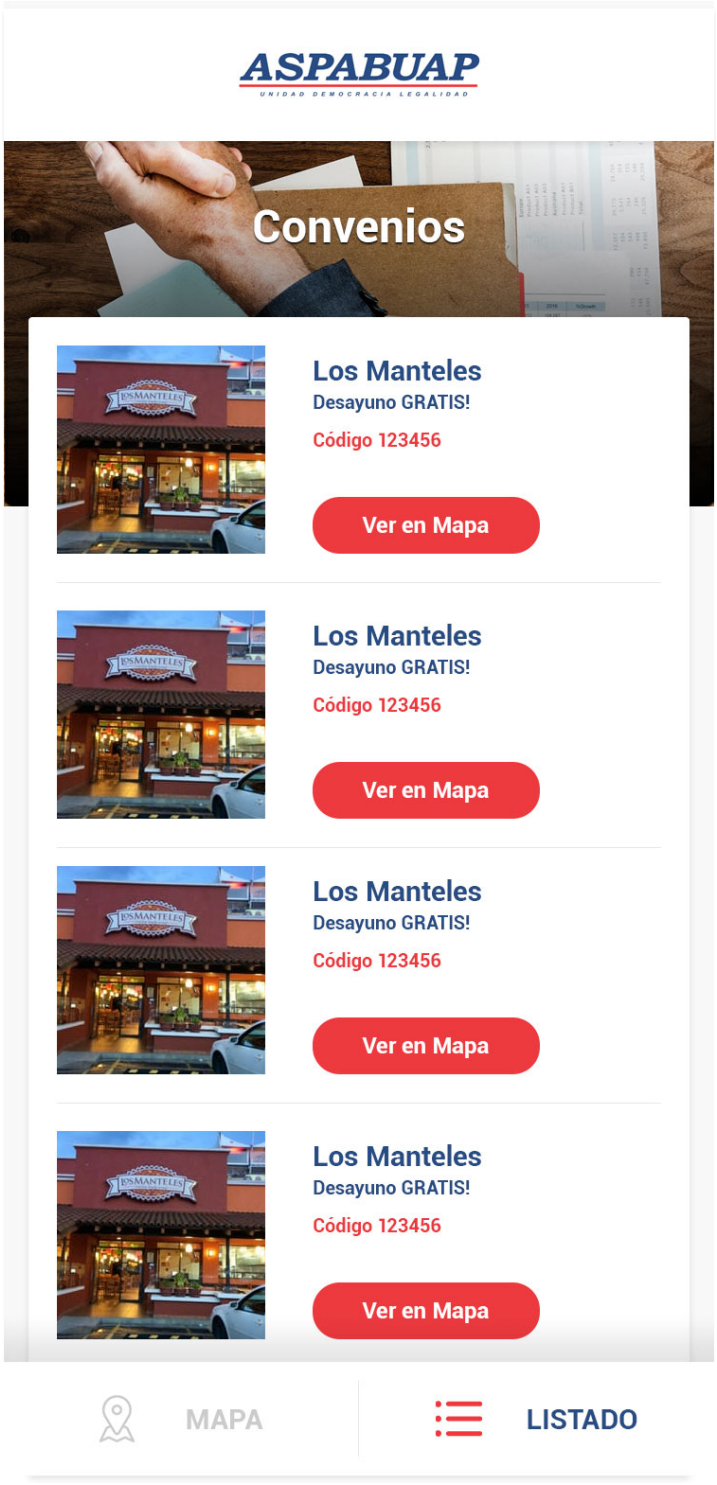
Pantalla 7

Descripción de la pantalla:Vista “Convenios Mapas” Esta pantalla muestra según la ubicación del usuario los convenios cercanos a el.



Pantalla 8

Descripción de la pantalla:Vista “Convenios Mapas Listado” Esta pantalla muestra una lista de todos convenios con los que el usuario cuenta sin importar sus ubicación.



Pantalla 9

Descripción de la pantalla:Vista “ASPABUAP” Esta pantalla muestra la información e historia del sindicato.

4.2 Metodos principales del sistema.

En este apartado colocare los codigos a los principales metodos del sistema como lo son: login, últimas noticias, mapa y convenios. Por otra parte colocare los componentes que se requieren para realizar las conexiones a los webservices.

1. url.servicios

   Tipo de archivo: “Servicios”.
   Objetivo del archivo: Almacenar las URL de origen de los webservices para posteriormente hacer el llamado a los mismos.

   Código: export const URL_SERVICIOS ="http://www.aspabuap.com/webservice/index.php/"

   Nota: esta linea solo exporta la ruta de los webservices.

2. Login

   Tipo de archivo: “Provider”
   Objetivo del archivo: validar si el usuario ya ha ingresado al sistema de no ser así pedir que se identifique, validar el usuario y la contraseña, guardar en la memoria del dispositivo el usuario y la contraseña para futuros inicios de sesión.

   Código:

```
activo():boolean{
    if (this.estado) {
        return true;
    }else{
      return false;
    }
  }

  logear(trabajador:string,password:string){
        let data = new URLSearchParams();
        data.append("trabajador",trabajador);
        data.append("password",password);
        let url = URL_SERVICIOS + 'login';
        return this.http.post(url,data).map(resp =>{
          let data_resp = resp.json();
          console.log(data_resp);
          if (data_resp.error) {
              this.alertCtrl.create({
                title:"Error al iniciar",
                subTitle:data_resp.mensaje,
```

```
                buttons:["ok"]
              }).present();
          }else{
            this.estado = data_resp.token;
            this.id_usuario = data_resp.id;
            console.log(this.estado);

            //guardar storage
            this.guardar_storage();

          }
        });
      }

      cerrar_sesion(){
        this.estado=null;
        this.id_usuario=null;
        this.guardar_storage();
      }

      private guardar_storage(){
        if (this.platform.is("cordova")) {
            this.storage.set('estado',this.estado);
            this.storage.set('id_usuario',this.id_usuario);
        }else{
          if (this.estado) {
          localStorage.setItem("estado",this.estado);
          localStorage.setItem("id_usuario",this.id_usuario);
        }else{
            localStorage.removeItem("estado");
            localStorage.removeItem("id_usuario");

         }
        }

      }

      cargar_storage(){
        let promesa = new Promise((resolve,reject)=>{
          if (this.platform.is("cordova")) {
            this.storage.ready().then(()=>{
              this.storage.get("estado").then(estado =>{
                if (estado) {
                    this.estado = estado;
                }
                resolve();
              })

              this.storage.get("id_usuario").then(id_usuario =>{
                if (id_usuario) {
                    this.id_usuario = id_usuario;
                }
                resolve();
              })

            })

```

```
78.             }else{
79.               if (localStorage.getItem("estado")) {
80.                   this.estado = localStorage.getItem("estado");
81.                   this.id_usuario = localStorage.getItem("id_usuari
   o");
82.               }
83.               resolve();
84.             }
85.           });
86.           return promesa;
87.         }
88.
89.       }
```

3. Cargar los convenios

   Tipo de archivo: “Provider”
   Objetivo del archivo: Mostrar el listado de todos los convenios disponibles.

   Codigo:

```
1. cargar_todos(){
2.     let promesa = new Promise((resolve,reject)=>{
3.       let url = URL_SERVICIOS + "convenios/todos/" + this.pagina;
4.       this.http.get(url).map(resp =>resp.json()).subscribe(data =>{
5.         console.log(data);
6.         if (data.error) {
7.
8.         }else{
9.         this.convenios.push(...data.empresas);
10.             this.pagina+=1;
11.           }
12.             resolve();
13.           })
14.
15.         });
16.         return promesa;
17.
18.       }
19.
20.     }
```

4. Cargar las ubicaciones

   Tipo de archivo: “Provider”
   Objetivo: precargar las ubicaciones de los convenios para despues implementar un metodo de selección

   Código:

```
1. cargar_todos(){
2.   let promesa = new Promise((resolve,reject)=>{
3.     let
   url = "http://localhost:8888/webservice/index.php/ubicaciones/todos
   ";
4.     this.http.get(url).map(resp =>resp.json()).subscribe(data =>{
5.       console.log(data);
6.       if (data.error) {
7.       }else{
8.       this.ubicaciones.push(...data.ubicaciones);
9.     }
10.            resolve();
11.          })
12.
13.        });
14.        return promesa;
15.
16.        }
17.      }
```

5. Cargar las noticias

   Tipo de archivo: “Provider”
   Objetivo: Mostrar todas las noticias que se encuentren en el sistema.

   Código:

```
1. cargar_todos(){
2.     let promesa = new Promise((resolve,reject)=>{
3.       let url = URL_SERVICIOS + "noticias/todos/" + this.pagina;
4.       this.http.get(url).map(resp =>resp.json()).subscribe(data =>{
5.         console.log(data);
6.         if (data.error) {
7.         }else{
8.         this.noticias.push(...data.noticias);
9.         this.pagina+=1;
10.            }
11.              resolve();
12.            })
13.
14.          });
15.          return promesa;
16.
17.        }
18.
19.
20.      }
```

6. Posición del usuario.

   Tipo de Archivo: “Page”
   Objetivo: Tomar y almacenar la ubicación del usuario en tiempo real.

   Código:

```
1. constructor(public navCtrl: NavController, public
   navParams: NavParams, private viewCtrl:ViewController,
2.     public geolocation: Geolocation,private
   googleMaps: GoogleMaps,private _ps:UbicacionProvider) {
3.
4.
5.       this.geolocation.getCurrentPosition().then((resp) => {
6.      console.log(resp.coords.latitude);
7.      }).catch((error) => {
8.       console.log('Error getting location', error);
9.     });
10.
11.          let watch = this.geolocation.watchPosition();
12.          watch.subscribe((data) => {
13.            this.milatitud=data.coords.latitude;
14.            this.milongitud=data.coords.longitude;
15.          });
16.        }
```

7. Mostrar el convenio que se encuentre a cierta distancia del usuario.

   Tipo de Archivo: “Page”
   Objetivo: Con base a la ubicación del usuario mostrar los convenios que se encuentren a una distancia establecida del usuario.

   Código:

```
1. this.map = GoogleMaps.create('map_canvas');
2.   this.map.one(GoogleMapsEvent.MAP_READY)
3.     .then(() => {
4.       this.getPosition();
5.
6.     })
7.     .catch(error =>{
8.       console.log(error);
9.     });
10.        }
11.
12.      getPosition():void{
13.
14.
15.        this.map.getMyLocation()
16.        .then(response => {
17.          this.map.moveCamera({
18.            target: response.latLng,
```

```
            zoom:15,
          });
          console.log(response);
          let marker: Marker = this.map.addMarkerSync({
          title: 'MiUbicacion',
          icon: 'red',
          animation: 'DROP',
          position: response.latLng
           });

        })
        .catch(error =>{
          console.log(error);
        });
      }




         mostrarU(){


          for (let i = 0; i < this._ps.ubicaciones.length; i++) {

              this.dLat = rad( this._ps.ubicaciones[i].latitud -
   this.milatitud );
              this.dLong = rad( this._ps.ubicaciones[i].longitud -
   this.milongitud );

              this.a = Math.sin(this.dLat/2) * Math.sin(this.dLat/2)
    + Math.cos(rad(this.milatitud)) * Math.cos(rad(this._ps.ubicacione
   s[i].latitud)) * Math.sin(this.dLong/2) * Math.sin(this.dLong/2);
              this.c = 2 * Math.atan2(Math.sqrt(this.a),
   Math.sqrt(1-this.a));
              this.d  = this.R * this.c;
              console.log(this.d);

              if (this.d <= 1000) {
                let marker: Marker = this.map.addMarkerSync({
                     title: this._ps.ubicaciones[i].nombre,
                     icon: 'blue',
                     animation: 'DROP',
                     position: {
                       lat: this._ps.ubicaciones[i].latitud,
                       lng: this._ps.ubicaciones[i].longitud
                     }
                   });
                }
              }



            function rad(x) {
              return x * Math.PI / 180;
            };
```

```
71.
72.             }
```

Bibliografía

Canavilhas, J. (2015). Nuevos Medios, Nuevo Ecosistema. El Profesional de La Información, 24(4), 357–362. https://www.proxydgb.buap.mx:2069/10.3145/epi.2015.jul.01

Durán Pérez, O., & Carrazana Castro, E. (2016). La preservación a largo plazo de la información digital. Revista Cubana de Ciencias Informáticas, 10(2), 6–8. Retrieved from https://www.proxydgb.buap.mx:3362/login.aspx?direct=true&db=fua&AN=115240854&lang=es&site=ehost-live

Kniberg, H. (2007). SCRUM Y XP DESDE LAS. infoq.

Schwab, K. (2016). La cuarta revolucion industrial. suiza: World Economic Forum.

INEGI (2018). Comunicado de prensa núm 208/18 Comunicación social.

https://www.consumidor.ftc.gov/articulos/s0018-aplicaciones-moviles-que-son-y-como-funcionan

INEGI. (2019). COMUNICADO DE PRENSA NÚM. 179/19. 2 ABRIL DE 2019, de INEGI Sitio web: https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2019/OtrTemEcon/ENDUTIH_2018.pdf

Gobiernos Locales, G. O. B. GOB. (s.f.). Estándar de aplicaciones móviles. Recuperado 19 mayo, 2019, de https://www.gob.mx/gobiernoslocales/documentos/estandar-de-aplicaciones-moviles?idiom=es

IBM. (s.f.). IBM Knowledge Center. Recuperado 19 mayo, 2019, de https://www.ibm.com/support/knowledgecenter/es/SS8H2S/com.ibm.mc.doc/dev_source/references/dev_about_app_types.htm

Scrum-institute. (s.f.). Scrum. Recuperado 19 mayo, 2019, de https://www.scrum-institute.org/The_Scrum_Product_Backlog.php

G. Booch, J. Rumbaugh y I. Jacobson, "El Lenguaje Unificado de Modelado"
Addison Wesley, 1999 2. I. Jacobson, G. Booch, J. Rumbaugh , "El Proceso Unificado de Desarrollo

Haversine Formula (from R.W. Sinnott, "Virtues of the Haversine",
Sky and Telescope, vol. 68, no. 2, 1984, p. 159):