



UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

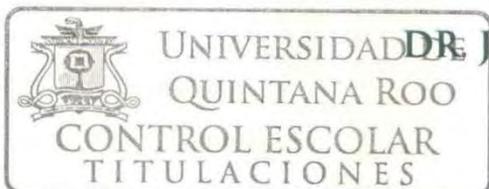
DESARROLLO DE UNA APLICACIÓN MÓVIL (NETATTEND) PARA LA GESTIÓN DE ASISTENCIA A EVENTOS ACADÉMICOS

TESIS
PARA OBTENER EL GRADO DE
INGENIERO EN REDES

PRESENTA
JAIR ALDO PASOS

DIRECTOR DE TESIS
MTI. VLADIMIR VENIAMIN CABAÑAS VICTORIA

ASESORES
M.T.I. MELISSA BLANQUETO ESTRADA
M.S.I. LAURA YÉSICA DÁVALOS CASTILLA
DR. JAVIER VÁZQUEZ CASTILLO
DR. JAIME S. ORTEGÓN AGUILAR





UNIVERSIDAD DE QUINTANA ROO

DIVISIÓN DE CIENCIAS E INGENIERÍA

TRABAJO DE TESIS TITULADO

“DESARROLLO DE UNA APLICACIÓN MÓVIL (NETATTEND) PARA LA GESTIÓN DE ASISTENCIA A EVENTOS ACADÉMICOS”

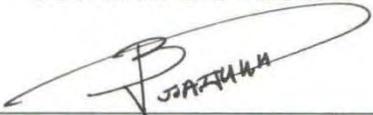
ELABORADO POR

JAIR ALDO PASOS

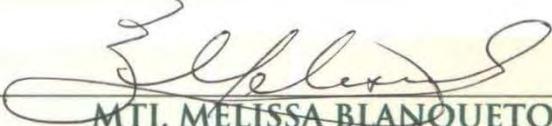
BAJO SUPERVISIÓN DEL COMITÉ DE ASESORÍA Y APROBADO COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE INGENIERO EN REDES

COMITÉ DE TESIS

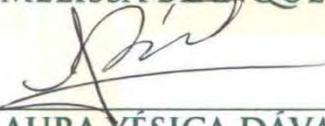
DIRECTOR:


M.T.I. VLADIMIR VENIAMIN CABAÑAS VICTORIA

ASESORA:


M.T.I. MELISSA BLANQUETO ESTRADA

ASESORA:

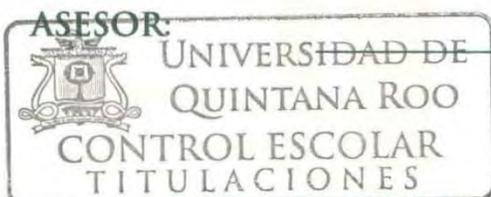

M.S.I. LAURA YÉSICA DÁVALOS CASTILLA

ASESOR:


DR. JAVIER VÁZQUEZ CASTILLO

ASESOR:


DR. JAIME SILVERIO ORTÉGÓN AGUILAR



Resumen

El presente proyecto describe la problemática en el control de asistencia a eventos académicos dentro y fuera de las instalaciones de la Universidad de Quintana Roo. Contiene el planteamiento del problema y el desarrollo de una aplicación móvil basada en la generación y uso de códigos QR (que se han vuelto muy populares) en una plataforma Android

NetAttend (aplicación móvil para el registro de asistencia a eventos) es el resultado del desarrollo de la aplicación móvil, la cual fue implementada y utilizada principalmente por la División de Ciencias Políticas y Humanidades (DCPH) del campus Chetumal de la Universidad de Quintana Roo para un evento que se ha venido realizando de manera periódica durante los últimos 14 años. Dicho evento es conocido actualmente como **Foro de Estudios en Lenguas Internacional (FEL)**.

El FEL es un evento que goza de un alto prestigio entre la comunidad educativa y de investigación orientada al estudio y enseñanza de diversas lenguas, dentro de este macro evento hay cabida para ponencias en diferentes idiomas (español e inglés principalmente), talleres, reuniones temáticas, mesas de trabajo, reunión de redes de colaboración, asesorías y simposios.

El comité organizador del FEL deseaba un mecanismo que le permitiera llevar un mejor control de la asistencia a los eventos descritos anteriormente, para la edición número 14 y 15 (años 2018-2019) de este importante evento utilizaron *NetAttend* con un alto nivel de aceptación y satisfacción tanto por parte del comité como de los usuarios (asistentes al evento) al usar la tecnología móvil, lo cual permitió desarrollar las actividades de registro de asistencia de una manera cómoda y rápida.

Agradecimientos

Durante la redacción de este documento de tesis, he recibido una gran cantidad de apoyo y asistencia. Primero quisiera agradecer a mi director, el M.T.I. Vladimir Veniamin Cabañas Victoria, cuya experiencia fue invaluable en la formulación de los temas y la metodología en particular.

Me gustaría agradecer a los organizadores del Foro de Estudios en Lenguas Internacional (FEL) por considerarme y darme la oportunidad de trabajar con ellos en el proyecto.

Me gustaría agradecer a mis colegas de Cuarta Revolución por su colaboración. Me apoyaron mucho y siempre estuvieron dispuestos a ayudarme. En particular, me gustaría destacar a mi supervisor, Ernesto de la Cruz, su ayuda fue realmente valiosa.

También me gustaría agradecer a mi tutora, la Maestra en Tecnologías de Información Melissa Blanqueto Estrada, por su valiosa orientación. Me dio el consejo que necesitaba para elegir la dirección correcta y completar con éxito mi carrera. A la maestra en Sistemas de Información Laura Dávalos Castilla por su apoyo en la revisión del presente proyecto.

Además, me gustaría agradecer a mis padres por su sabio consejo y todo lo que los llevó a ponerme en la posición que estoy hoy, a mi hermano y hermana a quienes siempre puedo admirar. Siempre están ahí para mí. Finalmente, mi novia por su apoyo y aliento sin fin y luego están mis amigos, que fueron de gran apoyo durante el transcurso de nuestras carreras. A todos ustedes, muchas gracias.

Dedicatoria

Dedico esta tesis a mis dulces y amorosos padres. Primero, le dedico esto a mi madre. No sería la persona que soy hoy si no fuera por ella. Mi madre me inculcó valores que me mantuvieron humilde y trabajando sin importar las circunstancias. En segundo lugar, dedico este trabajo a mi padre, quien me ha demostrado que cualquier cosa se puede lograr con dedicación y la mentalidad correcta.

Mis padres sirven de gran inspiración para mí para que me saque de cualquier obstáculo que enfrente en la vida.

Contenido

Resumen	i
Agradecimientos	ii
Dedicatoria.....	iii
Lista de ilustraciones	vi
Lista de Tablas	vi
Capítulo 1	1
1.1 Introducción	1
1.2 Planteamiento del problema	1
1.3 Objetivo General	2
1.4 Objetivos Específicos	2
1.5 Alcance	2
Capítulo 2 Marco Teórico	3
2.1 Sistemas Operativos para dispositivos móviles.....	3
2.1.1 Android	3
2.1.2 IOS	5
2.1.3 BlackBerry OS.....	6
2.2 Frameworks de desarrollo para dispositivos móviles.....	6
2.2.1 IONIC Framework.....	6
AngularJS.....	7
Arquitectura AngularJS	8
Apache Cordova	8
2.2.2 <i>Ruby on Rails</i>	10
2.3 Bases de Datos.....	12
2.3.1 <i>Firebase</i>	12
2.3.2 SQL	13
2.4 Aplicaciones móviles para eventos (conferencias)	14
2.4.1 <i>QR Attendance Control</i>	14

2.4.2 Hubstaff.....	14
2.5 Metodologías para desarrollo de aplicaciones móviles.....	15
2.5.1 WaterFall.....	15
2.5.1.1 Requerimientos.....	15
2.5.1.2 Análisis	15
2.5.1.3 Diseño.....	15
2.5.1.4 Codificar	15
2.5.1.5 Pruebas.....	16
2.5.1.6 Operaciones	16
2.5.2 Desarrollo Ágil.....	17
2.5.2.1 Scrum.....	17
2.5.2.2 Programación Extrema (Extreme Programming) (Pandu, s.f.)	18
2.5.2.3 Desarrollo Basado en Funciones (Feature-Driven Development) (Agile Methods, 2019)	19
2.5.2.4 Crystal (Agile 101: What Is Agile Methodology?, 2019)	20
2.5.2.5 Proceso Racional Unificado (Rational Unified Process)	20
2.5.2.6 Mobile D.....	21
2.5.4 Desarrollo Rápido de Aplicaciones	21
2.5.4.1 Fase 1: Planificación de Requisitos	22
2.5.4.2 Fase 2: Diseño de Usuario.....	22
2.5.4.3 Fase 3: Construcción Rápida	22
2.5.4.4 Fase 4: Corte	22
Capítulo 3 Desarrollo: Desarrollo Rápido de Aplicaciones.....	24
3.1.1 Fase 1: Planificación de Requerimientos	24
3.1.2 Fase 2: Diseño de Usuario.....	24
3.1.3 Fase 3: Construcción Rápida	25
3.1.4 Fase 4: Corte	26
Capítulo 4 Resultados Y Conclusiones	35
4.1 Introducción	35
4.2 Resultados	35
4.3 Conclusiones.....	38
Bibliografía.....	40
ANEXOS	42

Lista de ilustraciones

Ilustración 1: Arquitectura de Android. Elaboración propia basada en: https://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture	3
Ilustración 2: Capa de Servicios Básicos de iOS. Elaboración propia basada en: https://www.dotnettricks.com/learn/xamarin/understanding-xamarin-ios-build-native-ios-app	5
Ilustración 3: Arquitectura del Ionic Framework.....	7
Ilustración 4: Diagrama de Arquitectura AngularJS. Elaboración propia basada en:	8
Ilustración 5: Arquitectura de Apache Cordova	9
Ilustración 6: Arquitectura de plugins de Apache Cordova.....	10
Ilustración 7: Página de Login de la aplicación.....	26
Ilustración 8: Pantalla del menú principal Ilustración 9: Pantalla de perfil	27
Ilustración 10: Crear un nuevo evento	28
Ilustración 11: Creando un código QR usando un correo electrónico.....	29
Ilustración 12: Lista de eventos para registrar asistencia.....	30
Ilustración 13: Lista de eventos creados Ilustración 14: Registro detallado de un evento.....	31
Ilustración 15: Lista de eventos para editar	32
Ilustración 16: Proceso de Escaneo de Códigos QR en Net Attend. Elaboración propia.....	33
Ilustración 17 Resultados de la encuesta de usabilidad de la aplicación NetAttend	37

Lista de Tablas

Tabla 1 Información básica de los Eventos Acadpemicos que han utilizado NetAttend.....	35
--	----

Capítulo 1

1.1 Introducción

La Universidad de Quintana Roo es la máxima casa de estudios del Estado de Quintana Roo, en ella, se llevan diversos eventos académicos tales como: seminarios de investigación, conferencias, conferencias magistrales, talleres, cursos de actualización, asesorías grupales, círculos de conversación (del área de idiomas), entre otros. Estos eventos tienen como finalidad generar, promover y difundir diversas temáticas académicas y culturales que representan una actividad fundamental del quehacer universitario. Dada la importancia de los eventos académicos, resulta imprescindible contar con un mecanismo que permita gestionar de manera adecuada la asistencia de estudiantes, profesores, invitados y público en general, de tal forma que se puedan tomar mejores decisiones en la logística (como puede ser la selección de la cantidad y tipo de espacios que deben utilizarse), el control de asistencia, generación de constancias, entre otras.

1.2 Planteamiento del problema

El control de asistencia es un aspecto que ha sido complicado llevar a cabo en el área de logística y organización de eventos académicos. Por citar un ejemplo, desde el año 2004, los organizadores del Foro de Estudios en Lenguas Internacional (FEL), evento de talla internacional con 14 ediciones anuales, han requerido una estrategia que les permita determinar de manera rápida y exacta la asistencia a cada uno de los eventos que convergen en ese foro. Debe destacarse que muchas veces los eventos académicos tienen duración de 3 a 5 días, sus asistentes y participantes son de diferentes áreas geográficas y las constancias que se les emiten deben realizarse con la mayor rapidez posible. Por lo anterior, se requiere desarrollar una aplicación que satisfaga los requisitos funcionales para gestionar el registro de asistentes, generación de constancias y estadísticas del evento.

1.3 Objetivo General

Desarrollar una aplicación móvil que permita registrar la asistencia y gestionar la información de eventos académicos.

1.4 Objetivos Específicos

- i. Analizar los requisitos funcionales de la aplicación.
- ii. Diseñar la arquitectura de la aplicación.
- iii. Diseñar las interfaces de la aplicación.
- iv. Diseñar e implementar una base de datos en línea.
- v. Codificar las funciones de cada módulo de la aplicación.
- vi. Probar el funcionamiento de la aplicación.
- vii. Realizar modificaciones para asegurar el correcto funcionamiento de la aplicación.
- viii. Crear el instalador de la aplicación.

1.5 Alcance

La aplicación administrará sesiones de eventos académicos de manera simultánea, registrándolos en una base de datos en la nube para que posteriormente puedan ser generados los reportes correspondientes.

Capítulo 2 Marco Teórico

2.1 Sistemas Operativos para dispositivos móviles

2.1.1 Android

Android® es el nombre del sistema operativo utilizado en muchos *smartphones* y *tablets*. Es propiedad y está mantenido por Google. Google compró Android en 2005 (Ros, 2014) y lo lanzó para móviles en 2007.

Un sistema operativo es básicamente lo que aparece en la pantalla en un dispositivo, el *software* subyacente con el que interactúa. (Mas adelante, n.d.) Esto es diferente a las aplicaciones que puede descargar para usarlo como lo hace en la mayoría de los *smartphones*.

Cualquiera puede descargar el *SDK* (kit de desarrollo de *software*) y escribir aplicaciones para teléfonos Android y comenzar a desarrollar para *Google Play*. Los desarrolladores que venden aplicaciones en el mercado de *Google Play* pagan aproximadamente el 30% de su precio de venta en tarifas que se destinan a mantener el mercado de Google Play. (Transaction fees, n.d.) Algunos dispositivos no incluyen soporte para Google Play y pueden usar un mercado alternativo. Android es una plataforma muy atractiva para los consumidores y desarrolladores.

La arquitectura de Android se clasifica en cinco partes:

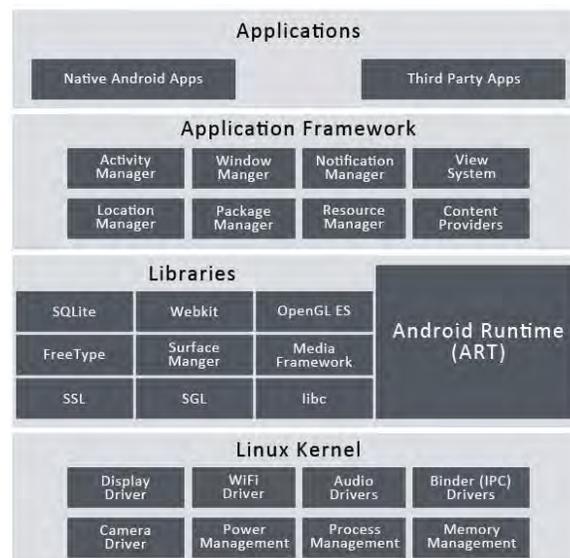


Ilustración 1: Arquitectura de Android. Elaboración propia basada en: https://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture

I. Kernel de Linux

Es el corazón de la arquitectura de Android que existe en la raíz de la arquitectura de Android. El kernel de Linux es responsable de los controladores de dispositivos, la administración de energía, la administración de memoria, la administración de dispositivos y el acceso a los recursos. (JavaTPoint, n.d.)

II. Librería Nativa

En la parte superior del *kernel* de Linux, hay librerías como *WebKit*, OpenGL, FreeType, *SQLite*, Media, librería de tiempo de ejecución de C (libc), etc. (JavaTPoint, n.d.)

La librería *WebKit* es responsable del soporte del navegador, *SQLite* es para la base de datos, FreeType para el soporte de fuentes, medios para reproducir y grabar formatos de audio y video. (JavaTPoint, n.d.)

III. Android Runtime

En el tiempo de ejecución de Android, existen librerías centrales y *DVM* (Dalvik Virtual Machine) que es responsable de ejecutar la aplicación de Android. *DVM* es como Java Virtual Machine, pero está optimizado para dispositivos móviles. Consume menos memoria y proporciona un rendimiento rápido. (JavaTPoint, n.d.)

IV. Android Framework

En la parte superior de las librerías nativas y el tiempo de ejecución de Android, hay un *framework* de Android. El *framework* de Android incluye *API* de Android como *UI* (Interfaz de usuario), telefonía, recursos, ubicaciones, Proveedores de contenido (datos) y gestores de paquetes. Proporciona muchas clases e interfaces para el desarrollo de aplicaciones de Android. (JavaTPoint, n.d.)

V. Aplicaciones

En la parte superior del *framework* de Android, hay aplicaciones. Todas las aplicaciones, como el hogar, los contactos, la configuración, los juegos y los navegadores utilizan un marco de Android que utiliza librerías y tiempo de ejecución de Android. El tiempo de ejecución de Android y las librerías nativas están utilizando Linux kernel. (JavaTPoint, n.d.)

2.1.2 IOS

iOS se deriva de Mac OS X y es un sistema operativo similar a Unix. Hay cuatro capas de abstracción dentro de iOS:

- I. Capa principal del sistema operativo: Proporciona funciones de bajo nivel, así como marcos para la seguridad e interacción con *hardware* externo. (iOS Architecture, n.d.)
- II. Capa de servicios principales: proporciona servicios requeridos por las capas superiores. (The iPhone OS Core Services Layer, n.d.)
- III. Media Layer: Proporciona las tecnologías necesarias para gráficos, audio y video. (iOS Architecture, n.d.)
- IV. Cocoa Touch Layer: donde se ubican los *frameworks*, que a menudo se usan al crear una aplicación. (The iPhone OS Cocoa Touch Layer, n.d.)



Ilustración 2: Capa de Servicios Básicos de iOS. Elaboración propia basada en:
<https://www.dotnettricks.com/learn/xamarin/understanding-xamarin-ios-build-native-ios-app>

iOS viene con muchas aplicaciones predeterminadas, que incluyen un cliente de correo electrónico, un navegador web *Safari*, un reproductor multimedia portátil (*iPod*) y la aplicación de teléfono.

Los desarrolladores pueden usar el kit de desarrollo de *software* (*SDK*) de iOS para crear aplicaciones para dispositivos móviles de Apple. El *SDK* incluye herramientas e interfaces para desarrollar, instalar, ejecutar y probar aplicaciones. Las aplicaciones nativas se pueden escribir utilizando los marcos del sistema iOS y el lenguaje de programación *Objective-C*. En el *SDK* de iOS se incluyen Xcode Tools, que incluye un entorno de desarrollo integrado (*IDE*) para administrar proyectos de aplicaciones, una

herramienta gráfica para crear la interfaz de usuario y una herramienta de depuración para analizar el rendimiento en tiempo de ejecución. También incluye un simulador de iOS, que permite a los desarrolladores probar aplicaciones en una Mac, y una biblioteca de desarrolladores de iOS, que proporciona toda la documentación necesaria y material de referencia.

2.1.3 BlackBerry OS

El sistema operativo BlackBerry es un sistema operativo móvil patentado desarrollado por BlackBerry para su uso en la popular línea de teléfonos inteligentes BlackBerry. Después de haber pasado por muchas transiciones desde su versión original 1.0 en 1999, el sistema operativo BlackBerry es mejor conocido por su seguridad, capacidad de multitarea e interoperabilidad con las infraestructuras de correo electrónico corporativo que utilizan servicios como Microsoft Exchange, Lotus Domino o Novell GroupWise. (Cassavoy, 2018)

2.2 Frameworks de desarrollo para dispositivos móviles

2.2.1 IONIC Framework

El Ionic *Framework* es una biblioteca de código abierto de componentes optimizados para dispositivos móviles en *JavaScript*, *HTML* y *CSS*. (Chapter 1: All About Ionic, n.d.) Esto implica que los desarrolladores pueden manejar la estructura del código de manera efectiva, ahorrando esfuerzos, dinero y tiempo. Proporciona a los desarrolladores herramientas para crear aplicaciones robustas y optimizadas que funcionan tanto en Android como en iOS. (Wiegert, 2019) Tiene una característica de un código base. Esto implica que las aplicaciones se implementan a través de Apache Cordova con una base de código solitaria, y la aplicación se adapta en consecuencia al dispositivo en el que está trabajando. Es, con mucho, el *framework* más rico por esta razón y porque es una combinación de dos *frameworks* extremadamente poderosos, Angular y Cordova. Se basa en el *framework* de la interfaz de usuario (UI) de Sass, que está especialmente optimizado y diseñado para *frameworks* operativos móviles que también proporcionan una gran cantidad de componentes de interfaz de usuario para crear aplicaciones móviles potentes.

Ionic es básicamente un módulo AngularJS y una hoja de estilo que se puede usar para crear aplicaciones que se parecen a las nativas. Por lo tanto, es muy fácil comenzar con Ionic, especialmente cuando uno ya conoce AngularJS. La mayoría de los componentes, donde uno de los componentes

es una lista, un cuadro de alternancia o de diapositiva, son solo directivas de AngularJS. Esto a menudo da como resultado un marcado altamente legible y una buena productividad.

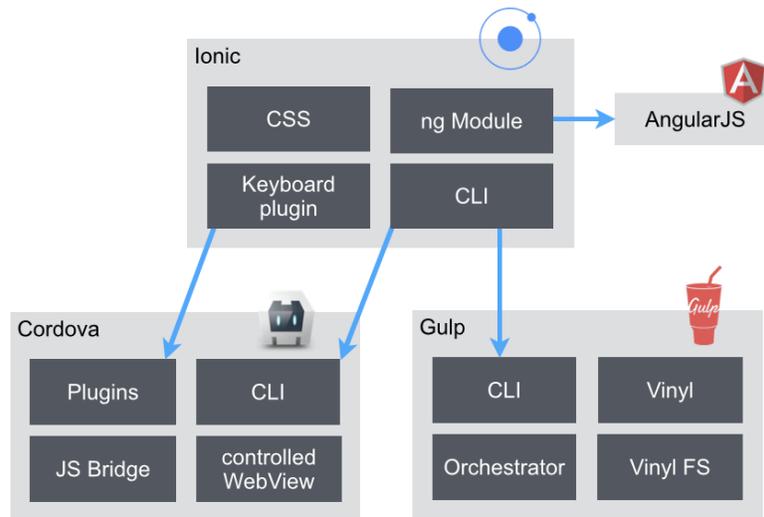


Ilustración 3: Arquitectura del Ionic Framework.

El Ionic *Framework*, en su núcleo, consta de cuatro partes: (Ripkens, 14)

1. Un *stylesheet* que define un diseño base móvil optimizado. Este diseño sirve como base para la aplicación. Incluye muchas mejores prácticas en las que ya no necesita aplicar.
2. El módulo AngularJS no es nada especial cuando estás acostumbrado a AngularJS. Define las directivas, patrones de navegación y mejores prácticas para que no tenga que pensar en ello.
3. El *CLI* lo despega rápidamente y puede actuar como una especie de proxy para el *CLI* de Cordova y Gulp.
4. El último componente es un *plugin* de teclado. Aunque técnicamente no es necesario, el complemento proporciona más información sobre el estado actual de la aplicación.

AngularJS

AngularJS es un *framework* de código abierto de Modelo-Vista-Controlador que es similar al *JavaScript framework*. AngularJS es probablemente uno de los *frameworks* modernos para web más populares disponibles en la actualidad. Este *framework* se utiliza para desarrollar aplicaciones en su mayoría de una sola página. Este *framework* ha sido desarrollado por un grupo de desarrolladores de Google. Debido al apoyo total de Google y las ideas de un foro amplio de la comunidad, el *framework* siempre

se mantiene actualizado. Además, siempre incorpora las últimas tendencias de desarrollo en el mercado.

Arquitectura AngularJS

AngularJS sigue la arquitectura MVC, el diagrama del *framework* MVC se muestra a continuación: (Guru99, 2019)

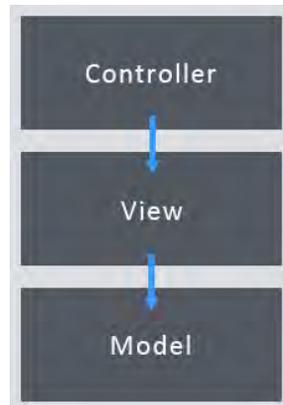


Ilustración 4: Diagrama de Arquitectura AngularJS. Elaboración propia basada en: <https://www.guru99.com/angularjs-introduction.html>

El Controlador representa la capa que tiene la lógica de negocio. Los eventos de usuario activan las funciones que están almacenadas dentro de su controlador. Los eventos del usuario son parte del controlador.

Las vistas se utilizan para representar la capa de presentación que se proporciona a los usuarios finales.

Los modelos se utilizan para representar los datos. Los datos en el modelo pueden ser tan simples como tener declaraciones primitivas. Por ejemplo, si se está manteniendo una aplicación de estudiante, su modelo de datos podría tener una identificación de estudiante y un nombre. O también puede ser complejo al tener un modelo de datos estructurado. Si se está manteniendo una solicitud de propiedad de automóvil, puede tener estructuras para definir el propio vehículo en términos de la capacidad de su motor, la capacidad de los asientos, etc.

Apache Cordova

Es posible desarrollar aplicaciones de varias maneras. Una de las opciones es volverse nativo y desarrollar una aplicación varias veces para lograr la mejor integración y, potencialmente, la mejor

experiencia de usuario. Las aplicaciones híbridas están en aumento y vienen en varias maneras. La opción más común es basar la aplicación en tecnologías web y utilizar el hecho de que la mayoría de los dispositivos son compatibles con los estándares web, es decir, *HTML*, *CSS* y *JavaScript*. La aplicación incluso sigue siendo instalable a través de *app stores*.

Cordova es una plataforma para aplicaciones híbridas basadas en web. Hace el trabajo de admitir e interactuar con varias *API* específicas del dispositivo y combinar la esencia de todas las *API* nativas en una *API* de *JavaScript* a la que puede acceder la aplicación híbrida. Las aplicaciones híbridas se ejecutan dentro de una vista web controlada por el *framework* de Cordova. Este es un trabajo complicado considerando la cantidad de plataformas que existen.

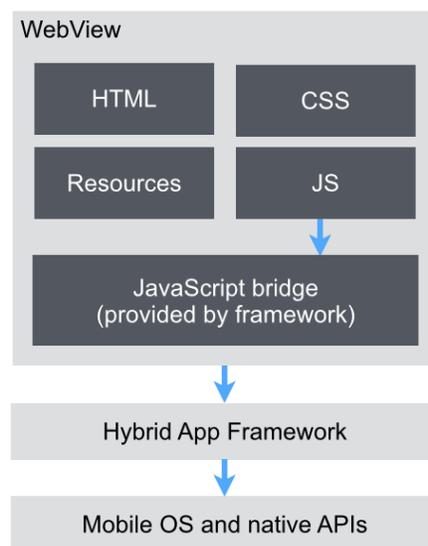


Ilustración 5: Arquitectura de Apache Cordova
<https://blog.codecentric.de/en/2014/11/ionic-angularjs-framework-on-the-rise/>

Además, Cordova tiene el concepto de complementos que encapsulan la interacción con una (a veces muchas) *API* de dispositivos nativos. Los complementos para la recuperación de información básica del dispositivo, el acceso a archivos locales y el acceso a la geolocalización del dispositivo son solo algunos ejemplos del ecosistema del complemento de Cordova. Los complementos siempre constan de dos partes. La primera es una parte de *JavaScript* que se ejecuta dentro de *WebView* que expone una *API* a la aplicación híbrida. La segunda parte es específica de la plataforma y escrita en el idioma nativo de la plataforma, por ejemplo, Java para Android y Objective-C para iOS. Las *API* nativas están

controladas por esa segunda parte. La siguiente imagen ilustra la arquitectura de *plugin* de alto nivel. (Ripkens, 14)

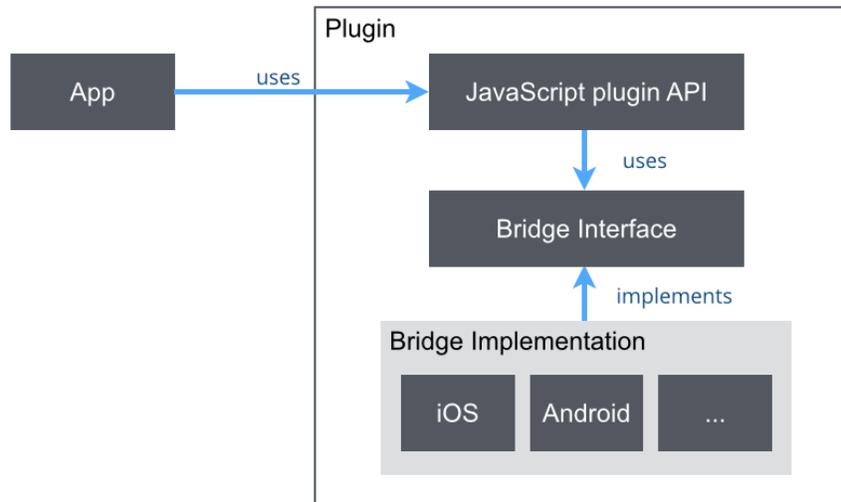


Ilustración 6: Arquitectura de plugins de Apache Cordova
<https://blog.codecentric.de/en/2014/11/ionic-angularjs-framework-on-the-rise/>

2.2.2 Ruby on Rails

Ruby on Rails (RoR) es uno de los *frameworks* web más populares y el que ha impulsado el lenguaje Ruby en sí. Fue inicialmente diseñado y traído a la vida por David Heinemeier Hansson. Se genera una gran cantidad de proyectos diariamente en la web usando *Ruby on Rails*. Su objetivo principal es ayudar al desarrollador a lidiar con los problemas y preocupaciones de la tecnología, y centrarse en resolver el problema comercial.

RoR sigue el patrón arquitectónico Modelo-Vista-Controlador (MVC). Además, sigue la regla de la convención sobre la configuración (también conocida como codificación por convención), lo que significa que ya ha tomado algunas decisiones por el programador el cual no tiene que preocuparse por estas cosas mientras siga las convenciones de RoR. (Panos, 2017)

Aquí está la lista de los 6 componentes básicos de la arquitectura RoR:

I. Modelos

Los modelos son el núcleo del negocio. Son la representación del dominio empresarial en el lenguaje Ruby, el modelo de dominio. En RoR, los modelos también son la interfaz de nuestro almacenamiento persistente. En otras palabras, ofrecen métodos para transferir el estado residente en memoria de una instancia del modelo a la base de datos. Los modelos no son la aplicación web. Sin embargo, uno puede hacer muchas cosas usando solo los modelos. (Panos, 2017)

II. *Ruby on Rails Server*

El servidor *Ruby on Rails*, es un ejecutable que crea una instancia de un servidor web. Se utiliza para aceptar las solicitudes desde el navegador. Escucha en una IP y un Puerto. Mientras realiza el desarrollo, *Rails Server* se inicia en su máquina local y escucha las solicitudes en el puerto 3000, de modo que puede acceder a él utilizando la siguiente dirección web: `http://0.0.0.0:3000/`. *Rails Server* toma una solicitud y delega el procesamiento a varios componentes de los que está compuesto. Además, sabe cómo escribir registros útiles dentro de la carpeta de registros. (Panos, 2017)

III. Rutas

Las rutas comprenden una forma de indicar qué rutas de *URL* entienden la aplicación y qué parte del código debe manejar las solicitudes correspondientes. Las rutas se configuran dentro de un archivo llamado `config/route.rb` y para cada ruta que queremos que maneje la aplicación, debemos especificar el controlador y la acción de este controlador responsable para manejar la solicitud correspondiente. (Panos, 2017)

IV. Controladores y Acciones

Una vez que el enrutador ha determinado qué controlador utilizar para una solicitud, el controlador es responsable de dar sentido a la solicitud y de producir la salida adecuada. Afortunadamente, el controlador hace la mayor parte del trabajo y utiliza convenciones inteligentes para que esto sea lo

más sencillo posible. Para la mayoría de las aplicaciones *RESTful* convencionales, el controlador recibirá la solicitud, invisible para el desarrollador, buscará o guardará datos de un modelo y usará una vista para crear una salida *HTML*. Si el controlador necesita hacer las cosas de manera un poco diferente, eso no es un problema, esta es la forma más común en que funciona un controlador.

Por lo tanto, un controlador puede considerarse como un intermediario entre los modelos y las vistas. Hace que los datos del modelo estén disponibles para la vista para que pueda mostrar esos datos al usuario, y guarda o actualiza los datos del usuario al modelo. (Panos, 2017)

V. Vistas

Las vistas son responsables de decirle a *Rails* cuál es el contenido del *HTML* que se enviará al navegador después de emitir la solicitud. Sin embargo, las vistas se basan en datos que deberían haber sido preparados por el controlador. Además de estos datos, la vista también necesita un diseño, que es una parte del formato *HTML* en el que estará la vista. Por lo tanto, diferentes vistas pueden usar el mismo diseño. En otras palabras, el diseño generalmente tiene el contenido que es común a muchas vistas y, esencialmente, incrusta el contenido de la vista. (Panos, 2017)

VI. Assets

El contenido de la vista como incrustado en el diseño se envía al navegador que ha solicitado una página. Pero, a excepción de ese contenido, los *assets* también se envían. Los *assets* son, por lo general, código *JavaScript*, código *CSS* y otros medios, como imágenes. (Panos, 2017)

2.3 Bases de Datos

2.3.1 *Firebase*

Firebase es un *Backend-as-a-Service* (BaaS) que ofrece varias características útiles cuando se trata de la gestión de datos. *Firebase* funciona en tiempo real. Envía nuevos datos tan pronto como se actualiza. Cuando se realiza un cambio, todos los clientes conectados reciben la actualización casi instantáneamente. *Firebase auth* tiene un sistema integrado de autenticación de correo electrónico / contraseña. También es compatible con *OAuth2 para Google, Facebook, Twitter y GitHub*. (Esplin, 2016)

2.3.2 SQL

SQLite es una biblioteca de *software* que proporciona un sistema de gestión de bases de datos relacionales. *SQLite* requiere configuración mínima, administración de base de datos y recursos.

Las características notables en esta base de datos incluyen: autónomo, sin servidor, de configuración cero, transaccional.

I. Autónomo

SQLite requiere un soporte mínimo del sistema operativo o de la biblioteca externa. Esto hace que *SQLite* sea utilizable en cualquier entorno, especialmente en dispositivos integrados como iPhones, teléfonos Android, consolas de juegos, reproductores de medios portátiles, etc. (What Is SQLite, n.d.)

II. Sin Servidor

Normalmente, un *RDBMS* como MySQL, PostgreSQL, etc., requiere un proceso de servidor separado para funcionar. Las aplicaciones que desean acceder al servidor de bases de datos utilizan el protocolo *TCP/IP* para enviar y recibir solicitudes. Esto se llama arquitectura cliente / servidor. *SQLite* no funciona de esta manera. *SQLite* no requiere un servidor para ejecutarse. La base de datos *SQLite* está integrada con la aplicación que accede a la base de datos. Las aplicaciones interactúan con la base de datos *SQLite*, leen y escriben directamente desde los archivos de la base de datos almacenados en el disco. (What Is SQLite, n.d.)

III. Cero Configuración

Debido a la arquitectura sin servidor, no es necesario que "instale" *SQLite* antes de usarlo. No hay ningún proceso de servidor que deba configurarse, iniciarse y detenerse. *SQLite* no utiliza ningún archivo de configuración. (What Is SQLite, n.d.)

IV. Transaccional

Todas las transacciones en *SQLite* son totalmente compatibles con *ACID*. Significa que todas las consultas y cambios son atómicos, consistentes, aislados y duraderos.

En otras palabras, todos los cambios dentro de una transacción se producen completamente o no se producen, incluso cuando se produce una situación inesperada como un fallo de la aplicación, un fallo de alimentación o un fallo del sistema operativo. (What Is SQLite, n.d.)

2.4 Aplicaciones móviles para eventos (conferencias)

2.4.1 QR *Attendance Control*

Las aplicaciones móviles que registran asistencia ya han existido en el mercado durante los últimos años. Si se compara todas las aplicaciones móviles de registro de asistencia, es posible que no vea mucha diferencia, ya que todas tienen características similares. Para comprender mejor la funcionalidad de la aplicación, nos centraremos en otras aplicaciones similares que ya existen y veremos cómo funcionan. QR *Attendance Control* es una aplicación que se destaca de todas las aplicaciones que existen. Esta aplicación fue creada para el control de asistencia mediante el escaneo de códigos QR. QR *Attendance*, coincide con la aplicación *NetAttend* en su simplicidad. Esta aplicación le permite controlar la asistencia a un evento mediante códigos QR, para que pueda registrar la hora de llegada y salida, y también le informa la cantidad de tiempo que cada persona estuvo presente en el evento. Se requiere el nombre de la persona que asiste para que se genere un código QR para que se escanee y se registre como asistencia.

2.4.2 *Hubstaff*

Hubstaff es una aplicación que fue creada para administrar a los trabajadores en una empresa. Es una aplicación de seguimiento de tiempo para empleados basada en la nube con monitoreo GPS y nómina. Una similitud encontrada en esta aplicación con la aplicación *NetAttend* es que está basada en la nube. *Hubstaff*, al igual que la aplicación *NetAttend*, funciona en tiempo real, lo que significa que cada vez que se registra un cambio, la aplicación se actualiza automáticamente. Se requiere una cuenta de *Hubstaff* antes de iniciar sesión en la aplicación. Un simple toque en una tarea específica que debe hacerse y así es cómo se registra un usuario. También permite la terminación de esa tarea cuando se realiza el trabajo. También hay una opción para ver los registros de los datos generados.

2.5 Metodologías para desarrollo de aplicaciones móviles

2.5.1 *WaterFall*

El modelo de cascada (*Waterfall*), introducido por primera vez por el Dr. Winston W. Royce, es un modelo de desarrollo que enfatiza que se debe seguir una progresión lógica a lo largo del ciclo de vida del desarrollo de software (*Software Development Life Cycle*). Si bien la popularidad del modelo de cascada ha disminuido en los últimos años a favor de metodologías más ágiles, la naturaleza lógica del proceso secuencial utilizado en el método de cascada no se puede negar, y sigue siendo un proceso de diseño común.

Implementar un modelo de cascada en un nuevo proyecto de software es un proceso sencillo. (Powell-Morse, 16)

2.5.1.1 Requerimientos

Durante esta fase inicial, los requisitos potenciales de la aplicación se analizan metódicamente y se escriben en un documento de especificaciones que sirve de base para todo desarrollo futuro. El resultado suele ser un documento de requisitos que define qué debe hacer la aplicación, pero no cómo debe hacerlo.

2.5.1.2 Análisis

Durante esta segunda etapa, el sistema se analiza para generar correctamente los modelos y la lógica empresarial que se utilizará en la aplicación.

2.5.1.3 Diseño

Esta etapa cubre en gran medida los requisitos de diseño técnico, como el lenguaje de programación, las capas de datos, los servicios, etc. Generalmente, se creará una especificación de diseño que describe cómo exactamente se implementará técnicamente la lógica de negocios cubierta en el análisis.

2.5.1.4 Codificar

El código fuente real finalmente se escribe en esta cuarta etapa, implementando todos los modelos, la lógica de negocios y las integraciones de servicios que se especificaron en las etapas anteriores.

2.5.1.5 Pruebas

Durante esta etapa, el control de calidad, los probadores beta y todos los demás probadores descubren e informan sistemáticamente los problemas dentro de la aplicación que deben resolverse. No es infrecuente que esta fase cause una "repetición necesaria" de la fase de codificación anterior, para que los errores revelados se eliminen correctamente.

2.5.1.6 Operaciones

Finalmente, la aplicación está lista para su implementación en un entorno en vivo. La etapa de operaciones implica no solo el despliegue de la aplicación, sino también el soporte y el mantenimiento posteriores que pueden ser necesarios para mantenerla funcional y actualizada.

Ventajas: (Waterfall Methodology: Advantages, Disadvantages And When to Use It?, 2018)

- Los requisitos se establecen de manera clara y precisa, permanecen sin cambios durante todo el desarrollo del proyecto.
- La documentación detallada de cada etapa de desarrollo proporciona resistencia a los cambios en los recursos humanos: un nuevo desarrollador puede obtener rápidamente toda la información necesaria.
- La planificación cuidadosa de la estructura de desarrollo del proyecto reduce el número de problemas problemáticos.
- Se establecen los puntos de inicio y finalización de cada fase, lo que facilita la medición del progreso.
- Las tareas permanecen lo más estables posible durante todo el proceso de desarrollo.
- Proporciona un fácil control y transparencia para el cliente debido a un estricto sistema de informes.
- La fecha de lanzamiento del producto terminado, así como su costo final, se pueden calcular antes del desarrollo.

Desventajas: (Waterfall Methodology: Advantages, Disadvantages And When to Use It?, 2018)

- Todos los requisitos deben ser conocidos antes del desarrollo, lo que retrasa considerablemente el inicio del proyecto.

- El bajo nivel de flexibilidad dificulta la realización de cambios durante el desarrollo, o incluso lo hace completamente imposible.
- Existe la necesidad de una gestión estricta y un monitoreo regular, para que el proyecto cumpla con el plazo.
- El cliente no tiene la oportunidad de familiarizarse con el sistema de antemano, por lo que no ve el producto hasta el momento de su finalización.
- En caso de que quede claro en el proceso de desarrollo que el producto no cumple con los requisitos del mercado, no habrá espacio para cambios.

2.5.2 Desarrollo Ágil

En 1970, el Dr. William Royce publicó un documento que discutía la administración y el desarrollo de grandes sistemas de software. El documento esbozó sus ideas específicas sobre el desarrollo secuencial. Su presentación indicó que un proyecto podría desarrollarse de manera muy similar a un producto en una línea de ensamblaje. Cada fase del desarrollo tenía que estar completa antes de que la siguiente fase pudiera comenzar. (Focus, 2018)

Existen varias metodologías ágiles; Todos comparten filosofías, características y prácticas similares. Sin embargo, desde el punto de implementación, cada ágil tiene sus propias prácticas, terminología y tácticas. Algunos de los principales componentes de la metodología de desarrollo de software ágil incluyen:

2.5.2.1 Scrum

Scrum es un marco de gestión con capacidades de gran alcance para controlar y gestionar las iteraciones e incrementos en todos los tipos de proyectos. Puede usarse con otras metodologías ágiles para diversas prácticas de ingeniería. *Scrums* ha crecido en popularidad dentro de la comunidad de desarrollo de software ágil porque es simple y tiene una tasa de productividad comprobada.

Ventajas: (Chandana, 2019)

- *Scrum* asegura el uso efectivo del tiempo y el dinero.
- Los grandes proyectos se dividen en *sprints* (incrementos) fácilmente manejables.
- Los desarrollos se codifican y se prueban durante las revisiones.
- Funciona bien para proyectos de desarrollo en rápido movimiento.

- *Scrum*, siendo ágil, adopta la retroalimentación de los clientes.

Desventajas: (Chandana, 2019)

- El *Scrum* a menudo conduce a un aumento de alcance, debido a la falta de una fecha final definida.
- Las posibilidades de fracaso del proyecto son altas si los individuos no son muy comprometidos o cooperativos.
- Adoptar el marco *Scrum* en grandes equipos es un reto.
- El marco puede tener éxito solo con miembros experimentados del equipo.
- Las reuniones diarias a veces frustran a los miembros del equipo.
- Si cualquier miembro del equipo se va en medio de un proyecto, puede tener un gran impacto negativo en el proyecto.
- La calidad es difícil de implementar, hasta que el equipo pasa por un proceso de prueba agresivo.

2.5.2.2 Programación Extrema (*Extreme Programming*) (Pandú, s.f.)

La programación extrema (XP) fue descrita originalmente por Kent Beck. Es una de las metodologías ágiles más populares y controvertidas. XP es un método altamente disciplinado de entregar continuamente software de alta calidad más rápido. El cliente está activamente involucrado con el equipo unido para realizar la planificación, las pruebas y la retroalimentación rápida para entregar software de trabajo con frecuencia. El software debe entregarse en intervalos cada 1 a 3 semanas.

Ventajas:

- Las pruebas regulares en la etapa de desarrollo aseguran la detección de todos los errores, y el uso de pruebas de validación aprobadas por el cliente para determinar la finalización exitosa de un bloque de codificación garantiza la implementación de solo lo que el cliente quiere y nada más.
- XP permite que se realicen cambios en los requisitos iniciales a través de retroalimentación durante la fase de desarrollo.
- La programación extrema recorta las actividades improductivas para reducir los costos y la frustración de todos los involucrados. Permite a los desarrolladores centrarse en la codificación

en lugar de perder el tiempo en papeleo y reuniones innecesarios y elimina la necesidad de evaluadores independientes.

- La programación extrema reduce los riesgos relacionados con la programación, al dividir las tareas en módulos, distribuye el riesgo y reduce la dependencia de cualquier arquitecto, gerente de proyecto o codificador individual.

Desventajas:

- La necesidad de un contacto constante con los clientes, ya que el éxito del proyecto depende de la recopilación de datos en muchas etapas del proceso de desarrollo.
- La falta de documentación adecuada crea problemas en los productos grandes cuando los miembros del proyecto se van y los miembros nuevos llegan más tarde.

2.5.2.3 Desarrollo Basado en Funciones (Feature-Driven Development) (Agile Methods, 2019)

Jeff De Luca, junto con colaboradores A.m. Rajashima, Lim Bak Wee, Paul Szego, Jon Kern y Stehen Palmer desarrollaron el desarrollo basado en funciones (FDD). Es un proceso de iteración corto impulsado por el modelo que comienza estableciendo primero la forma del modelo ágil. Las iteraciones sobre "diseño por característica, compilación por característica" se llevan a cabo cada dos semanas. Las características atraen a los clientes porque son pequeñas y útiles.

Ventajas:

- El desarrollo impulsado por funciones ayuda a mover proyectos de mayor tamaño y obtener un éxito repetible.
- Ayuda a llevar el trabajo hecho en un tiempo corto y de la manera más fácil;
- Este tipo de modelo está construido sobre estándares establecidos para la industria de desarrollo de software, por lo que ayuda a un desarrollo fácil y las mejores prácticas reconocidas en la industria;
- El desarrollo basado en características ofrece un seguimiento del progreso bien definido y capacidades de informes;
- FDD también soporta múltiples equipos trabajando en paralelo.

Desventajas:

- No es una metodología ideal para proyectos más pequeños, por lo que no es buena para un desarrollador de software individual;
- La alta dependencia del desarrollador principal significa que la persona debe estar completamente equipada para un acto como coordinador, diseñador principal y mentor;
- No se proporciona documentación escrita a los clientes en esta metodología, por lo que no pueden obtener una prueba de su propio software;
- Promueve la propiedad individual del código en lugar de la propiedad compartida del equipo;
- La iteración no está bien definida por el proceso como otras metodologías ágiles;
- El aspecto centrado en el modelo puede tener enormes impactos cuando se trabaja con un sistema antiguo que no tiene un modelo de sistema.

2.5.2.4 Crystal (Agile 101: What Is Agile Methodology?, 2019)

La metodología Crystal es uno de los enfoques más livianos y adaptables en el desarrollo de software. Está compuesto por varios procesos ágiles, entre los que se incluyen Clear, Crystal Yellow, Crystal Orange y otros métodos caracterizados de forma única. Hay varios factores que impulsan estos procesos, incluidos el tamaño del equipo, la criticidad del sistema y las prioridades del proyecto.

La familia Crystal se centra en la comprensión de que cada proyecto tiene características únicas, por lo tanto, las políticas y las prácticas deben personalizarse para adaptarse a estas características.

2.5.2.5 Proceso Racional Unificado (Rational Unified Process)

Proceso Racional Unificado (PRU) es una metodología de desarrollo de software ágil. PRU divide el ciclo de vida del proyecto en cuatro fases. Durante cada una de las fases, se llevan a cabo las seis disciplinas principales de desarrollo: modelado de negocios, requisitos, análisis y diseño, implementación, prueba y despliegue. Sin embargo, ciertos procesos son más importantes y requieren más tiempo en cada etapa. Cada una de las cuatro fases tiene el objetivo principal, que debe completarse antes de que el proyecto pueda avanzar a la siguiente fase. El objetivo principal de PRU es crear software de alta calidad con un presupuesto y un marco de tiempo predecibles. Cada una de las fases del ciclo de vida puede repetirse, si es necesario, hasta que se cumplan los objetivos principales. Una vez que la etapa de transición se completa con éxito, el proyecto se termina.

Ventajas:

- Es capaz de resolver de manera proactiva los riesgos del proyecto asociados con la evolución de los requisitos del cliente que requieren una gestión cuidadosa de las solicitudes de cambio;
- Se requiere menos tiempo para la integración a medida que el proceso de integración continúa a lo largo del ciclo de vida del desarrollo del software;
- El tiempo de desarrollo requerido es menor debido a la reutilización de componentes.

Desventajas:

- Los miembros del equipo deben ser expertos en su campo para desarrollar un software bajo esta metodología;
- El proceso de desarrollo es demasiado complejo y desorganizado;
- En proyectos que utilizan nueva tecnología, la reutilización de componentes no será posible. Por lo tanto, el ahorro de tiempo que uno podría haber hecho será imposible de cumplir;
- La integración a lo largo del proceso de desarrollo de software, en teoría suena bien. Pero en proyectos particularmente grandes con múltiples flujos de desarrollo, solo aumentará la confusión y causará más problemas durante las etapas de prueba.

2.5.2.6 Mobile D

Para superar los desafíos involucrados en el desarrollo de aplicaciones móviles, se desarrolló Mobile-D. El enfoque se basa en la Programación Extrema, las metodologías Crystal y el Rational Unified Process. El enfoque de Mobile-D está optimizado para un equipo de menos de diez desarrolladores que trabajan en un espacio de oficina, con el objetivo de ofrecer una aplicación móvil totalmente funcional en un corto período de tiempo. Un proyecto de desarrollo, siguiendo el enfoque de Mobile-D, se divide en cinco iteraciones. Estas fases son: configuración, núcleo, núcleo2, estabilización y conclusión. Cada fase consta de tres tipos diferentes de días de desarrollo: día de planificación, día laborable y día de lanzamiento. Si varios equipos desarrollan simultáneamente diferentes partes del mismo producto, también se necesita un día de la integración.

2.5.4 Desarrollo Rápido de Aplicaciones

El desarrollo rápido de aplicaciones (RAD) describe un método de desarrollo de software que enfatiza en gran medida el prototipado rápido y la entrega iterativa. El desarrollo rápido de aplicaciones es una forma de metodología de desarrollo de software ágil. A diferencia de los métodos de cascada, RAD

enfatisa el software de trabajo y la retroalimentación de los usuarios sobre la planificación estricta y el registro de requisitos. Esto significa que hay más acción, muchas pruebas.

2.5.4.1 Fase 1: Planificación de Requisitos

Cada pieza de software o aplicación está construida por una razón. RAD comienza con averiguar qué se supone que su proyecto debe lograr. En esta fase, los usuarios y los desarrolladores realizan una conferencia para establecer los requisitos de la aplicación. Una vez que se determinan los requisitos y se establece un cronograma estimado del proyecto.

2.5.4.2 Fase 2: Diseño de Usuario

Una vez que se determinan los requisitos, el plazo y el presupuesto, los desarrolladores deben comenzar a trabajar en modelos funcionales de inmediato. Los diseñadores crearán y mejorarán los prototipos de trabajo hasta que el producto final esté listo para ser revelado.

2.5.4.3 Fase 3: Construcción Rápida

Una vez que comienza el diseño básico del usuario y del sistema, la fase de construcción es donde se lleva a cabo la mayor parte de la codificación, las pruebas y la integración reales de la aplicación. Junto con el diseño del usuario, la fase de construcción rápida se repite tantas veces como sea necesario, a medida que se requieran nuevos componentes o se realicen modificaciones para satisfacer las necesidades del proyecto.

2.5.4.4 Fase 4: Corte

La etapa final del corte les permite a los desarrolladores a los componentes a un entorno de producción en vivo, donde se puede llevar a cabo un cabo las pruebas necesarias para una gran escala o la capacitación del equipo.

Ventajas:

- Reducción del tiempo de desarrollo;
- Incrementa la reutilización de componentes;
- Se producen revisiones iniciales rápidas;
- Alienta la retroalimentación de los clientes;
- La integración desde el principio resuelve muchos problemas de integración.

Desventajas:

- Depende de un equipo fuerte y actuaciones individuales para identificar los requisitos del cliente;
- Solo el sistema que puede modularizarse puede construirse usando RAD;
- Requiere desarrolladores / diseñadores altamente calificados;
- Alta dependencia en habilidades de modelado;
- No se puede aplicar a proyectos de bajo costo, ya que el costo de modelado y la generación automatizada de códigos es muy alto.

Capítulo 3 Desarrollo: Desarrollo Rápido de Aplicaciones

El Desarrollo Rápido de Aplicaciones (RAD) fue elegido como la metodología de desarrollo para *NetAttend* debido al poco tiempo que se le asignó para ejecutar el programa. Los comentarios de los usuarios son una parte importante en RAD y esto fue clave en la selección de la metodología, ya que los requisitos se modificaron en el proceso y se hicieron sugerencias durante todo el proceso de desarrollo.

3.1.1 Fase 1: Planificación de Requerimientos

Esta fase se llevó a cabo en este proyecto. Se realizó una reunión con los organizadores de la FEL para analizar las necesidades de la aplicación en función de los problemas encontrados en eventos pasados. El principal problema fue la falta de gestión en el área de asistencia. No había ningún control o registro seguro en el que se pudiera rastrear la asistencia. Todo se hizo manualmente, lo que causó confusión y fue tedioso. Otro problema fueron las estadísticas, y la asistencia se registró manualmente; había varias listas que se utilizarían y registrarían para encontrar a los asistentes que calificaban para una constancia. Luego se llegó a una conclusión de que la solución a todo esto sería una aplicación que registrara la asistencia en tiempo real. Para que esto sea posible, los asistentes necesitarían algún tipo de identificación. Se pensó que a cada asistente se le podría dar un código QR que contenía su correo electrónico. Este código QR luego se imprimirá para que el asistente lo muestre para escanear en cada entrada usando la misma aplicación. El proyecto final que se sugirió fue una aplicación que crea códigos QR y los escanea para registrar una asistencia en tiempo real.

3.1.2 Fase 2: Diseño de Usuario

Con los requisitos claramente establecidos, sentí que la forma más eficiente de entregar el producto final era a través del *framework* Ionic. Dado que la mayoría de los usuarios poseían dispositivos Android, ese fue el sistema operativo elegido. La primera página que se diseñó y codificó fue el menú principal, que solo contenía dos opciones: *Create QR Code* y *Take Attendance*. En el proceso de diseño, el formato hizo que pareciera que había pocas opciones y funciones en la aplicación. Allí fue donde decidí implementar varias funciones que facilitarían la experiencia del usuario y harían la aplicación más compleja. La aplicación debería tener una pantalla de inicio de sesión para poder

controlar el acceso a los datos y evitar la manipulación. Un botón de cerrar sesión fue agregado al menú principal. La otra función era permitir al usuario crear un evento para poder registrar a los asistentes. Presenté el diseño a los organizadores y expliqué las funcionalidades. Estaban bastante satisfechos con el progreso y me aseguraron que estaba avanzando muy bien en el proyecto. Hasta ese momento, la opción de escaneo era funcional, al abrir la cámara del dispositivo, lo que permitía al usuario escanear códigos QR y decodificarlos. También podría crear códigos QR que le permitían al usuario ingresar la información que deseaban convertir. Las funciones de inicio y cierre de sesión también funcionaban, esto fue posible con el uso de la autenticación de *Firestore*. Esto fue suficiente para presentar a los organizadores del evento para que puedan ofrecer sugerencias. La idea de la aplicación se amplió a medida que se hicieron sugerencias, como permitir que la aplicación envíe los códigos QR directamente a los asistentes una vez creados. Se hicieron otras adiciones menores a las páginas, como agregar campos a la página de creación de eventos.

3.1.3 Fase 3: Construcción Rápida

Después de la primera experiencia de usuario con el prototipo, me centré en la funcionalidad de la aplicación. Esto me llevó a integrar la creación de un evento con el escaneo y registro de un asistente a un evento. Esto fue exitoso con la integración de la base de datos de *Firestore*. Una vez que un evento fue creado y seleccionado y el código QR fue escaneado a ese evento, el asistente se registró en ese evento. Una vez que se ejecutaron estas funciones, me concentré en agregar el elemento de compartir. Se realizaron algunas investigaciones para obtener el conocimiento adecuado sobre cómo lograr esto. El *Social Share* es una característica del *framework* Ionic que facilitó esta tarea. Una vez instalado y programado, se implementó el botón compartir en redes sociales y se permitió al usuario compartir cualquier código QR generado en la aplicación. Después de esto, se llevó a cabo otra reunión para mostrar las actualizaciones en la aplicación. Esta vez, se invitó a todo el comité de FEL a formarse una opinión sobre la aplicación. Su principal preocupación eran las estadísticas, querían saber la cantidad de asistentes registrados para cada evento y la lista de esos asistentes por evento. También querían ser capaz de eliminar un asistente registrado. La reunión concluyó con esas sugerencias y una demostración de la aplicación y lo que contenía hasta ahora. Se realizaron las modificaciones adecuadas para satisfacer las necesidades del comité junto con algunas adiciones menores a la aplicación.

3.1.4 Fase 4: Corte

En esta fase del proyecto, la aplicación era completamente funcional y se desplegaron los archivos del proyecto. Dado que la aplicación era para uso interno, los APK de Android se generaron y no se implementaron en *Google Play Store*. La aplicación se instaló en un dispositivo Android para su prueba. Se realizaron pruebas adecuadas y se simularon diferentes escenarios. Se llevó a cabo otra reunión con los usuarios de la aplicación para instalarlos en sus respectivos dispositivos Android. Se presentaron instrucciones paso a paso para permitir a los usuarios hacer un seguimiento de las funciones de la aplicación.

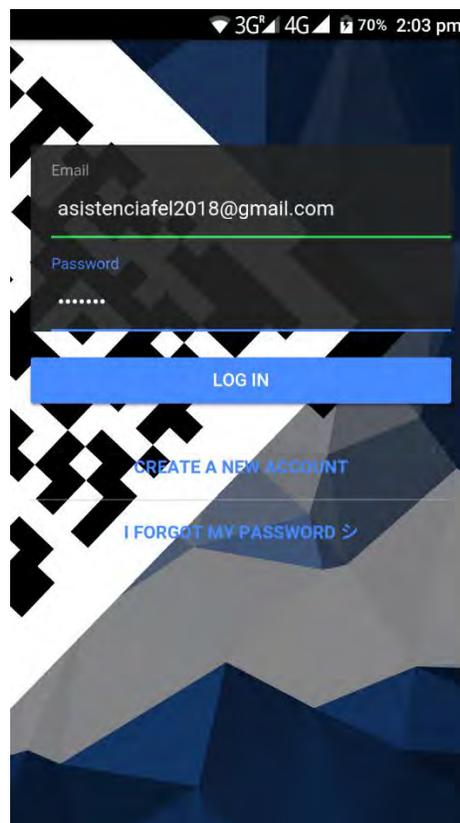


Ilustración 7: Página de Login de la aplicación

Esta es la entrada a toda la aplicación. Aquí el usuario puede crear una cuenta utilizando un correo electrónico. El usuario también puede recuperar el acceso a su cuenta si se pierde la contraseña. Se

debe ingresar un correo electrónico registrado y una contraseña para que el usuario ingrese a la aplicación.

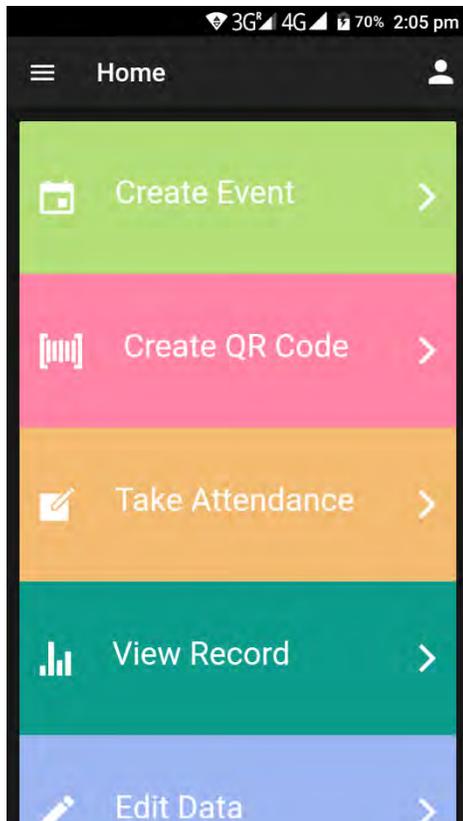


Ilustración 8: Pantalla del menú principal

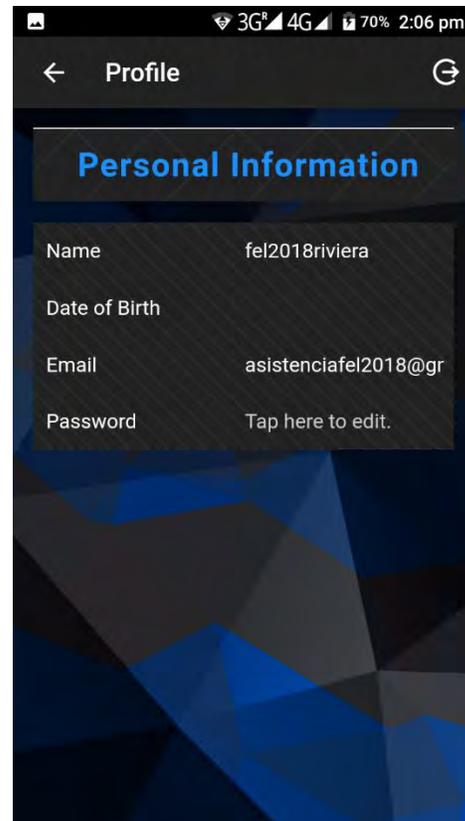


Ilustración 9: Pantalla de perfil

En la pantalla principal, aparece una lista de opciones. Estas son las tareas que puede realizar el usuario. También hay un icono de perfil en la esquina superior derecha de la pantalla que se puede seleccionar para ver y editar la información del usuario.

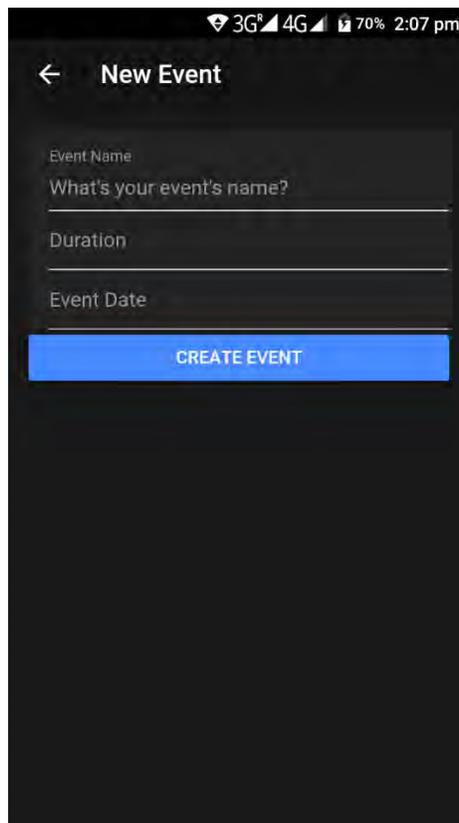
The image shows a mobile application screen titled "New Event". At the top, there is a status bar with icons for 3G, 4G, battery at 70%, and the time 2:07 pm. Below the title, there is a back arrow icon. The form consists of three input fields: "Event Name" with the placeholder text "What's your event's name?", "Duration", and "Event Date". At the bottom of the form is a prominent blue button labeled "CREATE EVENT".

Ilustración 10: Crear un nuevo evento

Para crear un nuevo evento, el usuario debe ingresar los campos requeridos. El nuevo evento se puede utilizar para registrar un asistente.



Ilustración 11: Creando un código QR usando un correo electrónico

Para registrar un asistente, se debe generar un código QR del correo electrónico del asistente. Este código QR se puede enviar al mismo correo electrónico para que el asistente lo reciba. Si el correo electrónico no se puede enviar por algún motivo, hay otras opciones para transmitir el código QR, como *Bluetooth*, *Messenger*, *WhatsApp*, etc. Esto será utilizado para registrar su asistencia.

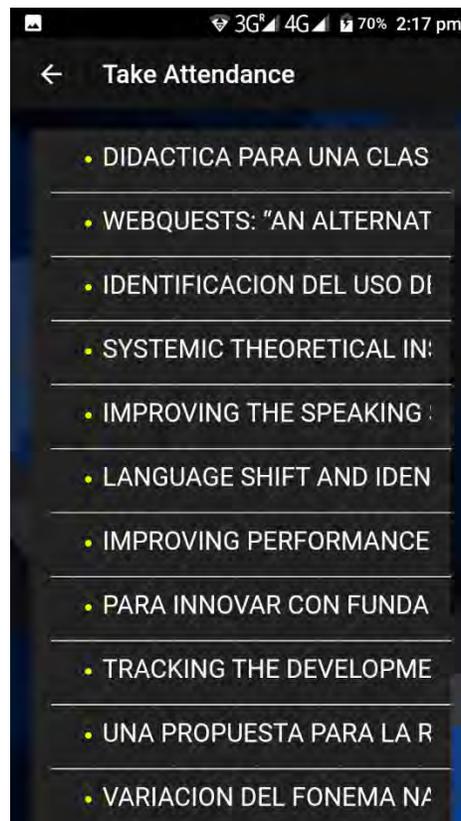


Ilustración 12: Lista de eventos para registrar asistencia

Para que el usuario registre una asistencia, el evento debe ser creado previamente. Ese evento se debe seleccionar para que la cámara se abra para escanear. El código QR se decodificará y se registrará en la base de datos de *Firebase* en tiempo real.



Ilustración 13: Lista de eventos creados

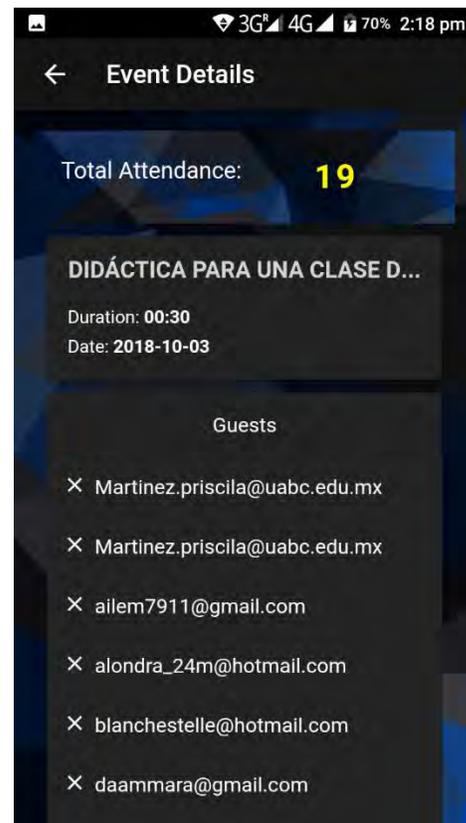


Ilustración 14: Registro detallado de un evento

Para ver el registro de asistencia, se selecciona cualquier evento para abrir la página de detalles del evento. En esta página, el usuario puede eliminar registros duplicados y realizar ajustes.

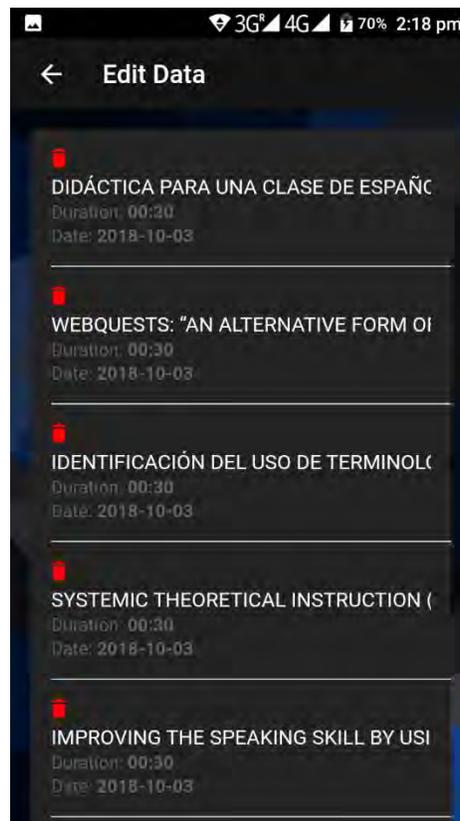


Ilustración 15: Lista de eventos para editar

Los eventos registrados se pueden eliminar de la base de datos haciendo clic en el icono de la papelera de reciclaje.

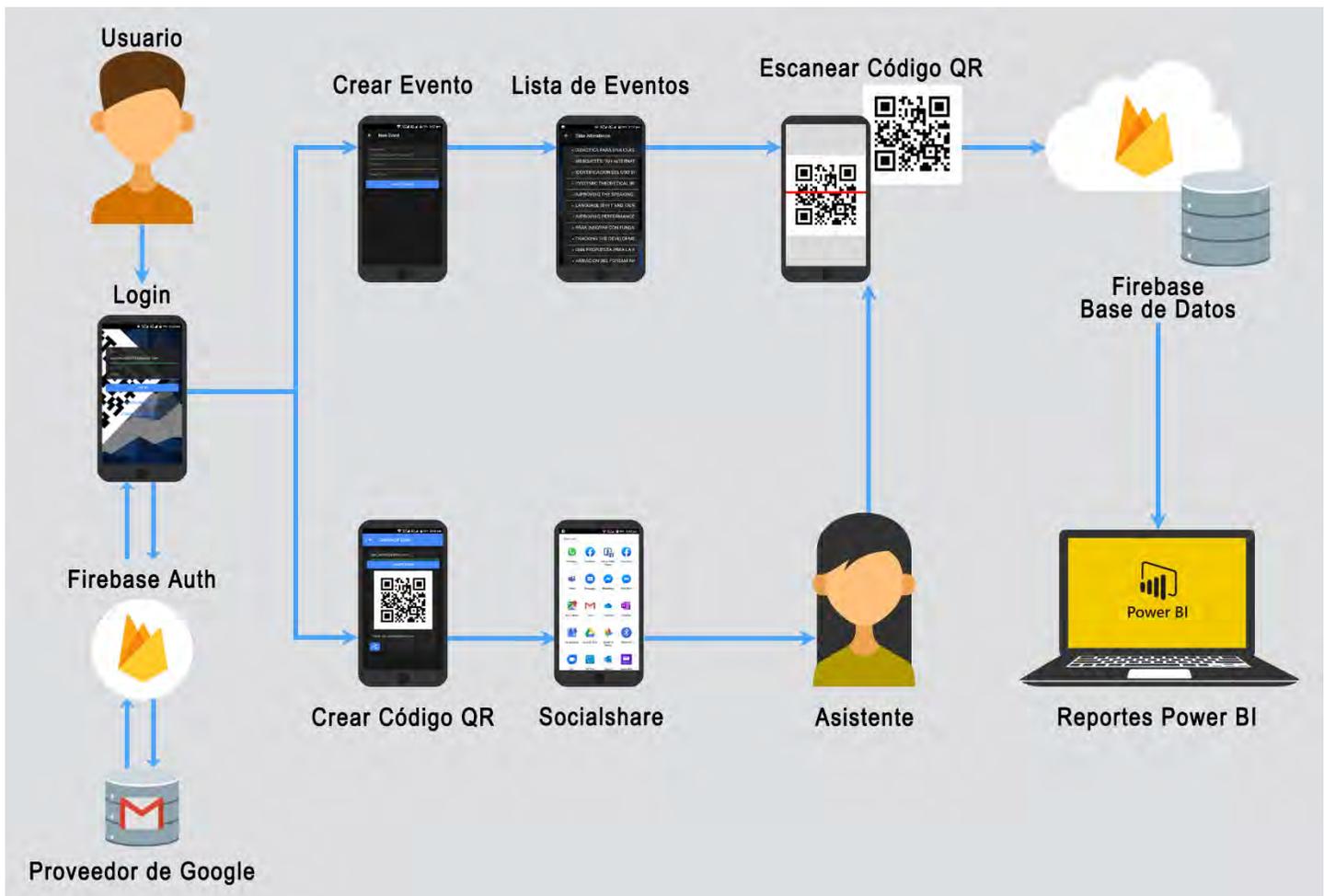


Ilustración 16: Proceso de Escaneo de Códigos QR en Net Attend. Elaboración propia.

La aplicación *NetAttend* funciona de tal manera que el proceso es fluido y todo sucede en tiempo real. Con la asistencia de la función *Firebase Authentication* proporcionada por el *Firebase Framework*, los usuarios simplemente crean una cuenta con sus correos electrónicos y una contraseña. Esto es importante ya que saber la identidad de un usuario permite que la aplicación guarde de forma segura los datos del usuario en la nube y proporcione la misma experiencia personalizada en todos los dispositivos del usuario.

Crear un código QR solo toma unos segundos. El primer paso es ir a "*Create QR Code*" e ingresar los datos apropiados, en este caso ingresamos el correo electrónico del asistente, por ejemplo **attende@gmail.com**. Se hace clic en el botón de "*Create Code*", y el código QR estará listo. Ahora solo se tiene que decidir cómo se quiere compartir el código QR para que el asistente lo tenga.

Todo esto depende de las otras aplicaciones y funciones instaladas en el dispositivo del usuario. Se puede enviar por correo electrónico, mensajería, Bluetooth o mediante cualquier otra aplicación.

Para escanear un código QR, debe seleccionar la opción “*Take assistance*” en el menú principal. Cuando haya ingresado, se inicia la aplicación y se mantiene la cámara del dispositivo sobre el código QR del asistente para escanearlo. Si el Código es legible, el correo electrónico codificada se escaneará automáticamente y se almacenará en la base de datos de *Firebase* en ese instante. En este punto, se puede acceder en línea a la base de datos de *Firebase* para ver los registros. En el portal de *Firebase*, el administrador puede descargar el archivo *JSON* y explotar todos los datos almacenados en la base de datos utilizando *PowerBI*. Se pueden generar reportes para dar un desglose visual de la actividad de la aplicación.

Capítulo 4 Resultados Y Conclusiones

4.1 Introducción

La idea de utilizar una aplicación móvil para llevar el registro de asistencias a eventos académicos surgió en el contexto del Foro de Estudios en Lenguas Internacional. Los organizadores (personal de la División de Ciencias Políticas y Humanidades de la Universidad de Quintana Roo, Campus Chetumal) requerían agilizar el control de asistencias, que les permitiera una mejor logística para la emisión de constancias de los asistentes.

4.2 Resultados

NetAttend se ha utilizado hasta este momento en 4 ocasiones:

Tabla 1 Información básica de los Eventos Acadpemicos que han utilizado NetAttend

Evento	Número de asistentes	Número de actividades
Foro de Estudios en Lenguas Internacional 2018 (FEL2018)	200	160
Foro de Estudios en Lenguas Internacional 2019 (FEL2019)	378	111
Jornada de Relaciones Internacionales 2019 (RRII2019)	130	14
UQROOMUN 2019	111	8

La tabla 1 muestra el nombre del evento que requirió el uso de la aplicación móvil *NetAttend*, el primero de ellos durante el año 2018 en el mes de octubre llevado a cabo en el municipio de Solidaridad, Quintana Roo, contó con una participación de 200 asistentes los cuales se distribuyeron durante 3 días en 160 actividades (inauguración, clausura, ponencias, exposición de carteles, redes de colaboración, etc.), ésta fue la primera vez que se realizaba el control de registro mediante el uso de un dispositivo móvil y la tecnología de códigos QR.

Al finalizar el evento, el comité organizador envió una encuesta de satisfacción la cual agregó preguntas acerca de la opinión de los asistentes con respecto al uso de la aplicación *NetAttend*.

Los resultados de dicha encuesta arrojaron que un 85% de usuarios opinaron de manera **favorable** (calificaciones de 4 y 5 en una escala del 1 al 5) a esta forma de llevar el control de asistencias. Un 11% emitieron recomendaciones sobre el registro de control de asistencia y un 3% no estuvo de acuerdo en el uso de la aplicación.

En octubre del 2019, la aplicación móvil *NetAttend* fue utilizado para la Jornada de Relaciones Internacionales 2019 (RRII2019). Se realizaron un total de 14 actividades en 2 días. Esta jornada obtuvo un total de 130 asistencias.

En el FEL2019 el número de asistentes creció de manera muy importante, con 378 asistentes registrados y distribuidos en 111 eventos, muchos de ellos que ya habían asistido al FEL2018 se encontraron bastante familiarizados con el uso de *NetAttend* para el registro de asistencias.

NetAttend fue utilizado exitosamente para la cuarta simulación de la Organización de las Naciones Unidas (ONU) de la Universidad de Quintana Roo que se llevó a cabo los días 30 y 31 de octubre del 2019. El comité organizador del IV Modelo de Naciones Unidas UQROOMUN organizó un total de 8 actividades los cuales el número total de asistencias fueron 111.

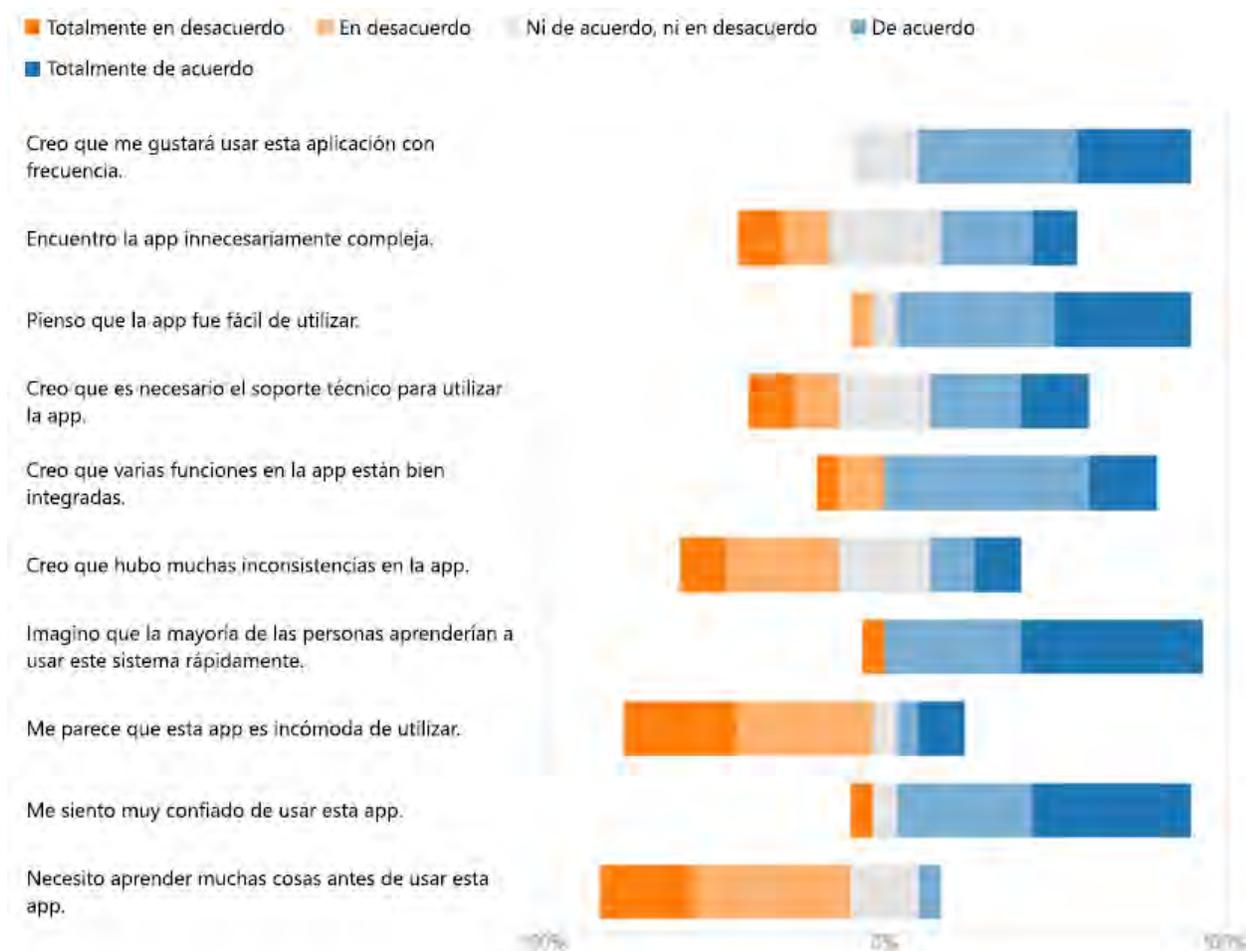


Ilustración 17 Resultados de la encuesta de usabilidad de la aplicación NetAttend

Se decidió realizar una encuesta de usabilidad de la aplicación a los operadores de los dispositivos móviles. Como puede apreciarse en la ilustración 17, un total de 15 personas respondieron a dicha encuesta. Conforme a los resultados de la encuesta, todos tuvieron facilidad en manejar la aplicación que hasta están dispuestos a usar *NetAttend* con frecuencia. Un 60% de los que respondieron están en desacuerdo en que la aplicación es complicada para usar (resultado favorable). Las funcionalidades fueron vistas como bien integradas, como se había planeado. Para complementar esto, todos estuvieron de acuerdo en que la aplicación era muy fácil de aprender. Un porcentaje mejor de las personas sienten la necesidad para soporte técnico para usar la aplicación. El 73% de las respuestas muestran que hubo inconsistencias con la aplicación. Todas las respuestas mostraron confianza utilizando la aplicación. 14 de las 15 personas sienten que la aplicación requiere conocimientos mínimos para manejar *NetAttend*.

4.3 Conclusiones

La aplicación *NetAttend* fue exitosamente utilizado en 4 ocasiones donde los usuarios estaban contentos con las funcionalidades y el potencial de la aplicación móvil. *NetAttend* cumplió con los requisitos de los comités de cada evento. *NetAttend* facilitó y automatizó el proceso de la toma de asistencias en una grande escala.

NetAttend fue creado para facilitar el proceso de toma de asistencia y seguimiento de registros. Se implementaron varias características para lograr esto. *NetAttend* ofrece la opción de crear un código QR personalizado. Simplemente ingresando el texto que desea convertir y con el clic de un botón, se genera el código QR. Para mejorar aún más la experiencia del usuario, *NetAttend* tiene la capacidad de compartir el código QR creado recientemente utilizando cualquier aplicación de redes sociales que esté instalada en el mismo dispositivo móvil. *NetAttend* también proporciona la función para ver y eliminar registros. Sin embargo, hay suficiente espacio para la mejora. Algunas sugerencias fueron recomendadas por los usuarios. De estas, una búsqueda de eventos para facilitar el proceso de registrar asistencias fue incluido. Esto implica que, al momento de escanear códigos de asistentes, el usuario tendría una manera de acceder al evento deseado con más facilidad y en menos tiempo. Esto serviría para acelerar el proceso del registro de asistencias. El control de usuario fue un tema de sugerencia de igual forma, esto implica que se implemente roles de usuarios administrativos y usuarios general. El usuario administrativo tendría acceso a todas las funciones de *NetAttend*, en cambio el usuario general tendría limitaciones al editar y crear eventos y registros. Otra idea fue que los reportes se puedan generar desde la aplicación. Esto reemplazaría la integración de PowerBI en total y minimizaría el trabajo tal cual.

Aparte de las sugerencias para la aplicación móvil, se presentaron algunos obstáculos. Una de las limitaciones que tuvo la aplicación móvil fueron la conexión de red disponible. El FEL 2018 fue organizado en el Hotel Iberostar en la Riviera Maya. Los eventos se llevaron a cabo en un centro de convenciones que cuenta con la misma red distribuida en toda el área de enfrente que incluye un restaurante, cuartos de hotel, cocina, y una tienda de regalos. El uso de la red causó una saturación que debilitó el rendimiento de *NetAttend*. Los registros de asistencias no se arrojaban en tiempo real.

Esto causó confusión entre los usuarios de la aplicación, quienes insistían con el escaneo de códigos QR hasta que se reflejaba en la lista de registros. Cuando se establecía la red, la base de datos se actualizaba causando que se reflejen registros duplicadas. Este error se solucionaba en Power BI al momento de generar los reportes.

NetAttend ofreció varias funciones para facilitar y acelerar el proceso de registro de asistencias. Antes de la creación e implementación de *NetAttend*, todo era hecho manualmente. *NetAttend* sirvió para obtener datos específicos para poder generar reportes de cada evento. La base de datos fue explotada para generar constancias conforme a los requisitos. En el mundo de hoy, donde la tecnología se hace cargo y revoluciona la forma en que se hacen las cosas, *NetAttend* ha demostrado ser y seguirá siendo beneficioso en la toma de asistencia en cualquier evento que se requiera.

Bibliografía

(s.f.). Obtenido de JavaTPoint: <https://www.javatpoint.com/android-software-stack>

(s.f.). Obtenido de Mas adelante: <https://www.masadelante.com/faqs/sistema-operativo>

(s.f.). Recuperado el 2019, de Google Support: <https://support.google.com/googleplay/android-developer/answer/112622?hl=en>

Agile 101: What Is Agile Methodology? (30 de abril de 2019). Obtenido de CollabNet: <https://resources.collab.net/agile-101/agile-methodologies>

Agile Methods. (30 de abril de 2019). Obtenido de What is six sigma: <https://www.whatissixsigma.net/agile-methods/>

Cassavoy, L. (23 de Diciembre de 2018). *What Is a BlackBerry?* Obtenido de Lifewire: <https://www.lifewire.com/what-is-a-blackberry-579602>

Chandana. (4 de octubre de 2019). *Scrum Project Management: Pros and Cons.* Obtenido de Simplilearn: <https://www.simplilearn.com/Scrum-project-management-article>

Chapter 1: All About Ionic. (s.f.). Obtenido de Ionic Framework: <https://ionicframework.com/docs/v1/guide/preface.html>

Esplin, C. (24 de 10 de 2016). *What is Firebase?* Obtenido de How to Firebase: <https://howtoFirebase.com/what-is-Firebase-fcb8614ba442>

Focus, A. (13 de noviembre de 2018). *METODOLOGÍA AGILE – QUE ES y QUE BENEFICIOS tiene.* Recuperado el 2019, de Noticias sobre tecnología: <https://www.digitalrubik.com/conocimientos-tecnicos/metodologia-agile-que-es-beneficios/>

Guru99. (2019). *What is AngularJS? Architecture & Features.* Obtenido de Guru99: <https://www.guru99.com/angularjs-introduction.html>

iOS Architecture. (s.f.). Recuperado el 05 de 2019, de intellipaat: <https://intellipaat.com/tutorial/ios-tutorial/ios-architecture/>

Netgate. (12 de 10 de 2018). *Openvpn.* Obtenido de Openvpn: <https://openvpn.net/>

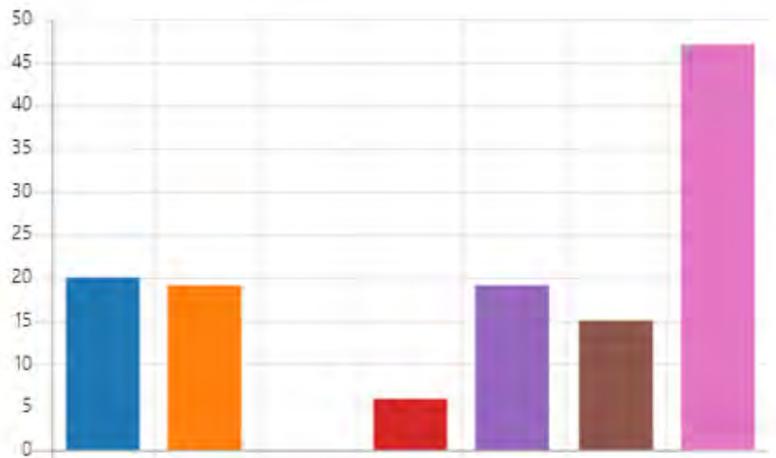
- Pandu, W. (s.f.). *What Is Agile Methodology?* Recuperado el 2019, de wcfpandu:
<https://sites.google.com/site/wcfpandu/what-is-agile-methodology>
- Panos, M. (27 de 04 de 2017). *Ruby on Rails - Architecture Overview For Beginners*. Recuperado el 2019, de Tech Career Booster: <https://www.techcareerbooster.com/blog/ruby-on-Rails-architecture-overview-for-beginners>
- Powell-Morse, A. (8 de diciembre de 16). *WaterFall Model: What Is It and When Should You Use It?* Obtenido de Airbrake : <https://airbrake.io/blog/sdlc/WaterFall-model>
- Ripkens, B. (28 de 11 de 14). *Ionic: An AngularJS based framework on the rise*. Obtenido de Codecentric Blog: <https://blog.codecentric.de/en/2014/11/ionic-angularjs-framework-on-the-rise/>
- Ros, I. (25 de Abril de 2014). *Google compró Android por 50 millones de dólares, Samsung lo rechazó*. Obtenido de Muy Computer: <https://www.muycomputer.com/2014/04/25/google-compro-android-samsung-lo-rechazo/>
- The iPhone OS Cocoa Touch Layer*. (s.f.). Obtenido de Techotopia:
https://www.techotopia.com/index.php/The_iPhone_OS_Cocoa_Touch_Layer
- The iPhone OS Core Services Layer*. (s.f.). Obtenido de Techotopia:
https://www.techotopia.com/index.php/The_iPhone_OS_Core_Services_Layer
- WaterFall Methodology: Advantages, Disadvantages And When to Use It?* (15 de marzo de 2018). Obtenido de Lvivity: <https://lvivity.com/WaterFall-model>
- What Is SQLite*. (s.f.). Recuperado el 2019, de SQLite Tutorial: <http://www.SQLitetutorial.net/what-is-SQLite/>
- Wiegert, C. (23 de 1 de 2019). *Android, iOS, and the Camera - Oh My!* Obtenido de Docs: Ionic Framework: <https://ionicframework.com/docs/developer-resources/guides/first-app-v3/ios-android-camera>

ANEXOS

1. ¿A través de cuáles medios se enteró del XIV Foro de Estudios en Lenguas 2018?

[Más detalles](#)

● Correo Electrónico	20
● Página de internet	19
● Revista	0
● Cartel	6
● Página Web del FEL	19
● Redes Sociales	15
● Otros	47



2. ¿Cuál es su consideración acerca de la calidad de las ponencias magistrales, ponencias, mesas redondas y talleres?

[Más detalles](#)

97

Respuestas



Clasificación promedio 4.27

3. ¿Cuál es su opinión sobre la sesión de pósters respecto a la dinámica de presentación ?

[Más detalles](#)

93

Respuestas



Clasificación promedio 4.12

4. ¿Qué tan adecuadas considera las instalaciones de la sede del XIV FEL 2018?

[Más detalles](#)

97

Respuestas



Clasificación promedio 4.66

5. ¿Cuál es su consideración acerca de la calidad de la organización del evento? Considere Registro, Moderadores, Constancias, Facturación, Equipo Técnico

[Más detalles](#)

97

Respuestas



Clasificación promedio 4.38

6. Deje un comentario acerca de la calificación de la pregunta anterior

Problemas con puntualidad y varios eventos cancelados

La iluminación no permitía visualizar de manera adecuada las diapositivas, al igual de los equipos y el moderador de las diapositivas obstruía la visión de las mismas.

Me parece que requieren mayor capacitación sobre los procesos a seguir

El evento estuvo muy bien organizado sin embargo hubo muchas ponencias canceladas

El registro un poco tardado pero lo demás muy bien

Las personas de staff fueron muy amables

Excelente servicio del Equipo Técnico, muy atentos y serviciales. El registro de Inscripción no estaba bien organizado y la mayoría desconocía la información necesaria a proporcionar. Considero que todos los que se van a encargar a atender deberían de recibir una capacitación previa al congreso para que conozcan toda la información necesaria. Los ponentes en su mayoría están muy bien preparados y fueron ponencias muy interesantes. Donde falta mucha atención es en la forma de registrar la asistencia. Que nosotros como participantes tengamos una forma para poder tener acceso y estar bien seguros de que se hizo apropiadamente el registro de las ponencias a las que asistimos ya que al ir a verificar para comparar el número de registros que tenían ustedes con el número de asistencias que tuvieron fue diferente y eso no es nada bien. Mientras uno tenía una cantidad de asistencias a ustedes les aparecía una cantidad mucho menor. De esa forma no es nada confiable ni seguro el sistema que están usando. Somos dos Maestras las que asistimos y ambas al ir a preguntar si se habían registrado correctamente nuestras asistencias nos encontramos con la muy desagradable sorpresa de que ustedes tenían menor cantidad de horas registradas. Varias veces comentaron cuando se hacía el registro de asistencia de que el WiFi no estaba funcionando. Eso es algo de suma importancia de atender. Gracias y Muchas Felicidades por todo lo demás

Excelente la facturación que enviaron a mi correo. Nunca pude hacer la reservación del hotel a través de la página proporcionada que no sirvió para nada por lo cual fue muy complicado reservar en línea. El precio del hotel fue muy caro. Por otra parte, me encantó que este año fuera en FEL en la playa. El link de la página de FEL 2018 no se puede cargar con facilidad y se necesita mucho tiempo para hacerlo.

Habían proyectores que fallaban o incluso las luces, se apagaban en medio de una exposición. Además, en algunos momentos se tuvo un problema con los registros.

Considero que tanto el lugar como el equipo de trabajo fue muy eficiente.

Muy buen trato por parte de todos

Gracias por su organización y su invitación.

Las constancias electrónicas no me resultaron tan convenientes. El escaneo de códigos me pareció un poco exagerado pero así se asegura la asistencia a las ponencias. La sede a pesar de ser lujosa, no estaba preparada para este evento, en la UQROO se sintió más el calor de compañerismo al convivir directamente con los estudiantes locales.

Todo en tiempo y forma, me gusto el uso de los celulares para pasar asistencia

Mayor organización probablemente

Excelente organización y sede

He asistido a eventos en el pasado, del mismo foro, los cuales han sido mejores que este.

Hubieron algunos problemas con el escaneo del código para contar las asistencias.

Incrementar apoyo técnico en las salas

No hubo una presentación inicial. Nadie instruía sobre a donde ir o cómo. Gracias a que tenían playeras se pudo identificar a quien preguntar.

Todos fueron muy atentos y se mostraron competentes en cada una de sus funciones

Desgraciadamente la facturación no tuene que ver directamente con FEL, pero considero que, al conocer la situación de las universidades públicas, deberían de informar a los hoteles sede de dicha situación. Repito, esto no es relacionado al evento en sí, pero siempre ayuda que los anfitriones informen a los encargados de la facturación. Por todo lo demás, constancias, soyo técnico, etc, excelente!!!

Se sintió mucho calor el primer día pues no había aire acondicionado.

La cuestión logística me ha parecido impecable. La facturación es un asunto pendiente que espero que sea tan buena como el resto del evento.

Todo muy organizado, las instalaciones excelentes, el registro un poco lento, lo hice en otro momento, tal vez hacer el registro como la entrada a las ponencias para no abarcar mucho tiempo. Mi factura y constancia de ponente muy a tiempo. Respecto a la sesión de posters, un poco triste, pues aunque se había mencionado premio al primer lugar, en la clausura se mencionaron tres lugares, considero que también podría darse un reconocimiento a esos 3 lugares, sobre todo por la satisfacción de llegar nuestros Centros de trabajo y mostrar el logro obtenido. en general muy bien!

7. ¿Cómo calificaría la actitud de los organizadores y personal de apoyo ?

[Más detalles](#)

93

Respuestas



Clasificación promedio 4.71

8. ¿Qué tan adecuado considera el control de asistencia utilizando códigos QR?

[Más detalles](#)

94

Respuestas



Clasificación promedio 4.41

9. Por favor, deje un comentario con respecto al control de asistencia

82 Respuestas

75	anonymous	Estuvo muy bien, aunque hubiera sido preferible poner el código QR en las tarjetas de ponente / asistente
76	anonymous	Excelente
77	anonymous	rápido y práctico
78	anonymous	Estuvo bien organizado
79	anonymous	La lectura de los códigos QR fue lento lo cual originó un poco de aglomeración a la entrada/salida.
80	anonymous	considero que es excelente porque agiliza el proceso y porque los asistentes pueden preguntar al final del día cuántas ponencias escucharon
81	anonymous	Podría mejorar si el código se pone en el mismo gafete.
82	anonymous	todo muy organizado

10. En general, ¿cuál es su nivel de satisfacción con respecto al XIV FEL 2018?

[Más detalles](#)

96

Respuestas



Clasificación promedio 4.65

11. ¿Recomendaría nuestro evento a amigos, compañeros y colegas?

[Más detalles](#)

96

Respuestas



Clasificación promedio 4.86