



Reporte Final de Estadía Antonio Adonai Molina Pérez

Desarrollo de módulos de funcionalidad para
aplicación móvil utilizando tecnologías de
desarrollo híbrido para la empresa BSD

Enterprise S.A de C.V.



**Desarrollo de módulos de funcionalidad para aplicación móvil
utilizando tecnologías de desarrollo híbrido para la empresa BSD**

Enterprise S.A de C.V.

Presenta:

Antonio Adonai Molina Pérez

Asesor Industrial:

Ing. Cristian Mercado Gallardo

Asesor Académico:

Lic. Rolando Rodríguez Vázquez

ÍNDICE

1	CAPÍTULO I: DATOS GENERALES.....	1
1.1	HISTORIA.....	1
1.2	MISIÓN	1
1.3	VISIÓN.....	1
1.4	Valores.....	2
1.5	Principios	2
1.6	Filosofía	2
2	Capítulo II: Análisis del problema y propuesta de la solución	3
2.1	Descripción de la problemática.....	3
2.2	Justificación.....	3
2.3	Objetivo General	4
2.4	Objetivos específicos	4
2.5	Alternativas de solución	5
2.5.1	React Native	5
2.5.2	NativeScript	6
2.6	Alcances	6
2.7	Limitaciones	7
2.8	Solución propuesta.....	7
3	Capitulo III: Fundamentación Teórica	10
3.1	Antecedentes de la Investigación	10
3.2	Metodología empleada para el desarrollo	11
3.2.1	Flujo de trabajo	13

3.2.2	Roles	13
3.2.3	Artefactos	14
3.3	Estrategias de solución	15
3.3.1	Framework Ionic.....	15
3.3.2	AngularJS	16
3.3.3	HTML	16
3.3.4	JavaScript.....	17
3.3.5	Android.....	17
3.3.6	iOS	17
3.3.7	Apache Cordova	18
4	Capítulo IV: Desarrollo de Módulos de Funcionalidad para Aplicación Móvi	19
5	Conclusiones	21
6	Recomendaciones	21
7	Referencias bibliográficas	23
8	Anexos	25
8.1	Cronograma	25

1 CAPÍTULO I: DATOS GENERALES

1.1 HISTORIA

BSD Enterprise, Business Solutions Developers Enterprise es una Empresa de servicios en Tecnología de Información con una propuesta creativa y dinámica que aporta soluciones para el crecimiento de su negocio, a través de la calidad, consultoría, análisis y desarrollo de sistemas. Se conforma por un experimentado y emprendedor grupo de talento empresarial, capaz de transformar los problemas de su negocio en grandes oportunidades de desarrollo y crecimiento.

1.2 MISIÓN

Nuestra misión se sintetiza en una frase: UNA VISIÓN INTELIGENTE.

Esta es la base que orienta el talento, sensibilidad y esfuerzo de nuestro equipo de trabajo para analizar y estudiar de forma constante los grandes retos del mercado, las constantes innovaciones y avances tecnológicos y adaptarlos para fortalecer las soluciones que ofrecemos a nuestros clientes, cuidando actuar con inteligencia, enfocados a solucionar las necesidades que inhiben su crecimiento.

1.3 VISIÓN

Ser la principal empresa de servicios de Tecnología de Información en México, con una amplia red de servicios en Europa, América, Asia y Oceanía.

1.4 VALORES

- **Disciplina:** La conforma una constancia en la investigación, actualización, reinversión, creación e innovación empresarial.
- **Ética:** Se basa en tres valores fundamentales: Honestidad, discreción y exclusividad.

1.5 PRINCIPIOS

- Lealtad y respeto.
- Trabajo en equipo y por objetivos.
- Capacitación constante.
- Innovación.
- Eficiencia en el uso de recursos y del tiempo.
- Transparencia en el trato a nuestros clientes.

1.6 FILOSOFÍA

Basada en una ideología en donde se centraliza nuestra Visión Inteligente, enfocada a la sensibilidad, disciplina, ética, valores, principios y esfuerzo de un equipo de trabajo con la finalidad de fortalecer las áreas de oportunidad, y lograr dar una respuesta apropiada a las necesidades de nuestros clientes.

2 CAPÍTULO II: ANÁLISIS DEL PROBLEMA Y PROPUESTA DE LA SOLUCIÓN

2.1 DESCRIPCIÓN DE LA PROBLEMÁTICA

La empresa se encuentra desarrollando actualmente una aplicación móvil que integra múltiples módulos de funcionalidad desarrollados en tecnologías híbridas y nativas que trabajan en conjunción. Esto se debe a que el equipo de desarrollo es multidisciplinario, los esfuerzos se pueden repartir de acuerdo a los recursos humanos con que cuenta la empresa y a los requisitos del producto. Para este proyecto se enfocarán los esfuerzos en la colaboración para el desarrollo de módulos realizados con tecnologías de desarrollo híbrido. Junto con el desarrollo se realizarán pruebas de humo para verificar de manera rápida que las funcionalidades programadas cumplan su cometido. De igual manera se llevará a cabo un análisis de los requisitos que permita orientar el desarrollo, tanto de la funcionalidad como de las interfaces de usuario y flujo de interacción.

El desarrollo se orienta con una metodología ágil utilizando iteraciones (modelo espiral), en cada iteración se analizan los requisitos o se agregan más y se desarrolla la funcionalidad en el transcurso de la iteración. Se hacen revisiones del producto a intervalos de tiempo para señalar áreas de mejora o correcciones. En ocasiones durante las revisiones se añaden también a la lista de requisitos nuevas funcionalidades.

2.2 JUSTIFICACIÓN

En cuanto al ámbito técnico, la combinación entre distintas tecnologías de desarrollo no es algo innovador, de hecho, muchas empresas como Instagram lo hacen en sus aplicaciones principales de manera que puedan aprovecharse las mejores características de cada plataforma y ahorrar en la mayor medida los tiempos de desarrollo. El desarrollo en tecnologías para aplicaciones híbridas es inherentemente más veloz que los desarrollos en tecnologías nativas en la gran mayoría de las ocasiones, casi siempre sin importar la

complejidad de los requisitos y con un gran plus que es la capacidad de escribir una sola fuente de código que podrá ejecutarse en dos plataformas distintas.

Para la aplicación se requiere tener ciertos módulos implementados en tecnología híbrida, normalmente aquellos que no demandan de mucho rendimiento. A corto plazo se disminuirán los tiempos de desarrollo, lo que además permitirá realizar mejoras a la funcionalidad en menor tiempo. Sin embargo, a largo plazo se pretende sustituir la tecnología híbrida por nativa para brindar una mejor experiencia de usuario y rendimiento, esto se realizará cuando el desarrollo se encuentre en una etapa mucho más madura.

La empresa se beneficiará significativamente con la disminución en los tiempos de entrega lo que también implica un ahorro de costes, permite implementar pruebas más prontamente para encontrar errores y mejorar la calidad del producto en un lapso de tiempo relativamente corto.

2.3 OBJETIVO GENERAL

Desarrollar múltiples módulos de funcionalidad mediante la utilización de tecnologías de desarrollo híbrido para una aplicación móvil nativa para la empresa BSD Enterprise utilizando técnicas de desarrollo ágil.

2.4 OBJETIVOS ESPECÍFICOS

- Realizar un análisis de los requisitos de manera ágil y breve en cada iteración para determinar la estructura de los módulos a desarrollar.
- Desarrollar módulos web con tecnología híbrida para la aplicación nativa acorde a los requerimientos.

- Realizar pruebas de humo para verificar el correcto funcionamiento de los módulos híbridos.
- Integrar y asegurar la correcta compatibilidad de los módulos híbridos con la aplicación nativa.

2.5 ALTERNATIVAS DE SOLUCIÓN

A continuación, se describen dos tecnologías alternativas basadas en tecnologías web (lenguaje JavaScript) para la realización de módulos que pueden integrarse con aplicaciones nativas para la conformación de una aplicación móvil:

2.5.1 REACT NATIVE

Se trata de un framework para crear interfaces de aplicaciones móviles nativas basado en la librería web React para crear interfaces. A diferencia de las tecnologías híbridas React Native no crea interfaces a través de un webView puesto que las vistas no son definidas utilizando HTML. Las interfaces son creadas utilizando un lenguaje similar a XML, pero con capacidades extendidas, como mezclar código JavaScript directo en la vista. React Native crea un puente de comunicación entre el api nativa del dispositivo móvil y el motor que ejecuta JavaScript haciendo posible la interacción sin la utilización de tecnologías como Apache Cordova con el SDK de Android o iOS.

React Native fue creado y es mantenido por Facebook. El hecho de contar con el respaldo de uno de los gigantes de la tecnología de hoy día les asegura a los desarrolladores que pueden confiar en la estabilidad del proyecto y que contará con un buen soporte.

React Native es utilizado por numerosas compañías de tecnología como Facebook, Instagram, Walmart, Airbnb entre muchas más, por lo que se puede suponer que es una tecnología apta para el desarrollo de aplicaciones móviles.

2.5.2 NATIVESCRIPT

Es una tecnología desarrollada por Telerik que funciona de forma similar a React Native en cuanto a que no emplea webViews para sus interfaces, crea vistas nativas a partir de definiciones escritas en XML que son conectadas con lógica escrita en JavaScript, aunque a diferencia del proyecto de Facebook, NativeScript transforma la misma definición de la interfaz a su equivalente en nativo. En React Native algunas interfaces deben ser escritas de manera personalizada para cada plataforma debido a la diferencia que existe entre plataformas de sus componentes nativos.

NativeScript es una tecnología con menos popularidad que React Native y también menor madurez, no la utiliza el mismo número de compañías como a React Native por lo que no es tan buena opción.

2.6 ALCANCES

- El producto comprende el cumplimiento de todos los requisitos que hayan sido aprobados previamente por el líder de proyecto y se asegura su correcta funcionalidad.
- Las interfaces de los módulos estarán acorde al diseño previamente estipulado por el líder de proyecto.

- El producto estará garantizado de tener un correcto funcionamiento e interoperabilidad con el ambiente de integración nativo en cualquiera de las plataformas que se haya definido utilizar.
- El proyecto utilizará las tecnologías que haya definido el líder de proyecto y el código fuente deberá apearse a los estándares que utilice el equipo de desarrollo.

2.7 LIMITACIONES

- El proyecto no comprende construir una aplicación standalone que se pueda instalar como un apk de Android, si no generar módulos que serán embebidos en una aplicación nativa ya existente.
- El producto no pretende igualar un producto equivalente realizado con tecnología nativa en ningún aspecto. Se pretende simplemente acelerar el proceso de desarrollo con tecnologías híbridas.

2.8 SOLUCIÓN PROPUESTA

Se optó por emplear la tecnología Ionic que emplea webViews para generar las interfaces utilizando HTML. Aunque los webViews no tienen el mejor rendimiento especialmente en dispositivos viejos, tiene la ventaja de utilizar HTML que es una tecnología con la que ya están familiarizados los desarrolladores del proyecto, al ser ésta ampliamente empleada en los desarrollos web que se han realizado anteriormente.

Los módulos a desarrollar en la tecnología híbrida no requieren de un muy alto rendimiento por lo que será suficiente utilizar webViews para las interfaces. El equipo de desarrollo ya cuenta con experiencia en la tecnología Ionic versión 1 por lo que el ritmo de desarrollo es

bastante rápido, que es lo que inicialmente se buscaba al evitar la implementación de tecnologías nativas.

Para la forma de trabajo el líder de proyecto especifica en forma de historias de usuario los requerimientos a la persona o personas que se van a encargar de desarrollar el módulo. Para esto el diseño gráfico de la interfaz ya está definido por lo que el desarrollador se enfoca principalmente en desarrollar la funcionalidad y los detalles estéticos se emplean al final.

Una vez se libere el módulo o alguna funcionalidad muy importante del mismo pasa al área de pruebas en otro departamento y en caso de encontrar fallas se le notifica al desarrollador encargado con detalles para que pueda replicar el defecto de ser necesario. Semanalmente el desarrollador reporta su progreso al líder del proyecto y le informa sobre aspectos que estén bloqueando su trabajo.

El entregable principal que debe proporcionar el desarrollador es el código fuente del módulo aunque no se requiere de una entrega formal, ya que el código se encuentra en los repositorios que el líder de proyecto especifica y hacia los cuales el desarrollador deberá estar subiendo entregas frecuentemente. Para el tipo de proyecto en cuestión por lo general no se requiere documentación aunque si el líder de proyecto lo requiere el desarrollador debe conformar un documento o serie de documentos que recopilen la información necesaria para documentar el proyecto, como pueden ser manual técnico, manual de usuario, entre otros.

Por lo general se realizan revisiones de nivel de alta dirección a ciertos intervalos de tiempo para verificar que la aplicación se apegue a la estrategia de negocio que se haya definido. En dichas revisiones participa el desarrollador principal del módulo evaluado y hace una presentación de las características del mismo y el flujo de interacción. Durante la revisión

se le hace saber al desarrollador acerca de detalles y mejoras que se puedan aplicar y en ocasiones se agregan requisitos para el módulo.

3 CAPITULO III: FUNDAMENTACIÓN TEÓRICA

3.1 ANTECEDENTES DE LA INVESTIGACIÓN

Las plataformas híbridas datan de hace casi diez años con la aparición de PhoneGap en el año 2008 (Wikipedia, 2016). En esa época los webViews de los dispositivos móviles se encontraban en una etapa demasiado infantil como para poder tomar en serio el desarrollo híbrido, sin embargo al pasar de los años esta situación ha ido cambiando, los proveedores de las plataformas móviles más utilizadas hoy día como son Android y iOS han mejorado en gran medida los webViews por lo que ahora con las versiones más recientes de estos sistemas operativos se puede obtener una experiencia de desarrollo y de uso mucho mejor que antes. Muchas aplicaciones han sido desarrolladas con esto en mente por lo que el desarrollo híbrido ha evolucionado al punto de volverse viable la creación de aplicaciones híbridas.

Las tecnologías de desarrollo híbrido permiten a los desarrolladores web crear experiencias móviles con el conocimiento y las tecnologías que ya conocen, con una curva de aprendizaje mínima y tiempos de desarrollo mucho más cortos, puesto que es un trabajo de codificar una vez y ejecutar donde sea, sin embargo, las aplicaciones híbridas tienen un rendimiento mucho menor que las aplicaciones nativas y la experiencia de usuario que proveen no es tan fluida como la que puede proporcionar una aplicación nativa. Esto quiere decir que si lo que se busca es desarrollar una aplicación con una gran carga de procesamiento y renderizado un enfoque híbrido no es la mejor opción. No obstante a pesar de las limitaciones mencionadas, se puede combinar lo mejor de ambos mundos, empresas como Instagram y Uber (Ormandy, 2017) crearon sus aplicaciones principales a partir de un desarrollo nativo que integra en algunos módulos webViews a través de los cuales se carga una aplicación híbrida. No hay duda que ambas aplicaciones cumplen su propósito con un alto rendimiento, de otro modo no tendrían la cantidad de usuarios con que cuentan

actualmente, por lo que se puede suponer que el enfoque que utilizan es viable para el desarrollo de aplicaciones móviles empresariales.

3.2 METODOLOGÍA EMPLEADA PARA EL DESARROLLO

Para el desarrollo de la solución del presente caso de estudio se necesita un marco de trabajo que guíe los esfuerzos del proyecto de manera óptima, de forma que se eviten esfuerzos innecesarios que comúnmente ocurren cuando no se cuenta con una forma de trabajo y una estructura definidas previamente. El desarrollo de software es una tarea compleja y si no se gestionan correctamente los procesos que conlleva se pueden tener graves consecuencias que pueden dirigir al fracaso del proyecto. Para ello existen múltiples metodologías y se debe ser cuidadoso al seleccionar una que se apegue a los requerimientos del proyecto, equipo de trabajo y situación actual del proyecto. No existe una metodología para todo, cada una se desarrolló teniendo en mente su adecuación a una serie de variables que representan la situación de un proyecto.

Para el desarrollo de este proyecto se optó por emplear la metodología Scrum, una metodología ágil que según estadísticas es la más usada por empresas en el mundo debido a la sencillez de su aplicación y a la efectividad que ha demostrado tener (VersionOne, 2015). Las metodologías ágiles se basan en el manifiesto ágil, una serie de principios agrupados en 4 valores:

1. Los individuos y su interacción, por encima de los procesos y las herramientas.
2. El software que funciona, frente a la documentación exhaustiva.
3. La colaboración con el cliente, por encima de la negociación contractual.
4. La respuesta al cambio, por encima del seguimiento de un plan.

Los principales beneficios de las metodologías ágiles según las empresas que aplican alguna son:

1. Habilidad de gestionar el cambio de prioridades
2. Incremento en la productividad del equipo de desarrollo
3. Mejora la visibilidad del proyecto

Se eligió emplear una metodología ágil como Scrum debido a una serie de razones. En primer lugar, los requerimientos pueden cambiar con el tiempo, en el mundo del software es una realidad que pasa casi siempre, no hay software que pueda mejorar sin cambiarse y esto ocurre porque en muchas ocasiones el cliente no tiene una noción exacta de lo que necesita. En el proyecto a desarrollar el cliente no tiene los requerimientos específicos, pero sí una idea de lo que necesita, además de la disponibilidad de tiempo suficiente para llevar a cabo las reuniones que son necesarias para el desarrollo de la metodología, cada 2 a 4 semanas.

Scrum permite tener una rápida y frecuente retroalimentación de los interesados, esto ocurre cada iteración que puede ser de 2 a 4 semanas. Esta rápida retroalimentación permite identificar si los esfuerzos tienen la dirección correcta y de no ser así se pueda corregir tempranamente. Esto es un punto muy importante puesto que se contempla que el desarrollador no es un experto desarrollador del tipo de solución que se ha planteado, de manera que es más propenso a incurrir en errores (incluso los desarrolladores expertos lo hacen).

Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales

y cuando se necesita capacidad de reacción ante la competencia. (IBM developerWorks, 2010)

3.2.1 FLUJO DE TRABAJO

En Scrum el trabajo se descompone en partes pequeñas que se van resolviendo de poco en poco. Se realiza apenas la suficiente planeación para empezar a construir la funcionalidad, se realizan los test y se hace una revisión del trabajo realizado, todo este proceso conforma una iteración conocida como Sprint en Scrum y tiene una duración de 2 a 4 semanas. Este proceso se repite hasta obtener un producto que satisfaga todas las características que solicita el cliente.

3.2.2 ROLES

Existen 3 roles necesarios para llevar a cabo el flujo de trabajo, estos son (IBM developerWorks, 2010):

- **Product Owner:** Es la persona encargada de definir las características que son necesarias para completar el producto, representa a todas las personas interesadas en los resultados del proyecto. Se encarga de definir los objetivos del producto, dirigir los resultados y maximizar el Retorno de Inversión (ROI). Para ello debe colaborar con el equipo en reuniones, mantener priorizados los requisitos, conocer el valor que aporta cada requisito y calcular su Retorno de Inversión, que es el valor que aporta cada requisito con respecto al coste de implementación.

En el caso de este proyecto el rol lo realiza el administrador de la base de taxis.

- **Scrum Master:** Es el líder del equipo y es responsable de facilitar al equipo la realización del trabajo, asegurándose de que el equipo siga el proceso que dicta Scrum y quitando los obstáculos que el equipo le indique. Se encarga de realizar las reuniones, asegurar que la lista de requisitos priorizada esté lista para antes de la siguiente iteración y proteger al equipo de interrupciones durante el Sprint.
- **Team:** Se refiere al equipo que desarrolla el producto. Sus responsabilidades son seleccionar los requisitos que se compromete a realizar durante el Sprint, estimar la complejidad de cada requisito, identificar las tareas para realizar cada requisito, demostrar al cliente lo completado y realizar una retrospectiva al final de cada iteración para evaluar qué se puede mejorar en la forma de trabajar.

3.2.3 ARTEFACTOS

- **Product Backlog:** Es la lista de requisitos priorizada, el Product Owner es el encargado de su creación y gestión. Los requisitos se expresan en forma de historias de usuario y se indica para cada uno el valor que aporta y el coste estimado. En la lista se consideran las posibles iteraciones y entregas esperadas por el cliente, de manera que especifique en qué punto necesita qué requerimientos, esto en función de la capacidad del equipo de desarrollo. Los riesgos son parte fundamental de la lista y se incluye una lista de tareas necesarios para mitigarlos.
- **Sprint Backlog:** Lista de tareas que el equipo elabora en la reunión de planificación de la iteración (Sprint planning) donde se especifican los requisitos que el equipo se compromete a entregar para el final de la iteración y los cuales van ordenados de acuerdo a la prioridad que el Product Owner asignó. Permite identificar tareas donde el equipo tenga problemas para avanzar y de esta manera tomar decisiones al respecto. Si una tarea

depende de otra, se coloca en algún punto por debajo de la que depende y las tareas deben tener un esfuerzo de entre 4 y 16 horas, de manera que sea fácil detectar progreso o estancamiento. Para cada objetivo se muestran las tareas necesarias para su realización, el esfuerzo pendiente y la autoasignación de requisitos que se han hecho los miembros del equipo.

- Gráfico de trabajo pendiente: Provee una referencia del progreso de desarrollo a través de un gráfico y permite estimar si el equipo será capaz de completar el trabajo en el tiempo estimado. Por lo general para este gráfico se utilizan como métrica los días pendientes para completar los requisitos del producto o proyecto (basado en el Product Backlog) y las horas pendientes para completar las tareas de la iteración (basado en el Sprint Backlog).

Las historias de usuario o user stories son descripciones cortas y simples de una característica del producto y son escritas desde la perspectiva de la persona que la necesita con el siguiente formato:

*Yo como un [Rol], necesito [Descripción de la funcionalidad], con la finalidad de
[Descripción de la consecuencia]*

3.3 ESTRATEGIAS DE SOLUCIÓN

3.3.1 FRAMEWORK IONIC

Ionic es un completo SDK de código abierto para el desarrollo de aplicaciones móviles híbridas. Construido sobre AngularJS y Apache Cordova, Ionic proporciona herramientas y servicios para desarrollar aplicaciones móviles híbridas usando tecnologías Web como CSS, HTML5 y Sass. Las aplicaciones se pueden construir con estas tecnologías Web y

luego se distribuyen a través de tiendas de aplicaciones nativas para que se instalen en dispositivos aprovechando Cordova. Ionic fue creado por Max Lynch, Ben Sperry, y Adam Bradley de Drifty Co. en 2013.

3.3.2 ANGULARJS

Es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos JSON estáticos o dinámicos.

3.3.3 HTML

HTML, sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

3.3.4 JAVASCRIPT

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor(Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

3.3.5 ANDROID

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tablets o tabléfonos; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró.

3.3.6 IOS

iOS es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad. No permite la instalación de iOS en hardware de terceros. Tenía el 26 % de cuota de mercado de sistemas operativos móviles vendidos en el último cuatrimestre de 2010, detrás

de Android y Windows Phone. Actualmente su sistema operativo se encuentra en la décima versión, mejor conocida como iOS 10.

3.3.7 APACHE CORDOVA

Apache Cordova (anteriormente PhoneGap) es un popular entorno de desarrollo de aplicaciones móviles, originalmente creado por Nitobi. Adobe compró Nitobi en 2011, le cambió el nombre a PhoneGap, y más tarde liberó una versión de código abierto del software llamado Apache Cordova. Apache Cordova permite, a los programadores de software, construir aplicaciones para dispositivos móviles utilizando CSS3, HTML5, y Javascript en vez de utilizar APIs específicas de cada plataforma como Android, iOS, o Windows Phone. Permite encapsular CSS, HTML, y código de Javascript dependiendo de la plataforma del dispositivo. Extiende las características de HTML y Javascript para trabajar con el dispositivo. Las aplicaciones resultantes son híbridas, lo que significa que no son ni una aplicación móvil nativa (porque toda la representación gráfica se realiza vía vistas de Web en vez del framework nativo) ni puramente basadas en web (porque no son solo aplicaciones web, sino que están empaquetadas como aplicaciones para su distribución y tienen acceso a las APIs nativas del dispositivo).

4 CAPÍTULO IV: DESARROLLO DE MÓDULOS DE FUNCIONALIDAD PARA APLICACIÓN MÓVI

El desarrollo del proyecto se realizó por medio de la utilización de tecnologías modernas de modo que su realización se concretó de manera relativamente rápida debido a que dichas tecnologías ofrecen niveles de abstracción cada vez más altos, de manera que el desarrollador se pueda concentrar casi exclusivamente en el desarrollo de la lógica de programación sin preocuparse por cuestiones como el ambiente de desarrollo o los procesos para la generación de versiones de distribución del producto.

Se realizó la integración con las funcionalidades existentes de la aplicación nativa en las plataformas iOS y Android. Para las tareas de integración de módulos web en la aplicación nativa se desarrolló un proceso automatizado y configurado para ahorrar tiempo y esfuerzo al desarrollador, el proceso fue desarrollado mediante un script de línea de comando.

Para el desarrollo e implementación de los módulos de funcionalidad se utilizaron diversos módulos de la plataforma Apache Cordova como utilerías que funcionan en ambos contextos, nativo y web. Dichos módulos, mejor conocidos como plugins permiten la utilización de las apis nativas desde la aplicación web con JavaScript como por ejemplo la api de networking que permite obtener información de la interfaz de red del dispositivo móvil o la api de geolocalización que permite utilizar el gps. Junto con los plugins se ha hecho uso de la librería ngCordova la cual ofrece un nivel de abstracción mayor para la utilización de los plugins y ofrece un alto nivel de compatibilidad con la forma de trabajo de AngularJS.

En cuanto a los entornos de desarrollo se utilizaron los editores de texto Sublime Text y Atom para la escritura de código y para la compilación de los módulos se utilizaron los IDEs Android Studio para Android y Xcode para iOS.

Como herramientas de debugging se utilizó principalmente Chrome Developer Tools debido a la gran cantidad de utilidades que ofrece y Safari para la versión que se ejecuta en iOS. En cuanto al ambiente nativo se utilizaron los mismo IDEs Android Studio y Xcode.

Para realizar pruebas de las peticiones web que debían realizar los módulos de la aplicación se utilizó Postman que es un cliente para realizar peticiones con una interfaz muy amigable y de fácil uso. También se utilizó Swagger, una herramienta web muy útil para la documentación de las apis que se utilizaron.

Los módulos de funcionalidad requirieron del uso de fuentes de íconos para las interfaces gráficas, entre las cuales se incluye la fuente que integra Ionic por defecto. Además de esta se utilizó un generador de fuentes de vectores llamado icomoon que proporciona un set de íconos gratuitos.

Una de los aspectos más destacables fue la utilización en uno de los módulos de una tecnología poco conocida y utilizada para la comunicación entre dispositivos sin servidores intermediarios, se trata de webRTC para la comunicación en tiempo real utilizando únicamente JavaScript y la especificación webRTC de HTML5. Su utilización ha representado un verdadero reto, ya que en Internet existen pocos ejemplos que demuestren su implementación. Existe documentación de la especificación de HTML5, sin embargo, ésta no es muy clara en cuanto al uso. Considerando los obstáculos que supone la implementación de webRTC resulta ser muy prometedor el hecho de crear un canal de comunicación directo entre dispositivos sin necesidad de servicios web, de manera que la latencia de las comunicaciones sería muy baja, inclusive para clientes que se encuentren en la misma red no haría falta el envío de paquetes a través de Internet, sino que las comunicaciones se realizarían localmente y con poco coste de implementación en cuanto a memoria y procesamiento se refiere.

Esta tecnología no ha sido implementado todavía en un ambiente de producción pues aún hace falta someterla a diversas pruebas de manera que se asegure su funcionamiento y se conozcan sus limitaciones, sin embargo ya hay trabajo en progreso y se espera que próximamente se implemente.

Finalmente se utilizaron los procesos de Scrum como ya se había mencionado. El flujo de trabajo de la metodología fue dirigido por el Líder de proyectos que fungió como Scrum Master. Frecuentemente se realizaban meetings en las que el Líder de proyecto explicaba la estrategia general de trabajo que seguiría el equipo en la semana, por lo cual estas juntas se realizaban usualmente los lunes o martes salvo las ocasiones en las que el Líder deseaba comunicar un cambio de estrategia, las cuales son más frecuentes de lo que se esperarían debido al constante cambio en las necesidades y requisitos que los proyectos suponen.

5 CONCLUSIONES

El verdadero reto fue realizar la completa integración de la aplicación web con toda su funcionalidad con la aplicación móvil nativa y la comunicación entre ambas para los procesos donde se necesitara la interacción, para esto se requirió de leer en ocasiones el código de la aplicación nativa, tanto en la plataforma iOS como Android, el hecho de no estar familiarizado con las tecnologías, especialmente con Objective-C, representó contratiempos y situaciones en las que ya se encontraba la funcionalidad lista pero no se podía comprobar su funcionamiento hasta realizar la integración. Se logró automatizar el proceso mediante la creación de un script de línea de comandos escrito en bash por lo que se puede decir que el objetivo de la integración se logró excepcionalmente.

El desarrollo de los módulos de funcionalidad se realizó con éxito y sin contratiempos debido a la familiaridad que ya se tenía con la tecnología de trabajo. La metodología Scrum permitió un rápido desarrollo y se distingue por no bloquear el proceso creativo con procesos burocráticos, lo cual para este proyecto y por sus necesidades resultó ser muy conveniente.

6 RECOMENDACIONES

Actualmente la empresa tiene un stack de tecnologías casi fijo para todos los proyectos que se realizan, entre estas tecnologías se encuentra el framework Ionic que permite incluir una aplicación web dentro de una aplicación móvil nativa. Resulta muy sencillo desarrollar aplicaciones en Ionic sin embargo durante el desarrollo se descubrieron varios inconvenientes con el framework como errores de diseño, bugs de funcionamiento inexplicables, issues registrados en el foro oficial de Ionic sin resolver de hace varios meses o hasta años. Lo más común era encontrar problemas de rendimiento los cuales no tenían una solución directa debido a las limitaciones que inherentemente conlleva el desarrollo de aplicaciones web para dispositivos móviles. Por lo cual una de las recomendaciones hacia la empresa sería un cambio en la tecnología utilizada para el desarrollo de aplicaciones móviles, ya que existen múltiples alternativas con mejores características como React Native. La otra recomendación tiene que ver más con el proceso de desarrollo y su control, en varias ocasiones se encontró a los desarrolladores teniendo retrasos para encontrar un bug que surgió a consecuencia de un cambio, esto se debe a que no existe una matriz de rastreo que permita visualizar las relaciones entre archivos. Otra recomendación importante

es el hecho de realizar una mejor planeación a la hora de crear una nueva aplicación o módulo, ya que en la forma de trabajo actual no hay un desarrollo que permita escalabilidad ni fácil mantenimiento, aunque esta recomendación es más difícil de llevar a cabo debido a que las aplicaciones se tienen que desarrollar en un lapso de tiempo demasiado corto, por lo que una correcta planeación está lejos de ser realizable.

7 REFERENCIAS BIBLIOGRÁFICAS

Apache Software Foundation. (25 de Marzo de 2017). *Cordova Plugins*. Recuperado el 25 de Marzo de 2017, de Cordova Plugins: <https://cordova.apache.org/plugins/>

Apple. (25 de Marzo de 2017). *Xcode - Apple Developer*. Recuperado el 25 de Marzo de 2017, de Xcode - Apple Developer: <https://developer.apple.com/xcode/>

Dutton, S. (21 de Febrero de 2014). *Getting Started with WebRTC*. Recuperado el 25 de Marzo de 2017, de html5rocks: <https://www.html5rocks.com/en/tutorials/webrtc/basics/>

Facebook. (19 de Febrero de 2017). *React Native | A framework for building native apps using React*. Recuperado el 19 de Febrero de 2017, de React Native: <https://facebook.github.io/react-native/>

Google Inc. (25 de Marzo de 2017). *Android Studio*. Recuperado el 25 de Marzo de 2017, de Android Studio: <https://developer.android.com/studio/index.html?hl=es-419>

IcoMoon. (1 de Noviembre de 2011). *icomoon app*. Recuperado el 25 de Marzo de 2017, de icomoon: <https://icomoon.io/app/#/select>

Ionic. (25 de Marzo de 2017). *ionicons*. Recuperado el 25 de Marzo de 2017, de ionicons: <http://ionicons.com/>

Ionic. (25 de Marzo de 2017). *ngCordova*. Recuperado el 25 de Marzo de 2017, de ngCordova: <http://ngcordova.com/>

Ormandy, K. (19 de Febrero de 2017). *Your favourite app isn't native*. Recuperado el 19 de Febrero de 2017, de Kenneth Ormandy: <http://kennethormandy.com/journal/your-favourite-app-isnt-native>

Postdot Technologies, Inc. (25 de Marzo de 2017). *Postman*. Recuperado el 25 de Marzo de 2017, de Postman: <https://www.getpostman.com/>

SATERAS, J. A. (12 de Julio de 2016). *Instagram User-Agent*. Recuperado el 19 de Febrero de 2017, de mpulp: <https://mpulp.mobi/2016/07/12/instagram-user-agent/>

SmartBear Software. (25 de Marzo de 2017). *Swagger*. Recuperado el 25 de Marzo de 2017, de Swagger: <http://swagger.io/>

Tataje, M. (22 de Noviembre de 2010). *Rational Team Concert for Scrum Projects SCRUM como metodología*. Recuperado el 19 de Febrero de 2017, de IBM developerWorks: <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Rational+Team+Concert+for+Scrum+Projects/page/SCRUM+como+metodolog%C3%ADa>

Telerik. (19 de Febrero de 2017). *Home*. Recuperado el 19 de Febrero de 2017, de NativeScript: <https://www.nativescript.org/>

w3c. (13 de Marzo de 2017). *WebRTC 1.0: Real-time Communication Between Browsers*. Recuperado el 25 de Marzo de 2017, de w3c: <http://w3c.github.io/webrtc-pc/#dom-rtcpeerconnection-createoffer>

Wikipedia. (28 de Diciembre de 2016). *Wikipedia*. Recuperado el 19 de Febrero de 2017, de AngularJS: <https://es.wikipedia.org/wiki/AngularJS>

Wikipedia. (29 de Noviembre de 2016). *Wikipedia*. Recuperado el 19 de Febrero de 2017, de Apache Cordova: https://es.wikipedia.org/wiki/Apache_Cordova

Wikipedia. (8 de Marzo de 2016). *Wikipedia*. Recuperado el 19 de Febrero de 2017, de PhoneGap: <https://es.wikipedia.org/wiki/PhoneGap>

Wikipedia. (3 de Febrero de 2017). *Wikipedia*. Recuperado el 19 de Febrero de 2017, de Wikipedia: [https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework))

Wikipedia. (14 de Febrero de 2017). *Wikipedia*. Recuperado el 19 de Febrero de 2017, de HTML: <https://es.wikipedia.org/wiki/HTML>

Wikipedia. (22 de Enero de 2017). *Wikipedia*. Recuperado el 19 de Febrero de 2017, de JavaScript: <https://es.wikipedia.org/wiki/JavaScript>

Wikipedia. (7 de Febrero de 2017). *Wikipedia*. Recuperado el 19 de Febrero de 2017, de IOS: <https://es.wikipedia.org/wiki/IOS>

Wikipedia. (17 de Febrero de 2017). *Wikipedia*. Recuperado el 19 de Febrero de 2017, de Android: <https://es.wikipedia.org/wiki/Android>

8 ANEXOS

8.1 CRONOGRAMA

CRONOGRAMA DE ACTIVIDADES		Desarrollo de módulos de funcionalidad para aplicación móvil utilizando tecnologías de desarrollo híbrido para la empresa BSD Enterprise S.A de C.V.															
No.	ACTIVIDAD	ENTREGABLE	P/R	ENERO			FEBRERO				MARZO				ABRIL		
				2	3	4	1	2	3	4	1	2	3	4	1	2	
SPRINT 1	1	Preparación del espacio de trabajo y planeación de iteración	1. Product Backlog 2. Sprint Backlog	P													
				R													
	2	Análisis de Requisitos y Diseño	1. Storyboard de la aplicación	P													
				R													
3	Codificación y Pruebas unitarias	1. Código fuente y test automáticos	P														
			R														
4	Pruebas de Integración y Retrospectiva	1. Retroalimentación del Sprint	P														
			R														
SPRINT 2	5	Planeación del Sprint, Análisis de Requisitos y Diseño	1. Storyboard de la aplicación	P													
				R													
	6 y 7	Codificación y Pruebas unitarias	1. Código fuente y test automáticos	P													
				R													
8	Pruebas de Integración y Retrospectiva	1. Retroalimentación del Sprint	P														
			R														
SPRINT 3	9	Planeación del Sprint, Análisis de Requisitos y Diseño	1. Storyboard de la aplicación	P													
				R													
10 y 11	Codificación y Pruebas unitarias	1. Código fuente y test automáticos	P														
			R														
SPRINT 4	12	Pruebas de Integración y Retrospectiva	1. Retroalimentación del Sprint	P													
				R													
13	Integración Final y Retrospectiva	1. Producto integrado	P														
			R														