

# Diseño de una aplicación móvil híbrida mediante Ionic

Dra. Griselda Cortés Barrera\*, Ing. Leonardo D. Mandujano\*\*,  
Dra. Mercedes Flores Flores\* y Mtro. Martín Verduzco Rodríguez\*



## Resumen

Los dispositivos móviles son unos de los elementos tecnológicos que han evolucionado más rápidamente desde su lanzamiento hasta el día de hoy, los cuales fueron desde un simple dispositivo de comunicación hasta los teléfonos inteligentes que se han vuelto indispensables en la actualidad. Dicha evolución ha ocasionado que existan diversos Sistemas Operativos Móviles, los cuales tienen lenguajes de programación nativos para codificar una aplicación, al presente existen herramientas que generan códigos híbridos, los cuales son de gran utilidad cuando se requiere usar la misma aplicación en diferentes plataformas. El presente trabajo, realiza una comparativa entre las aplicaciones híbridas y las nativas, indicando las ventajas y desventajas entre ellas, a fin de tomar una buena decisión a la hora de llevar a cabo el desarrollo de una aplicación móvil.

### Acerca de los autores...

\*Docente en la División de Ingeniería en Sistemas Computacionales del Tecnológico de Estudios Superiores de Ecatepec.

\*\* Investigador de AT&T Administración y Servicios, S. de R.L. de C.V.



**Palabras Clave:** Aplicaciones Móviles, Sistema Operativo, Framework.

## **Abstract**

*Mobile devices are some of the technological elements that have evolved more rapidly from launch until today, ranging from a simple communication device to smartphones that have become indispensable today. This evolution has caused that there are several Mobile Operating Systems, which have native programming languages to encode an application, at there are tools that generate hybrid codes, which are very useful when it is required to use the same application on different platforms. This work, makes a comparison between hybrid and native applications, indicating the advantages and disadvantages between them, in order to make a good decision when carrying out the development of a mobile application.*

**Keywords:** Mobile Applications, Operating System, Framework.

## **Introducción**

Con el paso del tiempo y la evolución tecnológica que se vive día con día, se ha dado la necesidad de contar con herramientas que faciliten la creación de aplicaciones móviles que no sean exclusivas de un sistema operativo, acelerando el proceso de desarrollo y minimizando costos. A finales de los años 90 del siglo anterior, se empezaron a desarrollar las primeras aplicaciones móviles [1], siendo demasiado sencillas, comparadas con las que contamos en la actualidad, como son: agendas, juegos sencillos y tonos, mismas que cumplían funciones muy elementales y su diseño era muy simple [2].

Posteriormente, el 27 de junio del año 2007 se realiza el lanzamiento del iPhone, el cual aparte de ser el primero en manejar pantallas táctiles, entre otras características de hardware, contaba con un sistema operativo iOS 1.0 [3], el cual marcó el principio de la evolución de dispositivos móviles. La evolución de las aplicaciones se dio gracias a las innovaciones de la tecnología Wireless Application Protocol (WAP) y en la transmisión de datos (EDGE), esto vino acompañado de un desarrollo muy fuerte de los mismos y de las aplicaciones ya existentes, mismas que eran descargables desde la App Store de Apple, [2]; en el año 2014 ya contaba con 1'200,000 aplicaciones descargables para su sistema operativo [1]. También se hizo el anuncio oficial de Android [4] en el cual se podrían instalar aplicaciones con tecnologías WAP y EDGE, dichas aplicaciones se pueden descargar desde Google Play lanzado en octubre del 2008 [2]. Aproximadamente 1'300,000 aplicaciones para Android que aprovechan perfectamente las características de los dispositivos móviles actuales [1].

Posteriormente, surgieron sistemas operativos sin tanto éxito como los anteriores, sin embargo, las aplicaciones desarrolladas para dichos sistemas operativos, se generaron en código nativo, dichas plataformas son [5]: Symbian, Windows Mobile, BlackBerry OS, Palm OS. Para la creación de aplicaciones móviles solamente existen dos vertientes: la primera es llevar a cabo el desarrollo de aplicaciones nativas, implementadas en el lenguaje nativo de las plataformas móviles [6]. La segunda opción, es llevar a cabo el desarrollo de la aplicación sobre un código híbrido, las cuales permiten crear aplicaciones nativas, esto es, que se ejecuten de la misma forma que cualquier otra aplicación y se puedan publicar en cualquiera de los mercados, como otra aplicación más [7], surgiendo así los frameworks especializados.

## **Aplicaciones para dispositivos móviles**

Las “apps” se utilizan cada vez más en smartphones y tablets para acceder a noticias, juegos, entretenimiento, tiempo y otras informaciones, por lo que se dice

que una APP es aquel software que utiliza en un dispositivo móvil como herramienta de comunicación, gestión, venta de servicios-productos orientados a proporcionar al usuario la respuesta a sus necesidades de forma automática e interactiva [8]. Las características de una metodología para el desarrollo de aplicaciones para móviles, son las siguientes [9]: agilidad, conciencia del mercado, soporte para la línea de productos de software, desarrollo basado en arquitectura y reusabilidad, inclusión de sesiones de revisión y aprendizaje, especificación temprana de la arquitectura física. Existen tres tipos de aplicaciones móviles: nativas, web e híbridas.

Las aplicaciones nativas se encuentran dentro de un Sistema Operativo Móvil (SOM) o plataforma por default, es decir, que pertenece al mismo sistema y son desarrolladas por la misma compañía, presentan gran compatibilidad, por lo que se reduce el porcentaje de error que se puede presentar dentro de las operaciones que realiza; su desventaja es que muchas veces sólo se aplica a un sistema operativo, sin importar si es de la misma empresa, pues al llegar las nuevas versiones del sistema, se necesita un nuevo desarrollo del mismo para la ejecución sin problemas de la compilación [10].

Por otra parte, cuando ingresamos a una página web desde nuestro dispositivo móvil, nos encontramos con que el navegador Web móvil nos redirecciona a una versión móvil de esa página. Esta versión está diseñada especialmente para este tipo de dispositivos, que son de tamaño reducido, pues muestran las opciones de navegación de más interés para el usuario, así como la información más concisa. Estas aplicaciones web para dispositivos móviles son accedidas desde un navegador web móvil, por lo que se considera una aplicación móvil web. Al no ser una aplicación nativa, cuenta con ventajas y desventajas, véase la Tabla 1 [11].

**TABLA 1**  
**CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE LAS APLICACIONES MÓVILES WEB**

Características	Ventajas	Desventajas
Se accede a través de un navegador de internet.	El código generado es reutilizable, abarcando la mayoría de los dispositivos móviles.	No se pueden utilizar componentes nativos.
	Para dar mantenimiento a la aplicación, se modifica solamente un código, sin importar en que SOM se utilice.	Pueden presentar problemas en algunos navegadores
En cada plataforma, la aplicación se verá casi de la misma forma.	Para realizar pruebas no es necesario descargar la aplicación e instalarla, únicamente se accede a la dirección URL.	
Para su desarrollo, se utilizan las mismas herramientas para crear cualquier aplicación web: por ejemplo, HTML, CSS y JS.	Las actualizaciones se cargan automáticamente y de inmediato, ya que no se instalan físicamente en el dispositivo.	No se puede utilizar offline, ya que difícilmente guardan la información necesaria en el dispositivo.
	Su desarrollo es sencillo y rápido, ya que es más fácil crear y diseñar dentro de un navegador con HTML, CSS y JavaScript, que con lenguajes de desarrollo para móviles.	

En el caso de las aplicaciones híbridas, no son otra cosa que la combinación de la programación nativa con la programación web. De este modo, se pueden aprovechar ciertas utilidades originarias y al mismo tiempo brindar la capacidad multiplataforma a dichas aplicaciones. El nivel de integración dependerá mucho del framework de desarrollo, que están basados en lenguajes de programación Web (HTML, CSS, JS). El framework funciona como un puente entre las API del sistema operativo nativo y el lenguaje web. Los desarrolladores pueden optar por codificar su propio puente o bien aprovechar soluciones ya construidas. Como Ionic, una biblioteca de fuente abierta que provee una interfaz JavaScript uniforme para los distintos sistemas operativos con el fin de que el desempeño sea adecuado. Al no ser un lenguaje nativo cuenta con ventajas y desventajas, como se muestra en la Tabla 2[1].

**TABLA 2**  
**VENTAJAS Y DESVENTAJAS DE LAS APLICACIONES HÍBRIDAS**

<b>Ventajas</b>	<b>Desventajas</b>
Posibilidad de distribución en las diferentes tiendas.	La experiencia de usuario es mucho más cercana a la aplicación web.
Instalación nativa, pero construida en HTML, CSS y JS.	El diseño visual no siempre es similar al ambiente de la plataforma en ejecución.
El código base puede reutilizarse para otras plataformas.	
Se puede acceder a partes del Hardware.	

Para el desarrollo de aplicaciones híbridas, existe una cantidad considerable de frameworks. Un framework es una estructura de software formada por componentes personalizables para la creación de una aplicación[2]. El uso de éstos en cualquier tipo de desarrollo reduce el tiempo de elaboración e implementación, y ayuda a hacer un trabajo sostenible y escalable. Le agrega funcionalidad extendida a un lenguaje de programación, automatiza muchos de los patrones de programación para orientarlos a un determinado propósito proporcionando una estructura al código, mejorándolo y haciéndolo más entendible y sostenible, y permite separar en capas la aplicación. En general, divide la aplicación en tres capas de lógica[3]: presentación, datos, y dominio o negocio. En el desarrollo de software, un framework es una aplicación reutilizable y semi-completa que puede ser especializada para producir aplicaciones individualizadas, ya que es una abstracción de software que proporciona una funcionalidad genérica que debe ser cambiada de manera selectiva por el código adicional escrito por el usuario, lo que genera el software de aplicación específica. Pueden incluir programas de apoyo, compiladores, bibliotecas de código, juegos de herramientas e interfaces de programación de aplicaciones (API, por sus siglas en inglés) que reúnen a todos los diferentes componentes para permitir el desarrollo de un proyecto [4].

Para realizar una aplicación móvil híbrida, existe una cantidad considerable de frameworks que permiten realizar dicho propósito. Los que encabezan la lista de popularidad, se lo deben a las API que manejan, las cuales permiten interactuar con el hardware de los dispositivos de manera nativa. Entre los más utilizados se encuentran:

- Xamarin: Plataforma de desarrollo de aplicaciones móviles para generar aplicaciones 100% nativas para iOS, Android y Windows, a partir de una base de código C#/ .NET común para conseguir entre un 75% y hasta casi un 100% de reutilización del código entre plataformas. Las aplicaciones escritas con Xamarin y C# disponen de acceso completo a las API de la plataforma subyacente, así como de la capacidad de crear interfaces de usuario nativas y de realizar la compilación en código nativo, por lo que el impacto en el rendimiento en tiempo de ejecución es escaso. Propone un enfoque de desarrollo multiplataforma en el que se comparte la codificación completa de la lógica del negocio. Sin embargo, las interfaces deben ser programadas de manera independiente para cada una de las plataformas destino. Así, la reutilización de código, según estudios estadísticos de la compañía Xamarin, es cercana al 85%. Si bien Xamarin cuenta con su propio IDE, denominado Xamarin Studio, es posible integrarlo con Microsoft Visual Studio, y de esta manera generar también aplicaciones para Windows [5].
- PhoneGap: Es una tecnología libre y de código abierto que permite desarrollar apps nativas para múltiples plataformas mediante tecnologías web. El UI para apps de PhoneGap es creado utilizando HTML, CSS, y JavaScript. La capa de UI de una app de PhoneGap consiste en un navegador web que ocupa el 100% de la pantalla del dispositivo. Por lo tanto, lo que PhoneGap permite es, en definitiva, desarrollar apps nativas empleando tecnologías web, es decir, con un código 100% reutilizable para cada una de las plataformas soportadas. Si se analiza la arquitectura de PhoneGap, se llega a la conclusión de que es un framework, que consiste esencialmente en una librería en JavaScript, que facilita a las apps desarrolladas acceder a características de los dispositivos. Así que, PhoneGap tendrá un API que permitirá desarrollar código en JavaScript capaz de implementar funcionalidades nativas, así como gestionar la comunicación del código con las características de los dispositivos.
- Ionic: Kit de desarrollo de software para programar aplicaciones móviles híbridas; provee herramientas y servicios para desarrollarlas mediante tecnología web. Incluye componentes móviles, tipografías, paradigmas interactivos y un esqueleto base extensible[6]; está optimizado con AngularJS. Requiere de Node.js como entorno de ejecución y de Apache Córdova como entorno de desarrollo de aplicaciones. Además de su uso desde el CLI, Ionic cuenta con herramientas como Ionic Creator, con lo cual brinda facilidad y rapidez para la creación de interfaces dentro de la aplicación [7].

## Comparativa entre aplicaciones móviles e híbridas

Cuando se conocen las características de los frameworks para la elaboración de aplicaciones híbridas, es fácil visualizar las diferencias entre ellos, como lo indica la Tabla 2, donde observamos que la mejor opción para el desarrollo de aplicaciones es Ionic, por ser una herramienta *open source*, con un gran número de plugin, los cuales hacen posible interactuar con el hardware de los móviles, y es soportado por un mayor número de plataformas, evitando realizar un desarrollo para cada una de ellas.



**TABLA 3**  
**COMPARATIVA DE FRAMEWORKS PARA APLICACIONES HÍBRIDAS**

Framework	Open Source	Lenguajes	Soporte Web	Plugin APIs	Su fuerte	Plataformas soportadas
Ionic	Sí	Javascript, html5, CSS	safari/crome	Adicionales a los de PhoneGap: 3D touch, gallery, Bluetooth serial, browser tab, calendar, date picker, Facebook, document viewer, Google Analytics, Google Maps, Gyroscope, Instagram, NFC, Photo Library, entre otras.	Vista, funcionamiento	Android, IOS, Windows, Blackberry, Ubuntu, Firefox OS, MacOS, WP8, Browser
PhoneGap	Sí	JavaScript HTML, CSS	safari/crome	Battery status, camera, contacts, device info, accelerometer, dialogs, file, geolocalization, browser, media capture, network information, splash screen, status bar, vibration.	Funcionamiento	IOS, Android, Windows, Windows Phone
Xamarin	No	C#/.net	Facilidades para compilar dentro de Visual Estudio	Librerías que te permiten acceder a las funciones nativas del teléfono	Vista, funcionamiento	IOS, Android, Windows, Windows Phone

## Desarrollo

En el framework Ionic únicamente se tiene que codificar una sola vez, éste a su vez genera el código en el lenguaje nativo al cual se quiere portar, he ahí su gran atractivo. Una vez que se ha descargado el framework desde su página oficial, se podrá comenzar con la codificación inmediatamente, sin embargo, es recomendable que ya se cuente con una metodología a seguir, para distribuir correctamente los módulos del sistema, y así lograr un desarrollo óptimo. Para generar una aplicación odontológica, la metodología más adecuada es la Programación Extrema (XP, por sus siglas en inglés), y una vez que se tiene el contacto con el usuario, se obtiene el diagrama de la Figura 1, donde se muestran las fases de dicha metodología, así como las historias que implican dicho proyecto.



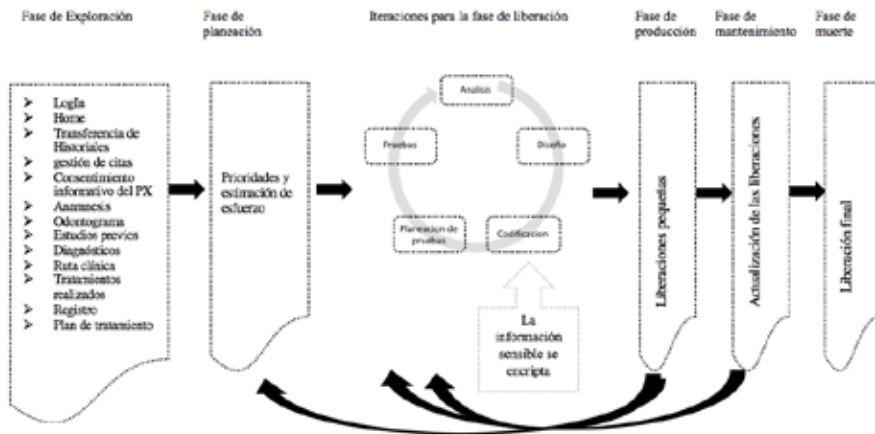


Figura 1

Diagrama XP aplicado al proyecto

Una vez que se realiza la fase de exploración, en donde se visualizan las historias necesarias, se pasa a la fase de planeación, en la cual se da prioridad a dichas historias para verificar el orden en que se desarrollarán, y comenzar la iteración de éstas. En una aplicación móvil, los prototipos suelen ser variados y deben cumplir con las expectativas del usuario. A continuación, se indican los prototipos iniciales por historia para este proyecto. En la Figura 2 se observa la tentativa de prototipos para las historias de iniciar sesión; historia desde la cual se podrá acceder al sistema; Pantalla principal: pantalla principal desde la cual se tendrá acceso al menú principal, mostrando los módulos que le permita el perfil del usuario; Transferencia de historiales: desde la cual se realizará la petición para transferir un historial médico. Además, se podrá gestionar y visualizar las citas de los pacientes.

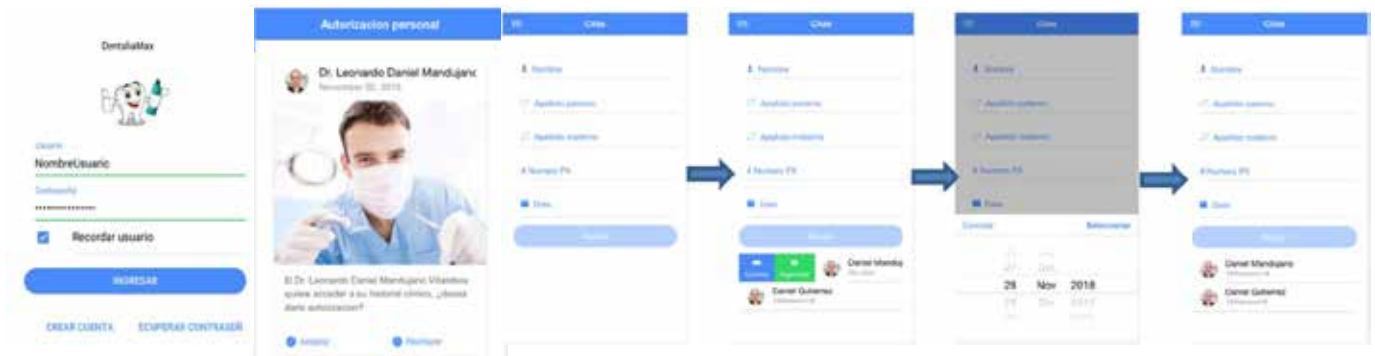


Figura 2

Prototipo de las historias de iniciar sesión, pantalla principal, transferencia de historiales y citas.

La Figura 3 indica la tentativa de prototipos para las historias de: Consentimiento informado: historia en la cual el paciente otorga el consentimiento para que se le brinde atención, esto después de haber escuchado todo el proceso indicado por el odontólogo; Odontograma: se empleará para indicar el estado actual de los dientes; Estudios previos: módulo en el cual se gestionarán los estudios realizados antes de tener la atención médica; Diagnósticos: historia en la cual se podrán consultar los diagnósticos anteriores así como dar de alta alguno nuevo en caso de ser requerido. Por otra parte la Anamnesis, es un conjunto de historias que guardan la información, personal y clínica del paciente, la cual consta de: ficha de identificación, antecedentes y exploraciones. En la ficha de identificación se visualizan los datos personales del paciente, como su nombre y dirección.

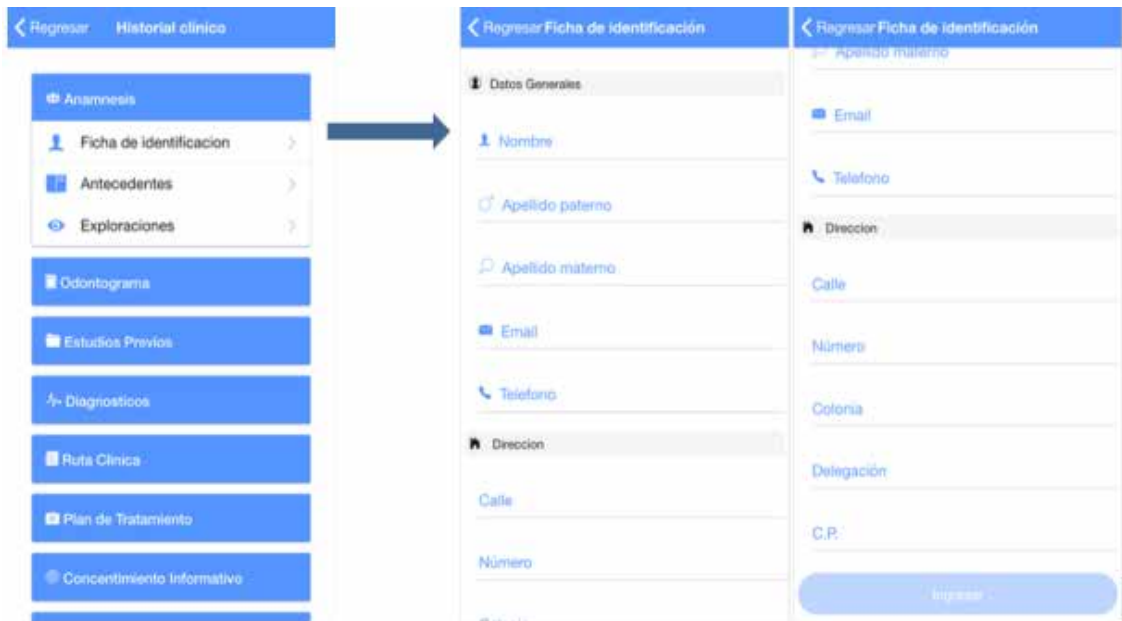
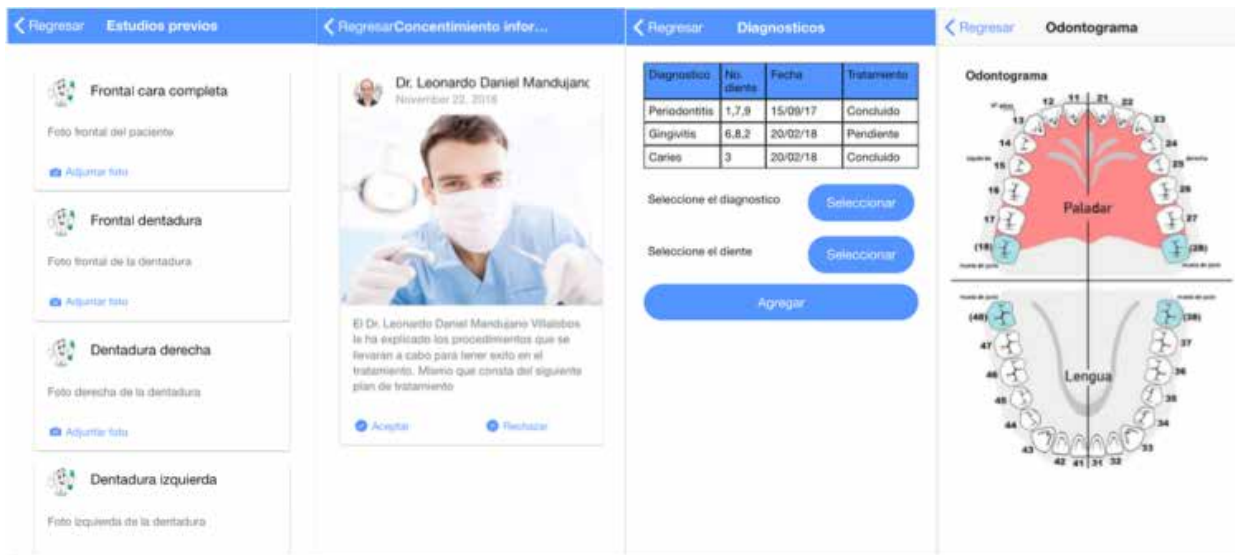


Figura 3

Prototipos del consentimiento del paciente, diagnósticos, odontograma, estudios previos e identificación.

La Figura 4 muestra el prototipo de los Antecedentes, dentro de la cual están los siguientes módulos: Antecedentes herodfamiliares: sección en donde se almacenan los padecimientos médicos de los familiares más cercanos (padres hermanos y abuelos); Antecedentes patológicos: refiere hábitos del px; Médicos: módulo en el cual se visualizan las enfermedades más comunes dentro de la familia.





Figura 4

Prototipo de las pantallas para los Antecedentes.

En la Figura 5 se presenta el prototipo de las exploraciones, módulo en el cual se visualizan las exploraciones físicas, anatómicas y orales del paciente. Además, muestra los prototipos de: Ruta clínica: en esta sección se indica el orden de los tratamientos necesarios para el paciente; Tratamientos realizados: se visualizará el historial de los tratamientos realizados; Plan de tratamiento: eventos cronológicamente organizados para atender los diagnósticos detectados por el odontólogo.



Figura 5

Prototipo de las exploraciones



Figura 6

Prototipos para la ruta clínica, tratamientos realizados y plan de tratamiento.

En la Figura 7 se muestra el registro, historia en donde se da de alta a un nuevo usuario del sistema, esta parte se adaptará dependiendo del perfil del paciente que se dará de alta. Por lo tanto, en el sistema actualmente se está trabajando sobre la fase de iteraciones, en la cual se completará el desarrollo de cada historia, para posteriormente contar con el sistema productivo.



Figura 7

Prototipo para el registro de usuarios y pacientes.

## Conclusiones

Para el desarrollo de aplicaciones híbridas, se tienen varias opciones a la hora de tomar la decisión del framework con el cual se desarrollará dicha aplicación, cada una tiene sus ventajas y desventajas. Si la aplicación a utilizar para codificar es de pocos recursos, la mejor opción es Ionic, ya que es un framework libre, que cuenta con una gran cantidad de APIs que permiten realizar aplicaciones que interactúan con las funciones nativas de los dispositivos (cámara, contactos, giroscopio, entre otros), y desplegarse en diferentes SOM; las plataformas más populares son iOS y Android; sin embargo, en ocasiones resurgen, lanzando nuevas versiones al mercado, por lo que Ionic toma precauciones en ese tema y soporta las plataformas existentes, aunque no estén presentes en el mercado actual. En este caso, la mejor opción fue Ionic, pero dicha decisión se deja al criterio del programador, ya que para la elección de un framework se tiene que tomar en cuenta la naturaleza del proyecto, el presupuesto con el que cuente y las plataformas en donde se piensa desplegar, motivos por los cuales la elección puede cambiar.

## AUTORES:

**GRISelda Cortés Barrera** recibió el grado de Licenciado en ciencias de la informática por el Instituto San Carlos, escuela incorporada al Instituto Politécnico Nacional, México, en 2000. Además, obtuvo la Maestría en Sistemas por ESIME Zacatenco del Instituto Politécnico Nacional, en 2008, También, el 2015 obtuvo el grado de doctor en Sistemas Computacionales por la Universidad Da Vinci. Actualmente, es profesora titular A de tiempo completo en la división de Ingeniería en Sistemas Computacionales del Tecnológico de Estudios Superiores de Ecatepec.

Líneas de investigación: procesamiento del lenguaje natural y reconocimiento de patrones con Redes Neuronales Artificiales. [grisicar7772@yahoo.com.mx](mailto:grisicar7772@yahoo.com.mx)

**LEONARDO D.MANDUJANO V.** Recibió el título de Ingeniero en Sistemas Computacionales por el Tecnológico de Estudios Superiores de Ecatepec, en 2013, labora en AT&T Administración y Servicios, S. de R.L. de C.V. en la dirección de AVP Desarrollo y Soporte de Aplicaciones desarrollando el puesto de desarrollador de software, desde el 2015 en la Línea de Investigación de Tecnología de Información y Comunicaciones.

**MERCEDES FLORES FLORES,** Ingeniera en Computación por la Universidad Nacional Autónoma de México, UNAM; Maestría en Ciencias de la Educación por la Universidad del Valle de México UVM; Dra. en Sistemas Computacionales por la Universidad Da Vinci. Líneas de investigación: cómputo científico y análisis de bioseñales. Profesora titular A de tiempo completo desde el 2005 en la división de Ingeniería en Sistemas Computacionales del Tecnológico de Estudios Superiores de Ecatepec. [merfloresflores@yahoo.com.mx](mailto:merfloresflores@yahoo.com.mx)

**MARTÍN VERDUZCO RODRÍGUEZ,** egresado desde 2001 del Tecnológico de Estudios Superiores de Ecatepec en Ingeniería en Sistemas Computacionales, obtuvo en 2005 el grado de Maestro en Ciencias de la Educación por parte de la Universidad del Valle de México. Labora actualmente como Docente Titular "A" impartiendo cátedra en nivel licenciatura y maestría en el TESE. Su principal interés como investigador son las nuevas tecnologías en el ámbito educativo. [martinverduzco@yahoo.com.mx](mailto:martinverduzco@yahoo.com.mx)

## Bibliografía

- [1] M. V. C. Chacaguasay, "Implementación de una aplicación de realidad aumentada para el control de costos de supermercados". 2016. [En línea]. Disponible en: [http://dspace.utpl.edu.ec/bitstream/123456789/14830/1/Chacaguasay\\_Chacaguasay\\_Marco\\_Vinicio.pdf](http://dspace.utpl.edu.ec/bitstream/123456789/14830/1/Chacaguasay_Chacaguasay_Marco_Vinicio.pdf). [Último acceso: 20 12 2017].
- [2] C. G. Aransay, "Desarrollo de una guía para dispositivos móviles de establecimientos para celiacos en logroño". 27 06 2013. [En línea]. Disponible en: <http://academica-e.unavarr.es/bitstream/handle/2454/7545/578082.pdf?sequence=1>. [Último acceso: 06 10 2017].
- [3] D. A. B. G. Germán Darío Camacho Sánchez, "Diseño de Framework web para el desarrollo dinámico de aplicaciones". 2010. [En línea]. Disponible en: <http://revistas.utp.edu.co/index.php/revistaciencia/articulo/viewFile/1817/1127>. [Último acceso: 13 11 2017].
- [4] S. C. Hohman, "Repositorio digital de la Facultad de Ingeniería - UNAM". 2014. [En línea]. Disponible en: [http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/4271/Tesis\\_Cuauv5.pdf?sequence=1](http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/4271/Tesis_Cuauv5.pdf?sequence=1). [Último acceso: 15 9 2017].
- [5] L. N. Delía, "Desarrollo de aplicaciones móviles multiplataforma". 2017. [En línea]. Disponible en: [http://sedici.unlp.edu.ar/bitstream/handle/10915/60497/Documento\\_completo\\_\\_pdf-PDFA.pdf?sequence=3](http://sedici.unlp.edu.ar/bitstream/handle/10915/60497/Documento_completo__pdf-PDFA.pdf?sequence=3). [Último acceso: 15 01 2018].
- [6] L. Ramos, "Desarrollo de aplicación móvil para mejorar la comunicación en pequeñas y medianas empresas". 2015. [En línea]. Disponible en: <http://159.90.80.55/tesis/000170749.pdf>. [Último acceso: 20 02 2018].
- [7] M. L. A. García y K. D. R. Calderón, "GeoPetFinder: Aplicación para dispositivos móviles para la búsqueda de perros extraviados en la ciudad de Bogotá". 2017. [En línea]. Disponible en: <http://repository.udistrital.edu.co/bitstream/11349/6510/1/AlmarioGarc%C3%ADaMarguilLorena2017.pdf>. [Último acceso: 20 02 2018].
- [8] SEP, "gob.mx," 2016. [En línea]. Disponible en: [https://www.gob.mx/cms/uploads/attachment/data/file/118382/El\\_Modelo\\_Educativo\\_2016.pdf](https://www.gob.mx/cms/uploads/attachment/data/file/118382/El_Modelo_Educativo_2016.pdf). [Último acceso: 22 Febrero 2017].
- [9] P. Rojas, "Historia y tipos de las aplicaciones móviles". 2015. [En línea]. Disponible en: <http://pedromrojas12.blogspot.mx/2015/12/historia-de-las-aplicaciones-moviles.html>. [Último acceso: 21 11 2017].
- [10] A. Ávila, "Historia y evolución de las aplicaciones móviles más conocidas". 07 08 2015. [En línea]. Disponible en: <http://alejandruplicacionesmoviles.blogspot.mx/2015/08/>. [Último acceso: 22 11 2017].
- [11] Redacción El Tiempo, "Así ha evolucionado el iPhone a lo largo de su historia". El Tiempo, 13 09 2017. [En línea]. Disponible en: <http://www.eltiempo.com/tecnosfera/tutoriales-tecnologia/evolucion-del-iphone-en-su-historia-40227>. [Último acceso: 22 11 2017].
- [12] D. Salas, "La historia y los comienzos de Android, el sistema operativo de Google". El androide libre, 18 08 2011. [En línea]. Disponible en: <https://elandroidelibre.elespanol.com/2011/08/la-historia-y-los-comienzos-de-android-el-sistema-operativo-de-google.html>. [Último acceso: 22 11 2017].
- [13] C. A. A. C. Jorge Armando Aparicio, "Tecnología móvil como herramienta de apoyo en la educación media". 2012. [En línea]. Disponible en: [http://utec.edu.sv/vips/uploads/investigaciones/tecnologia\\_movil\\_herramienta\\_apoyo.pdf](http://utec.edu.sv/vips/uploads/investigaciones/tecnologia_movil_herramienta_apoyo.pdf). [Último acceso: 09 10 2018].
- [14] I. M. Sola, "Desarrollo de apps multiplataforma: Un caso comparativo". 2014. [En línea]. Disponible en: <http://repositorio.upct.es/bitstream/handle/10317/4122/pfc5875.pdf?sequence=1>. [Último acceso: 07 11 2017].
- [15] S. O. Adriana, "Repositorio digital de la Facultad de Ingeniería - UNAM". 2012. [En línea]. Disponible en: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/2808/Tesis.pdf?sequence=1>. [Último acceso: 15 09 2017].
- [16] L. F. Benítez, "La implementación del mobile marketing como herramienta multidisciplinar en el sector turístico y aeroportuario". [En línea]. Disponible en: <http://www.eumed.net/libros-gratis/2016/1539/aplicacion.htm>. [Último acceso: 04 01 2018].
- [17] I. M. R. Villagrán, "Progresos en el desarrollo de aplicaciones para dispositivos móviles". 2013. [En línea]. Disponible en: <http://132.248.9.195/ptd2013/abril/300218002/300218002.pdf>. [Último acceso: 16 01 2018].
- [18] Universidad Interamericana para el Desarrollo, "Aplicaciones móviles nativas". 2013. [En línea]. Disponible en: [http://moodle2.unid.edu.mx/dts\\_cursos\\_md/pos/TI/MN/S02/MN02\\_Lectura.pdf](http://moodle2.unid.edu.mx/dts_cursos_md/pos/TI/MN/S02/MN02_Lectura.pdf). [Último acceso: 05 01 2018].
- [19] L. M. Zamores, "Aplicación web y móvil de consulta de proyectos SIRF (Sistema Institucional de Registro Financiero) para el lingen UNAM". 2014. [En línea]. Disponible en: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/3004/Tesis.pdf?sequence=3>. [Último acceso: 01 01 2018].
- [20] H. Jaimes, "Devs Xamarin CDMX". 2016. [En línea]. Disponible en: <http://www.hjr.com.mx/2016/04/lo-bueno-lo-malo-y-lo-feo-de-xamarin-mi.html>. [Último acceso: 23 11 2017].
- [21] C. A. A. Cáceres, "Desarrollo de la aplicación móvil multiplataforma-ToCook". 2105. [En línea]. Disponible en: [http://oa.upm.es/43116/8/TFM\\_CARLOS\\_ANTONIO\\_AZANON\\_CACERES.pdf](http://oa.upm.es/43116/8/TFM_CARLOS_ANTONIO_AZANON_CACERES.pdf). [Último acceso: 07 01 2018].