

COMPARATIVO DE FRAMEWORK OPEN SOURCE PARA EL DESARROLLO
DE APLICACIONES MÓVILES HÍBRIDAS EN CENTRO DE INVESTIGACIÓN
APLICADA Y DESARROLLO EN TECNOLOGÍAS DE LA INFORMACIÓN
(CIADTI), UNIVERSIDAD DE PAMPLONA

SANDRO ANTONIO CARDENAS LEAL

UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA, DEPARTAMENTO DE
ELECTRÓNICA, ELÉCTRICA, SISTEMAS Y TELECOMUNICACIONES
INGENIERÍA DE SISTEMAS
PAMPLONA

2018

COMPARATIVO DE FRAMEWORK OPEN SOURCE PARA EL DESARROLLO
DE APLICACIONES MÓVILES HÍBRIDAS EN CENTRO DE INVESTIGACIÓN
APLICADA Y DESARROLLO EN TECNOLOGÍAS DE LA INFORMACIÓN
(CIADTI), UNIVERSIDAD DE PAMPLONA

SANDRO ANTONIO CARDENAS LEAL

Trabajo presentado con el fin de obtener el título de ingeniero de sistemas y
obtener conocimientos de tecnologías para el desarrollo móvil.

DIRECTOR

MSc. AVILIO VILLAMIZAR

CO DIRECTOR

Esp. ELVIS NAVARRO VEGA

UNIVERSIDAD DE PAMPLONA

FACULTAD DE INGENIERÍAS Y ARQUITECTURA, DEPARTAMENTO DE
ELECTRÓNICA, ELÉCTRICA, SISTEMAS Y TELECOMUNICACIONES

INGENIERÍA DE SISTEMAS

PAMPLONA

2018

Agradecimientos

A mí amada madre Ernestina que con su esfuerzo, dedicación y trabajo duro saco adelante su familia y me permitió estar aquí. A mi novia Alejandra por sus consejos y apoyo que siempre están presentes y me permitieron culminar este proceso. A los ingenieros Elvis y Avilio por la oportunidad que me brindaron al darme la posibilidad de abordar este trabajo. A los docentes de la Universidad de Pamplona pertenecientes al programa ingeniería de sistemas, por su tiempo y conocimiento compartidos, en especial al profesor Mauricio rojas por su amabilidad y buena disposición en el proceso, aunque no tuviera relación con este.

Muchas gracias a todos.

Sandro Antonio Cardenas Leal

Dedicatoria

A mi madre

Ernestina Leal por haberme apoyado en todo momento, por sus consejos, por su atención, porque gracias a ella estoy cumpliendo uno de mis sueños, ella la persona más amada e importante para mí. Un gran ejemplo de vida y superación.

A mi padre

Antonio Cardenas que en paz descanse, espero que se sienta orgulloso de mis logros.

A mi hermano

Héctor José que, a pesar de las peleas, siempre lo tengo presente y espero que se sienta orgulloso.

A mi novia

Alejandra Rozo, que siempre está ahí para mí, es mi compañera de aprendizaje, de lucha, de superación, de desvelo, gracias por su amor, apoyo y comprensión.

Tabla de contenidos

1	INTRODUCCIÓN.....	11
1.1	PROBLEMA.....	12
1.2	JUSTIFICACIÓN.....	14
1.3	OBJETIVOS.....	16
1.3.1	Objetivo general	16
1.3.2	Objetivos específicos	16
1.4	METODOLOGÍA	17
1.4.1	Paradigma.....	17
1.4.2	Alcance	17
1.4.3	Población a la que va dirigido	17
1.4.4	Resultados esperados.....	17
1.4.5	Consideraciones éticas	17
2	MARCO DE REFERENCIA	18
2.1	INTRODUCCIÓN.....	18
2.2	ANTECEDENTES.....	18
2.2.1	Ámbito internacional.....	18
2.2.2	Ámbito nacional.....	20
2.2.3	Ámbito local.....	23
2.3	MARCO CONCEPTUAL	25
2.4	MARCO TERICO.....	26
2.4.1	Software	26
2.4.2	Framework	29
2.4.3	Aplicaciones móviles	30
2.4.4	Desarrollo back-end y front-end	31
2.4.5	Metodologías usadas en el desarrollo móvil	34
2.4.6	Opciones del desarrollo móvil	41
2.4.7	Revisión de framework híbridos	47
3	COMPARATIVO Y SELECCIÓN DEL FRAMEWORK.....	59
3.1	INTRODUCCIÓN.....	59

3.2	ESCALA DE MEDICIÓN.....	59
3.3	INSTRUMENTO DE EVALUACIÓN	60
3.4	CRITERIOS DE EVALUACIÓN	60
3.4.1	Funcionalidad.....	60
3.4.2	Mantenibilidad	61
3.4.3	Usabilidad	61
3.4.4	Portabilidad	61
3.4.5	Índices de interés	61
3.5	COMPARATIVO	62
3.5.1	Funcionalidad.....	62
3.5.2	Mantenibilidad	92
3.5.3	Usabilidad	101
3.5.4	Portabilidad	110
3.5.5	Índices de interés	111
3.6	SELECCIÓN DEL FRAMEWORK	119
3.6.1	Funcionalidad.....	119
3.6.2	Mantenibilidad	121
3.6.3	Usabilidad	122
3.6.4	Portabilidad	122
3.6.5	Índices de interés	123
4	CONSTRUCCIÓN DEL PROTOTIPO.....	126
4.1	ANÁLISIS.....	127
4.2	DISEÑO.....	135
4.3	DESARROLLO	141
5	CONCLUSIONES	152
6	TRABAJO FUTURO	153
7	FUENTES BIBLIOGRÁFICAS	154

Índice de tablas

Tabla 1 - Framework híbridos y sus tecnologías.....	47
Tabla 2 - Métricas de evaluación	59
Tabla 3 - Matriz de pesos	60
Tabla 4 - Matriz de pesos llena con los resultados del estudio comparativo	124
Tabla 5 - Especificación de requerimiento validar ingreso.....	128
Tabla 6 - Especificación de requerimiento seleccionar programa académico	129
Tabla 7 - Especificación de requerimiento pestaña o tab home	129
Tabla 8 - Especificación de requerimiento pestaña o tab de notificaciones.....	130
Tabla 9 - Especificación de requerimiento pestaña o tab de mapa.....	130
Tabla 10 - Especificación de requerimiento horario	130
Tabla 11 - Especificación de requerimiento registro extendido.....	131
Tabla 12 - Especificación de requerimiento información financiera	131
Tabla 13 - Especificación de requerimiento agenda	132
Tabla 14 - Especificación de requerimiento calificaciones.....	132
Tabla 15 - Especificación de requerimiento semáforo académico.....	133
Tabla 16 - Herramientas de software utilizadas para la construcción del prototipo	141

Índice de Ilustraciones

Ilustración 1 - Capas de Desarrollo.....	32
Ilustración 2 - Fases metodología de desarrollo móvil.....	35
Ilustración 3 - Diagramas UML para modelado.....	38
Ilustración 4 - App nativa vs App híbrida.....	46
Ilustración 5 - Estructura de ficheros Ionic.....	64
Ilustración 6 - Estructura de un módulo en Ionic.....	66
Ilustración 7 - Estructura de un template en Ionic.....	67
Ilustración 8 - Estructura de un componente en Ionic.....	67
Ilustración 9 - Estructura de un provider en Ionic.....	68
Ilustración 10 - Arquitectura de datos en Ionic.....	69
Ilustración 11 - Pantalla inicial en un nuevo proyecto de Ionic.....	70
Ilustración 12 - Puenteo en React Native.....	71
Ilustración 13- Estructura de un componente de React Native.....	74
Ilustración 14 - Estructura de un componente mensaje en React Native.....	75
Ilustración 15 – Llamado del componente mensaje en otro componente Root.....	75
Ilustración 16 - Estructura de un componente con cambio de state en React Native.....	76
Ilustración 17 - Estructura de un archivo HTML que importa Framework7.....	80
Ilustración 18 - Estructura de un botón HTML en framework7.....	80
Ilustración 19 - Archivos que conforman el componente button.....	81
Ilustración 20 - Objeto de Rutas de Framework7.....	82
Ilustración 21 - Arquitectura de una aplicación Cordova.....	84
Ilustración 22 - Estructura de un proyecto PhoneGap.....	86
Ilustración 23 - Navegación entre pantallas en PhoneGap.....	87
Ilustración 24 - Componente utilizados en una aplicación de NativeScript.....	88
Ilustración 25 - Estructura de una vista en NativeScript.....	90
Ilustración 26 - Estructura de código JavaScript de una vista.....	90
Ilustración 27 - Etiqueta de navegación.....	91
Ilustración 28 – Modulo de Navegación.....	91
Ilustración 29 - Interés en Ionic a nivel internacional 2013-2018.....	111
Ilustración 30 - Interés en Ionic a nivel nacional 2013-2018.....	112
Ilustración 31- Interés en Ionic 2 a nivel internacional 2013 - 2018.....	112
Ilustración 32 - Interés en Ionic 2 a nivel nacional 2013 - 2018.....	112
Ilustración 33 – Interés en Ionic 3 a nivel internacional 2013-2018.....	113
Ilustración 34 - Interés en Ionic 3 a nivel nacional 2013-2018.....	113
Ilustración 35 - Interés en Angular a nivel internacional 2013-2018.....	113
Ilustración 36 - Interés en Angular a nivel nacional 2013-2018.....	114
Ilustración 37 – Interés en React a nivel internacional 2012-2018.....	114
Ilustración 38 - Interés en React a nivel nacional 2012-2018.....	114

Ilustración 39 - Interés en React Native a nivel internacional 2012-2018	115
Ilustración 40 - Interés en React Native a nivel nacional 2012-2018	115
Ilustración 41 – Interés en Framework 7 a nivel internacional 2014-2018	115
Ilustración 42 – Interés en Framework 7 a nivel internacional 2014-2018	116
Ilustración 43 – Interés en PhoneGap a nivel internacional 2008-2018.....	116
Ilustración 44 - Interés en PhoneGap a nivel nacional 2008-2018.....	116
Ilustración 45 – Interés en Apache Cordova a nivel internacional 2008-2018	117
Ilustración 46 – Interés en Apache Cordova a nivel nacional 2008-2018	117
Ilustración 47 - Interés en NativeScript a nivel internacional 2014-2018.....	117
Ilustración 48 - Interés en NativeScript a nivel nacional 2014-2018.....	118
Ilustración 49 – Interés de búsqueda en los framework a nivel internacional	118
Ilustración 50 – Interés de búsqueda en los framework a nivel nacional	119
Ilustración 51 - Diagrama de casos de usos de la aplicación móvil	133
Ilustración 52 - Petición al API Rest de la Universidad de Pamplona	134
Ilustración 53 - Respuesta de petición a un API Rest, formato JSON	134
Ilustración 54 - Wireframe de la pantalla de login y cambio de programa.....	135
Ilustración 55 - Wireframe de los tabs home y mapa	136
Ilustración 56 - Wireframe de tab notificaciones	136
Ilustración 57 - Wireframe de menú desplegable y acceso a pantalla de horario	137
Ilustración 58 - Wireframe de acceso a pantalla calificaciones	137
Ilustración 59 - Wireframe acceso a pantalla de registro extendido.....	138
Ilustración 60 - Wireframe de acceso a pantalla de resumen financiero	138
Ilustración 61 - Wireframe de pantalla de semáforo académico	139
Ilustración 62 - Wireframe de acceso a pantalla de directorio	139
Ilustración 63 - Diagrama de secuencia del tab notificaciones	140
Ilustración 64- Diagrama de secuencia para el evento mostrar la pantalla del horario.....	140
Ilustración 65 - Diagrama de secuencia para el evento mostrar la pantalla del registro extendido	141
Ilustración 66 - Componente de login	142
Ilustración 67 - Pantalla de login	143
Ilustración 68 - Alerta de error en inicio de sesión	143
Ilustración 69 - Código formulario de logueo	144
Ilustración 70 - Pantalla seleccionar programa académico.....	144
Ilustración 71 – Pantalla home.....	145
Ilustración 72 – Pantalla mapa.....	145
Ilustración 73 – Menú desplegable	146
Ilustración 74 - Pantalla horario	147
Ilustración 75 - Pantalla registro extendido	147
Ilustración 76 - Pagina semáforo	148
Ilustración 77 - Pantalla información financiera.....	148

Ilustración 78 – Pantalla detalle otras liquidaciones.....	149
Ilustración 79 - Pantalla calificaciones	149
Ilustración 80 - Pantalla agenda.....	150
Ilustración 81 - Estructura de un provider en Ionic.....	151
Ilustración 82 - Petición http básica a un API Rest	151

1 INTRODUCCIÓN

Actualmente el mundo de la informática avanza constantemente a pasos agigantados, con el fin de satisfacer la demanda constante de quienes se encuentran inmersos en ese mundo, tan rápido es su avance, que los profesionales de área suelen sentir, que no terminan de dominar una tecnología cuando ya está emergiendo otra. Los profesionales del área de la informática se encuentran con una gran cantidad de posibilidades, que los suelen poner indecisos a la hora de escoger una, y así poder cumplir con sus obligaciones.

El caso de las aplicaciones móviles no es la excepción, en un mundo sumergido en Internet y en las facilidades que prestan los teléfonos inteligentes (Smartphone), las aplicaciones móviles cuentan con gran cantidad de tecnologías para su desarrollo y soporte. Dicho coloquialmente “Hay de dónde escoger”, pero para elegir una tecnología entre tantas se deben tener en cuenta ciertos factores, como en qué contexto se encontrará la aplicación, cuál será su función, será dinámica o estática, etc.

Dentro del desarrollo móvil se puede decir que existen dos tipos de desarrollo que se diferencian una del otro, por su forma de trabajar, las tecnologías que usa y su forma de ejecución sobre el dispositivo. Estos son el desarrollo móvil híbrido que se basan en tecnologías que son ajenas al sistema operativo del dispositivo y el desarrollo móvil nativo que trabaja con el lenguaje propio del sistema operativo.

Este proyecto se centra en las tecnologías de desarrollo de aplicaciones móviles híbridas, las cuales permiten un único desarrollo que pueda ser implementado en múltiples plataformas, a diferencia de las tecnologías de desarrollo móvil híbrido que necesitan un desarrollo por cada uno de los sistemas operativos.

Se pretende indagar las tecnologías usadas para el desarrollo de este tipo de framework móviles que en la actualidad se encuentran en auge, de ahí se realizó un estudio comparativo de algunos de los framework open source diseñados para este fin, se establecieron criterios de evaluación para comparar los framework y

permitir el análisis de cada uno, optando por el que mayor puntaje o aceptación tuviera según cada uno de los criterios que se seleccionaron.

Finalmente se desarrolló un prototipo con la tecnología seleccionada mediante el estudio comparativo. De este se partirá para próximos desarrollos dentro del centro de investigación aplicada y desarrollo en tecnologías de la información.

1.1 PROBLEMA

Las tecnologías móviles en la actualidad están en constante crecimiento, la gran mayoría de empresas cuentan con aplicativos móviles, ya que su uso, facilita la interacción y el acceso a los servicios en la nube ofrecidos por las empresas. En la actualidad una gran cantidad de personas cuentan con un Smartphone indiferente de la marca y el sistema operativo, capaz de almacenar y ejecutar estas aplicaciones, pero debido a esto mismo el desarrollo de aplicaciones se puede tornar complicado y costoso, es decir, el hecho que exista gran variedad de dispositivos y por ende sistemas operativos, implica diferentes tecnologías de desarrollo, y con tecnologías se hace referencia a el lenguaje nativo sobre el cual se soporta dispositivo.

Debido a esto surgió el desarrollo móvil híbrido el cual “permite combinar las tecnologías más usadas en la web como lo son HTML, CSS y JavaScript logrando un desarrollo con un alto impacto visual, mejorando la semántica y el posicionamiento SEO entre muchas otras ventajas”¹, con lenguajes de programación nativos, que se denominan así porque son el lenguaje propio de la terminal del sistema operativo, permitiendo acceder a recursos del dispositivo como lo son la cámara, el GPS, sensores, entre otros. En donde SEO (Search Engine Optimization) hace referencia a la optimización de búsqueda de una aplicación, en el lugar donde este alojada.

¹ www.programacion.net. Comparación de frameworks para desarrollar apps para móvil híbridas. Programación.net [en línea]. 23 de enero de 2016[revisado el 15 de julio de 2018]. Disponible en Internet: <http://programacion.net/articulo/comparacion-de-frameworks>

Entre los lenguajes nativos para dispositivos móviles más conocidos se encuentran java para el caso de Android, C# para el caso de Windows Phone, Swift en el caso de IOS.

La ventaja de las aplicaciones híbridas es que solo se necesita implementar una vez, y en la mayoría de los casos la aplicación se ejecutará sin problemas en dispositivos móviles independientemente de su sistema operativo ya sea Android, IOS, entre otros, ya que el núcleo de la aplicación está escrito en HTML, CSS y JavaScript y la traducción de estas instrucciones para que el dispositivo las entienda se hace generalmente a través de un intermediario.

Si se observa detenidamente la mayoría de las personas cuenta con un dispositivo móvil y los usuarios de la Universidad de Pamplona no son la excepción, cada uno cuenta con gustos particulares en cuanto a elección de dispositivo móvil y sistema operativo, pero si hablamos de cuota de mercado y éxito económico Android e IOS son los vencedores.

El Centro de Investigación Aplicada y Desarrollo de las Tecnologías de la Información (CIADTI) de la Universidad de Pamplona, cuenta con un conjunto de aplicaciones para el sistema operativo Android desarrollada con la librería JQuery Mobile, y a futuro no descarta la posibilidad de desarrollar en otras plataformas, por lo que surge la necesidad de indagar mediante un estudio comparativo acerca de framework de desarrollo móvil híbrido, buscando evitar desarrollos complejos, tiempo y esfuerzo innecesarios.

Haciendo una comparación de los beneficios potenciales de cada uno de los framework, a través de criterios de evaluación como documentación disponible, comunidad que apoya la tecnología, tasa de aprendizaje, entre otros, que serán seleccionados a partir de experiencias subidas en la web de desarrolladores en el área, trabajos relacionados con el área a través de las bases de datos bibliográficas y del personal experimentado en desarrollo dentro del (CIADTI).

1.2 JUSTIFICACIÓN

Debido a la gran cantidad de tecnologías que han emergido para solucionar los problemas de las empresas y de los desarrolladores, se hace necesario estar muy atento y ser astuto a la hora de escoger las que mejor se acomoden a las necesidades de una empresa, ya que muchas tecnologías pueden parecerse en finalidad y manera de usar pero cada una ofrece funcionalidades y características únicas, todo debido a la competitividad constante que ayuda en gran medida a un desarrollo más rápido y de calidad.

Una manera de escoger la tecnología que mejor se adecua a las necesidades, es hacer un estudio comparativo, en donde se tomaran criterios de evaluación de cada una de las tecnologías en este caso los framework de desarrollo móvil híbrido, y, después analizando y comparando estos criterios se puede llegar a resultados que permiten definir cuál framework es más apto para el desarrollo y cual se adecua más a un entorno y forma de trabajar específicos.

En una empresa como lo es el CIADTI, competitiva, que se encuentra en constante crecimiento y en búsqueda de nuevas tecnologías que faciliten las operaciones, la investigación y el desarrollo, se planea el uso de framework de desarrollo móvil híbrido buscando dar compatibilidad a múltiples plataformas (sistemas operativos) de manera sencilla ahorrando tiempo, costos, y permitiendo el desarrollo de código mucho más fácil y ordenado.

Actualmente se cuentan en el CIADTI con múltiples desarrollos móviles mediante la tecnología JQuery Mobile, que, aunque tiene la misma finalidad, la de desarrollar aplicaciones móviles híbridas, pero este, no cuenta con una estructura de desarrollo definida que permita mantenibilidad y escalabilidad de las aplicaciones, ya que más que un framework es una librería sin ningún tipo de patrón de desarrollo definido. Un estudio comparativo acerca de los framework de desarrollo móvil híbrido le da al CIADTI el soporte teórico que necesita, acerca de las tecnologías de desarrollo móvil híbrido, aprovechando todas sus ventajas y dando seguridad en desarrollos mediante estos.

A demás la implementación de un framework promueve las buenas prácticas de desarrollo, como la implementación de estructuras de ficheros definidas, patrones de diseño, reutilización de código, facilidad y agilidad de desarrollo, entre otras que darían un valor agregado a una Institución como el CIADTI.

1.3 OBJETIVOS

1.3.1 Objetivo general

Elaborar estudio comparativo de framework open source para el desarrollo de aplicaciones móviles híbridas en el Centro de Investigación Aplicada y Desarrollo en Tecnologías de la Información (CIADTI) Universidad de Pamplona.

1.3.2 Objetivos específicos

- Realizar una revisión bibliográfica digital o escrita de las tecnologías usadas para el desarrollo móvil híbrido.
- Establecer criterios para evaluar los framework, compararlos y analizarlos.
- Seleccionar el framework de desarrollo móvil híbrido que se adecue a las necesidades y los requerimientos del CIADTI.
- Validar el framework seleccionado según el análisis realizado, mediante un prototipo.

1.4 METODOLOGÍA

1.4.1 Paradigma

La presente investigación usa un paradigma cualitativo, puesto que se centra en la recolección de información de desarrollo móvil híbrido, mediante mesas de trabajo e investigación en fuentes digitales como foros, documentaciones oficiales y blogs, para su interpretación y uso en esta investigación.

1.4.2 Alcance

El alcance de la investigación es exploratorio, ya que el objeto de esta es examinar el mundo del desarrollo de aplicaciones móviles mediante los framework híbridos, siendo este un tema desconocido tanto en la base bibliográfica de la universidad, como para el investigador.

1.4.3 Población a la que va dirigido

El objetivo principal va enfocado a cumplir requerimientos dados por el CIADTI, en cuanto a la aplicación móvil actual, pero indirectamente se verán beneficiados estudiantes de la Universidad de Pamplona, esto en caso que el prototipo desarrollado con el framework resultado del estudio comparativo, finalice su fase de desarrollo, y sea implementado.

1.4.4 Resultados esperados

- Conocimiento general acerca del mundo del desarrollo móvil híbrido.
- Selección de un framework de desarrollo móvil híbrido.
- Desarrollo de un prototipo móvil híbrido.
- Dar a conocer las ventajas de usar un framework para el desarrollo de aplicaciones móviles.

1.4.5 Consideraciones éticas

Se tuvo en cuenta que los framework a utilizar fueran del tipo software libre, ya que no tienen ningún problema con el uso de su código. Se contó con acceso a información de la base de datos de prueba del CIADTI con la mayor responsabilidad, ya que esta información les pertenece y no puede ser tomada para otros fines.

2 MARCO DE REFERENCIA

2.1 INTRODUCCIÓN

En el siguiente capítulo se dará inicio a la construcción del marco de referencia, el cual tiene por objetivo dar a conocer los conocimientos que previamente se deben tener acerca de una temática en particular. Trabajos e investigaciones previas de la temática abordada, conceptos, definiciones y teorías propuestas hacen parte de la información que se abordó en el presente capítulo.

2.2 ANTECEDENTES

2.2.1 Ámbito internacional

Título: Estudio de la eficiencia de los frameworks híbridos y nativos en el desarrollo de aplicaciones móviles, basado en benchmark para el consorcio informega.

Autores: Salgado Escobar Stalin Sebastián.

Lugar: Universidad de las fuerzas armadas, Quito, Ecuador.

Fecha: 2018.

Link: <http://repositorio.espe.edu.ec/xmlui/handle/21000/14003>

Palabras clave: Android, Sistema Operativo, Interacción Celular, Aplicaciones Móviles, Lenguaje De Programación.

Descripción

Hoy en día, el crecimiento del uso de dispositivos móviles conjuntamente con la amplia disponibilidad de Internet, ha abierto varias aristas para brindar servicios a los usuarios, por medio de aplicaciones móviles, las mismas que desde su nacimiento ha existido un método nativo de desarrollo ya sea para Android usando Java como para IOS usando Objective C, esta característica sumado al crecimiento de las aplicaciones móviles ha motivado el nacimiento de nuevos métodos de desarrollo de aplicaciones móviles denominadas híbridas las mismas que permiten realizar un solo desarrollo para varias plataformas. Esta investigación plantea realizar un análisis comparativo de la eficiencia del comportamiento temporal de aplicaciones móviles desarrolladas para Android estructurando un modelo de calidad IQMC comparando el método nativo versus uno de los métodos híbridos en este caso Ionic Framework, planteándose un caso de estudio el cual se detalla en el Capítulo IV, lo cual permitirá evaluar la

eficiencia de los dos métodos determinado cual es el más eficiente entre los dos entregando a la comunidad de desarrollo datos que les permita decidir cuál de los dos métodos utilizar.²

Título: BiciPark: Web App con Ionic y acceso a servicios Rest.

Autores: Cano Insa, Esteban.

Lugar: Universidad de alicante, Alicante, España.

Fecha: 2017.

Link: <https://rua.ua.es/dspace/handle/10045/49988>

Palabras clave: Aplicación web, Dispositivos móviles, Ionic, Servicios Rest.

Descripción

En el presente Trabajos Fin de Máster (TFM). Se pretende diseñar e implementar una aplicación para dispositivos móviles híbrida. Mediante reutilización de datos ya existentes y accesibles bajo filosofía Open Data, facilitará información de utilidad y en tiempo real al usuario del servicio de alquiler de bicicletas de BICICARD. Creado por el ITCL, BICICARD es un sistema automático de bicicletas públicas. Ofrece un servicio en el cual las bicicletas están disponibles para su uso compartido dentro de las ciudades con los siguientes objetivos. Implantar un sistema de préstamo de bicicletas público, asequible, práctico y fácil de utilizar. Fomentar el uso de la bicicleta como alternativa de transporte habitual, limpio y saludable. Complementar la red de transporte público en las ciudades.³

Título: Comparativa de frameworks para el desarrollo de aplicaciones con PHP.

Autores: Aguirre Espinoza Belen Cindy.

Lugar: Universidad de Azuay, Cuenca, Ecuador.

² ESCOBAR, Salgado. Estudio de la eficiencia de los frameworks híbridos y nativos en el desarrollo de aplicaciones móviles, basado en Benchmark para el Consorcio Informega. Trabajo de grado: Universidad de las fuerzas armadas ESPE. 2018. 5 p.

³ CANO INSA, Esteban. BiciPark: web app con Ionic y acceso a servicios REST. Trabajo para fin de Master. España.: Universidad de alicante (2015). 12 p.

Fecha: 2012

Link: http://biblioteca.uazuay.edu.ec/opac_css/index.php?lvl=author_see&id=9563

Palabras clave: CakePHP, Calidad, Framework, Informática, Prado, Software, Zend.

Descripción:

Los desarrolladores eligen frameworks al azar, generando dificultades al proyecto, es por esta razón que se pretende comparar los frameworks más populares. Mi propuesta se basa en escoger la mejor herramienta, mediante la implementación de un caso práctico, evitando que los programadores pierdan tiempo identificando tediosos detalles de bajo nivel, en vez de requerimientos de software. Para la selección se utilizara modelos o métodos de calidad y en función del framework seleccionado generar aplicaciones uniformes que faciliten reutilizar código y promover buenas prácticas de desarrollo, como el uso de patrones y a su vez incremente las ganancias.⁴

2.2.2 Ámbito nacional

Título: Desarrollo de una aplicación prototipo para la localización de parqueaderos en la plataforma IOS.

Autores: Cortes Rojas, Miguel Dario.

Lugar: Universidad EAN, Bogotá, Colombia.

Fecha: 2013.

Link: <https://repository.ean.edu.co/bitstream/handle/10882/5991/RojasMiguel2013.pdf?sequence=3&isAllowed=y>

Palabras clave: IOS, PhoneGap, Aplicación móvil, Xcode, Georeferenciación, sistemas.

Resumen:

En la actualidad, la ubicación de un parqueadero en la ciudad de Bogotá no es tarea fácil, fenómeno generado por el atraso vial que caracteriza a la capital, sumado a la

⁴ AGUIRRE ESPINOZA BELEN, Cindy. Comparativa de frameworks para el desarrollo de aplicaciones con PHP. Trabajo de grado, Cuenca, Ecuador. Universidad de Azuay. 7 p.

pésima gestión de las obras para mejorar el estado de las vías, las facilidades para la adquisición de vehículo, los cambios de nomenclatura en las direcciones y las políticas establecidas para hacer uso del automóvil como medio de transporte las cuales, en ciertos horarios, generan congestiones vehiculares de grandes dimensiones entre otros aspectos. Los dispositivos móviles se presentan como la herramienta idónea para obtener de forma práctica la localización de un estacionamiento cercano. La constante evolución de las plataformas móviles y el boom de las “apps” encajan a la perfección en el contexto de esta situación.⁵

Título: Estudio comparativo de marcos de trabajo para el desarrollo de aplicaciones web de código abierto.

Autores: Guerrero Alarcón, Carlos Andrés.

Lugar: Politécnico Colombiano Jaime Isaza Cadavid, Medellín, Colombia.

Fecha: 2009

Link: <https://revistas.uis.edu.co/index.php/revistagti/article/view/808>

Palabras Clave: Comparativo, marcos de trabajo, web, código abierto, desarrollo, caracterización, criterios.

Resumen:

El trabajo realizado en la formulación y puesta en marcha de diversos proyectos, con soluciones informáticas, y la interacción con diversos equipos de trabajo, arroja un sin número de oportunidades de mejoramiento, una de ellas, es la forma de hacer aplicaciones informáticas. Los marcos de trabajo, tema de este artículo son una rueda importante en este engranaje del proceso de desarrollo software. Por tanto, con el presente artículo se pretende ampliar la visión acerca de las herramientas disponibles en el mercado para construir aplicativos software.⁶

⁵ CORTES ROJAS, Dario Contreras. Desarrollo de una aplicación prototipo para la localización de parqueaderos en la plataforma IOS. Trabajo de grado, Bogotá, Colombia. Universidad EAN. 2013. 13 p.

⁶ GUERRERO ALARCON, Carlos Andrés. Estudio comparativo de marcos de trabajo para el desarrollo de aplicaciones web de código abierto. Revista Gerencia Tecnológica Informática [en línea]. 10 de junio de 2009 [citado el 18 de julio de 2018]. Disponible en Internet: <https://revistas.uis.edu.co/index.php/revistagti/article/view/808>.

Título: Desarrollo de apps: Un estudio comparativo entre frameworks libres y privativos.

Autores: Argote Puetaman, Ivan Mauricio. Jimenez Toledo, Robinson Andres. Ceballos David, Castro Enrique Miguel.

Lugar: Universidad Mariana, Pasto, Nariño.

Fecha: 2016

Link: <http://ojseditorialumariana.com/index.php/BoletinInformativoCEI/article/view/1083>

Palabras Clave: Framework, aplicaciones móviles, libres, privativos, software.

Resumen:

El desarrollo de aplicaciones móviles se ha venido incrementando en la actualidad, dado el gran número de dispositivos que existen, razón por la cual los desarrolladores de software tipo Apps, tienen a su disposición diferentes frameworks que pueden utilizar en la fase de construcción. A nivel mundial existen investigaciones que dan a conocer los mejores frameworks para el desarrollo de aplicaciones móviles, así como las ventajas y desventajas que ofrece cada uno, desde una perspectiva teórica. En este sentido, el presente estudio, además de permitir dimensiones comparativas entre frameworks, realiza la construcción de una aplicación móvil, con la intencionalidad de indicar el que más contribuya en el proceso y producción de software.⁷

⁷ ARGOTE, Iván., JIMÉNEZ, Andrés., CASTRO, Danny., CEBALLOS, David. Desarrollo de apps: un estudio comparativo entre frameworks libres y privativos. Universidad mariana boletín informativo CEI [en línea]. 10 de junio de 2016 [citado el 18 de julio de 2018]. Disponible en Internet: <http://ojseditorialumariana.com/index.php/BoletinInformativoCEI/article/view/1083>.

2.2.3 Ámbito local

Título: Estudio comparativo de sistemas integrados de gestión bibliotecaria de software libre más populares en Colombia.

Autores: Escaño Turizo, Hugo Alejandro.

Lugar: Universidad de Pamplona, Pamplona, Colombia.

Fecha: 2016

Link: Sistema Bibliotecario de la Universidad de Pamplona.
<http://serviciosacademicos.unipamplona.edu.co/prestamo/>

Palabras Clave: Software libre, comparativo, sistemas, biblioteca

Resumen:

Los software para la gestión bibliotecaria existen desde los años 80 pero muchos de estos se han ido descontinuando, ya que, no han seguido evolucionando a medida que las bibliotecas fueron ampliando sus actividades y reglamentos que las rigen, para determinar cuál de estos cumple o no para su implementación, es necesario observar cómo están actualmente en el mercado este tipo de sistemas, para así saber cuál es la inversión a realizar, afortunadamente en el mundo existen software bibliotecarios completos tanto de licencia comercial como de libre y ambos son muy prometedores prospectos; el caso de los comerciales en donde se pueden encontrar a SIABUC, Primo, OCLC entre otros, todos estos muy completos pero con un costo monetario bastante considerable, mientras que la opción más económica, de fácil implementación y que también son competencia directa a los comerciales son los sistemas libres como Koha, ABCD, PMB, Openbiblio; que son muy competitivos y constantemente modificados para mejor manejo de las bibliotecas y no solo eso, también permitiendo una mejor adaptación de este sistema a la propia biblioteca con solo programar un poco o tan solo configurándola correctamente.⁸

⁸ ESCAÑO TURIZO, Hugo Alejandro. Estudio comparativo de sistemas integrados de gestión bibliotecaria de software libre más populares en Colombia. Trabajo de Grado, Pamplona, Colombia. Universidad de Pamplona. 2016. 8 p.

Título: Comparativo de Frameworks web MVC open source en java, para la migración del administrador vortal en el CIADTI.

Autores: Oscar Alberto Bolaño Narváez.

Lugar: Universidad de Pamplona, Pamplona, Norte de Santander.

Link: Sistema Bibliotecario de la Universidad de Pamplona.
<http://serviciosacademicos.unipamplona.edu.co/prestamo/>

Descripción

Debido al constante crecimiento en el ámbito de las tecnologías enfocadas hacia la evolución del desarrollo web, se observa que la creación de sistemas software es indispensable para casi cualquier organización y si se mira el desarrollo tradicional de este tipo de sistemas, es indispensable la normalización de los datos sin importar la forma como se manipulen los mismos, el problema que se maneja desarrollando de manera tradicional es que un sistema es más que solo datos, se debe considerar también código fuente, librerías, archivos de configuración etc., y todo esto depende del individuo que esté desarrollando el sistema de información, lo cual, hace más complicado el proceso. Un Framework web MVC es la solución a este inconveniente debido a que se encarga de automatizar muchos procesos en el desarrollo, sin embargo, la gran variedad de Frameworks web MVC existentes, se ha convertido en un factor de riesgo para los desarrollos en la medida que una mala selección, implica probablemente el fracaso del proyecto o en el mejor de los casos puede ser un obstáculo para el futuro mantenimiento de la aplicación por lo que el proceso de selección del Framework adecuado para el tipo de sistema a desarrollar se convierte en un factor determinante.⁹

⁹ NARVAEZ, Bolaño. Comparativo de Frameworks web MVC open source en java, para la migración del administrador vortal en el CIADTI. Trabajo de grado, Pamplona Colombia. Universidad de Pamplona. (2017). 8 p.

2.3 MARCO CONCEPTUAL

Aplicaciones móviles: las aplicaciones móviles son pequeños programas o aplicaciones informáticas que realizan funciones para las que han sido diseñadas: juegos, calculadoras de todo tipo, directorios, glosarios, programas formativos, presentaciones o catálogos de empresas, etc.

Framework: un framework es un conjunto de herramientas que permiten trabajar de forma sencilla, que permite ordenar y estructurar adecuadamente el código de un desarrollo, permite tener todo más organizado y en general la gran mayoría permite la reutilización de código.

Estudio comparativo: un comparativo aparece como un proceso repetitivo y constante en nuestro mundo usándose en cualquier contexto, por ejemplo para propósitos de este trabajo, si queremos emplear framework de desarrollo híbrido, usualmente se preguntaría, ¿cuáles son sus características?, ¿Qué ventajas tiene sobre otro framework usado para el mismo fin? , ¿Dentro de mis necesidades cual es el que mayores beneficios tiene según propósitos?, todas estas preguntas se deben responder y el proceso de comparar es el más adecuado, ya que teniendo el conocimiento necesario de los objetos a comparar podemos llegar a bosquejar una solución viable a nuestros problemas, porque de allí surge la necesidad de un comparativo, de un problema, de una incógnita, un estudio comparativo tiene como finalidad entonces, el estudio de objetos similares pero que se diferencian entre sí en algunos aspectos.

SDK: (Software Development Kit) como sus siglas lo indican es un kit de desarrollo de software, un conjunto de herramientas que permiten el desarrollo de aplicaciones móviles.

JSON: (JavaScript Object Notation) notación de objeto de JavaScript, es un formato de texto ligero similar a la notación de objetos usada por JavaScript.

Librería o Biblioteca: es un conjunto de funcionalidades desarrolladas en un lenguaje de programación, que integradas con un proyecto de desarrollo permiten ampliar las capacidades funcionales del software.

NPM: gestor de paquetes por excelencia en JavaScript, “gracias a él podremos tener cualquier librería disponible con solo una línea de código, npm nos ayudará a administrar nuestros módulos, distribuir paquetes y agregar dependencias de una manera sencilla”.¹⁰

URL: (Uniform Resource Locator) localizador uniforme de recursos, es una secuencia de caracteres con una estructura específica que permite la identificación de recursos en Internet.

HTML: (HyperText Markup Language) lenguaje de marcas de etiquetas que permite el desarrollo de páginas y sitios web, pero también de aplicaciones móviles, aplicaciones para televisores y gran cantidad de dispositivos que se conectan a Internet.

CSS: (Cascading Style Sheets) hojas de estilo en cascada, es un lenguaje de diseño que permite dar estilos y mejorar la presentación de un documento escrito mediante un lenguaje de marcado.

JavaScript: lenguaje de programación que funciona en los navegadores web y recientemente en los servidores, procesadores y otros dispositivos, forma parte del estándar para el desarrollo de tecnologías en la web junto a HTML y CSS.

2.4 MARCO TERICO

2.4.1 Software

Software no tiene una traducción exacta en el idioma español, pero debido a su universalidad y gran uso la real academia española la define como “Conjunto de

¹⁰ B. G, Alexander, ¿Qué es npm?, devcode.la [En Línea]. 20 de Abril de 2017 [revisado el 06 de Diciembre de 2018]. Disponible en Internet: <https://devcode.la/blog/que-es-npm/>

programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora”¹¹.

El software abarca todas las aplicaciones informáticas de cualquier tipo, y es desarrollado mediante distintos lenguajes de programación de acuerdo a las necesidades y el contexto. El software tiene como finalidad dar solución a problemas específicos.

Las aplicaciones móviles se encuentran clasificadas dentro del software de aplicación el cual “permite a los usuarios llevar a cabo una o varias tareas específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido”¹², y como software que son las aplicaciones móviles están enmarcadas dentro de un conjunto de técnicas para la resolución de tareas específicas, a este conjunto de técnicas se le conoce como ingeniería de software.

El software ha crecido de manera desmesurada, en la actualidad se cuenta con software para cada una de las necesidades que se puedan imaginar, y como era de esperarse este empezó a usarse con fines lucrativos. A raíz de esto surge la clasificación más general del software, que divide el software en software libre y software privativo.

2.4.1.1 Software privativo

El software privativo es aquel en el cual el código fuente no está a disposición de nadie en la comunidad y solo es propiedad del desarrollador o desarrolladores del mismo. Se le considera el termino opuesto a el software libre, y comenzó a llamarse privativo en el año 2003, cuando Richard Stallman fundador del movimiento software libre en una de sus conferencias, ya que anteriormente se le conocía como

¹¹Diccionario de la Real Academia Española (RAE). Definición de software. Real Academia Española [en línea]. 16 de julio de 2018 [revisado el 16 de julio de 2018]. Disponible en Internet: <http://dle.rae.es/?id=YErIG2H>

¹²Wikipedia. Software. Wikipedia [en línea]. 20 marzo de 2018 [revisado el 16 de julio de 2018]. Disponible en Internet: https://es.wikipedia.org/wiki/Software#Car%C3%A1cter_evolutivo_del_software

propietario, alegando que privativo se ajustaba más a la filosofía que el manejaba debido a que privaba y restringía los derechos y libertades sobre el código fuente.

En otras palabras, el software privativo es el que no nos concede total control sobre él, sino que cuenta con restricciones parciales o totales, en donde de forma unilateral el desarrollador del software tiene todos los privilegios sobre el mismo.

2.4.1.2 Software libre

“Software libre es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software.”¹³

Lo que se quiere expresar con este concepto, es que el software el libre no gratuito, libre de cualquier tipo de control sobre él o sobre las personas que se relacionan con él, dando así al software libre un carácter mucho más fuerte y representativo.

Para que un software este dentro de las características de libre debe cumplir con 4 libertades esenciales:

- **Libertad 0:** Libertad de ejecutar el programa como se desee, con cualquier propósito
- **Libertad 1:** Libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera. El acceso al código fuente es una condición necesaria para ello.
- **Libertad 2:** Libertad de redistribuir copias para ayudar a otros.
- **Libertad 3:** libertad de distribuir copias de sus versiones modificadas a terceros. Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.¹⁴

¹³GNU. ¿Qué es el software libre? El sistema operativo GNU [en línea]. 20 de julio de 2018 [revisado el 20 de julio de 2018]. Disponible en Internet: <https://www.gnu.org/philosophy/free-sw.es.html>

¹⁴Ibid.

2.4.1.3 ¿Software libre, igual a software open source?

El software open source nació de un desacuerdo con los objetivos del software libre, por lo que un aparte de la comunidad se dividió y dio inicio a la campaña para promover el software open source, ambos describen la misma categoría de software, pero su diferencia fundamental es en los valores que los respaldan.

La diferencia radica más en sus bases, en su filosofía, en sus valores. Para el software libre, para sus defensores es un movimiento social, es algo ético, que pone el concepto de libertad sobre lo demás y no admite al software que no es libre, para ellos el software que no es libre es un problema social, y por lo tanto se debe dejar de usar y empezar a usar software libre.

Mientras que el software de código abierto es una metodología de programación, que se centra en cómo mejorar el software y hacerlo competitivo con los otros software, tratando de hacerlo popular y exitoso, por lo que el software libre no puede aceptar las bases del código abierto, porque para ellos sus objetivos van en contra de la libertad.

2.4.2 Framework

Un framework “es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar”¹⁵ es decir, es un esquema que se usara para el desarrollo de una o varias aplicaciones y para su futuro mantenimiento.

Este esquema cuenta con componentes genéricos los cuales podemos usar para desarrollar una aplicación completa, para hacer del proceso de desarrollo algo más simple, brindando agregados en seguridad, arquitectura, diseño, etc.

Entonces ¿el porqué de los framework?, simplemente porque “se utilizan para facilitar el desarrollo de software, ya que permiten a los programadores dar más

¹⁵ www.wikipedia.org. Framework. Wikipedia [en línea]. 01 de mayo de 2017 [revisado el 21 de julio de 2018]. Disponible en Internet: <https://es.wikipedia.org/wiki/Framework>

tiempo para cumplir los requisitos del proyecto real en lugar de prestar atención a los detalles de bajo nivel del proyecto. Por lo tanto, ayudan a reducir el tiempo total de desarrollo.”¹⁶ No solo esto es tanta su utilidad que podemos ampliar la funcionalidad de nuestros proyectos y permitiendo llevar una estructura organizada de desarrollo, evitar el llamado código espagueti.

2.4.3 Aplicaciones móviles

Las aplicaciones móviles son el resultado de un proceso de desarrollo de software sobre dispositivos móviles (Smartphone), con el fin de generar un conjunto de funcionalidades que permitan llevar a cabo tareas de diferente índole, por ejemplo; profesional, de ocio, educativas, de acceso a servicios, etc.

El uso de aplicaciones móviles se vuelve una necesidad para las empresas, debido a que brinda comodidad y una experiencia eficiente a los usuarios. Su uso ya es extendido, pero muchas veces su desarrollo se torna complicado. Por lo que desarrolladores optan usar los marcos de trabajo conocidos como framework, los cuales logran hacer del proceso de desarrollo de una aplicación web algo mucho más simple.

Las aplicaciones móviles se dividen en 3, de acuerdo al tipo de diseño y desarrollo los cuales son: aplicaciones híbridas, aplicaciones nativas y aplicaciones web.

2.4.3.1 Origen de las aplicaciones móviles

No se cuenta con una fecha exacta en lo que concierne al surgimiento de las aplicaciones móviles como tal, sin embargo, se pueden situar a las primeras aplicaciones móviles en los años 90, donde los teléfonos eran conocidos como teléfonos básicos, los cuales no contaban con pantallas táctiles y eran de pantallas reducidas.

¹⁶ [www.postparaprogramadores.com](https://postparaprogramadores.com). Los tipos de framework y sus funcionalidades. Para programadores [en línea]. 21 de julio de 2018 [revisado el 2018]. Disponible en Internet: [https://postparaprogramadores.com/tipos-de-framework/#Tipos de Framework](https://postparaprogramadores.com/tipos-de-framework/#Tipos_de_Framework)

Pero se puede decir con certeza que las primeras aplicaciones móviles estaban destinadas al entretenimiento y la comodidad como lo son los videojuegos o los calendarios y recordatorios.

La primera aplicación de juegos instalada fue en el año de 1994 y fue el popular juego tetris, de ahí tres años más tarde Nokia lanzó el juego conocido como Snake el cual llegó a más de 350 millones de dispositivos móviles por todo el mundo.

Ya en el año 2000 con el nacimiento del protocolo de aplicaciones inalámbricas WAP permitió una mayor capacidad para la descarga de juegos y aplicativos. Pero el verdadero auge de las aplicaciones móviles se dio a partir del año 2008 con el lanzamiento de la tienda de aplicaciones App Store, por parte de la empresa de tecnología Apple donde se publicó la primera aplicación móvil.

Inmediatamente Google lanzó su primer SDK y su tienda de aplicaciones Google market, la cual en unos años se convertiría en Play Store la tienda de aplicaciones con mayor demanda de aplicaciones móviles ofrecidas.

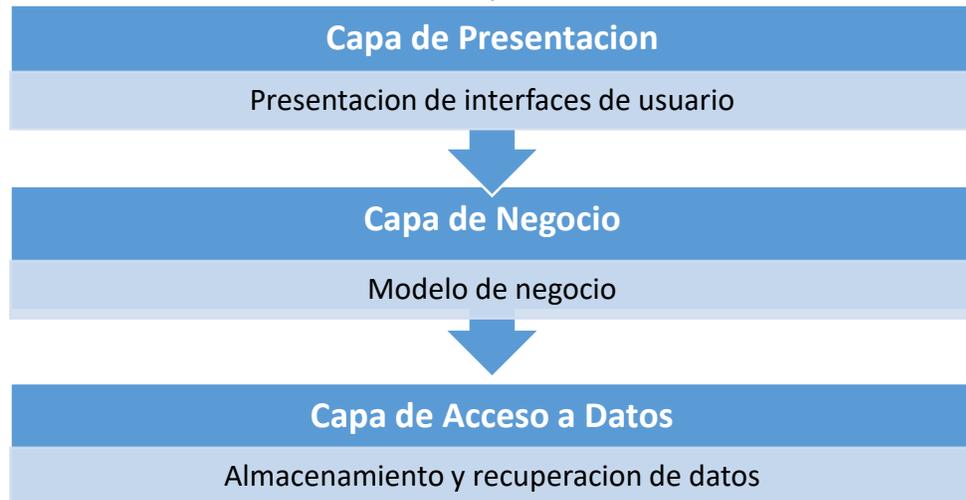
2.4.4 Desarrollo back-end y front-end

El desarrollo de software es la construcción de software para dar solución a una necesidad o un problema. Si se observa detenidamente la anterior definición, no parece ser algo complicado de entender, pero con el paso de los años se ha convertido en un proceso de alta complejidad, hasta el punto que ciencias como la ingeniería de software están dedicadas a este único fin.

Esta complejidad hizo que surgieran diferentes tipos de metodologías, marcos de trabajo, modelos y gran tipo de técnicas para organizar y simplificar la construcción de software. Entre las técnicas que nacieron, se encuentra la arquitectura basada en N capas, esta distribuye las responsabilidades del desarrollo de software en funciones las cuales suelen encontrarse en diferentes computadoras.

La distribución en 3 capas es la más usada, y distribuye el proceso de construcción de software en, la capa de presentación, en la capa de negocio y la capa de acceso a los datos.

Ilustración 1 - Capas de Desarrollo



Ya sabiendo lo anterior se puede hablar de lo que es el front-end y el back-end, términos que se refieren a la separación de intereses entre la capa de presentación y las capas de acceso a datos y de negocio, respectivamente, por lo general estos términos son usados cuando el desarrollo se encuentra distribuido o usa tecnologías de la web.

2.4.4.1 Front-end

El front-end es la parte frontal del desarrollo de software, es decir, lo que refiere a estilos, presentación, organización visual del software, colores, dinamismo y en general las interacciones del usuario con el software, lo que puede apreciar y con lo que puede interactuar.

Entonces el front-end es más artístico y creativo, y se suele pensar que es “fácil” y no implica ningún tipo de lógica, pero es algo que está lejos de ser cierto.

Se trabaja con lenguajes que funcionan en la capa de presentación, lo cual hace que se otorgue dinamismo visual al software. El lenguaje por excelencia en el front-end es JavaScript.

2.4.4.2 Back-end

El back-end se dedica a la parte lógica de un software, es decir es el encargado de que todo funcione como se debería. Toma los datos, los procesa y envía donde se necesitan, realiza la conexión con el servidor, conexión con base de datos, organiza la estructura del proyecto, seguridad, entre otras tareas.

Se puede decir que a diferencia del front-end uso solo lenguajes de programación, entre los cuales se pueden mencionar Java, Python, C, JavaScript. Se suelen usar herramientas como compiladores, debuggeadores, gestores de bases de datos, editores especializados y algunas otras cosas más.

Alrededor del mundo gran cantidad de organizaciones y desarrolladores de software se especializan en una de las dos áreas, y framework basan su forma de trabajar teniendo en cuenta a que área va a dirigir sus esfuerzos.

El presente trabajo centrara esfuerzos en framework que funcionan en la parte del front-end, pero que al trabajar sobre dispositivos móviles tienen acceso a funcionalidades propias del dispositivo y del sistema operativo, sin dejar de lado el back-end.

2.4.4.3 API Rest

Es un medio para la interacción de sistemas independientes mediante el uso del protocolo HTTP. Este define un conjunto de funciones para realizar peticiones y recibir respuestas a través de HTTP mediante métodos como GET, POST, PUT, DELETE sobre recursos los cuales se identifican mediante una URI. Las respuestas en un API Rest usualmente son hechas en formato JSON, pero también podría ser en XML.

Este es un enfoque de desarrollo mediante el uso de servicios web y sus siglas significan Transferencia de Estado Representacional, "Buscando una definición sencilla, REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como

XML y JSON. Es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP (Simple Object Access Protocol).”¹⁷

Características de un API Rest

- Separación clara entre el cliente y el servidor.
- Una separación entre el cliente y el servidor permite la fácil migración de códigos a distintos servidores por lo que proporciona escalabilidad.
- Al usar el protocolo HTTP los desarrollos se hacen independientes del lenguaje de programación a usar ya sea en el cliente o en el servidor.

2.4.5 Metodologías usadas en el desarrollo móvil

Las aplicaciones móviles son un tipo de software que se ejecuta sobre dispositivos móviles, por lo que el uso de metodologías de desarrollo de software tradicionales es aceptado, teniendo en cuenta aspectos específicos del hardware y software del dispositivo, como lo son la gran variedad de tamaños de pantalla, distintos sistemas operativos, portabilidad, actualizaciones constantes, rápida salida al mercado y otras características enmarcadas en este tipo de aplicaciones.

El desarrollo de aplicaciones para proveer servicios móviles, difiere del desarrollo de software tradicional en muchos aspectos, lo que provoca que las metodologías usadas para estos entornos móviles, también difieran de las del software clásico.¹⁸

¹⁷ BBVAOPEN4U, API REST : Que es y cuáles son sus ventajas a la hora de desarrollo de proyectos [en línea], 23 de marzo de 2016 [revisado el 15 de Noviembre de 2018], Disponible en Internet:

<https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>

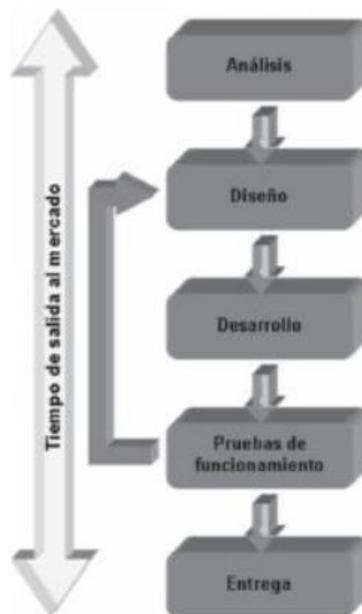
¹⁸ V, Rahimian, y R, Ramsin. Designing an agile methodology for mobile software development: A hybrid method engineering approach. IEEE explore digital library [en línea]. 03 de Junio de 2008 [revisado el 09 de agosto de 2018]. Disponible en Internet: <https://ieeexplore.ieee.org/document/4632123/>

Por lo general el proceso de diseño y desarrollo de aplicaciones cuenta con 5 etapas: análisis, diseño, desarrollo, pruebas y entrega, las cuales hacen parte de una metodología la cual es el resultado de un proceso de investigación hecho por la Universidad de Magdalena, Colombia.¹⁹

Algo a tener en cuenta es que una metodología de desarrollo es un marco de trabajo el cual permite la organización, planificación y desarrollo de software, pero los desarrollos en la actualidad no basan completamente su proceso en una metodología, por lo que los pasos y las etapas en el desarrollo pueden variar en función a las necesidades del equipo de trabajo y de los clientes de software.

En la siguiente figura se pueden apreciar las 5 fases para el diseño y desarrollo de la aplicación además de su enfoque de desarrollo, que claramente es repetitivo.

Ilustración 2 - Fases metodología de desarrollo móvil



Fuente <https://revistas.udistrital.edu.co/ojs/index.php/Tecnura/article/view/6972/8646>

¹⁹ G. M, Maira Cecilia, C. A, Luis Leonardo y D, Byron Medina. Metodología para el desarrollo de aplicaciones móviles. Trabajo de Grado, Colombia: Universidad de Magdalena. (2013). 5 p.

2.4.5.1 Análisis

El análisis es la fase en donde como lo dice su nombre se analizan las peticiones y los requerimientos de los clientes o las personas hacia los cuales está destinado el desarrollo del software.

A esto se le conoce con el nombre de obtención de requerimientos, los cuales son indispensables para que el desarrollo de software se acerque lo más posible a lo que el cliente espera, ya que muchas veces estos requisitos pueden estar incompletos, ambiguos e incluso contradictorios.

Según el artículo ¿Es necesaria la ingeniería de software? “Usualmente el cliente/usuario tiene una visión incompleta/inexacta de lo que necesita y es necesario ayudarlo para obtener la visión completa de los requerimientos. El contenido de comunicación en esta etapa es muy intenso ya que el objetivo es eliminar la ambigüedad en la medida de lo posible.”²⁰

2.4.5.1.1 Técnicas para la obtención de requerimientos

Se cuenta con una cantidad considerable de técnicas para la obtención de los requerimientos, los siguientes son los más conocidos:

- **Análisis de documentación:** consiste en obtener información acerca de requerimientos a partir de documentos ya elaborados.
- **Observación:** consiste en estudiar el entorno de trabajo de los interesados en el desarrollo.
- **Entrevistas:** reunión con los interesados.
- **Encuestas y cuestionarios:** se deben enfocar en los objetivos principales del desarrollo.
- **Mesas de trabajo:** son reuniones con los interesados y con todo el equipo de trabajo y de desarrollo.

²⁰ Xavi. Las cinco etapas de la ingeniería de software. Proyectosguerrilla [en línea]. 11 de noviembre de 2014 [revisado el 10 de agosto de 2018]. Disponible en Internet: <http://proyectosguerrilla.com/blog/author/xavi/>

- **Tormenta de ideas:** se reúne el equipo de trabajo con el fin de obtener la mayor cantidad de ideas posible.
- **Historias de usuario:** es una representación de un requisito en lenguaje común, y que es escrito por el usuario.

2.4.5.2 Diseño

“El objetivo de esta etapa es plasmar el pensamiento de la solución mediante diagramas o esquemas, considerando la mejor alternativa”²¹, es especificar como funcionara de forma general el software, diseñando componentes que describan la interacción de entre los elementos que conforman el software.

Esta etapa se divide en tres fases las cuales son:

- **Diseño del boceto:** Esto permite representar visualmente y de una forma muy sencilla y esquemática la estructura de una aplicación móvil, definiendo el contenido y la posición de los bloques que conforman una aplicación, describiendo de manera visual funcionalidades. Para el presente proyecto se usarán los Wireframe.
- **Definición del escenario:** “Las aplicaciones móviles se pueden diseñar para ejecutarse en diferentes escenarios, dependiendo del sistema de conexión y sincronización con el servidor o aplicación central; el proceso de sincronización se realiza para insertar, modificar o borrar información.”²² Los escenarios que se suelen tomar en cuenta son con acceso a Internet y sin acceso a Internet.
- **Estructurar el software:** En esta fase lo ideal es usar diagramas para estructurar el funcionamiento del software, por lo general se recomienda usar diagramas de Modelado de Lenguaje Unificado UML.

²¹ G. M, Maira Cecilia, C. A, Luis Leonardo y D, Byron Medina. OP. CIT. p 7.

²² Ibid. p. 8

Ilustración 3 - Diagramas UML para modelado



Fuente <https://revistas.udistrital.edu.co/ojs/index.php/Tecnura/article/view/6972/8646>

2.4.5.3 Desarrollo

En la fase de desarrollo se implementa el diseño, es decir, se traduce el diseño a código y por primera vez se pueden tener resultados tangibles de lo que es el software. En esta etapa se realizan las siguientes actividades:

- **Codificar:** Se expresa la información y los requerimientos de los usuarios en la etapa de diseño, en el lenguaje de programación seleccionado.
- **Documentar el código:** A medida que se codifica se debe documentar cada una de las partes del código, esta parte es una de las más tediosas, por lo que muchos desarrollos no cuentan con documentación.
- **Manual de ayudas:** Deben existir una serie de ayudas que informen de manera didáctica lo que puede hacer el usuario con la aplicación, estas ayudas deben ser codificadas en el mismo lenguaje de programación e integrada en la interfaz de aplicación para visualizarlas en el móvil.

2.4.5.4 Pruebas de funcionamiento

Esta fase consiste en comprobar que el software responda y realice correctamente las tareas para las cuales fue desarrollado, estas pruebas se hacen en diferentes escenarios y condiciones, descartando así cualquier tipo de fallas.

Las pruebas cuentan con una particularidad, la cual es que estas se pueden llevar a cabo en cualquier parte del proceso de desarrollo de software, dependiendo de su tipo se llevan a cabo en momentos diferentes.

2.4.5.4.1 Tipos de pruebas

Las pruebas las hay de muchos tipos, pero se pueden resumir las más conocidas y usadas dentro de los proyectos de software, en los siguientes incisos:

- **Pruebas unitarias:** Su objetivo es la de comprobar el correcto funcionamiento de una unidad de código, esta unidad de código está definida de acuerdo al estilo de programación con el cual se haga el desarrollo.
- **Pruebas de integración:** Son las pruebas que se realizan una vez realizadas las pruebas unitarias, estas como su nombre lo dice consiste en integrar las unidades del software y verificar el funcionamiento de las mismas en conjunto.
- **Pruebas del sistema:** “Tienen como objetivo ejercitar profundamente el sistema comprobando la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica.”²³
- **Pruebas de implantación:** Su objetivo ya el comprobar el correcto funcionamiento del sistema una vez implantado en su entorno real.
- **Pruebas de aceptación:** El objetivo de las pruebas de aceptación es permitir al usuario de dicho sistema determinar su aceptación, en cuanto a las funcionalidades del sistema.

²³ CILLERO, Manuel. Pruebas en ingeniería de software. *manuel.cillero.es* mi circunstancia similar [en línea]. 12 de Julio de 2017 [revisado el 12 de agosto 2018]. Disponible en Internet: <https://manuel.cillero.es/doc/metrica-3/tecnicas/pruebas/sistema/>

- **Pruebas de regresión:** Su objetivo es comprobar que los cambios sobre un componente no afecten el correcto funcionamiento del sistema en general, como por ejemplo con el surgimiento de errores.

2.4.5.5 Entrega

Una vez terminada la depuración de la aplicación y atendidos todos los requerimientos, se puede dar por finalizado el proyecto, por lo que se entrega el código fuente, manuales de uso, manuales de usuario y el ejecutable.

Una vez terminadas las formalidades de la entrega se da inicio a la fase de distribución en la cual la aplicación será comercializada. Existen diferentes formas de comercialización, la más común es en las tiendas de aplicaciones disponibles de acuerdo a cada sistema operativo, pero también en los sitios web oficiales de los distribuidores de las aplicaciones.

2.4.5.5.1 Tiendas de aplicaciones móviles

Existen diferentes tipos de tiendas, las cuales por lo general están ligadas al sistema operativo sobre el cual está desarrollada la aplicación. Las tiendas organizan y categorizan las aplicaciones, a fin de optimizar el proceso de distribución.

Cada tienda tiene normas diferentes de retribución y publicación, de acuerdo a las necesidades, es decisión del desarrollador escoger la tienda de aplicaciones sobre la cual publicar. Entre las tiendas de aplicaciones más populares están:

- Google Play.
- App Store.
- Windows Store.
- BlackBerry World.
- Amazon Appstore.
- F-Droid.

2.4.6 Opciones del desarrollo móvil

Se cuenta con tres tipos de desarrollos móviles teniendo en cuenta el diseño y la forma de desarrollo, las aplicaciones móviles nativas, las aplicaciones móviles híbridas y las aplicaciones móviles web. A continuación, se dará explicación de cada uno de los tipos de aplicaciones, sus características y se hará énfasis en las aplicaciones móviles híbridas ya que en ellas es que se basa el actual proyecto.

2.4.6.1 Aplicaciones móviles nativas

Las aplicaciones nativas son las que se desarrollan de manera específica para cada sistema operativo, de manera 100 % compatible con cada uno de los sistemas operativos respectivos. Esto permite mayor rendimiento y acceso a todas las funcionalidades y características del dispositivo, pero el hecho que estas se desarrollen en un lenguaje que es propio del sistema operativo supone múltiples desarrollos, uno por cada sistema operativo sobre el cual se desee trabajar, generando así mayores tiempos y costos de desarrollo de aplicaciones.

Lo anterior no quiere decir que sea una mala opción, solo que se deben tener siempre presente los requerimientos del desarrollo para poder tomar la mejor decisión en cuanto a aplicaciones. Este tipo de desarrollo se usa cuando las aplicaciones van destinadas a consumir gran cantidad de recursos del sistema operativo, por lo que un desarrollo en el lenguaje nativo del dispositivo ayuda a mejorar la experiencia de usuario, en caso que la prioridad sea el rendimiento.

Sus principales características son:

- **Manejo multitáctil:** aplicaciones que soportan todo tipo de eventos táctiles, multi toque, mantener el toque etc.
- **Componentes incorporados:** los dispositivos móviles cuentan con componentes nativos a los cuales se puede acceder “fácilmente”, como la cámara, GPS, almacenamiento encriptado, etc.
- **Gráficos rápidos y animaciones fluidas:** provee alta eficiencia y los mejores resultados en la optimización de gráficos y animaciones.

- **Alta confiabilidad:** se tiene un gran control sobre quienes acceden a tu app para evitar usos indeseados de la aplicación.
- **Experiencia de usuario y familiaridad:** los usuarios están acostumbrados a cierto tipo de interfaces, por lo que el acceso para el promedio de personas es más sencillo y familiar.
- **Documentación:** proporciona gran cantidad de documentación oficial, además de una gran comunidad y gran cantidad de recursos de apoyo como libros, revistas, repositorios.

Algunas desventajas para este tipo de desarrollos según la comunidad son:

- **Sistemas restringidos:** el paso de una aplicación móvil de un sistema a otro está restringido por el lenguaje del sistema operativo por lo que, por lo que proporcionar una aplicación a diferentes sistemas operativos es una tarea que se hace tediosa y complicada.
- **Dificultad de desarrollo:** el proceso de aprendizaje conlleva tiempo de estudio, y practica por lo que, aprender distintos lenguajes de programación de acuerdo a las necesidades resulta costoso y se ocupa más tiempo.
- **Alto nivel de experiencia requerido:** la dificultad de desarrollo es directamente proporcional el nivel de experiencia requerido, por lo que se puede decir que se necesita de gran experiencia para este tipo de desarrollos.
- **Entorno integrado de desarrollo:** tiene entornos integrados de desarrollos propios que les facilitan las tareas, si se tiene algún entorno de desarrollo preferido es posible que se tenga que dejar a un lado y acostumbrarse.

2.4.6.2 Aplicaciones móviles web

Son básicamente sitios web que se diseñaron de tal forma que cuando necesitan ser mostrados en pantallas más pequeñas con resoluciones bajas se ajustan a estas y simulan el diseño de una aplicación móvil, lo que es posible gracias al avance tecnológico y constante trabajo sobre los navegadores web. Tiene como

característica para comparar que no se necesita saber nada acerca del lenguaje nativo de los dispositivos móviles.

Sus principales características son:

- **No requiere descarga:** nunca se podrá encontrar dentro de una tienda de apps ya que es un sitio web optimizado para ser visualizado en navegadores móviles.
- **Fáciles de diseñar:** su complejidad técnica es mucho menor que la de una aplicación móvil por lo que encontrar a un desarrollador que haga un buen trabajo no es una tarea complicada.
- **Puede llegar a todos los usuarios:** ya que su funcionamiento no depende del lenguaje de programación del sistema operativo, puede llegar a todos los usuarios sin importar el dispositivo móvil que usen.

Algunas desventajas para este tipo de desarrollos según la comunidad son:

- **No puede servirse de las funciones del dispositivo:** es imposible para este tipo de aplicaciones el poder acceder a recursos propios del dispositivo como la cámara o el GPS ya que su desarrollo no usa ningún tipo de componente del dispositivo.
- **Carga continua:** cada vez que el usuario accede a la aplicación, carga cada vez su contenido por lo que el tiempo de acceso a la aplicación se hace costoso.
- **Pierde visibilidad:** al no encontrarse en el mismo sitio que las aplicaciones comunes como por ejemplo la Play Store pierde el alcance que pueda tener en cuanto a visibilidad.
- **Siempre conexión:** siempre se deberá tener una conexión a Internet para acceder a este tipo de aplicaciones.

2.4.6.3 Aplicaciones móviles híbridas

Entre las diferentes clases de framework se habla acerca de los usados para el desarrollo de aplicaciones híbridas, los cuales permiten el desarrollo de aplicaciones

programadas como un sitio web con HTML, CSS y JAVASCRIPT, y luego un framework que se encarga de adaptar la vista a la de un dispositivo móvil.

Este tipo de aplicaciones permiten hacer un desarrollo mucho más ágil, además de permitir ahorrar tiempo de mantenimiento debido que su fin último es el de un solo desarrollo que funcione en múltiples plataformas, siendo así un tipo de aplicaciones innovadoras que muchas empresas ahora solicitan, pero teniendo en cuenta que todo parte de las necesidades específicas de cada uno de los usuarios.

Sus principales características son:

- **Basado en tecnologías web:** su uso se basa en las tres tecnologías web más usadas HTML, CSS y JavaScript, por lo que las aplicaciones son mucho más sencillas de construir.
- **Menos costo:** su costo de desarrollo es mucho menor que el de una aplicación móvil nativa, además que su tiempo de desarrollo es mucho menor lo que implica menores costos para los usuarios que deseen este tipo de aplicaciones.
- **Multiplataforma:** al ser desarrollado en un lenguaje que no es propio del sistema operativo si no uno el cual es soportado por los navegadores internos de los dispositivos, estas aplicaciones pueden ser ejecutadas sobre varias plataformas, independientemente de sus características.
- **Uso de los recursos del sistema operativo:** a pesar de no estar desarrolladas mediante el lenguaje propio de los sistemas operativos, estas usan un intermediario para poder acceder a los recursos del sistema como la cámara, el almacenamiento, el GPS, etc.
- **Fácil distribución:** permite la distribución a través de las tiendas de aplicaciones de los diferentes sistemas operativos sin ningún problema.

Algunas de las desventajas que estas tienen frente a aplicaciones de otro tipo son:

- **Funciones Limitadas:** las funciones a las cuales puede acceder son limitadas debido a que por su forma de desarrollo no tienen acceso a todos

los recursos del sistema operativo.

- **Su rendimiento es menor:** al usar un intermediario y al no usar el lenguaje propio del sistema operativo del dispositivo, requiere algo más de tiempo en su ejecución.
- **Variaciones causadas por los diferentes dispositivos:** debido a la gran variedad de sistemas operativos, la apariencia puede cambiar dependiendo al dispositivo que se utilice, por lo que puede que no siempre se vea de la misma manera.

2.4.6.4 Aplicación híbrida contra nativa

El desarrollo de aplicaciones híbridas y nativas siempre han sido un tema de discusión y que ha causado controversia, debido a que el surgimiento de las aplicaciones web híbridas redujo de sobremanera el tiempo de desarrollo de las aplicaciones móviles, pero para poder hablar de esto se debe entender la principal diferencia entre una aplicación híbrida y una nativa, la cual no es como se programa , que por cierto depende de la tecnología, si no de la forma que se ejecutan las aplicaciones para el usuario final.

La aplicación nativa se ejecuta directamente sobre el sistema operativo, mientras que una aplicación híbrida requiere algo adicional como un contenedor o un navegador.

Ilustración 4 - App nativa vs App híbrida



Fuente: <https://platzi.com/blog/hibrida-o-nativa/>

2.4.6.5 Rechazo de las aplicaciones móviles híbridas

El nacimiento de este nuevo tipo de aplicaciones y rápido crecimiento llevo a pensar a muchas personas que las aplicaciones finalizadas tendrían problemas de rendimiento, y si, pero se habla de un año 2011 cuando todas las tecnologías para el desarrollo de estas aplicaciones estaban emergiendo, y como es lógico con el nacimiento de cada nueva tecnología vienen los problemas.

“Esta percepción se tuvo debido a que las primeras iteraciones de PhoneGap nos entregaron productos muy malos en las tiendas de aplicaciones.”²⁴

Pero en la actualidad los equipos y comunidades de desarrollo detrás de cada uno de los proyectos se encargan de solventar todos los problemas que van surgiendo y recibiendo todas las críticas como constructivas para mejorar la experiencia de desarrollo.

²⁴ ThespianArtist. Aplicación híbrida o nativa ¿Cuál es la mejor? Platzi [en línea]. 06 de junio de 2017 [revisado 15 de agosto de 2018]. Disponible en Internet: <https://platzi.com/blog/hibrida-o-nativa/>

2.4.6.6 Que elegir. Depende de las necesidades

Si se es desarrollador web, hacer un salto al desarrollo de aplicaciones móviles híbridas es mucho más sencillo, ya que todo lo aprendido con las tecnologías bases en el desarrollo web pueden ser usadas en la mayoría de framework de desarrollo móvil híbrido, incluso para gustos se tiene varias opciones de donde escoger.

Tabla 1 - Framework híbridos y sus tecnologías

Framework para el desarrollo de aplicaciones móviles híbridas	
Tecnologías en las que se basa	Nombre del framework
ReactJs, JavaScript, HTML, CSS	React Native
Angular2+, TypeScript, HTML, SASS	Ionic2+
XML, CSS, JavaScript	NativeScript
HTML, CSS, JavaScript	Framework7
HTML, CSS, JavaScript	Apache Cordova/PhoneGap

Se deben tener muy presentes los requerimientos de los desarrollos, para elegir la herramienta adecuada, porque habrá casos en que usar un desarrollo u otro sea la única opción. Por ejemplo, si lo que se necesita es que acceder a gran cantidad de recursos nativos del dispositivo y a las ultimas API del sistema operativo, la mejor opción es un desarrollo nativo, pero si no se tiene ninguna experiencia en el desarrollo de aplicaciones móviles, pero si en el ámbito web, y es necesario un producto de calidad en un periodo de tiempo corto, amigable con el usuario, un desarrollo híbrido es ideal.

Aunque con los avances tecnológicos y el paso del tiempo, la comunidad que respalda a los framework de desarrollo móvil híbrido, trabajan para hacer lo más parecido posible el comportamiento de los desarrollos híbridos al comportamiento de los desarrollos nativos.

2.4.7 Revisión de framework híbridos

Haciendo una revisión de los framework actuales a la fecha 27 de agosto de 2018, se documentó teniendo en cuenta dos aspectos, el primero es que cumplan con la

condición de ser framework bajo los estándares de software libre. El segundo aspecto es la popularidad en la comunidad lo que permite acotar la búsqueda, ya que a la fecha existen gran cantidad de framework para desarrollo móvil híbrido multiplataforma.

2.4.7.1 Ionic

Ionic es un SDK (kit de desarrollo de software) open source, “permite a los desarrolladores crear aplicaciones móviles de alta calidad utilizando tecnologías web familiares (HTML, CSS y JavaScript)”²⁵. Nació en el año 2013 creado por **Drifty Co**. Está basado en el framework Angular el cual le da gran soporte y flexibilidad para desarrollar y renderizar aplicaciones, y accede a las funciones nativas de los dispositivos mediante PhoneGap y Apache Cordova.

“Piense en Ionic como el marco de interfaz de usuario front-end que maneja todas las interacciones de apariencia y de interfaz de usuario que su aplicación necesita para ser convincente. Con soporte para una amplia gama de componentes móviles nativos comunes, animaciones ingeniosas y un hermoso diseño.”²⁶

Su lema es “*Write once, run everywhere*”, lo que significa que todo su código se podrá ejecutar en múltiples plataformas, la adaptación de sus componentes al diseño del sistema operativo es trabajo de Ionic, por lo que el desarrollador no tendrá que preocuparse, en su gran mayoría, cómo se verá la aplicación en dispositivos móviles distintos.

2.4.7.1.1 Historia

Fue creado por **Drifty Co** en 2013, productos anteriores de **Drifty Co** fueron diseñados para la construcción de interfaces del tipo arrastrar y soltar, las cuales fueron usadas por los usuarios de estas para desarrollar aplicativos móviles, debido

²⁵www.ionicframework.com. Acerca de, Documentación oficial. Ionic Framework [en línea]. 21 de agosto de 2018 [revisado el 21 agosto de 2018]. Disponible en Internet: <https://ionicframework.com/docs/>

²⁶www.ionicframework.com. Documentación oficial. Ionic Framework [en línea]. 21 de agosto de 2018 [revisado el 21 agosto de 2018]. Disponible en Internet: <https://ionicframework.com/docs/v1/guide/preface.html>

a esto la compañía decidió construir su propio marco de desarrollo especializado en el rendimiento de aplicaciones móviles usando las tecnologías web más usadas en su momento, construido sobre el marco de desarrollo Angular revoluciono permitiendo desarrollos más rápidos, mejor estructurados y sencillos.

Ya en el año 2016, el framework Angular sufrió una actualización que fue tan grande que ya no se consideraba el mismo framework, debido a esto Ionic también sufrió un cambio completo, en su forma de trabajar, pero permitió una mejor separación del código base en su núcleo, plugins nativos, Angular etc.

Ya Ionic siguiendo de la mano a Angular decidió adoptar un versionado semántico (SEMVER) y un ciclo de lanzamientos de cada seis meses, lo que según ellos permitirá una mejor evolución del framework, además de que no se tenga que reestructurar todo el código ya escrito, en su lugar se agregaran nuevas funcionalidades y características, pero manteniendo las anteriores.

“En la actualidad, Ionic es la plataforma de tecnología de desarrollo móvil multiplataforma más popular del mundo, impulsando tecnologías de rápido crecimiento para algunas de las compañías más grandes del mundo.”²⁷

2.4.7.1.2 Características

Alto rendimiento: no hay nada que desespere más que tener que esperar a que cargue una aplicación, por más que tengamos una conexión rápida a Internet, por lo que el framework está planteado para ser rápido y veloz.

Arquitectura robusta: la idea de Ionic al trabajar a base de Angular es que la aplicación se destaque por su funcionamiento, robusta y rica en funcionamiento para las aplicaciones.

²⁷ [www.ionicframework.com](https://ionicframework.com). Documentación oficial. Ionic Framework [en línea]. 21 de agosto de 2018 [revisado el 21 agosto de 2018]. Disponible en Internet: <https://ionicframework.com/about>

Centro Nativo: tiene un sistema que una vez se haya desarrollado el software se pueda compilar para diferentes sistemas operativos de forma sencilla, haciendo más fácil el trabajo cruzado multiplataforma.

Diseño Limpio: “El diseño de las aplicaciones es muy agradable e intuitivo, ofrecen una interfaz de producto llamativa que mejora la experiencia de los usuarios, con gran cantidad de elementos interactivos.”²⁸

IonicCLI: Ionic cuenta con una interfaz de línea de comando, permitiendo así crear de manera rápida y eficiente proyectos, y de la misma manera agregar nuevas funcionalidades y compilar los proyectos en tiempo real, agilizando los tiempos de desarrollo.

PWA: permite la creación a aplicaciones web progresivas, que son aplicaciones que se compartan como aplicaciones nativas. “Incrementan su funcionalidad, conforme las capacidades del dispositivo en el que se ejecutan, incrementan, de ahí la palabra progresiva, web, hace referencia a que se construyen utilizando estándares de desarrollo web y app es porque las Progressive Web Apps se comportan como aplicaciones web nativas, pero usan tecnologías web.”²⁹

Vista en tiempo real: Ionic ofrece una aplicación móvil gratuita que permite a los desarrolladores probar aplicaciones directamente en los dispositivos móviles de una manera más fácil y rápida, sin necesidad de instalar bibliotecas nativas complementarias.

Marco de trabajo de código abierto: licencia de software MIT. Esta es una licencia de software libre permisiva lo que significa que impone muy pocas limitaciones en la reutilización y por tanto posee una excelente Compatibilidad de licencia.

²⁸CARPINTERO, Irene. Ionic 3 un entorno para empezar a desarrollar apps moviles. Todoandroid [en línea]. 03 de septiembre de 2017 [revisado el 22 de agosto de 2018]. Disponible en Internet : www.todoandroid.es/ionic-3-desarrollar-apps-moviles

²⁹www.codigofacilito.com. Progressive web apps. Codigofacilito [en línea]. 22 de agosto de 2018 [revisado el 22 de agosto de 2018]. Disponible en Internet: <https://codigofacilito.com/articulos/progressive-apps>

Herramientas: es un framework que no depende de ningún entorno de desarrollo integrado, puede usarse el que el desarrollador desea, pero el sitio oficial de Ionic recomienda estos, con los plugins respectivos:

- Visual Studio Code
- Atom
- WebStorm
- ALM
- Angular IDE

2.4.7.2 React Native

React Native es un framework que permite la construcción de aplicaciones en plataformas nativas usando JavaScript, este ofrece un traductor de código para las plataformas de IOS y Android.

La idea detrás de React Native se refleja en su slogan "Aprende una vez, escribe en todas partes", permitiendo a los desarrolladores usar la misma estrategia para el desarrollo de cualquier plataforma.

React Native trabaja bajo los principios de React, pero no usa un DOM virtual como lo hace React, en lugar de eso maneja vistas nativas de cada dispositivo, y ejecuta un proceso en segundo plano que interpreta código JavaScript directamente en el dispositivo final y se comunica a partir de ahí con la plataforma nativa, por lo que no depende en absoluto de HTML5, sino de JavaScript y de los SDK nativos.

2.4.7.2.1 Historia

En 2012, Mark Zuckerberg creador de Facebook comentó: "El mayor error que cometimos como empresa fue apostar demasiado por HTML5 en lugar de por nativo". A partir de ese punto el equipo de Facebook empezó el trabajo para ofrecer una mejor experiencia móvil

“Dentro de Facebook, Jordan Walke encontró una manera de generar elementos de IU de IOS a partir de un hilo de JavaScript de fondo. Decidieron organizar

un hackathon interno para perfeccionar este prototipo a fin de poder crear aplicaciones nativas con esta tecnología. Después de algunos meses de desarrollo, Facebook lanzó la primera versión para React.js Conf 2015.”³⁰

2.4.7.2.2 Características

- **Actualización constante de la plataforma:** React Native es relativamente joven y por lo general hacen una actualización cada mes, adicionando nuevos componentes nativos, lo que indica que su comunidad es muy activa.
- **Componentes reutilizables:** React Native usa componentes nativos reutilizables que se compilan directamente sobre los nativos. Esta filosofía de desarrollo en componentes permite crear aplicaciones de una forma más ágil y de reutilización de código.
- **Comunidad amplia en crecimiento:** desarrollado por Facebook y su Core se basa en React, React Native cuenta con una amplia comunidad que aporta constantemente al sostenimiento de su código.
- **Live reload:** función que permite ver de inmediato los cambios que se hagan dentro del código de la aplicación.
- **Fuerte rendimiento para móviles:** las aplicaciones nativas suelen hacer uso intensivo de la CPU del dispositivo, pero React Native hace uso de la GPU (unidad de procesamiento de gráficos).
- **Rendimiento Nativo:** aunque no se puede comparar el rendimiento de una aplicación desarrollada en nativo a una desarrollada con React Native, la diferencia que puede percibir el usuario es casi imperceptible.
- **Interfaz de usuario simplificada:** React Native cuenta con una sólida base en la creación de interfaces de usuario, la cual permite tener claridad siempre en las rutas que puede tomar el usuario, lo que hace más sencilla la corrección de errores.

³⁰[www.wikipedia.org. React Native. Wikipedia \[en línea\]. 01 de agosto de 2018 \[revisado el 23 de agosto de 2018\]. Disponible en Internet:](https://en.wikipedia.org/wiki/React_(JavaScript_library)#React_Native)

[https://en.wikipedia.org/wiki/React_\(JavaScript_library\)#React_Native](https://en.wikipedia.org/wiki/React_(JavaScript_library)#React_Native)

2.4.7.3 Framework 7

“Es un framework de HTML móvil de fuente abierta y gratuita para desarrollar aplicaciones móviles híbridas o aplicaciones web con apariencia nativa de IOS y Android. También es una herramienta de creación de prototipos indispensable para mostrar prototipos de aplicaciones en funcionamiento lo antes posible en caso de que lo necesites.”³¹

El enfoque de framework 7 es la facilidad de desarrollo móvil mediante el uso de las tecnologías más usadas en la web, por lo que la creación de aplicaciones es como la creación de sitios web, usando en su gran mayoría HTML.

“Framework7 no es compatible con todas las plataformas, se enfoca solo en el diseño de IOS y Google Material para brindar la mejor experiencia y simplicidad.”³²

Tiene su propio marco de manipulación que lo llama DOM (DOM7) por lo que no es necesario trabajar con otros framework o librerías, pero al igual se pueden implementar, un ejemplo de esto es que ya contiene cargados gran cantidad de componentes de VueJS.

2.4.7.3.1 Historia

Framework7 se introdujo en el año 2014. La última versión 1.4.2 se lanzó en febrero de 2016 con licencia de MIT.

2.4.7.3.2 Características

Componentes de interfaz: Framework 7 contiene gran cantidad de elementos de interfaz de usuario listos para usar, por lo que se excluye el uso de JavaScript en absoluto.

Fácil de personalizar: los estilos están divididos en pequeños archivos que contiene la misma aplicación en su directorio de archivos.

³¹ www.framework7.io. Documentación oficial. Framework7 [en línea]. 23 de agosto de 2018 [revisado el 23 de agosto de 2018]. Disponible en Internet : <https://framework7.io/>

³² Ibid.

Biblioteca agnóstica: no utiliza ni depende de bibliotecas de terceros, lo que implica que sea más liviano y flexible.

Performant: hardware acelerado que permite mejores animaciones y transiciones CSS además de contar con un sistema de carga retrasada de imágenes para que estas no carguen sino se está sobre ellas.

Marco de código abierto: la última versión 1.4.2 fue lanzada en febrero de 2016 bajo la licencia del MIT. Esta es una Licencia de software libre permisiva lo que significa que impone muy pocas limitaciones en la reutilización y por tanto posee una excelente Compatibilidad de licencia.

Desplazamiento nativo: el desplazamiento por la pantalla es completamente nativo lo que hace que se haga más rápido y efectivo el mismo, además de poder hacer todo tipo de animaciones, muy a la par de cualquier lenguaje nativo.

2.4.7.4 PhoneGap

PhoneGap es un framework multiplataforma para el desarrollo de aplicaciones híbridas móviles. Permite la reutilización de habilidades de desarrollo web para la creación de aplicaciones híbridas con JavaScript, HTML y CSS y sus framework afines.

“El código PhoneGap fue aportado a la Apache Software Foundation (ASF) bajo el nombre de Apache Cordova y se graduó al estado de proyecto de nivel superior en octubre de 2012”³³

PhoneGap utiliza CSS3 y HTML5 para el renderizado y JavaScript para la lógica, además de poder usar las funciones nativas del dispositivo mediante una función una función de interfaz externa.

³³[www.PhoneGap.com](https://PhoneGap.com/about/). Acerca de, documentación oficial. Adobe PhoneGap [en línea]. 23 de agosto de 2018 [revisado el 23 de agosto de 2018]. Disponible en Internet: <https://PhoneGap.com/about/>

2.4.7.4.1 Historia

Nitobi fue el creador original y es uno de los principales contribuidores al framework, empresa encargada de desarrollar aplicaciones móviles, que, para economizar en costos y tiempos, desarrolla el framework con el objetivo de hacerla compatible con todos los dispositivos móviles y sistemas operativos.

En 2011 Adobe compra la empresa, con el requisito de que el proyecto fuera enviado a la fundación Apache para que siga siendo de código abierto, pasando a ser Adobe PhoneGap, parte de la suite de Adobe Creative Cloud siendo comercializado así por Adobe, pero el proyecto PhoneGap pasa a formar parte de la fundación Apache con el nombre de Apache Cordova.

2.4.7.4.2 Características

API: cuenta con un conjunto de funciones, comando, protocolos que permiten controlar recursos de los dispositivos móviles, como la cámara, el acelerómetro, GPS, notificaciones, almacenamiento, Sistemas de ficheros etc. Además, el motor sobre el cual se basa PhoneGap, Apache Cordova, sirve como intermediario para el acceso a recursos nativos de otros framework, como lo es el caso de Ionic.

PhoneGap Build: permite construir rápidamente apps móviles y compilarlas con facilidad sin necesidad de SDK o hardware específico. Es un servicio en la nube que brinda adobe, el cual es un compilador al cual se puede acceder mediante una cuenta adobe, y permite adjuntar un archivo a través de una cuenta en GitHub o un archivo .Zip desde un equipo. Este servicio no está disponible para Apache Cordova.

Licencia: cuenta con una licencia Apache 2.0, “la Licencia Apache permite al usuario del software la libertad de usarlo para cualquier propósito, distribuirlo, modificarlo, y distribuir versiones modificadas de ese software.”³⁴

³⁴Wiki.fdi.ucm.es. licencia Apache. Wiki de la asignatura de ética legislación y profesión (ELP) [en línea]. 03 de Julio de 2017 [revisado el 23 de agosto de 2018]. Disponible en Internet: http://wikis.fdi.ucm.es/ELP/Licencia_Apache

Rápido: “Si lo que buscas es hacer una aplicación sencilla y lo más rápidamente posible para ser el primero en hacerla, PhoneGap puede ser tu herramienta. En poco tiempo, si no necesitas muchos recursos, PhoneGap te permite desarrollar rápidamente con el plus que te da el hecho de que automáticamente el desarrollo es multiplataforma. Llegar a más dispositivos, más rápido.”³⁵

Integración: fácil integración con distintas librerías de desarrollo como JQuery Mobile, Dojo y Sencha Touch.

Herramientas: es un framework que no depende de ningún entorno de desarrollo integrado, puede usarse el que el desarrollador desea, de preferencia se debe escoger un editor al cual la comunidad haya hecho aportes y permitan la instalación de plugins, snippets, etc.

2.4.7.5 NativeScript

NativeScript es un framework que permite la construcción de aplicaciones móviles multiplataforma mediante el uso de XML, CSS y JavaScript, usando un motor de renderizado de plataforma proporcionando una experiencia nativa.

Se puede integrar con framework como Angular o VueJS para trabajar con una arquitectura más organizada, en este no todas las características CSS están disponibles, es decir cuenta con un subconjunto de CSS, y usa XML para describir la interfaz de usuario de la aplicación.

Con este se pueden construir casi cualquier tipo de aplicaciones los únicos factores limitantes son el desempeño y la disponibilidad de plugins.

2.4.7.5.1 Historia

Fue lanzado por primera vez al público en marzo de 2015, gana popularidad rápidamente en la plataforma GitHub, y en su Twitter oficial alcanzo alrededor de

³⁵www.Ecured.com. PhoneGap. Ecured [en línea]. 23 de septiembre de 2018 [revisado el 23 de agosto de 2018]. Disponible en Internet: <https://www.ecured.cu/PhoneGap>.

1500 seguidores después del lanzamiento de su versión 1.0.0, dos meses después de haber sido anunciado.

Desarrollado e ideado por progress, el cual hace grandes aportes su código fuente, contando ya con alrededor de 600 complementos disponibles, que vienen de la comunidad o son directamente aportados por progress.

2.4.7.5.2 Características

Rendimiento nativo: permite desarrollar una interfaz, similar a las de las plataformas nativas, “NativeScript utiliza las estructuras de datos XML que representan la abstracción multiplataforma para activar el código específico de la plataforma que interactúa directamente con los elementos nativos del sistema operativo objetivo.”³⁶

Licencia: NativeScript es de código abierto, pudiendo usar su código de la manera en que se desee ya que cuenta con la licencia de software Apache 2.0. “la Licencia Apache permite al usuario del software la libertad de usarlo para cualquier propósito, distribuirlo, modificarlo, y distribuir versiones modificadas de ese software.”³⁷

Soporte: la empresa desarrolladora de NativeScript ofrece soporte empresarial que se adapta a las necesidades de los usuarios, además de soporte profesional de ingenieros de NativeScript, garantizando asesoría constante en todos los procesos de desarrollo.

Fábrica de soporte 0 días: NativeScript puede usar las últimas API nativas y las últimas interfaces de usuario inmediatamente una vez Google, Facebook, Microsoft etc. Actualicen sus plataformas móviles. Siempre tiene un 100% de acceso a cada API y componente de la plataforma nativa desde el momento en que la función esté disponible en el sistema operativo móvil.

³⁶www.wikipedia.org. Nativescript. Wikipedia [en línea]. 19 de marzo de 2018 [revisado el 24 de agosto de 2018]. Disponible en Internet: <https://es.wikipedia.org/wiki/NativeScript>.

³⁷Wiki.fdi.ucm.es. OP. CIT.

No HTML: NativeScript no se incorpora en el navegador interno de los dispositivos móviles por lo que el modelo de objetos del documento (DOM) no existe, sin el DOM HTML es removido, por lo que de alguna manera se tiene mejor rendimiento.

Fácil aprendizaje: es mucho más fácil encontrar un desarrollador web que uno móvil por lo que esto ye es una ventaja en el caso de los framework para la creación de aplicaciones híbridas. Utilizando las habilidades en la web que se tienen de JavaScript y CSS.

Integración: “En NativeScript se puede seleccionar si se desea desarrollar con JavaScript, con TypeScript o con Angular 2, haciendo al framework más formidable debido a las características adicionales de Angular.”³⁸

Herramientas: es un framework que no depende de ningún entorno de desarrollo integrado, puede usarse el que el desarrollador desea, pero ellos recomiendan usar Visual Studio Code o WebStorm cada uno con sus respectivos plugin.

³⁸www.rigelnetworks.com. Going Native from Hybrid, with NativeScript. Rigel Networks le tus grow together. [en línea]. 28 de abril de 2017 [revisado el 24 de agosto de 2018]. Disponible en Internet: <https://www.rigelnetworks.com/6621-2/>

3 COMPARATIVO Y SELECCIÓN DEL FRAMEWORK

3.1 INTRODUCCIÓN

En el siguiente capítulo se hizo una evaluación comparativa de los framework de desarrollo móvil híbrido estudiados en el capítulo anterior, en la sección, revisión de framework híbridos.

Se definió una escala de medición que tendrá como fin calificar cada framework sobre criterios de evaluación. Los criterios usados para el estudio comparativo fueron seleccionados a partir de mesas de trabajo con un equipo en el CIADTI y recopilación de información de estudios similares a este.

Para finalizar se hizo un cuadro de resultados que tendrá la información de la comparación de los framework en función de los criterios, para así poder escoger con que tecnología se dará comienzo al desarrollo del prototipo.

3.2 ESCALA DE MEDICIÓN

Definir la escala de medición por lo general es trabajo del ejecutor del proyecto, no es necesario usar un modelo estandarizado para definir una escalada de medición, esta se define de acuerdo a las necesidades.

En este proyecto se usarán dos métricas una cualitativa para representar el nivel de aprobación del criterio y una escala normalizada con valores de 1 a 5 siendo 1 el más bajo y 5 el valor máximo.

Tabla 2 - Métricas de evaluación

Normalizada	Cualitativa
1	No cumple
2	Cumple parcialmente
3	Cumple con lo necesario
4	Cumple
5	Cumple a cabalidad

3.3 INSTRUMENTO DE EVALUACIÓN

El instrumento de evaluación que se uso fue una matriz de pesos, la cual está conformada por los framework, los criterios de evaluación y la métrica normalizada, esto con el objetivo de poder cuantificar los resultados del análisis hecho sobre los framework de desarrollo móvil híbrido.

Esta es una manera sencilla de observar los criterios en los cuales un framework tiene mayor desempeño, motivo por el cual se escogió el mismo. La matriz de pesos cuenta con la estructura mostrada en la siguiente tabla:

Tabla 3 - Matriz de pesos

		Framework					Subtotal
Criterios		#	#	#	#	#	Σ
		#					
		#					
		#					
		#					
	Subtotal	Σ					

3.4 CRITERIOS DE EVALUACIÓN

Los criterios expuestos a continuación fueron los que se consideraron necesarios en el proceso de selección del framework, que será usado en el desarrollo del prototipo de la aplicación móvil.

3.4.1 Funcionalidad

La funcionalidad es la capacidad que tiene un software de suplir las necesidades para las cuales fue desarrollado, es decir si brinda las herramientas necesarias para satisfacer las necesidades de quien lo usa. En términos de desarrollo de software, “se relaciona directamente con aquello que el software hace para

satisfacer necesidades, mientras que otras características se refieren al cómo y al cuándo.”³⁹

3.4.2 Mantenibilidad

Es la capacidad que tiene un framework para ser modificado, es decir, la respuesta con la que cuenta en caso que un cambio ocurra o que una nueva funcionalidad surja. También puede incluir comportamiento en cuanto a cambios del framework, actualizaciones, versiones, bibliotecas Apis.

3.4.3 Usabilidad

“La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva”⁴⁰, se tiene en cuenta la comunidad la cual respalda al framework y la documentación con la cual cuenta.

3.4.4 Portabilidad

Es la capacidad que tiene un software para ejecutarse en diferentes plataformas, en el caso de los framework de desarrollo móvil híbrido, se tomara a la portabilidad como los entornos o sistemas operativos sobre los cuales el framework tiene alcance.

3.4.5 Índices de interés

“Este criterio pretende mostrar qué tanto interés han tenido las personas que utilizan el buscador de Google, por medio de su herramienta Google Trends que permite a través de gráficas observar el comportamiento de la búsqueda especificada en el tiempo”.⁴¹

³⁹ www.diplomadogestioncalidadsoftware2015.com. ISO/IEC 9126. ISO/IEC 9126 [en línea]. [revisado el 2 octubre de 2018]. Disponible en Internet: <https://diplomadogestioncalidadsoftware2015.wordpress.com/norma-iso-9126/calidad-interna-y-externa/>

⁴⁰ Ibid.

⁴¹ NARVAEZ, Bolaño. Op. cit., p.66.

3.5 COMPARATIVO

3.5.1 Funcionalidad

Un aspecto importante en la funcionalidad y en el desarrollo de un sistema es la seguridad, para el presente trabajo los framework que se comparan trabajan en la parte del cliente en donde proporcionar seguridad no es algo de gran prioridad, esto se suele delegar al lado del servidor, un ejemplo de esto es que los nuevos navegadores permiten la edición del código del lado del cliente (front-end) de manera temporal, pero es posible, por lo que la seguridad se delega al lenguaje del servidor, al servidor en sí y al API Rest.

Si se desea proporcionar seguridad en el cliente se puede hacer uso de métodos de encriptación o mecanismos de firmado de llave pública y llave privada, pero para el caso no es necesario, ya que, la aplicación prototipo a desarrollar es de índole informativo y no hará ningún tipo de cambios en la base de datos o el servidor que puedan afectar la integridad del sistema de información de la Universidad de Pamplona, claro que esto puede cambiar con el tiempo.

3.5.1.1 Ionic

Las aplicaciones de Ionic se ejecutan sobre un componente interno de los dispositivos móviles llamado WebView que no es más que el navegador integrado que tienen los dispositivos móviles, el cual soporta las tecnologías sobre las cuales se basa el desarrollo de sitios web.

Ionic basa su núcleo en el framework Angular, por lo que cuando se crea un proyecto tiene una estructura similar por no decir que igual, su estructura básica de directorios y archivos está compuesto por:

src: es la carpeta donde se realiza la mayor parte del trabajo en Ionic, son los archivos fuente donde se escribirá el código para desarrollar la aplicación, el código será escrito en TypeScript, dentro de esta carpeta se encuentran otros directorios y archivos distribuidos de acuerdo a la función que prestan en la aplicación, entre ellos están:

pages: es la carpeta que contiene las páginas o pantallas de la aplicación.

asset: contiene los archivos multimedia como imágenes iconos u otros archivos externos que se usaran en las páginas de la aplicación.

app: carpeta raíz que tiene los archivos de arranque de la aplicación.

providers: contiene archivos TypeScript que proporcionan la información, los datos necesarios a nuestra información para ser representados en las pantallas de la aplicación.

pipes: contiene archivos de TypeScript que permiten hacer transformaciones a la información que será representada en la vista

index.html: archivo que es el punto de arranque de la aplicación web.

www: se alojan archivos que se producen de manera automática cuando se transpila el código TypeScript de la carpeta src, este código es el que se compila cuando una aplicación de Ionic se ejecuta en un navegador de una computadora.

node_modules: contiene todas las dependencias básicas necesarias para el funcionamiento de la aplicación, también se guardan las que se instalen después que se empiece a trabajar en la aplicación.

platforms: contiene los proyectos con el lenguaje nativo transpilado mediante el uso de Apache Cordova, para cada uno de los sistemas operativos sobre los cuales se trabaje.

resources: contiene el icono y la pantalla de presentación de la aplicación para cada una de las plataformas a las cuales va destinada la aplicación.

plugins: contiene los plugins nativos los cuales se instalan en la aplicación, pueden ser de terceros o del propio Ionic.

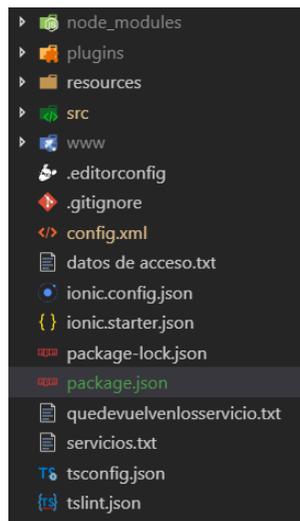
package.json: archivo que contiene la declaración de dependencias, script y paquetes que se integran en la carpeta node_modules.

config.xml: contiene los parámetros que se utilizan cuando se construye el proyecto nativo a partir de un proyecto de Ionic.

Tsconfig.json y tslint.json: son archivos que contienen información necesaria a la hora de compilar TypeScript.

Un proyecto con Ionic contiene más archivos y carpetas, pero cuando al momento de desarrollar una aplicación no son manejados de ninguna forma.

Ilustración 5 - Estructura de ficheros Ionic



En Ionic el bloque de construcción básico se les conoce como modulo, el cual encapsula templates construidos con HTML y etiquetas propias de Ionic, estos son controlados por lógica creada por el desarrollador la cual se escribe en una clase de TypeScript conocida como componente, y se le dan estilos al template con archivos SCSS. A demás la data o información proveniente de fuentes externas ajenas al componente, como peticiones HTTP de un servidor son manejadas por clases de TypeScript conocidas como providers.

Tanto los componentes como los providers son clases de TypeScript similares en estructura pero que se diferencian entre si gracias a los decoradores, que marcan

el tipo de clases que representan, y le proporcionan los datos necesarios para su funcionamiento.

Módulos

“Un módulo declara un contexto de compilación para un conjunto de componentes. Un módulo puede asociar sus componentes con código relacionado, como servicios, para formar unidades funcionales”⁴².

Cada aplicación generada cuenta con un módulo de raíz el cual implementa el mecanismo de arranque que inicia una aplicación. Pero se pueden modularizar las aplicaciones creándolos de acuerdo a las necesidades del desarrollador. Esto mejora el rendimiento de las aplicaciones debido a que solo se inician los módulos que se necesiten en su momento.

`@NgModule` () es el decorador que identifica una clase de TypeScript como un Módulo de la aplicación.

⁴² <https://angular.io>, descripción de la arquitectura, Angular Documentation [En línea] , [revisado el 1 de Noviembre de 2018]. Disponible en internet : <https://angular.io/guide/architecture>

Ilustración 6 - Estructura de un módulo en Ionic

```
1. /* JavaScript imports */
2. import { BrowserModule } from '@angular/platform-browser';
3. import { NgModule } from '@angular/core';
4. import { FormsModule } from '@angular/forms';
5. import { HttpClientModule } from '@angular/common/http';
6.
7. import { AppComponent } from './app.component';
8.
9. /* the AppModule class with the @NgModule decorator */
10. @NgModule({
11.   declarations: [
12.     AppComponent
13.   ],
14.   imports: [
15.     BrowserModule,
16.     FormsModule,
17.     HttpClientModule
18.   ],
19.   providers: [],
20.   bootstrap: [AppComponent]
21. })
22. export class AppModule { }
```

Fuente: <https://angular.io/guide/bootstrapping>

Componentes

Es la unidad que permite la construcción de las pantallas de una aplicación móvil en Ionic junto con los templates. Son clases de TypeScript que al igual que los módulos cuentan con un componente raíz que es el que da inicio a una jerarquía de pantallas en las aplicaciones móviles.

La comunicación de metadatos entre los templates y los componentes es posible mediante las directivas las cuales proveen una lógica de programación llamada data-binding que establece una comunicación entre los componentes y los templates, esta comunicación puede ser de eventos y propiedades.

`@Component` () es el decorador que identifica una clase de TypeScript como un componente.

Ilustración 7 - Estructura de un template en Ionic

```
<ion-header no-border>
  <ion-navbar>
  </ion-navbar>
</ion-header>
<ion-content>
</ion-content>
```

Ilustración 8 - Estructura de un componente en Ionic

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

/**
 * Generated class for the FinanciacionExternaPage page.
 *
 * See https://ionicframework.com/docs/components/#navigation for more info on
 * Ionic pages and navigation.
 */

Unsaved changes (cannot determine recent change or authors)
@IonicPage()
@Component({
  selector: 'page-financiacion-externa',
  templateUrl: 'financiacion-externa.html',
})
export class FinanciacionExternaPage {

  constructor(public navCtrl: NavController, public navParams: NavParams) {}
}
```

Providers

Los providers son clases las cuales contienen toda la lógica que no está asociada directamente con un template y que se quiere usar en una aplicación, por lo general contiene la información proveniente a peticiones HTTP. Al ser una clase de TypeScript los provider necesitan de decoradores para identificar que van a tener el comportamiento de uno, por lo que se usa el decorador `@Injectable ()`.

Ilustración 9 - Estructura de un provider en Ionic

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

/*
 * Generated class for the MapaProvider provider.
 *
 * See https://angular.io/guide/dependency-injection for more info on providers
 * and Angular DI.
 */
SandroCardenas, 2 months ago | 1 author (SandroCardenas)
@Injectable()
export class MapaProvider {

  constructor(public http: HttpClient) {
    console.log('Hello MapaProvider Provider');
  }

}
```

En la anterior ilustración se observa que Ionic hace un llamado a la clase HttpClient en el constructor de esta, esta clase es la que permite hacer el llamado de peticiones GET, POST, PUT o DELETE como se explicó en la sección que habla de servicios REST.

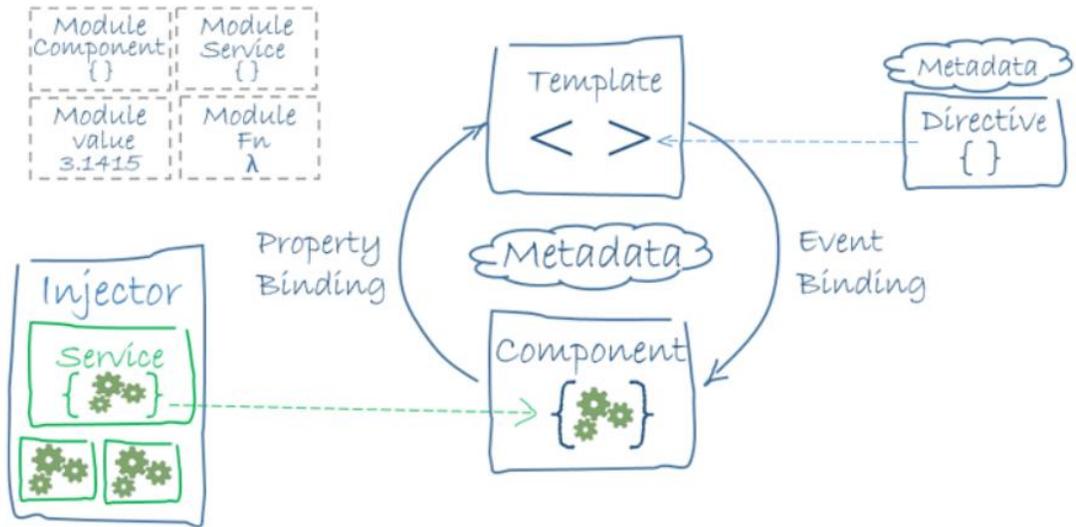
Enrutamiento

En una aplicación móvil hecha con Ionic el paso de una pantalla a otra se le conoce como enrutamiento, este sistema de enrutamiento está basado en la navegación nativa de los dispositivos de IOS la cual se basa en una pila de pantallas sobre las cuales se puede avanzar hacia adelante y atrás hasta llegar a una pantalla base que para el caso sería el template del componente raíz.

Su funcionamiento es sencillo solo es empujar y traer, pero hay un componente en Ionic que funciona de manera un poco distinta, y son los tabs, en donde cada uno de los tabs posee una pila de navegación independiente.

El componente de navegación básico de Ionic es la etiqueta <ion-nav>, la directiva [navPush] y el controlador NavController también permiten la navegación entre pantallas.

Ilustración 10 - Arquitectura de datos en Ionic



Fuente: <https://angular.io/guide/architecture>

Los anteriores elementos permiten la construcción de una aplicación en Ionic, pero sin ningún tipo de acceso al hardware del dispositivo móvil, para este fin Ionic usa un intermediario el cual es Apache Cordova, que contiene gran cantidad de plugins que permiten el acceso a componentes nativos del dispositivo.

Como se crea un proyecto en Ionic

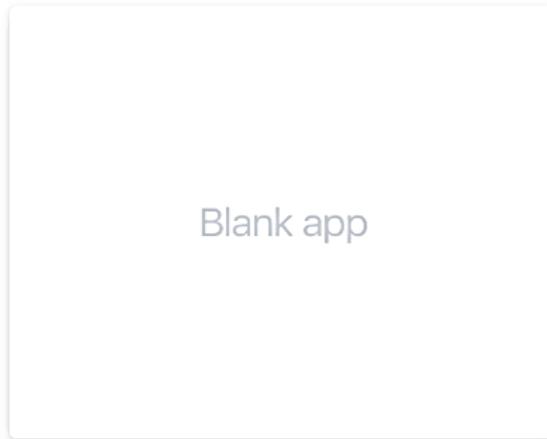
Para la creación de una aplicación con Ionic primero se debe instalar el gestor de paquetes npm el cual viene integrado con el entorno de ejecución de JavaScript Node.JS.

Mediante la consola independientemente del sistema operativo, se ejecuta el comando `npm install -g Ionic cordova`, este instalara Ionic en su última versión en el sistema, junto con él generador de código de Ionic y lo deja preparado para la creación de proyectos mediante el comando `Ionic start myApp blank`.

Esto genera la estructura de archivos de la que anteriormente se habló, dando paso al proceso de creación de aplicaciones móviles y automáticamente Ionic genera un módulo base y un componente base

Para probar las aplicaciones se puede hacer mediante el uso del navegador de la computadora mediante el comando `Ionic serve`, Ionic pondrá en marcha la aplicación sobre el servidor proporcionado por Node.JS.

Ilustración 11 - Pantalla inicial en un nuevo proyecto de Ionic



Fuente: <https://ionicframework.com/getting-started#cli>

Acceso al hardware del dispositivo

El acceso al hardware del dispositivo se hace mediante la intervención de los plugins de Apache Cordova. Ionic cuenta con toda una sección en su documentación oficial acerca del uso de estos plugins los cuales deben ser agregados al proyecto.

Por lo general este proceso se hace mediante los siguientes comandos: `Ionic cordova plugin add "nombre del plugin"` y `npm install --save @Ionic-native/" nombre del plugin"`

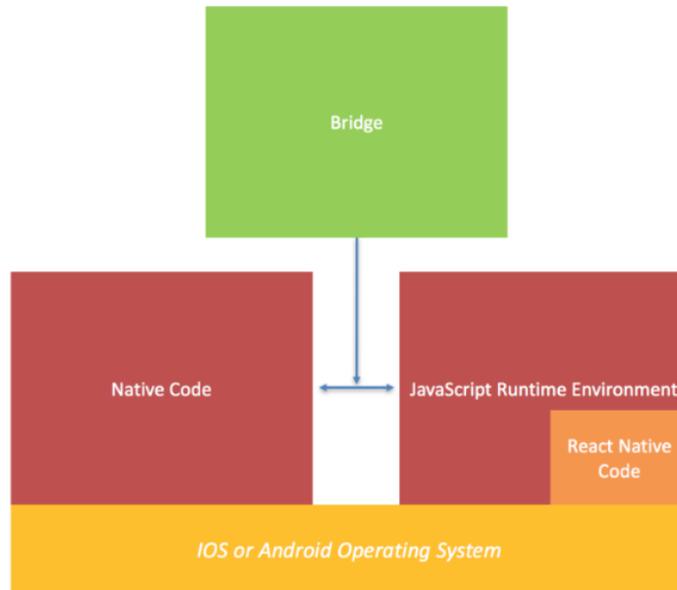
Para mayor seguridad el sitio proporciona un procedimiento paso a paso <https://ionicframework.com/docs/native/>.

3.5.1.2 React Native

React Native es un framework de JavaScript para el desarrollo de aplicaciones móviles nativas mediante el uso de JavaScript, el cual se basa su núcleo de desarrollo en el framework React.JS.

React Native no usa un web View para ejecutar las aplicaciones, en vez de eso usa una técnica conocida como “puenteo”, lo cual permite que JavaScript se comunique con el sistema operativo del dispositivo mediante un conjunto de funciones y utilidades que se ejecuten mientras el sistema operativo se está ejecutando, a esto se le conoce como el runtime de JavaScript.

Ilustración 12 - Punteo en React Native



Fuente: <https://www.bit.es/knowledge-center/react-native-una-nueva-arquitectura-para-la-generacion-de-apps-moviles/>

React Native ejecuta el código en dos entornos de máquina virtual dependiendo del sistema operativo:

- **JavaScriptCore:** motor de JavaScript que potencia el navegador safari en IOS.
- **V8:** motor de JavaScript que potencia al navegador Google Chrome en Android.

Esta ejecución independiente de acuerdo al sistema operativo hace que algunas partes del código en JavaScript no sea el mismo para las dos plataformas, en vez de esto se desarrollan componentes independientes dependiendo del sistema objetivo.

Cuando se crea un proyecto en React Native se crea la siguiente estructura de directorios y archivos:

IOS: contiene los archivos necesarios para la compilación de un proyecto en el sistema operativo IOS, en esta carpeta se crea un proyecto de Xcode el sdk de IOS para el desarrollo de aplicaciones móviles.

Android: contiene los archivos necesarios para la compilación de un proyecto en el sistema operativo Android, en esta carpeta se crea un proyecto de Java el lenguaje nativo de Android para desarrollo de aplicaciones móviles.

Node_modules: contiene todas las dependencias básicas necesarias para el funcionamiento de la aplicación, también se guardan las que se instalen después que se empiece a trabajar en la aplicación.

Package.json: archivo que contiene la declaración de dependencias, script y paquetes que se integran en la carpeta node_modules.

Index.ios.js: es el componente inicial de una aplicación desarrollada en IOS con React Native, conocido también como el componente raíz.

Index.Android.js: es el componente inicial de una aplicación desarrollada en Android con React Native, conocido también como el componente raíz.

.Flowconfig: archivo con la configuración de un analizador estático de lenguaje de JavaScript diseñado para detectar posibles errores causado por lo tipos de variables.

.Watchmanconfig: archivo de configuración de la librería watchman, la cual detecta si archivos cambian en el proyecto y desencadena las acciones para las cuales fu diseñado.

React Native no tiene una estructura de carpetas definida donde organizar sus componentes, la organización es dada por el desarrollador, pero los desarrolladores

del framework recomiendan el uso del patrón de diseño flux o el patrón de diseño redux.

En React Native el bloque de construcción básico se conoce como componente, el cual encapsula todo lo necesario para la creación de pantallas en la aplicación, como las vistas, los estilos, las propiedades, los estados y demás componentes que conforman una aplicación de React Native.

Si no se usa un patrón de diseño como flux se puede dar organización deseada por lo general se crea una carpeta en donde se guardarán todos los componentes separándolos de acuerdo a sistema operativo para el cual estén destinados y nombrándolos de acuerdo a esto con `ios.js` para un componente construido para el sistema operativo IOS y `.android.js` para un componente construido para el sistema operativo android; esta carpeta se suele nombrar como `src`.

Si los componentes no tienen ninguna diferencia en cuanto a su construcción no es necesario nombrarlos con `ios.js` o `.android.js`, solamente como el nombre del componente.`.js`.

La razón por la cual los componentes se nombran de esta forma es que, cuando se compila el proyecto se hace un puenteo y se crea código nativo a partir de, código JavaScript, y esta transformación es específica para cada plataforma.

Componentes

Un componente en React Native es una clase de JavaScript con unas variaciones, contiene las importaciones que vienen directo del framework React, luego el constructor, después el método `render ()` que es el encargado de renderizar el código usado por React Native, JSX y por último los métodos usados por `render ()`.

Es un elemento visual que tiene un estado, contiene su propia lógica, que recibe propiedades. Un componente es una sección de la aplicación de Native Script y es una parte lógica de la aplicación, el uso de componentes secciona el código en partes para poder reutilizarlo donde sea necesario.

En otras palabras, un componente es cualquier trozo o parte de una aplicación móvil que cumple con un objetivo en específico, como un menú de navegación, una barra de búsqueda, un slide de imágenes, etc.

Ilustración 13- Estructura de un componente de React Native

```
'use strict'

import React, {
  Component,
  View,
  Text
} from 'react-native'

class dashboardView extends Component{
  render(){
    return(

  )
}
}
```

Render ()

Toda componente de React Native tiene un método llamado Render el cual se encarga de renderizar el código JSX en código entendible para los dispositivos móviles como Java u Objective-c.

Propiedades

Las propiedades o props en React Native es la forma en la cual los componentes se pueden comunicar, es la manera en que un componente superior es decir un componente que contiene a otros, enviar información a un componente inferior es decir un componente contenido.

La gran mayoría de componentes predefinidos en React Native reciben props para ser construidos, un ejemplo es el siguiente componente mensaje el cual recibe tres props “size”, “color” y “contenido”, los cuales son necesarios para su construcción.

Ilustración 14 - Estructura de un componente mensaje en React Native

```
import React, {Component} from 'react'

class Mensaje extends Component {
  render () {
    const styles = {
      color: this.props.color,
      fontSize: this.props.size
    }
    return (
      <h3 style={styles}>{this.props.contenido}</h3>
    )
  }
}

export default Mensaje
```

Fuente: <https://medium.com/@taylorwebk/react-al-descubierto-ii-components-props-y-state-react-t%C3%A1stico-9888a5d9ce43>

Ilustración 15 – Llamado del componente mensaje en otro componente Root

```
import React, { Component } from 'react'
import {Container, Button} from 'semantic-ui-react'
import Mensaje from './Mensaje'
class Root extends Component {
  render () {
    return (
      <Container textAlign='center'>
        <Mensaje size='30px' color='#ff0000' contenido='AHORA' />
        <Mensaje size='40px' color='#00ff00' contenido='ENTIENDO' />
        <Mensaje size='60px' color='#0000ff' contenido='LOS PROPS
:D' />
        <Button content='Boton Cool' color='blue' />
      </Container>
    )
  }
}

export default Root
```

Fuente: <https://medium.com/@taylorwebk/react-al-descubierto-ii-components-props-y-state-react-t%C3%A1stico-9888a5d9ce43>

Estados

States como se conoce en React Native indican el estado de un componente en determinado momento, un ejemplo sería un componente que haga peticiones a un servidor, este podría tener dos posibles estados, cargando o finalizada la carga.

Los estados son determinados por el desarrollador y se pueden definir cuantos se crean necesarios para que un componente funcione adecuadamente.

La definición de los estados de un componente se hace mediante el constructor de la clase de JavaScript, para cambiar los estados se hace mediante métodos predefinidos en estos, en la siguiente imagen se puede ver un ejemplo de un componente cargando el cual cambia de estado cuando se presiona un botón.

Ilustración 16 - Estructura de un componente con cambio de state en React Native

```
import React, {Component} from 'react'
import {Button, Loader} from 'semantic-ui-react'
class Cargando extends Component {
  constructor (args) {
    super(args)
    this.state = {
      loading: false
    }
    this.changeLoading = this.changeLoading.bind(this)
  }
  changeLoading () {
    this.setState((prevState, props) {
      return {loading: !prevState.loading}
    })
  }
  render () {
    let contenido
    if (this.state.loading) {
      contenido = <Loader inline active />
    } else {
      contenido = <span>No esta cargando</span>
    }
    return (
      <div>
        {contenido}
        <Button content='CambiarEstado' onClick={this.changeLoading} />
      </div>
    )
  }
}
```

Fuente <https://medium.com/@taylorwebk/react-al-descubierto-ii-components-props-y-state-react-t%C3%A1stico-9888a5d9ce43>

Navegación

React Native para la navegación entre las pantallas de una aplicación hace uso de una librería de terceros llamada React – Navigation la cual se instala usando el

gestor de paquetes npm, mediante el comando `npm install --save react-navigation`.

Con React Navigation se establecen rutas de navegación sobre cada uno de los componentes mediante un objeto de JavaScript, al cual se le conoce como enrutador.

Los enrutadores hacen la lógica de navegación mucho más sencilla, gracias a que cada componente cuenta con su propio enrutador el cual analiza si debe empujar o sacar de la pila de navegación las vistas en los componentes.

Con los componentes anteriores ya es posible crear una aplicación básica en React Native. Algo a tener en cuenta es que una aplicación compleja en React Native requiere algunos conocimientos básicos sobre los lenguajes nativos de IOS y Android, lo que puede representar tanto como una ventaja para desarrolladores con experiencia en estos lenguajes, como una desventaja si lo que se busca evitar el uso de los lenguajes nativos.

Como crear un proyecto en React Native

La instalación de React Native es sencilla debido a que se hace mediante al gestor de paquetes npm el cual viene integrado con Node.JS. Una vez Instalado Node.JS independientemente del sistema operativo, se ejecuta en la consola el comando en la consola `npm install -g react-native-cli`, una vez hecho esto se empezará a descargar React Native y todas las dependencias necesarias para su funcionamiento.

Para crear un proyecto de se usa el comando `React-native init "nombre del proyecto"`, este proceso puede demorar un poco pero al finalizar ya se cuenta con un proyecto completo de React Native. Como complemento se recomienda instalar Xcode o Andorid Studio para IOS y Android respectivamente, ya que el código de React Native no se puede Probar directamente sobre el navegador de la computadora.

Una vez instalado Xcode o Android Studio, el proyecto de React Native puede ser compilado mediante el comando `react-native run-ios` o `react-native run-android`, según el sistema operativo sobre el cual se desea compilar la aplicación.

Acceso al hardware del dispositivo

El acceso al hardware del dispositivo también se lleva a cabo por componentes, estos componentes están disponibles en la documentación oficial <https://facebook.github.io/react-native/docs/> y deben ser agregados al proyecto mediante el uso de npm, el código necesario sería: `npm add react-native-“nombre del componente”`.

3.5.1.3 Framework 7

Framework 7 es un marco para el desarrollo de aplicaciones móviles híbridas mediante el uso de JavaScript, en su sitio web oficial cuenta con ayudas para empezar a crear aplicaciones. Las aplicaciones móviles desarrolladas en framework7.

Las aplicaciones desarrolladas con framework7 se ejecutan sobre un componente interno de los dispositivos móviles llamado WebView, que es el navegador nativo integrado en los dispositivos móviles, en si framework7 no cuenta con la capacidad de transpilar un proyecto en IOS o en Android, plataformas a las cuales está destinado, para este fin framework7 se apoya en Apache Cordova el cual le permite no solo transpilar el proyecto de framework7, si no que permite el acceso al hardware de los dispositivos móviles mediante los plugins desarrollados por su comunidad, por lo que framework7 se centra más en el rendimiento de la aplicación y que estas tengan un aspecto nativo de acuerdo al sistema operativo.

Los archivos necesarios del núcleo de framework7 y que permiten comenzar un desarrollo son:

Css: contiene archivos de estilos CSS los va orientados a imitar la apariencia nativa de las interfaces visuales de los dispositivos móviles de acuerdo al sistema operativo.

Js: contiene los archivos JavaScript con los componentes de visualización separados en archivos de acuerdo al sistema operativo al cual van orientados.

Index.html: archivo HTML raíz el cual contiene una vista a manera de ejemplo, con el uso de los principales componentes y estilos que framework7.

Como crear un proyecto en Framework7

Para crear un proyecto, se debe descargar el proyecto framework7 desde la página oficial o desde el repositorio oficial de GitHub. Una vez descargado el proyecto cuenta con una serie de archivos de los cuales el que interesa es la carpeta `package/core` la cual contiene el núcleo de desarrollo de framework7.

Esta carpeta contiene los archivos JavaScript, CSS y HTML necesarios para trabajar con framework7, además el `index.html` del proyecto es un archivo a manera de ejemplo que sirve como guía para empezar el desarrollo. Ahora desde el fichero que se va a trabajar y mediante el uso de la etiqueta de HTML `<link></link>` se hace el llamado de los archivos necesarios para trabajar, que por lo general son:
`framework7.material.min.css,` `framework7.material.colors.min.css,`
`framework7.min.js`

Ilustración 17 - Estructura de un archivo HTML que importa Framework7

```
<html>
<head>
  <!-- Required meta tags-->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui,
viewport-fit=cover">
  <meta name="apple-mobile-web-app-capable" content="yes">
  <!-- Color theme for statusbar -->
  <meta name="theme-color" content="#2196f3">
  <!-- Your app title -->
  <title>My App</title>
  <!-- Path to Framework7 Library CSS -->
  <link rel="stylesheet" href="path/to/framework7.min.css">
  <!-- Path to your custom app styles-->
  <link rel="stylesheet" href="path/to/my-app.css">
</head>
<body>
  <!-- App root element -->
  <div id="app">
    <!-- Statusbar overlay -->
    <div class="statusbar"></div>

    <!-- Your main view, should have "view-main" class -->
    <div class="view view-main">
      <!-- Initial Page, "data-name" contains page name -->
      <div data-name="home" class="page">

        <!-- Top Navbar -->
        <div class="navbar">
          <div class="navbar-inner">
            <div class="title">Awesome App</div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

Fuente: <https://framework7.io/docs/app-layout.html>

Framework7 le ha apostado a la integración con otras librerías como Vue.JS o React.JS ampliando las posibilidades de desarrollo, por lo que ahora en sus paquetes de desarrollo incluye núcleos para el trabajo con estos framework, y dan un mayor soporte y organización a su código por parte de la comunidad.

Una pantalla de una aplicación móvil la representa un archivo HTML el cual debe tener un componente raíz marcado con el atributo de clase app, esta etiqueta debe contener todos los demás elementos y componentes de la aplicación móvil.

Un componente en framework7, por ejemplo, un botón, tiene una estructura común y corriente HTML, solo se deben agregar atributos de clases a las etiquetas HTML.

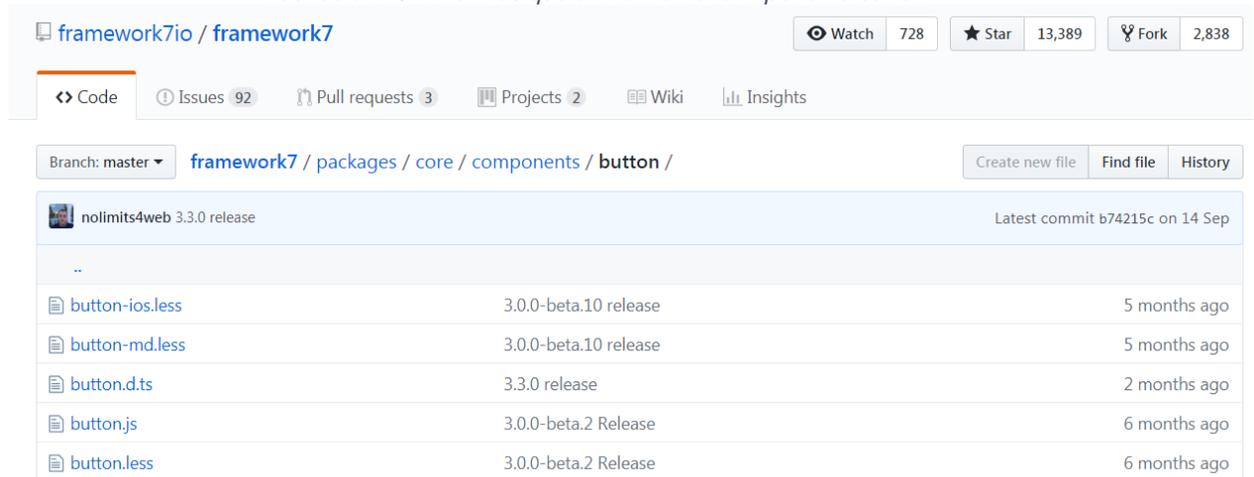
Ilustración 18 - Estructura de un botón HTML en framework7

```
<div class="block">
  <div class="row">
    <button class="button col">Button</button>
    <button class="button col">Button</button>
    <button class="button col button-round">Round</button>
  </div>
</div>
```

Fuente : <https://framework7.io/docs/button.html>

Esta estructura se encuentra definida en los componentes proporcionados por el framework, una revisión a la documentación y al código del núcleo puede dar algunos indicios acerca del uso de cada uno de los componentes:

Ilustración 19 - Archivos que conforman el componente button



The screenshot shows the GitHub interface for the repository 'framework7io / framework7'. The current branch is 'master'. The file path is 'framework7 / packages / core / components / button /'. The repository has 728 watchers, 13,389 stars, and 2,838 forks. The file list includes:

File Name	Version	Last Commit
..		
button-ios.less	3.0.0-beta.10 release	5 months ago
button-md.less	3.0.0-beta.10 release	5 months ago
button.d.ts	3.3.0 release	2 months ago
button.js	3.0.0-beta.2 Release	6 months ago
button.less	3.0.0-beta.2 Release	6 months ago

Fuente: <https://GitHub.com/framework7io/framework7/tree/master/packages/core/components/button>

Navegación entre las pantallas de la aplicación

La navegación entre las pantallas de la aplicación se hace mediante la definición de un objeto de rutas el cual define la URL de navegación que corresponde a cada una de las pantallas de la aplicación. Estas son definidas al inicio de la aplicación en el componente raíz, app, para que puedan ser accedidas en todas las nuevas pantallas que se creen.

Se pueden definir rutas adicionales de la misma manera, cuantas veces se necesite en la aplicación, pero con cuidado de no mezclar las rutas entre los objetos de rutas.

Ilustración 20 - Objeto de Rutas de Framework7

```
var app = new Framework7({
  routes: [
    {
      name: 'about',
      path: '/about/',
      url: './pages/about.html',
    },
    {
      name: 'news',
      path: '/news/',
      url: './pages/news.html',
      options: {
        animate: false,
      },
    },
    {
      name: 'users',
      path: '/users/',
      templateUrl: './pages/users.html',
      options: {
        context: {
          users: ['John Doe', 'Vladimir Kharlampidi', 'Timo Ernst'],
        },
      },
    },
    {
      on: {
        pageAfterIn: function test (e, page) {
          // do something after page gets into the view
        },
        pageInit: function (e, page) {
          // do something when page initialized
        },
      },
    },
    // Default route, match to all pages (e.g. 404 page)
    {
      path: '(.*)',
      url: './pages/404.html',
    },
  ],
});
```

Fuente: <https://framework7.io/docs/routes.html>

Framework7 cuenta con gran cantidad de componentes para la construcción y diseño de las interfaces de las aplicaciones móviles, junto con componentes de navegación simulan muy bien la experiencia de usuario nativa de dispositivos Android e IOS, pero solo es un framework visual, el acceso al hardware se hace mediante el framework Apache Cordova.

Algo a tener en cuenta es que framework7 permite una fácil integración con solo con los marcos de trabajo, con los cuales cuenta con un núcleo de desarrollo, sino con gran cantidad de tecnologías y librerías que usen Javascript.

Acceso al hardware del dispositivo

Primero un proyecto de framework 7 debe ser integrado con un proyecto de Apache Cordova, esta integración se logra agregando los archivos de framework7 al proyecto de Cordova.

El acceso al hardware del dispositivo se hace a través de los plugins de Apache Cordova, primero estos deben ser integrados mediante el comando `cordova plugin add "nombre del plugin"`, después se importan en el archivo donde se desea usar. Cada plugin tiene métodos y una forma diferente de usarlos, para entender esto la documentación oficial de Apache Cordova <https://cordova.apache.org/plugins/> proporciona una guía de uso.

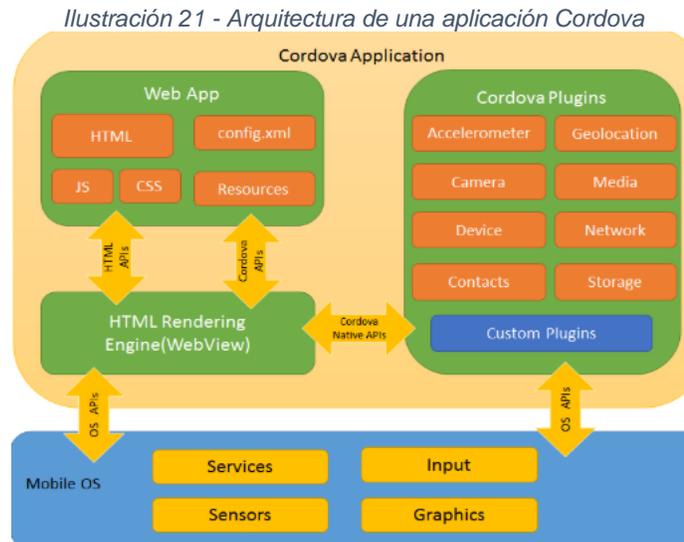
3.5.1.4 PhoneGap

PhoneGap es una versión de Apache Cordova que tienen mayor cantidad de herramientas de desarrollo y pruebas, como aplicaciones de prueba, generador y compilador de proyectos en la computadora, creación de aplicaciones en la nube, entre otras.

PhoneGap contiene una serie de librerías que facilitan todas las utilidades que se pueden llegar a necesitar sobre un teléfono. En vez de aprender las librerías propias de cada sistema con sus lenguajes, se hace uso de aquellas que proporciona el framework, usando un único lenguaje de programación, JavaScript. Las aplicaciones de PhoneGap se ejecutan sobre un componente interno de los dispositivos móviles llamado WebView que no es más que el navegador integrado que tienen los dispositivos móviles, el cual soporta las tecnologías sobre las cuales se basa el desarrollo de sitios web.

Todo el desarrollo que se hará, será empaquetado para el sistema operativo de elección mediante el comando `cordova build "sistema operativo"`, generando un proyecto en la carpeta platform el cual puede ser ejecutado en el SDK de preferencia según el sistema operativo Xcode, Android.

Este empaquetado es posible gracias a las APIS desarrolladas por el proyecto Apache Cordova, el cual es soporte para gran cantidad de framework que tiene por objetivo el desarrollo de aplicaciones móviles híbridas.



Fuente: <https://cordova.apache.org/docs/en/latest/guide/overview/#architecture>

Como se crea un proyecto en PhoneGap

Para poder crear un proyecto hay dos formas, una es mediante una aplicación de escritorio la cual provee una interfaz para la creación de proyectos en PhoneGap, en este caso solo es descargar el instalador dependiendo el sistema operativo que se utilice, ejecutarlo y seguir los pasos de instalación.

Otra forma de instalar PhoneGap es mediante la consola de comandos, para lo cual se necesita instalar Node.JS lo que permitirá usar el gestor de paquetes NPM, luego de esto ejecutar en la consola `npm install -g PhoneGap@latest`. Una vez instalado PhoneGap, la creación de un proyecto se hace mediante el comando `PhoneGap create "nombre del proyecto"`.

Cuando se crea un proyecto de PhoneGap este genera una estructura de carpetas y ficheros necesarios para el funcionamiento del framework y a medida que se avance el desarrollo del proyecto se irán creando más ficheros conforme sean necesarios. La estructura de un proyecto en PhoneGap es la siguiente:

Config.xml: es el archivo de configuración global del proyecto, permite controlar muchos aspectos del comportamiento de una aplicación, como configuraciones por defecto de la pantalla principal y de la orientación de la pantalla.

Platforms: contiene proyectos compilados de las plataformas a las cuales va destinada la aplicación móvil, se debe tener en cuenta el sistema operativo sobre el cual se genera un proyecto, debido a que, por ejemplo, un equipo de Windows no puede compilar un proyecto de IOS.

Plugins: es la carpeta que contiene los plugins instalados en una aplicación, los cuales brindan el acceso a características de hardware de un dispositivo móvil.

www: es la carpeta que contiene la aplicación, con la estructura básica de una aplicación web. Se debe trabajar exclusivamente sobre esta carpeta, cualquier cambio sobre estos archivos se verá reflejado en la aplicación.

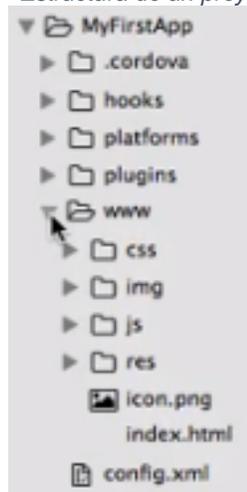
Css: carpeta que contiene los archivos CSS para dar estilos a la aplicación móvil.

Images: contiene los recursos interactivos como imágenes y videos.

Js: carpeta con los archivos JavaScript, encargados de proporcionar la lógica de negocio a la aplicación móvil.

Index.html: archivo raíz que inicializa la aplicación móvil.

Ilustración 22 - Estructura de un proyecto PhoneGap



Fuente. <https://www.youtube.com/watch?v=k5inoYeTKos>

Los archivos a editar se encuentran en la carpeta www y cualquier cambio que se haga sobre estos archivos se verán reflejados en la aplicación.

La secuencia de ejecución de la aplicación es la siguiente:

- La aplicación busca y lee el archivo index.html
- index.html importa los archivos necesarios para su construcción que son:
 - css/index.css: Estilos para la pantalla inicial de la aplicación, estos son estilos globales que se pueden usar en toda la aplicación.
 - PhoneGap.js/cordova.js: Es la librería de PhoneGap que permite la ejecución de la aplicación.
 - js/index.js: Ejecuta el código inicial de la aplicación móvil.
- index.html ejecuta app.initialize() función que inicializa la aplicación.

Una aplicación en PhoneGap es construida únicamente con código HTML, CSS y JavaScript, lo que facilita mucho más el desarrollo, solo se debe pensar en tamaños de pantallas más pequeños. Ya una vez lista la aplicación puede ser probada en el navegador de la computadora mediante el comando PhoneGap serve o mediante la aplicación móvil llamada PhoneGap Developer la cual se encuentra disponible en las tiendas de aplicaciones.

Navegación entre las pantallas de la aplicación

La navegación entre las pantallas de una aplicación en PhoneGap no requiere de ningún componente, la navegación se hace mediante las URLs de las pantallas, es decir de las rutas existentes entre los archivos HTML de la aplicación, por lo que se usa la etiqueta `<a><a/>` con su atributo `href=`” dirección de navegación”.

Ilustración 23 - Navegación entre pantallas en PhoneGap

```
<li><a href="#" class="ui-btn-active">Pantalla Principal</a></li>
<li><a href="pantalla2.html" rel="external">Pantalla 2</a></li>
<li><a href="pantalla3.html" rel="external">Pantalla 3</a></li>
```

Fuente: <http://aprenderaprogramarapps.es/2013/05/30/PhoneGap-navegando-entre-pantallas-en-una-app-web/>

Acceso al hardware del dispositivo

El acceso al hardware del dispositivo se hace a través de los plugins de Apache Cordova, primero estos deben ser integrados mediante el comando `cordova plugin add` “nombre del plugin”, después se importan en el archivo donde se desea usar, quedando listos para usarse. Cada plugin tiene métodos y una forma diferente de usarlos, para entender esto la documentación oficial de Apache Cordova <https://cordova.apache.org/plugins/> proporciona una guía de uso.

3.5.1.5 NativeScript

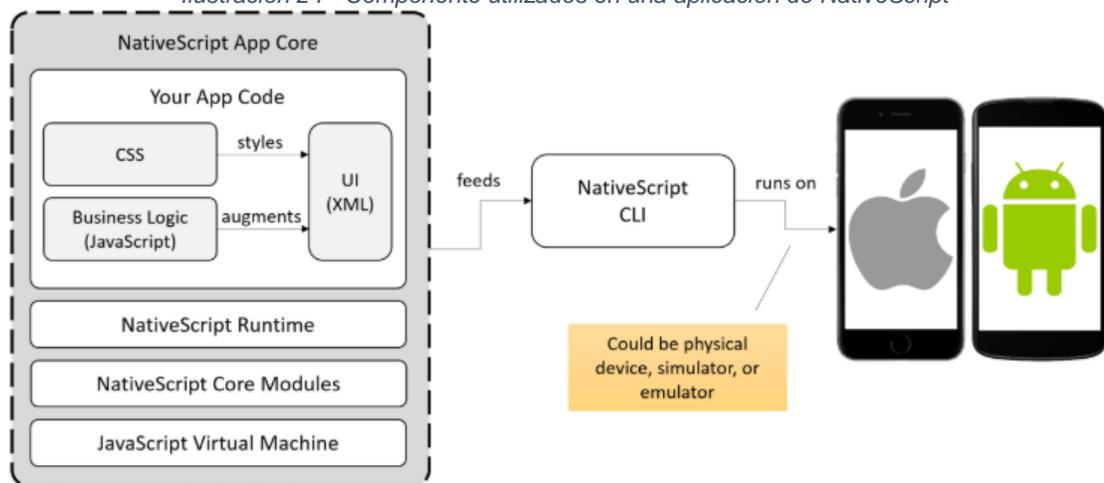
NativeScript es un framework que permite el desarrollo de aplicación es nativas mediante el uso de JavaScript, CSS y XML. En NativeScript no hay un DOM que manipular, se usa XML para las interfaces de usuario. Las aplicaciones de NativeScript se conocen como aplicaciones JIT (Just In Time) o compiladas justo en el momento es decir son compiladas en tiempo de ejecución es decir que “el código fuente es compilado a código maquina inmediatamente antes de ejecutar cada sentencia”.⁴³

⁴³ MERCADO, Alejandro. Desarrollo de aplicaciones móviles con NativeScript. Software Guru [en línea]. [revisado el 2 de noviembre de 2018]. Disponible en Internet: <https://sg.com.mx/revista/53/desarrollo-aplicaciones-m-viles-nativescript>

Tal vez la característica más importante de NativeScript es que puede brindar acceso nativo a las APIs nativas de cada plataforma, mediante el uso de NativeScript RunTime el cual usa la máquina virtual de JavaScript Propia para cada sistema operativo. JavaScriptCore en IOS y V8 en Android.

Estas máquinas virtuales contienen gran cantidad de código y APIs y son usadas por NativeScript para inyectar código que permita el acceso al código nativo del dispositivo, pero a través de código JavaScript. Por lo que no es necesario usar código Java y Objective-c en las aplicaciones desarrolladas por NativeScript.

Ilustración 24 - Componente utilizados en una aplicación de NativeScript



Fuente: <https://sq.com.mx/revista/53/desarrollo-aplicaciones-m-viles-NativeScript>

Como se crea un proyecto en NativeScript

Para crear un proyecto de NativeScript, como todos los anteriores necesita de Node.JS y su gestor de paquetes NPM, usando la consola del sistema se ejecuta el comando `npm install -g NativeScript`, lo que instalara NativeScript en el sistema.

Después de esto cada sistema operativo necesita instalación es y configuración es adicionales las cuales son necesarias para el adecuado funcionamiento de NativeScript, los requerimientos se encuentran detallados en el sitio oficial del framework, <https://docs.NativeScript.org/start/quick-setup>. Después de

completados los requerimientos necesarios para cada sistema operativo, se pueden crear aplicaciones mediante el comando `tns create "nombre de la aplicación"`. Hay dos formas para ver una aplicación corriendo con NativeScript mediante el uso de la aplicación móvil NativeScript Playground la cual permitirá escanear un código QR el cual es generado mediante la ejecución del comando `tns preview`.

Cuando se crea un proyecto de NativeScript puro cuenta con la siguiente estructura:

App: es el directorio que contiene los archivos que componen la aplicación, estos poseen extensiones XML, CSS y JS. Estas extensiones y estructura cambian de acuerdo a como se decida trabajar, si usando JavaScript, TypeScript, Angular o Vue.JS.

App-root.xml: vista que controla la interfaz de usuario, es la vista raíz de la aplicación.

App.css: archivo que controla los estilos de la vista, este es el archivo global de estilos, y todas las pantallas de la aplicación pueden acceder a este.

App.js: es el punto de entrada de la aplicación, es el primer archivo que NativeScript llama y ejecuta.

Carpeta_paginas: contiene las páginas que representan las pantallas de la aplicación

Node_modules: contiene todas las dependencias básicas necesarias para el funcionamiento de la aplicación, también se guardan las que se instalan después que se empieza a trabajar en la aplicación.

Platforms: este directorio está inicialmente vacío. Se rellena cuando se ejecuta el comando `tns platform add "sistema operativo"`. Contiene ficheros que son necesarios para el despliegue de la aplicación tanto en emulador como en dispositivo.

Package.json: archivo que contiene la declaración de dependencias, script y paquetes que se integran en la carpeta `node_modules`, además de algunas configuraciones de la aplicación.

La anterior estructura de carpetas es la que se crea en caso que el proyecto sea puro de NativeScript, pero en caso de utilizar otro marco de trabajo como Angular o Vue.JS la estructura del proyecto cambia para adecuarse a la forma de trabajar de cada uno de ellos.

Las aplicaciones de NativeScript consisten en páginas que representan pantallas en la aplicación, las paginas son instancias de la clase `page`, la interfaz de usuario se diseña en archivos XML y el código de la página es definido en un archivo JavaScript o TypeScript, se debe tener en cuenta que los nombres de los archivos XML y JavaScript deben coincidir para que haya una relación directa. Los estilos se aplican sobre una vista usando el atributo `Class` en las etiquetas de XML con se suele hacer con HTML.

Ilustración 25 - Estructura de una vista en NativeScript

```
<!-- main-page.xml -->
<Page loaded="onPageLoaded">
  <!-- Each page can have only a single root view --
  <StackLayout>
    <!-- content here -->
    <Label text="Hello, world!"/>
  </StackLayout>
</Page>
```

Fuente: <https://docs.nativescript.org/core-concepts/navigation>

Ilustración 26 - Estructura de código JavaScript de una vista

```
// main-page.js
function onPageLoaded(args) {
  console.log("Page Loaded");
}
exports.onPageLoaded = onPageLoaded;
```

Fuente: <https://docs.nativescript.org/core-concepts/navigation>

Navegación entre las pantallas de la aplicación

La navegación entre las vistas se hace mediante la etiqueta Frame o mediante el módulo `tns-core-modules/ui/frame` desde el código. La forma de navegación desde la vista es sencilla, se indica la ruta hacia la cual se desea navegar y un atributo `id` para identificar la etiqueta en caso de querer manejar la navegación desde el código JavaScript.

Ilustración 27 - Etiqueta de navegación

```
<Frame id="firstFrame" defaultPage="home/home-page"/>
```

Fuente: <https://docs.NativeScript.org/core-concepts/navigation>

Ilustración 28 – Modulo de Navegación

```
// e.g. home/home-page.js
const frameModule = require("tns-core-modules/ui/frame");

// Example using `getFrameById(frameId)` to get a `Frame`
// As an alternative, we could use `topmost()` method or `
const frame = frameModule.getFrameById("firstFrame");
frame.navigate("second/second-page");
```

Fuente: <https://docs.NativeScript.org/core-concepts/navigation>

Acceso al hardware del dispositivo

Para usar el hardware nativo de los dispositivos se debe integrar complementos instalándolos mediante npm usando el comando `npm install "nombre del plugin"`, luego este complemento se agrega al proyecto con el comando `tns plugin add "nombre del plugin"`.

Otra manera de usar hardware nativo y la más efectiva de todas es el acceso directo a las APIs nativas, usando JavaScript, mediante el motor de tiempo de ejecución de NativeScript. Lo que permite tener acceso a las nuevas APIs de los dispositivos casi de inmediato.

3.5.2 Mantenibilidad

3.5.2.1 Ionic

Cambios y actualizaciones

Ionic desde sus inicios ha basado su código fuente y el núcleo de su desarrollo en el framework Angular, por lo que su estructura de carpetas y arquitecturas se basan en el mismo. Los desarrolladores de Ionic tienen en cuenta los cambios en el framework Angular para sus futuras actualizaciones.

Ionic maneja un sistema de versionado semántico (SEMVER) el cual le ayuda a manejar de forma adecuada la gran cantidad de cambios y dependencias que el framework requiere, se maneja mediante números de versiones separados por puntos, estos indican el cambio de la versión de Ionic framework de acuerdo a la magnitud de este, su estructura es X.Y.Z.

Ionic actualmente se encuentra en su versión 3.2.1, en donde según lo anterior, el 3 representa la versión principal e indica cambios rupturistas, es decir cambios que no son compatibles con versiones anteriores, a este se le denomina MAJOR. El 2 representa la implementación de nuevas características que son compatibles con la versión actual, se le denomina MINOR y el 1 indica que se resolvió algún error o bug, a esto se le conoce como un PATCH.

Desde la versión 2 el tiempo estimado para el cambio de una versión MAJOR a otra es de 6 meses. Este tiempo es solo una estimación hecha por parte de los desarrolladores.

El generador de código de Ionic está a la escucha de nuevas actualizaciones en el repositorio, por lo que estará indicando el momento de la actualización del framework, pero solo eso, indicar.

Bibliotecas

Ionic framework basa su núcleo en Angular, el cual fue desarrollado usando JavaScript, específicamente TypeScript, el cual al igual que JavaScript otorga

dinamismo a sitios web. TypeScript permite la integración de bibliotecas dentro de su espacio de trabajo para que los desarrolladores puedan expandir las capacidades y alcance de una aplicación.

Ionic mediante TypeScript permite la integración de una amplia variedad de bibliotecas de terceros publicadas en NPM (Node Package Manager), gestor de paquetes por excelencia para proyectos que usen JavaScript, tanto del lado del cliente como del lado del servidor, permitiendo así la reutilización de código y optimización de trabajo.

Gran cantidad de librerías se pueden integrar a proyectos de Ionic, pero la comunidad recomienda tener cuidado con la integración de ciertas librerías, específicamente las librerías que manipulen de manera considerable el DOM (Document Object Model), ya que estas pueden afectar el funcionamiento las aplicaciones, ejemplos de estas el JQuery librería ampliamente usada en el mundo del desarrollo web.

Escalabilidad

Para dar escalabilidad y eficiencia a las aplicaciones Ionic usa módulos y patrones de diseño, lo cual le permite encapsular el código y reutilizarlo las veces que sea necesario, evitando la necesidad de repetir código.

En cuanto a patrones de diseño Ionic trabaja con el patrón de diseño MVVM (Modelo Vista Vista Modelo), el cual le ayuda a manejar la interacción de elementos entre los componentes de Ionic, basados en los componentes de Angular el cual usa el conocido Data Binding, que permite una comunica constante entre el modelo y la vista. Esta comunicación entre el modelo y la vista se hace mediante clases de TypeScript y de Hojas HTML.

En cuanto a interacciones con el servidor, Ionic tiene destinado un tipo especial de clases llamadas providers los cuales son los que se encargan de todas las peticiones con el servidor, centralizando la petición de información externa de las aplicaciones.

3.5.2.2 React Native

Cambios y actualizaciones

React Native maneja un sistema de versionamiento semántico (SEMVER) al igual que Ionic framework, el cual le ayuda a manejar adecuadamente sus versiones, cuenta con la estructura X.Y.Z en donde la X representa la versión MAJOR, Y la versión MINOR y z el PATCH.

Si se observa la versión actual del framework es la 0.57.2, indicando que React Native aún no cuenta con una versión MAJOR, pero el hecho que cuente con una versión 57 MINOR refleja la cantidad de trabajo y tiempo que los desarrolladores han invertido sobre el framework para darle madurez y estabilidad.

React Native no tiene un tiempo definido en cuanto a la liberación de una versión, y las nuevas versiones se lanzan cada dos o tres meses, tiempo corto debido a que los cambios a los cuales se comete actualmente el framework esta enfocados en el aumento de la funcionalidad, son graduales y son compatibles con versiones anteriores.

React Native no anuncia la disponibilidad de una nueva actualización en su núcleo ocurre por lo que una actualización que permita la compatibilidad completa del framework con nuevas funcionalidades depende del desarrollador que lo usa.

React Native no cuenta con generador de código, pero la comunidad se ha encargado de este problema y ofrecen varios generadores de código para facilitar el desarrollo, eso si algunos son de pago, ejemplos de estos son React Native App Generator y uuid-generator.

Los cambios y las actualizaciones del framework se hacen mediante el comando react-native upgrade, pero con la actualización, puede que se usen nuevas versiones de las dependencias necesarias para su funcionamiento, por lo que deben ser actualizadas a las versiones que se requiera.

Bibliotecas

React Native es un proyecto que se basa en el funcionamiento del framework React.js, el cual tiene mayor tiempo disponible en el mundo del desarrollo y cuenta con gran cantidad de componentes y bibliotecas las cuales son compatibles con el framework, las bibliotecas y los componentes de React Native son desarrollados por la comunidad alrededor de este, pero librerías tradicionales de JavaScript que trabajan con HTML como JQuery no son soportadas ya que este no es un framework que trabaje con el DOM de un documento web, en lugar de eso, React Native permite el desarrollo de aplicaciones que simula comportamiento nativo mediante el uso de JavaScript.

Los desarrolladores del proyecto le apuestan a los sistemas de versionamiento como GitHub en el cual almacenan todo el código fuente de la aplicación, pero también gran cantidad de librerías y paquetes, los cuales pueden ser descargados y manipulados por la comunidad mediante el gestor de paquetes NPM (Node Package Manager), o directamente desde el sitio oficial de GitHub.

Native directory y Awesome React Native son sitios web que ofrecen un directorio con un conjunto de librerías, componentes, templates y estilos disponibles de forma gratuita a la comunidad.

Escalabilidad

React Native hereda el patrón de diseño usado en React.js conocido como flux diseñado por Facebook, el cual fue desarrollado para mejorar la interacción en una arquitectura basada en componentes con comunicación bidireccional, usando estados de almacenamiento y una comunicación en un solo sentido con suscripciones a la escucha de cualquier cambio del estado de una aplicación, usando nuevas tecnologías en el desarrollo como lo es el caso de los observables.

Flux es un patrón de diseño como el MVC o el MVVM, y “su implementación proporciona una arquitectura de datos escalable, para proyectos que manejan una

cantidad de información variable, ya que flux ofrece un máximo control sobre el flujo y el estado de los datos en una aplicación”⁴⁴.

Algo a tener en cuenta es que flux es un patrón de diseño recomendado para proyectos de medianos a grandes, para proyectos pequeños puede ser confuso debido a que se necesita la creación de varios archivos, los cuales podrían llevar a confusiones, pero si se entiende adecuadamente el funcionamiento de flux, también se acopla muy bien con este tipo de proyectos.

3.5.2.3 Framework7

Cambios y actualizaciones

Framework7 es un proyecto independiente y uno de los que menos tiempo lleva en el mercado de los framework para desarrollo de aplicaciones móviles híbridas, pero su creador y desarrollador principal Vladimir Kharlampidi, junto con la comunidad que respalda el framework lo actualizan constantemente, aportándole una mayor cantidad de funcionalidades, rendimiento y atractivo visual.

Framework7 usa un sistema de versionamiento semántico (SEMVER), su primera versión lanzada al público bajo la licencia MIT se hizo a finales del año 2016, siendo esta la versión 1.4.1. Actualmente se encuentra en la versión 3.5.2 lo que refleja la disposición de sus colaboradores en función de que framework7 crezca.

La fecha de liberación de versiones no se encuentra definida, pero el desarrollador principal y líder del proyecto Vladimir Kharlampidi postea en su blog cuando una versión MAJOR se acerca, junto con esto publica las novedades que la versión tendrá e incluso en los foros publica el una guía y el nivel de complejidad que tendría el paso de una aplicación de una versión a otra.

Framework7 no cuenta con un generador de código por lo que el código debe ser escrito a mano, lo mismo sucede con la migración de una versión a otra debe

⁴⁴ DIEZ, Braulio. ¿Por qué REDUX?, Lemon Code- Master en Front End [En Línea], 18 de Junio de 2017 [revisado el 15 Octubre 2018], Disponible en Internet: <http://lemoncode.net/lemoncode-blog/2017/6/18/por-que-redux>

hacerse reemplazando a mano algunos archivos y actualizando algunas dependencias.

Bibliotecas

A diferencia de los dos anteriores framework 7 no cuenta con el respaldo de una gran compañía como Facebook o Google, aun así, framework 7 se basa en JavaScript, por lo que soporta la gran cantidad de librerías disponibles para el lenguaje.

Los desarrolladores de Framework7 con las últimas actualizaciones le apuestan a la integración de framework7 con otros marcos de trabajo como lo son reactJS y vueJS.

Framework7 cuenta con un API de complementos que le permite crear al desarrollador sus propios complementos y extensiones de forma rápida y segura sin alterar el código fuente de la aplicación. La instalación tanto de framework 7 como de librerías para ampliar su funcionalidad se hace mediante el gestor de paquetes NPM, a demás todo el código fuente del framework y de sus Apis se encuentra en GitHub.

Escalabilidad

Cuando se desarrolla una aplicación mediante el uso de framework7, se hace mediante el uso de HTML, CSS, JavaScript y las etiquetas que nos proporciona el framework, sus componentes y los plugins nativos mediante el uso de Cordova, pero en cuanto a un patrón de desarrollo o a una arquitectura en específico, framework7 no dispone de esto, por lo que está en manos del desarrollador proporcionar una estructura de trabajo organizada, entendible y que permita la fácil integración de nuevas funcionalidades.

Debido a esto los desarrolladores del framework le apuestan a la integración con otros marcos de trabajo, permitiendo estructurar y organizar el trabajo de los

desarrolladores, dando la posibilidad de escoger un marco de trabajo con un patrón de diseño y una arquitectura en la cual el desarrollador se sienta cómodo.

3.5.2.4 PhoneGap

Cambios y actualizaciones

PhoneGap usa un sistema de versionamiento semántico (SEMVER), lo que le permite controlar adecuadamente la liberación de versiones a su comunidad, actualmente PhoneGap se encuentra en la versión 8.1.1 siendo liberada con la última versión de Cordova, ya que el núcleo de PhoneGap se basa en este.

La liberación de versiones es constante pero los desarrolladores no anuncian la liberación de nuevas versiones, a excepción de las de tipo MAJOR las cuales afectan de alguna manera la estructura básica del framework, por lo que es indispensable para los desarrolladores. Cordova además de ser la base para PhoneGap, es la base para muchos otros framework y sus interacciones con el hardware de los dispositivos móviles.

Cordova se actualiza constantemente a la par que PhoneGap y es uno de los framework de desarrollo móvil híbrido con mayor tiempo disponibles en el mercado, por lo que lógico que cuente con una gran cantidad de versiones.

PhoneGap cumple con todos los estándares de los navegadores web modernos, usados en los dispositivos móviles los cuales se actualizan constantemente, siendo este el motivo de que PhoneGap se encuentre en versión actual. Se puede decir que PhoneGap y Cordova son el mismo proyecto, pero la diferencia es que uno cuenta con mayor cantidad herramientas de desarrollo y mantenimiento que el otro.

Bibliotecas

PhoneGap al tener como lenguaje base JavaScript es capaz de usar la gran cantidad de bibliotecas disponibles en el lenguaje.

Con el paso del tiempo su comunidad se ha vuelto menos activa, debido a que el framework ha perdido popularidad con el nacimiento de otros los cuales usan

Cordova como intermediario, aun así, cuenta con soporte y actualizaciones constantes de parte de su comunidad, siendo esto reflejo de la comunidad de Cordova la cual es más activa.

PhoneGap permite todas las interacciones con componentes de hardware, mediante la gran cantidad de librerías que su núcleo tiene destinado para este propósito, se puede acceder a componentes tales como la cámara, bluetooth, acelerómetro, brújula, red, notificaciones, archivos, almacenamiento, medios, y gran cantidad de componentes que permiten la construcción de aplicaciones con gran cantidad de funcionalidades.

Escalabilidad

La escalabilidad en un sistema, en este caso en un framework es la capacidad que tiene de crecer e implementar nuevas funcionalidades, sin que esto conlleve a cambios drásticos en el código desarrollado. En PhoneGap esto es una tarea que en gran medida depende el desarrollador, aunque nos da una estructura de carpetas para el trabajo, depende del desarrollador la aplicación de un patrón de diseño o el organizar los archivos de manera que sea fácil su comprensión, su reutilización y que el código no sea difícil de mantener.

EL fuerte de PhoneGap, son sus herramientas de desarrollo en la nube, como un compilador de ejecutables en línea, que evita la necesidad de compilación de los proyectos en el dispositivo móvil directamente, pero también cuentan con una app para emular la aplicación móvil desde el dispositivo móvil mediante el arranque de un servidor en una computadora.

3.5.2.5 NativeScript

Cambios y Actualizaciones

Al igual que los framework anteriores NativeScript maneja un sistema de versionamiento semántico (SEMVER). NativeScript es uno de los framework más recientes en el mundo del desarrollo de aplicaciones móviles híbridas, o como ellos lo prefieren llamar desarrollo cruzado, y cuenta con pocas versiones de su código

debido a que es un framework reciente, pero eso solo fue en sus inicios en la actualidad es estable y sus desarrolladores trabajan constantemente en la actualización de su núcleo.

Su versión actual es la 4.2.0 y el tiempo entre una actualización y la otra no está definido, pero estas se anuncian en el sitio oficial de NativeScript y en sus comunidades oficiales. NativeScript usa el gestor de paquetes npm para instalarse e instalar sus dependencias, el comando para su instalación es `npm install -g NativeScript`, con este se instala el código del núcleo de NativeScript y el generador de código, pero para que la instalación este completa, es necesario cumplir con requerimientos específicos de acuerdo a cada plataforma, esta información se encuentra disponible en la documentación oficial.

NativeScript permite la integración de código nativo, como Java, Objective-c, Swift o Kotlin, incluso se integra bien con gestores de paquetes como gradle, maven o CocoaPods.

Bibliotecas

NativeScript está construido mediante código JavaScript, por lo que puede usar gran cantidad de librerías de este lenguaje de programación, a excepción de las librerías que tengan dependencia con el DOM del navegador, como por ejemplo JQuery, esto se debe a que la vista de una aplicación hecha con NativeScript esta se hace mediante XML.

Los desarrolladores de NativeScript trabajan constantemente la integración de y el soporte de librerías y framework de terceros, es tanto así que ya en estos momentos soporta Angular y Vue.JS, dando opciones a los desarrolladores para que escojan el patrón de desarrollo con el cual se sientan cómodos.

NativeScript cuenta con una característica llamada reflexión dia cero la cual permite el uso de APIS nativas de los dispositivos móviles, casi inmediatamente, obteniendo información y metadatos de estas, para manejarlos desde NativeScript mediante código JavaScript.

Escalabilidad

NativeScript se basa en un modelo por componentes, lo cual permite dividir de manera eficaz tareas y trabajo, permitiendo construir y administrar código compartido e incentivar el trabajo colaborativo, pero el objetivo principal del modelo por componentes es la división y reutilización.

Al dividir las funcionalidades y centralizar los controladores, el mantenimiento del código se hace sencillo, y el tratamiento de los errores no se hace extenso, se focaliza en el componente que presenta el problema y se buscan soluciones sobre este sin afectar los componentes que se encuentran funcionando adecuadamente.

Es lo permite crecer a una aplicación basada en un modelo por componentes, en el caso de NativeScript usa el modelo por componentes y el lenguaje JavaScript, pero se puede mejorar la mantenibilidad de los desarrollos usando TypeScript que proporciona corrección de errores, permitiendo un proceso de depuración menos complejo.

Adicionar un framework como Angular o vue.JS en proyectos de mayor complejidad puede hacer la diferencia con un código limpio, escalable y mantenible, y NativeScript le apuesta a la integración de bibliotecas de terceros con su sencilla manipulación de componentes Nativos de Hardware.

3.5.3 Usabilidad

La usabilidad es un término que abarca una gran cantidad de conceptos y características, de los cuales no todos son aplicables al momento de evaluar un framework, por lo que el análisis del criterio usabilidad estará limitado a los siguientes aspectos: facilidad de aprendizaje, documentación proporcionada, foros y comunidades activas.

3.5.3.1 Ionic

Facilidad de aprendizaje

La versión actual de Ionic es la 3, y aunque muy pronto estará disponible versión 4, según la comunidad aun es inestable y no se recomienda el desarrollo de aplicaciones; pero la versión actual Ionic basa su desarrollo de aplicaciones móviles híbridas en 3 tecnologías, HTML, CSS y JavaScript, pero más específicamente HTML, CSS y TypeScript que es un supe conjunto de JavaScript, es decir JavaScript con corrección de errores y tipado fuerte por mencionar algunas cosas.

Para poder desarrollar aplicaciones más complejas y con muchas más funcionalidades, es necesario conocer acerca del framework Angular 2+ y el patrón de diseño por componentes, ya que Ionic basa su funcionamiento en esto. A pesar que se deba tener conocimiento previo de otras tecnologías para manejar adecuadamente el framework, su aprendizaje es sencillo, conociendo las tecnologías de la web el aprendizaje es sencillo y veloz.

Al momento de compilar y probarlo en el dispositivo móvil se puede hacer mediante un entorno de desarrollo o usando las herramientas como Ionic DevApp.

En producción el proyecto requiere unos cuantos pasos para generar el instalador correspondiente dependiendo de la plataforma.

Documentación

Ionic posee una documentación en su sitio web oficial <https://ionicframework.com/docs/> la cual cuenta con todo lo necesario para entender su funcionamiento, con ejemplos reales, componentes, guías e incluso demostraciones reales de funcionamiento. Su documentación se encuentra totalmente en idioma ingles y se encuentra estructurada de forma que la búsqueda de información se rápida y efectiva.

Foros y comunidades activas

Ionic cuenta con una gran comunidad, la cual hereda del framework Angular el cual es mantenido por Google LLC. Cuenta con un foro y un blog oficial en donde los desarrolladores pueden hacer sus publicaciones y compartir sus plugins y nuevos desarrollos. Los enlaces al foro y el blog oficiales son los siguientes <https://forum.ionicframework.com/> y <https://blog.ionicframework.com/> respectivamente.

Cuenta con una tienda oficial donde los desarrolladores pueden vender sus productos o pueden ofrecerlos gratuitamente a la comunidad <https://market.ionicframework.com/>. A demás si se crea una cuenta en el sitio oficial del framework se puede elegir planes de desarrollo, el básico y el pro. La diferencia entre el básico y el pro radica en la cantidad de herramientas de desarrollo disponibles en la plataforma, un ejemplo es la posibilidad de hacer desarrollos en línea si necesidad de hacer instalaciones locales o herramientas de despliegue y pruebas.

En cuanto a educación desarrolladores de todo el mundo ya usan Ionic y ponen a disposición curiosos por esta tecnología cursos, manuales videos tanto públicos como pagos, al ser una tecnología con varios años de experiencia ya es lo suficientemente madura como dar soporte suficiente a los nuevos interesados en aprender.

Ionic cuenta con su código fuente a disposición en el sistema de repositorios GitHub donde la comunidad puede revisarlo hacerle mejores y hacer los cambios que consideren convenientes para mejorar el framework.

3.5.3.2 React Native

Facilidad de aprendizaje

React Native es una de las tecnologías del momento, actualmente su versión en la 0.57, en la actual versión React Native trabaja con JavaScript y código JSX que es

una combinación entre HTML y JavaScript el cual al final termina por convertirse en código JavaScript, entonces, el conocimiento de JavaScript es fundamental.

Pero si se desea crear aplicaciones con muchas más funcionalidades y mayor complejidad es necesario conocer la librería React JS y los patrones de diseño Redux o flux, los cuales van de la mano para un desarrollo óptimo y organizado. Lo anterior hace que el aprendizaje del este, sea sencillo y rápido, más si ya se ha trabajado con tecnologías en la web.

Para probar la aplicación desarrollada antiguamente era necesario usar siempre un entorno de desarrollo como Xcode o Android Studio, pero con la última versión, nace un framework que permite utilizar, administrar, arrancar y compilar aplicaciones en React Native llamado expo.

Ya en producción el proyecto requiere unos cuantos pasos para generar el instalador correspondiente dependiendo de la plataforma. Sin la ayuda de expo framework sería necesario conocer acerca de los lenguajes nativos sobre los cuales deseamos generar un ejecutable, pero son conocimientos básicos en cada lenguaje así que no es una gran preocupación.

Documentación

React Native cuenta con una documentación amplia con totalmente en inglés, con ejemplos de aplicaciones, guías, componentes y plugins en su sitio oficial <https://facebook.github.io/react-native/docs/>.

Foros y comunidades activas

React Native cuenta con una gran comunidad de desarrolladores que aportan día a día en su crecimiento, su comunidad hereda la comunidad de React js y por consiguiente la comunidad de Facebook, que es quien da soporte constante y mantiene el proyecto.

Cuenta con un foro y un blog oficial a los cuales se pueden acceder mediante las siguientes direcciones <https://discuss.reactjs.org/t/welcome-react-native->

[community-group/10239](#) y <https://facebook.GitHub.io/react-native/blog/>. Además la comunidad aprovecha los servicios ofrecidos por Facebook y cuenta con gran cantidad de grupos y comunidades dentro de la red social.

Al ser Facebook quien está detrás de React Native, este se vincula a todo un ecosistema el cual le brinda el soporte necesario para estar en constante crecimiento.

En plataformas como udemy, coursera, código facilito, platzi, etc. Ya están disponibles cursos que facilitan el aprendizaje del framework desde cero, además de la gran cantidad de apasionados por las nuevas tecnologías que publican nuevas entradas en blogs con nueva información acerca del uso del framework y todas sus características.

3.5.3.3 Framework 7

Facilidad de aprendizaje

Framework 7 es uno de los framework para desarrollo móvil híbrido más reciente, actualmente se encuentra en la versión 3.4 y es un proyecto de código abierto independiente sostenido por su desarrollador Vladimir Kharlampidi y la comunidad que apoya el proyecto.

El desarrollo en framework 7 se basa en las tres tecnologías básicas de la web JavaScript, HTML y CSS, según su sitio web oficial de framework7 es “Ultra fácil de usar. ¡No más difícil que un sitio web! Intenta empezar y te sorprenderás”.⁴⁵ Framework7 no necesita ningún otro framework de desarrollo para cumplir con su tarea, pero cuenta con soporte para integrar React JS o VueJS para agregar funcionalidades, potencia, orden y simplicidad.

⁴⁵ www.framework7.io. About us. Framework7 [en línea]. [Revisado el 10 de octubre de 2018]. Disponible en Internet : <https://framework7.io/>

Para probar una aplicación basta con correrla en la computadora y si se desea poner en producción la aplicación es necesario comprender el proceso de construcción y de despliegue de las aplicaciones en Cordova.

Documentación

La documentación de framework 7 contiene todo lo referente a la instalación del framework y al uso de todos sus componentes de interfaz de usuario, con ejemplos reales y demostraciones de funcionamiento. La documentación se encuentra totalmente en inglés y está estructurada de manera que cualquier inconveniente con el uso de los componentes de la aplicación sea solucionado rápidamente.

Foros y comunidades activas

Framework 7 a pesar de ser un framework relativamente nuevo, sus desarrolladores, desde el comienzo pensaron en que su código fuente estuviera disponible para toda la comunidad, por lo que, con el objetivo de compartir ideas, desarrollar nuevos plugins y nuevas versiones, framework7 cuenta con un foro y blog y su código fuente disponible sobre la plataforma GitHub, los enlaces a los sitios son los siguientes en el mismo orden mencionado: <http://forum.framework7.io/>, <https://blog.framework7.io/>, <https://GitHub.com/framework7io/framework7>.

Debido al poco tiempo que el framework lleva disponible a la comunidad, su aprendizaje la gran mayoría de veces se limita a la lectura de la documentación, ya que cursos y video tutoriales para el aprendizaje del framework son limitados.

3.5.3.4 PhoneGap

Facilidad de aprendizaje

Se habla siempre de Cordova y de PhoneGap, aclarando nuevamente la única diferencia es que PhoneGap tiene un propietario y funciones de desarrollo en la nube, las cuales tienen un costo, mientras que Apache Cordova no cuenta con esta funcionalidad.

PhoneGap es un framework que permite el desarrollo de aplicaciones híbridas, el cual es uno de los que más tiempo lleva en el mundo del desarrollo de este tipo de aplicaciones, si tenemos en cuenta que se basa completamente en Apache Cordova, actualmente su versión estable es la 3.5.0.

PhoneGap permite el desarrollo de aplicaciones móviles mediante el uso de estándares de la web, JavaScript, HTML y CSS, permitiendo que en un futuro sigan funcionando en nuevos dispositivos, por lo que el aprendizaje de este resulta mucho más sencillo, que conocer acerca del desarrollo de aplicaciones mediante el uso de lenguajes nativos del dispositivo.

PhoneGap genera un proyecto de la plataforma, sobre la cual queremos publicar nuestra aplicación, lo cual implica solo tres líneas de código sobre la línea de comandos, teniendo así la posibilidad de disminuir los tiempos de desarrollo de las aplicaciones.

En la organización de la aplicación es responsabilidad del desarrollador darle un orden ya que trabajar en PhoneGap es como trabajar en un proyecto web y si se crean archivo y archivos sin ningún tipo de organización ni semántica, un proyecto con este framework puede llegar a confundir a nuevos desarrolladores.

Documentación

El framework cuenta con una documentación en su sitio oficial estructurada en inicio rápido, tutoriales referencias y construcción de la aplicación, además la documentación oficial de Apache Cordova igualmente es de utilidad para PhoneGap, el idioma principal de la documentación, pero también se encuentra en japonés, debido a que los desarrolladores de PhoneGap en un inicio eran de origen japonés. La documentación de Apache Cordova cuenta con parte de la documentación en español.

Foros y comunidades activas

A pesar que PhoneGap no saca al mercado una nueva actualización ya hace más de 5 años, la comunidad que lo rodea si es muy activa, PhoneGap cuenta con un foro y un blog oficiales donde la comunidad puede compartir nuevos avances en el desarrollo del framework o publicar sus dudas y opiniones, las siguientes direcciones son las del foro y el blog oficiales respectivamente <https://forums.adobe.com/community/PhoneGap> , <https://PhoneGap.com/blog/>.

Cuando el código fuente fue liberado y estuvo disponible para la comunidad, hubo un gran interés por los desarrolladores de todo el mundo en el framework, hasta el punto que países de habla hispana crearon comunidades, aceptadas oficialmente por adobe el propietario actual de PhoneGap. El siguiente enlace lleva al sitio oficial de PhoneGap España <https://www.PhoneGapSpain.com/> .

PhoneGap cuenta con un repositorio en GitHub el cual contiene todo su código fuente para que la comunidad lo acceda y lo tenga disponible en cualquier momento, incluso el uso de algunas APIS para PhoneGap solo se encuentra documentado en GitHub.

3.5.3.5 NativeScript

Facilidad de aprendizaje

NativeScript es uno de los framework para desarrollo móvil más recientes en el mercado, su primera versión estuvo disponible en 2014 y su última versión estable es la 3.4 disponible desde el 20 de diciembre de 2017.

NativeScript permite el desarrollo de aplicaciones móviles mediante el uso de JavaScript o TypeScript, XML y CSS, NativeScript no usa HTML, usa XML para el desarrollo de sus interfaces de usuario, esto implica que se debe aprender las etiquetas XML y CSS que NativeScript proporciona, y no las que usualmente se usan en desarrollos web con HTML, por lo que los conocimientos acerca de la web son necesarios pero se debe reemplazar el HTML con XML con etiquetas personalizadas.

Native script se puede integrar con otros framework para mayor organización a nivel de directorios, se puede integrar con Angular, VueJS y Jquery Mobile.

Para el desarrollo con NativeScript puede que sus componentes no estén totalmente optimizados en cuanto a la interfaz de la aplicación por lo que, un poco de conocimiento acerca de los lenguajes nativos del dispositivo se hace necesario, aunque según su sitio oficial, se pretende eliminar este comportamiento, pero que si el desarrollador desea usar lenguaje nativo para el desarrollo de su aplicación lo pueda usar sin problemas.

Cuenta con una plataforma en línea la cual le permite hacer desarrollos y despliegues ágiles llamada Platform Telerik la cual no solo se basa en desarrollo de NativeScript, sino de otras tecnologías como Xamarin, Jquery Mobile, JavaScript, ASP.Net, JSP, PHP y muchas otras tecnologías, pero la plataforma actualmente no se encuentra disponible debido a que la están actualizando y cambiando completamente.

Documentación

NativeScript cuenta con una documentación completa con mucha información sobre cómo empezar, guías de inicio, conceptos básicos, interfaces de usuario. También cuenta con ejemplos, tutoriales, foros y canales en foros de programación para que los que desean aprender NativeScript tengan como hacerlo y no perder el tiempo explorando en la web. Su documentación está totalmente en inglés y se encuentra distribuida de forma que si se desea usar tecnologías adicionales al núcleo de NativeScript se pueda encontrar la información dentro de la misma documentación.

Foros y comunidades activas

A pesar de ser un framework con poco tiempo en el mercado, cuenta con varias comunidades oficiales y foros a los cuales tiene su espacio oficial donde los desarrolladores pueden publicar sus inquietudes, además al contar con una licencia Apache sus desarrolladores tienen publicado su código en la plataforma de repositorios online GitHub.

En los siguientes enlaces se pueden encontrar el foro, el blog y el código fuente de GitHub respectivamente. <https://discourse.NativeScript.org/> , <https://www.NativeScript.org/blog> , <https://GitHub.com/NativeScript> .

3.5.4 Portabilidad

3.5.4.1 Ionic

Ionic fue diseñado para diseñar aplicaciones móviles mediante el uso de TypeScript HTML y CSS, usando Apache Cordova se empaqueta la aplicación para que sea compilada en la plataforma que se desea, entre las plataformas a las cuales están dirigidos los desarrollos en Ionic framework se encuentran: Android, IOS, Windows Phone, Web, Amazon FireOS, Blackberry 10, FirefoxOS.

Los sistemas operativos sobre los cuales Ionic puede construir proyectos, dependen de Apache Cordova y el soporte que este tenga en diferentes sistemas operativos.

3.5.4.2 React Native

React Native basa su desarrollo en el lema, aprende una vez, escribe en donde sea (learn once, write anywhere) permitiendo reutilizar el código escrito en JavaScript una vez, en muchos más desarrollos. React Native no corre la aplicación sobre un webView, como otros framework como Apache Cordova, en cambio el usa un puente nativo que está escrito en c++, que permite una comunicación serializada mediante JSON. Particularidad que lo distingue de otros framework híbridos.

Este puente nativo permite la conexión con aplicaciones desarrolladas para sistemas operativos Android e IOS, pero no de la web.

3.5.4.3 Framework 7

Framework 7 fue desarrollado originalmente para hacer aplicaciones para el sistema operativo IOS, por lo que los componentes de la interfaz de usuario están optimizados para esta plataforma, pero con el paso del tiempo también desarrollo componentes para el soporte de Android mediante el estilo de diseño de Material Design, e incluso si el desarrollador tiene una buena fundamentación en los estilos del desarrollo web los desarrollos pueden ser pasados como una web app.

3.5.4.4 PhoneGap

PhoneGap basa su núcleo de funcionamiento en Apache Cordova, por no decir que es absolutamente los mismo. PhoneGap empaqueta y construye aplicaciones para las siguientes plataformas: Android, IOS, Windows Phone, Amazon FireOS, Blackberry 10, FirefoxOS, Web, Symbian, Bada.

3.5.4.5 NativeScript

NativeScript permite crear aplicaciones para IOS y Android mediante el uso de JavaScript, según los desarrolladores del framework las aplicaciones desarrolladas con este framework se comportan completamente como aplicaciones nativas, ya que usa APIS que son propias de cada uno de los sistemas operativos, gracias a su sistema de reflexión del día cero.

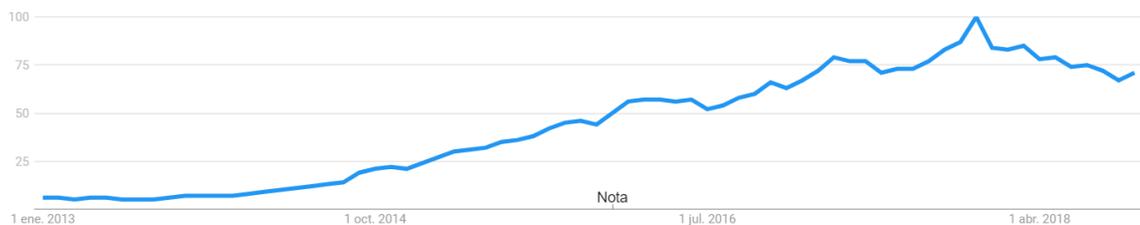
3.5.5 Índices de interés

3.5.5.1 Ionic

Para el caso de estudio se consultó en Google Trends los siguientes términos: Ionic, Ionic 2, Ionic 3 y Angular, términos que tienen relación directa con el framework, además las consultas se encuentran dentro del contexto nacional e internacional. Los resultados de la consulta se muestran en las gráficas a continuación.

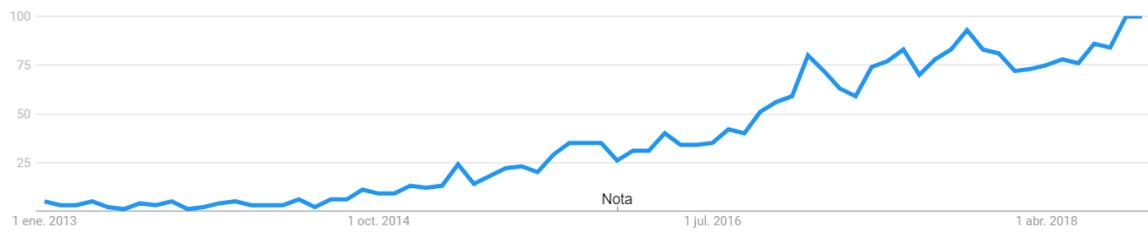
Búsqueda por el término Ionic:

Ilustración 29 - Interés en Ionic a nivel internacional 2013-2018



Fuente: <https://trends.google.es/trends/explore?date=2013-01-01%202018-10-14&q=ionic>

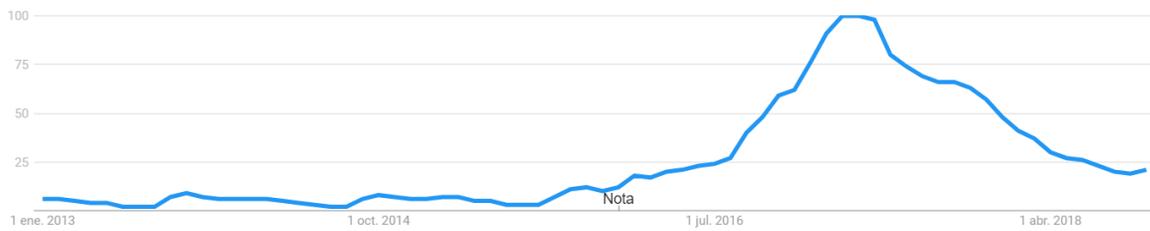
Ilustración 30 - Interés en Ionic a nivel nacional 2013-2018



Fuente: <https://trends.google.es/trends/explore?date=2013-01-01%202018-10-14&geo=CO&q=Ionic>

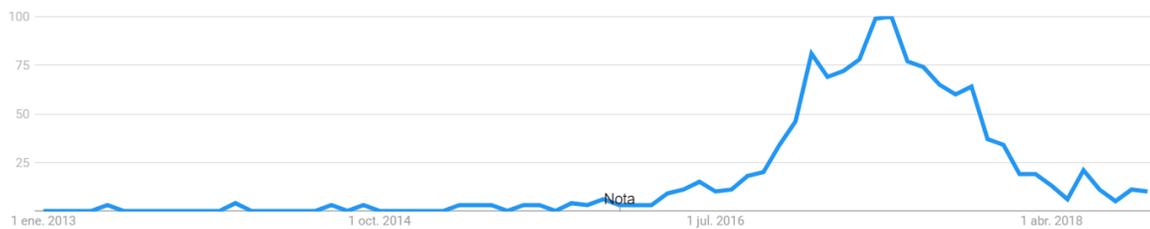
Búsqueda por el término Ionic 2:

Ilustración 31- Interés en Ionic 2 a nivel internacional 2013 - 2018



Fuente: <https://trends.google.es/trends/explore?date=2013-01-01%202018-10-14&geo=CO&q=Ionic%202>

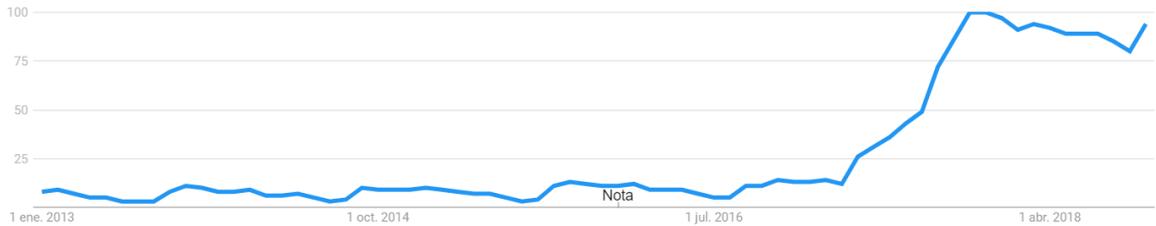
Ilustración 32 - Interés en Ionic 2 a nivel nacional 2013 - 2018



Fuente: <https://trends.google.es/trends/explore?date=2013-01-01%202018-10-14&geo=CO&q=Ionic%202>

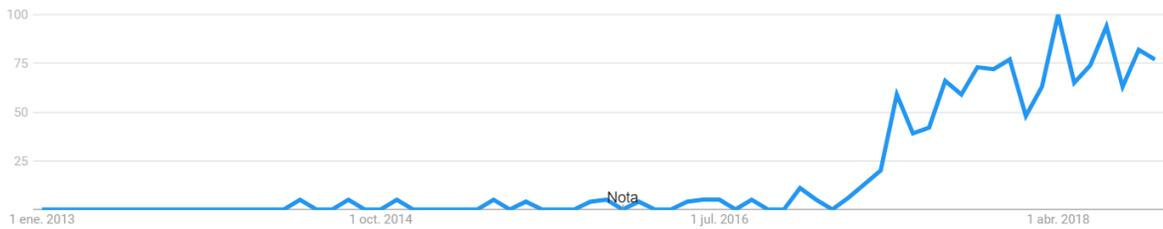
Búsqueda por el término Ionic 3:

Ilustración 33 – Interés en Ionic 3 a nivel internacional 2013-2018



Fuente: <https://trends.google.es/trends/explore?date=2013-01-01%202018-10-14&q=ionic%203>

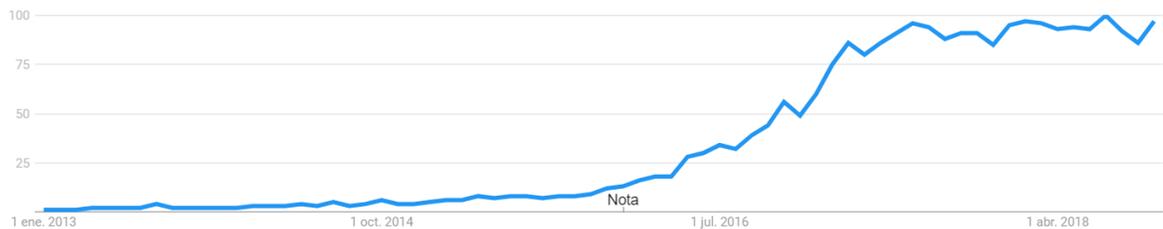
Ilustración 34 - Interés en Ionic 3 a nivel nacional 2013-2018



Fuente: <https://trends.google.es/trends/explore?date=2013-01-01%202018-10-14&geo=CO&q=ionic%203>

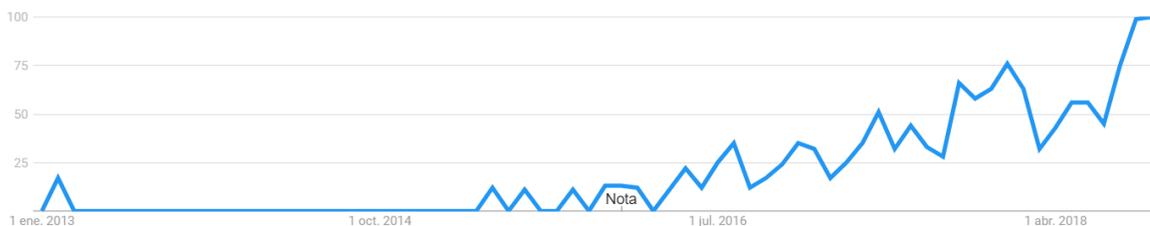
Búsqueda por el término Angular:

Ilustración 35 - Interés en Angular a nivel internacional 2013-2018



Fuente: <https://trends.google.es/trends/explore?date=2013-01-01%202018-10-14&q=%2Fg%2F11c6w0ddw9>

Ilustración 36 - Interés en Angular a nivel nacional 2013-2018



Fuente: <https://trends.google.es/trends/explore?date=2013-01-01%202018-10-14&geo=CO&q=%2Fq%2F11c6w0ddw9>

3.5.5.2 React Native

Para el caso de estudio se consultó en Google Trends los siguientes términos: React, React Native, términos que tienen relación directa con el framework, además las consultas se encuentran dentro del contexto nacional e internacional. Los resultados de la consulta se muestran en las gráficas a continuación.

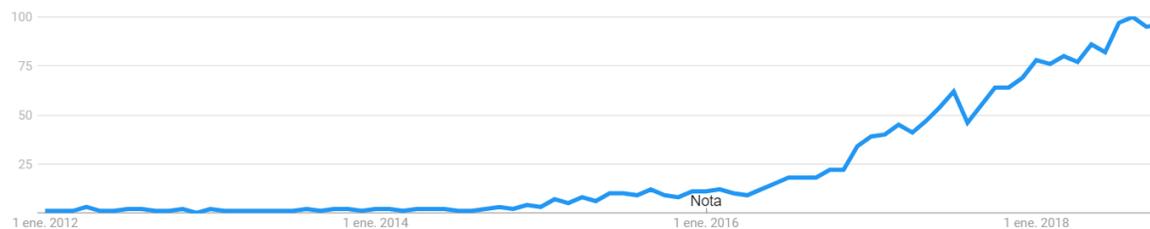
Búsqueda por el término React:

Ilustración 37 – Interés en React a nivel internacional 2012-2018



Fuente: <https://trends.google.es/trends/explore?date=2012-01-01%202018-10-14&q=React>

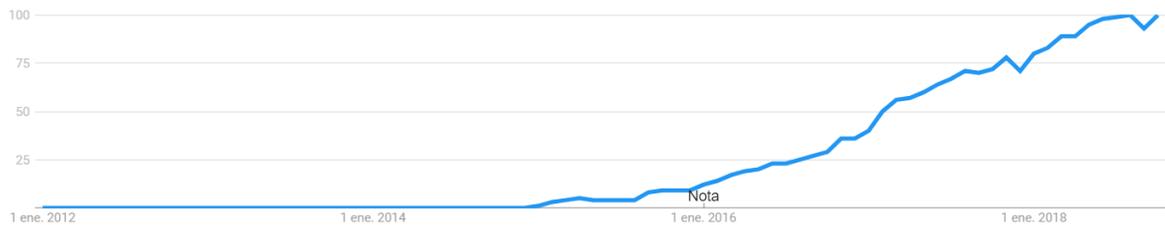
Ilustración 38 - Interés en React a nivel nacional 2012-2018



Fuente: <https://trends.google.es/trends/explore?date=2012-01-01%202018-10-14&geo=CO&q=React>

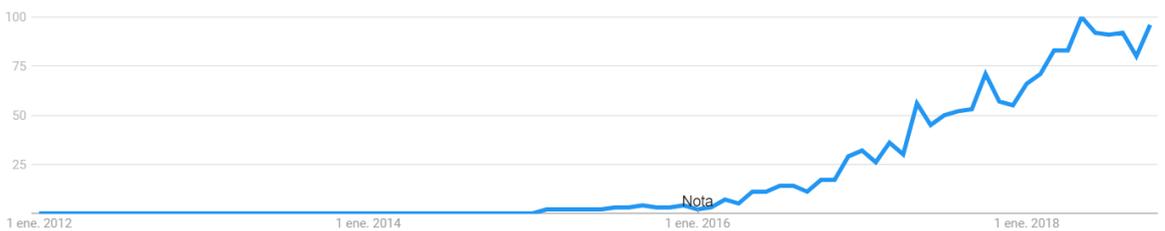
Búsqueda por el término React Native:

Ilustración 39 - Interés en React Native a nivel internacional 2012-2018



Fuente: <https://trends.google.es/trends/explore?date=2012-01-01%202018-10-14&q=React%20Native>

Ilustración 40 - Interés en React Native a nivel nacional 2012-2018



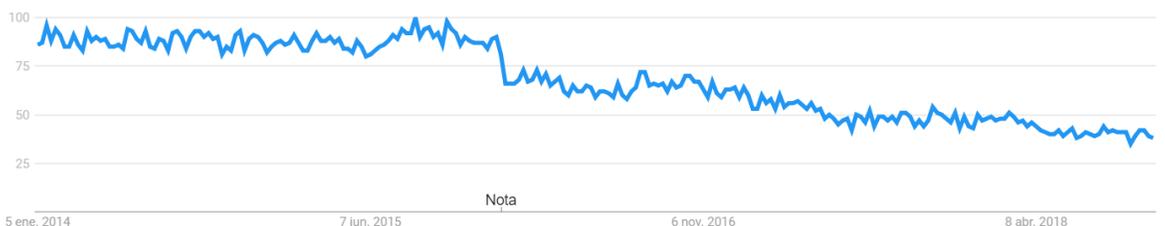
Fuente: <https://trends.google.es/trends/explore?date=2012-01-01%202018-10-14&geo=CO&q=React%20Native>

3.5.5.3 Framework 7

Para el caso de estudio se consultó en Google Trends el término framework 7, término que tienen relación directa con el framework, además las consultas se encuentran dentro del contexto nacional e internacional. Los resultados de la consulta se muestran en las gráficas a continuación.

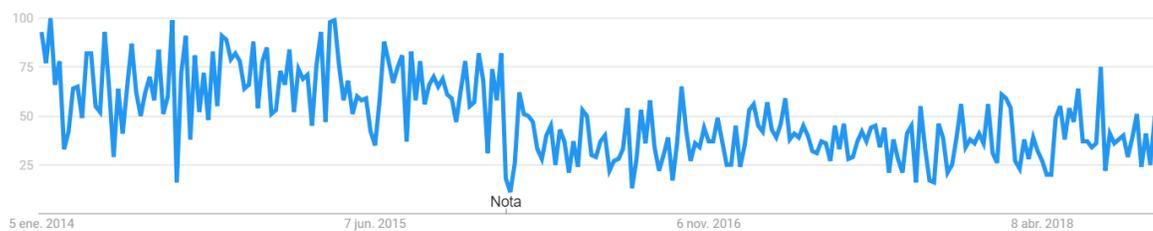
Búsqueda por el término framework 7:

Ilustración 41 – Interés en Framework 7 a nivel internacional 2014-2018



Fuente: <https://trends.google.es/trends/explore?date=2014-01-01%202018-10-14&q=framework%207>

Ilustración 42 – Interés en Framework 7 a nivel internacional 2014-2018



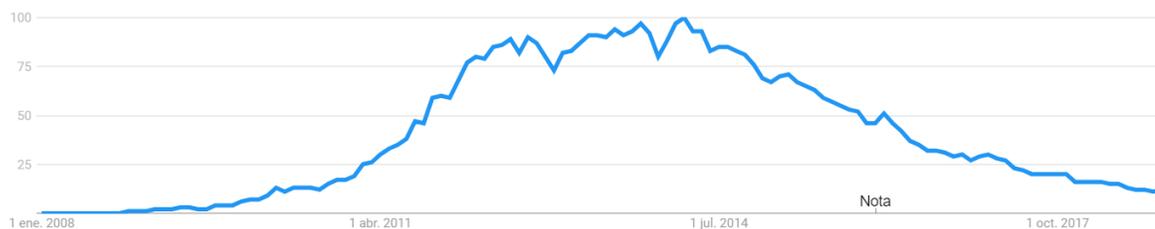
Fuente: <https://trends.google.es/trends/explore?date=2014-01-01%202018-10-14&geo=CO&q=framework%207>

3.5.5.4 PhoneGap

Para el caso de estudio se consultó en google trends los siguientes términos: PhoneGap, Apache Cordova, términos que tienen relación directa con el framework, además las consultas se encuentran dentro del contexto nacional e internacional. Los resultados de la consulta se muestran en las gráficas a continuación.

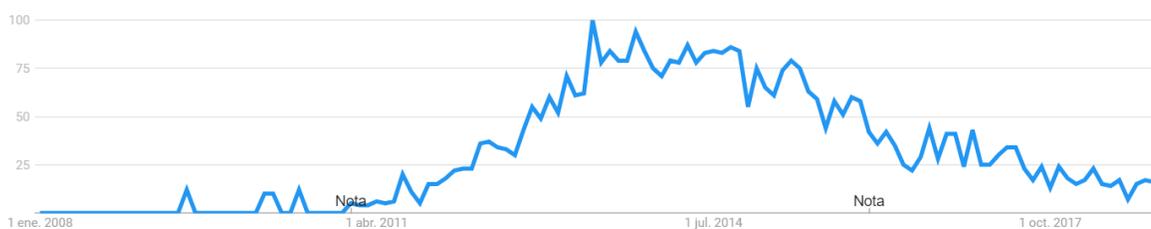
Búsqueda por el término PhoneGap:

Ilustración 43 – Interés en PhoneGap a nivel internacional 2008-2018



Fuente: <https://trends.google.es/trends/explore?date=2008-01-01%202018-10-14&q=PhoneGap>

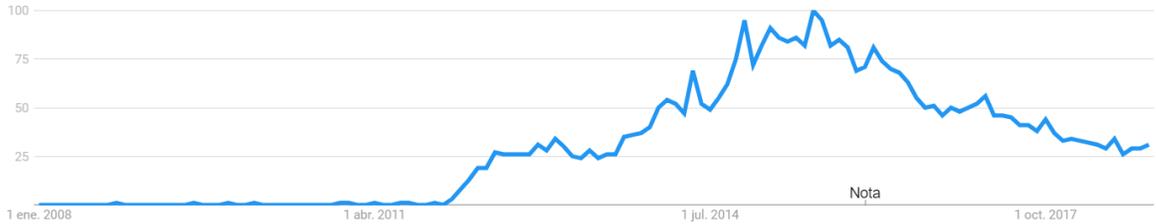
Ilustración 44 - Interés en PhoneGap a nivel nacional 2008-2018



Fuente: <https://trends.google.es/trends/explore?date=2008-01-01%202018-10-14&geo=CO&q=PhoneGap>

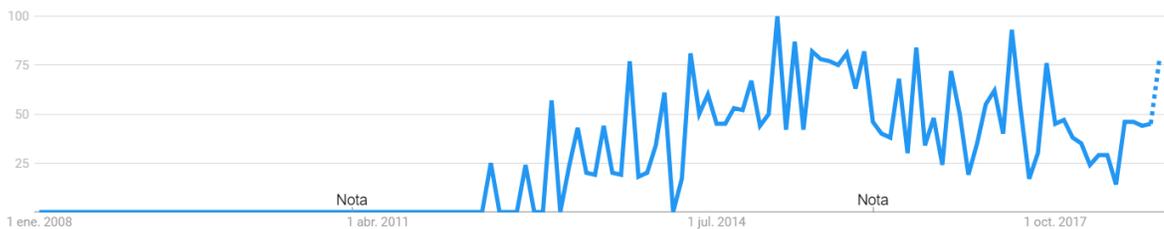
Búsqueda por el término Apache Cordova:

Ilustración 45 – Interés en Apache Cordova a nivel internacional 2008-2018



Fuente: <https://trends.google.es/trends/explore?date=2008-01-01%202018-10-14&q=%2Fm%2F06znsr5>

Ilustración 46 – Interés en Apache Cordova a nivel nacional 2008-2018



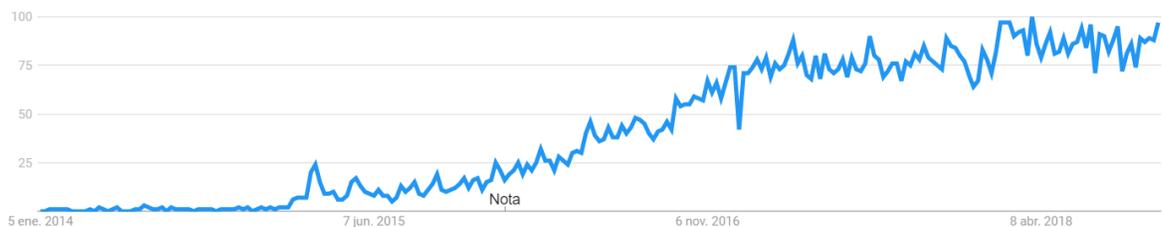
Fuente: <https://trends.google.es/trends/explore?date=2008-01-01%202018-10-14&geo=CO&q=%2Fm%2F06znsr5>

3.5.5.5 NativeScript

Para el caso de estudio se consultó en Google Trends el término NativeScript, término que tienen relación directa con el framework, además las consultas se encuentran dentro del contexto nacional e internacional. Los resultados de la consulta se muestran en las gráficas a continuación.

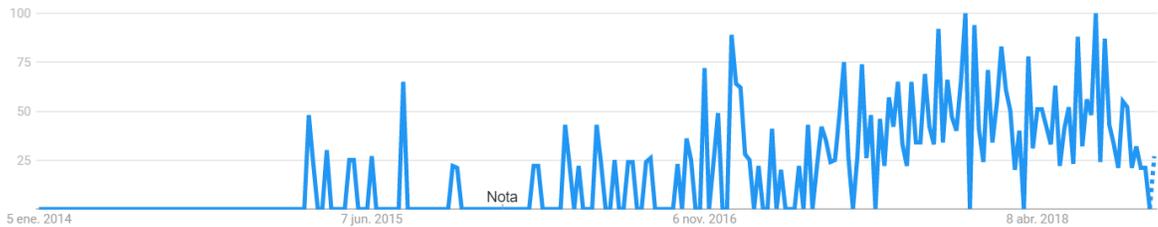
Búsqueda por el término NativeScript:

Ilustración 47 - Interés en NativeScript a nivel internacional 2014-2018



Fuente: <https://trends.google.es/trends/explore?date=2014-01-01%202018-10-14&q=%2Fg%2F11c57wr2yt>

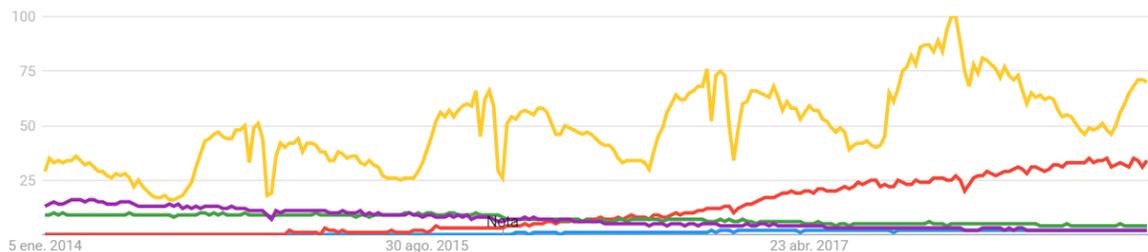
Ilustración 48 - Interés en NativeScript a nivel nacional 2014-2018



Fuente: <https://trends.google.es/trends/explore?date=2014-01-01%202018-10-14&geo=CO&q=%2Fg%2F11c57wr2yt>

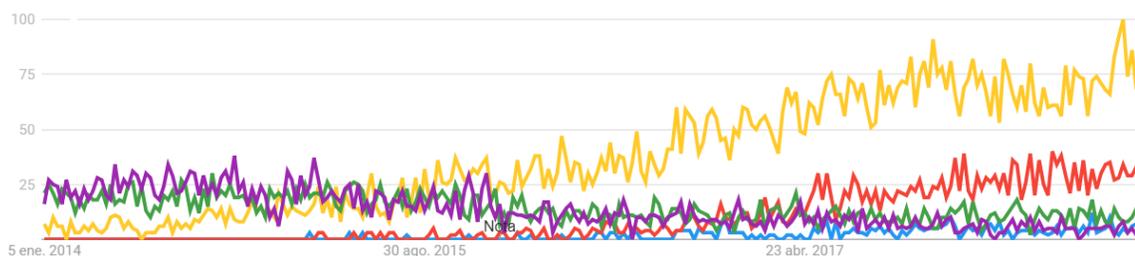
Ahora para dar mayor claridad, al comparar el interés de búsqueda de cada uno de los framework a continuación se muestra un gráfico de líneas, en el cual cada una de las variables representa los framework de desarrollo móvil, con respecto a un periodo de tiempo determinado. La correspondencia entre los colores de las gráficas y los framework es la siguiente: azul – NativeScript, rojo - React Native, amarillo – Ionic, verde - Framework 7, morado - PhoneGap.

Ilustración 49 – Interés de búsqueda en los framework a nivel internacional



Fuente: <https://trends.google.es/trends/explore?date=2014-01-01%202018-10-14&q=NativeScript,React%20Native,Ionic,framework%207,PhoneGap>

Ilustración 50 – Interés de búsqueda en los framework a nivel nacional



Fuente: <https://trends.google.es/trends/explore?date=2014-01-01%202018-10-14&geo=CO&q=NativeScript,React%20Native,Ionic,framework%207,PhoneGap>

3.6 SELECCIÓN DEL FRAMEWORK

A continuación, se determinará en cuales criterios de evaluación un framework es más eficiente que otro o cual cuenta con mayores capacidades y características, se dará una conclusión y se asignará una calificación de acuerdo, criterio personal según lo leído, a mesas de trabajo con los ingenieros del CIADTI y a lo observado a través del proceso de investigación.

3.6.1 Funcionalidad

Todos los framework evaluados permiten la construcción de aplicaciones móviles híbridas, considerando una aplicación híbrida como una aplicación que se desarrolla con un lenguaje ajeno a lenguaje del sistema operativo.

- React Native, Ionic y NativeScript trabajan mediante el uso de componentes lo cual permite la fácil reutilización de código, una mejor estructuración del proyecto y facilidades de integración con múltiples proyectos que trabajen mediante el uso del mismo framework, eso sí cada uno aplica en concepto de componente, pero su implementación es distinta en cada uno. Framework7 y PhoneGap en cambio trabajan HTML y JavaScript de forma tradicional.
- React Native y NativeScript se ejecutan como si fueran una aplicación nativa lo que mejora mucho el rendimiento de la una aplicación híbrida, pero esto implica tener conocimiento del lenguaje nativo del dispositivo, si se desea construir una aplicación compleja. Ionic, Framework7 y PhoneGap no necesitan ningún conocimiento del lenguaje nativo del dispositivo, debido a

que la aplicación se ejecuta sobre un navegador nativo, lo que permite que el desarrollo se haga mediante JavaScript en Framework7, PhoneGap y TypeScript para Ionic.

- Ionic y NativeScript usan el patrón de diseño MVVM similar a el MVC, lo que permite organización y escalabilidad, React Native usa el patrón de diseño flux, el cual permite el control total del flujo de la información en aplicaciones de gran tamaño. Framework7 y PhoneGap no hacen uso de un patrón de diseño en la construcción de las aplicaciones si el desarrollador desea implementar un patrón de desarrollo siempre es posible. Además, NativeScript y Framework7 son los dos únicos que le apuestan a la integración con otros marcos de trabajo.
- La navegación en todos es sencilla y trabaja como una pila de páginas a la cual se empuja y se extrae de acuerdo al comportamiento deseado.
- Todos los framework usan CSS para dar estilos a las pantallas de la aplicación.
- Para la construcción de las vistas de las pantallas de la aplicación Ionic, Framework7 y PhoneGap usan HTML, NativeScript hace uso de XML con etiquetas predefinidas por sus desarrolladores y React Native hace uso de JSX una combinación entre HTML y JavaScript.

Se asignó la puntuación “cumple a cabalidad”, que dentro de la escala normativa equivale a 5, a React Native y NativeScript, debido a que en comparación a los demás framework, estos permiten el acceso a los recursos de hardware de los dispositivos móviles sin la necesidad del uso de un intermediario.

A los demás framework se les asignó una puntuación de “cumple”, que dentro de la escala normativa equivale a 4, ya que su acceso a los recursos nativos del dispositivo se logra a través del intermediario Apache Cordova.

3.6.2 Mantenibilidad

- Ionic cuenta con fechas definidas para sus actualizaciones además de en sus comunidades oficiales hacen anuncios anticipados de actualización, React Native es el único que no cuenta con una versión MAJOR según el sistema de versionamiento SEMVER por lo que desde la primera versión no ha tenido cambios destructivos que afecten se estructura de trabajo. NativeScript también se encuentra en constante actualización ya que le apuesta a la integración con otros marcos de trabajo el camino de los desarrolladores del framework es arduo, y aunque Framework7 también se integra con otros marcos de trabajo su actualización no es tan constante. PhoneGap se basa en Apache Cordova por lo que su actualización es constante.
- Ionic se integra con todas las bibliotecas de JavaScript excepto con las que manipulan el DOM, Framework7 y PhoneGap si se integran con todas las bibliotecas de JavaScript. React Native no se integra con todas estas bibliotecas por lo que depende en gran medida de las desarrolladas por la comunidad, y NativeScript se integra con algunas bibliotecas de JavaScript y con gran cantidad de bibliotecas de los lenguajes nativos.
- Todos al ser framework permiten la escalabilidad de las aplicaciones, pero en Framework7 y PhoneGap esto depende de la forma en que el desarrollador construya la aplicación.

Se asignó la puntuación “cumple a cabalidad”, que dentro de la escala normativa equivale a 5, a React Native, Ionic y NativeScript, la razón es que cuentan con una estructura de ficheros y técnicas de programación unificadas, y aunque no cuenten con un soporte para integrar todas las librerías de JavaScript, sus desarrolladores y comunidades trabajan constantemente en la ampliación de sus funcionalidades.

A los framework PhoneGap y Framework7, se les asigno una puntuación de 3 debido a que no cuentan con técnicas de programación unificadas, las cuales permiten la estandarización del código.

3.6.3 Usabilidad

- La facilidad de aprendizaje es una medida relativa, no se puede determinar cuánto se demora una persona en aprender un framework de desarrollo híbrido, debido a que todas las personas cuentan con capacidades distintas, aprender su arquitectura y como estructuran el trabajo de una tecnología conlleva tiempo y trabajo, pero el tiempo de aprendizaje se hace más extenso si se debe conocer además de las tecnologías de la web el lenguaje nativo de los dispositivos, como en el caso de React Native y NativeScript.
- Todos los framework proporcionan documentación para aprender su manejo, con ejemplos de código, guías y tutoriales, Ionic y Framework7 muestran ejemplos en tiempo real. Ionic, NativeScript y PhoneGap proporcionan herramientas en línea para la construcción ágil de aplicaciones, con algunas funcionalidades pagas.
- Todos los framework cuentan con buenas comunidades, foros y blogs para publicar nuevo contenido y compartir entre los desarrolladores. React Native es apoyado directamente por Facebook, y otros como Ionic y NativeScript indirectamente por Google.

El tiempo de aprendizaje es una medida relativa, este puede variar en función de los conocimientos o capacidades de cada persona, pero la información proporcionada por sus comunidades y documentación oficial, hace de este proceso más sencillo.

Según lo anterior a los framework Ionic y Framework7 se les asigno una puntuación de 5, lo que representa en la escala cualitativa “cumple a cabalidad”. A los demás framework se les asigno una puntuación de 4.

3.6.4 Portabilidad

- En este caso NativeScript, React Native y Framework7 se orientan a ser compilados y ejecutados en Android e IOS. Ionic y PhoneGap están

orientados a más plataformas como Windows Phone, BlackBerry 10, Amazon fireOS, firefoxOS.

En cuanto a la portabilidad, todos los framework presentan un buen comportamiento, algunos permiten la compilación y generación de proyectos en más plataformas que en otros, pero el hecho de que pueda hacerse un desarrollo y usarse múltiples veces, es una gran ganancia del desarrollo móvil híbrido.

Por lo anterior se asignó una puntuación de 5, que en escala cualitativa representa un “cumple a cabalidad”, a todos los framework evaluados.

3.6.5 Índices de interés

- Según las gráficas de búsqueda en Google proporcionadas por Google trends los framework que más se buscan a nivel nacional como internacional son Ionic y React Native.
- Framework como NativeScript y PhoneGap han bajado con el tiempo en sus tasas de búsqueda, y Framework7 sube de a poco con el paso del tiempo.

El interés de búsqueda en el motor de búsqueda google es mayor para los framework Ionic y React Native, siendo esto un reflejo del respaldo de empresas como Facebook y Google, que respaldan directa o indirectamente los proyectos, por lo que se les asignó una puntuación de 5.

A los demás framework se les asignó una puntuación menor debido a que comparados con Ionic y React Native, el interés de las personas por estos no es muy alto.

Tabla 4 - Matriz de pesos llena con los resultados del estudio comparativo

		Framework					
		Ionic	React Native	Framework7	PhoneGap	NativeScript	Subtotal
Criterios	Funcionalidad	4	5	4	4	5	22
	Mantenibilidad	5	5	3	3	5	21
	Usabilidad	5	4	5	4	4	22
	Portabilidad	5	5	5	5	5	25
	Índices de interés	5	5	3	3	3	19
	Subtotal	24	24	20	19	22	

Se evidencia en la matriz de pesos un empate entre los framework Ionic y React Native con un puntaje de 24, por lo que fue necesario dar resolución a esta situación particular.

Para esto se hizo uso de mesas de trabajo con ingenieros del CIADTI en donde se determinó que Ionic era la mejor opción para el desarrollo, ya que el criterio sobre el cual cuenta con menor puntuación con respecto a React Native es la funcionalidad, específicamente en su capacidad de acceder a los recursos de hardware del dispositivo sin el uso de un intermediario como lo es Apache Cordova, lo cual no es algo de gran prioridad para el tipo de aplicación que se desea desarrollar, por lo que se consideró de mayor importancia el criterio de usabilidad, específicamente en la parte de la documentación para el aprendizaje de la tecnología.

Otro factor que se tuvo en cuenta para la selección del framework Ionic fue su estrecha relación con el framework Angular, el cual es usado actualmente en desarrollos dentro del CIADTI.

De la matriz de pesos también se puede concluir que los framework híbridos cobran mayor fuerza en el criterio de portabilidad, es decir la capacidad de ejecutarse en varios sistemas operativos, esto es lógico pensarlo, ya que las aplicaciones móviles híbridas desde un inicio fueron pensadas para ser desarrolladas una sola vez y ejecutarse sobre varios sistemas operativos.

4 CONSTRUCCIÓN DEL PROTOTIPO

La construcción del prototipo se hizo usando Ionic framework versión 3, mediante una metodología para el desarrollo de aplicaciones móviles, la metodología a usar fue explicada en sección metodologías usadas en el desarrollo móvil contenida en el capítulo marco teórico.

Se usó una arquitectura de tres capas contando con una separación bien definida entre el back-end, usando como lenguaje del servidor Java y el front-end, usando como lenguaje del cliente JavaScript, mediante el uso del framework Ionic. El motor de base de datos que se uso es postgresSQL, usado por el CIADTI habitualmente en sus desarrollos. Para la comunicación entre el servidor y el cliente se usó un API Rest, que se basa en el protocolo HTTP.

Al API Rest se le envían peticiones de parte del cliente; validándolas y procesándolas, respondiendo con un objeto en formato JSON, el cual contiene la respuesta de la petición, ya sea un error en caso de que se presente o la información que se necesite para trabajar. El CIADTI ya cuenta con un API Rest, el cual es consumido por la aplicación actual de estudiantes UP, se hizo uso de este para agilizar el desarrollo del prototipo.

Se desarrolló un prototipo en base a la aplicación móvil de estudiantes UP desarrollada por el CIADTI para proporcionar información relevante acerca de las actividades académicas de los estudiantes de la Universidad de Pamplona.

La razón de que el prototipo se haga bajo una metodología es para estructurar y organizar el desarrollo de este, ya que en el futuro el prototipo se proyecta como una aplicación que reemplace a la aplicación móvil actual, incluso que permita la integración de otras aplicaciones de la Universidad de Pamplona como la aplicación de docentes, que es de índole informativo en una sola aplicación, esto es posible gracias a la naturaleza escalable de los framework actuales.

4.1 ANÁLISIS

El proceso de análisis consiste en la obtención de los requerimientos necesarios que permitan el desarrollo de la aplicación, los requerimientos presentados a continuación fueron recolectados a través de mesas de trabajo y revisión de la sección de valores y opiniones de la aplicación estudiantes UP en la tienda de aplicaciones Google Play:

1. Al ingresar a la aplicación, esta solicitara las credenciales de usuario y contraseña los cuales deben estar almacenados a fin de verificar que el usuario tenga un vínculo con la institución.
2. Al usuario validado se le da la opción de seleccionar el programa académico del cual desea saber su información. En caso de que el usuario pertenezca a más de dos programas se le debe dar la posibilidad de escoger.
3. Una vez seleccionado un programa académico el sistema mostrara una página home y los iconos que permitan acceder al mapa, menú de opciones y anuncios
4. Al ingresar al tab o pestaña home se debe poder visualizar el avance del estudiante a través del programa académico y obligaciones del estudiante al día con el programa académico
5. Al ingresar al menú de la aplicación debe ser posible visualizar las opciones a elegir que sean relevantes para el estudiante.
 - a. Se debe poder visualizar el horario de clase del estudiante
 - b. Se debe poder acceder a las calificaciones actuales de los estudiantes, mostrando información de cómo se obtiene esta la calificación.
 - c. Visualizar el histórico de calificaciones del estudiante desde que inició su programa académico hasta donde lleve cursado del mismo.
 - d. Se debe poder visualizar información financiera del estudiante como su liquidación semestral, sus deudas, historial de pago.

- e. Debe contar con una estructura de información que permita visualizar el avance del estudiante a través del programa académico.
 - f. Se debe poder cerrar la sesión.
 - g. Debe contar con un directorio con los números relevantes de la Universidad de Pamplona.
 - h. Si el usuario cursa dos programas académicos se mostrará una opción para cambiar de programa académico, en caso de no cursar dos programas académicos esta opción no debe mostrarse.
6. Al ingresar al tab o pestaña de mapa se debe observar el mapa de la Universidad de Pamplona mediante los mapas de Google Maps
 7. Al ingresar al tab o pestaña notificaciones debe poder mostrar notificaciones en lo referente a compromisos, deudas, recordatorios, felicitaciones.
 8. Mostrar el formato de la hora en am y pm, no en hora militar.

Una vez definidos los requerimientos, estos fueron descritos y representados mediante un diagrama de casos de uso, esto con el fin de tener una descripción de las actividades que un usuario lleva a cabo con la aplicación móvil, a continuación, se muestra el proceso de especificación de requerimientos funcionales y el diagrama de casos de uso que representa el sistema:

Tabla 5 - Especificación de requerimiento validar ingreso

Identificador	RF01
Nombre	Validar Ingreso
Precondiciones	Que el usuario cuente con las credenciales respectivas nombre y usuario
Postcondiciones	El sistema validara credenciales y mostrara programas académicos.
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario abre la aplicación. 2. El sistema muestra formulario solicitando credenciales (usuario y contraseña). 3. El usuario digita credenciales.

	<p>4. El sistema valida las credenciales.</p> <p>5. El sistema redirige para poder escoger el programa académico</p>
Flujo Alternativo	4. a El sistema muestra un mensaje de error en caso de no poder validar las credenciales.

Tabla 6 - Especificación de requerimiento seleccionar programa académico

Identificador	RF02
Nombre	Seleccionar de programa académico
Precondiciones	Que el usuario haya introducido las credenciales de acceso correctas
Postcondiciones	Muestra la información académica del estudiante para el programa académico correspondiente
Flujo Principal	<p>1. El sistema muestra los programas académicos a los cuales pertenece el estudiante.</p> <p>2. El usuario selecciona el programa académico de cual desea saber información.</p> <p>3. El sistema muestra la información del estudiante en el programa académico.</p>
Flujo Alternativo	1. a En caso de solo pertenecer a un programa académico se pasará al paso 3.

Tabla 7 - Especificación de requerimiento pestaña o tab home

Identificador	RF03
Nombre	Tab Home
Precondiciones	Haber ingresado a la aplicación
Postcondiciones	ninguna
Flujo Principal	1. El sistema muestra información académica del estudiante.

	2. El estudiante puede visualizar la información mostrada
Flujo Alternativo	ninguno

Tabla 8 - Especificación de requerimiento pestaña o tab de notificaciones

Identificador	RF04
Nombre	Tab Notificaciones
Precondiciones	Ninguna
Postcondiciones	Ninguna
Flujo Principal	1.El sistema muestra información acerca de notificaciones 2. El estudiante puede visualizar la información mostrada y acceder a los enlaces de interés
Flujo Alternativo	ninguno

Tabla 9 - Especificación de requerimiento pestaña o tab de mapa

Identificador	RF05
Nombre	Tab Mapa
Precondiciones	Ninguna
Postcondiciones	Ninguna
Flujo Principal	1.El sistema muestra mapa con ubicaciones de la Universidad de Pamplona 2. El estudiante puede visualizar la información mostrada.
Flujo Alternativo	ninguno

Tabla 10 - Especificación de requerimiento horario

Identificador	RF06
Nombre	Horario
Precondiciones	Ninguna
Postcondiciones	Se muestra información acerca del horario de clases del estudiante
Flujo Principal	1. El sistema muestra en el menú desplegable la opción de horario. 2. El usuario selecciona la opción del horario

	3. El sistema muestra el horario del estudiante para el semestre actual.
Flujo Alternativo	ninguno

Tabla 11 - Especificación de requerimiento registro extendido

Identificador	RF07
Nombre	Registro Extendido
Precondiciones	Ninguna
Postcondiciones	Ninguna
Flujo Principal	<ol style="list-style-type: none"> 1. El sistema muestra en el menú desplegable la opción de registro extendido. 2. El usuario selecciona la opción de registro extendido 3. El sistema muestra el registro extendido del estudiante hasta el semestre actual.
Flujo Alternativo	ninguno

Tabla 12 - Especificación de requerimiento información financiera

Identificador	RF08
Nombre	Información financiera
Precondiciones	Ninguna
Postcondiciones	ninguna
Flujo Principal	<ol style="list-style-type: none"> 1. el sistema muestra la opción de información financiera. 2. El usuario selecciona la opción de información financiera. 3. El sistema muestra las de información de la opción seleccionada
Flujo Alternativo	ninguno

Tabla 13 - Especificación de requerimiento agenda

Identificador	RF09
Nombre	Agenda
Precondiciones	Ninguna
Postcondiciones	ninguna
Flujo Principal	<ol style="list-style-type: none"> 1. el sistema muestra la opción agenda. 2. El usuario selecciona la opción de agenda. 3. El sistema muestra números telefónicos relevantes para el estudiante
Flujo Alternativo	ninguno

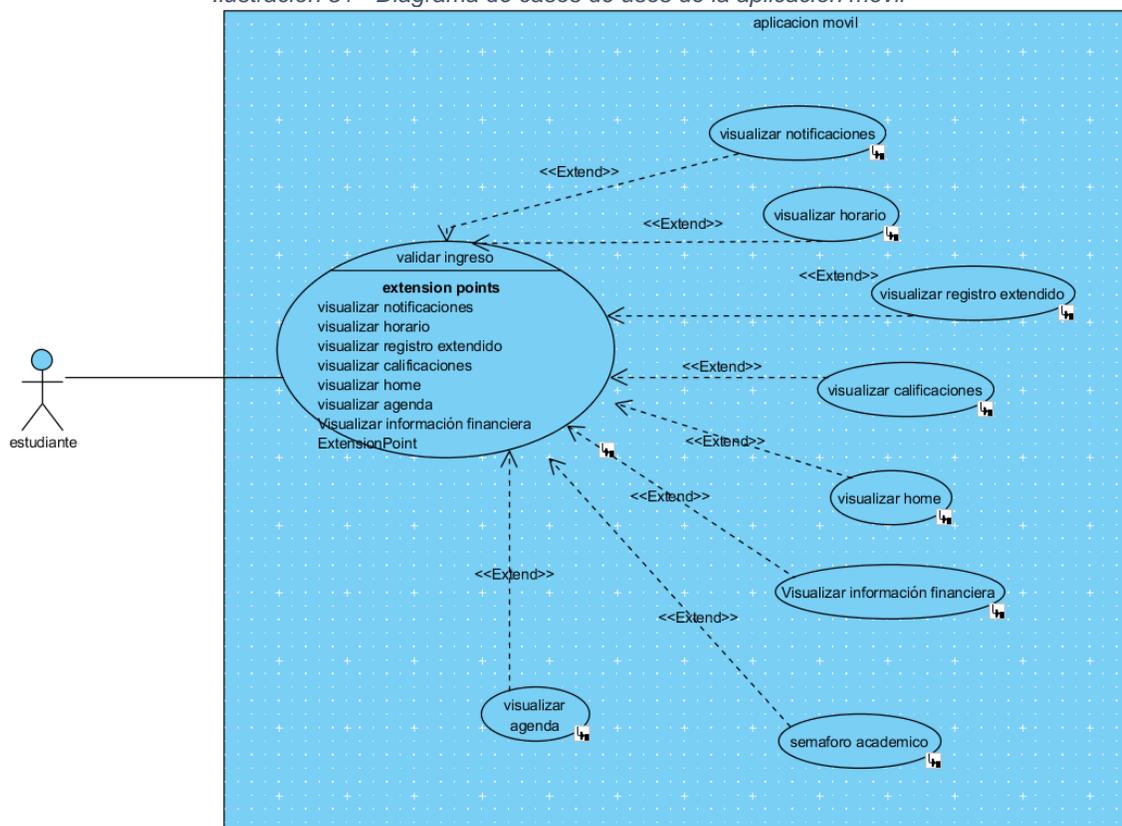
Tabla 14 - Especificación de requerimiento calificaciones

Identificador	RF10
Nombre	Calificaciones
Precondiciones	Ninguna
Postcondiciones	ninguna
Flujo Principal	<ol style="list-style-type: none"> 1. el sistema muestra la opción calificaciones. 2. El usuario selecciona la opción calificaciones. 3. El sistema muestra las calificaciones del estudiante del semestre en curso.
Flujo Alternativo	ninguno

Tabla 15 - Especificación de requerimiento semáforo académico

Identificador	RF11
Nombre	Semáforo Académico
Precondiciones	Ninguna
Postcondiciones	ninguna
Flujo Principal	<ol style="list-style-type: none"> 1. el sistema muestra la opción de semáforo académico. 2. El usuario selecciona la opción de semáforo académico. 3. El sistema muestra una estructura con el avance del estudiante a través del programa académico.
Flujo Alternativo	ninguno

Ilustración 51 - Diagrama de casos de usos de la aplicación móvil

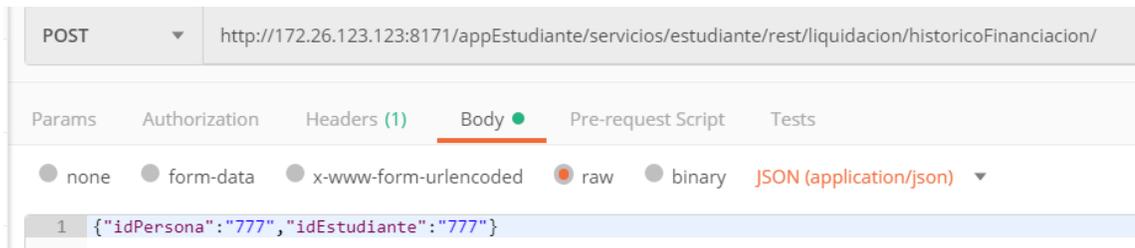


Modelo

El modelo de datos usado pertenece al CIADTI, este es usado en sus aplicaciones y desarrollos, pero por su gran tamaño no puede ser plasmado en el documento. Mediante el uso de este modelo se construyó un API Rest el cual es consumido por la aplicación móvil actual de estudiantes UP, este también se usó en el desarrollo del prototipo. Para probar la información que retorna un recurso del API Rest se usó el software Postman, el cual permite probar servicios web de cualquier tipo.

De la siguiente manera se hicieron peticiones al API Rest de la Universidad.

Ilustración 52 - Petición al API Rest de la Universidad de Pamplona



La información retornada el API Rest se retorna en formato JSON, para que su manejo y procesamiento fuera lo más sencillo posible:

Ilustración 53 - Respuesta de petición a un API Rest, formato JSON

```
{
  idPersona: "555",
  nombre: "ANALIA ",
  apellidos: "FRUZCA TORRADO",
  documento: "TI 458142196",
  emailInstitucional: "prueba2@g.com",
  emailPersonal: "abc@mailtemp.com",
  telefono: "555-12345",
  celular: "350-0203040",
  direccion: "CALLE 1 0-100",
  listadoProgramas: [
    {
      idEstudiante: "888",
      codigoPrograma: "789456",
      nombrePrograma: "ADMINISTRACION DE EMPRESAS",
      nombreUnidad: "UNIVERSIDAD DE PAMPLONA",
      situacionEstudiante: "ACTIVO",
      categoria: "ANTIGUO",
      metodologia: "PRESENCIAL",
      nivelEducativo: "PREGRADO",
      modalidad: "Universitaria",
      pensum: "ADMINISTRACION DE EMPRESAS 2006",
      pensumInicio: null
    }
  ]
}
```

4.2 DISEÑO

Definición del escenario

La aplicación tiene como fin último funcionar tanto con conexión a Internet como sin conexión a él, pero en este prototipo se mostrar el funcionamiento de la aplicación con conexión a la red.

Diseño del boceto

El diseño de un boceto permite una aproximación de lo que será la interfaz de usuario de un software en general, para este caso, de una aplicación móvil, un boceto permite que los desarrolladores se den una idea a grandes rasgos de cómo sería la interfaz de usuario del sistema.

Para este fin se hará uso de los wireframe los cuales son un medio para modelar el esquema de interacción del sistema con los usuarios, la idea es que se pueda plasmar la mayor cantidad de interacciones del sistema con el usuario. A continuación, se presentan los wireframe del prototipo de la aplicación móvil.

Ilustración 54 - Wireframe de la pantalla de login y cambio de programa

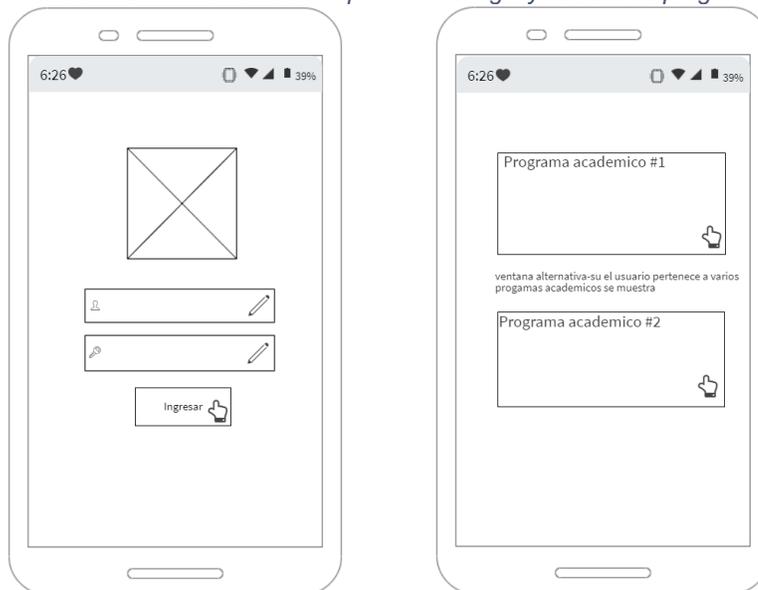


Ilustración 55 - Wireframe de los tabs home y mapa



Ilustración 56 - Wireframe de tab notificaciones

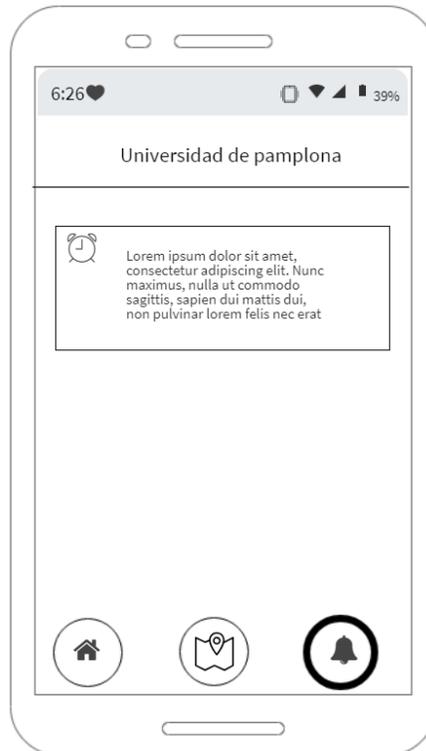


Ilustración 57 - Wireframe de menú desplegable y acceso a pantalla de horario



Ilustración 58 - Wireframe de acceso a pantalla calificaciones



Ilustración 59 - Wireframe acceso a pantalla de registro extendido



Ilustración 60 - Wireframe de acceso a pantalla de resumen financiero



Ilustración 61 - Wireframe de pantalla de semáforo académico

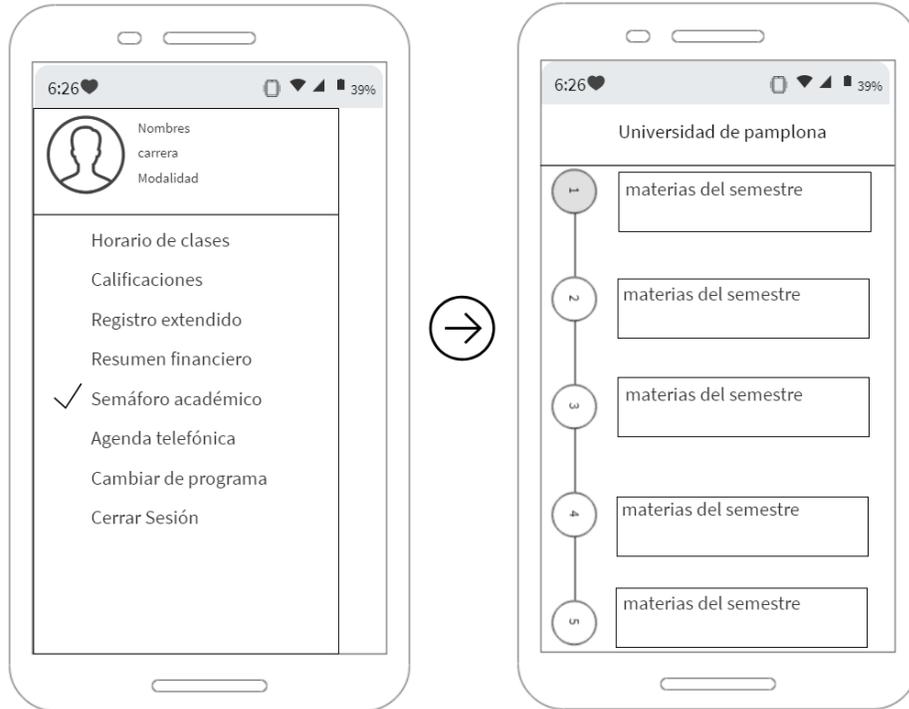
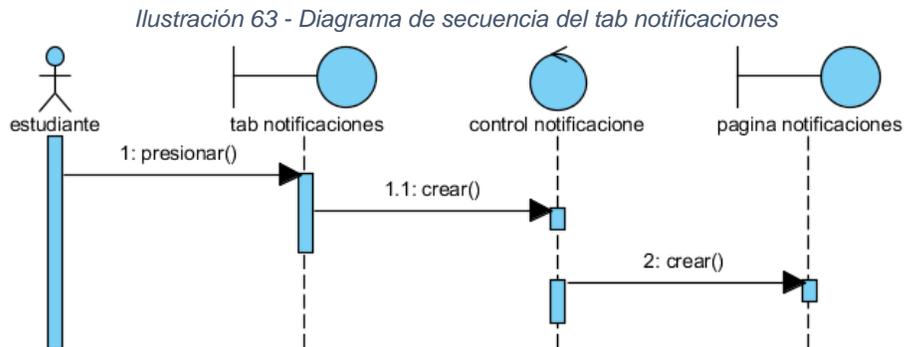


Ilustración 62 - Wireframe de acceso a pantalla de directorio



Estructuración de la aplicación

En este paso se muestra la interacción a través del tiempo entre los objetos del sistema, mostrando como es la comunicación entre los objetos que se muestran en las pantallas de la aplicación, entiéndase por objetos los componentes de la interfaz de usuario.



El diagrama de la ilustración 61 muestra la interacción entre los objetos en el evento presionar el tab notificaciones. Para los tab de mapa y home se presenta la misma interacción entre los objetos solo que cambia el objeto tab notificaciones, por tab home y tab mapa. Ahora se mostrarán los diagramas los cuales presentan una interacción con el objeto menú desplegable

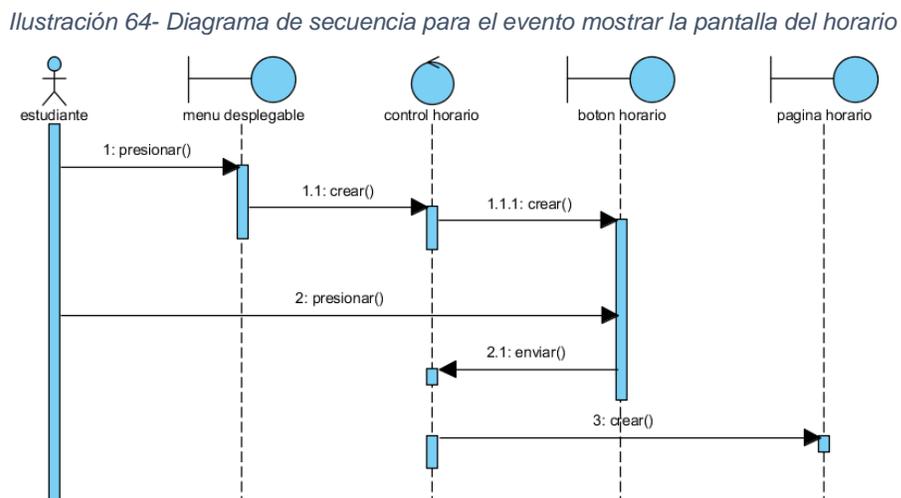
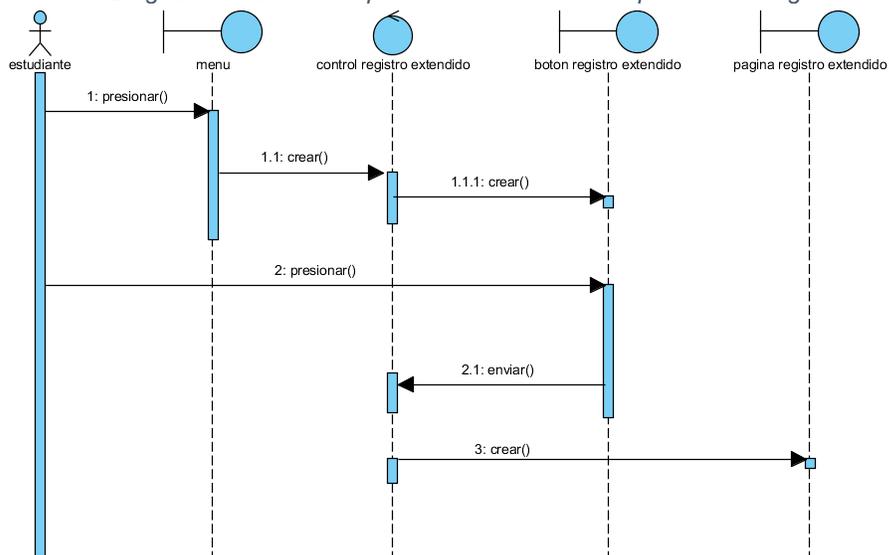


Ilustración 65 - Diagrama de secuencia para el evento mostrar la pantalla del registro extendido



Los demás diagramas de interacción con el objeto menú desplegable presentan la misma secuencia de eventos y acciones, lo único que cambia son los objetos que se muestran en el diagrama.

4.3 DESARROLLO

En esta sección se da inicio a la codificación del prototipo, se sigue el proceso desde la instalación del framework y el desarrollo de algunas pantallas, ya que muchas siguen la misma mecánica de creación, pero con diferente lógica, por lo que solo se mostraran las imágenes del resultado final de la pantalla mas no el procedimiento ni el código que les proporciona la lógica y la interfaz de usuario.

Herramientas de software utilizadas

Tabla 16 - Herramientas de software utilizadas para la construcción del prototipo

#	Nombre	Descripción
1	Visual Studio Code	Editor de código
2	Postman	Herramienta con múltiples características para el manejo de servicio web.
3	Android Studio	SDK de Google para el desarrollo de aplicaciones móviles en Android

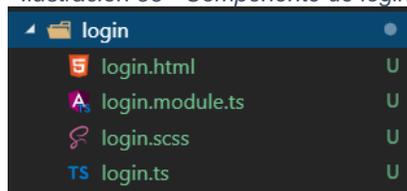
4	Google Chrome	Navegador web con herramientas de desarrollo, permite hacer pruebas con Ionic
5	Ionic Devapp	Aplicación que permite la simulación de la aplicación desarrollada
6	MockFlow	Herramienta en línea para la construcción de wireframe
7	Windows 7	Sistema operativo del computador
8	Node.JS	Entorno de ejecución de JavaScript para la capa de los servidores.

Como primer paso se procedió a instalar Ionic en el computador, pero primero se instaló Node.JS para poder hacer uso de su gestor de paquetes NPM, esto se hace descargando el instalador directamente del sitio oficial <https://nodejs.org/es/>, una vez instalado Node.JS se ejecutó en la consola el comando `npm install -g Ionic cordova`, el cual descargo Ionic y Apache Cordova en sus últimas versiones y las instalo en el sistema operativo.

Para crear un proyecto con Ionic se utilizó el comando `Ionic start estudiantes UP blank`, esto desplego un asistente en la consola, que permitió la creación de un proyecto en blanco. Para probarlo en el navegador en la consola se puede digitar el comando `Ionic serve` e `Ionic serve -c` si se desea probar en la aplicación Ionic DevApp. Algo a tener en cuenta es que el dispositivo móvil debe estar conectado en la misma red en la que se encuentra el computador.

Ionic proporciona un generador de código, para generar una pantalla usa el comando `Ionic g page "nombre de la página"`, esto genera el código necesario dentro de la carpeta `src`. Este comando genera un componente con sus archivos HTML, SCSS y TypeScript.

Ilustración 66 - Componente de login



El diseño de la interfaz de usuario se hace completamente en HTML y SCSS, además Ionic cuenta con componentes UI ya predefinidos, los cuales facilitan la creación de secciones en la página y simulan los efectos y transiciones por defecto de cada sistema operativo.

Ilustración 67 - Pantalla de login



En este momento el usuario puede ingresar sus credenciales para acceder al sistema, si estas son erróneas se lanza una alerta y si son correctas guarda en el storage (almacenamiento del dispositivo), información acerca de credenciales para llamar a los servicios, se re direcciona a la pantalla de seleccionar programa académico. En caso de solo pertenecer a un programa esta pantalla se omite.

Ilustración 68 - Alerta de error en inicio de sesión

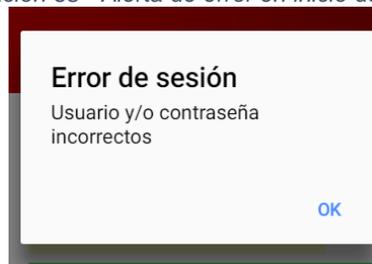


Ilustración 69 - Código formulario de logueo

```
<ion-row>
  <ion-col>
    <form>
      <ion-item class="animated bounceInLeft delay-1s slow">
        <ion-label floating color="dark">Usuario</ion-label>
        <ion-icon color="dark" name="person" item-end</ion-icon>
        <ion-input [(ngModel)]="user" type="text" name="user"></ion-input>
      </ion-item>
      <ion-item class="animated bounceInRight delay-1s slow">
        <ion-icon color="dark" name="eye" item-end</ion-icon>
        <ion-label floating color="dark">Contraseña</ion-label>
        <ion-input [(ngModel)]="pass" type="password" name="pass"></ion-input>
      </ion-item>
      <br>
      <br>
      <button [disabled]="user.length < 1 || pass.length < 1" (click)="login()"
        class="boton_general animated bounceInUp delay-1s slow" ion-button block>Ingresar</button>
    </form>
  </ion-col>
</ion-row>
```

Ilustración 70 - Pantalla seleccionar programa académico



Ahora se muestra la pantalla inicial de la aplicación conocida como home, en la cual se mostró información académica del usuario respecto al programa que cursa, un menú desplegable con las funcionalidades de la aplicación, y tres tabs de navegación, home, mapa y notificaciones.

Ilustración 71 – Pantalla home



Ilustración 72 – Pantalla mapa



Ilustración 73 – Menú desplegable



En la pantalla principal se mostró información académica del proceso del usuario en su programa, también se mostró gráficos a manera de slide con los cuales el usuario puede interactuar y notificaciones del día en curso a manera de recordatorio de las clases. En la pantalla del mapa se muestra el mapa de la sede principal de la Universidad de Pamplona con algunas ubicaciones.

Cada opción del menú representa una funcionalidad a la cual se accede presionándola, cuando esto pasa la funcionalidad se muestra en otra pantalla distinta. En las páginas de las funcionalidades también se incorporó el botón del menú lo que permite la navegabilidad entre ellas.

La funcionalidad de cambiar de programa solo se mostrará en pantalla en caso que el estudiante curse o haya cursado más de un programa académico. En el menú desplegable también se puede observar la información básica del estudiante, como el nombre completo, el programa en el que se encuentra, el correo electrónico y una foto.

Funcionalidades del menú desplegable

Horario: Muestra el horario para cada día de la semana, si no tiene clases ese día muestra un elemento que informa que no hay clase.

Ilustración 74 - Pantalla horario



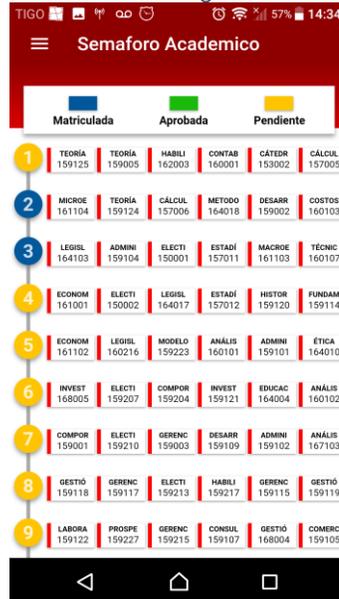
Registro extendido: Muestra el historial de calificaciones a través de los años.

Ilustración 75 - Pantalla registro extendido



Semáforo: Muestra información de las materias que se deben cursar y que pertenecen a un programa académico en específico.

Ilustración 76 - Pagina semáforo



Información Financiera: Muestra opciones con información financiera del usuario, las opciones llevan a otras pantallas donde se muestra a detalle la información.

Ilustración 77 - Pantalla información financiera



Ilustración 78 – Pantalla detalle otras liquidaciones

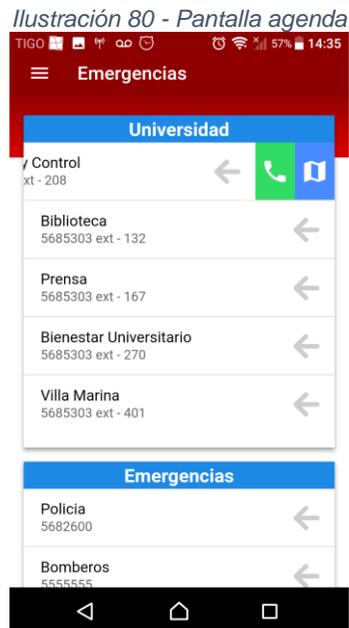


Calificaciones: Muestra calificaciones del semestre en curso y un detalle para ver que notas conforman la calificación final

Ilustración 79 - Pantalla calificaciones



Agenda: Muestra una pantalla con números telefónicos relacionados con la Universidad o con números de emergencia.



Cambiar programa: Muestra la pantalla de cambiar programa, solo se muestra si pertenece a más de dos programas.

Cerrar sesión: Elimina información de credenciales guardadas en el almacenamiento nativo y re direcciona a la pantalla de Login.

La información representada en las pantallas se obtiene mediante el consumo de un API Rest desarrollado por el CIADTI con anterioridad para el consumo de la aplicación móvil actual, el consumo de este API se hace a través de peticiones HTTP mediante la clase HttpClient usada en los providers de Ionic.

Esta clase permite hacer llamados a API Rest obteniendo respuestas de estos en un formato JSON, el cual es mucho más fácil de manejar para JavaScript, ya que cuenta con herramientas de conversión sencilla en su núcleo para este formato.

Ilustración 81 - Estructura de un provider en Ionic

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

/*
  Generated class for the MapaProvider provider.

  See https://angular.io/guide/dependency-injection for more info on providers
  and Angular DI.
*/
SandroCardenas, 2 months ago | 1 author (SandroCardenas)
@Injectable()
export class MapaProvider {

  constructor(public http: HttpClient) {
    console.log('Hello MapaProvider Provider');
  }

}
```

Ilustración 82 - Petición http básica a un API Rest

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';

SandroCardenas, 2 months ago | 1 author (SandroCardenas)
@Injectable()
export class ProductosProvider {
  url_base: string;

  constructor(public http: HttpClient) {
    console.log("Iniciado el servicio de productos");
    this.url_base = `https://restcountries.eu/rest/v2/name/united`;
  }

  public getProductosTipo() {
    return this.http.get(this.url_base);
  }
}
```

Las fases de pruebas y de entrega de la metodología usada, no se tomaron en cuenta para el desarrollo de la aplicación prototipo, si se realizaron pruebas durante el proceso, pero fueron pruebas sencillas del funcionamiento de la aplicación, más nada complejo.

5 CONCLUSIONES

Según lo expuesto a lo largo de este trabajo, se permite concluir lo siguiente:

- La elaboración de un estudio comparativo de framework móviles híbridos, permite dar una base teórica acerca de las distintas características y funcionalidades con las que cuentan este tipo de tecnologías, y como se adecuan a los desarrollos de acuerdo a las necesidades.
- Todos los framework de desarrollo móvil híbrido cumplen con el objetivo de desarrollar aplicaciones móviles, ofreciendo, funcionalidad, mantenibilidad, usabilidad, entre otras características que hacen parte de una tecnología, pero el más apropiado para su uso, según los requerimientos del CIADTI es el framework Ionic.
- El prototipo resultante se desarrolló en base a la aplicación actual de estudiantes UP imitando todas sus funcionalidades y adicionando otras, como recordatorios y agenda, en un periodo de tiempo corto, aportando fuerza a la selección de un framework, para el desarrollo a aplicaciones móviles en el CIADTI.
- Las aplicaciones móviles híbridas se desarrollan en un periodo de tiempo corto, ahorran costos y se destacan por su portabilidad, dando soporte a múltiples sistemas operativos en base al mismo desarrollo, por lo que son una opción viable para el desarrollo de aplicaciones móviles, teniendo en cuenta siempre el objetivo de la aplicación.
- No hay un framework de desarrollo móvil híbrido que represente la mejor alternativa para la creación de todo tipo de aplicación, y ante la gran cantidad de opciones para el desarrollo de este tipo de aplicaciones, la selección de una tecnología, depende del alcance, contexto y la finalidad de la aplicación.

6 TRABAJO FUTURO

- Desde que inició proyecto siempre estuvo pensado para que no quedara en el papel. En lugar de eso, se espera que el prototipo aquí presentado, pueda seguir su proceso desarrollo, llegando a una versión de la aplicación que pueda ser llevada a una fase de entrega y puesta en producción.
- Integración de la aplicación con servicios provistos en otras aplicaciones desarrolladas en el CIADTI (como la aplicación de docentes), a fin de contar con una sola, que supla las necesidades informativas de toda la comunidad universitaria, en lugar de varias aplicaciones, una, para cada necesidad.
- Uso de la aplicación sin conexión a Internet, con el fin de que los usuarios puedan navegar en esta y poder acceder al último contenido que se mostró cuando aún tenía conexión.
- Desarrollo de una sección de noticias, eventos e información compartida, similar a la página de inicio, del sitio web de la Universidad de Pamplona, con el fin de mantener informada a la comunidad universitaria.
- Desarrollo de la funcionalidad de notificaciones personales, para que los usuarios recuerden fechas de actividades o compromisos que tengan con la Universidad.

7 FUENTES BIBLIOGRÁFICAS

AGUIRRE ESPINOZA BELEN, Cindy. Comparativa de frameworks para el desarrollo de aplicaciones con PHP. Trabajo de grado, Cuenca, Ecuador. Universidad de Azuay. 7 p.

ARGOTE, Iván., JIMÉNEZ, Andrés., CASTRO, Danny., CEBALLOS, David. Desarrollo de apps: un estudio comparativo entre frameworks libres y privativos. Universidad mariana boletín informativo CEI [en línea]. 10 de junio de 2016 [citado el 18 de julio de 2018]. Disponible en Internet: <http://ojseditorialumariana.com/index.php/BoletinInformativoCEI/article/view/1083>.

BBVAOPEN4U, API REST: Que es y cuáles son sus ventajas a la hora de desarrollo de proyectos [en línea], 23 de marzo de 2016 [revisado el 15 de Noviembre de 2018], Disponible en Internet: <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>.

CANO INSA, Esteban. BiciPark: web app con Ionic y acceso a servicios REST. Trabajo para fin de Master. España.: Universidad de Alicante (2015). 12 p.

CARPINTERO, Irene. Ionic 3 un entorno para empezar a desarrollar apps moviles. Todoandroid [en línea]. 03 de septiembre de 2017 [revisado el 22 de agosto de 2018]. Disponible en Internet: www.todoandroid.es/ionic-3-desarrollar-apps-moviles.

CILLERO, Manuel. Pruebas en ingeniería de software. manuel.cillero.es mi cinscuntancia similar [en línea]. 12 de Julio de 2017 [revisado el 12 de agosto 2018]. Disponible en Internet: <https://manuel.cillero.es/doc/metrica-3/tecnicas/pruebas/sistema/>

CORTES ROJAS, Dario Contreras. Desarrollo de una aplicación prototipo para la localización de parqueaderos en la plataforma IOS. Trabajo de grado, Bogotá, Colombia. Universidad EAN. 2013. 13 p.

B. G, Alexander, ¿Qué es npm?, devcode.la [En Línea]. 20 de Abril de 2017 [revisado el 06 de Diciembre de 2018]. Disponible en Internet: <https://devcode.la/blog/que-es-npm/>

Diccionario de la Real Academia Española (RAE). Definición de software. Real Academia Española [en línea]. 16 de julio de 2018 [revisado el 16 de julio de 2018]. Disponible en Internet: <http://dle.rae.es/?id=YErIG2H>.

DIEZ, Braulio. ¿Por qué REDUX?, Lemon Code- Master en Front End [En Línea], 18 de Junio de 2017 [revisado el 15 Octubre 2018], Disponible en Internet: <http://lemoncode.net/lemoncode-blog/2017/6/18/por-que-redux>.

ESCAÑO TURIZO, Hugo Alejandro. Estudio comparativo de sistemas integrados de gestión bibliotecaria de software libre más populares en Colombia. Trabajo de Grado, Pamplona, Colombia. Universidad de Pamplona. 2016. 8 p.

ESCOBAR, Salgado. Estudio de la eficiencia de los frameworks híbridos y nativos en el desarrollo de aplicaciones móviles, basado en Benchmark para el Consorcio Informega. Trabajo de grado: Universidad de las fuerzas armadas ESPE. 2018. 5 p.

G. M, Maira Cecilia, C. A, Luis Leonardo y D, Byron Medina. Metodología para el desarrollo de aplicaciones móviles. Trabajo de Grado, Colombia: Universidad de Magdalena. (2013). 5 p.

GNU. ¿Qué es el software libre? El sistema operativo GNU [en línea]. 20 de julio de 2018 [revisado el 20 de julio de 2018]. Disponible en Internet: <https://www.gnu.org/philosophy/free-sw.es.html>.

GUERRERO ALARCON, Carlos Andres. Estudio comparativo de marcos de trabajo para el desarrollo de aplicaciones web de código abierto. Revista Gerencia Tecnológica Informática [en línea]. 10 de junio de 2009 [citado el 18 de julio de 2018]. Disponible en Internet: <https://revistas.uis.edu.co/index.php/revistagti/article/view/808>.

<https://angular.io>, descripción de la arquitectura, Angular Documentation [En línea], [revisado el 1 de Noviembre de 2018]. Disponible en Internet: <https://angular.io/guide/architecture>.

MERCADO, Alejandro. Desarrollo de aplicaciones móviles con NativeScript. Software Guru [en línea]. [Revisado el 2 de noviembre de 2018]. Disponible en Internet: <https://sg.com.mx/revista/53/desarrollo-aplicaciones-m-viles-NativeScript>.

NARVAEZ, Bolaño. Comparativo de Frameworks web MVC open source en java, para la migración del administrador vortal en el CIADTI. Trabajo de grado, Pamplona Colombia. Universidad de Pamplona. (2017). 8 p.

ThespianArtist. Aplicación híbrida o nativa ¿Cuál es la mejor? Platzi [en línea]. 06 de junio de 2017 [revisado 15 de agosto de 2018]. Disponible en Internet: <https://platzi.com/blog/hibrida-o-nativa/>.

V, Rahimian, y R, Ramsin. Designing an agile methodology for mobile software development: A hybrid method engineering approach. IEEE explore digital library [en línea]. 03 de junio de 2008 [revisado el 09 de agosto de 2018]. Disponible en Internet: <https://ieeexplore.ieee.org/document/4632123/>.

Wiki.fdi.ucm.es. licencia Apache. Wiki de la asignatura de ética legislación y profesión (ELP) [en línea]. 03 de Julio de 2017 [revisado el 23 de agosto de 2018]. Disponible en Internet: http://wikis.fdi.ucm.es/ELP/Licencia_Apache.

Wikipedia. Software. Wikipedia [en línea]. 20 marzo de 2018 [revisado el 16 de julio de 2018]. Disponible en Internet: https://es.wikipedia.org/wiki/Software#Car%C3%A1cter_evolutivo_del_software.

www.codigofacilito.com. Progressive web apps. Codigofacilito [en línea]. 22 de agosto de 2018 [revisado el 22 de agosto de 2018]. Disponible en Internet: <https://codigofacilito.com/articulos/progressive-apps>.

www.diplomadogestioncalidadsoftware2015.com. ISO/IEC 9126. ISO/IEC 9126 [en línea]. [revisado el 2 octubre de 2018]. Disponible en Internet:

<https://diplomadogestioncalidadsoftware2015.wordpress.com/norma-iso-9126/calidad-interna-y-externa/>.

www.Ecured.com. PhoneGap. Ecured [en línea]. 23 de septiembre de 2018 [revisado el 23 de agosto de 2018]. Disponible en Internet: <https://www.ecured.cu/PhoneGap>.

www.framework7.io. Documentación oficial. Framework7 [en línea]. 23 de agosto de 2018 [revisado el 23 de agosto de 2018]. Disponible en Internet: <https://framework7.io/>.

www.Ionicframework.com. Acerca de, Documentación oficial. Ionic Framework [en línea]. 21 de agosto de 2018 [revisado el 21 agosto de 2018]. Disponible en Internet: <https://ionicframework.com/docs/>.

www.Ionicframework.com. Documentación oficial. Ionic Framework [en línea]. 21 de agosto de 2018 [revisado el 21 agosto de 2018]. Disponible en Internet: <https://ionicframework.com/docs/v1/guide/preface.html>.

www.Ionicframework.com. Documentación oficial. Ionic Framework [en línea]. 21 de agosto de 2018 [revisado el 21 agosto de 2018]. Disponible en Internet: <https://ionicframework.com/about>.

www.PhoneGap.com. Acerca de, documentación oficial. Adobe PhoneGap [en línea]. 23 de agosto de 2018 [revisado el 23 de agosto de 2018]. Disponible en Internet: <https://PhoneGap.com/about/>

www.postparaprogramadores.com. Los tipos de frameworks y sus funcionalidades. Para programadores [en línea]. 21 de julio de 2018 [revisado el 2018]. Disponible en Internet: <https://postparaprogramadores.com/tipos-de-framework/#Tipos de Framework>.

www.programacion.net. Comparación de frameworks para desarrollar apps para móvil híbridas. Programación.net [en línea]. 23 de enero de 2016 [revisado el 15 de

julio de 2018]. Disponible en Internet:
http://programacion.net/articulo/comparacion_de_frameworks.

www.rigelnetworks.com. Going Native from Hybrid, with NativeScript. Rigel Networks le tus grow together. [en línea]. 28 de abril de 2017 [revisado el 24 de agosto de 2018]. Disponible en Internet: <https://www.rigelnetworks.com/6621-2/>

www.wikipedia.org. Framework. Wikipedia [en línea]. 01 de mayo de 2017 [revisado el 21 de julio de 2018]. Disponible en Internet:
<https://es.wikipedia.org/wiki/Framework>.

www.wikipedia.org. NativeScript. Wikipedia [en línea]. 19 de marzo de 2018 [revisado el 24 de agosto de 2018]. Disponible en Internet:
<https://es.wikipedia.org/wiki/NativeScript>.

www.wikipedia.org. React Native. Wikipedia [en línea]. 01 de agosto de 2018 [revisado el 23 de agosto de 2018]. Disponible en Internet:
[https://en.wikipedia.org/wiki/React_\(JavaScript_library\)#React_Native](https://en.wikipedia.org/wiki/React_(JavaScript_library)#React_Native)

Xavi. Las cinco etapas de la ingeniería de software. Proyectosguerrilla [en línea]. 11 de noviembre de 2014 [revisado el 10 de agosto de 2018]. Disponible en Internet:
<http://proyectosguerrilla.com/blog/author/xavi/>