

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Ingeniería en Sistemas Computacionales

**ESTUDIO DEL FRAMEWORK IONIC 2 PARA EL DESARROLLO DE  
APLICACIONES MÓVILES HÍBRIDAS**

Trabajo de grado previo a la obtención del título de Ingeniería en Sistemas  
Computacionales

**Autor:**

Luis Gonzalo Yaguapaz Madera

**Director:**

Ing. Pablo Landeta López

**Ibarra- Ecuador**

**Mayo - 2018**



**UNIVERSIDAD TÉCNICA DEL NORTE**

**BIBLIOTECA UNIVERSITARIA**

**AUTORIZACION DE USO Y PUBLICACIÓN**

**A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

## **1.- IDENTIFICACIÓN DE LA OBRA**

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital Institucional determina de disponer textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad. Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información.

<b>DATOS DEL AUTOR</b>	
<b>CÉDULA DE IDENTIDAD</b>	1003811716
<b>APELLIDOS Y NOMBRES</b>	YAGUAPAZ MADERA LUIS GONZALO
<b>DIRECCIÓN</b>	PIMAMPIRO, BOLIVAR Y ESPEJO
<b>E-MAIL</b>	lgyaguapaz@utn.edu.ec
<b>TELÉFONO MÓVIL</b>	0989186488
<b>DATOS DE LA OBRA</b>	
<b>TÍTULO</b>	“ESTUDIO DEL FRAMEWORK IONIC 2 PARA EL DESARROLLO DE APLICACIONES MÓVILES HIBRIDAS”
<b>AUTOR</b>	YAGUAPAZ MADERA LUIS GONZALO
<b>FECHA</b>	MAYO DEL 2018
<b>PROGRAMA</b>	PREGRADO
<b>TÍTULO POR EL QUE OPTA</b>	INGENIERÍA EN SISTEMAS COMPUTACIONALES
<b>DIRECTOR</b>	ING. PABLO LANDETA LÓPEZ, MSC.

## 2.- AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Luis Gonzalo Yaguapaz Madera, con cédula de identidad N° 100381171-6, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y Autorizo a la Universidad Técnica del Norte, la publicación de esta obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

## 3.- CONSTANCIAS

El autor manifiesta que la obra de la presente autorización es original y se desarrolló, sin violar derechos de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.



Firma

Nombre: Luis Gonzalo Yaguapaz Madera

Cédula: 1003811716

Ibarra, abril del 2018



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

**CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

Yo, Luis Gonzalo Yaguapaz Madera, con cédula de identidad N° 100381171-6 manifiesto mi voluntad de ceder a la Dirección Provincial de Educación de Imbabura los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor del trabajo de grado denominado "**ESTUDIO DEL FRAMEWORK IONIC 2 PARA EL DESARROLLO DE APLICACIONES MÓVILES HÍBRIDAS**", que ha sido desarrollado para obtener el título de Ingeniería en Sistemas Computacionales, en la Universidad Técnica del Norte, quedando en la Dirección de Educación para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital.

Firma

Nombre: Luis Gonzalo Yaguapaz Madera

Cédula: 1003811716

Ibarra, abril del 2018



UNIVERSIDAD TÉCNICA DEL NORTE  
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DIRECTOR DE TESIS

Por medio del presente yo Ing. Pablo Landeta, certifico que el Sr. Luis Gonzalo Yaguapaz Madera, portador de la cédula de identidad Nro. 100381171-6. Ha trabajado en el desarrollo del proyecto de tesis "ESTUDIO DEL FRAMEWORK IONIC 2 PARA EL DESARROLLO DE APLICACIONES MÓVILES HIBRIDAS", previo a la obtención del título de Ingeniero en Sistemas Computacionales, realizándolo con interés profesional y responsabilidad, lo cual certifico en honor a la verdad.

Ing. Pablo Landeta López  
DIRECTOR DE TRABAJO DE GRADO



Ing. Jean Arciniegas  
**DIRECTOR PLANIFICACIÓN,  
DESARROLLO Y COOPERACIÓN**

Tiene a bien,

**CERTIFICAR:**

Que el señor Luis Gonzalo Yaguapaz Madera, con cédula 1003811716, realizó la implementación y funcionamiento del aplicativo móvil “Destino Pimampiro”, misma que fue desarrollada conjuntamente con el GAD Municipal de Pimampiro.

Faculta al peticionario hacer uso del presente certificado en la forma que convenga a sus intereses, excepto para trámites judiciales.

Pimampiro, 22 de mayo de 2018



Ing. Jean Arciniegas  
**DIRECTOR PLANIFICACIÓN, DESARROLLO  
Y COOPERACIÓN GAD MUNICIPAL DE PIMAMPIRO**

## DEDICATORIA

Dedico este proyecto fin de carrera a todos mis familiares, amigos y compañeros que en algún momento durante estos años de estudio me acompañaron, guiaron y apoyaron. Y sobre todo a mi madre, que sin su esfuerzo e incansable apoyo nunca podría haber llegado a realizar este uno de los tantos objetivos planteados.

*“Llena tus ojos de ilusión -. Vive como si fueras a morir dentro de diez segundos. Ve al mundo. Es más fantástico que cualquier sueño real o imaginario. No pidas garantías, no pidas seguridad. Nunca ha existido algo así. Y, si existiera, estaría emparentado con el gran perezoso que cuelga boca abajo de un árbol, y todos y cada uno de los días, empleando la vida en dormir. Al diablo con eso-, sacude el árbol y has que el gran perezoso caiga sobre su trasero “*

Ray Bradbury

## **AGRADECIMIENTO**

A mi madre Patricia por todo el esfuerzo, atención y constante preocupación toda mi vida, mis logros son gracias a su ayuda y apoyo incondicional. Ella ha representado siempre el motor que mueve mi vida y me han motivado a avanzar y cumplir los objetivos trazados en mi vida.

A la Universidad Técnica del Norte por haberme acogido durante mi formación académica por formarme como persona, profesional y encaminarme para poder servir a la sociedad.

A mi familia, en especial a mi abuelo Gonzalo por sus palabras de aliento y apoyo, siempre me ha inspirado con su ejemplo de honradez, responsabilidad y perseverancia. Por sus palabras de aliento y apoyo.

**Luis Gonzalo Yaguapaz Madera**

## Tabla de Contenido

CAPITULO I .....	1
1. INTRODUCCIÓN.....	1
1.1. Problema .....	1
1.1.1. Antecedentes.....	1
1.1.2. Situación Actual.....	2
1.1.3. Prospectiva.....	3
1.1.4. Planteamiento del Problema. ....	3
1.2. Objetivos.....	4
1.2.1. Objetivo General. ....	4
1.2.2. Objetivos Específicos. ....	4
1.3. Alcance y Limitaciones.....	5
1.3.1. Alcance. ....	5
1.4. Justificación.....	6
1.4.1. De la investigación.....	6
1.4.2. De la aplicación. ....	6
CAPITULO II.....	7
2. MARCO TEÓRICO .....	7
2.1. Situación actual y proceso de difusión de turismo en el Gad Pimampiro.....	7
2.2. Introducción a herramientas de Desarrollo de Aplicaciones Móviles Híbridas	8
2.3. Metodología Scrum.....	10
2.3.1. Principios de Scrum.....	11
2.3.2. Los valores de Scrum.....	11
2.3.3. El equipo scrum (Team Scrum).....	12
2.3.4. El Dueño de Producto (Product Owner).....	12
2.3.5. El Equipo de Desarrollo (Development Team).....	12
2.3.6. El director del Proyecto (Scrum Master). ....	13
2.3.7. El Sprint. ....	13
2.4. Herramientas utilizadas para el desarrollo .....	13
2.4.1. Yii Framework 2.....	13
2.4.2. Base de datos MySQL. ....	19
2.4.3. JSON.....	20
2.4.4. RESTful web services. ....	21

CAPITULO III.....	23
3. FRAMEWORK IONIC 2 .....	23
3.1. Introducción al Framework .....	23
3.2. Instalación y Configuración .....	24
3.2.1. Pre-requisitos.....	24
3.2.2. Instalación.....	25
3.2.3. Configuración.....	26
3.2.4. Creación del proyecto. ....	26
3.2.5. Estructura del proyecto. ....	27
3.2.6. Proceso de compilación.....	29
3.3. Características del Framework .....	30
3.3.1. TypeScript.....	30
3.3.2. Angular 2.....	37
3.4. Componentes del Framework.....	38
3.4.1. Action Sheet (Acciones).....	38
3.4.2. Alerts (Alertas).....	38
3.4.3. Badges (Insignias).....	38
3.4.4. Buttons (Botones).....	39
3.4.5. Cards (Tarjetas).....	40
3.4.6. Checkbox (Botones de verificación).....	41
3.4.7. FAB's (Botones de acción flotante).....	41
3.4.8. Grid (tabla).....	42
3.5. API.....	42
3.5.1. API's de componentes.....	42
3.5.2. API's de servicios.....	45
3.6. Componentes Nativos .....	46
3.6.1. Device.....	47
3.6.2. Google Maps.....	48
CAPITULO IV: .....	49
4. DESARROLLO / FASE ELABORACIÓN .....	49
4.1. Planificación.....	49
4.1.1. Investigación preliminar.....	49
4.1.2. Prerrequisitos de sistema.....	49

4.1.3.	Prerrequisitos de hardware. ....	50
4.1.4.	Prerrequisitos de software. ....	50
4.2.	Especificación de requisitos de software (Product backlog).....	55
4.2.1.	Propósito.....	55
4.2.2.	Alcance. ....	55
4.2.3.	Personal Involucrado.....	55
4.2.4.	Requerimientos Funcionales.....	56
4.2.5.	Requerimientos no funcionales. ....	59
4.2.6.	Funcionalidad del Producto.....	62
4.2.7.	Características de los usuarios.....	63
4.2.8.	Planificación del proyecto (Sprint Planning). ....	64
4.3.	Proceso de desarrollo .....	69
4.3.1.	Arquitectura funcional.....	69
4.3.2.	Diseño del aplicativo móvil. ....	71
4.3.3.	Diseño del administrador de contenido. ....	71
4.3.4.	Diseño de la página web del aplicativo móvil. ....	72
4.3.5.	Módulos del administrador de contenido yii2. ....	72
4.4.	Desarrollo del administrador de contenido .....	74
4.4.1.	Configuración del entorno de desarrollo yii2. ....	74
4.4.2.	Base de datos.....	75
4.4.3.	Codificación del Administrador de Contenido y API.....	78
4.4.4.	Codificación de la Aplicación Móvil. ....	85
4.5.	Pruebas .....	92
CAPITULO V.....		94
5.	CONCLUSIONES Y RECOMENDACIONES .....	94
5.1.	Conclusiones .....	94
5.2.	Recomendaciones .....	95
ANEXOS .....		96
	Aplicación en Google Play .....	96
	Certificado - IARC Content Ratings.....	97
Bibliografía .....		98

## Índice de Figuras

Fig. 1: Árbol de Problemas .....	4
Fig. 2: Arquitectura de Ionic 2 orientado a servicios .....	6
Fig. 3: Logo Destino Pimampiro .....	7
Fig. 4: Logo Apache Cordova .....	9
Fig. 5: Funcionamiento de una Aplicación Híbrida .....	10
Fig. 6: Ciclo de trabajo de Scrum .....	11
Fig. 7: Principales características de Yii2 .....	14
Fig. 8: Migración RBAC Yii2 .....	16
Fig. 9: Uso de MVC en Yii2 .....	16
Fig. 10: Acciones y vistas en Yii2 .....	17
Fig. 11: Notación de un Controlador en Yli2.....	18
Fig. 12: Logotipo Bootstrap .....	18
Fig. 13: Logo MySQL.....	19
Fig. 14: Logo JSON .....	20
Fig. 15: Notación de un archivo JSON .....	21
Fig. 16: Ilustración Request y Response.....	22
Fig. 17: Plugin SQLite.....	24
Fig. 18: Creando un proyecto en Ionic 2 .....	26
Fig. 19: Estructura de archivos de un proyecto.....	28
Fig. 20: Vista previa del aplicativo en Ionic 2 .....	29
Fig. 21: Conversión de TypeScript a JavaScript.....	31
Fig. 22: Tipos de datos primitivos .....	31
Fig. 23: Estructura y representación de Tuplas .....	32
Fig. 24: Funciones con parámetros opcionales .....	33
Fig. 25: Funciones con parámetros de descanso .....	34
Fig. 26: Funciones de Flecha (Arrow Function) .....	34
Fig. 27: Promise o Promesas .....	35
Fig. 28: Interfaces.....	36
Fig. 29: TypeScript uso de clases .....	37
Fig. 30: Ejemplo de uso de Badges .....	38
Fig. 31 Botones en Ionic 2 .....	40
Fig. 32: Ejemplo de Cards en Ionic 2.....	41

Fig. 33: Ejemplo de FABs en Ionic 2.....	42
Fig. 34: Uso de ion-content y ion-header .....	43
Fig. 35: Segmento.....	44
Fig. 36: Inputs .....	45
Fig. 37: Tabs .....	46
Fig. 38: Plugin Device.....	48
Fig. 39: Instalación Composer .....	51
Fig. 40: Instalación Yii2 .....	51
Fig. 41: Instalación XAMPP.....	52
Fig. 42: Panel de control XAMPP .....	53
Fig. 43: Instalación de Node JS.....	53
Fig. 44: Configuración de variables de entorno Android SDK .....	54
Fig. 45: Arquitectura Aplicativo móvil.....	70
Fig. 46: Arquitectura administrador de contenido .....	70
Fig. 47: Diseño inicial del aplicativo móvil.....	71
Fig. 48: Diseño del administrador de contenido Yii2.....	72
Fig. 49: Diseño de la página web del aplicativo móvil.....	72
Fig. 50: Diagrama de secuencia módulo Administración de contenido .....	73
Fig. 51: Diagrama de secuencia módulo de administración de usuarios.....	73
Fig. 52: Diagrama de secuencia módulo de Respaldo y Recuperación .....	74
Fig. 53: Diagrama Entidad Relación .....	76
Fig. 54: Configuración del archivo hosts .....	78
Fig. 55: Configuración del archivo httpd-vhosts.....	79
Fig. 56: Configuración de la base de datos .....	79
Fig. 57: Creación de módulos en GII .....	80
Fig. 58: Modelo generado por CRUD .....	81
Fig. 59: Controlador generado por CRUD.....	82
Fig. 60: Vistas generadas por CRUD Generator .....	83
Fig. 61: Elaboración de CRUD en Yii2 .....	83
Fig. 62: Archivo de Configuración de API .....	84
Fig. 63: Consumo del ENDPOINT (API).....	87
Fig. 64: Resolver una petición HTTP .....	87
Fig. 65: Página de inicio del aplicativo.....	88

Fig. 66: Patrimonio Turístico.....	89
Figura 67: Detalle de lugar turístico .....	89
Fig. 68: Servicios.....	90
Fig. 69: Lista de eventos .....	91
Fig. 70: Pantalla de búsqueda .....	91

### Índice de tablas

Tabla 1: Sistemas Operativos y lenguajes de desarrollo para aplicaciones móviles híbridas.....	8
Tabla 2: Equivalencias de peticiones HTTP .....	22
Tabla 3: Requisitos de Sistema Operativo .....	25
Tabla 4: Descripción del contenido de un aplicativo Ionic 2 .....	27
Tabla 5: Descripción del Técnico de Turismo .....	55
Tabla 6: Descripción del Director Administrativo .....	55
Tabla 7: Descripción del Director de Sistemas.....	56
Tabla 8: Descripción del desarrollador .....	56
Tabla 9: Requisito funcional de perfil de acceso .....	57
Tabla 10: Requisitos funcionales de acceso de usuarios.....	58
Tabla 11: Requisitos funcionales sobre el ingreso de información.....	58
Tabla 12: Requisitos de Hardware y Software .....	59
Tabla 13: Acceso al sistema web de administración de contenido .....	59
Tabla 14: Acceso la aplicación móvil.....	60
Tabla 15: Requisitos de seguridad para el ingreso al sistema web de administración de contenido.....	60
Tabla 16: Seguridades de base de datos.....	61
Tabla 17: Especificación de los requisitos de mantenibilidad.....	61
Tabla 18: Especificaciones de portabilidad .....	62
Tabla 19: Descripción de la Base de Datos .....	75
Tabla 20: Diccionario de datos.....	77
Tabla 21: Definición de comandos usados por Ionic 2.....	86
Tabla 22: Pruebas en dispositivos .....	92

## RESUMEN

Ionic 2 es un potente framework de desarrollo de aplicaciones móviles híbridas, este nos permite desarrollar aplicaciones en corto tiempo ya que es posible realizar la compilación de estas en diferentes plataformas (Android, iOS, Windows Phone) haciendo uso del mismo código fuente.

Destino Pimampiro es una aplicación móvil que brinda al turista local, nacional, extranjero y público en general información sobre el turismo del Cantón Pimampiro en la que encontramos lugares turísticos, servicios, mapa, eventos y un apartado de interés.

En el capítulo 1: se presenta los objetivos, justificación y el alcance de la investigación como de la aplicación móvil.

En el capítulo 2: Se presenta una recopilación de información sobre la metodología de desarrollo, herramientas que fueron usadas mismas como IDE de desarrollo, herramientas de desarrollo para el aplicativo móvil y para el administrador de contenido.

En el capítulo 3: Se presenta una guía práctica acerca del Framework Ionic 2.

En el capítulo 4: Se presenta el proceso de desarrollo del aplicativo con las herramientas usadas, así como los resultados obtenidos dentro del desarrollo del software.

## SUMMARY

Ionic 2 is a powerful framework for the development of hybrid mobile applications. This allows us to develop applications in a short time since it is possible to compile these in different platforms (Android, iOS, Windows Phone)

Destino Pimampiro is a mobile application that provides to the tourist local, national, foreign and public in general information about tourism in the Pimampiro Canton, where we find tourist places, services, maps, events and a section of interest.

In chapter 1: the objectives, justification and scope of the research are presented as well as the mobile application.

In chapter 2: A compilation of information on the development methodology is presented, tools that were used as IDE of development, development tools for the mobile application and for the content administrator.

In chapter 3: A practical guide is presented about the Ionic 2 Framework.

In chapter 4: The development process of the application is presented with the tools used, as well as the results obtained within the software development.



# CAPITULO I

## 1. INTRODUCCIÓN

### 1.1. Problema

#### 1.1.1. Antecedentes.

Desde algunos años atrás se vienen desarrollando aplicaciones móviles con distintos y nuevos frameworks con diferentes lenguajes de programación, con el objetivo de acelerar el proceso de desarrollo y dejar de recurrir en costes innecesarios, fomentando la reutilización de código ya existente y promoviendo buenas prácticas de desarrollo.

Uno de los frameworks de desarrollo más usados en la actualidad es Ionic que es un marco de desarrollo de aplicaciones móviles HTML5 orientado a la creación de aplicaciones móviles híbridas. Creado por Drifty Co<sup>1</sup>. en 2013 y luego de su gran acogida esta prestigiosa plataforma ha tomado la retroalimentación de los clientes y/o usuario que intentaron o intentan construir aplicaciones móviles con esta herramienta.

Una de las razones de la gran acogida de Ionic entre los desarrolladores de aplicaciones móviles es que en su estructura interna implementa *Angular* que es un potente framework desarrollado por Google para la construcción de aplicaciones web. Otra de las razones de la gran acogida de Ionic es que también implementa *Cordova* antes conocido como *Apache Cordova* como enlace entre Ionic 2 y las API's<sup>2</sup> que proveen las plataformas de Android, iOS y Windows Phone.

---

<sup>1</sup>*Drifty Co*: Empresa de software radicada en Madison, Wisconsin en Estados Unidos.

<sup>2</sup>*API*: vienen del inglés Application Programming Interface (Interfaz de Programación de Aplicaciones).

Basado en estas plataformas el equipo de desarrollo decidió construir su propio marco que se centre en el rendimiento y en ser construido con los estándares web modernos, y después de lanzar una primera versión en mayo de 2015 y de la mano de la segunda versión de Angular denominada *Angular 2*, se lanza en octubre de 2016 la segunda versión del framework denominada Ionic 2.

Uno de los sectores en los que las aplicaciones móviles tienen gran acogida y se prestan mucho para ser desarrolladas es el sector turístico, luego de un profundo análisis se llega a la conclusión de que el GAD Pimampiro no dispone de un medio de difusión de información sobre sitios turísticos que este cantón dispone, es por esto el interés en la creación de una aplicación móvil para mejorar este proceso y así hacer del cantón un referente nacional en el ámbito turístico.

### **1.1.2. Situación Actual.**

#### **1.1.2.1. Investigación.**

Actualmente existe mucho desconocimiento e información escasa acerca del framework Ionic2 en su versión 2.0 por lo que es necesario analizar todas y cada una de sus funcionalidades, con el fin explorar nuevas tecnologías orientadas al desarrollo de aplicaciones móviles híbridas.

#### **1.1.2.2. Aplicación.**

En la actualidad, existe información deficiente y no acertada acerca de los atractivos turísticos y culturales del Cantón Pimampiro lo cual genera una escasa afluencia de turistas, debido a esto es necesario la implementación de medios que contribuyan con el proceso de difusión de información.

En el GAD Pimampiro, el proceso de difusión de atractivos turísticos no ha sido sistematizada es por eso que se requiere un sistema de difusión de información.

### **1.1.3. Prospectiva.**

Mediante esta investigación se logrará obtener una documentación consolidada, clara y estructurada que servirá de ayuda para las personas en general que deseen instruirse y/o necesiten una base sólida al momento de desarrollar aplicaciones móviles con el framework Ionic 2.

Con el desarrollo de la aplicación móvil planteada se automatizará el proceso de difusión de atractivos turísticos y culturales del Cantón Pimampiro, y observaremos una aplicación portable, con una interfaz amigable, rápida, estructurada y fácil de utilizar.

### **1.1.4. Planteamiento del Problema.**

La deficiente información para los desarrolladores tanto locales como nacionales incluso de los países de habla hispana que empiezan a involucrarse con el framework es uno de los obstáculos más difíciles de superar, es entonces que se ha propuesto como objetivo realizar un estudio profundo, elaborar una guía completa para los programadores que se inician con esta herramienta, finalmente teniendo en cuenta la deficiente información que existe en el sector turístico del GAD Pimampiro.

Se puede decir que la institución no cuenta con una aplicación móvil de difusión de información lo que genera pérdidas considerables en materia de turismo, negocios en riesgo de quebrar y temor al momento de impulsar un negocio propio. Se desarrollará una aplicación usando Ionic 2 para dar solución a dicho problema el cual se da a conocer en la *Fig. 1: Árbol de Problemas*.

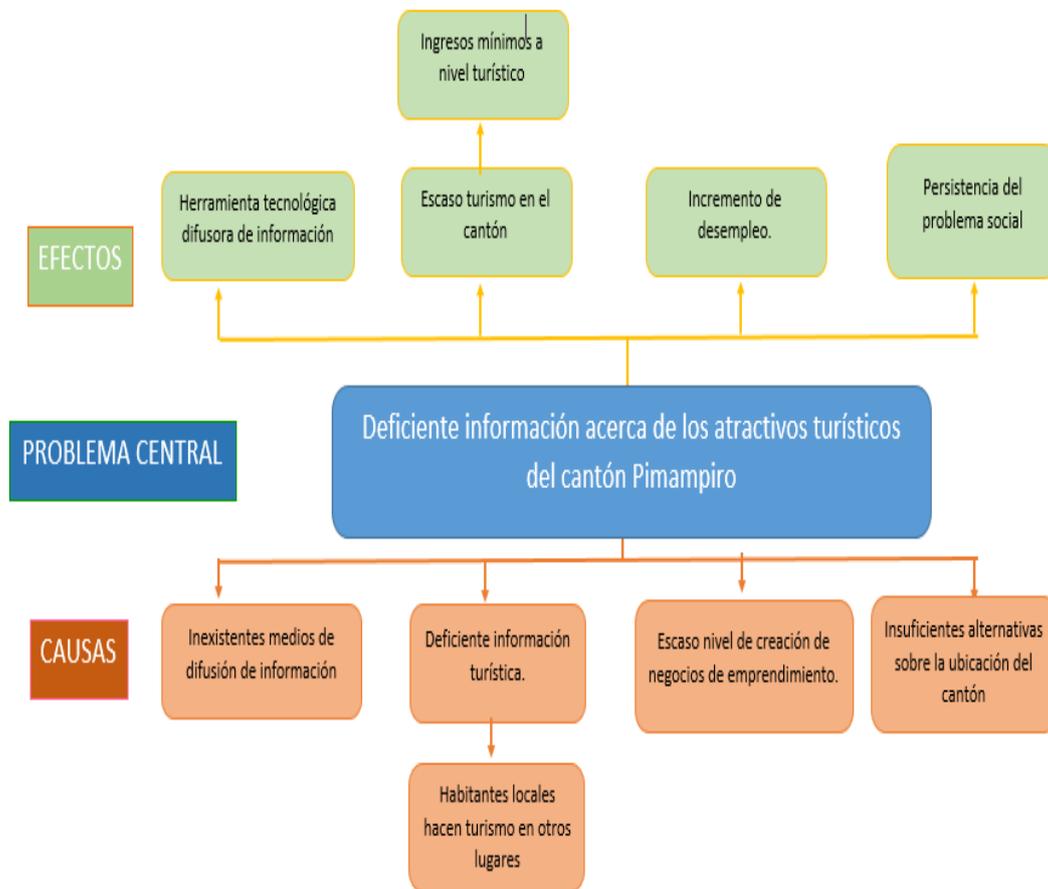


Fig. 1: Árbol de Problemas  
Fuente: Propia

## 1.2. Objetivos

### 1.2.1. Objetivo General.

Realizar un estudio del framework Ionic 2 para el desarrollo de aplicaciones móviles híbridas por medio de una aplicación para difundir atractivos turísticos.

### 1.2.2. Objetivos Específicos.

- Elaborar documentación técnica que servirá como guía para las personas interesadas en el framework Ionic 2.
- Implementar la metodología SCRUM en el proceso de desarrollo.
- Desarrollar una aplicación móvil para la promoción de atractivos turísticos en el Gobierno Autónomo Descentralizado del cantón Pimampiro.

### **1.3. Alcance y Limitaciones**

#### **1.3.1. Alcance.**

##### **1.3.1.1. De la aplicación.**

Se desarrolló una aplicación móvil para el Gobierno autónomo descentralizado del Cantón Pimampiro, la cual dará a conocer los atractivos turísticos, atractivos culturales, eventos a realizarse en dicho cantón la cual cuyo nombre fue Destino Pimampiro. Esta aplicación se desarrollará en el framework a ser estudiado mediante el consumo de servicios web RESTful.

#### **Estructura de la aplicación Destino Pimampiro**

Backend: En este apartado que lo llamaremos Apartado Administrativo, tendremos al Framework Yii2, en el que se ingresó toda la información recolectada y que es vista por los usuarios de la aplicación, los datos serán almacenados en una base de datos MySQL, en la Fig. 2: Arquitectura de Ionic 2 orientado a servicios se presenta los detalles de la arquitectura que fue utilizada.

Módulos del Apartado Administrativo:

- Lugares Turísticos
- Ubicaciones
- Backup o Respaldo de base de datos
- Administración de usuarios

Servicios: En este apartado que lo llamaremos API, tendremos al Framework Yii2, el que será el proveedor de los servicios web REST los que serán consumidos por la aplicación.

Aplicación: Esta será la encargada de consumir las API's que Yii2 nos ofrecerá, las cuales serán procesadas y finalmente brindará al usuario final la información deseada. Será desarrollada en Ionic 2 y está disponible en la tienda de *Google Play* únicamente para los dispositivos con sistema operativo Android.

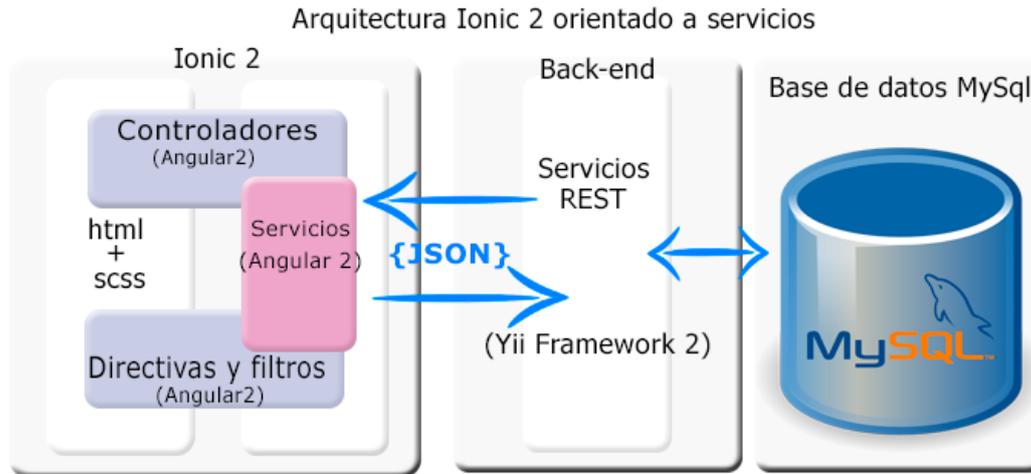


Fig. 2: Arquitectura de Ionic 2 orientado a servicios  
Fuente: Propia

#### 1.3.1.2. *De la investigación.*

Este trabajo de titulación tiene como propósito el análisis de características y funcionalidades del framework Ionic 2 cuyo enfoque es la realización de aplicaciones móviles multiplataforma. Se realizó un estudio y se elaboró documentación sólida que servirá como base para las personas que se interesen en esta tecnología.

### 1.4. Justificación

#### 1.4.1. *De la investigación.*

El presente proyecto de investigación fue realizado en vista de que, no se dispone de la información necesaria del framework Ionic 2, lo cual ocasiona un desconocimiento total de esta nueva tecnología para los desarrolladores de aplicaciones móviles.

#### 1.4.2. *De la aplicación.*

Los beneficiarios directos con el desarrollo de esta aplicación son y serán los ciudadanos tanto locales, nacionales y extranjeros. Este proyecto será de gran utilidad para las personas, ya que tendrán información real y actualizada de atractivos turísticos del cantón Pimampiro.

Después de finalizar el proyecto, este será de gran ayuda para los estudiantes y personas en general que deseen desarrollar aplicaciones móviles utilizando el framework Ionic.

## CAPITULO II

### 2. MARCO TEÓRICO

#### 2.1. Situación actual y proceso de difusión de turismo en el Gad Pimampiro

Pimampiro es un cantón de la Provincia de Imbabura ubicado a 50 kilómetros de la ciudad de Ibarra, en el encontramos hermosos paisajes, herencias ancestrales, lagunas, reservas ecológicas, gente cálida y amable. A más de eso es un cantón multicultural y multiétnico.

Actualmente el GAD municipal de Pimampiro ha dado gran importancia al sector turístico, es por esto que desde hace aproximadamente dos años y medio, se creó el área de turismo dentro de la unidad organizacional del cabildo, la cual entre uno de sus objetivos primordiales estaba la creación me medios de difusión tanto digitales como físicos entre los que se encontraban el levantamiento de la información turística del cantón, campañas publicitarias, stands en ferias turísticas del país, señalización de sitios turísticos, creación de redes sociales y un sitio web destinado al turismo local, entre otras.



Fuente: Destino Pimampiro  
Fig. 3: Logo Destino Pimampiro

Se ha podido apreciar que los esfuerzos han sido productivos y que poco a poco el cantón se ha dado a conocer a nivel local, nacional incluso internacional como un destino turístico y gran parte de esto es gracias a los medios digitales que incluyen en su mayor parte las redes sociales, el sitio web y que se ha dado prioridad a los eventos culturales, festivos y cívicos del cantón.

## 2.2. Introducción a herramientas de Desarrollo de Aplicaciones Móviles Híbridas

### ¿Qué son las aplicaciones móviles híbridas?

El enfoque híbrido combina desarrollo nativo con tecnología Web. Usando este enfoque, los desarrolladores escriben gran parte de su aplicación en tecnologías Web para múltiples plataformas, y mantienen el acceso directo a API's nativas cuando lo necesitan. (IBM Corporation, 2012)

Podemos decir que son ambientes de desarrollo en el cual el programador usa un mismo esquema de trabajo o también puede ser llamado código fuente que puede ser reutilizado en su mayor parte para producir una aplicación para las diferentes plataformas de dispositivos móviles.

Tabla 1: Sistemas Operativos y lenguajes de desarrollo para aplicaciones móviles híbridas.

Fuente: Propia

	Apple iOS	Android	BlackBerry OS	Windows Phone
Lenguajes	Objective C, C, C++	Java, C, C++	Java	C#, VB, .NET
Herramientas	XCode	Android SDK	BB Java Eclipse Plug-in	Visual Studio
Formato	.app	.apk	.cod	.xap
Tiendas	App Store	Play Store	BlackBerry World	Windows Phone Marketplace

Estas tecnologías de desarrollo en un gran porcentaje usan *Cordova* antes conocido como *Apache Cordova* que es un marco de desarrollo móvil de código abierto. Se permite el uso de tecnologías web estándar - HTML5, CSS3 y JavaScript para el desarrollo multiplataforma. Aplicaciones se ejecutan dentro de envolturas dirigidos a cada plataforma, y se basan en los

enlaces de API compatibles con los estándares para acceder a las capacidades de cada dispositivo.

Cordova se encarga de la explotación de las API's de los dispositivos sean estos, sensores, GPS, gestos de pantalla, gestión de almacenamiento, cámara, audio entre otros. Haciendo que el desarrollador utilice estas funciones fácilmente desde lenguaje web.



Fig. 4: Logo Apache Cordova  
Fuente: (Apache Software, 2014)

Las aplicaciones móviles híbridas en su gran mayoría usa un *Web View* o navegador web, es por eso que la mayoría de los lenguajes de programación usan lenguaje de marcado para elaborar páginas web (HTML) como una interfaz gráfica. Esta es una de las razones por la que se torna muy cómodo para los desarrolladores web involucrarse con este tipo de tecnologías, ya que dentro de estas se pueden usar etiquetas HTML, funciones JavaScript<sup>3</sup> y dar un diseño o acabado profesional usando CSS<sup>4</sup>.

En muchos casos dependiendo del avance de las tecnologías y herramientas de desarrollo se usan variaciones de los lenguajes de programación como HTML5 que es una versión avanzada de HTML, TypeScript que es un lenguaje de programación avanzado y potente del ya conocido JavaScript, ECMAScript<sup>5</sup> que es un compilador de lenguajes de programación

---

<sup>3</sup> JavaScript: Lenguaje de programación web, se usa para manipulación de objetos y clases HTML.

<sup>4</sup> CSS: lenguaje de hojas de estilos creado para controlar el aspecto de documentos definidos con HTML.

<sup>5</sup> ECMAScript: es una especificación de sintaxis de lenguajes de programación como JavaScript o JScript..

como TypeScript y finalmente SASS y/o SCSS que son lenguajes de programación basados y/o superiores al ya conocido CSS.

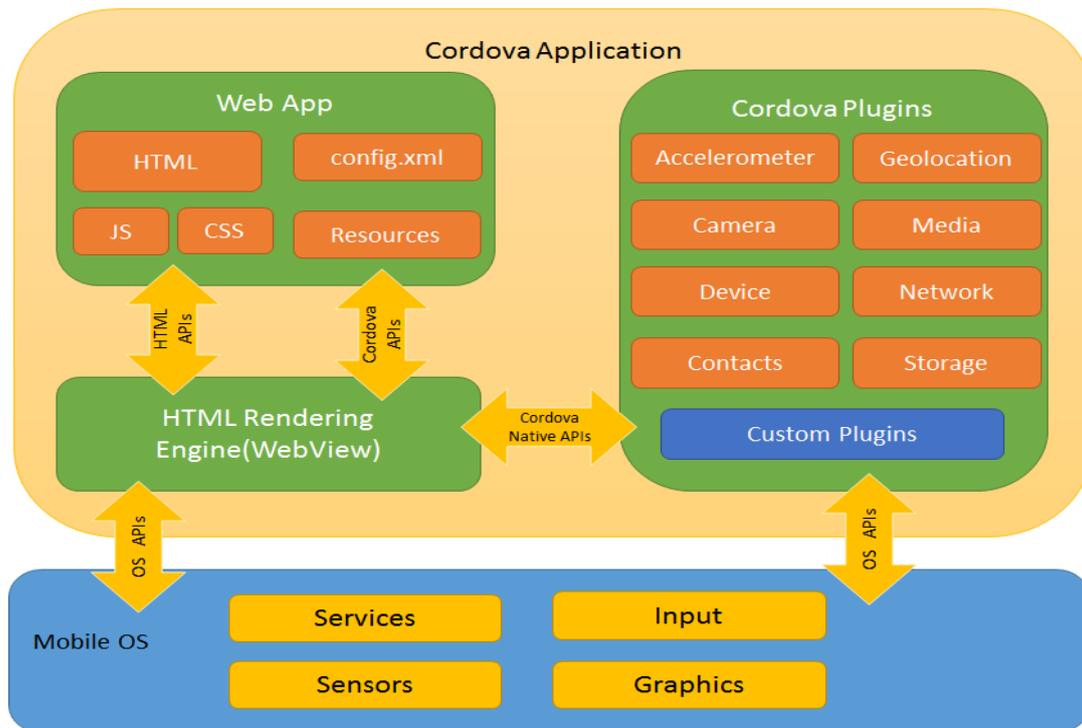


Fig. 5: Funcionamiento de una Aplicación Híbrida  
Fuente: (Apache Software, 2014)

### 2.3. Metodología Scrum

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. (Canós, Letelier, Penadés, & Valencia, 2003) Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas Sprints, con una duración de que varía entre 7 a 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante es las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (Canós et al., 2003)

Scrum es una de las metodologías ágiles más populares. Es una metodología de adaptación, iterativa, rápida, flexible y eficaz, diseñada para ofrecer un valor significativo de

forma rápida en todo el proyecto. Scrum garantiza transparencia en la comunicación y crea un ambiente de responsabilidad colectiva y de progreso continuo. (Tobergte & Curtis, 2013)

Un proyecto de desarrollo de software u otro producto que usa Scrum implica un esfuerzo de colaboración y la implementación de equipos multifuncionales y organizados denominados *Team Scrum*, estos equipos están en la capacidad de dividir su trabajo en lapsos o ciclos de tiempo que se denominan *Sprints*. La Fig. 6: *Ciclo de trabajo de Scrum* hace referencia al ciclo de trabajo en un proyecto de desarrollo *Scrum*.

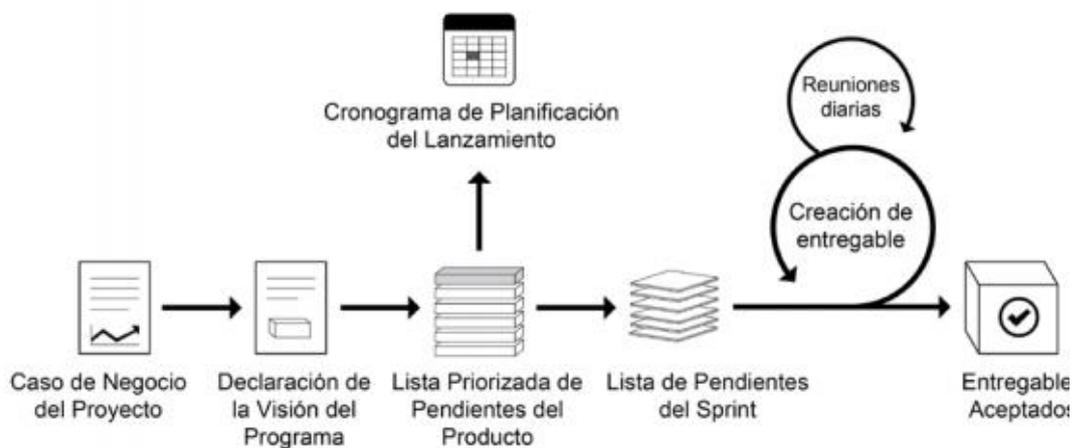


Fig. 6: Ciclo de trabajo de Scrum  
Fuente: (Tobergte & Curtis, 2013)

### 2.3.1. Principios de Scrum.

Los principios de la metodología son pautas y guías básicas aplicadas en el marco de desarrollo de proyectos Scrum y deberán ser aplicadas obligatoriamente para el éxito del producto final.

### 2.3.2. Los valores de Scrum.

En esta metodología se desarrollan valores y formas de convivencia entre los participantes del proyecto entre sí y entre el/los diseñadores del producto también llamado *Product Owner* valores como compromiso, respeto, responsabilidad y apertura al cambio además se basa en

pilares fundamentales como el trabajo en equipo, el aprendizaje a través de la experiencia y el compartir conocimientos.

### **2.3.3. El equipo scrum (Team Scrum).**

Los equipos Scrum son multifuncionales y a su vez organizados entre sí, en el que los multifuncionales tienen la capacidad de llevar a cabo tareas sin necesidad o dependencia de otra u otras personas que no son parte del equipo. Los equipos autoorganizados se encargan de su planificación para completar un sprint sin necesidad de tutoría o dirigencia de personas externas al equipo de trabajo. Este modelo de trabajo en equipo está basado en objetivos tales como impulsar la creatividad, la productividad y la flexibilidad al realizar tareas.

El equipo Scrum está conformado por: El dueño del producto (*Product Owner*), un grupo de desarrolladores (*Development Team*) y el director del proyecto (*Scrum Master*).

La forma de trabajo de un equipo Scrum es a través de entregables es decir entregar avances del producto final de forma iterativa, haciendo a si la retroalimentación más efectiva y potenciando el producto final.

### **2.3.4. El Dueño de Producto (Product Owner).**

El dueño del producto es aquel que guía al equipo de desarrollo y va de la mano con el Scrum Master con la finalidad de que maximizar el valor del producto. El *Product Owner* generalmente interactúa con el Scrum Master para brindar los requerimientos necesarios con la finalidad de tener un producto final acorde a las necesidades. (Tobergte & Curtis, 2013)

### **2.3.5. El Equipo de Desarrollo (Development Team).**

Está conformado por un grupo de personas que se encargan de crear o desarrollar según sea el caso un incrementable del producto final.

Los Equipos de Desarrollo son estructurados y empoderados por la organización para organizar y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del Equipo de Desarrollo.(Schwaber & Sutherland, 2013)

### **2.3.6. El director del Proyecto (Scrum Master).**

Es el encargado de que los requerimientos obtenidos del *Product Owner* sean entendidos a cabalidad por el *Team Scrum*. Desempeña el rol de líder y ayuda a que las iteraciones o *Sprints* incrementen el valor del producto final.

### **2.3.7. El Sprint.**

Es el corazón de la metodología SCRUM, es un ciclo de tiempo en el cual se desarrollan incrementables del producto final.

En el ciclo de desarrollo del producto puede contener varios *Sprints* y en cada uno de estos se debe realizar reuniones diarias, planificaciones y retrospectivas.

Durante un *Sprint* se puede realizar modificaciones al producto, cada *Sprint* tiene como duración entre 2 a 4 semanas, si llegara a sobrepasar este tiempo la complejidad y el riesgo se incrementan.

## **2.4. Herramientas utilizadas para el desarrollo**

### **2.4.1. Yii Framework 2.**

#### **2.4.1.1. Definición.**

Es un framework o entorno de desarrollo de software empresarial web. Proviene del vocablo chino *Yee* cuyo significado es “Simple y Revolucionario”, pero para mejoras en su pronunciación en occidente y Europa se lo nombró *Yii* que es un acrónimo de “Yes It Is” (Yii Software LLC., 2014)

*Yii2* es un framework MVC<sup>6</sup> de desarrollo de software libre basando en el lenguaje de programación PHP y orientado a objetos, virtudes que lo hacen muy potente y rápido al momento de desarrollar aplicaciones web.

Debido a su arquitectura basada en componentes y soporte de caché sofisticado, *Yii2* es especialmente adecuado para desarrollar aplicaciones de gran escala tales como portales,

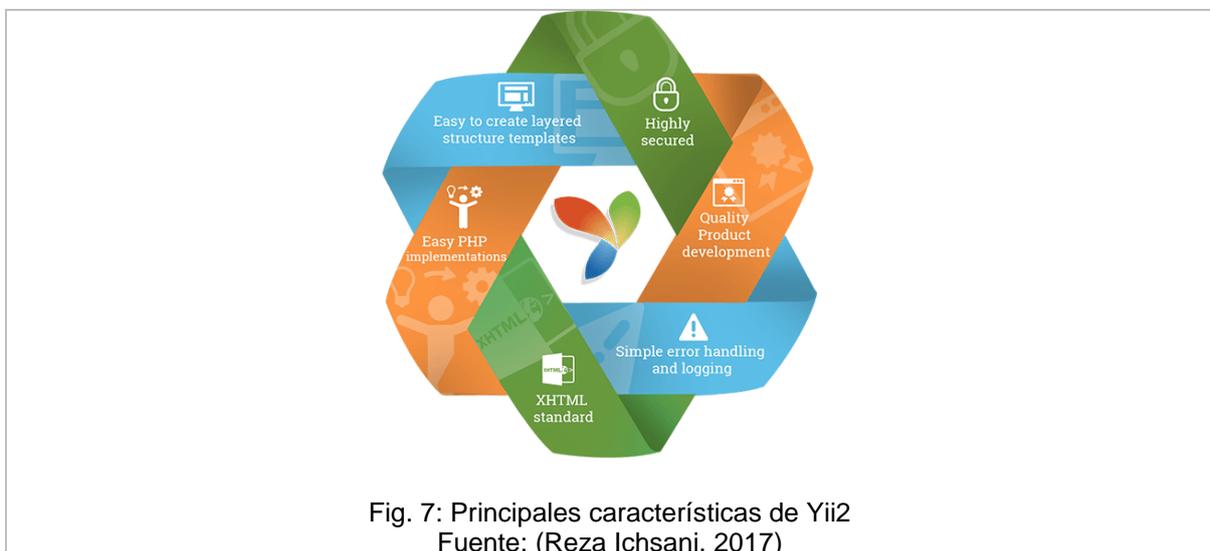
---

<sup>6</sup> MVC: Arquitectura modelo-vista-controlador para el desarrollo de software.

foros, sistemas de gestión de contenido (CMS), proyectos de comercio electrónico, servicios Web RESTful, etc. (Yii Software LLC., 2014)

#### 2.4.1.2. Características.

Yii2 es un framework de diseño de software con arquitectura MVC (Modelo - Vista - Controlador). Fomenta la reutilización de código fuente, incrementa su potencial con el uso de plantillas o *templates*, extensiones, módulos y/o componentes ajenos a su CORE<sup>7</sup> que pueden ser desarrollados por los programadores lo que hace que sea un framework amigable con el programador.



Soporta de forma nativa el desarrollo de servicios web de tipo REST y generador de código GII<sup>8</sup> cuya principal característica es la generación de módulos, modelos, formularios, extensiones y CRUD<sup>9</sup>.

Construido sobre PDO, Yii DAO (Database Access Objects) proporciona una API orientada a objetos para acceder a bases de datos relacionales y no relacionales. (LLC Yii, 2014)

<sup>7</sup> CORE: código fuente base de un framework o software

<sup>8</sup> GII: Herramienta generadora de código fuente del framework Yii2

<sup>9</sup> CRUD: Acrónimo que hace referencia a los verbos de programación *cr*éate, *r*ead, *u*pdate y *d*elete

Incorpora de forma nativa la configuración de acceso a base de datos PostgreSQL, MySQL, MariaDB, Sql Server, Oracle, Cubrid y SQLite

### **2.4.1.3. Plantillas de Yii2.**

#### 2.4.1.3.1. *Plantilla básica de Yii2.*

Esta plantilla incorpora una versión básica en su estructura, mas no reduce su potencia y normal funcionamiento, consta de módulos, modelos, controladores y vistas. Incorpora el generador de código GII para la generación de código fuente funcional desde la base de datos al entorno de desarrollo implementando la arquitectura MVC. En esta plantilla es necesario realizar la configuración manual del componente RBAC (Control de Acceso Basado en Roles) para la administración de usuarios y permisos dentro de la plataforma a desarrollarse.

#### 2.4.1.3.2. *Plantilla avanzada de Yii2.*

Incorpora dos ambientes de trabajo un *frontend* que generalmente es usado como la parte visual de la plataforma o el lado del cliente y el backend usado como capa de negocio es decir el lado administrador.

Se podría decir que la plantilla avanzada incorpora dos plantillas básicas incluso podrían ser tres o más, esta plantilla es ampliable en muchos de los casos se incorpora un módulo para servicios web dentro del *frontend* o *backend* o de forma externa. En esta plantilla no es necesaria la configuración del componente RBAC ya que trae incorporado la administración de usuarios mismo que puede ser inicializado con la siguiente línea de comandos de símbolo de sistema `yii migrate --migrationPath=@yii/rbac/migrations`.

```

drop table if exists `auth_assignment`;
drop table if exists `auth_item_child`;
drop table if exists `auth_item`;
drop table if exists `auth_rule`;

create table `auth_rule`
(
  `name`          varchar(64) not null,
  `data`          blob,
  `created_at`   integer,
  `updated_at`   integer,
  primary key (`name`)
) engine InnoDB;

create table `auth_item`
(
  `name`          varchar(64) not null,
  `type`          smallint not null,
  `description`   text,
  `rule_name`     varchar(64),
  `data`          blob,
  `created_at`   integer,
  `updated_at`   integer,
  primary key (`name`),
  foreign key (`rule_name`) references `auth_rule` (`name`) on delete set null on update cascade,
  key `type` (`type`)
) engine InnoDB;

create table `auth_item_child`
(
  `parent`        varchar(64) not null,
  `child`         varchar(64) not null,
  primary key (`parent`, `child`),
  foreign key (`parent`) references `auth_item` (`name`) on delete cascade on update cascade,
  foreign key (`child`) references `auth_item` (`name`) on delete cascade on update cascade
) engine InnoDB;

create table `auth_assignment`
(
  `item_name`     varchar(64) not null,
  `user_id`       varchar(64) not null,
  `created_at`   integer,
  primary key (`item_name`, `user_id`),
  foreign key (`item_name`) references `auth_item` (`name`) on delete cascade on update cascade
) engine InnoDB;

```

Fig. 8: Migración RBAC Yii2  
Fuente: Propia

2.4.1.4. **Uso de MVC en Yii2.**

El patrón arquitectónico usado por Yii2 en este caso MVC se detallan en la Fig. 9: *Uso de MVC en Yii2*

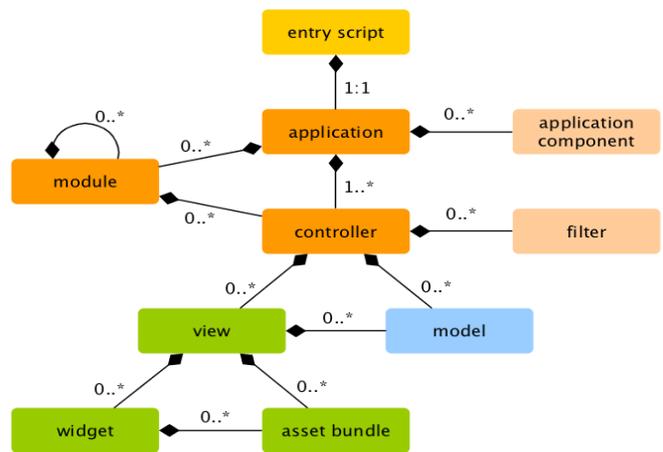


Fig. 9: Uso de MVC en Yii2  
Fuente: <http://www.yiiframework.com/doc-2.0/guide-structure-overview.html>

#### 2.4.1.4.1. Modelo.

Son un conjunto de clases entre ellas se encuentra la capa de abstracción de la base de datos, básicamente es la capa de acceso a datos y donde se realiza el proceso de negocio. (Bahit, 2011)

Yii2 está basado en programación orientada a objetos POO es por eso que los modelos usan herencia en este caso los modelos de acceso a datos heredan de *ActiveRecord*<sup>10</sup>.

#### 2.4.1.4.2. Vista.

La vista es la interfaz gráfica o capa de presentación en la que se da a conocer al usuario mediante textos o gráficos información. estructura la vista es código HTML, CSS, JavaScript y PHP. El nombre de una vista debe estar como una acción en un controlador como se puede apreciar en la *Fig. 10: Acciones y vistas en Yii2*.

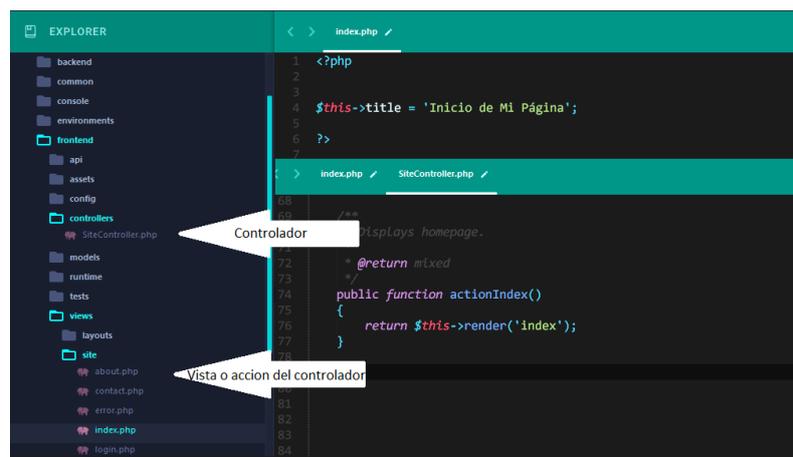


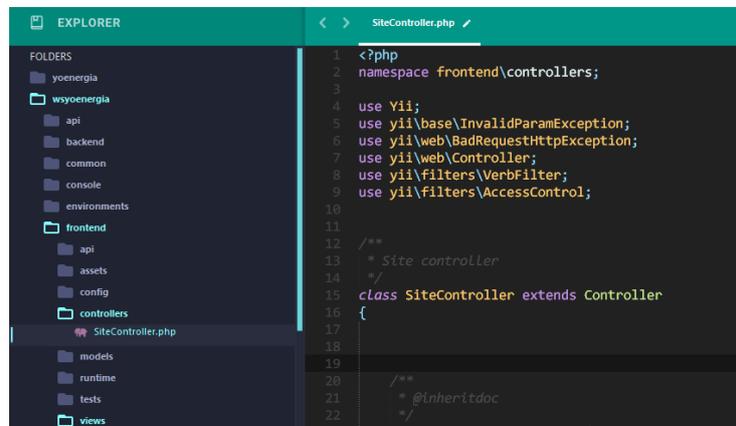
Fig. 10: Acciones y vistas en Yii2  
Fuente: Propia

#### 2.4.1.4.3. Controlador.

Son clases que extiende de *yii\base\Controller*, contienen métodos y acciones tales como peticiones y respuestas a las que acceden las vistas y a su vez los controladores acceden a los modelos mediante instancias y así los modelos poder acceder a datos y devolverlos a las

<sup>10</sup> ActiveRecord: Interfaz orientada a objetos para acceder y manipular una o más bases de datos.

vistas. En Fig. 11: Notación de un Controlador en Yii2 se observa detalladamente la notación de un controlador.



```
1 <?php
2 namespace frontend\controllers;
3
4 use Yii;
5 use yii\base\InvalidParamException;
6 use yii\web\BadRequestHttpException;
7 use yii\web\Controller;
8 use yii\filters\VerbFilter;
9 use yii\filters\AccessControl;
10
11
12 /**
13  * Site controller
14  */
15 class SiteController extends Controller
16 {
17
18
19
20 /**
21  * @inheritdoc
22  */
23 public function behaviors()
```

Fig. 11: Notación de un Controlador en Yii2  
Fuente: Propia

#### 2.4.1.5. **Bootstrap en Yii2.**

Bootstrap es un framework frontend potente, sencillo e intuitivo basado en html5, CSS3 y JQuery pionero en el desarrollo de páginas web *responsive* creado por Mark Otto y Jacob Thornton de Twitter, es un framework de código abierto lo que lo hace muy extensible y personalizable desarrollado por, en la actualidad existen miles de plantillas realizadas en base a Bootstrap.



Fig. 12: Logotipo Bootstrap  
Fuente: <https://getbootstrap.com>

Bootstrap se maneja bajo el concepto de *responsive*, esto es, que el desarrollador escribe un solo código y mediante este framework la página web se adapta al dispositivo sea este un ordenador, tabletas o teléfonos inteligentes.

## 2.4.2. Base de datos MySQL.

### 2.4.2.1. *Definición.*

MySQL es un motor de base de datos SQL<sup>11</sup> muy rápido, multihilo, multiusuario y robusto. MySQL está diseñado para sistemas de producción de carga crítica y de misión crítica, así como para incrustar en software de implementación masiva. (Reference Manual, 2017)

Actualmente MySQL es parte de Oracle, es una base de datos de dos tipos de licencias GNU<sup>12</sup> y pagada bajo licenciamiento por parte de Oracle.

### 2.4.2.2. *Características.*

Al igual que sus similares MySQL contiene características esenciales y básicas que un sistema gestor de base de datos relacionales como son transacciones, procedimientos almacenados, funciones, triggers o disparadores, integridad referencial, etc. Es muy popular entre los desarrolladores de sistemas web y páginas web.



Fig. 13: Logo MySQL

Fuente: [www.mysql.com](http://www.mysql.com)

Entre una de sus principales características MySQL es un sistema de gestión de base de datos relacionales es decir incorpora transacciones y claves o llaves foráneas. La conectividad a la base de datos es sumamente segura, el puerto que por defecto incorpora

---

<sup>11</sup> SQL: (Structured Query Language) es un lenguaje de programación estándar mediante el cual se accede a información de la base de datos.

<sup>12</sup> GNU: General Public Licence por sus siglas en inglés, es un estándar de licenciamiento gratuito.

es el 3306 que puede ser modificado si el usuario lo desea. La replicación es una de sus fortalezas ya que se puede realizar en tiempo real.

### **2.4.3. JSON.**

#### **2.4.3.1. Definición.**

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. (JSON, s.f.)



Fig. 14: Logo JSON  
Fuente: [www.byspel.com](http://www.byspel.com)

Es usado como intercambio de datos en servicios web cuyas peticiones y respuestas se realizan mediante HTTP y HTTPS. Uno de los lenguajes de programación que más aceptación dio a JSON e incremento su usabilidad en el universo de programadores es JavaScript.

En la actualidad es un competidor incluso con más aceptación que XML en el ámbito de servicios web, ya que el servicio web que usa JSON como formato es RESTful cuya implementación es mucho más sencilla que el tradicional SOAP que usa XML. Las plataformas o lenguajes de programación que dan más cabida a servicios web con este formato son PHP, PERL, Python, Rubi, Java, JavaScript y Angular.

#### **2.4.3.2. Características.**

Son muchos los aspectos que hace un potencial a esta notación es su simplicidad, la forma de leer o escribir hace que sea sumamente fácil de identificar e incluso legible para cualquier tipo de persona que no sea un programador.

Es un lenguaje independiente es decir que tiene su propia notación y a su vez compatible con la mayoría de los lenguajes de programación. Como todos lenguajes de programación podemos encontrar objetos y arreglos con la particularidad de ser multidimensionales y de fácil acceso a estos.

Muy fácilmente nos podemos dar cuenta que es un archivo es tipo JSON ya que su extensión es *.json*<sup>13</sup> y su tipo de dato en internet o protocolos HTTP es *application/json*.

#### 2.4.3.3. Notación de un archivo JSON.

En la figura podemos visualizar como usar JSON para almacenar información.

```
{
  "libro":
  [
    {
      "id": "01",
      "lenguaje": "POO en Java",
      "edicion": "third",
      "autor": "Juan Pérez"
    },
    {
      "id": "09",
      "lenguaje": "RESTful Web Service",
      "edicion": "third",
      "autor": "Marcelo Sánchez"
    }
  ]
}
```

Fig. 15: Notación de un archivo JSON  
Fuente: Propia

#### 2.4.4. RESTful web services.

En la actualidad existen un sinnúmero de protocolos y estándares, muchos de ellos orientados a HTTP y orientados a Servicios Web tales como WSDL y SOAP.

##### 2.4.4.1. Definición.

REST (Representational State Transfer) es un estilo de arquitectura de software para sistemas hipermedias distribuidos tales como la Web. (Marset, 2007)

---

<sup>13</sup> .json: Extensión de un archivo escrito en formato JSON

### 2.4.4.2. Características.

Hoy en día los servicios web de tipo REST tienen tanta acogida que grandes organizaciones como Spotify, PayPal, Facebook, Microsoft Azure, entre otros, proveen este tipo de servicios web que se los denomina API's que básicamente son rutas las cuales reciben parámetros (Request) y envían resultados (Response).

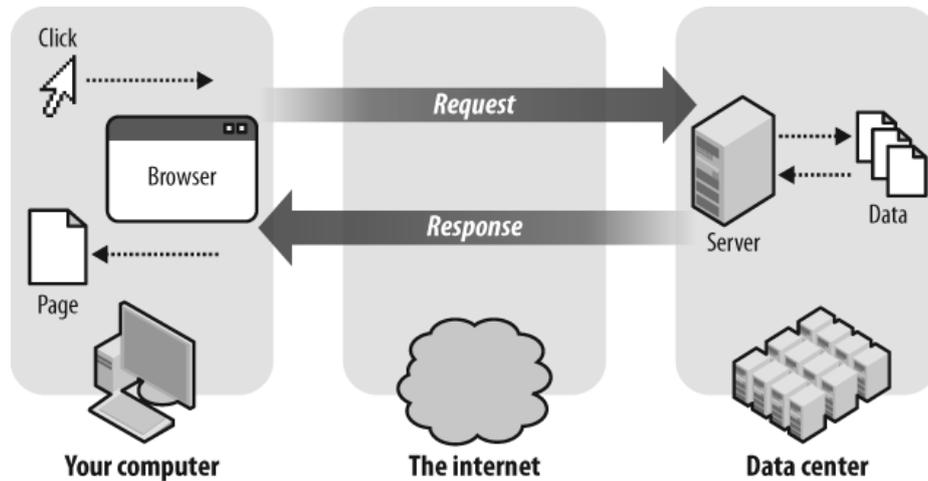


Fig. 16: Ilustración Request y Response  
Fuente: [www.medium.com](http://www.medium.com)

Los métodos que HTTP más usados y más importantes PUT, GET, POST y DELETE. En el lado de servidor o capa de negocio se transforman en operaciones de tipo CRUD como CREATE, UPDATE, READ y DELETE.

Fuente: Propia

Tabla 2: Equivalencias de peticiones HTTP

HTTP	Método Backend	SQL
<b>GET</b>	Read	Select
<b>POST</b>	Create	Insert
<b>PUT</b>	Update	Update
<b>DELETE</b>	Delete	Delete

## CAPITULO III

### 3. FRAMEWORK IONIC 2

#### 3.1. Introducción al Framework

Ionic ha evolucionado y mejorado continuamente desde su lanzamiento a lo largo de estos años desde su lanzamiento la versión 1, misma que desde sus inicios ya usaba Angular. Uno de los aspectos que impulso a Ionic como uno de los frameworks de desarrollo de aplicaciones móviles híbridas es la incorporación de Angular 2 y TypeScript en su versión 2.

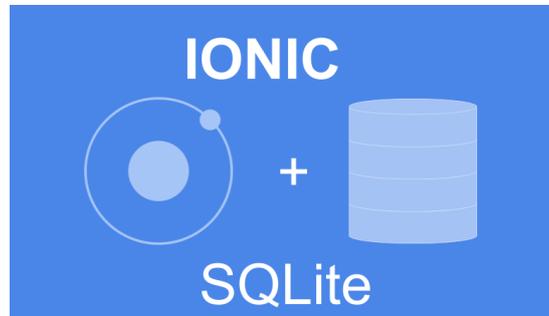
Adicional a la importante y trascendente incorporación de Angular 2 y TypeScript, es que usa versiones avanzadas de Cordova y misma que incluye mejoras en el rendimiento y permite aprovechar de mejor las API's liberadas por los Sistemas Operativos y la incorporación de nuevos componentes.

Aspectos importantes como una menor complejidad de código fuente lo que se traduce a un código de desarrollo más limpio y reutilizable mismo que facilita el desarrollo de aplicaciones escalables. Y con el uso de TypeScript llego a ser una ventaja aun mayor ya que facilita el desarrollo de aplicaciones empresariales y abrió al framework más mercado.

Entre las fortalezas de Ionic es que es un framework 100% libre y de código abierto licenciado por el Instituto Tecnológico de Massachussets (MIT), de desarrollo colaborativo masivo alrededor del mundo, tanto como en su SDK como en los plugins y componentes desarrollados por usuarios que gustan del framework.

Ionic posee más de cien plugins nativos como bluetooth, cámara, huella digital, entre otros y gracias a que es un framework basado en Cordova, tiene gran compatibilidad con plugins desarrollados para esta tecnología e incluso con los desarrollados para otros frameworks como PhoneGap lo que facilita mucho el desarrollo de aplicaciones de diferente complejidad.

Tener una base de datos en un dispositivo siempre es algo deseado y con Ionic 2 esto es posible haciendo uso del plugin SQLite como se indica en la Fig. 17: Plugin SQLite, tanto por la facilidad de i



```
import { SQLite, SQLiteObject } from '@ionic-native/sqlite';
constructor(private sqlite: SQLite) { }
this.sqlite.create({
  name: 'data.db', // INICIALIZAMOS LA BASE DE DATOS
  location: 'default'
})
.then((db: SQLiteObject) => {
  db.executeSql('create table telefono (cliente_id int, c
    liente_telefono VARCHAR(32))', {}) // CREACION DE A TABLA
    .then(() => console.log('Executed SQL')) // Si se ejecuta correctamente
    .catch(e => console.log(e)); // si existe un error al crear la tabla
  })
  .catch(e => console.log(e)); //si existe un error al crear la basede datos
```

Fig. 17: Plugin SQLite  
Fuente: [www.ionicframework.com](http://www.ionicframework.com)

Se encuentra entre las 5 mejores plataformas de desarrollo de aplicaciones híbridas junto a Xamarin de Microsoft, Phonegap, Framework 7 y Native Scritpt.

## 3.2. Instalación y Configuración

### 3.2.1. Pre-requisitos.

Para hacer uso del framework Ionic 2 es necesario cumplir ciertos requisitos para su óptimo funcionamiento.

#### 3.2.1.1. Sobre el Sistema Operativo.

Ionic 2 es compatible con los sistemas operativos que se indica en la *Tabla 3: Requisitos de Sistema Operativo*.

Tabla 3: Requisitos de Sistema Operativo  
Fuente: Propia

Sistema Operativo	Versión
Mac OS	iOS 7 o superior
Linux	Ubuntu 12.04 o superior
Windows	Windows Vista o Superior

### 3.2.1.2. Sobre memoria y almacenamiento.

Para el desarrollo de aplicaciones con Ionic 2 el almacenamiento no es un gran impedimento ya que tan solo se necesitaría 2 Gb adicionales de las que exige el sistema operativo en el que se va a instalar, en cuestión de memoria es necesario adicionar 512 Mb de memoria RAM que las requeridas por el sistema operativo, es recomendable que nuestro ordenador tenga un mínimo de 4 Gb de memoria RAM para que el desarrollo no se vea afectado.

### 3.2.1.3. Software adicional.

El framework que estamos estudiando funciona mediante dependencias es por esto por lo que es estrictamente necesario el uso de NODE JS y para su compilación JDK de Java y Android SDK en este caso la versión mínima es JDK 8.

### 3.2.2. Instalación.

Una vez instalado y configurado los pre-requisitos, procedemos con la instalación para lo es necesario una línea de comandos en Windows usaremos *CMD* y en Linux o Mac OS usaremos *Terminal* y procedemos a escribir el texto indicado a continuación:

```
npm install -g cordova ionic
```

### 3.2.3. Configuración.

La configuración no es más que ubicarnos en un directorio que será en el que crearemos nuestros aplicativos de Ionic 2.

### 3.2.4. Creación del proyecto.

La creación de un proyecto o aplicativo se lo realiza al igual que la instalación por medio de línea de comandos, existen ejemplos o temas para iniciar un proyecto en Ionic 2 estos son:

- *Blank* que es un aplicativo en el que se puede desarrollar desde cero, este tema o template tiene una sola página (home).
- *Tabs* mismo que brinda un espacio de trabajo con tres pantallas y lista horizontal que nos indica en que ventana del aplicativo estaremos.
- *Sidemenu* este tema nos coloca una barra de menú lateral en la que podemos escoger un sin número de opciones

En la *Fig. 18: Creando un proyecto en Ionic 2*. Se puede presenciar los diferentes templates de Ionic 2.

Comenzar una aplicación

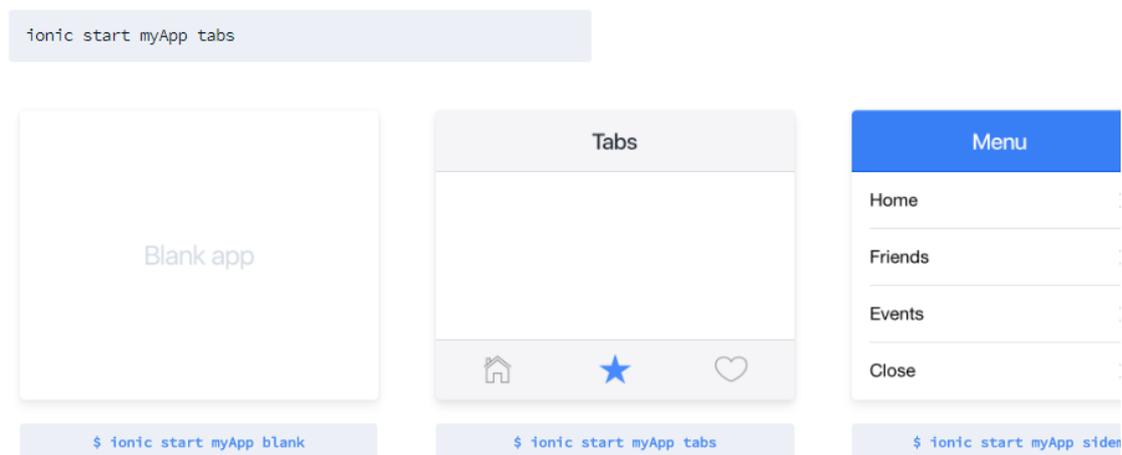


Fig. 18: Creando un proyecto en Ionic 2  
Fuente: [www.ionicframework.com](http://www.ionicframework.com)

### 3.2.5. Estructura del proyecto.

Dependiendo del tipo de proyecto que se requiere, se genera un directorio de archivos y carpetas que contiene información relevante para la construcción de un aplicativo.

Los aplicativos mantienen una estructura similar, la diferencia radica en la organización y estilo de las paginas o interfaces a ser mostradas al usuario final.

Tabla 4: Descripción del contenido de un aplicativo Ionic 2  
Fuente: Propia

Nombre	Tipo	Descripción
hooks	Directorio	Contiene scripts especiales que pueden ser añadidos por los desarrolladores de aplicaciones para personalizar los comandos de Cordova.
node_modules	Directorio	Contiene dependencias de Ionic 2 y Angular 2 para ser importadas o usadas por los desarrolladores durante el desarrollo de la aplicación.
Platforms	Directorio	En este directorio se encuentran las plataformas añadidas sean estas Android, iOS, Browser o Windows juntamente con los scripts o archivos necesarios para su compilación.
pluggins	Directorio	Es un directorio donde se almacenan los plugins nativos que la aplicación ha de usar.
resources	Directorio	Es un directorio con subcarpetas para cada plataforma añadida, esto es iconos e imágenes de splash estándares para diferentes tamaños de dispositivos.
src	Directorio	Es la carpeta más importante del aplicativo ya que en esta se encuentran subcarpetas de configuración del aplicativo como lo es <b>app</b> , interfaces o paginas ( <b>pages</b> ), <b>assets</b> o carpeta donde se almacena recursos de la aplicación como imágenes o tipografías y <b>theme</b> donde se encuentran archivos scss a ser aplicados en el aplicativo.
www	Directorio	Esta carpeta contiene el aplicativo compilado en formato html, css y javascript.
.editorconf	Archivo	Este archivo contiene información para los IDE de programación, es decir información de como presentar los archivos para que el desarrollo sea más cómodo.

.gitignore	Archivo	Archivo de uso para trabajar con repositorios Git.
config.xml	Archivo	Archivo de configuración en formato xml a ser utilizado por cordova, contiene nombre de la aplicación, versión y plugins a ser usados.
ionic.config.json	Archivo	Contiene el nombre de la aplicación, de utilidad para subir a repositorios de Ionic.
package.json	Archivo	Archivo de dependencias del aplicativo se encuentra todas las dependencias a ser usadas o ser instaladas en caso de una migración.
README.md	Archivo	Este es un archivo de información para repositorios Git.
tsconfig.ts	Archivo	Contiene información sobre la versión de TypeScript a ser usada en el aplicativo.
tslint.json	Archivo	Archivo que contiene información a ser usada en el proceso de compilación, brindara alertas cuando no se encuentre correcto.

En la Fig. 19: Estructura de archivos de un proyecto nos ilustraremos con la estructura.

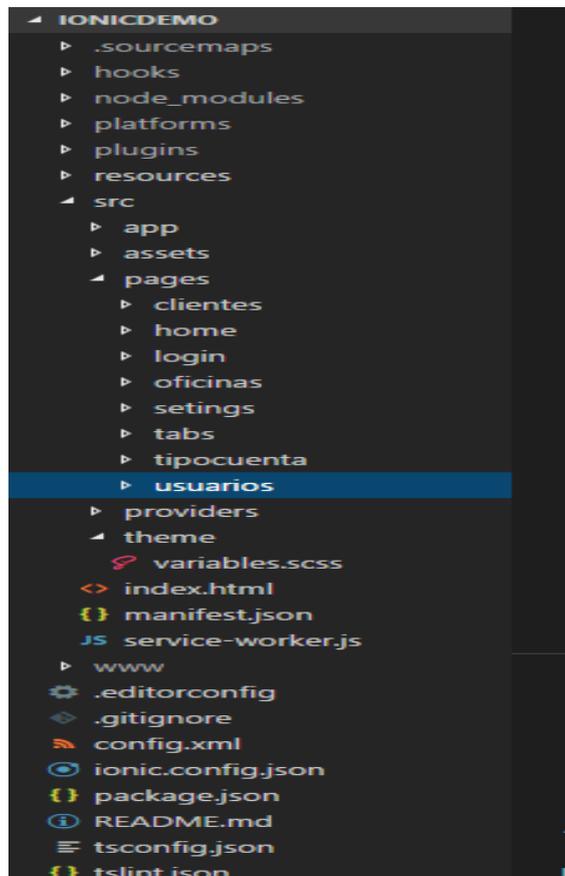


Fig. 19: Estructura de archivos de un proyecto  
Fuente: Propia

### 3.2.6. Proceso de compilación.

Ionic 2 nos brinda la posibilidad de ver la evolución de el aplicativo que se está desarrollando para las distintas plataformas es decir que podemos ir observando los cambios que realizamos y el diseño de la aplicación en tiempo real. Esto lo podemos realizar con el comando `ionic serve --lab` este comando compilará nuestro aplicativo para ser visto en un navegador web. Como lo indica la *Fig. 20: Vista previa del aplicativo en Ionic 2*.

Existe la posibilidad de realizar una compilación directa en el dispositivo, esto es sin tener que instalar la aplicación manualmente. Este proceso requiere algunas configuraciones en nuestro dispositivo en el caso de Android en las configuraciones debemos aceptar la *depuración USB* esto permitirá que podamos realizar pruebas de errores o bugs y aceptar *orígenes desconocidos* esto permitirá instalar aplicaciones que no sean descargadas de la tienda oficial.

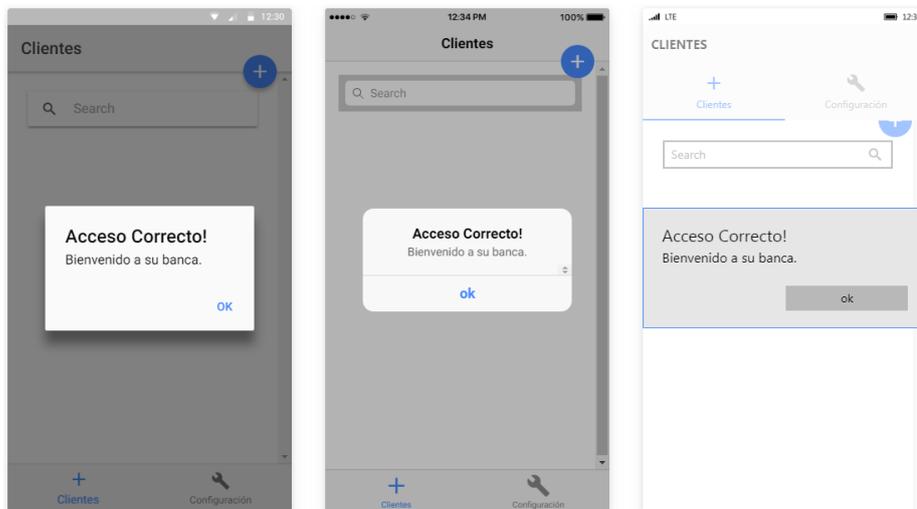


Fig. 20: Vista previa del aplicativo en Ionic 2  
Fuente: Propia

Previo activación de la depuración USB, aceptación de Orígenes desconocidos y tener conectado nuestro teléfono al computado, en la consola de comandos debemos escribir la siguiente instrucción `ionic cordova platform add android` esta sentencia descarga a nuestro aplicativo la plataforma Android para poder compilarla a este Sistema Operativo.

Para el proceso de compilación al dispositivo se debe escribir la instrucción *ionic cordova run android* se puede añadir a la instrucción la bandera *--livereload* que permitirá ver los cambios que se vaya realizando en el dispositivo siempre y cuando siga conectado por USB.

Si es requerida únicamente la aplicación y no se desea compilar a un dispositivo se *ionic cordova build android* y si se desea realizar la compilación definitiva se debe añadir la bandera *--prod --release* estos comandos adicionales dejaría nuestra aplicación lista para ser firmada y subida a la tienda.

### **3.3. Características del Framework**

La integración de Angular 2 y TypeScript fueron fundamentales para el crecimiento como framework y en el mundo de los desarrolladores de aplicaciones móviles.

#### **3.3.1. TypeScript.**

Es un lenguaje de programación moderno que permite crear aplicaciones web robustas en JavaScript. TypeScript no requiere de ningún tipo de plugin, puesto que lo que hace es generar código JavaScript que se ejecuta en cualquier navegador, plataforma o sistema operativo. (Valverde Ramos & Hernández-Mora de Fuentes, 2014).

Entonces TypeScript es un lenguaje de programación compilador o transcriptor de lógica de programación o instrucciones de un lenguaje a otro en este caso es JavaScript, un claro ejemplo de cómo funciona o que es TypeScript es el resultado de CSS previa aplicación o compilación de SASS o LESS.

Una de las ventajas que tiene sobre JavaScript es el resultado final en líneas de código en la *Fig. 21: Conversión de TypeScript a JavaScript* veremos un ejemplo de transcripción de un fragmento de código que lo que hace es recorrer un ciclo e imprimiéndolo en la consola del navegador.

Enfocado en la programación orientada a objetos TypeScript con la capacidad de interpretar con facilidad clases, constructores, herencia e incluso interfaces, esto impulsado por ES5, que es ECMAScript 5 que es una estandarización internacional ECMA internacional.

(Saini, 2017). Estándar que usan los desarrolladores de navegadores web o Web Browsers para compilación de JavaScript.

```
TypeScript

let list = ["a", "b", "c"];

for (let b of list) {
  console.log(b); // a, b, c
}

Javascript

var list = ["a", "b", "c"];

for (var _i = 0, list_1 = list; _i < list_1.length; _i++) {
  var b = list_1[_i];
  console.log(b); // a, b, c
}
```

Fig. 21: Conversión de TypeScript a JavaScript  
Fuente: Valverde Ramos & Hernandez- Mora. Pag 25

### 3.3.1.1. *Tipos de datos.*

#### 3.3.1.1.1. *Tipos Primitivos.*

Como en la mayoría de los lenguajes de programación, TypeScript posee tipos de datos primitivos como boolean, Number y String, tal como se describe en la Fig. 22: Tipos de datos primitivos. La estructura de programación permite usar templates o plantillas, en las que es posible inyectar variables directamente en la plantilla, misma que puede ser modificada dinámicamente en infinidad de veces.

```
1 let miVariableBoolean : boolean = false;
2 let miVariableNumerica : number = 25;
3 let miVariableString : string = 'Pedro Perez';
4 //USANDO UN TEMPLATE
5 let saludo : string = `Hola, mi nombre es ${ miVariableString } y
  tengo ${ miVariableNumerica } años.`
6
7 //EQUIVALENTE DEL TEMPLATE
8 let equivalente : string = "Hola, mi nombre es" +
  miVariableString + " y tengo " + miVariableNumerica + "años."
```

Fig. 22: Tipos de datos primitivos  
Fuente: Propia

### 3.3.1.1.2. Arrays o arreglos.

En la mayoría de los lenguajes de programación los arreglos son un dolor de cabeza para los programadores principiantes, pero con TypeScript no es el caso trabajar con arreglos es muy sencillo y des complicado. Aquí un ejemplo de su declaración.

```
let arregloNumeros: Array<number> = [1, 2, 3];  
let arregloString: Array<string> = ['azul', 'verde', 'rojo'];
```

Adicional se puede realizar trabajos de filtrado, eliminación, modificación y agregar elementos al array.

### 3.3.1.1.3. Tuplas.

En muchas de las ocasiones los desarrolladores tienen la necesidad de almacenar constantes o variables de diferentes tipos de datos, las matrices pueden que nos ayuden en muchos de los casos, pero ocupan muchos recursos ya que son arreglos multidimensionales, en este caso específico TypeScript nos da un tipo de dato que lo que hace es brindarnos una colección de valores heterogénea que puede contener uno o más tipos de datos primitivos que pueden ser almacenados en esta.

Las tuplas se inicializan asignando un nombre y en corchetes (`[]`) añadimos el o los tipos de datos que contiene la tupla separados por coma ( , ) como indica Fig. 23.



Fig. 23: Estructura y representación de Tuplas  
Fuente: Propia

Este tipo de datos es desconocido por la mayoría de los programadores ya que no es de fácil comprensión su funcionalidad, pero una de las mejores aplicaciones son el agrupamiento de constantes, ya que en una aplicación podemos encontrar constantes de distintos tipos de datos.

### 3.3.1.2. Funciones.

Las funciones con estructuras de una clase de código reusable, que sirve para realizar tareas específicas sean estas de cálculo, almacenamiento o modificación de datos. Al igual que JavaScript, en TypeScript podemos hacer uso de los diferentes tipos de funciones sean estas de retorno de datos, de cálculo de operaciones o de asignación también conocidas como callback.

Se puede hacer uso de parámetros opcionales, ya que en muchas ocasiones es requerido el uso de una función específica para realizar diferentes operaciones como se explica en la *Fig. 24*.

Al igual que los parámetros opcionales podemos usar parámetros de descanso o reposo, estos parámetros no restringen la cantidad de valores que una función puede recibir. Para indicar que se trata de este tipo parámetros, en la declaración de las funciones se debe ante poner 3 puntos (...) antes del parámetro de reposo tal como indica la *Fig. 25: Funciones con parámetros de descanso*

```
TypeScript (AngularJS 2.0.0-alpha.47)
1 function detalleorden(codigo : number, nombre :string, email?:string) {
2   let div=document.getElementById('div')
3   let div2=document.getElementById('div2')
4   if(email!=undefined) {
5     div.innerHTML ="Código : "+ codigo + " Nombre : "+nombre+ " Email : "+ email;
6   }else{
7     div2.innerHTML ="Código : "+ codigo + " Nombre : "+nombre;
8   }
9 }
10
11 detalleorden(123,"John");
12 detalleorden(111,"mary","mary@xyz.com");
```

Código : 111 Nombre : mary Email : mary@xy  
Código : 123 Nombre : John

Fig. 24: Funciones con parámetros opcionales  
Fuente: Propia

Las funciones Lambda han tomado fuerza en los últimos años ya que son funciones anónimas, también llamadas *Arrow Functions* o funciones de flecha. Son de fácil uso y simplifican código fuente al ser usadas ya que es una declaración de una función anónimas que hace referencia o apunta a un bloque de código de programación que realiza un algún tipo calculo.



```
TypeScript (No-Library (pure JS)) ▾
1 //FUNCION CON PARAMETROS DE DESCANSO
2 function totalFactura(... nums:number[]) {
3   let div=document.getElementById('div')
4
5   var i;
6   var suma:number = 0;
7
8   for(i = 0; i<nums.length; i++) {
9     suma = suma + nums[i];
10  }
11  div.innerHTML =("El valor de la factura es: <b>"+suma+"<b>")
12 }
13 totalFactura(13.99,43.67,31.25)
14
```

El valor de la factura es: **88.91**

Fig. 25: Funciones con parámetros de descanso  
Fuente: Propia

Dentro de las funciones no se necesita especificar el tipo de dato que ha de ser recibido como parámetro, esto es opcional ya que dentro de la función se puede hacer un reconocimiento interno en la función echemos un vistazo en la *Fig. 26: Funciones de Flecha (Arrow Function)*.

Este tipo de funciones son las más usadas dentro de los desarrolladores de Angular 2 e Ionic ya que tienen mucha versatilidad al momento de trabajar con servicios RESTfull, esto es por el tipo de respuesta que se tiene dentro de las API'S.



```
TypeScript (No-Library (pure JS)) ▾
1 var fArrow = (x) => {
2   let div = document.getElementById('div')
3   let div2 = document.getElementById('div2')
4   if(typeof x=="number") {
5     div.innerHTML = 'Parámetro => tipo de dato: ' + typeof x + ' valor: ' +x;
6   } else if(typeof x=="string") {
7     div2.innerHTML = 'Parámetro => tipo de dato: ' + typeof x + ' valor: ' +x;
8     //agregamos color para diferenciar
9     div2.style = 'color: green';
10  }
11 }
12 fArrow(199.23)
13 fArrow("Tomas Lopez")
```

Parámetro => tipo de dato: number valor: 199.23  
Parámetro => tipo de dato: string valor: Tomas Lopez

Fig. 26: Funciones de Flecha (Arrow Function)  
Fuente: Propia

### 3.3.1.3. Promesas o Promises.

Las promesas existen en muchos de los motores de JavaScript y en su mayoría se usan para capturar o manejar errores cuando se ejecutan operaciones asíncronas es decir operaciones que se ejecutan en este momento o en un futuro, y sus variables pueden ser asignadas en cualquier momento durante la ejecución del aplicativo.

Las promesas representan un resultado eventual de una operación asíncronica, la primera manera de interactuar con una promesa o *Promise* es a través del método *then* el cual registra el *Callback* que recibirá la respuesta o la razón por la cual la promesa no ha podido ser cumplida. (Valverde Ramos & Hernández-Mora de Fuentes, 2014).

Entonces entendemos que las promesas en la mayoría de los casos nos ayudan a capturar errores durante la ejecución de un método, un ejemplo básico sería el cálculo del impuesto al valor agregado de una factura en la que se recibe el subtotal de la factura como se indica en la *Fig. 27: Promise o Promesas*

### 3.3.1.4. Interfaces.

Dentro de TypeScript existen principios y uno de estos es verificar que los valores o datos estén asignados al tipo de dato correcto. Las interfaces son definidores de sintaxis y se encargan de comprobar los tipos de las variables que se pasan como argumentos. Contienen métodos, propiedades y eventos.



```
1 let subtotal = "87.58";
2 const promise = new Promise((resolve, reject) => {
3   if (typeof subtotal == 'number'){
4     resolve(subtotal)
5   }else if (typeof subtotal == 'string'){
6     reject(new Error("Algo malo ha pasado, el valor obtenido es : <b>" +typeof subtotal
7     + "</b>"));
8   }
9 });
10 promise.then((res) => {
11   let iva = 0.14;
12   document.body.innerHTML = 'valor de la factura: $'+ (res + res *iva).toFixed(2)
13 });
14 promise.catch((err) => {
15   document.body.innerHTML = err.message
16 });
```

Algo malo ha pasado, el valor obtenido es : string

Fig. 27: Promise o Promesas  
Fuente: Propia

TypeScript brinda la facilidad que dentro de la definición de una interfaz usar funciones de tipo flecha o *Arrow Function*, una de las ventajas de usar este tipo de interfaces es por ejemplo en una interfaz de tipo factura, en la que se especifique el subtotal y se calcule de nacimiento y automáticamente la Arrow Function calculara su edad como lo veremos en el siguiente ejemplo:

```
TypeScript (No-Library (pure JS)) ▼
1 interface ITotales {
2   subtotal : number,
3   impuesto : number,
4   total: () => number
5 }
6
7 var customer:ITotales = {
8   subtotal: 100,
9   impuesto: 0.12,
10  total: () =>{return customer.subtotal + customer.impuesto * customer.subtotal }
11 }
12
13 document.body.innerHTML = "Subtotal: "+customer.subtotal +
14   "<br> Impuesto: " +customer.impuesto+
15   "<br> Total: " + customer.total()
16
```

Subtotal: 100  
Impuesto: 0.12  
Total: 112

Fig. 28: Interfaces  
Fuente: Propia

### 3.3.1.5. Clases.

Para los desarrolladores de programación orientada a objetos JavaScript resulta un poco incomodo ya que hace sudo de funciones y de herencia basada en prototipos para omitir el uso de clases y construir componentes reusables, no siendo el caso de TypeScript ya que se puede hacer uso de clases e interfaces de manera dinámica, sencilla y de fácil aplicación.

La nomenclatura de una clase no varía mucho de los lenguajes de programación tradicionales, la declaración de la clase la podemos realizar con la palabra reservada *class* seguido del nombre de la clase. En caso de que se requiera de instanciar una clase esta debe contener un constructor con la palabra reservada *constructor*.

Dentro de una clase podemos hacer uso de funciones sean estas de cálculo o retorno como son los mutadores y accesoros conocidos como *Getters* y *Setters* para acceder a las variables declaradas dentro de los métodos.

Podemos hacer uso de clases e interfaces para acceder a los datos o realizar operaciones que se requiera.

```
TypeScript (No-Library (pure JS)) ▼
1 class Estudiante {
2   nombres : string;
3   constructor(public nombre, public primerApellido, public segundoApellido) {
4     this.nombres = nombre + " " + primerApellido + " " + segundoApellido;
5   }
6 }
7 interface Persona {
8   nombre: string;
9   primerApellido: string;
10 }
11 function saludo(persona : Persona) {
12   return "Hola, " + persona.nombre + " " + persona.primerApellido;
13 }
14 var usuario = new Estudiante("Juan", "Martinez", "Perez");
15
16 div = document.getElementById('div');
17 div2 = document.getElementById('div2');
18 div.innerHTML = "Usando interfaz: " + saludo(usuario);
19 div2.innerHTML = "Usando una instancia de la clase: " + usuario.nombres;
```

Usando interfaz: Hola, Juan Martinez  
Usando una instancia de la clase: Juan Martinez Perez

Fig. 29: TypeScript uso de clases  
Fuente: Propia

### 3.3.2. Angular 2.

Desde su lanzamiento hace más de cuatro años, Angular es uno de los frameworks más populares y preferidos por los desarrolladores JavaScript.

¿Por qué Angular? Su nacimiento radica en que JavaScript es un lenguaje dinámico, asíncrono y de una depuración compleja. Con estas particularidades hacen que sean más complejo de adaptarse a desarrolladores tradicionales ya que la misma lógica de programación no funciona en otros lenguajes de programación como Java, Python o C#. He aquí la necesidad crear un lenguaje de programación más lógicamente compatible con los lenguajes de programación tradicionales.

Aunque fue una de las actualizaciones más esperadas y que más dieron de hablar en el mundo de los desarrolladores web. Angular 1 al igual que la segunda versión fueron y son de gran utilidad para los desarrolladores web y de aplicaciones móviles híbridas en la que facilitó el desarrollo y la administración de aplicaciones del lado del cliente. Este framework se convirtió en una plataforma de desarrollo de aplicaciones web.

Angular 2 es otro framework, no simplemente una nueva versión. A los que no conocían Angular 1 esto les será indiferente, pero los que ya dominaban este framework sí deben entender que el conocimiento que necesitan adquirir es poco menos que si comenzasen desde cero. (Basalo, 2016).

Es un framework simplificado que permite a los desarrolladores crear aplicativos basados en programación orientada a objetos usando clases JavaScript. Los controladores y vistas se han de remplazar por componentes.

### 3.4. Componentes del Framework

#### 3.4.1. Action Sheet (Acciones).

Es un conjunto de opciones que se desliza desde la parte inferior del dispositivo y se sobrepone en la pantalla la misma que nos muestra acciones de confirmación o cancelación.

#### 3.4.2. Alerts (Alertas).

Las alertas son de gran utilidad al momento de indicar a el usuario información importante específica sobre algún evento, orientar al usuario a que tome una decisión especifica o en su defecto usarla para seleccionar una o más acciones a realizarse.

La mayoría de los desarrolladores consideran a las alertas como modales es decir ventanas flotantes para ejecutar acciones rápidas tales como notificaciones y verificación de datos, pero tienen muchas funcionalidades ya que son muy adaptables podemos usarlas como formularios e incluso para poder visualizar mapas o información de estos.

#### 3.4.3. Badges (Insignias).

Los Badges o insignias generalmente los usamos como indicadores que comúnmente son número y se usan dentro de un bloque de texto o en una lista. En la *Fig. 30: Ejemplo de uso de Badges* podemos visualizar un ejemplo de cómo usar un badge.

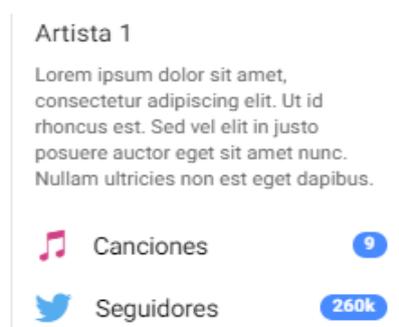


Fig. 30: Ejemplo de uso de Badges  
Fuente: Propia

### **3.4.4. Buttons (Botones).**

Los botones son la manera más sencilla de interactuar entre el dispositivo y el usuario, además permite navegar y realizar acciones específicas mismas que se realizaran luego de que el usuario toque el texto y/o icono que represente al botón.

Los botones están definidos por la etiqueta `<button>` pero se aplica la directiva `ion-button` para mejorar su presentación como tal.

#### **3.4.4.1. Colores.**

Los buttons pueden tener infinidad de colores y se los asigna con la directiva `color="danger"` (en este caso nos dará una tonalidad roja).

#### **3.4.4.2. Formas.**

Se tiene una variedad de formas de botones que tienen un estilo predefinido y que no ahorra tiempo cuando de diseño se trata.

##### **3.4.4.2.1. Outline.**

Esta propiedad nos muestra un botón con borde y texto del color requerido y el relleno color blanco. En este caso aplicamos la opción `outline`

##### **3.4.4.2.2. Clear.**

Esta propiedad nos muestra un botón limpio solo el texto del color requerido aplicando la opción `clear`.

##### **3.4.4.2.3. Round.**

Esta propiedad nos da un botón con las esquinas redondeadas dando un aspecto muy particular. Esta funcionalidad la conseguimos aplicando la opción `round` en la nomenclatura de nuestro botón.

#### 3.4.4.2.4. Icon.

Es una de las funcionalidades más usadas en los botones de Ionic 2, ya que podemos combinar texto e iconos o tan solo iconos. Para conseguir esta funcionalidad debemos que añadir la directiva *ion-icon*. En la Fig. 31 Botones en Ionic 2 se detalla el uso de los botones.

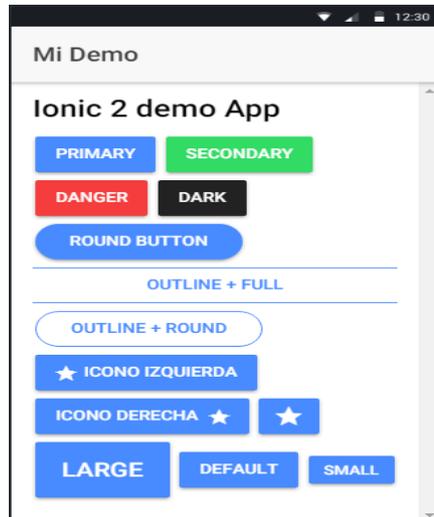


Fig. 31 Botones en Ionic 2  
Fuente: Propia

#### 3.4.5. Cards (Tarjetas).

Las tarjetas son una estupenda manera de presentar contenido y organizar información. Se presentan con un diseño atractivo y facilitan la visualización de información en distintos tamaños de pantalla sin perder estilo. Además, contienen un amplio grado de personalización y componentes que pueden ser usados en su interior como contenido.

Al igual que los botones tenemos una gran variedad de tipos de Cards o tarjetas que pueden ser usados y personalizados por los desarrolladores.

Un Card tiene tres componentes principales *ion-card-header*, *ion-card-content* y *ion-card-footer*, de los cuales *ion-card-content* es obligatorio usar ya que lleva el contenido a ser visualizado por el usuario.

En un Card podemos incorporar listas, imágenes, listas e imágenes e incluso mapas haciendo uso de las API's de Google Maps, Open Street Maps entre otros. En la Fig. 32: Ejemplo de Cards en Ionic 2 podremos visualizar como se presentan los Cards.

### 3.4.6. Checkbox (Botones de verificación).

Los botones de verificación o Checkbox son componentes que permiten marcar casillas con valores booleanos que dan a conocer a la aplicación la opción que el usuario desea seleccionar.

Tienen la misma funcionalidad que en el HTML tradicional, pero en Ionic 2 se puede añadir diseño y funcionalidad con el fin de mejorar su rendimiento en una aplicación.

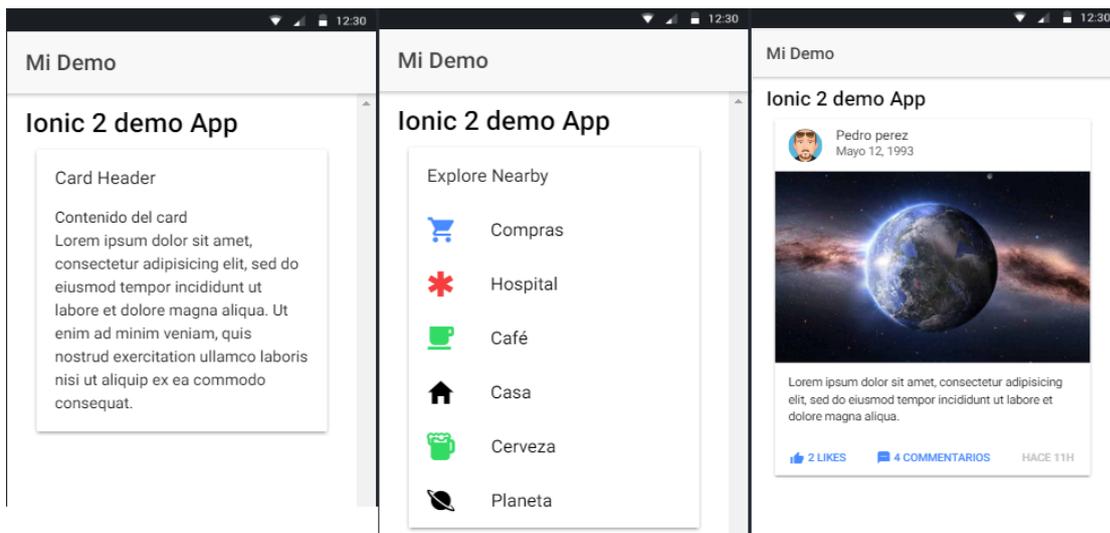


Fig. 32: Ejemplo de Cards en Ionic 2  
Fuente: Propia

### 3.4.7. FAB's (Botones de acción flotante).

FABs (Floating Action Buttons) son componentes estándares de *Material Design*<sup>14</sup>, cuya forma de presentación son botones con apariencia flotante que representan acciones que la aplicación o el dispositivo va a efectuar.

La directiva que se usa para formar un FAB es *ion-fab* misma que puede ser acompañada de atributos de ubicación como son *top*, *right*, *left* y *bottom*.

<sup>14</sup> *Material Design*: Guía integral de diseño, movimientos e interacción del usuario con el dispositivo

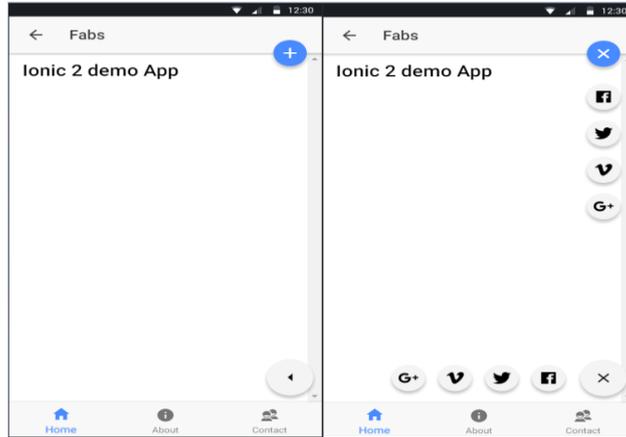


Fig. 33: Ejemplo de FABs en Ionic 2  
Fuente: Propia

### 3.4.8. Grid (tabla).

Las Grid son un sistema para construir layouts o diseños de distintas formas y tamaños y este a su vez utiliza una serie de filas, columnas y contenedores para alinear el contenido de una vista siempre manejado bajo del concepto de *Responsive Web Design*.

La siguiente caja de texto contiene un fragmento de código de la formación GRID.

```

<ion-grid>
  <ion-row>
    <ion-col> 1 de 3 </ion-col>
    <ion-col> 2 de 3 </ion-col>
    <ion-col> 3 de 3 </ion-col>
  </ion-row>
</ion-grid>

```

## 3.5. API

La documentación de Ionic muestra dos tipos de API's, apis de componentes y de servicios.

### 3.5.1. API's de componentes.

Las apis de componentes son aquellas que incluyen entre su nomenclatura de uso clases como Item, Toggle, Icon y Card. Nos indican sus selectores, propiedades y eventos que podemos usar dentro de estas directivas.

Dentro de las API'S de componentes más usadas de Ionic se encuentran:

### 3.5.1.1. *Content y Header.*

Es la directiva *Content* indica el cuerpo del contenido de la interfaz a ser vista por el usuario, su equivalente en html5 es la etiqueta `<body>` la etiqueta que indica su uso es:

```
<ion-content> Soy el contenido!!! </ion-content>
```

Es la directiva *Header* indica el en encabezado de la pantalla que puede ser visualizada por el usuario, juntamente con `<ion-navbar>`, su equivalente en html5 es la etiqueta `<header>`.

```
<ion-header> Página Demo </ion-header>
```

A estos componentes se puede inyectar atributos adicionales para modificar su interfaz y su funcionalidad, como indica la *Fig. 34: Uso de ion-content y ion-header.* en la que se inyecta un atributo *padding* mismo que dará un margen en el contenido del aplicativo.

```
1
2 <ion-header>
3   <ion-navbar>
4     <ion-title>Página Demo</ion-title>
5   </ion-navbar>
6 </ion-header>
7
8 <ion-content padding>
9   <h2> Soy el contenido !! </h2>
10 </ion-content >
11
```

Fig. 34: Uso de ion-content y ion-header  
Fuente: Propia

### 3.5.1.2. *Item.*

De manera general se usa dentro de una lista, juntamente con otros elementos, este componente se puede manipular fácilmente con acciones como deslizar, eliminar, ordenar y reordenar y editar fácilmente, el único requisito o requerimiento es que se encuentre dentro de una lista. Podemos identificar o dar uso de la siguiente manera:

```
<ion-list><ion-item> Soy un Item!!! </ion-item ></ion-list>
```

### 3.5.1.3. Segment.

Un segment o segmento es un grupo de botones mismo que dan la facilidad al usuario de interactuar con controles y/o páginas. Un segmento brinda una funcionalidad igual o parecida en muchos aspectos a una pestaña.

En los segmentos podemos utilizar infinidad de componentes como listas, botones, cards, sliders e iconos, tantos elementos como se requiera para interactuar con el usuario.

A continuación, un ejemplo de un segmento de como seria su implementación en las plataformas IOS y Android.

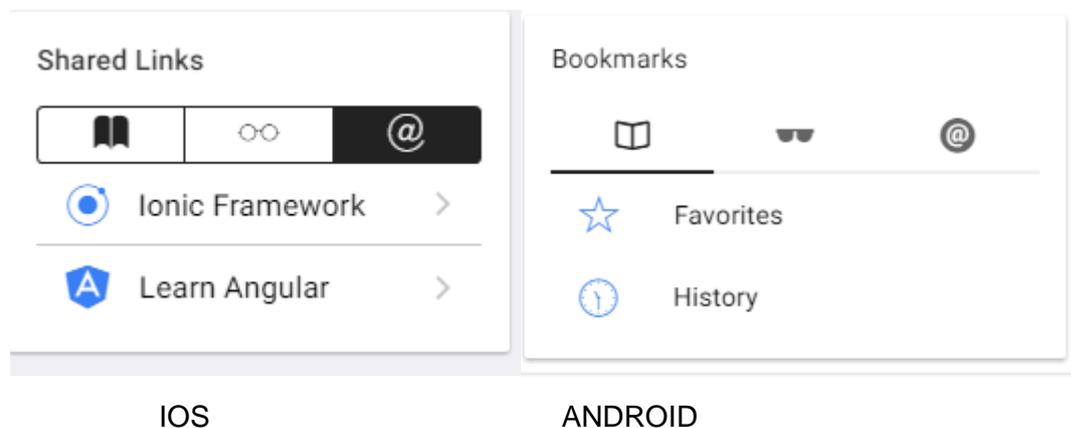


Fig. 35: Segmento  
Fuente: Documentación Oficial

### 3.5.1.4. Input.

Los inputs o entradas de texto son nos ayudan a capturar información, Ionic es compatibles con los elementos HTML tradicionales con la finalidad de facilitar el desarrollo para los programadores y mejorar la experiencia del usuario. La directiva de uso es `<ion-input>` acompañado de atributos tales como *text*, *number* e *email* al igual que el HTML nativo.

Los inputs poseen atributos y eventos que pueden ser inyectados directamente que facilitan su uso y potencian su alcance. Al igual que la mayoría de las componentes de Ionic tienen la facultad de incorporar en su uso eventos de Angular 2.

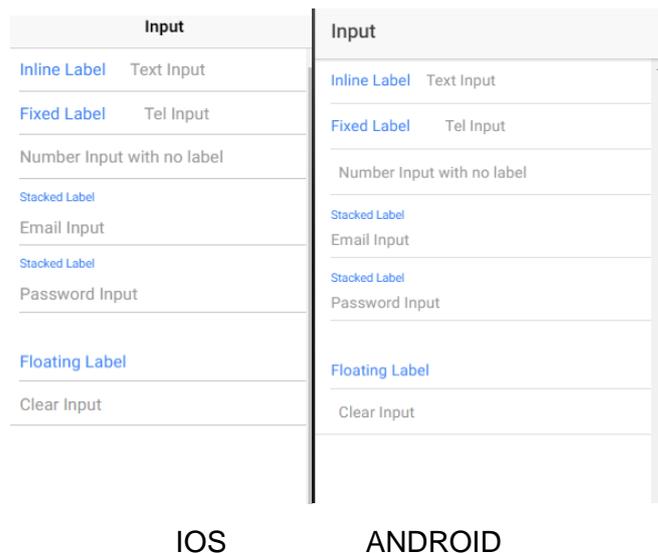


Fig. 36: Inputs  
Fuente: Documentación Oficial

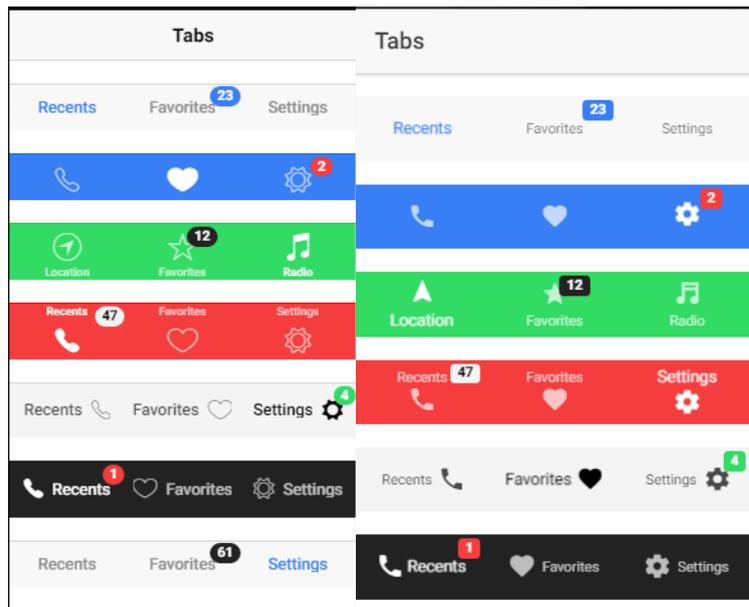
A los ion-input se puede inyectar clases CSS y atributos adicionales que modifican su comportamiento ante el usuario final y su forma de presentación.

### 3.5.2. API's de servicios.

Las API de servicio son clases como *MenuController*, *Config* o *Platform*. Son servicios provistos por el marco de desarrollo de Ionic que pueden ser inyectados directamente las clases. Dentro de las API de servicio más usadas por los desarrolladores se encuentran:

#### 3.5.2.1. Tabs.

Al igual que los segmentos o *segment* las Tabs o pestañas facilitan la navegación entre diferentes páginas. Su nomenclatura `<ion-tabs>` Y pueden abarcar un sinnúmero de componentes individuales, es decir no tienen un límite.



IOS

ANDROID

<https://ionicframework.com/docs/components/#tabs>

Fig. 37: Tabs

Fuente: Documentación Oficial

La ubicación del conjunto de pestañas puede ser modificado sea este en la parte superior o inferior de la pantalla visible al usuario. De forma predeterminada en Windows Phone se sitúa en la parte superior de la pantalla y en la parte superior para las plataformas Android e iOS.

Las API'S de servicios son aquellas que hacen uso clases como *Nav*, *Toolbar*, *Platform* y *Tabs*. Mismos que son provistos por Ionic u que pueden ser usados directamente o inyectados dentro de clases.

### 3.6. Componentes Nativos

Dentro del desarrollo de aplicaciones podemos encontrar inconvenientes esto dependiendo del tipo de aplicación que estemos desarrollando, en muchas ocasiones tenemos que obtener información del dispositivo u otros factores externos como direcciones IP, almacenamiento de datos, notificaciones y mapas. Para estos casos Ionic provee plugins adicionales para poder realizar estas funcionalidades.

Veremos algunos plugins, los más usados por los desarrolladores y los más útiles al momento de desarrollar aplicaciones tales como Device, Google Maps, SQLite y Storage, veremos cómo usarlo y los resultados de su aplicación.

Todos los plugins se deben instalar vía consola o terminal del sistema operativo esto es *CMD*<sup>15</sup> o *Windows Power Shell*<sup>16</sup> en Windows, para Linux y Mac a través de *Terminal*<sup>17</sup> para lo que debemos navegar hasta la carpeta donde se encuentra nuestro aplicativo y escribir el nombre del plugin a ser instalado de la siguiente manera.

1. *ionic cordova plugin add cordova-plugin-device*
2. *npm install --save @ionic-native/device*

Estos comandos instalaran los archivos y dependencias necesarias para que podamos usar el plugin.

### 3.6.1. Device.

Este componente nativo nos permite acceder a la información de nuestro dispositivo, su funcionamiento es adecuado para todas las plataformas de Android, iOS, Windows Phone, MacOS y navegadores.

Dentro del componente ponemos instanciar los siguientes objetos que nos brindaran la información del dispositivo, entre las más importantes se encuentran:

- **Cordova:** Obtiene la versión de Cordova que está corriendo en el dispositivo.
- **Model:** esta instancia nos da a conocer el modelo o producto, este valor puede cambiar para distintos dispositivos un ejemplo de esto es para el Samsung Galaxy S5 tenemos los modelos SM-G900F, SM-G900H, SM-G900M entre otros.
- **Platform:** nos brinda información de en qué sistema operativo está corriendo la aplicación.

---

<sup>15</sup> CMD: Consola de Windows, es un intérprete de comandos.

<sup>16</sup> Windows Power Shell: interprete de comandos similar a CDM

<sup>17</sup> Terminal: interprete de comandos para los sistemas operativo basados en Unix (Linux, Mac)

- **Version:** obtiene la versión del sistema operativo por ejemplo podría ser 5.0 en Android y en iOS 11.

Para su uso debemos realizar la importación del componente e instanciar el objeto en el constructor de la clase como indica la *Fig. 38: Plugin Device*

```
1 import { Device } from '@ionic-native/device';
2
3 constructor(private device: Device) {
4 }
5
6
7 obtenerModeloDispositivo(){
8     console.log('El model es: ' + this.device.model);
9 }
0
1
```

Fig. 38: Plugin Device  
Fuente: Propia

### 3.6.2. Google Maps.

La implementación de mapas es uno de los requerimientos que está presente en muchos de los desarrollos, una de las opciones y la más valorada por los desarrolladores es Google Maps.

Para la implantación de mapas de Google es necesario un *API KEY*<sup>18</sup> estas claves o llaves se las debe crear en [Google Api Console](#). En este caso en la instalación del plugin se debe añadir algunas variables adicionales en este caso el API KEY creado la cual.

```
ionic cordova plugin add cordova-plugin-googlemaps --variable
API_KEY_FOR_ANDROID="ANDROID_API_KEY" --variable
API_KEY_FOR_IOS="IOS_API_KEY"
```

---

<sup>18</sup> API KEY: Clave o llaves para el uso de servicios de Google

## CAPITULO IV:

### 4. DESARROLLO / FASE ELABORACIÓN

#### 4.1. Planificación

##### 4.1.1. Investigación preliminar.

Antes de iniciar la fase de diseño y proceder con el desarrollo del aplicativo, se tomará en cuenta ciertos prerrequisitos para avanzar.

El proyecto es un aplicativo orientado a servicios esto nos dice que debemos realizar un backend que ofrezca servicios RESTful que es Yii2 para ser consumidos por la aplicación móvil es por esto por lo que tenemos que definir prerrequisitos para el proveedor de servicios y para la aplicación móvil a realizarse en Ionic 2.

##### 4.1.2. Prerrequisitos de sistema.

En el presente proyecto se requiere desarrollo en diferentes tecnologías cuyas funcionalidades son completamente diferentes, para lo cual debemos de realizar un análisis diferente.

###### 4.1.2.1. *Backend Yii2.*

- Windows Vista en adelante.
- MacOS X 10.5.8 o superior.
- Linux (Ubuntu, Fedora, Mint).

###### 4.1.2.2. *Aplicativo Móvil.*

- Windows 7 en adelante.
- MacOS X 10.5.8 o superior.
- Linux (Ubuntu, Fedora, Mint).

No existe restricción en sistemas operativos de 32 o 64 bits, en este caso se recomienda el de 64 por su velocidad de procesamiento de datos.

### **4.1.3. Prerrequisitos de hardware.**

Las restricciones de hardware son mínimas en el campo de almacenamiento se recomienda un disco duro de al menos 50 Gb de espacio de almacenamiento ya que el SDK de Android y sus máquinas virtuales suelen usar mucho espacio en disco.

En cuanto a la memoria RAM y procesador se recomienda un mínimo de 4 Gb y un procesador i3 o superior o su equivalente para que el desarrollo sea fluido y no se tenga problemas con el rendimiento del computador.

### **4.1.4. Prerrequisitos de software.**

Los requerimientos de software se detallan a continuación y son necesidades básicas para la configuración y funcionamiento del software que vamos a construir.

#### **4.1.4.1. Backend Yii2.**

Como habíamos mencionado, usaremos Yii2 para la administración del contenido a ser mostrado en el aplicativo móvil y como proveedor de Servicios Web de tipo REST. Para el proceso de desarrollo de esta parte del proyecto usaremos:

##### *4.1.4.1.1. Composer.*

Es un manejador de dependencias para el lenguaje de programación PHP y podemos descargarlo de su página oficial, el instalador está disponible para Windows y sistemas operativos basados en Unix como indica la *Fig. 39: Instalación Composer*.

En nuestro caso usaremos Windows como sistema operativo para desarrollo para lo cual tenemos que:

- Descargar el instalador.
- Ejecutar el instalador.
- Añadir al PATH la ruta de instalación de *composer* (si no se añade durante la instalación).

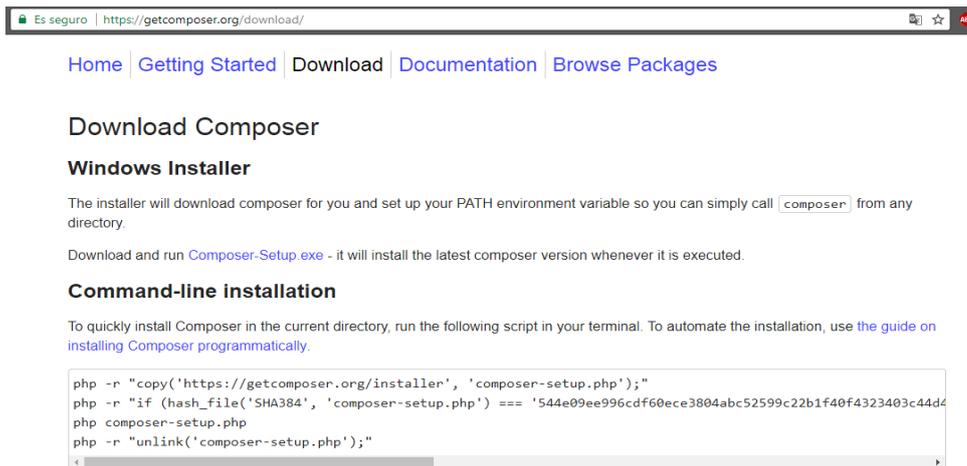


Fig. 39: Instalación Composer  
Fuente: (Composer, 2017)

#### 4.1.4.1.2. Framework Yii2.

La instalación de Yii2 la realizaremos mediante el manejador de dependencias composer para lo cual debemos dirigirnos a la página web oficial del framework misma que se indica la *Fig. 40: Instalación Yii2*. En este caso instalaremos la plantilla avanzada en la que encontramos dos aplicativos backend que es la administración y frontend es el que se presenta al usuario final.

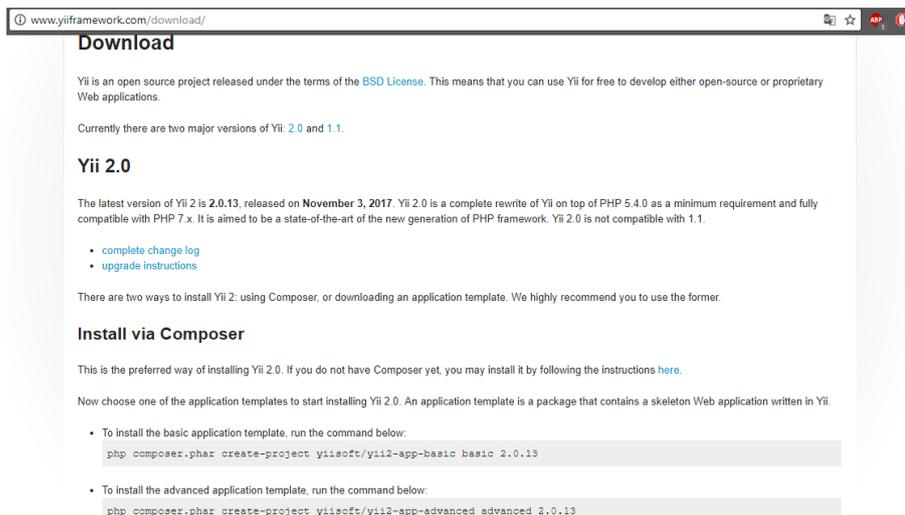


Fig. 40: Instalación Yii2  
Fuente: (LLC Yii, 2014)

En este caso usaremos la plantilla avanzada, esta nos permitirá crear un backend modular y orientado a servicios.

En proceso de instalación consiste en:

- Definir un directorio donde crearemos el aplicativo.
- Abrir la consola de Windows o terminal en otro sistema operativo.
- Escribir o pegar el comando `php composer.phar create-project yiisoft/yii2-app-advanced turismoapp`.
- El proceso anterior creará un directorio donde se instalará Yii2.

#### 4.1.4.1.3. Xampp.

Es necesario un servidor web para poder visualizar los cambios que hemos realizado en nuestro backend Yii2 para lo cual usaremos XAMPP que es una solución basada en Apache, MySQL y PHP. En esta Ocasión utilizaremos la versión 5.6.33 de este software en este caso para la plataforma Windows, también existe distribuciones para Linux y MacOS.



Fig. 41: Instalación XAMPP  
Fuente: (Apache Friends, 2017)

Proceso de instalación del software:

- Descargar XAMPP desde su página web oficial.
- Ejecutar el instalador.
- Seleccionar PHP, MySQL y Apache.
- Finalizar la instalación
- Ejecutar la aplicación.

- Verificar que una vez no iniciado Apache y MySQL se inicien los servicios. De no ser el caso cambiar de puertos.

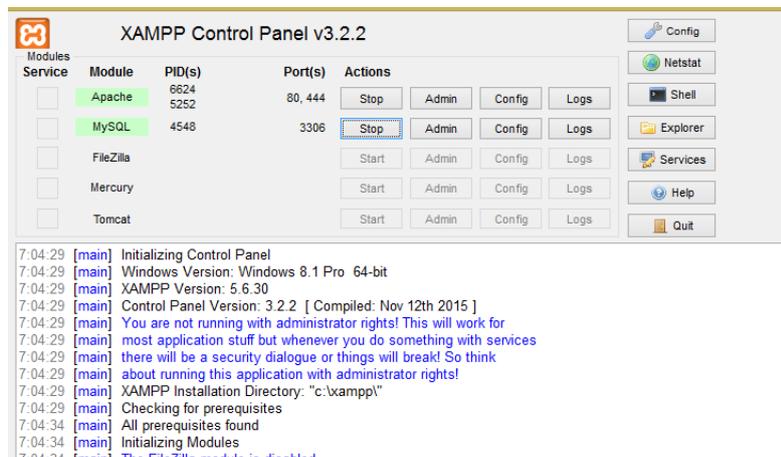


Fig. 42: Panel de control XAMPP  
Fuente: Propia

#### 4.1.4.2. **Aplicativo Móvil.**

##### 4.1.4.2.1. **Node JS.**

Node JS es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el más grande de librerías de código abierto en el mundo. (Clements, 2014)

Node JS contiene el gestor de paquetes NPM, el cual instalará dependencias necesarias y adicionales para nuestro aplicativo a ser desarrollado.



Fig. 43: Instalación de Node JS  
Fuente: Sitio Oficial NODE JS

#### 4.1.4.2.2. SDK Android.

El SDK de Android nos permitirá compilar nuestra aplicación para este sistema operativo. Su instalador lo podemos encontrar en el sitio oficial en este caso descargaremos Android Studio ya que este instalador lleva incluido el SDK de Android.

Una vez instalado el SDK de Android se debe añadir a las variables de entorno, en este caso añadiremos una variable nueva llamada ANDROID\_HOME cuyo valor será el directorio de instalación de Android SDK como se indica en la Fig. 44: Configuración de variables de entorno Android SDK. Luego en la variable PATH añadiremos los accesos a la carpeta *platform-tools* y *tools* que son directorios que se utilizara para poder compilar nuestro aplicativo de la siguiente manera `%JAVA_HOME%;%JAVA_HOME%\platform-tools;%JAVA_HOME%\tools`.

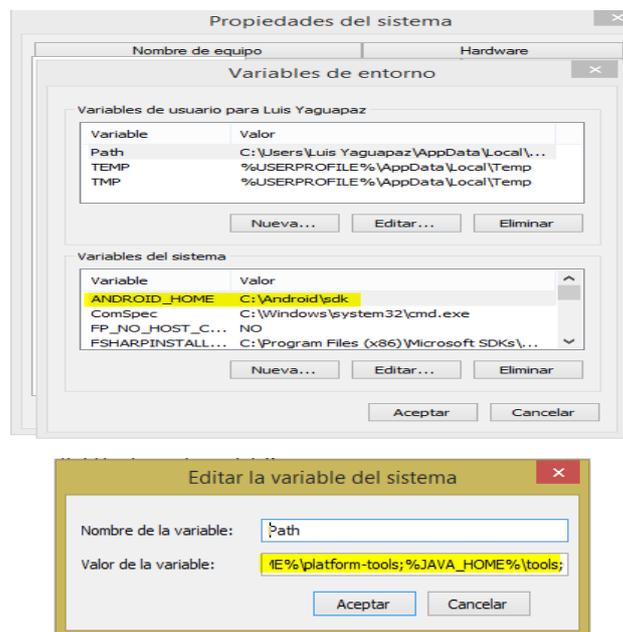


Fig. 44: Configuración de variables de entorno Android SDK  
Fuente: Propia

## 4.2. Especificación de requisitos de software (Product backlog)

### 4.2.1. Propósito.

Está dirigido a las Autoridades, Coordinadores y Directores Departamentales, con el propósito de realizar una descripción detallada de los requisitos que concierne al desarrollo del aplicativo para analizar el funcionamiento y alcance del proyecto a ser realizado, determinando así la funcionalidad que exige el aplicativo móvil, así como las condiciones que se deberá tener en cuenta durante el periodo de desarrollo.

### 4.2.2. Alcance.

El alcance del presente desarrollo es dar a conocer al ciudadano tanto local, nacional e internacional el patrimonio turístico que el cantón Pimampiro ofrece y los lugares turísticos con mayor afluencia para trabajar en el mejoramiento y promoción de estos.

### 4.2.3. Personal Involucrado.

Tabla 5: Descripción del Técnico de Turismo  
Fuente: Propia

Nombre	Lic. Daniel Rivera
Rol	Usuario – Técnico de Turismo - Product Owner
Categoría profesional	Licenciado en Turismo
Responsabilidades	Validar funcionalidades del proyecto Dueño del proceso
Información de contacto	Pimampiro, Tel: 062937117.
Aprobación	SI

Tabla 6: Descripción del Director Administrativo  
Fuente: Propia

Nombre	Ing. Jean Arciniega
Rol	Usuario – Director Administrativo - Product Owner
Categoría profesional	
Responsabilidades	Validar funcionalidades del proyecto Dueño del proceso
Información de contacto	Pimampiro, Tel: 062937117
Aprobación	SI

Tabla 7: Descripción del Director de Sistemas  
Fuente: Propia

Nombre	Nixson Carrera
Rol	Usuario – Director de sistemas
Categoría profesional	
Responsabilidades	Colaborador para subir aplicación al servidor
Información de contacto	Pimampiro, Tel: 062037117
Aprobación	SI

Tabla 8: Descripción del desarrollador  
Fuente: Propia

Nombre	Luis Yaguapaz
Rol	Programador – Scrum Team
Categoría profesional	Estudiante de Ingeniería en Sistemas
Responsabilidades	Validar funcionalidades del proyecto Dueño del proceso
Información de contacto	Pimampiro, Tel: 0989186488
Aprobación	SI

#### 4.2.4. Requerimientos Funcionales.

##### 4.2.4.1. *Requerimientos Del Negocio.*

La aplicación móvil Destino Pimampiro se desarrolló con la finalidad de permitir a los usuarios (ciudadanos locales, nacionales e internacionales) llevar la información de los atractivos turísticos del Cantón Pimampiro de una manera cómoda y de fácil usabilidad, sus funciones son confiables, prácticas para enriquecer el conocimiento de esto sobre lo que el cantón ofrece al turista en lugares naturales, turismo de aventura, eventos y festividades.

El problema que solucionó Destino Pimampiro es la confusión y falta de información en los turistas y ciudadanos en general con respecto a que ofrece Pimampiro en el ámbito turístico y cultural ya que tendrá la información a su alcance de manera rápida, ordenada y actualizada, todo ello para mejorar la afluencia de turistas y fomentar la economía local.

#### 4.2.4.2. **Requerimientos De Usuario.**

##### 4.2.4.2.1. *Necesidades de los Usuarios.*

- **Fácil de manejar:** Todos los formularios de ingreso de información deben tener una forma fácil de usar, con botones, listas desplegables y cuadros de selección que al solo ver sepan la funcionalidad que va a tener de cada uno.
- **Diseño amigable:** Cada formulario debe ser claro e intuitivo, es decir, que va a realizar dicho formulario con sus respectivos botones con iconos referentes a lo que se está realizando.
- **No muy colorido:** Los colores serán puestos mediante un estándar brindados por el diseñador de la institución ya que los formularios y páginas de la aplicación con colores muy fuertes provocan cansancio a la vista, o por otra parte que tenga los colores de la marca Destino Pimampiro o en su defecto de la institución.
- **Al iniciar el sistema:** Mostrará una pantalla principal donde saldrá un mensaje de bienvenida juntamente con el logotipo de la empresa.
- **Mensajes de aceptación:** En el momento de llenar el formulario y enviarlo, debe haber mensajes que notifique si estamos o no seguros de realizar la operación.
- **Mensajes de proceso:** En el momento de ingresar a una página si esta demora en cargarse, debe haber mensajes que notifique sí que se está realizando la operación.

##### 4.2.4.3. **Requerimientos de la aplicación.**

Dentro de los requerimientos del sistema encontraremos requerimiento del aplicativo web administrador de contenido que detallamos a continuación.

Tabla 9: Requisito funcional de perfil de acceso  
Fuente: Propia

Número de requisito	RF.APL.01	
Nombre de requisito	Perfil de Acceso	
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción
Detalle del requisito	A los usuarios dados de alta en el sistema se les asociará un perfil de acceso y se identificará el nivel de actuación de su organización.	

Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
-------------------------	--	--	---

Tabla 10:Requisitos funcionales de acceso de usuarios  
Fuente: Propia

Número de requisito	RF.APL.02		
Nombre de requisito	Límite de acceso de usuarios		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Detalle del requisito	<p>Existirá 2 tipos de usuarios:</p> <p><b>Edición.</b> Además de tener permisos de Lectura, podrá acceder a la aplicación para ejecutar ciertas funcionalidades del sistema como edición y modificación.</p> <p><b>Administración.</b> Además de poseer los permisos de Lectura y ejecución, tendrá acceso a la parte de administración de la información en el sistema.</p> <p><b>Auditoria.</b> Además de poseer los permisos de Lectura, tendrá acceso a la parte de auditoria en la que se detallan modificaciones realizadas.</p>		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

#### 4.2.4.4. Datos de entrada.

Tabla 11: Requisitos funcionales sobre el ingreso de información  
Fuente: Propia

Número de requisito	RF.ENT.01		
Nombre de requisito	Ingreso de la Información		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Detalle del requisito	<p>La base de datos de MySQL debe ser capaz de recibir, validar, cargar y consolidar los datos de sus socios y transacciones registradas.</p> <p>A continuación, se enumeran los datos que inicialmente se recibirán:</p> <p>Datos atractivos nuevos.</p> <p>Datos de usuarios de aplicativo web con permisos de uso del sistema.</p> <p>Datos de categorías de lugares turísticos.</p> <p>Datos e imágenes de lugares turísticos.</p> <p>Datos de eventos a realizarse.</p> <p>Datos de distribución política de la provincia, cantón y parroquia.</p>		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

#### 4.2.5. Requerimientos no funcionales.

##### 4.2.5.1. Arquitectura.

Requisitos necesarios para la implementación del sistema.

Tabla 12: Requisitos de Hardware y Software  
Fuente: Propia

Número de requisito	RF.ARQ.01		
Nombre de requisito	Requisitos de hardware y de software		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Detalle del requisito	Sistema Web de administración de contenido se instalará bajo sistema operativo Linux, sobre un servidor de aplicaciones en la nube, que dispondrá de, un servidor web Apache, MySQL. La aplicación móvil estará disponible en la tienda de Google (Google Play)		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

##### 4.2.5.2. Usabilidad.

Especificaciones destinadas a cubrir la capacidad del software para que éste sea entendido, aprendido, operado y usable para los usuarios tanto del administrador de contenido como de la aplicación móvil.

Tabla 13: Acceso al sistema web de administración de contenido  
Fuente: Propia

Número de requisito	RF.USA.01		
Nombre de requisito	Acceso al sistema web de administración de contenido.		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Detalle del requisito	La aplicación será accesible mediante un enlace web situado en la internet. Mantendrá los estándares de usabilidad que cumpla con la facilidad, seguridad y operabilidad accesible por los usuarios autorizados.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

Tabla 14: Acceso la aplicación móvil  
Fuente: Propia

Número de requisito	RF.USA.02		
Nombre de requisito	Acceso al sistema web de administración de contenido.		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Detalle del requisito	La aplicación será accesible mediante en la tienda de Google esto des Google Play Store. Mantendrá los estándares de usabilidad que cumpla con la facilidad y operabilidad accesible para el usuario final.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

#### 4.2.5.3. Seguridad.

Especificaciones destinadas a cubrir la capacidad del administrador de contenido con la finalidad de proteger información y datos, de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniega el acceso a las personas o sistemas autorizados.

Tabla 15: Requisitos de seguridad para el ingreso al sistema web de administración de contenido.  
Fuente: Propia

Número de requisito	RF.SEG.01		
Nombre de requisito	Seguridad de ingreso al sistema web de administración de contenido.		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Detalle del requisito	Los usuarios para ingresar al sistema web de administración de contenido utilizarán un usuario y una contraseña, que será proporcionados por el administrador.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

Tabla 16: Seguridades de base de datos  
Fuente: Propia

Número de requisito	RF.SEG.02		
Nombre de requisito	Seguridad de base de datos		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Detalle del requisito	La seguridad de la base de datos estará bajo la responsabilidad del Director de Sistemas de la Institución o en su defecto del desarrollador del sistema.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

#### 4.2.5.4. **Mantenibilidad.**

La versión inicial se desarrollará con los requerimientos y recursos e información actual, tanto el aplicativo web de administración de contenido como el aplicativo móvil tienen escalabilidad, se puede añadir nuevos módulos o modificar los actuales o en su defecto modificar la estructura del aplicativo para versiones futuras.

Tabla 17: Especificación de los requisitos de mantenibilidad.  
Fuente: Propia

Número de requisito	RF.MAN.01		
Nombre de requisito	Tipo de mantenimiento		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Detalle del requisito	El mantenimiento del sistema estará bajo la responsabilidad del administrador del sistema de la institución o en su defecto el desarrollador, que llevará a cabo los procesos de actualización y corrección de errores del sistema.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

#### 4.2.5.5. **Portabilidad.**

Se especificará los atributos que el sistema web y aplicativo móvil debe mantener para facilitar su traslado a otras plataformas u entornos.

Tabla 18: Especificaciones de portabilidad  
Fuente: Propia

Número de requisito	RF.POR.01		
Nombre de requisito	Especificaciones de Portabilidad		
Tipo	<input checked="" type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Detalle del requisito	<p>No existe restricción de sistema operativo y se debe tener navegadores web con soporte de PHP, CSS, HTML, JAVASCRIPT, AJAX, VENTANAS EMERGENTES ACTIVADO, Y DESACTIVADO EL USO DE ADBLOCKERS para esa dirección o dominio.</p> <p>El servidor apache debe tener asignado un puerto que no ocupe otra aplicación en el mismo entorno para evitar conflictos.</p> <p>En el caso del aplicativo móvil no se debe volver subir a la tienda previa compilación y cumpliendo los pre-requisitos de la misma.</p>		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

#### 4.2.6. Funcionalidad del Producto.

El sistema que implementarse consta de los siguientes aspectos:

##### 4.2.6.1. Administrador de contenido.

###### 4.2.6.1.1. Módulo de lugares turísticos.

Permitirá el ingreso, modificación y verificación de información sobre servicios, atractivos turísticos y eventos a realizarse dentro del cantón.

###### 4.2.6.1.2. Módulo localidades.

En este módulo se permitirá la modificación, ingreso y verificación de información referente al ordenamiento territorial como parroquias del cantón.

###### 4.2.6.1.3. Módulo respaldo y recuperación.

Permitirá la generación e importación de respaldo de la base de datos que usa el aplicativo web.

#### 4.2.6.1.4. *Modulo administrador de usuarios.*

En este módulo tiene la finalidad de administrar los usuarios, roles y acciones que un usuario puede realizar dentro del aplicativo web.

#### 4.2.6.1.5. *Proveer de Servicios Web REST.*

Encargado de proveer los servicios web de tipo REST a ser consumidos por el aplicativo móvil.

### **4.2.6.2. *Aplicativo móvil.***

#### 4.2.6.2.1. *Patrimonio Turístico.*

Conjunto de páginas o vistas donde se indica el patrimonio turístico del Cantón Pimampiro.

#### 4.2.6.2.2. *Servicios.*

Conjunto de páginas o vistas donde se indican los servicios ofertados en el Cantón Pimampiro sean estos restaurantes, hoteles, hosterías, etc.

#### 4.2.6.2.3. *Interés.*

Conjunto de páginas o vistas donde se indica las opciones de interés al público en el Cantón Pimampiro como son zonas wifi, libros o revistas digitales y eventos.

#### 4.2.6.2.4. *Mapa.*

Página que indica el mapa del cantón Pimampiro.

#### 4.2.6.2.5. *Búsqueda de lugares turísticos o servicios.*

Conjunto de páginas o vistas donde permite buscar los distintos lugares turísticos, servicios y lugares de interés en el Cantón Pimampiro.

### **4.2.7. Características de los usuarios.**

Tipo de usuario	Director de Sistemas
Formación	
Habilidades	Administración de usuarios y contenido de la aplicación
Actividades	Gestionar, supervisar y aprobar los usuarios del aplicativo web de administración de contenido

Tipo de usuario	Técnico de Turismo
Formación	
Habilidades	Editor de Contenido
Actividades	Gestionar, supervisar y aprobar el contenido (texto o imágenes) del aplicativo web de administración de contenido

Tipo de usuario	Usuario común
Formación	
Habilidades	Visualizar Contenido
Actividades	Visualizar contenido en la aplicación móvil.

#### 4.2.8. Planificación del proyecto (Sprint Planning).

El proyecto tiene un tiempo estimado de 600 horas un aproximado de 75 días laborables para su culminación, para lo que se ligará a los Sprints y sus requerimientos. A continuación, se detalla las actividades que se realizaron en cada iteración:

##### 4.2.8.1. *Sprint 1.*

En el Sprint 1 se tubo planificado del diseño e implementación de la base de datos, así como la configuración del aplicativo y la subida al manejador de versiones.

Development:	Luis Yaguapaz
Product Owner	Daniel Rivera
Sprint	1
Fechas	01/02/2017
	15/02/2017
Total horas	53
<b>TAREA</b>	<b>TIEMPO ESTIMADO (Horas)</b>
Adquisición e instalación de herramientas	2
Configuración del framework Yii2	12
Subida da BitBucket del proyecto de Yii2	1
Modelado de Base de Datos	5
Normalización de Base de Datos	4
Puesta a prueba de la base de datos	6
Modificación de la estructura de base de datos	4
Subida da BitBucket de la base de datos.	1
Creación de Módulos en el backend de Yii2	2
Configuración de generadores de CRUD's	3
Pruebas de generadores de CRUD's	2

Socialización Scrum, Retrospectiva Sprint	4
Planificación.	4
Revisión de desarrollos.	3
<b>TOTAL</b>	<b>53</b>

#### 4.2.8.2. *Sprint 2.*

En el Sprint 2 se tubo planificado la creación de CRUD's (Provincias, Cantones y Parroquias) y añadir las vistas adicionales, así como las pruebas de funcionamiento de las mismas.

Development:	Luis Yaguapaz
Product Owner	Daniel Rivera
Sprint	2
Fechas	16/02/2017
	28/02/2017
Total horas	34
<b>TAREA</b>	<b>TIEMPO ESTIMADO (Horas)</b>
Generación y edición de crud de Provincias	2
Generación y edición de crud de Cantones	2
Generación y edición de crud de Parroquias	2
Pruebas de CRUD's de Provincias, Cantones y Parroquias	5
Corrección de Errores en CRUD's de Provincias, Cantones y Parroquias	2
Generación y edición de crud Atractivo	2
Generación y edición de crud de CategoriaLugar	2
Pruebas de CRUD's de Provincias, Cantones y Parroquias	4
Corrección de Errores en CRUD's de Atractivo y CategoriaLugar	2
Socialización Scrum, Retrospectiva Sprint	4
Planificación.	4
Revisión de desarrollos.	3
<b>TOTAL</b>	<b>34</b>

#### 4.2.8.3. *Sprint 3.*

En el Sprint 3 se tubo planificado la creación de CRUD's (Lugares Turisticos,, Eventos), subida de imagenes y añadir las vistas adicionales así como las pruebas de funcionamiento de las mismas. Adicional se creó el módulo para servicios web REST.

Development:	Luis Yaguapaz
Product Owner	Daniel Rivera
Sprint	3
Fechas	01/03/2017
	15/03/2017
Total horas	54
<b>TAREA</b>	<b>TIEMPO ESTIMADO (Horas)</b>
Generación y edición de crud de Lugares	5
Generación y edición de crud de Imágenes	4
Generación y edición de crud de Eventos	3
Pruebas de CRUD's de Lugares, Imágenes y Eventos	5
Corrección de Errores en CRUD's de Lugares, Imágenes y Eventos	4
Generación y edición de crud Visitas	3
Pruebas de CRUD's de Visitas	1
Corrección de Errores en CRUD's Visitas	1
Pruebas del aplicativo en Yii2	6
Corrección de errores y diseño del aplicativo en Yii2	6
Configuración de directorio/modulo para API	5
Socialización Scrum, Retrospectiva Sprint	4
Planificación.	4
Revisión de desarrollos.	3
<b>TOTAL</b>	<b>54</b>

#### 4.2.8.4. *Sprint 4.*

En el Sprint 4 se tubo planificado la configuración del ambiente de trabajo de Ionic 2, un prototipo del aplicativo y la creación de servicios web REST.

Development:	Luis Yaguapaz
Product Owner	Daniel Rivera
Sprint	4
Fechas	15/03/2017
	31/03/2017
Total horas	84
<b>TAREA</b>	<b>TIEMPO ESTIMADO (Horas)</b>
Instalación y configuración de ambiente de trabajo para IONIC 2	5
Creación y edición de páginas Home, Mapa, Buscar e Información (PROTOTIPO)	10

Creación y edición de páginas Atractivos Turísticos, Zonas Seguras, Servicios (PROTOTIPO)	12
Creación y edición de páginas Atractivos Turísticos, Zonas Seguras, Servicios (PROTOTIPO)	5
Creación de Servicios Web REST para el prototipo	12
Pruebas de Servicios Web REST para el prototipo	3
Corrección de errores de Servicios Web REST para el prototipo	3
Consumo de servicios web REST en el prototipo	12
Pruebas de consumo de servicios web REST en el prototipo	3
Corrección de errores en el consumo de servicios web REST en el prototipo	3
Corrección de errores de diseño en el prototipo	5
Socialización Scrum, Retrospectiva Sprint	4
Planificación.	4
Revisión de desarrollos.	3
<b>TOTAL</b>	<b>84</b>

#### 4.2.8.5. *Sprint 5.*

En el Sprint 5 se presentó el prototipo y se corrigió lo solicitado por el Product Owner así como se recibió nuevos requerimientos y se procedió a cumplir con los mismos.

Development:	Luis Yaguapaz
Product Owner	Daniel Rivera
Sprint	4
Fechas	01/04/2017
	15/04/2017
Total horas	51
<b>TAREA</b>	<b>TIEMPO ESTIMADO (Horas)</b>
Cambios en estructura de la APP (Servicios e Interés)	5
Cambios en estructura de la web app (Yii2) para servicios e interés	2
Pruebas de Cambios en la estructura de la APP (Servicios e interés)	2
Corrección de errores en la estructura de la APP (Servicios e interés)	5
Cambios en la Sección About (Sobre la App)	4
Cambios en estructura de la APP (Servicios e Interés)	5
Cambios en estructura de la web app (Yii2) para Eventos	2
Pruebas de Cambios en la estructura de la APP Eventos	2
Corrección de errores en la estructura de la APP Eventos	5
Corrección de errores en el consumo de servicios web REST en el prototipo	3
Corrección de errores de diseño y funcionalidad en el prototipo	5

Socialización Scrum, Retrospectiva Sprint	4
Planificación.	4
Revisión de desarrollos.	3
<b>TOTAL</b>	<b>51</b>

#### 4.2.8.6. *Sprint 6.*

En el Sprint 6 se llevó a cabo el periodo de pruebas y el feedback de las pruebas del aplicativo.

Development:	Luis Yaguapaz
Product Owner	Daniel Rivera
Sprint	4
Fechas	15/04/2017
	30/04/2017
Total horas	38
<b>TAREA</b>	<b>TIEMPO ESTIMADO (Horas)</b>
Periodo de pruebas (Personal Municipio)	15
Aprobación de presupuesto Dpto Finanzas	10
Feedback de pruebas	4
Socialización Scrum, Retrospectiva Sprint	4
Planificación.	2
Revisión de desarrollos.	3
<b>TOTAL</b>	<b>38</b>

#### 4.2.8.7. *Sprint 7.*

En el Sprint 6 se realizó la revisión de ortografía.

Development:	Luis Yaguapaz
Product Owner	Daniel Rivera
Sprint	4
Fechas	01/05/2017
	15/05/2017
Total horas	23
<b>TAREA</b>	<b>TIEMPO ESTIMADO (Horas)</b>
Revisión faltas ortográficas y posibles errores no detectados aplicación web (Yii2)	10

Deploy de la web app (Yii2)	3
Solicitud por parte de GAD Pimampiro de hosting y dominio	4
Socialización Scrum, Retrospectiva Sprint	4
Planificación.	2
Revisión de desarrollos.	2
<b>TOTAL</b>	<b>23</b>

#### 4.2.8.8. *Sprint 8.*

En el Sprint 6 se realizó la subida a producción tanto del administrador de contenido como del aplicativo móvil a la Google Play Store.

Development:	Luis Yaguapaz
Product Owner	Daniel Rivera
Sprint	4
Fechas	15/05/2017
	30/05/2017
Total horas	36
<b>TAREA</b>	<b>TIEMPO ESTIMADO (Horas)</b>
Subida de aplicación web a hosting	4
Configuración de base de datos y permisos en hosting	3
Deploy de app móvil Destino Pimampiro	1
Compra de cuenta en Developers Google Play Store	1
Proceso de firmar y alinear app Destino Pimampiro	2
Subida de aplicación a Google Play Store	2
Presentación de app Destino Pimampiro	3
Difusión de app Destino Pimampiro en medios de comunicación	20
<b>TOTAL</b>	<b>36</b>

### 4.3. Proceso de desarrollo

#### 4.3.1. Arquitectura funcional.

Se ha dividido la arquitectura principal del proyecto en arquitectura del aplicativo móvil y arquitectura del administrador de contenido, ya que cumplen diferentes funciones para el usuario final y para el administrador o editor de contenido del sitio web.

#### 4.3.1.1. **Aplicación Android.**

El dispositivo Android contenedor del aplicativo se conectará servicios REST mediante protocolo HTTP al proveedor de servicios, mismo que realizará una conexión directa a la base de datos como lo indicamos en la *Fig. 45: Arquitectura Aplicativo móvil*

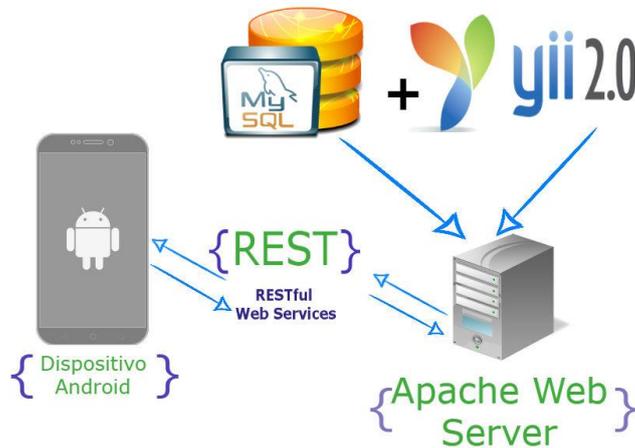


Fig. 45: Arquitectura Aplicativo móvil  
Fuente: Propia

#### 4.3.1.2. **Administrador de contenido YII2.**

El administrador de contenido es un aplicativo web desarrollado en el Framework Yii2 que se encuentra en un servidor Apache, mismo aplicativo almacenará los datos la base de datos MySQL este diagrama se muestra en la *Fig. 46: Arquitectura administrador de contenido.*



Fig. 46: Arquitectura administrador de contenido  
Fuente: Propia

### 4.3.2. Diseño del aplicativo móvil.

El diseño del aplicativo móvil propuesto por el técnico de turismo del GAD Pimampiro, el mismo que no es el definitivo puesto que se prevé cambiar durante el desarrollo se muestra en la Fig. 48: Diseño del administrador de contenido Yii2 se tiene previsto un aproximado de un 10% de cambios al diseño original en el transcurso del desarrollo del aplicativo.

Inicialmente se trabajará en el acoplamiento y desarrollo de la interfaz de usuario final de la aplicación al diseño otorgado por el GAD Pimampiro esto es con datos ficticios con la finalidad de lograr la aprobación de este.



Fig. 47: Diseño inicial del aplicativo móvil  
Fuente: Daniel Rivera, GAD Pimampiro.

### 4.3.3. Diseño del administrador de contenido.

Para el diseño del administrador de contenido se utilizó una plantilla de Bootstrap denominada AdminLTE, esta plantilla está disponible en el sitio web del desarrollador en Github (<https://github.com/dmstr/yii2-adminlte-asset>).

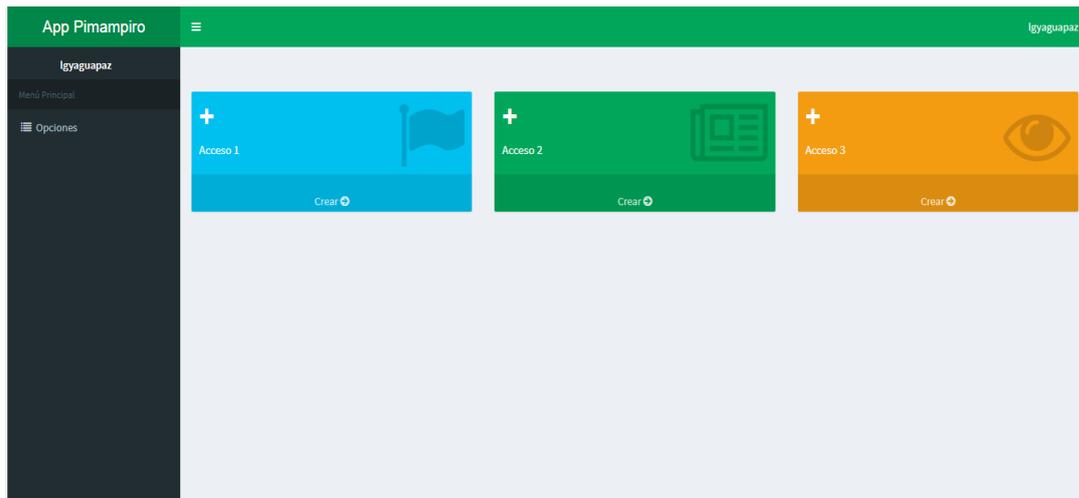


Fig. 48: Diseño del administrador de contenido Yii2  
Fuente: Propia

#### 4.3.4. Diseño de la página web del aplicativo móvil.

Adicional al administrador de contenido se tuvo la necesidad de realizar una página web informativa sobre la aplicación móvil



Fig. 49: Diseño de la página web del aplicativo móvil  
Fuente: Propia

#### 4.3.5. Módulos del administrador de contenido yii2.

Con la finalidad de tener organizado nuestro aplicativo de administración de contenido separaremos las funciones principales en módulos, estos nos ayudarán a segmentar la programación y hacerlo más ordenado.

#### 4.3.5.1. Módulo de contenido.

En este módulo se administrará el contenido a ser presentado el aplicativo móvil, se añadirá fotografías y eventos.

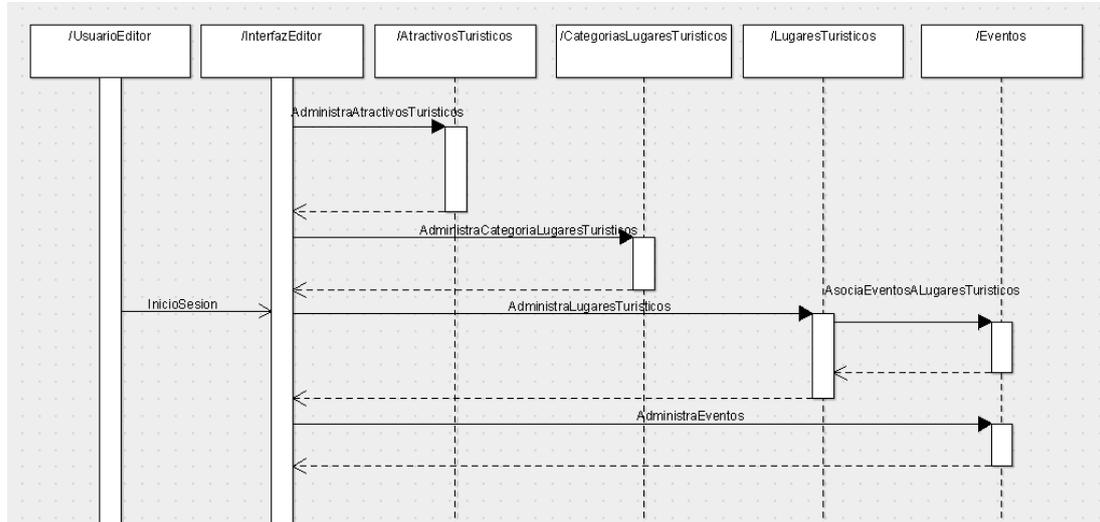


Fig. 50: Diagrama de secuencia módulo Administración de contenido  
Fuente: Propia

#### 4.3.5.2. Módulo de administración de usuarios.

Esta parte del aplicativo web se desarrollará una correcta forma de administración de usuarios basado en roles y acciones para los usuarios.

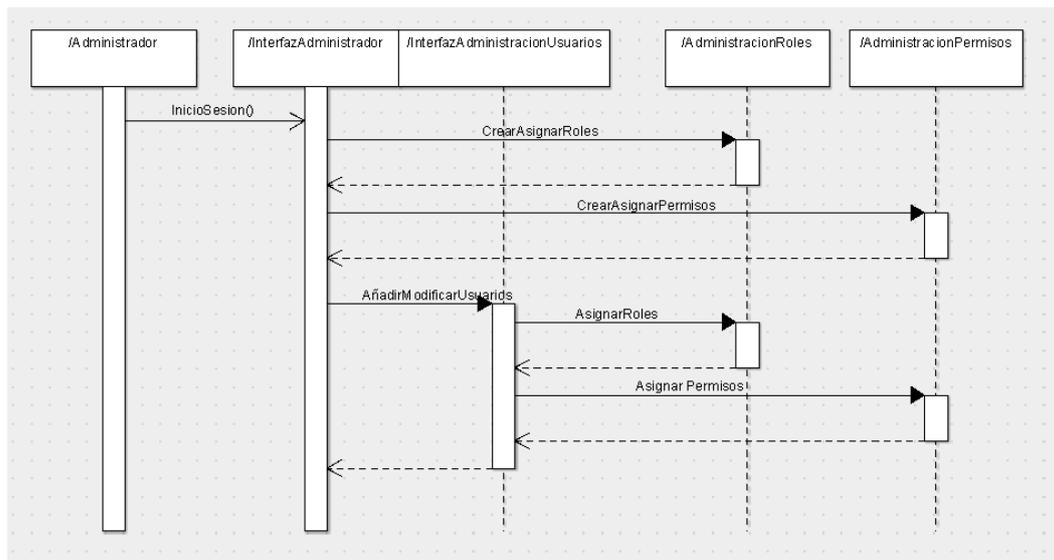


Fig. 51: Diagrama de secuencia módulo de administración de usuarios  
Fuente: Propia

### 4.3.5.3. Módulo de respaldo y recuperación.

El apartado de respaldo y recuperación de información será de utilidad en caso de migraciones, adicional es necesario mantener la información segura a través del tiempo.

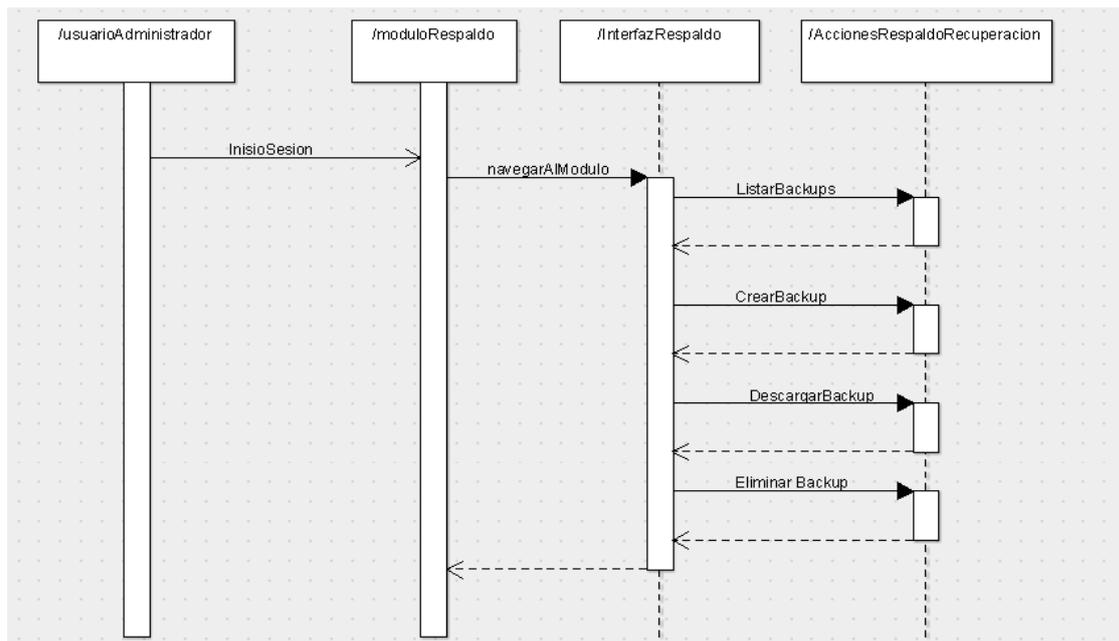


Fig. 52: Diagrama de secuencia módulo de Respaldo y Recuperación  
Fuente: Propia

## 4.4. Desarrollo del administrador de contenido

### 4.4.1. Configuración del entorno de desarrollo yii2.

#### 4.4.1.1. Modelador de Base de Datos.

Para el modelamiento de base de datos y adicional como manejador usaremos Mysql Workbench, se usó la más actual en este caso lo descargaremos de su sitio oficial <https://dev.mysql.com/downloads/workbench>. El proceso de instalación no es complicado se trata de ejecutar el instalador, seleccionar los componentes necesarios.

#### 4.4.1.2. **Servidor web.**

El servidor web es apache en su versión 2.4 en este caso instalaremos XAMPP ya que nos provee una solución muy conveniente y adecuada para el desarrollo, su instalación no es complicada ya que se procederá a instalar el servidor apache y PHP.

El instalador de XAMPP lo encontramos en la página oficial el enlace de descarga <https://www.apachefriends.org/es/download.html>.

#### 4.4.1.3. **Sublime Text.**

Para el desarrollo es necesario un editor de texto, en este caso la solución es Sublime Text en su versión 3 ya que es una herramienta muy poderosa y muy liviano al momento de ejecutar.

Sublime Text 3 tiene la capacidad de instalar extensiones y plugins que nos serán de mucha ayuda durante el desarrollo.

Esta es una herramienta de pago, pero es posible usarla de forma gratuita sin necesidad de licencia alguna, el instalador lo podemos encontrar en la página oficial de la herramienta <https://www.sublimetext.com/3>.

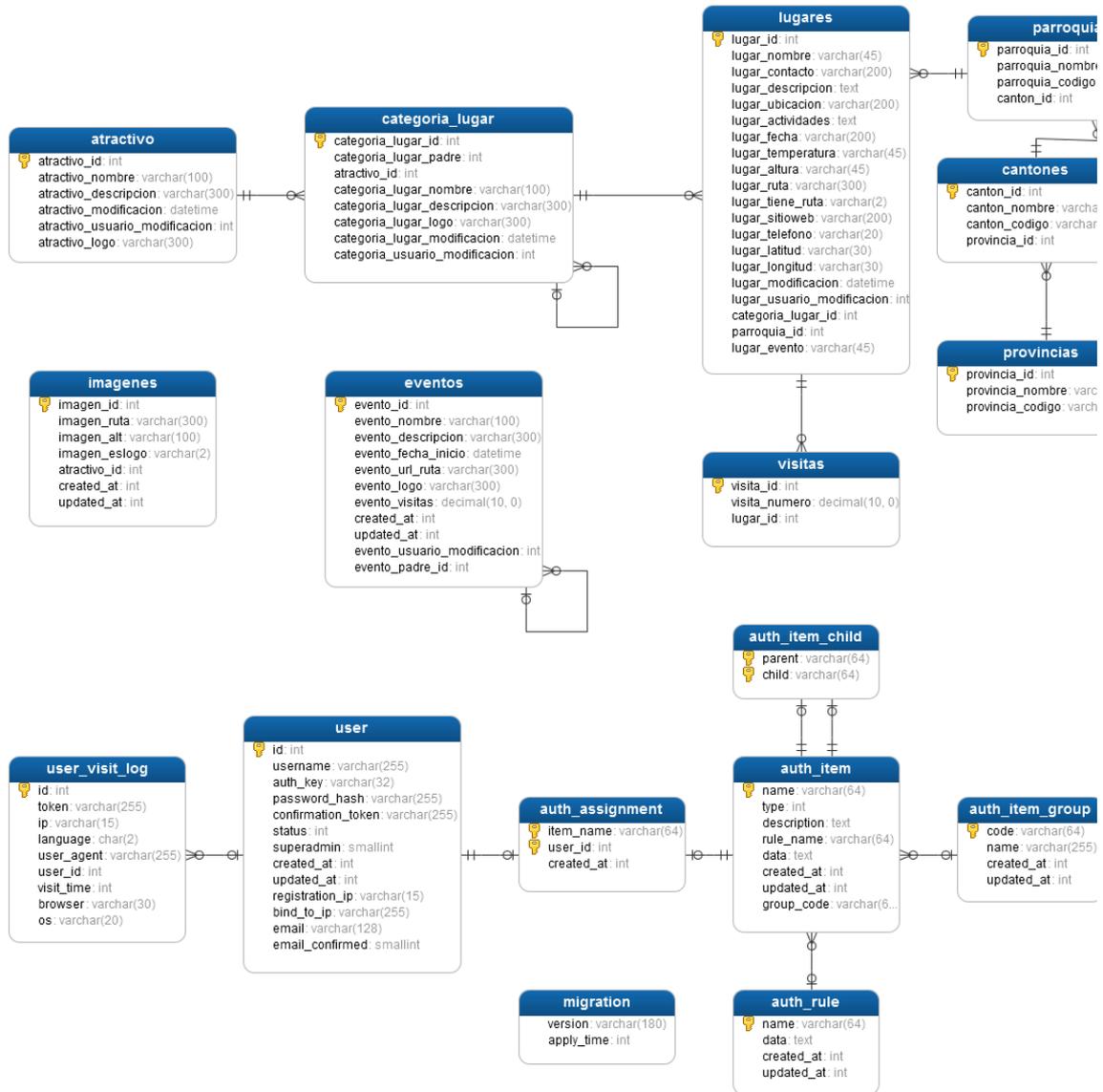
#### 4.4.2. **Base de datos.**

##### 4.4.2.1. **Modelo de base de datos.**

Tabla 19: Descripción de la Base de Datos  
Fuente: Propia

Descripción	Desarrollo de una aplicación móvil para promover los atractivos turísticos del GAD Pimampiro.
Autor	Luis Gonzalo Yaguapaz Madera
Base de datos	Appmovil
Tipo base de datos	MySQL
Versión de base de datos	5.7
Fecha	05/02/2017

#### 4.4.2.2. Diagrama de Entidad – Relación.



Fuente: Propia  
Fig. 53: Diagrama Entidad Relación

#### 4.4.2.3. Diccionario de datos.

Tabla 20: Diccionario de datos  
Fuente: Propia

Name	Primary Contrait	Key	Numero de columnas	Comentario
atractivo	atractivo_id		6	Tabla de almacenamiento de los atractivos turísticos
cantones	canton_id		4	Tabla de almacenamiento de cantones según la distribución política del país
categoria_lugar	categoria_lugar_id		8	Tabla de almacenamiento de las categorías y subcategorías de los lugares turísticos
eventos	evento_id		11	Tabla de almacenamiento de eventos del cantón
imagenes	imagen_id		7	Tabla de almacenamiento de las imágenes de los lugares turísticos
lugares	lugar_id		20	Tabla de almacenamiento de los lugares turísticos
parroquias	parroquia_id		4	Tabla de almacenamiento de parroquias según la distribución política del país
provincias	provincia_id		3	Tabla de almacenamiento de parroquias según la distribución política del país
migration			2	Tabla de almacenamiento de migraciones que se realizaron en el aplicativo
user	id		14	Tabla de almacenamiento de usuarios de la aplicación
user_visit_log	id		9	Tabla de almacenamiento del log de inicio de sesión de los usuarios.
visitas	visita_id		3	Tabla de almacenamiento de visitas realizadas a los lugares turísticos

auth_assignment	Ítem_name	3	Tabla de almacenamiento de roles asignados a usuarios
auth_item	name	8	Tabla de almacenamiento de las rutas o permisos a ser asignados
auth_item_child	parent, child	2	Tabla de almacenamiento de los permisos asignados a los roles
auth_item_group	code	4	Tabla de almacenamiento de los grupos de usuarios
auth_rule	name	4	Tabla de almacenamiento de reglas o acciones permitidas por los roles.

#### 4.4.3. Codificación del Administrador de Contenido y API.

##### 4.4.3.1. Configuración del Aplicativo.

Para el desarrollo y codificación de nuestro aplicativo usaremos un dominio local para lo que se tiene que configurar un host virtual en nuestro ordenador para lo cual se debe modificar el archivo host de nuestro sistema operativo mismo archivo se encuentra en la ruta “C:\Windows\System32\drivers\etc” tal como nos indica la *Fig. 54: Configuración del archivo hosts*.

```

# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
#       127.0.0.1       localhost
#       127.0.0.1       destinopimampiro.local

```

Fig. 54: Configuración del archivo hosts  
Fuente: Propia

Posterior a esto nos dirigimos al directorio de instalación de XAMPP y navegaremos por este directorio por las carpetas “apache\conf\extra” y modificaremos el archivo *httpd-vhosts*

y configuramos el archivo de la forma que indica la *Fig. 55: Configuración del archivo httpd-vhosts*.



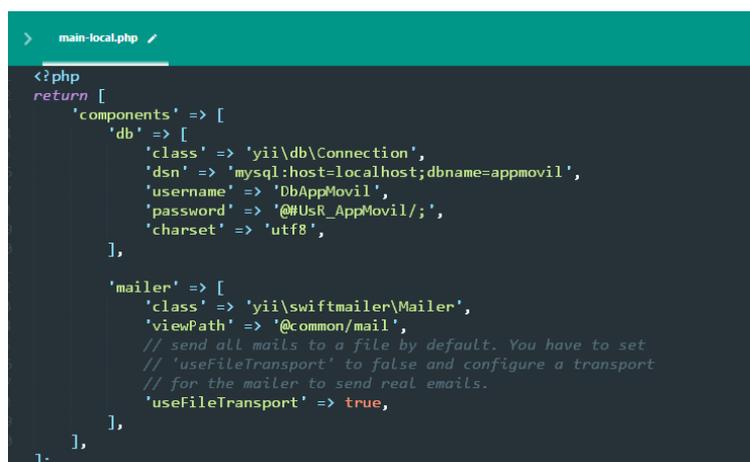
```
httpd-vhosts: Bloc de notas
Archivo Edición Formato Ver Ayuda

NameVirtualHost *:80
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs"
    ServerName localhost
    <Directory "C:/xampp/htdocs">
        AllowOverride All
    </Directory>
</VirtualHost>
<VirtualHost *:80>
    DocumentRoot "C:/xampp/htdocs/turismoapp"
    ServerName destinopimampiro.local
    <Directory "C:/xampp/htdocs/turismoapp">
        AllowOverride All
    </Directory>
</VirtualHost>
```

Fig. 55: Configuración del archivo httpd-vhosts  
Fuente: Propia

Ahora se procede con la configuración del proyecto en Yii2 para lo cual abrimos el símbolo del sistema (CMD) y nos dirigimos a la ruta *htdocs* del directorio de instalación de XAMPP y colocamos *init* este comando inicializa nuestro aplicativo en modo desarrollo o producción y crea archivos de configuración de base de datos.

Una vez inicializado el aplicativo se debe configurar la base de datos a la que accede el aplicativo para lo cual configuramos el archivo *main-local* que se encuentra en la ruta "common/config" de nuestro aplicativo como se puede apreciar en la *Fig. 56: Configuración de la base de datos* .



```
main-local.php
<?php
return [
    'components' => [
        'db' => [
            'class' => 'yii\db\Connection',
            'dsn' => 'mysql:host=localhost;dbname=appmovil',
            'username' => 'DbAppMovil',
            'password' => '@#UsR_AppMovil;',
            'charset' => 'utf8',
        ],
        'mailer' => [
            'class' => 'yii\swiftmailer\Mailer',
            'viewPath' => '@common/mail',
            // send all mails to a file by default. You have to set
            // 'useFileTransport' to false and configure a transport
            // for the mailer to send real emails.
            'useFileTransport' => true,
        ],
    ],
];
```

Fig. 56: Configuración de la base de datos  
Fuente: Propia

Yii2 trae en su core un sistema de administración de usuarios básico, en este caso no lo usaremos. Procedemos con la instalación de una extensión de terceros para la administración de los usuarios en este caso es la extensión user-management del desarrollador [Webvimark](#) que se encuentra alojada en *GitHub* en la misma podemos encontrar las instrucciones de configuración.

Para la instalación de la extensión en la línea de comandos se debe escribir la instrucción `composer require webvimark/module-user-management`.

#### 4.4.3.2. Creación de módulos.

Para la creación de módulos haremos uso del módulo generador de código que nuestro framework en este caso es GII, para lo que entraremos a la ruta `/backend/web/gii`.

Inicialmente crearemos el módulo de localidades en *la Fig. 57: Creación de módulos en GII* podemos apreciar la forma de proceder para generarlo, una vez generado se debe añadir el módulo creado a la sección de módulos (*modules*) en el archivo `main.php` ubicado en la carpeta `backend/config`. Se debe realizar acciones similares para los módulos de atractivo turístico y respaldo.

**Module Generator**

This generator helps you to generate the skeleton code needed by a Yii module.

**Module Class**

**Module ID**

**Code Template**

default (C:\xampp\htdocs\yii2\vendor\yiisoft\yii2-gii\generators\module\default)

Click on the above **Generate** button to generate the files selected below:

Create  Unchanged  Overwrite

Code File	Action	<input checked="" type="checkbox"/>
modules\localidades\Module.php	create	<input checked="" type="checkbox"/>
modules\localidades\controllers\DefaultController.php	create	<input checked="" type="checkbox"/>
modules\localidades\views\default\index.php	create	<input checked="" type="checkbox"/>

Fig. 57: Creación de módulos en GII  
Fuente: Propia

#### 4.4.3.3. *Elaboración de CRUD's.*

La elaboración de las páginas de creación, modificación y eliminación se puede realizar de una manera fácil, dinámica y con las validaciones respectivas campos a ser ingresados.

En primera instancia se deberá crear un modelo en la sección de Model Generator para generar un modelo con atributos y validaciones necesarias para poder administrar los registros de la tabla de la base de datos.

Posterior a la creación de los modelos respectivos modelos, se deberá crear los controladores y vistas en el apartado CRUD Generator. Adicional a la creación de vistas y controladores, Yii2 nos da la posibilidad de crear una clase o modelo de búsqueda avanzada denominada SearchModel.

Al finar la generación en el caso de *Eventos* se obtiene los siguiente ficheros o archivos:

#### **Modelos**

- EventoModel. – Clase de acceso a datos, se encuentran declaraciones de atributo y etiquetas de atributos. Aquí se ubica la lógica del negocio.
- EventoSearch. - Métodos de búsqueda avanzada.

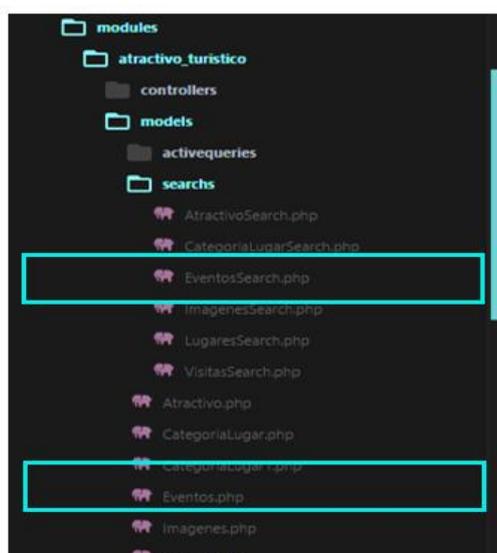


Fig. 58: Modelo generado por CRUD  
Fuente: Propia

## Controlador

- EventoController.- Encargados de procesar las solicitudes, procesar y generar las respuestas mismas que serán inyectadas en las vistas.

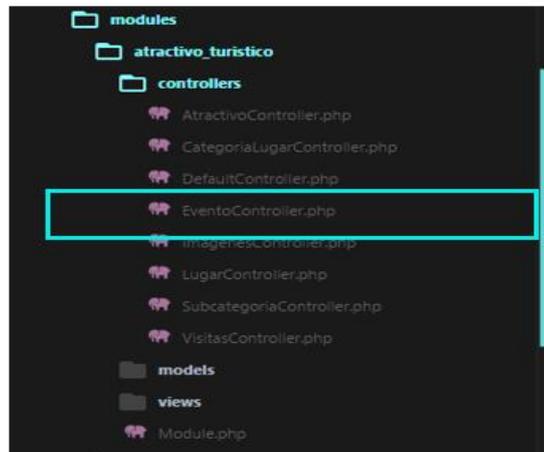


Fig. 59: Controlador generado por CRUD  
Fuente: Propia

## Vistas

- Index. - Archivo o directorio que lista los registros del modelo en una tabla (GRID) misma que permite realizar los filtros requeridos por el usuario.
- Update. – Vista que permite al usuario modificar registros del modelo.
- Create. – Permite crear un registro e insertarlo en la base de datos con las validaciones especificadas en el modelo.
- View. – Presenta una visualización del registro creado o modificado.

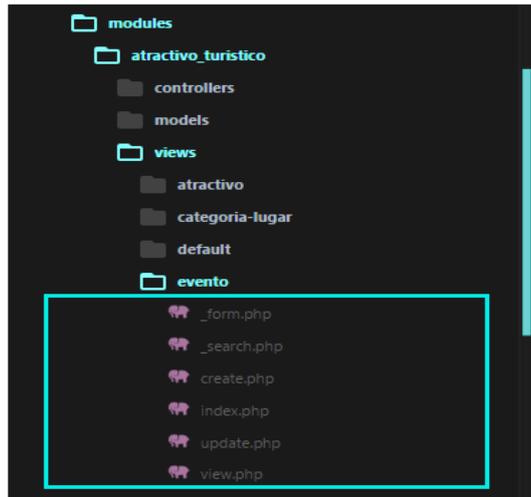


Fig. 60: Vistas generadas por CRUD Generator  
Fuente: Propia.

Se deberá modificar las vistas los CRUD's generados con anterioridad en nuestro aplicativo en cuanto sea necesario para que sea agradable a la vista de del usuario final, los modelos deben aplicar las validaciones necesarias para que sean ingresados datos que no sean de utilidad a nuestro aplicativo.

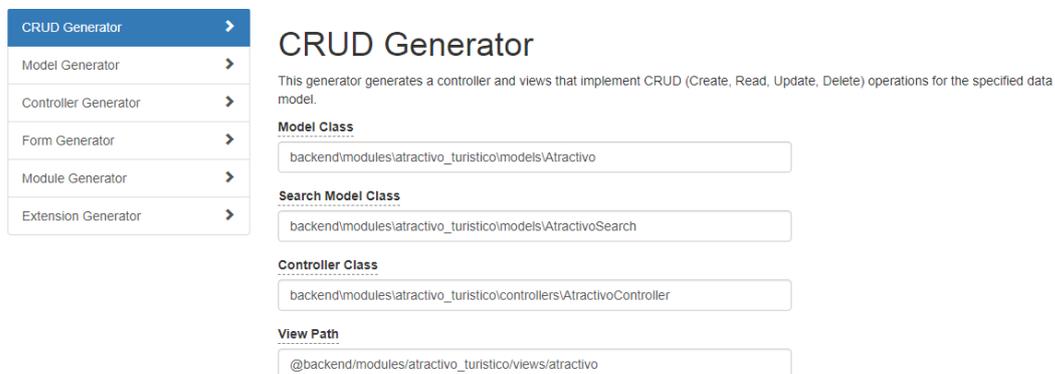


Fig. 61: Elaboración de CRUD en Yii2  
Fuente: Propia

Para las validaciones en las clases modelos existe una función denominada *rules* en lo que se especifica el tipo de datos y validaciones adicionales.

```
[[ 'atractivo_nombre', 'required' ],
```

```
[[ 'atractivo_nombre', 'string', 'max' => 100, 'message' => 'Maximo de 100 caracteres' ],
```

#### 4.4.3.4. Creación de API

Para la creación del API se deberá crear un directorio en la raíz nuestro aplicativo, en las que se creará directorios *config*, *modules*, *runtime* y *web*. Similar a los directorios *backend* y *frontend* de proyecto.

En el archivo *aliases.php* ubicado en el archivo *common/config* se deberá añadir la siguiente línea de código “`Yii::setAlias('api', dirname(dirname(__DIR__)). '/api');`”. En la

Fig. 62: Archivo de Configuración de API en la que se configura las rutas de acceso a los métodos que serán consumidos por nuestra aplicación Android.

```

<?php
$params = array_merge(
    require(__DIR__ . '/../common/config/params.php'),
    require(__DIR__ . '/params.php')
);

return [
    'id' => 'app-api',
    'basePath' => dirname(__DIR__),
    'bootstrap' => ['log'],
    'timezone' => 'America/Lima',
    'modules' => [
        'v1' => [
            'basePath' => '@app/modules/v1',
            'class' => 'api\modules\v1\Module'
        ],
    ],
    'components' => [
        'request' => [
            'parsers' => [
                'application/json' => 'yii\web\JsonParser',
            ]
        ],
        'urlManager' => [
            'enablePrettyUrl' => true,
            'enableStrictParsing' => true,
            'showScriptName' => false,
            'rules' => [
                [
                    'pluralize' => false,
                    'class' => 'yii\rest\UrlRule',
                    'controller' => ['v1/lugar'],
                    'tokens' => [
                        '{id}' => '<id:\w+>'
                    ],
                    'extraPatterns' => [
                        'GET findLugar' => 'getLugar',
                        'GET findLugarCategoria' => 'getLugar_categoria',
                        'GET findInfoGeo' => 'getinformacion_geografica',
                        'GET findImagenesByAtractivo' => 'getimagenesbyatractivo',
                        'POST addVisita' => 'postcontadorLugar',
                        'GET search' => 'searchatractivo',
                    ],
                ],
            ],
        ],
    ],
    'user' => [
        'identityClass' => 'common\models\User',
        'enableAutoLogin' => false,
    ],
    'log' => [
        'traceLevel' => YII_DEBUG ? 3 : 0,
        'targets' => [
            [
                'class' => 'yii\log\FileTarget',
                'levels' => ['error', 'warning'],
            ],
        ],
    ],
],
'params' => $params,
];

```

Fig. 62: Archivo de Configuración de API  
Fuente: Propia

Dentro del archivo se encuentran el módulo v1 que es el cual se crearan los controladores y clases necesarias para brindar servicios la primera versión de nuestra aplicación.

En los métodos de los controladores se hará uso de métodos propios del framework como *find()*, *findOne()*, *findWhere()*, etc. ya por que tardan más tiempo en ejecución en este caso haremos uso de sentencias SQL haciendo uso de conexión directa a base de datos.

#### **4.4.3.5. Creación de Aplicativo Móvil**

#### **4.4.4. Codificación de la Aplicación Móvil.**

Para el aplicativo móvil haremos uso de la plantilla Tabs propia del framework misma que nos brinda navegación entre páginas en el inferior de la pantalla del dispositivo. Para la instalación de esta plantilla es necesario *ionic start destinopimampiro tabs*.

Durante el desarrollo se usará del IDE de programación Visual Studio Code que es una herramienta libre proporcionada por Microsoft, se recomienda usarla ya que incorpora interfaz de línea de comandos misma que es de utilidad para ejecución del aplicativo.

Se debe tomar en cuenta los comandos que podemos ejecutar en la línea de comandos junto con sus variables las cuales son:

Tabla 21: Definición de comandos usados por Ionic 2  
Fuente: Propia

Comando	Descripción
<i>ionic serve</i>	Permite emular en el navegador el aplicativo, se usa durante el desarrollo de la aplicación. Se puede añadir <i>-android</i> , <i>-ios</i> , <i>etc.</i> con la finalidad de emular una plataforma solicitada.
<i>ionic generate page miPagina</i>	Permite crear una página, este comando crea un directorio con el nombre de la página con sus componentes (html, scss y ts)
<i>ionic generate pipe formatoFecha</i>	Permite crear un pipe, este comando crea un directorio con el nombre del pipe con su componente (ts)
<i>ionic generate provider sesión</i>	Permite crear un provider, este comando crea un directorio con el nombre del provider con su componente (ts)
<i>ionic generate component pantallaBloqueo</i>	Permite crear un componente, este comando crea directorio con el nombre de la página con sus componentes (html, scss y ts)
<i>ionic cordova run android</i> <i>ionic cordova build android -prod --release</i>	Permite ejecutar o compilar la aplicación en un dispositivo físico o emulador. Se puede añadir <i>-prod</i> en el caso de requerirla para producción. Si se desea subir a la tienda Google Play Store se deberá añadir la bandera <i>-release</i> .

#### 4.4.4.1. Maquetación

En el proceso de maquetación se procederá con la estructura del aplicativo, en el caso se tomará el prototipo realizado por el diseñador gráfico del GAD Pimampiro y será acoplado al aplicativo a ser desarrollado.

La presentación se presentará al usuario en forma de árbol que partirá desde los él patrimonios turísticos, categorías de lugares turísticos, lugares turísticos y detalles de lugares turísticos. Adicional se presentará un mapa y la sección de contacto.

#### 4.4.4.2. Consumo de servicios

El consumo de servicios REST publicados por el administrador de contenido se lo realizará mediante un *provider* que es un componente del framework que hace uso de paquetes que especializados el protocolo HTTP y de mapeo de datos.

La petición de datos parte del aplicativo mediante un conjunto de peticiones que indican una acción a realizarse, sea esta de lectura o escritura en el servicio web. El aplicativo realiza una petición de tipo OPTIONS el cual describe la acción a realizarse en la petición esta puede ser únicamente GET y POST.

```
getAtractivos() {
  return this._http.get(this.apiUrl + 'atractivo/findAtractivo')
    .map(res => res.json());
}
```

Fig. 63: Consumo del ENDPOINT (API)

Fuente: Propia

Una vez realizada la petición el aplicativo procede a realizar un mapeo de datos y mediante una suscripción al paquete de respuesta puede asignar los datos devueltos a una variable dentro de una determinada página para que: si contiene un valor sea mostrado directamente en la vista o si tiene más de un elemento sea iterado y/o procesado para posterior ser presentado al usuario.

```
getlugares() {
  this._Service.getlugares(this.categoriaId).subscribe( // nos suscribimos al evento
    result => {
      this.items = result; // Asigna a una variable el resultado

      if (result.code == "ERROR_NO_DATA_FOUND") {
        this.items = null; /// si no existe datos
        this.presentarAlerta('Ups!', 'No Existen eventos por el momento!');
      } else {
        this.itera = true; /// si existe datos, permite mostrar en la vista
      }
    },
    error => {
      this.errorMessage = <any>error;
      if (this.errorMessage != null) {
        this.presentarAlerta('Sin Conexión !', 'Al parecer no tienes internet!');
      }
    }
  );
}
```

Fig. 64: Resolver una petición HTTP

Fuente: Propia

#### 4.4.4.3. **Resultados del desarrollo**

Una vez concluido la creación de las paginas basada en los diseños, la implementación de mapas, el consumo de web services y la aplicación de estos en las vistas. Procedemos a la incorporación del plugin launch navigator mismo que nos permitirá abrir aplicaciones de mapas como Google Maps, Waze, Maps.ME, HERE Maps entre otros, mismo plugin lo descargaremos de su página oficial de Github <https://github.com/dpa99c/phonegap-launch-navigator>.



Fig. 65: Página de inicio del aplicativo  
Fuente: Propia

En Fig. 65: *Página de inicio del aplicativo* se puede evidenciar la pantalla principal del aplicativo y en la parte inferior el menú de navegación de tipo Tabs mismo que contiene la página principal, pagina de búsqueda, mapa y sección de contacto.



Fig. 66: Patrimonio Turístico  
Fuente: Propia

En la *Fig. 66: Patrimonio Turístico* se visualiza la pantalla correspondiente a patrimonio turístico del Cantón Pimampiro, sección que puede ser modificada en el administrador de contenido.



Figura 67: Detalle de lugar turístico  
Fuente: Propia

La Figura 67: Detalle de lugar turístico da a conocer detalles de ubicaciones, actividades, teléfonos, etc. del lugar turístico, adicional al pulsar el botón ir traza una ruta desde nuestro punto actual hasta el lugar turístico seleccionado.



Fig. 68: Servicios  
Fuente: Propia

En la *Fig. 68: Servicios* evidenciamos los distintos servicios que esperan al turista en el Cantón Pimampiro, las distintas categorías de los servicios pueden ser administradas en el administrador de contenido, de la misma forma añadir las subcategorías necesarias.

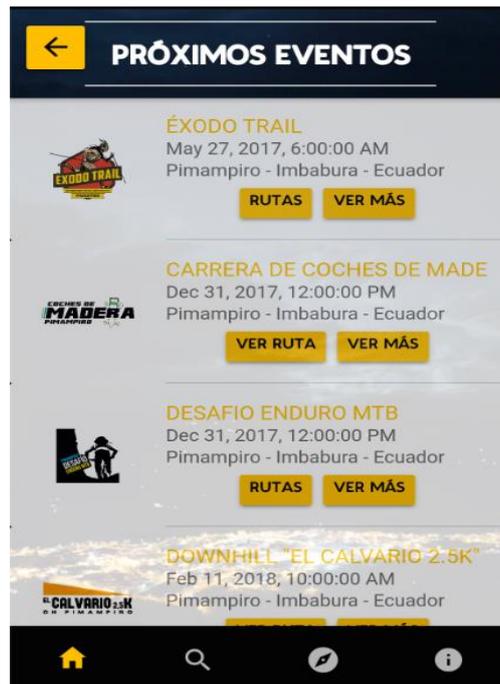


Fig. 69: Lista de eventos  
Fuente: Propia

En la Fig. 69: Lista de eventos se puede evidenciar los eventos a realizarse en un futuro con los detalles de sus rutas, al pulsar dentro de cada ruta la mismas muestra un mapa con los puntos de partida y llegada.

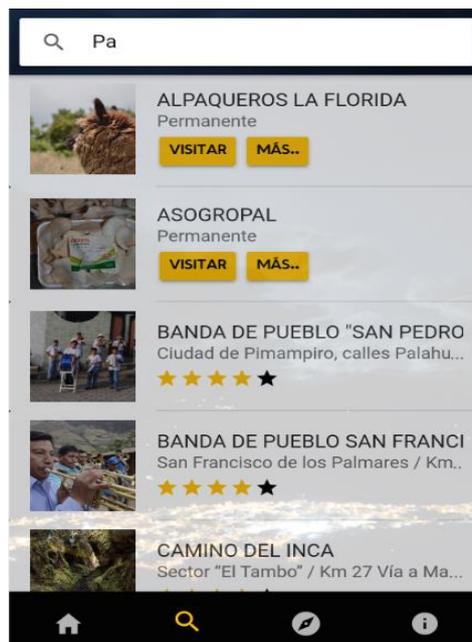


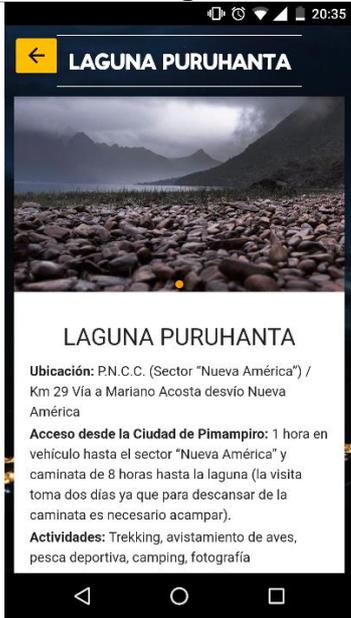
Fig. 70: Pantalla de búsqueda  
Fuente: Propia

En la Fig. 70: Pantalla de búsqueda se evidencia un cuadro de búsqueda en el cual se deberá ingresar un texto y nos dará a conocer una lista con los atractivos turísticos que coinciden con el texto ingresado.

#### 4.5. Pruebas

Para realizar las pruebas de la aplicación Destino Pimampiro se utilizó diferentes dispositivos de diferentes marcas como:

Tabla 22: Pruebas en dispositivos  
Fuente: Propia

Características del Dispositivo	Imagen
<p>Motorola Moto X2 Android 5.0</p>	 <p>The screenshot shows a mobile application interface for 'LAGUNA PURUHANTA'. At the top, there is a back arrow and the title 'LAGUNA PURUHANTA'. Below the title is a landscape photograph of a rocky shore with mountains in the background. Underneath the photo, the title 'LAGUNA PURUHANTA' is repeated. The text below provides location information: 'Ubicación: PN.C.C. (Sector "Nueva América") / Km 29 Vía a Mariano Acosta desvío Nueva América'. It also describes access: 'Acceso desde la Ciudad de Pimampiro: 1 hora en vehículo hasta el sector "Nueva América" y caminata de 8 horas hasta la laguna (la visita toma dos días ya que para descansar de la caminata es necesario acampar)'. Activities listed are: 'Actividades: Trekking, avistamiento de aves, pesca deportiva, camping, fotografía'. The Android navigation bar is visible at the bottom.</p>
<p>Samsung Galaxy S5 Android 6.0.1</p>	 <p>The screenshot shows a mobile application interface for 'CARNAVAL DEL SOL'. At the top, there is a back arrow and the title 'CARNAVAL DEL SOL'. Below the title is a photograph of a large crowd of people at a festival. Underneath the photo, the title 'CARNAVAL DEL SOL' is repeated. The text below provides location information: 'Ubicación: Ciudad de Pimampiro, Polideportivo Municipal'. Activities listed are: 'Actividades: Festival musical'. It also states: 'Abierto: Domingo antes de miércoles de ceniza' and 'Altura: 2200 msnm'. Temperature is listed as 'Temperatura: 19°C'. At the bottom, there is a logo for 'GAD Municipal de Pimampiro' and two yellow buttons labeled 'IR' and 'VER MÁS'. The Android navigation bar is visible at the bottom.</p>

Samsung Galaxy J5  
Android 7.0



# CAPITULO V

## 5. CONCLUSIONES Y RECOMENDACIONES

### 5.1. Conclusiones

Una vez analizado los objetivos y el alcance planteados en este proyecto de titulación se han llegado a las conclusiones siguientes:

- Las nuevas tecnologías, nuevos lenguajes de programación y las tendencias de desarrollo como los es Ionic 2 ayudan a los programadores y dueños del producto a ahorrar costos y tiempo significativamente en este caso generar aplicaciones para diferentes plataformas partiendo de un solo proyecto de desarrollo que se basa en un único lenguaje de programación.
- El estudio de una herramienta debe ser lo más minucioso posible ya que brinda información y ayuda a aprovechar al máximo las bondades de un framework o lenguaje de programación, incluso nos brinda una breve visión de si el framework tiene estabilidad a futuro.
- Las metodologías de desarrollo de software como Scrum tienen como objetivo agilizar la generación de productos ya que se trata de una metodología de desarrollo rápido y garantiza tener un producto a medida que el proyecto de desarrollo se realiza.
- Las nuevas tecnologías nos dan nuevos métodos de conectividad entre aplicaciones, esto es los servicios RESTful, son servicios web completos, livianos y de fácil implementación ya que usan JSON lo que hace que los datos sean multidimensionales y de fácil manipulación.

## 5.2. Recomendaciones

Una vez analizado los objetivos y el alcance planteados en este proyecto de titulación se han llegado a las recomendaciones siguientes:

- Profundizar el estudio del framework Ionic 2 para sacar ventajas de toda la capacidad que este tiene para el desarrollo de aplicaciones móviles.
- Usar Ionic framework en su versión 2 para el desarrollo de aplicaciones móviles nos brinda facilidad al desarrollar para múltiples plataformas en un solo proceso de desarrollo de software.
- El uso de la metodología Scrum en proyectos de desarrollo de software ya que nos permite realizar entregables y acoplarnos a nuevos cambios o requerimientos, de la misma manera la generación de aplicativos escalables y detectar errores de forma oportuna.
- Usar herramientas de código libre (Open Source) en los proyectos de desarrollo brindan rentabilidad.
- Se recomienda el uso de servicios web RESTfull ya que son escalables y multilenguaje ya que usan el protocolo HTTP, adicional usa JSON lo que hace que sus objetos sean de fácil manipulación.

# ANEXOS

## Aplicación en Google Play

**Destino Pimampiro**  
Luis Yaguapaz Viajes  
★★★★★ 11  
Todos  
Esta app es compatible con todos tus dispositivos.  
**Instalada**

**PIMAMPIRO**  
— TIERRA DEL SOL —

EXISTE UN LUGAR

**PIMAMPIRO**  
— TIERRA DEL SOL —  
PATRIMONIO TURÍSTICO  
Nuestros atractivos turísticos...  
SERVICIOS  
Alquileres, organización...  
INTERÉS  
Agenda, seguridad, contactos...

Montañas, sol, ríos, arcoiris, música, alegría, fiesta, tradición, adrenalina, aire puro y frutas los 365 días del año. ¿Se imagina, todo esto en un solo lugar? Pimampiro es diversidad a la carta, donde usted se sentirá como en casa. Su gama climática y tierras fértiles lo convierten en un auténtico jardín productivo. Tierra ancestral donde el valor del dinero queda congelado en el tiempo con el trueque del sol. Terrazas agrícolas, petroglifos, centros ceremoniales, tumbas pre-colombinas forman parte de la arqueología local. Cuna intercultural donde conviven armónicamente mestizos, afro descendientes e

## Certificado - IARC Content Ratings

Certificación de Destino Pimampiro cuyo contenido apto para todo publico

 <b>Rating Certificate</b>			
<b>App Title:</b> Destino Pimampiro <b>Certificate ID:</b> b414841b-ecf8-4097-b918-cba7034402f8 <b>Date Issued:</b> Thursday, May 18, 2017		<b>Certificate Issued To:</b> Luis Yaguapaz <b>Storefront:</b> Google Play	
This rating may <b>only</b> be used on storefronts participating in IARC. It may <b>not</b> be used on physical products.			
Rating Authority	Region	Rating Category	Content Descriptors
ClassInd	Brazil		Classificação Livre
ESRB	The Americas		Everyone
PEGI	Europe		PEGI 3
USK	Germany		USK ab 0 Jahren
Generic	Other Regions		
<b>Interactive Elements:</b> Unrestricted Internet			
A permanent record of this certificate may be accessed <a href="#">here</a> .			

## Bibliografía

Apache Friends, A. (2017). *Apache Friends Org*. Retrieved from <https://www.apachefriends.org/es/download.html>

Apache Software, F. (2014, Enero). *Cordova Apache*. Retrieved from Cordova Apache: <https://cordova.apache.org/>

Basalo, A. (09 de Junio de 2016). *DesarrolloWeb.com*. Obtenido de <https://www.desarrolloweb.com/articulos/introduccion-angular2.html>

Clements, D. M. (2014). *Node Cookbook: Second Edition*.

Composer, D. M. (2017, 12 01). *Composer*. Retrieved from <https://getcomposer.org/download/>

JSON, O. (n.d.). *JSON Oficial Web Site*. Retrieved from <http://www.json.org/json-es.html>

LLC Yii, S. (2014). *The Definitive Guide to Yii 2.0*. Retrieved from <http://www.yiiframework.com/doc-2.0/guide-db-dao.html>

Phan, H. (2016). *Ionic 2 Cookbook Second Edition*. Birmingham: Packt Publishing Ltd.

Reference Manual, M. (2017). *Oracle Corporation and/or its affiliates*. Retrieved from <https://dev.mysql.com/doc/refman/5.7/en/introduction.html>

Reza Ichsani, M. (2017, Octubre 11). *Go Pensil*. Retrieved from Go Pensil: <http://www.gopensil.com/2017/10/pengertian-yii-framework.html>

Saini, G. (2017). *Hybrid Mobile Development with Ionic*.

Bahit, E. (2011). POO y MVC en PHP. *Argentina: Openlibra.*, 66.

- Canós, J. H., Letelier, P., Penadés, C., & Valencia, D. P. De. (2003). Metodologías Ágiles en el Desarrollo de Software. *Development*, 1–8. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.4553&rep=rep1&type=pdf>
- Marset, R. N. (2007). REST vs Web Services. *Internet*, 1–19.
- Schwaber, K., & Sutherland, J. (2013). La Guía de Scrum. *Scrumguides.Org*, 1, 21. Retrieved from <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- Tobergte, D. R., & Curtis, S. (2013). *Sbok. Journal of Chemical Information and Modeling* (Vol. 53). <https://doi.org/10.1017/CBO9781107415324.004>
- Valverde Ramos, E., & Hernández-Mora de Fuentes, P. (2014). TypeScript, 1–112. <https://doi.org/10.1002/ejoc.201200111>
- Yii Software LLC. (2014). Yii2 The Definitive Guide, (May).