



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN SISTEMAS**

**“DESARROLLO DE UNA APLICACIÓN WEB  
MULTIPLATAFORMA USANDO EL FRAMEWORK DJANGO,  
PARA PUBLICITAR EVENTOS SOCIALES, APLICADO EN EL  
MUNICIPIO DEL CANTÓN MORONA”**

TRABAJO DE TITULACIÓN: **PROYECTO TÉCNICO**

Para optar al Grado Académico de:

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**AUTOR: JUAN ALFREDO SOLÓRZANO ÁVILA**

**TUTORA: Ing. Linda Noralma Aguilar Moncayo Mg.**

Macas – Ecuador

2018

## ESCUELA DE INGENIERÍA EN SISTEMAS

El Tribunal de Trabajo de titulación certifica que: El trabajo de investigación: “DESARROLLO DE UNA APLICACIÓN WEB MULTIPLATAFORMA USANDO EL FRAMEWORK DJANGO, PARA PUBLICITAR EVENTOS SOCIALES, APLICADO EN EL MUNICIPIO DEL CANTÓN MORONA”, de responsabilidad del señor Juan Alfredo Solórzano Ávila, ha sido minuciosamente revisado por los miembros del Tribunal de Trabajo de Titulación, quedando autorizada su presentación.

Dr. Julio Santillán

VICEDECANO

\_\_\_\_\_

Ing. Patricio Moreno

DIRECTOR DE ESCUELA

\_\_\_\_\_

Mg. Linda Aguilar Moncayo

DIRECTOR DE TRABAJO DE TITULACIÓN

\_\_\_\_\_

Ing. Miguel Duque

MIEMBRO DE TRABAJO DE TITULACIÓN

\_\_\_\_\_

## **RESPONSABILIDAD DEL AUTOR**

“Yo, Juan Alfredo Solórzano Ávila soy responsable de las ideas, doctrinas y resultados expuestos en esta tesis, y el patrimonio intelectual de la Tesis de Grado pertenece a la Escuela Superior Politécnica de Chimborazo Ext. Morona Santiago”

---

Juan Alfredo Solórzano Ávila

## **AGRADECIMIENTO**

Agradezco a Dios que ha estado en mi camino incluso en momentos en los que no creía, gracias a mi madre quien me ha enseñado de valores y me ha mostrado ejemplos de fe, gracias a mi hermano y mis compañeros de carrera que han estado conmigo para aprender y desaprender, a la universidad, en especial a quienes he llamado maestros, al cuerpo administrativo: desde el encargado de la limpieza y mantenimiento hasta los directores de la extensión de Morona Santiago quienes han colaborado generosamente por una politécnica en la provincia. Gracias a mis amigos y familia por estar conmigo incondicionalmente apoyando mis proyectos.

## **DEDICATORIA**

A todos mis maestros y compañeros, a mi familia, en especial a Maria Elena.

Juan Solórzano.

## **RESUMEN**

Se ha culminado exitosamente el desarrollo de Mape; una Aplicación Web Multiplataforma que sirve de mapa turístico, aplicado en el Gobierno Municipal del Cantón Morona de la provincia de Morona Santiago, con el propósito de promocionar los lugares y eventos. Se realizó una encuesta sobre las necesidades de los turistas/consumidores y los comerciantes que trabajan como servidores turísticos registrados en la municipalidad. Dicha investigación se ha realizado con el objetivo de identificar o validar una problemática a resolver. Los resultados del análisis de las encuestas han servido para la planificación del proyecto Mape, realizada bajo la metodología de desarrollo ágil Scrum que ha sido utilizado para la gestión del proyecto; el documento de especificación de requerimientos de software (SRS) ha sido introducido dentro de esta metodología como resultado de la planificación inicial para dar pauta a la creación de las herramientas de Scrum: Producto Backlog y Sprint Backlogs 1, 2 y 3. La arquitectura del proyecto se basa en el Modelo Vista Template (MVT) de Django, en la programación de la aplicación se ha hecho uso de tecnologías de software libre de última generación como el framework front-endMaterializeCSS para diseño en HTML5 y CSS3, el sistema está basado en el paradigma Orientado a Objetos, escrito en lenguaje de programación Python. La implementación se realizó bajo el nombre de dominio mape.live, accesible desde la página oficial del Gobierno Municipal del Cantón Morona, el sistema corre sobre un servidor Nginx, en Ubuntu 16. También se han implementado los servicios del web-service de Mapbox.com así como para la autenticación con Facebook, además de un módulo de administración incluido en Django. El estudio revela que 63.3% de los 43 consumidores encuestados cree que el turismo mejoraría con mayor difusión de información, el 87,92% de los 38 servidores turísticos intervienen en el desarrollo de eventos y 9 de cada 10 se interesa en usar una aplicación para invitar a sus clientes a sus eventos. Se ha evidenciado los beneficios del uso del framework Django, además del mismo uso del lenguaje Python, este entorno de trabajo representa un gran aporte a la Programación Orientada a Objetos. Se recomienda el uso del documento SRS para la especificación de los principales requerimientos de un proyecto de software.

**PALABRAS CLAVE:** <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INGENIERIA DE SOFTWARE>, <TURISMO>, <SCRUM (METODOLOGÍA DE DESARROLLO ÁGIL)>,<PROGRAMACIÓN DE APLICACIONES WEB>, <SOFTWARE LIBRE>, <PYTHON (SOFTWARE)>.



## ÍNDICE DE ABREVIATURAS

**APP:** Aplicación de software

**CSS:** Cascading Style Sheets

**GUI:** Graphical User Interface, Unidad de Interfaz Gráfica.

**HTML5:** Hypermedia Text Markup Language, o traducido; lenguaje de marcas de hipertexto.

**HW:** Hardware

**JRE:** Java Runtime Environment, entorno de tiempo ejecución para Java.

**MVT:** Modelo Vista Template.

**NYC1:** New York City 1, región de internet.

**PIL:** Python Imaging Library o (traducido) librería de imágenes de Python

**SRS:** Acrónimo de Software Requirements Specification (Especificación de Requerimientos de Software)

**SGDB:** Sistema Gestor de Bases de Datos.

**SW:** Software

**VPS:** Virtual Private Server, Servidor Virtual Privado en español.

## DEFINICIONES

**API:** De las siglas en inglés Application Programming Interface.

**Bug/s:** “Bicho/s” en español, son errores en los sistemas que no afectan de sobremanera en el funcionamiento, provienen principalmente de incompatibilidad de plataformas.

**Brainstorming:** Lluvia de ideas, técnica de organización de ideas interrelacionadas.

**Conexión:** Comunicación entre dos entes que tienen similares características de comunicarse.

**Django:** Framework para desarrollo de aplicaciones móviles, facilita la organización, gestionabilidad del proyecto de software.

**Framework:** Entorno de trabajo en español, se refiere a la plataforma sobre la cual se desarrolla un proyecto de software, incluyendo sus técnicas y metodologías.

**Google Fonts:** Tipos de letras obtenidos de los servidores de Google.

**Herramientas CASE:** Aplicaciones informáticas que nos permiten ayudar en todos los aspectos de ciclo de vida del software, en tareas como el proceso de realizar el diseño del proyecto, cálculo de costes, documentación, etc. Permite aumentar nuestra productividad en el desarrollo del mismo y reducir el coste en términos de tiempo y dinero. Son aplicaciones software que respaldan el desarrollo y el mantenimiento del software.

**JavaScript:** Lenguaje de programación de categoría Orientada a Objetos.

**jQuery:** Librería de JavaScript para la interacción con documentos HTML.

**Link:** Enlace en español, es un vínculo que lleva hacia otro sitio del sistema.

**Login/logueado:** Requisito de ingreso como usuario del Sistema informático, consta de una identificación pública y una contraseña privada.

**MapBox:** Servidor de mapas en la web.

**Mape.live:** Dominio web de la aplicación informática producto del presente trabajo práctico, el nombre Mape proviene de la unión de dos conceptos: mapa y eventos.

**Paradigma de programación orientado a objetos:** es aquella en la que se usa objetos como formas de representación de objetos reales, para ello emplea estrictos métodos de abstracción, encapsulamiento, polimorfismo, herencia, acoplamiento y cohesión.

**Password/Pass:** Contraseña en español. Es una clave asociada a una cuenta de usuario en un sistema informático.

**Servidor:** Computador de características especiales con el objetivo de proveer a los sistemas informáticos: almacenamiento, procesamiento, servicios web, alojamiento de base de datos, y otras configuraciones de datos.

**Sistema de gestión de datos:** dedicadas a servir de interfaz entre las bases de datos, el usuario y las aplicaciones clientes que las utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y un lenguaje de consulta. El propósito de estas aplicaciones es de manejar de manera clara, sencilla y ordenada un conjunto de datos.

**Sistema Operativo:** Software que se encarga de ofrecer un entorno para el uso de dispositivos o hardware.

**SublimeText:** Editor de texto, diseñado para desarrollo de software.

**Slide:** Colección de objetos que se muestran de forma especial para que el usuario las deslice una tras otra en la pantalla.

**Ubuntu:** Sistema operativo de código abierto, distribuido por Linux.

**Web-service:** Estándares para establecer comunicaciones entre aplicaciones.

**Wearables:** Son dispositivos externos de uso como prenda o complemento para vestir que se comunica con smartphones para realizar alguna función.

## ÍNDICE GENERAL

FIRMAS DE RESPONSABILIDAD.....	2
RESPONSABILIDAD DEL AUTOR.....	3
AGRADECIMIENTO.....	4
DEDICATORIA.....	5
RESUMEN.....	6
SUMARY.....	7
ÍNDICE DE ABREVIATURAS.....	8
DEFINICIONES.....	9
ÍNDICE GENERAL.....	10
ÍNDICE DE FIGURAS.....	13
ÍNDICE DE TABLAS.....	15
INTRODUCCIÓN .....	17
CAPÍTULO I: .....	18
MARCO REFERENCIAL .....	18
1.1 ANTECEDENTES DEL PROYECTO DE TESIS .....	18
1.2 Justificación del trabajo de tesis .....	19
1.2.1 Justificación teórica .....	19
1.2.2 Justificación aplicativa.....	20
1.3.1 Objetivo general .....	21

CAPÍTULO II: .....	22
MARCO TEÓRICO.....	22
2.1 POO con Python .....	22
2.3 Materialize (HTML5, CSS3, JavaScript) .....	39
2.3.1 Material Design .....	39
2.3.2 Características.....	40
2.3.3 Ventajas y desventajas .....	40
2.4 Métodos y técnicas.....	42
2.4.1 Metodología Ágil Scrum. ....	42
2.4.2 Investigación.....	48
2.4.3 Base de datos .....	49
CAPÍTULO III:.....	51
DESARROLLO DE LA APLICACIÓN MÓVIL MAPE .....	51
3.1 Planificación del proyecto.....	52
3.1.1 Cronograma de actividades.....	52
3.1.2 Análisis de resultados de encuesta a Usuarios .....	52
3.1.2 Definición del Ámbito .....	65
3.1.3 Estudio de Factibilidad .....	70
3.2 Desarrollo de tareas.....	75
3.2.1 Preparación de entorno de trabajo.....	75
3.2.2 Agenda según Scrum .....	76
3.3 Implementación y pruebas .....	84
3.3.1 Definición de Estándares de programación.....	84

3.3.2 Pruebas de Verificación .....	85
3.4 Resultados.....	87
4. CONCLUSIONES .....	91
5. RECOMENDACIONES .....	92
8. BIBLIOGRAFÍA .....	93
9. ANEXOS .....	97

## INDICE DE FIGURAS

### CAPITULO I

<b>Figura 1-1:</b> Estadísticas globales de uso de navegadores.....	18
<b>Figura 1-2:</b> Estadísticas globales de uso de navegadores en dispositivos móviles de américa latina.....	19

### CAPITULO II

<b>Figura 2-1:</b> Arquitectura de proyecto Django.....	30
<b>Figura 2-2:</b> Material Design Concept.....	39
<b>Figura 2-3:</b> Scrum en acción.....	42

### CAPITULO III

<b>Figura 3-1:</b> Cronograma de actividades.....	51
<b>Figura 3-2:</b> Usuarios_Pregunta_Piloto.....	52
<b>Figura 3-3:</b> Usuarios_Pregunta_1.....	52
<b>Figura 3-4:</b> Usuarios_Pregunta_2.....	55
<b>Figura 3-5:</b> Usuarios_Pregunta_3.....	55
<b>Figura 3-6:</b> Usuarios_Pregunta_4.....	56
<b>Figura 3-7:</b> Usuarios_Pregunta_5.....	57
<b>Figura 3-8:</b> Servidores_Pregunta_1.....	58
<b>Figura 3-9:</b> Servidores_Pregunta_2.....	59
<b>Figura 3-10:</b> Servidores_Pregunta_3.....	60
<b>Figura 3-11:</b> Servidores_Pregunta_4.....	61
<b>Figura 3-12:</b> Servidores_Pregunta_5.....	62
<b>Figura 3-13:</b> Objetos principales de la aplicación.....	66

<b>Figura 3-14:</b> ProductBacklog del proyecto Mape.....	75
<b>Figura 3-15:</b> Captura de pantalla del “Registro”.....	78
<b>Figura 3-16:</b> Captura de pantalla “Principal”.....	78
<b>Figura 3-17:</b> Captura de pantalla de “Formulario de usuario”.....	79
<b>Figura 3-18:</b> Captura de pantalla “login”.....	88
<b>Figura 3-19:</b> Captura de pantalla “signup”.....	88
<b>Figura 3-20:</b> Captura de pantalla “principles”.....	89
<b>Figura 3-21:</b> Captura de pantalla “map”.....	89

## INDICE DE TABLAS

### CAPITULO II

<b>Tabla 2-1:</b> Versiones de Django hasta 1-dic-2017. ....	28
<b>Tabla 2-2:</b> Tabla de resultados entre frameworks Django y RoR.....	36

### CAPITULO III

<b>Tabla 3-1:</b> Tamaño de las muestras poblacionales.....	52
<b>Tabla 3-2:</b> Características de usuario administrador. ....	68
<b>Tabla 3-3:</b> Características de usuario normal.....	68
<b>Tabla 3-4:</b> Características de usuario comercial. ....	69
<b>Tabla 3-5:</b> Personal de trabajo. ....	70
<b>Tabla 3-6:</b> Equipamiento de trabajo.....	70
<b>Tabla 3-7:</b> Equipo de software. ....	71
<b>Tabla 3-8:</b> Valoración de hardware. ....	72
<b>Tabla 3-9:</b> Valoración de software. ....	72
<b>Tabla 3-10:</b> Valoración de suministros de oficina. ....	73
<b>Tabla 3-11:</b> Valoración de otros rubros. ....	73
<b>Tabla 3-12:</b> Valoración total.....	74
<b>Tabla 3-13:</b> Nomenclatura de tabla de tareas.....	76
<b>Tabla 3-14:</b> Tareas del Sprint 1.....	76
<b>Tabla 3-15:</b> Resultado de Sprint 2.....	79
<b>Tabla 3-16:</b> Resultado de Sprint 3.....	82
<b>Tabla 3-17:</b> Caso de prueba 1.....	84
<b>Tabla 3-18:</b> Caso de prueba 2.....	84
<b>Tabla 3-19:</b> Caso de prueba 3.....	84

<b>Tabla 3-20:</b> Caso de prueba 4.....	85
<b>Tabla 3-21:</b> Caso de prueba 5.....	85
<b>Tabla 3-22:</b> Caso de prueba 6.....	85
<b>Tabla 3-23:</b> Caso de prueba 7.....	85
<b>Tabla 3-24:</b> Relación peso/carga de páginas estáticas.....	87

## INTRODUCCIÓN

Dado el aumento en la población turística en el Ecuador gracias las campañas publicitarias generadas por el gobierno central y sobre todo por nuestra riqueza turística; se pretende apoyar este sector a través del departamento de Turismo del Gobierno Municipal del Cantón Morona de la provincia de Morona Santiago, investigando una problemática y resolviéndola con la implementación de una aplicación web que apoye las necesidades de este ámbito, desarrollando un sistema basado en lenguaje de programación Python, enfocado en la promoción de eventos sociales y localidades turísticamente atractivas que los usuarios podrán ingresar libremente.

Ya que se cuenta con entornos de desarrollo que nos permiten simplificar las tareas repetitivas de programación, se hace uso de Django Framework el cual básicamente nos ayudará a concentrar los esfuerzos en la implementación del modelo de negocio, es decir, la funcionalidad misma de la aplicación mas no en la tediosa programación de un controlador personalizado para una aplicación.

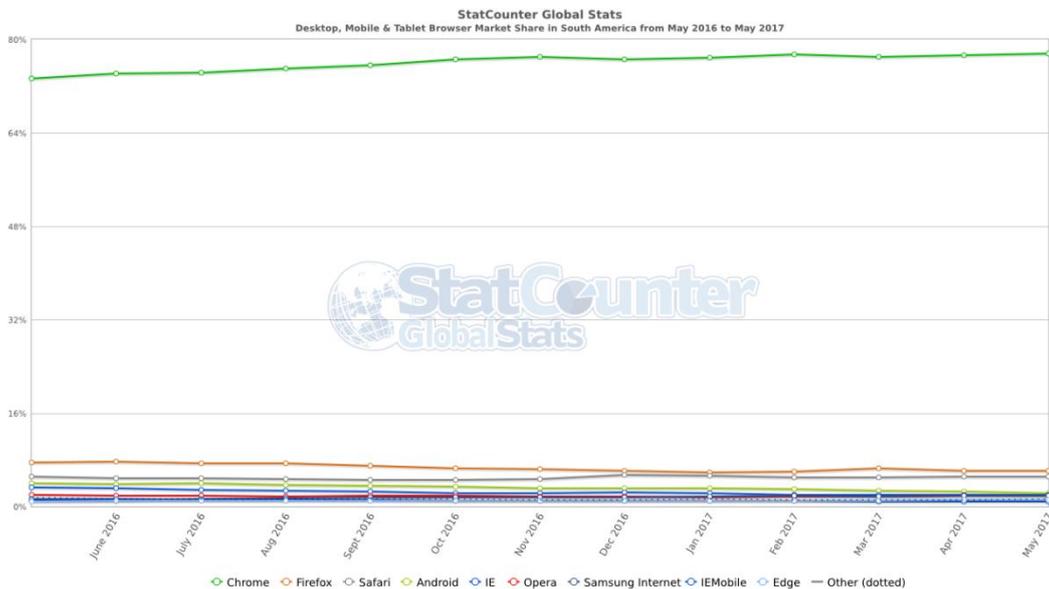
El presente trabajo práctico pretende demostrar los beneficios en el uso del framework Django basado en el lenguaje Python, e implícitamente, el uso de una arquitectura de aplicación basada en MTV: Modelo Vista Template. Con la implementación de una aplicación web multiplataforma que apoye el sector turístico en Morona Santiago a través de un mapa de eventos y localidades.

# CAPÍTULO I:

## MARCO REFERENCIAL

### 1.1 ANTECEDENTES DEL PROYECTO DE TESIS

- Según los datos obtenidos del portal web StatCounter mostrados en la **Figura 1-1**, las estadísticas muestran que el navegador que se mantiene como el más usado es Google Chrome hasta la fecha (mayo del 2017) con 54.2% de todos los navegadores usados desde plataformas de escritorio y móviles. Únicamente en Sudamérica el porcentaje es de 77.53% de todos los accesos desde éste navegador (Línea color verde), incluidos desde plataformas móviles Android y otros.

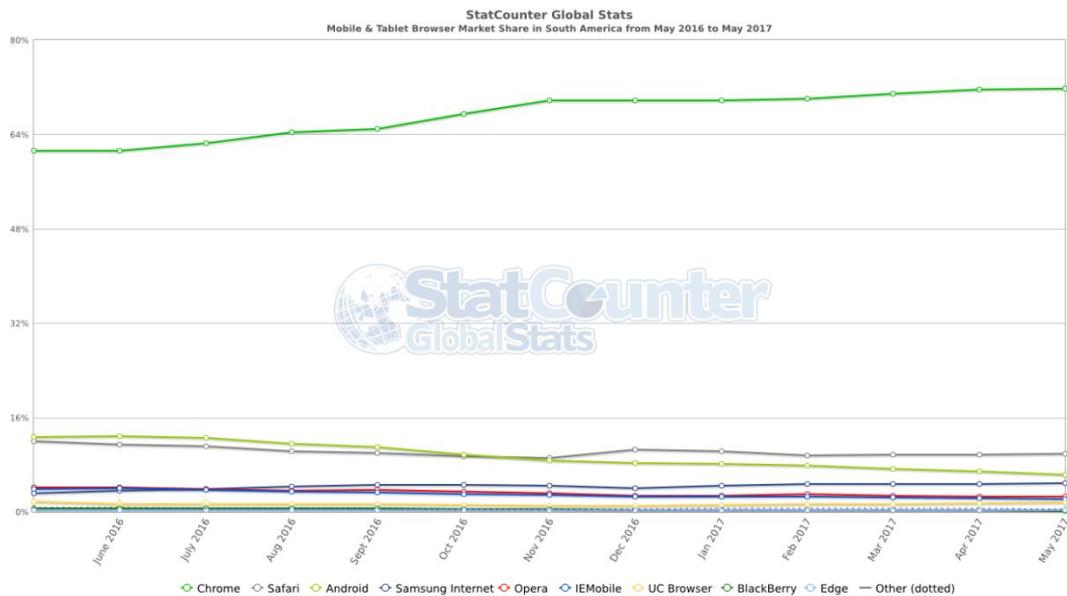


**Figura 1-1:** Estadísticas globales de uso de navegadores.

Fuente: StatCounter.com

- En Sudamérica el uso de los navegadores para dispositivos móviles también está dominado por el de Google, según las estadísticas de StatCounter.com el 73.6% de todo el

acceso desde los dispositivos móviles y tabletas se realiza a través de Chrome. Seguido por los navegadores Safari (9.78%), Android(6.26%), Samsung Internet(4.86%) y Opera(2.6%).



**Figura 1-2:** Estadísticas globales de uso de navegadores en dispositivos móviles de América Latina.

Fuente: StatCounter.com

- Está claro que los navegadores estarán entre las herramientas más usadas durante mucho tiempo, además de involucrarse cada vez más en la vida cotidiana de los seres humanos en casi todos los ámbitos, “en los próximos años, una parte muy importante de las operaciones comerciales de una organización, será soportada por aplicaciones móviles confiables y seguras”. (Alberto Balderas, 2014)

El gobierno central del Ecuador ha impulsado grandes campañas turísticas a nivel mundial con lo cual se pretende atraer a más turistas a nuestro país, por lo que debemos estar preparados con lo que sea necesario para dar un servicio de calidad, esto implica el uso de medios tecnológicos para optimizar el contacto entre empresa y cliente, usando una aplicación web que sea accesible para la mayoría de gente, y así exponer los servicios, productos, necesidades y eventos cuando estos sean requeridos por los usuarios de internet.

## 1.2 Justificación del trabajo de tesis

### 1.2.1 Justificación teórica

Muchas necesidades contemporáneas están siendo satisfechas por aplicaciones web que encontramos en internet, desde editores de texto a editores de audio y video, muchas de estas

aplicaciones han sido diseñadas usando herramientas comerciales o de pago y otras con una filosofía de gratuidad, para el presente proyecto se ha de usar únicamente herramientas de licencia abiertas y gratuitas, además del hecho que abarcan todo lo necesario para el desarrollo contemporáneo de aplicaciones de punta.

Los avances en la informática ha abierto un mercado creciente para el cual debemos estar preparados con las mejores técnicas de desarrollo, sin dejar cerrada la posibilidad de contribuir al cambio, siendo el desarrollo web multiplataforma uno de los grandes pilares que marcan historia con la utilización de nuevas tecnologías basadas en la *POO*.

El presente proyecto, en su parte teórica-metodológica pretende mostrar los grandes beneficios en el uso de nuevas tendencias en programación y plataformas tecnológicas de libre acceso; Estructura *MVT*, Metodología de desarrollo *SCRUM*, lenguajes de programación tales como *HTML5*, *CSS3*, *PYTHON*, Y *JAVASCRIPT* y *PostgreSql* para la base de datos.

### ***1.2.2 Justificación aplicativa***

El cantón Morona tiene mucho potencial turístico, ya que presenta mucha variedad en eventos de categorías como: deportes extremos, fiestas populares, paseos, conciertos musicales, historia, tradiciones y cultura. Las ciudades tienen mucho movimiento durante sus festividades, además de los eventos generados por sus habitantes durante todo el año, es decir, los eventos nos rodean en todo momento.

Ya que los eventos se están dando en todas partes; sería una ventaja tener una visión clara de qué, dónde y cuándo se realizan. Se pretende crear un mapa con los eventos y los lugares en donde se realizan, informar sobre lo que hay en la localidad de la que se desee conocer. Es así que se observa una oportunidad de mejorar el potencial turístico no solamente del cantón morona sino también de todas las ciudades del Ecuador.

El presente proyecto pretende ayudar a fomentar el ámbito turístico a través del desarrollo de una aplicación web, a la cual podrán acceder desde cualquier navegador que encontramos en PCs, teléfonos inteligentes, tabletas y portátiles. Se trata de una *App web* encargada de mostrar un mapa con eventos y localidades subidas por cualquier usuario de internet, estos podrán acceder a la aplicación como invitados o registrarse en el sistema con una cuenta personal. De esta manera la aplicación contendrá información actualizada de los eventos del sector propuestos por los mismos usuarios registrados.

### **1.3 Objetivos**

#### ***1.3.1 Objetivo general***

Desarrollar una aplicación web multiplataforma para publicitar eventos sociales y aplicarlo en el Municipio del Cantón Morona.

#### ***1.3.2 Objetivos específicos***

1. Realizar una investigación para determinar la necesidad de los turistas, consumidores y comerciantes en cuanto a eventos sociales.
2. Analizar las factibilidades técnicas para la realización de una aplicación multiplataforma.
3. Realizar la aplicación móvil *Mapebajo* los estándares *HTML5*, *CSS3*, *Python* bajo el *framework Django* y *Javascript*.

## **CAPÍTULO II:**

### **MARCO TEÓRICO**

En este capítulo se describe con detalle los aspectos tecnológicos involucrados en el desarrollo de la tesis como tal. Involucra la parte investigativa y la parte aplicativa de desarrollo del sistema de software.

Se empieza describiendo los temas que definen el proceso de desarrollo del proyecto, introduciendo a las bases fundamentales con la Programación Orientada a Objetos con Python, luego sigue lo referente a la historia y versiones del Framework Django que implementa la arquitectura MVT (Modelo Vista Template) para sus proyectos.

Luego se muestra la definición del lenguaje de etiquetas HTML5, CSS3 y JavaScript, estos tres últimos van de la mano en el diseño de interfaces de usuario para nuestro proyecto y es comúnmente usado en la programación de páginas web. Por último se continúa con la descripción de la técnica de encuesta aplicada para la obtención de información, en este caso referente a las necesidades de publicidad para eventos de los potenciales usuarios de la aplicación.

#### **2.1 POO con Python**

En este apartado veremos en resumen como se crean clases, variables o campos, definición de funciones, métodos, sentencias y cómo usarlos todos en un breve ejemplo. Para entender los siguientes ejemplos es necesario un conocimiento básico en programación.

Cabe recalcar que entre la documentación de Python Argentina se encuentran también algunos tutoriales sobre este tema, mucho más didácticos y extensos que los aquí mostrados. Además contiene información sobre el módulo Django 1.8.

#### **Sintaxis para definición de Clases:**

```
>>>class Clase:  
...     pass
```

```
... .
... .
... .
... <declaración-N>
```

El código anterior muestra lo sencillo que es el código Python, en ella se declara una clase entera (sin ningún objeto). Una vez ejecutado el código la clase “Clase” pasa a formar parte de las variables de entorno dentro del Python Shell.

Si lo aplicamos a una clase Insecto se vería algo así:

```
>>>class Insecto:
...     ojos = none
...     pelo = none
...     antenas = none
```

El código anterior define una clase Insecto con las variables vacías (none). Como se puede observar en el código Python se reducen los caracteres al final de las sentencias como el punto y coma, también la declaración del tipo de variable (String, Number, Serial, Array...), dejándolo como un objeto abierto a cualquier posible tipo de variable u objeto (funcion, metodo, variable, instancia de clase).

### **Sintaxis para definición de objetos de clase:**

Las clases creadas soportan dos tipos de operaciones como se describe en el tutorial de Python 3.5 de la comunidad argentina que indica: “Los objetos clase soportan dos tipos de operaciones: hacer referencia a atributos e instanciación”. (Python Software Fundation, 2017)

Para referirse a los atributos se usa el nombre del objeto seguido por un punto, por ejemplo *insecto.alas*. Los nombres después de los puntos son objetos, variables y funciones definidas en la clase. A continuación un ejemplo:

```
>>>class Insecto:
...     ojos = none
...     pelo = none
...     antenas = none
```

```
... def f(self):  
...     return 'hola mundo'
```

En el código se pueden observar los atributos ojos, pelo y antenas con valores none por defecto, estos valores pueden cambiar a cualquier tipo de valor antes descritos de forma dinámica, de hecho es necesario cambiar el anterior código ya que las variables estarían compartidas con todas las instancias de su clase y eso no es deseable.

Se define como último objeto el método “f” que devuelve la cadena ‘hola mundo’. Existen algunos métodos predefinidos en las clases de Python, estos métodos o comportamientos de ciertos objetos de python son reconocidos por usar doble guion bajo “\_\_” antes y después de sus nombres, el siguiente ejemplo muestra el uso de `__init__`:

```
>>> class Insecto:  
...     def __init__(self, a, b, c):  
...         self.ojos = a  
...         self.pelo= b  
...         self.antenas = c  
...escarabajo= Insecto(2, 320, 2)  
...escarabajo.pelos  
...320
```

El método `__init__(self, argumentos)` nos ayuda a redefinir el método de instanciación, puede requerir de argumentos los cuales pueden ser usados para instanciar un objeto de la clase.

### **Objetos Métodos**

Como regla general un objeto método debería instanciarse antes de referirse a él.

```
>>>insecto= Insecto()  
>>>insecto.f()  
'hola mundo'  
>>>insecto_f = insecto.f  
>>>while True:  
...     print(insecto_f())
```

```
'hola mundo'  
'hola mundo'  
... #seguirá imprimiendo por siempre...
```

Como se habrá notado el llamado a la función `insecto.f()` sin ningún argumento a pesar de que se definió que `f(self)` especifica un argumento “self”. En Python este argumento `self` (sí mismo) le indica a la clase que debe instanciarse con el objeto mismo como primer argumento. es decir que `insecto.f()` es lo mismo que `Insecto.f(insecto)`.

### **Variables de clase y de instancia**

Hablar de variables de instancia es hablar de variables con un grado de permiso y restricción; permiso para usar la variable o método dentro de la instancia de un objeto y restricción para que no sea usada fuera del mismo. Las variables de clase son variables y métodos que son compartidos por todas las instancias de la clase.

```
>>>class Insecto:  
...     tipo:'omnívoro' #variable de clase compartida entre  
instancias  
...     def __init__(self, a, b, c):  
...         self.ojos = a #variables de instancia con valores  
únicos  
...         self.pelo= b  
...         self.antenas = c  
...escarabajo= Insecto(2, 320, 2)  
...lombris= Insecto(0,0,0)  
...escarabajo.pelo  
320  
...lombris.pelo  
0
```

### **Herencia**

Para crear una clase basada en otra se usa la siguiente sintaxis:

```
>>>class Mosca(Insecto):  
...     <declaración-1>
```

```
... .  
... .  
... .  
... <declaración-N>
```

También se pueden usar otras expresiones para indicar la clase padre, por ejemplo, cuando la clase padre está definida en otro módulo:

```
>>>import modulo  
>>>class Mosca(modulo.Insecto):  
... <declaración-1>  
... .
```

En el código anterior se define una clase “Mosca” que contiene todo lo definido para su clase padre a lo que puede acceder (según lo permita su clase padre), de esta forma hemos llegado al inicio de este apartado, concluyendo en el inicio de un nuevo eslabón de un sistema llamado Clase, en este caso la clase “Mosca” que puede contener más objetos dentro de sí, como se ha visto en los ejemplos anteriores. De esta forma se puede observar en un plano general como se componen las clases para formar un sistema y su grado de abstracción de la realidad.

## 2.2 Django Web Framework

Antes de avanzar con el tema de Django es necesario entender bien qué es y para qué sirve un framework; según el ingeniero Jordi Sánchez “es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación.”(Sanchez, 2006). Un framework web es una herramienta de programación de alto nivel, se le llama web porque se obtiene una página web como interfaz para el usuario, por lo general se usa con tecnologías como JavaScript, CSS, HTML, XML y PHP, independientemente del lenguaje de programación. Un framework web es un entorno de trabajo que nos ofrece herramientas para la creación de páginas web interactivas con el objetivo de conectar el modelo de negocios con la GUI de usuarios.

Según la traducción de los textos de los autores de Django: “Utilizando Django puedes crear y mantener aplicaciones Web de alta calidad con un mínimo esfuerzo, en el mejor de los casos, el desarrollo web es un acto entretenido y creativo; en el peor, puede ser una molestia repetitiva y

frustrante. Django te permite enfocarte en la parte divertida [...] al mismo tiempo que mitiga el esfuerzo de las partes repetitivas”. (Holovaty y Kaplan-Moss, 2015)

### 2.2.1 Historia

La historia de Django no puede ser descrita de forma más precisa que la ya realizada por sus fundadores Adrian Holovaty y Jacob Kaplan-Moss en “El libro de Django”:

*“Es útil entender por qué se creó el framework, ya que el conocimiento de la historia pone en contexto la razón por la cual Django trabaja de la forma en que lo hace.*

*[...]El camino clásico de un desarrollador Web es algo como esto:*

- 1. Escribir una aplicación Web desde cero.*
- 2. Escribir otra aplicación Web desde cero.*
- 3. Darse cuenta de que la aplicación del paso 1 tiene muchas cosas en común con la aplicación del paso 2.*
- 4. Refactorizar el código para que la aplicación 1 comparta código con la aplicación 2.*
- 5. Repetir los pasos 2-4 varias veces.*
- 6. Darse cuenta de que acabamos de inventar un framework.*

*Así es precisamente como fue creado Django.*

*Django nació naturalmente de aplicaciones de la vida real escritas por un equipo de desarrolladores Web en Lawrence, Kansas. Nació en el otoño boreal de 2003, cuando los programadores Web del diario Lawrence Journal-World, Adrian Holovaty y Simon Willison, comenzaron a usar Python para crear sus aplicaciones.*

*El equipo de TheWorld Online, responsable de la producción y mantenimiento de varios sitios locales de noticias, prosperaban en un entorno de desarrollo dictado por las fechas límite del periodismo. Para los sitios — incluidos LJWorld.com, Lawrence.com y KUsports.com — los periodistas (y los directivos) exigían que se agregaran nuevas características y que aplicaciones enteras se crearan a una velocidad vertiginosa, a menudo con sólo días u horas de preaviso. Es así que Adrian y Simon desarrollaron por necesidad un framework de desarrollo Web que les ahorrara tiempo — era la única forma en que podían crear aplicaciones mantenibles en tan poco tiempo —.*

*En el verano boreal de 2005, luego de haber desarrollado este framework hasta el punto en que estaba haciendo funcionar la mayoría de los sitios World Online, el equipo de World Online, que ahora incluía a Jacob Kaplan-Moss, decidió liberar el framework como software de código abierto. Lo liberaron en julio de 2005 y lo llamaron Django, por el guitarrista de jazz Django Reinhardt.”.(Garcia, 2015)*

### 2.2.2 Versiones

Las primeras versiones de Django carecían de soporte técnico, además son incompletas y contienen ciertos bugs que son pulidas en versiones posteriores hasta llegar a la versión 1.8 y su última versión 1.9 como se expone en la siguiente tabla:

**Tabla 2-1:** Versiones de Django hasta 1-dic-2017.

Versión	Fecha	Notas
0.9	16 de noviembre de 2005	
0.91	11 de enero de 2006	"new-admin"
0.95	29 de julio de 2006	"magicremoval"
0.96	23 de marzo de 2007	"newforms", herramientas de testeo
1.0	03 de septiembre de 2008	Estabilidad de la API, administrador desacoplado, unicode
1.1	29 de julio de 2009	Agregados, testeos basados en transacción
1.2	17 de mayo de 2010	Múltiples conexiones de bd, CSRF, validación de modelo
1.3	23 de marzo de 2011	Vistas basadas en clases, archivos estáticos
1.4	23 de marzo de 2012	Zonas horarias, pruebas de navegador, plantillas de aplicación.
1.5	26 de febrero de 2013	Soporte para Python 3, modelo de usuario configurable
1.6	06 de noviembre de 2013	Dedicado a Malcolm Tredinnick,

		Administración de transacciones de bd, agrupación de conexiones.
1.7	02 de septiembre de 2014	Migraciones, carga de aplicación y configuración.
1.8	01 de abril de 2015	Soporte nativo para múltiples motores de plantillas. comunicado de apoyo a largo plazo, soportado hasta por lo menos, abril del 2018
1.9	01 de diciembre de 2015	Validación automática de contraseñas. Nuevos estilos para la interfaz de administrador.

**Realizado por:** Juan Solórzano

**Fuente:** Blog Django

### 2.2.3 Características de Django 1.8

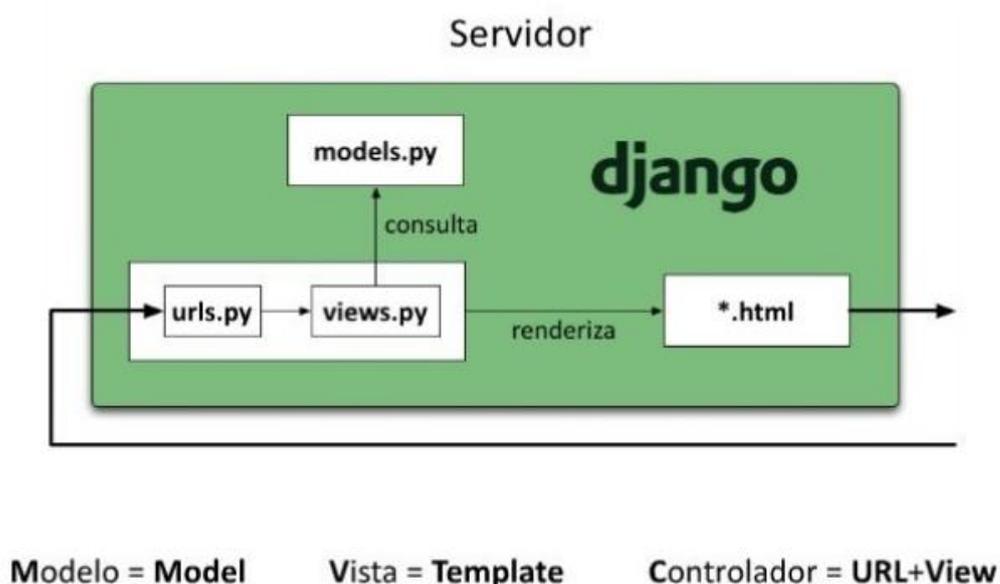
Debido a sus orígenes antes mencionados; existen muchas características únicas en su clase en Django pero se ha resumido en las más importantes, enfocadas al desarrollo rápido de aplicaciones web y mejorar su gestión:

- Django está desarrollado como un módulo de Python, una de sus principales características es su fácil aprendizaje, heredado de su lenguaje de programación: “Si no tienes experiencia programando en Python, te espera una grata sorpresa. Es fácil de aprender y muy divertido de usar.”(Holovaty& Kaplan-Moss, 2008, p.5)
- El acoplamiento entre los módulos de un proyecto son débiles (*looselycouple*): “Eso significa que cada pieza de la aplicación Web que funciona sobre Django tiene un único propósito clave, que puede ser modificado independientemente sin afectar las otras piezas.” (Holovaty& Kaplan-Moss, 2008, p.3).
- Django registra cada uno de los cambios en los modelos la base de datos y se encarga de realizar las migraciones correspondientes: “... hemos agregado nuestra propia facilidad para la migración de la base por lo que los desarrolladores no tienen que aplicar parches SQL para mantener los actuales esquemas de base de datos funcionando” (Holovaty& Kaplan-Moss, 2008, p.248).
- La arquitectura de un proyecto en Django es bajo el esquema Modelo Vista Template (*MVT*), “Django define su estructura... llamada modelo-vista-template”(Cumba & Barreto, 2012)

### 2.2.4 Arquitectura MVC en Django

Django está inspirado en la clásica arquitectura de programación Modelo Vista Controlador, pero en Django esta tendencia se aplica resumiendo una nueva estructura denominada Modelo Vista Template (Template: plantilla).

Como resultado se obtiene lo que sería un “controlador” en MVC como “vista” en MVT y lo que tiene que ver con la “vista” se denomina “template” tal como se muestra en la **Figura 2-1**. Parte del trabajo de controlador lo maneja el mismo framework con sus funciones para manejar el acceso a la base de datos y demás, por otro lado el denominado “modelo” en los dos significan lo mismo exceptuando en el acceso a la base de datos que ya viene implementada en Django.



## **Figura 2-1:** Arquitectura de proyecto Django.

**Realizado por:** Ander Beaskoetxea (2009)

M significa "Model" (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.

T significa "Template" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web o otro tipo de documento.

V significa "View" (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre los modelos y las plantillas. (Holovaty & Kaplan-Moss, 2008, p.48)

### **2.2.5 Ventajas y desventajas**

Las ventajas de usar Django se han recopilado en base a un estudio realizado por estudiantes de la Escuela Superior Politécnica de Chimborazo, que incluye un análisis donde se comparan a Django frente a su homónimo Ruby onRails. Al igual que en el presente caso se busca el framework con mejores prestaciones para el desarrollo de una aplicación web con políticas de acceso, base de datos relacional, y manejo de GIS:

*"Se necesita crear una solución informática que permitirá automatizar el proceso de contratación docente, el que conste con una interfaz amigable con el usuario, estableciendo políticas de acceso para el ingreso del sistema, además de tener un control de la información de los docentes los cuales están y serán contratados por la Dirección.*

*La aplicación web facilitara la realización de reportes, os mismos que detallaran información de los docentes, las escuelas y la ubicación geográfica donde encuentran laborando."* (Cumba & Barreto, 2012)

Para realizar este análisis comparativo se ha considerado 3 módulos de prueba que han establecido los autores:

#### ***"Módulo 1***

*Es el módulo que será desarrollador para probar los siguientes parámetros de:*

- *Manejo del patrón MVC*

- *Principio DRY*

### **Módulo 2**

*En este módulo implementara para probar los parámetros de seguridad.*

*En este módulo se desarrollara un login para la autenticación de los usuarios en una aplicación web, mediante el uso de sesiones, protección de los datos que viajan en la web como las contraseñas, el uso de cookies y la validación de usuarios existentes.*

### **Módulo 3**

*En este módulo se probaran los parámetros del Producto, en el cual se realizan cuadros comparativos que reflejan las características que presenta cada framework al momento de desarrollar aplicaciones web. Esto nos permitirá distinguir las ventajas en una producción final.” (Cumba & Barreto, 2012)*

### **Ventajas frente a RoR**

- **Uso de plantillas:** *Django framework presenta un completo sistema de plantillas, el cual su implementación es de manera muy sencilla, nos permite realizar paginas de manera rápida y ágil, pero al ser un sistema de plantillas ideal para diseñadores mas no para programadores por lo que la creación de la plantilla posee un pequeño grado de complejidad es por eso que obtiene una calificación de 17 que equivale a Satisfactorio. Por otra parte Ruby onRails no posee un sistema de plantillas pero tiene la ventaja de que permite la Herencia de Plantillas en la cual la aplicación mantiene cargada una plantilla en el navegador y sobre esta se van cargando las vistas necesarias, debido a que no posee un sistema propio de plantillas obtiene una calificación de 16 que equivale a Satisfactorio.*
- **Adaptación con hojas de estilo:** *La utilización de estilos en las paginas HTML es muy importante para un diseño personalizado, Django permite la administración de los recursos a través del sistema de plantillas, permite crear estilos para la adaptación de una sola página base de fácil implementación por lo que obtiene una calificación de 18 que equivale a Muy Satisfactorio. Por otro lado RoR, se adapta perfectamente con las hojas de estilo, además puede ser utilizado con una mejora a las hojas de estilo CSS llamado SCSS (de “Sassy CSS”),[...] por esto RoR obtiene una calificación Satisfactoria de 17 puntos.*
- **Fusión de código y diseño:** *Como se dijo anteriormente el sistema de plantillas que utiliza Django es ideal para los diseñadores mas no para los programadores por lo que*

de manera intencional permite la inclusión directa de código en el diseño HTML, permitiendo una correcta fusión por lo que se valora a Django con una calificación de 19 que equivale a Muy Satisfactorio. Mientras que RoR también permite esta fusión, pero resulta un tanto complejo debido a que se debe crear muchas vistas lo cual en base a lo experimentado hace el trabajo más tedioso, y por tanto se requiere algo más de tiempo, por estas razones se obtiene una calificación de 17 puntos equivalente a Satisfactorio. [...]

- **Generación automática de formularios:** Django permite una generación automática de formulario a partir de los modelos lo cual brinda un ahorro en el tiempo y facilita la reutilización, los formularios se generan creando modelos de tipo formulario y se generan automáticamente tomando los atributos de los modelos, es por esto que obtuvo una calificación de 19 que equivale a Muy Sobresaliente. Por otra parte RoR igualmente realiza la generación de los formularios de forma automática[...], pero la forma automática que hace esta tarea tiene su desventaja ya que se crea el formulario a la manera de RoR y en idioma inglés, lo que obliga a que tengamos que personalizar el formulario, una tarea no compleja pero que va a requerir un poco más de tiempo, por lo mencionado RoR obtiene un valor Muy Satisfactorio de 18 puntos.[...]
- **Validación de formularios:** La validación de formularios es muy importante a nivel de aplicación, Django permite una validación automática de la información que se ingresa en el formulario debido a que el formulario se maneja como un objeto creado a partir de una clase de tipo FORM, este objeto presenta propiedades y métodos que permiten validar de una manera sencilla y automática el formulario de acuerdo a tipo de dato que se requiere que ingrese por lo que Django obtiene una calificación de 20 que equivale a - 116- Muy Satisfactorio. Mientras que en RoR la validación también se lo hace de forma sencilla escribiendo simples líneas de código en el Modelo correspondiente a la tabla de la BD, a diferencia de Django RoR realiza la validación en el modelo, lo cual le hace obtener la calificación de 19 siendo muy satisfactorio.[...]
- **Herencia de plantillas:** El sistema de plantillas de Django permite crear una plantilla base de la cual permite heredar todas las propiedades de la misma hacia sus plantillas hijas, permitiendo incluir archivos HTML y bloques de contenedores, lo cual nos beneficia al momento de crear rápidamente distintas páginas, razones que le permiten obtener una calificación de 20 puntos que equivale a Muy Satisfactorio. En RoR existe una similitud en este sentido ya que igualmente se mantiene una plantilla de base y sobre esta se van cargando pedazos de código HTML, y otros con código mixto Ruby que en este caso son las Vistas, mismas que son invocadas con simples líneas de código, es por esto que se le ha calificado con 19 siendo Muy Satisfactorio. [...]
- **Tamaño de la Aplicación:** Este indicador es muy importante ya que permite obtener una clara conjetura de la reutilización a la baja utilización de Bytes para realizar las mismas funciones con un tamaño de aplicación que se encuentra en el rango de 0 a 1

*MB, Django obtiene una calificación de 20 puntos. Mientras tanto que Ruby onRails presenta características en la creación de la aplicación que generan archivos los cuales incrementan el tamaño de la aplicación haciéndola menos ágil, teniendo esta un tamaño de aplicación mayor a 3MB, por lo que obtiene una calificación de 10 puntos que equivale a No Satisfactorio. [...]*

- **Validación de datos:** *Django permite validar sus datos automáticamente mediante la utilización de vistas propias del framework sin tener que crearlas de manera manual indicando mediante una plantilla el error en la validación de los usuarios razones por la cual obtiene una calificación de 20 puntos que equivale a Muy Satisfactorio. En este sentido RoR difiere ya que las validaciones las realiza en el Modelo, es aquí en donde se deben añadir líneas de código para realizar las validaciones las cuales funcionan muy bien por eso su calificación de 19, Muy Satisfactorio.[...]*
- **Costo:** *Este indicador es muy importante ya que nos representa la rentabilidad en el desarrollo de un producto de software, Django y Ruby onRails son herramientas de software libre con similar licenciamiento en lo que se diferencia es en su tiempo de implementación por lo que Django presenta una ventaja sobre RoR, por lo que obtiene una calificación de 20 puntos contra 19 puntos de RoR, que equivalen a Muy Satisfactorio.[...]*
- **Entorno de Desarrollo:** *Existen multiples IDEs como entornos de desarrollo los cuales son muy bien utilizados por Django, debido a esa compatibilidad de entornos Django obtiene una calificación de 19 puntos que equivale a Muy Satisfactorio. Por otra parte en RoR si bien existen varios IDEs los que mejor se adaptan a este framework son de pago, y el mejor de los gratuitos (NetBeans) la empresa propietaria ha dejado de seguir dando soporte para esta Tecnología, si bien aun existe todavía los plugins para su utilización a medida q vayan saliendo mas versiones de RoR este quedará obsoleto, razones que le dan una calificación de 15 equivalente a Satisfactorio. [...]*
- **Tiempo de Instalación:** *Este indicador es fundamental debido a la facilidad que presenta cada tecnología respecto a su instalación, debido a que el tiempo y facilidad que presta Django siendo esta inferior a los 10 minutos obtiene una calificación de 20 puntos que equivale a Muy Satisfactorio. Por otro lado Ruby onRails presenta ciertas complejidades que incrementan el uso del tiempo en su instalación siendo esta realizada en el rango de 21 a 30 minutos teniendo una calificación de 10 puntos que equivale a Poco Satisfactorio. [...]*

### **Desventajas frente a RoR**

- **Desempeño con la base de datos:** *El desarrollo de los modelos en los respectivos frameworks, nos permitió determinar el desempeño que posee cada tecnología con la*

base de datos. Puesto que el tiempo de implementación de los modelos en Django estuvo en el rango de 6 a 10 minutos obtuvo una calificación de 3 puntos que equivale a Satisfactorio, no obstante Ruby onRails presenta una mejor manipulación y facilidad en la creación de los modelos siendo estos desarrollados en un rango de 1 a 5 minutos por lo que RoR obtiene una calificación de 4 puntos que equivale a Muy Satisfactorio.[...]

- **Manejo visual de componentes de formulario:** Los componentes de los formularios son manejados como componentes de un objeto, dependiendo de la implementación del formulario, pero al no contar Django con un entorno que permita gestionar de manera visual los componentes de un formulario obtiene una calificación de 10 que equivale a No Satisfactorio. Al igual RoR, tampoco posee esta característica por lo tanto obtiene la misma calificación. [...]
- **Herencia de clase base:** El framework Django en la implementación de los modelos permite fácilmente heredar las propiedades y atributos desde la clase base MODELS, posibilitando la reutilización de objetos, atributos y propiedades que ya se encuentran implementadas por lo que obtiene una calificación de 18 que equivale a Muy Satisfactorio. En cambio RoR y debido a que el lenguaje mismo Ruby es puramente orientado a objetos permite la herencia inclusive de las características de cualquier tipo de dato, igualmente en el framework la herencia de la clase base (Active Record), se la utiliza en todas las clases de la aplicación, con lo mencionado RoR obtiene una puntuación de 19, siendo Muy Satisfactorio.[...]
- **Manejo de cookies:** La manipulación de las cookies ya son fijadas por el propio framework, lo cual permite a un navegador u otro saber si es el mismo usuario que esta realizando la petición, en Django es increíblemente sencillo ya que cada objeto de petición tiene un objeto COOKIES que actúa como un diccionario lo cual lo podemos usar para leer cualquier cookie desde el navegador por lo que obtiene una calificación de 18 que equivale a Muy Satisfactorio. Por otro lado en RoR esto es configurado por el mismo componente de autenticación, las cookies son manejadas por ActionController, pero como se menciona, esto ya el framework lo hace de forma automática la calificación obtenida es de 19 equivalente a Muy Satisfactorio.[...]

Después del análisis de los indicadores de productividad propuestos, mediante la aplicación de experimentación y observación en la construcción de los módulos de prueba, se obtuvieron como resultado valores cuantitativos que reflejan el desenvolvimiento de cada una de las tecnologías de acuerdo a sus características en función a la productividad de desarrollo. Estos valores son el resultado de las pruebas efectuadas tanto a los módulos de prueba 1, 2 y 3, tomando en consideración factores como el tiempo de desarrollo, implementación, tamaño de la aplicación, facilidad de uso. (Cumba y Barreto, 2012, pp.110-136)

En la siguiente tabla se demuestra con los resultados la leve pero notable superioridad de Django frente al framework que compite en su categoría. Cabe señalar que los indicadores del estudio en los que empatan no han sido considerados como ventajas o desventajas.

**Tabla 2-2:** Tabla de resultados entre frameworks Django y RoR.

CLASIFICACION	PARAMETROS	INDICADORES	Django	RoR
		formularios		
		<ul style="list-style-type: none"> <li>Validación de formularios</li> </ul>	20	19
		<ul style="list-style-type: none"> <li>Manejo visual de componentes de formulario</li> </ul>	10	10
Principio DRY Don't repeat yourself	Reutilización	<ul style="list-style-type: none"> <li>Herencia de clases base</li> </ul>	18	19
		<ul style="list-style-type: none"> <li>Herencia de plantillas</li> </ul>	20	19
		<ul style="list-style-type: none"> <li>Personalización de código heredado</li> </ul>	17	17
		<ul style="list-style-type: none"> <li>Tamaño de la aplicación</li> </ul>	20	10
Seguridad	Seguridad de Aplicación	<ul style="list-style-type: none"> <li>Variable uso de sesiones</li> </ul>	19	19
		<ul style="list-style-type: none"> <li>Manejo de cookies</li> </ul>	18	19
		<ul style="list-style-type: none"> <li>Encriptación de datos</li> </ul>	19	19
		<ul style="list-style-type: none"> <li>Validación de datos</li> </ul>	20	19
Producto	Madurez de producto	<ul style="list-style-type: none"> <li>Especificaciones</li> </ul>	17	17
		<ul style="list-style-type: none"> <li>Licenciamiento</li> </ul>	19	19
		<ul style="list-style-type: none"> <li>Costo de aplicaciones</li> </ul>	18	18
		Instalación	<ul style="list-style-type: none"> <li>Entorno de</li> </ul>	19

		desarrollo		
		• Tiempo de instalación	20	10
<b>Totales</b>			382	352

**Realizado por:** Paul Cumba & Byron Barreto, 2012.

**Fuente:** (ANÁLISIS DE PYTHON CON DJANGO FRENTE A RUBY ON RAILS PARA DESARROLLO ÁGIL DE APLICACIONES WEB. CASO PRÁCTICO: DECH., 2012, pp.136-138)

### 2.2.6 Filosofía Django

Django al provenir de Python hereda mucho de su filosofía, por lo cual desarrollar un sistema con Django resulta una tarea muy enriquecedora en términos de Programación Orientada a Objetos, ya que resulta muy didáctico comprender la forma correcta de usar los objetos y hacerlos reusables, además está permitido realizar modificaciones en su código fuente por si se desea contribuir o personalizar el entorno de trabajo.

El Libro de Django en su traducción al español realizada por Saul Garcia describe de mejor manera sobre la lógica de este framework:

*Los inicios de Django también marcan ciertas limitaciones intencionales, ya que nace de la necesidad de generar páginas de contenido en un tiempo corto, el desarrollo en Django usa un sistema de plantillas para agilizar el trabajo de diseño con lo cual se hace necesario (como en todo proyecto) una vinculación entre el diseño y el modelado de negocio, en Django dicha vinculación es débil (a propósito) basándose en las siguientes sentencias:*

- *La lógica de negocios debe ser separada de la presentación lógica. Vemos al sistema de plantillas como una herramienta que controla la presentación y la lógica relacionado a esta — y eso es todo. El sistema de plantillas no debería admitir funcionalidad que vaya más allá de este concepto básico.*
- *Por esta razón, es imposible llamar a código Python directamente dentro de las plantillas de Django. Todo "programador" está fundamentalmente limitado al alcance de lo que una etiqueta puede hacer. Es posible escribir etiquetas personalizadas que hagan cosas arbitrarias, pero las etiquetas de Django intencionalmente no permiten ejecutar código arbitrario de Python.*

- *La sintaxis debe ser independiente de HTML/XML. Aunque el sistemas de plantillas de Django es usado principalmente para producir HTML, este pretende ser útil para formatos no HTML, como texto plano. Algunos otros lenguajes de plantillas están basados en XML, poniendo toda la lógica de plantilla con etiquetas XML o atributos, pero Django evita deliberadamente esta limitación. Requerir un XML válido para escribir plantillas introduce un mundo de errores humanos y mensajes difícil de entender, y usando un motor de XML para parsear plantillas implica un inaceptable nivel de overhead en el procesamiento de la plantilla.*
- *Los diseñadores se supone que se sienten más cómodos con el código HTML. El sistema de plantillas no está diseñado para que las plantillas necesariamente sean mostradas de forma agradable en los editores WYSIWYG tales como Dreamweaver. Eso es también una limitación severa y no permitiría que la sintaxis sea tan clara como lo es. Django espera las plantillas de los autores para estar cómodo editando HTML directamente.*
- *Se supone que los diseñadores no son programadores Python. El sistema de plantillas de los autores reconoce que las plantillas de las páginas web son en al mayoría de los casos escritos por diseñadores, no por programadores, y por esto no debería asumir ningún conocimiento de Python. Sin embargo, el sistema también pretende acomodar pequeños grupos en los cuales las plantillas sean creadas por programadores de Python. Esto ofrece otro camino para extender la sintaxis del sistema escribiendo código Python puro. (Más de esto en el Capítulo 10).*
- *El objetivo no es inventar un lenguaje de programación. El objetivo es ofrecer sólo la suficiente funcionalidad de programación, tales como ramificación e iteración, que son esenciales para hacer presentaciones relacionadas a decisiones.*

*Como resultado de esta filosofía, el lenguaje de plantillas de Django tiene las siguientes limitaciones:*

- *Una plantilla no puede asignar una variable o cambiar el valor de esta. Esto es posible escribiendo una etiqueta personalizada para cumplir con esta meta [...], pero la pila de etiquetas de Django no lo permite.*
- *Una plantilla no puede llamar código Python crudo. No hay forma de ingresar en "modo Python" o usar sentencias puras de Python. De nuevo, esto es posible creando*

*plantillas personalizadas, pero la pila de etiquetas de Django no lo permiten.*(Garcia, 2015, pp.60-61)

## **2.3 Materialize (HTML5, CSS3, JavaScript)**

Materialize se define en su página web como “un framework web front-end moderno y responsiva creado basado en Material Design”(Wang et al., 2014), ofrece la oportunidad de diseñar una página web con objetos como componentes, transiciones, forms, imágenes y colores basados, todo basado en HTML5, CSS3 y JavaScript, que, en conjunto forman una herramienta de diseño muy completa.

### **2.3.1 Material Design**

Para conocer el trasfondo de Materialize se debe entender el concepto de Material Design, el cual fue creado por Google como una normativa de diseño para el diseño de las interfaces para el sistema operativo Android.

De acuerdo con el blog “Material Design” (Duarte citado en Marina, 2015) en donde cita al vicepresidente de diseño de Google: “... a diferencia del papel, nuestro material digital se puede ampliar y redimensionar de manera inteligente. Material Design tiene superficies físicas y bordes. Las escenas y sombras proporcionan significado sobre lo que se puede tocar y cómo se va a mover.”



## **Figura 2-2:** Material Design Concept.

**Realizado por:**RichFulcher(2015)

Su nombre proviene del concepto de las cosas materiales, es decir que está enfocado a diseñar objetos con relieves, sombras, bordes, movimientos, curvas y picos que pretenden generar una sensación en el usuario y ayudar a dar significado a cada elemento como se puede observar en el demo mostrado en la **Figura 2-2**. Material Design es una normativa de diseño llevada a otro nivel, como una filosofía para el diseño web, el equivalente a la Programación Orientada a Objetos para el diseño orientado a materiales.

### **2.3.2 Características**

Además de ser de código abierto la comunidad de Materialize destaca algunas características:

- *Permite (al igual que que otros frameworks como Bootstrap) crear interfaces que se adapten a distintas resoluciones y navegadores.*
- *Crear menú lateral abierto o desplegable en función de la resolución.*
- *Animaciones con filosofía Material Design.*
- *Crear diseños con Material Design como botones, tarjetas, colecciones, iconos, barras de navegación, etiquetas, 'toast', modales, etc.*
- *Integra Parallax, Pushpin y scrollspy para hacer un diseño más atractivo. (Guillem, 2015)*

### **2.3.3 Ventajas y desventajas**

Materialize al ser una tecnología relativamente nueva se podría decir que no ha generado mayor impacto en la comunidad desarrolladora, por otro lado su tecnología base: Material Design está dando mucho de qué hablar, por lo que se pueden considerar además de sus ventajas y desventajas (propias de Materialize), las de usar un framework front-end según el portal web Aula Informativa:

#### ***Ventajas de utilizar una framework CSS***

- *1. Agilización del proceso de desarrollo: Una framework CSS te permite desarrollar más rápido mediante el uso de código base y tareas comunes, de modo que no tienes que empezar desde cero. Si estás trabajando con un equipo, todos tendrán el mismo código, así que la colaboración en equipo también se beneficia de una framework CSS.*

- *2. Facilita la funcionalidad en diversos navegadores: Una framework CSS ya viene programada para que el resultado (es decir, tu aplicación o sitio web) final luzca similar en todos los navegadores, de modo que puedes concentrarte en personalizar y crear contenido en vez de preocuparte en modificar la apariencia base.*
- *3. Te permite trabajar con layout simétrica: Una framework CSS con sistemas de cuadrículas asigna columnas con anchos predefinidos, de modo que puedes concentrarte en la creación de contenido en vez de la alineación de bloques de texto.*
- *4. Refuerza buenos hábitos de diseño: Una framework CSS te provee una serie de selectores que puedes utilizar para tus sitios web y aplicaciones, de este modo el diseño se hace congruente ya que no debes adivinar ni acordarte que hiciste en proyectos pasados.*

#### ***Desventajas de utilizar una framework CSS***

- *1. Limita tu libertad: En general, sólo se utiliza una framework CSS para todos tus proyectos ya que aprender una tras otra realmente no es muy práctico. Ya que una framework CSS tiene una serie de retículas, selectores y otro tipo de código limita tu diseño en cuanto al tamaño de los elementos y el estilo.*
- *2. Añade código extra: Es poco probable que emplees todas las características de una framework CSS por lo que definitivamente tus aplicaciones tendrán código que no necesitas. Esto podría ser un problema si deseas que tu aplicación sea súper ligera y cada byte cuenta.*
- *3. Te fuerza a usar la semántica propia de la framework: Al usar una framework CSS tienes que acostumbrarte a trabajar con la semántica de ésta. En general, esto no suele ser un problema para muchos desarrolladores que trabajan solos. Sin embargo, si estás colaborando con varias personas deberás asegurarte que todos tengan el mismo nivel de conocimiento sobre la framework, lo que es un escenario ideal, por supuesto. Así que debes esperar que tus compañeros se familiaricen con la framework primero.*
- *4. Podría ser una pérdida de tiempo: Si ya estás acostumbrado a trabajar y ordenar tu código de cierta manera y debido a una petición de cliente o a un proyecto en equipo te ves obligado a trabajar con cierta framework CSS, obviamente primero debes aprender cómo funciona. Esto podría ser tiempo valioso que podrías haber invertido en*

*comenzar a codificar sin el uso de una framework CSS. (Blog Aula Informativa, 2014)*

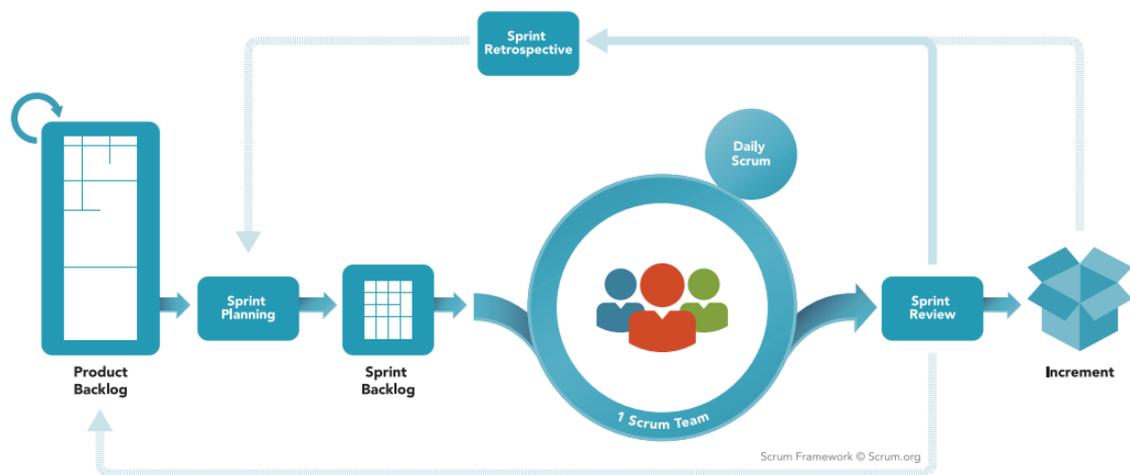
## **2.4 Métodos y técnicas**

### **2.4.1 Metodología Ágil Scrum.**

Scrum es una metodología ágil muy popular entre la comunidad de desarrolladores que ofrece una gran flexibilidad frente al cambio y combina técnicas que la hacen una herramienta completa a la vez que se puede implementar con otras herramientas CASE para adaptarla a las necesidades del proyecto o el equipo de programadores. Esta metodología es muy versátil ya que permite al gestor del proyecto entrar en contacto con todos los involucrados en el proyecto de manera organizada, para tratar el tema en este apartado se ha utilizado “La Guía de Scrum” escrito por sus autores Ken Schwaber y Jeff Sutherland:

*“Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar.”(Ken S., 2013, p.4)*

# SCRUM FRAMEWORK



**Figura 2-3:** Scrum en acción.

**Realizado por:** Ken Schwaber and Jeff Sutherland (2012)

## 2.4.1.1 Filosofía ágil en Scrum

*“Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación”. (Ken S., 2013, p.4)*

Ken Schwaber y Jeff Sutherland, autores de Scrum, aseguran que esta metodología nace de la experiencia en el desarrollo de muchos proyectos que ayudaron a definir cuáles son las verdaderas necesidades del gestor de proyecto y su grupo desarrollador, incluyendo además al cliente entre sus actividades de planificación.

Estos pilares fundamentales son descritos por Schwaber y Sutherland en la guía de Scrum (2013, p.5):

***“Transparencia***

*Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se está viendo.*

### ***Inspección***

*Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones. Su inspección no debe ser tan frecuente como para que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos, en el mismo lugar de trabajo.*

### ***Adaptación***

*Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado deben ser ajustados. Dicho ajuste debe realizarse cuanto antes para minimizar desviaciones mayores.”*

#### ***2.4.1.2 Equipo Scrum***

En Scrum se especifican roles que cumplen ciertas tareas, estos roles son ocupados por quienes están a cargo de que el proyecto se ejecute, quienes lo financian y el equipo encargado del desarrollo.

*“El Equipo Scrum consiste en un Dueño de Producto (ProductOwner), el Equipo de Desarrollo (DevelopmentTeam) y un Scrum Master. Los Equipos Scrum son autoorganizados y multifuncionales. Los equipos autoorganizados eligen la mejor forma de llevar a cabo su trabajo” (Ken S., 2013, p.5)*

### 2.4.1.3 Eventos de Scrum

Como se muestra en la **Figura 2-3**; en Scrum se implementan eventos que deben ser gestionados, estos sirven para implementar el principio de inspección en Scrum:

- **Reunión de Planificación del Sprint (Sprint Planning Meeting):** Inicia el calendario del Sprint, reúne al Equipo Scrum para una conversación como lluvia de ideas con el objetivo de visualizar el producto y sus funcionalidades principales, de las cuales se extenderá una lista de requerimientos con los cuales se iniciaría el trabajo de desarrollo además del Goal Sprint. Según sus autores Schwaber y Sutherland el tiempo máximo para esta reunión es de 8 horas por Sprint de 1 mes (2013), según esta guía se debe responder a las siguientes preguntas: **¿Qué puede entregarse en el Incremento resultante del Sprint que comienza?** y **¿Cómo se conseguirá hacer el trabajo necesario para entregar el Incremento?**

**Sprint Goal:** Según Schwaber es un objetivo general para el Sprint, se delimita con el Equipo Scrum en las Reuniones de Planificación, resume las tareas programadas para el Sprint en cuanto al producto se refiere y ofrece una guía a los desarrolladores para dar un enfoque general de lo que se está haciendo para el Sprint en transcurso.

*Duración:* 8 horas para sprints de 1 mes

*Integrantes:* Product Owner, Master Scrum y Scrum Team

*Recursividad:* Mensual

- **Scrum Diario (DailyScrum):** Son reuniones cortas para sincronizar las actividades entre compañeros al final de cada jornada de trabajo. En su libro mencionan algunas preguntas a resolver: “¿Qué hice ayer que ayudó al Equipo de Desarrollo a lograr el Objetivo del Sprint?, ¿Qué haré hoy para ayudar al Equipo de Desarrollo a lograr el Objetivo del Sprint?, ¿Veo algún impedimento que evite que el Equipo de Desarrollo o yo logremos el Objetivo del Sprint?” (Schwaber, 2014, pp.12).

*Duración:* 15 minutos

*Integrantes:* Master Scrum y Scrum Team

*Recursividad:* al inicio de cada jornada

- **Revisión del Sprint (Sprint Review):** Se trata de una reunión para revisar el Sprint y dar conclusiones y recomendaciones para las siguientes reuniones de planificación del siguiente Sprint. Esta revisión tiene las siguientes características o elementos:
  - *“Los asistentes son el Equipo Scrum y los interesados clave invitados por el Dueño de Producto;*
  - *El Dueño de Producto explica qué elementos de la Lista de Producto se han “Terminado” y cuales no se han “Terminado”;*
  - *El Equipo de Desarrollo habla acerca de qué fue bien durante el Sprint, qué problemas aparecieron y cómo fueron resueltos esos problemas;*
  - *El Equipo de Desarrollo demuestra el trabajo que ha “Terminado” y responde preguntas acerca del Incremento;*
  - *El Dueño de Producto habla acerca de la Lista de Producto en el estado actual. Proyecta fechas de finalización probables en el tiempo basándose en el progreso obtenido hasta la fecha (si es necesario);*
  - *El grupo completo colabora acerca de qué hacer a continuación, de modo que la Revisión del Sprint proporcione información de entrada valiosa para Reuniones de Planificación de Sprints subsiguientes*
  - *Revisión de cómo el mercado o el uso potencial del producto podría haber cambiado lo que es de más valor para hacer a continuación; y,*
  - *Revisión de la línea de tiempo, presupuesto, capacidades potenciales y mercado para la próxima entrega prevista del producto” (Schwaber, 2014, pp.13).*

*Duración:* 4 horas

*Integrantes:* ProductOwner, Master Scrum y ScrumTeam

*Recursividad:* al término de cada Sprint

- **Retrospectiva del Sprint (Sprint Retrospective):** La Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint. (Schwaber, 2013, p.14)

El propósito de la Retrospectiva de Sprint es:

- *“Inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas;*
- *Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras; y,*

- *Crear un plan para implementar las mejoras a la forma en la que el Equipo Scrum desempeña su trabajo” (Schwaber, 2013, p.14).*

*Duración:* 3 horas

*Integrantes:* Master Scrum y ScrumTeam

*Recursividad:* antes de planificación del siguiente Sprint

### 2.4.1.3 Herramientas Scrum

#### **ProductBacklog (Lista de Producto)**

Esta lista consta de todas las tareas necesarias para llevar a su término el producto final, esta lista es definida en la primera reunión de planificación y es redefinida si es necesario en planificaciones posteriores. La lista nunca está completa en un inicio por lo que es necesario estudiar bien los requerimientos para desglosar todas las tareas, esta lista se caracteriza precisamente por ser dinámica y ligada al producto de forma permanente.

Esta lista requiere de refinamiento continuo por parte del Equipo Scrum, los cuales deberán revisar y analizar los elementos de la lista durante su desarrollo sin sobrepasar el 10% de su dedicación al proyecto, el ProductOwner se encarga de mantener la lista priorizada y si lo ve necesario puede modificar la lista en coordinación con el Scrum Master. A medida que avanza el desarrollo se agregan más detalles a las tareas de mayor prioridad que por lo general están por encima de las demás tareas de menor prioridad y menor detalle.

#### **Sprint Backlog (Pendientes del Sprint)**

Esta lista está constituida por los elementos de la Lista de Producto seleccionadas para el Sprint, define el trabajo a realizar durante el sprint para completar el objetivo o Sprint Goal. Incluye además un plan para “terminar” el producto con la funcionalidad propuesta por el objetivo, se debe buscar una estrategia para seleccionar y ejecutar las tareas pendientes en la lista hasta “terminar”.

Nota: Se debe tomar muy en cuenta el significado de la palabra “terminado”, es necesario manejar un concepto claro de cuando uno de los elementos de la lista esta cumplido a cabalidad, desde un principio se debe definir los estados que puede tener un elemento hasta llegar a “terminado”.

#### **Incremento**

El Incremento es la suma de todos los elementos de la Lista de Producto completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores. Al final de un Sprint, el nuevo Incremento debe estar terminado, lo cual significa que está en condiciones de ser utilizado y que cumple la Definición de “Terminado” del Equipo Scrum, como se describe en la guía: “El incremento debe estar en condiciones de utilizarse sin importar si el Dueño de Producto decide liberarlo o no”(Schwaber, 2013, p.17).

#### **2.4.2 Investigación**

Para la obtención de datos sobre el entorno en donde se desenvolverá la aplicación se debe usar métodos de investigación que ayuden a mejorar los resultados esperados, es por ello que para investigar los requerimientos adecuados para el desarrollo del presente proyecto se ha optado por el uso de encuestas.

##### **2.4.2.1 Encuestas**

*“Una encuesta es un estudio observacional en el cual el investigador no modifica el entorno ni controla el proceso que está en observación, obteniendo los datos a partir de la formulación de un conjunto de preguntas normalizadas dirigidas a una muestra representativa o al conjunto total de la población estadística en estudio, con el fin de conocer estados de opinión, características o hechos específicos”.*(Johnson & Kuby, 2013)

Para el presente proyecto ha realizado una encuesta hacia la una población de servidores turísticos y otra a parte de la población del cantón Morona, además de enfocar preguntas sobre temas turísticos se añaden algunas en búsqueda de indicios que contrasten con los resultados una vez implementada la aplicación, ayudando posteriormente a medir el impacto del uso de un mapa web de eventos.

#### **Fórmula para calcular el tamaño de una muestra**

Se utilizó muestreo probabilístico tomando en cuenta varios parámetros: el error muestral, el nivel de confianza y la varianza poblacional. Como se conoce el tamaño de la población se utilizó la siguiente ecuación:

$$n = \frac{N * Z^2 * p * q}{d^2 * (N - 1) + Z^2 * p * q}$$

**Ecuación:** Tamaño de muestra cuando se conoce la población.

**Fuente:** Ing. Mariela Torres(s.f.)

En donde:

**n**= Tamaño de la muestra

**N**= Universo de la población

**Z**= Nivel de confianza

**p**= Probabilidad de Éxito

**q**= Probabilidad de Fracaso

**d**= Precisión (error máximo admisible en términos de proporción)

### 2.4.3 Base de datos

Se necesita una forma de representar un usuario en el sistema, es decir, guardar su información en tablas, a estas representaciones abstractas las definimos como modelos (clases en términos de programación) en Django, se necesita un sistema (*SGBD*) que se encargue de almacenar esas tablas y que gestione las transacciones de consulta y actualización desde el sistema a desarrollar, para ello debe haber compatibilidad ya que se trata de un sistema que trabaja de forma virtualmente aislada. Django soporta la conexión con algunos de los más usados entre la comunidad de programadores: PostgreSQL, SQLite, MySQL, Oracle y por supuesto existen módulos desarrollados para dar soporte a otros motores de bases de datos según Cumba y Barreto (2012, p.62).

Ya que se prioriza las alternativas basadas en software libre, se ha escogido *PostgreSql* como Sistema Gestor de Base de Datos (*SGDB*).

*“Django es un framework donde cada uno de los modelos de datos utilizados tiene su correspondencia en una tabla de la Base de Datos, por ello Django está preparado*

*para soportar las más comunes en el mercado simplemente cambiando la variable DATABASE\_ENGINE en el settings.py” (Cumba & Barreto, 2012)*

### **CAPÍTULO III:**

#### **DESARROLLO DE LA APLICACIÓN MÓVIL MAPE**

Para el desarrollo de la aplicación web Mape se involucra el uso de tecnologías de desarrollo ágil, paradigmas orientados a objetos y filosofías inherentes al software libre. Se hace uso de herramientas disponibles en la internet de forma gratuita; API Mapbox, Django, Python, MaterializeCss, jQuery, SublimeText, Google Fonts. El proceso de desarrollo con dichas herramientas se reduce considerablemente ya que cubren muchos de los aspectos que hasta hace algunos años dependían de mucho esfuerzo de programación y costos elevados en aspectos como la robustez, seguridad y compatibilidad.

La aplicación ofrece una alternativa gratuita para la publicidad de eventos de cualquier índole, con un valor agregado en el uso de mapas interactivos con lo cual los usuarios podrán encontrar la ubicación exacta de eventos, productos, servicios, localidades que se ofrezcan a través de la aplicación. Para determinar el enfoque que se le daría a la aplicación se ha realizado una encuesta que ayudase a encontrar una problemática a resolver. Para ver más detalles sobre la información de este capítulo puede buscar entre la documentación anexada.

### 3.1 Planificación del proyecto

#### 3.1.1 Cronograma de actividades

Se presenta la planificación general para el desarrollo del proyecto de tesis incluyendo su preparación previa hasta el inicio del desarrollo basado en la metodología de desarrollo Scrum.



**Figura 3-1:** Cronograma de actividades

Realizado por: Juan Solórzano

#### 3.1.2 Análisis de resultados de encuesta a Usuarios

En total se han realizado 40 encuestas para los servidores turísticos y 43 encuestas a los potenciales usuarios turísticos. Antes de iniciar las encuestas se realizó una pregunta piloto para determinar el nivel de confianza que tendrán las respuestas respecto al turismo en el cantón Morona. El resultado podría determinar qué nivel de confianza se usaría para encontrar un tamaño de muestra poblacional.



**Figura 3-2: Usuarios\_Pregunta\_Piloto**

Realizado por: Juan Solórzano

Para identificar la problemática a resolver con la implementación de una aplicación web, primero se ha realizado una encuesta que ayude a comprobar que efectivamente existe una oportunidad de mejorar algo. En este caso se ha realizado dos encuestas dirigidas a una población de posibles usuarios de una aplicación web en sus móviles y a los servidores turísticos del cantón Morona respecto a sus necesidades comerciales. Los resultados obtenidos aplicando una fórmula para el cálculo de las muestras poblacionales en total se muestran en la siguiente tabla:

**Tabla 3-1:**Tamaño de las muestras poblacionales.

	Tamaño de la muestra para encuestas a:	
	Comerciantes/Servidores Turísticos	Usuarios/Consumidores
<i>N</i>	179	18.984
<i>Z<sub>α</sub></i>	1,4	1,65
<i>p</i>	0,5	0,8

<i>q</i>	0,5	0,2
<i>d</i>	0,1	0,1
<b>Tamaño de la muestra (n)</b>	<b>38,63</b>	<b>43,46</b>

Fuente: Datos de muestra

El tamaño de la población urbana ha sido obtenido del CENSO DE POBLACIÓN Y VIVIENDA (CPV-2010)

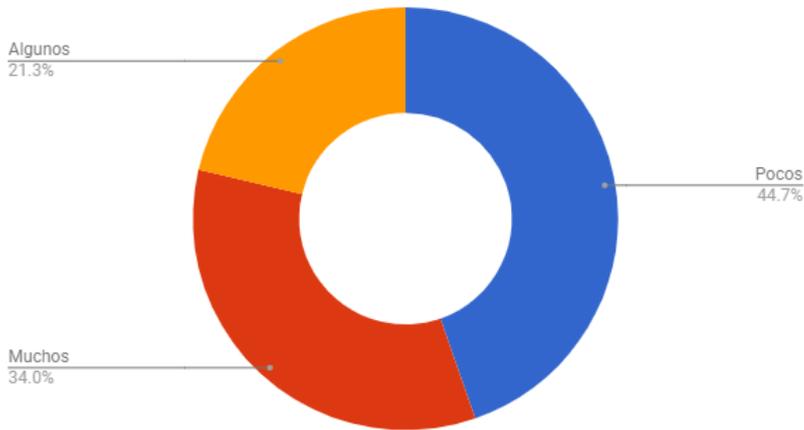
**Encuesta Servidores Turísticos:** Para la encuesta de los servidores turísticos se ha considerado un nivel de confianza (Z) de 91% y para la probabilidad de éxito (p) se ha predeterminado el valor de 50% de probabilidad de éxito ya que no se desconoce la proporción esperada.

**Encuesta a Consumidores/Usuarios:** En el caso de los consumidores potenciales del servicio Mape son tomados de la población de redes sociales y parte urbana. El *nivel de confianza* utilizado es de 95% pero en la probabilidad de éxito (p) se ha determinado el valor de 80% para el peor de los casos, esto debido a que 9 de 10 personas respondieron positivamente a la siguiente pregunta piloto de la **Figura 3-2**.

## ENCUESTA A CONSUMIDORES

### 3.1.2.1 *Pregunta usuarios 1*

¿Conoce buenos lugares para visitar?



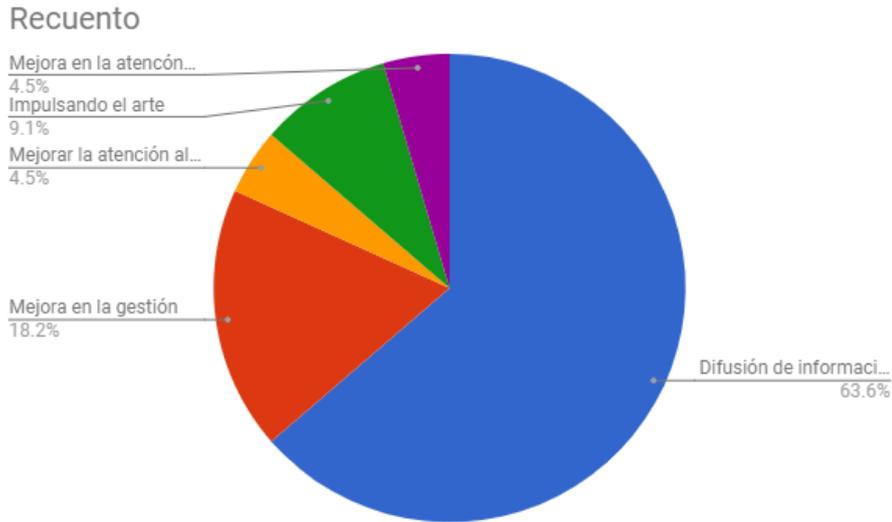
**Figura 3-3: Usuarios\_Pregunta\_1**

Realizado por: Juan Solórzano

**Interpretación:** El 55,3% de los encuestados señala conocer entre algunos y muchos lugares para visitar, lo que representa casi la mitad de las personas que solo saben de pocos lugares.

### 3.1.2.2 *Pregunta usuarios 2*

Para la siguiente pregunta se tuvieron que clasificar sus respuestas según el ámbito en el que se desempeña para poder identificar la necesidad de mayor recurrencia. **¿Cómo fomentaría usted el turismo en Macas?**



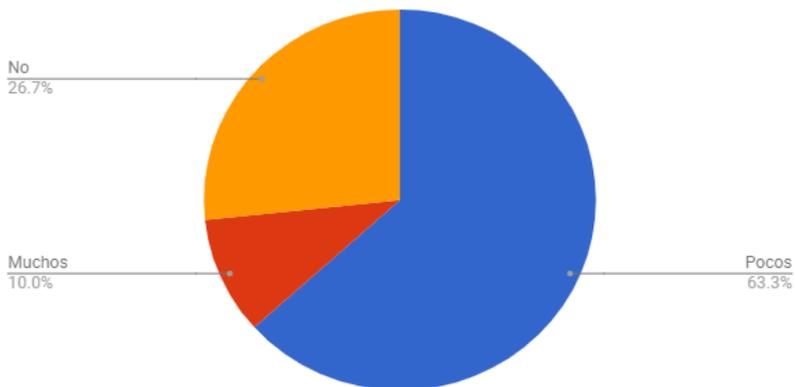
**Figura 3-4: Usuarios\_Pregunta\_2**

Realizado por: Juan Solórzano

**Interpretación:** La gran mayoría propone alternativas para la difusión de información con un 50%, mientras que el 15% empata en segundo lugar con tres de los tópicos, dejando al final con un 3% para alternativas relacionadas al arte.

### 3.1.2.3 Pregunta usuarios 3

¿Sabe usted de guías turísticas y opciones para visitar lugares atractivos?



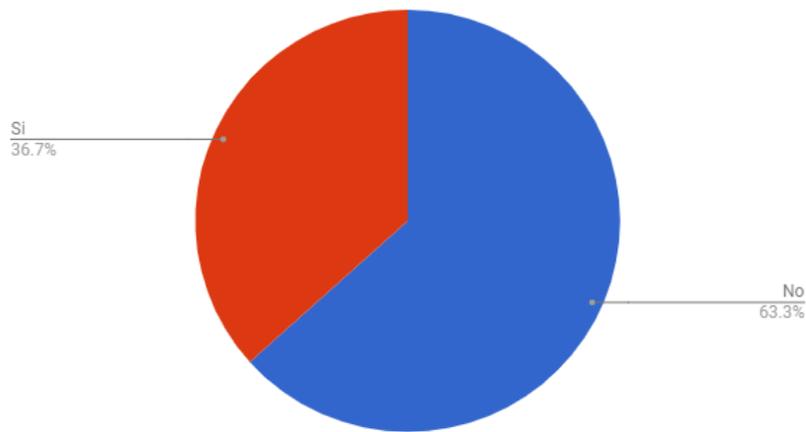
**Figura 3-5: Usuarios\_Pregunta\_3**

Realizado por: Juan Solórzano

**Interpretación:** Sólo uno de cada diez encuestados sabe de dónde obtener información turística, la gran mayoría de las personas desconoce de una guía turística, esto sin considerar la eficiencia de las guías disponibles.

#### 3.1.2.4 Pregunta usuarios 4

¿Ha utilizado alguna app turística?



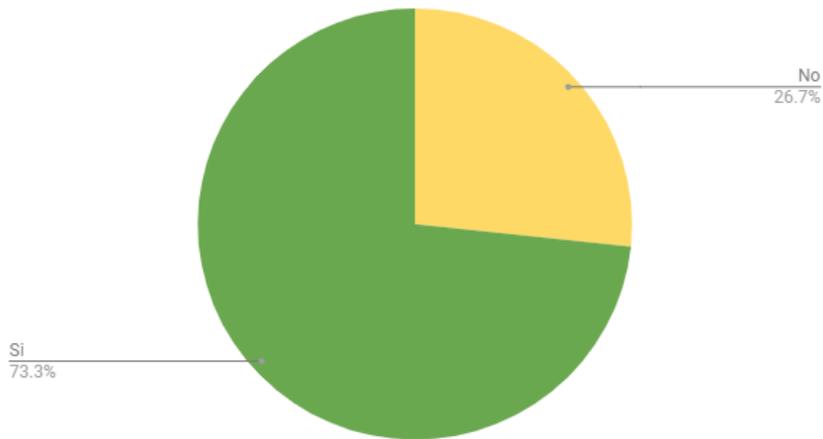
**Figura 3-6: Usuarios\_Pregunta\_4**

**Realizado por:** Juan Solórzano

**Interpretación:** El 63,3% que representa una gran mayoría; dos terceras partes de la población de encuestados que no ha consultado de una aplicación turística.

### 3.1.2.5 Pregunta usuarios 5

¿Ha usado una app o pagina web con mapa?



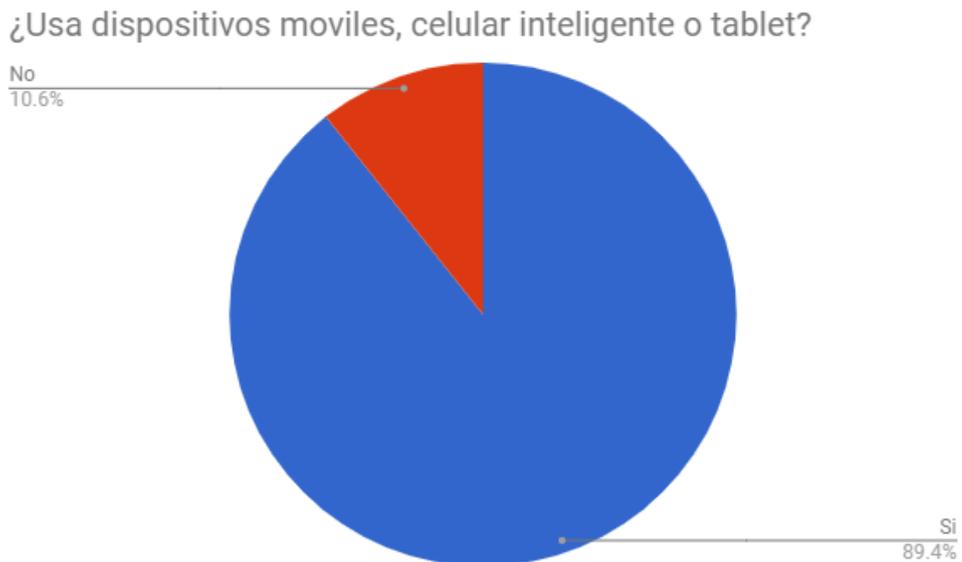
**Figura 3-7: Usuarios\_Pregunta\_5**

**Realizado por:** Juan Solórzano

**Interpretación:** El 73,3% de los encuestados ha utilizado alguna vez un mapa a través de un dispositivo. Existe gran población de personas que ha hecho uso de un mapa desde una página web o app sin embargo la pregunta anterior revela que el usuario no encontró utilidad en el ámbito turístico (a pesar de ser un mapa), se podría especular que la ubicación tiene mucha relación (no existen marcas cercanas).

## ENCUESTA A COMERCIANTES/SERVIDORES TURÍSTICOS

### 3.1.2.6 Pregunta servidores 1



**Figura 3-8:** Servidores\_Pregunta\_1

**Realizado por:** Juan Solórzano

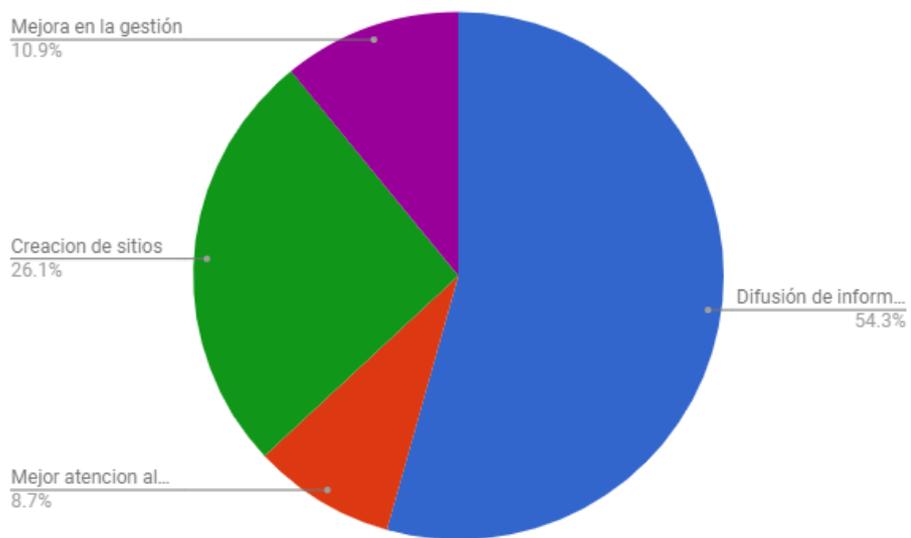
**Interpretación:** Uno de cada diez personas encuestadas en la zona urbana usa un dispositivo móvil. Este resultado confirma el gran apogeo de los teléfonos inteligentes y sobre todo nos indica que hay una plataforma física para desplegar un sistema.

### 3.1.2.7 Pregunta servidores 2

#### ¿Cómo fomentaría usted el turismo en Macas?

De igual manera se han determinado los principales tópicos a los que hacen referencia los encuestados:

- Difusión de información
- Mejora en la atención al cliente
- Creación de sitios turísticos
- Mejora en la gestión



**Figura 3-9:** Servidores\_Pregunta\_2

**Realizado por:** Juan Solórzano

**Interpretación:** De igual forma que en la encuesta para consumidores, la gran mayoría expresa la necesidad de difusión de información turística de forma eficiente.

### 3.1.2.8 Pregunta servidores 3

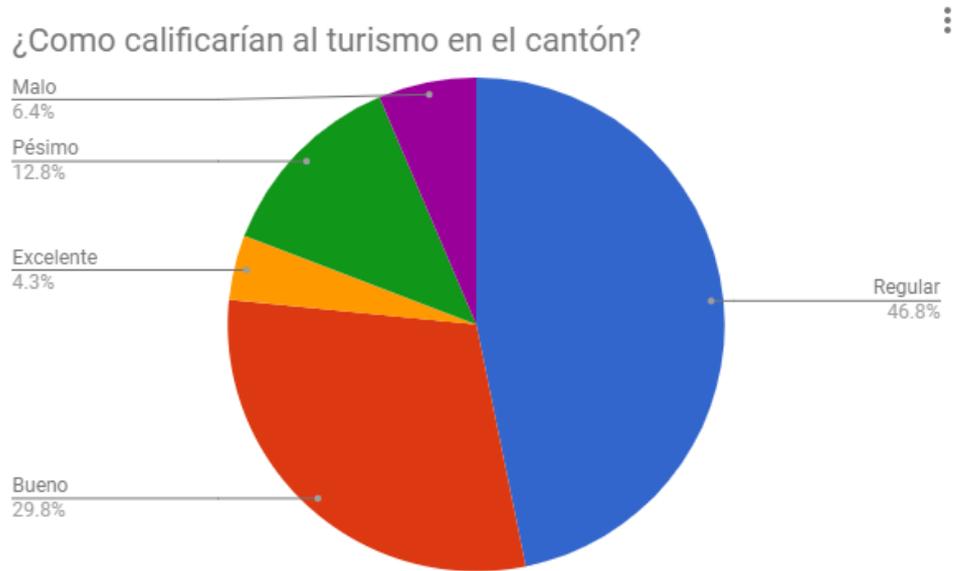


**Figura 3-10:** Servidores\_Pregunta\_3

**Realizado por:** Juan Solórzano

**Interpretación:** Las cifras difieren a las que se pueden encontrar como resultado de la anterior encuesta para los consumidores, en la que casi la mitad de los encuestados desconocen de muchos lugares turísticos, en cambio aquí el 33% sabe de muchos lugares, esto puede deberse a la diferencia de edades, sin embargo las cifras son bajas para un cantón con mucho potencial turístico.

### 3.1.2.9 Pregunta servidores 4



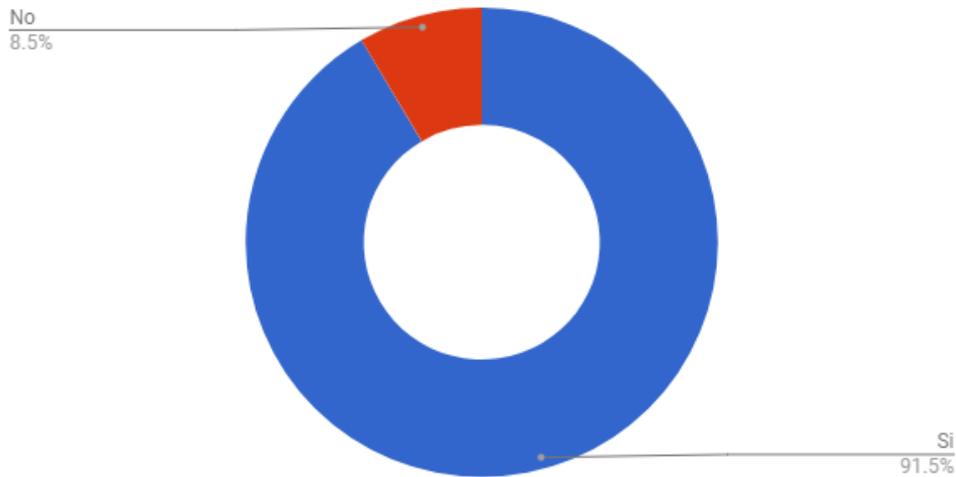
**Figura 3-11:** Servidores\_Pregunta\_4

**Realizado por:** Juan Solórzano

**Interpretación:** La mayoría de las selecciones de los calificativos con 46,8% + 6,4% + 12,8% son negativos para el turismo, lo cual refleja una discordancia con los resultados para la pregunta anterior y con la creencia del cantón es rico en el ámbito turístico; la gente reconoce los potenciales turísticos pero existe uno o varios factores que impide que aflore.

### 3.1.2.10 Pregunta servidores 5

¿Usaría una sencilla aplicación para publicar eventos e invitar a sus clientes?



**Figura 3-12:** Servidores\_Pregunta\_5

**Realizado por:** Juan Solórzano

**Interpretación:** Definitivamente la gran mayoría de los servidores turísticos están interesados en el uso de una aplicación para promocionar eventos.

### 3.1.2.11 Pregunta servidores 6

**¿Qué desearías saber de un evento?**

Se determinaron las siguientes características de un evento según lo que desearía conocer el usuario.

- **Dirección**
- **Nombre del sitio**
- **Precio**
- **Hora**
- **Fecha**
- **Duración**
- **Fotos**

Otros: *Clima, Promociones, Cronograma de Actividades, Sobre que trata el evento, Transporte.*

### 3.1.2.1 Conclusiones de las encuestas

Luego de recopilar los datos obtenidos de la encuesta se ha procedido al análisis e interpretación de la información en una visión general, con criterios objetivos sobre el turismo y las necesidades de los comerciantes, que en este caso son los servidores turísticos que se encuentran en los registros de la municipalidad.

1. El 94% de las personas reconocen que tienen “mucho o bastante” potencial turístico.
2. A pesar del dato anterior; el 44,7% representa la casi la mitad de encuestados que desconocen de sus atractivos turísticos.
3. Una de las principales necesidades expuesta por los encuestados es la falta de difusión de información por cualquier medio.
4. Una tercera parte de la muestra poblacional nunca ha utilizado una aplicación para hacer turismo, sin embargo el 73,3% ha usado un mapa mediante una página web o app; lo que indica que la mayoría de esos mapas no ha servido al ámbito turístico en el cantón morona.
5. La mayoría de los representantes con un 89% de los sitios turísticos tienen disponible un teléfono inteligente para su gestión.
6. Se restablece una cifra alarmante para el tema de difusión de información en el ámbito turístico con un 54% de recurrencia sobre entre las respuestas a soluciones para fomentar el turismo en el cantón Morona.
7. La población conoce buenos lugares entre “Pocos” para 23,3%, el 76,6% conoce entre “muchos” y “algunos”.
8. En total 56% de los encuestados califica al turismo en el cantón como “malo”, “regular” y “pésimo”. Existe un gran vacío en la oportunidad de mejora para las riquezas turísticas del cantón que aún no se ha concretado.
9. 9 de cada 10 (91,5%) servidores turísticos está interesado en promocionar sus eventos a través de una sencilla aplicación web.

### *3.1.3 Definición del Ámbito*

La Municipalidad del cantón Morona en concordancia con su misión de fomentar el desarrollo cantonal en forma planificada y sostenible, avalan el uso de una aplicación que fomente el turismo para el cantón y la provincia según el *Anexo 5: Aval para aplicación en el Municipio de Morona Santiago*. Para ello se comprometen a asumir roles dentro del proyecto como “cliente”, dentro del marco de la metodología de desarrollo Scrum en beneficio del producto final.

#### *3.1.3.1 Perspectiva del Producto*

Mape es una aplicación que muestra un mapa con eventos, localidades comerciales, entidades públicas y eventos privados, cuenta con una interfaz sencilla y de última generación. Sirve para que los usuarios puedan consultar sobre los eventos que lo rodean en la ciudad en la que se encuentren, marcar sus propios puntos en el mapa, consultar sobre dónde encontrar sus necesidades tanto en productos como servicios, beneficiarse de los eventos de sus locales de suscripción. Los locales comerciales son gestionados en nuestro sistema a través de cuentas comerciales que promocionan sus eventos a través de Mape, mostrando dichos eventos en un mapa público. Con la suma de todas las funciones, los usuarios tienen la capacidad de abrir una nueva perspectiva de su entorno, visualizando y modificando de forma personalizada sus intereses.

Los elementos principales para el usuario son:

- Cuenta de usuario: se crea con el primer uso y puede contener otra cuenta comercial.
- Contactos: lista de usuarios agregados a quienes puede realizar invitaciones.
- Cuenta comercial: puede ser creada a partir de una cuenta de usuario, puede acceder a beneficios comerciales.
- Eventos Públicos: son aquellos eventos a los que todos están invitados, se muestran en el mapa público.
- Eventos Privados: son aquellos que han sido creados desde cuentas de usuarios y requieren de invitación personalizada.
- Localidades públicas: son pertenecientes a cuentas comerciales o entidades públicas o de gobierno.

- Localidades privadas: son aquellas que se muestran únicamente en las cuentas de usuarios que las crean, pueden estar sujetas a invitación.
- Búsqueda: el texto ingresado para la búsqueda puede referirse a productos, servicios, localidades y eventos.
- Vistas: Muestran la información personalizada de los eventos, mapa público y configuraciones.

Este proyecto de software podría complementarse con otros sistemas como Facebook, Twitter y Google, etc Por la naturaleza de las *apps* y el concepto de internet 3.0, el cual demanda una simbiosis entre sistemas de información, esta *app* permitirá crear cuentas a partir de otros sistemas de información como los antes mencionados, con el fin de ser más eficiente con el uso de los datos.

### 3.1.3.2 Funcionalidad del Producto

En un principio de planificación se han definido los principales elementos que interactuarían dentro del sistema para que funcione para satisfacer los requerimientos, es así que se definieron los principales objetos, tal como se muestran en la **Figura 3-8** a continuación:



**Figura 3-13:** Objetos principales de la aplicación

**Fuente:** Especificación de Requerimientos Mape, 2016 (*Anexo N°1: Planificación Mape*)

Mape consiste en un mapa de eventos mostrado al usuario a través de su navegador, ya sea en un dispositivo móvil o computador, estaría accesible a su uso desde una dirección web, serviría para publicar eventos ingresados y administrados por los usuarios de la aplicación. Su clasificación dentro del entorno de aplicaciones es la de “Viajes y Guías” ya que serviría directamente al ámbito turístico. Entre las principales funciones del producto se encuentran:

- Registrar y gestionar cuentas de usuarios
- Registrar y gestionar información de localidades públicas y comerciales de usuarios
- Registrar y gestionar marcas de localidades
- Registrar y gestionar eventos

### *3.1.3.3 Alcance*

Identificación del producto como “Mape”, una aplicación informática dentro de la categoría de app web multiplataforma, responsiva.

La aplicación está enfocada a ser una fuente de información acerca de las opciones de entretenimiento a través de eventos propuestos de primera mano por locales comerciales y usuarios. Ofrecer al usuario una plataforma para publicar no solo los eventos sino también las localidades en donde se realizan (en un intento por registrar las vivencias encontradas en un lugar en específico).

Una vez verificada la información proporcionada por el usuario, este podrá acceder a la opción de publicar a nivel global sus eventos (todos los usuarios de Mape, aun sin registrarse podrían conocer sobre el evento en el sistema) beneficios como administrador de una localidad con fines publicitarios (comerciales), con un módico costo anual.

La aplicación, se alojaría en un host abierto a toda la web por lo cual es importante implementar controles para asegurar la integridad de la información guardada y mostrada a los demás usuarios. De tal forma Mape puede llegar a convertirse en una aplicación de ámbito internacional, no obstante el lenguaje oficial de la app web es el español.

### *3.1.3.4 Características de Usuarios*

El sistema está dirigido al público en general, por lo tanto no se requieren características específicas para su uso además del conocimiento básico sobre el uso de un navegador de internet.

Tipos de usuarios y permisos de acceso al sistema de software:

- Administrador: Acceso permitido a la interfaz de administrador Django; acceso a la base de datos con funciones de modificación, creación y eliminación.
- Usuario Normal: Acceso al sistema a través de su cuenta activa, con permisos de creación de eventos, localidades y gestión de cuentas comerciales a su cargo.

- Usuario Comercial: Se deriva del usuario normal, a diferencia del primero, este debe completar información personal que será validada por el sistema.

**Tabla 3-2:** Características de usuario administrador.

<b>Tipo de Usuario</b>	Usuario Administrador
<b>Formación</b>	Manejo de ordenadores, conocimiento en informática avanzada.
<b>Habilidades</b>	Manejo de ordenadores y/o dispositivos móviles
<b>Actividades</b>	<ul style="list-style-type: none"> <li>● Generar búsqueda</li> <li>● Creación de Localidades públicas/privadas</li> <li>● Gestión de eventos</li> <li>● Agregar contactos</li> <li>● Gestión de cuenta comercial</li> </ul>

**Fuente:** Especificación de Requerimientos Mape, 2016 (*Anexo N°1: Planificación Mape*)

**Tabla 3-3:** Características de usuario normal.

<b>Tipo de Usuario</b>	Usuario Normal
<b>Formación</b>	Ninguna
<b>Habilidades</b>	Manejo de ordenadores y/o dispositivos móviles
<b>Actividades</b>	<ul style="list-style-type: none"> <li>● Generar búsqueda</li> <li>● Creación de Localidades públicas/privadas</li> <li>● Gestión de eventos</li> <li>● Agregar contactos</li> <li>● Gestión de cuenta comercial</li> </ul>

**Fuente:** Especificación de Requerimientos Mape, 2016 (*Anexo N°1: Planificación Mape*)

**Tabla 3-4:** Características de usuario comercial.

<b>Tipo de Usuario</b>	Usuario Comercial
<b>Formación</b>	Conocimiento sobre redes sociales
<b>Habilidades</b>	Manejo de ordenadores y/o dispositivos móviles
<b>Actividades</b>	Gestión de eventos Gestionar localidad comercial

**Fuente:** Especificación de Requerimientos Mape, 2016 (*Anexo N°1: Planificación Mape*)

### ***3.1.4 Estudio de Factibilidad***

Luego de establecer el ámbito y pre visualizar el alcance que tendrá el sistema a desarrollar de forma general y no necesariamente de forma oficial, antes de empezar con el desarrollo, se debe medir la factibilidad desde las diferentes perspectivas de la aplicación: técnica, operativa, económica y legal. Esto con el fin de establecer si el sistema de software a desarrollar es factible o no.

#### ***3.1.4.1 Factibilidad Técnica***

##### **Recurso Humano**

El desarrollo del presente proyecto de tesis cuenta con el personal adecuado, capacitado y comprometido con los objetivos propuestos.

En este apartado se muestra el Equipo Scrum, el documento oficial donde se asumen las responsabilidades es el SRS (*Anexo N°1: Planificación Mape*).

El proyecto está propuesto por el estudiante Juan Solórzano con el apoyo del Municipio del cantón Morona a través del Departamento de Turismo y bajo la guía del director de tesis sobre quienes recaen los siguientes roles:

**Tabla 3-5:** Personal de trabajo.

<b>Nombre</b>	Juan Alfredo Solórzano Ávila
<b>Rol</b>	<i>Desarrollador/ProductOwner</i>
<b>Responsabilidades</b>	Diseño y desarrollo del sistema,
<b>Información de contacto</b>	juan.s.avila.2014@gmail.com

**Fuente:** Especificación de Requerimientos Mape, 2016 (*Anexo N°1: Planificación Mape*)

### **Recurso Hardware**

Se ha gestionado para obtener los siguientes dispositivos de hardware necesarios para la puesta en marcha del proyecto:

**Tabla 3-6:** Equipamiento de trabajo.

<b>Equipo</b>	<b>Descripción</b>	<b>Estado</b>
Pc de escritorio	Procesador Intel Core i5 64Bits, 3.5Ghz, RAM 8Gb.	Bueno
Monitor	18 pulgadas.	Bueno
Mouse, teclado	-	Bueno
Celular	Xperia Z3 Compact	Bueno

**Fuente:** Especificación de Requerimientos Mape, 2016

## Recurso Software

Se dispone de herramientas disponibles en internet de forma gratuita o bajo un mínimo costo de inversión:

**Tabla 3-7:** Equipo de software.

Software	Descripción	Costo
Python 2.7	Lenguaje de programación interpretado, orientado a objetos.	Gratuito
Django 1.8	Framework para desarrollo de aplicaciones web, basado en Python.	Gratuito
SublimeText 2	Editor de código fuente.	Gratuito/Registro
PostgreSQL 9.6	Sistema de gestión de bases de datos relacional orientado a objetos.	Gratuito
Mapbox API	Interfaz para uso de servicios web de Mapbox limitados.	Gratuito/Pago
MaterializeCSS	Framework CSS para diseño de páginas HTML, basado en Material Design.	Gratuito

**Realizado por:** Juan Solórzano

### 3.1.4.2 Factibilidad Operativa

La aplicación es totalmente operativa en cuanto al personal para administración posterior a su instalación, además de la facilidad que ofrece Django en cuanto a su soporte para la administración de sus aplicaciones.

El sistema estará montado en los servidores administrados por el departamento de sistemas del Municipio de Morona.

### 3.1.4.3 Factibilidad Económica

## HARDWARE

**Tabla 3-8:** Valoración de hardware.

CANTIDAD	RECURSO	V. UNITARIO	V. TOTAL
1	● Computador	\$750.00	\$750.00
1	● Smartphone	\$650.00	\$650.00
1	● Impresora	\$160.00	\$160.00
		<b>Total</b>	\$1560.00USD

**Fuente:** Especificación de Requerimientos Mape, 2016 (*Anexo N°1: Planificación Mape*)

## SOFTWARE

**Tabla 3-9:** Valoración de software.

CANTIDAD	RECURSO	V. UNITARIO	V. TOTAL
1	Python 2.7.12	Gratis	0
1	Framework Django 1.8	Gratis	0
1	Google Chrome	Gratis	0
1	Sublime Text 2	Gratis	0
1	PostgreSql 9.6	Gratis	0
1	Mapbox API	Gratis	0
1	MaterializeCSS	Gratis	0
		<b>Total</b>	\$0.00USD

**Fuente:** Especificación de Requerimientos Mape, 2016 (*Anexo N°1: Planificación Mape*)

**Nota:** Se ha hecho énfasis en el uso de tecnologías populares, libres y económicas, en este caso gratuitas.

## SUMINISTROS DE OFICINA

**Tabla 3-10:** Valoración de suministros de oficina.

CANTIDAD	RECURSO	V. UNITARIO	V. TOTAL
2	• Resmas de papel Bond A4	\$4.00USD	\$8.00USD
5	• CDs	\$0.50USD	\$2.50USD
1	• Flash de 8gb	\$12.00USD	\$12.00USD
4	• Tintas de colores	\$3.00USD	\$12.00USD
		<b>Total</b>	<b>\$34.50USD</b>

**Fuente:** Especificación de Requerimientos Mape, 2016 (*Anexo N°1: Planificación Mape*)

## OTROS

**Tabla 3-11:** Valoración de otros rubros.

CANTIDAD	RECURSO	V. UNITARIO	V. TOTAL
6 MESES	Electricidad	\$20.00USD/mes	\$120.00USD
6 MESES	Internet	\$18.00USD/mes	\$108.00USD
1	Libro de Django	Gratis	Gratis
		<b>Total</b>	<b>\$228.00USD</b>

**Fuente:** Especificación de Requerimientos Mape, 2016 (*Anexo N°1: Planificación Mape*)

**Tabla 3-12:** Valoración total.

<b>RECURSO</b>	<b>V. TOTAL</b>
Hardware	\$1560.00USD
Software	\$0.00USD
Suministros	\$34.50USD
Otros	\$228.00USD
<b>TOTAL</b>	<b>\$1822.50USD</b>

**Fuente:** Especificación de Requerimientos Mape, 2016 (*Anexo N°1: Planificación Mape*)

El costo de producción para el presente trabajo de titulación es asumido por el proponente en su totalidad.

#### *3.1.4.4 Factibilidad legal*

Las autoridades competentes están de acuerdo con los objetivos y funcionalidades del sistema, las leyes permiten la creación e implementación del sistema sin ningún inconveniente ni requerimiento previo por lo que la factibilidad legal no es un problema actualmente.

### **3.2 Desarrollo de tareas**

En este subcapítulo se muestra como se invierte el esfuerzo para el desarrollo de las principales tareas que fueron necesarias para construir la aplicación: desde la preparación del PC de trabajo hasta el despliegue del sistema con sus pruebas y los resultados arrojados.

#### *3.2.1 Preparación de entorno de trabajo*

El presente proyecto requiere la utilización de herramientas de software que deben instalarse en un sistema operativo que sea compatible con todas estas herramientas, por lo que se recomienda el uso de un sistema basado en Linux.

La guía para realizar las instalaciones respectivas en el sistema operativo Windows 10 se encuentra en el *Anexo3: Preparación de entorno de trabajo*.

### 3.2.2 Agenda según Scrum

#### 3.2.2.1 Definición del ProductBacklog

El ProductBacklogmostrado a continuación muestra una de las herramientas más importantes del proyecto Scrum, definido a partir de la planificación inicial, muestra los requisitos del sistema a desarrollarse en orden de prioridades y representa un primer plano del trabajo en su totalidad. Se recopila la información específicamente de la lista de requerimientos (*Anexo N°1: Planificación Mape*).

Product Backlog										
ID	ID	Requisitos	Departamento	Horas estimadas	Factor de ajuste	Estimación ajustada	Iteracion:			
							Pendiente:	1	2	3
							240	165	93	18
1	RF1	Permitir la autenticación de usuario	Programación	45	-0.222222	-10		35	0	0
2	RF2	Permitir la gestión de su cuenta de usuario	Programación	35	0.1428571	5		40	0	0
		<b>Totales Sprint1:</b>		<b>80</b>	<b>-0.0625</b>	<b>-5</b>		<b>75</b>	<b>0</b>	<b>0</b>
<b>Primera entrega</b>										
6	RF6	Permitir la gestión de localidades públicas/privadas	Programación	25	0.28	7		0	32	0
3	RF3	Permitir la gestión de una localidad comercial	Programación	20	-0.25	-5		0	15	0
5	RF5	Permitir la gestión de Eventos	Programación	25	0	0		0	25	0
		<b>Totales Sprint2:</b>		<b>70</b>	<b>0.0285714</b>	<b>2</b>		<b>0</b>	<b>72</b>	<b>0</b>
<b>Segunda entrega</b>										
4	RF4	Permitir guardar contactos	Programación	20	0.25	5		0	0	25
7	RF7	Permitir al usuario realizar una búsqueda	Programación	25	0	0		0	0	25
8	RF8	Permitir al usuario aceptar o rechazar una solicitud de	Programación	35	-0.285714	-10		0	0	25
9	RF9	Gestionar palabras Hashtag en Posts	Programación	10	-1	-10		0	0	0
		<b>Totales Sprint3:</b>		<b>90</b>	<b>-0.166666</b>	<b>-15</b>		<b>0</b>	<b>0</b>	<b>75</b>
<b>Tercera entrega</b>										

**Figura 3-14:ProductBacklog del proyecto Mape**

Realizado por: Juan Solórzano

#### 3.2.2.2 Sprint 1

El conjunto de tareas que se exponen a continuación son generadas a partir de las reuniones de planificación de las cuales se obtienen los principales requisitos y funcionalidades para el sistema, dichos requerimientos representan un conjunto de actividades o tareas que están incluidas en la lista del primer Sprint que en última instancia formarán la estructura del proyecto reflejado en los componentes principales de Django.

**Tabla 3-13:** Nomenclatura de tabla de tareas

<b>Requisito Origen</b>	Requisito funcional que origina la tarea, descritas anteriormente en la especificación de requerimientos de software en formato extendido.
<b>ID</b>	Número de tarea con la que se identifica.
<b>Estado</b>	Puede ser; Pendiente (P), Desarrollando (D) y Listo (L).
<b>Componente</b>	Objeto final o parcial que se obtiene del cumplimiento de la tarea en cuestión, representa un componente dentro del entorno de Django o dentro del producto final.

Realizado por: Juan Solórzano

**Tabla 3-14:** Tareas del Sprint 1

Requisito Origen	ID	Estado	Tareas	Componente
RF1	0	L	Configuración de proyecto: creación de proyecto Mape, app social, cuenta admin.	Configuración (settings.py)
RF1	1	L	Modelos version 1: User, profiles, events, hashtags, sponsors, trades, categories_od_trades, categories, offers_of_trade, offers	Configuración (settings.py)
RF1	2	L	Configuración de proyecto: base de datos, idiomas, apps, plantillas, zona horaria, login/logout	Configuración (settings.py)
RF1	3	L	Maquetación UI: Login / logout.	Diseño de interfaces UI
RF1	4	P	Views: logout_view, login_view	Vistas (views.py)
RF1	5	L	Template: login/logout	<ul style="list-style-type: none"> <li>• Templates o plantillas (.html)</li> </ul>
RF1	6	L	Funciones de materialize: rango_jquery.js	Diseño de interfaces UI
RF1	7	L	URLs: login, logout	URLs (urls.py)
RF1	8	D	Diccionario de datos: User, profiles, events, hashtags, sponsors, trades,	Documentación

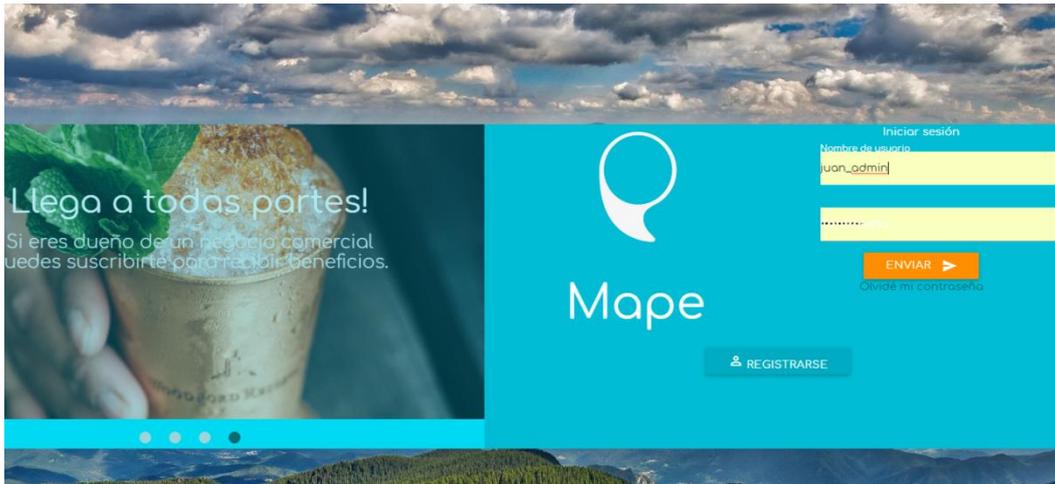
		categories_od_trades, offers_of_trade, offers	categories,	
RF1	9 L	Configurar admin.py	Módulo de Administración (admin.py)	
RF1	10 L	Templates: base maccounts	Templates o plantillas (.html)	
RF2	11 L	Forms: RegisterProfileForm, RegisterUserForm	Formularios (forms.py)	
RF2	12 L	Maquetacion UI: configuracion de cuenta	Diseño de interfaces UI	
RF2	13 L	Views: profile_register_view, user_register_view, accounts_view	Vistas (views.py)	
RF2	14 D	Templates: base mape, mape, profile_register, user_register, accounts	Templates o plantillas (.html)	
RF2	15 D	URLs: accounts	URLs (urls.py)	

**Realizado por:** Juan Solórzano

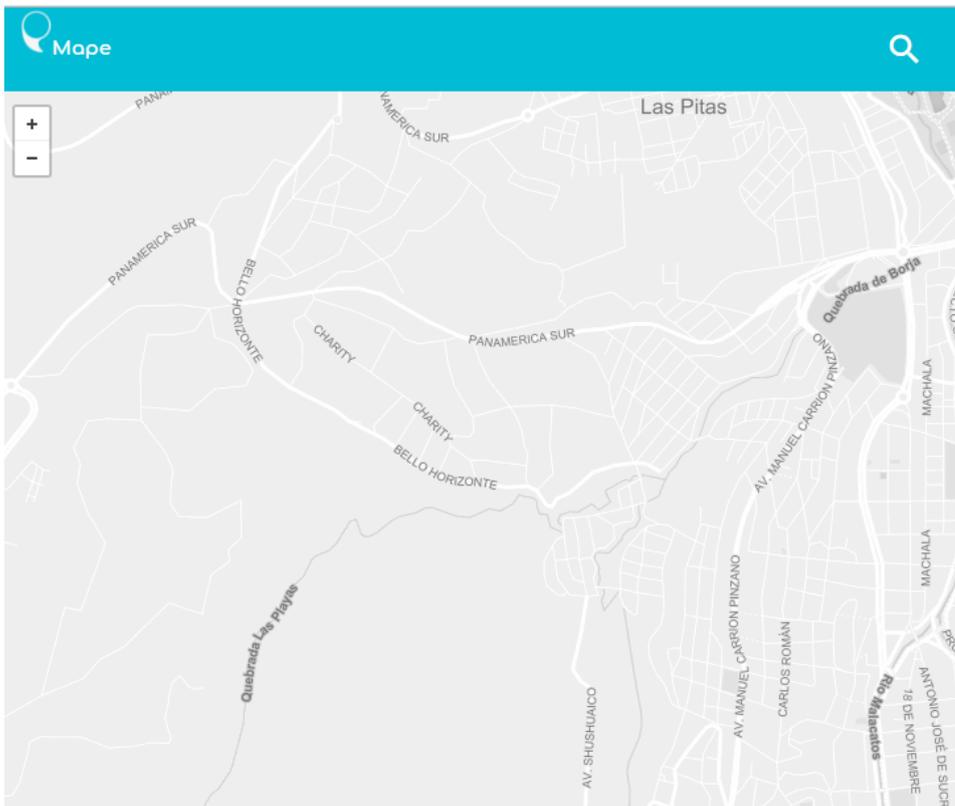
**Fuente:** Sprint Backlog 1, 2016 (Anexo N°2: Product Backlog, Sprint Backlog 1, Sprint Backlog 2 y Sprint Backlog 3)

Cada Sprint tiene una duración de 30 días, para el primer sprint se inicia el 8 de Marzo del 2016, culminando las tareas el día 30 de Marzo. Posteriormente de un Sprint inicia el evento de retrospectiva para pulir la forma de trabajo entre el equipo de desarrollo y realizar modificaciones en el ProductBacklog y el siguiente Sprint Backlog de ser necesario.

Algunas de las capturas del producto al final del primer Sprint:



**Figura 3-15:** Captura de pantalla del “Registro”



**Figura 3-16:** Captura de pantalla “Principal”

**Juanito ASudnaisdfas**

Nombre de usuario juan	Dirección de correo electrónico solovino.16.2012@gmail.com
Nombre Juanito	Apellidos ASudnaisdfas
Cédula 1234567890	Fecha de nacimiento 26/05/2017
Número de celular 1234567890	

IMAGEN DE PERFIL ACTUALMENTE:  
SOCIAL/AVATARES/13091825\_938764772908590\_1656332065\_0.JPG  LIMPIAR

**GUARDAR** >

**Figura 3-17:** Captura de pantalla de “Formulario de usuario”

### 3.2.2.3 Sprint 2

Existen tareas “arrastradas” del Sprint 1 las cuales se tuvieron que resolver durante el siguiente Sprint, además de añadir adaptaciones en los requerimientos. Para el inicio del Sprint 2 se definieron las siguientes modificaciones:

RF1: Permitir la autenticación de usuario

RF3: Permitir la gestión de una localidad comercial

- RF1: Permitir la autenticación con facebook
- RF3: Se requiere una serie de objetos para la implementación de productos y servicios de localidades comerciales.

El inicio del segundo sprint fue el día lunes 4 de abril, tras 4 semanas de trabajo se culminó el día Viernes 28 del mismo mes, obteniendo el siguiente avance en el proyecto:

**Tabla 3-15:** Resultado de Sprint 2

Requisito Origen	ID	Estado	Tareas	Requisito Final
RF1	4	L	Views: logout_view, login_view	Vistas (views.py)

RF1	6	L	Funciones de materialize: rango_jquery.js	Diseño de interfaces UI
RF1	8	L	Diccionario de datos: User, profiles, events, hashtags, sponsors, trades, categories_od_trades, categories, offers_of_trade, offers	Documentación
RF6	16	L	View: Location_register_view	Vistas (views.py)
RF6	17	D	View: Shower_view	Vistas (views.py)
RF6	18	L	Forms: LocationRegisterForm	Formularios (forms.py)
RF6	19	L	Template: Location_register	Plantillas o plantillas (.html)
RF6	20	L	Template: Shower	Plantillas o plantillas (.html)
RF6	21	L	Static files: rango_mape: Mapboxjs	Javascript
RF6	22	L	Static files: rango_jquery: Materialize js	JavaScript
RF6	23	L	Template: Base_mape: scripts Mapboxjs	Plantillas o plantillas (.html)
RF6	24	L	Template: Base_mape: scripts Materialize js	Plantillas o plantillas (.html)
RF6	25	L	URLs: location_register, shower	URLs (urls.py)
RF3	26	L	cssmape: configuracion de colores, tamaños, letras, etiquetas.	Diseño de interfaces UI
RF3	27	L	Maquetacion UI: pantalla principal, registro de comercial, opciones de pantalla principal.	Diseño de interfaces UI
RF3	28	L	Forms: RegisterCommercialForm	Formularios (forms.py)
RF3	29	L	Views: Commercial_register_view	Vistas (views.py)
RF3	30	D	Template: Commercial_register	Plantillas o plantillas (.html)
RF3	31	L	URLs: Commercial_register	URLs (urls.py)

RF5	32	L	View: Event_register_view	Vistas (views.py)
RF5	33	L	Forms: EventRegisterForm	Formularios (forms.py)
RF5	34	D	Template: Event_register, Event_detail	Plantillas o plantillas (.html)
RF5	35	L	URLs: event_register,	URLs (urls.py)
RF6	-	L	Template: comercial_chip, profile_chip,	Plantillas o plantillas (.html)
RF4	36	L	Modelos version 1: configuracion de modelo Groups	Configuración (settings.py)
RF4	37	L	Views: contacts, add_contact_view	Vistas (views.py)
RF4	38	L	Template: base_contacts, contacts_stack	Diseño de interfaces UI
	39	L	URLs: contacts	URLs (urls.py)

**Realizado por:** Juan Solórzano

**Fuente:** Sprint Backlog 2 Mape, 2016

### 3.2.2.4 Sprint 3

El Sprint 3 se da inicio el día lunes 1 de mayo del cronograma de actividades, con algunas tareas que vinculan al Sprint 2 referente a:

RF4: Permitir guardar contactos

- RF4: Se requiere un componente (app Django) para la gestión de suscriptores.
- RF4: Se requiere un módulo para resolver solicitudes.
- RF4: Se implementa solicitudes especiales para usuarios comerciales.
- RF4: Se implementa un módulo para el manejo de notificaciones.

**Tabla 3-16:** Resultado de Sprint 3

Requisito Origen	ID	Estado	Tareas	Requisito Final
RF4	40	L	View: RelationshipListView	Vistas (views.py)
RF4	41	L	Models: Friendship	Modelos (Models.py)
RF4	42	L	Template: Relationship.html	Plantillas o plantillas (.html)
RF4	43	L	URLs: Friendship	URLs (urls.py)
RF7	44	L	View: Search_view, Shower_view, Map_view	Diseño de interfaces UI
RF7	45	L	Template: Base_mape: scripts Mapboxjs	Plantillas o plantillas (.html)
RF7	46	L	Template: Base_mape: scripts Materialize js	Plantillas o plantillas (.html)
RF7	47	L	Static files: rango_jquery: Mapboxjs	Javascript

RF7	48	L	Static files: rango_jquery: Materialize js	JavaScript
RF7	49	L	Settings: Implementacion de Mapbox.py	Configuración(settings.py)
RF7	50	L	Template: Mape.html, Shower_event.html	Plantillas o plantillas (.html)
RF8	51	L	View: Notification	Vistas (views.py)
RF8	52	L	Templates: Notification_list	
RF8	53	L	Url: notifications, notification_read	Diseño de interfaces UI

**Realizado por:** Juan Solórzano

**Fuente:** Sprint Backlog 2 Mape, 2016

### 3.3 Implementación y pruebas

#### 3.3.1 Definición de Estándares de programación

La programación se ha realizado en el lenguaje de programación Python, JavaScript y JQuery (definido en inglés), además se aplicaron los siguientes estándares de programación:

1. Se usa el idioma inglés (estándar internacional) para definir las **sentencias** (código fuente).
2. La escritura de los **comentarios** es importante, se describe la función de los bloques de código en idioma español o inglés.
3. Los **objetos** dentro del proyecto no pueden nombrarse anteponiendo doble guión bajo, y en general en los casos que se aplique para nombres reservados del framework Django.
4. Los nombres de los **documentos, archivos de cualquier tipo y carpetas** usan letras minúsculas y separando las palabras con guión bajo.
5. Únicamente los nombres de las **clases** se escriben con las primeras letras de las palabras mayúsculas (capitalizadas) y si son más de dos deben ir juntas (sin separación de guión bajo).

- 6. Los nombres para nombres de **funciones, variables e instancias** de clases y objetos, deben ser escritas en letras minúsculas y con separación de guión bajo.

### 3.3.2 Pruebas de Verificación

Esta técnica prueba la habilidad del programa para manejar datos que se encuentren en los límites aceptables, como por ejemplo el número de dígitos para un numero de cedula. Este tipo de pruebas se han realizado sobre el tiempo de desarrollo, usando las herramientas del navegador y depurando las fallas en tiempo de ejecución (permitido en Django).

- Nombre de usuario con espacios y/o símbolos, menor a 3 caracteres:

**Tabla 3-17:** Caso de prueba 1

Valor ingresado	Valor esperado	Valor obtenido	Aprobado
abcd12	error	none	False
abc d	error	none	False
ab	error	ab	False
abcde	abcde	abcde	True

**Realizado por:** Juan Solórzano

**Fuente:** Casos de prueba Mape, 2016

- Correo electrónico erróneo:

**Tabla 3-18:** Caso de prueba 2

Valor ingresado	Valor esperado	Valor obtenido	Aprobado
asns@@aks.com	error	none	False
asns @kds.com	error	none	False
karen12@mail.ku	karen12@mail.ku	karen12@mail.ku	True

**Realizado por:** Juan Solórzano

**Fuente:** Casos de prueba Mape, 2016

- Número de cédula de 10 dígitos:

**Tabla 3-19:** Caso de prueba 3

Valor ingresado	Valor esperado	Valor obtenido	Aprobado
12345678901	truncated	truncated	True

12345678a2	error	12345678a2	False
1234567890	1234567890	1234567890	True
1400700488	1400700488	1400700488	True
0348002303	0348002303	0348002303	True

**Realizado por:** Juan Solórzano

**Fuente:** Casos de prueba Mape, 2016

- Número de teléfono celular de 10 dígitos:

**Tabla 3-20:** Caso de prueba 4

Valor ingresado	Valor esperado	Valor obtenido	Aprobado
123asd12	error	123asd12	False
1234567890123	error	1234567890123	False
1400700488	1400700488	1400700488	True
0348002303	0348002303	0348002303	True

**Realizado por:** Juan Solórzano

**Fuente:** Casos de prueba Mape, 2016

- Fecha de nacimiento correcto, no menores de 12 años:

**Tabla 3-21:** Caso de prueba 5

Valor ingresado	Valor esperado	Valor obtenido	Aprobado
12/04/2005	12/04/2005	12/04/2005	True
12/04/2016	restricted	restricted	True

**Realizado por:** Juan Solórzano

**Fuente:** Casos de prueba Mape, 2016

- Evento de entidad privada no realiza eventos en localidades privadas (no comerciales):

**Tabla 3-22:** Caso de prueba 6

Valor ingresado	Valor esperado	Valor obtenido	Aprobado
Event: [ public: "True",] Locality.Commercial = True	Restricted	"Saved"	False

12/04/2016

restricted

restricted

True

**Realizado por:** Juan Solórzano

**Fuente:** Casos de prueba Mape, 2016

- El evento contiene una fecha anterior a la actual:

**Tabla 3-23:** Caso de prueba 7

<b>Valor ingresado</b>	<b>Valor esperado</b>	<b>Valor obtenido</b>	<b>Aprobado</b>
12/04/2016	restricted	restricted	True
12/04/2018	12/04/2018	12/04/2018	True

**Realizado por:** Juan Solórzano

**Fuente:** Casos de prueba Mape, 2016

Las pruebas de Módulos y otras clases de testeo se muestran con sus resultados en el *Anexo N°4: Pruebas Mape*.

### 3.4 Resultados

El proceso de encuestas dio como resultado algunos datos interesantes sobre la percepción de la gente sobre el turismo local, como la tasa de 54% de personas que opina que el turismo es “malo, regular o pésimo”, sin embargo la gran mayoría reconocen el potencial turístico que tienen, con una tasa de 94% de personas que creen que tienen “mucho o bastante” potencial turístico en Morona Santiago. Estos y otros resultados han sido interpretados como una oportunidad para mejorar la promoción turística existente, y han servido de apoyo para la planificación de la aplicación web multiplataforma Mape.

Previo al desarrollo se ha realizado un estudio sobre la factibilidad para el desarrollo de dicha aplicación, los resultados obtenidos en lo que concierne a la factibilidad económica de hardware y otros llegan a \$1.822USD aproximadamente (asumidos por el proponente), los recursos software suman un total de \$0USD gracias a que se ha utilizado software libre, en cuanto a

factibilidad legal y operativa no se han detectado ningún impedimento para el desarrollo, por sobre todo está la apertura que se le ha dado al proyecto en el Municipio de Morona Santiago.

Los datos de la investigación han ofrecido grandes beneficios a la hora de tomar decisiones sobre la funcionalidad del sistema, dando como resultado un mapa digital (Mape) enfocado a responder al ¿Qué, dónde y cuándo? de los eventos, productos y servicios que ofrecen los servidores turísticos registrados en el Municipio de Morona Santiago.

Las herramientas utilizadas dieron como resultado un código limpio, flexible a cambios, cubriendo los requisitos planteados con relativa facilidad, su normal funcionamiento en los principales navegadores incluyendo de dispositivos móviles. La *app* está disponible en internet bajo el nombre de dominio *mape.live*, en un servidor alojado en New York (NYC1). URL: <http://www.mape.live>

La aplicación responde con fluidez y muestra gran calidad tanto en el diseño como en su funcionamiento. Los tiempos de respuesta para las páginas estáticas son:

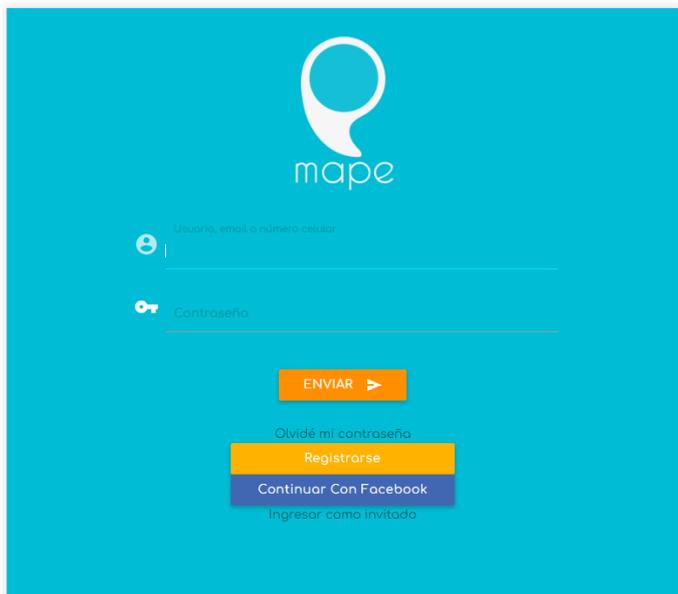
**Tabla 3-24:** Relación peso/carga de páginas estáticas

<i>Página</i>	<b>Cache</b>	<i>Tiempos de carga / Banda</i>	
		<b>12.5MB/s</b>	<b>2MB/s</b>
<i>/login</i>	772KB	2.20s	7.4s
<i>/signup</i>	1.6MB	7.55s	18.2s
<i>/conditions</i>	329KB	1.87s	4.3s
<i>/principles</i>	852KB	1.87s	4.9s

**Realizado por:** Juan Solórzano

**Fuente:** Casos de prueba Mape, 2016

A continuación se muestran las capturas de las pantallas antes mencionadas en el orden respectivo:



**Figura 3-18:** Captura de pantalla “login”



**Figura 3-19:** Captura de pantalla “signup”



Figura 3-20: Captura de pantalla “principles”

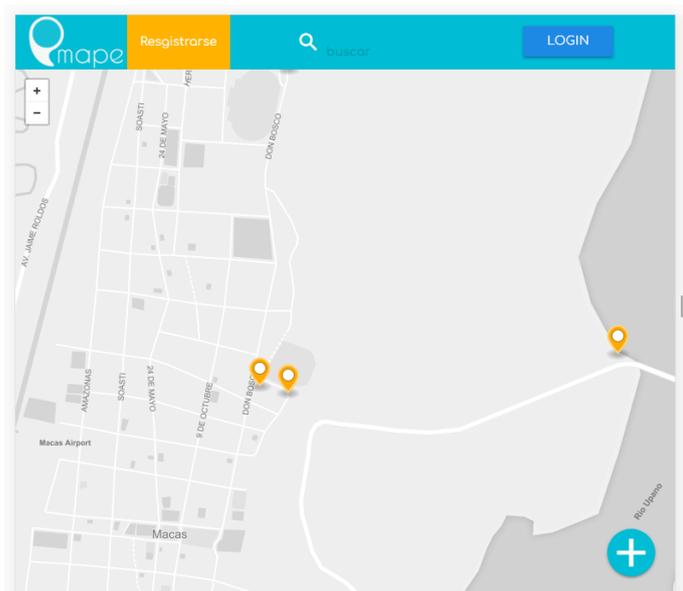


Figura 3-21: Captura de pantalla “map”

#### 4. CONCLUSIONES

- Una encuesta realizada a una muestra poblacional de 43 personas, determina que el 94% de las personas reconocen que tienen “mucho o bastante” potencial turístico, y el 54% menciona una falta de difusión de información turística en el cantón Morona, 35 de 38 servidores turísticos señalaron estar interesados en usar una sencilla aplicación para invitar a sus clientes a participar de sus eventos. Se establece la falta de promoción turística como la problemática a resolver mediante el desarrollo de una app que apoye al sector turístico.
- Se desarrolló la app web Mape con éxito y en el tiempo predeterminado según la agenda Scrum; *Mape.live* es el dominio usado para el despliegue del primer mapa de eventos en la provincia de Morona Santiago, prestando servicios para promocionar de eventos y lugares turísticos, disponible desde cualquier dispositivo móvil con acceso a internet. Implementado con tecnologías contemporáneas como lo son Django, Material Design, JQuery, incluyendo servicios web de Mapbox y Google Fonts. El uso de estas herramientas tecnológicas y metodológicas han supuesto un gran ahorro en los costos de producción.
- Se ha evidenciado los beneficios del uso del framework Django, además del mismo uso del lenguaje Python, este entorno de trabajo representa un gran aporte a la Programación Orientada a Objetos; flexible a cualquier necesidad de acoplamiento con los principales estándares sobre los que se trabaja a nivel empresarial. Django ayudó a optimizar significativamente la codificación del sistema, además de la facilidad de configurar nuevos módulos internos o externos, todo ello se traduce en una reducción en el tiempo de producción.
- La metodología de desarrollo Scrum resulta muy cómoda de implementar ya que es fácil de entender y se puede adaptar al tamaño del proyecto, el equipo de trabajo disponible, tiempos y otras metodologías. Las herramientas Scrum resultan muy útiles a la hora de tomar decisiones sobre la marcha y en los Eventos Scrum integran de forma eficiente las opiniones de todos los involucrados en el proyecto.

## 5. RECOMENDACIONES

- Para el desarrollo de una aplicación web, o una app móvil que consuma muchos recursos (según los requerimientos), se recomienda desarrollar una plataforma base como lo es una aplicación web alojada en un servidor propio, para posteriormente hacer uso de un *web-service* partiendo de una aplicación web estable, para desarrollo de su aplicativo en diferentes sistemas operativos o plataformas web, de esta manera las aplicaciones posteriores se libera de carga en la parte del cliente.
- Se recomienda el uso de *Scrum* para el desarrollo ágil de aplicaciones, en especial para proyectos de páginas web, en general para proyectos de corto o medio plazo de producción, es posible adaptar el entorno *Scrum* para proyectos más extensos ya que no es una metodología estática. De hecho se ha modificado para el uso de *SRS* en la planificación del presente proyecto.
- Es recomendable usar un *framework front-end* para el maquetado e implementación temprana de interfaces de usuario, posiblemente sea necesario manejar un nivel de personalización para cada tipo de proyecto en la fase. Este tipo de frameworks ayudan mucho en la etapa de producción pero puede que el producto final lleve una leve sobrecarga de métodos que podrían ser optimizados con la implementación de interfaces diseñadas personalmente.
- La interacción con el cliente en la fase de planificación de un proyecto de software es sumamente importante, por lo que se recomienda llevar un registro de las ideas y puntos de vista expresados en los eventos *Scrum*, inicialmente con *brainstorming* lluvia de ideas para organizar los requerimientos y todo lo necesario para completar el *SRS* en este caso.
- Se recomienda tener a disposición una tarjeta de crédito o débito para la etapa de despliegue de una página web, ya que será necesario contratar algunos servicios por internet que solo están disponibles bajo este medio de pago.

## 6. BIBLIOGRAFÍA

**Beaskoetxea, Ander**, *Django: el framework web definitivo*[blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<http://es.slideshare.net/alatar/django-el-framework-web-definitivo-1362169>>

Guido van Rossum. *El Tutorial de Python (Traducido y empaquetado por la comunidad de Python Argentina)* [en línea], Cap. 10. Clases. [Consulta: 25 Mayo 2016]. Disponible en <<http://docs.python.org.ar/tutorial/3/classes.html>>

**Garcia M, Saul**. *El Libro de Django*. Boston, MA 02111-1307, USA.2015, pp 6-61

**Holovaty, Adrian**, *Introducing Django 0.90*[blog]. [Consulta: 27 de septiembre de 2016] Disponible en <https://www.djangoproject.com/weblog/2005/nov/16/firstrelease/>

**Holovaty, Adrian**, *Django 0.91 released*[blog]. [Consulta: 27 de septiembre de 2016] Disponible en <https://www.djangoproject.com/weblog/2006/jan/11/091/>

**Holovaty, Adrian**, *Introducing Django 0.95*[blog]. [Consulta: 27 de septiembre de 2016] Disponible en <https://www.djangoproject.com/weblog/2006/jul/29/095/>

**Holovaty, Adrian**, *Announcing Django 0.96!*[blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<https://www.djangoproject.com/weblog/2007/mar/23/096/>>

**Holovaty, Adrian**, *Django 1.0 released!*[blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<https://www.djangoproject.com/weblog/2008/sep/03/1/>>

**Bennett, James**, *Django 1.1 released* [blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<https://www.djangoproject.com/weblog/2009/jul/29/1-point-1/>>

**Bennett, James**, *Django 1.2 released* [blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<https://www.djangoproject.com/weblog/2010/may/17/12/>>

**Bennett, James**, *Django 1.3 released* [blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<https://www.djangoproject.com/weblog/2011/mar/23/13/>>

**Bennett, James**, *Django 1.4 released* [blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<https://www.djangoproject.com/weblog/2012/mar/23/14/>>

**Bennett, James**, *Django 1.5 released* [blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<https://www.djangoproject.com/weblog/2013/feb/26/15/>>

**Bennett, James**, *Django 1.6 released* [blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<https://www.djangoproject.com/weblog/2013/nov/06/django-16-released/>>

**Bennett, James**, *Django 1.7 released* [blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<https://www.djangoproject.com/weblog/2014/sep/02/release-17-final/>>

**Bennett, James**, *Django 1.8 released* [blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<https://www.djangoproject.com/weblog/2015/apr/01/release-18-final/>>

**Graham, Tim**, *Django 1.9 released* [blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<https://www.djangoproject.com/weblog/2015/dec/01/django-19-released/>>

**CUMBA ARMIJOS, Paul David & BARRENO PILCO, Byron Augusto.** *Análisis de Python con Django frente A Ruby OnRails para desarrollo ágil de aplicaciones web. Caso práctico: DECH(Tesis).* (Ingeniería) Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería en Sistemas. Riobamba-Ecuador. 2012. pp.110-136

**GUILLEM, Victor.** *Materializecss, Framework front-end para Material design*[blog]. [Consulta: 27 de septiembre de 2016]. Disponible en <<https://www.genbetadev.com/frameworks/materializecss-framework-front-end-para-material-design>>

Johnson, Robert; &Kuby, Patricia, *Estadística Elemental, lo esencial.* 3ª ed. Av Santa Fe núm 505, piso 21, Col, Cruz Manca, Santa Fe. CP 05349, D.F. Mexico. Pp.509

Fulcher, Rich, *Material Design Awards 2015* [blog]. [Consulta: 15 de Octubre de 2016] Disponible en <<https://design.google/library/material-design-awards-2015/>>

**Marinas, Francisco,** *Material Design*[blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<http://www.titaniodigital.com/divulgacion-md.html>>

*Principales ventajas y desventajas de emplear una framework CSS* [blog]. [Consulta: 27 de septiembre de 2016] Disponible en <<http://blog.aulaformativa.com/principales-ventajas-y-desventajas-de-emplear-una-framework-css/>>

**Sánchez, Jordi.** *¿Qué es un 'framework'?* [blog], [Consulta: 24 Mayo 2016]. Disponible en <<https://jordisan.net/blog/2006/que-es-un-framework/>>

**Schwaber, Ken; & Sutherland, Jeff**, *La Guía definitiva de Scrum: Las reglas del juego* [en línea]. 2014. [Consulta: 27 de septiembre 2016]. Disponible en <<https://www.scrum.org/resources/scrum-guide>>

**Schwaber, Ken; & Sutherland, Jeff**, *Software in 30 Days*. Hoboken, New Jersey: John Wiley & Sons, 2012, pp.

**TORRES, Mariela; KARIM, Paz A.** “Tamaño de una Muestra para una Investigación de Mercado”. Boletín Electrónico, No. 02. (s.f.). (Guatemala) pp.11.

**WANG, Alvin & CHANG, Alan & MARK, Alex & LOUIE, Kevin.** *Materialize* [en línea]. Sobre Materialize [Consulta: 27 de septiembre del 2016]. Disponible en <<http://materializecss.com/about.html>>

## **7. ANEXOS**

### **Anexo N°1:**

PlanificaciónMape

**Anexo N°2:**

[Product Backlog, Sprint Backlog 1, Sprint Backlog 2,  
Sprint Backlog 3]

## **Anexo N°3:**

### **Preparación de entorno de trabajo**

**Anexo N°4:**  
Pruebas Mape

**Anexo N°5:**

**Aval para aplicación en el Municipio de Morona Santiago**