



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

**“DESARROLLO DE UN SISTEMA INTELIGENTE WEB  
ADAPTATIVO PARA LA RECOMENDACIÓN DE DESTINOS  
TURÍSTICOS EN EL CANTÓN MORONA BASADO EN REGLAS Y  
HECHOS”**

**TRABAJO DE TITULACIÓN**

Tipo: **PROYECTO TÉCNICO**

Para optar al Grado Académico de:

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**AUTORES: HURTADO JERVES JULIO CÉSAR**

**ZABALA BUSTAMANTE BRAYAN ANDRÉS**

**TUTOR: ING. ÁNGEL FLORES MGS.**

Macas - Ecuador

2018

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

El Tribunal del Trabajo de Titulación certifica que: El proyecto técnico: “DESARROLLO DE UN SISTEMA INTELIGENTE WEB ADAPTATIVO PARA LA RECOMENDACIÓN DE DESTINOS TURÍSTICOS EN EL CANTÓN MORONA BASADO EN REGLAS Y HECHOS”, de responsabilidad de los señores Julio César Hurtado Jerves y Brayan Andrés Zabala Bustamante, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

Dr. Julio Santillán.

**VICEDECANO DE LA FACULTAD DE** -----  
**INFORMÁTICA Y ELECTRÓNICA**

Ing. Patricio Moreno.

**DIRECTOR DE ESCUELA DE** -----  
**INGENIERÍA EN SISTEMAS**

Ing. Ángel Flores.

**DIRECTOR DEL TRABAJO DE** -----  
**TITULACIÓN**

Ing. Ligia Niama.

**MIEMBRO DE TRIBUNAL** -----

Nosotros, JULIO CÉSAR HURTADO JERVES y BRAYAN ANDRÉS ZABALA BUSTAMANTE, somos responsables de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del mismo pertenece a la Escuela Superior Politécnica de Chimborazo.

---

Julio César Hurtado Jerves

---

Brayan Andrés Zabala Bustamante

## **DEDICATORIA**

A nuestros padres por habernos apoyado en todo momento, por sus consejos, por su ejemplo de perseverancia y constancia, y por la motivación que nos han permitido ser personas de bien, pero más que nada, por su amor.

A nuestros maestros, aquellos que desde niños han sabido transmitirnos de la mejor manera sus conocimientos y algunos de sus valores. Por sus consejos y enseñanzas.

A todos nuestros amigos, por los buenos y los malos momentos. Por aquellas situaciones de alegría y de tristeza, por los abrazos y las peleas. Ellos han hecho que esta travesía sea mucho más llevadera.

Julio y Brayan.

## TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	vi
ÍNDICE DE FIGURAS.....	vii
ÍNDICE DE GRÁFICOS.....	viii
ÍNDICE DE ANEXOS.....	ix
ÍNDICE DE ABREVIATURAS.....	x
INTRODUCCIÓN .....	1
ANTECEDENTES .....	1
FORMULACIÓN DEL PROBLEMA .....	2
SISTEMATIZACIÓN DEL PROBLEMA .....	2
JUSTIFICACIÓN .....	2
JUSTIFICACIÓN TEÓRICA.....	2
JUSTIFICACIÓN APLICATIVA.....	3
OBJETIVOS.....	4
OBJETIVO GENERAL. ....	4
OBJETIVOS ESPECÍFICOS. ....	4
<b>CAPÍTULO I</b>	
<b>1. MARCO TEÓRICO .....</b>	<b>5</b>
<b>1.1. Turismo.....</b>	<b>5</b>
<b>1.2. Sector Turístico en Ecuador .....</b>	<b>6</b>
<b>1.3. Inteligencia Artificial.....</b>	<b>7</b>
<i>1.3.1. Sistemas Expertos .....</i>	<i>8</i>
<i>1.3.2. Lógica difusa.....</i>	<i>9</i>
<i>1.3.3. Sistemas Expertos basados en Redes Neuronales .....</i>	<i>9</i>
<i>1.3.4. Sistemas Expertos Basados en Reglas y Hechos .....</i>	<i>11</i>
<b>CAPÍTULO II .....</b>	<b>13</b>
<b>2. DESARROLLO DE UN SISTEMA INTELIGENTE WEB ADAPTATIVO PARA LA RECOMENDACIÓN DE DESTINOS TURÍSTICOS EN EL CANTÓN MORONA BASADO EN REGLAS Y HECHOS.....</b>	<b>13</b>
<b>2.1. Visión y Alcance.....</b>	<b>13</b>
<i>2.1.1. Visión del proyecto .....</i>	<i>13</i>
<i>2.1.2. Perfiles de Usuario.....</i>	<i>13</i>
<i>2.1.3. Ámbito del Proyecto .....</i>	<i>14</i>
<i>2.1.4. Herramientas que utilizar.....</i>	<i>14</i>
<i>2.1.5. Objetivos del Proyecto .....</i>	<i>15</i>
<i>2.1.6. Análisis y gestión de riesgos.....</i>	<i>15</i>
<i>2.1.7. Planificación Inicial .....</i>	<i>19</i>

2.1.8.	<i>Estudio de Factibilidad</i> .....	19
2.2.	<b>Planificación</b> .....	21
2.3.	<b>Especificación de Requisitos</b> .....	21
2.3.1.	<i>Visión General</i> .....	21
2.3.2.	<i>Descripción General</i> .....	21
2.3.3.	<i>Perspectiva del Proyecto</i> .....	21
2.3.4.	<i>Funciones de la Aplicación</i> .....	21
2.3.5.	<i>Limitaciones Generales</i> .....	22
2.3.6.	<i>Suposiciones y Dependencias</i> .....	22
2.3.7.	<i>Requisitos Futuros</i> .....	22
2.3.8.	<i>Requerimientos funcionales</i> .....	22
2.3.9.	<i>Requerimientos no funcionales</i> .....	25
2.3.10.	<i>Casos de Uso</i> .....	26
2.3.11.	<i>Arquitectura</i> .....	28
2.3.12.	<i>Diseño Lógico</i> .....	29
2.3.13.	<i>Diseño Físico</i> .....	29
2.3.13.1.	<i>Diagrama Físico de la Base de Datos</i> .....	29
2.3.13.2.	<i>Diagrama de Componentes.</i> .....	30
2.3.13.3.	<i>Diagrama de Despliegue</i> .....	30
2.4.	<b>Desarrollo</b> .....	31
2.5.	<b>Estabilización</b> .....	34
2.5.1.	<i>Plan de Prueba</i> .....	34
<b>CAPÍTULO III</b>		
3.	<b>MARCO DE RESULTADOS</b> .....	43
<b>CONCLUSIONES</b> .....		46
<b>RECOMENDACIONES</b> .....		47
<b>BIBLIOGRAFÍA</b>		
<b>ANEXOS</b>		

## ÍNDICE DE TABLAS

<b>Tabla 1-2</b>	Riesgos del proyecto.....	15
<b>Tabla 2-2</b>	Exposición de Riesgos.....	16
<b>Tabla 3-2</b>	Prioridad de Riesgos.....	16
<b>Tabla 4-2</b>	Código de color de la exposición del riesgo.....	17
<b>Tabla 5-2</b>	Gestión del Riesgo R6.....	17
<b>Tabla 6-2</b>	Gestión del Riesgo R4.....	18
<b>Tabla 7-2</b>	Hardware requerido.....	19
<b>Tabla 8-2</b>	Software requerido.....	20
<b>Tabla 9-2</b>	Factibilidad económica.....	20
<b>Tabla 10-2</b>	Requerimiento: Registro de usuario.....	22
<b>Tabla 11-2</b>	Requerimiento: Validar usuario.....	23
<b>Tabla 12-2</b>	Requerimiento: Crear Destino Turístico.....	23
<b>Tabla 13-2</b>	Requerimiento: Modificar Destino Turístico.....	23
<b>Tabla 14-2</b>	Requerimiento: Crear Servicio Turístico.....	24
<b>Tabla 15-2</b>	Requerimiento: Modificar Servicio Turístico.....	24
<b>Tabla 16-2</b>	Requerimiento: Eliminar Servicio Turístico.....	24
<b>Tabla 17-2</b>	Requerimiento: Gestión del conocimiento.....	24
<b>Tabla 18-2</b>	Requerimiento: Módulo de inferencia.....	25
<b>Tabla 19-2</b>	Caso de Uso Extendido: Gestión de Usuarios.....	26
<b>Tabla 20-2</b>	Caso de Uso Extendido: Gestión de Destinos Turísticos.....	27
<b>Tabla 21-2</b>	Caso de Uso Extendido: Gestión del conocimiento.....	28
<b>Tabla 22-2</b>	Extensiones de Archivos.....	33

## ÍNDICE DE FIGURAS

<b>Figura 1-1.</b>	Gráfica de definición de turismo.....	6
<b>Figura 2-1.</b>	Estructura de un Sistema Experto .....	9
<b>Figura 3-1.</b>	Diagrama de una Neurona Artificial (PE).....	10
<b>Figura 4-1.</b>	Arquitectura de una Red Neuronal Simple.....	10
<b>Figura 1-2.</b>	Diagrama de Casos de Uso: Gestión de Usuarios .....	26
<b>Figura 2-2.</b>	Diagrama de Casos de Uso: Gestión de Destinos Turísticos .....	27
<b>Figura 3-2.</b>	Diagrama de Casos de Uso: Gestión del Conocimiento.....	28
<b>Figura 4-2.</b>	Arquitectura de la Aplicación - Django .....	28
<b>Figura 5-2.</b>	Diseño Lógico de la Base de Datos .....	29
<b>Figura 6-2.</b>	Diagrama Físico de la Base de datos.....	29
<b>Figura 7-2.</b>	Diagrama de Componentes .....	30
<b>Figura 8-2.</b>	Diagrama de Despliegue .....	30
<b>Figura 9-2.</b>	Modelo Center.....	31
<b>Figura 10-2.</b>	Modelo Picture .....	31
<b>Figura 11-2.</b>	Modelo Picture .....	31
<b>Figura 12-2.</b>	Modelo Picture .....	32
<b>Figura 13-2.</b>	Vista CenterCreateView.....	32
<b>Figura 14-2.</b>	Vista PictureAdd .....	32
<b>Figura 15-2.</b>	Vista CreateServiceView .....	32
<b>Figura 16-2.</b>	Vista RegisterUserCreateView.....	33
<b>Figura 17-2.</b>	Código de la Función Listar destinos turísticos. ....	34
<b>Figura 18-2.</b>	Código de la Función Listar destinos turísticos. ....	35
<b>Figura 19-2.</b>	Código de la Función Detalle de un destino turístico.....	35
<b>Figura 20-2.</b>	Código de la Función Crear destino turístico.....	36
<b>Figura 21-2.</b>	Código de la Función Editar destino turístico. ....	37
<b>Figura 22-2.</b>	Código de la Función crear conocimiento.....	38
<b>Figura 23-2.</b>	Código de la función actualizar conocimiento.....	38
<b>Figura 24-2.</b>	Código de la Función realizar encuesta.....	39
<b>Figura 25-2.</b>	Código de la Función listar servicios. ....	40
<b>Figura 26-2.</b>	Código de la Función ver destinos turísticos de un servicio. ....	40
<b>Figura 27-2.</b>	Código de la Función crear servicios. ....	41
<b>Figura 28-2.</b>	Código de la Función crear servicios. ....	42
<b>Figura 29-2.</b>	Código de la Función crear servicios. ....	42
<b>Figura 29-2.</b>	Código de la Función crear servicios. ....	42



## ÍNDICE DE GRÁFICOS

<b>Gráfico 1-3.</b>	Personas que solicitaron información para recomendar el destino turístico .....	43
<b>Gráfico 2-3.</b>	Aciertos en las recomendaciones por parte de los ciudadanos. ....	44
<b>Gráfico 3-3.</b>	Aciertos en las recomendaciones utilizando el sistema. ....	44

## ÍNDICE DE ANEXOS

**Anexo A:** Certificado de experto humano sobre resultados de inferencia.

**Anexo B:** Manual de usuario.

## ÍNDICE DE ABREVIATURAS

MSF	Microsoft Solution Framework
GAD	Gobierno Autónomo Descentralizado
ISO	International Organization for Standardization
MVC	Model, View, Controller
MTV	Model, Template, View
DRY	Don't Repeat Yourself
SQL	Structured Query Language
DBMS	Database Management System
HTML	HyperText Markup Language
URL	Uniform Resource Locator
IEC	International Electrotechnical Commission
CASE	Computer Aided Software Engineering
UML	Unified Modeling Language
CSS	Cascading Stylesheets
ORM	Object-Relational Mapping
RNAs	Redes Neuronales Artificiales

## **RESUMEN**

El siguiente trabajo tuvo como objetivo el desarrollo de un sistema inteligente web adaptativo para la recomendación de destinos turísticos en el cantón Morona, el sistema se diseñó con la finalidad de automatizar la recomendación de destinos turísticos en el cantón. El desarrollo de la aplicación se realizó utilizando la metodología Microsoft Solution Framework, que permitió la división del proyecto en iteraciones que generan entregables funcionales. La aplicación se desarrolló utilizando el entorno de desarrollo integrado (IDE) Pycharm for Education, con el cual fue posible la utilización de entornos virtuales para el desarrollo de la aplicación en Python. Se utilizó el paradigma orientado a objetos y la división en módulos de la aplicación desarrollada. Para verificar la exactitud de las recomendaciones turísticas, se probó la aplicación realizando 30 ejecuciones de la encuesta de selección de destinos, los resultados de esta inferencia fueron posteriormente validados por el usuario, obteniendo una aceptación satisfactoria de las recomendaciones brindadas por el sistema. De esta forma se pudo concluir que la utilización de un sistema inteligente permite que la recomendación de destinos turísticos se realice sin la presencia de un experto humano en turismo. Para asegurar la fiabilidad de la inferencia, se recomienda al personal de la oficina de turismo la realización de auditorías periódicas a la información ingresada por los usuarios encargados de la gestión del conocimiento.

**PALABRAS CLAVE:** <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INGENIERÍA DE SOFTWARE>, <TURISMO>, <PROGRAMACIÓN DE APLICACIONES WEB>, <MORONA (CANTÓN)>, <PYTHON (SOFTWARE)>, <SISTEMA EXPERTO>.

## **ABSTRACT**

This work aimed at the development of an adaptive and intelligent web system for the recommendation of tourist destinations in Morona canton, the system was designed with the purpose of automating the recommendation of tourist destinations in the canton. The development of the application was conducted using the Microsoft Solution Framework methodology, which allowed the division project in iterations that generate deliverable functions. The application was developed using the Integrated Development Environment (IDE) Pycharm for Education, which was made possible to use of virtual environments for the development of the application in Python. The oriented paradigm was used to objects and the division into modules of the developed application. To verify the accuracy of the tourist recommendations, the application was tested by carrying out 30 executions of the survey of the selection of destination, the results of this inference were subsequently validated by the user, obtaining a satisfactory acceptance of the recommendations provided by the system. In this way, it was possible to conclude that the use of an intelligent system allows the recommendation of tourist destinations are made without the presence of a human expert in tourism. To ensure the reliability of the inference, the staff of the tourism office is recommended to carry out periodic audits of the information entered by the users in charge of the management of the knowledge.

**KEY WORDS:** <TECHNOLOGY AND ENGINEERING SCIENCE>, <SOFTWARE ENGINEERING>, <TOURISM>, <PROGRAMMING OF WEB APPLICATIONS>, <MORONA CANTON>, <PYTHON SOFTWARE>, <EXPERT SYSTEM>.

## **INTRODUCCIÓN**

El presente documento comprende el desarrollo de un sistema inteligente web adaptativo para la recomendación de destinos turísticos en el cantón Morona basado en reglas y hechos. El sistema es una aplicación inteligente que tiene como principal funcionalidad el recomendar destinos turísticos a los usuarios mediante una interfaz web.

El capítulo 1, describe los conceptos correspondientes al sector turístico, la inteligencia artificial, los sistemas expertos y las herramientas tecnológicas utilizadas tales como Python, Django y Postgresql. Adicionalmente, se define la norma ISO (*International Organization for Standardization*) 25010 utilizadas para realizar la medición de la calidad del software.

En el capítulo 2, se detalla el desarrollo de la aplicación, utilizando la metodología *Microsoft Solution Framework*. Se define el alcance y visión del proyecto, los perfiles de usuario y adicionalmente se especifican requerimientos funcionales. Se realizan las pruebas automatizadas de software según el plan de pruebas. Con la asistencia de un experto en turismo del Gobierno Autónomo Descentralizado del Cantón Morona, se construyó la base de conocimientos y las reglas de inferencia para la recomendación de los destinos turísticos.

El capítulo 3, muestra los resultados obtenidos al realizar las pruebas de inferencia, y la medición de la exactitud en las recomendaciones turísticas utilizando el sistema experto; así como también la medición sin la utilización de la aplicación.

En la parte final del documento, se destallan las conclusiones obtenidas durante la ejecución del proyecto, así como también las recomendaciones.

## **ANTECEDENTES**

La oficina de Turismo del Gobierno Autónomo Descentralizado del Cantón Morona tiene como principal objetivo el promover la actividad turística en el cantón, mediante la ejecución de campañas de información y un punto fijo de información turística ubicada en la calle Domingo Comín junto al Consejo Provincial. La recomendación de destinos turísticos, lugares de comida y de diversión se realiza, en su mayoría, por parte de la ciudadanía por lo que carecen de una base técnica.

Según la encargada de la oficina de turismo, la Lic. Yadira Delgado, esto genera efectos negativos ya que quienes recomiendan los lugares turísticos no lo hacen utilizando datos técnicos ni exactos causando que los turistas no consigan la experiencia que esperaban conseguir.

El principal problema radica en la falta de difusión de los destinos y atractivos turísticos en medios digitales tales como redes sociales o aplicaciones web en la que consten los servicios y valores de

los diferentes destinos turísticos; ya que esta información es asequible únicamente en el punto de información turística de la municipalidad.

### **FORMULACIÓN DEL PROBLEMA**

¿Cómo ayudará el desarrollo de un sistema inteligente web adaptativo basado en reglas y hechos a la recomendación de destinos turísticos del cantón Morona?

### **SISTEMATIZACIÓN DEL PROBLEMA**

¿Cómo se realiza la recomendación de destinos turísticos actualmente?

¿Qué son los sistemas de Inteligencia Artificial?

¿Es posible desarrollar un sistema experto web adaptativo para la recomendación de destinos turísticos?

¿Qué tan precisas son las recomendaciones de destinos turísticos de un sistema experto web?

### **JUSTIFICACIÓN**

#### **JUSTIFICACIÓN TEÓRICA**

El artículo 3, literal b de la Ley de Turismo indica: “La participación de los gobiernos provincial y cantonal para impulsar y apoyar el desarrollo turístico, dentro del marco de la descentralización” esta sección de dicha ley; en tal virtud, los Gobiernos Autónomos Descentralizados tienen que brindar las facilidades y mecanismos para el correcto desarrollo de la actividad turística. Brindando apertura a los distintos proyectos que tengan como finalidad la de brindar apoyo al sector turístico de la provincia o el cantón.

Según el anuario estadístico del Ministerio de Turismo, en los periodos comprendidos desde el 2011 al 2015, el sector hotelero y alimentario de la provincia de Morona Santiago representa unos ingresos que oscilan apenas entre el 0.2% y el 0.36% respectivamente del total nacional. Lo cual se ve reflejado en que, en el final del periodo, el sector turístico de la provincia generó apenas 779 plazas de trabajo e ingresos económicos de 291.047 dólares.

La utilización de sistemas inteligentes que simulan el comportamiento de una persona experta en el área en que se desempeña han sido utilizados en las áreas de agricultura, medicina, educación, etc, brindando resultados favorables. Entre otros, destacan:

- Nieve, MG, Fallat, RJ, Tyler, WR, Hsu, SP, “Consulta pulmonar: concepto a la aplicación de un sistema experto”, Revista de Ingeniería Clínica 13:03, págs. 201 a 205, de 1988
- Richard E. Planta, Nicholas D. duela, "Los sistemas de conocimiento basados en la agricultura", McGraw-Hill, 1991.

- Zorica Nedic, Nedic Vladimir y Jan Machotka (2002) Sistema Tutor Inteligente para la enseñanza de primero de ingeniería, las transacciones Mundial de Ingeniería y Tecnología, Vol. 1, N° 2, 2002

En el área de turismo, se han realizado trabajos titulados: “Diseño e implementación de un Sistema experto de Software para brindar información turística en el estado de Guanajuato.” y “Desarrollo de un Sistema Experto en Turismo para la Ciudad de Bogotá” en México y Colombia respectivamente; en ambos casos, los sistemas están desarrollados con la finalidad de realizar una recomendación acerca de los destinos según las preferencias e información proporcionada por el turista.

### **JUSTIFICACIÓN APLICATIVA**

La propuesta está enfocada en el desarrollo de un Sistema Inteligente Web adaptativo para la recomendación de destinos turísticos en el cantón Morona basado en reglas y hechos. Que permita mostrar la información de los distintos destinos turísticos, así como también recomendar (basadas en un grado de probabilidad) uno de esos destinos a los turistas.

El sistema será desarrollado utilizando la metodología Microsoft Solution Framework ya que permite la gestión iterativa controlada por hitos. El ciclo de vida del software está orientado a la gestión temprana de riesgos utilizando pocos recursos humanos.

Se utilizará el lenguaje de programación python ya que este lenguaje se caracteriza por su velocidad de ejecución y flexibilidad. Como característica adicional, es un lenguaje multiplataforma.

La persistencia de los datos se realizará con PostgreSQL. Es un motor de gestión de base de datos probado en varios entornos por usuarios de diferente nivel de conocimiento técnico. Permite las transacciones y cuenta con amplia información oficial.

En todos los casos, las herramientas a utilizar no tienen son gratuitas por lo que se reduce el costo de desarrollo de la aplicación.

### **Funciones del Sistema:**

- Presentar la información (ubicación, galería multimedia, precios y servicios) de los destinos turísticos registrados.
- Elegir el destino turístico con mayor probabilidad de satisfacción del usuario mediante el motor de inferencia.
- Permitir la creación y validación de cuentas de usuario y la generación de reportes.



## **OBJETIVOS**

### **OBJETIVO GENERAL.**

Desarrollar un Sistema Inteligente Web adaptativo para influir en la precisión de las recomendaciones de destinos turísticos en el cantón Morona.

### **OBJETIVOS ESPECÍFICOS.**

- Analizar la forma en que se realiza actualmente la recomendación de destinos turísticos en el cantón Morona.
- Investigar acerca de los sistemas de inteligencia artificial.
- Implementar el sistema inteligente web adaptativo para la recomendación de destinos turísticos en el cantón Morona.
- Medir la precisión de las recomendaciones de destinos turísticos dadas por el sistema experto.

## CAPÍTULO I

### 1. MARCO TEÓRICO

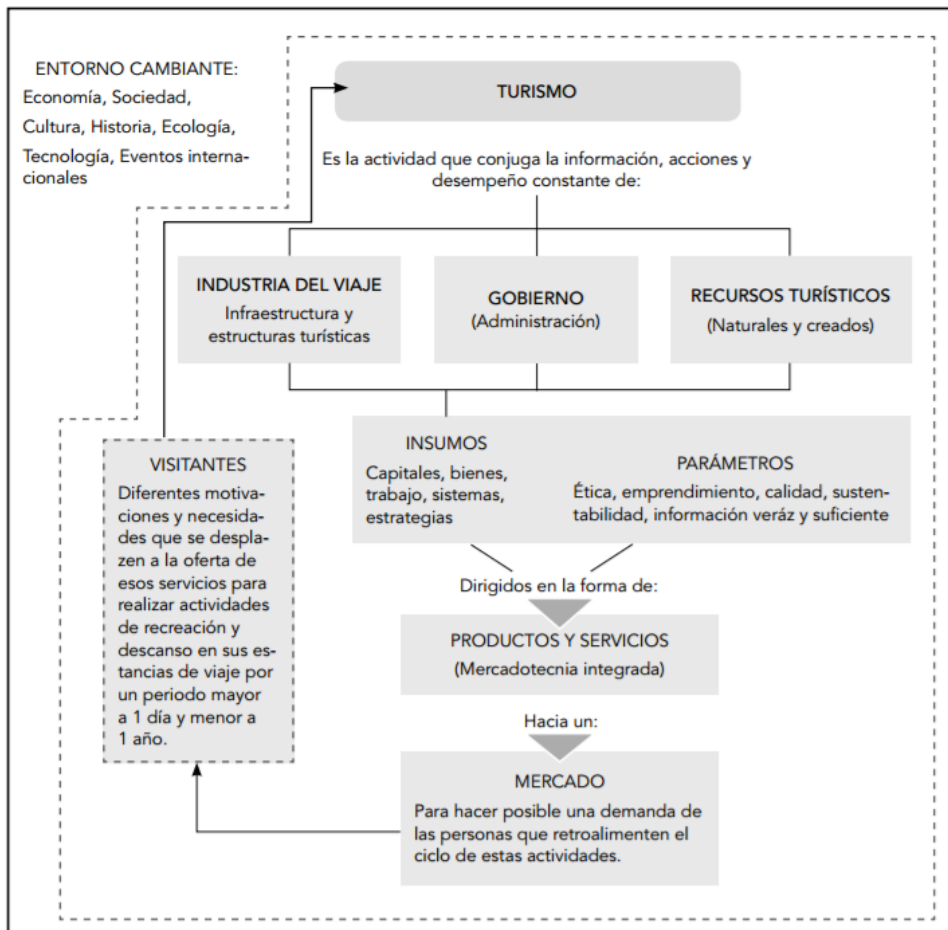
#### 1.1. Turismo

*“La conceptualización del turismo ha causado controversias debido a las variadas interpretaciones que se le han brindado a esta actividad. Las controversias surgen unas por las definiciones es que brindan algunas disciplinas que ven en el turismo un campo de estudio y otras que se basan en puntos de vista de algunos tratadistas que pretenden explicar este fenómeno de la ideología.” (Fuentes, 2013: p.59)*

Estas definiciones han evolucionado conforme la misma sociedad lo ha hecho para considerar aspectos que antes no se consideraron. Miguel Ángel Acerenza ha estudiado las diferentes conceptualizaciones del turismo armando una definición bastante completa:

*“El turismo para la sociedad es, en esencia, una actividad relacionada con el descanso, el placer, el desarrollo cultural y la recreación. [...] el turismo para la sociedad constituye una forma particular del uso del tiempo libre y una forma particular de recreación, pero que no cubre todas las formas de uso que las personas pueden hacer de su tiempo libre, ni todas las formas posibles de recreación.” (Acerenza, 2006: p.27)*

*“El concepto de la Organización Mundial del Turismo (1991) revisó y actualizó sus conceptos estableciendo que el turismo comprende las actividades de personas que viajan y permanecen en lugares fuera de su ambiente usualmente durante no más de un año consecutivo con fines de gozar de tiempo libre, negocios u otros.” (Fuentes, 2013: p.62)*



**Figura 1-1.** Gráfica de definición de turismo.

Fuente: Guerrero y Ramos, 2014: p.33

## 1.2. Sector Turístico en Ecuador

La población del Ecuador es de aproximadamente catorce millones de habitantes y cuenta con una gran cantidad de recursos culturales y sociales en el interior de sus fronteras. Está habitado por un 65% de mestizos, 25% de indígenas, 10% de caucásicos y 7% de africanos. (Fuentes, 2013: p.121)

Ecuador cuenta con una frontera cultural bien marcada dividida entre las tres regiones continentales: Sierra, Costa y Amazonía con una gran densidad poblacional en la Costa y la Sierra. (Fuentes, 2013: p.121)

Tiene un calendario de eventos folclóricos y festividades religiosas tan diversas como su misma población. Cuenta con muchas festividades anuales correspondientes a la fundación e independencia de las localidades y también en honor a la Virgen y a los santos. Estas festividades se caracterizan por la fusión entre las tradiciones indígenas y católicas. (Fuentes, 2013: p.122)

Cuenta con una historia de cuatro décadas de acciones de conservación de su naturaleza desde que en 1959 la Islas Galápagos como área protegida. En la actualidad, el Sistema Nacional de Áreas Protegidas tiene a su cargo 45 áreas protegidas que representan el 18.5% del territorio nacional con una superficie de aproximadamente 5 hectáreas. (Fuentes, 2013: p.124)

La principal actividad predominante es la de Alimentos y Bebidas representando el 60% seguida por alojamiento (23%) y agencias de viaje (9%); el 8% de las demás actividades se reparte entre recreación, transporte y empresas de promoción turística. (Fuentes, 2013: p.124)

Según el anuario estadístico del Ministerio de Turismo, en los periodos comprendidos desde el 2011 al 2015, el sector hotelero y alimentario de la provincia de Morona Santiago representa unos ingresos que oscilan apenas entre el 0.2% y el 0.36% respectivamente del total nacional. Lo cual se ve reflejado en que, en el final del periodo, el sector turístico de la provincia generó apenas 779 plazas de trabajo e ingresos económicos de 291.047 dólares.

### **1.3. Inteligencia Artificial**

En una primera aproximación podemos definir la Inteligencia Artificial como una ciencia que tiene como objetivo el diseño y construcción de máquinas capaces de imitar el comportamiento inteligente de las personas. Una rama especializada de la informática que investiga y produce razonamiento por medio de máquinas automáticas y que pretende fabricar artefactos dotados de la capacidad de pensar. (Álvarez, 1994: p 19)

El nacimiento de la AI como disciplina de investigación se remonta a 1956, durante una conferencia de informática teórica que tuvo lugar en Dartmouth College (Estados Unidos). A dicha conferencia asistieron algunos de los científicos que se encargarían de desarrollar la disciplina en diferentes ámbitos y dotarla de una estructura teórica y computacional apropiada. (Benítez; 2013: p6)

Según Turing (1950; citado en García, 2012: p.2) se debe considerar a una máquina como inteligente si esta es capaz de actuar como un humano. Para ello propone una prueba llamada Test de Turing que permite comprobar si una máquina es o no inteligente.

El Test de Turing aún es utilizado ya que exige lo que actualmente es conocido como inteligencia artificial: Reconocimiento del lenguaje natural, razonamiento, aprendizaje y representación del conocimiento (García, 2012: p.2)

*“En la actualidad [...] empresas como Google, gobiernos y otras instituciones invierten grandes cantidades de dinero. Mucha culpa de todo esto la tiene internet, cuya gran cantidad de información facilitan el acceso a grandes cantidades de datos analizables y a su vez demanda nuevas técnicas que permitan manejar tales cantidades de información.*

[...]

*La empresa automovilística también trabaja actualmente en coches autónomos que se conducen solos que son capaces de prever situaciones complicadas y actuar en consecuencia.”* (García, 2011: p.6)

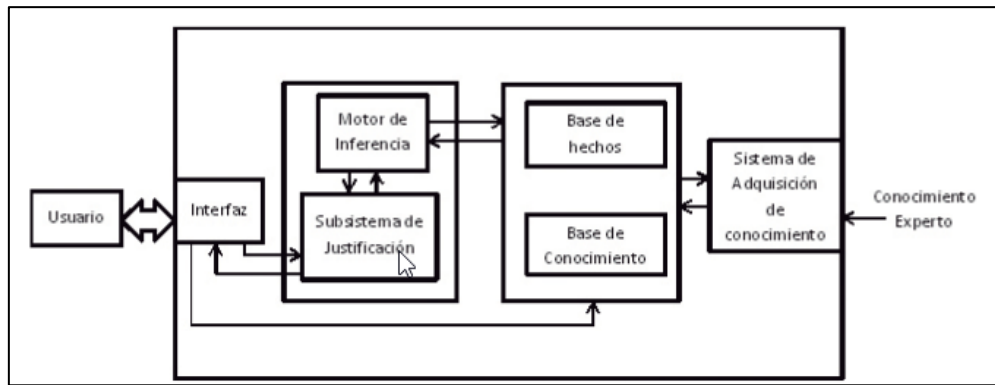
### **1.3.1. Sistemas Expertos**

Según Pinos (2001: p.10) un sistema experto se define como la aplicación que contiene el conocimiento de un especialista humano experimentado en un determinado campo de aplicación.

Según Rossini (2000; citado en Badaró et al, 2013 p.351), los sistemas expertos pueden ser considerador como un subconjunto de la inteligencia artificial.

Turban (1995; citado en Badaró et al, 2013 p.351-354) afirma que los sistemas expertos pueden funcionar mejor que cualquier experto humano tomando decisiones de forma individual en un área específica de pericia y están compuestos por cuatro componentes básicos: Subsistema de adquisición de conocimiento, base de conocimiento, base de hechos, el motor de inferencias y el subsistema de justificación. (Pino, 2001: p.10).

Los sistemas expertos tienen ventajas sobre los humanos. El conocimiento contenido en los sistemas expertos es más fácil de documentar y de transferir que el de los expertos humanos. Dicho conocimiento es permanente en el tiempo. (Pino, 2001: p.10).



**Figura 2-1.** Estructura de un Sistema Experto

Fuente: Badaró et al, 2013: p.351

### 1.3.2. *Lógica difusa*

La lógica difusa (*fuzzy logic*) es considerada una generalización de la teoría general de conjuntos que permite que elementos de un universo tengan grados intermedios de pertenencia a conjunto por medio de una función característica. (Pérez, 2007: p.12).

Según Morales-Luna (2002: p.1) se debe hablar de la lógica difusa en plural, y son lógicas de valor múltiple que extiende de las lógicas clásicas. Mientras que en las lógicas clásicas se tiene valores de verdadero o falso, en las lógicas difusas se busca crear aproximaciones matemáticas que permitan resolver problemas. Lo difuso puede entenderse como la posibilidad de tener valores de verdad distintos a verdadero o falso.

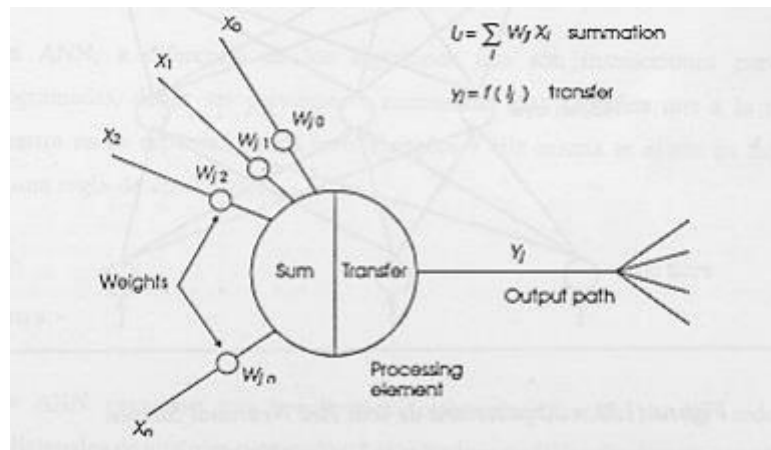
### 1.3.3. *Sistemas Expertos basados en Redes Neuronales*

*“Las redes neuronales artificiales (RNAs) son modelos computacionales que surgieron como un intento de conseguir formalizaciones matemáticas acerca de la estructura del cerebro. Los (RNAs) imitan la estructura hardware del sistema nervioso, concentrándose en la estructura del cerebro humano, basado en el aprendizaje a través de la experiencia, con la consiguiente extracción de conocimiento a partir de la misma.”* (Flores, 2008: p.16).

La capacidad de aprendizaje adaptativo es una de las características más atractivas de redes neuronales. Esto es, aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos. Como las redes neuronales pueden aprender a diferenciar patrones mediante ejemplos y entrenamientos, no es necesario elaborar modelos a priori ni necesidad de especificar funciones de distribución de probabilidad. Las redes neuronales son sistemas dinámicos auto adaptativos. Son adaptables debido a

la capacidad de autoajuste de los elementos procesales (neuronas) que componen el sistema. Son dinámicos, pues son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones (Matich, 2001: p.9).

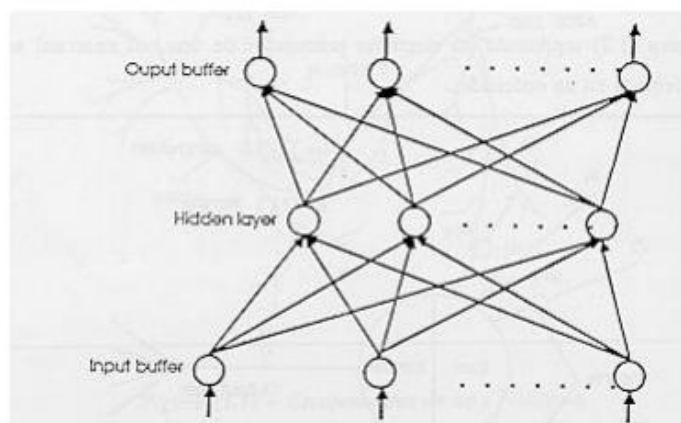
En una RNA la unidad análoga a la neurona biológica es el elemento procesador que tiene varias entradas y las combina con una suma básica. Las entradas son modificadas por una función de transferencia y el valor de la salida al elemento procesador (Basogain, 2008: p.3).



**Figura 3-1.** Diagrama de una Neurona Artificial (PE).

Fuente: Basogain, 2008: p.3

Una red neuronal consiste en un conjunto de unidades elementales conectadas de una forma concreta. Los elementos están organizados en grupos llamados niveles o capas. Una red típica consiste en una secuencia de capas con conexiones entre capas adyacentes consecutivas. Existen dos capas con conexiones con el mundo exterior. Una capa de entrada, buffer de entrada, donde se presentan los datos a la red, y una capa buffer de salida que mantiene la respuesta de la red a una entrada. El resto de las capas reciben el nombre de capas ocultas. (Basogain, 2008: p.4).



**Figura 4-1.** Arquitectura de una Red Neuronal Simple.

Fuente: Basogain, 2008: p.4

#### ***1.3.4. Sistemas Expertos Basados en Reglas y Hechos***

Los sistemas expertos basados en reglas y hechos tratan problemas deterministas, esto quiere decir que sacan sus conclusiones basándose en un conjunto de reglas utilizando un mecanismo de razonamiento lógico. (Castillo, 1996: p.11)

Según la definición de Castillo (Castillo, 1996: pp.11-12) los componentes principales de los sistemas expertos basados en reglas y hechos son los siguientes:

- **La Base de Conocimiento.**

Mientras el conocimiento es permanente, los datos no forman parte permanente de un sistema y son desechados luego de utilizarlos.

- **Subsistema de Adquisición de Conocimiento.**

Controla el flujo del nuevo conocimiento. El sistema determina qué nuevo conocimiento se necesita y, en caso necesario, incorpora estos conocimientos a la misma.

- **El Motor de Inferencia.**

Es el núcleo de un sistema experto. Su objetivo principal es sacar conclusiones aplicando el conocimiento a los datos.

En un sistema experto basado en reglas, el motor de inferencia se encarga de seleccionar de la base de conocimiento aquellas que son aplicables. Para ello, emplea la condición de las reglas en un conjunto de hechos (almacenados en la base de datos) y en caso de ser ciertos aplica las reglas obteniendo (infiere) nuevos hechos que se incorporan a la base de datos. Repitiendo este proceso se produce un encadenamiento de conclusiones.

*“El principal problema que se plantea es crear un conjunto de inferencias que nos permita llegar desde la definición inicial del problema a la solución. En este sentido podemos encontrar dos estrategias principales:*

- *Avanzar desde el conjunto de datos o hechos hacia las conclusiones o razonamiento hacia delante. La regla de inferencia necesaria para realizar este tipo de razonamiento es el MODUS PONENS.*
- *Seleccionar una posible conclusión e intentar demostrar su validez encontrando algunas evidencias que lo soporten o razonamiento hacia atrás. La regla de inferencia que se utiliza en este sentido es MODUS TOLLEN.”* (Gámez, 1998: p6)

Los valores de verdad de reglas y hechos son verdadero y falso, mientras que en sistemas que utilizan lógica difusa se hace uso valor de verdad que se encuentran entre el verdadero



o falso. Mientras que las redes neuronales representan un modelo en el que las neuronas aprenden y se forman a sí mismas. Luego del aprendizaje, algunos sistemas representan una solución adecuada al problema mientras que otros no funcionan bien. Generalmente, el entrenamiento de una red neuronal necesita de grandes cantidades de datos y repeticiones y en la actualidad la oficina de turismo del GAD no cuenta con información actualizada ni precisa de todos los destinos turísticos del cantón. Con la implantación futura del sistema

Actualmente, el problema planteado por la unidad de turismo del GAD cantonal de Morona presentaba un modelo bastante parecido al de reglas hechos. Es decir, se planteaba la resolución de la recomendación utilizando preguntas enmarcadas en enunciados del tipo “Si ... actividad, entonces ... recomendar” y “Si ... tipo de personas, entonces ... recomendar”.

Por lo que, para el desarrollo de la aplicación, se eligió desarrollar el sistema experto usando un motor de inferencia basado en reglas y hechos con la regla MODUS PONENS ya que cumple con las especificaciones necesarias planteadas por la experta humana.

## CAPÍTULO II

### 2. DESARROLLO DE UN SISTEMA INTELIGENTE WEB ADAPTATIVO PARA LA RECOMENDACIÓN DE DESTINOS TURÍSTICOS EN EL CANTÓN MORONA BASADO EN REGLAS Y HECHOS.

#### 2.1. Visión y Alcance

##### 2.1.1. Visión del proyecto

Desarrollar una aplicación web que sirva como medio para promocionar y dar a conocer los destinos turísticos del cantón Morona. Presentará la información de los destinos turísticos tales como: ubicación, servicios con los que cuenta y los valores de dichos servicios.

Haciendo uso de una base de conocimientos, el sistema solicitará las preferencias del usuario para posteriormente indicar la probabilidad de satisfacción del turista en destino recomendado.

##### 2.1.2. Perfiles de Usuario

###### **Administrador:**

El perfil de usuario administrador tiene como finalidad gestionar todos los aspectos para el correcto funcionamiento de la aplicación.

Podrá realizar las siguientes actividades:

- Validar cuentas de usuario *premium*.
- Gestionar la base de Conocimiento.
- Gestionar Usuarios
- Gestionar Centros Turísticos.

###### **Usuario Estándar**

El usuario estándar es el usuario final de la aplicación. Será quien proporcionará al sistema la información de sus preferencias para recibir como respuesta un destino turístico y la probabilidad de satisfacción en el mismo.

Podrá realizar las siguientes actividades:

- Modificar su perfil de usuario.

- Solicitar la recomendación de un destino turístico.

### **Usuario Premium.**

El usuario premium es aquel que puede crear un destino turístico en el sistema y posteriormente completar la base de conocimiento de ese destino turístico. Es quien ofrecerá un destino turístico y los distintos servicios que provee.

Podrá realizar las siguientes actividades:

- Crear destino turístico.
- Gestionar el destino turístico creado.
- Gestionar el conocimiento del centro turístico creado.

### **2.1.3. *Ámbito del Proyecto***

La aplicación web fue nombrada “Macas Turística”, la misma sirve para la promoción de los destinos turísticos del Cantón Morona, así como también, el módulo de experticia permite la recomendación de destinos turísticos (con una probabilidad de satisfacción) a los usuarios finales que hagan uso de esta.

La aplicación cumple con los estándares actuales de desarrollo web, haciendo uso de URLs amigables, así como también interfaces adaptativas que permiten la correcta visualización en dispositivos móviles y de escritorio. Funciona de forma correcta en todos los navegadores web modernos que cumplen los estándares de W3C respecto a HTML5.

### **2.1.4. *Herramientas que utilizar***

#### **2.1.4.1. *Python 3.5.2***

Es el lenguaje de programación con el que se desarrolla la aplicación.

#### **2.1.4.2. *Django 1.10.3***

Es un framework de Python que permite el desarrollo de aplicaciones web cumpliendo el paradigma de la programación orientada a objetos y de la arquitectura de software a tres capas (modelo-vista-template)

#### **2.1.4.3. *Postgresql 9.5***

Sistema Gestor de Base de Datos que permitirá realizar la persistencia de los datos.

#### **2.1.4.4. *Microsoft Solution Framework***

Es una metodología de desarrollo ágil propuesta por Microsoft que permite el desarrollo de aplicaciones con equipos pequeños.

#### 2.1.4.5. StarUML2.8

Es una herramienta CASE que nos permite realizar los diagramas UML de la aplicación, tales como casos de uso o diagramas de clases.

### 2.1.5. *Objetivos del Proyecto*

#### 2.1.5.1. *Objetivo General:*

Desarrollar un Sistema Inteligente Web adaptativo para la recomendación de destinos turísticos en el cantón Morona basado en reglas y hechos.

#### 2.1.5.2. *Objetivos Específicos:*

- Desarrollar el módulo de gestión de usuarios.
- Desarrollar el módulo de gestión de destinos turísticos.
- Desarrollar el módulo de gestión del conocimiento.

### 2.1.6. *Análisis y gestión de riesgos*

La siguiente tabla presenta los probables riesgos que pueden presentarse en el proyecto.

**Tabla 1-2** Riesgos del proyecto.

<b>ID</b>	<b>RIESGO</b>	<b>CATEGORÍA</b>
R1	El usuario Premium no gestiona la información correspondiente a su destino turístico de forma correcta.	R. Negocios
R2	Mayor número de usuarios del software de lo planificado inicialmente	R. Técnico
R3	La implementación del sistema en un entorno real podría presentar errores antes inexistentes.	R. Proyecto
R4	La utilización del sistema es más baja de lo esperado.	R. Negocios
R5	Los usuarios del sistema no cuentan con los conocimientos necesarios en informática para la utilización del sistema.	R. Negocios
R6	La aplicación podría no incentivar al turista a visitar los destinos turísticos.	R. Proyecto
R7	No existe el equipo hardware adecuado para la implementación de la aplicación	R. Técnico

Realizado por: Hurtado J, Brayan Z. 2018

2.1.6.1. *Determinación de la exposición del riesgo.*

**Tabla 2-2** Exposición de Riesgos

ID	PROBABILIDAD			IMPACTO		EXPOSICIÓN	
	PORCENTAJE	DESCRIPCIÓN	VALOR	IMPACTO	VALOR	EXPOSICIÓN	VALOR
R1	30%	BAJA	1	BAJO	1	BAJA	1
R2	60%	MEDIA	2	MEDIA	2	MEDIA	4
R3	10%	BAJA	1	CRÍTICO	4	MEDIA	4
R4	50%	MEDIA	2	CRÍTICO	4	ALTA	8
R5	30%	BAJA	1	ALTA	3	MEDIA	3
R6	40%	MEDIA	2	CRÍTICO	4	ALTA	8
R7	15%	BAJA	1	CRÍTICO	4	MEDIA	4

Realizado por: Hurtado J, Brayan Z. 2018

2.1.6.2. *Determinación de la prioridad del riesgo*

**Tabla 3-2** Prioridad de Riesgos

ID	EXPOSICIÓN		PRIORIDAD
	EXPOSICIÓN	VALOR	
R6	ALTA	9	1
R4	ALTA	8	2
R2	MEDIA	8	3
R3	MEDIA	6	4
R5	MEDIA	6	4
R7	MEDIA	4	5
R1	BAJA	4	5

Realizado por: Hurtado J, Brayan Z. 2018

**Tabla 4-2** Código de color de la exposición del riesgo

<b>CÓDIGO DE COLOR</b>	<b>EXPOSICIÓN DEL RIESGO</b>
VERDE	BAJA
AMARILLO	MEDIA
ROJO	ALTA

Realizado por: Hurtado J, Brayan Z. 2018

**Tabla 5-2** Gestión del Riesgo R6

<b>HOJA DE INFORMACIÓN DEL RIESGO</b>			
<b>ID DEL RIESGO:</b> R6		<b>FECHA:</b> 05/10/2017	
<b>PROBABILIDAD:</b> 40% (MEDIA)	<b>IMPACTO:</b> 4 (CRÍTICO)	<b>EXPOSICIÓN:</b> 8 (ALTA)	<b>PRIORIDAD:</b> 1
<b>DESCRIPCIÓN:</b> La utilización del sistema es más baja de lo esperado.			
<b>REFINAMIENTO:</b> Dado que la utilización del sistema es más baja de lo esperado entonces existe la preocupación de que la aplicación sea descartada y no se planeé darle soporte futuro.			
<b>CAUSAS:</b> <ul style="list-style-type: none"> <li>Falta de publicidad de la aplicación.</li> <li>Exista alguna otra aplicación que posea una funcionalidad similar.</li> </ul>			
<b>CONSECUENCIAS:</b> <ul style="list-style-type: none"> <li>El sistema no será descartado.</li> </ul>			
<b>REDUCCIÓN:</b> <ul style="list-style-type: none"> <li>Dar a conocer a la aplicación a la población turista.</li> </ul>			
<b>SUPERVISIÓN:</b> <ul style="list-style-type: none"> <li>Verificar y controlar que la información acerca de la aplicación sea publicitada.</li> </ul>			
<b>GESTIÓN:</b> <ul style="list-style-type: none"> <li>Solicitar el apoyo de las autoridades competentes para difundir información de la aplicación.</li> </ul>			
<b>ESTADO ACTUAL:</b> FASE DE REDUCCIÓN INICIAL X			

<p>FASE DE SUSPENSIÓN</p> <p>FASE DE GESTIÓN</p>
<p><b>RESPONSABLES</b></p> <p>Julio Hurtado</p> <p>Brayan Zabala</p>

Realizado por: Hurtado J, Brayan Z. 2018

**Tabla 6-2** Gestión del Riesgo R4

<b>HOJA DE INFORMACIÓN DEL RIESGO</b>			
<b>ID DEL RIESGO:</b> R4		<b>FECHA:</b> 05/10/2017	
<b>PROBABILIDAD:</b> 50% (MEDIA)	<b>IMPACTO:</b> 4 (CRÍTICO)	<b>EXPOSICIÓN:</b> 8 (ALTA)	<b>PRIORIDAD:</b> 2
<b>DESCRIPCIÓN:</b> La aplicación podría no incentivar al turista a visitar los destinos turísticos.			
<p><b>REFINAMIENTO:</b></p> <p>Dado que La aplicación podría no incentivar al turista a visitar los destinos turísticos entonces existe la preocupación de que la aplicación cumpla con las expectativas establecidas en la planificación del proyecto.</p> <p><b>CAUSAS:</b></p> <ul style="list-style-type: none"> <li>• Poco interés por el sector turístico en la aplicación.</li> <li>• La información que brinda la aplicación acerca de los destinos turísticos es insuficiente o inadecuada.</li> </ul> <p><b>CONSECUENCIAS:</b></p> <ul style="list-style-type: none"> <li>• El sistema no será utilizado</li> </ul>			
<p><b>REDUCCIÓN:</b></p> <ul style="list-style-type: none"> <li>• Establecer parámetros de información necesaria que debe tener cada usuario premium de su destino turístico.</li> </ul>			
<b>SUPERVISIÓN:</b>			

<ul style="list-style-type: none"> <li>• Verificar y controlar que la información existente cumpla con los parámetros establecidos.</li> </ul>
<b>GESTIÓN:</b> <ul style="list-style-type: none"> <li>• Cambio en el diseño del software en la manera de presentar la información turística</li> </ul>
<b>ESTADO ACTUAL:</b> FASE DE REDUCCIÓN INICIAL X FASE DE SUSPENSIÓN FASE DE GESTIÓN
<b>RESPONSABLES</b> Julio Hurtado Brayan Zabala

Realizado por: Hurtado J, Brayan Z. 2018

#### 2.1.7. Planificación Inicial

#### 2.1.8. Estudio de Factibilidad

##### Factibilidad Técnica

- **Hardware Requerido**

**Tabla 7-2** Hardware requerido

Cantidad	Descripción
2	Computador Portátil: <ul style="list-style-type: none"> <li>• Memoria 4GB o superior</li> <li>• Disco duro 500GB o superior</li> <li>• Procesador Intel Core i5 (o equivalente) o superior</li> </ul>
1	Cámara Digital
1	Smartphone
1	Impresora a Color
1	Unidad de Almacenamiento USB

Realizado por: Hurtado J, Brayan Z. 2018



- **Software Requerido**

**Tabla 8-2** Software requerido

<b>Cantidad</b>	<b>Descripción</b>
1	Motor de Base de Datos Postgresql
1	Herramienta CASE “Pycharm”
1	Herramienta CASE “Star UML”
1	Servidor Web Apache
1	Lenguaje de Programación Python
1	Framework Django

Realizado por: Hurtado J, Brayan Z. 2018

**Factibilidad Operativa**

La aplicación contará con interfaces adaptativas e intuitivas, lo que permitirá su utilización en dispositivos móviles y de escritorio; además, será de fácil uso por lo que el usuario final no necesitará grandes conocimientos para poder utilizarla.

**Factibilidad Legal**

La aplicación web desarrollada no incumple con ningún reglamento u ordenanza emitido por el GAD Municipal de Morona, así como tampoco con ningún reglamento de ley, ley, ley orgánica o constitución de la república.

Al contrario, la aplicación tiene como una finalidad aportar con el cumplimiento de algunos de los objetivos del Plan Nacional del Buen Vivir y del artículo 3, literal b de la Ley de Turismo.

**Tabla 9-2** Factibilidad económica

<b>Hardware</b>			
<b>Cantidad</b>	<b>Descripción</b>	<b>Precio Unitario</b>	<b>Precio Total</b>
2	Computadores para el Desarrollo de la Aplicación	500.00	1000.00
1	Smartphone	250.00	250.00
1	Pendrive	10.00	10.00
<b>Software</b>			
<b>Cantidad</b>	<b>Descripción</b>	<b>Precio Unitario</b>	<b>Precio Total</b>
1	DBMS Postgresql	0.00	0.00
1	Pgadmin	0.00	0.00

1	StarUML	49.00	49.00
1	Pycharm	89.00	89.00
1	Sublime Text 3	0.00	0.00
1	Django Framework	0.00	0.00
<b>Recursos Humanos</b>			
<b>Cantidad</b>	<b>Descripción</b>	<b>Precio Unitario</b>	<b>Precio Total</b>
2	Desarrolladores 6 meses	2500.00	5000.00
<b>Otros Gastos</b>			
<b>Cantidad</b>	<b>Descripción</b>	<b>Precio Unitario</b>	<b>Precio Total</b>
2	Transporte 6 meses	50.00	100.00
6	Servicio de internet mensual	25.00	150.00
<b>Costo Total del Proyecto</b>			<b>6648.00</b>

Realizado por: Hurtado J, Brayán Z. 2018

## 2.2. Planificación

## 2.3. Especificación de Requisitos

### 2.3.1. Visión General

En esta etapa se detallan los requerimientos de la aplicación teniendo en cuenta las especificaciones planteadas por el GAD Municipal de Morona. Estos requisitos serán implementados durante la etapa de desarrollo del proyecto.

### 2.3.2. Descripción General

Se caracterizará por ser un Software inteligente basado en reglas y hechos que tomará decisiones asistidas sobre a qué destino turístico debería visitar un usuario, dependiendo de las preferencias de este. El software tomará como criterio para decidir el destino el conocimiento de un experto humano.

### 2.3.3. Perspectiva del Proyecto

La aplicación podrá ser utilizada desde dispositivos móviles y de escritorio gracias a que sus interfaces son adaptativas y se pueden visualizar de forma adecuada en cualquier tamaño de pantalla en navegadores webs modernos.

### 2.3.4. Funciones de la Aplicación

Las principales funciones de la aplicación son:

- Registro y Validación de usuarios
- Creación y modificación de Destinos Turísticos
- Creación, modificación y eliminación de Destinos Turísticos
- Gestionar el Conocimiento
- Recomendar Destino Turístico al Usuario

### 2.3.5. *Limitaciones Generales*

- Se necesita una conexión a internet para acceder a la aplicación

### 2.3.6. *Suposiciones y Dependencias*

- El sistema se desarrollará de forma modular por lo que se podrá agregar nuevas funcionalidades en el futuro; estas podrán ser desarrolladas en cualquier lenguaje de programación posterior al desarrollo de un API como middleware de comunicación.
- Se podrá migrar el sistema para utilizar otro motor de base de datos soportado por Django.

### 2.3.7. *Requisitos Futuros*

- Desarrollo de aplicaciones móviles

### 2.3.8. *Requerimientos funcionales*

#### **Módulo de Gestión de Usuarios**

**Tabla 10-2** Requerimiento: Registro de usuario

Id. Requerimiento	1.1 Registro de usuario
Descripción	Permite a un usuario registrarse en el sistema.
Entradas	Nombres Apellidos Email Nombre de Usuario Contraseña
Salidas	Confirmar registro de usuario.
Proceso	El sistema verificará y validará el nombre de usuario y contraseña. Si los datos son correctos creará la sesión de usuario
Precondiciones	
Postcondiciones	
Efectos Colaterales	
Rol que ejecuta	Usuario

Realizado por: Hurtado J, Brayan Z. 2018

**Tabla 11-2** Requerimiento: Validar usuario

Id. Requerimiento	1.2 Validar usuario
Descripción	Permite al administrador validar un usuario.
Entradas	Id de usuario
Salidas	Confirmar validación de usuario.
Proceso	El sistema modificará el usuario permitiendo que este tenga los permisos necesarios para crear destinos turísticos
Precondiciones	El usuario debe estar activo
Postcondiciones	
Efectos Colaterales	
Rol que ejecuta	Administrador

Realizado por: Hurtado J, Brayan Z. 2018

### Módulo de Gestión de Destinos Turísticos

**Tabla 12-2** Requerimiento: Crear Destino Turístico

Id. Requerimiento	2.1 Crear Destino Turístico
Descripción	Permite al usuario crear un destino turístico
Entradas	Nombre Coordenadas Información Adicional
Salidas	Confirmar registro de destino turístico.
Proceso	
Precondiciones	El usuario debe de estar registrado en el sistema
Postcondiciones	
Efectos Colaterales	
Rol que ejecuta	Usuario Premium

Realizado por: Hurtado J, Brayan Z. 2018

**Tabla 13-2** Requerimiento: Modificar Destino Turístico

Id. Requerimiento	2.2 Modificar destino turístico
Descripción	Permite al usuario modificar la información acerca del centro turísticos.
Entradas	Ubicación del centro Descripción
Salidas	Confirmar aplicar cambios.
Proceso	El sistema aplicará los cambios correspondientes.
Precondiciones	Debe existir el destino turístico
Postcondiciones	
Efectos Colaterales	
Rol que ejecuta	Usuario Premium

Realizado por: Hurtado J, Brayan Z. 2018

**Tabla 14-2** Requerimiento: Crear Servicio Turístico

Id. Requerimiento	2.3 Crear servicios turísticos
Descripción	Permite al usuario crear servicios turísticos.
Entradas	Nombre del servicio Precio del servicio. Descripción del servicio.
Salidas	Confirmar creación del servicio.
Proceso	El sistema almacenará el servicio creado.
Precondiciones	El usuario debe estar logueado
Postcondiciones	
Efectos Colaterales	
Rol que ejecuta	Usuario Premium

Realizado por: Hurtado J, Brayan Z. 2018

**Tabla 15-2** Requerimiento: Modificar Servicio Turístico

Id. Requerimiento	2.4 Modificar servicio turístico
Descripción	Permite al usuario modificar servicios turísticos.
Entradas	Id servicio turístico.
Salidas	Confirmar cambios correspondientes.
Proceso	El sistema aplicará los cambios correspondientes.
Precondiciones	El servicio debe de estar previamente creado
Postcondiciones	
Efectos Colaterales	
Rol que ejecuta	Usuario Premium

Realizado por: Hurtado J, Brayan Z. 2018

**Tabla 16-2** Requerimiento: Eliminar Servicio Turístico

Id. Requerimiento	2.5 Eliminar servicio turístico
Descripción	Permite al usuario eliminar servicios turísticos.
Entradas	Id servicio turístico.
Salidas	Confirmar eliminación del servicio.
Proceso	El sistema eliminará el servicio turístico.
Precondiciones	El servicio debe de estar previamente creado.
Postcondiciones	
Efectos Colaterales	
Rol que ejecuta	Usuario Premium

Realizado por: Hurtado J, Brayan Z. 2018

## Módulo de conocimiento

**Tabla 17-2** Requerimiento: Gestión del conocimiento

Id. Requerimiento	3.1 Gestión del conocimiento
Descripción	Permite al usuario ingresar la información del destino turístico
Entradas	Datos del Destino Turístico respecto a la base de conocimientos
Salidas	Confirmar aplicar cambios.
Proceso	El sistema almacenará los valores ingresados.
Precondiciones	
Postcondiciones	
Efectos Colaterales	
Rol que ejecuta	Usuario Premium

Realizado por: Hurtado J, Brayan Z. 2018

**Tabla 18-2** Requerimiento: Módulo de inferencia

Id. Requerimiento	3.2 Módulo de inferencia
Descripción	Permite al sistema sugerir un destino turístico adecuado a las necesidades del usuario.
Entradas	Valores de preferencia ingresados por el usuario.
Salidas	Sugerencia de destino turístico.
Proceso	El sistema inferirá la información obtenida de las preferencias ingresadas por el usuario y sugerirá el destino turístico que se acerque más a sus necesidades.
Precondiciones	La base de conocimientos debe de estar previamente creada.
Postcondiciones	
Efectos Colaterales	
Rol que ejecuta	Usuario

Realizado por: Hurtado J, Brayan Z. 2018

### 2.3.9. *Requerimientos no funcionales*

**SEGURIDAD:** Para acceder a la aplicación el usuario deberá realizar el proceso de autenticación. Se controlarán los permisos de usuario de tal forma que solamente puedan acceder a secciones a los que están autorizados. Se permitirá el cambio de contraseñas de usuario. Además, se podrá realizar una auditoría total sobre las acciones administrativas.

**MANTENIBILIDAD:** El sistema será mantenible por módulos, lo que permitirá una mayor escalabilidad, detección y corrección de errores.

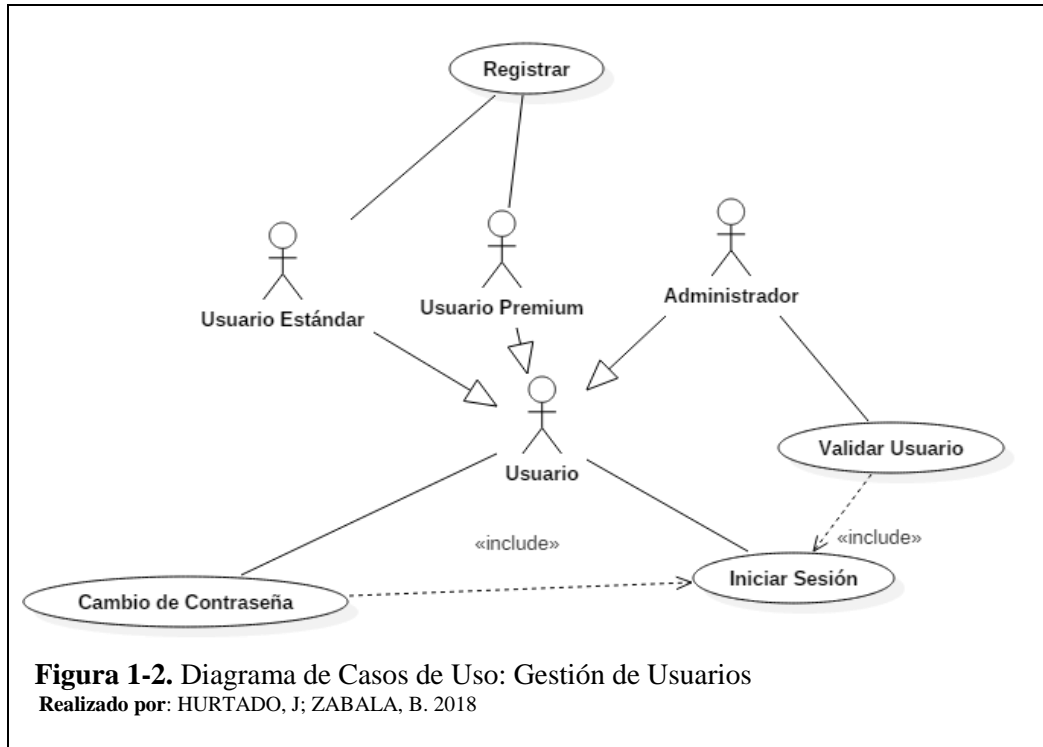
**PORTABILIDAD:** La aplicación tendrá la capacidad de ser ejecutada fácilmente sobre diferentes plataformas de software y hardware en las que originalmente fue diseñada.

**FACILIDAD DE USO:** El sistema hará uso de tecnologías web modernas como HTML5 y CSS3 para que el sitio sea *responsive web design* (adaptativo) y se pueda utilizar fácilmente desde cualquier dispositivo móvil o de escritorio.

2.3.10. Casos de Uso

2.3.10.1. Gestión de Usuarios

Tabla 19-2 Caso de Uso Extendido: Gestión de Usuarios

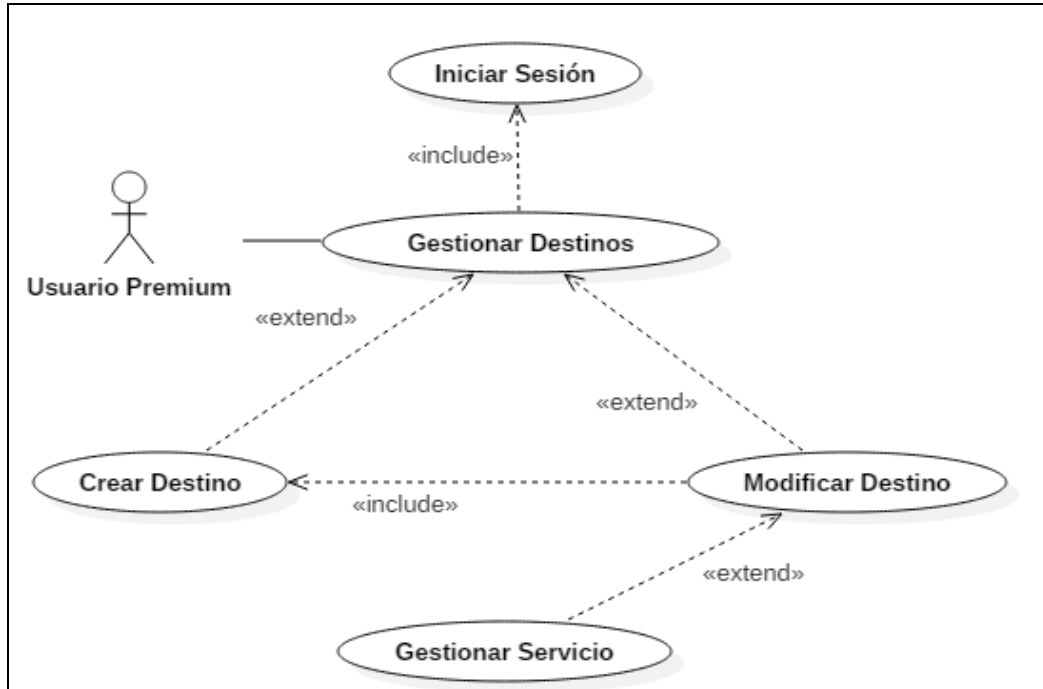


<b>Nombre</b>	Gestión de Usuarios
<b>Actores:</b>	Usuario, Administrador
<b>Propósito:</b>	Gestionar el registro, acceso y validación de usuarios
<b>Pre-condiciones:</b>	Acceder al Sistema
<b>Flujo de Eventos:</b>	<ol style="list-style-type: none"> <li>1. El usuario presiona el botón de “Registrarse”</li> <li>2. Ingresar los datos solicitados (Nombre de usuario, nombres y apellidos, correo, contraseña)</li> <li>3. El usuario inicia sesión                         <ol style="list-style-type: none"> <li>a. Podrá acceder al submódulo de inferencia.</li> <li>b. Si el usuario es un usuario validado, podrá acceder al módulo de gestión de destinos turísticos.</li> <li>c. Si es un usuario administrador accede a la lista de usuarios no validados, es decir, a la lista de usuarios que no pueden crear Destinos Turísticos en el Sistema. Podrá validarlos si es necesario</li> </ol> </li> <li>4. El usuario puede modificar sus datos de perfil.</li> <li>5. El usuario cierra sesión.</li> </ol>
<b>Post-condiciones:</b>	Se registra el usuario en la base de datos Se modifica el usuario en la base de datos

Realizado por: HURTADO, J; ZABALA, B. 2018

2.3.10.2. Gestión de Destinos Turísticos

**Tabla 20-2** Caso de Uso Extendido: Gestión de Destinos Turísticos



**Figura 2-2.** Diagrama de Casos de Uso: Gestión de Destinos Turísticos

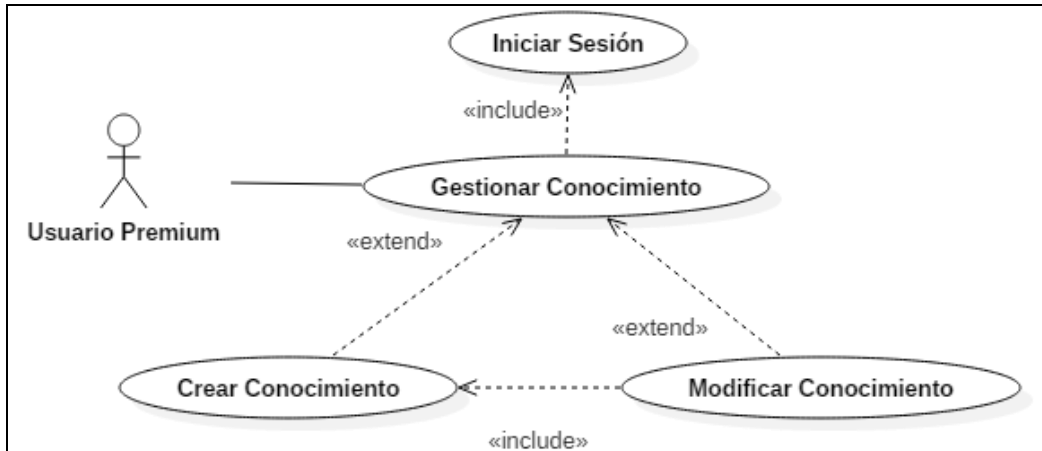
Realizado por: HURTADO, J; ZABALA, B. 2018

<b>Nombre</b>	Gestión de Destinos Turísticos
<b>Actores:</b>	Usuario
<b>Propósito:</b>	Gestionar la creación y modificación de destinos turísticos y servicios
<b>Pre-condiciones:</b>	Acceder al Sistema Cuenta Validada
<b>Flujo de Eventos:</b>	<ol style="list-style-type: none"> <li>1. El usuario accede al sistema mediante sus datos de autenticación</li> <li>2. El sistema verifica que el usuario esté validado</li> <li>3. El sistema verifica que el usuario no haya creado un destino turístico             <ol style="list-style-type: none"> <li>a. Si no ha creado un destino turístico el usuario puede crear uno ingresando el nombre, las coordenadas y demás información</li> <li>b. Si existe un destino para ese usuario puede crear, modificar o eliminar servicios</li> </ol> </li> </ol>
<b>Post-condiciones:</b>	Se crea el registro del destino turístico en la base de datos. Se crea, modifica o elimina el registro de los servicios.

Realizado por: HURTADO, J; ZABALA, B. 2018



**Tabla 21-2** Caso de Uso Extendido: Gestión del conocimiento



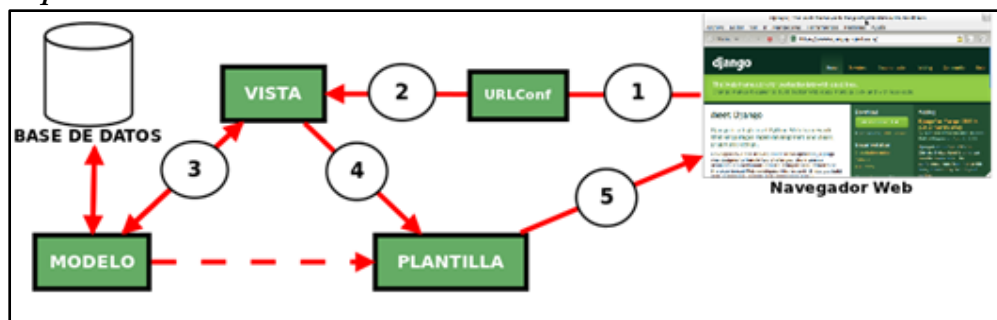
**Figura 3-2.** Diagrama de Casos de Uso: Gestión del Conocimiento

Fuente: HURTADO, J; ZABALA, B. 2016

<b>Nombre</b>	Gestión del Conocimiento
<b>Actores:</b>	Usuario
<b>Propósito:</b>	Crear y modificar el conocimiento de un destino turístico
<b>Pre-condiciones:</b>	Acceder al Sistema Cuenta Validada
<b>Flujo de Eventos:</b>	<ol style="list-style-type: none"> <li>1. El usuario accede al sistema mediante sus datos de autenticación</li> <li>2. El sistema verifica que el usuario esté validado</li> <li>3. Si el usuario cuenta con un destino turístico creado tiene la posibilidad de modificar la base de conocimiento de ese destino turístico.</li> </ol>
<b>Post-condiciones:</b>	Se crea o modifica en la base de datos el registro del conocimiento sobre el destino turístico del usuario

Realizado por: HURTADO, J; ZABALA, B. 2018

### 2.3.11. Arquitectura



**Figura 4-2.** Arquitectura de la Aplicación - Django

Fuente: INFANTE, F. 2012

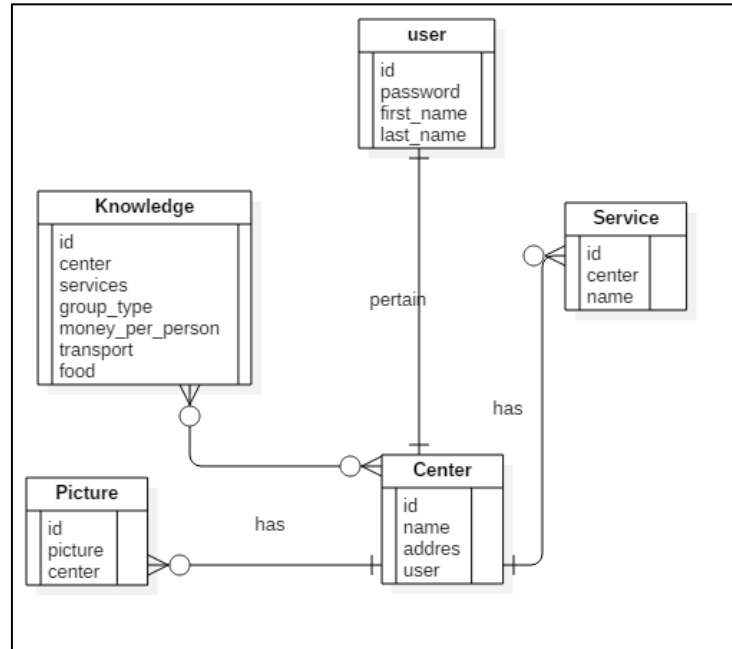
La Figura 1-2 representa la arquitectura de la aplicación y el flujo lógico que sigue una petición desde el cliente (usuario) hacia el servidor de la aplicación.

- El navegador envía una solicitud.
- El URLConf interpreta la solicitud y ubica la vista apropiada.

- La vista interactúa con el modelo para obtener datos.
- La vista llama a una plantilla.
- La plantilla renderiza la respuesta a la solicitud del navegador.

### 2.3.12. Diseño Lógico

#### 2.3.12.1. Diseño lógico de la Base de Datos

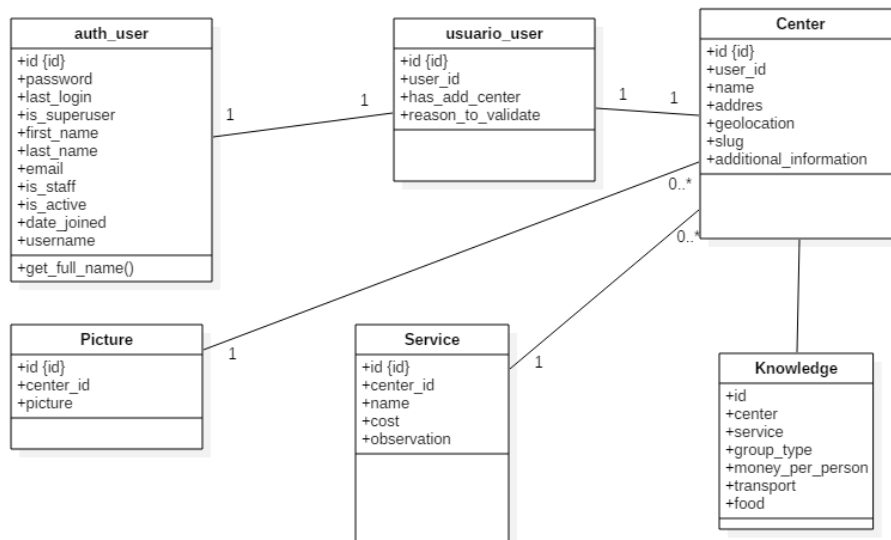


**Figura 5-2.** Diseño Lógico de la Base de Datos  
Realizado por: HURTADO, J; ZABALA, B. 2018

El Gráfico 4-2 muestra la representación lógica de la base de datos, incluyendo las tablas con la que cuenta, los atributos de estas y las relaciones y cardinalidad entre ellas.

### 2.3.13. Diseño Físico

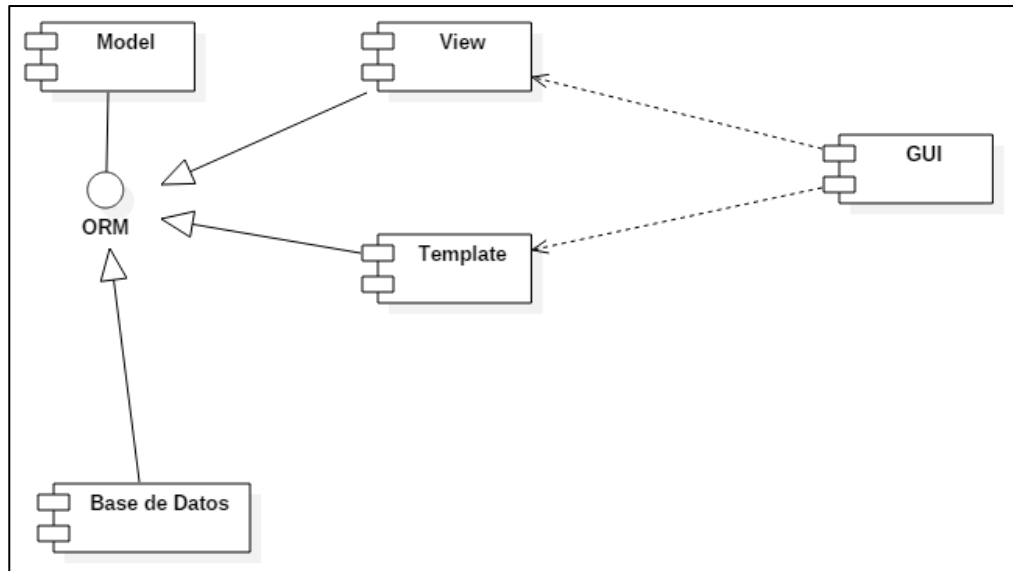
#### 2.3.13.1. Diagrama Físico de la Base de Datos



**Figura 6-2.** Diagrama Físico de la Base de datos  
Realizado por: HURTADO, J; ZABALA, B. 2018

La figura 6-2 es la representación física de la base de datos, en donde se muestran los nombres de las tablas ya implementadas en el motor de gestión de base de datos.

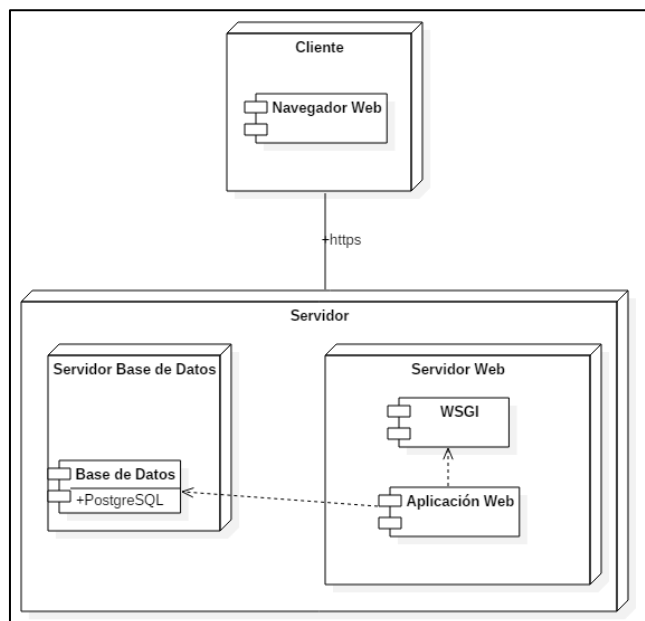
2.3.13.2. *Diagrama de Componentes.*



**Figura 7-2.** Diagrama de Componentes  
Realizado por: HURTADO, J; ZABALA, B. 2018

La Figura 7-2 muestra la forma en que los componentes de la aplicación están relacionados desde el ORM hacia la interfaz de usuario pasando por las vistas y las plantillas.

2.3.13.3. *Diagrama de Despliegue*



**Figura 8-2.** Diagrama de Despliegue  
Realizado por: HURTADO, J; ZABALA, B. 2018

## 2.4. Desarrollo

Para la etapa de desarrollo de la aplicación web se tendrá en cuenta el patrón de diseño Modelo-Vista-Template otorgado por el framework django. Adicionalmente, el nombramiento y definición de variables, clases y métodos seguirán las recomendaciones planteadas por el lenguaje de programación Python.

Los atributos y métodos estarán escritos en inglés. En caso de que estos estén compuestos por más de una palabra, se separarán las palabras con guiones bajos. Esto se aplicará siempre y cuando los nombres sean naturales y representen la función que dichos atributos, métodos y clases representan en la aplicación.

Modelos. Serán nombrados como objeto que representan en singular. Comienza con mayúscula y las demás letras se escribirán en minúscula.

```
class Center(models.Model):
    name = models.TextField(blank=False)
    address = map_fields.AddressField(max_length=200)
    additional_information = models.TextField(blank=True)
    geolocation = map_fields.GeoLocationField(max_length=100)
    slug = AutoSlugField(unique=True, populate_from='name', always_update=True)
    user = models.OneToOneField(User, unique=True, null=True)
```

**Figura 9-2.** Modelo Center

Realizado por: HURTADO, J; ZABALA, B. 2018

```
class Picture(models.Model):
    picture = models.ImageField()
    center = models.ForeignKey(Center)
```

**Figura 10-2.** Modelo Picture

Realizado por: HURTADO, J; ZABALA, B. 2018

```
class Service(models.Model):
    name = models.TextField(blank=False)
    center = models.ForeignKey(Center, null=True)
    cost = models.DecimalField(max_digits=10,
                               decimal_places=2,
                               blank=False,
                               default=00.00)
    observation = models.TextField(blank=False,
                                   default=" ")
```

**Figura 11-2.** Modelo Picture

Realizado por: HURTADO, J; ZABALA, B. 2018

```

class User(models.Model):
    user = models.OneToOneField(Usuario)
    has_add_center = models.BooleanField(default=False)
    reason_to_validate = models.TextField(blank=True)

```

**Figura 12-2.** Modelo Picture

Realizado por: HURTADO, J; ZABALA, B. 2018

Vistas. Las vistas serán todas basadas en clases para aprovechar las bondades del paradigma de la orientación a objetos. Los nombres representarán la función principal de la vista, los nombres compuestos por más de dos palabras, estas se escribirán juntas y comenzarán en mayúscula. Por ejemplo: *UserCreateView*, *ServiceUpdateView*

```

class CenterCreateView(CreateView):
    model = Center
    template_name = 'center_create.html'
    form_class = CenterCreateForm
    context_object_name = 'center'

    @method_decorator(permission_required('usuario.add_center'))
    def dispatch(self, *args, **kwargs):
        return super(CenterCreateView, self).dispatch(*args, **kwargs)

```

**Figura 13-2.** Vista CenterCreateView

Realizado por: HURTADO, J; ZABALA, B. 2018

```

class PictureAdd(PermissionRequiredMixin, FormView):
    permission_required = "usuario.add_center"
    template_name = "add_picture.html"
    form_class = PictureAddForm

```

**Figura 14-2.** Vista PictureAdd

Realizado por: HURTADO, J; ZABALA, B. 2018

```

class CreateServiceView(CreateView):
    model = Service
    template_name = "service_create.html"
    context_object_name = "service"
    fields = ['name', 'center', 'cost', 'observation', ]
    success_url = "/"

```

**Figura 15-2.** Vista CreateServiceView

Realizado por: HURTADO, J; ZABALA, B. 2018

```

class RegisterUserCreateView(CreateView):
    model = User
    template_name = "registro.html"
    form_class = UsuarioForm
    success_url = reverse_lazy('Center:home')

    def get_context_data(self, **kwargs):
        context = super(RegisterUserCreateView, self).get_context_data(**kwargs)
        context['verification'] = True
        context['request'] = self.request
        return context

    def post(self, request, *args, **kwargs):
        self.object = self.get_object
        form = self.form_class(request.POST)
        if form.is_valid():
            user = form.save()
            profile = Profile.objects.get_or_create(user_=user)
            return HttpResponseRedirect(self.get_success_url())
        else:
            return self.render_to_response(self.get_context_data(form=form))

```

**Figura 16-2.** Vista RegisterUserCreateView

Realizado por: HURTADO, J; ZABALA, B. 2018

Atributos. Serán nombrados con sustantivos y se escriben en minúscula. Por ejemplo: *name, age, description*

Métodos. Serán nombrados como verbos y se escriben en minúscula. Por ejemplo: *calculate, save, get\_full\_name*

### Base de Datos

El acceso a datos se realizará mediante el ORM de django, por lo que el acceso a datos está compuesto por los modelos y la configuración de acceso definida en el archivo *settings.py*

### Extensiones de Archivos

**Tabla 22-2** Extensiones de Archivos

Tipo de Archivo	Extensión	Nomenclatura
Python	.py	model.py
Lenguaje de Marcado de Hipertexto	.html	user_list.html
Hojas de Estilos	.css	style.css
Archivos JavaScript	.js	script.js
Imágenes	.jpg	image.jpg
	.png	image.png
	.gif	image.gif
Archivo de Fuente OpenType	.otf	font.otf

Realizado por: HURTADO, J; ZABALA, B. 2018

### Capas de la Aplicación

La aplicación maneja un modelo a tres capas basado en Modelo-Vista-Controlador. Al ser una implementación propia de ese modelo, se conoce como Model-View-Template

en donde *Model* representan la capa de acceso a datos, *View* la capa de negocio y *Template* la capa de presentación.

## 2.5. Estabilización

### 2.5.1. Plan de Prueba

Las pruebas son rutinas simples que verifican el funcionamiento del código comprobando que se produzcan las salidas esperadas. Las pruebas automatizadas permiten verificar que el código siga funcionando de la forma prevista incluso cuando se realizan cambios sin tener que volver a ejecutar las pruebas de forma manual. Permiten identificar e impedir que sucedan problemas. (Django Software Foundation).

Las pruebas se realizarán sobre las funciones principales del sistema y se realizará utilizando la metodología TDD.

#### 2.5.1.1. Función: Listar destinos turísticos.

```
1 class HomeTestCase(TestCase):
2     def setUp(self):
3         self.url = reverse('Center:home')
4         user = User.objects.create_user(
5             username='juliohurtado',
6             first_name="Julio",
7             last_name='Hurtado',
8             email='juliohurtado@email.com',
9             is_active=True,
10            password='examplePass'
11        )
12        self.client = Client()
13
14    def test_unlogged_user_can_view_home_page(self):
15        response = self.client.get(self.url)
16        view = resolve(self.url)
17        self.assertEqual(response.status_code, 200)
18        self.assertEqual(view.func.view_class, CenterListView)
19
20    def test_logged_user_can_view_home_page(self):
21        self.client.login(username='juliohurtado', password='examplePass')
22        self.test_unlogged_user_can_view_home_page()
```

**Figura 17-2.** Código de la Función Listar destinos turísticos.

Realizado por: HURTADO, J; ZABALA, B. 2018

La Figura 17-2 muestra el código que se realizará en cada ejecución de la prueba. El método setUp configura la clase de la prueba, es el primer método que se ejecuta. Posteriormente se ejecutan los métodos test\_, el primero con un usuario sin autenticar y el segundo con un usuario autenticado.

### 2.5.1.2. Función: Listar destinos turísticos gratuitos

```
1 class HomeFreeTestCase(HomeTestCase):
2     def setUp(self):
3         super(HomeFreeTestCase, self).setUp()
4         self.url = reverse('Center:home_free')
```

**Figura 18-2.** Código de la Función Listar destinos turísticos.

Realizado por: HURTADO, J; ZABALA, B. 2018

La Figura 18-2 muestra la clase HomeFreeTestCase que prueba la funcionalidad de listar destinos turísticos gratuitos, es igual a la prueba HomeTestCase (Figura 17-2) salvo en la url. Por ello hereda de la clase HomeTestCase y sobrescribe la url para ambos casos de prueba.

### 2.5.1.3. Función: Detalle de un destino turístico

```
1 class CenterDetailTestCase(TestCase):
2     def setUp(self):
3         user = User.objects.create_user(
4             username='juliohurtado',
5             first_name="Julio",
6             last_name='Hurtado',
7             email='juliohurtado@email.com',
8             is_active=True,
9             password='examplePass'
10        )
11        profile = Profile.objects.create(
12            user=user,
13            has_add_center=True,
14            reason_to_validate='Lorem insup'
15        )
16        center = Center.objects.create(
17            name='Center Name',
18            adres='-2.251579, -78.132993',
19            additional_information='Lorem ipsum dolor sit amet, consetetuer adipiscing elit.',
20            user=profile,
21        )
22        self.url = reverse('Center:center_detail', kwargs={'slug': center.slug})
23        self.client = Client()
24
25    def test_unlogged_user_can_view_detail_center(self):
26        response = self.client.get(self.url)
27        view = resolve(self.url)
28        self.assertEqual(response.status_code, 200)
29        self.assertEqual(view.func.view_class, CenterDetailView)
30
31    def test_logged_user_can_view_detail_center(self):
32        self.client.login(username='juliohurtado', password='examplePass')
33        self.test_unlogged_user_can_view_detail_center()
34
35    def test_detail_center_404(self):
36        self.url = reverse('Center:center_detail', kwargs={'slug': 'errorslug'})
37        response = self.client.get(self.url)
38        self.assertEqual(response.status_code, 404)
```

**Figura 19-2.** Código de la Función Detalle de un destino turístico.

Realizado por: HURTADO, J; ZABALA, B. 2018

La Figura 19-2 muestra el código que se realizará en cada ejecución de la prueba. El método setUp configura la clase de la prueba, es el primer método que se ejecuta. Posteriormente se ejecutan los métodos test\_, el primero prueba los detalles de un destino existente mientras no se está logueado, el segundo mientras está logueado y el tercero verifica que se muestre un error 404 cuando el destino turístico no existe.



#### 2.5.1.4. Función: Crear destino turístico

```
1 class CenterCreateTestCase(TestCase):
2     def setUp(self):
3         user = User.objects.create_user(
4             username='juliohurtado',
5             first_name="Julio",
6             last_name='Hurtado',
7             email='juliohurtado@email.com',
8             is_active=True,
9             password='examplePass'
10        )
11        self.profile = Profile.objects.create(
12            user=user,
13            has_add_center=True,
14            reason_to_validate='Loremp insup'
15        )
16        self.permissions = Permission.objects.get(name='Puede Crear Centros Turísticos')
17        self.client = Client()
18        self.url = reverse('Center:center_create')
19
20    def test_user_with_permission_can_create_center(self):
21        self.client.login(username='juliohurtado', password='examplePass')
22        self.profile.user.user_permissions.add(self.permissions)
23        response = self.client.get(self.url)
24        view = resolve(self.url)
25
26        self.assertEqual(response.status_code, 200)
27        self.assertEqual(view.func.view_class, CenterCreateView)
28
29    def test_user_without_permission_cant_create_center(self):
30        self.client.login(username='juliohurtado', password='examplePass')
31        response = self.client.get(self.url)
32        view = resolve(self.url)
33        self.assertEqual(response.status_code, 302)
34        self.assertEqual(view.func.view_class, CenterCreateView)
```

**Figura 20-2.** Código de la Función Crear destino turístico.

Realizado por: HURTADO, J; ZABALA, B. 2018

La Figura 20-2 muestra el código que se realizará en cada ejecución de la prueba. El método setUp configura la clase de la prueba, es el primer método que se ejecuta. Posteriormente se ejecutan los métodos test\_, el primero ejecuta la prueba de crear un destino turístico cuando se tiene el permiso “Puede Crear Centros Turísticos” en donde el servidor devuelve un estado 200. La segunda prueba el acceder a crear destinos turísticos sin tener el permiso adecuado, que devuelve un código 302 correspondiente a una redirección temporal debido a que el usuario es llevado a la página de login.

### 2.5.1.5. Función: Editar un destino turístico

```
1 class CenterUpdateTestCase(TestCase):
2     def setUp(self):
3         user = User.objects.create_user(username='juliohurtado', first_name='Julio', last_name='Hurtado',
4             email='juliohurtado@gmail.com', is_active=True, password='examplePass')
5         self.profile = Profile.objects.create(user=user, has_add_center=True, reason_to_validate='Lorem insup')
6         self.permissions = Permission.objects.get(name='Puede Crear Centros Turísticos')
7         self.client = Client()
8         self.center = Center.objects.create(name='Center Name', adres='-2.251579, -78.132993',
9             additional_information='Lorem ipsum dolor sit amet, consectetur adipiscing elit.', user=self.profile, )
10        self.url = reverse('Center:center_edit')
11
12
13    def test_user_with_created_center_get(self):
14        self.client.login(username='juliohurtado', password='examplePass')
15        response = self.client.get(self.url)
16        view = resolve(self.url)
17        self.assertEqual(response.status_code, 200)
18        self.assertContains(response, self.center.name)
19        self.assertEqual(view.func.view_class, CenterUpdateView)
20
21    def test_user_without_created_center_cant_edit_a_center(self):
22        otheruser = User.objects.create_user(username='cesarjerves',
23            first_name='Cesar',
24            last_name='Jerves',
25            email='cesarjerves@email.com',
26            is_active=True,
27            password='examplePass')
28        profile = Profile.objects.create(user=otheruser, has_add_center=True, reason_to_validate='Lorem insup')
29        self.client.login(username='cesarjerves', password='examplePass')
30        response = self.client.get(self.url)
31        view = resolve(self.url)
32        self.assertEqual(response.status_code, 302)
33        self.assertEqual(view.func.view_class, CenterUpdateView)
34
35    def test_user_unlogged_cant_edit_center(self):
36        response = self.client.get(self.url)
37        view = resolve(self.url)
38        self.assertEqual(response.status_code, 302)
39        self.assertEqual(view.func.view_class, CenterUpdateView)
40
41    def test_user_with_permission_can_edit_center_post(self):
42        self.client.login(username='juliohurtado', password='examplePass')
43        new_name = self.center.name + 'other.'
44        data = {'name': new_name, 'adres': self.center.adres, 'additional_information': self.center.additional_information, }
45        response = self.client.post(self.url, data=data)
46        self.assertEqual(response.status_code, 302)
47        self.assertEqual(response.wsgi_request._post['name'], new_name)
48
```

**Figura 21-2.** Código de la Función Editar destino turístico.

Realizado por: HURTADO, J; ZABALA, B. 2018

La Figura 21-2 muestra el código que se realizará en cada ejecución de la prueba. El método setUp configura la clase de la prueba, es el primer método que se ejecuta. Posteriormente se ejecutan los métodos test\_, el primero ejecuta la prueba de un usuario que accede a editar un destino turístico si lo tiene creado (se verifica al recibir un código de estado 200). El segundo método prueba a un usuario accediendo a la url de editar destino sin tenerlo creado (se verifica al obtener un código de estado 302 pues es redirigido a la página de creación). El tercer método verifica la petición post al editar un destino, se verifica al obtener el código 302 (redirección a la página de detalle) y que posteriormente se muestre el nuevo nombre para verificar que se haya guardado.

### 2.5.1.6. Función crear y actualizar conocimiento

```
1 class KnowledgeCreateTestCase(TestCase):
2     def setUp(self):
3         user = create_user()
4         self.profile = create_profile(user)
5         self.center = create_center(self.profile)
6         permissions = Permission.objects.get(name='Puede Crear Centros Turísticos')
7         user.user_permissions.add(permissions)
8         self.url = reverse('Center:knowledge_create')
9
10    def test_create_knowledge_with_created_center(self):
11        self.client.login(username='juliohurtado', password='examplePass')
12        response = self.client.get(self.url)
13        view = resolve(self.url)
14        self.assertEqual(response.status_code, 200)
15        self.assertEqual(view.func.view_class, CreateKnowledge)
16        self.assertContains(response, self.center.name)
17
18    def test_create_knowledge_without_created_center(self):
19        response = self.client.get(self.url)
20        view = resolve(self.url)
21        self.assertEqual(response.status_code, 302)
22        self.assertEqual(view.func.view_class, CreateKnowledge)
```

**Figura 22-2.** Código de la Función crear conocimiento.

Realizado por: HURTADO, J; ZABALA, B. 2018

```
1 class KnowledgeUpdateTestCase(KnowledgeCreateTestCase):
2     def setUp(self):
3         self.url = reverse('Center:knowledge_update')
4         super(KnowledgeUpdateTestCase, self).setUp()
```

**Figura 23-2.** Código de la función actualizar conocimiento.

Realizado por: HURTADO, J; ZABALA, B. 2018

La Figura 22-2 y 23-2 muestran el código que se realizará en cada ejecución de la prueba. El método setUp configura la clase de la prueba, es el primer método que se ejecuta. El segundo y tercer método prueban, respectivamente, la funcionalidad de crear conocimiento cuando se ha creado un destino y cuando no. En el primer caso, se verifica con un estado 200 (Muestra correctamente la página esperada) y muestra el nombre del destino creado. El tercer método redirecciona a la página de creación de un destino (estado 302). La segunda figura 22-3 prueba la funcionalidad de actualizar conocimiento.

### 2.5.1.7. Función realizar encuesta y ver resultados

```
1 class TestTestCase(TestCase):
2     def setUp(self):
3         user = create_user()
4         self.profile = create_profile(user)
5         self.center = create_center(self.profile)
6         self.url = reverse('Center:encuesta')
7
8         self.activity = ActivityForKnowledge.objects.create(name='Actividad 1')
9         self.group = GroupTypeForKnowledge.objects.create(name='Group 1')
10        self.food = FoodForKnowledge.objects.create(name='Food 1')
11        self.transport = TransportForKnowledge.objects.create(name='Transport 1')
12
13        self.knowledge = Knowledge.objects.create(center=self.center, money_per_person=10)
14        self.knowledge.activities.add(self.activity)
15        self.knowledge.group_type.add(self.group)
16        self.knowledge.food.add(self.food)
17        self.knowledge.transport.add(self.transport)
18
19
20    def user_can_view_test_get(self):
21        self.client.login(username='juliohurtado', password='examplePass')
22        response = self.client.get(self.url)
23        self.assertEqual(response.status_code, 200)
24        self.assertContains(response, "Conformación del grupo de Turistas")
25        self.assertContains(response, "Actividades que desea desarrollar")
26        self.assertContains(response, "Método de transporte preferido")
27        self.assertContains(response, "Comida preferida")
28        self.assertContains(response, "Dinero máximo por persona")
29
30    def user_can_view_test_post(self):
31        self.client.login(username='juliohurtado', password='examplePass')
32        response = self.client.get(self.url)
33        view = resolve(self.url)
34        self.client.login(username='juliohurtado', password='examplePass')
35
36        data = {
37            'group_type': self.group.id,
38            'activity': self.activity.id,
39            'transport': self.transport.id,
40            'food': self.food.id,
41            'money_per_people': self.knowledge.money_per_person,
42        }
43        response = self.client.post(self.url, data=data)
44        self.assertEqual(response.status_code, 302)
45        self.assertEqual(response._headers['location'][1], '/encuesta/1/')
46        response2 = self.client.get(response._headers['location'][1])
47        self.assertEqual(response2.status_code, 200)
48        self.assertContains(response2, "El sistema le recomienda visitar el destino turístico Center Name")
49
50    def unlogged_user_cant_view_test(self):
51        response = self.client.get(self.url)
52        self.assertEqual(response.status_code, 302)
```

**Figura 24-2.** Código de la Función realizar encuesta.

Realizado por: HURTADO, J; ZABALA, B. 2018

La Figura 24-2 muestra el código que se realizará en cada ejecución de la prueba. El método setUp configura la clase de la prueba, es el primer método que se ejecuta. El segundo método prueba que los usuarios autenticados puedan visualizar el formulario de la encuesta verificando que el estado de la petición sea correcto (código 200). El tercer método ejecuta una solicitud POST a la url, y verifica que obtiene una redirección (código 302) y posteriormente que se puede acceder a la url del resultado generado. El último método verifica que se obtiene una redirección a la página de login para los usuarios que no están autenticados.

### 2.5.1.8. Función listar servicios

```
1 class ServicelistTestCase(TestCase):
2     def setUp(self):
3         self.user = create_user()
4         self.profile = create_profile(self.user)
5         self.center = create_center(self.profile)
6         self.url = reverse('Service:service_list')
7         self.services = (
8             Service.objects.create(name='Servicio 1'),
9             Service.objects.create(name='Servicio 2'),
10        )
11
12    def test_anonymous_user_can_see_services_list(self):
13        response = self.client.get(self.url)
14        self.assertEqual(response.status_code, 200)
15        for service in self.services:
16            self.assertContains(response, service.name)
17
18    def test_logued_user_can_see_services_list(self):
19        self.client.login(username='juliohurtado', password='examplePass')
20        self.test_anonymous_user_can_see_services_list()
```

**Figura 25-2.** Código de la Función listar servicios.

Realizado por: HURTADO, J; ZABALA, B. 2018

La Figura 24-2 muestra el código que se realizará en cada ejecución de la prueba. El método setUp configura la clase de la prueba, es el primer método que se ejecuta. El segundo y tercer método prueba, respectivamente, que los usuarios sin autenticar y los autenticados puedan visualizar la lista de servicios creados, en ambos casos se obtiene un código 200.

### 2.5.1.9. Función listar destinos turísticos de un servicio

```
1 class CenterListWithServiceTestCase(TestCase):
2     def setUp(self):
3         self.user = create_user()
4         self.profile = create_profile(self.user)
5         self.center = create_center(self.profile)
6         self.service = Service.objects.create(name='Piscinas')
7         self.url = reverse('Service:center_service_list', kwargs={'slug': self.service.slug})
8
9
10    def test_anonymous_user_can_see_centers_list_with_a_service(self):
11        ServiceCenter.objects.create(center=self.center, service=self.service, cost=25,
12                                     observation='Generic Observation')
13        response = self.client.get(self.url)
14        self.assertContains(response, self.center.name)
15
16    def test_user_can_see_centers_list_with_a_service(self):
17        self.client.login(username='juliohurtado', password='examplePass')
18        self.test_anonymous_user_can_see_centers_list_with_a_service()
```

**Figura 26-2.** Código de la Función ver destinos turísticos de un servicio.

Realizado por: HURTADO, J; ZABALA, B. 2018

La Figura 24-2 muestra el código que se realizará en cada ejecución de la prueba. El método setUp configura la clase de la prueba, es el primer método que se ejecuta. El segundo y tercer método prueba, respectivamente, que los usuarios sin autenticar y los autenticados puedan visualizar la lista de centros de un servicio, en ambos casos se obtiene un código 200.

### 2.5.1.10. Función crear servicios

```
1 class CreateServiceTestCase(TestCase):
2     def setUp(self):
3         self.user = create_user()
4         self.profile = create_profile(self.user)
5         self.center = create_center(self.profile)
6         self.service = Service.objects.create(name='Piscinas')
7         self.url = reverse('Service:service_create')
8         self.permissions = Permission.objects.get(name='Puede Crear Centros Turísticos')
9
10    def test_user_without_permission_cant_create_user(self):
11        self.client.login(username='juliohurtado', password='examplePass')
12        response = self.client.get(self.url)
13        self.assertEqual(response.status_code, 302) # Redirige al login
14
15    def test_user_with_permission_can_create_user_get(self):
16        self.client.login(username='juliohurtado', password='examplePass')
17        self.profile.user.user_permissions.add(self.permissions)
18        response = self.client.get(self.url)
19        self.assertEqual(response.status_code, 200)
20        self.assertContains(response, 'Crear servicios')
21
22    def test_user_with_permission_can_create_user_post(self):
23        self.client.login(username='juliohurtado', password='examplePass')
24        self.profile.user.user_permissions.add(self.permissions)
25        data = {'service': self.service.id, 'cost': 10, 'observation': 'Generic Observation'}
26        response = self.client.post(self.url, data=data)
27        new_response = self.client.get(response._headers['location'][1]) # Test redirect
28        self.assertContains(new_response, self.center.name) # Test correct redirect
```

**Figura 27-2.** Código de la Función crear servicios.

Realizado por: HURTADO, J; ZABALA, B. 2018

La Figura 24-2 muestra el código que se realizará en cada ejecución de la prueba. El método setUp configura la clase de la prueba, es el primer método que se ejecuta. El segundo método verifica que se produzca la redirección al login cuando un usuario sin permisos accede a la url. El tercer método prueba una petición GET y verifica que los usuarios con permisos puedan visualizar el formulario para crear un servicio. El último método realiza una petición POST enviando el formulario con los datos necesarios y verifica que se redirija a la página de detalle del destino turístico (accediendo a la url desde los headers de la respuesta)

### 2.5.1.11. Ejecución de las pruebas

Utilizando la herramienta de pruebas de django, se pueden ejecutar las pruebas automatizadas de varias formas

- Ejecutar todas las pruebas de la aplicación:

```
+ (sertum) C:\Users\Juliens\Proyectos\sertum>python manage.py test
x Creating test database for alias 'default'...
.....
-----
Ran 27 tests in 1.671s

OK
Destroying test database for alias 'default'...

(sertum) C:\Users\Juliens\Proyectos\sertum>
```

**Figura 28-2.** Código de la Función crear servicios.

Realizado por: HURTADO, J; ZABALA, B. 2018

- Ejecutar las pruebas correspondientes a un módulo:

```
+ (sertum) C:\Users\Juliens\Proyectos\sertum>python manage.py test centro
x Creating test database for alias 'default'...
.....
-----
Ran 20 tests in 1.101s

OK
Destroying test database for alias 'default'...
```

**Figura 29-2.** Código de la Función crear servicios.

Realizado por: HURTADO, J; ZABALA, B. 2018

- Ejecutar las pruebas correspondientes a una función:

```
+ (sertum) C:\Users\Juliens\Proyectos\sertum>python manage.py test servicio.tests.ServicesListTestCase
x Creating test database for alias 'default'...
..
-----
Ran 2 tests in 0.200s

OK
Destroying test database for alias 'default'...
```

**Figura 29-2.** Código de la Función crear servicios.

Realizado por: HURTADO, J; ZABALA, B. 2018

Después de realizadas las pruebas automatizadas del sistema, se concluye que las mismas favorecen a la localización de errores causados por nuevos requerimientos o cambios realizados en los ya existentes; ya que permiten verificar de forma automática las funciones contempladas en el plan de pruebas, ejecutando los archivos en dónde se encuentran programadas las pruebas.

Por tal motivo, en caso de que se realicen cambios en la aplicación, se recomienda programar pruebas automatizadas que engloben los nuevos requerimientos, para identificar los errores potenciales tanto en las funciones nuevas como en las ya existentes.

De igual manera, se recomienda que, para mejorar el rendimiento del sistema, se desarrollen aplicaciones nativas para dispositivos móviles para Android y iOS.

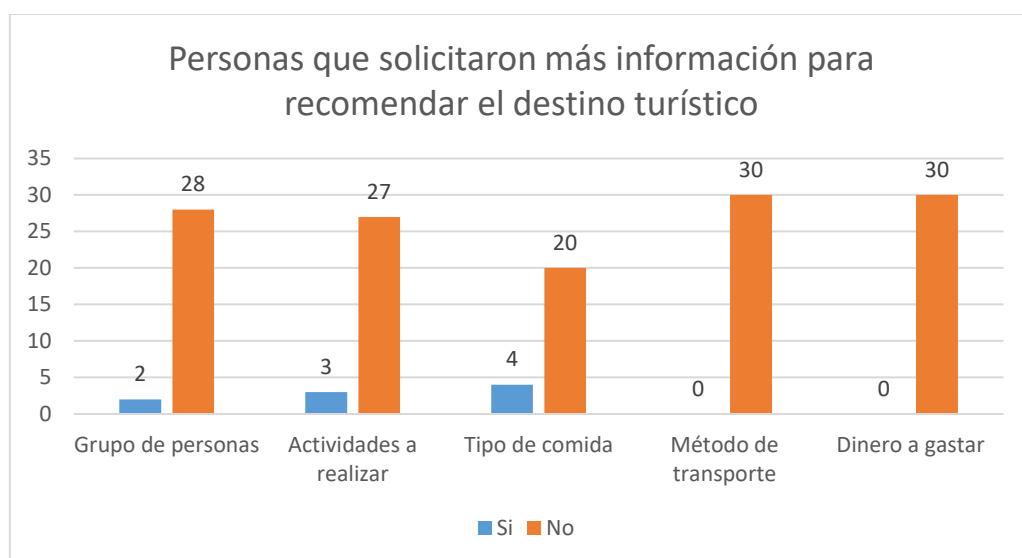
### CAPÍTULO III

#### 3. MARCO DE RESULTADOS

Para verificar la influencia del sistema en la exactitud de las inferencias, se realizaron treinta pruebas de recomendación de destinos turísticos sin utilizar el sistema, y posteriormente la misma cantidad de pruebas utilizando el sistema; en ambos casos, las recomendaciones fueron analizadas por la experta en turismo del GAD del Cantón Morona.

##### Resultados de exactitud utilizando el método de recomendación de boca en boca

Actualmente, la recomendación de destinos turísticos la realiza la ciudadanía teniendo en cuenta únicamente sus propias preferencias o experiencias; sin considerar en ningún momento los gustos o deseos que en ese momento tiene el turista.

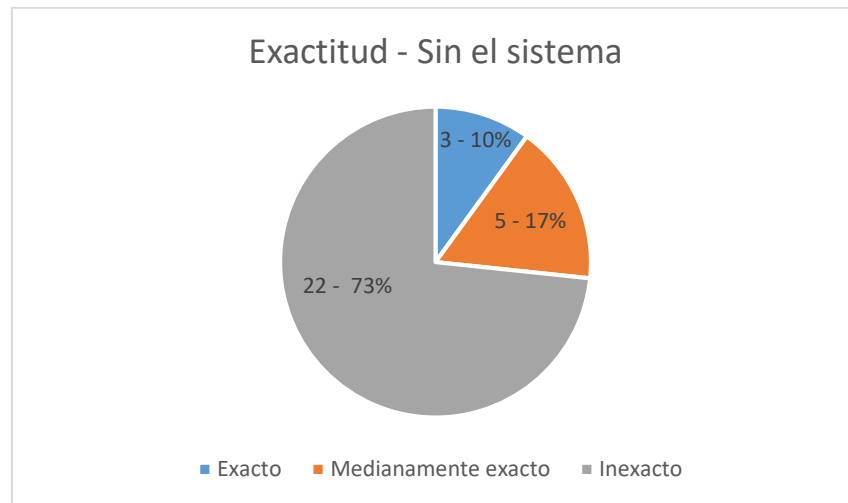


**Gráfico 1-3.** Personas que solicitaron información para recomendar el destino turístico  
Realizado por: HURTADO, J; ZABALA, B. 2018

Para realizar estas pruebas, se solicitó a treinta personas distintas en las calles céntricas de la ciudad de Macas que recomienden un lugar turístico; en la mayoría de los casos, las personas no solicitaron información detallada respecto a lo que se buscaba del destino turístico, sino que se limitaron a realizar una recomendación subjetiva de acuerdo con sus propias experiencias y a la poca información solicitada. Como muestra el *Gráfico 1-3*, solamente **2** personas preguntaron acerca del tipo de personas que visitarían el destino

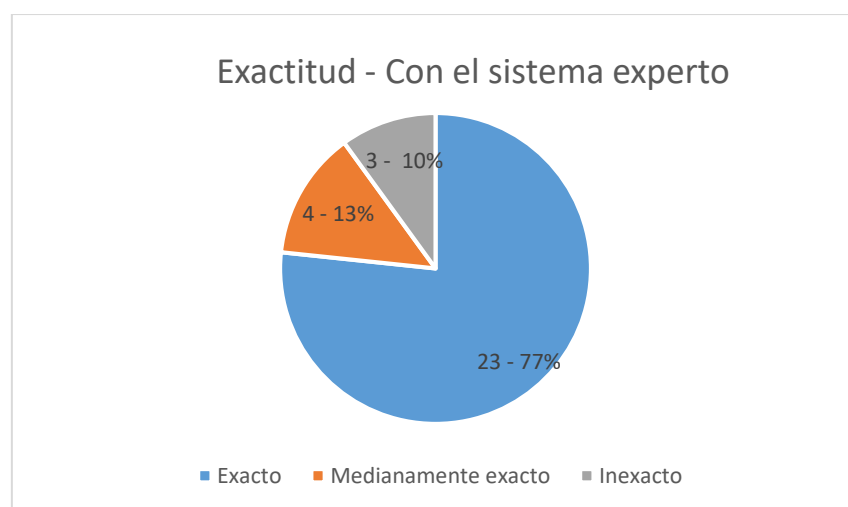


turístico, **3** sobre las actividades que deseaban realizar, **4** sobre el tipo de comida y **ninguna** acerca del método de transporte y el dinero que está dispuesto a gastar.



**Gráfico 2-3.** Aciertos en las recomendaciones por parte de los ciudadanos.  
Realizado por: HURTADO, J; ZABALA, B. 2018

Posteriormente, el experto humano en turismo analiza las recomendaciones teniendo en cuenta las preferencias del turista. En el *Gráfico 2-3* se puede observar que solamente en **3** (10%) casos la recomendación es la más adecuada, en **5** (17%) casos la recomendación no es la más adecuada, sin embargo, es aceptable. Y en **22** (73%) casos no se recomendó la opción más adecuada. Lo que significa que únicamente en el **27%** de los casos se recomendó la opción correcta al turista.



**Gráfico 3-3.** Aciertos en las recomendaciones utilizando el sistema.  
Realizado por: HURTADO, J; ZABALA, B. 2018

El nivel de exactitud del sistema es realizado de igual forma por el experto humano (*Anexo 1*), los resultados se detallan en el *Gráfico 3-3* obteniendo como resultado que en el **77%** de los casos el sistema brindó la recomendación exacta, que coincide con lo que hubiese recomendado el experto humano, en un **13%** la recomendación es aceptable y cumple con las expectativas del turista, sin embargo, el experto habría recomendado otro lugar más adecuado. Y apenas en el **10%** de los casos el sistema recomendó un lugar inadecuado teniendo en cuenta todas las preferencias del turista. Por lo que, en el **90%** de los casos, el sistema brinda una recomendación adecuada; una mejora del **63%** de precisión.

## CONCLUSIONES

1. Tras realizar las pruebas del sistema, se concluye que el desarrollo del sistema experto influyó positivamente en la exactitud de la recomendación de destinos turísticos en el cantón Morona en un 63%.
2. En la actualidad la recomendación de destinos turísticos se realiza utilizando la experiencia o preferencias de la persona que realiza dicha recomendación, es decir, los habitantes del cantón a quienes el turista solicita la información acerca de los destinos turísticos; al no tener en cuenta las preferencias del turista, la exactitud de las recomendaciones es de apenas un 27%; lo que se puede observar en el gráfico 2-3.
3. Una vez realizada la investigación acerca de sistemas expertos se concluye que, un sistema experto tiene la capacidad de simular el comportamiento humano en un área de trabajo determinada, que se lo realiza, utilizando el conocimiento del experto en combinación de tecnologías de desarrollo de sistemas inteligentes.
4. Luego de desarrollar el sistema experto se concluyó que, es posible utilizar un sistema inteligente basado en reglas y hechos que simule el conocimiento de un experto humano para realizar la recomendación de destinos turísticos.
5. Después de realizadas las pruebas de inferencia, y tras la validación de los resultados por parte de la experta humana, se concluye que al utilizar el sistema se obtiene una precisión del 90% en las recomendaciones turísticas, según se observa en el gráfico 3-3 y en el Anexo A.

## RECOMENDACIONES

1. Con la finalidad de que los resultados de la investigación se puedan reflejar en la realidad de la actividad turística, se recomienda a la oficina de Turismo y al departamento de TICs del GAD de Morona se proceda con la etapa de implantación del sistema.
2. Se recomienda al GAD de Morona aumentar la difusión de información acerca de los destinos turísticos, esto con la finalidad de que los turistas se puedan informar de los servicios que ofertan los distintos destinos turísticos. Además, socializar a los ciudadanos del cantón los aspectos que se deben tener en cuenta para recomendar a un turista un destino turístico, restaurante, u otro centro de diversión y esparcimiento.
3. Se recomienda, en un futuro, luego de la implantación del Sistema Experto, realizar un estudio técnico que tenga como objetivo la reestructuración del motor de inferencia para que haga uso de técnicas más avanzadas tales como las redes neuronales en caso de ser necesario.
4. Para que la aplicación recomiende destinos turísticos con mayor exactitud, se recomienda que los datos ingresados en la base de conocimiento sean auditados de forma periódica por un delegado del departamento de turismo del GAD municipal de Morona y que se socialice la aplicación de tal forma que se consiga registrar la mayor cantidad de destinos turísticos para aumentar la información disponible en la base de conocimientos.
5. Para mejorar la experiencia de usuario en la utilización del sistema, se recomienda realizar en un futuro aplicaciones nativas para los distintos sistemas operativos móviles; de la misma forma, implementar un módulo que permita al usuario valorar la recomendación dada por el sistema luego de visitar el destino. Adicionalmente, agregar al módulo de administración funcionalidades que permitan presentar reportes sobre el uso de la aplicación y sobre las inferencias realizadas.

## BIBLIOGRAFÍA

- **ACERENZA, Miguel.** *Conceptualización, Origen y Evolución del Turismo*. México-México: Trillas, 2006, p 27.
- **ALVA, Elena.** Metodología de Medición y Evaluación de la Usabilidad en Sitios Web Educativos (tesis). (Doctoral). Universidad de Oviedo, Departamento de Informática, Oviedo, España. 2005, p.32 [Consulta: 14 marzo 2018]. Disponible en: <http://di002.edv.uniovi.es/~cueva/investigacion/tesis/Elena.pdf>
- **ÁLVAREZ, Luis.** *Fundamentos de inteligencia artificial*. Murcia-España: Universidad de Murcia. Secretariado de Publicaciones, 1994, p 19.
- **BASOGAIN, Xavier.** *Redes neuronales artificiales y sus aplicaciones*. Bilbao-España, Escuela Superior de Ingeniería de Bilbao, 2008, pp. 3-4.
- **BENITEZ, Raúl; et al.** *INTELIGENCIA ARTIFICIAL AVANZADA*. Barcelona-España: Editorial UOC, 2013, p 6.
- **BLÉ, Carlos.** *Diseño Ágil con TDD*. iExpertos.com, 2010, pp.48-50
- **BUCHANAN, Bruce; y SHORTLIFFE, Edward.** *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. MA-Estados Unidos: Addison, Wesley, 1984, p 3.
- **CASTILLO, Enrique; et al.** *Sistemas Expertos y Modelos de redes probabilísticas*. Santander-España: Academia de ingeniería, Universidad de Cantabria, 1996, p.11, p.12.
- **CONDORI, José.** “Python – Django. Framework de desarrollo web para perfeccionistas. Basado en el Modelo MTV” [en línea]. *Revista de Información, Tecnología y Sociedad*, n°7 (2012), (Bolivia) pp. 36-37. Disponible en: [http://www.revistasbolivianas.org.bo/scielo.php?script=sci\\_arttext&pid=S1997-40442012000200016&lng=es&nrm=iso](http://www.revistasbolivianas.org.bo/scielo.php?script=sci_arttext&pid=S1997-40442012000200016&lng=es&nrm=iso)
- **DABARÓ, Sebastián; et al.** “Sistemas Expertos: Fundamentos, Metodologías y Aplicaciones” *Revista Ciencia y Tecnología de la Universidad de Palermo* [en línea]. 2013,

(Argentina) vol.13, pp.349-364. [Consulta: 10 marzo 2018]. ISSN 1850-0870. Disponible en:  
[http://www.palermo.edu/ingenieria/pdf2014/13/CyT\\_13\\_24.pdf](http://www.palermo.edu/ingenieria/pdf2014/13/CyT_13_24.pdf)

- **DJANGO SOFTWARE FOUNDATION.** *Django Documentation* [en línea]. Writing your first Django app, part 5. Disponible en:  
<https://docs.djangoproject.com/en/1.11/intro/tutorial05/>
- **DRAKE, Fred.** *El tutorial de Python* [en línea]. Argentina, Python Software Foundation: 2017, pp.1-3-8-9-11-12-13-66. [Consulta: 10 enero 2017] Disponible en:  
<http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf>
- **FLÓREZ, Raquel y FERNÁNDEZ, José.** *Las redes neuronales artificiales fundamentos teóricos y aplicaciones prácticas.* La Coruña-España: Netbiblo S.L., 2008, p.16
- **FUENTES, Aníbal.** *Teoría y Desarrollo del Turismo en el ECUADOR.* Quito-Ecuador: Efecto Gráfico, 2013, pp.121-122-124
- **GÁMEZ, José y PUERTA, José.** *Sistemas Expertos probabilísticas.* Castilla-España: Ediciones de la universidad de Castilla la Mancha, 1998, p.6, p.7.
- **GARCÍA, Alberto.** *INTELIGENCIA ARTIFICIAL: Fundamentos, práctica y aplicaciones.* Madrid-España: Grupo RC, 2012, pp.2-6
- **GARCÍA, Ana.** *Evaluación de métricas de calidad del software sobre un programa Java (Tesis) (maestría).* Universidad Complutense de Madrid, Facultad de Informática, Departamento de Sistemas Informáticos y Computación. Madrid-España. 2010. pp. 7-8 [Consulta: 15 julio 2007]. Disponible en: <http://eprints.ucm.es/11487/>
- **GRANOLLERS, T., & Lorés, J.** “Esfuerzo de Usabilidad: un nuevo concepto para medir la usabilidad de un sistema interactivo basada en el Diseño Centrado en el Usuario.” *Asociación Interacción Persona-Ordenador* [en línea], 2004, (España), pp.123-124. [Consultado: 17 marzo 2018] Disponible en: <https://aipo.es/articulos/3/18.pdf>
- **GUERRERO, Perla; & RAMOS, José.** *Introducción al Turismo* [en línea]. México-México: Patria, 2014, p.33. [Consulta: 30 noviembre 2016]. Disponible en:  
<http://www.editorialpatria.com.mx/pdffiles/9786074384130.pdf>

- **INFANTE, Sergio.** *Maestros del Web* [en línea]. 2012. Curso Django: Entendiendo como Funciona Django. [Consulta: 14 febrero 2017] Disponible en: <http://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django>
- **MATICH, Jorge.** *Redes neuronales: conceptos básicos y aplicaciones*. Rosario-Argentina: Universidad Tecnológica Nacional, 2001, p.9.
- **MÍNGUEZ, R.** *Microsoft Developer Network* [en línea]. 2005. MockObjects y TDD en .Net Framework [Consulta: 10 marzo 2018]. Disponible en: <https://msdn.microsoft.com/es-es/library/bb932285.aspx#EOAA>
- **MORALES-LUNA, Guillermo.** *Introducción a la lógica difusa*. México-México: Centro de Investigación y Estudios Avanzados del IPN, 2002, p.1
- **PÉREZ, Iván.** *Lógica difusa para principiantes*. Caracas-Venezuela: Universidad Católica Andrés Bello, 2007, p.12.
- **PERUREN, Lilliam y MORAGUE, Mercedes.** “Usability of Web sites, methods and evaluation techniques” *Rev. cuba. inf. cienc. salud* [en línea]. 2013, (Cuba) vol.24, n.2, p.188. [Consulta: 18 marzo 2018]. ISSN 2307-2113. Disponible en: [http://scielo.sld.cu/scielo.php?pid=S2307-21132013000200007&script=sci\\_arttext&tlng=en](http://scielo.sld.cu/scielo.php?pid=S2307-21132013000200007&script=sci_arttext&tlng=en)
- **PINO, Raúl; et al.** *Introducción a la inteligencia artificial: Sistemas Expertos, Redes neuronales artificiales y computación evolutiva*. Madrid-España: Servicio de publicaciones de la Universidad de Oviedo, 2001, p.1.
- **RAMOS, M<sup>a</sup> Jesús; et al.** *Sistemas gestores de bases de datos*. Madrid-España: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, 2002, pp.8-9.
- **SILBERSCHATSZ, Abraham; et al.** *Fundamentos de Bases de Datos. 4<sup>ta</sup> ed.* Madrid-España: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, 2002, p.1.
- **SUMMERFIELD, Mark.** *Python3*. Madrid-España: GRUPO ANAYA, 2010, p.1.
- **WORSLEY, John; & DRAKE, Josua.** *Practical PostgreSQL*. Sebastopol, CA-Estados Unidos: O'Reilly & Associates Inc, 2002, p.3.

**ANEXO A. Certificado de experto humano sobre resultados de inferencia**



**ANEXO B. Manual de usuario.**