

UNIVERSIDAD PERUANA UNIÓN
FACULTAD DE INGENIERÍA Y ARQUITECTURA
Escuela Profesional de Ingeniería de Sistemas



Una Institución Adventista

Diseño e implementación de un circuito turístico inteligente en la
Región Puno mediante la metaheurística Búsqueda Tabú.

Por:

Oscar David Mendoza Apaza

Asesor:

Mg. Abel Angel Sullon Macalupu

Co-Asesor:

Dra. Sc. Nélide Gladys Maquera Sosa

Juliaca, marzo de 2017

Cómo citar:

Estilo Apa

Mendoza A. (2017). *Diseño e implementación de un circuito turístico inteligente en la Región Puno mediante la metaheurística Búsqueda Tabú* (Tesis de grado). Universidad Peruana Unión, Juliaca, Puno, Perú.

Estilo Iso

MENDOZA APAZA Oscar David. *Diseño e implementación de un circuito turístico inteligente en la Región Puno mediante la metaheurística Búsqueda Tabú*, Tesis de grado inédita, Universidad Peruana Unión, Juliaca, 2017.

Estilo IEEE

O. Mendoza, *Diseño e implementación de un circuito turístico inteligente en la Región Puno mediante la metaheurística Búsqueda Tabú*. Juliaca: Universidad Peruana Unión, 2017.

Ficha catalográfica elaborada por el Centro de Recursos para el Aprendizaje y la Investigación (CRAI) de la UPeU

TIS	Mendoza Apaza, Oscar David
2	Diseño e implementación de un circuito turístico inteligente en la Región Puno mediante la metaheurística Búsqueda Tabú / Autor: Oscar David Mendoza Apaza;
M42	Asesor: Mg. Abel Angel Sullon Macalupu - Juliaca, 2017.
2017	76 páginas: anexos, figuras, tablas.
	Tesis (Licenciatura) -- Universidad Peruana Unión. Facultad de Ingeniería y Arquitectura. EP. Ingeniería de Sistemas, 2017.
	Incluye referencias y resumen.
	Campo del conocimiento: Ingeniería de Sistemas
	1. Problema del agente viajero. 2. Optimización. 3. Búsqueda Tabú. 4. Heurística de Construcción. 4. Heurística de Mejoría.

DECLARACIÓN JURADA
DE AUTORÍA DEL INFORME DE TESIS

Abel Angel Sullon Macalupu de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas de la Universidad Peruana Unión.

HACE CONSTAR:

Que el presente informe de investigación titulado "DISEÑO E IMPLEMENTACIÓN DE UN CIRCUITO TURÍSTICO INTELIGENTE EN LA REGIÓN PUNO MEDIANTE LA METAHEURISTICA BÚSQUEDA TABÚ", constituye la memoria que representa el Bachiller Oscar David Mendoza Apaza para aspirar al título de Profesional de Ingeniero de Sistemas, ha sido realizada en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente constancia en Juliaca, a los treinta y un días de octubre del dos mil diecisiete




Mg, Abel Angel Sullon Macalupu

Diseño e implementación de un circuito turístico
inteligente en la Región Puno mediante la metaheurística
Búsqueda Tabú

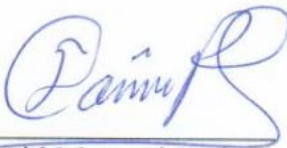
TESIS

Presentada para optar el título profesional de Ingeniero de Sistemas

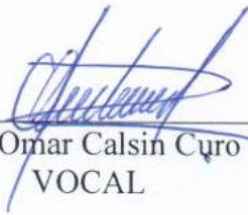
JURADO CALIFICADOR



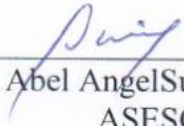
Ing. Fredy Abel Huanca Torres
PRESIDENTE



Ing. David Mamani Pari
SECRETARIO



Ing. Omar Calsin Curo
VOCAL



Mg. Abel Angel Sullon Macalupu
ASESOR

Juliaca, 31 de marzo de 2017

DEDICATORIA

A mi madre Cristina, tenerla es un verdadero privilegio; a la memoria de mi padre Vicente, por ser mis primeros maestros e inculcarme valores para ser una mejor persona. A la Dra. Gladys Maquera por motivarme constantemente en la investigación científica y superación académica. A toda mi familia por respaldarme en la culminación de mis estudios.

AGRADECIMIENTOS

A Dios, el supremo investigador, por brindarme la vida, salud y la fuerza e inteligencia necesaria para la elaboración de esta investigación; a la Universidad Peruana Unión por incentivar la investigación y desarrollar nuestros conocimientos; al Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica (CONCYTEC), por promover e impulsar la investigación científica, patrocinando esta investigación. Al Mg. Abel Angel Sullon Macalupu y a la Dra. Sc. Nélica Gladys Maquera Sosa por el apoyo, colaboración y asesoramiento; a mis padres por su apoyo y sacrificio incondicional concedido constantemente en todo el tiempo de estudio, así como también a todas aquellas personas quienes contribuyeron con sus aportes.

ÍNDICE GENERAL

DEDICATORIA.....	v
AGRADECIMIENTOS.....	vi
ÍNDICE GENERAL.....	vii
ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE ANEXOS.....	xii
ABREVIATURAS Y ACRÓNIMOS.....	xiii
RESUMEN.....	xiv
ABSTRACT.....	xv
CAPÍTULO I Introducción.....	16
1.1.Descripción del problema.....	16
1.2.Objetivos.....	18
1.2.1.Objetivo general.....	18
1.2.2.Objetivos específicos.....	18
1.3.Justificación de la investigación.....	18
CAPÍTULO II Marco Teórico.....	22
2.1.Antecedentes de la investigación.....	22
2.1.1.El problema del agente viajero: un algoritmo determinístico usando Búsqueda Tabú.....	22
2.1.2.Búsqueda Tabú para un problema de diseño de red multiproducto con capacidad finita en las aristas.....	22
2.1.3.Interactive design of personalized tourism routes.....	23
2.1.4.Intelligent tourism recommender systems: A survey.....	24
2.2.Bases teóricas.....	25
2.2.1.Inteligencia Artificial.....	25
2.2.2.Algoritmos de búsqueda.....	26
2.2.2.1.Búsqueda no informada.....	26
2.2.2.1.1.Búsqueda en anchura.....	27
2.2.2.1.2.Búsqueda en profundidad.....	27
2.2.2.1.3.Búsqueda bidireccional.....	28
2.2.2.2.Búsqueda informada.....	29
2.2.2.2.1.Búsqueda voraz primero el mejor.....	29
2.2.2.2.2.Búsqueda A*: Minimizar el costo total de la solución.....	31

2.2.3.Métodos exactos y heurísticos	34
2.2.3.1.Métodos exactos	34
2.2.3.2.Método heurístico	34
2.2.4.Heurística.....	35
2.2.4.1.Heurística de construcción	35
2.2.4.2.Heurística del vecino más cercano	35
2.2.4.3.Heurística de mejoría.....	36
2.2.4.4.Heurística de 2 intercambio (2-opt).....	36
2.2.5.Metaheurística	37
2.2.6.Metaheurística Búsqueda Tabú	37
2.2.7.Investigación de Operaciones	38
2.2.8.Python.....	41
2.2.9.Programación Extrema (XP)	41
CAPÍTULO III Materiales y métodos	42
3.1.Materiales	42
3.2.Tipo de investigación	43
3.3.Diseño de la solución propuesta	43
3.3.1.Definición del problema basado en el Problema del Agente Viajero.....	43
3.3.2.Diseño e implementación de la heurística de construcción.....	43
3.3.3.Diseño e implementación de la heurística de mejoría.	46
3.3.4.Diseño e implementación de la metaheurística Búsqueda Tabú.	47
3.3.4.1.Estructura de datos.	48
3.3.4.2.Entorno de intercambio.	48
3.3.4.3.Memoria de corto plazo.....	49
3.3.4.4.Criterio de aspiración.....	50
3.3.4.5.Memoria de largo plazo.....	50
3.3.4.6.Diversificación.....	51
CAPÍTULO IV Validación del algoritmo en el proyecto UPeU-CONCYTEC.....	52
4.1.Acerca del proyecto UPeU-CONCYTEC.	52
4.2.Desarrollo de la aplicación web.....	52
4.2.1.Casos de uso.	52
4.2.1.1.Caso de uso del registro de emprendimientos	52
4.2.1.1.1.Actores.....	53
4.2.1.1.2.Flujo Básico	53

4.2.1.1.3.Prototipos.....	54
4.2.1.2.Caso de uso para generar la ruta inteligente	55
4.2.1.2.1.Actores.....	56
4.2.1.2.2.Flujo Básico.....	56
4.2.1.2.3.Prototipos.....	57
4.2.2.Programación.....	58
4.3.Validación de la metaheurística con datos del Distrito de Juli.....	60
CAPÍTULO V Resultados y discusión	62
5.1.Resultados.....	62
5.1.1.Validación de la metaheurística con instancias artificiales	62
5.2.Discusión	65
CAPÍTULO VI Conclusiones y recomendaciones	67
6.1.Conclusiones.....	67
6.2.Recomendaciones	67
7.LISTA DE REFERENCIAS.....	69
ANEXOS	72

ÍNDICE DE TABLAS

Tabla 1 Valores de <i>hDLR</i> distancias en línea recta a Bucarest.	30
Tabla 2 Aplicaciones de la investigación de operaciones en organizaciones reales y su impacto.	39
Tabla 3 Matriz de distancias en km. de 6 recursos turísticos.	48
Tabla 4 Entorno de intercambio para la primera iteración.	49
Tabla 5 Matriz tabú en una iteración dada.	50
Tabla 6 Matriz de frecuencia al final de una ejecución.	51
Tabla 7 Resultados de la validación con instancias artificiales.	63
Tabla 8 Resultados de la validación con instancias artificiales.	64

ÍNDICE DE FIGURAS

Figura 1. Número anual de artículos publicados con relación al turismo y sistemas de información en Beijing.	19
Figura 2. Representación de un agente interactuando con el medioambiente.	26
Figura 3. Búsqueda en anchura sobre un árbol binario sencillo.	27
Figura 4. Búsqueda en profundidad sobre un árbol binario.	28
Figura 5. Un esquema de una búsqueda bidireccional que está a punto de tener éxito.	29
Figura 6. Etapas de una búsqueda primero el mejor para Bucarest utilizando la heurística distancia en línea recta.	31
Figura 7. Etapas en una búsqueda A* para Bucarest etiquetando los nodos con $f=g+h$	33
Figura 8. Diagrama de flujo del método de construcción del vecino más cercano.	45
Figura 9. Solución inicial.	46
Figura 10. Solución mejorada.	46
Figura 11. Codificación del algoritmo heurístico de mejoría 2-óptimo.	47
Figura 12. Permutación inicial.	49
Figura 13. Intercambio de los recursos 1 y 4.	49
Figura 14. Criterio de aspiración.	50
Figura 15. Diagrama de caso de uso del registro de emprendimientos.	53
Figura 16. Prototipo del registro de emprendimientos.	54
Figura 17. Prototipo del registro de actividades del emprendimiento.	55
Figura 18. Diagrama de caso de uso de la ruta inteligente.	56
Figura 19. Prototipo para seleccionar los recursos que se desea visitar.	57
Figura 20. Prototipo para registrar las preferencias del turista.	57
Figura 21. Prototipo de la ruta propuesta por el algoritmo.	58
Figura 22. Codificación de la metaheurística Búsqueda Tabú.	59
Figura 23. Aplicación web puesta en producción.	59
Figura 24. Validación del algoritmo con instancias reales.	60
Figura 25. Circuito personalizado para un día de viaje y un presupuesto de 100 soles.	61
Figura 26. Eficiencia de los tres métodos.	64
Figura 27. Búsqueda global procesando la instancia dantzig42 mediante TS.	65

ÍNDICE DE ANEXOS

Anexo A Información recopilada del distrito de Juli.	72
Anexo B Formulario 1 utilizados para el levantamiento de información.....	74
Anexo C Formulario 2 utilizados para el levantamiento de información.....	74
Anexo D Fotos del levantamiento de la información	75
Anexo E Validación de la aplicación web por parte de los actores del TRC de Puno	75
Anexo F Mapeo de tres rutas hacia los recursos turísticos.....	76
Anexo G Instancia dantzig42	77
Anexo H Servidor local del API que contiene los recursos para la ejecución de los algoritmos	78

ABREVIATURAS Y ACRÓNIMOS

- UPeU: Abreviatura de Universidad Peruana Unión.
- TS: Abreviatura en ingles de Búsqueda Tabú.
- 2-OPT: Método dos intercambios para la heurística de construcción.
- TSP: Abreviatura en ingles de Problema del Agente Viajero.
- IA: Abreviatura de Inteligencia Artificial.
- IO: Abreviatura de Investigación de Operaciones.
- XP: Abreviatura en ingles de la metodología de desarrollo de software ágil Programación Extrema.
- NP-Duro: Definición de un Problema combinatorio con una complejidad exponencial.
- PROMPERÚ: Abreviatura de Comisión de Promoción del Perú para la Exportación y el Turismo.
- MINCETUR: Abreviatura de Ministerio de Comercio Exterior y Turismo.
- TRC: Abreviatura de Turismo Rural Comunitario.

RESUMEN

En esta investigación se aborda una variante del problema de optimización combinatoria del área de Investigación de Operaciones y que también es estudiada por la Inteligencia Artificial; famoso por ser sencillo de enunciar, pero complejo de resolver: “el problema del agente viajero”, una de sus aplicaciones en la vida real se refleja en la planificación de viajes turísticos en la región Puno, cuando un turista desea diseñar un circuito turístico personalizado teniendo en cuenta sus necesidades, gustos y experiencias. Dada la dificultad de dar solución a este problema combinatorio de tipo NP-Duro, en donde se requiere ofrecer una solución inmediata (segundos), surge la necesidad del uso de heurísticas y metaheurísticas (algoritmos aproximados); en consecuencia, la solución planteada a este problema, fue diseñar e implementar la metaheurística Búsqueda Tabú contemplando las variables propias del lugar de ejecución. Para el desarrollo de la aplicación web que integrará el algoritmo se adoptó la metodología ágil XP, y para la implementación de la metaheurística seguimos el procedimiento de método TS, iniciamos con dos valores de la función objetivo obtenidas mediante una heurística de construcción basada en el método del vecino más cercano y una heurística de mejoría mediante el método del intercambio de dos aristas interceptadas (2 - opt). TS logró evitar el bucle en soluciones óptimas locales explorando nuevas regiones en un entorno global. El algoritmo se validó utilizando cinco instancias artificiales definidas en el ámbito científico contrastando con sus mejores resultados obtenidos hasta la fecha; logrando igualar al 60% de las mejores soluciones. También se utilizó información recopilada de los atractivos turísticos ubicados en el distrito de Juli región Puno, y posteriormente en dos sesiones los actores directos del Turismo Rural Comunitario pudieron validar la aplicación web desplegada en un servidor de aplicación.

Palabras clave: Problema del agente viajero, Optimización, Heurística de construcción, Heurística de Mejoría, Búsqueda Tabú.

ABSTRACT

This research deals with a variant of the combinatorial optimization problem of the Operations Research area, which is also studied by Artificial Intelligence; famous for being simple to enunciate, but complex to solve: " Travelling salesman problem", one of its applications in real life is reflected in the planning of tourist trips in the Puno region, when a tourist wants to design a personalized tourist circuit taking into account their needs, tastes and experiences. Given the difficulty of solving this NP-Hard type combinatorial problem, where it is necessary to offer an immediate solution (seconds), the need to use heuristics and metaheuristics (approximate algorithms) arose; consequently, the solution to this problem was to design and implement the metaheuristic Tabu Search (TS) taking into account the variables of the place of execution. For the development of the web application that will integrate the algorithm, the XP agile methodology was adopted, and for the implementation of the metaheuristics we follow the TS method procedure, starting with two values of the objective function obtained by means of a construction heuristic based on the nearest neighbor's method and an improvement heuristic by means of the intercepted two-edged exchange method (2 - opt). TS was able to avoid the loop in local optimal solutions by exploring new regions in a global environment. The algorithm was validated using five artificial instances defined in the scientific field, contrasting with the best results obtained to date; achieving 60% of the best solutions. Information gathered from the tourist attractions located in the district of Juli region Puno was also used, and later in two sessions the direct actors of Community Rural Tourism (TRC) were able to validate the web application deployed on an application server.

Key words: Travel agent problem, Optimization, Heuristics of construction, Heuristics of Improvement, Tabu search.

CAPÍTULO I

Introducción

1.1. Descripción del problema

El Perú es un país reconocido en todo el mundo por su autenticidad, biodiversidad, legado histórico y por ser cuna de una de las civilizaciones más antiguas del planeta, y cuenta con una afluencia importante de turistas (Mohme, 2013, pág. 4). El turismo se ha convertido en uno de los motores económicos del país y viene acompañado por los efectos positivos sobre la totalidad de las condiciones sociales, culturales y ambientales. Puno es la 5ta región más grande del país y la 2da más visitada, donde encontramos culturas que aún se mantienen vivas y atractivos turísticos naturales aún no difundidos (Machado, 2011, pág. 8), cuenta también con un potencial turístico de nivel internacional por estar localizado estratégicamente entre Brasil, Chile y Bolivia, es también importante destacar que es uno de los circuitos turístico más comercializados y con más relevancia en el país; por integrar las regiones de Lima-Arequipa-Cusco-Puno (Moncloa, 2013, pág. 17).

Una publicación de la Comisión de Promoción del Perú para la Exportación y el Turismo (PROMPERÚ) afirma que: “Los millennials, las personas que oscilan entre los 18 y los 34 años de edad, usan la tecnología todo el tiempo y una de sus actividades preferidas es viajar para vivir experiencias únicas. En el Perú, este segmento representa el 35% del total de sus habitantes y es el segundo grupo más importante dentro de la población que viaja al interior de nuestro país por vacaciones, descanso o diversión” (PROMPERU, 2015, pág. 2) , asimismo en las cifras que brinda esta misma entidad indica que el 86% de los turistas extranjeros planifica su viaje vía internet (PROMPERU, 2013, pág. 7). Por lo tanto, existen sistemas de información que brindan un fácil acceso a un gran volumen de datos de los diferentes lugares turísticos de la Región Puno, esta información está dirigida a los turistas locales y extranjeros. Para el turista independiente, el cual no contrata los servicios de una agencia, diseña su propia ruta de viaje y posee recursos limitados como tiempo y dinero; le resulta muy complicado aplicar una serie de cálculos con el fin de generar una ruta turística a medida y que también demandará un tiempo indefinido. Revisando los sitios web oficiales y más promocionados por el Ministerio de Comercio Exterior y Turismo (MINCETUR) como son: www.peru.travel, www.ytuqueplanes.com, www.turismoruralcomunitario.com.pe, se advierte que no cuentan con un componente que integre un modelo matemático para la toma de decisiones, que sea capaz de evaluar todas las diferentes alternativas con una determinada cantidad de variables, con el objetivo de diseñar un viaje a medida según las preferencias y necesidades del turista; afirmación que

es también corroborada por el representante del Turismo Rural Comunitario en la Región Puno (Canales, 2016).

En la industria, las decisiones tomadas incorrectamente pueden traer pérdidas de recursos tanto tangibles como intangibles al tener que evaluar diferentes alternativas para lograr una solución óptima, por tal motivo los criterios a tomar en cuenta se deben basar en características que determinen que la decisión tomada sea la mejor (Yunbulema, 2005, págs. 1-2). En los países desarrollados están ahorrando recursos y/o aumentando ganancias por la optimización de rutas; el saber escoger la ruta es importante a la hora de brindar un servicio, existen problemas en empresas, como visitar sitios turísticos, transporte de personas, de mercadería, entre otros; éste es un problema combinatorio de Investigación de Operaciones (IO) y se le denomina Problema del Agente Viajero (frase en inglés - TSP) (Araujo, 2006, págs. 1-2), también es estudiada por la Inteligencia Artificial (IA) debido a su complejidad computacional. Se puede deducir por lo anterior, que al no aplicar métodos de Investigación de Operaciones y técnicas de IA; en especial la de optimización de rutas turísticas en la Región Puno, disminuyen las posibilidades de ser competitivos en el mercado turístico.

El Problema del Agente Viajero es uno de los problemas más complejos que se conoce de la programación matemática de la actualidad, por su complejidad computacional, y está clasificado dentro de aquellos problemas considerados NP-Duros (que no tienen un tiempo polinomial de respuesta). Un posible enunciado general del problema podría ser dado por (Calviño, 2011, pág. 7) donde define: “Si un viajante parte de una ciudad y las distancias a otras ciudades son conocidas, ¿cuál es la ruta óptima que debe elegir para visitar todas las ciudades y volver a la ciudad de partida?”.

Existen diversos tipos de problemas de rutas (en la literatura se le denomina con distintos nombres), los cuales están diferenciados por las restricciones que se le impongan. En el caso particular de la presente investigación se añadirán las limitaciones de costo y tiempo, así como también restricciones propias del atractivo turístico como son: categorías, accesibilidad, transporte y servicios.

La gran diferencia al TSP de esta investigación es, en el primer caso, la ruta óptima a determinar debe visitar todas las ciudades, mientras que, en el segundo, para lograr la ruta óptima no es necesario que el turista visite todas las ciudades.

Se considera que la alternativa de solución que será propuesta, ayudará a los turistas, así como también a la inserción de nuevos problemas en IO e IA.

1.2. Objetivos

1.2.1 Objetivo general.

Diseñar e implementar un circuito turístico inteligente en la Región Puno mediante la Metaheurística Búsqueda Tabú.

1.2.2 Objetivos específicos.

- a) Diseñar una Heurística de Construcción
- b) Diseñar una Heurística de Mejoría.
- c) Implementar la Metaheurística Búsqueda Tabú.
- d) Validar la Metaheurística Búsqueda Tabú con instancias artificiales y datos reales del Distrito de Juli Región Puno.
- e) Desarrollo de una aplicación web que integre la Metaheurística Búsqueda Tabú.

1.3. Justificación de la investigación

Con la presente investigación se pretende la creación y/o integración de los atractivos turísticos en los circuitos turísticos existentes en la Región Puno, estos atractivos no son promocionados por PROMPERÚ y MINCETUR, muchos de ellos son desconocidos y no son considerados en el itinerario de viaje por las agencias turísticas y los turistas independientes, la mayoría están ubicados en comunidades alejadas con un alto índice de pobreza en donde el trabajo diario es de tipo artesanal (agricultura, ganadería y pesca); es por ello que se busca tener mejores ingresos tentando así, a una alternativa viable como es el turismo (Canales, 2016). El turismo podría ayudar a revalorar sus culturas, que con el pasar de los tiempos se ha ido perdiendo; y, sin embargo, forman parte de los atractivos turísticos con gran demanda en el Perú (Suaña, 2016).

La Ley General del Ambiente, Ley N° 28611, señala que el estado, a través de sus órganos y entidades, tiene la función de diseñar y aplicar las políticas, normas, instrumentos, incentivos y sanciones necesarias para garantizar el ejercicio efectivo, así como el cumplimiento de los derechos, obligaciones y responsabilidades de carácter ambiental. Al percibir afluencia turística un recurso natural podría convertirse en un atractivo turístico, el cual estará regido por las leyes del Perú mediante los diferentes Ministerios y Autoridades Locales y Regionales; de esta manera se podrá obtener beneficios como financiamiento e incorporación en proyectos de desarrollo y/o investigación, es decir: que lograrán formalizarse siendo reconocidos como un emprendimiento real por el Estado Peruano (Congreso de la República del Perú, 2009, pág. 1). Por consiguiente, se aportaría en la conservación responsable del medio ambiente.

Cuando deseamos viajar algunas veces nos dirigimos a una agencia de viajes para que el agente nos asesore sobre las ofertas turísticas. Sin embargo, poco a poco los turistas vienen sustituyendo la agencia de viajes por la consulta y reserva a través de internet; a este tipo de turista se les denomina “Turista 2.0” y/o “Turista Inteligente” (Buhalis, 2003). El turista no sólo desea la información que la mayoría de software/aplicaciones brindan, si no desea que le recomiende la secuencia de la ruta turística de los lugares que ha elegido en el tiempo requerido contemplando el presupuesto y tiempo con el que el turista dispone; además de las diferentes necesidades que ellos tienen.

La investigación interdisciplinaria en las Tecnologías de la Información y el Turismo está creciendo rápidamente (Li, Buhalis, & Zhang, 2013, pág. 2), lo que demuestra la importancia tecnológica en su desarrollo. Esto se refleja en la cantidad de investigaciones y publicaciones enfocadas en estas áreas realizadas en la ciudad de Beijing, y se presentan en la Figura 1.

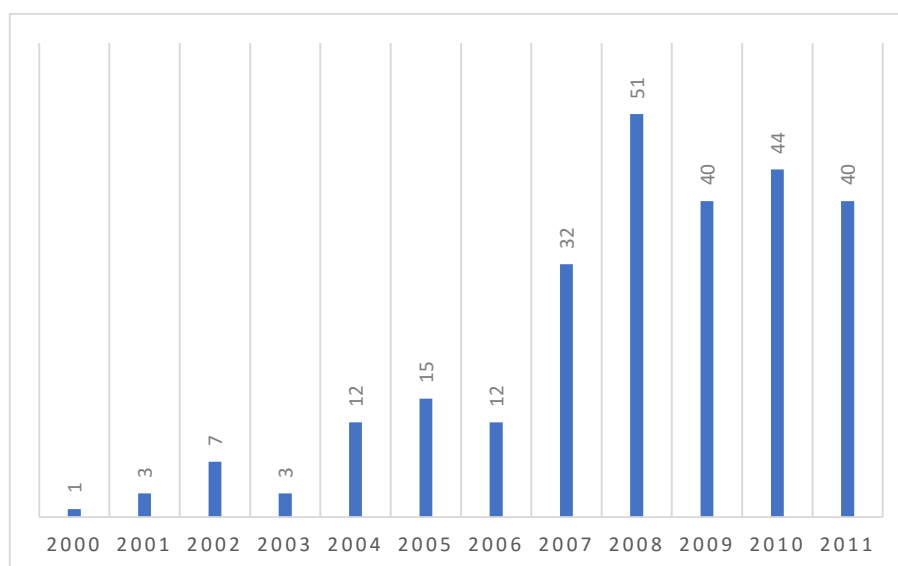


Figura 1. Número anual de artículos publicados con relación al turismo y sistemas de información en Beijing.

Fuente: Interdisciplinary Research on Information Science and Tourism. Information and Communication Technologies in Tourism, 2013.

La mayor parte de aplicaciones web o móvil relacionadas con los viajes de apoyo tales como sistemas de recomendación, sistema de guiado, sistema de servicio de vigilancia de posición, recurso turístico y el sistema de alerta temprana necesitan contar con un sistema de información geográfica. La Inteligencia Artificial es el método más usado en la investigación de las Tecnologías de la Información relacionada con el Turismo, estas investigaciones se pueden dividir en varios aspectos: el razonamiento, los datos del blog, la

minería de datos, los agentes inteligentes, evaluación y toma de decisiones de apoyo (Li, Buhalis, & Zhang, 2013, pág. 2).

Dada la dificultad de dar solución a problemas combinatorios de tipo NP-Duros, y la necesidad de ofrecer una solución inmediata (segundos) dado su interés práctico, surge la necesidad del uso de heurísticas (algoritmos aproximados). “Se califica de heurístico a un procedimiento que tiene un alto grado de confianza, que encuentra soluciones de alta calidad con un coste computacional razonable, aunque no se garantice que los resultados sean optimos y en algunos casos, no se llegue a establecer lo cerca que se está de dicha situación” (Melián, Pérez, & Vega, 2003). Además, se usa el calificativo heurístico en contraposición a los métodos exactos, que se aplica en los procedimientos en donde se les exige que la solución aportada sea óptima o factible, sin tener en cuenta el tiempo computacional. Considerando la definición anterior y el tiempo disponible del turista, el cual razonablemente no puede aguardar demasiado tiempo para obtener una respuesta, que usando métodos exactos se extendería indefinidamente; se optó por la implementación de dos Heurísticas y una Metaheurística.

El problema propuesto en este trabajo de investigación es una variante del Problema del Agente Viajero: El Problema del Ciclo Simple, en este problema el ciclo solución no tienen por qué estar incluidas todas las ciudades (García, 2014, pág. 27). El turista propondrá todos los recursos turísticos que desea visitar, pero las restricciones de tiempo y dinero pueden restringir la visita a todos los estos recursos.

En la investigación realizada por (Isaza, 2012), se identificó que los algoritmos que ofrecen mayores ventajas para la resolución del TSP están basados en las metaheurísticas como: Colonia de Hormigas, Simulated Annealing y Búsqueda Tabú porque convergen en menor número de iteraciones. En los casos que involucran muchas ciudades (nodos), los métodos que más facilidades presentan son el Simulated Annealing, Colonia de Hormigas y el Búsqueda Tabú. Así mismo (Rejas Cano, 2014, pág. 2) afirma que TS llega a la solución en un menor tiempo que otros métodos existentes, obteniendo resultados iguales o mejores; de esta manera la propuesta de la presente investigación, es la implementación de la metaheurística Búsqueda Tabú.

La solución desarrollada será un entorno que sea capaz de ofrecer al turista un circuito turístico personalizado y que se generará automáticamente para él, teniendo en cuenta sus necesidades, gustos y experiencias; conceptos caracterizados mediante el modelado de perfiles específicos.

Es necesario mencionar que la visita de turistas responsables a los lugares que ofrezcan sus paquetes turísticos permitirá el mejoramiento de la calidad de vida del anfitrión; cumpliendo de este modo con la misión de nuestra universidad.

CAPÍTULO II

Marco Teórico

2.1. Antecedentes de la investigación

Con el fin de complementar el respaldo a la presente investigación, evitar la duplicidad de esfuerzos en cuanto a investigaciones similares en el mismo lugar de ejecución, resolver conflictos y señalar el camino para futuras investigaciones, se realizó la revisión de las diferentes fuentes comprendidas en el marco de esta investigación, y se presentan a continuación:

2.1.1 El problema del agente viajero: un algoritmo determinístico usando Búsqueda Tabú

En el trabajo de investigación realizado por (Lopez, Salas, & Murillo, 2013) se implementó un algoritmo basado al método Búsqueda Tabú. Consideraron tres modelos reales, dos para el Agente de la Salud y otro para el Asesor de Matemáticas, cuyas coordenadas se determinaron con el dispositivo GPS MAP Garmin 60CX y las distancias son tomadas de carreteras registradas (esto según datos del Instituto de Fomento y Asesoría Municipal). En la implementación de la metaheurística cada vecindario fue de tamaño $n-1$, se utilizó una portátil DELL Vostro 1000 con 1 GB de memoria RAM, procesador Mobile AMD Sempron (tm) Procesador 3600 + 2 GHz. Para probar la funcionalidad del modelo, se compararon con algunas instancias de la Travel Salesman Problem Library (TSPLIB), algunas instancias aleatorias y se aplicaron a tres situaciones reales.

Se determinó un conjunto de variables iniciales (longitud_tabú, número de iteraciones y número de diversificaciones) haciendo que el método tenga un mejor desempeño computacional para cada uno de los casos de prueba, calibrándolos por prueba y error.

Se verificó la eficiencia de la técnica búsqueda tabú comparando los resultados obtenidos por el algoritmo, con las mejores respuestas reportadas en la TSPLIB para los casos gr17, gr21, gr24, fri26, bayg29, gr48 y Berlin52, siendo los más representativos los casos gr17 y fri26.

2.1.2 Búsqueda Tabú para un problema de diseño de red multiproducto con capacidad finita en las aristas

En la Tesis de maestría realizada por (Cobos, 2004), se aborda un problema de diseño de red capacitada multiproducto, que tiene como objetivo determinar qué aristas deben considerarse en el diseño de forma que se garantice la operación de la red y que el costo total en que se incurra (considerando costos de diseño y de operación) sea el menor posible.

En este problema se deben transportar varios productos a través de una red con capacidades finitas en las aristas desde su origen hasta su respectivo destino. Al hacer esto, se incurre en un costo por transportar flujo llamado “costo variable”, así como un costo de construcción llamado “costo fijo”, por el uso de cada arista. Asociadas a cada arista está la capacidad, que será compartida por todos los productos que la usen sin importar la dirección del flujo. El objetivo es decidir cuáles aristas deben ser incluidas en el diseño de la red de forma que el costo total que se incurre por construcción y operación de la misma sea el menor posible.

Los objetivos de esta tesis fueron: desarrollar e implementar un sistema de optimización basado en técnicas inteligentes de búsqueda local aplicados a resolver el problema de diseño de red capacitada; evaluar computacionalmente el algoritmo, es decir, encontrar los valores de los parámetros algoritmos que provean soluciones de mayor calidad; y comparar los resultados obtenidos con un algoritmo evolutivo.

En relación con el desempeño del método de solución propuesto se puede notar, que, si bien los resultados son buenos en general, se desempeñó mejor en redes con costos variables predominantes sobre costos fijos. Esto se debe a que se seleccionan aristas caras lo cual conlleva a soluciones de mala calidad.

2.1.3 Interactive design of personalized tourism routes

En este artículo realizado por (Rodríguez, Molina, Pérez, & Caballero, 2012) refieren que en España las preferencias y comportamientos de los turistas, se inclinan cada vez más por abandonar los servicios de las agencias de viajes, buscando un itinerario que se adapte a sus necesidades. Las nuevas tecnologías ayudan a los turistas a planificar su viaje, se ofrece un fácil acceso a un gran volumen de información que permite al turista reunir información de los diferentes destinos; resulta difícil obtener una ruta óptima procesando toda la información de forma manual. Por lo tanto, existe la oportunidad de mejorar este proceso desarrollando una herramienta que facilite la organización de un viaje personalizado. El trabajo consistió en desarrollar una aplicación que considera los diferentes objetivos, las restricciones existentes, tanto impuestas por el turista, las impuestas por el destino y las actividades, ofreciendo así un circuito personalizado.

El problema que se enfrentó es multiobjetivo de actividades de selección y secuenciación. Debido a la complejidad de este problema se optó por aplicar el método metaheurístico Búsqueda Tabú.

La aplicación lleva el nombre de SAT, un sistema de apoyo turístico que incluye un modelo matemático que se adapta a la realidad del sector turismo. Además, hace uso de un

sistema de información geográfica que mide las distancias reales entre actividades y distancias no lineales.

2.1.4 Intelligent tourism recommender systems: A survey

Esta investigación realizada por (Borrás, Moreno, & Valls, 2014) centra su aplicación en el turismo, realiza una búsqueda exhaustiva de los recomendadores inteligentes de e-Tourism (Turismo electrónico) informado en las revistas y conferencias de Inteligencia Artificial desde el año 2008. El documento proporciona una encuesta detallada y actualizada de este entorno, considerando los diferentes tipos de interfaces, diversidad de algoritmos de recomendación, las funcionalidades ofrecidas por estos sistemas y sus técnicas de Inteligencia Artificial.

Analizando las interfaces utilizadas por los sistemas actuales de e-Tourism (turismo electrónico), los autores advierten que existe un predominio de los enfoques basados en la web, que son especialmente útiles para los turistas cuando planean una visita.

Algunas de las aplicaciones en estudio realizan de forma automática el proceso de planificar un viaje a medida. La ruta es casi completa tomando el contexto del usuario como la ubicación, la hora y el horario de apertura, entre otros.

Por último, las técnicas de optimización ofrecen soluciones a complejos problemas de planificación y programación cuando el sistema quiere construir automáticamente una ruta de varios días.

Como resultado de esta revisión analítica, algunas pautas básicas pueden ser seguidas en el diseño y desarrollo de los sistemas inteligentes de turismo; manifiestan los autores.

2.2. Bases teóricas

2.2.1 Inteligencia Artificial

(Romero, 2007, pág. 10) afirma que, “la IA es la rama de la ciencia que se encarga del estudio de la inteligencia en elementos artificiales y, desde el punto de vista de la ingeniería, propone la creación de elementos que posean un comportamiento inteligente. Dicho de otra forma, la IA pretende construir sistemas y máquinas que presenten un comportamiento que, si fuera llevado a cabo por una persona, se diría que es inteligente”.

Mediante la prueba de Turing. propuesta por Alan Turing (1950), se intenta una satisfactoria definición de lo que es la inteligencia. Turing definió una conducta inteligente como la capacidad de lograr eficiencia a nivel humano en todas las actividades de tipo cognoscitivo, suficiente para engañar a un evaluador. Brevemente, la prueba que Turing propuso que consistía en que un humano interrogase a una computadora por medio de un teletipo; la prueba se consideraba aprobada si el evaluador era incapaz de determinar si una computadora o un humano era quien había respondido las preguntas en el otro extremo de la terminal. Actualmente, el trabajo que entraña programar una computadora para pasar la prueba es considerable. La computadora debería ser capaz de lo siguiente:

- Procesar un lenguaje natural, para así poder establecer comunicación satisfactoria, sea en inglés o en cualquier otro idioma humano.
- Representar el conocimiento, para así guardar toda la información que se le haya dado antes o durante el interrogatorio.
- Razonar automáticamente, con el fin de utilizar la información guardada al responder preguntas y obtener nuevas conclusiones.
- Autoaprendizaje de la máquina, para que se adapte a nuevas circunstancias y para detectar y extrapolar esquemas determinados.

“La necesidad de actuar como los humanos se presenta básicamente cuando los programas de IA deben interactuar con la gente, por ejemplo, cuando un sistema experto ha de explicar lo justificado de un diagnóstico, o cuando un sistema de procesamiento de lenguaje natural entabla diálogo con un usuario” (Russell, 2008, pág. 15).

En la IA un agente es cualquier cosa capaz de percibir su medioambiente con la ayuda de sensores y actuar en ese medio utilizando actuadores, la Figura 2 ilustra esta idea de manera gráfica.

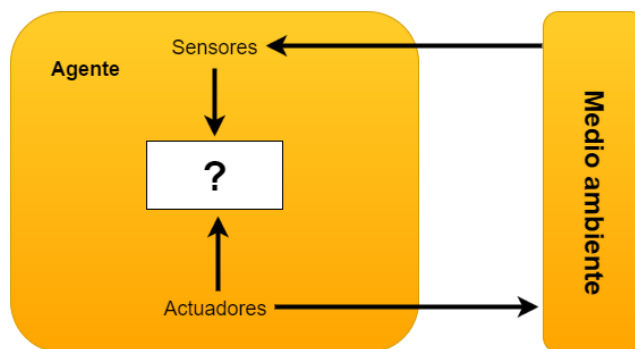


Figura 2. Representación de un agente interactuando con el medioambiente.

Fuente: Inteligencia Artificial Un Enfoque Moderno Segunda edición. Madrid: Ara Isabel Garcia Borno, 2008.

“Un agente humano tiene ojos, oídos y otros órganos sensoriales. Un agente robot recibe pulsaciones del teclado, archivos de información y paquetes vía red. Se trabajará con la hipótesis general de que cada agente puede percibir sus propias acciones” (Russell, 2008, pág. 38).

Los agentes inteligentes deben maximizar su procesos de rendimiento, esto puede simplificarse algunas veces si el agente puede seleccionar un objetivo y tratar de satisfacerlo. Se considera un objetivo como un conjunto de estados del mundo real (exactamente aquellos estados que satisfacen el objetivo). La trabajo del agente es encontrar qué secuencia de acciones permite obtener un estado objetivo. Para esto se necesita decidir qué acciones y estados permite obtener un estado objetivo (Russell, 2008, pág. 38).

La solución a estos problemas puede ser dado mediante algoritmos. “Los algoritmos son procedimientos sistemáticos de solución para resolver problemas, algunos de estos algoritmos se utilizan para solucionar problemas que incluyen cientos o miles de variables” (Hillier, 2010, pág. 4). Existen varios tipos de algoritmos, en esta investigación se describen los algoritmos de búsqueda como premisa para el desarrollo de la investigación, y se presentan a continuación.

2.2.2 Algoritmos de búsqueda

Los Algoritmos de Búsqueda constituyen una de las tecnologías más importantes de la IA, cuyo objetivo principal consiste en hallar una solución válida dentro del espacio de estados. Entre otras técnicas que se pueden implementar con estos algoritmos se conocen las de búsqueda no informada y búsqueda con información (Hossian, 2015, pág. 197), las que se presentan seguidamente.

2.2.2.1 Búsqueda no informada

También llamada Búsqueda a Ciegas. “El término significa que ellas no tienen información adicional acerca de los estados más allá de la que proporciona la definición del

problema. Todo lo que ellas pueden hacer es generar los sucesores y distinguir entre un estado objetivo de uno que no lo es” (Russell, 2008, pág. 82). Todas las estrategias se distinguen por el orden de expansión de los nodos, existen dos búsquedas de anchura; Búsqueda en Profundidad y Búsqueda Bidireccional, las cuales serán descritas a continuación

2.2.2.1.1 Búsqueda en anchura

La búsqueda en anchura es una estrategia sencilla en la que se expande primero el nodo raíz, a continuación, se expanden todos los sucesores del nodo raíz, después los sus sucesores de estos, etc. En general se expanden todos los nodos a una profundidad en el árbol de búsqueda antes de expandir cualquier nodo del próximo nivel.

La búsqueda primero en anchura se puede implementar llamando a la Búsqueda de tipo Árbol, asegurando que los nodos primeros visitados serán los primeros en entrar primero en salir, asegurando que los nodos primero visitados serán los primeros expandidos (Russell, 2008, pág. 82).

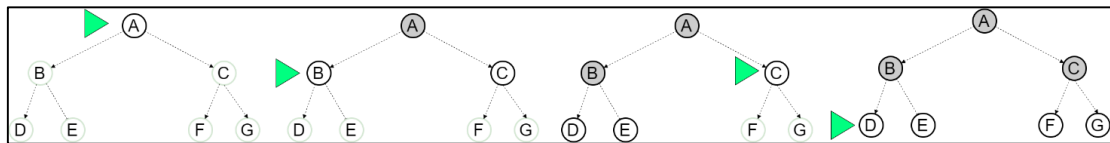


Figura 3. Búsqueda en anchura sobre un árbol binario sencillo.

Fuente: Inteligencia Artificial Un Enfoque Moderno Segunda edición. Madrid: Ara Isabel Garcia Borno, 2008.

2.2.2.1.2 Búsqueda en profundidad

La búsqueda primero en profundidad siempre expande el nodo más profundo en la frontera actual del árbol de búsqueda. El progreso de la búsqueda se ilustra en la figura 4. La figura inicia procesando inmediatamente al nivel más profundo del árbol de búsqueda, donde los nodos no tienen cuanran con ningún sucesor. Cuando dichos nodos se expanden, son eliminados de la frontera, y la búsqueda retrocede al siguiente nodo más superficial que todavía tenga sucesores sin explorar ((Russell, 2008, pág. 86).

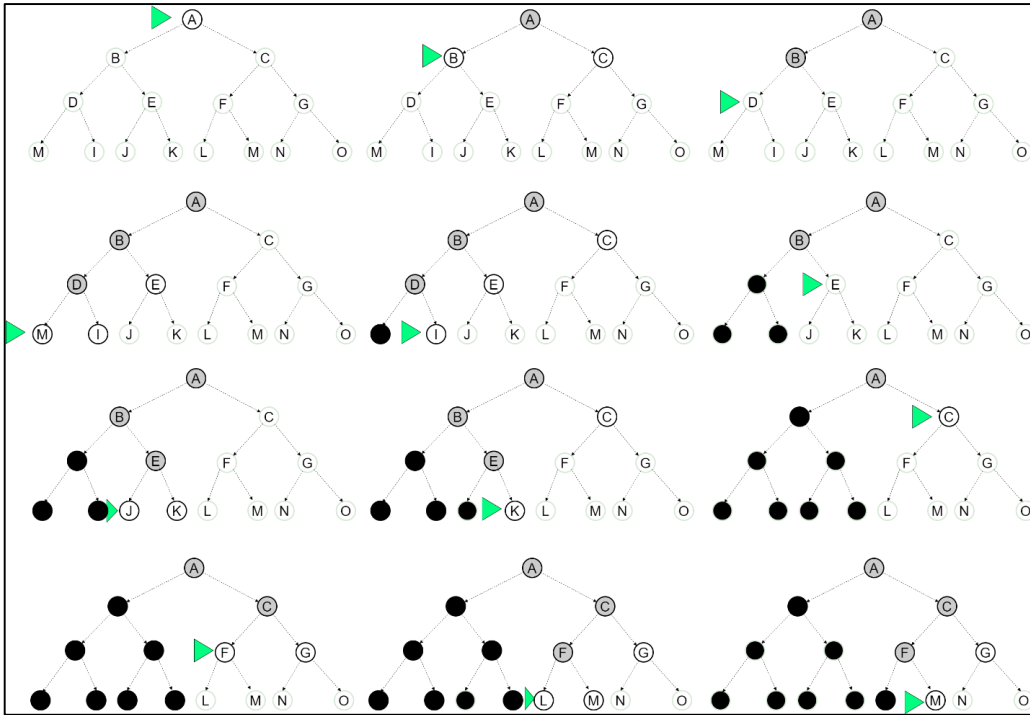


Figura 4. Búsqueda en profundidad sobre un árbol binario.

Fuente: Inteligencia Artificial Un Enfoque Moderno Segunda edición. Madrid: Ara Isabel Garcia Borno, 2008.

2.2.2.1.3 Búsqueda bidireccional

El objetivo de una búsqueda bidireccional es ejecutar dos búsquedas en paralelo: una hacia adelante comenzando del estado inicial y la otra hacia atrás desde el objetivo, deteniéndose cuando las búsquedas se interceptan en la parte central (Figura 5). La motivación es que $b^{\frac{d}{2}} + b^{\frac{d}{2}}$ es mucho menor que b^d donde $d=profundidad$ y $b=factor\ de\ ramificación$, el área de los dos círculos pequeños es menor que el área de un círculo grande centrado en el inicio y que alcance al objetivo (Russell, 2008, pág. 89).

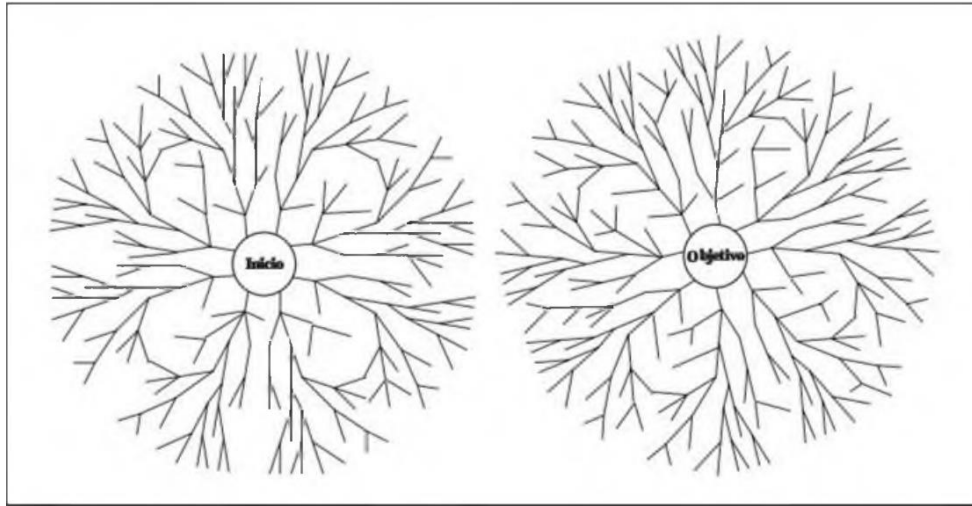


Figura 5. Un esquema de una búsqueda bidireccional que está a punto de tener éxito.

Fuente: Russell, S. (2008). Inteligencia Artificial Un Enfoque Moderno Segunda edicion. Madrid: Ara Isabel Garcia Borno.

2.2.2.2 Búsqueda informada

“La búsqueda informada (la que utiliza el conocimiento específico del problema más allá de la definición del problema en sí mismo) puede encontrar soluciones, de una manera más eficiente que una estrategia no informada” (Russell, 2008, pág. 107).

Para el desarrollo de la heurística de construcción y mejoría de esta investigación se utilizará la técnica de Búsqueda Informada; se tomarán en cuenta los datos recopilados de las distancias entre atractivos turísticos, costos y tiempo.

2.2.2.2.1 Búsqueda voraz primero el mejor

Este método realizar una expansión del vertice más cercano al objetivo, justificando que probablemente nos conduzca en un tiempo razonable a una solución. Así, evalúa los vertices utilizando solamente la función heurística: $f(n) = h(n)$.

Se presenta el trabajo de este método para el problema de encontrar una ruta en Rumanía utilizando las distancia en línea recta, que se denominarán h_{DLR} . Si el objetivo es Bucarest, se conocen las distancias en línea recta a Bucarest, que se muestran en la Tabla 1.

Tabla 1

Valores de h_{DLR} distancias en línea recta a Bucarest.

Ciudad	Distancia	Ciudad	Distancia
Arad	366	Mehadla	241
Bucarest	0	Neamt	234
Cralova	160	Oradea	380
Dobreta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Glurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vasiui	199
Lugoj	244	Zerind	374

Fuente: Inteligencia Artificial Un Enfoque Moderno Segunda edicion. Madrid: Ara Isabel Garcia Borno, 2008.

En la Figura 6 se muestra el progreso de una búsqueda primero el mejor avatar con h_{DLR} para encontrar un camino desde Arad a Bucarest. El primer nodo a expandir desde Arad será Sibiu, porque está más cerca de Bucarest que Zerind o que Timisoara. El siguiente nodo a expandir será Fagaras, porque es la más cercana. Fagaras en su turno genera Bucarest, que es el objetivo.

La búsqueda voraz primero el mejor es similar a la búsqueda primero en profundidad en donde prefiere seguir un camino hacia el objetivo, pero retornará atrás cuando llegue a un callejón sin salida. Sufre los mismos resultados que la búsqueda primero en profundidad, no es del todo óptima, y es incompleta (porque se puede dirigir hacia abajo en un camino infinito y jamas volver para intentar otras posibilidades). La complejidad en tiempo y espacio en el peor caso es igual a: $O(b)^m$, donde m es la profundidad máxima del espacio de búsqueda. (Russell, 2008, pág. 108).

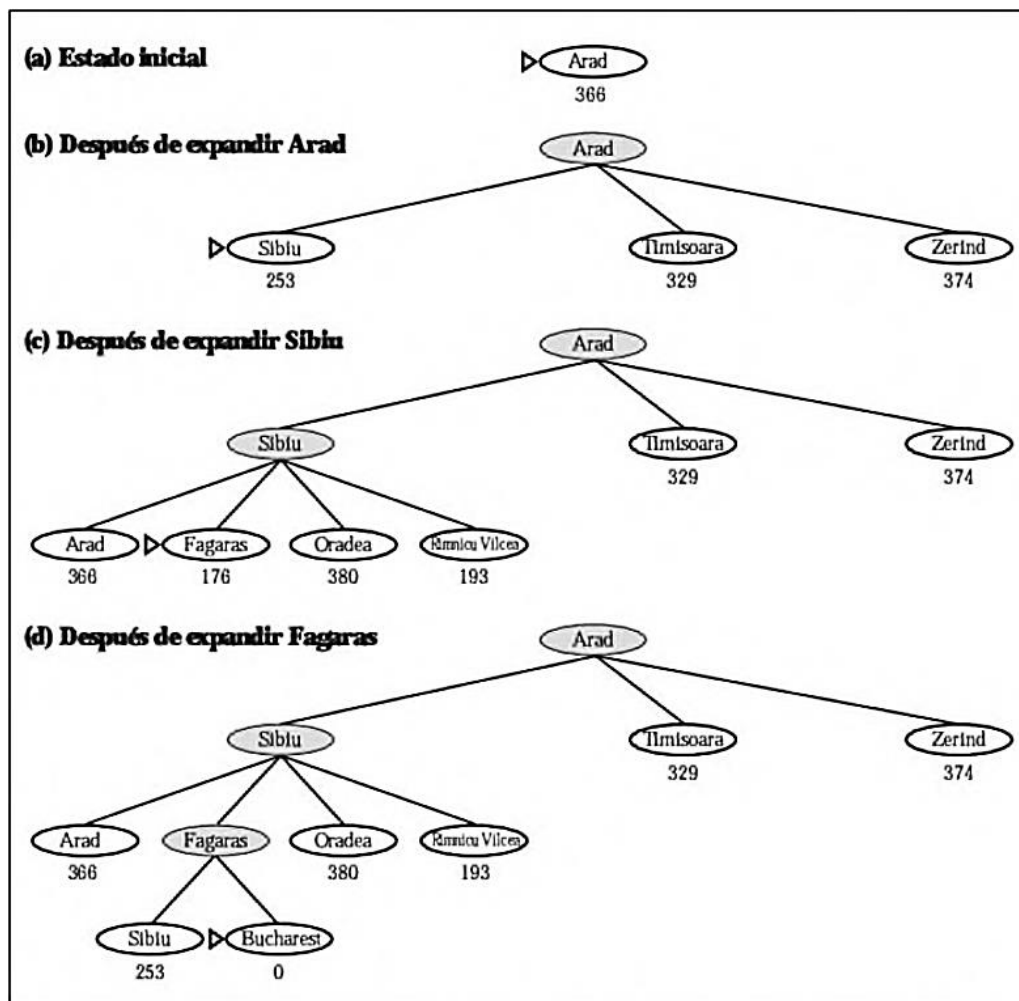


Figura 6. Etapas de una búsqueda primero el mejor para Bucarest utilizando la heurística distancia en línea recta.

Fuente: Inteligencia Artificial Un Enfoque Moderno Segunda edición. Madrid: Ara Isabel Garcia Borno, 2008.

2.2.2.2.2 Búsqueda A*: Minimizar el costo total de la solución

A la forma más ampliamente conocida de la búsqueda primero el mejor se le llama búsqueda A* (pronunciada: búsqueda A-estrella). Evalúa los nodos combinando $g(n)$, el coste para alcanzar nodo, y $h(n)$, el coste de ir al nodo objetivo:

$$f(n) = g(n) + h(n)$$

Ya que la $g(n)$ nos da el coste del camino desde el nodo inicio al nodo n , y la $h(n)$ el coste estimado es más barato desde n al objetivo, tenemos:

$$f(n) = \text{coste más barato estimado de la solución a través de } n$$

Así, si se trata de encontrar la solución más barata, es razonable intentar primero el nodo con el más bajo de $g(n) + h(n)$. Esta estrategia es más que razonable: con tal de que la función heurística $h(n)$ satisfaga ciertas condiciones, la búsqueda A* es tanto completa como óptima.

La optimalidad de A^* es sencilla de analizar si se usa con la Búsqueda-Árboles. En este caso, A^* es óptima si $h(n)$ es una heurística admisible, es decir, con tal de que la $h(n)$ nunca sobrestime el coste de alcanzar el objetivo. Las heurísticas admisibles son por naturaleza optimistas, porque piensan que el coste de resolver el problema es menor que el que es en realidad. Ya que $g(n)$ es el coste exacto para alcanzar n , se tiene como consecuencia inmediata que la $f(n)$ nunca sobrestima el coste verdadero de una solución a través de n (Russell, 2008, pág. 111).

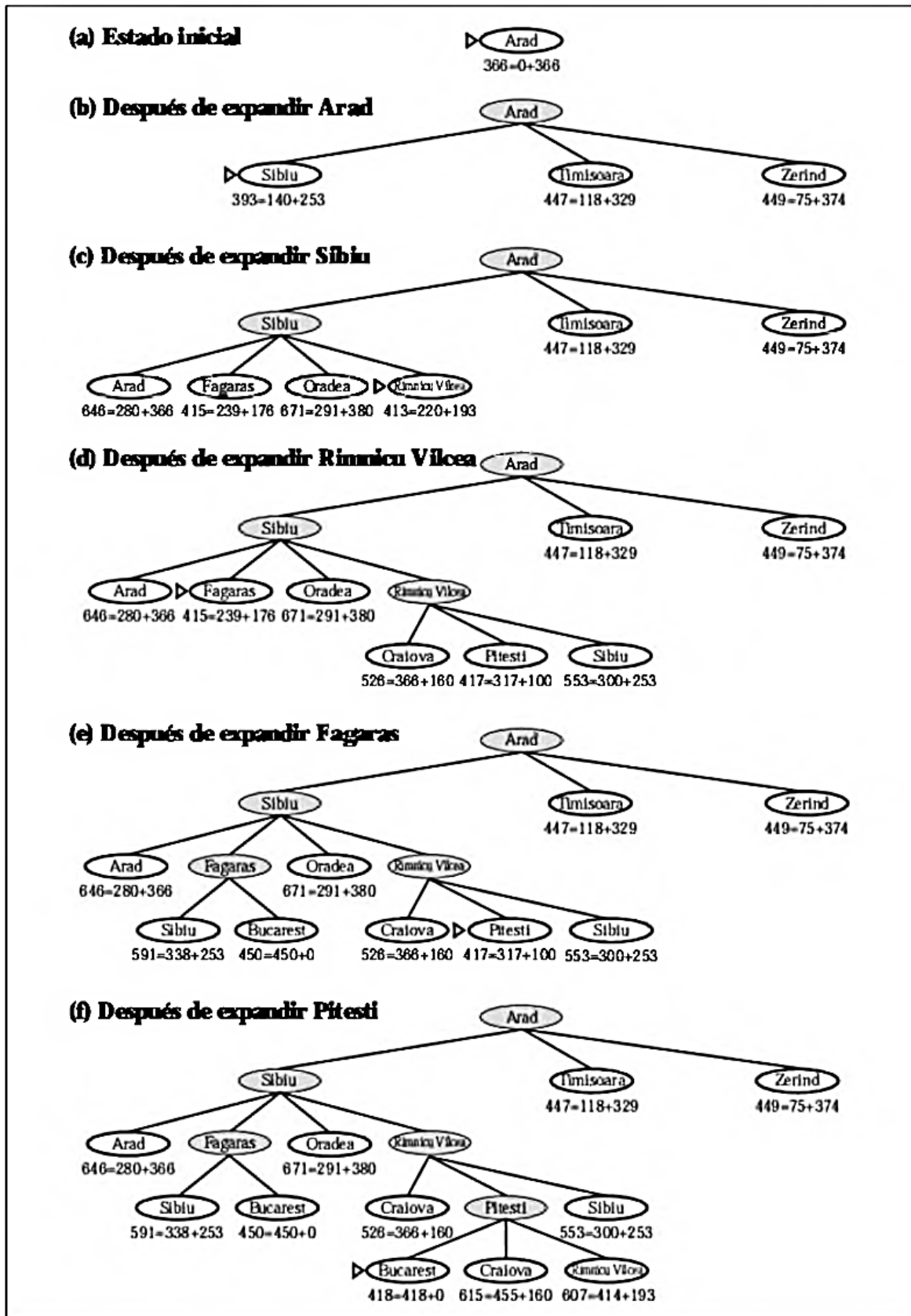


Figura 7. Etapas en una búsqueda A* para Bucarest etiquetando los nodos con $f=g+h$.

Fuente: Inteligencia Artificial Un Enfoque Moderno Segunda edición. Madrid: Ara Isabel Garcia Borno, 2008.

2.2.3 Métodos exactos y heurísticos

Los métodos para resolver problemas de optimización combinatoria se pueden clasificar de diferentes maneras, una de sus clasificaciones más generales según los investigadores de estas áreas corresponde a los métodos exactos y heurísticos, los cuales son presentados a continuación.

2.2.3.1 Métodos exactos

Según (García, 2014, pág. 8), los métodos exactos proporcionan una solución óptima del problema, sin tomar en consideración el tiempo computacional. Procesarán todas las soluciones posibles en función a las restricciones impuestas y el número de nodos, con gran cantidad de nodos el tiempo de procesamiento crecerá exponencialmente siendo en algunos casos indefinido. Algunos de estos modelos son:

- a) Ramificación y acotación
- b) Hiperplanos de corte
- c) Ramificación y corte

2.2.3.2 Método heurístico

Una de sus definiciones sería: “Es método (algoritmo) que con poco esfuerzo computacional proporciona una solución aproximada, pero no necesariamente óptima del problema” (García, 2014, pág. 8). Una clasificación de este tipo de problemas son:

- a) Algoritmos constructivos: Heurísticas que tratan de construir una solución aceptable sin depender de ninguna previa.
- b) Heurísticas voraces: El algoritmo hace uso del mejor elemento, teniendo en cuenta el objetivo a optimizar, pero no toma en cuenta consecuencias en el futuro, ya que tras tomar una decisión esta no vuelve a ser considerada en futuras iteraciones.
- c) Heurística de mejoría: EL objetivo principal es comenzar con una solución inicial en donde se eliminarán n aristas o arcos para luego añadirle la misma cantidad con el objetivo de mejorar la solución.
- d) Heurística de multiarranque: Trabaja con probabilidades y procesos aleatorios que nos dan mejores soluciones.

Tras este análisis general de los diferentes aspectos las cuales forman parte de la denominada “Programación Matemática” y están relacionadas con la presente investigación, se hace evidente la necesidad de trabajar con Métodos Heurísticos por contemplar un tiempo computacional razonable; de esta manera el usuario final (Turista) no tendrá que esperar un tiempo prolongado para obtener un circuito turístico a su medida.

2.2.4 Heurística

“En Inteligencia Artificial (IA), se habla de heurística para referirse a una técnica, método o procedimiento inteligente para realizar una tarea que no es producto de un riguroso análisis formal, sino de conocimiento experto sobre la tarea” (Melián, Pérez, & Vega, 2003, pág. 1).

La heurística está diseñada para encontrar buenas soluciones aproximadas de problemas combinatorios difíciles que de lo contrario no pueden resolverse mediante los algoritmos de optimización disponibles. Una heurística es una técnica de búsqueda directa que utiliza reglas favorables prácticas para localizar soluciones mejoradas. La ventaja de la heurística es que en general determina buenas soluciones con rapidez, utilizando reglas de solución simples. La desventaja es que la calidad de la solución con respecto a la óptima suele desconocerse (Taha, 2012, pág. 351).

(Taha, 2012, pág. 351) también manifiesta que: “Las primeras generaciones de heurísticas se basan en la regla de búsqueda codiciosa que dicta que se mejore el valor de la función objetivo con cada movimiento de búsqueda. La búsqueda termina en un óptimo local donde ya no son posibles más mejoras”.

En los últimos años ha habido un crecimiento en el desarrollo de procedimientos heurísticos para resolver problemas combinatorios, debido a la necesidad de disponer de herramientas que permitan ofrecer soluciones rápidas a problemas reales. Los problemas de optimización combinatoria en particular son de gran dificultad debido a su complejidad, ya que crecen exponencialmente con el tamaño del problema, por lo cual se pretende que los métodos heurísticos se acercan a la solución óptima en un tiempo razonable. (Cobos, 2004, pág. 6)

2.2.4.1 Heurística de construcción

Un concepto correcto sería el dado por (Gonzales Silos, 2014, pág. 6) “Las heurísticas de construcción son las que contribuyen una solución factible inicial que normalmente se usa para dar la primera solución inicial, tienen que ser sencillas, pero con soluciones aceptables”. Para la implementación de la heurística de mejoría, se tomará como datos de entrada, a la mejor solución obtenida mediante la heurística de construcción.

2.2.4.2 Heurística del vecino más cercano

Una de las heurísticas para el TSP es el llamado “vecino más cercano”, que trata de construir un ciclo Hamiltoniano de bajo coste basándose en el vértice cercano a uno dado.

Su código, en una versión standard, es el siguiente:

Inicialización

Seleccionar un vértice j al azar.

Hacer $t = j$ y $W = V \setminus \{j\}$.

Mientras ($W \neq \emptyset$)

Tomar $j \in / c_{tj} = \min \{c_{ti} / i \in W \}$

Conectar t a j

Hacer $W = W \setminus \{j\}$ y $t = j$

Este procedimiento ejecuta un cantidad de operaciones de orden $O(n^2)$. Siguiendo con la evolución del algoritmo al construir una solución, se identifica que comienza bien, seleccionando aristas de bajo coste. Pero sin embargo, al final de la ejecución probablemente quedarán nodos cuya conexión obligará a introducir aristas con un costo elevado. Esto es lo que se conoce como miopía del procedimiento, ya que, en una iteración elige la mejor opción disponible sin ver que esto puede obligar a realizar malas elecciones en iteraciones posteriores.

El algoritmo base puede ser programado en unas pocas líneas de código. Pero, una implementación directa será muy lenta al ejecutar problemas de gran tamaño. Así pues, incluso para una heurística tan sencilla como ésta, es importante pensar en la eficiencia y velocidad de su código.

2.2.4.3 Heurística de mejoría

(Riojas, 2005, pág. 24) afirma que las Heurísticas de mejoría, “comienzan con una solución del problema y la mejoran progresivamente. El método finaliza cuando no existe ninguna solución accesible que mejore la anterior”. La solución a mejorar en esta investigación, será la que se obtenga de la heurística del vecino más cercano.

2.2.4.4 Heurística de 2 intercambio (2-opt)

Este procedimiento está basado en la siguiente observación para grafos con distancias euclídeas (o en general con costes cumpliendo la desigualdad triangular): Si un ciclo Hamiltoniano se cruza a sí mismo, puede ser fácilmente acortado, basta con eliminar las dos aristas que se cruzan y reconectar los dos caminos resultantes mediante aristas que no se corten. El ciclo final es más corto que el inicial.

“Un movimiento 2-opt consiste en eliminar dos aristas y reconectar los dos caminos resultantes de una manera diferente para obtener un nuevo ciclo” (Marti, 2013, pág. 26) .

El siguiente código recoge el algoritmo heurístico de mejora 2-óptimo y se presenta a continuación:

Inicializar

*Considerar un ciclo Hamiltoniano inicial
movimiento=1*

Mientras (movimiento=1)

movimiento=0. Etiquetar todos los vértices como no explorados.

Mientras (Queden vértices por explorar)

*Seleccionar un vértice i no explorado. Examinar todos los
movimientos 2-opt que incluyan la arista de i a su sucesor en el ciclo.*

*Si alguno de los movimientos examinados reduce la longitud del
ciclo, realizar el mejor de todos y hacer movimiento= 1.*

En otro caso etiquetar i como explorado

2.2.5 Metaheurística

“El sufijo ‘meta’ significa ‘más allá’, a un nivel superior, las metaheurísticas son estrategias para diseñar o mejorar los procedimientos heurísticos con miras a obtener un alto rendimiento. El término metaheurística fue introducido por Fred Glover en 1986 y a partir de entonces han aparecido muchas propuestas de pautas o guías para diseñar mejores procedimientos de solución de problemas combinatorios” (Riojas, 2005, pág. 25).

“Una metaheurística también es un tipo general de método de solución que organiza la iteración entre los procedimientos de mejora local y las estrategias de más alto nivel para crear un proceso que sea capaz de escapar de un óptimo local realizando una búsqueda vigorosa de una región factible”. (Taha, 2012, pág. 351). Lo anterior hace referencia a la característica clave de las metaheurísticas; que es la capacidad de escapar de un óptimo local.

2.2.6 Metaheurística Búsqueda Tabú

La búsqueda Tabú es una técnica iterativa de búsqueda local inteligente que trata de evitar que las soluciones caigan en óptimos locales. Para esto se utilizan unas estructuras de memoria de corto y largo plazo, acompañadas de criterios de aspiración. En esta técnica en una iteración se pretende pasar de una solución a la mejor solución vecina, sin importar si esta es mejor o peor que la solución actual. El criterio de terminación puede ser un cierto número máximo de iteraciones o un valor de la función por optimizar. Entre las características relevantes que posee este método se encuentran la denominada lista tabú y el criterio de aspiración. El objetivo más general de la lista tabú es continuar estimulando el descubrimiento de soluciones de alta calidad. En general, un tipo común de restricción opera seleccionando algún subconjunto de atributos y declarando un movimiento tabú un determinado número de veces. (Daza, Montoya, & Narducci, 2009, pág. 5).

El esquema de funcionamiento de este algoritmo en pseudocódigo es el siguiente:

Procedimiento Búsqueda Tabú (X , var X^m):

Hacer $x^m = x$ y $niter = 0$

Repetir

Hacer $k = 0$, $x^0 = x$ y $inter = inter + 1$:

Repetir

Hacer $k = k + 1$;

Definir $N = \left\{ \begin{array}{l} j \text{ no es tabú o cumple criterio de aspiración} \\ j \\ y x_j^0 > 0 \end{array} \right\}$

Si $N \neq \emptyset$ hacer

Determinar $beta_{j^*} = \max\{beta_j / j \in N\}$

Hacer $= -1$ y, $i=1 \dots h$

Hacer $x^1 = x^0$ y ejecutar Proyección(X^1)

Hasta ($k = ka$) o ($N = \emptyset$) o ($f(x^1) < f(x^m)$) {Fin paso 1}

Definir $A = \{j / x_j^i > x_j\}$ y Hacer $lista_tabú_j = inter$, $j \in A$ {Actualizar}

Hacer $x = x^1$ y actualizar x^m {Mejor Solución encontrada}

Hasta criterio de parada.

Como se puede observar este procedimiento sigue una estrategia y es: alejarse de la región factible quitando elementos y volver a ella a continuación con el procedimiento de proyección (Pacheco, 2003, pág. 547)

2.2.7 Investigación de Operaciones

Como su nombre lo indica, el objetivo de esta disciplina implica “investigar sobre las operaciones”. En consecuencia, se aplica a la problemática relacionada con la conducción y la coordinación de actividades en una organización.

La IO incluye el término investigación en su nombre porque utiliza un enfoque similar al que se aplica en las áreas científicas establecidas. El método científico se utiliza para explorar los diversos problemas que deben ser tratados, pero en ocasiones se usa el término management science o ciencia de la administración como sinónimo de investigación de operaciones. El proceso comienza por la observación cuidadosa y la formulación del problema, lo cual incluye el levantamiento de la información (datos) pertinentes. El siguiente paso es la construcción de un modelo científico (generalmente matemático) con el cual se intenta abstraer la esencia del problema del mundo real. En esta etapa se propone la hipótesis de que el modelo será la representación precisa de las características esenciales de

la realidad, permitirá también que las conclusiones sean validas. Después se llevan a cabo los experimentos necesarios para probar la hipótesis, modificarla si es necesario y para verificarla en determinados momentos, paso que se conoce como validación del modelo.

“En cierto sentido, la IO involucra la investigación científica creativa de las propiedades fundamentales de las operaciones. Sin embargo, es más que esto, la IO se ocupa también de la administración práctica de la organización; por lo tanto, para tener éxito, también debe proporcionar conclusiones claras que el tomador de decisiones pueda usar cuando sea necesario” (Hillier, 2010, pág. 3). En la tabla 2 se presentan los beneficios de su aplicación en instituciones reales.

Tabla 2

Aplicaciones de la investigación de operaciones en organizaciones reales y su impacto.

Organización	Área de aplicación	Ahorros anuales
Federal Express	Planeación logística de envíos.	No estimado.
Continental Airlines	Reasignación de tripulaciones a vuelos cuando ocurren interrupciones en el itinerario.	\$40 millones.
Swift & Company	Mejora del desempeño en ventas y manufactura.	\$12 millones.
Memorial Sloan-Kettering Cancer Center	Diseño de terapia de radiación.	\$459 millones.
United Airlines	Plan para los programas de trabajo de los empleados en aeropuertos y oficinas de reservación.	\$6 millones.
Welch 's	Optimización del uso y movimiento de materias primas.	\$150 000
Samsung Electronics	Reducción de tiempos de manufactura y niveles de inventario.	Ganancias adicionales de \$200 millones.
Pacific Lumber Company	Gestión de ecosistemas forestales a largo plazo.	\$398 millones VPN.
Procter & Gamble	Rediseño del sistema de producción y distribución.	\$200 millones.
Canadian Pacific Railway	Plan de rutas para un tren de carga.	\$100 millones.
United Airlines	Reasignación de aviones a vuelos cuando ocurren interrupciones.	No estimado.
Ejército de Estados Unidos	Planeación logística de la Operación Tormenta del Desierto.	No estimados.
Air New Zealand	Programación de tripulaciones en una aerolínea.	\$6.7 millones.

Taco Bell	Planeación de los programas de trabajo de los empleados de restaurantes.	\$13 millones.
Gestión de desperdicios	Desarrollo de un sistema de administración de rutas para la recolección y disposición de basuras.	\$100 millones.
Bank Hapoalim Group	Desarrollo de un sistema de apoyo a las decisiones de asesores en inversiones.	Ganancias adicionales de \$31 millones.
Sears	Rutas y programación de vehículos para servicios y entregas a domicilio.	\$42 millones.
Conoco-Philips	Evaluación de proyectos de exploración petrolera.	No estimado.
Oficina de compensaciones a los trabajadores	Gestión de solicitudes de incapacidad y rehabilitación de alto riesgo.	\$4 millones.
Westinghouse	Evaluación de proyectos de investigación y desarrollo.	No estimado.
Merril Lynch	Administración de riesgos de liquidez de líneas de crédito revolventes.	Liquidez adicional de \$4 millones.
PSA Peugeot Citroen	Guía para el proceso de diseño de plantas de ensamble de automóviles eficientes.	Utilidades adicionales de \$4 mil millones.
KeyCorp	Mejora de la eficiencia de los cajeros de banco.	\$20 millones
General Motors	Mejora de la eficiencia de líneas de producción.	\$90 millones
Deere & Company	Administración de inventarios a lo largo de una cadena de suministro.	Reducción de \$1000 millones en inventario.
Time Inc.	Administración de canales de distribución de revistas.	Utilidades adicionales de 3.5 millones.
Bank One Corporation	Administración de líneas de crédito y tasas de interés de tarjetas de crédito.	Utilidades adicionales de \$75 millones.
Merril Lynch	Análisis de precios de provisión de servicios financieros.	Ganancias adicionales de \$50 millones.
AT&T	Diseño y operación de centros de atención telefónica.	Utilidades adicionales de \$750 millones.

Fuente: Introducción a la Investigación de Operaciones. México: Edmundo Carlos Zúñiga Gutiérrez, 2010.

2.2.8 Python

Python es un lenguaje de programación que cuenta con estructuras de datos eficientes, de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripling y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas existentes. El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C). Python también puede usarse como un lenguaje de extensiones para aplicaciones personalizables (Van, 2009).

2.2.9 Programación Extrema (XP)

La Programación Extrema o Extreme Programing, es un enfoque de la ingeniería de software formulado por Kent Beck, se considera el más destacado de los procesos ágiles de desarrollo de software. Al igual que estos, la programación extrema se diferencia de los métodos tradicionales principalmente en que presenta más énfasis en la adaptabilidad que en la previsibilidad (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016, pág. 24). Algunas de sus características son:

- a) Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.
- b) Se aplica de manera dinámica durante el ciclo de vida del software.
- c) Es capaz de adaptarse a los cambios de requisitos.
- d) Los individuos e interacciones son más importantes que los procesos y herramientas.
- e) Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.

CAPÍTULO III

Materiales y métodos

El método para el desarrollo del algoritmo de Búsqueda Tabú a implementar durante esta investigación seguirá el esquema básico del algoritmo descrito por (Daza, Montoya, & Narducci, 2009) presentado en el marco conceptual.

La programación del algoritmo y la aplicación web que contendrá a éste, se realizará adoptando la metodología ágil denominada Programación Extrema (XP), XP brinda avances con cierta regularidad y para ser evaluados por el cliente. De esta manera se presentaron avances semanales del proyecto, los cuales eran evaluados por el cliente (Actores del TRC).

3.1. Materiales

Para levantar la información de los atractivos turísticos del distrito de Juli, fue necesario elaborar dos formularios basados en el “Manual para la formulación del Inventario de Recursos Turísticos a Nivel Nacional” del MINCETUR. Estos formularios se modificaron de acuerdo a las variables necesarias para la validación de la Metaheurística, y se presentan en los Anexos B y C. Los datos recopilados en estos formularios se transcribieron en una lista consolidada, la cual es presentada en el Anexo A.

Asimismo, se obtuvo datos artificiales (instancias) de repositorios especializados para validar este tipo de algoritmos, algunos grupos de instancias se presentan en el Anexo G.

Las herramientas para el desarrollo de la investigación y la validación de la Metaheurística fueron:

- Laptop TOSHIBA Disco duro 1 TB, memoria RAM 16 Gb, procesador i7 2.60GHz, pantalla 15". Computador donde se ejecutó el algoritmo.
- Python 3.2. Lenguaje de programación ágil.
- Django 1.9.2. Framework para el desarrollo de aplicaciones web.
- Angular 1.5. Framework para el lado del cliente.
- PyCharm 2016.1.3. Ambiente de desarrollo integrado para Python
- WebStorm 2016.1.2. Ambiente de desarrollo integrado para Angular JS.
- Django Rest 3.3.3. Framework para creación de servicios para ser consumidos por cualquier dispositivo.
- Balsamiq 2.1. Diseño de prototipos para el desarrollo software.

3.2. Tipo de investigación

La presente investigación es de tipo aplicada. Ya que se realizaron experimentos en el diseño y desarrollo de los algoritmos, tomamos experiencia de los procedimientos iniciales, estableciendo nuevas estrategias para lograr soluciones finales aceptables.

3.3. Diseño de la solución propuesta

3.3.1 Definición del problema basado en el Problema del Agente Viajero

Como primer paso, se definió el problema del mundo real en un problema combinatorio establecido en la literatura (TSP). La variante del TSP que representa el problema real, se denomina Problema del Ciclo Simple (frase en inglés - SCP). En el ciclo solución no tienen por qué estar incluidas todas las ciudades, esto se debe a que, el visitante limitará su itinerario con las restricciones de tiempo y/o costos. El modelo matemático para representar el problema real es el siguiente:

- Un viajante debe partir de su ciudad y recorrer N ciudades volviendo a la ciudad de origen sin repetir ninguna. La distancia entre la ciudad i y j es c . Se asocia una variable binaria Y_{ij} a cada par de ciudades i, j .
- La función objetivo consiste en minimizar las distancias y/o costos sujetos a múltiples restricciones que se le impongan.

$$\min_y \sum_{i=1}^N \sum_{j=1}^N c_{ij} y_{ij} \quad y_{ij} = \begin{cases} 0 & \text{si el viajante no va de la ciudad } i \text{ a la } j \\ 1 & \text{si el viajante va de la ciudad } i \text{ a la } j \end{cases} \quad y_{ii} = 0$$

- Restricciones

$$\sum_{i=1}^N y_{ij} = 1 \quad j = 1, \dots, N \quad \text{debe llegar una vez y solo una de la ciudad } j.$$

$$\sum_{j=1}^N y_{ij} = 1 \quad i = 1, \dots, N \quad \text{debe llegar una vez y solo una de la ciudad } i.$$

3.3.2 Diseño e implementación de la heurística de construcción

La primera solución que satisfaga las restricciones impuestas en el problema establecido, se obtuvo de la implementación de una heurística de construcción basada en el método llamado “el vecino más cercano”, el objetivo fue construir un ciclo Hamiltoniano de bajo coste basándose en el vértice (nodo o recurso turístico) más cercano a un dado. El pseudocódigo de este método está especificado en las bases teóricas de esta investigación.

Se diseñó un diagrama de flujo que representa al modelo y que satisface a la función objetivo como se muestra en la Figura 8. El proceso comienza definiendo las siguientes variables:

- a) Nombre de la función: tspOnlyneDistances
- b) Matriz de distancias de tamaño $n*n$: distances [n,n]
- c) Vector de nodos por visitar: nodes_by_visit[n]
- d) Vector del conjunto total de nodos: nodes[n]
- e) Vector con la primera solución óptima: optimal_rute[n]
- f) Costo total de la primera solución: total_distances

El siguiente paso será definir el punto de partida, luego se evaluará cual es el siguiente nodo a visitar en función a la distancia más cercana, tomando en cuenta no volver a un nodo ya visitado. El proceso se repetirá hasta visitar todos los nodos o se cumpla con el límite de distancias y/o costos.

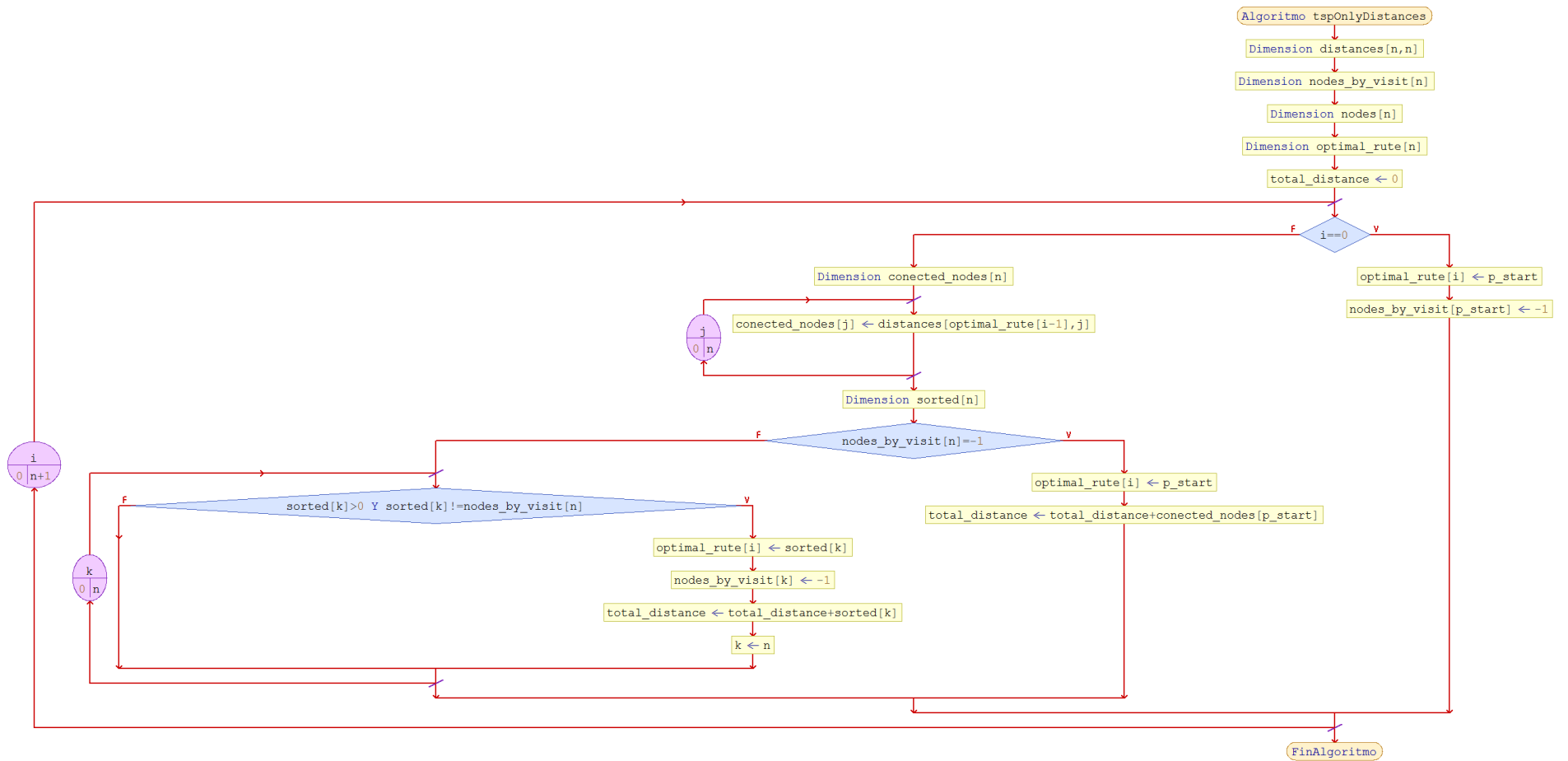


Figura 8. Diagrama de flujo del método de construcción del vecino más cercano.

Fuente: Elaboración propia, 2017.

3.3.3 Diseño e implementación de la heurística de mejoría.

En general las soluciones que se obtienen de los métodos constructivos suelen ser de calidad moderada y demandan un tiempo computacional elevado (Martí, 2013, pág. 26); por tal motivo surge la necesidad de implementar heurísticas de mejoría.

El método que se utilizó es denominado “Procedimiento de 2 intercambio o 2 óptimo (2 - opt)” y consiste en eliminar dos aristas (nodos conectados) y reconectar los dos caminos resultantes de una manera diferente para obtener un nuevo ciclo (ruta turística). Las figuras 9 y 10 ilustran este movimiento en donde las aristas (i, j) y (l, k) son reemplazadas por (l, j) y (i, k) . Solo existe una manera de reconectar los dos caminos y formar un único circuito turístico.

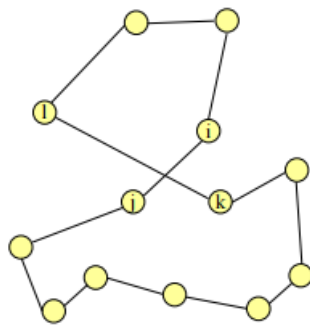


Figura 9. Solución inicial.

Fuente: Procedimientos Metaheurísticos en Optimización Combinatoria, 2010.

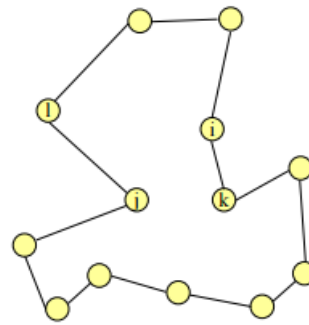


Figura 10. Solución mejorada.

Fuente: Procedimientos Metaheurísticos en Optimización Combinatoria, 2010.

La codificación estructurada de este método está encapsulada en una función recibiendo como argumentos de entrada los valores de la primera solución obtenida en el algoritmo de construcción y son: valor de la primera solución, valor de la función objetivo, matriz de distancias y costos. El algoritmo evaluará los mejores movimientos posibles guardando la mejor solución en cada iteración, esta reemplazará al valor de la primera solución y la función objetivo en la siguiente iteración; es decir adopta la técnica de IA “primero el mejor”, la codificación se presenta seguidamente en la Figura 11.

```

def improvement(optimal_route, total_distance, distances):
    best = total_distance
    optimal_route_1 = optimal_route[:]
    tam = len(optimal_route_1)
    opc = 0
    while opc != 1:
        routes = []
        # recorre todos los nodos que pueden realizar un movimiento
        for i in range(tam - 3):
            # se obtiene la distancia de la primera arista
            d1 = distances[optimal_route_1[i]][optimal_route_1[i + 1]]
            for j in range(i + 2, tam - 1):
                ruta_ = optimal_route_1[:]
                # se obtiene la distancia de la primera nueva arista
                d1_ = distances[optimal_route_1[i]][optimal_route_1[j]]
                # se obtiene la distancia de la segunda arista
                d2 = distances[optimal_route_1[j]][optimal_route_1[j + 1]]
                temp = best - (d1 + d2)
                # se obtiene la distancia de la segunda nueva arista
                d2_ = distances[optimal_route_1[i + 1]][optimal_route_1[j + 1]]
                best_ = temp + (d1_ + d2_)
                # verifica si se minimizó la distancia total
                if best_ < best:
                    # se realiza el movimiento
                    ruta_[i + 1] = optimal_route_1[j]
                    ruta_[j] = optimal_route_1[i + 1]
                    routes.append((best_, ruta_))
                    m = i + 2
                    k = j - 1
                    # Redireccionar los nodos intermedarios
                    while k > i + 1:
                        ruta_[k] = optimal_route_1[m]
                        k -= 1
                        m += 1

        if (routes):
            # Actualizar la nueva mejor ruta
            sorted_routes = sorted(routes)
            optimal_route_1 = sorted_routes[0][1]
            best = sorted_routes[0][0]
        # Salir del bucle si no existen mejores resultados
        if len(routes) == 0:
            opc = 1
    return optimal_route_1, best

```

Figura 11. Codificación del algoritmo heurístico de mejoría 2-óptimo.

Fuente: Elaboración propia, 2017.

3.3.4 Diseño e implementación de la metaheurística Búsqueda Tabú.

TS está basada en principios generales de la IA y puede ser utilizado para guiar cualquier procedimiento de búsqueda local en la búsqueda agresiva del óptimo del problema. Por agresiva se refiere a la estrategia de evitar que la búsqueda quede atrapada en un óptimo local que no sea global. A tal efecto, TS toma de la IA el concepto de memoria y lo implementa mediante estructuras simples con el objetivo de dirigir la búsqueda teniendo en cuenta la historia de ésta. Es decir, el procedimiento trata de extraer información de lo sucedido y actuar en consecuencia. En este sentido puede decirse que hay cierto aprendizaje y que la búsqueda es inteligente (Cobos, 2004, p. 21). Por lo referido anteriormente TS ayudará a mejorar la solución que se obtendrá de la heurística de mejoría (solución local).

Las estructuras de memoria implementadas en esta investigación fueron de dos dimensiones: hechos recientes y frecuencia (memoria de largo plazo).

3.3.4.1 Estructura de datos.

Las distancias y costos entre nodos se representan en una estructura de una matriz simétrica como se muestra en la Tabla 3, los datos artificiales (instancias) y los datos recopilados del Distrito de Juli serán estructurados de esta manera. Las instancias originales tienen un formato similar, separadas por espacios en blanco y almacenadas en un texto plano, mientras que los datos de Juli se almacenaron relacionamente en una Base de Datos MySQL.

Tabla 3

Matriz de distancias en km. de 6 recursos turísticos.

	(0)	(1)	(2)	(3)	(4)	(5)
(0)	0	24	33	24	16	71
(1)	24	0	124	21	52	28
(2)	33	124	0	333	92	143
(3)	24	21	333	0	63	323
(4)	16	52	92	63	0	53
(5)	71	28	143	323	53	0

Fuente: Elaboración propia, 2017.

3.3.4.2 Entorno de intercambio.

En las Figuras 12 y 13 se visualiza un ejemplo de la primera iteración con intercambio de dos recursos turísticos. El intercambio forma parte de un conjunto de posibles intercambios por pares, a este conjunto se le denomina “entorno de intercambio” en donde se asocia a cada par de intercambio, el valor de la función objetivo y el valor del movimiento, como se visualiza en la Tabla 4.

Tabla 4

Entorno de intercambio para la primera iteración.

Recursos turísticos		Valor de la función objetivo	Valor del movimiento
<i>I</i>	<i>j</i>		
1	4	365	-303
0	5	431	-237
1	4	561	-107
2	5	563	-105
0	3	594	-74
1	3	657	-11
4	5	873	205

Fuente: Elaboración propia, 2017.

[0, 1, 2, 3, 4, 5, 0]

Figura 12. Permutación inicial.

Fuente: Elaboración propia, 2017.

[0, 4, 2, 3, 1, 5, 0]

Figura 13. Intercambio de los recursos 1 y 4.

Fuente: Elaboración propia, 2017.

El ordenamiento de la Figura 12 especifica la ruta turística inicial en donde el punto de partida es el valor 0, se aprecia que el valor inicial es igual al valor final puesto que, en el problema original refiere que el viajante vuelva al punto inicial; esta restricción puede ser ignorada tomando en cuenta la variable limitante de tiempo y/o costo.

El mejor movimiento de la tabla 4 resulta del intercambio de los recursos 1 y 4 (Figura 13), cuyo valor de movimiento minimiza el valor de la función objetivo en 303 unidades.

3.3.4.3 Memoria de corto plazo.

La memoria de corto plazo lleva la cuenta de las soluciones recientes visitadas y su objetivo es explorar a fondo una región dada del espacio de soluciones.

Para evitar la búsqueda desde combinaciones de intercambio repetidas usadas en el pasado reciente, invirtiendo potencialmente los efectos de movimientos anteriores que podrían devolver a soluciones previas, se clasifica como tabú todos los intercambios compuestos por recursos turísticos recientes.

El objetivo principal de etiquetar soluciones visitadas como tabú es evitar que la búsqueda se cicle. Por ello se considera que tras un cierto número de iteraciones la búsqueda está en una región distinta y puede liberarse del estatus tabú a las soluciones antiguas. De esta forma se reduce el esfuerzo computacional de calcular el entorno reducido en cada iteración.

Para este proceso se creó una lista tabú programada en una matriz de $n \times n$ siendo n la cantidad total de recursos turísticos a visitar. En la Tabla 5 se muestra el estado final de la memoria de corto plazo del ejemplo anterior.

La duración tabú de cada movimiento se calcula mediante la fórmula: $\left\lfloor \frac{n}{2}, \frac{n}{4} \right\rfloor$ (Maquera, 2017).

Tabla 5

Matriz tabú en una iteración dada.

	(0)	(1)	(2)	(3)	(4)	(5)
(0)	0	0	0	0	0	0
(1)	0	0	0	0	0	0
(2)	0	0	0	0	0	0
(3)	0	0	0	0	0	0
(4)	0	4	0	0	0	0
(5)	0	0	4	0	0	0

Fuente: Elaboración propia, 2017.

En cada iteración los movimientos con estado tabú disminuyen en 1, de modo que en el futuro vuelvan a considerarse en nuevas soluciones. En la tabla 5 las columnas representan los nodos y las filas las posiciones; identificamos que los recursos 1 y 2 en las posiciones 4 y 5 respectivamente no podrán ser consideradas en 4 movimientos.

3.3.4.4 Criterio de aspiración.

Los niveles de aspiración o criterios de aspiración, son introducidos para determinar cuándo pueden ser ignoradas las reglas de activación tabú. En el momento que los nodos con estado tabú ofrezcan mejorar el valor de la función objetivo actual, podrán ser considerados en el intercambio, de esta forma se introduce cierta flexibilidad en la búsqueda y se mantiene su carácter agresivo. Esta funcionalidad codificada se presenta en las líneas de código 151 – 153 de la Figura 14.

```

150     for j in exchange_environment:
151         if exchange_environment[1] < best_solution[1]:
152             short_memory[j[2]][j[3]] = 0
153             short_memory[j[3]][j[2]] = 0
154         if short_memory[j[2]][j[3]] == 0:

```

Figura 14. Criterio de aspiración.

Fuente: Elaboración propia, 2017.

3.3.4.5 Memoria de largo plazo.

En esta memoria se registra la frecuencia de los movimientos, complementando la información proporcionada por la memoria basada en lo reciente, ampliando la base para

seleccionar movimientos aún no explorados. La estructura es de tipo matriz de $n * n$ y aún inicio comenzará con un valor de 0 en cada casilla, esta matriz se presenta en la Tabla 6.

Tabla 6

Matriz de frecuencia al final de una ejecución.

	(0)	(1)	(2)	(3)	(4)	(5)
(0)	2	0	0	0	0	0
(1)	0	4	6	0	3	0
(2)	0	0	0	0	0	0
(3)	3	0	0	0	0	4
(4)	0	6	2	0	3	0
(5)	0	0	4	0	0	0

Fuente: Elaboración propia, 2017.

La Tabla 6 también muestra todos los movimientos realizados en la ejecución del algoritmo con 6 instancias, y se identifica que el nodo 1 y 2 en la posición 4 y 1 respectivamente fueron considerados 6 veces en total, esta información ayudará a determinar una estrategia en un movimiento futuro.

3.3.4.6 Diversificación.

La diversificación ayudará a buscar soluciones a un no exploradas, el algoritmo llamará a este procedimiento en número de iteración definida por $\frac{n}{3}$ en cuanto no se haya actualizado aún el valor de la función objetivo.

CAPÍTULO IV

Validación del algoritmo en el proyecto UPeU-CONCYTEC

En este capítulo se describe la aplicación y validación del algoritmo en un módulo del proyecto UPeU-CONCYTEC. Para la validación se desarrolló una aplicación web que forma parte de la Plataforma Digital Inteligente y Big Data para el Turismo Rural Comunitario en la Región Puno (patrocinado por CONCYTEC y la UPeU), denominado “Caminos del Inkari”.

4.1. Acerca del proyecto UPeU-CONCYTEC.

El proyecto UPeU-CONCYTEC 2015-2018, es un proyecto de investigación presentado por la Dra. Sc. Nélide Gladys Maquera Sosa y la UPeU (Universidad Peruana Unión) en el año 2015 al CONCYTEC denominado “Plataforma Digital Inteligente y Big Data para el Turismo Rural Comunitario en la Región Puno” con Resolución de Dirección Ejecutiva N° 144-2015-FONDECYT-DE, financiado por el Consejo Nacional de Ciencia, Tecnología e Innovación del Perú (CONCYTEC) y por la Universidad Peruana Unión (UPeU). Y uno de sus objetivos es aplicar las estrategias de Inteligencia Artificial y Big Data a la cadena productiva del Turismo Rural Comunitario (TRC) en la Región Puno.

4.2. Desarrollo de la aplicación web.

La Metaheurística diseñada e implementada será el motor (cerebro) de la aplicación web a desarrollarse, la aplicación será un entorno que sea capaz de soportar este motor adaptándose a sus funcionalidades, y como producto final será capaz de ofrecer una interacción más eficiente e intuitiva al usuario (turista).

4.2.1 Casos de uso.

Como parte del análisis se diseñaron dos casos de uso validados por los actores directos del TRC en la región Puno. Fue necesario realizar visitas a estos actores para definir los requerimientos funcionales y no funcionales de la aplicación web y por ende del algoritmo.

4.2.1.1 Caso de uso del registro de emprendimientos

Este caso de uso permitirá a los actores involucrados en el TRC en la Región Puno en especial a los emprendedores, registrar la información de sus emprendimientos turísticos (Hospedajes, Artesanía, Alimentación, Guía, entre otros) y los atractivos turísticos cercanos que aún no se encuentren registrados en la plataforma. De esta manera se alimentará la Base de Datos para generar la ruta inteligente en más distritos y provincias. El diagrama del caso de uso de presenta en la Figura 15.

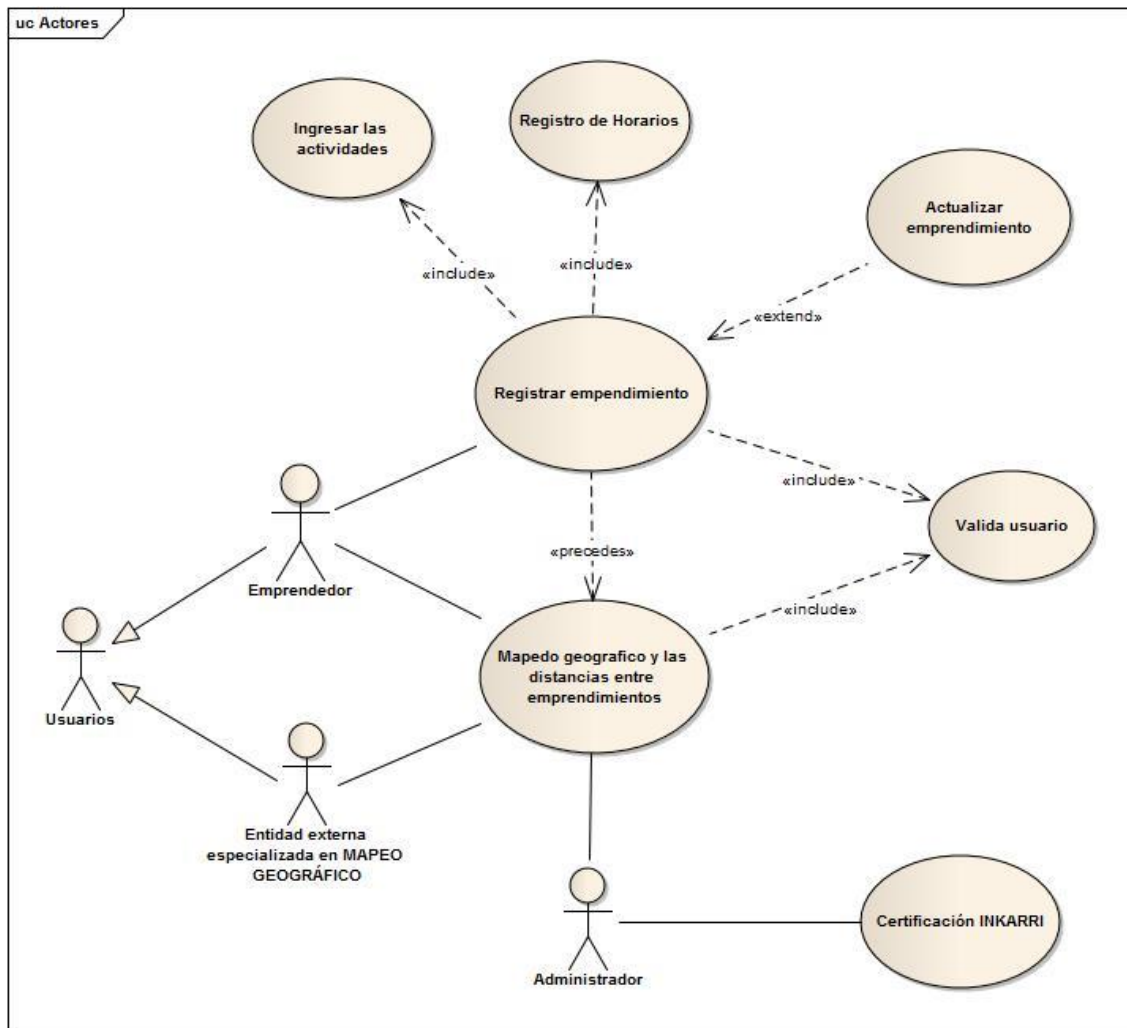


Figura 15. Diagrama de caso de uso del registro de emprendimientos.

Fuente: Elaboración propia, 2017.

4.2.1.1.1 Actores

- Emprendedor: La persona propietaria de uno o más emprendimientos.
- Entidad Externa: Cualquier tipo de empresa calificada para ingresar datos específicos.
- Administrador: Persona responsable, y con accesos privilegiados al sistema.
- Usuarios: Cualquier Persona externas al sistema que utilicen las distintas funcionalidades de la misma.

4.2.1.1.2 Flujo Básico

- El emprendedor se autenticará en el sistema.
- El sistema mostrará una vista con un formulario para registrar su emprendimiento.
- El emprendedor ingresará la información requerida en el formulario.
- Si el registro se completó y el sistema no muestra ninguna notificación, se podrá continuar pulsando el botón guardar.

- e) El sistema registrará toda la información en un servidor de Base de Datos desplegada en la nube, luego mostrará la información de la misma al emprendedor.
- f) El emprendedor podrá actualizar la información de forma remota, si hubiera algún cambio en su emprendimiento esta información se debe reflejar en el sistema.
- g) El emprendedor y/o una entidad externa y/o administrador podrán registrar los datos mapeados geográficamente y las distancias entre los emprendimientos y atractivos turísticos.
- h) El sistema solicitará seleccionar los emprendimientos a mapear.
- i) El sistema muestra un formulario para registrar la información.
- j) El sistema verificará toda la información enviada.
- k) Un Administrados tomará la decisión de certificar el emprendimiento según determinen los resultados de la verificación.

4.2.1.1.3 Prototipos

Fue necesario elaborar prototipos para validar las funcionalidades, diseño y maquetado de la aplicación web, y son los siguientes:

- Registro de Horarios y Estado del emprendimiento, se presenta en la Figura 16.

Figura 16. Prototipo del registro de emprendimientos.

Fuente: Elaboración propia, 2017.

- Registro de actividades, se presenta en la Figura 17.

El prototipo muestra una interfaz de usuario en un navegador web. La barra de direcciones muestra 'http://www.caminosdelinkarri.com'. En la parte superior derecha, hay un ícono de usuario y el nombre 'Alberto Molinedo'. El título principal es 'Taypi Qamaña Establo'. Dentro de un recuadro, hay un interruptor de 'Paseo en bicicleta' que está activado. Debajo, hay tres campos de entrada etiquetados 'Tipo', 'Duración' y 'Costo'. Un recuadro 'Horario' contiene tres campos para 'Mañana', 'Tarde' y 'Noche'. Al final, hay un campo de texto 'Descripción'.

Figura 17. Prototipo del registro de actividades del emprendimiento.

Fuente: Elaboración propia, 2017.

4.2.1.2 Caso de uso para generar la ruta inteligente

Este caso de uso permitirá a los visitantes planificar un viaje a medida generando una ruta turística que se adapte a sus necesidades, gustos y preferencias. El diagrama del caso de uso se presenta a continuación en la Figura 18:

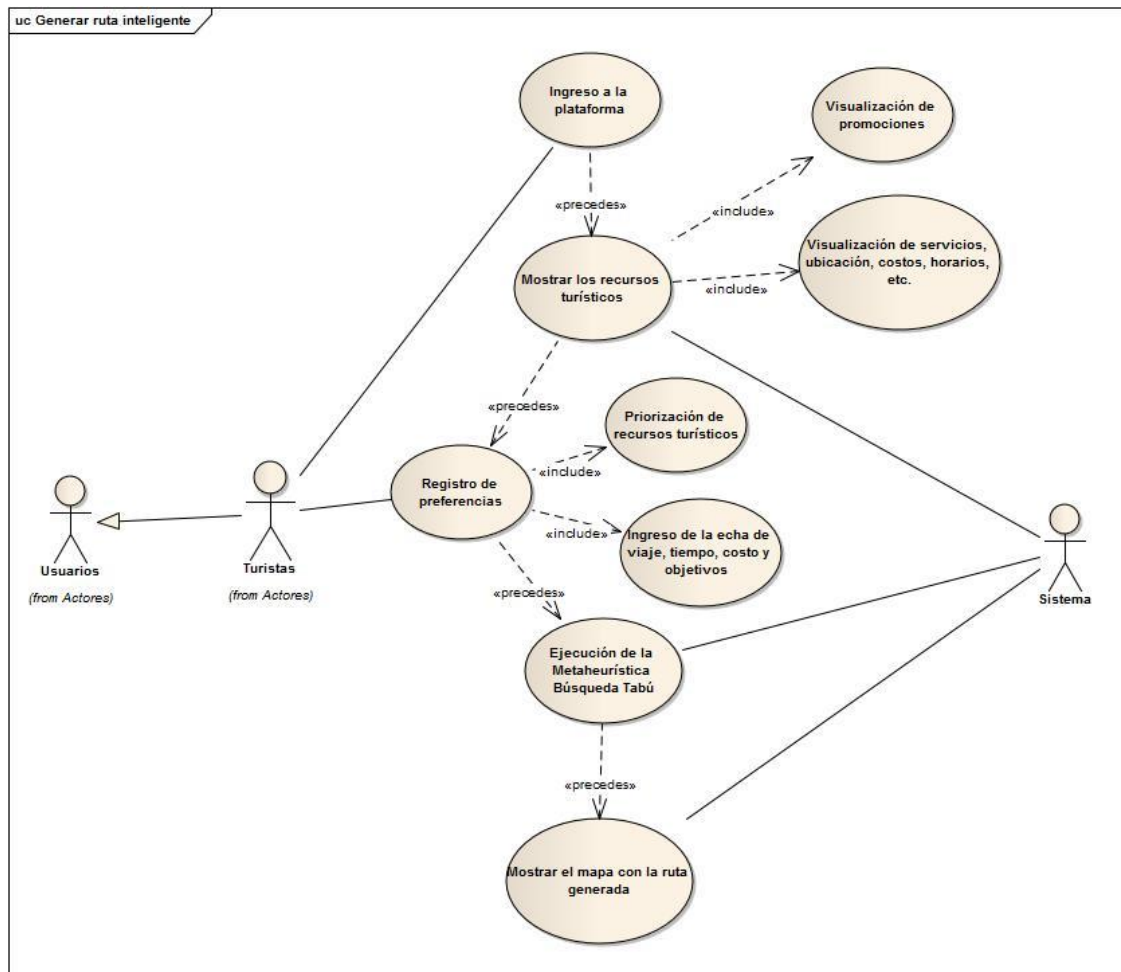


Figura 18. Diagrama de caso de uso de la ruta inteligente.

Fuente: Elaboración propia, 2017.

4.2.1.2.1 Actores

- Turista: El visitante que desea una ruta turística personalizada.
- Sistema: Funciones automatizadas de la plataforma.
- Usuarios: Cualquier Persona externas al sistema que utilicen las distintas funcionalidades de la misma.

4.2.1.2.2 Flujo Básico

- El turista se dirige al sitio web: www.turismoruralcomunitario.pe.
- El sistema muestra la lista de los recursos turísticos con información detallada.
- El turista pulsa el botón “Diseñar mi viaje”.
- El sistema pedirá el destino de viaje, mostrando una lista de destinos almacenados en su Base de Datos.
- El turista seleccionará un destino.
- El Sistema cargará un conjunto de recursos turísticos del destino seleccionado.

- g) El turista puede seleccionar los recursos que desea visitar, priorizando los que le demanden más interés. También ingresara las preferencias como tiempo y dinero.
- h) El Sistema ejecuta la metaheurística Búsqueda Tabú procesando la información enviada por el turista.
- i) El sistema carga una vista con un mapa y la ruta generada.

4.2.1.2.3 Prototipos

- Seleccionar los recursos turísticos preferidos, se muestra en la Figura 19.

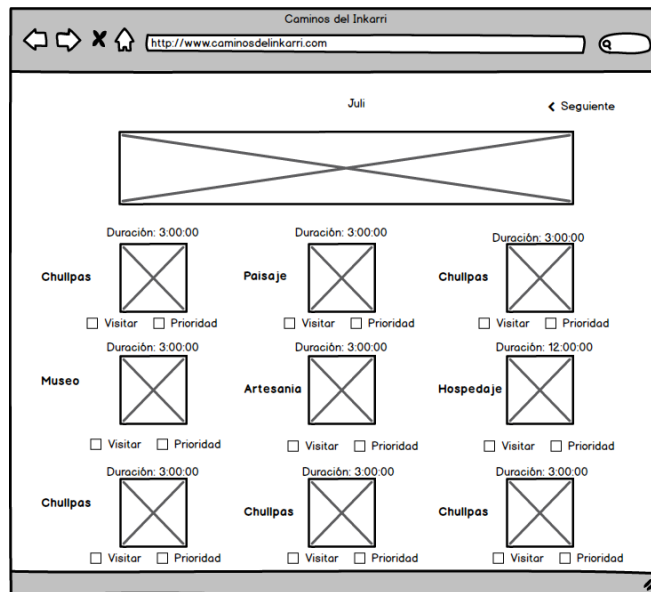


Figura 19. Prototipo para seleccionar los recursos que se desea visitar.
Fuente: Elaboración propia, 2017.

- Ingreso de las preferencias del turista, se presenta en la Figura 20.

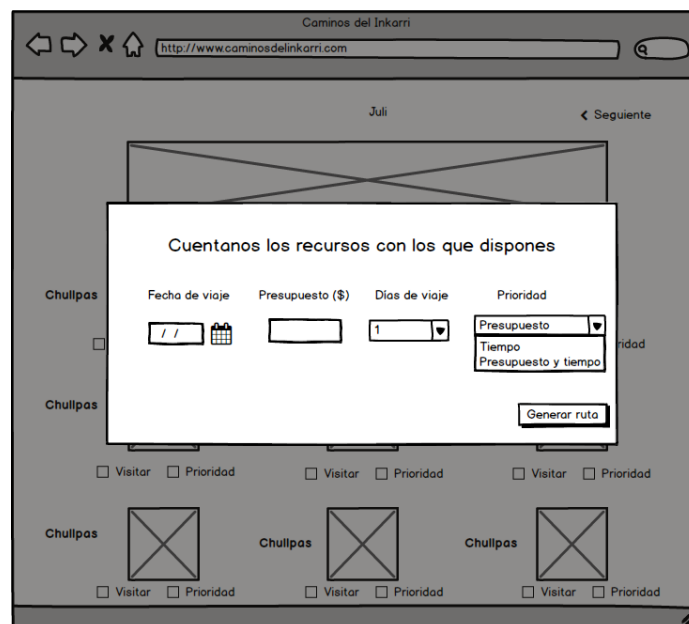


Figura 20. Prototipo para registrar las preferencias del turista.
Fuente: Elaboración propia, 2017.

- Ruta inteligente generada, se presenta en la Figura 21.

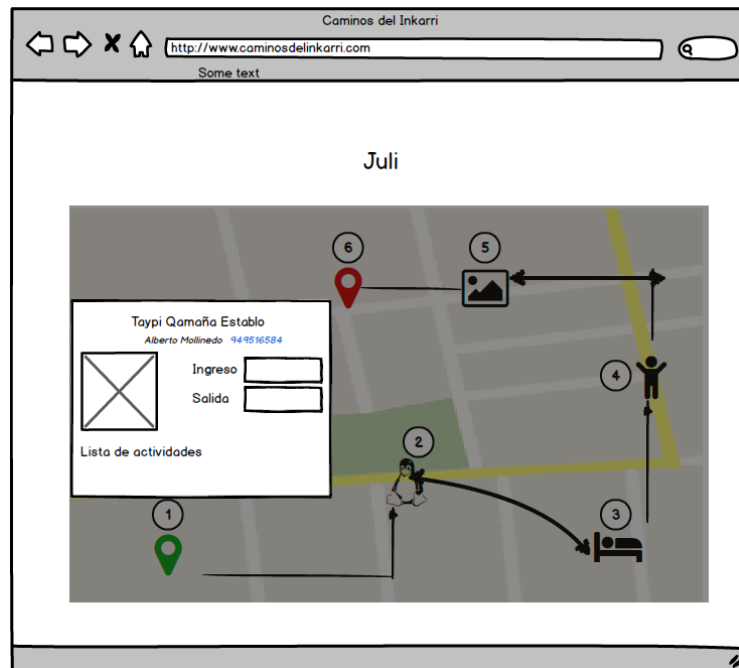


Figura 21. Prototipo de la ruta propuesta por el algoritmo.

Fuente: Elaboración propia, 2017.

4.2.2 Programación.

La metaheurística fue programada en una función que recibe como argumentos de entrada los valores obtenidos en la heurística de mejoría, estos son: lista de recursos (ruta), valor de la función objetivo y las distancias en forma de matriz, retornando los valores óptimos globales de la solución y el valor de la función objetivo.

Con el fin de optimizar el código se programó funciones independientes que alimentan a la función principal, la cual se muestra en la Figura 22.

```

def tabuSearch(route, value_fo, distances):
    route_size = len(route)
    new_route = route[:]
    new_value_fo = value_fo
    short_memory = np.zeros((route_size - 1, route_size - 1), dtype=np.int)
    large_memory = np.zeros((route_size - 1, route_size - 1), dtype=np.int)
    best_solution = [route, value_fo]
    count = 0
    while count < route_size:
        exchange_environment = sorted(getSwaps(new_route, new_value_fo, distances))
        for j in exchange_environment:
            if exchange_environment[j][1] < best_solution[1]:
                short_memory[j[2]][j[3]] = 0
                short_memory[j[3]][j[2]] = 0
                if short_memory[j[2]][j[3]] == 0:
                    new_route[j[4]] = j[3]
                    new_route[j[5]] = j[2]
                    updateShortMemory(short_memory)
                    duration_tabu = randint(route_size / 4, route_size / 2)
                    short_memory[j[2]][j[3]] = duration_tabu
                    short_memory[j[3]][j[2]] = duration_tabu
                    large_memory[j[2]][j[3]] += 1
                    large_memory[j[3]][j[2]] += 1
                    new_value_fo = j[1]
                if new_value_fo < best_solution[1]:
                    optimal_route = new_route[:]
                    best_solution = [optimal_route, new_value_fo]
                break
        count += 1
    return best_solution

```

Figura 22. Codificación de la metaheurística Búsqueda Tabú.

Fuente: Elaboración propia, 2017.

La aplicación web desarrollada que integra las dos heurísticas y la metaheurística se muestra en la Figura 23.

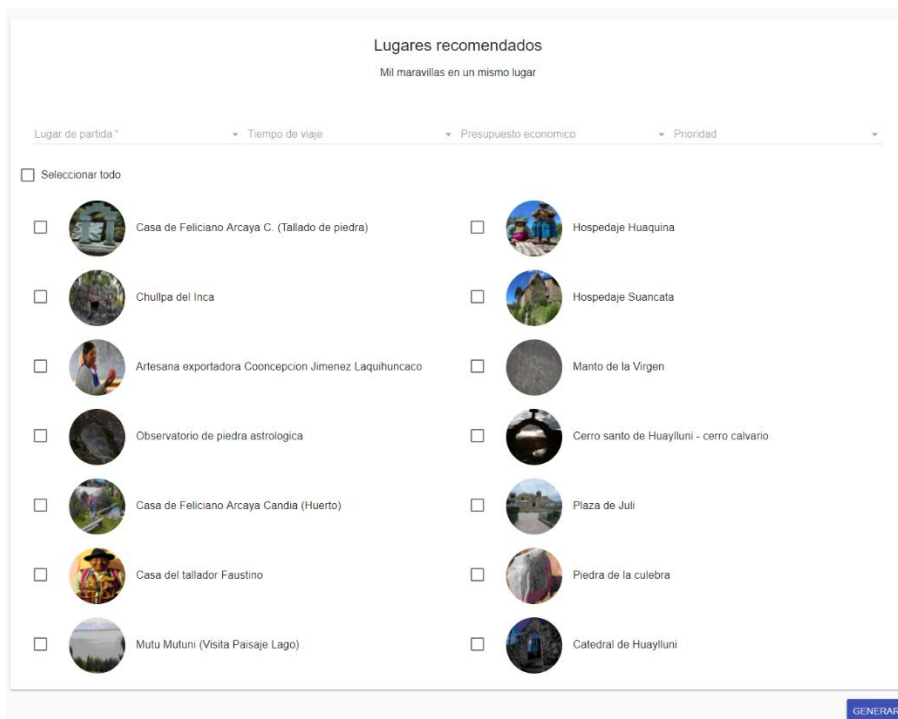


Figura 23. Aplicación web puesta en producción.

Fuente: Elaboración propia, 2017.

4.3. Validación de la metaheurística con datos del Distrito de Juli

Los resultados del algoritmo al procesar instancias artificiales serían suficientes para validarlo, pues es así como muchos investigadores de estos temas presentan y someten a juicio sus trabajos de investigación. Pero en el caso particular de este proyecto de tesis, se consideró la necesidad de desarrollar una aplicación web que integre el algoritmo y se adapte a las necesidades del turista y los actores del TRC de la Región Puno, siendo estos últimos los que validaron la aplicación web y que también fueron capacitados por el investigador, como se muestra en el Anexo E.

La información recopilada de los recursos turísticos y emprendimientos del Distrito de Juli se presentan en el Anexo A, esta información fue registrada en una base de datos relacional MySQL. Para la ejecución del algoritmo los datos de las distancias, costos y tiempos se extrajeron en una variable de tipo arreglo (matriz). En la Figura 26 se aprecia la ejecución de la aplicación web, en donde el algoritmo traza un circuito turístico formando un ciclo hamiltoniano y volviendo al lugar de partida, como estipula el TSP.

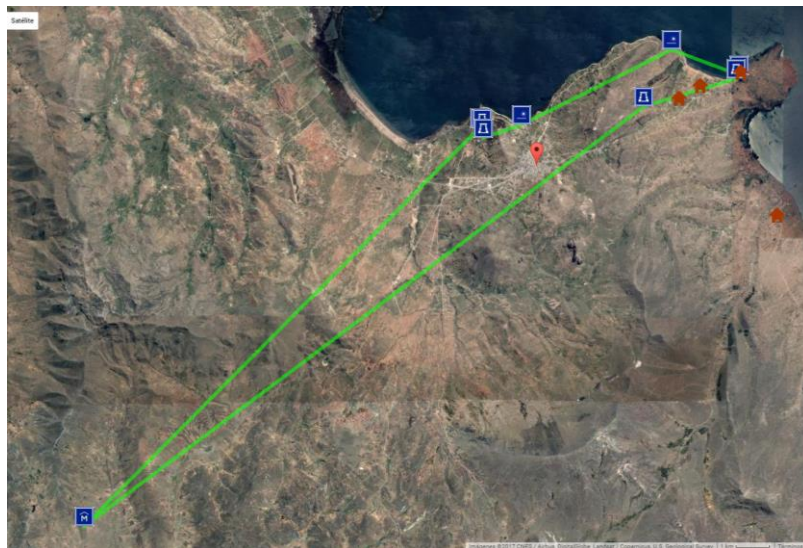


Figura 24. Validación del algoritmo con instancias reales.

Fuente: Elaboración propia, 2017.

En la Figura 27 se presenta un caso de la validación, en donde el turista tiene un presupuesto de S/ 100 soles, y desea realizar una visita a Juli. El algoritmo le diseña un recorrido con los lugares que podrá visitar contemplando su presupuesto y en una duración de un día. El circuito propuesto comenzará y terminará en el mirador natural a orillas del Lago Titicaca, que lleva en nombre de “Chinchalavi”.

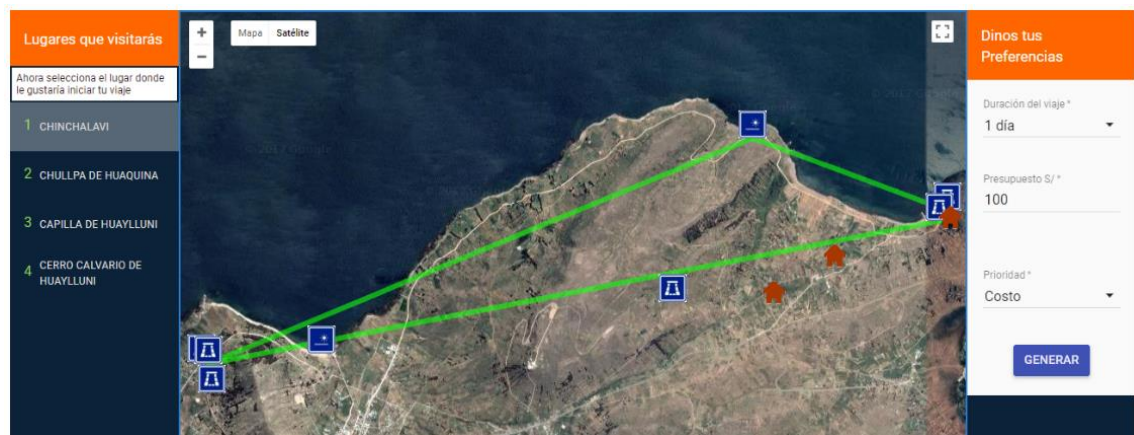


Figura 25. Circuito personalizado para un día de viaje y un presupuesto de 100 soles.

Fuente: Elaboración propia, 2017.

Las preferencias pueden ser editadas, en el caso de la prioridad el turista podrá elegir entre costo, distancia y tiempo; dependiendo de sus necesidades, preferencias y gustos.

CAPÍTULO V

Resultados y discusión

En este capítulo se describe la validación de la metaheurística procesando instancias utilizadas en el ámbito científico para validar este tipo de problemas combinatorios. También se utilizó la información recopilada de recursos turísticos y emprendimientos del distrito de Juli.

5.1. Resultados

5.1.1 Validación de la metaheurística con instancias artificiales

Las instancias artificiales que se utilizaron, vienen en librerías, son de dominio público y muchos investigadores lo usan para validar sus métodos, La librería también contienen la mejor solución obtenida hasta la fecha y/o la óptima, el tiempo de ejecución, características del computador en donde se ejecutó al algoritmo y las distancias en formato de una matriz simétrica o coordenadas.

Uno de los métodos para validar nuevas heurísticas y metaheurísticas con dichas instancias, consiste en comparar, la nueva solución, con la mejor registrada hasta el momento. Esta comparación se realiza mediante la desviación porcentual, en donde: se define C_h al coste de la solución del nuevo algoritmo y C_{opt} al coste de la solución óptima, en el problema de minimización la desviación porcentual viene dada por la expresión: $\frac{C_h - C_{opt}}{C_{opt}} \cdot 100$ (Marti, 2010, pág 13).

El objetivo de la validación, es que, al aplicar la desviación porcentual al costo de las soluciones obtenidas por los tres métodos, el resultado sea igual a cero, igualando de esta manera a la mejor solución conocida hasta la fecha.

Los resultados obtenidos procesando 5 grupos de instancias artificiales se muestran en la Tabla 7 presentada a continuación.

Tabla 7

Distancia porcentual de los resultados obtenidos.

Instancias	Nro. ciudades	C_{opt}	C_h			Desviación porcentual		
			C	M	TS	C	M	TS
FRI26	26	937	1112	955	937	18.68	1.92	0
dantzig42	42	699	956	731	699	36.77	4.58	0
GR17	17	2085	2187	2085	2085	4.89	0	0
Western_Sahara	29	27603	36388	28031	27750	31.83	1.55	0.53
Qatar	194	9352	11640	9901	9894	24.47	5.87	5.80

Fuente: Elaboración propia, 2017.

La Tabla 7 muestra los nombres de las instancias con su número de nodos y su solución óptima hasta la fecha. En las siguientes columnas se describen el costo de las soluciones obtenidas mediante la Heurística de Construcción (C), la Heurística de Mejoría (M) y la Metaheurística Búsqueda Tabú (TS) diseñadas e implementas en esta investigación. Finalmente se observa la desviación porcentual del costo de las tres soluciones obtenidas mediante los tres métodos.

Mediante la heurística de construcción, no se pudo igualar a la mejor solución en todas las instancias. Mediante la heurística de mejoría en la instancia “GR17” se logró igualar a la mejor solución y se obtuvo mejoras en las otras 4 instancias restantes.

Mediante la metaheurística Búsqueda Tabú, se logró igualar a la mejor solución en las instancias “FRI26”, “GR17”, “dantzig42” y en la instancia “Western_Sahara” la desviación porcentual llego al valor de 0.53, mientras que en la instancia “Qatar” la desviación porcentual disminuyó a 5.80. En la Figura 24 se presenta un gráfico que contrasta la desviación porcentual de los tres métodos.

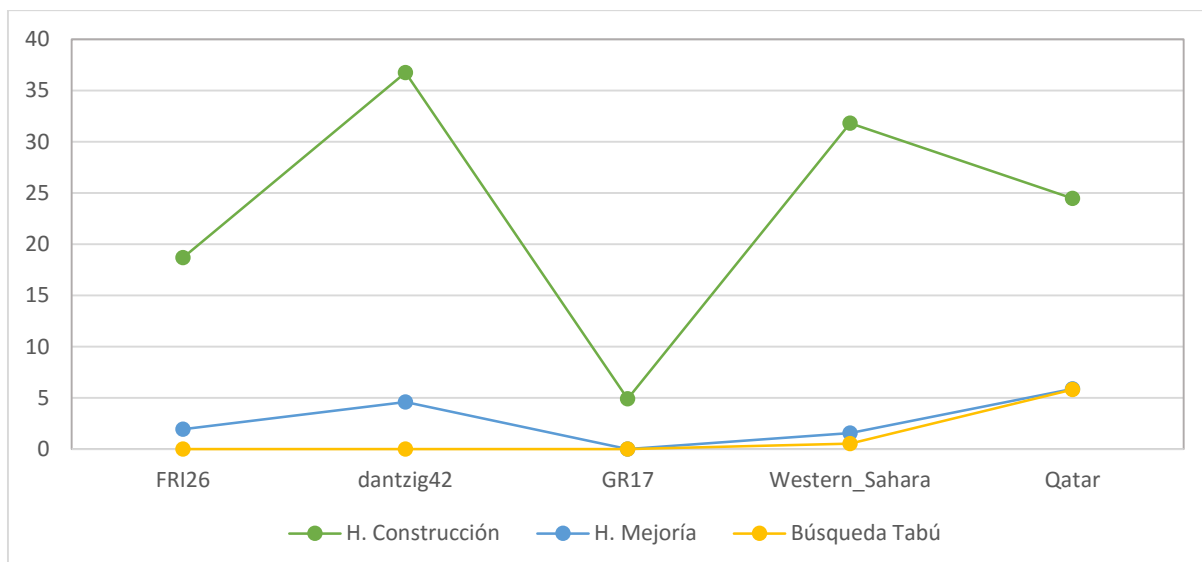


Figura 26. Eficiencia de los tres métodos.

Fuente: Elaboración propia, 2017.

La diferencia es notoria entre las soluciones obtenidas mediante la Heurística de construcción y la de Mejoría; esto demuestra la importancia de la implementación de la heurística de mejoría. Y con la implementación de TS se logró mejorar las dos soluciones anteriores, validando su eficiencia en la mejora de soluciones locales.

En la tabla 8 se aprecia los tiempos de ejecución de las mejores soluciones de las instancias en comparación con los tiempos de ejecución de TS, las características de la computadora utilizada para procesar la metaheurística se describen en la sección de materiales de esta investigación.

Tabla 8

Tiempos de ejecución de la metaheurística.

Instancias	Mejor tiempo de ejecución (Segundos)	Tiempo de ejecución de TS (Segundos)
FRI26	0.093	0.007
dantzig42	0.23	0.031
GR17	0.08	0.002
Western_Sahara	0.09	0.01
Qatar	2.09	3.68

Fuente: Elaboración propia, 2017.

El espacio de soluciones en 42 iteraciones procesando la instancia “dantsig42” se presenta en la Figura 25, identificamos como TS adopta soluciones que empeoran el valor de la función objetivo (731) ampliando el espacio de búsqueda de local a global. Se puede observar también que el algoritmo logra igualar a la mejor solución (699) en la iteración 10.

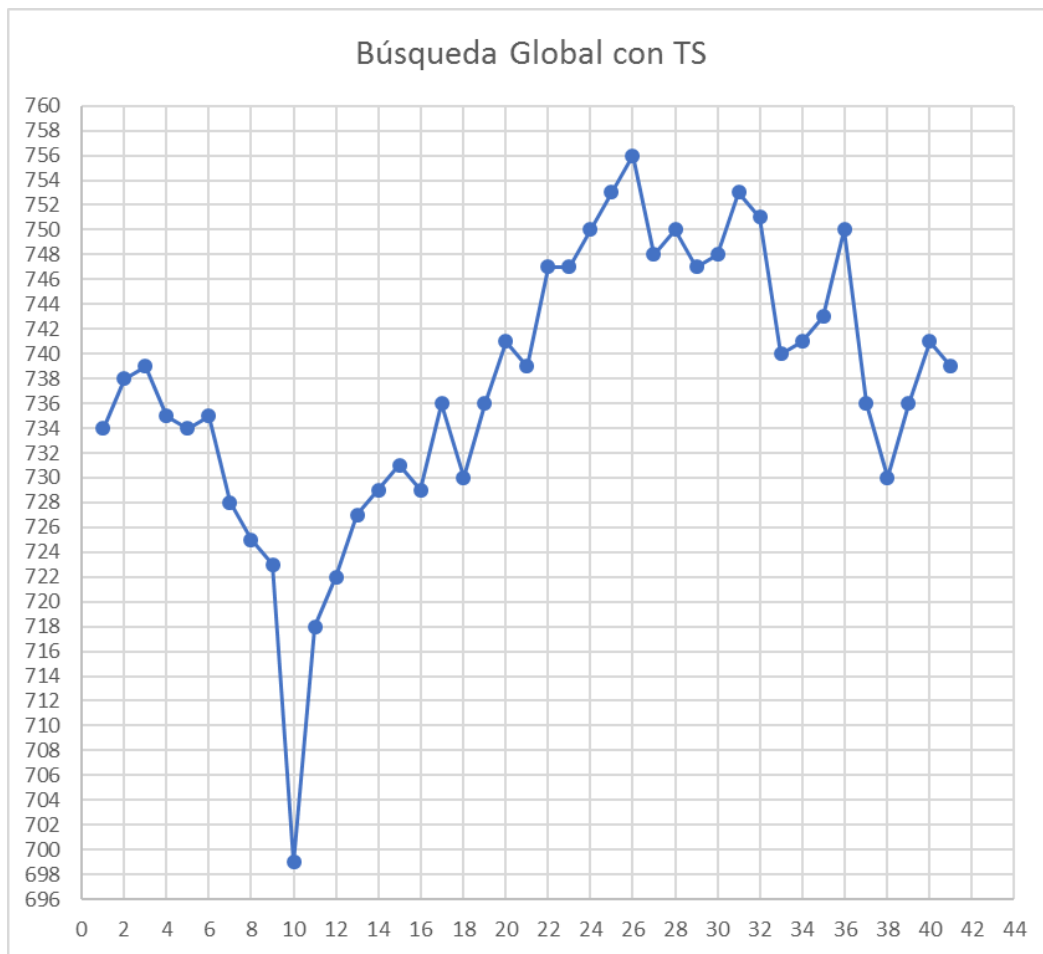


Figura 27. Búsqueda global procesando la instancia dantzig42 mediante TS.

Fuente: Elaboración propia, 2017.

5.2. Discusión

Una de las contribuciones de importancia en esta investigación es la aplicación de tecnologías de Inteligencia Artificial e Investigación de Operaciones al Turismo Rural Comunitario en la Región Puno. Sin embargo, uno de los inconvenientes es el acceso a la información por parte de los investigadores; por ejemplo, el inventario de los recursos turísticos es responsabilidad del MINCETUR y Gobiernos locales y regionales, pero no tienen una Base de Datos completa ni actualizada.

En la validación de la aplicación web los emprendedores quedaron satisfechos pues los resultados coincidían con la información que ellos tienen, pues son ellos quienes viven cerca a los recursos turísticos.

La metodología usada para el diseño e implementación de la metaheurística también está descrita en el artículo de Fred Glover publicado en 1986, quien fue el primero en introducir el término Búsqueda Tabú y metaheurística. (Hillier, 2010). Con esto se asegura que la metodología usada está validada por su mismo autor.

En los resultados no se pudo igualar a todas las mejores soluciones, esto se debe a que, en muchos casos, estas soluciones fueron obtenidas mediante métodos exactos y en otros casos las Heurísticas usadas incorporaban más estrategias especializadas que demandan estudios avanzados en el área de Ciencias de la Computación.

Esta investigación está focalizada en la Región Puno, sin embargo, el algoritmo puede procesar información de otras regiones del Perú.

CAPÍTULO VI

Conclusiones y recomendaciones

6.1. Conclusiones

Conforme a los datos y resultados presentados en esta investigación de diseño e implementación de un circuito turístico inteligente en la región puno mediante la metaheurística búsqueda tabú, se llegó a las siguientes conclusiones:

Respecto al objetivo general, se diseñó e implementó un circuito inteligente mediante la metaheurística Búsqueda Tabú, validado con instancias artificiales e información real del Distrito de Juli, en donde los emprendedores y diferentes actores validaron satisfactoriamente los resultados finales.

En cuanto al primer objetivo específico se implementó la heurística de Construcción del vecino más cercano, la cual obtuvo las primeras soluciones que alimentaron a la heurística de mejoría.

Referente al segundo objetivo específico se diseñó e implementó la heurística de mejoría con el método del intercambio de dos aristas (2 opt), esta heurística mejoró las soluciones obtenidas con la heurística de construcción.

Con relación al tercer objetivo específico, se implementó la metaheurística Búsqueda Tabú con la estrategia de diversificación.

En cuanto al cuarto objetivo específico, se validó la metaheurística Búsqueda Tabú con 5 instancias artificiales y la información de 13 emprendimientos y 7 recursos turísticos del distrito de Juli. Con las instancias artificiales se logró igualar al 60% de las mejores soluciones encontradas hasta la fecha, y con los datos reales se obtuvo un ciclo hamiltoniano proponiendo un circuito turístico a medida para el turista, que inicia y termina en un mismo lugar y minimiza el costo del recorrido.

Con relación al quinto objetivo específico, se desarrolló una aplicación web que integra a la metaheurística Búsqueda Tabú, dicha aplicación web se desplegó en un servidor de aplicación con un el dominio www.turismoruralcomunitario.pe en donde se realizaron las pruebas de despliegue respectivas.

6.2. Recomendaciones

Para investigaciones futuras con un nivel avanzado se propone el uso de las estructuras de memoria basadas en cuatro dimensiones, contemplando las propiedades en: hechos recientes, frecuencia, la calidad e influencia.

El gobierno debería optar por proyecto de Ciencia y Tecnología en mercados emergentes como es el del Turismo Rural Comunitario, en vista que la mayoría de los visitantes utilizan

las TICs para buscar información, realizar reservas e interactuar con los proveedores de servicios; si los pobladores rurales están ajenos a estas tecnologías se disminuyen las posibilidades del crecimiento de sus negocios y por ende mejorar su calidad de vida.

Se necesita más investigaciones enfocadas a obtener el perfil del turista. Por medio de la aplicación desarrollada en este proyecto de tesis se podría capturar la interacción del visitante y luego analizarla obteniendo patrones que ayuden a identificar el perfil del turista; y en consecuencia los emprendedores podrían estar preparados a las necesidades del turista incluso antes que éste realice una reserva.

LISTA DE REFERENCIAS

- Araujo, E. C. (2006). Diseño e Implementación de un Algoritmo de tipo Recocido Simulado para la resolución del Problema del Agente Viajero. Guayaquil: Instituto de Ciencias matemáticas.
- Borrás, J., Moreno, A., & Valls, A. (2014). Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, 7370–7389.
- Buhalis, D. (2003). *eTourism: information technology for strategic tourism management*, Pearson. London: Financial Times/Prentice Hall.
- Calviño, A. M. (2011). Cooperación en los problemas del viajante (TSP) y de rutas de vehículos (VRP): una panorámica. Santiago de Compostela: Universidad de Santiago de Compostela.
- Canales, V. R. (17 de Mayo de 2016). Turismo Rural Comunitario en la Región Puno. (O. M. Apaza, Entrevistador)
- Cobos, N. Z. (2004). Búsqueda Tabú para el problema de diseño de red multiproducto con capacidad finita en las aristas. México: Universidad Autónoma de Nuevo León.
- Cobos, N. Z. (2004). Búsqueda tabu para un problema de diseño de red multiproducto con capacidad finita en las aristas. México: Universidad Autonoma de Nuevo León.
- Congreso de la República del Perú. (2009). Ley N° 29408, LEY GENERAL DE TURISMO. Lima: El Congreso de la Republica.
- Daza, J. M., Montoya, J. R., & Narducci, F. (2009). Resolución del problema de enrutamiento de vehículos con limitaciones de capacidad utilizando un procedimiento metaheurístico de dos fases. *EIA (Escuela de Ingeniería de Antioquia)*, 23-38.
- Fred, G., & Rafael, M. (2013). *TABU SEARCH*. Colorado: Leeds School of Business, University of Colorado.
- García, M. T. (2014). Problema del Viajante de Comercio (TSP) Métodos Exactos de resolución. La Laguna: Universidad de la Laguna.
- Hillier, F. (2010). *Introducción a la Investigación de Operaciones*. Mexico: Edmundo Carlos Zúñiga Gutiérrez.
- Hossian, A. (2015). Aplicación de Tecnologías Inteligentes para el Estudio de Conductas de Robots Móviles en Ambientes de Trabajo con Obstáculos Fijos. *Revista Latinoamericana de Ingeniería de Software*, 198-205.

- Isaza, R. A. (2012). Técnicas Heurísticas Aplicadas al Problema del Cartero Viajante (TSP). Pereira: Universidad Tecnológica de Pereira.
- Li, N., Buhalis, D., & Zhang, L. (2013). Interdisciplinary Research on Information Science and Tourism. *Information and Communication Technologies in Tourism 2013*, 302-313.
- Lopez, E., Salas, Ó., & Murillo, Á. (2013). El problema del agente viajero: un algoritmo determinístico usando Búsqueda Tabú. *Uála: Revista de matemáticas: Teoría y aplicaciones*.
- Machado, R. A. (2011). Plan Estratégico Regional de Turismo Puno - PERTUR 2021. Puno: DIRCETUR.
- Maquera, N. S. (17 de Enero de 2017). Introducción a la metaheurística Búsqueda Tabú. (O. A. Mendoza, Entrevistador)
- Martí, R. (2010). *Procedimientos Metaheurísticos en Optimización Combinatoria*. Valencia: Universidad de Valencia.
- Martí, R. (2013). *Procedimientos Metaheurísticos en Optimización Combinatoria*. Valencia: Universitat de Valencia - Departamento de Estadística e Investigación Operativa.
- Meléndez Valladares, S. M., Gaitan, M. E., & Pérez Reyes, N. N. (2016). **METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION. MANAGUA: UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, MANAGUA - FACULTAD DE CIENCIAS E INGENIERÍA.**
- Melián, B., Pérez, J. A., & Vega, J. M. (2003). Metaheuristics A global view. *Revista Iberoamericana de Inteligencia Artificial*, 7-28.
- Ministerio del Ambiente. (2011). **COMPENDIO DE LA LEGISLACIÓN AMBIENTAL PERUANA VOLUMEN VI** . Lima: Ministerio del ambiente del Peru.
- Mohme, C. C. (2013). Plan Estratégico Nacional de Turismo. Lima: MARKA - CULTURA & MARKETING.
- Moncloa, M. B. (2013). Perfil Sectorial del Ecoturismo. Lima: Dirección Corporativa de Ambiente y Cambio Climático .
- Pacheco, B. (2003). Estudio comparativo de diferentes estrategias metaheurísticas para la resolución del labor schedulin problem. *Estudios de Economía Aplicada*, 537-557.
- PROMPERU. (2013). **MILLENNIALS - TURISMO RECEPTIVO (2013)**. Lima: Rapimagen S.A.

- PROMPERU. (2015). Perfil del Vacacionista Nacional 2015. Lima: PROMPERU.
- Rejas Cano, E. A. (2014). DISEÑO DE UN ALGORITMO DE BÚSQUEDA TABÚ PARA RESOLVER EL PROBLEMA DE LA SELECCIÓN DE PROYECTOS. IIMA: PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ - FACULTAD DE CIENCIAS E INGENIERÍA.
- Riojas, A. C. (2005). Conceptos, algoritmo y aplicación al problema de las N - reinas. Lima: Universidad Nacional Mayor de San Marcos.
- Rodríguez, B., Molina, J., Pérez, F., & Caballero, R. (2012). Interactive design of personalised tourism routes. *Tourism Management*, 926-940.
- Romero, J. J. (2007). *Inteligencia Artificial y Computación Avanzada*. Santiago de Compostela: Fundación Alfredo Brañas.
- Russell, S. (2008). *Inteligencia Artificial Un Enfoque Moderno Segunda edicion*. Madrid: Ara Isabel Garcia Borno.
- Stockdale, M. L. (2011). El problema del viajante: un algortimo heurístico y una aplicación. Buenos Aires: Facultad de Ciencias Exactas y Naturales.
- Suaña, C. C. (17 de Mayo de 2016). Estado actual del turismo en el destino Lago Titicaca. (O. M. Apaza, Entrevistador)
- Taha, H. A. (2012). *Investigación de Operaciones*. Mexico: Gabriela Lopéz Ballesteros.
- Van, G. R. (2009). *Python*. Ámsterdam: python software foundation.
- Yunbulema, D. R. (2005). Diseño e implementación de un algoritmo de tipo recocido simulado para el reconocimiento de rutas en tiempo real. Guayaquil: Ingenieria en Estadística Informática.

ANEXOS

Anexo A Información recopilada del distrito de Juli.

Fecha	Recurso	Nombre	GPS			Ruta	Distancia	Tiempo	Costo	Tipo de transporte	Tipo de carretera
			Altura	Este	Norte						
	Atractivos	Comunidad de Suancata	3827	19k 444715	UTM 8207538	Juli - Suancata	7 km.	00:10 min.	S/. 5.00	Combi	Asfaltado - trocha
26/01/2017	Recursos	Casa de los hermanos Mamani Villalva	3911	19k 443815	UTM 8207584	Suancata - Recurso	824 mts.	00:30 min.	S/. 0.00	A pie	Trocha
26/01/2017		Casa de La Sra. Carmen Mamani Espinoza	3900	19k 443881	UTM 8207574	Suancata - Recurso	932 mts.	00:30 min	S/. 0.00	A pie	Trocha
26/01/2017		Casa de La Sra. Eduardo Mamani Anchapuri	3830	19k 444631	UTM 8206396	Suancata - Recurso	652 mts.	00:15 min	S/. 0.00	A pie	Trocha
26/01/2017		Casa de la Sra. Emilia Anchapuri Choquegonza	3833	19k 445444	UTM 8206380	Suancata - Recurso	1.7 Km.	00:10 min	S/. 0.50	Combi	Trocha
	Atractivo	Comunidad de Huaquina visita a la Casa de Alcidez Velazco Olivera	3839	19k 449033	UTM 8207950	Juli Huaquina	3.73 km	00:15 min	S/. 5.00	Combi	Asfaltado - trocha
26/01/2017	Recursos	Visita a la casa de Edgar Zaga Olivera	3345	19k 449248	UTM 8207846	Huaquina - recurso	239 mts.	00:05 min	S/. 0.00	A pie	Trocha
26/01/2017		Piedra de la culebra	3893	19k 449469	UTM 8208149	Huaquina - recurso	307 mts	00:20 min	S/. 0.00	A pie	Trocha
26/01/2017		Observatorio de piedra astrológica	3863	19k 449359	UTM 8208466	Casa de Alcidez - Recurso	631 mts.	00:05 min	S/. 1.00	Combi	Trocha
26/01/2017		Chullpa del Inca	3890	19k 449437	UTM 8208448	Piedra astrológica - Recurso	1 km.	00:05 min	S/. 1.00	Combi	Trocha
	Atractivo	Comunidad de Ccajje									
26/02/2017	Recursos	Complejo de Huaylluni	3344	19k 457147	UTM 8209959	Juli - Recurso	6.7 km.	00:15 min	S/. 5.00	Combi	Asfaltado - Trocha
26/02/2017		Manto de la Virgen	3855	19k 457184	UTM 8210157	Complejo - Recurso	2013 mts.	00:05 min	S/. 0.00	A pie	Trocha
26/02/2017		Cerro santo de Huaylluni - cerro calvario	3881	19k 457235	UTM 8210070	Complejo - Recurso	144 mts.	00:08 min	S/. 0.00	A pie	Trocha
26/02/2017		Casa del tallador Faustino	3874	19k 456021	UTM 8209437	Complejo - Recurso	1.2 km.	00:08 min	S/. 2.00	Combi	Trocha

	Atractivo	Comunidad de Olla	3904	19k 45271 0	UTM 82096 72							
27/02/2017	Recursos	Textileria y bordado	3904	19k 45271 0	UTM 82096 72	Juli - Recurso	4 km.	00:14 min	S/. 6.00	Combi	Trocha	
		Mutu Mutuni (Visita Paisaje Lago)	3856	19k 55278 2	UTM 82102 82	Juli - Recurso	3.4 km.	00:07 min	S/. 4.00	Combi	Trocha	
	Atractivo	Comunidad de Aurina	3895	19k 45970 7	UTM 82031 66	Jui - Comunid ad	12.5 km.	00:25 min	S/. 8.00	Combi	Asfalta do - Trocha	
27/02/2017	Recursos	Casa de Feliciano Arcaya C. (Tallado de piedra)	3909	19k 45956 0	UTM 82026 58	Plaza - Recurso	545 mts.	00:02 min	S/. 1.00	Combi	Trocha	
27/02/2017		Casa de Juana Herminia Calamullo	3946	19k 45943 4	UTM 82022 48	Paza - Recurso	976 mts.	00:03 min	S/. 1.00	Combi	Trocha	
27/02/2017		Casa de Rosalia Candia Apaza	3915	19k 45932 1	UTM 82024 62	Plaza - Recurso	825 mts.	00:02 min	S/. 1.00	Combi	Trocha	
27/02/2017		Casa de Feliciano Arcaya Candia (Huerto)	3905	19k 45920 7	UTM 82028 14	Paza - Recurso	634 mts.	00:04 min	S/. 1.50	Combi	Trocha	
	Atractivo	Comunidad de Morojahua										
27/02/2017	Recursos	Visita a la casa de la artesana Cooncepcion Jimenez Laquihunco	3903	19k 45750 7	UTM 82034 92	Juli - recurso	12.9 km.	00:30 min	S/. 10.0 0	Combi	Asfalta do - Trocha	

Fuente: Elaboración propia, 2017.

Anexo B Formulario 1 utilizados para el levantamiento de información.

Ficha _____ Fecha _____

Nombre del atractivo: _____

Ubicación: Región _____ Provincia _____ Distrito _____

Lugar de procedencia: _____ Hora de salida: _____ Hora de llegada _____

Costo viaje: _____ Distancia _____ Tipo de transporte: _____

Tipo carretera: _____

Otros: _____

RUTA DE ACCESO AL RECURSO

Recorrido	Tramo	Acceso	Medio de transporte	Via Acceso	Distancia

ÉPOCA PROPICIA DE VISITA AL RECURSO

Todo el Año

Esporádicamente-algunos meses.....

Fines de semana.....

Feriados.....

Días de visita:

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
-------	--------	-----------	--------	---------	--------	---------

Actividades:

Actividad	Costo	Duración	Hora

Anexo C Formulario 2 utilizados para el levantamiento de información

Recurso	GPS	Ruta	Distancia	Tiempo	Costo	Tipo de transporte	Tipo de carretera	Hora Salida	Hora Llegada

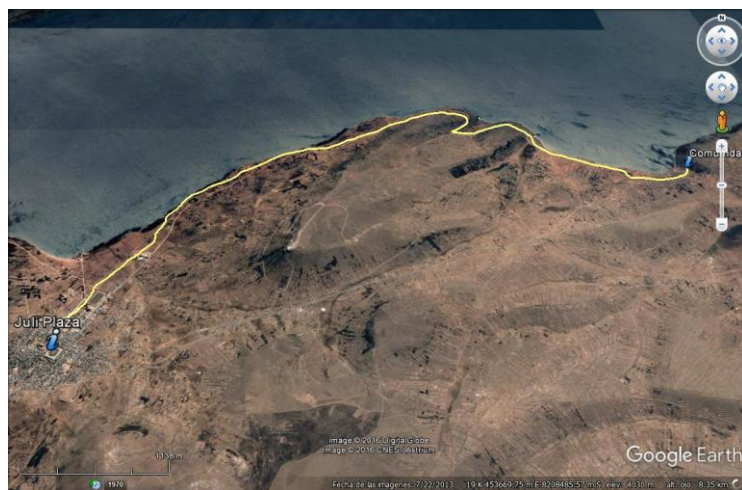
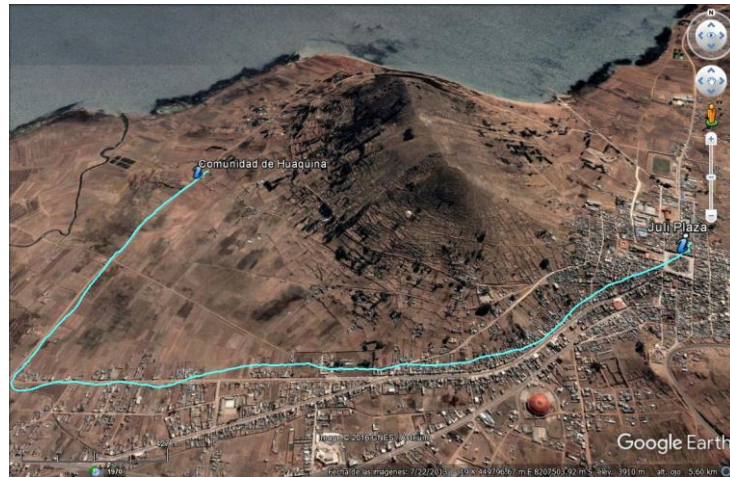
Anexo D Fotos del levantamiento de la información



Anexo E Validación de la aplicación web por parte de los actores del TRC de Puno



Anexo F Mapeo de tres rutas hacia los recursos turísticos.



Anexo G Instancia dantzig42

0	8	39	37	50	61	58	59	62	81	103	108	145	181	187	161	142	174	185	164	137	117	114	85	77	87	91	105	111	91	83	89	95	74	67	74	57	45	35	29	3	5
8	0	45	47	49	62	60	60	66	81	107	117	149	185	191	170	146	178	186	165	139	122	118	89	80	89	93	106	113	92	85	91	97	81	69	76	59	46	37	33	11	12
39	45	0	9	21	21	16	15	20	40	62	66	104	140	146	120	101	133	142	120	94	77	73	44	36	44	48	62	69	50	42	55	64	44	42	61	46	41	35	30	41	55
37	47	9	0	15	20	17	20	25	44	67	71	108	144	150	124	104	138	143	123	96	80	78	48	40	46	50	63	71	51	43	55	63	43	41	60	41	34	26	21	37	41
50	49	21	15	0	17	18	26	31	50	72	77	114	150	156	130	111	143	140	124	94	83	84	53	46	46	48	64	66	46	38	50	56	35	31	42	25	20	18	18	47	53
61	62	21	20	17	0	6	17	22	41	63	68	106	142	142	115	97	129	130	106	80	68	69	41	34	30	34	47	51	30	22	34	42	23	25	44	30	34	34	35	57	64
58	60	16	17	18	6	0	10	15	35	57	61	99	135	137	110	91	123	126	106	78	62	63	34	27	28	32	46	53	34	26	39	49	30	32	51	36	38	36	33	55	61
59	60	15	20	26	17	10	0	5	24	46	51	88	124	130	104	85	117	124	105	77	60	57	28	19	29	33	49	56	38	32	44	56	39	41	60	47	48	46	40	58	61
62	66	20	25	31	22	15	5	0	20	41	46	84	120	125	105	86	118	128	110	84	61	59	29	21	32	36	54	61	43	36	49	60	44	46	66	52	53	51	45	63	66
81	81	40	44	50	41	35	24	20	0	23	26	63	99	105	90	75	107	118	104	77	50	48	22	14	27	30	48	57	49	51	63	75	62	64	83	71	73	70	65	83	84
103	107	62	67	72	63	57	46	41	23	0	11	49	85	90	72	51	83	93	86	56	34	28	23	29	36	34	46	59	60	63	76	86	78	83	102	93	96	93	87	105	111
108	117	66	71	77	68	61	51	46	26	11	0	40	76	81	62	59	84	101	97	64	42	36	35	40	47	45	59	71	71	75	87	97	89	90	110	98	99	97	91	109	113
145	149	104	108	114	106	99	88	84	63	49	40	0	35	41	34	29	54	72	71	65	49	43	69	77	78	77	85	96	103	106	120	126	121	130	147	136	137	134	117	147	150
181	185	140	144	150	142	135	124	120	99	85	76	35	0	10	31	53	46	69	93	90	82	77	105	114	116	115	119	130	141	142	155	160	159	164	185	172	176	171	166	186	186
187	191	146	150	156	142	137	130	125	105	90	81	41	10	0	27	48	35	58	82	87	77	72	102	111	112	110	115	126	136	140	150	155	155	160	179	172	178	176	171	188	192
161	170	120	124	130	115	110	104	105	90	72	62	34	31	27	0	21	26	58	62	58	60	45	74	84	84	83	88	98	109	112	123	128	127	133	155	148	151	151	144	164	166
142	146	101	104	111	97	91	85	86	75	51	59	29	53	48	21	0	31	43	42	36	30	27	56	64	66	63	66	75	90	93	100	104	108	114	133	126	131	129	125	144	147
174	178	133	138	143	129	123	117	118	107	83	84	54	46	35	26	31	0	26	45	68	62	59	88	96	98	97	98	98	115	126	123	128	136	146	159	158	163	161	157	176	180
185	186	142	143	140	130	126	124	128	118	93	101	72	69	58	58	43	26	0	22	50	70	69	99	107	95	91	79	85	99	108	109	113	124	134	146	147	159	163	156	182	188
164	165	120	123	124	106	106	105	110	104	86	97	71	93	82	62	42	45	22	0	30	49	55	81	87	75	72	59	62	81	88	86	90	101	111	122	124	135	139	139	161	167
137	139	94	96	94	80	78	77	84	77	56	64	65	90	87	58	36	68	50	30	0	21	27	54	60	47	44	31	38	53	60	62	67	75	85	98	121	108	118	113	134	140
117	122	77	80	83	68	62	60	61	50	34	42	49	82	77	60	30	62	70	49	21	0	5	32	40	36	32	36	47	61	64	71	76	79	84	105	97	102	102	95	119	124
114	118	73	78	84	69	63	57	59	48	28	36	43	77	72	45	27	59	69	55	27	5	0	29	37	39	36	42	53	62	66	78	82	81	86	107	99	103	101	97	116	119
85	89	44	48	53	41	34	28	29	22	23	35	69	105	102	74	56	88	99	81	54	32	29	0	8	12	9	28	39	36	39	52	62	54	59	79	71	73	71	67	86	90
77	80	36	40	46	34	27	19	21	14	29	40	77	114	111	84	64	96	107	87	60	40	37	8	0	11	15	33	42	34	36	49	59	50	52	71	65	67	65	60	78	87
87	89	44	46	46	30	28	29	32	27	36	47	78	116	112	84	66	98	95	75	47	36	39	12	11	0	3	21	29	24	27	39	49	42	47	66	59	64	65	62	84	90
91	93	48	50	48	34	32	33	36	30	34	45	77	115	110	83	63	97	91	72	44	32	36	9	15	3	0	20	30	28	31	44	53	46	51	70	63	69	70	67	88	94
105	106	62	63	64	47	46	49	54	48	46	59	85	119	115	88	66	98	79	59	31	36	42	28	33	21	20	0	12	20	28	35	40	43	53	70	67	75	84	79	101	107
111	113	69	71	66	51	53	56	61	57	59	71	96	130	126	98	75	98	85	62	38	47	53	39	42	29	30	12	0	20	28	24	29	39	49	60	62	72	78	82	108	114
91	92	50	51	46	30	34	38	43	49	60	71	103	141	136	109	90	115	99	81	53	61	62	36	34	24	28	20	20	0	8	15	25	23	32	48	46	54	58	62	88	77
83	85	42	43	38	22	26	32	36	51	63	75	106	142	140	112	93	126	108	88	60	64	66	39	36	27	31	28	28	8	0	12	23	14	24	40	38	46	50	53	80	86
89	91	55	55	50	34	39	44	49	63	76	87	120	155	150	123	100	123	109	86	62	71	78	52	49	39	44	35	24	15	12	0	11	14	24	36	37	49	56	59	86	92
95	97	64	63	56	42	49	56	60	75	86	97	126	160	155	128	104	128	113	90	67	76	82	62	59	49	53	40	29	25	23	11	0	21	30	33	43	54	62	66	92	98
74	81	44	43	35	23	30	39	44	62	78	89	121	159	155	127	108	136	124	101	75	79	81	54	50	42	46	43	39	23	14	14	21	0	9	25	23	34	41	45	71	80
67	69	42	41	31	25	32	41	46	64	83	90	130	164	160	133	114	146	134	111	85	84	86	59	52	47	51	53	49	32	24	24	30	9	0	18	13	24	32	38	64	74
74	76	61	60	42	44	51	60	66	83	102	110	147	185	179	155	133	159	146	122	98	105	107	79	71	66	70	70	60	48	40	36	33	25	18	0	17	29	38	45	71	77
57	59	46	41	25	30	36	47	52	71	93	98	136	172	172	148	126	158	147	124	121	97	99	71	65	59	63	67	62	46	38	37	43	23	13	17	0	12	21	27	54	60
45	46	41	34	20	34	38	48	53	73	96	99	137	176	178	151	131	163	159	135	108	102	103	73	67	64	69	75	72	54	46	49	54	34	24	29	12	0	9	15	41	48
35	37	35	26	18	34	36	46	51	70	93	97	134	171	176	151	129	161	163	139	118	102	101	71	65	65	70	84	78	58	50	56	62	41	32	38	21	9	0	6	32	38
29	33	30	21	18	35	33	40	45	65	87	91	117	166	171	144	125	157	156	139	113	95	97	67	60	62	67	79	82	62	53	59	66	45	38	45	27	15	6	0	25	32</

Anexo H Servidor local del API que contiene los recursos para la ejecución de los algoritmos

The image shows a screenshot of an IDE with two main panels. The top panel displays Python code for a routing algorithm, and the bottom panel shows a Threading graph.

Python Code (Left Panel):

```
21 nodes_by_visit[p_start] = -1
22 # Recorrer a partir del segundo nodo
23 for i in range(1, (len(distances) + 1)):
24     connected_nodes = []
25     # Obtener los vecinos del nodo
26     for idx, distance in enumerate(distances[optimal_route[-1]]):
27         val = distance, idx
28         connected_nodes.append(val)
29     # Obtenemos el vecino mas cercano
30     sorted_connected_nodes = sorted(connected_nodes)
31     for l in sorted_connected_nodes:
32         # Verificar si el nodo aún no fue visitado
33         if (l[0] > 0 and l[1]) in nodes_by_visit:
34             optimal_route.append(l[1])
35             nodes_by_visit[l[1]] = -1
36             total_distance = total_distance + l[0]
37             break
38     # Retornamos al nodo inicial
39     total_distance = total_distance + distances[p_start][optimal_route
40     optimal_route.append(p_start)
41     return optimal_route, total_distance
42
43
44 def improvement(optimal_route, total_distance, distances):
45     best = total_distance
```

Python Code (Right Panel):

```
59     new_value_to = value_to + value_movement
60     exchange_environment = (value_movement, new_value
61     swaps.append(exchange_environment)
62     return swaps
63
64 import numpy as np
65
66 def updateShortMemory(short_memory):
67     for idx, i in enumerate(short_memory):
68         for idy, j in enumerate(i):
69             if j > 0:
70                 short_memory[idx][idy] = j - 1
71     return short_memory
72
73 def tabuSearch(route, value_fo, distances):
74     new_route = route[:]
75     new_value_fo = value_fo
76     short_memory = np.zeros((len(route) - 1, len(route) - 1), dtype=
77     best_solution = [route, value_fo]
78     for i in range(len(route) * 2):
79         print(" i -> ", i)
80         exchange_environment = sorted(getSwaps(new_route, new_valu
81         for j in exchange_environment:
```

Threading graph (Bottom Panel):

Name	Waiting time (s)
MainThread	0.00 (0%)

The Threading graph shows a single thread named 'MainThread' running for approximately 10.5 seconds. The graph includes a legend for thread states: Running (green), Waiting for lock (yellow), Running with lock (blue), and Deadlock (red).