



Metodología integrativa para la búsqueda de caminos en Instituciones Educativas contextualizada en plano tridimensional

Ivan Adrianzén Olano
Gisella Luisa Elena Maquen Niño
Alejandro Chayán Coloma
Diana Mercedes Castro Cárdenas
Franklin Edinson Terán Santa Cruz
Freddy William Campos Flores



**Metodología integrativa para la búsqueda
de caminos en Instituciones Educativas
contextualizada en plano tridimensional**

**Metodología integrativa para la búsqueda de caminos
en Instituciones Educativas
contextualizada en plano tridimensional**

Ivan Adrianzén Olano
Gisella Luisa Elena Maquen Niño
Alejandro Chayán Coloma
Diana Mercedes Castro Cárdenas
Franklin Edinson Terán Santa Cruz
Freddy William Campos Flores



Ivan Adrianzén Olano
Gisella Luisa Elena Maquen Niño
Alejandro Chayán Coloma
Diana Mercedes Castro Cárdenas
Franklin Edinson Terán Santa Cruz
Freddy William Campos Flores

Metodología integrativa para la búsqueda de caminos en Instituciones
Educativas contextualizada en plano tridimensional

ISBN: 978-9942-603-34-0

Savez editorial

Título: Metodología integrativa para la búsqueda de caminos en
Instituciones Educativas contextualizada en plano tridimensional

Primera Edición: Marzo 2022

ISBN: 978-9942-603-34-0

Obra revisada previamente por la modalidad doble par ciego, en caso
de requerir información sobre el proceso comunicarse al correo
electrónico
editor@savezeditorial.com

Queda prohibida la reproducción total o parcial de esta obra por cualquier
medio (electrónico, mecánico, fotocopia, grabación u otros), sin la previa
autorización por escrito del titular de los derechos de autor, bajo las sanciones
establecidas por la ley. El contenido de esta publicación puede ser reproducido
citando la fuente.

El trabajo publicado expresa exclusivamente la opinión de los autores, de
manera que no compromete el pensamiento ni la responsabilidad del Savez
editorial

ÍNDICE

PRÓLOGO	7
CAPÍTULO I: CONTEXTO ACTUAL DE LA INTELIGENCIA ARTIFICIAL.....	8
DESARROLLO DE HABILIDADES E INVERSIÓN EN IA A NIVEL MUNDIAL	8
EUROPA	13
AMÉRICA	15
PERÚ	18
CAPITULO II: EVOLUCIÓN DE LA BÚSQUEDA DE CAMINOS Y PLANIFICACIÓN DE RUTAS DENTRO DE LA INTELIGENCIA ARTIFICIAL.....	20
NACIMIENTO DE LA IA (1956 - 1969)	20
LA IA COMIENZA A INDUSTRIALIZARSE (1980 A 1995)	22
SURGIMIENTO DE LOS AGENTES INTELIGENTES (1995 A 2000)	22
HACIA LA PLANIFICACIÓN DE RUTAS (2001 A LA ACTUALIDAD)	23
EXPANSIÓN DE LA BÚSQUEDA DE CAMINOS (2011 A LA ACTUALIDAD)	24
CAPÍTULO III: INVESTIGACIONES REALIZADAS EN BÚSQUEDA DE CAMINOS Y PLANIFICACIÓN DE RUTAS .	28
INVESTIGACIONES PREVIAS A NIVEL MUNDIAL	28
INVESTIGACIONES PREVIAS A NIVEL NACIONAL	38
INVESTIGACIONES PREVIAS A NIVEL LOCAL.....	42
CAPÍTULO IV: CONCEPTOS RELACIONADOS CON BÚSQUEDA DE CAMINOS	46
BÚSQUEDA DE CAMINOS	46
ESTRATEGIAS DE BÚSQUEDA DE CAMINOS.....	47
ESTRATEGIAS DE BÚSQUEDA A CIEGAS	48
ESTRATEGIAS DE BÚSQUEDA INFORMADA.....	59
PLANIFICACIÓN DE RUTAS.....	65
MÉTODO DE LA RUTA MÁS CORTA.....	66
CAPÍTULO V: ELABORACIÓN DE INSTRUMENTOS DE RECOJO DE DATOS SOBRE LA BÚSQUEDA DE CAMINOS EN UNA INSTITUCIÓN EDUCATIVA PUBLICA.....	70
TÉCNICAS E INSTRUMENTOS DE RECOJO DE DATOS	70
MARCO ÉTICO EN LA APLICACIÓN DE INSTRUMENTOS.....	74
CAPÍTULO VI: DIAGNÓSTICO DEL ESTADO ACTUAL DEL PROCESO DE BÚSQUEDA DE CAMINOS EN UNA INSTITUCIÓN EDUCATIVA PÚBLICA DE PERÚ.....	77
RESULTADOS DEL PROCESAMIENTO DE INFORMACIÓN DEL CUESTIONARIO A PADRES DE FAMILIA	78
RESULTADOS DEL PROCESAMIENTO DE INFORMACIÓN DE LA GUÍA DE ENTREVISTA AL ENCARGADO DE PORTERÍA	79
RESULTADOS DEL PROCESAMIENTO DE INFORMACIÓN DEL CUESTIONARIO A ESPECIALISTAS EN INTELIGENCIA ARTIFICIAL	80
INTERPRETACIÓN Y DISCUSIÓN DE RESULTADOS	87
CONCLUSIONES.....	95
REFERENCIAS BIBLIOGRÀFICAS	97

Prólogo

Esta investigación nace de la inquietud por situar a los procesos de búsqueda de caminos y planificación de rutas dentro de la inteligencia artificial y la dinámica que han tenido para poder evolucionar. El crecimiento de sistemas informáticos para búsqueda y planificación de rutas es vertiginoso y desordenado, y no hay una metodología específica para este tipo de software, de tal manera que su desarrollo e implementación es engorroso, disperso y difícil, lo que redundaría en que investigadores inexpertos abandonen su desarrollo. Por eso es que se traza como objetivo realizar un estudio de la evolución histórica de ambos procesos para posteriormente realizar el estudio del proceso de búsqueda de caminos dentro de una Institución Educativa Pública de educación básica regular de la ciudad de Lambayeque – Perú que permita determinar su problemática.

Para poder realizar esta investigación se realizó la investigación bibliográfica a través de la consulta a Base de Datos confiables entre las que se cita a IEEEExplore, Dialnet, Scopus, Scielo, Alicia y el metabuscador Google académico.

Como resultados se identifica que la búsqueda de caminos nace en el área de la construcción de agentes inteligentes dentro de la Inteligencia Artificial, mientras que la planificación de rutas es propia del área de Investigación de operaciones pero que actualmente ambos procesos utilizan algoritmos comunes para poder resolver problemas cada vez más complejos y que el éxito del sistema de búsqueda dependerá de la buena elección del algoritmo o algoritmos, que se adecuen a los criterios de optimalidad que se deben satisfacer dentro de un escenario determinado. Se concluye que la búsqueda de caminos y la planificación de rutas son procesos dinámicos que están relacionados entre sí y que su aplicación exitosa depende de la buena elección del algoritmo de búsqueda.

Capítulo I: Contexto actual de la Inteligencia Artificial.

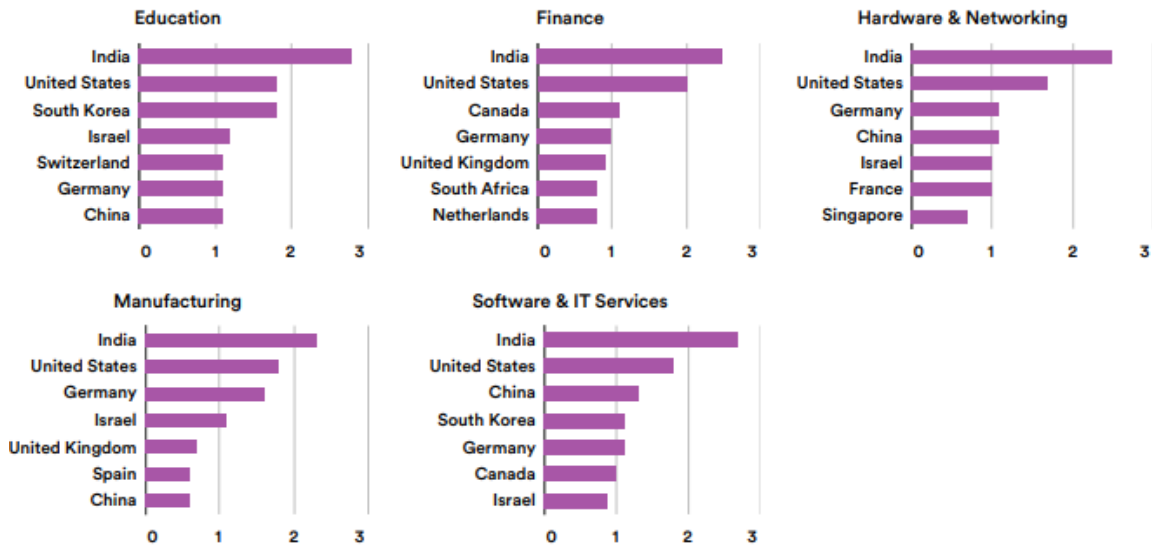
Desarrollo de habilidades e Inversión en IA a nivel mundial

A nivel mundial se viene incrementando la utilización de Inteligencia Artificial en diversas áreas de la actividad humana, y se puede medir a partir de la penetración que ha tenido las diferentes áreas de Inteligencia Artificial (IA) en sectores empresariales. Para proporcionar una descomposición sectorial de la penetración de IA en la industria, se ha analizado los datos de los cinco rubros industriales con la penetración de habilidades de IA más alta a nivel mundial en los últimos cinco años: educación, finanzas, hardware - redes, manufactura, software - Tecnologías de la Información (D. Zhang et al., 2021).

India tiene la penetración relativa más alta de habilidades de inteligencia artificial en las cinco industrias, mientras que Estados Unidos aparece en el segundo puesto. China aparece con frecuencia en los primeros lugares de la lista en cuatro de los cinco sectores empresariales. Otro país que vale la pena destacar es Alemania, con tasas de penetración muy altas en hardware-redes, así como en manufactura; e Israel en manufactura y educación.

Figura 1:

Tasa de penetración de Habilidades en IA por sector industrial. Periodo 2015-2020.

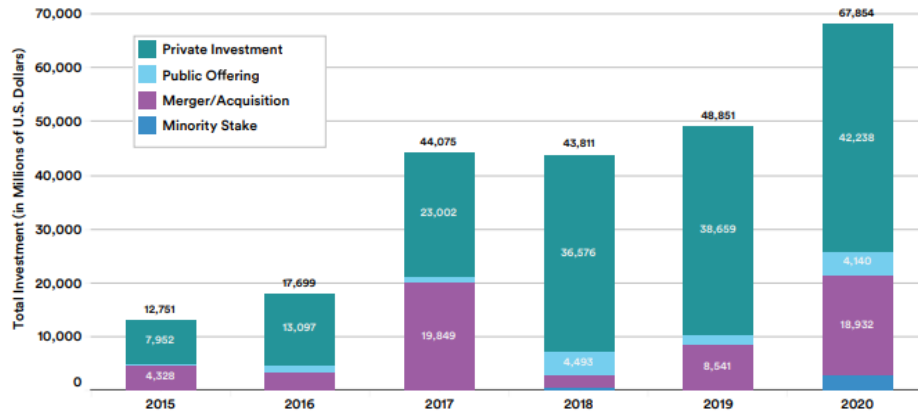


Fuente: (D. Zhang et al., 2021)

Otro rubro importante a analizar, es la inversión que están teniendo los países en inteligencia artificial. Se ha encontrado que la inversión global total en Inteligencia Artificial, aumentó en un 40% en 2020 en relación con el año 2019, siendo total de 67 864 millones de dólares para el año 2020. Las fusiones y adquisiciones constituyeron la mayor parte del monto total de la inversión en 2020, aumentando un 121,7% en relación con 2019, debido a que por la pandemia varias empresas pequeñas han tenido que fusionarse lo que está impulsando la inversión corporativa en IA. (D. Zhang et al., 2021)

Figura 2:

Inversión corporativa global en IA por actividad de inversión. Periodo 2015-2020.

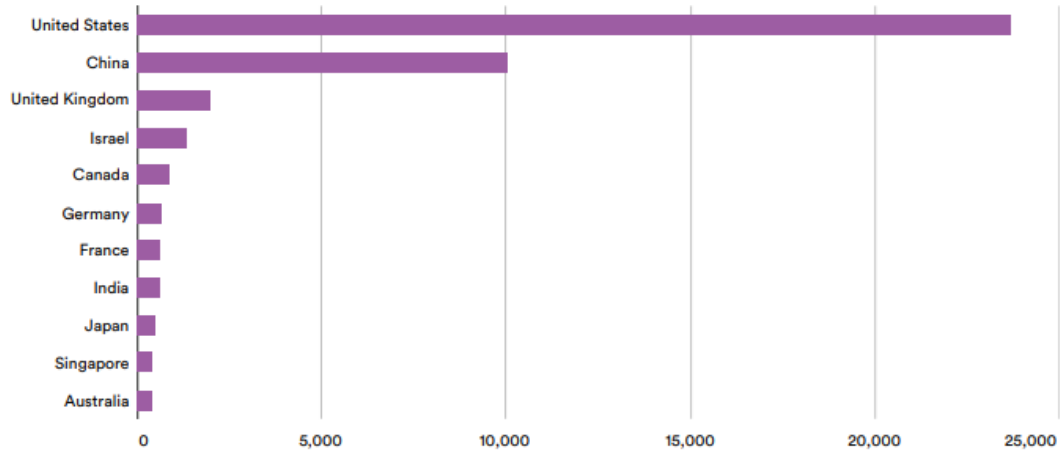


Fuente: (D. Zhang et al., 2021)

Con respecto a la inversión sectorizada por país, Estados Unidos sigue siendo el principal destino de la inversión privada, con más de USD 23,6 billones en financiamiento en el 2020, seguido por China con USD 9,9 billones de dólares y Reino Unido con USD 1,9 billones de dólares (D. Zhang et al., 2021). Con respecto a China, que ocupa el segundo puesto en inversión en IA a nivel mundial, si bien tuvo una cantidad excepcionalmente alta de inversión privada en inteligencia artificial en el 2018, su nivel de inversión en el 2020 es menor a la mitad de la inversión en EEUU. No obstante, la diferencia que existe entre EEUU y China es que esta última tiene fuertes inversiones públicas en IA. Tanto el gobierno central como el gobierno local están gastando mucho en proyectos de I + D (inversión más desarrollo) en Inteligencia Artificial.

Figura 3:

Inversión privada en IA por país año 2020 CapIQ, Crunchbase y NetBase Quid

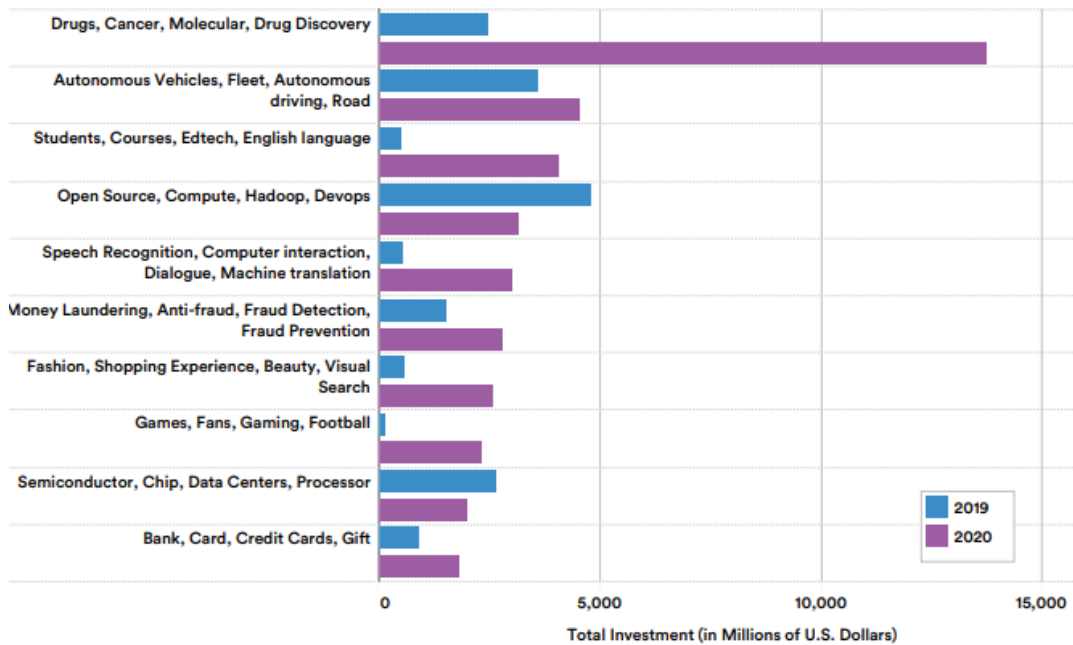


Fuente: (D. Zhang et al., 2021)

Con respecto a las áreas de enfoque principales que recibieron la mayor cantidad de inversión privada en 2020, tenemos en segundo puesto a "Vehículos autónomos, flota, conducción autónoma, carretera", con USD 4.5 billones de dólares. Además, tanto " Juegos, aficionados, videojuegos, fútbol "y" Estudiantes, cursos, tecnología educativa, idioma inglés "vieron un aumento significativo en la cantidad de inversión privada en IA de 2019 a 2020. El primero está impulsado en gran medida por varios proyectos financieros impulsados por empresas de juegos y deportes en Estados Unidos y Corea del Sur, mientras que este último se encuentra promovido por inversiones de una plataforma de educación en línea en China. (D. Zhang et al., 2021)

Figura 4:

Inversión privada global en IA por área de foco 2019-2020 obtenido en CapIQ, Crunchbase y NetBase Quid



Fuente: (D. Zhang et al., 2021)

Esto indica que existe una fuerte inversión en investigaciones que mejoren la conducción autónoma y donde la búsqueda de caminos y planificación de rutas juega un papel importante. Así mismo, el sector de videojuegos utiliza algoritmos para búsqueda de caminos que involucren un sistema de posicionamiento dentro de un plano que son puntos requeridos en la presente investigación.

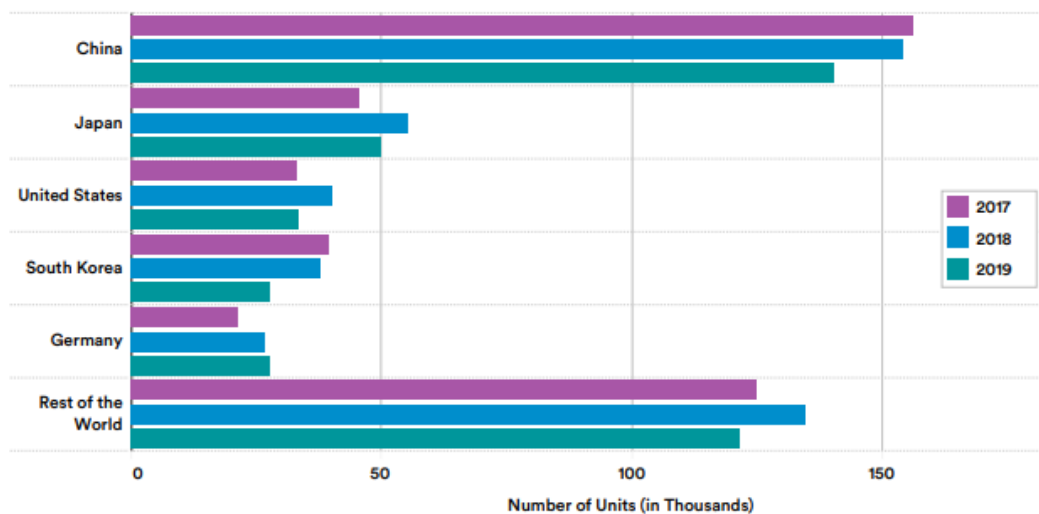
También se han creado empresas que fabrican robots industriales que poseen algoritmos de IA para poder atender el mercado. Asia, Europa y América del Norte, son tres de los mercados más grandes de robots industriales y después de un período de crecimiento de seis años, en el año 2019 obtuvieron un decremento en las instalaciones de robots. América del

Norte experimentó la disminución más pronunciada, del 16%, en 2019, en comparación con el 5% en Europa y el 13% en Asia. (D. Zhang et al., 2021)

Los cinco países que poseen la mayor cantidad de empresas de robots industriales son: China, Japón, Estados Unidos, Corea del Sur y Alemania, con el 73% de las instalaciones de empresas de robots industriales, experimentaron disminuciones en sus instalaciones, a excepción de Alemania, que experimentó un ligero aumento en las instalaciones entre 2018 y 2019. Aún existe una tendencia decreciente en China, se debe resaltar que el país tuvo más robots industriales en el 2019 que los otros cuatro países juntos.

Figura 5:

Nuevas Instalaciones de robots industriales en cinco mercados principales, 2017-2019



Fuente: (D. Zhang et al., 2021)

Europa

En este contexto, en **Europa** se vienen desarrollando investigaciones en Inteligencia Artificial. La Comisión Europa ha publicado "El Libro Blanco

sobre inteligencia artificial: un enfoque europeo de la excelencia y la confianza”, que establece un marco de trabajo común en todos los países de la Unión Europea con respecto a reglamentación y trabajo con Inteligencia Artificial.

Durante el periodo del 2018 al 2020, el financiamiento que ha destinado la Unión Europea para investigaciones e innovaciones en Inteligencia Artificial se ha elevado en 1500 millones de euros, es decir un aumento del 70% en comparación con el periodo de hace tres años atrás. Sin embargo, esta inversión sigue siendo mucho menor que la que se ha dado en otras regiones del mundo. En 2018 se invirtió 3200 millones de euros en IA en Europa, en comparación con 12100 millones de euros en América del Norte y 6500 millones en Asia. Para poder incrementar la inversión en Inteligencia Artificial la Comisión Europea propondrá a los estados miembros financiar 20000 millones de euros anuales en IA durante la próxima década, tanto de inversiones públicas como privadas (Comisión Europea; 2020).

En el “Libro Blanco sobre Inteligencia Artificial”, también se habla de la posible creación de una personalidad jurídica, en términos de una “personalidad electrónica” para aquellos robots que estén programados para tomar decisiones autónomas inteligentes. Si bien el Parlamento Europeo hace alusión a una “persona electrónica”, estos derechos solo serán aplicados a aquellos robots inteligentes que sean ciber-físicos, siendo la denominación más correcta “persona electro-física” ya que este término es el que más se acomoda a la configuración misma de la visión que se tiene de la estructura de un robot, componentes físicos (hardware) con equipamiento lógico de computación, almacenamiento y comunicación (software) para controlar e interactuar en un entorno, de

acuerdo a algoritmos computacionales previamente configurados y con acceso a internet para la actualización de información (Valdivia, 2021) .

También se debe destacar que Europa produce más de una cuarta parte de todos los servicios industriales y profesionales por medio de agentes inteligentes en áreas como logística (Comisión Europea;, 2020), y realiza un papel importante en desarrollar y utilizar aplicaciones de software que automatizan la cadena de valor, encontrándose dentro de este rubro los sistemas de búsqueda y planificación de rutas.

La mayoría de estudios con respecto la planificación de rutas, se están dando principalmente en **Asia**, donde se ve la necesidad de ver algoritmos cada vez más eficiente para poder afrontar la complejidad de los entornos donde se desarrollarán la búsqueda de rutas, teniendo como principales criterios de optimalidad hallar la ruta más corta y lograr el menor tiempo de cálculo de la ruta.

América

Con respecto a **América**, el país principal que invierte en Inteligencia Artificial es Estados Unidos, siendo sus niveles de inversión comparables a los de Asia, y por tanto liderando en el mundo.

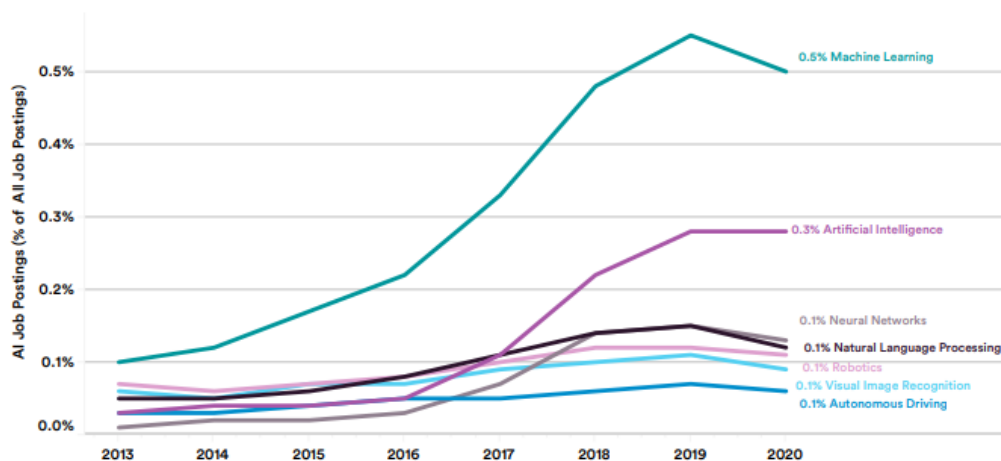
Según el Artificial Intelligence Index Report (2021), informe anual de la Universidad de Stanford, el interés de **Estados Unidos** en Inteligencia Artificial (IA) se hace evidente en ámbitos como las ofertas de empleo. En Estados Unidos las ofertas de trabajo han disminuido por el contexto de la pandemia del COVID- 19, y las ofertas de trabajo dedicadas a inteligencia artificial no han sido la excepción, teniendo una caída que no se veía hace

6 años del 8.2%, de 325724 trabajos en el 2019 a 300999 en el 2020. (D. Zhang et al., 2021)

Pero sin embargo si lo vemos en un periodo mayor, del 2013 al 2020, los trabajos relacionados con la línea de aprendizaje automático e inteligencia artificial experimentaron un crecimiento en las ofertas de trabajo en los Estados Unidos, aumentando del 0,1% al 0,5% y del 0,03% al 0,3% del total de puestos de trabajo, respectivamente. (D. Zhang et al., 2021)

Figura 6:

Porcentaje de publicaciones de ofertas laborales agrupadas por habilidades en Ramas de Inteligencia Artificial en Estados Unidos. Periodo 2013-2020.



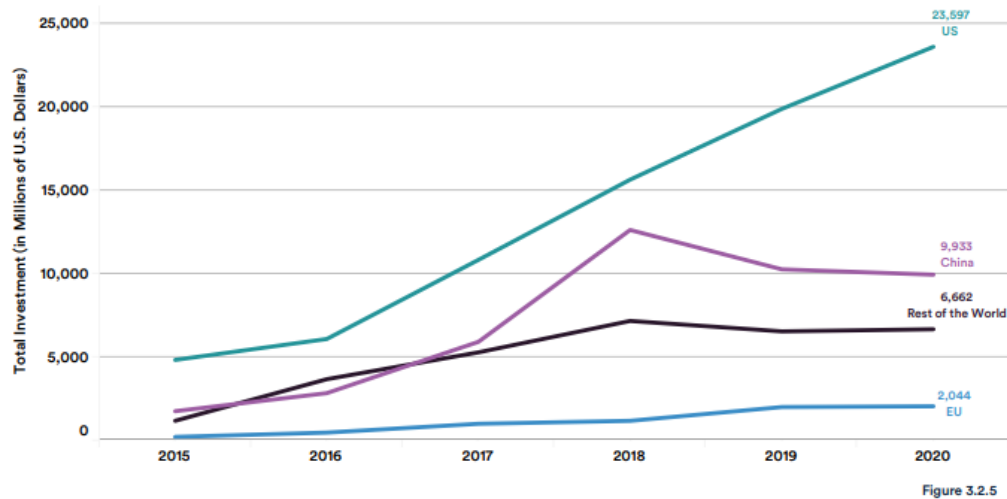
Fuente: (D. Zhang et al., 2021)

Con respecto a los cinco sectores industriales educación, finanzas, hardware - redes, manufactura, software - Tecnologías de la Información, que tuvieron mayor penetración de habilidades de Inteligencia Artificial en el año 2020 Estados Unidos ocupa el segundo puesto en los cinco sectores industriales, con índice mayor a 1.5 y menor igual a dos veces la tasa mostrada en el año 2015. (D. Zhang et al., 2021)

Otro aspecto a tener en cuenta para poder analizar la importancia de la Inteligencia Artificial en los países, es la inversión de cada uno de ellos en este rubro. Estados Unidos sigue siendo el principal destino de la inversión privada, con más de USD 23,6 billones en financiamiento en el 2020, demostrando que año a año la inversión ha ido incrementando, desde el 2015 donde solo se daba una inversión de USD 5 billones de dólares, y que a pesar de la pandemia ha logrado seguir incrementando en el año 2018 de aproximadamente USD 15 billones a aproximadamente USD 20 billones en el año 2019. (D. Zhang et al., 2021)

Figura 7:

Inversión privada en IA por área geográfica periodo 2015-2020 extraída de CAPIQ, Crunchbase y NetBase Quid



Fuente: (D. Zhang et al., 2021)

Hay muchos países de América cuyo puesto de empleo en IA han aumentado en el periodo del 2016 al 2020. Brasil, Canadá y Argentina se encuentran dentro de los 14 países con el mayor crecimiento en la contratación de Inteligencia Artificial en el mundo. Para el país líder, Brasil, el índice de contratación creció más de 3,5 veces, a pesar de la pandemia de COVID-19. En Canadá que ocupó el tercer puesto, la tasa de contratación creció en 2.7 veces, mientras que Argentina que ocupó el noveno puesto, la tasa de contratación creció al doble que lo ofrecido en el año 2016.

Perú

A nivel Nacional, se ha encontrado que Perú y 41 países miembros y socios de la Organización para la Cooperación y el Desarrollo Económicos (OCDE) firmaron en el año 2019, nuevos principios sobre Inteligencia Artificial, compuesto por un conjunto de directrices de política intergubernamental sobre Inteligencia Artificial (IA), acordando respetar estándares internacionales que tienen como objetivo certificar que los sistemas hechos dentro del rubro de Inteligencia Artificial estén diseñados para ser robustos, seguros, y confiables.

Estas directrices fueron elaboradas con la orientación de un grupo de experto, entre los que encontramos: "miembros de gobiernos, academia, empresas, sociedad civil, organismos internacionales entre otros y abarca cinco principios para el despliegue responsable de proyectos en IA y cinco recomendaciones de política pública y cooperación internacional" (OCDE, 2019). Su objetivo es guiar a los gobiernos, las organizaciones y las personas en el diseño y la ejecución de sistemas de IA de manera que priorice los intereses de las personas y se garantice que los diseñadores y operadores

sean responsables del correcto funcionamiento de los sistemas.(OCDE, 2019)

Capitulo II: Evolución de la búsqueda de caminos y planificación de rutas dentro de la inteligencia artificial

La búsqueda de caminos estuvo de la mano con la evolución de la Inteligencia Artificial ya que es a través de esta disciplina de la computación e informática que se utilizó algoritmos que permitan la búsqueda de soluciones a problemas

En tal sentido se puede esbozar un análisis cronológico de la evolución del proceso de búsqueda de caminos.

Nacimiento de la IA (1956 - 1969)

Comienza con el acuñamiento del nombre de Inteligencia Artificial en el taller del Dartmouth College que conglomeró a los científicos más reconocidos en este campo de investigación,

En esta etapa, el proceso de búsqueda de caminos iba orientado a la búsqueda de soluciones de problemas, creándose el SRGP (Sistema General de resolución de problemas) creado por Newell y Simón quienes querían construir un programa que simulara los procesos humanos para resolver problemas por medio de encontrar soluciones a puzzles, sin embargo, a pesar de que funcionaba perfectamente para problemas conocidos cuando se cambiaba alguna de las variables el programa no daba solución. Su más grande contribución está en el hecho de que demostraron que se pueden resolver problemas manipulando estructuras de datos simbólicas.

Otro gran aporte de esta etapa es el Generador de consejos, programa creado por McCarthy que permitía buscar solución de un problema utilizando representaciones del conocimiento a través de axiomas.

A pesar que estos programas sentaron la base del desarrollo de algoritmos de inteligencia artificial, el aporte más importante lo tuvo Minsky quien con ayuda de sus estudiantes, planteo la búsqueda de soluciones a problemas limitados a un dominio específico, conocidos como los Micro mundos. Así se creaban programas que abarcaran ramas específicas del quehacer científico, creandose ANALOGY programa que resolvía problemas de analogía geométrica. STUDENT programa para resolución de problemas de algebra, BLOQUES, que permitía la reordenación de bloques en un espacio geográfico limitado.

Era de hielo de la IA (1966 -1973)

Uno de los principales problemas a los que se tuvo que enfrentar en el proceso de búsqueda de caminos, fue resolver mediante Inteligencia Artificial problemas "intratables". Los primeros programas utilizaban un conjunto de pasos iterativos hasta encontrar la solución. Esto funcionó en los micromundos por que la cantidad de variables a tener en cuenta en la colusión eran limitadas, por tanto, la secuencia de la solución era muy corta y la búsqueda contenía pocas acciones. En tal sentido, la solución que el programa encontraba no encerraba todas las posibles soluciones que se podía encontrar en la solución práctica.

Es así como los investigadores buscaron hacer una serie de pequeñas mutaciones al programa en código de bajo nivel para generar un programa con un rendimiento más adecuado y por tanto pudiera aprovechar de mejor manera los recursos limitados de procesamiento y memoria. Después, se

probaron mutaciones aleatorias aplicando un proceso de selección con el fin de guardar aquellas mutaciones que resultaron incrementar de mejor modo la eficiencia del programa, sentando las bases teóricas para la creación posterior de los algoritmos genéticos.

La IA comienza a industrializarse (1980 a 1995)

La industria de IA comenzó a obtener grandes fuentes de financiamiento tanto en Estados Unidos por parte de empresas privadas que buscaban tener procesos con altos rendimientos, así como Japón quien apostaba por el financiamiento estatal a proyectos de Inteligencia artificial que tuvieran relevancia en la industria.

En esta etapa comenzó a utilizarse conceptos estadísticos como las redes de bayes para facilitar la representación y el razonamiento al momento de realizar búsqueda de soluciones. Esto permitía la mejor comprensión de los problemas y ayudaba a tratar su complejidad, a través de algoritmos más robustos.

Surgimiento de los agentes inteligentes (1995 a 2000)

Se creó un movimiento "situado" que intenta entender la forma en como un agente actúa en entornos reales, captando su entorno a través de sensores y realizando una acción a través de efectores. Uno de los medios más importante para los agentes es Internet, ya que las aplicaciones que se crean se conectan a internet para realizar una función específica, es así como se crearon motores de búsqueda de información y sistemas para la construcción de portales web basados en agentes artificiales.

Hay dos grandes aportes de esta etapa: el primero es que se constató que los sistemas sensoriales no pueden generar información totalmente fidedigna del medio real y lo segundo los agentes artificiales permitieron la interacción de Inteligencia Artificial con otras ramas del saber cómo son investigación de operaciones, matemática, estadística, teoría de control, economía, entre otros.

Hacia la planificación de rutas (2001 a la actualidad)

La planificación de rutas de transporte ha sido abordada por medio de problemas del vendedor viajante conocido por sus siglas en inglés como TSP (Traveling salesman problem) que trata de encontrar la ruta mínima recorriendo los nodos de puntos geográficamente dispersos, problemas de rutas de vehículos cuyas siglas en inglés son VRP (Vehicle Routing Problem) donde un conjunto de vehículos tiene que recorrer todos los nodos asignados como árbol del problema en un determinado ámbito geográfico y últimamente hablamos de los problemas de rutas de vehículos con intervalo de tiempo cuyas siglas en inglés son VRPTW (Vehicle Routing Problem with Time Windows) donde la característica adicional está estipulada en que los clientes deben ser atendidos dentro de un intervalo de tiempo establecido (ventana de tiempo), solo una vez y por un solo vehículo (Konstantakopoulos et al., 2020).

Los algoritmos para la planificación de rutas tienen como finalidad explorar un espacio del cual se tiene poca o ninguna información “con el fin de encontrar una ruta que pueda conectar dos puntos con coordenadas para simplificar las tareas de búsqueda” (Ríos, 2015).

También se da el proceso de búsqueda y trazabilidad de rutas en el área de redes y telecomunicaciones, sin embargo difiere del problema de

investigación, en tanto que el enrutamiento en redes busca el intercambio de información sea de voz, video, texto o imágenes, que se dan por medio de redes informáticas o celulares (Amit & Sandeep Joshi, 2020), a diferencia de la presente investigación que pretende buscar una ruta óptima que una un nodo origen y uno destino, sin necesidad de intercambiar información entre nodos.

La planificación de rutas ha ido evolucionando desde métodos exactos, métodos heurísticos y ahora hablamos de métodos metaheurísticos, que permiten atender de mejor manera los problemas NP-duros, que se caracterizan por ser problemas que no admiten soluciones exactas eficientes, y que se han ido dando porque los sistemas blandos se han ido complejizando y abarcan múltiples variables (Korkmaz & Durdu, 2018).

Durante muchos años, la rama de Investigación de operaciones ha sido la encargada de abordar soluciones en planificación de rutas mediante métodos de búsqueda, sin embargo, con la mejora de los sistemas computacionales, que cada vez tienen mejor y mayor procesamiento, se ha incluido dentro de la inteligencia artificial, la creación de sistemas informáticos que permitan resolver este tipo de problemas mediante algoritmos más eficientes y complejos.

En tal sentido se podría indicar que la planificación de rutas cimienta las bases para el desarrollo del proceso de búsqueda de caminos.

Expansión de la Búsqueda de caminos (2011 a la actualidad)

La búsqueda de caminos se ha aplicado en diferentes entornos, para la detección de minas terrestres, servicio de personas mayores, transferencia de mercancías en fábricas, exploración de planetas, que pueden ser

abordados como localización, construcción de mapas y planificación de rutas.

La elección de un método, está relacionado con las características del problema a resolver, si la cantidad de rutas a cubrir es grande, si existen intervalos de tiempo como restricción para llegar a un determinado nodo, si hay una capacidad máxima de visitas al nodo, la cantidad de interconexiones entre los nodos, y si hay duplicidad de interconexiones entre nodos. En tal sentido, existen investigaciones que afirman que los métodos exactos tienen igual rendimiento que los métodos heurísticos cuando la cantidad de nodos a unir son pocos y no tienen muchas interconexiones, pero que difieren en forma sustancial cuando la cantidad de nodos aumenta al doble o al triple y la cantidad de interconexiones entre nodo es mayor (Putri et al., 2021).

En la actualidad, se han desarrollado agentes artificiales solo de software también conocido como softbots, así también como aquellos que incluyen hardware conocidos como robots que permitan realizar diferentes actividades del quehacer humano, y dentro de ellos ubicamos a los agentes de búsqueda, que en base a una estrategia realizan el proceso de encontrar una ruta óptima. Incluso actualmente hay una rama de la inteligencia artificial que se encargan del estudio de este tipo de agentes denominada "robótica".

En este contexto, encontramos a los robots móviles, cuya función es desplazarse en un entorno desconocido, para realizar una función específica. Para que los robots móviles puedan realizar su función, deben conocer su ubicación dentro del entorno y comprender su entorno, y a partir de ese conocimiento previo elegir un camino óptimo desde el punto inicial

hacia un punto objetivo teniendo en cuenta las restricciones para la elección del camino. Este problema se conoce en la literatura como búsqueda de caminos.(Korkmaz & Durdu, 2018).

Los agentes de búsqueda difieren de los robots móviles en el sentido que este último en cada paso tiene que sensor el entorno porque tiene un desconocimiento del mapa y de los obstáculos que hayan, en cambio el agente de búsqueda tiene un mapa de su entorno, constituido por un conjunto de nodos posibles que le permita hallar un camino específico.

Los agentes de búsqueda han ido evolucionando a medida que han ido evolucionando las estrategias que utilizan para realizar su función, así encontramos dentro de los agentes menos evolucionados a los que utilizan la estrategia de búsqueda no informada o búsqueda a ciegas, entre los que se encuentran: búsqueda preferente en anchura, búsqueda preferente en profundidad, búsqueda bidireccional, búsqueda iterativa, luego encontramos los que utilizan estrategias de búsqueda heurísticas, entre los cuales encontramos búsqueda voraz primero el mejor, A asterisco, con memoria acotada, y dentro los agentes más evolucionados encontramos los que utilizan algoritmos de optimización de búsquedas, como son ascensión de colinas, recocido simulado, haz local y algoritmos genéticos. Estos últimos algoritmos confluyen con los establecidos por los métodos metaheurísticos de la rama científica de investigación de operaciones.

A pesar que cada vez se realizan investigaciones para lograr un algoritmo cada vez más eficiente, no se ven investigaciones que promuevan el desarrollo de una metodología que permita uniformizar y ordenar el proceso de crecimiento de los sistemas de búsqueda de caminos en ambientes delimitados.

Capítulo III: Investigaciones realizadas en búsqueda de caminos y planificación de rutas

A través de revisión bibliográfica en diferentes bases de datos, se constata que el proceso de búsqueda de caminos y planificación de rutas se han estudiado a nivel local, nacional e internacional, conforme se visualiza en los siguientes párrafos.

Investigaciones previas a nivel mundial

En el artículo científico titulado "A Fast Bi-directional A* Algorithm Based on Quad-tree Decomposition and Hierarchical Map" (Yijun et al., 2021), se tuvo como objetivo proponer un algoritmo QH-A* (algoritmo bidireccional A* basado en una cuádruple descomposición y mapa jerárquico), algoritmo basado en el preprocesamiento que utiliza el método de descomposición del mapa en cuatro árboles, que a través de un mapa jerárquico reduce el espacio de búsqueda, y con el algoritmo A* bidireccional se realiza la búsqueda desde el comienzo y desde el final en el mapa jerárquico, siendo una investigación de tipo experimental tuvieron como resultado que el método de descomposición basado en cuatro árboles no ayuda al preprocesamiento, pero que sin embargo al combinarse con el uso del algoritmo de búsqueda bidireccional A* en un mapa jerárquico, se llega a reducir hasta el 70% del área de búsqueda en comparación del algoritmo A* original, y al reducir el área de búsqueda en forma efectiva mejora la eficiencia de la búsqueda, controlando el tiempo de cálculo de preprocesamiento dentro de un rango razonable, llegando como conclusión que el algoritmo QHA* sirve para la planificación de rutas en entorno

complejo y solo puede ser usado para mapa de cuadrícula, además, reduce el área de búsqueda en comparación con algoritmo A* original.

En el artículo científico titulado "A Multiobjective Large Neighborhood Search Metaheuristic for the Vehicle Routing Problem with Time Windows" (Konstantakopoulos et al., 2020), los autores tuvieron como objetivo implementar un algoritmo de búsqueda multiobjetivo de gran vecindario (MOLNS) para resolver problemas de enrutamiento de vehículos con intervalo de tiempo, utilizando el lenguaje de programación Python, evaluándose 56 instancias de referencia de Solomon con 100 clientes, así como en Instancias de referencia de Gehring y Homberger con 1000 clientes, siendo una investigación experimental descriptiva, donde cada instancia se ejecutó tres veces, mientras que el límite de tiempo computacional se estableció en 5 minutos, lo que es un tiempo razonable para que una empresa de logística obtenga la ruta de sus vehículos y la programación de sus entregas.

Las instancias de Salomón con clientes agrupados geográficamente, tienen resultados más eficientes (Desviación media de 0.30) que con los datos geográficos generados aleatoriamente (Desviación media de 3,37%), y los datos geográficos mixtos (Desviación media 3.47%). Por último, la desviación media para todas las instancias es inferior al 2,85%, lo que puede considerarse muy eficiente. En las instancias de referencia de Gehring y Homberger con 1000 clientes, la desviación media para todos los casos relacionados con el número de vehículos es del 6,63% y del 10,60%, para la distancia recorrida. Comparando la desviación media para todas las instancias en los casos de Salomón de 2,85% y la de Gehring y Homberger

de 6.63%, que indica que los resultados se ven afectados en pequeña medida cuando el número de clientes aumenta.

Los investigadores llegaron a la conclusión que el algoritmo MOLNS propuesto es suficiente para minimizar tanto el número de vehículos como la distancia total recorrida aplicando operadores de extracción e inserción manteniendo al mismo tiempo el concepto de optimización de Pareto. El algoritmo MOLNS ha demostrado ser eficiente tanto en la calidad de los resultados, ya que ofrece tres nuevas soluciones óptimas en el conjunto de datos de Solomon y produce resultados casi óptimos en la mayoría de los casos, en términos de tiempo computacional, ya que, incluso en casos de hasta 1000 clientes, se obtienen resultados de buena calidad en menos de 15 min.

En una investigación de tesis realizada para obtener el grado de Maestro en la Universidad Católica de Chile titulada: "Compilación en programación de conjuntos de respuestas para el problema de búsqueda de caminos con múltiples agentes" (Gómez, 2020), se tuvo como objetivo encontrar todos los caminos libres de conflictos que resuelva el problema de búsqueda de caminos multiagente en una grilla, minimizando el costo total de los caminos o el makespan de la búsqueda, utilizando dos tipos de solvers: un solver basado en búsqueda y otro basado en compilaciones. El solver basado en la búsqueda está programado para búsqueda heurística y utiliza el algoritmo Conflict-Based Search (CBS) que funciona minimizando la suma del largo de los caminos entre todos los agentes. Este valor está definido como la suma de costos de una solución. El solver basados en compilaciones, utiliza compilaciones de Satisfacción Booleana (SAT) o Programación de Conjuntos

de Respuestas (ASP). minimizando el camino más largo entre todos los agentes, es decir, busca optimizar el tiempo total necesario para que todos los agentes lleguen a sus respectivos objetivos. Este valor está definido como el makespan de una solución. Dado que MAPF es un problema combinatorio, se utiliza la compilación de Programación de Conjuntos de Respuestas (ASP) que resuelven la variante de suma de costos de MAPF sobre grillas 4-conectadas, haciendo que el número de cláusulas totales (después de la instanciación) crezca linealmente con el número de agentes. Además, el objetivo de optimización es tal que su tamaño después de la instanciación no depende del tamaño de la grilla. El autor concluye que el Answer-Set Programming (ASP) es un enfoque viable para resolver problemas MAPF, especialmente en problemas con alta congestión. Los solvers basados en ASP permiten encontrar soluciones en menor tiempo y a mayor escala, sacrificando menos del 1% de la calidad media de la solución. Por tanto, no se debe centrar en soluciones óptimas en cuanto a costos sino óptimas en cuanto a makespan para problemas MAPF.

En el artículo de investigación "An Efficient Optimal Path Finding for Mobile Robot Based on Dijkstra Method" (Alyasin et al., 2019) se plantea como objetivo desarrollar un robot móvil inteligente que utilice el algoritmo de Dijkstra que elija la ruta más corta entre dos nodos en la búsqueda gráfica, para llegar a un lugar objetivo en una red de carreteras, siendo una investigación experimental, que trabajó con un croquis de una red de carreteras que consta de 10 nodos, que tiene 1 nodo origen y 3 nodos destino, con 21 líneas conectadas entre ellos que poseen diferentes pesos, y un robot móvil del tamaño de un carro de juguete a quien se le dotó de

programación a través del algoritmo Dijkstra para transitar por el croquis de la red de carreteras. El método Dijkstra que se aplicó en esta investigación consiste en asignar el valor 0 al vértice inicial y todos los demás se asigna un valor infinito, luego se identifican los vértices que están conectados al vértice inicial, luego se calcula la distancia de cada vértice al inicial colocándose como etiqueta en las líneas de conexión, luego si la distancia (costo de la ruta o peso de la ruta) es menor que la que ya está asignada, se tacha esa distancia y se reemplaza por la menor, por último, si está con valor infinito, se escribe la nueva distancia. Se repite el paso 4 hasta que se llegue al vértice de destino. En los resultados obtenidos, se asignó como nodo inicial el vértice A, como nodo destino el vértice H, encontrando que el costo de la ruta óptima encontrada es 10, siendo el camino más corto desde el punto A, luego al punto C, luego al punto F hasta llegar al punto (H). La forma para calcular el tiempo es mediante una regla de tres simple, donde cada 8 unidades de peso corresponden a 1 segundo. El tiempo entre el nodo (A) y nodo (C) = 0.375 S y ángulo 0. El tiempo entre el nodo (C) y el nodo (F) = 0,125 S y el ángulo 0. El tiempo entre el nodo (F) y el nodo (H) = 0,75 S en un ángulo de +45. La ruta encontrada desde el nodo A al nodo H, tomó 1.25 segundos con 10 unidades de peso. Los autores concluyeron que el algoritmo Dijkstra permitió alcanzar el objetivo de ubicar la ruta más corta en el menor tiempo y con el menor costo, en el croquis de la red de carreteras a través de un robot móvil de tamaño pequeño que transitó por el croquis con alto grado de precisión.

En el artículo de investigación titulado: "Apply A* Search for Robot Path Finding" (Angkuldee et al., 2019) se plantean como objetivo crear un

algoritmo para que un robot planifique la ruta más corta desde una posición inicial hasta un destino para lo cual se verificó la eficacia de cuatro métodos distintos: algoritmo de ángulo de dirección, reenvío de paquetes como algoritmo, algoritmo de búsqueda A* con GPS y algoritmo A* sin utilizar GPS. El tipo de investigación fue experimental descriptivo, probando cada uno de los cuatro métodos diferentes de búsqueda y evaluándolos a través de tres criterios: automatización con peso de 0.3, velocidad con peso 0.2 y precisión con peso 0.5, en cada criterio se le asignó una de las tres puntuaciones disponibles: 1 bajo, 2 medio y 3 alto, hallando los promedios ponderados por cada método. El algoritmo de ángulo de dirección resultó ser altamente automatizado y altamente rápido, pero fue bajo en precisión, debido a que el robot fijaba la dirección en el objetivo y no en los obstáculos, lo que podría crear un bucle infinito por intentar moverse directamente al destino sin tener en cuenta el obstáculo. El algoritmo de punto de control con envío de paquetes, obtuvo un alto puntaje en velocidad y alto en precisión, pero bajo en automatización ya que requiere del trabajo humano para colocar los puntos de control. El algoritmo de búsqueda A* con GPS, obtuvo alto puntaje en automatización y precisión y medio en velocidad, por tanto, es el más eficiente con mayor puntaje promedio, pero no se puede aplicar a interiores de un edificio. El algoritmo de búsqueda A* con cálculo de la distancia de viaje, obtuvo alto puntaje en automatización, medio en velocidad y medio en precisión, ya que calcula su posición actual de acuerdo a la distancia de recorrido y al ángulo de giro, sin embargo, esto puede generar unas leves imprecisiones. En conclusión, el mejor algoritmo para el desplazamiento de un robot en un lugar con obstáculos en exteriores es el algoritmo A* con GPS, y en interiores es el algoritmo A* con cálculo de

distancia de viaje, teniendo en consideración que el algoritmo A* calcula la distancia más corta entre dos puntos que utiliza una función heurística para encontrar la celda de vecina a la celda actual.

En el artículo científico titulado "Shortest Path Search Simulation on Busway Line using Ant Algorithm" (Prmono et al., 2019) se tuvo como objetivo diseñar un software de simulación de búsqueda de la ruta más corta en una línea de bus, utilizando el algoritmo Colonia de Hormiga y utilizando como método de desarrollo de software la iteración en cascada. Se plantea una investigación de tipo experimental donde se usará 100 hormigas para rastrear todo el camino en un carril de la vía de autobuses, dejando caer feromonas en cada camino, cuanto más corto es un camino, más feromonas están acumuladas en cada iteración. Después de pasar por un número de iteraciones, la feromona se acumulará en el camino más corto y todas las hormigas seguirán el camino más corto. Según las pruebas de la aplicación, se puede ver que el algoritmo "Colonia de Hormiga" permite hallar la ruta más corta desde un número dado de terminales. Para un máximo de 30 terminales, el algoritmo solo requiere 28 segundos para hallar la ruta más corta. Sin duda, esto es mucho más rápido para buscar la ruta más corta, que cuando se usa un esquema de ataque de fuerza bruta. El tiempo de ejecución es proporcional al número de puntos. Cuantos más puntos o terminales, más tiempo tarda el algoritmo "Hormiga" en encontrar la ruta más corta. Se llegó a la conclusión de que la aplicación encuentra la ruta más corta en la línea de carriles en Medan utilizando el algoritmo "Hormiga" para un máximo de 30 terminales con tiempo de ejecución de búsqueda alrededor de 28,53

segundos, dependiendo de los puntos terminales y la especificación de la computadora utilizada para ejecutar la solicitud.

En la investigación titulada "Engaging undergraduates in artificial intelligence: Path finding in game development" (Liburd & Boumedine, 2018), se tuvo como objetivo diseñar y desarrollar una interfaz que visualice el espacio de búsqueda y las rutas de solución de un problema de búsqueda de caminos, utilizando tres algoritmos: Dijkstra, A* y Greedy Best First. El trabajo fue de tipo experimental, utilizando los algoritmos de búsqueda: Dijkstra, A* y Greedy best frist en mapas de cuadrículas 2D con obstáculos, con la visualización del comportamiento de los algoritmos y la cantidad de nodos expandidos para alcanzar el nodo objetivo, basándose en 8 direcciones 4 direcciones cardinales y 4 diagonales a través de la distancia euclidiana. Los resultados se obtuvieron en dos escenarios con el mismo plano, pero diferente ubicación de obstáculos y diferentes nodos de partida y llegada. En el primer escenario, los tres algoritmos encontraron la ruta más corta, el algoritmo A* expandió 158 nodos, el método avaro primero el mejor expandió 806 nodos, y el método Dijkstra expandió 1606 nodos. En el segundo escenario la ruta más corta fue encontrada por A* y Dijkstra, pero no por el método avaro primero el mejor, a pesar que este último fue el que expandió menor cantidad de nodos, su ruta encontrada fue mucho más larga. Esto se debe porque primero expande el nodo con una distancia estimada más pequeña a la meta, aunque esto a veces puede seleccionar un camino no óptimo. Todos los resultados encontrados están conforme a la literatura. Los investigadores concluyeron que los estudiantes pudieron captar rápidamente el concepto de optimalidad, y admisibilidad, inherentes

a las funciones heurísticas, cuando la heurística es admisible hay la certeza que encuentra la ruta más corta, a través de la ejecución de los tres algoritmos: Dijkstra, A* y Ávaro primero el mejor (Greedy best first), viéndose que el que expande menor número de nodos en la mayoría de escenarios es el de búsqueda avara, sin embargo no siempre encuentra la ruta más corta ya que se inclina por buscar en las rutas que están más cerca al objetivo sin analizar el coste de distancias desde el inicio. El algoritmo A* es el más seguro ya que siempre encontró la ruta más corta, sin embargo, en rendimiento tuvo un valor intermedio, ya que en promedio fue el que expandió nodos en cantidades medias.

En el artículo científico titulado: "Comparison of optimal path planning algorithms" (Korkmaz & Durdu, 2018) se planteó como objetivo comparar los algoritmos de búsqueda que involucren métodos de análisis estadísticos: A*, Algoritmo genéticos (GA), RRT (Árbol aleatorio de exploración rápida), B-RRT (Bidireccional RRT), Probabilistic Roadmap (PRM), para poder determinar cuál es más adecuado en cuanto a los criterios de tiempo y ruta más corta en el desplazamiento de un robot móvil autónomo. La investigación fue de tipo experimental y utilizó un algoritmo de mapeo tipo SLAM, llamado gmapping que construye un mapa del entorno, a partir de los datos de sensores y odometría del robot, para lo cual se ha establecido dos coordenadas X, Y una para el inicio y otra para la meta. Con el algoritmo A* se obtuvo un tiempo de 76,96 h y una distancia de 368 cm, con el algoritmo B-RRT se obtuvo un tiempo de 1.98 h y 434 cms., con el algoritmo genético 24,62 h y 524 cms., el algoritmo PMR 0,95 h y 397 cms., RRT 2.38 h y 446 cms. Se realizó un factor de eficiencia en base a la multiplicación de

tiempo y la distancia obteniendo para A* 1, para B-RRT 32.96, para GA 2.196, para PMR 75.09 y para RRT 26.68. Los investigadores llegaron a la conclusión que el algoritmo A* encuentra la ruta más corta. Sin embargo, se observa que la eficiencia de tiempo de este algoritmo es muy baja por su largo tiempo de cálculo. Por otro lado, el algoritmo PRM es el método más adecuado en términos de tiempo transcurrido. Además de esto, la longitud de la ruta del algoritmo está más cerca del algoritmo A*.

En el artículo científico titulado "Research and Application of Path-finding Algorithm Based on Unity 3D"(H. Zhang et al., 2016), se tuvo como objetivo demostrar la aplicación del algoritmo A* en la búsqueda de rutas en la producción de un videojuego usando Unity 3D. El algoritmo A* es el método más eficaz para hallar el camino más corto en una cuadrícula estática, hay una buena aplicación en la estrategia de búsqueda del juego, así como a la búsqueda de la ruta en la cuadrícula. El algoritmo de cuadrícula de navegación Navmesh se usa más en la búsqueda de caminos en la escena del juego y evitación dinámica de obstáculos de la escena. Se llega a la conclusión que se puede aplicar el algoritmo A* en juegos para resolver problemas de búsqueda de caminos. Con el desarrollo de los juegos gráficos, las escenas del juego son cada vez más complejas, un algoritmo único no se puede aplicar a todos los escenarios de búsqueda de rutas. Entonces para la aplicación en diferentes escenas, todavía necesita mejorar el algoritmo A*.

En la tesis de pregrado para obtener el título en Ingeniería en Organización Industrial en la Universidad de Valladolid titulada "Algoritmos heurísticos y

metaheurísticos basados en búsqueda local aplicados a Problemas de Rutas de Vehículos” (Fernandez, 2016) se aborda el problema de planificación de rutas, y se plantea como objetivo “realizar un compendio de diferentes métodos heurísticos y metaheurísticos que ayude a resolver problemas de enrutamiento de vehículos, así como obtener una guía con procedimientos confiables que permitan llegar a resultados confiables” (Fernandez, 2016). La investigación fue de tipo experimental y se obtuvo como resultados que el método GRASP obtiene mejores tiempos el 66,17% de las veces con respecto al método Simulated Annealing. Con respecto a la comparación del método GRASP con un método exacto, se obtiene que el 84,62% de los casos se obtuvo mejores resultados con el método GRASP. “Adicionalmente, se evidencia que para mapas pequeños ambos métodos ofrecen resultados equivalentes, pero que a medida que va aumentando el tamaño del mapa el método GRASP es mucho más eficiente” (Fernandez, 2016). Se llega a la conclusión que el método GRASP consiguió mejores resultados con respecto a la calidad y al tiempo de cálculo. Este método está constituido por un método avaro aleatorizado de la heurística de Inserción Secuencial. Los algoritmos exactos calculan todas las soluciones posibles hasta que se alcanza la mejor, pero, sin embargo, se desempeña muy mal en términos de tiempo computacional, incluso cuando se trata de pocas instancias.

Investigaciones previas a nivel nacional

En el artículo de investigación titulado “Evaluación de un sistema de búsqueda de rutas de evacuación eficientes de un establecimiento usando el algoritmo D estrella (D*)” (Pariona, 2019) se tuvo como objetivo: desarrollar un sistema informático de evacuación ante posibles desastres naturales a través de la creación de un simulador que brinde la ruta de salida más adecuada y segura a un área libre de peligro usando el algoritmo de búsqueda D estrella (D*), fue una investigación experimental donde se establecieron tres escenarios:

“El primer escenario estuvo constituido de 3 variaciones del plano con respecto a su tamaño y con 10 obstáculos iniciales.

El segundo escenario estuvo constituido de 3 variaciones del plano con respecto a la cantidad de obstáculos iniciales sobre un mismo plano de escala 5.

El tercer escenario estuvo constituido de seis escenarios propuestos. Así, se generaron cinco ejecuciones sobre un plano de escala 10”. (Pariona, 2019)

Con esto se comprobó la consistencia del tiempo de ejecución obtenido en las ejecuciones realizadas. “En el primer escenario, el tiempo de ejecución aumenta conforme aumenta la escala en el plano virtual. En el segundo escenario, el tiempo de ejecución incrementa en baja proporción de acuerdo al incremento en la cantidad de obstáculos. En el tercer escenario, el tiempo fue variable en las 5 iteraciones consecutivas de la ejecución de la simulación, cuya variación no excede a los 16 milisegundos” (Pariona, 2019). Se llega a la conclusión que, el algoritmo D* encuentra el camino de búsqueda en solo 22 milisegundos, lo que permitiría inferir que en un proceso real de evacuación el usuario de la aplicación conocería de forma inmediata el

camino más corto por el cual evacuar, en menos de un segundo. También se logró computarizar el tiempo empírico que le tomaría el algoritmo recalcularse el camino más corto en caso se evidencie la presencia de obstáculos que interfirieran en el recorrido, tomándole al algoritmo 3 milisegundos en recalcularse una ruta de escape.

En la tesis para obtener el grado en Ingeniería Mecatrónica en la Universidad Nacional de Ingeniería titulada "Desarrollo de un software para la gestión de vehículos con generación y planificación automática de rutas"(Pumaricra Rojas, 2019) se planteó como objetivo "desarrollar una aplicación informática para el enrutamiento de vehículos con generación automática de rutas basado en el uso de algoritmos de planificación de rutas, visión computacional y redes" (Pumaricra Rojas, 2019), siendo una investigación experimental llegó a la conclusión que el software fue capaz de manipular al vehículo de guiado automático, planear su movimiento de acuerdo con rutas obtenidas a partir de una vista de planta mediante el uso de algoritmo, y esto lo hace a través de la generación de un archivo de texto correspondiente a las imágenes importadas de los planos pudiendo llevar el vehículo de un nodo a otro con comunicación TCP de internet inalámbrico..

En la tesis de pregrado de la Universidad Nacional Mayor de San Marcos, titulada "Aplicación de un modelo de ruteo de vehículos para optimizar el recorrido en el servicio de visitas turísticas" (Acuña, 2018), tuvo como objetivo: "aplicar un método exacto para la planificación de rutas en un problema de enrutamiento de vehículo periódico para el sector turístico", siendo una investigación de tipo experimental, se llegó a la conclusión que

en el caso de del sector turístico no basta con planificar rutas sino se debe aplicar técnicas más avanzadas, heurísticas y metaheurísticas para generar soluciones con mejores resultados,

En la tesis para obtener el título de Ingeniero de Sistema en la Universidad Peruana Unión, titulada "Diseño e implementación de un circuito turístico inteligente en la Región Puno mediante la metaheurística Búsqueda Tabú" (Mendoza, 2017), se tuvo como objetivo "diseñar y desarrollar un circuito turístico inteligente para la Región Puno mediante la Búsqueda Tabú" (Mendoza, 2017), siendo una investigación de tipo experimental descriptivo donde se utilizó tres algoritmos de búsqueda: Heurística de Construcción (C), Heurística de Mejoría (M) y la Metaheurística Búsqueda Tabú (TS) que fueron aplicados en cinco mapas, el primero tenía 26 nodos, el segundo 42 nodos, el tercero 17, el cuarto 29 nodos y el quinto 194 nodos, Se comparó la desviación porcentual del costo de los tres algoritmos. Con la heurística de construcción, no se alcanzó la mejor solución en ninguno de los mapas. Con la heurística de mejoría solo se logró igualar a la mejor solución en un mapa, de cinco mapas en total. Al final se concluye que la metaheurística Búsqueda Tabú implementada con la estrategia de diversificación, fue validada con 5 mapas (instancias artificiales) igualando al 60% de las mejores soluciones encontradas hasta la fecha, "siendo sus datos de entrada obtenidos de agencias turísticas del distrito de Juli, y dando como resultado un ciclo hamiltoniano que conformaría un circuito turístico".

Investigaciones previas a nivel local

En una tesis para obtener el título profesional en Ingeniería de Sistemas de la Universidad Señor de Sipán denominada "Desarrollo de un planificador de rutas para recojo de desechos sólidos utilizando algoritmo de Bellman Ford" (Reyes, 2018), se planteó como objetivo "desarrollar un planificador de rutas para el recojo de basura en el Distrito de Chiclayo utilizando el Algoritmo de Bellman-Ford" (Reyes, 2018). Fue una investigación experimental donde los resultados de la planificación de rutas demuestran que, a mayor cantidad de metros recorridos, así como mayor cantidad de nodos destino, mayor es el tiempo de ejecución, "así cuando hay un solo nodo destino con 799 metros a recorrer el tiempo de ejecución es de 0.0586 milisegundos y en cambio cuando hay 30 nodos destino y 5462 mts por recorrer, el tiempo de ejecución es de 2.2962 milisegundos". Llegó a la conclusión que el algoritmo utilizado permitió realizar la planificación de rutas para el recojo de basura, usando la geolocalización en una API de Google maps que permitiera visualizar el plano de la ciudad de Chiclayo que serían utilizados por los vehículos recolectores de basura. El tiempo para mostrar las rutas optimas varía según el procesador del computador donde se ejecute.

En la tesis de pregrado denominada "Desarrollo de un planificador de rutas para recojo de desechos sólidos en el distrito de Chiclayo utilizando algoritmo de Dijkstra"(Anton, 2018) se trazó como objetivo implementar un sistema informático de planificación de rutas para recojo de basura en el distrito de Chiclayo utilizando dicho algoritmo, siendo una investigación de tipo experimental que obtuvo como resultados que la eficiencia del

algoritmo utilizado depende del procesador del computador en que se ejecute, teniendo en comparación la diferencia en milisegundos, el primer tiempo de 1.52 seg con un segundo tiempo de 1.66 seg. sin embargo, cuando hablamos de longitud de la trayectoria de búsqueda encontrada es invariable para 7 casos de prueba, obteniendo los mismos resultados en ambos computadores” (Anton, 2018) , por lo que se asume que la longitud de la trayectoria es independiente del computador en que se ejecute, y lo que si varía es el tiempo de búsqueda. El autor llegó a la conclusión que el algoritmo Dijkstra ayuda en la planificación de rutas utilizando como valores iniciales un Grafo ponderado que utilice una matriz de adyacencia, tomando en cuenta los puntos de búsqueda y el punto inicial, utilizando el lenguaje PHP para poder implementar el sistema informático” (Anton, 2018) .

En la tesis de pregrado denominada “Desarrollo de un sistema para la optimización de rutas de trabajo utilizando el algoritmo de Dijkstra y diagramas de Voronoi” (Marchena, 2015) se plateó como objetivo implementar un sistema informático que permita optimizar la planificación de rutas utilizando el algoritmo de Dijkstra y diagramas de Voronoi para la empresa EPSEL S.A para el departamento de medición del fluido eléctrico (Marchena, 2015), siendo una investigación de tipo experimental con pre test y post test, se obtuvieron resultados de mejora al realizar la comparación de los resultados del pre test con la manera de trabajo tradicional de los empleados (notificadores, verificadores e inspectores) y los resultados del post test con el trabajo realizado por los empleados utilizando el sistema informático, verificándose que los notificadores tienen una mejora de la eficiencia de su trabajo de un 16% para un día de semana y 13% para un día

de fin de semana, y para los verificadores mejora su eficiencia en un 13.78% para un día de semana y 12% para un día de fin de semana , y con respecto a los inspectores su eficiencia de trabajo mejoró en un 18% para un día de semana y 24% para un día de fin de semana (Marchena, 2015), aun cuando la localización y posicionamiento de los predios muchas veces fueron inexactos por las actualizaciones propias de la api de google maps utilizada en esta investigación para la visualización del plano de la ciudad de Chiclayo. Se llegó a la conclusión que el algoritmo de Dijkstra permite hallar las rutas óptimas de búsqueda para que los empleados de la empresa EPSEL puedan realizar su trabajo teniendo como datos de entrada los puntos de búsqueda, aplicándose el algoritmo de Voronoi sobre plano al cual previamente se le realizó la sectorización de zonas para que los empleados tengan en cuenta los límites en el mapa del espacio geográfico de donde deben realizar su trabajo” (Marchena, 2015).

En la tesis de pregrado denominada “Sistema de búsqueda heurística de la Universidad nacional Pedro Ruíz Gallo utilizando el algoritmo A Star” (Quiroz & Ramirez, 2019) se planteó como objetivo Imp elementar un sistema de búsqueda heurística en la Universidad Nacional Pedro Ruiz Gallo, siendo una investigación de tipo descriptiva logro identificar que el algoritmo A star no tiene en cuenta la distribución de los nodos o el coste de las aristas. Se estableció 435 nodos para las vías transitables de la Universidad Nacional Pedro Ruiz Gallo, que permitió realizar búsquedas utilizando una base de datos en MySql que contiene principalmente cuatro tablas: arista, nodo, destino y usuario para la administración del grafo y de los usuarios del

sistema de búsqueda heurística. Se realizaron 27 pruebas de los módulos del sistema realizados en el catálogo de pruebas que dieron como resultado que funcionaban correctamente cada uno de los componentes del software, que permitió localizar las principales oficinas de la Universidad Nacional Pedro Ruiz Gallo con mayor facilidad gracias a las rutas generadas por el algoritmo elegido.

Capítulo IV: Conceptos relacionados con búsqueda de Caminos

Se abordarán los conceptos relacionados con las variables de la investigación como son: Búsqueda de caminos, agente artificial, así como los diferentes algoritmos que se utilizan en las diversas estrategias de búsqueda de caminos.

Búsqueda de caminos

La búsqueda de caminos, es un proceso realizado con algoritmos de Inteligencia Artificial para encontrar la ruta más corta o el camino más corto de una ubicación a otra dentro de un mapa o un plano (Angkuldee et al., 2019). Se puede aplicar a un agente artificial para decidir el camino más corto desde su posición actual hasta el destino de forma rápida y correcta.

Visto técnicamente, para poder realizar una búsqueda, se debe conformar un espacio de solución que comúnmente es representado por un grafo, donde a partir de un nodo inicial se va realizando acciones para expandir los nodos hasta llegar a un nodo final. El agente es el que decide qué hacer en cada nodo, examinando una secuencia posible de acciones que le conducen a nuevos nodo hasta llegar al objetivo. Por tanto se podría decir que la búsqueda es el proceso de hallar la secuencia de nodos que conforman la solución de un problema teniendo como datos de entrada un espacio de solución, un nodo inicial y un nodo final, y siguiendo las acciones dadas por un algoritmo (Russell & Norvig, 2010).

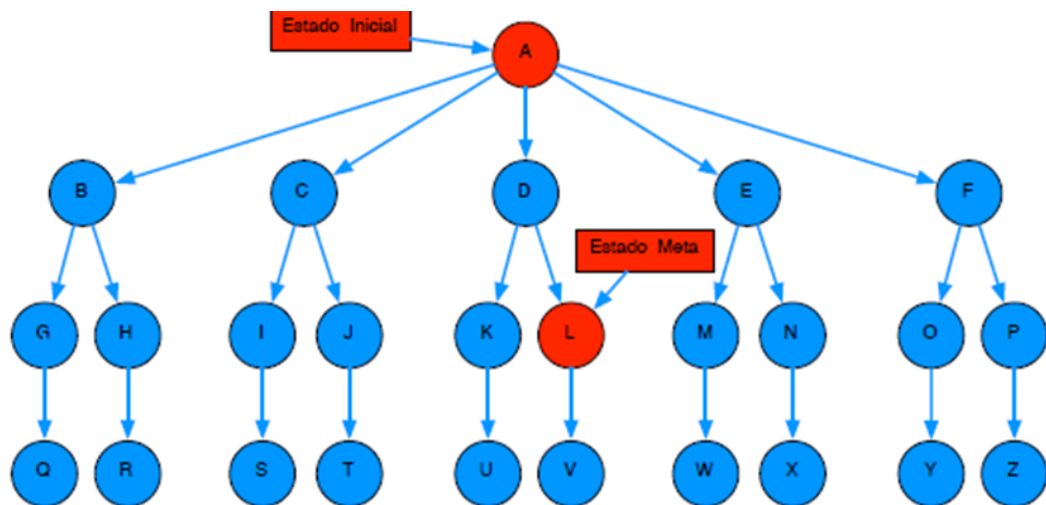
Durante el proceso de búsqueda el algoritmo va construyendo un árbol que almacena en cada nodo, información sobre los nodos que se van obteniendo al realizar las acciones. La copa del árbol corresponde al nodo inicial y sus ramas a los nodos hijos que resultan de realizar acciones a ese nodo. Se continúa ramificando el árbol, aplicando el mismo razonamiento a cada hijo (sin repetir

nodos en cada camino), hasta que llegue al nodo final, y es aquí donde se crea el camino.

El camino viene definido por la secuencia de nodos conectados a una secuencia de acciones que une el estado inicial con el nodo objetivo.

Figura 8:

Ejemplo de Árbol de búsqueda



Estrategias de búsqueda de caminos

La estrategia de búsqueda consiste en definir cómo se expandirán los nodos dentro del árbol de búsqueda, utilizando un concepto de frontera, que es donde se almacenan los nodos del árbol que han sido generados, pero no expandidos. Al comienzo la frontera únicamente tendrá el nodo inicial. Los nodos generados son aquellos que están en lista de espera a ser expandidos, y los nodos expandidos son aquellos a los cuales se han generado sus nodos hijos. (Russell & Norvig, 2010).

La forma en cómo se guardan los nodos en la frontera depende del algoritmo, puede ser como una pila o como una cola.

El algoritmo general de una estrategia de búsqueda es:

Para comenzar necesita como datos de entrada:

Nodo inicial

Lista de nodos

Lista de nodos finales (objetivos)

Grafo con el espacio de solución o funciones que permitan determinar las acciones

Durante el Funcionamiento:

Árbol de búsqueda

Pila o cola que representa la frontera

Al terminar se muestra:

Lista con secuencia de nodos

El algoritmo de búsqueda corresponde a una estrategia de búsqueda que es el criterio para elegir cuál será el siguiente nodo a expandir, siendo las dos estrategias más conocidas; la búsqueda a ciegas y la búsqueda informada.

Estrategias de búsqueda a ciegas

Esta estrategia se utiliza cuando no se tiene información adicional acerca de los nodos. La única información es la que proporciona la formulación del problema, y para determinar si se ha llegado al objetivo, se debe verificar en cada nodo generado si corresponde al nodo objetivo.

Para describir un problema de búsqueda a ciegas, se propondrá un ejemplo, supongamos que hay tres caminos que salen de A (C, D y E), pero ninguno directo a B, y el objetivo es llegar a B. El agente no tiene un mapa del lugar,

así que podría tomar cualquier ruta ya que no sabría cuál de ellas es la más directa para llevarlo a su objetivo. El camino viene determinado por la sucesión de nodos conectados a una serie de actividades que junta el estado inicial con el nodo objetivo, por eso se le conoce como “a ciegas” o “a tientas”.

A continuación, se describirán los algoritmos más utilizados bajo este enfoque:

Búsqueda Primero en Amplitud

La búsqueda primero en amplitud requiere guardar cada nivel del árbol transversal para más procesamiento. En consecuencia, esto conduce a un mayor uso de la memoria y agotará fácilmente la memoria disponible en sistemas con poca memoria (Kolhe et al., 2018). Además, el tiempo consumido por el algoritmo es directamente proporcional al trayecto entre el nodo objetivo y el nodo raíz.

La búsqueda preferente en amplitud o anchura, también conocida como BFS (Breadth-first search) halla el camino más corto desde un vértice de origen eligiendo un nodo de un grafo como elemento raíz, a todos los demás vértices que tienen como punto de partida el nodo raíz. Posteriormente, para cada uno de los nodos vecinos se expanden sus respectivos vecinos adyacentes, y así se continua hasta que se recorra todo el árbol en busca del modo objetivo (Kolhe et al., 2018).

Figura 9:

Pseudocódigo de primero en amplitud

Pseudo code:

```
bfs(Graph, start_node, end_node)
{
  Create Empty List VISITED
  Create Empty QUEUE
  VISITED.add(start_node)
  QUEUE.enqueue(start_node)
  current = start_node
  while current != end_node
  if v not in VISITED
    current = QUEUE.dequeue()
    for all neighbours n of current in Graph
      if n not in VISITED
        QUEUE.enqueue(n)
      end for
    VISITED.add(current)
  end if
end while
}
```

Fuente: (Kolhe et al., 2018)

Búsqueda Primero en profundidad

En la búsqueda preferente en profundidad, se exploran los bordes del vértice descubierto recientemente. Este proceso continúa hasta que hayamos descubierto todos los vértices accesibles desde el vértice fuente original. Si quedan vértices no descubiertos, se selecciona uno de los vértices como nueva fuente y se repite la búsqueda desde esa fuente. Todo este proceso se repite hasta que se descubren todos los vértices. Al igual que en la búsqueda primero en amplitud, siempre que se descubre un vértice durante un escaneo de la lista de adyacencia de un vértice o ya descubierto, la búsqueda en profundidad primero registra este evento estableciendo el campo predecesor de $\pi[v]$ en u . A diferencia de la búsqueda primero en

amplitud, cuyo subgrafo predecesor forma un árbol, el subgrafo predecesor producido por una búsqueda en profundidad puede estar compuesto por varios árboles, porque la búsqueda puede repetirse desde múltiples fuentes. El subgrafo predecesor de una búsqueda en primer lugar se define, por lo tanto, de forma ligeramente diferente al de una búsqueda en primer lugar: dejamos

$$G_x = (V, E), \text{ donde}$$

$$E = \{([v], v) : v \text{ and } [v] \text{ NIL}\}$$

El subgrafo predecesor de una búsqueda en profundidad forma un bosque en profundidad compuesto por varios árboles en profundidad. Las aristas en E_π se denominan aristas de árbol. Como en la búsqueda en amplitud, los vértices se colorean durante la búsqueda para indicar su estado. Cada vértice es inicialmente blanco, aparece en gris cuando se descubre en la búsqueda y ennegrecido cuando se termina, es decir, cuando se ha examinado completamente su lista de adjuntos (Cormen et al., 2001).

Figura 10:

Pseudocódigo de primero en profundidad

Pseudo Code:

```
dfs(Graph, start_node, end_node)
{
    Create Empty List VISITED
    Create Empty STACK
    VISITED.add(start_node)
    STACK.pust(start_node)
    current = start_node
    while current != end_node
    current = STACK.top()
    STACK.pop()
    for all neighbours n of current in Graph
    if n not in visited
        STACK.push(n)
        VISITED.add(n)
    end if
    end for
    end while
}
```

Fuente: (Kolhe et al., 2018)

Búsqueda de Costo uniforme

Cuando todos los costos de las aristas del grafo de búsqueda son iguales, la búsqueda primero en amplitud es óptima ya que siempre expande el nodo menos profundo. Mediante una simple extensión, se puede configurar un algoritmo que sea óptimo con cualquier función de costo escalonado. En lugar de expandir el nodo más superficial, el algoritmo de costo uniforme expande el nodo n con el costo de ruta más bajo $g(n)$. Esto se hace almacenando la frontera como una cola de prioridad ordenada por el costo de la arista de la ruta.

Figura 11:

Pseudocódigo de Costo uniforme

```
function UNIFORM-COST-SEARCH(problem) returns a solution, or failure
  node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  frontier ← a priority queue ordered by PATH-COST, with node as the only element
  explored ← an empty set
  loop do
    if EMPTY?(frontier) then return failure
    node ← POP(frontier) /* chooses the lowest-cost node in frontier */
    if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
    add node.STATE to explored
    for each action in problem.ACTIONS(node.STATE) do
      child ← CHILD-NODE(problem, node, action)
      if child.STATE is not in explored or frontier then
        frontier ← INSERT(child, frontier)
      else if child.STATE is in frontier with higher PATH-COST then
        replace that frontier node with child
```

Fuente: (Russell & Norvig, 2010).

Además del orden de la cola por costo de ruta, existen otras dos diferencias significativas con respecto a la búsqueda en amplitud. La primera es que la prueba de objetivo se aplica a un nodo cuando se selecciona para expansión en lugar de cuando se genera por primera vez. La razón es que el primer nodo objetivo que se genera puede estar en una ruta subóptima.

La búsqueda de costos uniformes no tiene en cuenta la cantidad de pasos que tiene una ruta, sino solo por su costo total. Por lo tanto, se atascará en un bucle infinito si hay una ruta con una secuencia infinita de acciones de costo cero, por ejemplo, una secuencia de acciones NoOp. La integridad está garantizada siempre que el costo de cada paso exceda una pequeña constante positiva ϵ .

La búsqueda de costo uniforme se guía por los costos de la ruta en lugar del nivel de profundización. Sea C el costo de la solución óptima, cada acción costaría al menos e , debido a que la búsqueda de costo uniforme primero explora grandes árboles que involucren pasos pequeños antes de explorar caminos que involucren pasos grandes y quizás útiles. Cuando todos los costos de paso son iguales, la búsqueda de costo uniforme es similar a la búsqueda preferente en amplitud " (Russell & Norvig, 2010), excepto que esta última se detiene tan pronto como genera un objetivo, mientras que la búsqueda de costo uniforme examina todos los nodos en la profundidad del objetivo para ver si uno tiene un costo menor; por lo tanto, la búsqueda de costo uniforme hace estrictamente más trabajo al expandir los nodos a una profundidad innecesaria.

Búsqueda en Profundidad limitada (Depth-limited)

El vergonzoso fracaso de la búsqueda en profundidad en espacios de estados infinitos puede aliviarse proporcionando la búsqueda preferente en profundidad con un límite predeterminado " l ", de tal manera que en la profundidad " l " los nodos se exploran, pero no se expanden sus sucesores. Poner un límite como máximo nivel de profundidad resuelve el problema del camino infinito. Desafortunadamente, también introduce un riesgo de incompletitud si elegimos $l < d$, es decir, si el objetivo está a nivel mayor de profundidad que el límite de profundidad. Esto es probable cuando se desconoce d . La búsqueda de este tipo tampoco será eficiente si

elegimos $Q > d$. Su complejidad temporal es $O(b^d)$ y su complejidad espacial es $O(bd)$. La búsqueda en profundidad puede verse como un caso especial de búsqueda con profundidad limitada con $l = \infty$ (Russell & Norvig, 2010)

Figura 12:

Pseudocódigo de Profundidad limitada

```

function DEPTH-LIMITED-SEARCH(problem, limit) returns a solution, or failure/cutoff
return RECURSIVE-DLS(MAKE-NODE(problem.INITIAL-STATE), problem, limit)

function RECURSIVE-DLS(node, problem, limit) returns a solution, or failure/cutoff
if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
else if limit = 0 then return cutoff
else
  cutoff_occurred? ← false
  for each action in problem.ACTIONS(node.STATE) do
    child ← CHILD-NODE(problem, node, action)
    result ← RECURSIVE-DLS(child, problem, limit - 1)
    if result = cutoff then cutoff_occurred? ← true
    else if result ≠ failure then return result
  if cutoff_occurred? then return cutoff else return failure

```

Fuente: (Russell & Norvig, 2010).

A menudo, los límites de profundidad pueden establecerse en base al problema. Por ejemplo, en el mapa del agente viajante de Rumanía hay una cantidad de 20 puntos de búsqueda. Por lo tanto, sabemos que, si hay una solución, debe tener una longitud máxima de 19, por lo que $t = 19$ es una opción posible. Pero, si estudiamos el mapa detenidamente, descubriremos que se puede llegar a cualquier punto de búsqueda (ciudad) desde cualquier otra ciudad en un máximo de 9 pasos. Este número, conocido como el espacio de estados (la totalidad de estados posibles), permite establecer un mejor límite de profundidad, lo que conduce a una búsqueda más eficiente con

limitación de profundidad. No obstante, no se sabrá el correcto límite de profundidad hasta que se haya resuelto el problema". (Russell & Norvig, 2010).

La búsqueda de profundidad limitada se puede llevar a cabo como una sencilla modificación del algoritmo general de búsqueda de árbol. Alternativamente, se puede implementar como un algoritmo recursivo simple. Se puede inferir que la búsqueda de profundidad limitada puede concluir con dos tipos de fallas: que no se encuentre la solución; o que no haya solución dentro del límite de profundidad" (Russell & Norvig, 2010).

Primero en profundidad con profundidad iterativa

La búsqueda iterativa de profundización (o búsqueda iterativa de profundización primero en profundidad) se realiza combinando la búsqueda de árboles en profundidad, que encuentra el mejor límite de profundidad. Para ello, aumenta gradualmente el límite (primero 0, luego 1, luego 2, y así sucesivamente) hasta que se encuentra una meta.

Esto ocurrirá cuando el límite de profundidad llegue a la profundidad más superficial del nodo objetivo. La profundización iterativa es una mixtura de los beneficios de la búsqueda en profundidad y en amplitud. Al igual que la búsqueda en profundidad, sus requisitos de memoria son modestos. Al igual que la búsqueda en amplitud, está completa cuando el costo de la ruta es una función no decreciente de la profundidad del nodo" (Russell & Norvig, 2010)

Figura 13:

Pseudocódigo de Primero en profundidad con profundidad iterativa

```
function ITERATIVE-DEEPENING-SEARCH(problem) returns a solution, or failure  
for depth = 0 to  $\infty$  do  
    result  $\leftarrow$  DEPTH-LIMITED-SEARCH(problem, depth)  
    if result  $\neq$  cutoff then return result
```

Fuente: (Russell & Norvig, 2010).

La búsqueda iterativa de profundización puede parecer un desperdicio porque los estados se generan varias veces. Resulta que esto no es demasiado costoso. Lo que da una complejidad de tiempo de $O(b^d)$ —asintóticamente lo mismo que la búsqueda preferente en amplitud. Existe un costo adicional para generar los niveles superiores varias veces, pero es enormemente grande.

Si realmente le preocupa la repetición de la repetición, puede utilizar un enfoque híbrido que ejecute la búsqueda en amplitud hasta que se consuma casi toda la memoria disponible y luego una profundización iterativa de todos los nodos en la frontera. En general, la profundización iterativa es el método de búsqueda desinformado preferido cuando el espacio de búsqueda es grande y la profundidad de la solución no se conoce.

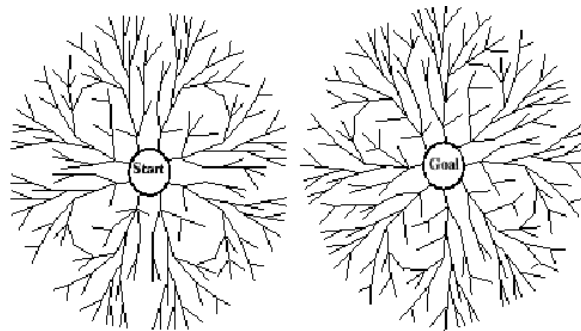
La búsqueda iterativa de profundización es análoga a la búsqueda en amplitud en el sentido de que explora una capa completa de nuevos nodos en cada iteración antes de pasar a la siguiente capa.

Búsqueda Bidireccional

La búsqueda bidireccional ejecuta dos búsquedas simultáneas, una hacia adelante desde el estado inicial y la otra hacia atrás desde el objetivo, con la esperanza de que las dos búsquedas se encuentren en el medio (Figura)". (Russell & Norvig, 2010)

Figura 14:

Vista esquemática de una búsqueda bidireccional



Fuente: (Russell & Norvig, 2010).

La búsqueda bidireccional se implementa reemplazando la prueba del objetivo con una verificación para ver si las fronteras de las dos búsquedas se cruzan; si lo hacen, se ha encontrado una solución. (Es importante darse cuenta de que la primera solución de este tipo encontrada puede no ser óptima, incluso si las dos búsquedas son ambas en amplitud; se requiere una búsqueda adicional para asegurarse de que no haya otro atajo a través de la brecha). La verificación se puede hacer cuando cada nodo se genera o se selecciona para expansión y, con una tabla hash, tomará un tiempo constante.

Podemos reducir el espacio de búsqueda si una de las dos búsquedas se realiza mediante profundización iterativa, pero al menos una de las

fronteras debe mantenerse en la memoria para que se pueda realizar la verificación de intersección. Este requisito de espacio es la debilidad más significativa de la búsqueda bidireccional. La búsqueda bidireccional requiere un método para calcular predecesores. Cuando todas las acciones en el espacio de estados son reversibles, los predecesores de x son solo sus sucesores. Otros casos pueden requerir un ingenio sustancial.

Estrategias de Búsqueda informada

Indica la forma en cómo se expandirán los nodos dentro de un árbol de búsqueda, sabiendo cuales nodos no son objetivos, y cuales son más "prometedores" que otros, es decir aquellos nodos que permiten llegar más rápidamente al objetivo, siguiendo un determinado algoritmo de búsqueda (Russell & Norvig, 2010)

Se propondrá un ejemplo de búsqueda informada, supongamos que hay tres caminos que salen de A (C, D y E), pero ninguno directo a B, siendo B el nodo objetivo. El agente tiene un mapa que le brinda información sobre los nodos en que podría estar, así como las acciones que podría tomar, y de acuerdo a esta información que tiene guardada puede seguir los nodos que le llevarán en menor tiempo al nodo objetivo. Por eso se dice que es informada.

A continuación, se presentan los algoritmos más utilizados de este tipo de estrategia:

Búsqueda voraz primero el mejor (Greedy best first)

Una búsqueda voraz primero el mejor es una BÚSQUEDA-GRAFO donde los nodos de costo mínimo que no han sido expandidos se escogen para la expansión. Para ello, utiliza típicamente una función heurística $h(n)$ que estima el costo De ese punto actual (nodo n) hasta el nodo objetivo". (Russell & Norvig, 2010)

$h(n)$ = coste estimado desde el nodo n a un nodo objetivo.

Esta función devuelve un número que ayuda a indicar lo deseable o indeseable que sería la expansión de ese nodo. Se expande primero aquel nodo que tiene mejor evaluación, se escoge el que parece ser el mejor, aunque podría no serlo. La función heurística viene dada por la siguiente fórmula".. (Russell & Norvig, 2010)

$$h(n) = dE(n', ng) = (ngx - n'x)^2 + (ngy - n'y)^2$$

donde:

- n' es un nodo cercano al nodo actual con coordenadas cartesianas $(n'x, n'y)$ el cual no está ocupado por un obstáculo y es posiblemente un siguiente nodo en la ruta.
- ng es el nodo objetivo con coordenadas cartesianas (ngx, ngy) .

La búsqueda voraz primero el mejor propaga nodos con el valor $h(n)$ menor. No siempre encuentra la ruta óptima, pero sin embargo

es eficiente con respecto a la cantidad de nodos explotados para encontrar el objetivo.

Búsqueda A*

La búsqueda A * (se pronuncia "búsqueda A estrella") evalúa los nodos sumando dos funciones: La función $g(n)$ que calcula el costo para llegar hasta el nodo actual, y $h(n)$ que calcula el costo para llegar hacia el nodo objetivo:

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

Ya que $g(n)$ da el costo de la ruta a partir del nodo inicial al nodo n , y $h(n)$ es el costo estimado de la ruta más barata de n a la meta, tenemos, $f(n) =$ costo estimado de la solución más barata a través de todos. Por lo tanto, si estamos tratando de encontrar la solución más barata, una cosa razonable para probar primero es el nodo con el valor más bajo de $f(n)$. Resulta que esta estrategia es más que razonable: siempre que la función heurística $h(n)$ satisfaga ciertas condiciones, la búsqueda A * es completa y óptima. El algoritmo es idéntico a búsqueda de costos uniformes excepto que A * usa $g + h$ en lugar de g . Como mencionamos anteriormente, A* tiene las siguientes propiedades: la versión de búsqueda de árbol de A * es óptima si $h(n)$ es admisible, mientras que la versión de búsqueda de gráfico es óptima si $h(n)$ es consistente. Mostramos la segunda de estas dos afirmaciones ya que es más útil. El argumento esencialmente refleja el argumento de la optimización de la búsqueda de costo uniforme, con g reemplazado por f , al igual que en el algoritmo A * mismo. "La complejidad de A * a

menudo hace que no sea práctico insistir en encontrar una solución óptima. Se pueden usar variantes de A* que encuentran soluciones subóptimas rápidamente, o algunas veces se pueden diseñar heurísticas que son más precisas pero no estrictamente admisibles. En cualquier caso, el uso de una buena heurística aún proporciona enormes ahorros en comparación con el uso de una búsqueda desinformada. El tiempo de cálculo no es, sin embargo, un pequeño inconveniente. Debido a que mantiene todos los nodos generados en la memoria (al igual que todos los algoritmos búsqueda gráfica), A* generalmente se queda sin espacio mucho antes de que se agote el tiempo. Por esta razón, A* no es práctico para muchos problemas a gran escala. Sin embargo, existen algoritmos que superan el problema del espacio sin sacrificar". (Russell & Norvig, 2010)

Figura 15 :

*Pseudocódigo del algoritmo A**

```
Pseudo Code:
Astar(Graph,start_node,end_node)
{
    Create Empty List CLOSEDSET
    Create Empty List OPENSET
    OPENSET.add(start_node)
    while OPENSET.count > 0
        find the node in OPENSET with lowest f cost and call it current
        current.visited = true
        CLOSEDSET.add(current)
        OPENSET.remove(current)
        if current == end_node
            display path found
            return
        end if
        for all node n adjacent to current
            if not n.visited
                n.g_cost = current.g_cost + distance between g and n
                n.h_cost = distance between n and end_node
                n.f_cost = n.g_cost + n.h_cost
            end if
            if n not in OPENSET
                OPENSET.add(n)
            end if
        end for
    end while
}
```

Fuente: (Kolhe et al., 2018)

Dirección de Profundidad A*

La dirección de profundidad A * es una modificación de A * en la que cuando no se encuentran obstáculos en las proximidades del nodo actual donde ha llegado la ruta, calcula el siguiente nodo usando la teoría de grafos usando la fórmula: $y = mx + c$

donde y es el índice y y x es el índice x y m es la pendiente de la línea desde la corriente apunta al nodo final y c es la intersección de esa misma línea. m se calcula usando la fórmula

$$m=(y_2 - y_1)/(x_2 - x_1)$$

donde x_2 e y_2 son las coordenadas x e y del nodo objetivo, respectivamente, x_1 e y_1 son las coordenadas x e y del cabeceo actual. De manera similar, c se puede encontrar sustituyendo valor de x , y y m en la ecuación de la recta. Por lo tanto, usando m , c , y el nuevo valor de coordenada x , es decir, ya sea la antigua coordenada $x + 1$ o la antigua coordenada $x - 1$, podemos encontrar el valor coordenada y para el siguiente nodo en la ruta, mientras que si encontramos un obstáculo utilizamos el algoritmo A * tradicional para evitar el obstáculo.

También utilizamos un montón en lugar de una lista para OPENSET, ya que usamos un montón

Reducir el tiempo requerido para encontrar el elemento que tiene el costo más bajo es un asunto muy costoso donde la mayor parte del tiempo es gastado por el algoritmo. (Kolhe et al., 2018)

Figura 16 :

Pseudocódigo de primero en profundidad

```
Depth Direction A* pseudo code:
Depth_Direction_Astar(Graph,root)
{
    Find initial value of  $m = (\text{Goal Y} - \text{Start Y})/(\text{Goal X} - \text{Start X})$ 
    Create Empty List CLOSESET
    Create Empty Heap OPENSET
    OPENSET.add(root)
    While OPENSET not empty
        current = OPENSET.removefirst()
        if current's neighbors not obstacles
            find next node using linear graph theory
        else
            find next node using A*
        end if
    end while
}
```

Fuente: (Kolhe et al., 2018)

Planificación de rutas

Los métodos para la planificación de rutas son tomados desde la rama científica de investigación de operaciones. Los algoritmos para la planificación de rutas tienen como objetivo explorar un espacio del cual se tiene poca o ninguna información "con el fin de encontrar una ruta que pueda conectar dos puntos con coordenadas para simplificar las tareas de búsqueda" (Ríos, 2015). La planificación de rutas permite resolver problemas del viajante "TSP Traveling Salesman Problem" en la que se debe encontrar la ruta mínima por lo cual un vehículo viajará para recorrer puntos distantes geográficamente, y problemas de enrutamiento de vehículos (VRP Vehicle Routing Problem) que se relaciona con la necesidad de que varios vehículos que circulan en una misma unidad geográfica deban recorrer todos los puntos de un problema específico (Fernandez, 2016).

Actualmente hablamos de los problemas de rutas de vehículos con intervalo de tiempo cuyas siglas en inglés son VRPTW (Vehicle Routing Problem with Time Windows) donde la característica adicional está estipulada en que los clientes deben ser atendidos dentro de un intervalo de tiempo establecido (ventana de tiempo), solo una vez y por un solo vehículo. (Konstantakopoulos et al., 2020)

Una planificación de la ruta más corta puede reducir el tiempo consumido al intentar trasladarse a otra ubicación. Además, si el agente artificial puede detectar obstáculos y planificar un camino más corto para evitar esos obstáculos, ayudará a reducir aún más el tiempo y los errores del camino óptimo encontrado. (Angkuldee et al., 2019)

Método de la Ruta más corta

Generalmente, se emplea una aplicación informática para encontrar la ruta más corta entre dos puntos. Además, se utilizan varias técnicas para calcular la ruta óptima y más rápida mediante el uso de métodos de búsqueda de gráficos.

El problema de la ruta más corta se refiere a la búsqueda de la ruta entre los problemas de la ruta más corta calculando su suma de ponderaciones en una teoría de grafos. Dos problemas son categorías basadas en el problema de la ruta más corta, la ruta más corta de una sola fuente y todos los problemas de algoritmo de pares más cortos. En una sola fuente, apunte a encontrar la ruta más rápida desde un vértice dado a todos los vértices en un gráfico mientras que en todos los pares la ruta más disparada, los objetivos buscarán la ruta más corta entre todos los pares de vértices en el gráfico (Alyasin et al., 2019).

Dijkstra

El algoritmo de Dijkstra es un algoritmo iterativo que proporciona la ruta más corta desde un nodo inicial particular a todos los otros nodos en el grafo. Se expande hacia afuera desde el punto de partida hasta llegar a la meta

El algoritmo de Dijkstra es uno de los mejores métodos para estimar la ruta mínima para llegar al punto. Dijkstra encuentra la ruta más corta entre dos nodos con un solo nodo como nodo "fuente" y encuentra las rutas más cortas desde la fuente a todos los demás nodos en el gráfico, produciendo un árbol de ruta más corto. (Alyasin et al., 2019)

"Se trata de una especialización de la búsqueda de costo uniforme y, como tal, no funciona en grafos con aristas de coste negativo (al elegir siempre el nodo con distancia menor, pueden quedar excluidos de la búsqueda nodos que en próximas iteraciones bajaría el costo general del camino al pasar por una arista con costo negativo)". (Alyasin et al., 2019)

Usando el algoritmo de Dijkstra, es posible determinar la distancia más corta (o el menor esfuerzo / menor costo) entre un nodo de inicio y cualquier otro nodo en un gráfico. La idea del algoritmo es calcular continuamente la distancia más corta a partir de un punto de partida y excluir distancias más largas al realizar una actualización.

Figura 17:

Pseudocódigo del algoritmo Dijkstra

Pseudo Code:

```
dijkstra(Graph, start_node, end_node)
Create Empty List VISITED
Create Empty List NEIGHBOUR
current = start_node
VISITED.add(current)
while current != end_node
for each neighbour n of current in GRAPH
NEIGHBOUR.add(n)
end for
current = NEIGHBOUR.min()
VISITED.add(current)
end while
```

Fuente: (Kolhe et al., 2018)

A*

Es un algoritmo que también se utiliza en la búsqueda de caminos y tiene su mismo funcionamiento, por eso es que no se detalla en este apartado su algoritmo, solo se resalta que el algoritmo A*, tiene como objetivo hallar la ruta más corta utilizando un heurístico que permite estimar la distancia desde un nodo hasta el destino, a través de la suma de dos funciones: la función $g(n)$ que haya el costo de la ruta desde el nodo inicial al nodo n , y la función $h(n)$ que es el costo estimado de la ruta más barata del nodo inicial al nodo meta,

Búsqueda Voraz primero el mejor

Es un algoritmo que también es usado dentro de la búsqueda de caminos que se vio anteriormente, por eso solo se resumirá en el siguiente concepto: "El algoritmo de búsqueda voraz primero el mejor expande todos los nodos vecinos del nodo actual donde se encuentra el agente, buscando el siguiente nodo que tenga la

distancia más corta hasta el nodo objetivo. Cada nodo vecino que es expandido se evalúa con la función heurística" (Reyes, 2018).

Capítulo V: Elaboración de Instrumentos de recojo de datos sobre la búsqueda de caminos en una Institución Educativa Publica

Los métodos empíricos utilizados para demostrar las deficiencias en la búsqueda de caminos en una Institución Educativa Pública de la ciudad de Lambayeque - Peru conllevan a la utilización de las siguientes técnicas e instrumentos:

Técnicas e Instrumentos de recojo de datos

Entrevista (técnica) al encargado de portería de la Institución Educativa Pública mediante una guía de entrevista (instrumento), se realizó preguntas acerca del requerimiento del público usuario para la búsqueda de caminos de los diferentes ambientes de la Institución educativa, afluencia de público en condiciones diferentes a la emergencia nacional del COVID- 19 y la atención que se brinda al usuario para brindarle la información solicitada

Encuesta (técnica) a los padres de familia de la Institución Educativa mediante cuestionario (instrumento) para conocer la percepción que tienen con respecto a la ayuda que se le brinda en la búsqueda de caminos durante el proceso de búsqueda de caminos de los diferentes ambientes educativos de la Institución Educativa. También se utilizó un cuestionario a los especialistas en Inteligencia Artificial para conocer la problemática con respecto a búsqueda de caminos y la necesidad de establecer una metodología que unifique la creación de sistemas de búsqueda de caminos.

Análisis Bibliográfico (técnica) sobre el objeto de estudio, de las variables de investigación, así como la propuesta de la metodología integrativa para la creación de agentes artificiales para búsqueda de caminos. Se hará análisis

documental de material bibliográfico en formato impreso y digital mediante fichas bibliográficas digitales (instrumento)

Técnica	Instrumento	Sujetos	Explicación
Entrevista	Guía de entrevista	Encargado de Portería	Se aplicó una guía de entrevista de 7 preguntas abiertas a través de la plataforma Google meet para medir la problemática con respecto a la búsqueda de caminos de ambientes educativos y la aceptación de un software informático que ayude a mejorarla.
Encuesta	Cuestionario	Padres de familia	Se aplicó un cuestionario de 11 ítems, 2 dicotómicas, 1 de respuesta múltiple y 8 de respuesta única. A los ítems de respuesta única, los valores estaban de acuerdo con una escala de Likert de 5 opciones. La finalidad fue recolectar la percepción sobre la

Técnica	Instrumento	Sujetos	Explicación
			realidad problemática de la localización de ambientes educativos.
Encuesta	Cuestionario	Especialistas en Inteligencia artificial	Se aplicó un cuestionario de 12 ítems, 1 dicotómica, 5 de respuesta múltiple y 6 de respuesta única. A los ítems de respuesta única, los valores estaban de acuerdo con una escala de Likert de 5 opciones. Se realizó para recabar la percepción de especialistas en IA sobre la búsqueda de caminos y sus conceptos relacionados.

La **validez**, “que alude a la capacidad del instrumento de medir el constructo que pretende cuantificar” fue analizada por medio de un juicio de expertos conformado por 5 especialistas: 02 doctores en Ciencias de la computación e Ingeniería, 01 Doctor en Matemática Aplicada con título en Ingeniería en Computación e Informática, 01 Doctora en Educación con mención en Gestión Educativa y 01 Ingeniero Civil.

La **confiabilidad** fue establecida por la medida estadística del Alfa de Cronbach a los instrumentos tipo cuestionario.

Para el cuestionario aplicado a 46 padres de familia se obtuvo un valor del Alfa de Cronbach de 0.806, lo que indica un valor “bueno”.

Resumen del procesamiento de los casos

		N	%
Casos	Válidos	46	100,0
	<u>Excluidos^a</u>	0	,0
	Total	46	100,0

a. Eliminación por lista basada en todas las variables del procedimiento.

Estadísticos de fiabilidad

Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,806	,797	8

Para el cuestionario a especialistas en Inteligencia Artificial se aplicó a 5 encuestados y se obtuvo un valor del Alfa de Cronbach de 0.825, lo que indica un valor “bueno”.

Resumen del procesamiento de los casos

	N	%
Válidos	5	100,0
Casos <u>Excluidos^a</u>	0	,0
Total	5	100,0

a. Eliminación por lista basada en todas las variables del procedimiento.

Estadísticos de fiabilidad

Alfa de Cronbach	Alfa de Cronbach basada en los elementos tipificados	N de elementos
,825	,861	5

Marco ético en la aplicación de instrumentos

Esta investigación se realizó en el Marco de los principios éticos establecidos para toda investigación científica (Comisión nacional para la protección de los sujetos humanos de investigación biomédica y del Comportamiento; 1979).

En todo momento se respetó la autonomía de las personas a quienes se les aplique los instrumentos de recojo de información, quienes participarán en la investigación voluntariamente y a quienes se les brindará la información sobre los objetivos de la investigación y la importancia de su participación, sin ejercer ningún tipo de coacción para que otorguen la información. Complementando con lo anterior se le pedirá a la persona a entrevistar y/o encuestar su consentimiento informado, respetando los tres principios básicos: Información, por medio del cual se le brindará en todo momento al usuario la información del porqué de la investigación, sin ocultarle ningún detalle importante, e indicándole la forma en cómo se resguardará y se dará protección a la

información que este brinde. Así mismo, se les indicará a los directivos los resultados de la investigación una vez que ésta haya culminado.

Otra forma de respetar la autonomía fue evitando tanto en los cuestionarios como en las guías de entrevista, preguntas de tipo tendenciosas, que hagan que el usuario no tenga libertad de expresar lo que realmente piensa o siente sobre lo que se le está preguntando.

También se dio protección a los datos que se extraigan de la institución. En tal sentido, los planos y demás datos obtenidos de la Institución Educativa serán utilizados única y exclusivamente para fines académicos que permitan desarrollar la presente investigación, los cuales serán resguardados y en ningún momento serán publicados en Internet.

La Confidencialidad, que es parte del secreto profesional, indica ser celoso con la investigación, evitando divulgar la información de datos personales que se obtienen producto de esta investigación; tal como lo expone La ley N° 29733: "Ley de protección de datos personales", en su Título IV: "Obligaciones del titular y del encargado del banco de datos personales" señala: Que recopilar datos personales por medios ilícitos, fraudulentos o desleales está penado por el estado peruano. En tal sentido, se cuidará la identidad de las personas a quienes se les aplique la entrevista y/o encuesta, evitando den sus datos personales como nombres, apellidos o número de documento de identificación, por tanto, serán totalmente anónimas para así evitar revelar la identidad del encuestado o entrevistado.

La presente investigación respetó los criterios de objetividad, validez, originalidad y actualidad de la información.

La Objetividad, que obliga dar a conocer aspectos con veracidad en relación a los estudios realizados sobre el proceso de búsqueda de caminos en una

Institución Educativa dejando de lado la subjetividad del investigador. En este sentido, se respetó los resultados obtenidos de la investigación, evitando manipular datos, que conlleven a información dudosa o inexacta, aún si los resultados sean adversos a la hipótesis planteada, sobretodo en la parte de la ejemplificación práctica parcial de la búsqueda de caminos a través del software informático.

La validez, será resguardada por el juicio de expertos que se hará a los instrumentos utilizados para el recojo de datos, así como para el instrumento que validará la metodología integrativa propuesta.

La Originalidad, respetando el derecho de autor de los documentos físicos o digitales de donde se extraiga la información, de tal forma que se citarán las fuentes bibliográficas de la información mostrada en la presente investigación, a fin de evitar el plagio intelectual.

Sobre la actualidad de la información, se respetó que más del 80% de las fuentes de información sean de los últimos 5 años. Con respecto a la calidad de la información del sustento teórico de la presente investigación se buscó información de bases de datos confiables como IEEEExplore, Researchgate, Scopus, Alicia y la base de datos de la Universidad Señor de Sipan.

Capítulo VI: Diagnóstico del estado actual del proceso de búsqueda de caminos en una Institución Educativa Pública de Perú

Para obtener el diagnóstico del estado actual del proceso de búsqueda de caminos de ambientes educativos, se aplicó un cuestionario a 46 padres de familia de la Institución Educativa Pública de Educación básica regular de la ciudad de Lambayeque- Perú, que fue de carácter anónimo, es decir no se consignó los nombres y apellidos de los encuestados, a través de un formulario para cuestionarios de Google Form cuyo enlace fue: <https://forms.gle/yHWnEGstyUJ6BtA39> y que fue enviado a algunos docentes y estos a su vez a los padres de familia, a través del whatsapp. El cuestionario está conformado por 11 ítems, 2 dicotómicas, 1 de respuesta múltiple y 8 de respuesta única. A los ítems de respuesta única se les asignó valores que estaban de acuerdo con una escala de Likert de 5 opciones

Así mismo, se aplicó una guía de entrevista al encargado de portería de la Institución Educativa a través de la plataforma de video conferencia Google meet. La guía contenía 7 preguntas abiertas para obtener información sobre su percepción del proceso de búsqueda de caminos de ambientes en la institución educativa

Resultados del procesamiento de información del Cuestionario a Padres de Familia

Se mostrarán los resultados de los principales ítems del cuestionario de padres de familia que corresponde a las preguntas de opción única.

Se evidencia que la mayoría de encuestados opina que la señalética en la Institución Educativa es "regular" con 39.1% y solo el 26.1% opina que es "Inadecuada". Estos resultados afianzan la necesidad de contar con procedimientos en la búsqueda de caminos mas eficientes que ayuden a mejorar la búsqueda de caminos.

Se evidencia que la mayoría de encuestados ha tenido necesidad de encontrar algún ambiente u oficina de forma "ocasional" con un 37% y solo el 4.4% opina que "nunca" ha tenido dificultad, con lo que se demuestra la existencia del problema de búsqueda de caminos dentro de la Institución Educativa.

Se evidencia que la mayoría de encuestados ha llegado a una ubicación incorrecta a pesar de haber preguntado por su localización de forma "ocasional" con un 34.8% y solo el 4.3% opina que "siempre" han tenido ese inconveniente, la mayoría de encuestados "ocasionalmente" ha demorado mucho para llegar a un ambiente educativo que previamente se preguntó por su localización con un 45.7% y solo el 2.1% opina que "nunca" ha tenido ese inconveniente, la mayoría de encuestados tienen un grado de satisfacción "regular" con un 58.7% y solo el 2.2% opina que se encuentran "insatisfechos" cuando ha requerido información en la ubicación de algún ambiente u oficina.

Se evidencia que la mayoría de encuestados están "de acuerdo" con utilizar un software para la búsqueda de ambientes educativos con un 47.8% y solo el 2.2% opina que estarían "totalmente en desacuerdo" con esta solución.

Para ver el las puntuaciones obtenidas por la aplicación del cuestionario a padres de familia “Resultados obtenidos en la aplicación de instrumentos de recolección de datos”.

Resultados del procesamiento de información de la guía de Entrevista al Encargado de Portería

Se mostrarán un resumen de las respuestas dadas por el encargado de portería que se obtuvieron al aplicarle la guía de entrevista.

En la pregunta 1: ¿En qué periodos hay gran afluencia de público, en que meses y para que actividades? La respuesta que se obtuvo por el encargado de portería demuestra que existe gran afluencia de público sobre todo para los meses de enero a Marzo, mes de Julio y mes de diciembre.

En la pregunta 2: ¿Desarrolla actividades que le obligan a moverse a diferentes ambientes de la IIEE? ¿Cuán a menudo ocurre esto? indicó que hay actividades propias de sus funciones que le exigen moverse por diferentes ambientes de la Institución Educativa, También aquí se indicó el proceso de búsqueda de caminos de ambientes educativos, que se diferencia por el tipo de usuario, cuando es una autoridad sea UGEL, municipio u otro, se le acompaña hasta el ambiente de destino, cuando es una persona natural, cualquier miembro de la comunidad, se le da pautas teniendo en cuenta puntos importantes como esquinas, escaleras, para que llegue al destino, pero que aun así se iban a un ambiente que no era el que estaban buscando.

Con respecto a la pregunta ¿Hay otras personas encargadas en la portería del colegio en clases presenciales? ¿Cómo se turnan? Indicó que solo era él y que había falta de personal administrativo, porque a pesar que se ha solicitado a la unidad de gestión local, hasta la actualidad no dan trámite a la solicitud.

Con respecto a la pregunta, ¿Ha habido ocasiones en que le han preguntado sobre la ubicación de algún ambiente u oficina de la IIEE? ¿En qué ocasiones ocurre esto con mayor frecuencia? Indica que siempre se da esto, y que los usuarios constantemente preguntan sobre la búsqueda de ambientes y oficinas educativas.

Con respecto a la pregunta ¿Considera que una de sus funciones es atender al público sobre la ubicación de algún ambiente u oficina de la IIEE? Si no es así, ¿Por qué realiza esta función? Mencionó que como servidor público su función era siempre atender a la comunidad educativa que llega a la institución.

Con respecto a la pregunta ¿Considera que es importante brindar una buena atención al público? ¿Por qué? Indicó que los servidores públicos deben brindar una buena atención al público.

Con respecto a la pregunta: ¿Cree que sería importante contar con una aplicación desde el celular que ayude a la localización de oficinas? mencionó que la institución educativa es bastante amplia para que pueda ubicarse las principales oficinas, y que sería importante contar con tecnología que ayude a mejorar este procedimiento.

Resultados del procesamiento de información del Cuestionario a Especialistas en Inteligencia Artificial

Para obtener el diagnóstico del estado actual del proceso de búsqueda de caminos, se aplicó un cuestionario a 5 especialistas investigadores en Inteligencia Artificial de la ciudad de Lambayeque, que fue de carácter anónimo, es decir no se consignó los nombres y apellidos de los encuestados, a través de un

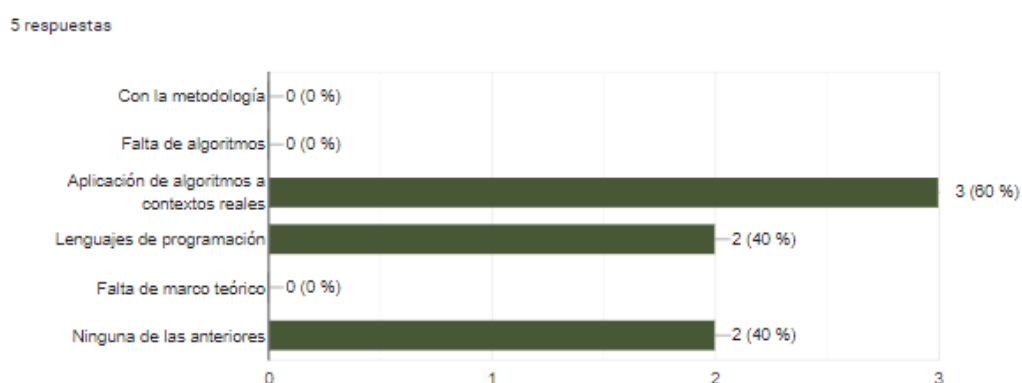
formulario de cuestionario Google Form cuyo enlace fue: <https://forms.gle/G7vkGW6pYHsT7Wxy5> y que fue enviado a través de la mensajería instantánea whatsapp. El cuestionario está conformado por 12 ítems, 1 dicotómico, 5 de respuesta múltiple y 6 de respuesta única.

A los items de respuesta única se les asignó valores que estaban de acuerdo con una escala de Likert de 5 opciones

Se mostrarán los resultados de los principales ítems del cuestionario de padres de familia que corresponde a las preguntas de opción única.

Figura 18

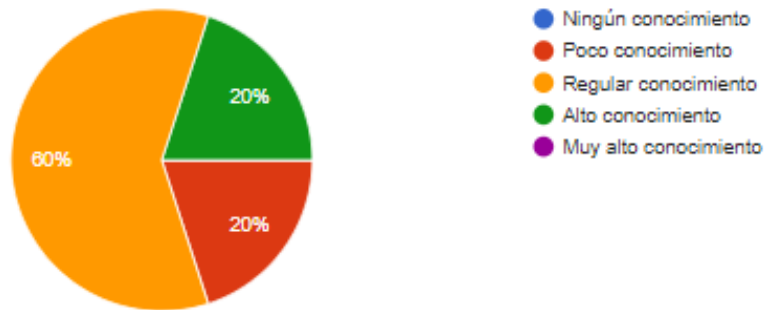
Ítem 3 referido a la dificultad al momento de realizar una aplicación en algún tema de IA



Fue una pregunta con respuesta de elección múltiple, de tal forma que un encuestado podría marcar más de una alternativa. Se evidencia que la mayoría de encuestados opina que la dificultad para realizar un software en IA es la aplicación de algoritmos a contextos reales con un 60% del 100%, así mismo marcaron como dificultad la falta de referentes teóricos.

Figura 19

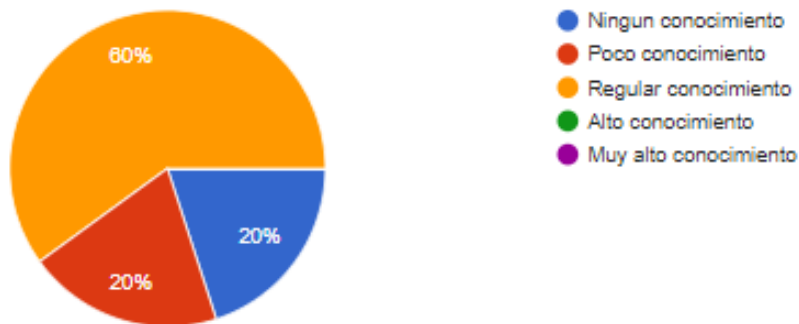
Ítem 4 referido al conocimiento en la creación de sistemas para búsqueda de caminos o planificación de rutas



Se evidencia que la mayoría de encuestados indica tener un “regular conocimiento” con un 60% en la creación de sistemas de búsqueda de caminos o planificación de rutas y solo el 20% “alto conocimiento”.

Figura 20

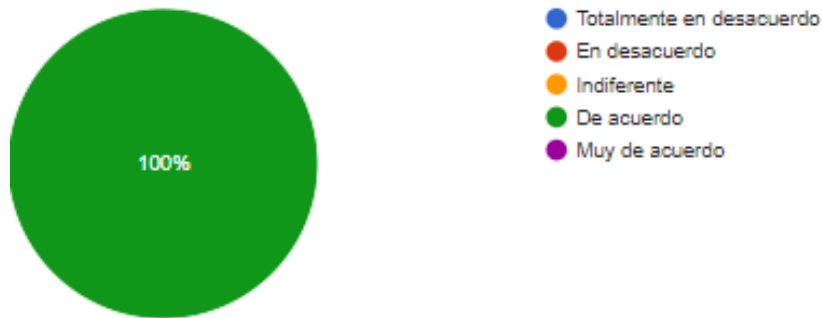
Ítem 5 referido al conocimiento en la creación de agentes artificiales



Se evidencia que la mayoría de encuestados indica tener un “regular conocimiento” con un 60% en la creación de agentes artificiales y solo 20% “ningún conocimiento”.

Figura 21

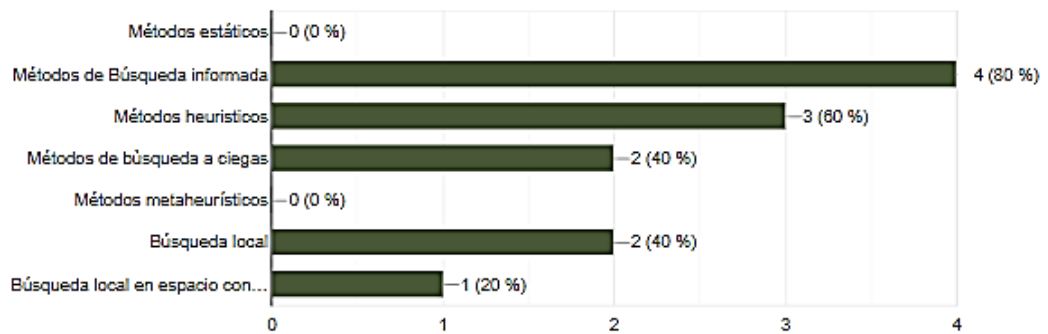
Ítem 6 referido a la percepción de vigencia del tema de búsqueda de caminos o planificación de rutas



Se evidencia que todos los encuestados opinan que el tema de búsqueda de caminos o planificación de rutas es un tema vigente, con un 100%. Esto indica que es importante abordar este tema para generar nuevo conocimiento y que para la investigación vendría a ser el objeto de estudio.

Figura 22

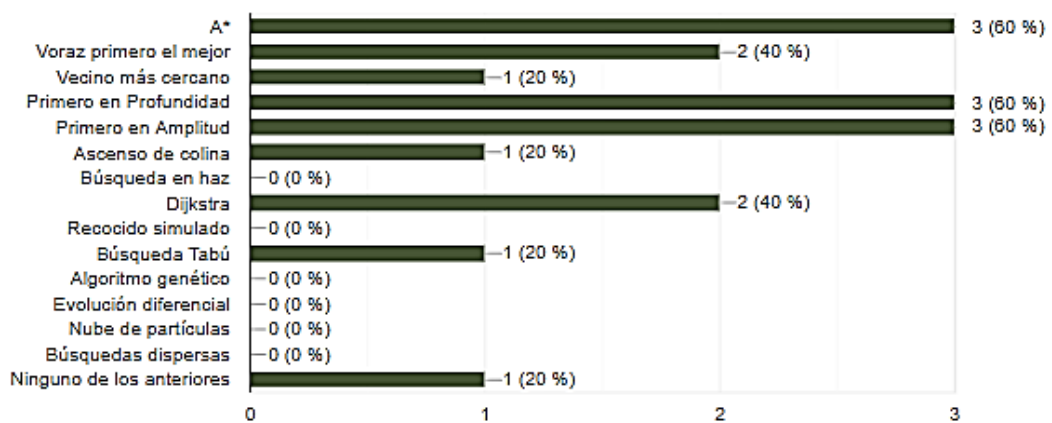
Ítem 7 referido al conocimiento de métodos para búsqueda de caminos



Fue una pregunta con respuesta de elección múltiple, de tal forma que un encuestado podría marcar más de una alternativa. Se evidencia que la mayoría de encuestados conoce los métodos de búsqueda informada 80% de 100%, y ninguno conoce los métodos estáticos y métodos metaheurísticos.

Figura 23

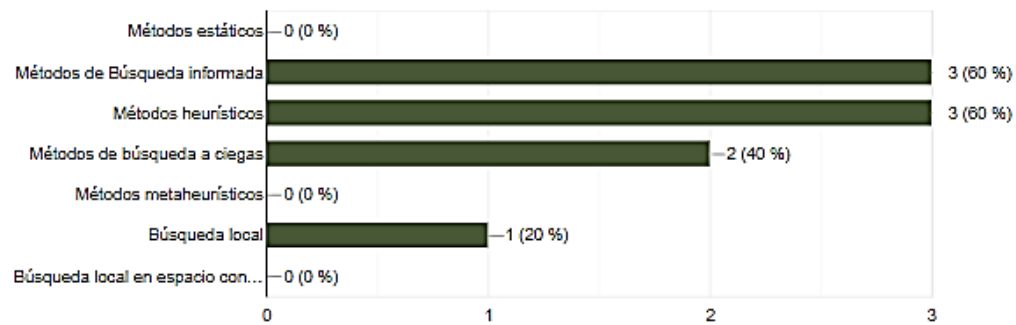
Ítem 8 referido al conocimiento de algoritmos para búsqueda de caminos



Fue una pregunta con respuesta de elección múltiple, de tal forma que un encuestado podría marcar más de una alternativa. Se evidencia que la mayoría de encuestados conoce el algoritmo A*, primero en profundidad, primero en amplitud con un 60% de 100% cada uno respectivamente. Se evidencia también que se confunden con los algoritmos de planificación de rutas como son vecino más cercano, Dijkstra, búsqueda Tabú y que no son aplicados comúnmente para búsqueda de caminos.

Figura 24

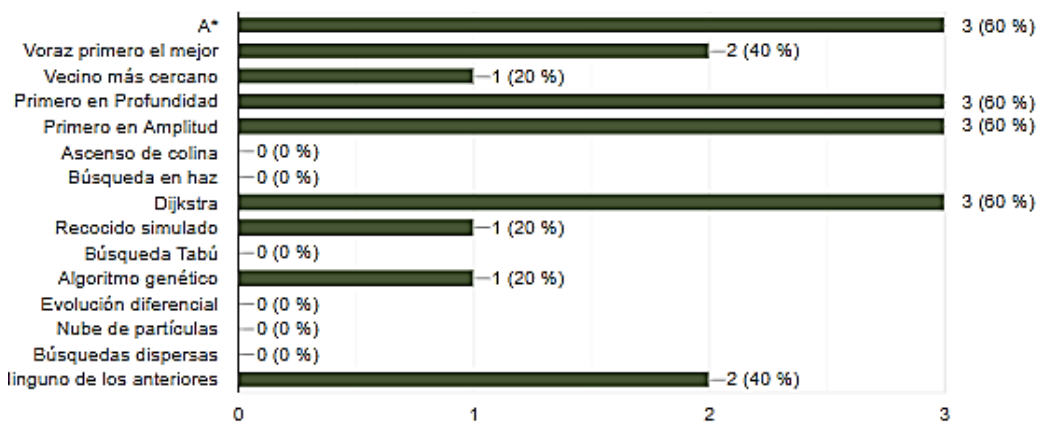
Ítem 9 referido al conocimiento de métodos para la planificación de rutas



Fue una pregunta con respuesta de elección múltiple, de tal forma que un encuestado pudo marcar más de una alternativa. Se evidencia que la mayoría de encuestados se confunde con los métodos utilizados para la búsqueda de caminos como son: métodos de búsqueda informada, métodos de búsqueda a ciegas. Se evidencia también que la mayoría conoce los métodos heurísticos para la planificación de rutas, con un 60% de 100%.

Figura 25

Ítem 10 referido al conocimiento de algoritmos para la planificación de rutas

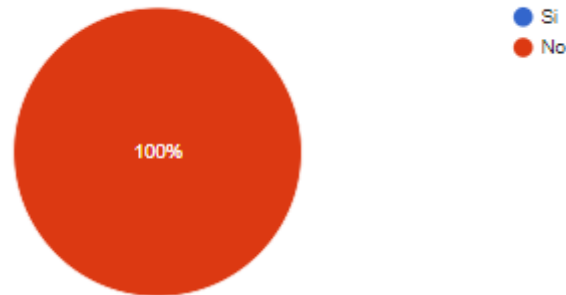


Fue una pregunta con respuesta de elección múltiple, de tal forma que un encuestado podría marcar más de una alternativa. Se evidencia que la mayoría de encuestados se confunde con los algoritmos utilizados para la búsqueda de caminos como son: primero en profundidad, primero en amplitud. Se evidencia

también que la mayoría conoce los algoritmos A* y Dijkstra para la planificación de rutas, con un 60% de 100% cada uno respectivamente.

Figura 26

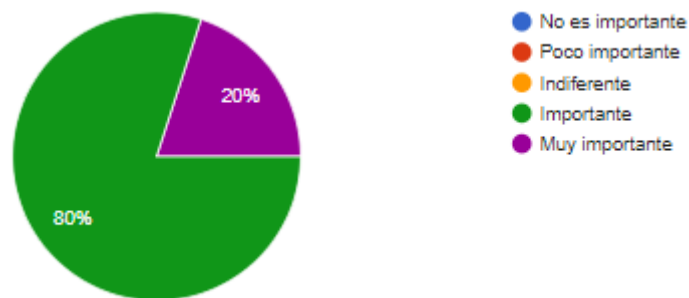
Ítem 11 sobre el conocimiento de alguna metodología de búsqueda de caminos



Se evidencia que ninguno tiene conocimiento sobre la existencia de una metodología para búsqueda de caminos, siendo el 100% los que marcaron "no".

Figura 27

Ítem 12 sobre la importancia de la existencia de una metodología de búsqueda de caminos



Se evidencia que la mayoría de encuestados opina que es "importante" que exista una metodología en búsqueda de caminos, con un 80%, mientras que solamente el 20% indica que es "muy importante".

Interpretación y discusión de resultados

Del diagnóstico situacional del proceso de búsqueda de caminos, cuya información de los indicadores de la variable dependiente se obtuvo a través de la aplicación de cuestionarios a 46 padres de familia, cuestionario a 5 especialistas en Inteligencia Artificial y una guía de entrevista al encargado de portería de la Institución Educativa, que luego fueron procesadas para destacar los hallazgos más importantes.

La mayoría de padres de familia opinan que han tenido necesidad de localizar algún ambiente u oficina dentro de la Institución Educativa en un contexto diferente a la pandemia del COVID 19, en forma ocasional un 37%, siempre un 21.7% en forma frecuente un 13%. Así mismo, el encargado de portería de la Institución Educativa opinó que siempre hay necesidad de búsqueda de ambientes y oficinas y que genera la necesidad de guiar a los usuarios, porque ellos preguntan con mayor frecuencia la ubicación de la oficina de Dirección, pero que, sin embargo, de acuerdo a su consulta la respuesta la tienen que obtener en otras oficinas. Otra afirmación que hizo es que la amplitud del colegio hace que sea necesario un sistema para la búsqueda de ambientes.

Se aprecia en las respuestas que hay la necesidad de buscar ambientes y oficinas en la institución educativa, y que la falta de procedimiento establecido en la búsqueda de caminos. En este sentido (Alyasin et al., 2019) nos mencionan para solucionar problemas relacionados con localización y construcción de mapas, la búsqueda de caminos es una buena alternativa, y se ha aplicado en diferentes entornos, desde la detección de minas hasta la exploración de planetas. Así mismo, nos indican que existen dos categorías de problemas de localización, hallar la ruta más corta de una sola fuente y la ruta más corta entre

pares de vértices de un mapa. La presente investigación se enmarcará en el primer tipo problema de una sola fuente y un solo nodo objetivo.

Con respecto a llegar a un destino errado, a pesar de que previamente se preguntó por la localización del ambiente buscado, los padres de familia opinaron que "ocasionalmente" sucedía esto con un 34.8%, con un 28.3% se daba de forma frecuente y con un 4.3% que siempre les ocurría esto cuando han preguntado por un ambiente u oficina. Así mismo cuando se le preguntó al encargado de portería sobre la necesidad de ubicación de algún ambiente u oficina de la Institución Educativa, refirió que ha habido ocasiones que a pesar que se les indica a los usuarios (padres de familia, visitantes) el camino para llegar, resultan ingresando al pabellón de las aulas porque se equivocaron de camino, trayendo incomodidad para el docente y pérdida de tiempo para el usuario.

Los movimientos planificados o aleatorios de una búsqueda pueden afectar el tiempo de ejecución e incluso si están mal planificados, puede que no se llegue al destino. Para que se pueda llegar a un destino, se tiene que conocer su ubicación dentro del entorno, que constituyen problemas en el ámbito de la localización y proceso de mapeo en la literatura robótica. Una vez tenga conocimiento de su ubicación, deben elegir el camino óptimo de principio a fin para alcanzar un destino objetivo, donde también se debe delimitar las especificaciones o restricciones dentro del entorno, como pueden ser la presencia de obstáculos. (Korkmaz & Durdu, 2018)

Con respecto a la demora en llegar a un ambiente u oficina en la Institución educativa, por el cual previamente se preguntó, se evidencia que el 45.7% opinó que "ocasionalmente" le sucedió esto y el 10.9% "frecuentemente".

El factor tiempo siempre es importante al momento de realizar búsqueda de caminos, cuando se desconoce las rutas para llegar a un destino, trae como consecuencia “problemas pérdida de tiempo durante la búsqueda y el acceso de los puntos de acción (nodos)” (Marchena, 2015).

Si bien existe pérdida de tiempo en las búsquedas de caminos en el entorno real, este problema también se da en los sistemas informáticos creados para ese fin. En tal sentido, el tiempo para la búsqueda de caminos se ve afectado por, la cantidad de metros a recorrer y la cantidad de nodos destinos (puntos importantes como oficinas, ambientes), y esto se tiene que tener en cuenta al momento de desarrollar el software informático. Para poder entender el grado de afectación en el tiempo, se menciona los resultados de una investigación local, “cuando hay un solo nodo destino con 799 metros a recorrer el tiempo de ejecución es de 0.0586 milisegundos, en cambio cuando hay 30 nodos destino y 5462 metros por recorrer, el tiempo de ejecución es de 2.2962 milisegundos.” (Reyes, 2018)

El tiempo es considerado a menudo un criterio de optimalidad para la búsqueda de camino, en tal sentido se debe tener en cuenta al momento de elegir un algoritmo de búsqueda para poder hallar el camino en un tiempo prudencial para el usuario final, ya que de nada nos serviría encontrar la ruta más corta (con menos cantidad de metros en el recorrido) si es que demora mucho tiempo en calcularla y al final obligaría al usuario a abandonar el uso del software. Muchos autores han identificado que el tiempo es uno de los criterios más importantes al momento de analizar la eficiencia en la búsqueda de caminos. Generalmente se busca tener la distancia más corta entre los puntos de inicio y finalización en una búsqueda, sin embargo, por querer cumplir con este objetivo el tiempo de cálculo es muy alto, lo que afecta la continuidad de

la tarea actual y de las tareas sucesivas. En este contexto será mejor evaluar un algoritmo a través de una combinación de criterios como distancia y tiempo.(Korkmaz & Durdu, 2018).

Hay que tener en cuenta que un punto importante que afecta el tiempo de búsqueda es la cantidad de nodos que se tienen que examinar, ya que en cada uno de ellos se tiene que verificar si se ha llegado al nodo final, esto es muy importante sobre todo en contextos donde sea un tipo de búsqueda a ciegas, es decir no haya un mapa que indique la cantidad de nodos y la ubicación de ellos, y el software tiene que ir censando los nodos que se encuentren alrededor de él. La presente investigación esta enmarcada dentro de una investigación informada (no es una investigación a ciegas), sin embargo, es importante tener claro que la cantidad de nodos que se examinen y que no necesariamente son parte de la ruta optima que será la solución al problema, afecta el tiempo de cálculo y por tanto el tiempo de respuesta del sistema informático.

Al comparar tres algoritmos orientados a encontrar la ruta más corta: A*, Búsqueda ávara primero el mejor y Dijkstra, en un primer mapa los tres encontraron la misma ruta, pero en el proceso tuvieron que examinar y expandir nodos para poder encontrarla, el primero expandió 158 nodos, el segundo expandió 806 nodos, y el tercero expandió 1606 nodos (Liburd & Boumedine, 2018). A más cantidad de nodos examinados, mayor es la cantidad de tiempo que necesita el sistema informático para poder encontrar la ruta de búsqueda, por eso es importante elegir el algoritmo que sea más adecuado al tipo de búsqueda y a las características del entorno donde se realiza la búsqueda.

La longitud de la ruta obtenida como resultado de la búsqueda de caminos, también es un criterio de optimalidad. Esta define el espacio de recorrido para unir un punto de partida con un punto de llegada. Varios investigadores tratan de comparar diversos algoritmos orientados a obtener la ruta más corta, obteniendo ventajas y desventajas en cada uno de ellos. Una investigación internacional comparó los algoritmos de A*, Dijkstra y "búsqueda avara primero el mejor", encontrando que las dos primeras hallaron la ruta más corta (cumplieron con el criterio de admisibilidad), en cambio el método avaro primero el mejor, a pesar que fue el que expandió menor cantidad de nodos, su ruta encontrada fue mucho más larga (Liburd & Boumedine, 2018). Esto se debe porque el primer algoritmo expande el nodo teniendo en cuenta la suma de una distancia estimada al objetivo y una distancia estimada desde el punto de partida, en cambio el ultimo algoritmo se inclina por buscar en las rutas que están más cerca al objetivo sin analizar el coste de distancias desde el inicio, sin embargo, este último algoritmo es el que obtuvo menor tiempo de cálculo, ya que es el que examina menos cantidad de nodos.

En este aspecto, se debe tener en cuenta que el algoritmo elegido sea el correcto según las características del problema a resolver, y es una de las mayores dificultades que afrontan los investigadores en inteligencia Artificial, ya que, de 5 investigadores encuestados, se obtuvo que el 60% del 100% de ellos obtuvo dificultad en aplicar los algoritmos a contexto reales. Así mismo, cuando se les preguntó que algoritmos de búsqueda de caminos conocen, el 60% de ellos respondió que conocen tres algoritmos, el algoritmo primero en profundidad, el algoritmo primero en amplitud (ambos algoritmos pertenecientes al tipo de búsqueda a ciegas) y el algoritmo A* (algoritmo

perteneciente al tipo de búsqueda informada). También se evidenció que confunden los algoritmos destinados para planificación de rutas, con los que búsquedas de caminos, ya que en esta pregunta marcaron vecino más cercano, Dijkstra, búsqueda Tabú, que son algoritmos pertenecientes a planificación de rutas.

Para ilustrar como afecta el algoritmo elegido para resolver un problema de búsqueda de caminos en el tiempo de cálculo de la solución, se cita los resultados obtenidos en una investigación internacional, donde se compararon 5 algoritmos diferentes para calcular una misma ruta (mismo nodo de inicio y mismo nodo de finalización), con el primer algoritmo se obtuvo un tiempo de 76,96 s y una distancia de 368 cm, con el segundo algoritmo se obtuvo un tiempo de 1.98 s y 434 cms., con el tercer algoritmo se obtuvo 24,62 s y 524 cms., con el cuarto algoritmo 0,95 s y 397 cms., y con el ultimo 2.38 s y 446 cms.(Korkmaz & Durdu, 2018)

Se puede apreciar que la diferencia entre el algoritmo que obtuvo mayor tiempo (79.96 segundos) y el que tuvo menor tiempo (0,95 segundos) es de 79.01 segundos, tiempo más que suficiente para que cualquier usuario abandone el uso del software y sea calificado como solución inviable para ese contexto.

Por otro lado, con respecto a la disposición que tienen los padres de familia para usar un software informático en la búsqueda de caminos de ambientes y oficinas, el 47.8% está "de acuerdo" y el 21.7% está "muy de acuerdo". En cuanto al encargado de portería, cuando se le hizo la pregunta relacionada con la importancia de contar con una aplicación que desde el celular le ayude a la

búsqueda de oficinas, respondió que considera que es muy importancia y que será de gran ayuda para poder ubicar a las principales oficinas.

En la actualidad las compañías exigen manejar los materiales necesarios para sus procesos productivos cada vez con más rapidez, con más precisión y con más exactitud y es allí donde los sistemas informáticos de búsqueda de caminos ayudan a automatizar el proceso de localización geográfica de nodos dentro de un árbol de búsqueda.(Pumaricra Rojas, 2019)

Los sistemas informáticos con algoritmos de búsqueda heurísticos suelen aplicarse e implementarse en un espacio de búsqueda pequeño y simultáneamente producir soluciones eficientes en un tiempo informático limitado.(Konstantakopoulos et al., 2020).

En tal sentido, vemos que se hace necesario la realización de un sistema informático de búsqueda de caminos, teniendo en cuenta el tamaño del espacio de búsqueda y la cantidad de nodos, que influirán en la elección del algoritmo de búsqueda.

Con respecto a la ubicación espacial se debe determinar la forma de localización interna del agente dentro del sistema informático y la sincronización con la localización externa del usuario dentro de la Institución Educativa. En tal sentido se debe escoger el sistema de trabajo con el tipo de coordenadas que se va utilizar para localizar al usuario, siendo las coordenadas proyectadas siendo el sistema geodésico WGS84 con quien trabaja el GPS "Sistema de posicionamiento global" en latinoamerica.

En el caso de movilidad en interiores dentro de edificios, el problema está en que no se puede usar GPS, por tanto, hay falta de precisión en el sistema de posicionamiento. El robot no conoce su posición exacta, solo calcula qué tan

lejos se ha desplazado y actualiza su índice de posición en la cuadrícula de acuerdo a esos valores.(Angkuldee et al., 2019).

Las rutas que se trazarán serán en ambientes externos, es decir no será dentro de una oficina, sino fuera de las oficinas y demás ambientes de la institución educativa, y se optará por la utilización del GPS para poder ubicar correctamente al usuario.

CONCLUSIONES

Se determinó las tendencias históricas sobre el proceso de búsqueda de caminos, indicando que al comienzo se buscaba la planificación de rutas para mejorar la logística de las empresas que permitan solucionar problemas del tipo TSP y VRP a través de algoritmos de Investigación de operaciones, y que al hacerse más complejo, se vio la necesidad de interactuar con algoritmos de agentes resolventes de problemas de Inteligencia Artificial, aplicándose actualmente a diversos campos como son videojuegos, robótica, vehículos auto conducidos, redes y comunicaciones, turismo, evidenciándose una alta necesidad en organizaciones públicas y privadas en la construcción de este tipo de sistemas de búsqueda.

Se precisaron técnicas e instrumentos de recolección de datos que luego serán procesados con métodos estadísticos utilizando el software ofimático Microsoft Excel y el procesador estadístico SSPS para el análisis e interpretación de datos. Así mismo, la información será resumida en tablas y gráficos estadísticos de Microsoft Excel. También se establecieron los criterios éticos y de rigor científico que regirán la investigación teniendo en cuenta el informe Belmont que guía los principios éticos en una investigación científica.

Finalmente se diagnosticó el estado actual del proceso de búsqueda de caminos en una institución educativa de Educación básica regular de la ciudad de Lambayeque – Perú, mediante la aplicación de instrumentos validados por expertos: una encuesta a padres de familia de la Institución Educativa y una entrevista al encargado de atención al usuario y portería, en la que se observó la ausencia de procedimientos adecuados que ocasionaban malestar en el usuario, ya que el 34.8% opinaban

ocasionalmente llegaban a un destino errado a pesar de que previamente se preguntó por la localización del ambiente buscado. También se aplicó una encuesta a investigadores en Inteligencia Artificial para determinar la falta de una metodología específica para búsqueda de caminos, opinando la mayoría de encuestados en una 80% que debería crearse una metodología específica para la búsqueda de caminos.

REFERENCIAS BIBLIOGRÁFICAS

- Acuña, H. (2018). Aplicación de un modelo de ruteo de vehículos para optimizar el recorrido en el servicio de visitas turísticas. In *Universidad Nacional Mayor de San Marcos*. Universidad Nacional Mayor de San Marcos.
- Alyasin, A., Abbas, E., & Hasan, S. (2019). An Efficient Optimal Path Finding for Mobile Robot Based on Dijkstra Method. *4th Scientific International Conference – Najaf – IRAQ*.
<https://doi.org/10.1109/SICN47020.2019.9019345>
- Amit, B., & Sandeep Joshi. (2020). An agent based routing search methodology for improving QoS in MANET. *Revista Chilena de Ingeniería*, 28(4), 558–564.
<http://web.b.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=0&sid=4ed3354e-9ead-4edc-95ab-d627d1562da4%40sessionmgr101>
- Angkuldee, A., Shih, K. P., & Ruengittinun, S. (2019). Apply A* Search for Robot Path Finding. *Proceedings - 2019 12th International Conference on Ubi-Media Computing, Ubi-Media 2019*, 183–186. <https://doi.org/10.1109/Ubi-Media.2019.00043>
- Anton, J. (2018). DESARROLLO DE UN PLANIFICADOR DE RUTAS PARA RECOJO DE DESECHOS SÓLIDOS EN EL DISTRITO DE CHICLAYO UTILIZANDO ALGORITMO DE DIJKSTRA. In *Anales de la Universidad de Chile* (Vol. 0, Issue 13).
<https://repositorio.uss.edu.pe/handle/20.500.12802/4666>
- Comisión Europea; (2020). *White Paper on Artificial Intelligence A European approach to excellence and trust*. https://ec.europa.eu/info/files/white-paper-artificial-intelligence-european-approach-excellence-and-trust_es
- Comisión nacional para la protección de los sujetos humanos de investigación biomédica y del Comportamiento; (1979). *EL INFORME BELMONT PRINCIPIOS Y GUÍAS ÉTICOS PARA LA PROTECCIÓN DE LOS SUJETOS HUMANOS DE INVESTIGACIÓN*. [www.bioeticayderecho.ub.es-
www.bioeticaidret.cat](http://www.bioeticayderecho.ub.es-www.bioeticaidret.cat)
- Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2001). *Introduction To Algorithms* (2da Ed.).
https://books.google.com.pe/books?id=NLngYyWFI_YC&pg=PA540&dq=breadth-first+search&hl=es&sa=X&ved=2ahUKEwiTtef9l4LyAhU2GbkGHQASCglQ6AEwAXoECAYQAg#v=onepage&q=breadth-first search&f=false
- Fernandez, A. (2016). *Algoritmos heurísticos y metaheurísticos basados en búsqueda local aplicados a problemas de rutas de vehiculos*. Universidad de Valladolid.
- Gómez, R. (2020). COMPILACION EN PROGRAMACION DE CONJUNTOS DE RESPUESTAS PARA EL PROBLEMA DE BUSQUEDA DE CAMINOS CON MULTIPLES AGENTES. *Tesis de Pregrado*, 1–74.
<https://www.proquest.com/openview/87e180d74d8317ec4b29c18b5fa8660b/1?pq-origsite=gscholar&cbl=2026366&diss=y>
- Kolhe, M. L., Munesh, ., Trivedi, C., Tiwari, S., Vikash, ., & Singh, K. (2018). *Lecture Notes in Networks and Systems 38* (Vol. 1). <http://www.springer.com/series/15179>
- Konstantakopoulos, G. D., Gayialis, S. P., Kechagias, E. P., Papadopoulos, G. A., & Tatsiopoulos, I. P. (2020). A multiobjective large neighborhood search metaheuristic for the vehicle routing problem with time windows. *Algorithms*, 13(10). <https://doi.org/10.3390/a13100243>

- Korkmaz, M., & Durdu, A. (2018). Comparison of optimal path planning algorithms. *14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, 255–258. <https://doi.org/10.1109/tcset.2018.8336197>
- Liburd, K., & Boumedine, M. (2018). Engaging undergraduates in artificial intelligence: Path finding in game development. *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018*, 1475–1478. <https://doi.org/10.1109/CSCI46756.2018.00294>
- Marchena, D. J. (2015). Desarrollo de un sistema para la optimización de rutas de trabajo utilizando el algoritmo de Dijkstra y diagramas de Voronoi. *Repositorio Institucional - USS*. <https://repositorio.uss.edu.pe/handle/20.500.12802/157>
- Mendoza, O. D. (2017). Diseño e implementación de un circuito turístico inteligente en la Región Puno mediante la metaheurística Búsqueda Tabú. In *Universidad Peruana Unión*. http://repositorio.upeu.edu.pe/bitstream/handle/UPEU/899/Oscar_Mendoza_Tesis_Bachiller_2017.pdf?sequence=1&isAllowed=y
- OCDE. (2019). *PISA 2018 Results (Volume I) WHAT STUDENTS KNOW AND CAN DO*. <https://doi.org/10.1787/5f07c754-en>
- Pariona, W. (2019). Evaluación de un sistema de búsqueda de rutas de evacuación eficientes de un establecimiento usando el algoritmo D estrella (D*). *Actas Del II Congreso Internacional de Ingeniería de Sistemas. Innovando La Educación En Tecnología*.
- Pramono, K., Wijaya, K., Cuosman, W., Hartanto, D., Dharma, A., & Wardani, S. (2019). Shortest Path Search Simulation on Busway Line using Ant Algorithm. *Journal of Physics: Conf. Series*, 12094. <https://doi.org/10.1088/1742-6596/1230/1/012094>
- Pumaricra Rojas, D. R. (2019). Desarrollo de un software para la gestión de vehículos con generación y planificación automática de rutas [Universidad Nacional de Ingeniería]. In *Universidad Nacional de Ingeniería*. <http://cybertesis.uni.edu.pe/handle/uni/17832>
- Putri, K. A., Rachmawati, N. L., Lusiani, M., Ngurah, A. A., & Redi, P. (2021). Genetic Algorithm with Cluster-first Route-second to Solve the Capacitated Vehicle Routing Problem with Time Windows: A Case Study. *Jurnal Teknik Industri*, 23(1). <https://doi.org/10.9744/jti.23.1.75-82>
- Quiroz, J., & Ramirez, K. (2019). Sistema de búsqueda heurística de las principales oficinas de la Universidad nacional Pedro Ruíz Gallo utilizando el algoritmo A Star [Universidad Nacional Pedro Ruíz Gallo]. In *Universidad Nacional Pedro Ruíz Gallo*. <https://repositorio.unprg.edu.pe/bitstream/handle/20.500.12893/8314/BC-4714 QUIROZ DAVILA-RAMIREZ LOPEZ.pdf?sequence=1&isAllowed=y>
- Reyes, J. (2018). DESARROLLO DE UN PLANIFICADOR DE RUTAS PARA RECOJO DE DESECHOS SÓLIDOS UTILIZANDO ALGORITMO DE BELLMAN FORD. *Anales de La Universidad de Chile*, 0(13), Pág. 95-131-131. <https://repositorio.uss.edu.pe/handle/20.500.12802/5779>
- Ríos, A. (2015). Desarrollo de un algoritmo planificador de rutas con capacidad de implementación en diversas aplicaciones de la robótica móvil. In *tesis de pregrado*.
- Russell, S., & Norvig, P. (2010). Artificial intelligence: a modern approach. In *Prentice Hall* (Vol. 11, Issue 1). <https://doi.org/10.1017/s0269888900007724>

- Valdivia, A. K. C. (2021). HACIA OTRA DIMENSIÓN JURIDICA: EL DERECHO DE LOS ROBOTS. *REVISTA IUS*, 15(48). <https://doi.org/10.35487/RIUS.V15I48.2021.681>
- Yijun, Z., Jiadong, X., & Chen, L. (2021). A Fast Bi-directional A* Algorithm Based on Quad-tree Decomposition and Hierarchical Map. *IEEE Access*, 1–1. <https://doi.org/10.1109/ACCESS.2021.3094854>
- Zhang, D., Mishra, S., Brynjolfsson, E., Etchemendy, J., Ganguli, D., Grosz, B., Lyons, T., Manyika, J., Niebles, J., Sellitto, M., Shoham, Y., Clark, J., & Perrault, R. (2021). *Artificial Intelligence Index Report 2021*.
- Zhang, H., Minyong, S., & Chunfang, L. (2016). Research and Application of Path-finding Algorithm Based on Unity 3D. *IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. <https://doi.org/10.1109/ICIS.2016.7550934>



Ivan Adrianzén Olano

Universidad Nacional Toribio Rodríguez de Mendoza de Amazonas – Amazonas, Perú
<https://orcid.org/0000-0002-1910-2854>
ivan.adrianzen@untrm.edu.pe

Ingeniero en Computación e Informática, Maestro en Ciencias de la Educación con mención en Docencia y Gestión Universitaria por la Universidad Nacional Pedro Ruiz Gallo, estudios concluidos de Maestría en Ingeniería de Sistemas con mención en Sistemas de Información en Universidad Privada Antenor Orrego de Trujillo. Docente nombrado en la Universidad Nacional Toribio Rodríguez de Mendoza de Amazonas – Perú, adscrito al Departamento Académico de Ingeniería de la Facultad de Ingeniería de Sistemas y Mecánica Eléctrica.

Gisella Luisa Elena Maquen Niño

Universidad Nacional Pedro Ruiz Gallo – Lambayeque, Perú
<https://orcid.org/0000-0002-9224-5456>
gmaquenn@unprg.edu.pe

Ingeniero en Computación e Informática, Licenciada en Educación con especialidad en Matemática y Computación. Maestra en Ciencias de la Educación con mención en tecnologías de la información e informática educativa. Doctora en Ciencias de la Educación con mención en Administración de la Educación. Doctorado en Ciencias de la Computación y Sistemas en trámite. Docente nombrada en la Universidad Nacional Pedro Ruiz Gallo de Lambayeque – Perú, adscrita al Departamento académico de Computación y Electrónica. Asesora de tesis de pregrado y posgrado con más de 10 años en la docencia universitaria.



Alejandro Chayán Coloma

Universidad Nacional Pedro Ruiz Gallo – Lambayeque, Perú
<https://orcid.org/0000-0003-2445-5037>
achayanc@unprg.edu.pe

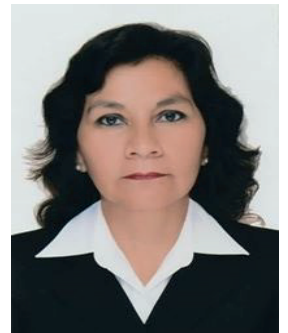
Ingeniero en Computación e Informática, Maestro en Ingeniería de Sistemas, actualmente cursando el doctorado de Tecnologías de la Información y Comunicaciones, en la Universidad Nacional de Piura. Docente adscrito al Departamento Académico de Computación y Electrónica de la Universidad Nacional Pedro Ruiz Gallo de Lambayeque - Perú.



Diana Mercedes Castro Cárdenas

Universidad Nacional Pedro Ruiz Gallo – Lambayeque, Perú
<https://orcid.org/0000-0001-8489-9671>
dcastroc@unprg.edu.pe

Licenciada en Matemática. Maestra en Ciencias con mención en Matemática Aplicada. Doctora en Educación. Docente ordinario en la Universidad Nacional Pedro Ruiz Gallo de Lambayeque – Perú, adscrito al Departamento académico de Matemáticas. Más de 25 años de experiencia en la docencia universitaria.



Franklin Edinson Terán Santa Cruz

Universidad Nacional Pedro Ruiz Gallo – Lambayeque, Perú
<https://orcid.org/0000-0002-3197-7979>
fteran@unprg.edu.pe

Ingeniero en Computación e Informática y Maestro en Ingeniería de Sistemas con mención en Gerencia de tecnologías de la información y gestión del software. Docente nombrado en la Universidad Nacional Pedro Ruiz Gallo de Lambayeque – Perú, adscrito al Departamento académico de Computación y Electrónica y asesor de tesis de pregrado con más de 10 años en la docencia universitaria y más de 20 años como desarrollador independiente.



Freddy William Campos Flores

Universidad Nacional Pedro Ruiz Gallo – Lambayeque, Perú
<https://orcid.org/0000-0002-9624-2930> fcampos@unprg.edu.pe

Ingeniero en Computación e Informática. Maestro en Ingeniería de Sistemas, Universidad César Vallejo, Lambayeque, Perú. Docente nombrado en la Universidad Nacional Pedro Ruiz Gallo de Lambayeque – Perú, adscrito al Departamento académico de Computación y Electrónica.





ISBN: 978-9942-603-34-0

