



FACULTAD DE INGENIERÍA

CARRERA PROFESIONAL DE INGENIERÍA DE SOFTWARE

**Modelo de Asignación Dinámica de Tareas Utilizando el Sistema
Multiagente en la Policía Nacional del Perú de la ciudad de Arequipa**

**Tesis presentada por:
Sergio Miguel Dueñas Vera**

**Asesor:
Dr. Yasiel Pérez Vera**

Para optar por el título profesional de:

Ingeniero de Software

AREQUIPA – PERÚ

2024

Modelo de Asignación Dinámica de Tareas Utilizando el Sistema Multiagente en la Policía Nacional del Perú de la ciudad de Arequipa

INFORME DE ORIGINALIDAD

2 %	2 %	1 %	0 %
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	hdl.handle.net Fuente de Internet	< 1 %
2	repositorio.unp.edu.pe Fuente de Internet	< 1 %
3	"Evaluación de interfaces de visualización de modelo de tópicos de consultas ciudadanas : un estudio de caso del proceso constituyente participativo del 2016 en Chile", Pontificia Universidad Catolica de Chile, 2021 Publicación	< 1 %
4	dspace.otalca.cl Fuente de Internet	< 1 %
5	alicia.concytec.gob.pe Fuente de Internet	< 1 %
6	Submitted to Universidad Manuela Beltrán Trabajo del estudiante	< 1 %
7	repositorio.unu.edu.pe Fuente de Internet	

		<1 %
8	core.ac.uk Fuente de Internet	<1 %
9	www.revespcardiol.org Fuente de Internet	<1 %
10	acervodigital.ufpr.br Fuente de Internet	<1 %
11	arribasalud.com Fuente de Internet	<1 %
12	es.scribd.com Fuente de Internet	<1 %
13	www.coursehero.com Fuente de Internet	<1 %
14	www.slideshare.net Fuente de Internet	<1 %
15	www.unify.com Fuente de Internet	<1 %
16	Babeş-Bolyai University Publicación	<1 %
17	colposdigital.colpos.mx:8080 Fuente de Internet	<1 %
18	peru21.pe Fuente de Internet	<1 %

19	sedici.unlp.edu.ar Fuente de Internet	<1 %
20	www.oreilly.com Fuente de Internet	<1 %
21	Encyclopedia of Complexity and Systems Science, 2009. Publicación	<1 %
22	ensani.ir Fuente de Internet	<1 %
23	libros.catedu.es Fuente de Internet	<1 %
24	repositories.lib.utexas.edu Fuente de Internet	<1 %
25	test.contactoet.com Fuente de Internet	<1 %
26	www.sodexo.es Fuente de Internet	<1 %
27	"Inter-American Yearbook on Human Rights / Anuario Interamericano de Derechos Humanos, Volume 17 (2001)", Brill, 2005 Publicación	<1 %

Excluir citas

Activo

Excluir coincidencias

Apagado

Excluir bibliografía

Activo

Dedicatoria

Con amor y gratitud a mi querida madre Naldy que me demostró su gran amor y apoyo incondicional en todo momento, y a mi hermana Pamela que siempre me dio palabras de aliento.

Agradecimientos

Agradecer a Dios por permitirme alcanzar esta meta tan anhelada.

Mi reconocimiento y gratitud eterna a la Universidad “La Salle” mi alma mater, por haberme formado integralmente como persona y profesional

A mis maestros que fueron mis mentores y vieron en mí una promesa profesional

A mi maestro asesor que con mucho profesionalismo me oriento en el desarrollo de este trabajo de investigación.

Índice General

Dedicatoria	v
Agradecimientos	vi
Índice General	vii
Índice de Abreviaturas y Siglas	xi
Índice de Tablas.....	xii
Índice de Figuras	xiii
Resumen.....	xiv
Palabras clave	xv
Abstract	xvi
Keywords	xvii
Capítulo I - Planteamiento de la Investigación	1
1.1. Planteamiento del problema	1
1.2. Objetivos de la Investigación	3
1.3. Tipo y Nivel de Investigación	4
1.4. Preguntas de Investigación.....	5
1.5. Justificación	5
Capítulo II - Revisión y Fundamentación Teórica	7
2.1. Estado del Arte.....	7
2.2. Antecedentes Investigativos.....	7
2.2.1. Estrategia para la elaboración de los antecedentes investigativos	7
2.2.2. Aplicación de estrategia para la elaboración de los antecedentes investigativos	9
2.3. Antecedentes de proyectos.....	12
2.4. Fundamentos teóricos.....	20
2.4.1. Inteligencia artificial	20
2.4.2. Agentes	21
2.4.3. Tipos de entornos para dispositivos inteligentes agentes	25

2.4.4. Asignación de Tareas Dinámicas	27
2.4.4.1 Definición y contexto.....	27
2.4.4.2 Importancia	28
2.4.4.3 Problemáticas	28
2.4.4.4 Futuro.....	29
2.4.5. Herramientas para el desarrollo de Agentes Inteligentes.....	29
Capítulo III - Solución propuesta	31
3.1. Recopilar información	31
3.1.1 Entrevistas con la Policía Nacional del Perú	32
3.1.2 Análisis de Documentos Estatales.....	32
3.2. Diseño de Modelo.....	32
3.2.1. Diseño general de la aplicación	32
3.2.2. Diseño General de un Agente.....	34
3.2.3. Diseño del Aprendizaje de un Agente.....	35
3.3. Codificación del Modelo	36
3.3.1. Codificación de la Aplicación	36
3.3.1.1 Requisitos del Sistema	36
3.3.1.2 Base de Datos	37
3.3.1.3 Programación.....	39
3.3.2. Codificación del Sistema multiagente.....	39
3.4. Entrenamiento del modelo	42
3.4.1 Inicialización del Entorno	42
3.4.2 Toma de Decisiones de los Patrulleros	43
3.4.3 Actualización del Estado del Entorno	43
3.4.4 Cálculo de Recompensa	44
3.4.5 Aprendizaje y Actualización de Parámetros.....	44
3.5. Diseño de las Pruebas	48

Capítulo IV: Resultados y Discusión.....	51
4.1. Recopilar información	51
4.1.1 Entrevistas con la Policía Nacional del Perú	51
4.1.2 Análisis de Documentos Estatales.....	51
4.2. Diseño Modelo.....	52
4.2.1 Diseño General de un Agente	52
4.2.2 Diseño del Aprendizaje de un Agente.....	53
4.3. Codificación del Modelo	54
4.4. Entrenamiento del modelo	55
4.4.1 Inicialización del Entorno	56
4.4.2 Toma de Decisiones de los Patrulleros	57
4.4.3 Actualización del Estado del Entorno	58
4.4.4 Cálculo de Recompensa	60
4.4.5 Aprendizaje y Actualización de Parámetros.....	61
4.5. Pruebas	62
4.6. Resultados de la Investigación.....	64
4.7. Discusión de los Resultados.....	67
4.7.1. Análisis del Tiempo de Ejecución del Algoritmo	67
4.7.2. Comparativas Modificando los Hiperparámetros.....	69
4.7.2.1. Comparativa de Resultados para Diferentes Valores de ϵ	69
4.7.2.2. Comparativa de Resultados para Diferentes Valores de α	70
4.7.2.3. Comparativa de Resultados para Diferentes Valores de γ	71
4.7.3. Interpretación de los Resultados.....	72
Conclusiones.....	74
Recomendaciones	76
Referencias	77
Anexos	81

Anexo 1: Recojo de información de la PNP	81
Anexo 2: Cuestionario aplicado al personal de la PNP	82
Anexo 3: Requisitos del Sistema	83

Índice de Abreviaturas y Siglas

AI	Artificial Intelligence (Inteligencia Artificial)
LEPH	Law Enforcement problem Heterogeneous (Problema de aplicación de la ley)
MAS	Multi Agent System (Sistema multi agente)
ML	Machine learning (Aprendizaje Automático)
DRL	Deep Reinforcement Learning (Aprendizaje profundo por refuerzo)
RL	Reinforcing Learning (Aprendizaje Reforzado)
MAPF	Multi-Agent Path Finding (Búsqueda de ruta de múltiples agentes)
API	Application Programming Interface (Interfaz de programación de aplicaciones)
ORS	Open Route Service (Servicio de ruta abierta)

Índice de Tablas

Tabla 3.1: Registro de Denuncias	38
Tabla 4.1: Resultados de las Pruebas Unitarias, de Aceptación y de Humo	62
Tabla 4.2: Medición de Latencia y Tiempo de Respuesta.....	64
Tabla 4.3: Monitoreo del Uso de Recursos del Sistema.....	64
Tabla 4.4: Aceptación de tecnología por parte del personal policial de la PNP	65
Tabla 4.5: Opinión del personal policial sobre la digitalización de asignación de tareas	66
Tabla 4.6: Disponibilidad del personal policial para capacitación tecnológica	66
Tabla 4.7: Aceptación del personal policial para digitalizar la información policial	66

Índice de Figuras

Figura 2.2: Distribución de artículos encontrados en la revisión.....	10
Figura 2.3: Criterios de exclusión aplicados a los artículos.	11
Figura 2.4: Criterios de eliminación aplicados a los artículos.	12
Figura 2.5: Composición de un agente inteligente en el sistema.	21
Figura 3.1: Esquema del modelo propuesto en la tesis.....	31
Figura 3.2: Flujo de actividad para asignación de denuncias.....	33
Figura 3.3: Estructura del Sistema Multiagente Policial.	34
Figura 3.4: Proceso de aprendizaje de un agente virtual.	35
Figura 3.5: Tabla de Denuncias.	38
Figura 3.6: Diagrama de Patrullaje de Policías en el Sistema Multiagente.	40
Figura 3.7: Diagrama de Respuesta del Patrullero Más Cercano ante una Denuncia en el Sistema Multiagente.	41
Figura 4.1: Proceso de inicialización del entorno simulado.....	56
Figura 4.2: Decisiones de patrulleros ante denuncias.....	57
Figura 4.3: Estado inicial del entorno tras una denuncia.....	58
Figura 4.4: Estado intermedio del entorno en respuesta.....	59
Figura 4.5: Estado final del entorno post-respuesta.....	60
Figura 4.6: Tiempo de Ejecución vs Número de Agentes.....	67
Figura 4.7: Distancia de Ejecución vs Número de Agentes.....	68
Figura 4.8: Comparativa de Resultados para Diferentes Valores de ϵ	69
Figura 4.9: Comparativa de Resultados para Diferentes Valores de α	70
Figura 4.10: Comparativa de Resultados para Diferentes Valores de γ	71
Figura 4.11: Tiempo Real vs Tiempo Agente por Mes.....	72
Figura 4.12: Distancia Real vs Distancia Agente por Mes.....	73

Resumen

La Policía Nacional del Perú desempeña un papel crucial en la protección y seguridad de los ciudadanos. No obstante, los métodos tradicionales de asignación de tareas han demostrado ser insuficientes ante la creciente complejidad de los desafíos delictivos. Este estudio se centra en mejorar la asignación de tareas en la policía de Arequipa mediante un enfoque innovador: un modelo de asignación dinámica de tareas utilizando un sistema multiagente respaldado por inteligencia artificial.

La investigación se llevó a cabo en la ciudad de Arequipa, impulsada por la necesidad de optimizar la respuesta policial a las denuncias ciudadanas y reconociendo la importancia de la tecnología en la modernización de los métodos policiales. Se realizó una revisión sistemática de la literatura para fundamentar conceptualmente el modelo propuesto, seguida de un diseño e implementación práctica utilizando técnicas de ingeniería de software y aprendizaje automático.

El modelo desarrollado, denominado LEPH (Law Enforcement Problem Heterogeneous), se basa en la idea de asignar dinámicamente múltiples agentes a tareas variadas y desconocidas en tiempo real. Esto permite una distribución óptima de recursos humanos y una respuesta más rápida y efectiva a los incidentes delictivos, considerando factores como la ubicación, la urgencia y la importancia de cada tarea.

Los resultados obtenidos muestran una mejora significativa en la eficiencia y efectividad de la policía de Arequipa. Se evidenció una reducción en el tiempo de respuesta a las denuncias del 30%, aunque fue necesario aumentar el presupuesto de la policía en un 3%. Además, los análisis revelaron una reducción notable en la distancia real recorrida por los patrulleros, destacando la eficiencia del modelo en acercar a los agentes a las denuncias. También se identificó que mantener un número óptimo de agentes es crucial para evitar incrementos significativos en la distancia de ejecución.

Palabras clave

Asignación dinámica, Problema aplicando ley, Sistema Multiagente, Aprendizaje reforzado.

Abstract

The National Police of Peru plays a crucial role in the protection and security of citizens. However, traditional task assignment methods have proven insufficient in the face of the increasing complexity of criminal challenges. This study focuses on improving task assignment in the Arequipa police through an innovative approach: a dynamic task assignment model using a multi-agent system supported by artificial intelligence.

The research was conducted in the city of Arequipa, driven by the need to optimize police response to citizen complaints and recognizing the importance of technology in modernizing police methods. A systematic literature review was conducted to conceptually underpin the proposed model, followed by practical design and implementation using software engineering and machine learning techniques.

The developed model, called LEPH (Law Enforcement Problem Heterogeneous), is based on the idea of dynamically assigning multiple agents to various and unknown tasks in real-time. This allows for optimal distribution of human resources and a faster and more effective response to criminal incidents, considering factors such as location, urgency, and the importance of each task.

The results obtained show a significant improvement in the efficiency and effectiveness of the Arequipa police. There was a 30% reduction in response time to complaints, although it was necessary to increase the police budget by 3%. Additionally, the analysis revealed a notable reduction in the actual distance traveled by patrol cars, highlighting the model's efficiency in bringing agents closer to complaints. It was also identified that maintaining an optimal number of agents is crucial to avoiding significant increases in execution distance.

Keywords

Dynamic assignment, Problem applying law, Multiagent System, Reinforcement learning

Capítulo I - Planteamiento de la Investigación

1.1. Planteamiento del problema

La Policía Nacional del Perú es una institución que cumple una función importante dentro de la sociedad, pues contribuye a resguardar la seguridad y bienestar de los ciudadanos, indiscutiblemente es cierto que pese a los esfuerzos del personal de la policía por cumplir con eficiencia su tarea diaria, no siempre logran su objetivo. La violencia y la delincuencia generan un alto grado de inseguridad en el país, obstaculizando el crecimiento económico y la reducción de la pobreza. Se estima que solo se denuncia alrededor del 25 % de los actos delictivos. Estos hallazgos respaldan la afirmación de que la técnica de trabajo de la Policía Nacional del Perú es obsoleta y requiere una alternativa de solución, como la digitalización integral del sistema policial [1].

En la ciudad de Arequipa del Perú, el personal de la policía lleva a cabo patrullajes de rutina y responde a los incidentes denunciados por los ciudadanos. Cada tarea policial puede variar en importancia, desde quejas por ruido hasta casos de mayor gravedad, como asesinatos. La carga de trabajo asociada con cada tarea puede ser diferente, lo que requiere asignar múltiples agentes para trabajar juntos en tareas relevantes y compartirla, lo que mejora la eficiencia y el tiempo de respuesta. Para optimizar la asignación de tareas, es necesario considerar la ubicación y hora, esto permitirá mejorar la coordinación y respuesta de la policía en la ciudad.

Uno de los principales problemas es la distribución desigual de las denuncias y la incapacidad del sistema actual para asignar adecuadamente las tareas en función de la disponibilidad y cercanía de los agentes. Además, la coordinación entre los diferentes agentes en el campo es limitada, lo que puede resultar en duplicidad de esfuerzos o áreas no cubiertas. La necesidad de considerar factores como la ubicación y la hora de llegada añade una capa adicional de complejidad que el sistema actual no maneja de manera efectiva.

Este proyecto de investigación se centra en la implementación de un modelo de asignación dinámica de tareas utilizando un sistema multiagente apoyado por inteligencia artificial. Este modelo busca optimizar la asignación de tareas en la Policía Nacional del Perú, específicamente en Arequipa, mejorando así la respuesta a las denuncias ciudadanas. La utilización de un sistema multiagente permitirá una mejor coordinación y distribución de los recursos policiales, asegurando que las tareas reciban la atención adecuada de manera oportuna.

La investigación aborda la necesidad de modernizar las técnicas de patrullaje y respuesta policial mediante la integración de tecnología avanzada. Al implementar este modelo, se espera reducir los tiempos de respuesta a las denuncias, optimizar el uso de recursos y, en última instancia, mejorar la seguridad y calidad de vida de los ciudadanos en Arequipa. Este enfoque no solo es relevante para la ciudad de Arequipa, sino que también puede servir como un modelo replicable para otras ciudades en Perú y en otros países enfrentando desafíos similares en la asignación de tareas policiales.

La mayoría cree que la inteligencia es exclusiva del ser humano, en los últimos años se ha investigado y se ha logrado un progreso importante sobre inteligencia artificial, pues la búsqueda de una copia artificial del hombre no es nueva. En 1956 Marvin Minsky declaró en su libro “Búsqueda por asalto de la inteligencia artificial” que “El problema del modelado de la Inteligencia Artificial (IA) será resuelto dentro de una generación “[2]. Alan Turing matemático británico demostró que una calculadora universal, ahora conocida como Turing máquina es capaz de resolver cualquier problema siempre que pueda ser representado y resuelto por un algoritmo, para luego ser transferido a la inteligencia humana; esto significa que los procesos cognitivos son algoritmos que se dividen en finitos pasos individuales para luego ser ejecutados por una máquina [2].

Machine Learning (ML) es un subcampo de la IA, pero también se relaciona con muchas otras disciplinas científicas como la estadística, la ciencia cognitiva y teoría de la información [3]. Los algoritmos de ML se organizan en taxonomías según el resultado deseado del algoritmo. Los tipos de algoritmos comunes incluyen: (Aprendizaje supervisado, Aprendizaje no supervisado, Aprendizaje semi-supervisado, Aprendizaje por refuerzo, Predicción, Aprender a aprender) [4].

El Deep Reinforcement Learning (DRL) es un algoritmo que aprende una política de cómo actuar dada una observación del mundo, cada acción tiene algún impacto en el entorno y el entorno proporciona comentarios que guían el algoritmo de aprendizaje. Actualmente en varios proyectos donde se usa AI se está tomando la decisión de basarse en la arquitectura neurocognitiva de Multi Agent System (MAS), la cual tienen una similitud arquitectónica con el cerebro humano [5].

Un agente es todo aquello que percibe su entorno, realiza acciones de forma autónoma para lograr objetivos y puede mejorar su desempeño con el aprendizaje o puede utilizar el conocimiento. Pueden ser simples o complejos: un termostato se considera un ejemplo de un

agente inteligente, al igual que un ser humano, como cualquier sistema que cumpla con la definición, como una empresa o un estado. MAS es una extensión de la tecnología de agentes donde un grupo de agentes autónomos débilmente conectados actúan en un entorno para lograr un objetivo. Esto se hace cooperando o compitiendo, compartiendo o no compartiendo conocimiento entre sí [6].

Law Enforcement Problem (LEPH) asignación en tiempo real de múltiples agentes a tareas desconocidas que llegan en momentos y lugares inéditos. Estas tareas poseen diferentes características, complejidades y urgencia. Algunas tareas pueden ser demasiado grandes o complejas para ser manejadas por un solo agente, mientras que otras tareas pueden requerir múltiples habilidades o campos de experiencia que ningún agente posee. Además, al combinar capacidades y esfuerzo conjunto de varios agentes, el tiempo necesario para completar una tarea puede reducirse, incluso si la tarea no requiere dicha cooperación [7].

La arquitectura interna que tendrán los agentes será una estructura homogénea, ya que todas las patrullas de la policía tienen la misma arquitectura interna, esta se refiere a los objetivos locales, sensor capacidades, estados internos, mecanismo de inferencia y posibles acciones.

1.2. Objetivos de la Investigación

A. Objetivo General

Desarrollar un modelo de asignación dinámica de tareas utilizando el Sistema Multiagente en la Policía Nacional del Perú de la ciudad de Arequipa para mejorar la seguridad ciudadana.

B. Objetivos Específicos

- Investigar los conceptos asociados con los sistemas multiagentes y la asignación dinámica de tareas para comprender el objeto de estudio.
- Diseñar un Modelo de Asignación Dinámica de Tareas Utilizando el Sistema Multiagente para mejorar la seguridad ciudadana.
- Implementar el Modelo de Asignación Dinámica de Tareas Utilizando el Sistema Multiagente en la Policía Nacional del Perú de la ciudad de Arequipa.
- Evaluar el modelo Multiagente para la asignación dinámica de tareas de la policía nacional del Perú en la ciudad de Arequipa.

1.3. Tipo y Nivel de Investigación

Desarrollar un modelo de asignación dinámica de tareas utilizando el Sistema Multiagente en la Policía Nacional del Perú de la ciudad de Arequipa para mejorar la seguridad ciudadana.

Según Hernández Sampieri et al. (2014), existen diferentes clasificaciones para determinar el tipo y nivel de investigación. Sobre el tipo de investigación, se distinguen dos según su propósito: la investigación básica y la investigación aplicada. La investigación básica se enfoca en generar conocimientos novedosos y la exploración de áreas de investigación sin una finalidad práctica inmediata. Por otro lado, la investigación aplicada busca resolver problemas prácticos mediante la aplicación de los descubrimientos encontrados. En el caso de esta investigación, se clasifica como aplicada, ya que tiene como objetivo tomar acciones con la finalidad de resolver problemas prácticos relacionados con la asignación de tareas.

En cuanto al enfoque de investigación, Hernández Sampieri et al. (2014) mencionan tres tipos principales: cualitativo, cuantitativo y mixto. El enfoque cualitativo se centra en investigaciones profundas e intensivas, aplicadas a muestras relativamente pequeñas, con el propósito de interpretar el fenómeno estudiado. El enfoque cuantitativo, por otro lado, busca establecer interrelaciones causales y explicar el fenómeno estudiado. Se utiliza en muestras grandes, con alta representatividad de la población, y se emplea la estadística para el análisis de datos y la generalización del fenómeno. Por último, el enfoque mixto combina elementos de los enfoques cualitativo y cuantitativo.

Considerando la investigación realizada, se clasifica como cuantitativa, pues se comparan los datos estadísticos obtenidos antes y después de la aplicación del LEPH en la Institución para analizar los cambios, además se procede a utilizar métodos estadísticos para el análisis de los datos y así establecer conclusiones basadas en evidencia numérica.

En términos del nivel de profundización, se considera una tesis explicativa, ya que tiene como objetivo profundizar en el tema de la asignación de tareas utilizando la metodología multiagente. A través de la investigación realizada, se busca obtener un mejor entendimiento de dicho tema y proporcionar explicaciones claras y fundamentadas.

1.4. Preguntas de Investigación

- ¿La aplicación del Modelo de Asignación Dinámica de Tareas mediante el sistema multiagente va a permitir mejorar la asignación dinámica de tareas en la Policía Nacional del Perú de la ciudad de Arequipa?
- ¿La simulación del Modelo de Asignación Dinámica de Tareas en el mapa de la ciudad de Arequipa evidenciará su eficacia en la asignación de tareas?
- ¿Los resultados de la presente investigación con la estadística actual de la Institución policial, será evidencia de un producto favorable?

1.5. Justificación

En la actualidad, nos encontramos en la era de la Industria 4.0, también conocida como la cuarta revolución industrial que se caracteriza por el constante avance tecnológico y la aparición de innovaciones que satisfacen las necesidades de la sociedad en diversos ámbitos, como la seguridad ciudadana, la industria, el comercio, la educación y el hogar. Ante esta realidad, resulta imprescindible desarrollar nuevos modelos que optimicen las tareas en instituciones clave [8].

Una de estas instituciones fundamentales es la Policía, encargada de salvaguardar las leyes y garantizar la seguridad civil de los ciudadanos. Por tanto, esta investigación tiene como objetivo contribuir a la optimización de la asignación de tareas a la policía en la ciudad de Arequipa. Mediante la implementación de enfoques y tecnologías innovadoras, se busca beneficiar a la sociedad en general, logrando un uso más eficiente del tiempo y de los recursos económicos disponibles.

Este estudio radica en su impacto positivo en la comunidad, pues al optimizar la asignación de tareas a la policía, se espera mejorar la seguridad y el bienestar de los ciudadanos en la ciudad de Arequipa. Al reducir los tiempos de respuesta, aumentar la eficacia en la prevención del delito y mejorar la capacidad de atención a las necesidades de la comunidad, se generará un resultado positivo en la calidad de vida de los habitantes. Además, al optimizar el uso de recursos, se podrán asignar más efectivamente los presupuestos disponibles, logrando un uso más eficiente y equitativo de los recursos públicos.

Esta investigación es práctica porque se centra en la resolución de un problema concreto. La asignación de tareas a la policía es un desafío complejo, especialmente en entornos urbanos

dinámicos y con recursos limitados. Al desarrollar modelos y enfoques innovadores para la asignación de tareas, se espera mejorar la eficiencia y la efectividad de la labor policial en la ciudad de Arequipa. Esto implica una mejor distribución de los recursos humanos y materiales, una respuesta más rápida a las situaciones de emergencia y una mayor capacidad de adaptación ante los cambios en el entorno. En última instancia, esto contribuirá a fortalecer el sistema de seguridad y a generar un entorno más seguro y confiable para los ciudadanos.

Esta investigación tiene una justificación social y práctica importante. Busca beneficiar a la sociedad mediante la optimización de la asignación de tareas a la policía en la ciudad de Arequipa, mejorando la seguridad ciudadana y la calidad de vida de los habitantes. Además, busca resolver un problema práctico al desarrollar modelos y enfoques innovadores que optimicen el uso de recursos y mejoren la eficiencia de la labor policial.

Capítulo II - Revisión y Fundamentación Teórica

2.1. Estado del Arte

El Aprendizaje Reforzado (RL) aplicado a los MAS de última generación ha logrado un notable éxito en los últimos años. Este éxito se ha basado principalmente en el supuesto de que todos los compañeros de equipo cooperan perfectamente para optimizar un objetivo global y alcanzar una meta común. Sin embargo, en la realidad, los sistemas multiagentes pueden experimentar fallas y problemas de cooperación. Para abordar estas dificultades, existe una aplicación que da un enfoque de juego cooperativo-competitivo en los MAS. En este enfoque, un grupo de agentes protagonistas representa los objetivos de los agentes en el MAS, mientras que un grupo de agentes antagonistas representa las posibles fallas en el MAS. La detección de fallas en configuraciones aparentemente fáciles se puede lograr mediante el uso del algoritmo QMixMax, que se basa en la técnica Minimax de búsqueda adversaria y la poda alfa-beta [9].

La situación actual de la comunicación entre los agentes para la cooperación en la resolución de tareas complejas en los MAS, se enfrentan desafíos adicionales, como la observación parcial, los estados continuos y los espacios de acción. Aunque se han realizado esfuerzos para ampliar el RL tradicional a RL profundo en el dominio de agentes múltiples, todavía existen limitaciones en esta área [10].

Se tiene en cuenta el conjunto de trabajos previos realizados por otros autores o instituciones sobre el tema de investigación, para ello se aplica una estrategia para la elaboración de los antecedentes investigativos, que tiene en cuenta lo siguiente: se procede a identificar las palabras clave, luego se utiliza estrategias de búsqueda, criterios de selección y finalmente la selección de trabajos de investigación útiles para la investigación.

2.2. Antecedentes Investigativos

En esta sección se presentarán los estudios previos y la literatura existente que fundamentan la investigación actual. Se abordarán las diferentes metodologías, enfoques y hallazgos relevantes para el desarrollo del proyecto, proporcionando un marco teórico robusto.

2.2.1. Estrategia para la elaboración de los antecedentes investigativos

Para la elaboración del estado del arte se ha considerado aportes teóricos importantes y los distintos autores que realizaron trabajos relacionados con la presente investigación, en un

proceso que contribuya a obtener una mejor calidad de resultados con respecto al tema de la presente tesis.

A. Cadena de Búsqueda

Para determinar la cadena de búsqueda se selecciona distintas fuentes de información confiables, es decir, se recurre a un conjunto de documentos necesarios para resolver el problema de investigación, para ello se identifica las palabras claves que permite indexar documentos relacionados en conjunto con las estrategias de búsqueda:

*(“Multiagent systems” OR “Autonomous agents” OR “cooperative learning”)
AND (“Task assignment” OR “cooperation” AND (“police officers” OR
“robots”) AND (“simulator” OR “videogames”))*

Se ilustra el proceso sistemático para la recopilación de antecedentes investigativos relevantes para la tesis. Este diagrama representa una serie de pasos secuenciales que aseguran una búsqueda exhaustiva y efectiva en distintas bases de datos científica.

B. Bases de datos científicas

Se utiliza una forma segura y eficiente de almacenar datos útiles para la investigación y posteriormente puedan responder las preguntas que formulen los usuarios, los datos pueden ser números o nombres organizados en filas, denominados registros y columnas denominados campos, se identifican las bases de datos científicas para proporcionar la búsqueda necesaria con respecto a la cadena de búsqueda ya definida, se consideran las siguientes bases de datos:

- ACM Digital Library
- IEEE Xplore Digital Library
- Science Direct
- Scopus
- Springer Link

C. Criterios de inclusión

Para delimitar la población elegible se tiene en cuenta todas las características particulares que debe tener un sujeto u objeto de estudio de la investigación, para ello se ha considerado

investigaciones con implementación de sistema de agentes y algoritmos de enjambre con enfoque a asignación de tareas, además de investigaciones publicadas en inglés exclusivamente, otras investigaciones publicadas con un margen de 5 años de antigüedad máxima, también investigaciones publicadas en conferencias, congresos, revistas y finalmente publicaciones en base de datos fiables expuestas en la sección.

D. Criterios de exclusión

Se tiene en cuenta publicaciones cuyas características impiden su participación en la investigación, porque pueden alterar o modificar los resultados, en este caso se ha excluido publicaciones que no contienen el tema de asignación de tareas o sistemas agentes.

E. Criterios de eliminación

Se considera no tomar en cuenta investigaciones que contienen solo una sola palabra del conjunto principal de la cadena de búsqueda y la aparición de circunstancias que ocurren después de iniciada la investigación.

2.2.2. Aplicación de estrategia para la elaboración de los antecedentes investigativos

Para elaborar los antecedentes investigativos, se ha seguido una estrategia rigurosa de búsqueda y selección de información. A continuación, se detallan los pasos seguidos y los criterios aplicados para garantizar la relevancia y calidad de los estudios seleccionados.

1. Artículos encontrados

Se realizaron búsquedas en varias bases de datos científicas confiables para encontrar documentos relevantes para la tesis, teniendo en cuenta únicamente publicaciones de los últimos 5 años. Los resultados obtenidos fueron los siguientes: en la ACM Digital Library se encontraron 2,653 documentos, en la IEEE Xplore Digital Library se encontraron 896 documentos, en Science Direct se encontraron 2,598 documentos y en Springer Link se encontraron 896 documentos. Estos números representan la cantidad de documentos encontrados en cada base de datos que están relacionados con la cadena de búsqueda previamente definida y publicados dentro del período de los últimos 5 años. Ahora se puede proceder a revisar y analizar estos documentos para determinar su relevancia para la tesis. Es importante tener en cuenta que estos resultados son solo un punto de partida, y es posible que sea necesario refinar la búsqueda o explorar otras fuentes de información para obtener más artículos pertinentes dentro del período específico considerado.

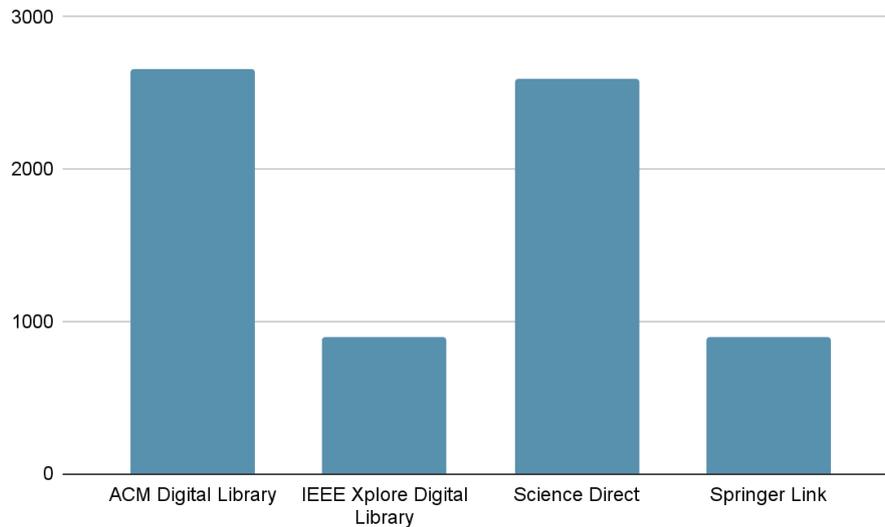


Figura 2.2: Distribución de artículos encontrados en la revisión

Fuente: Elaboración propia.

La Figura 2.2 muestra la distribución de los artículos encontrados durante la revisión literaria. Los artículos se agrupan según diferentes categorías, como el año de publicación, el área temática y la relevancia para el tema de estudio. Esta figura ayuda a visualizar el volumen y la relevancia de la literatura existente, permitiendo una mejor comprensión del estado actual del conocimiento en el campo investigado.

2. Criterios de exclusión

En el proceso de investigación, se aplicaron criterios de exclusión para seleccionar publicaciones relevantes en relación al tema de asignación de tareas o sistemas agentes. Se tuvo en cuenta que las características de algunas publicaciones pudieran interferir o modificar los resultados, lo que llevó a excluir aquellas que no abordaban específicamente este tema. Tras el análisis, se redujo la cantidad de publicaciones de ACM Digital Library a 174, de IEEE Xplore Digital Library a 145, de Science Direct a 450 y de Springer Link a 364. Estos números representan una disminución con respecto a los valores iniciales, donde se tenían 61,623 publicaciones de ACM Digital Library, 896 de IEEE Xplore Digital Library, 21,998 de Science Direct y 896 de Springer Link. El proceso de exclusión se realizó con el objetivo de garantizar que las publicaciones seleccionadas fueran relevantes y proporcionaran información pertinente para la investigación en curso.

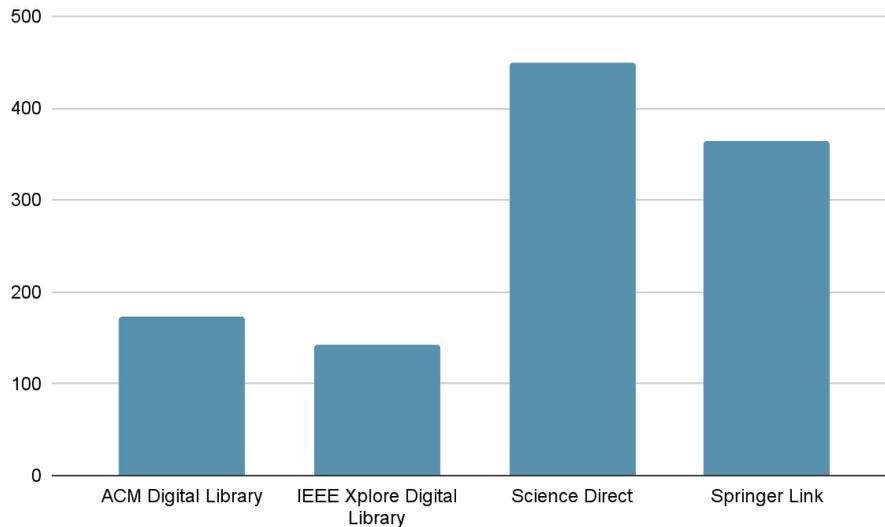


Figura 2.3: Criterios de exclusión aplicados a los artículos.

Fuente: Elaboración propia.

La Figura 2.3 ilustra los criterios de exclusión que se aplicaron a los artículos revisados. Estos criterios incluyen la eliminación de artículos que no estaban en idioma español o inglés, aquellos que no se publicaron en revistas indexadas, y los que no abordaban directamente el tema de investigación. La figura proporciona una visualización clara de cómo se redujo el conjunto inicial de artículos a través de estos filtros, asegurando que solo los estudios más relevantes y de alta calidad fueran considerados.

3. Criterios de eliminación

Además de aplicar criterios de exclusión, también se llevó a cabo un proceso de eliminación para refinar aún más la selección de publicaciones. En esta etapa, se descartaron aquellas que no cumplían con los requisitos específicos y no se consideraban relevantes para la investigación sobre asignación de tareas o sistemas agentes. Después de aplicar el criterio de eliminación, el número de publicaciones se redujo aún más. En la ACM Digital Library, se seleccionaron 66 publicaciones, en IEEE Xplore Digital Library quedaron 23, en Science Direct se encontraron 80 y en Springer Link se eligieron 52. Estos valores representan una disminución significativa en comparación con los números anteriores, lo cual indica un proceso riguroso de selección. El objetivo de esta etapa de eliminación fue asegurar que sólo se consideraran las publicaciones más relevantes y de mayor calidad para el estudio en cuestión. Al reducir la cantidad de publicaciones, se buscó obtener un conjunto más selecto de fuentes que proporcionan información precisa sobre el tema de investigación.

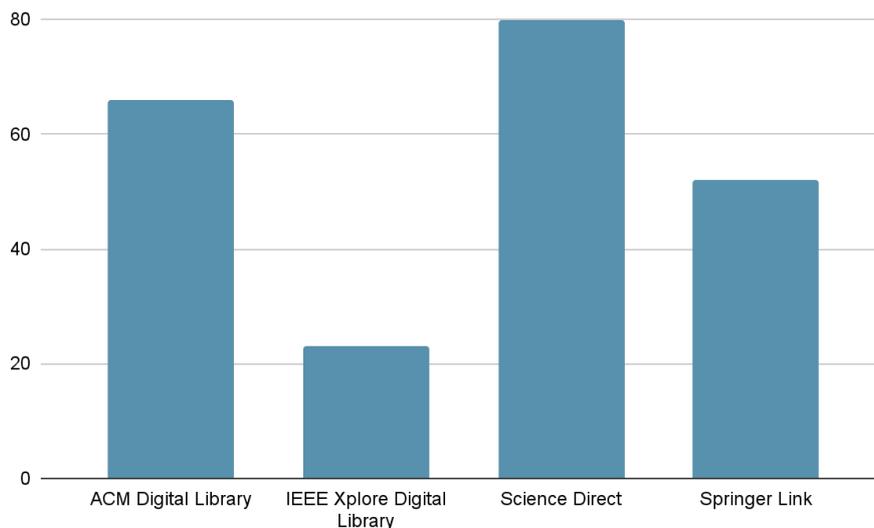


Figura 2.4: Criterios de eliminación aplicados a los artículos.

Fuente: Elaboración propia.

La Figura 2.4 detalla los criterios de eliminación aplicados a los artículos seleccionados. Estos criterios incluyen la eliminación de artículos con datos insuficientes, estudios con sesgos metodológicos, y aquellos cuyos resultados no eran reproducibles. La figura muestra el proceso de refinamiento del conjunto de artículos, garantizando que solo las fuentes más robustas y pertinentes se utilizaron en la revisión final.

2.3. Antecedentes de proyectos

Esta sección revisa proyectos anteriores y actuales que están relacionados con la investigación. Se examinan sus objetivos, metodologías, resultados y lecciones aprendidas, proporcionando un contexto más amplio y una comprensión de cómo se ha abordado el tema en diferentes estudios.

Se seleccionaron los 8 artículos más relevantes relacionados con la asignación dinámica de tareas utilizando sistemas multiagentes. Estos trabajos abordan diversas problemáticas y aplican diferentes enfoques de aprendizaje para mejorar la eficiencia y efectividad en la planificación y ejecución de tareas. En el ámbito del aprendizaje por refuerzo, varios estudios destacan su uso para la planificación y asignación dinámica de tareas. Por ejemplo, uno de los artículos se enfoca en la planificación de tareas dinámicas mediante el aprendizaje por refuerzo, mostrando cómo esta técnica puede optimizar la distribución de recursos en entornos cambiantes, especialmente en situaciones de alta incertidumbre [11]. Otro trabajo relevante aplica un

enfoque profundo de aprendizaje por refuerzo para resolver problemas complejos de programación de talleres, demostrando mejoras significativas en la coordinación y eficiencia operativa, incluso en escenarios con restricciones temporales y recursos limitados [7]. Asimismo, un modelo dinámico basado en el aprendizaje por refuerzo para el rescate de emergencias en aviación ilustra cómo los sistemas multiagentes pueden ser utilizados en situaciones críticas para mejorar la respuesta y coordinación, destacando la capacidad de estos sistemas para adaptarse rápidamente a cambios en el entorno y las necesidades emergentes [12].

Por otro lado, en el contexto de la asignación de tareas y la coordinación entre múltiples agentes, algunos estudios han desarrollado métodos específicos para distintos entornos. La asignación dinámica de tareas en la aplicación de la ley se ha explorado para mejorar la eficiencia y efectividad en la operativa policial, destacando la importancia de una asignación adaptativa y optimizada para responder a incidentes en tiempo real [13]. La búsqueda de rutas en entornos configurables también ha sido un área de interés, donde se ha investigado cómo los agentes pueden adaptar sus rutas de manera óptima en función de las condiciones del entorno, permitiendo una navegación más eficiente y segura [14]. Además, se han diseñado mecanismos de pago para la coordinación en juegos de asignación de tareas, lo cual es crucial para incentivar la cooperación entre agentes autónomos, resolviendo problemas de equidad y motivación en la distribución de tareas [15]. En el ámbito de los vehículos submarinos no tripulados (UUV), se ha propuesto un método de entrenamiento para la asignación dinámica de tareas heterogéneas, mostrando avances en la eficiencia operativa de estos sistemas y su capacidad para realizar misiones complejas y diversas [16]. Finalmente, la adaptación consciente del desgaste para el patrullaje de múltiples agentes ha sido investigada para mejorar la durabilidad y efectividad en tareas de vigilancia, destacando la necesidad de adaptar las estrategias en función del estado de los agentes, reduciendo el desgaste y prolongando la vida útil de los sistemas [17].

Aprendizaje por refuerzo en la planificación de tareas dinámicas

El artículo revisado, "Reinforcement Learning in Dynamic Task Scheduling: A Review" [11], escrito por Shyalika, Silva y Karunananda en 2020, aborda el problema de la planificación óptima de tareas y recursos de forma dinámica utilizando técnicas de aprendizaje por refuerzo. El objetivo del artículo es proporcionar un estado del arte sobre las técnicas de aprendizaje por refuerzo utilizadas en la planificación dinámica de tareas y presentar una revisión comparativa de estas técnicas. La metodología del estudio se basa en la revisión exhaustiva de la literatura existente en el campo.

Los resultados obtenidos en el artículo demuestran que las técnicas de aprendizaje por refuerzo han sido efectivas en la planificación dinámica de tareas. Se identifican y analizan diferentes enfoques y algoritmos utilizados en estudios previos, lo que proporciona una visión amplia del estado actual del campo.

En cuanto a la conclusión del artículo, se destaca la importancia del aprendizaje por refuerzo como una herramienta prometedora para la planificación de tareas dinámicas, pero también se señalan las limitaciones y desafíos que aún existen en este ámbito de investigación.

Este artículo fue útil para comprender las técnicas de aprendizaje por refuerzo aplicadas a la planificación dinámica de tareas. Sin embargo, debido a que el problema de investigación se enfoca en otro aspecto específico de la planificación, este artículo no aborda directamente en la investigación de la presente tesis y no proporciona una solución directa para resolverlo. A pesar de eso, el artículo ha brindado un marco teórico sólido y ha permitido obtener una perspectiva general del uso del aprendizaje por refuerzo en la planificación dinámica de tareas.

Asignación dinámica de tareas de múltiples agentes en la aplicación de la ley

El artículo "Towards addressing dynamic multi-agent task allocation in law enforcement" [7], escrito por Tkach, A. Itshak y Sofia en 2021, aborda el problema de la asignación dinámica de tareas a agentes policiales en la aplicación de la ley. El objetivo de la investigación es resolver este problema mediante la comparación de tres métodos: Fisher basado en el mercado FMC_TAH+, inteligencia de enjambre HDBA y algoritmos SA de recocido simulado.

En el estudio, se utiliza el método FMC_TAH+ que implementa agentes como compradores y tareas como bienes, con el fin de calcular asignaciones justas y eficientes en un tiempo polinomial o pseudopolinomial. El algoritmo HDBA, basado en la inteligencia de enjambre e inspirado en el comportamiento de las abejas sociales, asigna agentes a tareas según el desempeño de los agentes, las prioridades de las tareas y las distancias entre ellos. Por otro lado, SA es un algoritmo de recocido simulado utilizado como metaheurística para la optimización en problemas grandes.

Los tres métodos fueron comparados en el estudio utilizando cinco medidas de desempeño comúnmente utilizadas por las autoridades encargadas de hacer cumplir la ley. Los resultados indican que el método FMC_TAH+ tiene ventajas en términos de utilidad total y tiempo medio de llegada a las tareas. Además, en comparación con HDBA y SA, FMC_TAH+ genera un 34% y 32% más de utilidad del equipo en la carga de trabajo de mayor demanda.

Este artículo resultó útil para entender los métodos utilizados en la asignación dinámica de tareas en la aplicación de la ley. Sin embargo, no aborda directamente el problema de investigación de forma específica, ya que se centra en la aplicación de la ley y la asignación de tareas a agentes policiales. Aunque el estudio ofrece información relevante sobre métodos de asignación, no es directamente aplicable a esta investigación sobre otro ámbito o contexto.

Búsqueda de rutas de múltiples agentes en entornos configurables

El artículo "Multi-Agent Path Finding in Configurable Environments" [12], escrito por B. Matteo, B. Nicola y A. Francesco en 2020, aborda el problema de la búsqueda de rutas de múltiples agentes (Multi-Agent Path Finding, MAPF) en entornos configurables. En muchas aplicaciones de la vida real, es crucial la coordinación de agentes autónomos para alcanzar objetivos comunes.

En el artículo, se presenta C-MAPF, una variante del problema MAPF en la que el entorno es configurable, lo que significa que su estructura y topología se pueden controlar dentro de ciertas reglas predefinidas. Se utiliza como ejemplo una aplicación de logística de almacén, donde los gerentes pueden modificar el entorno mediante la reorganización de estantes o la adición/remoción de paredes temporales.

Se estudian las propiedades del problema C-MAPF y se diseñan dos algoritmos basados: uno en Conflict-Based Search (CBS) y otro un algoritmo avanzado para MAPF. En primer lugar, se presenta Parallel CBS (P-CBS), que busca una solución considerando simultáneamente todas las configuraciones posibles del entorno. Luego se introduce Abstract CBS (A-CBS), una versión extendida del algoritmo CBS que resuelve problemas de C-MAPF al permitir un nuevo tipo de conflicto en las configuraciones del entorno, entonces se demuestra que ambos algoritmos son completos y óptimos.

Se realiza una evaluación experimental del rendimiento de los solucionadores en diferentes entornos, generando soluciones simultáneas considerando todas las posibles configuraciones del entorno.

Este artículo fue de utilidad para comprender cómo abordar el problema de búsqueda de rutas en entornos configurables con múltiples agentes. Sin embargo, no es directamente aplicable al problema de investigación, ya que el enfoque se centra en otro contexto o dominio. Aun así, el artículo proporcionó una base teórica y metodológica valiosa para la comprensión de los desafíos y enfoques en la búsqueda de rutas de múltiples agentes en entornos configurables.

Un modelo dinámico de asignación de tareas para el rescate de emergencias en aviación basado en el aprendizaje por refuerzo de múltiples agentes

Este artículo titulado "A dynamic task assignment model for aviation emergency rescue based on multi-agent reinforcement learning" [13], aborda el desafío de emparejar las demandas de rescate en constante cambio con la disponibilidad limitada de aeronaves de rescate de emergencia de aviación.

El objetivo principal de este artículo es mejorar la eficiencia de los rescates de emergencia de aviación mediante el desarrollo de un modelo de asignación de tareas dinámicas basado en aprendizaje por refuerzo multiagente.

La metodología utilizada se basa en el enfoque del aprendizaje por refuerzo multiagente. Se construye un modelo de asignación de tareas dinámicas que representa los elementos relevantes del rescate de emergencia de aviación y su estructura organizativa. Además, se establece un marco de entrenamiento basado en el algoritmo PPO (Proximal Policy Optimization) para entrenar el modelo.

Los resultados obtenidos en este estudio demuestran que el modelo propuesto logra mayores beneficios de asignación al considerar la naturaleza dinámica de las demandas de rescate y la disponibilidad limitada de equipos de rescate en helicóptero. También se observa un aumento en las tasas de asignación de tareas y una mayor satisfacción promedio en cuanto al tiempo de respuesta, lo que indica una asignación más eficiente y oportuna de las tareas.

En conclusión, este artículo presenta un enfoque prometedor para mejorar la eficiencia de los rescates de emergencia de aviación a través de un modelo de asignación de tareas dinámicas basado en aprendizaje por refuerzo multiagente. Los resultados obtenidos son alentadores y muestran mejoras significativas en la asignación de tareas y en la satisfacción del tiempo de respuesta. Sin embargo, se señalan algunas limitaciones y desafíos futuros que podrían ser abordados en investigaciones posteriores, como una mayor consideración de la calidad del rescate y la inclusión de diferentes niveles de habilidad del personal de rescate en el modelo.

Este artículo es valioso, ya que aborda un problema relevante y propone una metodología innovadora para resolverlo. Los resultados obtenidos respaldan la efectividad del modelo propuesto y sugieren su potencial aplicación en el ámbito de los rescates de emergencia de aviación. Sin embargo, es importante tener en cuenta las limitaciones y desafíos mencionados para seguir mejorando y perfilando este enfoque en futuras investigaciones.

Un enfoque profundo de aprendizaje por refuerzo de múltiples agentes para resolver el problema dinámico de programación de talleres

El artículo titulado "A deep multi-agent reinforcement learning approach to solve dynamic job shop scheduling problem" [14], aborda el desafío de la planificación dinámica de la producción en entornos de talleres de trabajo, donde los problemas de planificación son volátiles y complejos. El objetivo principal del artículo es desarrollar un enfoque basado en aprendizaje por refuerzo multiagente profundo para abordar este problema y mejorar la eficiencia de la planificación en tiempo real.

La metodología utilizada se basa en el aprendizaje por refuerzo multiagente profundo. Se propone un algoritmo integrado que permite que cada máquina o agente determine su orden de operaciones en el taller de trabajo. Además, se desarrollan representaciones de estado y acción especializadas que pueden adaptarse a problemas de planificación de diferentes tamaños. Se utiliza un marco de acción cronológica para abordar la dificultad de asignación de crédito en la planificación dinámica. También se emplean técnicas de modelado de recompensas basadas en conocimiento para fomentar la cooperación entre los agentes y mejorar el rendimiento general del sistema.

Los resultados obtenidos en los estudios de simulación muestran que el enfoque propuesto proporciona un rendimiento competitivo y consistente en comparación con enfoques contemporáneos basados en aprendizaje por refuerzo profundo y estrategias de planificación en tiempo real existentes. Se demuestra la eficacia del enfoque propuesto para abordar la planificación dinámica en entornos de talleres de trabajo.

En conclusión, este artículo presenta un enfoque innovador basado en aprendizaje por refuerzo multiagente profundo para resolver el problema de la planificación dinámica de talleres de trabajo. Los resultados obtenidos respaldan la efectividad del enfoque propuesto y muestran mejoras significativas en la eficiencia de la planificación en tiempo real. Este enfoque tiene el potencial de ser aplicado en diversos escenarios de planificación en la industria manufacturera.

Este artículo es muy interesante y ofrece una solución innovadora para abordar el desafío de la planificación dinámica en entornos de talleres de trabajo. El enfoque basado en aprendizaje por refuerzo multiagente profundo parece ser prometedor, ya que permite que cada máquina o agente tome decisiones de planificación en tiempo real de manera autónoma.

Diseño de mecanismos de pago para la coordinación en juegos de asignación de tareas de múltiples agentes

El artículo titulado "Payoff Mechanism Design for Coordination in Multi-Agent Task Allocation Games" [15], aborda el problema de la toma de decisiones multiagente en la asignación de tareas, donde una población de agentes debe coordinar sus estrategias para minimizar los trabajos restantes en todas las tareas asignadas. El objetivo principal del trabajo es diseñar un modelo descentralizado que permita a los agentes adoptar estrategias óptimas, coordinadas y que minimicen los trabajos restantes.

En cuanto a la metodología utilizada, se emplea el enfoque de juegos de población para formular el problema como un juego de asignación de tareas. Se propone un mecanismo de recompensa basado en pagos para incentivar la coordinación entre los agentes en la selección de estrategias. Se utiliza la teoría de control de retroalimentación para derivar condiciones técnicas que el modelo debe satisfacer, y se presenta un enfoque numérico para calcular el modelo.

Los resultados obtenidos demuestran la efectividad del mecanismo de recompensa propuesto para coordinar a los agentes en juegos de asignación de tareas. Se muestran ejemplos numéricos que ilustran cómo el diseño óptimo del mecanismo de recompensa puede influir en la adopción de estrategias óptimas por parte de los agentes y en la minimización de los trabajos restantes.

En conclusión, este artículo presenta un enfoque innovador para el diseño de un mecanismo de recompensa que coordina la asignación de tareas en entornos multiagente. Los resultados demuestran la eficacia del enfoque propuesto y su capacidad para mejorar la coordinación entre los agentes. Además, se mencionan posibles extensiones del trabajo, como el uso de mapeos no lineales y el aprendizaje automático para mejorar aún más el mecanismo de recompensa y la asignación de tareas.

Este artículo ofrece una contribución significativa al campo de la toma de decisiones multiagente y la asignación de tareas. El enfoque propuesto es sólido y los resultados obtenidos respaldan la efectividad del mecanismo de recompensa diseñado. Además, las posibles extensiones mencionadas abren el camino para futuras investigaciones y mejoras en el campo de la coordinación en juegos de asignación de tareas.

Un método de entrenamiento de períodos para la asignación dinámica de tareas heterogéneas de UUV

El artículo "A Period Training Method for Heterogeneous UUV Dynamic Task Allocation. Electronics" [16], presenta un enfoque interesante y prometedor para abordar el desafío de la asignación dinámica de tareas en UUVs heterogéneos. La combinación del aprendizaje por refuerzo multiagente (MARL) y el método de entrenamiento periódico (PTM) proporciona una solución eficiente y adaptable para asignar tareas a los UUVs en tiempo real.

El uso del MARL permite una asignación rápida de tareas a cada UUV, teniendo en cuenta las características heterogéneas de los mismos. Además, el método de entrenamiento periódico mejora la robustez del algoritmo al optimizar los parámetros del modelo en diferentes entornos de entrenamiento.

Los resultados obtenidos en las pruebas de simulación respaldan la efectividad del método propuesto, mostrando una asignación rápida y precisa de tareas a los UUVs heterogéneos. Esto es especialmente importante en situaciones de emergencia, donde la capacidad de respuesta rápida y eficiente es crucial.

Este artículo ofrece una contribución significativa al campo de la asignación de tareas en UUVs. El enfoque propuesto puede mejorar la eficiencia operativa de los UUVs y su capacidad para adaptarse a cambios dinámicos en el entorno de las tareas. Además, abre la puerta a futuras investigaciones y mejoras en la asignación dinámica de tareas en otros contextos y escenarios.

Adaptación consciente del desgaste para el patrullaje de múltiples agentes

El artículo "Attrition-Aware Adaptation for Multi-Agent Patrolling" [17], aborda el problema del patrullaje multiagente, que es relevante para la Policía Nacional del Perú de la ciudad de Arequipa. El problema implica realizar patrullas eficientes y efectivas, asignando tareas a los agentes de manera dinámica y considerando la posibilidad de pérdida de agentes debido a diversas circunstancias. El objetivo del artículo es presentar un enfoque adaptativo para el patrullaje multiagente que tome en cuenta la pérdida de agentes y permita la adaptación eficiente de las asignaciones de tareas.

En cuanto a la metodología y técnicas utilizadas, el artículo propone el algoritmo Adaptive Heuristic-based Patrolling Algorithm (AHPA). Este algoritmo se basa en la formulación matemática del problema de patrullaje multiagente, tanto en su versión centralizada como

distribuida. Además, se emplea la partición de Voronoi como una herramienta para asignar nodos a los agentes y se implementa una heurística basada en el problema del viajante para determinar el orden de visita de cada agente. El enfoque adaptativo del algoritmo AHPA permite una adaptación efectiva y escalable ante la pérdida de agentes, minimizando la necesidad de comunicación entre ellos.

En términos de resultados, el artículo demuestra que el enfoque adaptativo propuesto, AHPA, es capaz de adaptarse eficientemente ante la pérdida de agentes en el patrullaje multiagente. Se realizan comparaciones con varios algoritmos de referencia utilizando un entorno de simulación realista basado en el Robot Operating System (ROS), y se muestra que AHPA ofrece una solución efectiva y escalable para la adaptación de tareas en presencia de pérdida de agentes.

En conclusión, el enfoque adaptativo presentado en el artículo, basado en el algoritmo Adaptive Heuristic-based Patrolling Algorithm (AHPA), ofrece una solución innovadora y relevante para abordar el problema de asignación dinámica de tareas en el patrullaje multiagente. El artículo destaca la capacidad de adaptación del algoritmo, su eficiencia en términos de comunicación y su valor en entornos de patrullaje realistas. En el contexto de la tesis sobre el modelo de asignación dinámica de tareas utilizando el sistema multiagente en la Policía Nacional del Perú de la ciudad de Arequipa, este antecedente proporciona un enfoque prometedor que se puede considerar para mejorar la eficiencia y efectividad de la asignación de tareas en la policía.

2.4. Fundamentos teóricos

Aquí se exploran los conceptos y teorías clave que sustentan el estudio. Se discuten los principios fundamentales y los modelos teóricos que guían la investigación, asegurando una base sólida para el análisis y la interpretación de los datos.

2.4.1. Inteligencia artificial

La inteligencia es el factor decisivo de cómo los seres humanos se convierten en las formas de vida más dominantes en la tierra. A lo largo de la historia, los seres humanos han desarrollado herramientas y tecnologías que ayudan a las civilizaciones a evolucionar y crecer. Computadoras y por extensión, IA, ha jugado un papel importante en ese continuo de tecnologías [18].

A. Aprendizaje por Refuerzo Profundo

Aunque ML se ha convertido en sinónimo de IA; Recientemente, Deep Learning (DL) se está utilizando en lugar del aprendizaje automático de forma persistente. Si bien el aprendizaje automático está ocupado con métodos supervisados y no supervisados, el aprendizaje profundo continúa su motivación para replicar el sistema nervioso humano mediante la incorporación de tipos avanzados de Neural Network (NN). Debido a su viabilidad, el aprendizaje profundo ha encontrado en sus aplicaciones en varias soluciones de inteligencia artificial, como la visión por computadora, el procesamiento del lenguaje natural, el análisis de video inteligente, el análisis de imágenes hiperespectrales de satélites, etc [19].

2.4.2. Agentes

El concepto de agente se ha vuelto importante tanto en AI como en la informática convencional. Un agente en ciencias de la computación es una entidad de tipo computacional con inteligencia. Por tanto, un agente es entidad capaz de obtener información sobre su medio ambiente o percibir su entorno por sensores y luego intente seleccionar la acción apropiada dentro de varias acciones que están disponibles e intentan lograr las metas esperadas actuando a través de actuadores [19].

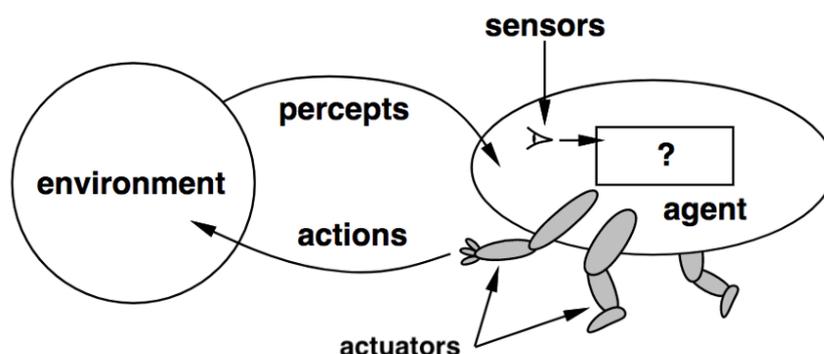


Figura 2.5: Composición de un agente inteligente en el sistema.

Fuente: Attrition-Aware Adaptation for Multi-Agent Patrolling [17].

En la Figura 2.5 ilustra cómo cada agente está diseñado con módulos específicos para la percepción del entorno, toma de decisiones, y ejecución de acciones. Los agentes perciben información del entorno a través de sensores, procesan esta información para tomar decisiones y finalmente ejecutan las acciones decididas. Esta composición permite a los agentes adaptarse dinámicamente a cambios en el entorno [17].

A. Características Internas

Un agente inteligente tiene alguna característica interna incluyendo autonomía, aprendizaje y razonamiento y basado en objetivos. A continuación, se analiza sobre cada característica debajo [20].

a) Autonomía

Los agentes inteligentes tienen la capacidad de percibir su entorno y actuar de manera autónoma, basándose en su conocimiento y las reglas establecidas. Se destaca el de los agentes Belief Desire Intention (BDI). Estos agentes reconocen la importancia de las intenciones y permiten categorizar las interacciones entre creencias, deseos e intenciones. Además, los agentes BDI pueden aprender de sus experiencias y utilizar ese aprendizaje para adaptar su comportamiento. Esta capacidad de aprendizaje les permite mejorar su toma de decisiones y su interacción con el entorno [21].

b) Reactividad

La reactividad es un principio fundamental en el diseño de agentes inteligentes. Se refiere a la capacidad de un agente para reaccionar en tiempo real a los estímulos del entorno y tomar decisiones inmediatas en función de esa información. La reactividad implica una respuesta rápida y eficiente a los cambios y eventos que ocurren en el entorno. Los agentes reactivos se centran en la toma de decisiones basada en la información actual del entorno, sin considerar el historial de eventos o el estado pasado. Estos agentes actúan de manera reflexiva, respondiendo directamente a los estímulos percibidos sin la necesidad de una planificación o consideración a largo plazo. La reactividad es especialmente útil en entornos dinámicos y cambiantes, donde las respuestas rápidas son necesarias para lograr los objetivos [22].

c) Aprendizaje/Razonamiento

En la actualidad, el aprendizaje y razonamiento de agentes ha experimentado avances significativos basados en los principios del aprendizaje profundo y el procesamiento de datos a gran escala. El enfoque del aprendizaje profundo ha revolucionado campos como el reconocimiento de voz, la detección de objetos visuales y el procesamiento de texto, entre otros. La capacidad de los modelos de aprendizaje profundo para aprender representaciones de datos con múltiples niveles de abstracción ha llevado a mejoras impresionantes en la capacidad de los agentes para comprender y procesar información compleja. Además, se ha observado un

creciente interés en la aplicación del aprendizaje no supervisado, reconociendo que gran parte del aprendizaje humano y animal se realiza mediante la observación y la inferencia de la estructura del mundo. La combinación de redes neuronales convolucionales y recurrentes ha permitido avances notables en la visión por computadora y en el procesamiento de datos secuenciales, aprovechando el aprendizaje por refuerzo para guiar la atención y la toma de decisiones. Sin embargo, a pesar de estos logros, aún se requieren investigaciones adicionales para integrar de manera efectiva el aprendizaje de representaciones con el razonamiento complejo, lo que permitirá a los agentes realizar tareas más sofisticadas que involucren la manipulación y comprensión de símbolos y conceptos abstractos [23].

d) Basado en Objetivos

Ha ganado un creciente interés dentro de la comunidad de agentes, particularmente en relación con el uso de técnicas de la teoría de decisiones y la teoría de juegos. El artículo "Game Theory and Decision Theory in Multi-Agent Systems" proporciona una visión general de estos conceptos clave y discute cómo se aplican estas herramientas en la investigación de sistemas de agentes. La utilización de la teoría de decisiones y la teoría de juegos ha permitido a los agentes tomar decisiones racionales y estratégicas en entornos complejos y dinámicos. La teoría de decisiones, basada en modelos como las redes bayesianas y los diagramas de influencia, ha proporcionado un marco formal para la toma de decisiones en presencia de incertidumbre. Por otro lado, la teoría de juegos ha permitido a los agentes razonar sobre las interacciones estratégicas entre ellos, lo que ha impulsado el desarrollo de enfoques de negociación y cooperación entre agentes. Estas técnicas continúan siendo objeto de investigación activa en sistemas de agentes, y su aplicación efectiva ha demostrado mejorar la capacidad de los agentes para alcanzar sus objetivos en entornos complejos y competitivos [24].

B. Características Externas

Además, cada agente inteligente tiene algunas características tales como: comunicación, cooperación, movilidad que se discutirán más a continuación [20].

a) Comunicación

Cada agente requiere interactuar con su entorno, incluyendo otros agentes y seres humanos, para alcanzar sus objetivos. Actualmente se propone un esquema de comunicación denominado Intention Sharing (IS) en el ámbito del aprendizaje por refuerzo multiagente. Esta propuesta introduce el esquema IS como una manera de mejorar la coordinación entre los agentes. En este

esquema, cada agente genera una trayectoria imaginada que modela la dinámica del entorno y las acciones de los demás agentes, lo cual representa su plan de acción futuro. Mediante un mecanismo de atención, se determina la importancia relativa de las diferentes componentes de la trayectoria imaginada, basándose en los mensajes recibidos de otros agentes. Los resultados obtenidos en este trabajo demuestran que el esquema IS propuesto supera de manera significativa a otros esquemas de comunicación en el aprendizaje por refuerzo multiagente [25].

b) Cooperativa

Para realizar tareas complejas, un agente necesita cooperar con otros agentes para aumentar su propia capacidad y lograr sus objetivos de manera más efectiva. La cooperatividad es una característica externa crucial para los agentes, ya que les permite trabajar en conjunto y realizar tareas con mayor facilidad. Un artículo relevante para este tema es "A Review of Cooperative Multi-Agent Deep Reinforcement Learning". Este artículo se enfoca en presentar enfoques recientes sobre algoritmos de Aprendizaje por Reforzamiento Multiagente (MARL) que abordan problemas de aprendizaje cooperativo. Se exploran cinco enfoques comunes, como el uso de aprendices independientes, críticos completamente observables, factorización de la función de valor, consenso y aprendizaje de la comunicación. Estos enfoques ofrecen formas diferentes de modelar y resolver desafíos en entornos cooperativos. Además, el artículo revisa aplicaciones del MARL en el mundo real y proporciona una lista de entornos disponibles para la investigación en este campo. Al considerar la cooperatividad como una característica externa de un agente, el análisis de este artículo ofrece una perspectiva valiosa para comprender cómo los agentes pueden trabajar juntos y aumentar su capacidad para lograr sus objetivos de manera más efectiva [26].

c) Movilidad

La movilidad es una característica clave de los agentes inteligentes, ya que les permite navegar dentro de las redes de comunicación. En el artículo 'Distributed Machine Learning with Self-Organizing Mobile Agents for Earthquake Monitoring', se propone el uso de agentes móviles para fusionar el Internet de las Cosas (IoT) con entornos móviles y en la nube. Estos agentes móviles pueden migrar entre diferentes plataformas, como los navegadores web, llevando consigo su código, estado y datos. Esta capacidad de movilidad facilita la adaptación y el desplazamiento autónomo de los agentes en entornos heterogéneos, lo cual es fundamental para la creación de sistemas autoorganizados y aplicaciones de alto impacto, como el monitoreo de terremotos [27].

2.4.3. Tipos de entornos para dispositivos inteligentes agentes

Existen diferentes tipos de entornos, cada uno tiene características específicas en función a su estructura. A continuación, se muestran los siguientes entornos:

a) Totalmente observable vs a Parcialmente observable

Si los sensores de un agente pueden proporcionar información completa sobre su entorno desde diferentes puntos o dimensiones, significa que el entorno es completamente observable. Sin embargo, a veces, por diferentes razones (como ruido o problemas en sensores, etc.), no es posible que los sensores se llenen de información o ayuda para percibir el entorno total, en este caso, el entorno parcialmente observable será asignado a ese agente. En este contexto, el artículo 'Partially Observable Markov Decision Processes for Artificial Intelligence' aborda el problema de elegir acciones óptimas en dominios estocásticos parcialmente observables mediante técnicas de investigación de operaciones. El artículo introduce la teoría de los procesos de decisión de Markov (MDPs) y los procesos de decisión de Markov parcialmente observables (POMDPs). También presenta un algoritmo novedoso para resolver los POMDPs offline, mostrando cómo se puede extraer un controlador de memoria finita a partir de la solución de un POMDP. Asimismo, se destaca el uso del paquete 'pomdp' en el lenguaje R para abordar el Problema del Tigre como ejemplo ilustrativo, y se exploran las extensiones y aplicaciones de los POMDPs en diversos contextos [28].

b) Determinista vs Estocástico

El tipo de entorno en el que opera un agente inteligente puede clasificarse en determinista o estocástico. En un entorno determinista, el agente tiene la capacidad de tomar decisiones precisas basadas en la información completa del estado actual, lo que le permite seleccionar la acción adecuada para lograr el próximo estado deseado con certeza. En este tipo de entorno, el agente puede planificar de manera óptima y predecir el resultado de sus acciones con precisión. Por otro lado, en los entornos estocásticos, el estado actual del agente no proporciona suficiente información para determinar completamente el siguiente estado o la acción precisa necesaria para alcanzarlo. En lugar de eso, el siguiente estado está influenciado tanto por el estado actual como por una distribución de probabilidad. Esto introduce incertidumbre y azar en el entorno, lo que dificulta la predicción precisa y la planificación óptima por parte del agente en entornos estocásticos [29].

c) Episódico vs. Secuencial

La diferencia entre entornos episódicos y secuenciales radica en cómo se estructuran las interacciones entre un agente y su entorno. En un entorno episódico, las interacciones se dividen en unidades autónomas llamadas episodios, donde el agente puede actuar y lograr objetivos específicos. No hay una dependencia explícita de eventos pasados o futuros entre los episodios. En contraste, en un entorno secuencial, las interacciones ocurren en una secuencia continua, y las acciones pasadas influyen en las decisiones actuales del agente. Esta distinción tiene implicaciones importantes para el diseño de estrategias de aprendizaje y toma de decisiones de los agentes inteligentes, ya que en entornos episódicos se maximiza la recompensa dentro de cada episodio, mientras que en entornos secuenciales se considera el impacto a largo plazo de las acciones. Adaptar las estrategias al tipo de entorno es crucial para el éxito del agente [30].

d) Estático vs Dinámico

El entorno estático no cambia durante el paso del tiempo, lo que significa que el agente no necesita monitorear continuamente el entorno para realizar cambios en sí mismo. Sin embargo, en un entorno dinámico, el tiempo, las acciones del agente y los diferentes estados del mundo pueden afectar el entorno. En este contexto, un paper relevante destaca la importancia de considerar la toma de decisiones en etapas sucesivas y la ponderación de los costos presentes y futuros. En este enfoque, se clasifican las decisiones en función de una combinación del costo presente y el costo futuro esperado, asumiendo una toma de decisiones óptima en etapas posteriores. Esto subraya la necesidad de que un agente en un entorno dinámico considere la evolución del sistema a lo largo del tiempo y anticipe los costos futuros al tomar decisiones [31].

e) Discreto vs. Continuo

En el caso de un entorno discreto, hay un número limitado de percepciones, estados o acciones disponibles para un agente. Esto significa que el agente solo puede interactuar con un conjunto específico y finito de opciones en su entorno. Por ejemplo, en un juego de ajedrez, las posiciones del tablero y las posibles jugadas son discretas y predefinidas.

Por otro lado, en un entorno continuo no hay este tipo de limitaciones para las percepciones, acciones o estados actuales y siguientes. En este tipo de entorno, las posibilidades son infinitas y pueden variar en un continuo. Esto implica que los agentes deben lidiar con una gran cantidad de información y tomar decisiones en un espacio de acción más amplio [32].

f) Agente único vs. Multi-agente

En RL se pueden distinguir dos categorías principales de entornos para dispositivos inteligentes agentes: Single Agent (agente único) y Multi-Agent (multiagente). En un entorno Single Agent, un agente opera de manera autónoma sin depender de otros agentes para llevar a cabo sus tareas y alcanzar sus objetivos. Este enfoque es adecuado cuando el agente tiene la capacidad de realizar todas las acciones necesarias de forma independiente. Por otro lado, en un entorno Multi-Agent, los agentes requieren colaborar y cooperar entre sí para lograr un comportamiento deseado. En este tipo de entorno, varios agentes interactúan y se comunican, intercambiando información y tomando decisiones conjuntas para alcanzar metas comunes. La colaboración entre agentes es fundamental en los entornos Multi-Agent, ya que cada agente puede poseer conocimientos y habilidades específicas que contribuyen al éxito colectivo [33].

2.4.4. Asignación de Tareas Dinámicas

La asignación de tareas dinámica es un componente crucial en sistemas multiagente. En esta sección, se analizarán las teorías y enfoques que permiten la distribución eficiente y adaptativa de tareas entre agentes, destacando las ventajas y desafíos asociados.

2.4.4.1 Definición y contexto

La asignación dinámica de tareas es un problema ampliamente reconocido y aplicado en diversos campos, como el diseño de sistemas informáticos distribuidos, la asignación de trabajos en redes de telecomunicaciones y la resolución de problemas de ubicación, rutas de camiones y programación en talleres de trabajo. Este problema se basa en el modelo clásico de investigación de operaciones, que busca asignar de manera óptima un conjunto de tareas a un grupo de agentes, asegurando que cada agente tenga una tarea asignada y que cada tarea sea asignada a un solo agente [34].

En el contexto de la asignación de tareas, surgen problemáticas relacionadas con las limitaciones de presupuesto, que definen los diferentes consumos de recursos, y las relaciones entre los agentes y las tareas. Estas problemáticas pueden incluir permitir que un agente realice múltiples tareas y/o que una tarea sea realizada por varios agentes [35].

La asignación de tareas dinámicas se refiere a la asignación óptima de un conjunto de tareas a un grupo de agentes en un entorno en constante cambio. Este problema ha sido ampliamente estudiado en campos como los sistemas informáticos distribuidos, las redes de

telecomunicaciones y la programación en talleres de trabajo. Para abordar estas situaciones, es necesario considerar las limitaciones de presupuesto, los diferentes consumos de recursos y las relaciones entre los agentes y las tareas. Al asignar de manera adecuada las tareas a los agentes, se logra un mejor uso de los recursos disponibles y se maximiza la eficiencia del sistema en su conjunto. Además, en entornos cambiantes, la capacidad de adaptarse y reasignar tareas de manera dinámica es esencial para mantener un funcionamiento óptimo del sistema [36].

2.4.4.2 Importancia

La importancia de este modelo de asignación dinámica de tareas utilizando el sistema multiagente en la Policía Nacional del Perú de la ciudad de Arequipa radica en la mejora de la eficiencia operativa al optimizar la asignación de tareas y utilizar de manera óptima los recursos disponibles. Esto permite una respuesta más rápida y efectiva ante situaciones de emergencia, así como una mayor capacidad para adaptarse a los cambios en el entorno operativo. Además, fomenta la coordinación y colaboración entre los agentes, mejorando la comunicación y la distribución equitativa de responsabilidades. Asimismo, contribuye a la optimización del uso de los recursos humanos, materiales y tecnológicos disponibles, asegurando una asignación adecuada de personal y una utilización eficiente de vehículos, equipos y otros recursos necesarios para la ejecución de las tareas [37].

2.4.4.3 Problemáticas

La asignación dinámica de tareas en la Policía Nacional del Perú de la ciudad de Arequipa enfrenta diversas problemáticas. La complejidad de la coordinación en un entorno operativo dinámico requiere una comunicación clara, protocolos colaborativos y procedimientos de coordinación efectivos. La gestión de dependencias entre tareas es crucial para evitar retrasos, considerando la secuencia de ejecución y los recursos compartidos. Además, la adaptación a situaciones imprevistas exige una reasignación ágil de tareas ante emergencias. La consideración de la carga de trabajo y la capacidad de los agentes garantiza una asignación equitativa y eficiente, evitando la sobrecarga o subutilización de recursos. Asimismo, la integridad y confidencialidad de la información son fundamentales, requiriendo medidas de seguridad para proteger los datos y prevenir accesos no autorizados. Estas problemáticas deben abordarse adecuadamente para lograr una asignación dinámica de tareas efectiva en la Policía Nacional del Perú de la ciudad de Arequipa [36].

2.4.4.4 Futuro

En el futuro, se espera que la asignación de tareas dinámicas siga siendo un área de investigación activa y relevante. Se prevé una mejora continua de los algoritmos y métodos utilizados, con enfoques basados en inteligencia artificial, aprendizaje automático y optimización para lograr asignaciones más eficientes y adaptativas. Además, se investigarán y desarrollarán métodos para garantizar la seguridad y la confiabilidad de la asignación de tareas, especialmente en entornos críticos. La integración con tecnologías emergentes, como IoT y la computación en la nube, permitirá una asignación más eficiente y escalable, aprovechando la recopilación de datos y el procesamiento distribuido. Asimismo, se deben considerar los aspectos éticos y sociales, incluyendo la equidad en la asignación de tareas, la transparencia de los algoritmos y el impacto en la fuerza laboral y la sociedad en general. Estas direcciones futuras en la asignación de tareas dinámicas prometen continuar impulsando avances significativos en este campo [38].

2.4.5. Herramientas para el desarrollo de Agentes Inteligentes

Las tecnologías específicas empleadas en este estudio para la simulación, enfocándonos en su aplicación para agentes inteligentes.

a) Mesa

Mesa es un framework especializado en modelado basado en agentes (ABM) diseñado para Python. Proporciona una estructura flexible para la creación de simulaciones donde múltiples agentes autónomos interactúan con su entorno y entre ellos. En este proyecto, Mesa se utiliza para definir los agentes vehiculares y su comportamiento, así como para gestionar la ejecución de la simulación [39].

b) Shapely

Shapely es una biblioteca de Python que ofrece funcionalidades para la manipulación y análisis de objetos geométricos. Esta herramienta resulta fundamental en la verificación de relaciones espaciales. En el contexto de los agentes inteligentes para el transporte urbano, Shapely facilita la implementación de lógica espacial para la toma de decisiones de los agentes [40].

c) IPython Display

IPython Display es un módulo que permite la visualización interactiva de objetos en entornos IPython, como Jupyter Notebooks. En este proyecto, IPython Display se emplea para mostrar mapas interactivos generados durante la simulación, facilitando la comprensión y análisis de los resultados obtenidos [41].

Capítulo III - Solución propuesta

Para esta problemática se va a desarrollar y aplicar el LEPH mediante el MAS para mejorar la asignación dinámica de tareas en la Policía Nacional del Perú de la ciudad de Arequipa.

La Figura 3.1 muestra el modelo de tesis propuesto, consiste en recopilar la información necesaria y pertinente que permita diseñar el modelo LEPH con una buena arquitectura y diagramas, luego codificar el modelo utilizando mesa, a continuación, entrenar el modelo y finalmente aplicar métricas de evaluación, con el fin de optimizar la asignación dinámica de tareas en la Institución policial de Arequipa y de nuestro país.

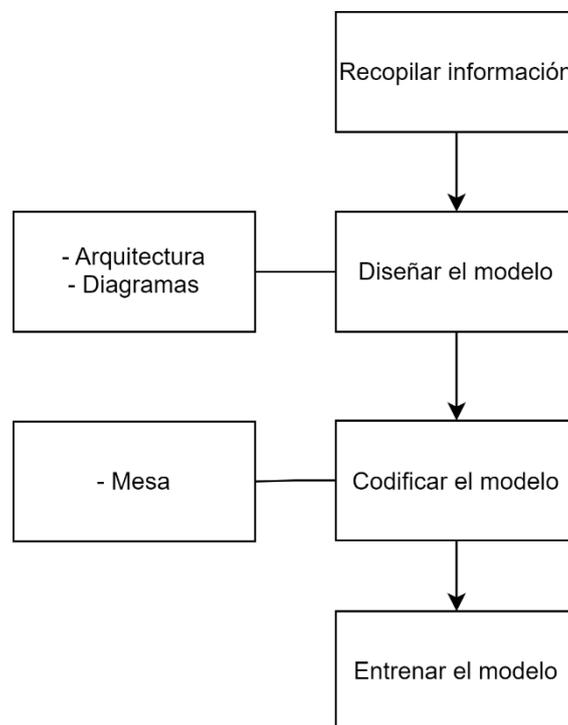


Figura 3.1: Esquema del modelo propuesto en la tesis.

Fuente: Elaboración propia.

3.1. Recopilar información

En esta investigación, se enfoca en la obtención de información esencial para el desarrollo de la aplicación propuesta. La recopilación de datos se llevará a cabo mediante dos enfoques complementarios, garantizando una comprensión completa de la realidad de las comisarías en la ciudad de Arequipa. La información detallada sobre los datos recopilados se encuentra en el Anexo 1.

3.1.1 Entrevistas con la Policía Nacional del Perú

La Policía Nacional del Perú se identifica como una fuente directa y confiable para obtener información clave. Se elaboró un conjunto estructurado de preguntas que abarcan diversos aspectos, con el objetivo de obtener una visión completa de las necesidades y desafíos enfrentados por las comisarías en la ciudad de Arequipa. Estas entrevistas proporcionarán una comprensión detallada de los procedimientos actuales, protocolos de seguridad y cualquier otra información relevante para el alcance de la investigación.

3.1.2 Análisis de Documentos Estatales

Además de las entrevistas con la Policía Nacional, se ha llevado a cabo un análisis exhaustivo de documentos estatales relacionados con la seguridad ciudadana. Entre estos documentos se incluye el "Plan de Acción Distrital de Seguridad Ciudadana de Yanahuara". Este análisis proporcionará una perspectiva complementaria, permitiendo una contextualización más amplia de las estrategias gubernamentales en relación con la seguridad en la región.

Este enfoque de recopilación de información se ha diseñado para garantizar la obtención de datos exhaustivos y multidimensionales, fundamentales para la implementación efectiva de la solución propuesta.

3.2. Diseño de Modelo

El diseño del modelo es una etapa crítica en la investigación. En esta sección, se describirá la conceptualización y estructuración del modelo propuesto, incluyendo los componentes y las interacciones entre ellos.

3.2.1. Diseño general de la aplicación

Para mejor comprensión sobre la funcionalidad del comportamiento del personal policial cuando se le asigna una tarea, se muestra un diagrama de actividad, que muestra el desplazamiento de la policía para asistir a la denuncia presentada.

A. Diagrama de Actividad para Resolver Denuncias Asignadas

En toda comisaria hay una central de recepción de denuncias donde recurren los ciudadanos ya sea en forma presencial o por vía celular, en ambos casos se ingresa la denuncia en el sistema, éste notificará el patrullero y designará los agentes especializados según el tipo de la denuncia

y su localización, a su vez el sistema irá mostrando en el mapa el camino para llegar a la denuncia en el menor tiempo posible.

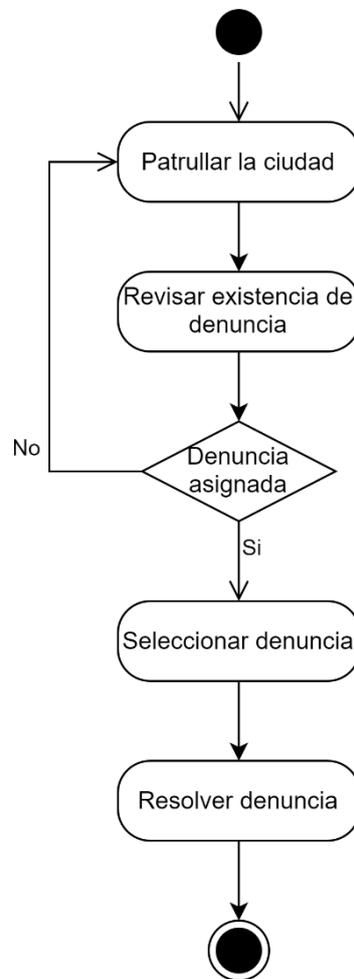


Figura 3.2: Flujo de actividad para asignación de denuncias.

Fuente: Elaboración propia.

Figura 3.2: Flujo de actividad para asignación de denuncias muestra el proceso secuencial para la gestión y distribución de denuncias en un sistema multiagente. Este diagrama ilustra cómo las denuncias son recibidas, procesadas y asignadas a los agentes correspondientes. Inicialmente, una denuncia es registrada en el sistema con su ubicación. A continuación, se selecciona un agente disponible que se encuentre más cercano al incidente y se le asigna la tarea de responder. El agente, tras recibir la asignación, se dirige al lugar del incidente, realiza las acciones necesarias y actualiza su estado en el sistema. Este flujo de actividad asegura una respuesta eficiente y coordinada a las denuncias, optimizando los recursos y tiempos de reacción dentro del sistema.

3.2.2. Diseño General de un Agente

Un agente inteligente es aquel que se le asigna un objetivo y es capaz de realizar acciones constantes que maximizan su función objetivo, optimizando su desempeño. En el contexto de un sistema de patrullaje policial, cada agente representa a un policía virtual con la capacidad de moverse, percibir su entorno, tomar decisiones y actuar en consecuencia. Los componentes del diseño de un agente consisten en:

- **Percepción:** Los agentes están equipados con sensores virtuales que les permiten recoger información del entorno. Estos sensores pueden simular la recepción de datos como la ubicación de denuncias y la posición de otros agentes.
- **Decisión:** Con la información percibida, el agente puede tomar una decisión para determinar la mejor acción a seguir. Esto puede incluir la selección de la ruta más rápida hacia una denuncia utilizando el algoritmo de Dijkstra y la coordinación con otros agentes para optimizar la cobertura.
- **Acción:** Una vez tomada la decisión, el agente ejecuta la acción correspondiente, como desplazarse hacia una ubicación específica, comunicarse con otros agentes y actualizar su estado en el sistema.

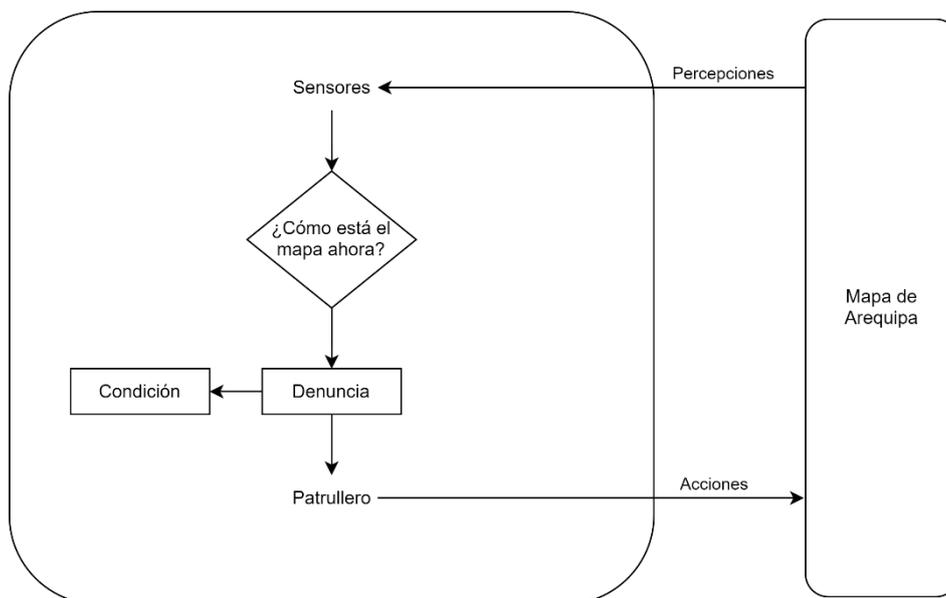


Figura 3.3: Estructura del Sistema Multiagente Policial.

Fuente: Elaboración propia.

En la Figura 3.3 se puede observar el comportamiento que deben seguir los agentes de la policía, donde múltiples agentes interactúan en un entorno compartido. Cada agente tiene un conjunto de sensores, un módulo de decisión y capacidades de actuación. La Figura 3.3 muestra cómo los agentes pueden intercambiar información y coordinarse para atender las denuncias de manera eficiente.

3.2.3. Diseño del Aprendizaje de un Agente

El aprendizaje de múltiples agentes se basa en el uso de técnicas de aprendizaje automático en un sistema de múltiples agentes. Generalmente, los agentes mejoran sus decisiones a través de la experiencia acumulada y es fundamental que un agente aprenda a coordinarse con los demás agentes para optimizar su desempeño.

La Figura 3.4 ilustra el proceso de aprendizaje de un agente. En este contexto, la percepción juega un rol crucial. La percepción de un agente se refiere a su capacidad para recibir e interpretar información de su entorno mediante sensores. Estos sensores incluyen datos de ubicación GPS.

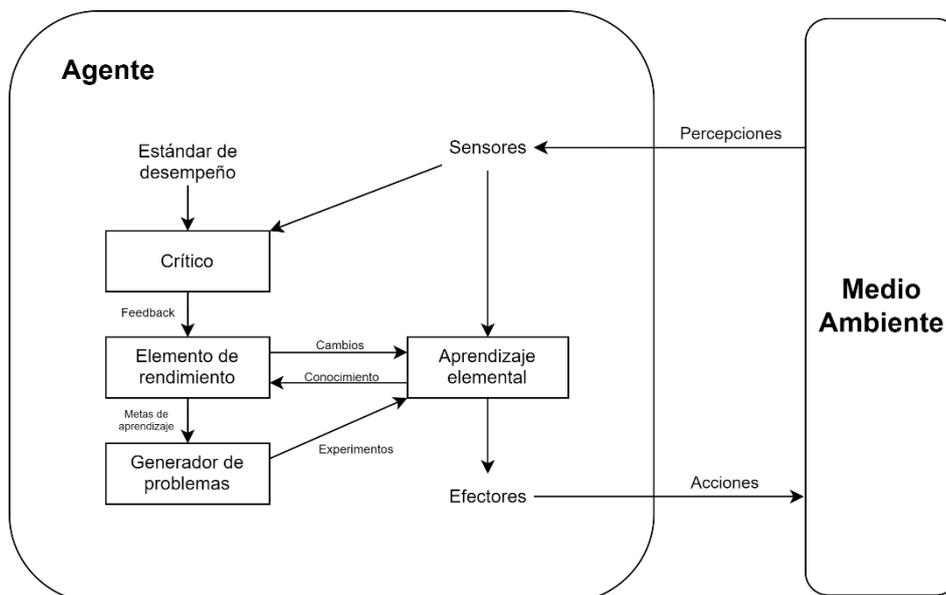


Figura 3.4: Proceso de aprendizaje de un agente virtual.

Fuente: Elaboración propia.

Cuando un agente recibe una denuncia, utiliza su percepción para evaluar su ubicación actual, la ubicación de la denuncia y la posición de otros agentes. El agente más cercano debe responder a la denuncia y, para ello, utiliza esta información para calcular la ruta más rápida hacia el lugar del incidente.

Después de atender la denuncia, los agentes restantes que están patrullando utilizan la experiencia acumulada para optimizar la respuesta a futuras denuncias. Estos agentes analizan los datos históricos de denuncias para ajustar sus rutas de patrullaje, asegurándose de que están posicionados estratégicamente para responder de manera eficiente a nuevas denuncias. Este proceso de decisión se ajusta continuamente mediante el uso de algoritmos de aprendizaje por refuerzo que permiten a los agentes aprender de sus experiencias y mejorar su desempeño con el tiempo.

Dada la Figura 3.4 se puede observar que el agente mejora su toma de decisiones a través de la simulación de múltiples escenarios de denuncias en la ciudad de Arequipa. Cada simulación proporciona nuevos datos de percepción que el agente utiliza para refinar sus estrategias de acción. A través de la experiencia continua, el agente no solo aprende a responder de manera más eficiente a las denuncias, sino que también mejora su capacidad para coordinarse con otros agentes en tiempo real.

El aprendizaje también implica la capacidad del agente para predecir el impacto de sus acciones en el entorno. Esto se logra mediante la implementación de modelos predictivos que utilizan datos históricos y actuales para anticipar posibles escenarios futuros y tomar decisiones informadas.

3.3. Codificación del Modelo

Una vez diseñado el modelo, el siguiente paso es su implementación. Esta sección proporciona una visión general del proceso de codificación, incluyendo las herramientas y lenguajes de programación utilizados.

3.3.1. Codificación de la Aplicación

En esta subsección, se detallará el proceso específico de codificación de la aplicación. Se describirán los módulos y funciones desarrollados, así como los desafíos técnicos enfrentados y las soluciones implementadas.

3.3.1.1 Requisitos del Sistema

Antes de iniciar la codificación, se identificaron y definieron los requisitos del sistema, estableciendo las funciones y características esenciales que la aplicación debía abordar. Estos requisitos fueron el resultado de una exhaustiva revisión de los objetivos de la investigación y

de las necesidades específicas de la Policía Nacional del Perú en la ciudad de Arequipa. Los detalles de estos requisitos se encuentran disponibles en el Anexo 3.

3.3.1.2 Base de Datos

Las bases de datos son componentes críticos en cualquier sistema de información, especialmente en aquellos que manejan datos geoespaciales y requieren alta precisión y eficiencia. En este apartado se detalla la implementación y estructura de la base de datos utilizada en el modelo de agentes para la gestión de denuncias policiales en la ciudad de Arequipa.

Esta base de datos es fundamental para el correcto funcionamiento del sistema, ya que permite almacenar y consultar información geoespacial de manera eficiente. Además, su integración con el modelo de agentes facilita la planificación y respuesta a incidentes en tiempo real, mejorando la capacidad de la Policía Nacional del Perú para gestionar y resolver denuncias en la ciudad de Arequipa.

Para respaldar la funcionalidad espacial del modelo, se implementará una base de datos utilizando PostGIS, una extensión espacial para PostgreSQL. PostGIS ofrece un conjunto robusto de funciones y capacidades espaciales que son esenciales para el manejo eficiente de datos geoespaciales.

La base de datos contiene un total de 1000 registros. Estos registros son generados aleatoriamente y representan denuncias policiales, las cuales se utilizan para modelar y simular las operaciones de patrullaje y respuesta de la Policía Nacional del Perú en la ciudad de Arequipa. Cada registro en la base de datos incluye la siguiente información:

- **ID de la Denuncia:** Un identificador único para cada denuncia.
- **Ubicación:** Coordenadas geográficas (latitud y longitud) de la denuncia.
- **Fecha hora:** El momento en que se realizó la denuncia

Denuncias
id_denuncia
latitud
longitud
fecha_hora

Figura 3.5: Tabla de Denuncias.

Fuente: Elaboración propia.

En la Figura 3.5 se presenta la Tabla de Denuncias, que muestra la distribución de denuncias registradas en el sistema multiagente diseñado para la Policía Nacional del Perú (PNP), facilitando la planificación y respuesta a incidentes en tiempo real. Para respaldar la funcionalidad espacial del modelo, se implementó una base de datos utilizando PostGIS, una extensión espacial para PostgreSQL. PostGIS ofrece un conjunto robusto de funciones y capacidades espaciales esenciales para el manejo eficiente de datos geoespaciales.

La tabla principal Denuncia tiene la siguiente estructura:

```
CREATE TABLE Denuncia (
  id_denuncia SERIAL PRIMARY KEY,
  latitud DOUBLE PRECISION NOT NULL,
  longitud DOUBLE PRECISION NOT NULL,
  fecha_hora TIMESTAMP NOT NULL
);
```

A continuación, se presentan algunos ejemplos de los datos almacenados en la tabla **Denuncia**:

Id denuncia	Latitud	Longitud	Fecha hora
4321	-16.4090	-71.5374	2024-06-10 08:30:00
4322	-16.3988	-71.5352	2024-06-10 09:15:00
4323	-16.4072	-71.5400	2024-06-10 10:15:00

Tabla 3.1: Registro de Denuncias

La Tabla 3.1 muestra el Registro de Denuncias, donde se detalla información crucial para cada denuncia registrada en el sistema, proporcionando un marco temporal preciso para el análisis y seguimiento de los casos.

3.3.1.3 Programación

Para la programación del modelo se utiliza el lenguaje de programación Python, con la herramienta OpenStreetMap que proporciona datos de mapas, necesaria para la visualización de las denuncias activas y la ruta más rápida generada.

3.3.2. Codificación del Sistema multiagente

Se utilizará un entorno de múltiples agentes como base, inspirado en la simulación de un escenario policial dinámico. En este entorno, los agentes representan el personal policial asignado en los patrulleros, y las situaciones de emergencia corresponden a las denuncias presentadas por los ciudadanos.

También se utilizará un modelo de Aprendizaje por Refuerzo (RL) para optimizar las decisiones de los agentes en el sistema multiagente. El algoritmo específico implementado fue Q-learning, uno de los métodos más conocidos y utilizados en RL.

Algoritmo del Sistema Multiagente

1. Inicialización del Sistema Multiagente:

Inicializar el sistema con un número específico de agentes policiales.

Entrada: Número de agentes, será entero que representa la cantidad de agentes a crear.

Proceso: Se crea una lista de agentes y se genera un AgentePolicial en una posición aleatoria. Luego, se añade el AgentePolicial a la lista de agentes.

Salida: Lista de agentes inicializados.

2. Ejecución del Sistema Multiagente:

Ejecutar el sistema durante un número específico de iteraciones.

Entrada: Las iteraciones son un entero que representa la cantidad de iteraciones a ejecutar, mientras que los agentes son una lista de agentes inicializados.

Proceso: Empieza la iteración, cada agente va a Patrullar o resolver una denuncia

Salida: Estado final de los agentes después de todas las iteraciones.

3. Funciones del Agente Policial:

Determinar la acción del agente basado en su estado actual (patrullando o respondiendo).

Hay denuncia: Se debe verificar si hay una denuncia en la ubicación actual, devolviendo Verdadero si hay una denuncia y Falso en caso contrario.

Responder a denuncia: Cambiar el estado del agente a "respondiendo" y dirigir al agente hacia la denuncia.

Patrullar: Realizar acciones de patrullaje en la zona.

Dirigirse a denuncia: Determinar la ruta más corta hacia la denuncia utilizando algoritmos de ruta (Dijkstra).

Cambiar estado: Cambiar el estado actual del agente al nuevo estado proporcionado.

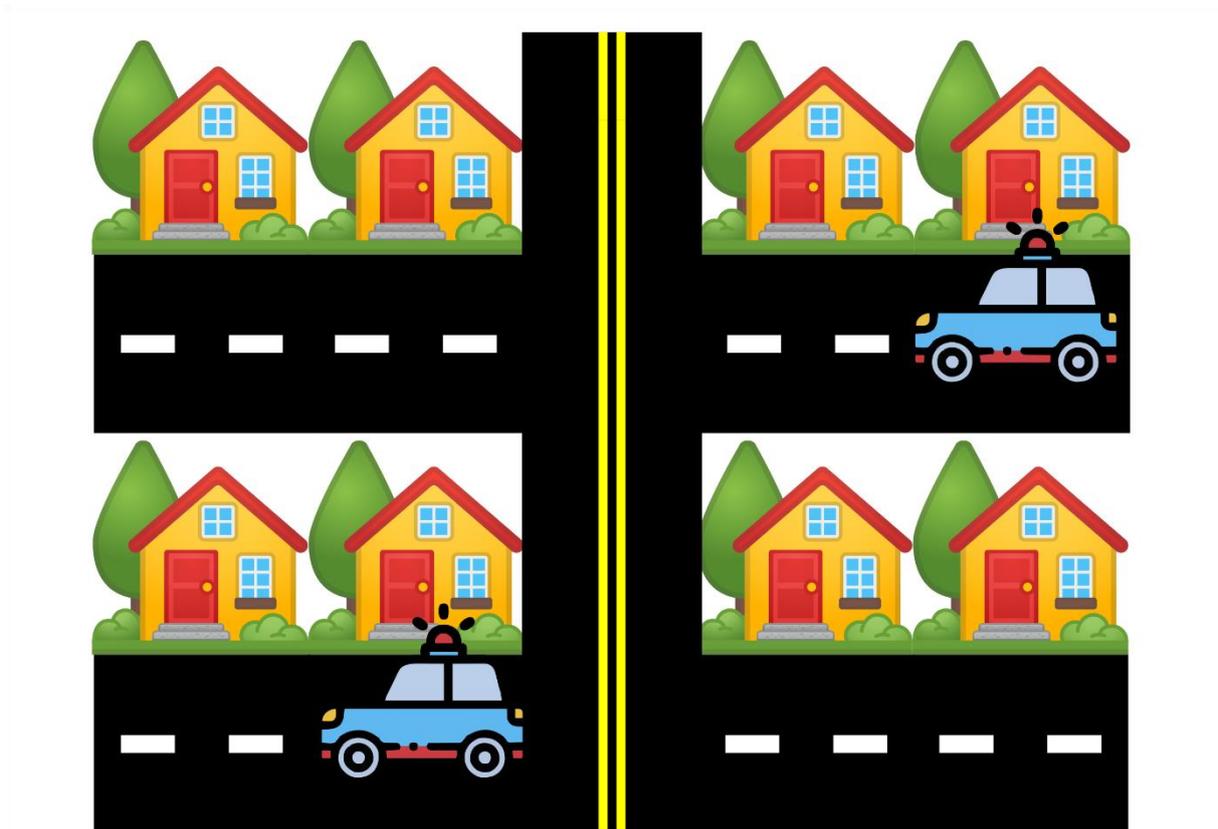


Figura 3.6: Diagrama de Patrullaje de Policías en el Sistema Multiagente.

Fuente: Elaboración propia.

Figura 3.6: Esquema de codificación del sistema multiagente muestra cómo se estructuran y operan los agentes policiales dentro del sistema. Inicialmente, los agentes son creados y posicionados aleatoriamente en la función Inicializar de la clase SistemaMultiagente, donde cada agente (AgentePolicial) comienza patrullando. En cada iteración del sistema, la función Actuar verifica si hay una denuncia con hay_denuncia.

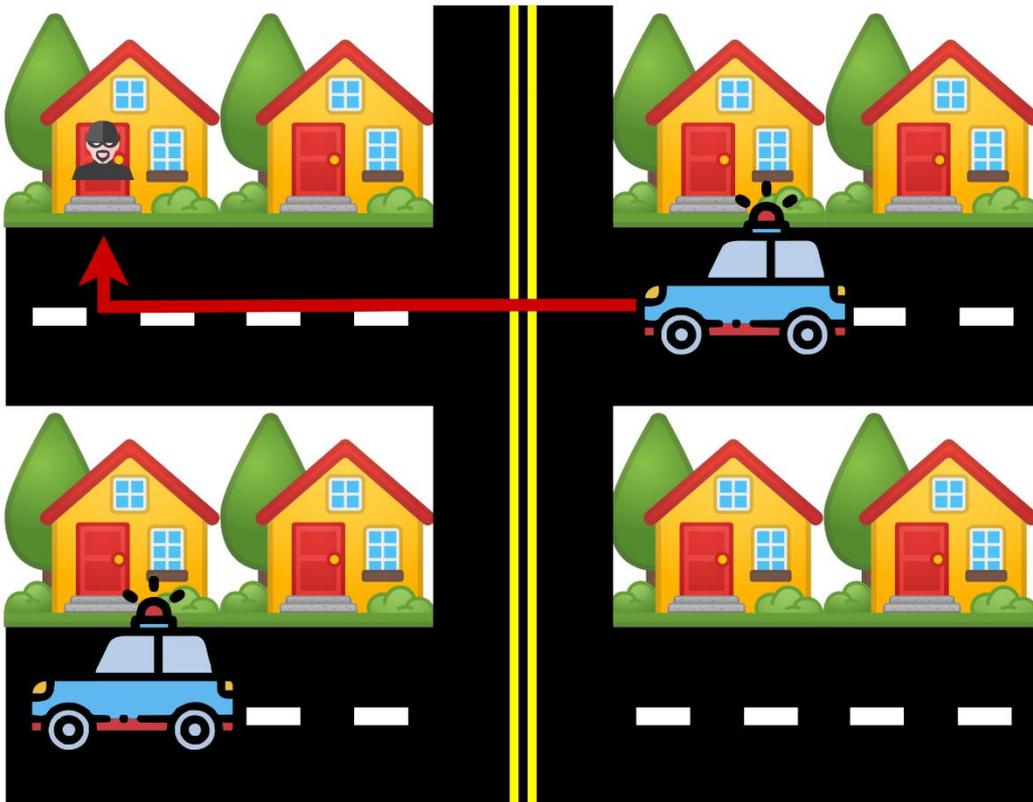


Figura 3.7: Diagrama de Respuesta del Patrullero Más Cercano ante una Denuncia en el Sistema Multiagente.

Fuente: Elaboración propia.

En la Figura 3.7 el agente cambia su estado a "respondiendo" y se dirige al lugar del incidente utilizando la función `dirigirse_a_denuncia`. Si no hay denuncias, el agente continúa patrullando mediante la función `patrullar`. En la figura se puede ver que el agente más cercano a la denuncia va a resolverla y el otro agente sigue en su patrullaje. Este ciclo continuo de patrullaje y respuesta, asegura que los agentes policiales respondan dinámicamente a los eventos en tiempo real, tal como se ilustra en el esquema de la figura.

3.4. Entrenamiento del modelo

En esta sección, se describe en detalle el proceso de entrenamiento del modelo de asignación dinámica de tareas utilizando un enfoque de aprendizaje por refuerzo. Se utilizó un código base que integra entornos con múltiples agentes basados en Gym OpenAI, fusionándose con la aplicación de OpenStreetMap para utilizar el mapa como entorno de ejecución.

El proceso de entrenamiento implica la interacción dinámica entre los agentes de policía (patrulleros) y el entorno simulado de denuncias. A continuación, se detallan los pasos clave del proceso de entrenamiento:

3.4.1 Inicialización del Entorno

En la fase inicial, se da el puntapié inicial a nuestra simulación mediante la "Inicialización del Entorno". Aquí, se configuran las condiciones iniciales de nuestro modelo, generando patrulleros con sus atributos característicos y denuncias con detalles aleatorios. Estos elementos son cuidadosamente situados en el mapa virtual, estableciendo las bases para la interacción dinámica entre los agentes y las situaciones de denuncias que se desarrollarán durante el entrenamiento.

Algoritmo de Inicialización del Entorno

Configurar el entorno de simulación creando agentes patrulleros, generando denuncias aleatorias y posicionando estos elementos en el mapa.

Entrada: Parámetros del entorno como el tamaño del mapa y la cantidad de objetos.

Proceso:

1. **Crear Agentes Patrulleros:** Crear un agente con atributos iniciales (ubicación, estado) y asignar el agente a un vehículo
2. **Crear Denuncias Aleatorias:** Genera ubicaciones aleatorias para la creación de denuncias
3. **Posicionar en el Mapa:** Para cada agente y denuncia, se selecciona una ubicación inicial aleatoria dentro del mapa, verificando que esta no esté ocupada por otro agente o denuncia. Luego, se coloca el agente o la denuncia en la ubicación seleccionada.

Salida: Mapa inicializado con todos los agentes y denuncias distribuidos.

3.4.2 Toma de Decisiones de los Patrulleros

Una vez preparado el entorno, los "Patrulleros" comienzan a evaluar la situación actual y a tomar decisiones estratégicas. Esta fase implica no solo la cuidadosa selección de denuncias a atender y la elección de rutas óptimas para maximizar la eficiencia en la atención de las denuncias, sino también la asignación de rutas de patrullaje cuando no hay denuncias disponibles. De esta manera, los patrulleros aseguran una presencia constante y vigilancia proactiva en el área asignada.

Algoritmo de Toma de Decisiones de los Patrulleros

Determinar y ejecutar las acciones necesarias para que los patrulleros respondan a las denuncias de manera eficiente.

Entrada: Estado actual del entorno incluyendo la ubicación de los patrulleros y las denuncias.

Proceso:

1. **Evaluar Estado Actual:** Para cada patrullero se evalúa la ubicación actual y el estado de las denuncias cercanas.
2. **Seleccionar Denuncias:** Para cada patrullero se le asignan rutas óptimas para llegar a las denuncias seleccionadas.
3. **Interactuar con el Entorno:** Para cada patrullero, es crucial moverse hacia la denuncia asignada siguiendo la ruta calculada, atender la denuncia y actualizar el estado del entorno según las acciones tomadas. Si no hay una denuncia asignada, se le asigna una ruta de patrullaje y se mueve a lo largo de esta ruta.

Salida: Acciones de los patrulleros, rutas seguidas y estado actualizado del entorno.

3.4.3 Actualización del Estado del Entorno

Una vez que los patrulleros han tomado acciones, el estado del entorno se ajusta para reflejar las consecuencias de esas acciones. Las ubicaciones de agentes y denuncias se modifican, se actualizan para mantener un entorno dinámico.

Algoritmo de Actualización del Estado del Entorno

Actualizar el estado del entorno basándose en las acciones tomadas por los patrulleros y las denuncias atendidas.

Entrada: Acciones de los patrulleros y estado actual del entorno.

Proceso:

1. **Cuando el patrullero llegue a la denuncia:** Para cada patrullero, es esencial moverse según la ruta asignada. Si se llega a una denuncia, se debe eliminar la marca de la denuncia, incrementar el contador de denuncias resueltas y actualizar el estado del entorno. En caso de no tener una denuncia asignada, el patrullero debe continuar patrullando.
2. **Actualizar posiciones:** Actualizar las posiciones de los vehículos y sus marcadores en el mapa.

Salida: Estado actualizado del entorno, incluyendo la posición de los patrulleros y el estado de las denuncias.

3.4.4 Cálculo de Recompensa

La eficiencia y pertinencia en la atención de las denuncias se traducen en una "recompensa" esencial para guiar el aprendizaje del modelo. Este proceso implica la evaluación de la actuación, asignación de recompensas según la eficiencia y ajuste de la recompensa acumulativa del modelo.

Algoritmo Cálculo de Recompensa:

- Obtener la recompensa máxima esperada del siguiente estado.
- Obtener el valor Q actual del estado y nodo actual.
- Calcular el nuevo valor Q utilizando la fórmula de actualización de Q-Learning.
- Si el estado no existe en la tabla Q, inicializarlo.
- Actualizar el valor Q del estado y nodo actual en la tabla Q.

3.4.5 Aprendizaje y Actualización de Parámetros

En este proyecto, se utiliza el algoritmo de aprendizaje por refuerzo Q-learning para mejorar la toma de decisiones de los agentes de vehículos. La elección de Q-learning se basa en varias razones: es un algoritmo sencillo y eficiente que no requiere un modelo del entorno, lo cual facilita su implementación y adaptación a diferentes contextos. Además, Q-learning puede

manejar espacios de estado y acción discretos y continuos, lo que lo hace adecuado para la variedad de escenarios y decisiones que deben tomar los agentes de vehículos. También permite que los agentes aprendan y mejoren sus decisiones en tiempo real basándose en la retroalimentación del entorno, lo que es crucial para adaptarse a cambios dinámicos en las condiciones de patrullaje. Finalmente, Q-learning busca maximizar la recompensa acumulada a largo plazo, lo que se alinea con el objetivo de optimizar las rutas de patrullaje para reducir el tiempo de respuesta a las denuncias.

La fórmula se utiliza para actualizar la tabla Q, que es una especie de memoria donde el agente (vehículo) guarda el valor esperado de resolver cada denuncia en cada estado. Esto le permite aprender y mejorar su toma de decisiones con el tiempo.

$$Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s,a)]$$

Ecuación 1: Formula de Q-Learning.

Fuente: Learning From Delayed Rewards [42].

Como se ve en la Ecuación 1, se puede comprender por los siguientes componentes:

1. Q(s, a):

- **Q:** Representa el valor de tomar la acción en el estado s.
- **s:** Representa la ubicación actual de la denuncia.
- **a:** Representa la acción, es el nodo donde el agente se encuentra actualmente.

2. Tasa de aprendizaje (α): Controla cuánto influye la nueva información en la actualización de los valores Q. Un valor alto de α significa que las nuevas recompensas tienen un impacto mayor en los valores Q.

3. Recompensa (r): Es la recompensa inmediata que el agente recibe después de tomar la acción a en el estado s. Como por ejemplo: Si el vehículo llega a resolver una denuncia, puede recibir una recompensa positiva.

4. Factor de descuento (γ): Determina la importancia de las recompensas futuras. Un valor de γ cercano a 1 enfoca el algoritmo en las recompensas a largo plazo, mientras que un valor más bajo prioriza las recompensas inmediatas.

5. **Max $Q(s',a')$** : Toma el valor máximo Q entre todas las acciones posibles en ese estado.

Si bien existen otros algoritmos de aprendizaje por refuerzo que pueden ayudar al modelo, se decidió utilizar Q-learning por su simplicidad y eficiencia, lo cual es crucial cuando se trabaja con datos simulados. Aunque Double Q-learning ofrece mejoras en la precisión de la estimación de valores Q al reducir el sesgo, esta ventaja puede no ser tan crítica cuando se entrenan modelos con datos generados artificialmente. La facilidad de implementación y la menor complejidad de Q-learning lo hacen más adecuado para este proyecto. También se consideró el uso de Deep Q-Networks (DQN), pero requiere una mayor cantidad de datos, así como recursos computacionales y tiempo de entrenamiento, lo cual no era práctico para los objetivos del proyecto.

El proceso de Q-learning comienza con la inicialización de la tabla Q con valores arbitrarios. En cada paso del tiempo, el agente elegirá una acción que balancee la exploración y la explotación. En el código, se utilizará una estrategia epsilon-greedy, donde epsilon es la probabilidad de seleccionar una ruta aleatoria para explorar nuevas posibilidades. Esto permite al agente, en cada momento de decisión, elegir entre explorar nuevas rutas (exploración) o seguir la mejor ruta conocida según su tabla Q (explotación). Posteriormente, el agente ejecuta la acción seleccionada y observa la recompensa obtenida y el nuevo estado al que se transita. La fórmula de actualización ajusta el valor Q para el par estado-acción actual, basándose en la recompensa inmediata y el valor Q del estado siguiente, permitiendo que el agente aprenda a tomar decisiones que maximicen la recompensa acumulada a largo plazo. La tasa de aprendizaje (α) y el factor de descuento (γ) controlan cuánto influyen las nuevas recompensas y las recompensas futuras, respectivamente. Este ciclo se repite hasta que se cumpla un criterio de finalización, como un número máximo de episodios o la convergencia de la tabla Q .

En el algoritmo Q-learning, los hiperparámetros son cruciales para el desempeño del agente. Anteriormente, en esta sección se han descrito dos conceptos clave: la tasa de aprendizaje (α) y el factor de descuento (γ). A continuación, se profundiza en estos parámetros en el contexto de su aplicación específica, y lo demás principales hiperparámetros utilizados en este proyecto:

a) Tasa de aprendizaje (α): La tasa de aprendizaje controla cuánto influye la nueva información en la actualización de los valores Q . Un valor alto de α significa que las nuevas

recompensas tienen un impacto mayor en los valores Q. Como se discutió anteriormente, este parámetro impacta directamente en la eficiencia del aprendizaje del modelo.

b) Factor de descuento (γ): El factor de descuento, como se mencionó anteriormente, determina la importancia de las recompensas futuras en comparación con las inmediatas. Este parámetro es crucial para influenciar la toma de decisiones y la priorización de rutas a largo plazo.

c) Epsilon (ϵ): Define la probabilidad de que el agente elija una ruta aleatoria en lugar de la mejor acción conocida. Este valor decrece con el tiempo para favorecer la explotación de acciones conocidas a medida que el agente aprende.

El siguiente apartado detalla el algoritmo desarrollado para el agente policial, abarcando desde la inicialización de sus parámetros hasta el proceso de aprendizaje y la actualización continua de su conocimiento. Este algoritmo está diseñado para optimizar las rutas de patrullaje del agente, balanceando entre la exploración de nuevas rutas y la explotación de rutas previamente conocidas y evaluadas como efectivas.

Parámetros Agente Policial:

Asigna valores iniciales a los parámetros del agente:

- **Velocidad:** velocidad predeterminada.
- **Estado:** "patrullando".
- **Posición:** posición inicial del agente.
- **Epsilon:** valor inicial para la exploración.
- **Alpha:** tasa de aprendizaje inicial.
- **Gamma:** factor de descuento inicial.
- **Table:** inicializa la tabla Q vacía.

Algoritmo de Aprendizaje

1. **Retroalimentación de Recompensa:** Cuando el agente llega a la ubicación de una denuncia, obtiene la recompensa correspondiente. Luego, evalúa la calidad de la acción

basada en la recompensa obtenida, lo que le permite determinar la efectividad de las estrategias implementadas.

2. **Algoritmo de Aprendizaje:** El agente decide si tomar una ruta de patrullaje aleatoria o una acción basada en la tabla Q. Dependiendo de la decisión, el agente puede optar por la exploración o la explotación. Una vez que toma una acción y se mueve a un nuevo nodo, recibe una recompensa solo si llega a la ubicación de una denuncia. Posteriormente, la tabla Q se actualiza utilizando la fórmula de Q-learning después de recibir la recompensa.
3. **Actualización Continua:** Implica que el agente está en constante aprendizaje y ajuste mientras opera. En cuanto a la decisión de movimiento, si el agente tiene una ruta definida, se mueve al siguiente nodo en la ruta. Si no tiene una ruta definida, decide su siguiente movimiento basado en exploración o explotación. Cuando el agente llega a la ubicación de una denuncia, actualiza la tabla Q con una recompensa. Si el agente no está respondiendo a una denuncia, se le asigna una nueva ruta de patrullaje a una denuncia, asigna una nueva ruta de patrullaje.

3.5. Diseño de las Pruebas

Las pruebas son fundamentales para garantizar que el sistema multiagente funcione de manera correcta y eficiente. En esta sección, se describirán los diferentes tipos de pruebas realizadas, incluyendo pruebas unitarias, de integración y de rendimiento, así como los resultados obtenidos y los ajustes necesarios para mejorar el funcionamiento del sistema.

1. Pruebas Unitarias

Las pruebas unitarias son esenciales para verificar el funcionamiento preciso de funciones y métodos individuales en el código. Para asegurar la funcionalidad básica de movimiento de los agentes se creará una instancia del modelo de transporte y se obtiene el primer agente del modelo junto con su nodo inicial. Al ejecutar un paso en el modelo, el agente se mueve, y se verifica que el nodo actual del agente no sea igual al nodo inicial.

2. Pruebas de sistema

Las pruebas de sistema deben asegurar que el sistema cumpla con los requisitos y funcionalidades clave, como el manejo de denuncias. Se propone una prueba de verificación de

reinicio que se asegure de que la posición inicial de los agentes sea diferente en cada reinicio, al igual que la posición inicial de las denuncias.

3. Prueba de Humo

Las pruebas de humo son pruebas básicas que verifican la ejecución sin errores graves. En el siguiente pseudocódigo, se ejecuta una simulación básica y se verifica que la simulación se haya completado exitosamente: Para lograrlo en esta prueba, se verificará que el número de agentes y denuncias creadas sean correctos y sin ningún error

4. Pruebas Rendimiento

Las pruebas de rendimiento son esenciales para evaluar cómo nuestro sistema de simulación de agentes responde y se comporta bajo condiciones de carga y estrés. Estas pruebas se centran en medir la capacidad del sistema para manejar un número significativo de agentes y la eficiencia general en términos de tiempo de respuesta y uso de recursos. A continuación, se presentan pruebas específicas diseñadas para evaluar el rendimiento del sistema desde diferentes perspectivas.

Prueba 1: Evaluación del rendimiento con un gran número de agentes

Esta prueba simula la ejecución del sistema con un gran número de agentes concurrentes, permitiéndonos evaluar la escalabilidad y la estabilidad del sistema. Para lograrlo en esta prueba, se realizará una iteración de 300 pasos con 600 agentes. Los agentes patrullarán o patrullarán y crearán una denuncia cada cierto tiempo. Esta prueba simula la ejecución del sistema con un gran número de agentes concurrentes, permitiéndonos evaluar la escalabilidad y la estabilidad del sistema

Prueba 2: Evaluación del rendimiento con una simulación de larga duración

En esta prueba, realizamos una simulación extensa para evaluar la capacidad del sistema para manejar escenarios de larga duración sin degradación significativa del rendimiento. Para lograrlo en esta prueba, se realizará una iteración de 5000 pasos con 200 agentes. Los agentes patrullarán o patrullarán y crearán una denuncia cada cierto tiempo. Esta prueba simula una ejecución extensa para evaluar la capacidad del sistema para manejar escenarios de larga duración sin degradación significativa del rendimiento

Prueba 3: Medición de latencia y tiempo de respuesta

Esta prueba se enfoca en medir la latencia del sistema y el tiempo de respuesta durante la ejecución de una acción, proporcionando información sobre la eficiencia en términos de velocidad de respuesta. Para lograrlo esta prueba se enfoca en medir la latencia del sistema y el tiempo de respuesta durante la ejecución de una acción, proporcionando información sobre la eficiencia en términos de velocidad de respuesta. Para medir la latencia y el tiempo de respuesta, se reiniciará el entorno y se registrará el tiempo inicial y el tiempo que se demora en comenzar el patrullaje.

Prueba 4: Monitoreo del uso de recursos del sistema

En esta prueba, monitoreamos el uso de recursos como la CPU y la memoria durante la simulación para asegurarnos de que el sistema no genere una carga excesiva. Para lograrlo en esta prueba, se reiniciará el entorno y se creará una lista para guardar los recursos usados. Se realizará una iteración de 1000 pasos en la que los agentes patrullarán o resolverán denuncias, y las denuncias se crearán cada cierto tiempo. Se monitorearán los recursos de CPU y memoria, y estos datos se guardarán en la lista

Capítulo IV: Resultados y Discusión

Los resultados obtenidos se procedieron a analizar, dando respuesta a las preguntas de investigación, las evidencias se explican a continuación.

4.1. Recopilar información

En este capítulo, se presentan los resultados derivados de la recopilación de información, que incluye la entrevista realizada con la Policía Nacional del Perú y el análisis de documentos estatales relacionados con la seguridad ciudadana en la ciudad de Arequipa.

4.1.1 Entrevistas con la Policía Nacional del Perú

Durante las entrevistas con la Policía Nacional del Perú, se obtuvieron percepciones valiosas sobre el funcionamiento de las comisarías en la ciudad de Arequipa. El Capitán de la Policía Nacional del Perú que por cuestiones de confidencialidad de datos no se puede mencionar su nombre proporcionó información detallada sobre los procedimientos actuales de recepción de denuncias y su impacto en el patrullaje policial.

Entre los aspectos destacados de las entrevistas se incluyen:

Procedimientos de recepción de denuncias: El Capitán de la Policía Nacional del Perú explicó en detalle cómo la Policía Nacional del Perú recibe y procesa las denuncias ciudadanas, destacando los desafíos y limitaciones que enfrentan en este proceso.

Impacto de denuncias anteriores en el patrullaje: Se discutió cómo las denuncias anteriores pueden influir en la asignación de recursos y la planificación del patrullaje policial, lo que proporciona una comprensión más profunda de la dinámica de seguridad en la ciudad.

4.1.2 Análisis de Documentos Estatales

El análisis de documentos estatales, incluido el "Plan de Acción Distrital de Seguridad Ciudadana de Yanahuara" y el "Plan Anual de Codisex Yanahuara", arrojó luz sobre las estrategias gubernamentales en relación con la seguridad ciudadana en la región. Estos documentos proporcionan una perspectiva complementaria a las entrevistas con la Policía Nacional del Perú.

Además, se utilizó el mapa de delitos proporcionado por el Instituto Nacional de Estadística e Informática (INEI) para contextualizar los datos recopilados y comprender mejor la distribución geográfica de los delitos en la ciudad de Arequipa.

4.2. Diseño Modelo

El diseño del modelo es un paso crítico que define la estructura y el comportamiento del sistema. En esta sección, se presentará el proceso de diseño del modelo propuesto, destacando los componentes clave, sus interacciones y las consideraciones técnicas que guiaron su desarrollo.

4.2.1 Diseño General de un Agente

El siguiente algoritmo representa lo implementado en código y el diseño general de un agente de vehículo para su implementación en un entorno de simulación:

Algoritmo para la Clase Vehicle Agent

Entrada: identificador unico y modelo se pasan a la superclase mesa.Agent.

Parámetros Agente Policial:

Asigna valores iniciales a los parámetros del agente:

- **Posición:** Se establece en un nodo aleatorio del grafo del modelo, representando la posición inicial del agente.
- **Ruta planificada:** Se inicializa como una lista vacía, que almacenará la ruta planeada.
- **Velocidad:** se establece en un valor aleatorio entre 5 y 15, simulando velocidades variables de los agentes.
- **Estado:** es un booleano que indica si el agente está respondiendo a una queja.
- **Ubicación actual queja:** es inicialmente **None**, y se actualizará con la ubicación de una queja cuando se asigne.
- **Epsilon:** valor inicial para la exploración.
- **Alpha:** tasa de aprendizaje inicial.
- **Gamma:** factor de descuento inicial.

- **Table:** se inicializa como un diccionario vacío para almacenar los valores Q para las parejas estado-acción en el algoritmo de aprendizaje Q.

Este agente se caracteriza por su capacidad para moverse a través de un grafo que representa la red vial de la ciudad, seleccionando un nodo aleatorio como punto de partida. El agente puede responder a denuncias estableciendo su ubicación actual y planificando una ruta hacia la ubicación de la denuncia.

4.2.2 Diseño del Aprendizaje de un Agente

Dado el entorno dinámico y complejo en el que operan los agentes de seguridad, es crucial que puedan aprender y adaptarse a nuevas situaciones. Se implementó un enfoque de aprendizaje automático que permita a los agentes mejorar sus decisiones a través de la experiencia acumulada.

El aprendizaje se generará mediante la simulación de múltiples escenarios de denuncias en la ciudad de Arequipa. Los agentes aprenderán de forma continua a partir de esta experiencia, refinando sus estrategias y mejorando su capacidad para coordinarse con otros agentes en tiempo real.

Para facilitar la implementación de este enfoque, se diseñó un modelo de transporte que servirá como entorno de simulación para los agentes de seguridad. El siguiente código representa la definición de este modelo:

Algoritmo para la Clase Vehicle Agent

Entrada: ninguna

Proceso:

- Llama al constructor de la superclase mesa.Model para garantiza que toda la configuración inicial y se ejecuten correctamente.
- **Definición de Límites Geográficos:** Se establecen los límites norte, oeste, sur y este del área geográfica de interés.
- **Creación del Grafo de Transporte:** Se crea un grafo de la red de carreteras dentro de los límites geográficos definidos, utilizando datos de OpenStreetMap.

- **Inicialización de la Cuadrícula Espacial:** Se inicializa una cuadrícula espacial que representa la estructura de la red vial.
- **Inicialización del Programador de Agentes:** Se inicializa un scheduler para la activación simultánea de los agentes.
- **Marcadores de Vehículos:** Diccionario para marcar la ubicación de los vehículos.
- **Marcadores de Denuncias:** Diccionario para marcar la ubicación de las denuncias.
- **Marcadores Guardados:** Diccionario para almacenar marcadores guardados.
- **Denuncias Resueltas:** Lista para almacenar las denuncias que han sido resueltas en cada paso de tiempo.

Este modelo proporciona una representación del sistema de transporte en la ciudad de Arequipa, lo que permitirá a los agentes moverse y responder a las denuncias de manera realista dentro del entorno simulado.

4.3. Codificación del Modelo

La implementación de la aplicación es la etapa en la que se traduce el diseño conceptual en código funcional. En esta sección, se detalla la implementación del modelo de simulación de patrullaje policial basado en agentes, explicando el proceso de codificación del modelo, los lenguajes y herramientas utilizados, las principales funcionalidades implementadas y los retos enfrentados durante el desarrollo.

Se utilizó el lenguaje de programación Python para la implementación del modelo. La herramienta OpenStreetMap proporcionó los datos de mapas necesarios para la visualización de las denuncias activas y la generación de la ruta más rápida.

El algoritmo para el modelo de simulación de patrullaje policial se presenta a continuación:

1. Algoritmo para la Clase Agente Policial

Entrada: Posición inicial del agente en el plano y estado actual del agente (inicialmente "patrullando").

Funciones:

- **actuar:** Determina la acción del agente basado en su estado (patrullar o responder a una denuncia).
- **hay_denuncia:** Verifica si hay una denuncia en la ubicación actual, devolviendo True o False aleatoriamente.
- **responder_a_denuncia:** Cambia el estado del agente a "respondiendo" y dirige al agente hacia la denuncia.
- **patrullar:** Realiza acciones de patrullaje en la zona (por ejemplo, imprimir "Patrullando la zona").
- **dirigirse_a_denuncia:** Determina la ruta más corta hacia la denuncia utilizando algoritmos de ruta Dijkstra.
- **cambiar_estado(nuevo_estado):** Cambia el estado actual del agente al nuevo estado proporcionado.

2. Algoritmo para la Clase Sistema Multiagente

Inicialización

Propósito: Crear una instancia del sistema multiagente con un número específico de agentes.

Entrada: Lista de agentes del sistema.

Proceso:

- **Iteraciones:** Ejecuta la simulación durante un número específico de iteraciones. En cada iteración de la simulación, cada agente realiza su patrullaje y, en caso de recibir una denuncia cercana, se dirige a solucionarla.

4.4. Entrenamiento del modelo

El entrenamiento del modelo se realiza mediante el algoritmo de aprendizaje por refuerzo Q-learning, el cual se descompone en varias etapas que se detallan a continuación.

4.4.1 Inicialización del Entorno

Durante la fase de inicialización del entorno, se configuraron las condiciones iniciales del modelo para simular el escenario de asignación dinámica de tareas. Como se muestra en la captura de pantalla del mapa a continuación, se generaron 248 patrulleros (marcados con marcadores azules) y 5 denuncias (marcadas con marcadores rojos) con detalles aleatorios. Estos elementos fueron situados en el mapa virtual para establecer las bases de interacción y simular el escenario de patrullaje y respuesta a denuncias.

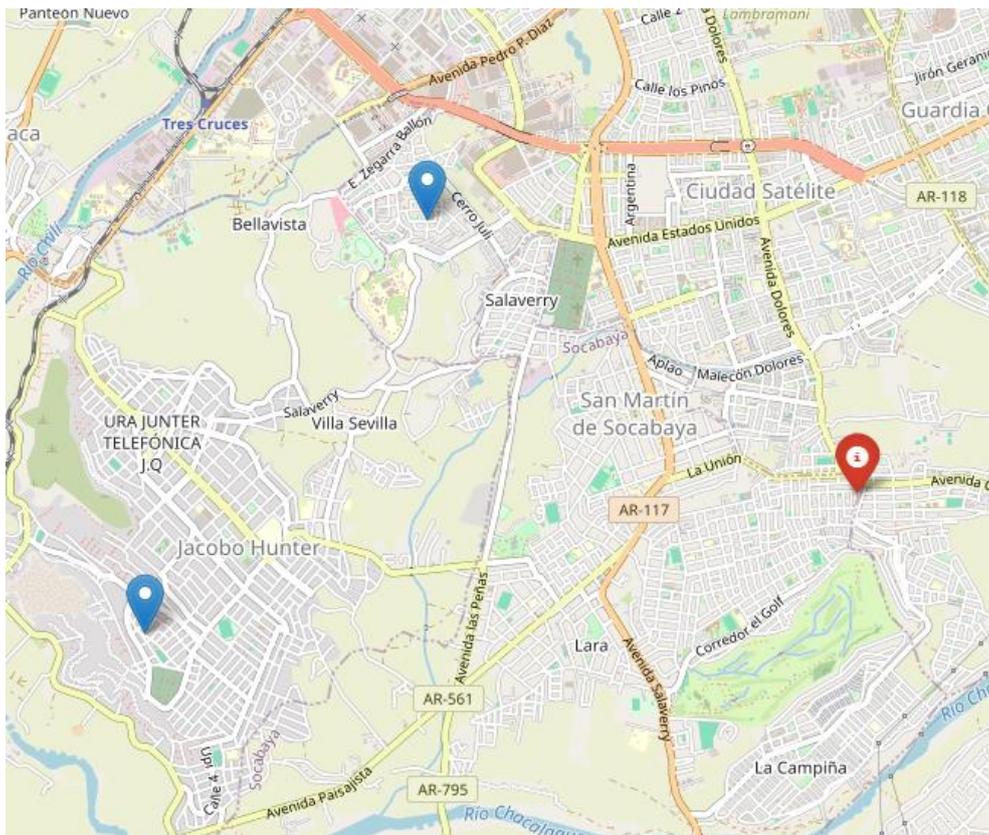


Figura 4.1: Proceso de inicialización del entorno simulado.

Fuente: Elaboración propia.

En la Figura 4.1 los marcadores azules representan la ubicación inicial de los patrulleros, cada uno con sus atributos característicos y asignados a una ubicación aleatoria dentro del área del mapa. La denuncia generada, representada por el marcador rojo, también se encuentra en una ubicación aleatoria dentro del área del mapa, lista para ser atendida por los patrulleros.

4.4.2 Toma de Decisiones de los Patrulleros

Durante cada paso del tiempo, los patrulleros deben decidir si explorar nuevas rutas o patrullar por la información almacenada en la tabla Q para maximizar la recompensa esperada. Esta decisión se toma utilizando una política epsilon-greedy, donde con probabilidad epsilon se elige una acción aleatoria (patrullaje) y con probabilidad $1 - \text{epsilon}$ se elige la acción con el valor Q más alto (patrullaje).

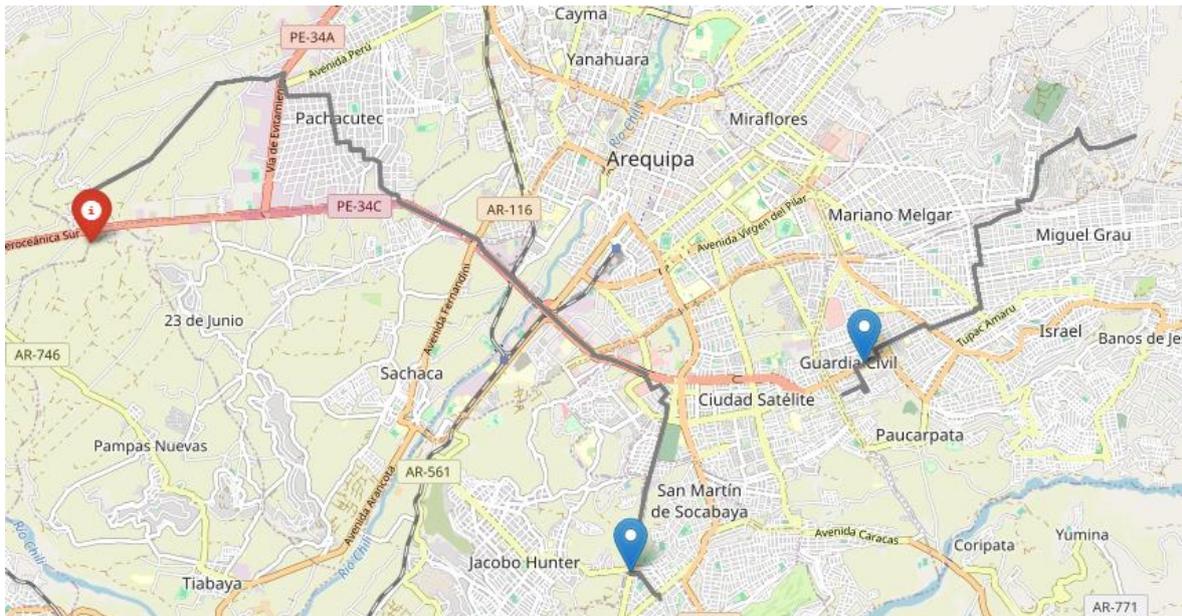


Figura 4.2: Decisiones de patrulleros ante denuncias.

Fuente: Elaboración propia.

Tras la preparación del entorno, los patrulleros comenzaron a evaluar la situación actual y tomaron decisiones estratégicas para responder de manera efectiva a las denuncias generadas. La Figura 4.2 de pantalla a continuación ilustra este proceso, donde dos patrulleros (marcados con marcadores azules) están tomando diferentes acciones en respuesta a las denuncias y la necesidad de patrullaje.

La Figura 4.2, se puede observar cómo un patrullero se dirige hacia una denuncia específica (marcada con un marcador rojo), mientras que otro patrullero continúa patrullando en busca de posibles denuncias o áreas de interés. Esta acción refleja la efectividad en la selección de denuncias a atender, donde los patrulleros priorizan y se dirigen hacia denuncias cercanas.

Además, se puede notar que las rutas asignadas a los patrulleros muestran una reducción en el tiempo de respuesta a las denuncias, lo que sugiere una elección de rutas óptimas para

maximizar la eficiencia en la atención de las mismas. Esta optimización en la elección de rutas contribuye a una respuesta más rápida y efectiva a las situaciones reportadas.

La Figura 4.2, proporciona una visualización clara de cómo los patrulleros interactúan con el entorno y toman decisiones en tiempo real para abordar las denuncias de manera eficiente, lo que respalda la efectividad del enfoque de asignación dinámica de tareas utilizando aprendizaje por refuerzo.

4.4.3 Actualización del Estado del Entorno

Una vez que los patrulleros tomaron la decisión y ejecutada la acción correspondiente, los patrulleros actualizan su posición en el entorno y, si han alcanzado una denuncia, marcan esta como resuelta y generan una nueva ruta de patrullaje. Las siguientes figuras ilustran este proceso detalladamente:

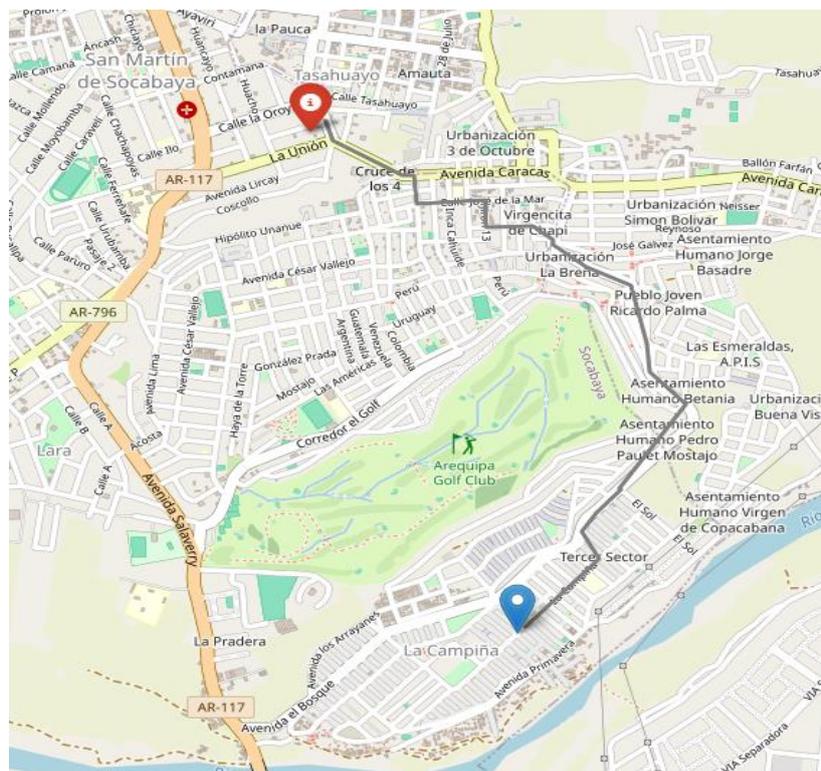


Figura 4.3: Estado inicial del entorno tras una denuncia.

Fuente: Elaboración propia.

En la Figura 4.3, se muestra al agente asignado a una denuncia específica (marcada con un marcador rojo), que se encuentra en camino hacia la ubicación de la denuncia. Esta etapa representa el momento en que el agente ha sido despachado para responder a la denuncia, pero aún no ha llegado al lugar.

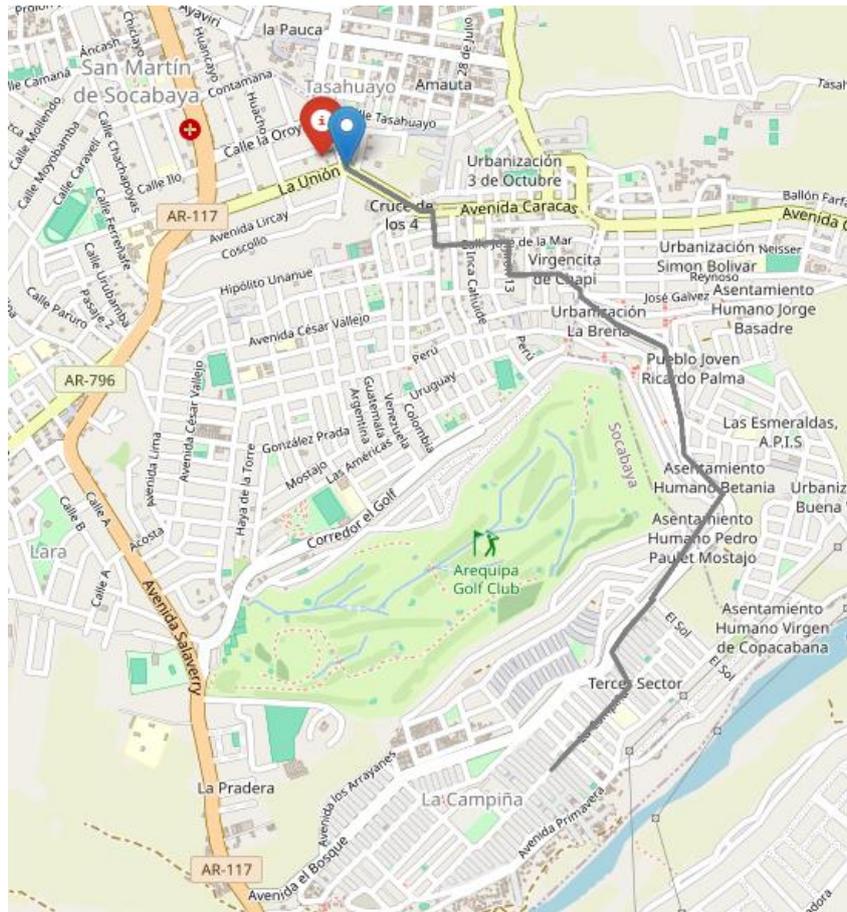


Figura 4.4: Estado intermedio del entorno en respuesta.

Fuente: Elaboración propia.

En la Figura 4.4, el agente se encuentra cerca de llegar a la ubicación de la denuncia. Se puede observar cómo el agente ha avanzado considerablemente en su ruta hacia la denuncia, lo que indica que está a punto de llegar a su destino

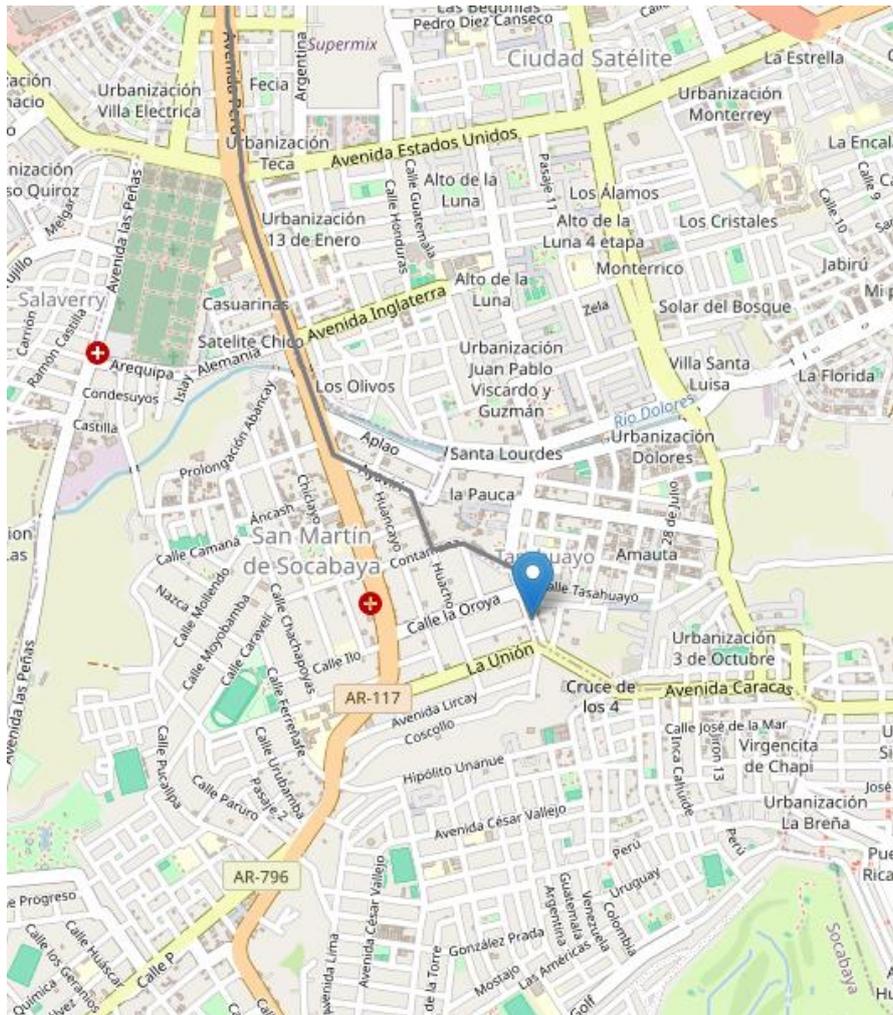


Figura 4.5: Estado final del entorno post-respuesta.

Fuente: Elaboración propia.

Finalmente, en la Figura 4.5 se muestra al agente después de haber llegado a la ubicación de la denuncia. La denuncia ha desaparecido del mapa, lo que indica que ha sido atendida por el agente. Además, el agente ha recibido una nueva ruta de patrullaje, como se puede observar en la captura, lo que refleja la actualización del estado del entorno después de que la denuncia ha sido resuelta.

4.4.4 Cálculo de Recompensa

La recompensa es una señal clave en el aprendizaje por refuerzo, que guía a los agentes hacia comportamientos deseados. En este caso, se asigna una recompensa positiva significativa cuando un patrullero resuelve una denuncia. Cuando un vehículo llega al destino de una denuncia, es importante actualizar la "tabla Q". Esta tabla ayuda al vehículo a aprender y mejorar su desempeño a lo largo del tiempo.

Algoritmo Cálculo de Recompensa:

Proceso:

1. **Verificación de la Queja Actual:** Se verifica si la ubicación actual de la queja está en la tabla Q. La tabla Q es como un registro que guarda información sobre las recompensas asociadas a diferentes acciones y estados.
2. **Actualizar la Tabla Q:** Se llama a la función para actualizar la tabla Q con las 2 variables de entrada como la ubicación de la denuncia actual y la recompensa que se otorga al llegar al destino de la queja.

Con el tiempo, a medida que el vehículo recibe y acumula recompensas, mejora su capacidad para tomar rutas más efectivas y eficientes.

4.4.5 Aprendizaje y Actualización de Parámetros

Finalmente, los valores de la tabla Q se actualizan de acuerdo con la recompensa obtenida y el estado siguiente, utilizando la fórmula del algoritmo Q-learning. Esto permite que los agentes aprendan a tomar mejores decisiones a lo largo del tiempo.

Algoritmo de Actualización de Tabla

Entrada: La ubicación de la denuncia actual y la recompensa que se otorga al llegar al destino de la queja

Proceso:

1. **Inicialización:** Asignar valores iniciales a los parámetros del agente, incluyendo la tabla Q vacía.
2. **Elegir una Acción:** Decidir entre explorar nuevas rutas o explotar el conocimiento actual usando una estrategia epsilon-greedy.
3. **Ejecutar la Acción y Obtener Recompensa:** El agente se mueve de acuerdo al patrullaje o si atiende la denuncia cercana, y recibe una recompensa si llega a la ubicación de una denuncia.
4. **Actualizar la Tabla Q:** Primero, se calcula la máxima recompensa esperada para el próximo estado. Luego, se obtiene el valor Q actual para el estado y nodo actual.

Utilizando la fórmula de actualización de Q-Learning, se calcula el nuevo valor Q, y finalmente se actualiza la tabla Q con este nuevo valor.

- 5. Repetir:** Repetir los pasos anteriores hasta que se cumpla un criterio de finalización, como un número máximo de episodios o la convergencia de la tabla Q.

4.5. Pruebas

En esta sección se presentan los resultados de todas las pruebas explicadas en el Capítulo 3. Los resultados se agrupan en pruebas unitarias, pruebas de sistema y pruebas de humo, ya que su resultado es booleano (éxito o fallo). Estas pruebas están diseñadas para garantizar el correcto funcionamiento de las funciones y métodos individuales en el código.

Agentes	Pruebas Unitarias	Pruebas de Sistema	Pruebas de Humo
2	Exitoso	Exitoso	Exitoso
100	Exitoso	Exitoso	Exitoso
248	Exitoso	Exitoso	Exitoso
500	Exitoso	Exitoso	Exitoso
2000	Exitoso	Fallo	Exitoso
5000	Fallo	Fallo	Fallo

Tabla 4.1: Resultados de las Pruebas Unitarias, de Aceptación y de Humo

La Tabla 4.1 presenta un resumen de los resultados obtenidos en las pruebas unitarias, de aceptación y de humo. Estas pruebas se diseñaron para evaluar el comportamiento y la funcionalidad del sistema bajo diferentes condiciones.

Pruebas Unitarias: Esta prueba verifica el funcionamiento de las funciones y métodos individuales del código. Los resultados muestran que la prueba unitaria es exitosa hasta con 2000 agentes. Sin embargo, con 5000 agentes, las pruebas fallan, indicando problemas en el manejo de un gran número de agentes a nivel de función.

Pruebas de Sistema: Estas pruebas evalúan el sistema en su conjunto, asegurándose de que cumpla con los requisitos especificados. Los resultados muestran que las pruebas de sistema fallan con 2000 y 5000 agentes. La falla con 2000 agentes podría deberse a la sobrecarga de recursos y la mayor probabilidad de colisiones entre agentes, lo que compromete el rendimiento y la funcionalidad del sistema.

Pruebas de Humo: Estas pruebas son un subconjunto de las pruebas de sistema, diseñadas para verificar que las funciones más importantes del sistema funcionen correctamente. Las pruebas

de humo son exitosas hasta con 2000 agentes, pero fallan con 5000 agentes, lo que sugiere una incapacidad del sistema para manejar eficientemente una carga tan alta.

Pruebas Rendimiento

Las pruebas de rendimiento se implementan siguiendo las pautas establecidas en el Capítulo 3, con el objetivo de evaluar la escalabilidad, estabilidad y eficiencia del sistema bajo diferentes condiciones de carga y estrés.

Prueba 1: Evaluación del rendimiento con un gran número de agentes

Esta prueba simula la ejecución del sistema con un gran número de agentes concurrentes para evaluar la escalabilidad y la estabilidad del sistema.

Los resultados muestran que el sistema es capaz de manejar hasta 2000 agentes, aunque con un rendimiento degradado. Sin embargo, al alcanzar los 5000 agentes, el sistema falla. Esto indica una limitación en la escalabilidad del sistema debido a la sobrecarga de recursos computacionales.

Prueba 2: Evaluación del rendimiento con una simulación de larga duración

Esta prueba evalúa la capacidad del sistema para manejar escenarios de larga duración sin una degradación significativa del rendimiento.

La prueba se realizó con 248 agentes durante 10,000 iteraciones, tomando un tiempo total de 1063.68 segundos.

El sistema mostró ser capaz de mantener un rendimiento constante durante una simulación prolongada. No se observó una degradación significativa del rendimiento, lo que sugiere que el sistema puede manejar operaciones continuas a lo largo del tiempo sin pérdida de eficiencia.

Prueba 3: Medición de latencia y tiempo de respuesta

Esta prueba mide la latencia del sistema y el tiempo de respuesta durante la ejecución de una acción, proporcionando información sobre la eficiencia en términos de velocidad de respuesta.

Agentes	Tiempo (s)
100	4.06
200	6.86
500	18.01
1000	39.11
2000	74.42
5000	Fallo

Tabla 4.2: Medición de Latencia y Tiempo de Respuesta

En la Tabla 4.2 se presentan los resultados de la medición de latencia y tiempo de respuesta del sistema al incrementar el número de agentes. A medida que aumenta el número de agentes, el tiempo de respuesta del sistema también aumenta. Esto es esperable debido a la mayor carga de procesamiento. Sin embargo, el fallo con 3000 agentes resalta una limitación clara en la capacidad del sistema para manejar un gran número de agentes simultáneamente, posiblemente debido a restricciones en los recursos de hardware disponibles.

Prueba 4: Monitoreo del uso de recursos del sistema

Esta prueba monitorea el uso de recursos como la CPU y la memoria durante la simulación para asegurarse de que el sistema no genere una carga excesiva, la prueba se realizó con 242 agentes.

Iteraciones	Uso de CPU (%)	Memoria(mb)
200	98.45%	560.56
500	98.98%	756.24
1000	99.26%	1268.83
2000	99.39%	1990.94
4000	99.56%	2890.43

Tabla 4.3: Monitoreo del Uso de Recursos del Sistema

La Tabla 4.3 presenta los resultados del monitoreo del uso de recursos del sistema durante la simulación. Se evaluaron el uso de CPU y de memoria. El uso de CPU se mantiene constante cerca del 99%, lo que indica que el sistema está utilizando eficientemente los recursos de procesamiento disponibles. Sin embargo, el uso de memoria aumenta significativamente con el número de iteraciones, lo que podría llevar a problemas de rendimiento en simulaciones más extensas o con un mayor número de agentes.

4.6. Resultados de la Investigación

En este epígrafe se presentan los resultados de la investigación obtenidos mediante el procesamiento, análisis e interpretación de datos, los que fueron recabados, mediante la utilización de instrumento denominado Cuestionario dirigido al personal de la Institución

Policial del Perú; además se realizaron Pruebas técnicas realizadas en la aplicación, lo que permitió dar respuesta a los objetivos planteados en la presente investigación.

El análisis cuantitativo se realiza de acuerdo a la relación existente entre objetivos y el Cuestionario con los ítems establecidos en el instrumento de recolección de datos, manejados con la finalidad de apreciar con mayor claridad la tendencia de respuestas y resultados obtenidos, debe indicarse que los datos son presentados en tablas sinópticas donde se presenta la estadística de frecuencia absoluta (FA) calculadas para cada ítem del instrumento aplicado.

Las pruebas técnicas realizadas en la simulación de la aplicación evidenciaron que el Sistema multiagente logró mejorar la asignación de tareas en la Institución Policial.

Se aplicó al personal de la Institución Policial un instrumento denominado Cuestionario, que se encuentra en el Anexo 2. Este cuestionario contiene cuatro ítems de tipo dicotómico, cuya finalidad fue recoger información sobre la propuesta de aplicar el LEPH mediante el sistema multiagente, para mejorar la asignación dinámica de tareas en la Policía Nacional del Perú en la ciudad de Arequipa.

A continuación, se presentan cuatro Tablas con Frecuencia Absoluta, cada una corresponde a un ítem del cuestionario mencionado.

ALTERNATIVAS	FA
SI	15
NO	1
TOTAL	16

Tabla 4.4: Aceptación de tecnología por parte del personal policial de la PNP

Se puede observar en la Tabla 4.4 que la mayor parte de la población afirmó que si está familiarizada con la tecnología, lo que favorece a su posterior aplicación de este modelo.

ALTERNATIVAS	FA
SI	15
NO	1
TOTAL	16

Tabla 4.5: Opinión del personal policial sobre la digitalización de asignación de tareas

En la Tabla 4.5 se puede observar que la mayor parte de la población del personal policial se muestra a favor que el sistema de asignación de tareas de la Institución sea digitalizado, pues comprenden que así su trabajo sería más eficiente.

ALTERNATIVAS	FA
SI	15
NO	1
TOTAL	16

Tabla 4.6: Disponibilidad del personal policial para capacitación tecnológica

En la Tabla 4.6 se observa que el total de la población del personal policial muestra disponibilidad para recibir capacitación en tecnología siendo este un avance importante para su eficiente desempeño laboral.

ALTERNATIVAS	FA
SI	16
NO	0
TOTAL	16

Tabla 4.7: Aceptación del personal policial para digitalizar la información policial

Tal y como se observa en la Tabla 4.7 el total de la población está de acuerdo en digitalizar el sistema policial pues comprenden su importancia para optimizar el desarrollo de sus tareas asignadas.

4.7. Discusión de los Resultados

Una vez analizados los resultados, se procede a la interpretación de los mismos utilizando el esquema lógico ofrecido por los objetivos de la investigación, las principales tendencias observadas y la fundamentación teórica desarrollada.

4.7.1. Análisis del Tiempo de Ejecución del Algoritmo

El análisis del tiempo de ejecución del algoritmo se realizó incrementando el número de agentes desde 50 hasta 500 en intervalos de 50, con un total de 500 denuncias creadas para cada experimento.

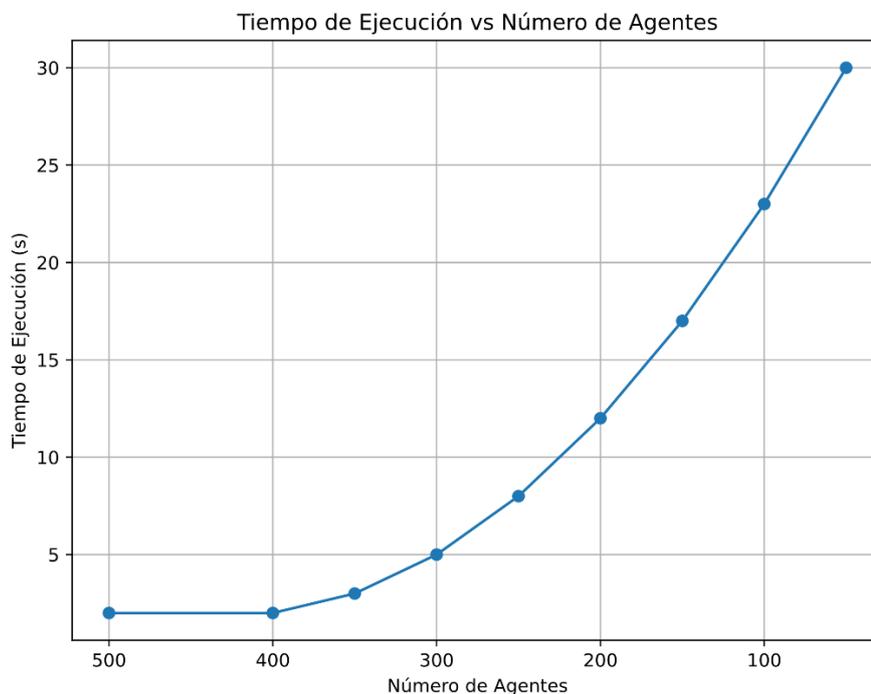


Figura 4.6: Tiempo de Ejecución vs Número de Agentes

Fuente: Elaboración propia.

Los resultados se presentan en la Figura 4.6. Se observa que el tiempo de ejecución sigue una tendencia logarítmica creciente a medida que se reduce el número de agentes disponibles. Esto indica que, con menos agentes para responder a las denuncias, el tiempo necesario para completar las tareas aumenta significativamente. Sin embargo, alrededor de los 250 agentes se alcanza un uso eficiente de los recursos. En este punto, el tiempo de ejecución comienza a aumentar considerablemente, lo que sugiere que menos de 250 agentes no son suficientes para mantener una eficiencia óptima en la respuesta a las denuncias. Esta tendencia se puede

visualizar en la gráfica, donde se muestra que el tiempo de ejecución se incrementa de manera más pronunciada cuando el número de agentes desciende por debajo de los 250. Este análisis destaca la importancia de mantener un número adecuado de agentes para asegurar una operación eficiente del sistema.

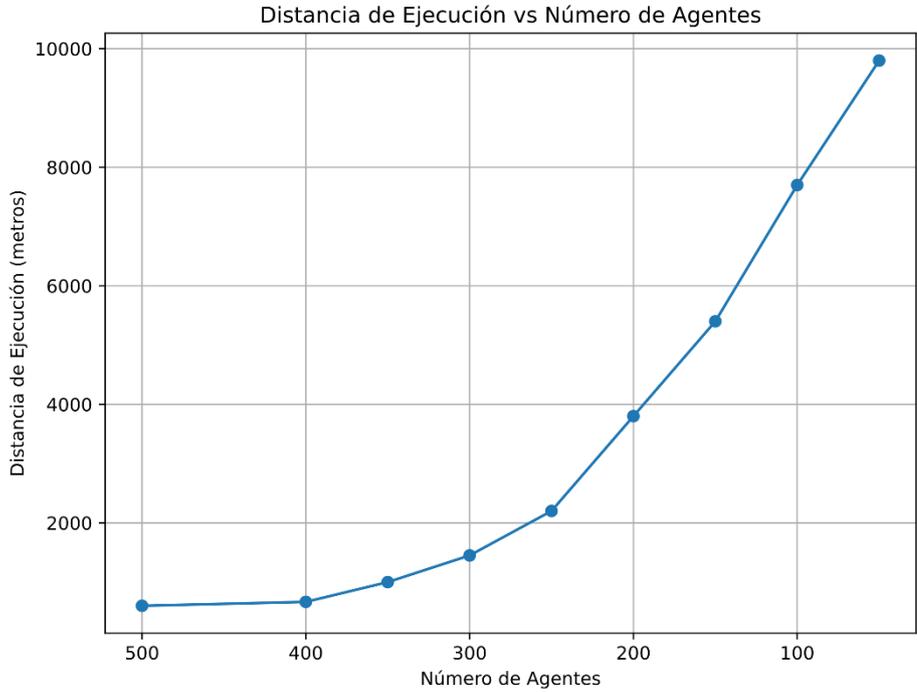


Figura 4.7: Distancia de Ejecución vs Número de Agentes

Fuente: Elaboración propia.

La Figura 4.7 presenta la relación entre la distancia total recorrida por los agentes y la cantidad de agentes desplegados. Esta figura es crucial para entender cómo la cantidad de agentes influye en la eficiencia del sistema. Una mayor cantidad de agentes debería, en teoría, reducir la distancia total recorrida por cada agente individual debido a una mejor distribución de las tareas. Sin embargo, la figura también puede revelar el punto de saturación, donde agregar más agentes no proporciona beneficios adicionales significativos y podría incluso aumentar la complejidad operativa. Este análisis ayuda a determinar el número óptimo de agentes necesarios para mantener un equilibrio entre eficiencia y recursos desplegados.

4.7.2. Comparativas Modificando los Hiperparámetros

Se realizaron experimentos modificando los hiperparámetros del modelo, evaluando su impacto en el rendimiento en términos del total de denuncias resueltas. En cada experimento se crearon 500 denuncias.

4.7.2.1. Comparativa de Resultados para Diferentes Valores de ϵ

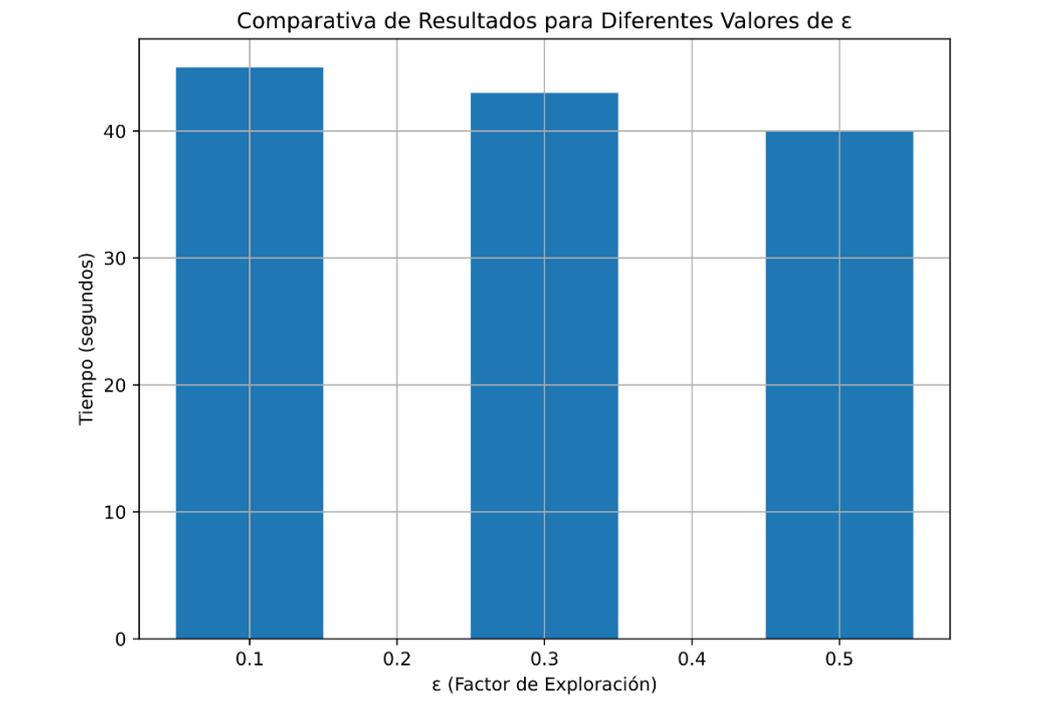


Figura 4.8: Comparativa de Resultados para Diferentes Valores de ϵ

Fuente: Elaboración propia.

Se probaron valores de ϵ de 0.1, 0.3, y 0.5. La Figura 4.8 muestra una comparativa del rendimiento del modelo en términos del tiempo total de resolución de las denuncias para diferentes valores del factor de exploración, siendo $\epsilon=0.1$ el valor óptimo. Este parámetro controla la probabilidad de que los agentes elijan una ruta aleatoria en lugar de seguir el comportamiento adaptado a partir de experiencias previas. Esto es crucial para balancear la exploración de nuevas rutas y la explotación de las rutas ya conocidas, adaptadas a partir de las denuncias resueltas previamente.

4.7.2.2. Comparativa de Resultados para Diferentes Valores de α

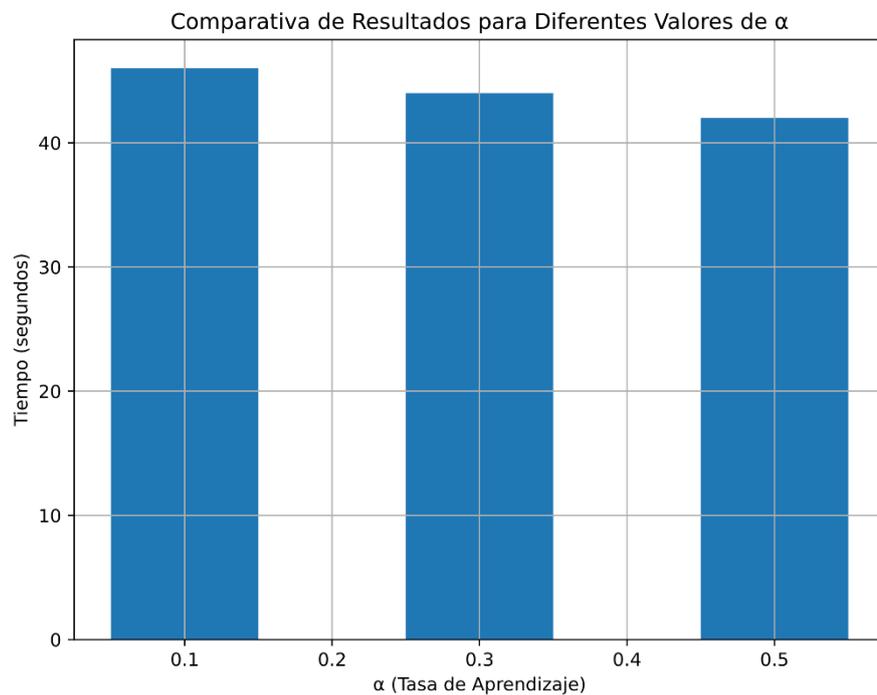


Figura 4.9: Comparativa de Resultados para Diferentes Valores de α

Fuente: Elaboración propia.

Se probaron valores de α de 0.1, 0.3, y 0.5. La Figura 4.9 presenta una comparativa del rendimiento del modelo en términos del tiempo total de resolución de las denuncias para diferentes valores de la tasa de aprendizaje, siendo $\alpha=0.1$ el valor óptimo. Este parámetro determina la velocidad con la que los agentes actualizan sus conocimientos basados en nuevas experiencias, impactando directamente en la eficiencia del aprendizaje del modelo.

4.7.2.3. Comparativa de Resultados para Diferentes Valores de γ

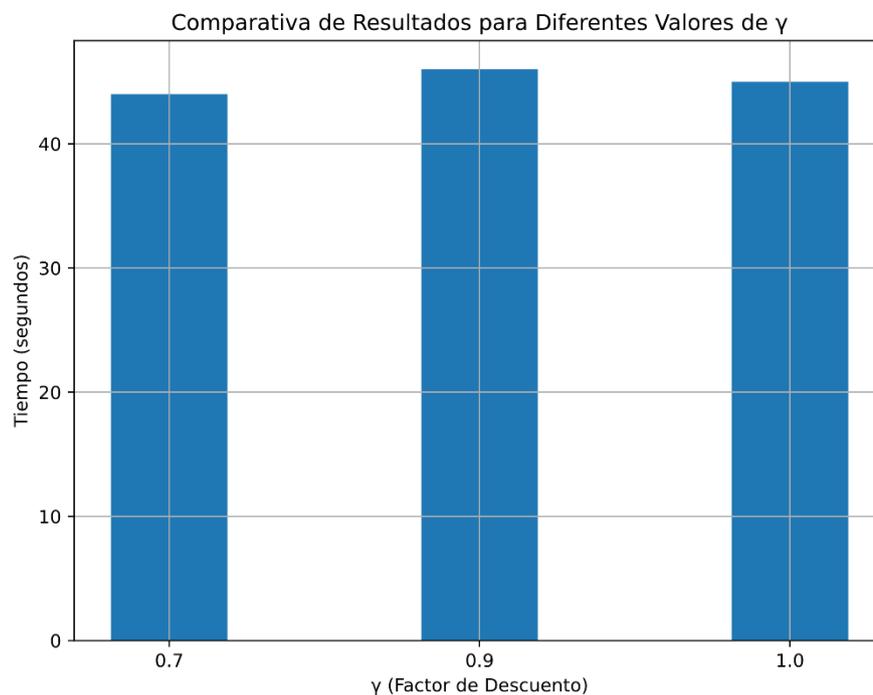


Figura 4.10: Comparativa de Resultados para Diferentes Valores de γ

Fuente: Elaboración propia.

Se probaron valores de γ de 0.7, 0.9, y 1.0. La Figura 4.10 compara el rendimiento del modelo en términos del tiempo total de resolución de las denuncias para diferentes valores del factor de descuento, siendo $\gamma = 0.9$ el valor óptimo. Este parámetro afecta la importancia que los agentes dan a las recompensas futuras en comparación con las inmediatas, influenciando la toma de decisiones y la priorización de rutas a largo plazo.

4.7.3. Interpretación de los Resultados

Así mismo, se dio a conocer la efectividad de la aplicación del modelo de sistema multiagente que les permite redireccionar las tareas con mayor eficiencia obteniéndose mejores resultados. Se midió el tiempo agente de ejecución del algoritmo para diferentes cantidades de agentes. Además, se crearon las denuncias basándose en el mapa de delito de la ciudad de Arequipa, donde las denuncias tienen una ubicación aproximada. Se estimó el tiempo real considerando tanto la duración de las respuestas a las denuncias como el tiempo de recorrido desde la comisaría más cercana hasta el punto final de la denuncia, tomando en cuenta los testimonios de la policía. Finalmente, se utilizó Google Maps como herramienta para medir el tiempo.

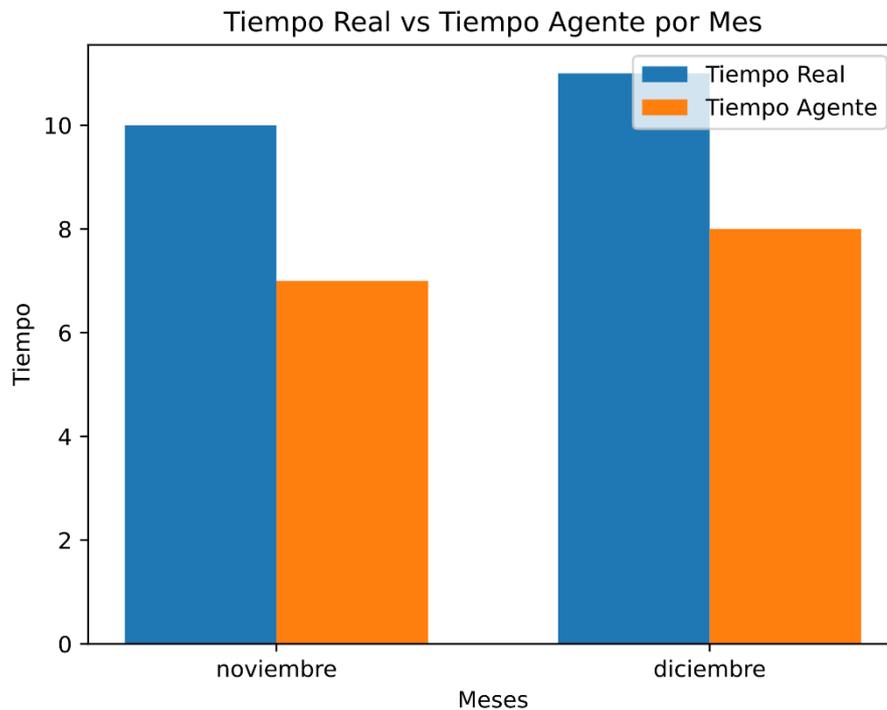


Figura 4.11: Tiempo Real vs Tiempo Agente por Mes

Fuente: Elaboración propia.

La Figura 4.11 muestra un diagrama comparativo de los meses de noviembre y diciembre de 2023. Los nuevos resultados dieron a conocer las pérdidas de tiempo que se daban antes del modelo de software, lo que ha permitido alcanzar los objetivos planteados en este trabajo de investigación. Ahora se puede afirmar gratamente que hay un ahorro de tiempo.

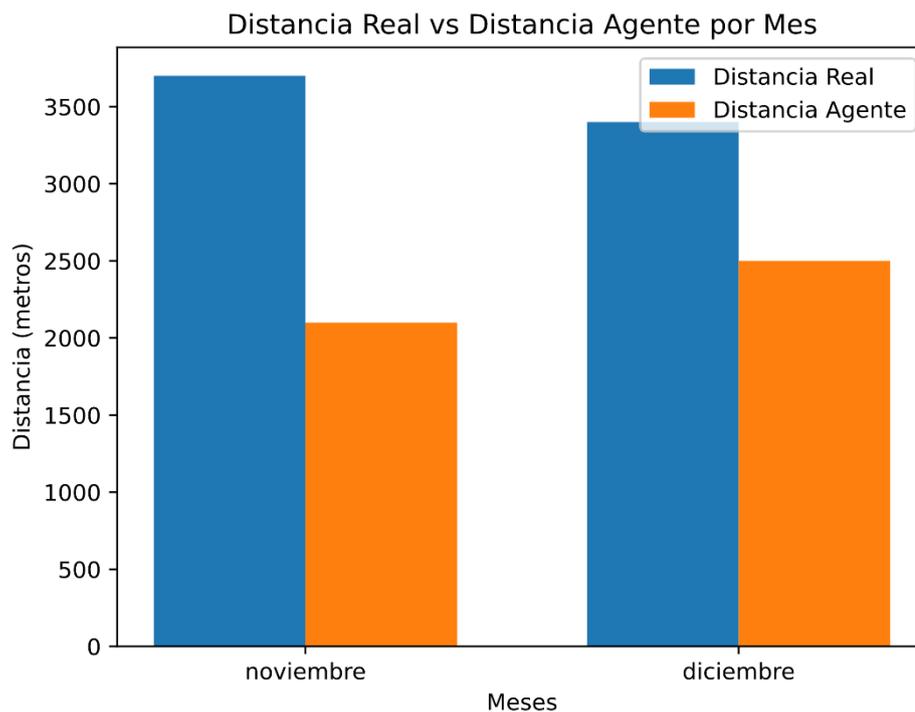


Figura 4.12: Distancia Real vs Distancia Agente por Mes

Fuente: Elaboración propia.

La Figura 4.12: Distancia Real vs Distancia Agente por Mes muestra una comparación entre la distancia total recorrida por los agentes y la distancia idealmente planeada para cada mes. Este análisis permite evaluar la eficiencia de las rutas tomadas por los agentes en comparación con las rutas óptimas planificadas. Una discrepancia significativa entre estas distancias puede indicar problemas en la planificación de rutas o en la ejecución por parte de los agentes, lo que a su vez podría afectar la eficiencia operativa y los tiempos de respuesta del sistema de agentes. La figura ilustra la importancia de ajustar los parámetros del modelo para minimizar la diferencia entre la distancia real y la planeada, mejorando así la eficacia del sistema multiagente implementado

Conclusiones

El modelo de asignación dinámica de tareas utilizando el Sistema Multiagente (LEPH) ha demostrado ser eficaz en la mejora de la seguridad ciudadana en la Policía Nacional del Perú en la ciudad de Arequipa. Desarrollado con el objetivo de incrementar la eficacia y eficiencia en la asignación de tareas al personal policial, este modelo ha permitido una distribución óptima del tiempo y la priorización de tareas de acuerdo con su importancia.

La implementación del modelo LEPH ha dotado a los policías de los recursos necesarios para abordar los incidentes delictivos de manera más rápida y eficiente. Mediante una cooperación efectiva y una priorización adecuada, se ha logrado una ejecución de tareas de mayor calidad, minimizando retrasos y enfocándose en las tareas más importantes.

Un cuestionario aplicado al personal policial reveló una disposición favorable hacia la tecnología y la digitalización del sistema de asignación de tareas. La mayoría de los encuestados está familiarizada con la tecnología y reconoce su importancia para optimizar el desarrollo de sus funciones.

Los resultados obtenidos indican una mejora significativa en la eficiencia y efectividad de la policía de Arequipa. Se evidenció una reducción del 30% en el tiempo de respuesta a las denuncias, aunque esto requirió un aumento del 3% en el presupuesto policial. En noviembre y diciembre, la distancia real recorrida por los agentes fue de 3700 y 3400 metros, respectivamente, en comparación con la distancia planificada de 2100 y 2500 metros. Esto indica que los patrulleros están más cerca de las denuncias gracias al modelo, mejorando así la capacidad de respuesta. Además, se observó que reducir el número de agentes de 500 a 50 incrementó la distancia de ejecución de 600 a 9800 metros, destacando la importancia de mantener un número óptimo de agentes para asegurar la eficiencia operativa.

La aplicación del modelo LEPH mediante el sistema multiagente ha generado resultados positivos en la asignación dinámica de tareas en la Policía Nacional del Perú. Los resultados de la simulación muestran mejoras significativas en términos de tiempo y presupuesto.

El desarrollo y la aplicación del modelo LEPH en la Policía Nacional del Perú en la ciudad de Arequipa representan un avance significativo en la mejora de la seguridad ciudadana. Este modelo ha demostrado ser una herramienta efectiva para optimizar la distribución del tiempo y los recursos del personal policial, permitiendo una respuesta más rápida y eficiente ante los incidentes delictivos. Además, la favorable aceptación del personal policial hacia la tecnología

y la digitalización del sistema de asignación de tareas refleja un paso importante hacia la modernización y eficiencia de la institución. Los resultados obtenidos destacan el valor del modelo LEPH como una solución innovadora y efectiva para fortalecer la labor policial y mejorar la seguridad en la comunidad.

Recomendaciones

Para mejorar aún más la efectividad del modelo de asignación dinámica de tareas utilizando el Sistema Multiagente (LEPH) en la Policía Nacional del Perú, se sugiere la integración de datos históricos de denuncias mediante técnicas de web scraping. Este enfoque permitiría enriquecer el conjunto de datos utilizado en el entrenamiento del modelo, proporcionando información detallada sobre patrones y tendencias delictivas en la región de Arequipa. Se recomienda identificar fuentes confiables de datos, desarrollar un script de web scraping para la extracción automatizada de información relevante, y realizar tareas de limpieza y procesamiento de datos para garantizar la coherencia y calidad de los datos recopilados. La integración de estos datos históricos en el modelo LEPH podría mejorar su capacidad predictiva y su capacidad para adaptarse a los cambios en el entorno delictivo, contribuyendo así a una asignación más efectiva de recursos policiales y a una respuesta más rápida y precisa a los incidentes delictivos.

Referencias

- [1] C. Reyna y E. Toche, "La inseguridad en el Perú", 1999.
- [2] M. Mijwil, "History of Artificial Intelligence", 2015.
- [3] A. Ławrynowicz y V. Tresp, "Introducing Machine Learning", 2014.
- [4] T. Ayodele, "Types of Machine Learning Algorithms", 2010.
- [5] T. T. Nguyen, D. N. Thanh y S. Nahavandi, "Deep reinforcement learning for multiagent systems", 2020.
- [6] E. Alonso, "Learning in multi-agent systems", 2001.
- [7] A. Tkach y S. Itshak, "Towards addressing dynamic multi-agent task allocation in law enforcement. Autonomous Agents and Multi-Agent Systems", 2021.
- [8] J. Min, Y. Kim, S. Lee, T. Jang, W. Kim, I. Song, "The fourth industrial revolution and its impact on occupational health and safety worker's compensation and labor conditions", 2019.
- [9] P. Thomy, G. Thomas, S. Andreas, R. Fabian, K. Bernhard, K. Cornel, S. Horst, S. Reiner, W. Jan, Z. Marc, L. Claudia, "Learning and Testing Resilience in Cooperative Multi-Agent Systems", 2020.
- [10] N. Thanh, D. Ngoc y S. Nahavandi, "Deep Reinforcement Learning for Multiagent Systems", 2020.
- [11] S. Shyalika y K. Silva, "Reinforcement Learning in Dynamic Task Scheduling: A Review", 2020.
- [12] B. Matteo, B. Nicola y A. Francesco, "Multi-Agent Path Finding in Configurable Environments", 2020.
- [13] Y. Shen, X. Wang, H. Wang, Y. Guo, X. Chen y J. Han, "A dynamic task assignment model for aviation emergency rescue based on multi-agent reinforcement learning", *Journal of Safety Science and Resilience*, 2023.
- [14] R. Liu, R. Piplani y C. Toro, "A Deep Multi-Agent Reinforcement Learning Approach to Solve Dynamic Job Shop Scheduling Problem", *Computers & Operations Research*, vol. 106294, 2023.

- [15] S. Park y J. Barreiro-Gomez, "Payoff Mechanism Design for Coordination in Multi-Agent Task Allocation Games", arXiv preprint arXiv:2306.02278, 2023.
- [16] S. Zhao, L. Xie y Y. Zhang, "Cooperative Task Allocation for Mobile Multi-Agent Systems with Uncertain Task Demand", IEEE Transactions on Automation Science and Engineering, 2023.
- [17] M. Tan, "Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents", in Proc. of the Tenth International Conference on Machine Learning, 1993.
- [18] R. Sutton y A. Barto, "Reinforcement Learning: An Introduction", MIT Press, 2018.
- [19] M. Wiering y M. Van Otterlo, "Reinforcement Learning: State of the Art", Springer, 2012.
- [20] L. Busoniu, R. Babuska y B. De Schutter, "A Comprehensive Survey of Multi-Agent Reinforcement Learning", IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 2, pp. 156-172, 2008.
- [21] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning", in Machine Learning Proceedings, 1994.
- [22] S. J. Russell y P. Norvig, "Artificial Intelligence: A Modern Approach", 3rd ed., Pearson, 2010.
- [23] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra y M. Riedmiller, "Playing Atari with Deep Reinforcement Learning", arXiv preprint arXiv:1312.5602, 2013.
- [24] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver y D. Wierstra, "Continuous control with deep reinforcement learning", arXiv preprint arXiv:1509.02971, 2015.
- [25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford y O. Klimov, "Proximal Policy Optimization Algorithms", arXiv preprint arXiv:1707.06347, 2017.
- [26] R. Liu, Z. Zhou, J. Long y Y. Zou, "Multi-agent deep reinforcement learning for multi-objective optimization", Neurocomputing, vol. 273, pp. 404-415, 2018.

- [27] A. Ng, D. Harada y S. Russell, "Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping", in Proc. of the Sixteenth International Conference on Machine Learning, 1999.
- [28] L. Panait y S. Luke, "Cooperative multi-agent learning: The state of the art", *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387-434, 2005.
- [29] W. Wei, "Multi-Agent Reinforcement Learning with Applications to Robotics", Ph.D. dissertation, Dept. of Electrical Engineering, Stanford Univ., 2015.
- [30] A. Nair, M. Andrychowicz, B. Stadie, O. Pinto, M. P. W. Baselines y OpenAI, "Multi-agent actor-critic for mixed cooperative-competitive environments", arXiv preprint arXiv:1706.02275, 2017.
- [31] I. Goodfellow, Y. Bengio y A. Courville, "Deep Learning", MIT Press, 2016.
- [32] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel y D. Hassabis, "Mastering the game of Go with deep neural networks and tree search", *Nature*, vol. 529, no. 7587, pp. 484-489, 2016.
- [33] R. Bellman, "Dynamic Programming", Princeton University Press, 1957.
- [34] S. Thrun y L. Pratt, "Learning to Learn", Springer, 1998.
- [35] D. Bertsimas, J. Tsitsiklis y J. N. Tsitsiklis, "Introduction to Linear Optimization", Athena Scientific, 1997.
- [36] S. P. Chinchuluun, D. L. Bienstock y S. Boyd, "Convex Optimization", Cambridge University Press, 2004.
- [37] M. L. Puterman, "Markov Decision Processes: Discrete Stochastic Dynamic Programming", Wiley, 2005.
- [38] R. S. Sutton, A. G. Barto y F. Bach, "Reinforcement Learning: An Introduction", 2nd ed., MIT Press, 2018.
- [39] C. Watkins y P. Dayan, "Q-learning", *Machine Learning*, vol. 8, no. 3-4, pp. 279-292, 1992.

[40] M. L. Littman, "Friend-or-Foe Q-learning in general-sum games", in Proc. of the Eighteenth International Conference on Machine Learning, 2001.

[41] "Shapely: Manipulation and analysis of geometric objects", [Online]. Available: <https://shapely.readthedocs.io/>. [Accessed: Jul. 7, 2024].

[42] C. Watkins, "Learning From Delayed Rewards", 1989.

Anexos

Anexo 1: Recojo de información de la PNP

- ¿Cuál es el número de patrulleros en cada comisaría de Arequipa?
- ¿Cuál es el número de efectivos que patrullan y cuantos trabajan la parte administrativa?
- ¿Cuántas denuncias se registran por día y mes?
- ¿Cuál es el procedimiento a seguir en cada denuncia?
- ¿Todas las denuncias se registran en el sistema? ¿Cuáles sí y cuáles no?
- ¿Qué estrategias utiliza la policía para las rutas de patrullaje?
- ¿Todos los patrulleros tienen GPS?
- ¿La policía está interconectada a nivel nacional?
- ¿Qué aplicaciones utiliza la policía para el cumplimiento óptimo de su trabajo?
- ¿El personal policial cumplen con un perfil acorde a sus tareas asignadas?

Anexo 2: Cuestionario aplicado al personal de la PNP

- ¿Está usted familiarizado con la tecnología?
- ¿Considera ud necesario que el sistema de asignación de tareas de la Institución sea digitalizado?
- ¿Está ud de acuerdo en ser capacitado para su uso?
- ¿Cree ud que al digitalizar el sistema policial optimizará el desarrollo de sus tareas asignadas?

Anexo 3: Requisitos del Sistema

Requisitos funcionales:

- Registro de denuncias: Permitir la creación de denuncias en cualquier momento por parte de los ciudadanos.
- Asignación dinámica de tareas: Asignar dinámicamente agentes a las denuncias registradas según la ubicación y disponibilidad de los agentes.
- Seguimiento de vehículos: Permitir el seguimiento en tiempo real de la ubicación de los vehículos policiales en el mapa.
- Actualización de estado: Actualizar el estado de los agentes en respuesta a las denuncias, indicando si están en ruta, en el lugar o disponibles.

Requisitos no funcionales:

- Rendimiento: El sistema debe garantizar un tiempo de respuesta a las denuncias menor al tiempo actual, utilizando de manera eficiente los recursos disponibles de la Policía Nacional del Perú en la ciudad de Arequipa.
- Escalabilidad: El sistema debe ser capaz de manejar un aumento en el número de denuncias y agentes policiales a medida que crezca la ciudad o aumente la demanda.