

# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

## **Facultad de Ingeniería en Electricidad y Computación**

Despliegue dinámico de drones en situaciones críticas utilizando Machine Learning

INGE-2743

### **PROYECTO INTEGRADOR**

Previo a la obtención del Título de:  
**Ingeniero en Telecomunicaciones**

Presentado por:

Carlos Fernando Arroyo Barrera

Boris Josue Piyasagua Avila

Guayaquil – Ecuador

Año: 2024

## Dedicatoria

---

A mis padres, **Carlos y Narcisa**, por ser mi pilar inquebrantable, por su amor, sacrificio y apoyo incondicional en cada paso de mi vida. Su esfuerzo y dedicación han sido la inspiración para alcanzar cada meta.

A mi compañero de tesis y gran amigo, **Boris**, por su compromiso, paciencia y colaboración en este desafío. Compartir este camino contigo ha sido un aprendizaje invaluable y una experiencia que siempre recordaré con gratitud.

A mis amistades de la universidad, **Maritza y Daleska**, por ser compañeras de lucha en este viaje académico, por su apoyo, consejos y por los momentos de alegría que hicieron más llevadero este proceso.

A mi hermana, **Valeria**, por ser un motor constante de motivación, por su cariño y por recordarme siempre la importancia de seguir adelante con determinación y entusiasmo.

A mis **tíos**, por su apoyo silencioso, por sus palabras de aliento y por ser parte de mi vida en todo momento. Su confianza en mí ha sido un gran impulso para llegar hasta aquí.

A todos ustedes, les dedico este logro con todo mi cariño y gratitud.

## Dedicatoria

---

A mi madre, **Dennis**, quien siempre ha estado apoyándome y esforzándose para que yo salga adelante. A mi padre, **José**, quien me ha enseñado a no rendirme y que, gracias a su sacrificio, soy ese profesional que alguna vez anhelaron tener.

A mis hermanas, **Suanny** y **Danna**, por compartir los momentos de risa y peleas de hermanos, que este logro lo tomen como una motivación y superación.

A mis primos, **Gregory**, **Matheus**, **Ariel**, **Anthony**, por compartir momentos buenos de la infancia y apoyarme en todo lo que he logrado.

A mi mejor amigo, **Jonathan**, por ser como un hermano y brindarme su amistad por varios años, por escucharme y motivarme a cumplir mis sueños.

A **Carlitos**, quien no solo ha sido mi compañero durante toda la carrera, sino el SÍ A TODO: reuniones, viajes, salidas, y quien me ha apoyado tanto en la parte académica como personal.

A **Lilly**, quien ha estado a mi lado brindándome su apoyo, su amistad y motivándome a lograr lo que me propongo.

Este mérito es para ustedes,

**Boris Josue Piyasagua Avila**

## Agradecimientos

---

Al culminar esta etapa, no puedo dejar de expresar mi más sincero agradecimiento a todas las personas que, de una u otra manera, han sido parte fundamental de este proceso. A mis padres, **Carlos** y **Narcisa**, por ser mi mayor fuente de apoyo y motivación. Su esfuerzo, sacrificio y amor incondicional han sido la base sobre la que he construido mis logros.

A mi compañero de tesis y amigo, **Boris**, por su dedicación, compromiso y compañerismo. Trabajar juntos ha sido un reto y un aprendizaje constante, y no podría haber pedido un mejor aliado para este proyecto.

A mis amigos de la universidad, **Maritza, Daleska, Steveen, Stefany y María**, por su apoyo, amistad y por ser parte de esta etapa llena de desafíos.

A **Dayanna**, una persona valiosa que sigo conservando y que ha sido una guía y un apoyo en mi crecimiento tanto personal como profesional. Su confianza, consejos y enseñanzas han marcado una diferencia significativa en mi vida, y siempre estaré agradecido por ello.

A todos ustedes, gracias por ser parte de mi historia y por acompañarme en este camino.

## Agradecimientos

---

Agradezco a mi tío, **John**, quien ha sido mi guía y soporte a lo largo de mi vida, así como una motivación para seguir adelante.

A mi tutora, **Patricia**, quien me ha orientado tanto en la parte académica como en la parte profesional/laboral.

A los padres de mi compañero **Carlos**, quienes me acogieron como un hijo dentro de su hogar.

A mis compañeros de carrera, **Maritza, Andrés, Steveen, Stefany, José, Darwin, Cristopher**, por el apoyo y compañerismo durante los años de universidad.

A mis compañeros de escuela, **Jorge, Karen y David**, quienes me han brindado una amistad increíble durante varios años.

A mi compañero de trabajo, **Widmany**, quien ha estado presente durante este tiempo, apoyándome para culminar mis metas.

Agradezco a todos los que me apoyaron y confiaron en mí.

Con mucho cariño,

**Boris Josue Piyasagua Avila**

## Declaración Expresa

---

Nosotros **Carlos Fernando Arroyo Barrera** y **Boris Josue Piyasagua Avila** acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor del proyecto corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí/nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que me/nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi/nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique a los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

---

Carlos Fernando Arroyo Barrera

---

Boris Josue Piyasagua Avila

## **Evaluadores**

---

**PhD. María Álvarez Villanueva**

Profesora de Materia

---

**PhD. Patricia Chávez Burbano**

Tutora de proyecto

## Resumen

Este proyecto aborda el diseño de un modelo de despliegue dinámico de drones basado en Machine Learning, utilizando algoritmos para el entrenamiento de una red neuronal. El trabajo se centra en el diseño de una red neuronal de tipo LSTM, la cual optimiza la ubicación de drones en tiempo real, garantizando una cobertura efectiva de zonas críticas. Mediante mapas de calor y simulaciones de pruebas alimentado por una base de datos, con variables como coordenadas de latitud, altitud y longitud, los drones son capaces de identificar zonas y/o áreas con alta densidad poblacional y nivel de prioridad con el fin de no solo posicionarse estratégicamente, sino reducir el consumo de energía debido el uso de rutas optimizadas. Esto asegura que cada uno de los 50 drones logren estar en el lugar adecuado, en el momento exacto.

Finalmente, la implementación de la red neuronal de tipo LSTM con la red de emergencia conformada por drones demostró ser una alternativa viable para solventar las limitaciones de funcionamiento de los sistemas de comunicaciones tradicionales durante y después de una situación crítica, debido al establecimiento de nodos autónomos y temporales, facilitando la comunicación entre los grupos de rescatistas y las personas afectadas de una localidad en específico y optimizando los tiempos de búsqueda.

**Palabras Clave:** Machine Learning, LSTM, Densidad poblacional

## ***Abstract***

*This Project focuses on the design of a dynamic drone deployment model based on Machine Learning (ML), utilizing algorithms for training a neural network. The work focuses on designing a Long-Short Term Memory (LSTM) neural network model, which optimizes the real-time positioning of drones to ensure effective coverage of critical areas. Through heat maps and test simulations powered by a database containing variables such as latitude, altitude, and longitude coordinates, the drones can identify zones and/or areas with high population density and priority levels. This allows them not only to be strategically positioned but also to reduce energy consumption by using optimized routes. This ensures that each of the 50 drones will be in the right place at the right time.*

*Finally, the implementation of the LSTM-type neural network with the emergency network composed of drones proved to be a viable alternative to overcome the operational limitations of traditional communications systems during and after a critical situation. This is due to the establishment of autonomous and temporary nodes, facilitating communication between rescue teams and affected individuals in a specific location while optimizing search time.*

***Keywords:*** *Machine Learning, LSTM, population density*

# Índice general

Resumen.....	1
<i>Abstract</i> .....	2
Índice general.....	3
Abreviaturas.....	5
Índice de figuras.....	6
Índice de tablas.....	7
Capítulo 1: Introducción.....	8
1.1 Introducción.....	9
1.2 Descripción del Problema.....	10
1.3 Justificación del Problema.....	11
1.4 Objetivos.....	12
<i>1.4.1 Objetivo General</i> .....	12
<i>1.4.2 Objetivos específicos</i> .....	12
1.5 Marco Teórico.....	13
<i>1.5.1 Machine Learning (ML)</i> .....	13
<i>1.5.2 Redes Neuronales</i> .....	13
<i>1.5.3 Modelo MPL (Perceptrón Multi Capa)</i> .....	15
<i>1.5.4 Modelo LSTM (Memoria a Largo-Corto Plazo)</i> .....	15
<i>1.5.5 Aprendizaje profundo por refuerzo (DRL)</i> .....	15
<i>1.5.6 Algoritmo Optimización de la política próxima (PPO)</i> .....	15
Capítulo 2: Estrategia de Despliegue de Drones en Redes de Emergencia.....	16
2.1 Preparación de datos.....	17
<i>2.1.1 Descripción del conjunto de datos</i> .....	17
2.2 Diseño del entorno de simulación.....	18
<i>2.2.1 Definición del entorno y sus parámetros</i> .....	19
<i>2.2.2 Determinación del punto central de operaciones</i> .....	20

2.2.3 Inicialización de drones y rutas .....	20
2.3 Modelo de Aprendizaje para Optimización de Ubicación.....	21
2.3.1 Selección de Modelo LSTM.....	22
2.3.2 Función de Recompensa Basada en Prioridades de Emergencia.....	23
2.4 Estrategias de Exploración y Comunicación .....	24
2.4.1 Exploración de Rutas para Cobertura Óptima .....	24
2.4.2 Comunicación y Conectividad Entre Drones .....	25
Capítulo 3: Condiciones de Prueba y Evaluación del Despliegue de Drones .....	26
3.1 Análisis de la configuración inicial.....	27
3.1.1 Descripción de las condiciones de prueba .....	27
3.1.2 Ubicación inicial y conectividad de los drones .....	28
3.2 Desempeño del modelo LSTM en el despliegue de drones .....	28
3.2.1 Evaluación del modelo de aprendizaje LSTM .....	29
3.3 Cobertura y eficiencia del despliegue .....	31
3.3.1 Cobertura de zonas prioritarias.....	31
3.3.2 Distribución y consumo de baterías .....	32
3.4 Interpretación de los mapas generados .....	32
3.4.1 Mapa de calor de zonas de emergencias .....	32
3.4.2 Mapa del punto central de operaciones .....	34
3.4.3 Mapa de posiciones iniciales y finales de drones .....	35
3.4.4 Mapa de comunicación entre drones.....	37
3.4.5 Mapa de calor de zonas de emergencias con posición final de drones .....	38
Capítulo 4: Conclusiones y Recomendaciones.....	39
4.1 Conclusiones .....	40
4.2 Recomendaciones.....	41
Bibliografía.....	42

## Abreviaturas

ML	Machine Learning	Aprendizaje Automático
LSTM	Long Short-Term Memory	Memoria a corto-largo plazo
MLP	Multilayer Perceptron	Perceptrón Multicapa
PPO	Proximal Policy Optimization	Optimización de la política próxima
DRL	Deep Reinforcement Learning	Aprendizaje profundo por refuerzo

## Índice de figuras

Ilustración 2.1 Parámetros de preparación de datos.....	18
Ilustración 2.2 Secuencia para el diseño del entorno de Simulación.....	18
Ilustración 2.3 Parámetros Necesarios para la Simulación .....	19
Ilustración 2.4 Representación de una red neuronal estándar [14].....	21
Ilustración 2.5 Caracterización de una etapa de una Red LSTM [14].....	22
Ilustración 3.1 Mapa de despliegue utilizando MLP .....	30
Ilustración 3.2 Mapa de calor de densidad poblacional y limitación de zona de estudio .....	33
Ilustración 3.3 Punto Central de Operaciones.....	35
Ilustración 3.4 Posición inicial de Drones .....	35
Ilustración 3.5 Posición Final de Drones .....	36
Ilustración 3.6 Mapa de comunicación entre drones.....	37
Ilustración 3.7 Mapa de calor de zonas de emergencias con posición final de drones .....	38

## Índice de tablas

Tabla 1.1 Características entre redes neuronales MLP y LSTM .....	14
Tabla 3.1 Parámetros de simulación de despliegue de drones.....	27

# **Capítulo 1: Introducción**

## 1.1 Introducción

Al momento de presenciar desastres naturales como terremotos, inundaciones, tsunamis o huracanes, por lo general, las redes e infraestructuras de comunicaciones suelen colapsar, dejando así a las personas aisladas y sin manera alguna de solicitar ayuda. Esta pérdida de comunicación no solamente dificulta el trabajo a los equipos de rescate, sino que aumenta el riesgo de aquellas personas que esperan el auxilio luego de la tragedia natural. Con los medios de comunicación colapsados, las autoridades competentes, y los equipos de emergencia enfrentan obstáculos significativos para poder socorrer a las personas afectadas de manera oportuna. En los escenarios donde los caminos están bloqueados y ciertas zonas son difíciles de alcanzar, coordinar un rescate es simplemente imposible si no se tiene una comunicación estable y confiable, lo que, inevitablemente podría llevar a una respuesta tardía. En esta situación se ve reflejado lo importante que resulta contar con una solución que permita mantener la comunicación para poder actuar de forma eficiente en situaciones donde los sistemas de comunicación tradicional fallan. Es fundamental contar con vínculos de comunicación eficientes y eficaces antes, durante y después de un desastre, en particular entre el personal de emergencia para ayudar en la respuesta y recuperación ante desastres. Sin embargo, la falla de los sistemas de comunicación ya sea total o parcial, como los sistemas de radio o satélites, ha causado ineficiencia y demoras en las tareas de socorro y respuesta ante emergencias, esto conduce a la pérdida de vidas y lesiones evitables. La falla de los sistemas de comunicación puede causar daños catastróficos a la humanidad y a las actividades financieras-económicas, ya que las personas no pueden tener comunicación entre sí de manera oportuna y con excelente calidad de servicio [1].

La implementación de drones en redes de telecomunicaciones diseñadas para situaciones de emergencia ha impulsado innovaciones y avances tecnológicos en el despliegue de redes de comunicaciones móviles. Se obtuvo un avance significativo en la fase inicial del proyecto “Implementación de red de acceso de red de drones para comunicaciones de emergencia”, en la que fue desarrollada una red de interconexión de drones que actuaban modelando un backbone móvil. Esta infraestructura demostró ser de suma importancia al momento de facilitar la conexión y comunicación entre dispositivos, ya que, de esta manera se otorgó a la red de drones un papel central en caso de colapso de las redes de comunicación convencionales [2].

Con los resultados de trabajos anteriores, se logró que los drones se posicionen como una herramienta comparable a los de sistemas de radio móvil, siendo capaces de crear nodos de red autónomos y temporales incluso teniendo en cuenta entornos complejos. No obstante, aún existen

desafíos y dificultades al momento de definir la disposición óptima de despliegue para la red de drones y en la expansión de la capacidad de esta para alcanzar una cobertura más amplia. Con la finalidad de contrarrestar dichas limitaciones, se plantea este proyecto, en el que se explorará recientes avances de tecnología de radiofrecuencia y en las capacidades avanzadas de drones, adaptando así nuevos modelos de red y soluciones.

La solución implementada en el presente proyecto tiene como punto central a las redes neuronales, que son una forma de inteligencia artificial que simula el funcionamiento del cerebro humano. Están compuestas por elementos como neuronas artificiales, que se disponen en varias capas. El aprendizaje es fundamental para la plasticidad de una red neuronal y, básicamente, es el proceso mediante el cual las conexiones sinápticas se ajustan, permitiendo que la red responda de manera diferente a los estímulos del entorno [3].

Las redes neuronales, con el paso de los años, se han convertido en una herramienta de suma importancia para resolver problemas que requieran una alta capacidad de procesamiento de datos, y toma de decisiones óptimas y sub-óptimas. La aplicación de este tipo de redes será fundamental para la realización de este proyecto. Ya que, debido a la capacidad de las redes neuronales de aprender sobre situaciones anteriores, identificación de patrones, y predicción de resultados a través de múltiples factores y parámetros, se presentan como una herramienta clave para la definición de ubicaciones óptimas de despliegue para la red propuesta, manteniendo una conexión ininterrumpida y estable.

## **1.2 Descripción del Problema**

La problemática por tratar en este proyecto es la ineficiencia de la comunicación efectiva durante y después de situaciones de emergencia, especialmente en casos donde las infraestructuras de telecomunicaciones tienden a colapsar, es decir, es casos de catástrofes naturales o eventos con emergencias múltiples, donde una falta de comunicación entre los organismos de rescate podría empeorar las condiciones y limitar los esfuerzos de los rescatistas. Partiendo de la situación mencionada, el trabajo se enfoca especialmente en el despliegue óptimo de una red de drones en el campus de ESPOL, ubicándolos de manera estratégica para asistir de forma rápida y efectiva, en las operaciones de rescate tras una situación de emergencia. Lo que se propone con esta red, es que, sea activada en aquellas áreas de mayor vulnerabilidad, facilitando una respuesta coordinada a los grupos de rescate. Esto se vuelve complejo teniendo en cuenta diversos factores y parámetros imprescindibles para modelar una situación de estas características, y que se le deben tomar en cuenta si lo que se quiere lograr es una correcta simulación; tales como la densidad poblacional, la geografía urbana, la accesibilidad, las condiciones

naturales del terreno de despliegue, además de las limitaciones técnicas de los equipos utilizados para el despliegue e interconexión de la red, como la autonomía y radio de comunicación de los drones.

Par dar solución a estas limitaciones, la red implementará una red neuronal que se entrena con un historial de datos, y datos en tiempo real sobre desastres naturales antes mencionados, además, se tiene la capacidad de trabajar empleando patrones de tráfico, y parámetros como la densidad poblacional. Esta implementación permitirá que la red desplegada adapte su ubicación de manera dinámica, optimizando el tiempo de respuesta en cada emergencia que se presente, y mejorando de forma escalable y constante el tiempo de respuesta de las operaciones de rescate ante una emergencia.

### **1.3 Justificación del Problema**

Definir ubicaciones óptimas de despliegue para una red de emergencias es un paso de suma importancia para optimizar el tiempo de respuesta ante situaciones catastróficas, como los desastres mencionados en la sección anterior, o incluso para emergencias médicas cotidianas. Además, esta acción permitirá cubrir áreas más extensas y amplias en menos tiempo, así como el facilitar el acceso a nuevas zonas complejas para una vía tradicional. Esta idea es crucial en regiones remotas donde la infraestructura de las redes no se encuentra en condiciones idóneas y simplemente están limitadas, ya que, facilita una llegada eficiente de ayuda, lo cual definitivamente marcará una diferencia al momento de salvar una vida luego de una emergencia.

Definir correctamente las ubicaciones de despliegue también permite usar eficientemente los recursos disponibles, teniendo en cuenta las características de estos. Es decir, teniendo en cuenta la autonomía de los dispositivos, se puede obtener un mayor tiempo de vuelo reduciendo costos de operación. Además, teniendo en cuenta otros factores y parámetros como el terreno y obstáculos potenciales, se garantiza que los drones se encontrarán en operación de manera secuencial y sin interrupciones o inconvenientes, incluso en condiciones y limitaciones reales. Distribuir los drones de manera estratégica mejorará la cobertura de los equipos de rescate en las áreas afectadas, además de facilitar la comunicación y coordinación entre los equipos de rescate, aumentando la flexibilidad, escalabilidad y eficiencia de la red ante los cambios de las necesidades en momentos de emergencia.

Teniendo en cuenta los beneficios a largo plazo de esta solución, su aplicación no sólo mejorará y mejorará la capacidad de respuesta en situaciones post emergencia, sino que ayudará a las comunidades afectadas a estar mejor preparadas para enfrentar desastres naturales. En el apartado científico, el proyecto contribuirá notablemente al crecimiento de las gestiones de infraestructuras de redes de

telecomunicaciones en situaciones de crisis, otorgando un modelo a replicar en distintos proyectos, contextos y países.

Mencionando trabajos anteriores, los drones utilizados para la entrega de desfibriladores funcionan mediante vuelos automáticos dirigidos a los lugares donde se reportan emergencias cardíacas. Equipados con sistemas GPS y navegación avanzada, pueden calcular las rutas más rápidas, esquivando obstáculos como el tráfico o barreras geográficas que podrían retrasar a los servicios de emergencia tradicionales. Esta capacidad permite reducir considerablemente el tiempo de respuesta en situaciones críticas, lo que incrementa las probabilidades de supervivencia al brindar asistencia médica de manera más rápida y eficiente [4].

El trabajo mencionado anteriormente “Drones para la entrega de desfibriladores” podría tener una mejora con el uso o implementación de redes neuronales a su diseño de red, ayudando a mejorar los tiempos de respuesta y precisión de las entregas. Utilizando este diseño de red neuronal, los drones podrán aprender y analizar datos en tiempo real, como las condiciones del tráfico y clima además como las ubicaciones de pedidos y emergencias previas para la entrega de equipo, optimizando tiempos y rutas de vuelo. Además, con ayuda de la red neuronal, la red podrá predecir p anticipar patrones de emergencia y priorizar los casos más críticos, estableciendo parámetros d prioridad, aumentando de esta manera la eficiencia del sistema y de la red de drones.

## **1.4 Objetivos**

En este apartado se presentarán los objetivos, tanto general como específicos

### ***1.4.1 Objetivo General***

Determinar ubicaciones óptimas de drones utilizando Machine Learning para el despliegue de una red de emergencia considerando limitaciones reales como densidad poblacional, cantidad de drones y cobertura de red.

### ***1.4.2 Objetivos específicos***

- Realizar una revisión bibliográfica de las aplicaciones de redes neuronales, algoritmo de Machine Learning y técnicas de optimización en la determinación de ubicaciones óptimas para el despliegue de drones.
- Implementar un modelo basado en redes neuronales para la óptima ubicación de drones en el despliegue de una red de emergencia.

- Analizar el desempeño del modelo propuesto basado en redes neuronales para la optimización de la ubicación de drones en un entorno con diferentes niveles de densidad poblacional, usando métricas como limitaciones de la zona, tiempo de vuelo, consumo de energía y cobertura de comunicación entre los drones desplegados.

## **1.5 Marco Teórico**

En esta sección se mencionarán conceptos teóricos que están ligados y relacionados a la implementación del código que permitirá el desarrollo de los objetivos del proyecto.

### ***1.5.1 Machine Learning (ML)***

El machine learning (ML), o aprendizaje automático, es un campo dentro de la inteligencia artificial que permite a las máquinas aprender por sí mismas a partir de los datos, sin necesidad de ser programadas de manera directa para cada tarea. A lo largo del tiempo, ML ha avanzado enormemente, integrando el uso de redes neuronales artificiales, como MLP y LSTM, y algoritmos como Nearest Neighbor, que permiten a las máquinas reconocer patrones y tomar decisiones más precisas [5].

ML puede ser una herramienta muy útil para optimizar dónde posicionar drones de rescate, ya que permite analizar una gran cantidad de datos sobre el terreno y otros factores importantes, como el tiempo que se tarda en llegar, los obstáculos, las rutas disponibles y las áreas más riesgosas. Utilizando algoritmos que aprenden a partir de estos datos, es posible identificar patrones que ayuden a predecir los mejores lugares para colocar los drones, mejorando así la cobertura y reduciendo el tiempo de respuesta en situaciones de emergencia. Además, estos sistemas pueden ajustarse sobre la marcha a medida que cambian las condiciones, haciéndolos aún más efectivos.

### ***1.5.2 Redes Neuronales***

Es un tipo de modelo de ML, el que permite a las computadoras procesar información de forma similar al cerebro humano. Es una red formada por neuronas artificiales, que se organizan en capas, las cuales van desde una de entrada, donde llega la información, hasta una de salida, que genera el resultado, pasando por capas intermedias que pueden ser pocas o muchas. Cada neurona envía su señal a otras a través de conexiones llamadas "sinapsis", donde la información se ajusta con un valor (peso) y luego se transforma a través de una función de activación, que puede ser, por ejemplo, una función sigmoideal. Esta transformación decide qué tan activa estará la neurona receptora [6].

Los pesos se ajustan con el objetivo de minimizar un error global que refleja el objetivo del problema, generalmente usando un algoritmo de optimización. Aunque se suele comparar este modelo con el funcionamiento del cerebro humano, su valor real está en su capacidad para aprender y resolver problemas complejos como reconocimiento facial, ubicaciones geográficas, aprendizajes autónomos, con una precisión mayor debido al entrenamiento otorgado [7].

**Tabla 1.1**

*Comparación de Características entre redes neuronales MPL y LSTM*

<b>Característica</b>	<b>MPL (Perceptrón Multicapa)</b>	<b>LSTM (Memoria a Largo-Corto Plazo)</b>
<b>Tipo de datos</b>	Datos estructurados y estáticos como coordenadas.	Datos temporales, patrones históricos de tráfico, clima y eventos emergentes.
<b>Capacidad de contexto temporal</b>	Baja, cada entrada es independiente.	Alta, optimizada para analizar secuencias temporales con el fin de predecir zonas de emergencia.
<b>Aplicación en ubicación de drones</b>	Útil para identificación de zonas generales, pero no considera cambios en el tiempo.	Predice y ajusta la ubicación del dron en respuesta a cambios temporales como aumento poblacional o patrones de desastres.
<b>Eficiencia en secuencias</b>	Limitada a datos fijos.	Excelente para secuencias temporales, se adapta a emergencias prolongadas.
<b>Adaptación a cambios en tiempo real</b>	Limitada, requiere reentrenamiento [8].	Alta, permite ajustar la ubicación de drones en tiempo real [9].

### ***1.5.3 Modelo MPL (Perceptrón Multi Capa)***

Se trata de un modelo neuronal ampliamente utilizado en diversas aplicaciones. Su estructura se organiza en capas, con el flujo de información moviéndose en una única dirección, desde la capa de entrada (que actúa como receptores) hasta la capa que genera la salida del sistema. Durante este proceso, la información es procesada gradualmente por diferentes capas intermedias (ocultas) de unidades [8].

Es importante señalar la relevancia de seleccionar una topología adecuada para la red, lo que se define por el número de capas ocultas y la cantidad de unidades en cada una. Si la red es demasiado pequeña, no podrá modelar correctamente los datos. Por otro lado, si la red es muy grande, puede generar varias soluciones coherentes, lo que se conoce como "overfitting" [10] [11].

### ***1.5.4 Modelo LSTM (Memoria a Largo-Corto Plazo)***

Por sus siglas en inglés, LSTM (Long-Short Term Memory) es un tipo de red neuronal, cuyo propósito es modelar secuencias temporales y sus dependencias de largo alcance con una mayor exactitud y precisión que una red neuronal RNN convencional. LSTM incluye unidades especiales denominadas bloques de memoria en su capa oculta recurrente, estos bloques contienen celdas de memorias con auto conexiones que retienen el estado temporal de la red, y trabajan con unidades multiplicativas especiales llamadas puertas que se encargan de regular el flujo de información [12].

### ***1.5.5 Aprendizaje profundo por refuerzo (DRL)***

Por sus siglas en inglés DRL (Deep Reinforcement Learning), es la unión de un aprendizaje profundo y un aprendizaje por refuerzo, el cual es utilizado para que un agente aprenda a tomar decisiones bajo un entorno definido, logrando maximizar una recompensa a medida que aprende a largo plazo, este algoritmo trabaja en conjunto con las redes neuronales profunda para un óptimo entrenamiento bajo un escenario planteado [13].

### ***1.5.6 Algoritmo Optimización de la política próxima (PPO)***

Por sus siglas en inglés PPO (Proximal Policy Optimization), es un algoritmo que está englobado en el campo de aprendizaje por refuerzo (RL), cuyo objetivo principal es maximizar las políticas de un agente, en este caso un dron. Estas políticas vienen a ser un conjunto de directrices o comandos que utiliza el agente para una mejor toma de decisiones bajo un entorno definido, es decir, la PPO de un agente trata de identificar idealmente una decisión con el fin de maximizar el beneficio acumulado [14].

# **Capítulo 2: Estrategia de Despliegue de Drones en Redes de Emergencia**

## **2.1 Preparación de datos**

La preparación de datos es importante en este trabajo, ya que se requiere un aprendizaje automático. Esto es debido a que se debe asegurar que los datos usados para entrenar la red y evaluar el modelo de solución sean de alta fidelidad y consistentes. Preparar datos consiste en la transformación de los mismos sin procesarlos, de tal manera que pueden ser extraídos de muchas fuentes de información, de manera ordenada y manteniendo un formato sin errores y adecuados para realizar un análisis y modelado correspondiente. El propósito principal de esta fase es optimizar los datos para adecuarlos al uso de algoritmos de aprendizaje empleados en el código de la solución, a través de una limpieza y unificación de parámetros.

### ***2.1.1 Descripción del conjunto de datos***

El principal objetivo que se requiere lograr es identificar y definir las ubicación óptimas y eficientes para el despliegue de una red de drones de emergencia. Por esta razón, los parámetros seleccionados son vitales para la representación de las condiciones del entorno y el establecimiento de las zonas de mayor nivel prioritario.

El parámetro “Densidad Poblacional” es fundamental para definir ubicaciones óptimas de despliegue para una red, debido a que, permite saber que zonas tienen más personas, y por ende más usuarios en necesidad de rescate en caso de una catástrofe. De esta manera, se puede empezar priorizando las áreas que cuentan con mayor tráfico de personas y garantizar una respuesta rápida para una cantidad importante de usuarios en un caso de post emergencia. En otras palabras, esto va a permitir que los drones lleguen en primer lugar a las zonas de más alta demanda, ayudando a los organismos de rescate en la búsqueda y rescate del mayor número de personas posible.

También se trabaja con el parámetro denominado “Nivel de Prioridad”, el cual ayudará a identificar las áreas de mayor importancia durante una situación de emergencia. Un ejemplo de estas zonas, teniendo en cuenta aplicaciones a mayor escala, sería un hospital, o una escuela, o aquellos lugares con infraestructuras críticas o gravemente afectadas por la situación, por lo cual requieren atención inmediata. En el despliegue de una red de drones, este indicador se usó para enviar a los lugares seleccionados como prioritarios, es decir, en lugares donde más se los requiera, y de esta manera mejorar la eficiencia de la red. Al combinar este parámetro con la densidad poblacional, se planifica de mejor manera una respuesta óptima, obteniendo una mejor gestión de los recursos disponibles en las zonas más vulnerables.

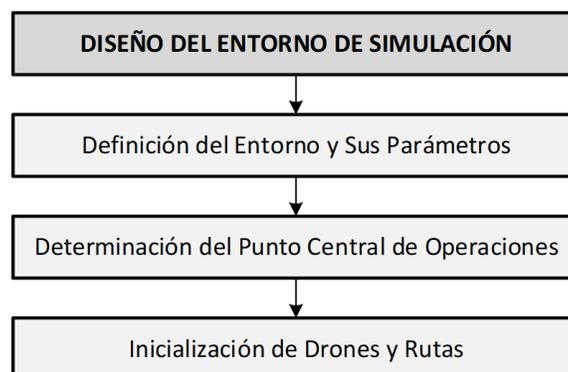
Los indicadores geográficos, tales como “latitud”, “longitud” y “altitud”, son aquellos parámetros que le permitirán a la red a mapear puntos específicos con una alta precisión. Estos ayudarán a definir las coordenadas prioritarias para proceder con el despliegue de la red, además de mantener un monitoreo constante de las rutas de vuelo de los drones y un control de las zonas más vulnerable durante las emergencias. En términos generales, la integración con mapas intensivos garantiza que los drones lleguen a donde más se necesitan y trabajen correctamente en un estado de emergencia, y también posibilita un control adecuado de los equipos de la red, y las zonas de vulnerabilidad.



**Ilustración 2.1** Parámetros de preparación de datos

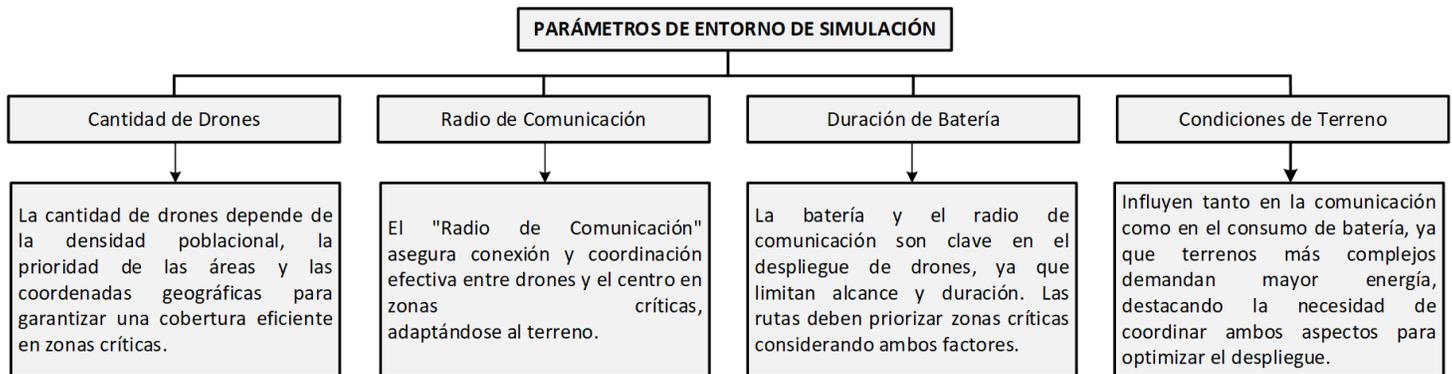
## 2.2 Diseño del entorno de simulación

El entorno de simulación debe ser correctamente diseñado debido a que, esto es fundamental para definir si el modelo propuesto funciona y cumple con los resultados esperados. Un entorno de simulación correctamente diseñado e implementado permite recrear un escenario realista con un alto porcentaje de semejanza con la realidad, en este caso, un escenario real e ideal sería aquel donde los drones puedan operar de manera eficiente, sin dejar de lado aquellas limitaciones mencionadas en secciones anteriores, y los desafíos presentes en cualquier situación de emergencia.



**Ilustración 2.2** Secuencia para el diseño del entorno de Simulación

### 2.2.1 Definición del entorno y sus parámetros



**Ilustración 2.3** Parámetros Necesarios para la Simulación

Si se requiere despegar de forma óptima una red de emergencia conformada por drones, el número disponible de equipos para abastecer una correcta operación es de suma importancia, teniendo en cuenta que el objetivo principal es cubrir la mayor cantidad de áreas de interés. Para poder determinar esta cantidad, el proyecto toma un enfoque principalmente en el parámetro de la densidad poblacional. Se parte de la base de que, a mayor cantidad de habitantes, usuarios de la red, mayor será la demanda de drones para abastecer las operaciones de rescate en situaciones de post emergencia. Teniendo en cuenta esta idea, se planteó que posean la mayor cantidad de estudiantes en el campus politécnico, serán aquellas que requieren una mayor cantidad de drones desplegados. También se asignó un nivel de prioridad a cada una de las facultades de ESPOL para diferenciarlas en función de su densidad poblacional con la finalidad de distribuir los drones de manera guiada y organizada en orden jerárquico, garantizando la presencia de drones necesarios en los puntos donde la demanda sea mayor.

El radio de comunicación es la distancia máxima al punto central de operaciones en la que un equipo de la red puede posicionarse para poder permanecer en funcionamiento. Este dato es fundamental para que las operaciones de la red puedan coordinarse de manera eficiente en las distantes zonas de cada nivel de prioridad. Configurar de manera correcta este parámetro asegura que los drones se mantengan conectados incluso en las áreas críticas donde entran en juego las limitaciones de la infraestructura de la red. La distancia máxima de comunicación deberá ajustarse a las condiciones del terreno, es decir, va a depender explícitamente de las coordenadas geográficas y la ubicación exacta del despliegue de la red de emergencias.

La autonomía de los equipos que conforman la red se relaciona de manera directa al radio de comunicación, ya que el rendimiento y la funcionalidad de la red dependen estrictamente de estos

parámetros. Mientras que el radio de comunicación define la longitud máxima en la que un dron puede mantenerse conectado al punto central de operaciones, la autonomía de los dispositivos limita el tiempo en el que pueden mantenerse en operación en este rango de distancia. Por esta razón, se debe considerar la autonomía y el radio de comunicación al momento de planificar las rutas de operación de la red desplegada. También es importante considerar que, la altitud y características del terreno no sólo afectan el alcance de la red desplegada, sino que también afecta el tiempo de operación de los dispositivos, ya que estos requerirán un mayor consumo de energía en terrenos complejos.

### ***2.2.2 Determinación del punto central de operaciones***

La densidad poblacional es el parámetro inicial para determinar el punto central de operaciones de la desplegada mediante la función *calculate\_central\_point*; Se implementó esta función con la finalidad de identificar cuál es el área con mayor número de usuarios gestionando los datos proporcionados por la base de datos generada; Gracias a esta función, se asegura que la operación de la red de emergencias comience sus operaciones en el lugar de más alta prioridad maximizando su impacto. El código implementado también utiliza los parámetros de datos geográficos correspondientes a cada zona de diferente nivel de prioridad para ubicar de manera exacta el punto inicial y ubicarlo de manera correcta, y esto a su vez, brinda una apertura a las actividades de exploración y procesamiento de drones para una futura ubicación de despliegue.

### ***2.2.3 Inicialización de drones y rutas***

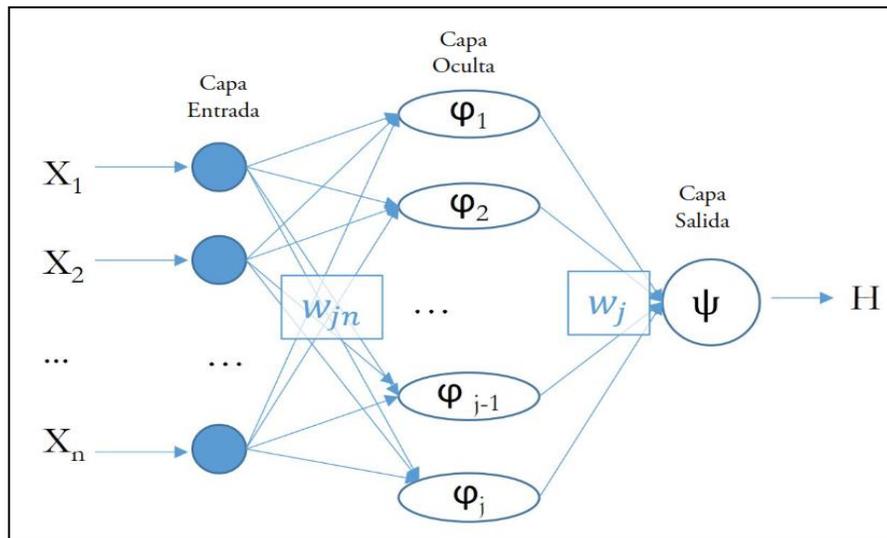
Para la inicialización de los equipos de la red, el punto central de operaciones, y las rutas de vuelo, se implementa la función *initialize\_drone\_positions*. Para conectar el procesado de datos del código implementado para la solución, la función *calculate\_central\_point* calcula la ubicación del punto central de operaciones, luego, la función *initialize\_drone\_positions* posiciona todos los equipos pertenecientes a la red en la ubicación calculada para el inicio del despliegue. El parámetro *max\_battery* define el estado de batería que cada dron posee en un estado inicial antes de comenzar la distribución de la red, además una ruta vacía registra las posiciones de las visitas de cada operación de la red para sumarlas a los datos por la red neuronal; Dicha red neuronal implementada permite que los drones registren y analicen sus posiciones en función de las acciones tomadas, para que las rutas almacenadas sean consideradas al momento de desplegar nuevamente la red de emergencias, mejorando su desempeño.

## 2.3 Modelo de Aprendizaje para Optimización de Ubicación

Las redes neuronales son imprescindibles al momento de predecir fenómenos y secuencias complejas, debido a que tienen la capacidad de realizar un modelado de funciones no lineales, así como generar patrones asimétricos en un periodo largo de tiempo. Una red neuronal sencilla, con una única capa intermedia, como la de la *Ecuación 1*, permite relacionar “ $n$ ” variables de entrada,  $x_n$ , con una variable de salida,  $H$  [15]. La capa intermedia tendría pesos de entrada,  $w_{jn}$ , y de salida,  $w_j$ . Las variables de entrada  $x_n$  se conectan con las “ $j$ ” neuronas de la capa intermedia, y esta con la variable de salida,  $H$ , a través de los pesos y las funciones de activación,  $\varphi_j$  y  $\psi$ , respectivamente [15].

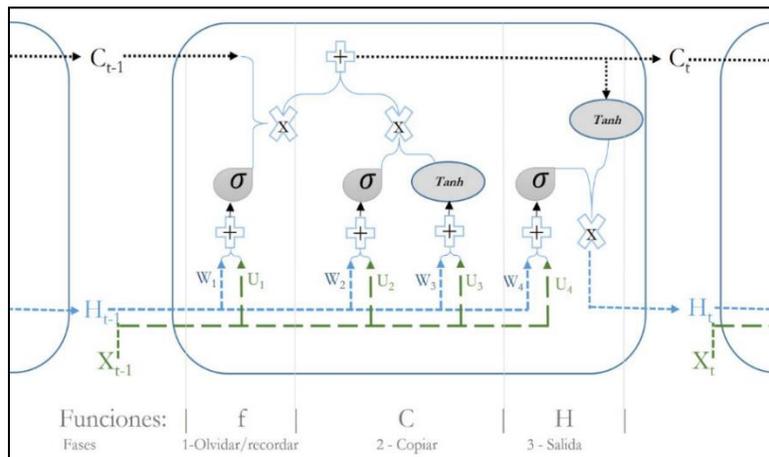
$$H = \psi \left[ \sum_j w_j \varphi_j \left( \sum_n w_{jn} x_n \right) \right]$$

La *ilustración 2.4* propone una representación gráfica de una red neuronal sencilla como la ilustrada en la *ecuación 1*. Tiene una capa de entrada de variables y otra capa de salida. En este caso, ambas están conectadas mediante una sola capa oculta intermedia. Los pesos de salida,  $w_j$  y, de entrada,  $w_{ji}$ , se modifican en cada iteración. El comportamiento de la red ilustrada en la *ilustración 2.4* está gobernado por los parámetros y funciones de activación de la red desplegada [15].



**Ilustración 2.4** Representación de una red neuronal estándar [14]

El funcionamiento de una etapa en una red neuronal LSTM se muestra en la *ilustración 2.5*, resaltando la interrelación entre sus elementos fundamentales: el estado celular  $C_t$ , el estado oculto  $H_t$  y la entrada actual  $X_t$ . La arquitectura comprende tres puertas esenciales: la puerta de olvido  $f_t$ , la entrada  $i_t$  y la salida  $o_t$ , todas ellas controladas por funciones de activación como la sigmoide ( $\sigma$ ) y la tangente hiperbólica ( $\tanh$ ). La puerta de olvido determina qué datos del estado anterior ( $C_{t-1}$ ) deben ser eliminados o preservados, mientras que la puerta de entrada posibilita la adición de nueva información al estado de la celda  $C_t$ , incorporando valores de  $X_t$  y  $H_{t-1}$ . En última instancia, la puerta de salida determina qué segmento del estado actual de la celda se emplea para producir la salida presente  $H_t$ . Este procedimiento está mediado por pesos ( $W$  y  $U$ ) particulares para cada entrada, que modifican las aportaciones de los ingresos y los estados anteriores. La incorporación exacta de estos componentes posibilita que la red LSTM adquiera dependencias a largo plazo en secuencias, ajustándose de manera dinámica al contexto y optimizando la eficacia de la memoria.



**Ilustración 2.5** Caracterización de una etapa de una Red LSTM [14]

### 2.3.1 Selección de Modelo LSTM

En la sección 1.5.1 se mencionó al modelo de red neuronal LSTM como una herramienta importante para generar modelados de secuencias temporales y dependencias a largo plazo con mayor precisión y optimización que una red neuronal recurrente tradicional; Esta función es posible gracias a los bloques de memoria, y las puertas lógicas cuya función es la regularización del flujo de información y la conservación del estado temporal y dinámico de la red [12]. Para implementar este modelo de red neuronal de forma óptima con la red de drones desplegada. Junto con la definición de los puntos estratégicos de inicio de operación, es necesario realizar un seguimiento del proceso de modelado de secuencia de datos, entrenamiento de la red e integración con el sistema de control nativo de la red.

El diseño de red LSTM será muy útil para trabajar con datos que cambian a lo largo del tiempo. En una red de emergencia con drones, se pueden usar para analizar tanto información pasada como datos en tiempo real. Esto incluye aspectos clave como cuántas personas hay en una zona, dónde y con qué frecuencia han ocurrido emergencias antes, cómo están las condiciones del entorno y cómo han funcionado los drones en despliegues anteriores, considerando tiempos de respuesta, consumo de energía y cobertura. Toda esta información ayuda a identificar patrones y a decidir cuáles son las áreas más importantes en caso de emergencia.

Para que la red LSTM pueda predecir dónde deberían ubicarse los drones, se entrena utilizando un conjunto de datos que refleje cómo se comportan las emergencias en la región de interés. Estos datos incluyen las variables clave mencionadas en el capítulo anterior, mientras que el objetivo principal es encontrar las ubicaciones óptimas que aseguren una mejor cobertura y reduzcan el tiempo de respuesta. Este entrenamiento permite que la red entienda patrones complejos y relaciones en el comportamiento de las emergencias, haciendo que las decisiones sean más precisas y efectivas.

Una vez entrenada, la red LSTM se conecta al sistema de navegación de los drones para que, en situaciones de emergencia, pueda usar datos en tiempo real de sensores y redes y así identificar las zonas más críticas donde deben posicionarse. Además, la red no solo se basa en datos históricos, sino que también ajusta sus predicciones constantemente a medida que los drones recopilan nueva información del terreno y las condiciones cambian. Esto permite que las decisiones sean rápidas y se adapten al contexto, haciendo que los drones se reubiquen de manera eficiente conforme la emergencia evoluciona.

### ***2.3.2 Función de Recompensa Basada en Prioridades de Emergencia***

La función de recompensa está diseñada para incentivar a los drones a ubicarse en áreas con alta densidad poblacional. Durante cada paso de la simulación, la recompensa se calcula basándose en la densidad de población de la zona más cercana al dron en su posición actual. Esto se implementa en el método `step`, donde se utiliza la función `get_population_density` para determinar la densidad poblacional en función de la latitud, longitud y altitud del dron. Este enfoque guía a los drones hacia áreas de mayor prioridad en términos de densidad poblacional, reflejando las necesidades críticas en un escenario de emergencia.

## 2.4 Estrategias de Exploración y Comunicación

En esta sección se analizan los aspectos fundamentales para proceder con la implementación de la red de despliegue; Como la exploración de rutas y la comunicación efectiva de todos los equipos de la red con el punto central de operaciones, abordando el diseño del sistema de exploración y optimizando su funcionamiento en función del parámetro de la densidad poblacional y el sistema de prioridad de las distintas zonas de despliegue.

### 2.4.1 Exploración de Rutas para Cobertura Óptima

La exploración de rutas para asegurar una cobertura óptima se realiza de forma iterativa, donde los drones se mueven estratégicamente para cubrir las áreas con mayor densidad de población. En cada paso de la simulación, se asignan movimientos específicos a cada dron, con el objetivo de maximizar la cobertura de estas zonas clave.

Los drones tienen la capacidad de moverse en un espacio bidimensional (latitud, longitud) o tridimensional (latitud, longitud, altitud) mediante acciones predefinidas. Esto les permite ajustar su posición a lo largo de los ejes de latitud, longitud y altitud:

- Movimiento en latitud: Los drones pueden desplazarse hacia el norte o hacia el sur.
- Movimiento en longitud: Los drones pueden moverse hacia el este o el oeste.
- Movimiento en altitud: Los drones tienen la capacidad de subir o bajar en altitud.

El sistema asigna las acciones de movimiento según las coordenadas actuales del dron, asegurando que siempre se mantenga dentro de los límites geográficos y altitudinales definidos.

Cada vez que un dron se mueve, el sistema calcula la densidad poblacional en su nueva ubicación utilizando una función específica. Esta densidad sirve como un indicador para otorgar recompensas, lo que motiva a los drones a dirigirse hacia las áreas con mayor concentración de población. El proceso de exploración se realiza en tiempo real, lo que permite ajustar la posición del dron de manera continua para mejorar la cobertura y adaptarse a los cambios en el entorno. De esta forma, los drones se mueven activamente hacia las zonas más pobladas, modificando su ruta en función de las recompensas obtenidas por las áreas cercanas con mayor densidad.

### ***2.4.2 Comunicación y Conectividad Entre Drones***

La comunicación y conectividad entre drones es esencial para garantizar el despliegue eficiente de una red de drones en situaciones de emergencia. El código implementa un sistema en el que los drones establecen vínculos de comunicación en función de su cercanía, formando pares de drones que se encuentran dentro del rango de comunicación para mantener una red de intercambio de información. A continuación, se explica cómo se desarrolla este proceso:

La comunicación entre drones se basa en un radio de comunicación predefinido, el cual está representado por el parámetro *communication\_radius*. Cuando la distancia entre dos drones es menor o igual a este radio, se considera que ambos drones pueden comunicarse entre sí.

El cálculo de la proximidad se realiza en el método *get\_communication\_pairs()*, que evalúa la distancia euclidiana entre los drones considerando solo las coordenadas de latitud y longitud (se omite la altitud). Si la distancia entre dos drones es menor o igual al valor de *communication\_radius*, se crea un par de comunicación entre esos drones.

# **Capítulo 3: Condiciones de Prueba y Evaluación del Despliegue de Drones**

### 3.1 Análisis de la configuración inicial

Esta sección está enfocada en analizar el diseño y las condiciones de prueba del entorno de simulación de la red de drones de emergencia, tomando en cuenta la configuración de los parámetros de simulación y estableciendo un entorno semejante a un escenario realista. Además, se brinda una explicación detallada de la estrategia de simulación, incluyendo la determinación del punto central de operaciones, estableciendo zonas críticas, garantizando una cobertura eficiente y una conectividad continua para coordinar los equipos de búsqueda y rescate en las zonas altamente pobladas. Esta explicación detallada considera un enfoque en la interacción de drones y su despliegue para aumentar su efectividad en situaciones de post emergencia.

#### 3.1.1 Descripción de las condiciones de prueba

Para las simulaciones realizadas a partir del código implementado, se utilizan datos reales los cuales fueron representados en el archivo *espol\_locations.csv*. El archivo contiene la información de los parámetros de densidad poblacional y los datos geográficos correspondientes a las diversas áreas de interés dentro de ESPOL. Los datos representan la correcta ubicación del despliegue de la red de emergencia en función de los parámetros presentados en la **tabla 3.1**

**Tabla 3.1**

*Parámetros de simulación de despliegue de drones*

<b>Parámetros de Simulación</b>	
<b>Número de drones</b>	Se definieron 50 drones activos para la simulación inicial. Esta cifra es una representación de una cantidad de despliegue adecuada para cubrir las áreas de interés.
<b>Capacidad máxima de baterías</b>	Cada dron tiene una batería inicial de 100%.
<b>Número máximo de pasos</b>	Este parámetro representa los ciclos de operación de la red, el cual está limitado a 500.
<b>Radio de comunicación</b>	Los drones tienen un radio de comunicación de 1,5 Km. Estar dentro de este parámetro garantiza su conexión adecuada con el punto central de operaciones.
<b>Acciones disponibles para cada dron</b>	Los drones pueden desplazar en seis direcciones; norte, sur, este y oeste. Además de poder ascender y descender.

El entorno de simulación fue diseñado utilizando un espacio de acción discreto, y se cuenta con un espacio de observación continuo contando con la posición geográfica de los equipos desplegados. Con esta acción, se asegura que el modelo simulado logre capturar tanto las condiciones estáticas del entorno, así como las condiciones dinámicas que son generadas a partir del movimiento de los drones. Para definir las coordenadas geográficas dentro del código, se utilizaron las variables  $\{\{lat\_centro\}\}$ ,  $\{\{lon\_centro\}\}$  y  $\{\{alt\_centro\}\}$ , para representar la latitud, longitud y altitud respectivamente. Con estas coordenadas la red puede desplegarse y comenzar sus operaciones.

### ***3.1.2 Ubicación inicial y conectividad de los drones***

Con la finalidad de garantizar que la red comience sus operaciones en la zona crítica más relevante, la ubicación inicial de la red desplegada se determina automáticamente definiendo como punto inicial la ubicación con la máxima densidad poblacional dentro de la base de datos proporcionada.

La cercanía de los drones en una posición centralizada, es decir, el punto central de operaciones asegura que todos los equipos se mantengan dentro del radio de 1,5 kilómetros en la fase inicial del despliegue, creando de esta manera una red de interconexión donde cada dron puede interactuar con los otros 49. El despliegue está enfocado en las zonas más afectadas luego de una emergencia, y a su vez, en las zonas de mayor densidad poblacional, para obtener una cobertura óptima y fundamental para coordinar las estrategias de distribución de los grupos de rescate.

## **3.2 Desempeño del modelo LSTM en el despliegue de drones**

Se escogió el algoritmo Proximal Policy Optimization (Recurrent PPO) como el modelo de aprendizaje por refuerzo a utilizarse en este proyecto; Esto se logró implementando una política cuya infraestructura lógica está basada en el uso de redes neuronales LSTM. Para comenzar, se utiliza la biblioteca Stable-Baselines3 para la configuración y entrenamiento inicial de la red de despliegue. El entorno de simulación personalizado DroneDeploymentEnv es modelado basándose en un DummyVecEnv para permitir su interconexión total. Para asegurar que las decisiones de los drones sean robustas y adaptativas, durante el proceso de entrenamiento de la red, el modelo simulado mejora el aprendizaje y la optimización de las acciones de drones para maximizar la recompensa acumulada en un intervalo de tiempo definido, adaptándose a las características dinámicas del entorno, como las limitantes en la autonomía y la comunicación con el punto central de operaciones.

El uso de redes neuronales se relacionó directamente con el modelo de aprendizaje por refuerzo Recurrent PPO. Esta solución se basa en la combinación de una red MLP y una red LSTM, con la finalidad de obtener un procesamiento de datos obtenidos del entorno, en este caso, de simulación. La función de la red neuronal implementada es recibir un conjunto de información sobre las características que describen el estado actual de los equipos de la red, como su posición, la densidad poblacional del área en donde se encuentran, y su nivel de batería. A partir del análisis de este conjunto de datos, la red aprende a tomar decisiones de la manera óptima para maximizar su desempeño. Esta mejora de desempeño permite que los drones realicen ciertas tareas complejas, tales como la exploración de zonas críticas, tener una correcta comunicación entre sí anclándose a un punto central de operaciones, y adaptarse a los cambios de las características del entorno en el que se opera.

La implementación de una red LSTM en la arquitectura de la red de drones, permite que dicho sistema capture relaciones temporales en el análisis de datos. Para un correcto despliegue de los equipos de la red, en donde la toma de decisiones en un momento específico, afectan un estado futuro de despliegue y los resultados de esta. Este modelo de red maneja secuencias de datos, recolectando datos para mantener un historial de acciones previas, entrenando y mejorando la red de drones para crear patrones de operación a medida que pasa el tiempo.

### ***3.2.1 Evaluación del modelo de aprendizaje LSTM***

Se implementó el modelo de aprendizaje Recurrent PPO con el tipo de red neuronal LSTM, ya que, esta solución tiene la capacidad de manejar datos secuenciales y dependencias temporales; Esta característica convierte esta solución en una herramienta óptima para abordar el problema de despliegue de drones. La arquitectura de la red neuronal implementada permite que la red capture y procese patrones a lo largo del tiempo, tales como la posición actual de los equipos desplegados y las recompensas y aprendizajes a futuro. El despliegue de una red de emergencia depende de la toma de decisiones estratégicas tomando en cuenta estados pasados, y estados activos de los drones, de esta manera se proyectan las acciones futuras, tales como un redespiegue en una zona de mayor demanda de usuarios, exploración de zonas críticas y mantener una comunicación interrumpida entre todos los drones. De esta manera, la integración con el algoritmo PPO, crea una red más estable y eficiente en entornos complejos, con una convergencia mejorada en los aprendizajes y toma de decisiones en un entorno simulado.

No obstante, la implementación de la red neuronal con el algoritmo PPO presenta ciertas limitantes que repercuten directamente en el desempeño de la red. En primer lugar, la representación de los datos observados en el entorno de simulación, que incluye las variables presentadas en la tabla x, podrían carecer de una correcta normalización, lo que afectaría de manera directa la capacidad de la red LSTM para tomar decisiones eficientes para un despliegue posterior. A su vez, la arquitectura del entorno simulado, y la cantidad de drones activos pertenecientes a la red, podrían no representar correctamente la complejidad del mundo real, ni todas las limitantes existentes en una infraestructura de telecomunicaciones, donde existen factores adicionales, como las interferencias que ocasionan las condiciones climáticas.

Finalmente, se configuró la cantidad de 1000 pasos de entrenamiento, es decir, 1000 ciclos de operación; Esta cantidad fue suficiente para capturar los patrones complejos por completo en un espacio de observación multidimensional. Para obtener una evaluación más precisa y eficiente de las capacidades de la implementación del modelo de red LSTM, junto con su optimización en escenarios semejantes a los del mundo real, se deben mejorar estos aspectos procesando una mayor cantidad de datos, ajustando parámetros en la arquitectura de red y diseñando un entorno de simulación más robusto.

### 3.2.2 Comparación con el modelo MLP

Se realizó el diseño de una red neuronal de tipo MLP, con el fin de mostrar una comparación entre LSTM, y así obtener resultados de su desempeño de manera visual con un mapa y su distribución de drones.

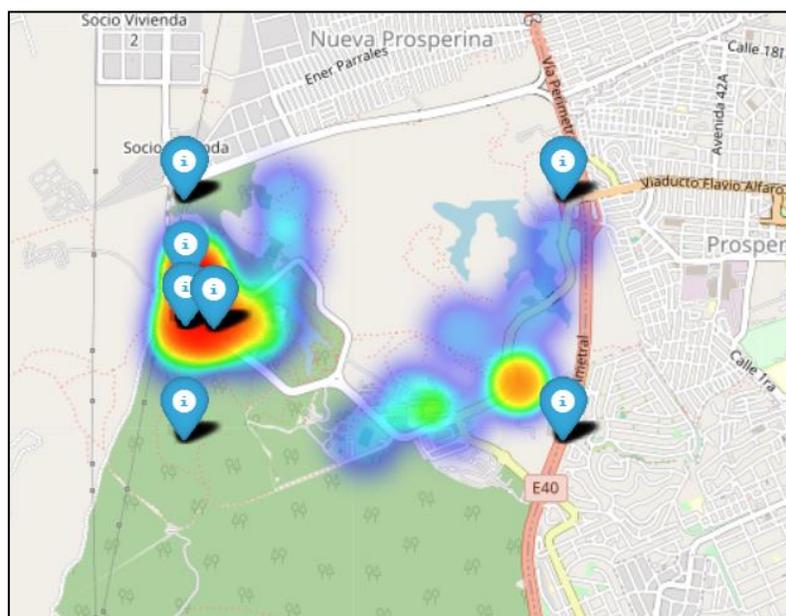


Ilustración 3.1 Mapa de despliegue utilizando MLP

En la *ilustración 3.1* se puede visualizar el mapa de despliegue de drones utilizando MLP, teniendo en cuenta que se debe dar una cobertura en zonas críticas, esta se muestra limitada y presenta concentraciones alteradas de los drones. Los drones tienden a realizar agrupaciones en puntos con alta densidad poblacional, sin embargo, no existe una redistribución efectiva hacia otras zonas del mapa donde se requiere atención, de esta manera, se observa una carencia de optimización de los recursos disponibles, dejando zonas sin cubrir.

Con respecto a la adaptabilidad, MLP es un modelo más sólido ya que procesa los datos de manera estática, esto hace que limite su capacidad de adaptación a situaciones dinámicas, lo cual es crucial para la redistribución de drones en tiempo real durante una situación de emergencia. Esto va de la mano con la exploración de rutas, ya que carece de memoria interna, es decir, no tendrá la capacidad de registrar un historial de ejecuciones y/o decisiones, reduciendo la eficiencia en el aprendizaje en este tipo de entornos.

Como se evidencia en la *ilustración 3.1*, existen drones que están agrupados de manera eficiente en áreas con alta densidad poblacional, dejando zonas críticas sin cobertura, justificando que este modelo no es capaz de gestionar de manera correcta la distribución en entornos más dinámicos y complejos.

### **3.3 Cobertura y eficiencia del despliegue**

La cobertura y eficiencia del despliegue de drones en una red de emergencia representa la base fundamental en el correcto funcionamiento del modelo implementado. Esta sección aborda la capacidad del modelo para optimizar la ubicación de drones en puntos críticos, administrando los recursos disponibles como el consumo de energía y la redistribución de los drones. Utilizando LSTM, este modelo garantiza una respuesta adaptativa y dinámica, optimizando el impacto en áreas con alta densidad poblacional.

#### **3.3.1 Cobertura de zonas prioritarias**

La cobertura del área de análisis se logra mediante la integración de una red neuronal LSTM, la cual utiliza varias entradas como datos históricos y en tiempo real con el fin de identificar y optimizar la ubicación de los drones. En la *ilustración 3.7* se muestra una distribución eficiente, con drones posicionados de una manera estratégica, no solo cubriendo puntos con alta densidad poblacional, sino áreas secundarias o con bajo nivel de prioridad.

El modelo LSTM se enfoca en procesar variables como densidad poblacional y rutas exploradas con anterioridad, esto permite el ajuste de las ubicaciones dinámicas conforme se desarrollen los posibles eventos. El uso de funciones como *calculate\_central\_point* y *redistribute\_drones* dentro del código implementado facilitan el entrenamiento y aprendizaje de nuestra red, haciendo posible un balance adecuado para que los drones cubran tanto las zonas críticas como sus alrededores.

### ***3.3.2 Distribución y consumo de baterías***

El consumo de energía es un factor crítico para el éxito del despliegue de la red. El modelo LSTM incorpora cálculos y funciones de la batería, así como la optimización de rutas, permitiendo que los drones operen de manera eficiente sin comprometer la cobertura del área de estudio.

Los drones comienzan con una capacidad de batería máxima, la cual está definida dentro del código como *max\_battery*, y esta se actualiza por cada paso realizado. Las rutas calculadas por el modelo LSTM minimizan los movimientos innecesarios, con el fin de garantizar un mejor consumo energético. Esto hace que tengan las baterías suficientes para explorar y poder redistribuirse hacia áreas de alta prioridad.

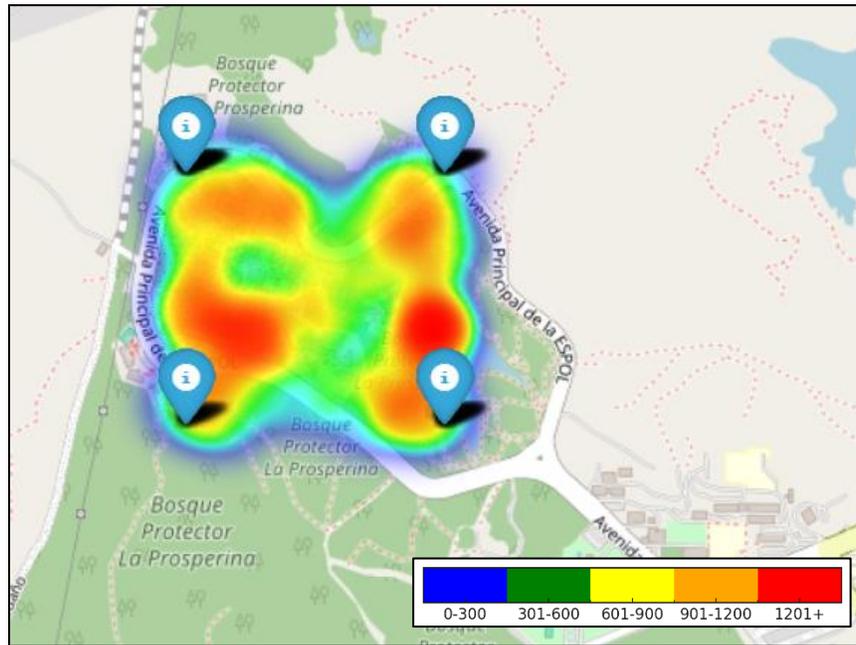
## **3.4 Interpretación de los mapas generados**

En este apartado, se van a mostrar mapas generados por el código implementado, con el fin de analizar los resultados de cada ilustración.

### ***3.4.1 Mapa de calor de zonas de emergencias***

Para proceder con la generación de un mapa de calor, se ingresó una base de datos con el fin de realizar pruebas simulando una situación de emergencia, tomando en cuenta variables como coordenadas (latitud, longitud y altitud), así como la densidad poblacional y el nivel de prioridad en zonas críticas.

Estas variables son primordiales para proceder con la limitación de la región de estudio, así como la exploración y ubicación óptima de los drones dependiendo de las zonas críticas.



**Ilustración 3.2** Mapa de calor de densidad poblacional y limitación de zona de estudio

En la *ilustración 3.2* se puede visualizar el mapa de calor representando zonas con mayor densidad poblacional de color naranja/rojo en un rango entre 900 hasta mayor a 1200 personas por área de estudio, y zonas con menor densidad poblacional con colores más claros, como el verde y/o azul, en un rango entre 0 a 600 personas por área analizada, verificando la base de datos el código ha generado correctamente las respectivas marcaciones de las zonas.

Por otro lado, puede visualizar una marcación por cada esquina, formando así un cuadrado el cual representa la zona de estudio y análisis, por la cual los drones realizarán sus respectivas misiones de exploración, recopilación y aprendizaje limitándose a estos cuatro puntos.

Para hacer posible este proceso, se tuvo que implementar y modificar el código con el fin de establecer los límites geográficos, latitud, longitud y altitud, los cuales se encuentra dentro de la función *DronesDeploymentEnv*. A su vez, se tiene que restringir la zona de estudio, para que los drones no pasen estos límites y sean penalizados, esto se puede evidenciar dentro del código en la función *step*.

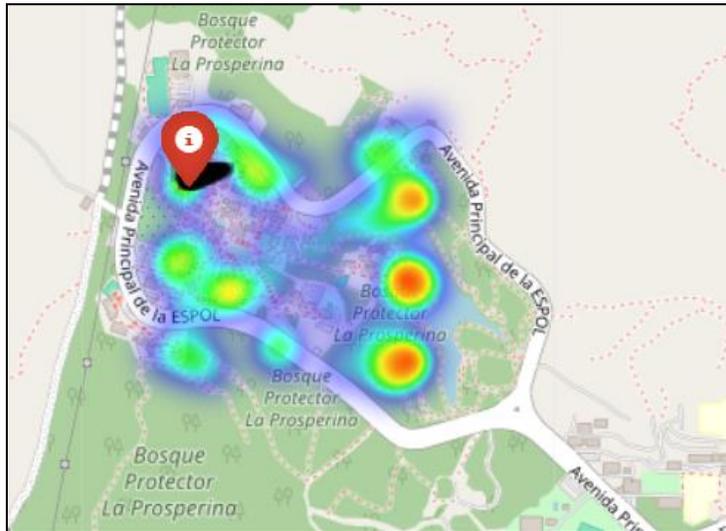
### ***3.4.2 Mapa del punto central de operaciones***

Antes de proceder con el despliegue óptimo de drones, se realiza un análisis del caso de estudio, por ende, el código limita las zonas, la cual forma parte del Campus ESPOL, a partir de eso, el código evalúa las zonas más afectadas según la base de datos otorgada en un inicio, donde se requieran más drones, para así establecer un punto central, el cual se lo ha denominado Base de Operaciones.

Esta base de operaciones es la que permitirá que estén todos los drones para proceder con el despliegue, y así ser el punto de comunicación para que todo esté conectado entre sí. Es importante determinar esto, ya que las situaciones pueden ser cambiantes, y normalmente se necesitaría de una base móvil para eventos de rescate, entonces es preferible evaluar el entorno para proceder con un punto inicial que desde allí empiece la exploración de los dispositivos aéreos no tripulados.

A continuación, se enlistará motivos por los cuales es crucial que en situaciones de emergencias se establezca un punto central de operaciones:

- **Coordinación eficiente:** Facilitará el manejo y gestión de drones, optimizando las rutas y sus misiones de exploración. Además, que permite centralizar una recopilación y análisis de datos para la toma de decisiones en tiempo real.
- **Reducción de distancias:** Los drones tendrán que explorar la zona como primera fase, y para optimizar los recursos, como batería y tiempo de vuelo, es necesario que se busquen posiciones estratégicas con respecto a la base de operaciones, reduciendo así el recorrido para cubrir zonas críticas.
- **Flexibilidad y adaptabilidad:** Establecer una base de operaciones que dependa de otras variables, como zonas críticas, densidad poblacional y nivel de prioridad, es importante ya que estos datos son cambiantes según la situación de emergencia, no habrá datos fijos. Es decir que, según datos recopilados antes, durante o después de la emergencia, habrá una nueva posición inicial de la base de operaciones para proceder con el despliegue, adaptándose dinámicamente al entorno.
- **Seguridad y Reabastecimiento:** Al ser una base sirve como estación para un posible mantenimiento preventivo y/o correctivo para los drones, así como su carga y recarga de baterías durante la operación en la emergencia.

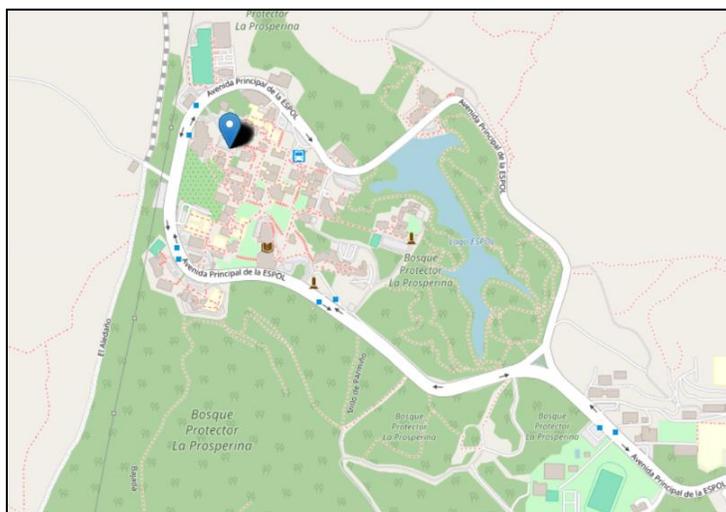


**Ilustración 3.3** Punto Central de Operaciones

En la **ilustración 3.3** se puede visualizar un punto, el cual fue denominado base de operaciones, este resultado fue producto del método *calculate\_central\_point*, el cual está definido dentro del código y asociada a la base de datos proporcionadas para realizar las respectivas pruebas. Es cálculo procesa los datos con el fin de identificar un punto central óptimo para posteriormente usarlo como base para el despliegue de los drones, esto fue logrado de la mano con las variables de densidad poblacional y nivel de prioridad, ya que se busca establecer una base de operaciones que esté lo más cercana a zonas con alto nivel de riesgo, por otro lado, se procura optimizar el recorrido de drones buscando la mejor ruta y estar siempre en comunicación entre ellos y la base de operaciones.

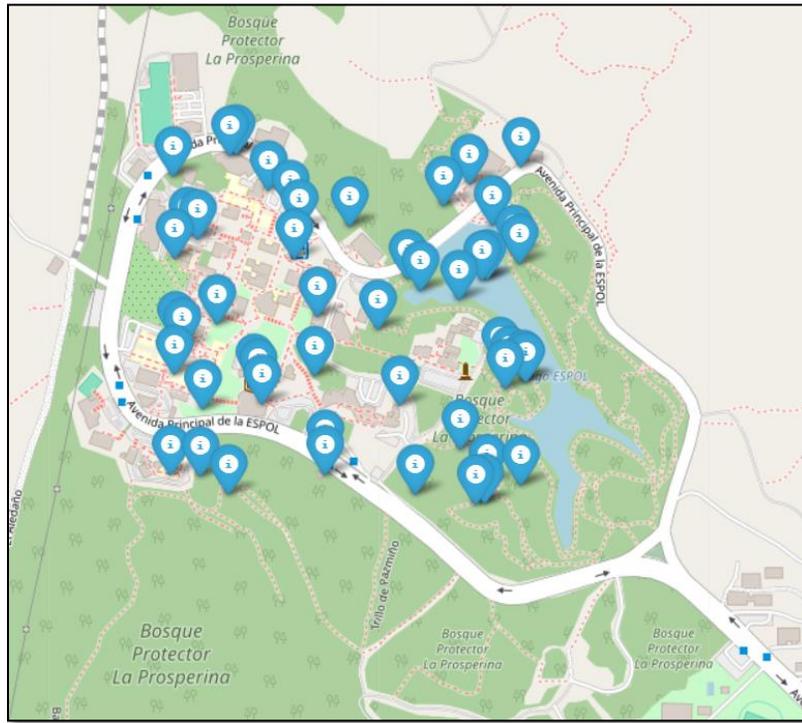
### 3.4.3 Mapa de posiciones iniciales y finales de drones

A continuación, se mostrarán dos ilustraciones las cuales muestran las posiciones iniciales y finales de los drones.



**Ilustración 3.4** Posición inicial de Drones

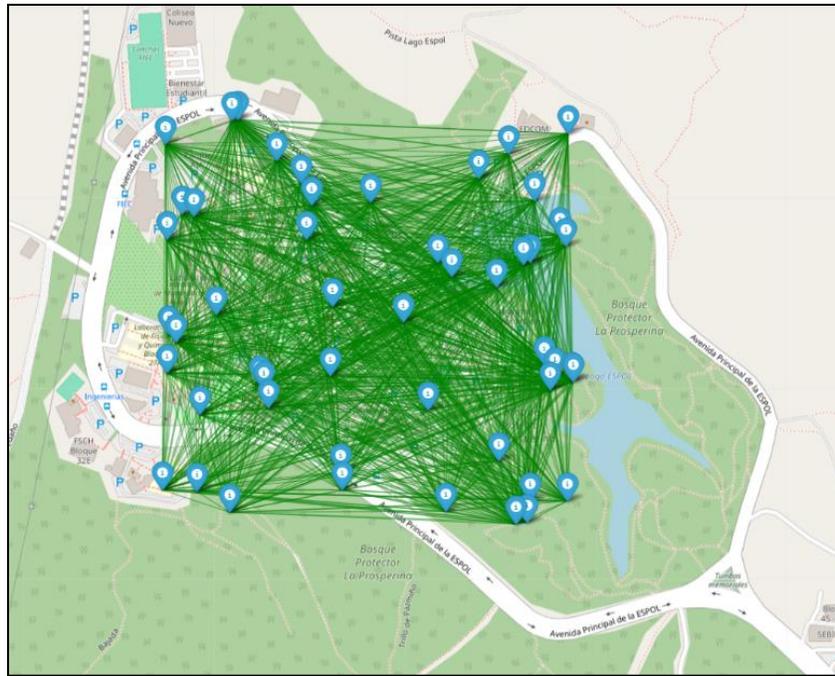
En la **ilustración 3.4** se puede visualizar un solo punto, este coincide con la base de operaciones, el cual fue calculado y analizado en el apartado 3.4.2. En este punto se van a encontrar los 50 drones, los cuales serán utilizados para la exploración de zonas en una situación de emergencia. Estos drones se van a encargar de analizar la zona de estudio para establecerse en los puntos críticos del mapa y enviarán información a la base de operaciones de los datos recopilados de su entrenamiento.



**Ilustración 3.5** Posición Final de Drones

En la **ilustración 3.5** se puede observar varios puntos en una parte del mapa de ESPOL, estos puntos representan los 50 drones. Esto fue posible debido al entrenamiento que se les dio a los drones en la fase de exploración. Más adelante se mostrará como estos puntos están netamente relacionado a los parámetros de nuestra base de datos, los cuales están vinculados a las variables de densidad poblacional y nivel de prioridad en una situación de emergencia.

### 3.4.4 Mapa de comunicación entre drones

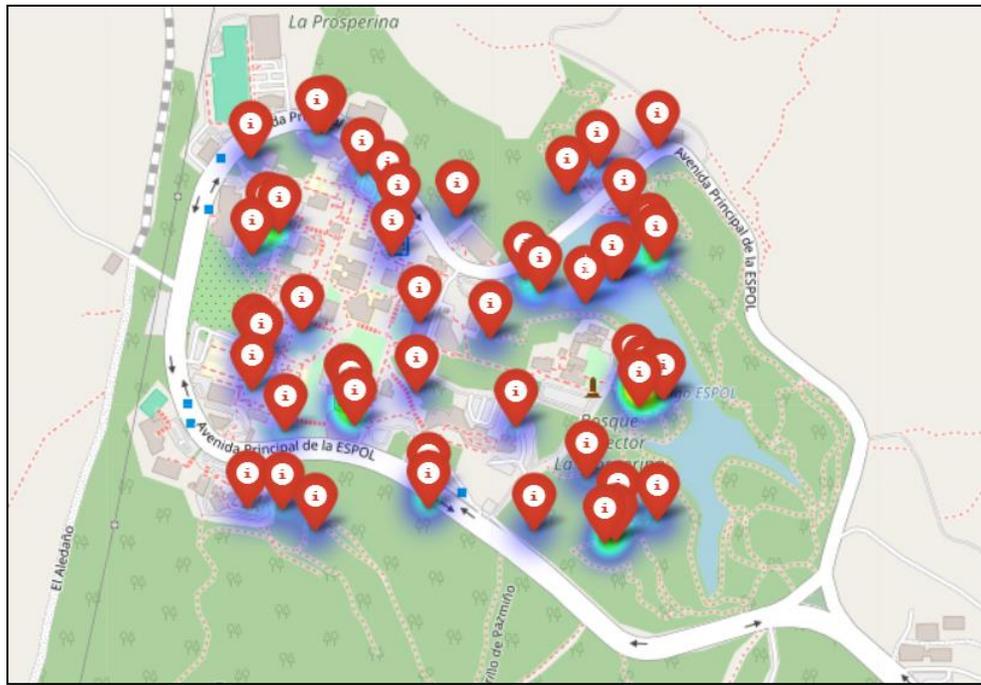


**Ilustración 3.6** Mapa de comunicación entre drones

Es necesario verificar que los drones estén conectados entre sí para proceder con el envío de información recopilada, y que esta llegue a la base de operaciones, en otras palabras, que la comunicación entre ellos sea efectiva. Esto se puede visualizar en la **ilustración 3.6**, donde muestran que los 50 drones están en comunicación, esto es representado por las líneas verdes, solo para confirmar que la conectividad es efectiva. Al realizar las respectivas pruebas, es importante revisar las especificaciones técnicas de los drones a utilizar, no obstante, para el despliegue de drones, se debe considerar obtener drones para situaciones de emergencias, ya que resistirían la gravedad del asunto.

Sin embargo, al investigar sobre los modelos de drones ligados con el radio de cobertura para una comunicación efectiva entre sí, el promedio de drones comerciales tiene un radio de 1Km hasta 5Km, entonces se puede afirmar que en el caso de estudio los drones cumplen con esto, ya que el radio máximo de nuestro análisis es de 1,5 Km, dentro de ese radio están dispersos los drones para recopilar y enviar información entre sí [16].

### 3.4.5 Mapa de calor de zonas de emergencias con posición final de drones



**Ilustración 3.7** Mapa de calor de zonas de emergencias con posición final de drones

En la **ilustración 3.7**, se evidencia una cobertura óptima y estratégica, optimizando los recursos disponibles y asegurando una respuesta efectiva a las necesidades del entorno. Se puede evidenciar como la posición de los drones está acorde al mapa de calor, priorizando zonas críticas con alta densidad poblacional, así como zonas secundarias, donde al menos se sitúa un dron para obtener información y de ser el caso comunicarse entre sí. El uso de la red LSTM aseguró que este despliegue sea exitoso, utilizando los diferentes tipos de variables y funciones dentro del entrenamiento de los drones para así obtener un óptimo despliegue y ubicación de drones en situaciones de emergencia.

# **Capítulo 4: Conclusiones y Recomendaciones**

## 4.1 Conclusiones

- La implementación del tipo de red neuronal LSTM con la red de emergencia conformada por drones demostró ser una alternativa sumamente viable para solventar las limitaciones de funcionamiento de las redes y sistemas de comunicaciones tradicionales durante y después de una situación de emergencia, debido al establecimiento de nodos autónomos y temporales, facilitando la comunicación entre los grupos de rescatistas y las personas afectadas de una localidad en específico y optimizando los tiempos de búsqueda. No obstante, la efectividad del despliegue es dependiente de los factores analizados en el código implementado, como la densidad poblacional, los datos geográficos y la capacidad de batería de los equipos, lo que denota más la importancia de establecer rutas y planes de despliegue estratégicos.
- La red LSTM fue clave para maximizar la eficiencia, adaptabilidad y escalabilidad de la red de emergencias. Su principal ventaja frente a los modelos tradicionales como MLP, es que permite un procesamiento de datos secuenciales y adaptarse a los cambios de condiciones en tiempo real, asegurando que el redespliegue dinámico de los drones teniendo en cuenta la evolución de los eventos de emergencia. Esto se traduce en una cobertura más amplia de las áreas de interés, un mejor uso de recursos energéticos y una mayor precisión en la ubicación del punto central de operaciones mediante la cual los equipos de la red están interconectados.
- El algoritmo Proximal Policy Optimization (PPO) resultó ser un buen complemento de la red LSTM, debido a las mejores obtenidas en la toma de decisiones de la red de emergencia. La función de recompensa basada en cubrir las áreas de mayor nivel de prioridad permitió guiar la operación de los drones hacia las áreas con un mayor número de usuarios, garantizando una distribución equitativa y efectiva de los recursos disponibles, además de permitir la identificación de patrones en la ocurrencia de emergencia, posibilitando que la red aprenda a anticipar y priorizar respuestas en situaciones críticas.
- Esta solución además de representar un avance importante en la optimización de despliegue de una red de drones de emergencia constituye un modelo replicable para distintos conceptos y escenarios en donde se requiera que un sistema aprenda cada vez que inicie sus operaciones. A nivel científico, contribuye al desarrollo de infraestructuras de telecomunicaciones mediante el uso de inteligencia artificial, mientras que, a nivel de operatividad, ofrece una alternativa de alta escalabilidad para gestionar situaciones de emergencia donde la cobertura de redes tradicionales es baja. Además, ofrece nuevas oportunidades en entornos educativos y de investigación, para poder mejorar la preparación y respuesta de las entidades de rescate ante una situación de emergencia.

## 4.2 Recomendaciones

- Es recomendable desarrollar y practicar procedimientos de emergencia específicos para mejorar la gestión de crisis de manera rápida y eficaz, incluyendo protocolos para fallos mecánicos, condiciones climáticas extremas y otras posibles limitantes variadas. El entrenamiento constante de la red y la implementación de nuevas funciones de emergencia del dron son esenciales para una mantener una respuesta efectiva.
- Utilizar metodologías de diseño basado en distintos modelos para la creación y el análisis de los sistemas antes de construir un entorno simulado. Si se requiere una implementación física, herramientas como Simulink son recomendables para desarrollar modelos de simulación para drones, ya que, facilitan las relaciones de simulaciones completas de sistemas, subsistemas y componentes junto con el algoritmo que desee implementar para la solución.
- Al momento de desarrollar un entorno de simulación, se recomienda tomar en cuenta factores con la precisión de la emulación de dinámicas físicas de lo drones, y las condiciones reales del escenario de emergencia. Esta recomendación tiene como objetivo mejorar la validez de los algoritmos implementados para el control de vuelo y las estrategias de misión sin riesgos asociados a las pruebas en el mundo real. Por ejemplo, para implantaciones futuras, se puede usar motores de videojuegos como Unity, que tiene la característica de facilitar la creación de escenarios virtuales detallados para misiones de búsqueda y rescate con enjambres de drones

## Bibliografía

- [1] Z. Khaled y H. Mcheick, «Estudios de casos de sistemas de comunicación en entornos hostiles: una revisión de enfoques, debilidades y limitaciones para mejorar la calidad del servicio,» *International Journal of Distributed Sensor Networks*, vol. 15, nº 2, 2019.
- [2] C. A. Terán Otero y I. F. Gómez Suárez , «Implementación de red de acceso a red de drones para comunicaciones de Emergencia,» Guayaquil, 2023.
- [3] S. C. Izaurieta F, «Redes Neuronales Artificiales,» Chile, 2000.
- [4] M. C. García, «Uso de drones para la entrega de desfibriladores en ataques cardíacos extrahospitalarios: una revisión sistemática,» Universidad de Valladolid, Facultad de Enfermería de Valladolid, Valladolid, 2023.
- [5] H. D, «EL MACHINE LEARNING A TRAVÉS DE LOS TIEMPOS, Y LOS APORTES A LA HUMANIDAD,» PEREIRA, Bogotá, 2018.
- [6] IBM, «¿Qué es una red neuronal? | IBM,» 16 Mayo 2023. [En línea]. Available: <https://www.ibm.com/es-es/topics/neural-networks>. [Último acceso: 29 Octubre 2024].
- [7] C. J, «Inteligencia artificial generativa,» de *Anales de la Real Academia de Doctores de España*, vol. 8, 2023, pp. 475-489.
- [8] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2ª edición ed., Prentice Hall, 1998.
- [9] J. M. Cantuña Guano, «MARCO DE EVALUACIÓN PARA MODELOS PREDICTIVOS,» Febrero 2024. [En línea]. Available: MARCO DE EVALUACIÓN PARA MODELOS PREDICTIVOS. [Último acceso: 04 Noviembre 2024].
- [10] S. B. E. & D. R. Geman, «Neural Networks and the Bias/Variance Dilemma,» *Neural Computation*, vol. 4, pp. 1-58, 1992.
- [11] S. G. C. L. & T. A. C. Lawrence, «Lessons in Neural Network Training: Overfitting May Be Harder than Expected,» de *Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 1997.
- [12] H. Sak, A. Senior y F. Beaufays, «Long short-term memory recurrent neural network architectures for large scale acoustic modeling,» ISCA, ISCA, 2014.

- [13] Z. M. Nayeri, T. Ghafarian y B. Javadi, «Application placement in Fog computing with AI approach: Taxonomy and a state of the art survey,» *Journal of network and computer applications*, vol. 185, nº 103078, p. 103078, 2021.
- [14] N. Angarita, «Edu.co,» [En línea]. Available: <https://repositorio.uniandes.edu.co/server/api/core/bitstreams/6d342d58-3229-457b-be54-2ace07198a50/content#:~:text=El%20algoritmo%20conocido%20como%20Optimizaci%C3%B3n,tomar%20decisiones%20en%20su%20entorno..> [Último acceso: 16 Octubre 2024].
- [15] J. de Lucio Fernández, «Estimación adelantada del crecimiento regional mediante redes neuronales LSTM,» *Investigaciones Regionales - Journal of Regional Research*, nº 49, pp. 45-64, 2021.
- [16] A. Iturri Torrens, «Configuración y test de enlace,» 26 Junio 2017. [En línea]. Available: <https://academica-e.unavarra.es/entities/publication/1d9e71c4-85cf-44bf-903d-c0d21426cf01>. [Último acceso: 15 Enero 2024].
- [17] UNNOBA, «¿Hacia un colapso mundial de las comunicaciones?,» *El Universitario*, Marzo 2013. [En línea]. Available: <https://eluniversitario.unnoba.edu.ar/2013/03/18/hacia-un-colapso-mundial-de-las-comunicaciones/>. [Último acceso: 10 Octubre 2024].

## Anexos

A continuación, se presenta el código ajustado e implementado para el desarrollo de la materia integradora.

```
import pandas as pd
import numpy as np
import folium
from folium.plugins import HeatMap, TimestampedGeoJson
from datetime import datetime, timedelta
import gymnasium
from gymnasium import spaces
from sb3_contrib import RecurrentPPO
from stable_baselines3.common.vec_env import DummyVecEnv

# Cargar los datos sin normalización
df_original = pd.read_csv(r'C:\Users\bpiya\Escritorio\U ESPOL\TESIS\Code\espol_locations.csv')
df = df_original.copy()

# Clase del entorno personalizado
class DroneDeploymentEnv(gymnasium.Env):
    def __init__(self, data, num_drones=50, active_drones=50, max_steps=500, max_battery=100,
communication_radius=0.015):
        super(DroneDeploymentEnv, self).__init__()
        self.data = data
        self.num_drones = num_drones
        self.active_drones = active_drones
        self.max_steps = max_steps
        self.max_battery = max_battery
        self.communication_radius = communication_radius
        self.current_step = 0

        # Calcular el punto central de operaciones
        self.lat_centro, self.lon_centro, self.alt_centro = self._calculate_central_point()

        # Inicializar límites de latitud, longitud y altitud basados en los datos
        self.lat_min = self.data['Latitude'].min()
        self.lat_max = self.data['Latitude'].max()
        self.lon_min = self.data['Longitude'].min()
        self.lon_max = self.data['Longitude'].max()
        self.alt_min = self.data['Altitude'].min()
        self.alt_max = self.data['Altitude'].max()

        # Inicializar posiciones de drones y baterías
        self.drones_position = self._initialize_drone_positions()
```

```

self.drone_batteries = np.full(self.num_drones, self.max_battery)
self.drone_routes = {i: [] for i in range(self.num_drones)}
for i in range(self.num_drones):
    self.drone_routes[i].append(self.drones_position[i].tolist())

# Espacios de acción y observación
self.action_space = spaces.MultiDiscrete([7] * self.active_drones)
self.observation_space = spaces.Box(low=-np.inf, high=np.inf, shape=(self.active_drones,
5), dtype=np.float32)

def _calculate_central_point(self):
    max_density_area = self.data.loc[self.data['Population Density'].idxmax()]
    lat_centro, lon_centro, alt_centro = max_density_area['Latitude'],
max_density_area['Longitude'], max_density_area['Altitude']
    return lat_centro, lon_centro, alt_centro

def _initialize_drone_positions(self):
    return np.array([[self.lat_centro, self.lon_centro, self.alt_centro]] * self.num_drones)

def step(self, actions):
    rewards = []
    for i, action in enumerate(actions):
        # Actualiza la posición y la batería de los drones
        prev_position = self.drones_position[i].copy()
        if action == 1:
            self.drones_position[i][0] = min(self.drones_position[i][0] + 0.00025,
self.lat_max)
        elif action == 2:
            self.drones_position[i][0] = max(self.drones_position[i][0] - 0.00025,
self.lat_min)
        elif action == 3:
            self.drones_position[i][1] = min(self.drones_position[i][1] + 0.00025,
self.lon_max)
        elif action == 4:
            self.drones_position[i][1] = max(self.drones_position[i][1] - 0.00025,
self.lon_min)
        elif action == 5:
            self.drones_position[i][2] = min(self.drones_position[i][2] + 10, self.alt_max)
        elif action == 6:
            self.drones_position[i][2] = max(self.drones_position[i][2] - 10, self.alt_min)

        # Actualiza las rutas de cada dron
        self.drone_routes[i].append(self.drones_position[i].tolist())

```

```

rewards.append(self.data.iloc[self._get_nearest_zone_index(self.drones_position[i])]['Population
Density'])
return self._get_observation(), sum(rewards), self.current_step >= self.max_steps, False,
{}

```

```

def _get_population_density(self, lat, lon, alt):
    distances = np.sqrt(
        (self.data['Latitude'] - lat) ** 2 +
        (self.data['Longitude'] - lon) ** 2 +
        (self.data['Altitude'] - alt) ** 2
    )
    closest_index = distances.idxmin()
    return self.data.iloc[closest_index]['Population Density']

```

```

def _get_observation(self):
    obs = []
    for i, drone_pos in enumerate(self.drones_position[:self.active_drones]):
        density = self._get_population_density(*drone_pos)
        battery = self.drone_batteries[i]
        obs.append([*drone_pos, density, battery])
    return np.array(obs)

```

```

def _get_nearest_zone_index(self, position):
    distances = np.sqrt(
        (self.data['Latitude'] - position[0]) ** 2 +
        (self.data['Longitude'] - position[1]) ** 2
    )
    return distances.idxmin()

```

```

def reset(self, seed=None, **kwargs):
    self.drones_position = self._initialize_drone_positions()
    self.current_step = 0
    self.drone_routes = {i: [self.drones_position[i].tolist()] for i in
range(self.num_drones)}
    return self._get_observation(), {}

```

```

def get_communication_pairs(self):
    pairs = []
    for i in range(self.num_drones):
        for j in range(i + 1, self.num_drones):
            if np.linalg.norm(self.drones_position[i][:2] - self.drones_position[j][:2]) <=
self.communication_radius:
                pairs.append((i, j))
    return pairs

```

```

def redistribute_drones(self):
    """Redistribuye los drones en función de la densidad poblacional después de la
    exploración."""
    zones_density = []
    for i in range(len(self.data)):
        density = self.data.iloc[i]['Population Density']
        lat, lon, alt = self.data.iloc[i]['Latitude'], self.data.iloc[i]['Longitude'],
self.data.iloc[i]['Altitude']
        zones_density.append((density, [lat, lon, alt]))

    zones_density.sort(reverse=True, key=lambda x: x[0])
    num_drones_per_zone = max(1, self.num_drones // len(zones_density))
    new_positions = []

    for density, zone in zones_density:
        for _ in range(num_drones_per_zone):
            if len(new_positions) < self.num_drones:
                new_positions.append(zone)
            else:
                break

    self.drones_position = np.array(new_positions[:self.num_drones])

# Funciones para generación de mapas
def generar_mapa_de_calor(data, drones_positions, output_file):
    mapa = folium.Map(location=[data['Latitude'].mean(), data['Longitude'].mean()], zoom_start=13)
    HeatMap([[row['Latitude'], row['Longitude'], row['Population Density']] for _, row in
data.iterrows()], radius=15).add_to(mapa)
    for i, pos in enumerate(drones_positions):
        folium.Marker(location=[pos[0], pos[1]], popup=f"Dron {i}",
icon=folium.Icon(color="red")).add_to(mapa)
    mapa.save(output_file)

def generar_mapa_de_exploracion(data, drone_routes, output_file):
    mapa = folium.Map(location=[data['Latitude'].mean(), data['Longitude'].mean()], zoom_start=13)
    for dron_id, ruta in drone_routes.items():
        folium.PolyLine([[pos[0], pos[1]] for pos in ruta], color='blue').add_to(mapa)
    mapa.save(output_file)

def generar_mapa_con_comunicacion(data, drone_routes, communication_pairs, output_file,
lat_centro, lon_centro):
    mapa = folium.Map(location=[data['Latitude'].mean(), data['Longitude'].mean()], zoom_start=13)
    for (i, j) in communication_pairs:

```

```

        folium.PolyLine([drone_routes[i][-1][:2], drone_routes[j][-1][:2]],
color='green').add_to(mapa)
        mapa.save(output_file)

def generar_mapa_animado(data, drone_routes, output_file):
    """
    Genera un mapa animado con las rutas de los drones utilizando TimestampedGeoJson.

    Args:
        data: DataFrame con los datos de la simulación.
        drone_routes: Diccionario con las rutas de cada dron.
        output_file: Nombre del archivo HTML donde se guardará el mapa.
    """
    # Crear el mapa centrado en la región de interés
    mapa = folium.Map(location=[data['Latitude'].mean(), data['Longitude'].mean()], zoom_start=13)

    features = []
    start_time = datetime.now()

    for dron_id, ruta in drone_routes.items():
        for step, pos in enumerate(ruta):
            # Asegurarse de que las coordenadas estén en el formato [longitud, latitud]
            timestamp = (start_time + timedelta(seconds=step)).isoformat()
            features.append({
                "type": "Feature",
                "geometry": {
                    "type": "Point",
                    "coordinates": [pos[1], pos[0]] # [longitud, latitud]
                },
                "properties": {
                    "time": timestamp,
                    "popup": f"Dron {dron_id}<br>Lat: {pos[0]:.6f}<br>Lon: {pos[1]:.6f}<br>Alt:
{pos[2]:.2f} m",
                    "style": {"color": "blue", "icon": "circle", "iconstyle": {"fillColor":
"blue", "radius": 5}}
                }
            })

    # Construir el GeoJSON animado
    geojson = {
        "type": "FeatureCollection",
        "features": features
    }

    TimestampedGeoJson(

```

```

    geojson,
    transition_time=200, # Duración de la transición entre puntos (en milisegundos)
    period="PT1S" # Intervalo entre puntos (en segundos)
).add_to(mapa)

# Guardar el mapa como archivo HTML
mapa.save(output_file)

# Crear el entorno y entrenar el modelo
env = DummyVecEnv([lambda: DroneDeploymentEnv(data=df)])
model = RecurrentPPO("MlpLstmPolicy", env, verbose=1)
model.learn(total_timesteps=10000)

# Generar y guardar mapas después de la exploración
observation = env.reset()
communication_pairs = env.get_attr('get_communication_pairs', 0)[0]()
lat_centro, lon_centro = env.get_attr('lat_centro', 0)[0], env.get_attr('lon_centro', 0)[0]

generar_mapa_de_calor(df_original, [route[-1] for route in env.get_attr('drone_routes',
0)[0].values()], "mapa_drones_final.html")
generar_mapa_de_exploracion(df_original, env.get_attr('drone_routes', 0)[0],
"mapa_exploracion_fase1.html")
generar_mapa_con_comunicacion(df_original, env.get_attr('drone_routes', 0)[0],
communication_pairs, "mapa_drones_comunicacion.html", lat_centro, lon_centro)
generar_mapa_animado(df_original, env.get_attr('drone_routes', 0)[0], "mapa_animado_drones.html")

# Redistribuir drones y generar el mapa final con drones en posiciones óptimas
env.get_attr('redistribute_drones', 0)[0]()
generar_mapa_de_calor(df_original, env.get_attr('drones_position', 0)[0],
"mapa_drones_optimos.html")

```