



**UAI**  
UNIVERSIDAD ADOLFO IBÁÑEZ

# **Enjambre de Drones para Detección de Embarcaciones Nocivas dentro de una Zona Marítima de Pesca Artesanal**

**Estudiantes:**

Eduardo Carrasco

Argenis Chirinos

Gustavo Jara

**Profesor:**

Enrique Canessa

**Curso:**

Simulación Basada en Agentes. MIA, 2022

# Tabla de contenido

<b>Tabla de contenido</b>	<b>1</b>
Introducción y descripción del fenómeno:	2
Propuesta de solución:	3
Capítulo 2: Descripción del modelo (ABM):	3
Características del escenario y limitaciones:	4
Mundo:	4
Sensores:	4
Supuestos:	5
Características de la Arquitectura utilizada:	7
Parámetros de entrada por defecto:	8
Métricas de salida:	8
Descripción de la implementación en NetLogo:	9
Representación visual de los objetos del entorno y agentes en NetLogo:	9
Instancia de un modelo configurado con 5 drones:	10
Elementos en la interfaz de NetLogo:	10
Elementos de control:	10
Variables de entrada:	10
Variables de salida:	11
Capítulo 3: Experimentos con el modelo:	13
Selección de Variables de entrada:	14
Selección de la variable de salida (estudio):	14
Desarrollo del Modelo 2 <sup>2</sup> :	15
Lecturas obtenidas durante los experimentos:	16
Tabla de respuestas:	16
Interpretación de los resultados:	16
Análisis de Varianza (ANOVA), Modelo 2 <sup>2</sup> :	19
Desarrollo del Modelo 2 <sup>3</sup> :	20
Lecturas obtenidas durante los experimentos:	20
Tabla de respuestas:	21
Interpretación de los resultados:	21
Análisis de Varianza (ANOVA), Modelo 2 <sup>3</sup> :	23
Capítulo 4: Propuesta de Validación del Modelo:	24
Verificación del modelo:	24
Validación del modelo:	25
Conclusiones:	29
Referencias bibliográficas utilizadas:	30

## I. Introducción y descripción del fenómeno:

El sector pesquero chileno se encuentra dividido por los siguientes subsectores: Pesca Artesanal, Pesca Industrial y Subsector Acuicultura<sup>1</sup>; con respecto a la pesca artesanal, cabe destacar que el número de pescadores artesanales inscritos en el Registro Pesquero Artesanal (R.P.A.) al año 2021 es de aproximadamente 100.000<sup>2</sup> personas, donde Chile se ubica en el 8vo lugar del mundo por volumen extractivo, representando una alta operación de embarcaciones a lo largo de más de 3 millones de km<sup>2</sup> (4 veces el territorio nacional) de mar territorial y zona económica exclusiva, que permiten sustentar actividades marinas.

La sustentabilidad de los recursos pesqueros es un desafío permanente, por lo cual, la Armada de Chile realiza diversos esfuerzos para combatir la pesca ilegal y fiscalizar a las embarcaciones que no cumplan con la normativa nacional o internacional, amenazando directamente a la conservación de las poblaciones de peces y ecosistemas marinos, tal como lo describe la Ley General de Pesca.

Para cumplir lo anterior, la marina ha construido 4 unidades (OPV<sup>3</sup>), capaces de controlar la zona económica exclusiva (ZEE), que cumplen misiones de escolta, vigilancia e inspección; este buque multipropósito cuenta con un alto grado de flexibilidad y es capaz de efectuar fiscalización a las actividades de pesca utilizando 2 embarcaciones (RHIB<sup>4</sup>) propias.

Cabe destacar que el volumen de fiscalizaciones al año que realiza la Armada a pesqueros artesanales es cercano a las 15.000 embarcaciones, según la cuenta pública del año 2021<sup>5</sup>.

El modo de operación de la unidad para cada actividad de fiscalización consiste en desplegar ambos botes con dotaciones, los cuales, deben ser capaces de seleccionar visualmente a cual de todas las embarcaciones fiscalizar, viéndose enfrentados a los siguientes escenarios:

a) Seleccionar una embarcación a fiscalizar.

---

<sup>1</sup> DIRECTEMAR (2022), Ley General de Pesca y Acuicultura, D.S. N° 430. [En línea] Disponible en: [https://www.directemar.cl/directemar/site/docs/20170125/20170125102709/tm\\_066\\_actualizada\\_10\\_mayo\\_2022.pdf](https://www.directemar.cl/directemar/site/docs/20170125/20170125102709/tm_066_actualizada_10_mayo_2022.pdf), pág. 14.

<sup>2</sup> SERNAPESCA (2021). Anuario Estadístico de Pesca y Acuicultura 2021. [En Línea] Disponible en: [http://www.sernapesca.cl/sites/default/files/2021\\_030201\\_subsector\\_artesanal.pdf](http://www.sernapesca.cl/sites/default/files/2021_030201_subsector_artesanal.pdf), pág. 1.

<sup>3</sup> OPV: Offshore Patrol Vessel (Patrullero de Alta Mar).

<sup>4</sup> RHIB: Rigid Hull Inflate Boat (Bote Inflable de Quilla Rígida).

<sup>5</sup> DIRECTEMAR (2021), Cuenta Pública de Fiscalización Pesquera del año 2021. [en línea] Disponible en: [https://www.directemar.cl/directemar/site/docs/20170125/20170125102709/tm\\_066\\_actualizada\\_10\\_mayo\\_2022.pdf](https://www.directemar.cl/directemar/site/docs/20170125/20170125102709/tm_066_actualizada_10_mayo_2022.pdf), pág. 25.

- b) Considerar que el desplazamiento es lento y a nivel de superficie (nivel del mar), en el cual se tiene poca visibilidad para reconocer las actividades de pesca.
- c) Que existe un componente humano que afecta el nivel de decisión.
- d) Que el desplazamiento en malas condiciones de mar afecta a las dotaciones de los botes, los cuales se ven enfrentados a largos trayectos con mareo a causa del movimiento propio del mar.

Lo anterior, representa sólo parte de la cantidad de escenarios propios de una fiscalización pesquera, por lo cual, se puede reconocer el siguiente problema:

“El proceso de ejecución de la fiscalización pesquera a través de botes, es ineficiente porque utiliza medios con poca capacidad y tiene un componente humano que decide la embarcación a fiscalizar en base a información incompleta”.

#### A. Propuesta de solución:

Para resolver el problema enunciado anteriormente, se propone desarrollar un sistema basado de agentes que permita a través de un enjambre de drones del tipo UAV<sup>6</sup>, efectuar un reconocimiento fotográfico de cada embarcación en actividad de pesca, de manera de seleccionar las que representan **nocividad** o que posean características particulares o de interés, para posteriormente utilizar un bote fiscalizador.

Lo anterior, permitirá disminuir los tiempos de reconocimiento de embarcaciones y aumentar la información disponible para la toma de decisiones.

## II. Capítulo 2: Descripción del modelo (ABM):

Para el modelo propuesto, se plantea un escenario en el que existan embarcaciones con dos características principales: **NO nocivas** y **nocivas**.

Estas características serán determinadas a través de un reconocimiento visual utilizando drones (UAV), controlados a través de una estación incorporada en el buque fiscalizador.

Para poder determinar la posición de las embarcaciones, se utilizará un radar<sup>7</sup> que permite identificar cada embarcación y se encuentra limitado a su radio de cobertura.

La estación controladora incorporada, utiliza los datos del radar y asigna rutas de recorrido a los drones que efectuarán el reconocimiento visual para determinar su

---

<sup>6</sup> UAV: Unmanned Aerial Vehicle: aeronave no tripulada que utiliza guiado autónomo o radio controlado (teleguiado), para su desplazamiento a través de una estación terrestre, aérea o marina.

<sup>7</sup> Radar: Sistema que utiliza ondas electromagnéticas para medir distancias, altitudes, rumbos y velocidades de elementos (embarcaciones, barcos, aeronaves, entre otros).

nivel de nocividad, lo que es reportado por el dron al buque fiscalizador como agente controlador.

#### A. Características del escenario y limitaciones:

##### Mundo:

1. Escenario de  $n \times n$  dimensiones con límites verticales y horizontales (NO toroide).
2. Escenario de un mundo en dos dimensiones, por lo cual, no existe un componente Z relacionado a la altitud de vuelo de cada dron.
3. Contiene “ $m$ ” que representa la cantidad de embarcaciones, con el siguiente dominio [5 - 200], cada objeto es simbolizado con una etiqueta del tipo velero. Esta cantidad es seleccionable dentro del modelo.
4. Contiene “ $o$ ” que representa el porcentaje del total de embarcaciones que son reconocidas como nocivas, con un dominio [0 - 100 %]. Este porcentaje es seleccionable dentro del modelo
5. Contiene “ $q$ ” que representa la cantidad de drones UAV, con un dominio [1 - 5] que recorrerán el mundo dentro del radio de cobertura del radar.
6. Contiene un radio de cobertura de radar representado por una línea circular punteada con una distancia “ $r$ ” de la unidad controladora.
7. Contiene colores de etiquetado con las siguientes consideraciones:
  - a. Verde : Embarcación reconocida como No Nociva.
  - b. Rojo : Embarcación reconocida como Nociva.
  - c. Gris : Embarcación fuera del radio de cobertura de radar
8. Contiene una unidad controladora (agente controlador) que por sus dimensiones, limita la posición de las embarcaciones, es decir, no se pueden sobreponer. Además, posee posiciones específicas de lanzamiento del dron alrededor de su estructura.

##### Sensores:

1. Para la torre de control (estación controladora):
  - a. Radar que permite detectar todos los objetos del mundo, pero no es capaz de identificar si son nocivos o no nocivos.
  - b. Emitter de radiofrecuencia para retransmitir la posición GPS (determinada en demarcación y distancia<sup>8</sup>) de las embarcaciones (objetos), a los drones.
  - c. Receiver de radiofrecuencia para recibir información respecto de la nocividad de cada objeto.
  - d. Radio de cobertura de las radiofrecuencias asociadas al traspaso de información dron/antena es similar al radio de cobertura de

---

<sup>8</sup> Esto representa un vector con posiciones X e Y, de manera de determinar distancias utilizando reglas trigonométricas.

detección del radar, por lo cual, el drone no volará más allá de dicho radio.

- e. La estación controladora se encuentra estacionaria, sin rumbo ni velocidad.
2. Para los drones:
- a. Propulsores (4) con motores rotacionales, UAV de ala rotatoria (En Netlogo solo se dibujarán).
  - b. Considera un tiempo de operación determinado.
  - c. Considera un tiempo de recarga determinado.
  - d. Considera un margen de seguridad de 90% y un nivel mínimo de recarga para operar de 75%.
  - e. Giroscopio y acelerómetro para controlar la estabilidad y giros (En Netlogo esto no se simulará).
  - f. Radio Altimetro, para reconocer la altitud de vuelo del drone (En Netlogo esto no se simulará).
  - g. GPS para reconocer la posición y girocompás como sistema de navegación inercial para orientarse y moverse a cada objeto según instrucción de la torre de control (En Netlogo el GPS se simulará con coordenadas X e Y, por lo que no se requeriría la demarcación con respecto del objetivo a alcanzar).
  - h. Cámara para determinar la nocividad del objeto en base a características, para este modelo, serán seleccionados en forma aleatoria (En Netlogo se simulará al momento en que el drone se posiciona sobre el objeto).
  - i. Receiver de radiofrecuencia para recibir instrucciones desde la torre de control.
  - j. Emitter de radiofrecuencia para enviar objetivos completados y nivel de nocividad de cada objeto.

#### Supuestos:

- 1. Para el mundo:
  - a. Es plano por lo que no considera altura (sólo componente X e Y).
  - b. No existen condiciones meteorológicas que afecten a los drones, embarcaciones ni torre de control.
  - c. El mundo es una superficie plana, por lo tanto, la determinación de las distancias es trigonométrica, no geodésica<sup>9</sup>.
- 2. Para los objetos:
  - a. Existen 2 tipos de objetos dentro del radio de cobertura del radar: nocivos y NO nocivos.
  - b. Fuera del radio de cobertura, todos los objetos son desconocidos.

---

<sup>9</sup> Geodésica: Cálculo en un espacio esférico 3D, que permite determinar distancias considerando la curvatura terrestre.

- c. Todos los objetos (embarcaciones), poseen una velocidad (no superior a la del drone) y rumbo aleatorio.
  - d. No hay embarcaciones que se encuentren estacionarias (sin rumbo ni velocidad).
  - e. Sólo pueden ser reconocidos (etiquetados), por un drone.
  - f. Una vez reconocidos, permanecen estacionarios (controlados), para efectos de simulación.
3. Para la torre de control:
- a. Conoce la ubicación de los objetos en todo el mundo a través de su radar, pero desconoce su nocividad.
  - b. El radio de cobertura de radar es un círculo de radio determinado que no se ve afectado por la curvatura terrestre ni por condiciones meteorológicas.
  - c. El radio de cobertura de los emitter/receiver es un círculo de radio determinado de igual distancia al radio de cobertura de radar, que no se ve afectado por la curvatura terrestre ni por condiciones meteorológicas.
  - d. Una vez detectado un objeto nocivo (embarcación nociva), no se toman acciones, solo se identifican con un etiquetado rojo y se genera un listado con su posición.
  - e. Una vez detectado un objeto no nocivo (embarcación no nociva), no se toman acciones, sólo se identifican con un etiquetado verde y se genera un listado con su posición.
  - f. Ningún elemento posee fallas.
4. Para los drones:
- a. Tienen limitación de energía de 13.3 min de operación, para efecto de simulación esto es equivalente a 800 ticks (100 %), o 1/800 de su capacidad por cada tick.
  - b. Tiene una alarma de operación seteada al 10% de la capacidad remanente de operación (80 ticks - 1 minuto 20 segundos), donde debe volver a la estación controladora para recarga.
  - c. Tiene un tiempo de recarga de 2 min, para efecto de simulación esto equivale a 120 ticks (100%), o 1/120 de recarga por cada tick.
  - d. Considerar que al recargar un 75%, equivalente a 1 min 30 segundos, está capacitado para operar.
  - e. No poseen límites de operación en altitud (no se simula este componente).
  - f. Son 100% estables, no presentando fallas en su operación o desplazamiento.
  - g. No son afectados por condiciones meteorológicas (no simuladas).
  - h. El emitter y el receiver nunca fallan.

- i. Inician su desplazamiento desde la estación controladora (unidad fiscalizadora).
- j. Todos se mueven a la misma velocidad.
- k. Reconocen un objeto nocivo/no nocivo, una vez que sobrevuelan su posición.
- l. El tiempo de reconocimiento es instantáneo.
- m. Determinan el próximo objetivo a reconocer en base a la distancia más cercana.
- n. Una vez finalizado el reconocimiento, vuelven a la estación controladora.
- o. No vuelan más allá del radio de cobertura de radar/emitter/receiver.

## B. Características de la Arquitectura utilizada:

La arquitectura utilizada es **subsunción** con drones **reactivos**, la que permite acoplar la información sensorial (percepción - *"Information Sensor"*), dependiente de la posición del objeto, para organizar los comportamientos a través de capas por niveles de competencia.

En la siguiente figura, se muestra un ejemplo de la arquitectura de subsunción:

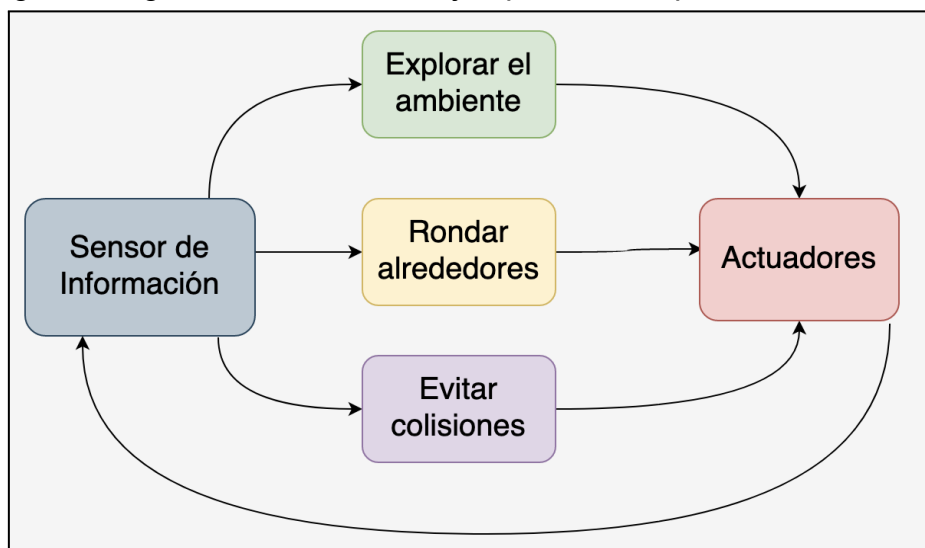


Figura N° 1: Diagrama básico de una arquitectura de subsunción.

En particular para el modelo en estudio, se cuenta con las siguientes capas:

1. Percepción (senzar):
  - a. Captura de los objetos del mundo.
2. Planeamiento (controlador del modelo):
  - a. Analizar la ubicación de los objetos y generar rutas de recorrido para cada drone, en base al más cercano.
3. Ejecución de tareas (actuadores):



- a. La torre central instruye a cada drone con los objetivos a realizar, donde cada objetivo se basa en dirigirse a un objeto e identificar su nocividad.
- b. A medida que los drones ejecutan cada objetivo, se genera un listado de objetos nocivos y se le entrega un siguiente objetivo.
- c. Cuando no quedan más objetos por identificar, el drone vuelve a la estación controladora.
- d. Al finalizar, la torre de control genera reporte de objetos nocivos y su posición (esto permite que la estación controladora envíe el bote con dotación a fiscalizar las embarcaciones nocivas).

#### Parámetros de entrada por defecto:

1. Cantidad de drones : [1 - 5], default 3.
2. Cantidad de objetos totales : [10 - 200], default 50.
3. Probabilidad objeto nocivo : [0% - 100%], default 10%.
4. Probabilidad de pérdida : [0% - 100%], default 1%.
5. Limitación de Energía : [Sí - No], default Sí.
6. Nivel de Recarga de batería : Todos default, con un nivel aleatorio entre [50 % - 100%].

#### Métricas de salida:

1. Distancia Recorrida [metros]: representa la cantidad de metros recorridos por cada drone.
2. Distancia Recorrida Total [metros]: representa la cantidad total de metros recorridos por el enjambre de drones.
3. Promedio de Distancia Recorrida [metros]: representa el promedio de la cantidad de metros recorridos por los drones.
4. Desviación Estándar de Distancia Recorrida: representa la desviación estándar respecto a la cantidad de metros recorridos por los drones, permitiendo verificar la variación/dispersión.
5. Tiempo de Vuelo [segundos]: representa la cantidad de segundos de operación durante la operación de cada drone.
6. Tiempo Total de Vuelo [segundos]: representa la cantidad total de segundos de operación de todos los drones.
7. Promedio del Tiempo de Vuelo [segundos]: representa el promedio de la cantidad total de segundos de operación de todos los drones.
8. Desviación Estándar del Tiempo de Vuelo: representa la desviación estándar respecto a la cantidad de segundos de operación de todos los drones, permitiendo verificar la variación/dispersión.
9. Objetivos Alcanzados por Drone [cantidades]: representa la cantidad de objetivos alcanzados por los drones.
10. Total de Objetivos [cantidades]: total de objetivos alcanzados.
11. Promedio Objetivos [cantidades]: promedio de objetivos alcanzados por cada drone.

12. Desviación Estándar de la Cantidad de Objetivos Alcanzados: desviación estándar de la cantidad de objetivos alcanzados.
13. Consumo [watts]: watts consumidos por los drones.
14. Total Objetivos [watts]: total de watts consumidos por los drones.
15. Promedio Consumo: [watts]: watts consumidos en promedio por los drones.
16. Desviación Estándar del Consumo [watts]: desviación estándar del consumo energético de los drones.
17. Tiempo Medio de Clasificación: cada cuanto tiempo, en promedio, los drones alcanzan objetivos.

### C. Descripción de la implementación en NetLogo:

A continuación, se describen elementos relacionados directamente a la implementación en NetLogo del modelo.

Representación visual de los objetos del entorno y agentes en NetLogo:


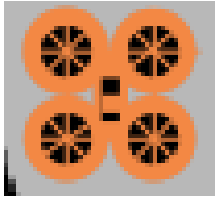
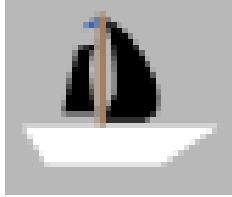
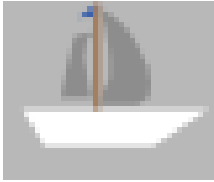
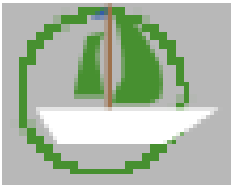
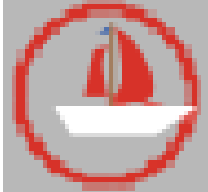
<p>Torre de control</p> 	<p>Drone (agente)</p> 	<p>Embarcación dentro del radar</p> 
<p>Embarcación fuera del radar</p> 	<p>Embarcación <b>no</b> nociva</p> 	<p>Embarcación nociva</p> 

Figura N° 2: Representación visual de los objetos.

Instancia de un modelo configurado con 5 drones:

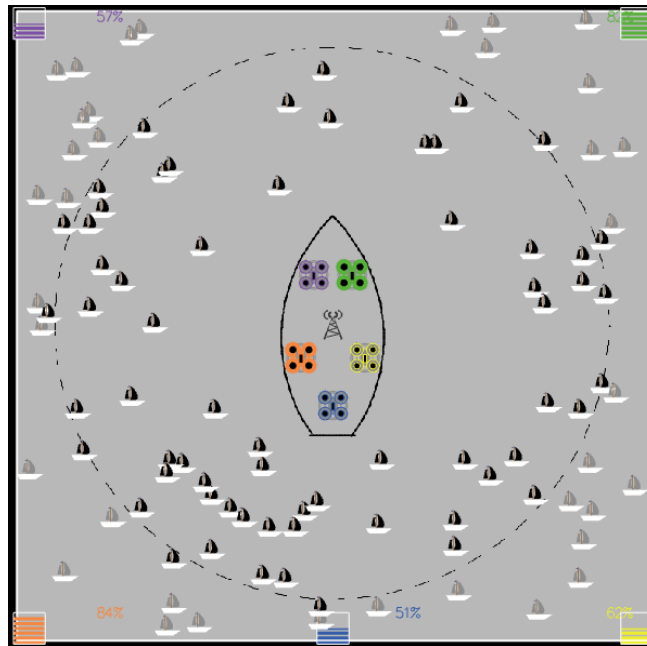


Figura N° 3: Modelo NetLogo para 5 drones.

En esta instancia del modelo, se puede apreciar la embarcación principal que cuenta con la torre de control y 5 drones disponibles para realizar tareas. Además, se puede visualizar el perímetro abarcado por el radar y embarcaciones dentro y fuera de este perímetro.

De igual manera, se puede observar el nivel de recarga de cada batería, como un cuadrado ubicado en los extremos que representa el porcentaje de recarga.

Elementos en la interfaz de NetLogo:

1. **Elementos de control:**

- Setup** (botón): inicializa el mundo (plano) ubicando la torre de control, los elementos nocivos y no nocivos, y los drones que realizan las tareas.
- Go** (botón): inicia la simulación del modelo.
- 1 tick** (botón): ejecuta un “step” y actualiza el estado de los elementos del modelo.

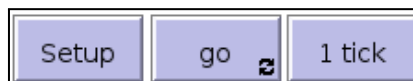


Figura N° 4: Elementos de Control del modelo.

2. **Variables de entrada:**

- q-drones** (slider): define la cantidad de drones que se crearán para realizar las tareas (máximo 5 drones).

- b. **q-objetos** (slider): define la cantidad de embarcaciones que estarán en el entorno.
- c. **probabilidad-nocividad** (slider): define la probabilidad que tiene cada embarcación de ser considerada como nociva.
- d. **administración-energía**: define la activación de utilizar drones con límites de operación basado en energía disponible.
- e. **probabilidad-pérdida**: probabilidad de que un dron se pierda o presente fallas.

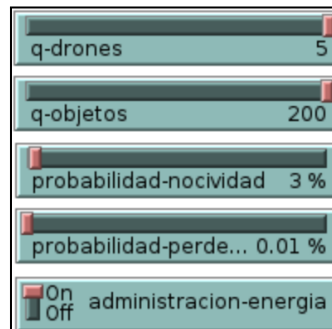


Figura N° 5: Elementos de Entrada/Interacción del modelo.

### 3. Variables de salida:

- a. **Monitor de distancia recorrida (metros)**: exhibe curvas que representan la distancia recorrida por cada dron. Este monitor cuenta con los siguientes indicadores específicos:
  - i. **distancia-recorrida-total**: sumatoria de la distancia recorrida por los drones.
  - ii. **promedio**: distancia promedio recorrida por cada dron.
  - iii. **desviación-estándar**: desviación estándar de las distancias recorridas por los drones.

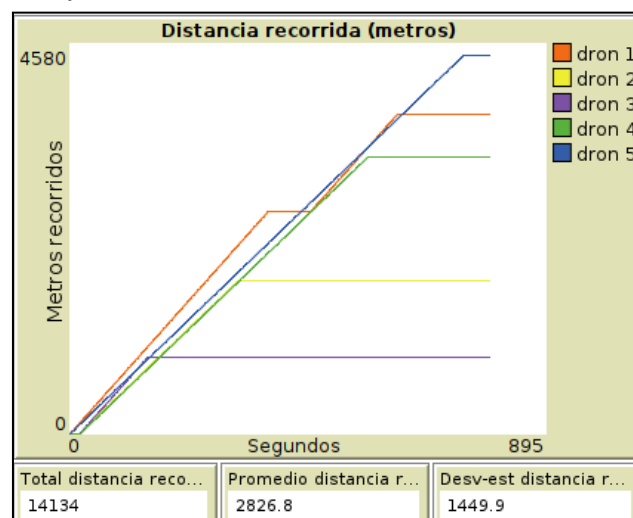


Figura N° 6: Elementos de Salida del modelo, Distancia Recorrida.

- b. **Monitor de tiempo de vuelo (segundos):** exhibe curvas que representan el tiempo de vuelo de cada drone. Este monitor cuenta con los siguientes indicadores específicos:
- tiempo-vuelto-total:** sumatoria de los tiempos de vuelo de cada drone.
  - promedio-tiempo-vuelo:** tiempo de vuelo promedio por cada drone.
  - desviación-estándar-tiempo-vuelo:** desviación estándar de los tiempos de vuelo de los drones.

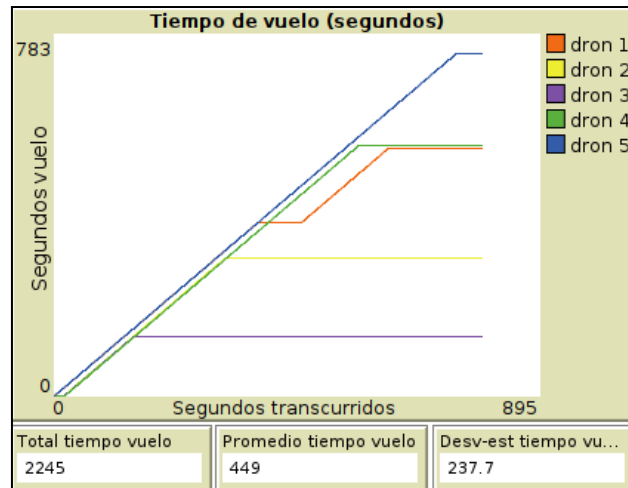


Figura N° 7: Elementos de Salida del modelo, Tiempo de Vuelo.

- c. **Monitor de objetivos alcanzados por drone (cantidades):** cantidad de objetivos alcanzados por los drones. Indicadores específicos:
- total objetivos:** total de objetivos alcanzados.
  - promedio objetivos:** promedio de objetivos alcanzados por unidad de tiempo.
  - desv-est objetivos:** desviación estándar de la cantidad de objetivos alcanzados por los drones.

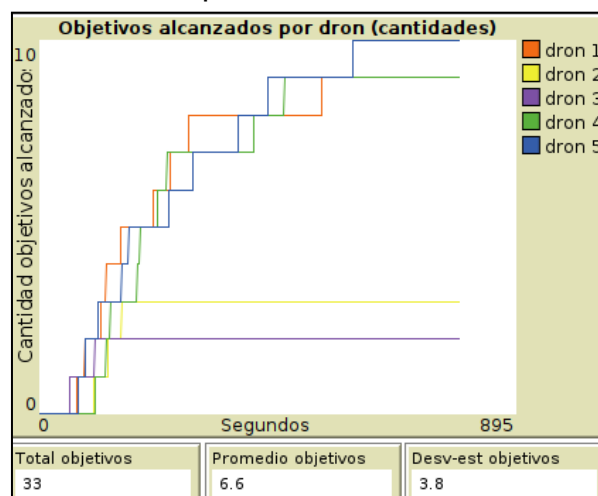


Figura N° 8: Elementos de Salida del modelo, objetivos alcanzados.

- d. **Monitor de consumo (watts)**: medición de la energía consumida por los drones. Indicadores específicos:
- total consumo**: total de watts consumidos por los drones.
  - promedio consumo**: watts consumidos en promedio por los drones.
  - desv-est consumo**: desviación estándar de los watts consumidos por los drones.

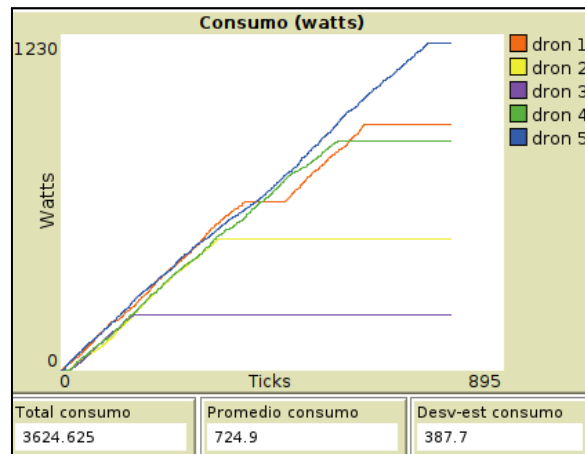


Figura N° 9: Elementos de Salida del modelo, Consumo.

- e. **Monitor del tiempo medio de clasificación (TMC)**: cada cuánto tiempo, en promedio, los drones alcanzan un objetivo.

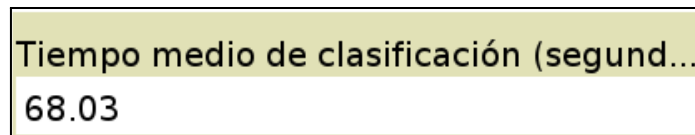


Figura N° 10: Elementos de Salida del modelo, Tiempo medio de clasificación.

### III. Capítulo 3: Experimentos con el modelo:

Con intención de estudiar el modelo y entender cómo las variables definidas afectan su desempeño, se realizaron experimentos basados en técnicas de Diseño y Análisis de Experimentos Factoriales. El interés se centra en los efectos que tienen las variables sobre el **Tiempo Medio de Clasificación (TMC)**. En consecuencia, esta será nuestra variable de estudio para medir el desempeño del modelo durante los experimentos.

Objeto comprender de mejor manera las variables a utilizar, previamente descritas en el punto C.2 y C.3, se seleccionaron como objeto de estudio las siguientes:

#### A. Selección de Variables de entrada:

- a) **q-objetos** (cantidad de embarcaciones):  
Como objeto de estudio, se decidió utilizar las siguientes cantidades: **150** y **200**, entendiendo que no representan un escenario real pero permiten compensar las falsas detecciones de radar<sup>10</sup>.
- b) **q-drones** (cantidad de drones):  
Como objeto de estudio, se decidió utilizar las siguientes cantidades: **3** y **5**, entendiendo que es un número razonable de dispositivos por adquirir, teniendo en cuenta que forma parte de una prueba de conceptos (prueba práctica).
- c) **probabilidad-pérdida** (probabilidad de que un drone presente fallas y quede inoperativo):  
Como objeto de estudio, se decidió utilizar valores de: **0%** y **1%**, lo anterior, de acuerdo a lo publicado por diversos fabricantes de drone que definen un nivel de confiabilidad (Reliability) de los drones civiles cercanos al 99% y un MTBF<sup>11</sup> mayor a 33.000 horas (1 falla cada 49 meses<sup>12</sup>). Cabe destacar que se considera un UAV nuevo o en su efecto, con mantenimiento preventivo al día.

#### B. Selección de la variable de salida (estudio):

- a) **Tiempo Medio de Clasificación (segundos):**  
Este valor está definido como **tiempo total de vuelo** dividido entre **cantidad de drones**.

Dado que las variables de entrada se analizarán para dos valores de interés correspondientemente, los experimentos factoriales serán de dos niveles, es decir, experimentos  $2^k$ ; donde k representa la cantidad de variables.

---

<sup>10</sup> Esto es común por el tamaño de la embarcación artesanal (pesquera), con respecto a la ola y la cercanía de la misma con la Unidad fiscalizadora, lo que produce falsos ecos y con ello, falsas detecciones. Pudiendo potencialmente generar múltiples contactos que pueden ser clasificados de igual manera con el UAV.

<sup>11</sup> MTBF: Mean Time Between Failures (Tiempo medio entre fallas).

<sup>12</sup> Petritoli et al (2018). Reliability and Maintenance Analysis of Unmanned Aerial Vehicles. Sensors, MDPI Journal. [En Línea], Disponible en: <https://www.mdpi.com/1424-8220/18/9/3171>, Pág. 11.

En el primer modelo ( $2^2$ ), es de nuestro interés analizar los efectos que tienen las variables **q-objetos** y **q-drones**, es decir, ignorando la posibilidad de falla que puedan tener los drones.

De esta forma se evalúa el modelo dejando de lado los aspectos técnicos que no están directamente relacionados con el diseño en sí.

De igual manera, en el segundo modelo ( $2^3$ ), se efectúa un experimento considerando este aspecto (representado por la variable **probabilidad-pérdida**), con el objeto de evaluar cuánto afecta una falla técnica que provoque la indisponibilidad de los drones, afectando consecuentemente el desempeño del modelo.

Teniendo en cuenta lo anterior, se presenta la definición de los experimentos factoriales  $2^k$ .

- 1) Cada variable tomará dos valores de interés. Es decir, habrá dos niveles.
- 2) Se realizarán experimentos con  $k = 2$  (ignorando fallas técnicas), y con  $k = 3$  (considerando fallas técnicas)
- 3) Cada tratamiento será evaluado en cinco iteraciones.

### C. Desarrollo del Modelo $2^2$ :

**Los experimentos fueron realizados con la siguiente definición:**

- 1) Cantidad de drones (Factor A):
  - a) Nivel 1: 3.
  - b) Nivel 2: 5.
- 2) Cantidad de embarcaciones (Factor B):
  - a) Nivel 1: 150.
  - b) Nivel 2: 200.
- 3) Probabilidad de nocividad/anomalía: 3% (constante).
- 4) Probabilidad de falla/perder drone: 0% (constante).
- 5) Administración de energía: True (constante).
- 6) Tiempo Medio de Clasificación (TMC): variable de salida (estudio), medida en segundos.
- 7) Cantidad de iteraciones: 5.



### Lecturas obtenidas durante los experimentos:

Combinación de Tratamientos	Réplica 1	Réplica 2	Réplica 3	Réplica 4	Réplica 5	Respuesta (promedio réplicas)
A <sub>1</sub> = 3, B <sub>1</sub> = 150	39.33	36.41	39.79	48.81	43.21	41.51
A <sub>2</sub> = 5, B <sub>1</sub> = 150	30.22	30.50	43.42	30.42	43.72	35.66
A <sub>1</sub> = 3, B <sub>2</sub> = 200	31.66	33.91	32.39	30.68	33.31	32.39
A <sub>2</sub> = 5, B <sub>2</sub> = 200	27.58	33.29	29.93	34.75	29.12	30.93
<b>Efecto global promedio</b>						<b>35.12</b>

Figura N° 11: Tabla de observaciones obtenidas, 2 ^ 2.

### Tabla de respuestas:

COMPONENTE	IDENTIDAD	Cantidad drones		Cantidad embarcaciones		Interacción	
	1	A		B		AB	
A1 = 3, B1 = 150	41,51	41,51		41,51			41,51
A1 = 3, B2 = 200	32,39	32,39			32,39	32,39	
A2 = 5, B1 = 150	35,66		35,66	35,66		35,66	
A2 = 5, B2 = 200	30,93		30,93		30,93		30,93
<b>TOTAL</b>	<b>140,49</b>	<b>73,90</b>	<b>66,59</b>	<b>77,17</b>	<b>63,32</b>	<b>68,05</b>	<b>72,44</b>
<b>VERIFICACIÓN</b>	<b>140,49</b>	<b>140,49</b>		<b>140,49</b>		<b>140,49</b>	
FACTOR		-1	1	-1	1	-1	1
NETO (sin abs)	140,49	-7,31		-13,84		4,40	
DIVISOR	4	2		2		2	
EFFECTO	35,12	-3,66		-6,92		2,20	
<b>RANKING</b>		<b>2</b>		<b>1</b>		<b>3</b>	

Figura N° 12: Tabla de resultado en base a su interacción, 2 ^ 2.

### Interpretación de los resultados:

#### 1. Efectos principales:

- a. **Cantidad de drones (Factor A):** Pasar de 3 a 5 drones disminuye cerca de 3.65 segundos el tiempo medio de clasificación. Lo cual hace sentido, teniendo en cuenta que hay más drones para alcanzar los objetivos. De acuerdo a los resultados obtenidos, esta variable ocupa el lugar número 2 (la 2da variable que más efecto tiene) en cuanto a influencia sobre el TMC.

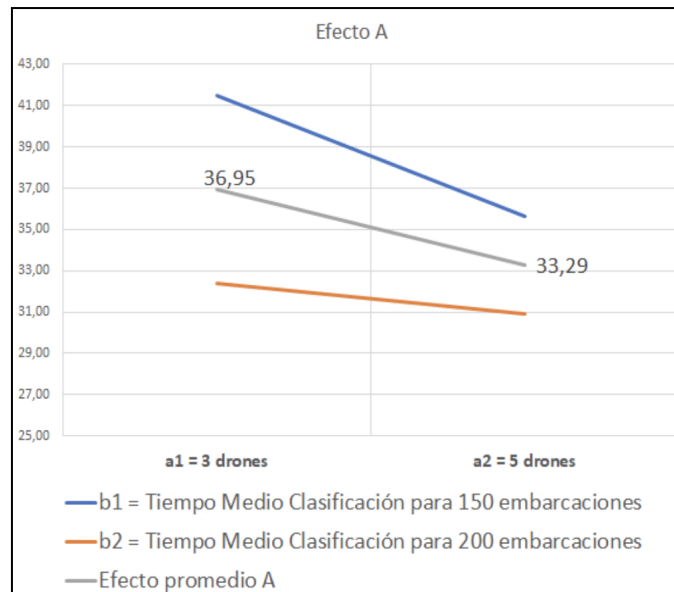


Figura N° 13: Gráfico de efectos por Factor A (q-objetos),  $2^2$ .

- b. **Cantidad de embarcaciones (Factor B):** Pasar de 150 a 200 embarcaciones disminuye cerca de 6.92 segundos el tiempo medio de clasificación. Tiene sentido, ya que hay más embarcaciones y estas pueden encontrarse relativamente cerca entre ellas. De acuerdo a los resultados obtenidos, ocupa el lugar número 1 (la variable que más efecto tiene) en cuanto a influencia sobre el TMC.

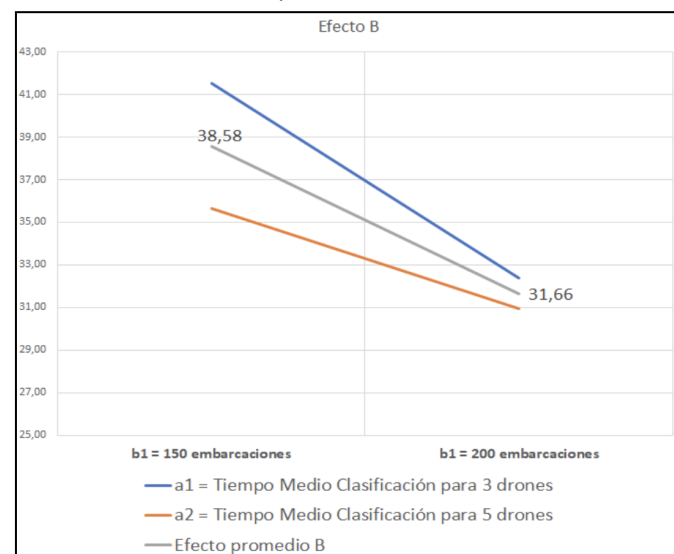


Figura N° 14: Gráfico de efectos por Factor B (q-drones),  $2^2$ .

## 2. Interacciones:

- a. **Interacción AB:** Pasar de 3 a 5 drones bajo el factor b (cantidad de embarcaciones), aumenta cerca de 2.20 segundos el TMC. La interacción de estas variables ocupa el lugar número 3 (la que menos efecto tiene) en cuanto a influencia en el TMC.

Intuitivamente podríamos pensar que la interacción AB lleva a disminuir el tiempo medio de clasificación, ya que los efectos principales de A y B disminuyen el tiempo medio. Sin embargo, sucede lo contrario: el tiempo medio incrementa. Entre las razones para que exista este comportamiento podríamos considerar el hecho de que los drones recargan energía periódicamente, lo cual implica viajar a la unidad central (buque fiscalizador) y desplegarse nuevamente. Este tiempo no es invertido en la inspección de las embarcaciones propiamente, sino tiempo destinado a mantener los drones operativos. En consecuencia, mientras los drones están **activamente inspeccionando**, mayor será el consumo de energía y mayor será, también, el tiempo invertido en viajar a la unidad central para recargar energía. Los viajes de recarga también pueden verse incrementados a medida que haya más embarcaciones. Una posible opción a estudiar para mejorar esta interacción, es aumentar la autonomía de los drones con la intención de disminuir los viajes de recarga.

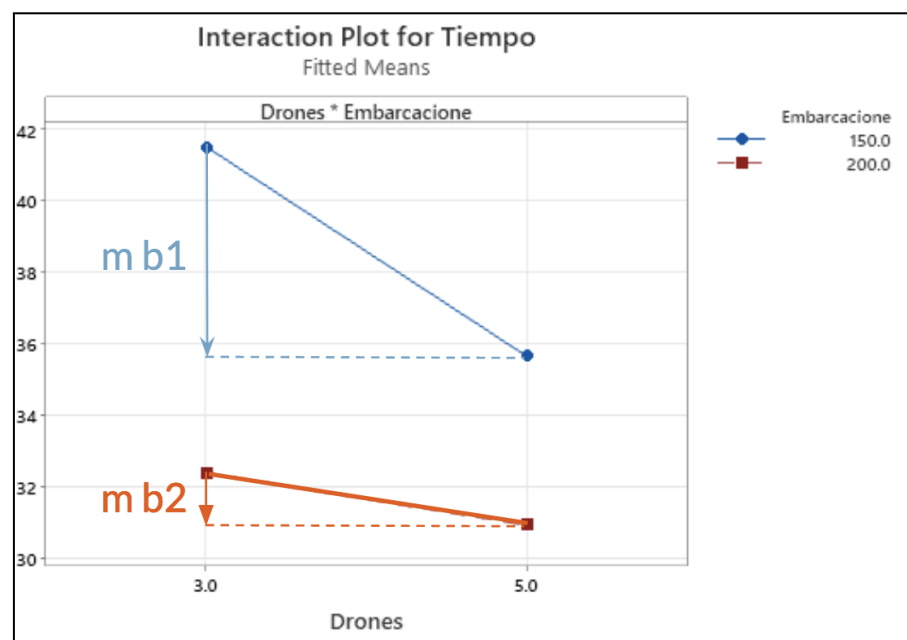


Figura N° 14: Gráfico de interacción de factores (AB),  $2^2$ .

Un dato interesante obtenido a través de los experimentos, es que la variable de estudio se comporta mejor cuando ambos factores se encuentran en sus niveles más altos (5 y 200 respectivamente), arrojando un TMC de 30.93 segundos.

### **Análisis de Varianza (ANOVA), Modelo 2<sup>2</sup>:**

En estricto rigor, dado que los experimentos realizados son muestrales (no poblacionales), no podríamos concluir directamente que los efectos vistos hasta ahora también se verán reflejados a nivel poblacional. En consecuencia, debemos utilizar un proceso estadístico de inferencia para ver si los efectos muestrales (experimentales) son lo suficientemente relevantes para que garanticen efectos poblacionales. Un proceso para realizar este estudio sería el Análisis de Varianza (ANOVA), el cual nos permite inferir si los efectos de cada variable se verán reflejados a nivel poblacional.

Es importante tener en cuenta que este análisis intenta explicar qué variables causan la diferencia de los resultados en comparación a la media de la variable de estudio, con lo cual, no es del todo necesario cuando los resultados obtenidos para cada tratamiento son parecidos a la media obtenida durante los experimentos.

Para el experimento 2<sup>2</sup> obtuvimos los siguientes valores de TMC:

Tratamiento	TCM
A1, B1	41.51
A2, B1	35.66
A1, B2	32.39
A2, B2	30.93

Tabla N° 1: Valores obtenidos de TCM para el modelo 2<sup>2</sup>.

Teniendo en cuenta que la media del TMC es de 35.12 y la desviación estándar es de 4.69, podríamos decir que los efectos obtenidos a partir de cada tratamiento, son muy parecidos, y el análisis ANOVA no sería necesario. Sin embargo, para fines de estudio, realizamos el análisis ANOVA para tener una visión más amplia sobre los niveles de significancia de cada variable. A continuación, los niveles de significancia obtenidos para cada variable:

Variable	Significancia
Drones	0.550
Embarcaciones	0.010

Tabla N° 2: Niveles de significancia para el modelo 2<sup>2</sup>.

Las variables se consideran significativas cuando el análisis ANOVA arroja un valor menor a 0.05. En este caso, la variable que más impacto tiene sobre el TMC es la cantidad de embarcaciones. A su vez, esto verifica el análisis de los experimentos factoriales presentados anteriormente, donde la cantidad de embarcaciones se encuentra en la posición número 1 del ranking de influencia, seguida de la cantidad de drones.

#### D. Desarrollo del Modelo $2^3$ :

**Los experimentos fueron realizados con la siguiente definición:**

- 1) Cantidad de drones (Factor A):
  - a) Nivel 1: 3.
  - b) Nivel 2: 5.
- 2) Cantidad de embarcaciones (Factor B):
  - a) Nivel 1: 150.
  - b) Nivel 2: 200.
- 3) Probabilidad de pérdida/falla (Factor C):
  - a) Nivel 1: 0%.
  - b) Nivel 2: 0.01%.
- 4) Probabilidad de nocividad/anomalía: 3% (constante).
- 5) Administración de energía: True (constante).
- 6) Tiempo Medio de Clasificación (TMC): variable de salida (estudio), medida en segundos.
- 7) Cantidad de iteraciones: 5.

**Lecturas obtenidas durante los experimentos:**

Combinación de Tratamientos	Réplica 1	Réplica 2	Réplica 3	Réplica 4	Réplica 5	Respuesta (promedio réplicas)
$A_1 = 3, B_1 = 150, C_1 = 0\%$	43.19	38.02	36.34	33.92	38.97	38.09
$A_2 = 5, B_1 = 150, C_1 = 0\%$	42.74	30.54	38.55	37.29	47.69	39.36
$A_1 = 3, B_2 = 200, C_1 = 0\%$	29.53	38.73	31.67	37.99	37.74	35.13
$A_2 = 5, B_2 = 200, C_1 = 0\%$	29.78	24.92	27.96	30.24	30.05	28.59
$A_1 = 3, B_1 = 150, C_2 = 0.01\%$	42.19	39.89	40.54	43.90	35.71	40.45
$A_2 = 5, B_1 = 150, C_2 = 0.01\%$	46.60	36.11	41.45	33.80	36.64	38.92
$A_1 = 3, B_2 = 200, C_2 = 0.01\%$	34.93	34.74	43.63	37.84	32.97	36.82
$A_2 = 5, B_2 = 200, C_2 = 0.01\%$	26.77	30.80	30.81	31.05	33.68	30.62
<b>Efecto global promedio</b>						<b>36.00</b>

Figura N° 15: Tabla de observaciones obtenidas,  $2^3$ .

## Tabla de respuestas:

COMPONENTE	IDENTIDAD	Cantidad drones		Cantidad embarcaciones		Probabilidad pérdida/falla dron		Interacción		Interacción		Interacción		Interacción	
	1	A		B		C		AB		AC		BC		ABC	
A1 = 3, B1 = 150, C1 = 0%	38.09	38.09		38.09		38.09		38.09		38.09		38.09		38.09	
A2 = 5, B1 = 150, C1 = 0%	39.36		39.36	39.36		39.36		39.36		39.36		39.36		39.36	
A1 = 3, B2 = 200, C1 = 0%	35.13	35.13			35.13	35.13			35.13		35.13	35.13			35.13
A2 = 5, B2 = 200, C1 = 0%	28.59		28.59		28.59	28.59		28.59		28.59		28.59		28.59	
A1 = 3, B1 = 150, C2 = 0.01%	40.45	40.45		40.45			40.45			40.45		40.45			40.45
A2 = 5, B1 = 150, C2 = 0.01%	38.92		38.92	38.92			38.92	38.92			38.92	38.92		38.92	
A1 = 3, B2 = 200, C2 = 0.01%	36.82	36.82			36.82		36.82	36.82		36.82			36.82	36.82	
A2 = 5, B2 = 200, C2 = 0.01%	30.62		30.62		30.62	30.62		30.62		30.62		30.62		30.62	
TOTAL	287.98	150.48	137.49	156.82	131.16	141.17	146.81	150.23	137.74	145.22	142.76	143.09	144.89	142.42	145.56
VERIFICACIÓN	287.98	287.98		287.98		287.98		287.98		287.98		287.98		287.98	
FACTOR		-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1
NETO (sin abs)	287.98	-12.99		-25.66		5.64		-12.49		-2.46		1.80		3.14	
DIVISOR	8	4		4		4		4		4		4		4	
EFFECTO	36.00	-3.25		-6.41		1.41		-3.12		-0.61		0.45		0.78	
RANKING		2		1		4		3		6		7		5	

Figura N° 16: Tabla de resultado en base a su interacción,  $2^3$ .

## Interpretación de los resultados:

### 1. Efectos principales:

- Cantidad de drones (Factor A):** Pasar de 3 a 5 drones disminuye en promedio cerca de 3.24 segundos. Hace sentido ya que hay más drones para inspeccionar. Esta es la 2da variable más incluyente sobre el TMC.
- Cantidad de embarcaciones (Factor B):** Pasar de 150 a 200 embarcaciones disminuye cerca de 6.41 segundos. Hay más embarcaciones y estas posiblemente estén cerca. Es la 1ra variable más influyente sobre el TMC.
- Probabilidad de pérdida (Factor C):** Pasar de 0 a 0.01 la prob. de falla, aumenta cerca de 1.4 segundos. Ocupa el lugar número 4 en el ranking de influencia.

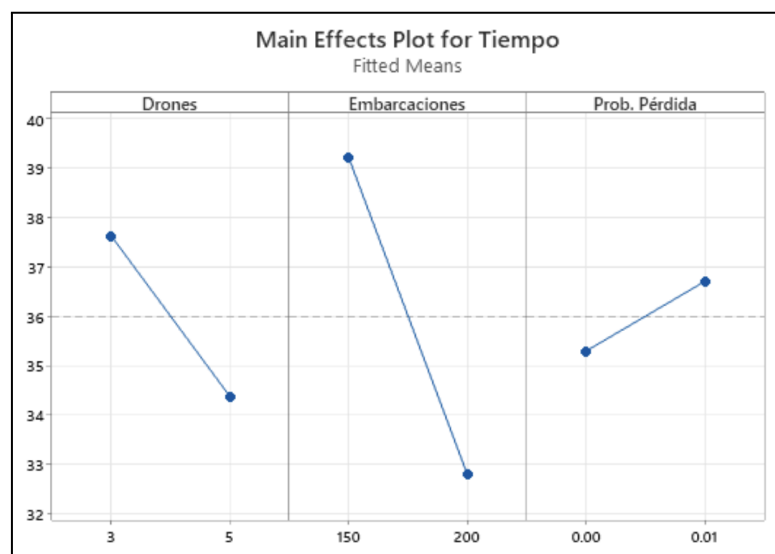


Figura N° 17: Efecto de la modificación de cada variable con respecto al tiempo,  $2^3$ .

## 2. Interacciones:

- a. **Interacción AB:** Pasar de 3 a 5 drones bajo el factor b (cantidad de embarcaciones), disminuye cerca de -3.12 segundos el tiempo medio de clasificación. La razón para que esta interacción se comporte de forma distinta al experimento  $2^2$ , es que en este experimento los drones pueden fallar/perderse. Con lo cual, aumentar la cantidad de drones (junto con los viajes de recarga que ello implica) compensa un poco la pérdida de drones. La interacción ocupa el lugar número 3 en el ranking de influencia.
- b. **Interacción AC:** Pasar de 3 a 5 drones bajo el factor c (probabilidad de perder drone), disminuye cerca de -0.61 segundos el tiempo medio de clasificación. Aumentar la cantidad de drones bajo el incremento de la probabilidad de pérdida, muestra cierta disminución en el tiempo medio de clasificación. Esta disminución podría hacerse más evidente con valores más altos para ambos factores. Ocupa el lugar número 6 en el ranking de influencia.
- c. **Interacción BC:** Pasar de 150 a 200 drones bajo el factor c (probabilidad de perder drone), aumenta cerca de 0.45 segundos el tiempo medio de clasificación. Ocupa el lugar número 7 en el ranking de influencia.
- d. **Interacción ABC:** Pasar de 3 a 5 drones bajo el factor b, bajo el factor c, incrementa alrededor de 0.78 segundos el tiempo medio de clasificación. Ocupa el lugar número 5 en el ranking de influencia.

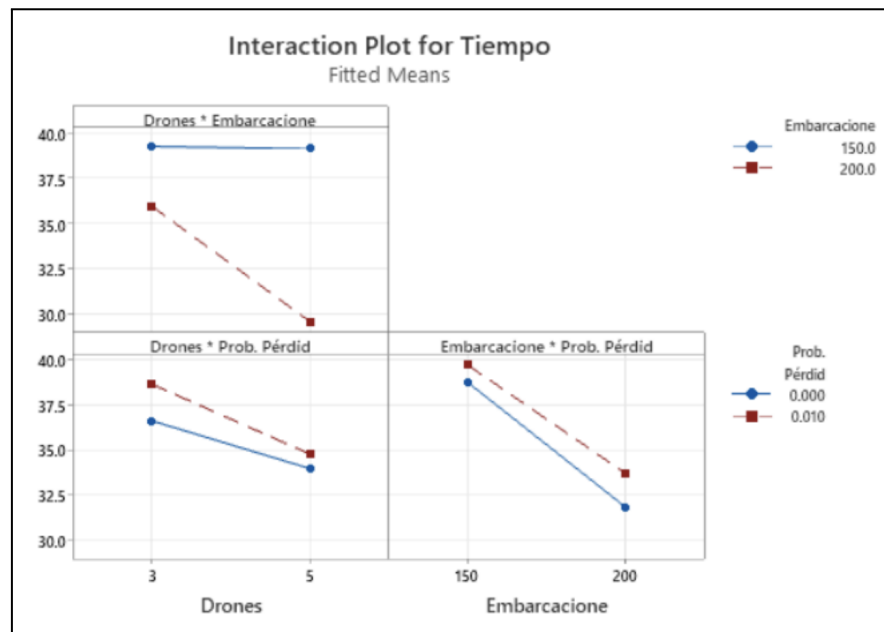


Figura N° 18: Gráfico de interacción de factores,  $2^3$ .

Durante estos experimentos, el mejor TMC se obtuvo cuando A y B estaban en sus niveles más altos y C en su nivel más bajo.

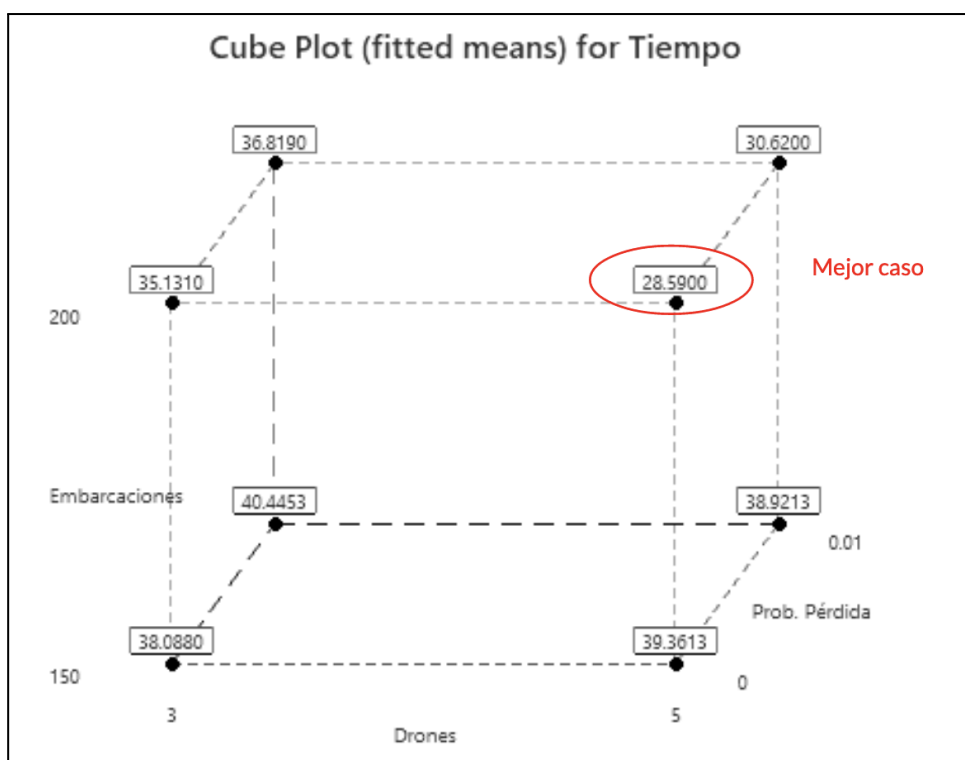


Figura N° 19: Gráfico cúbico de cada factor con respecto al tiempo,  $2^3$ .

### Análisis de Varianza (ANOVA), Modelo $2^3$ :

Para el experimento  $2^3$  obtuvimos los siguientes valores de TMC:

Tratamiento	TCM
A1, B1, C1	38.09
A2, B1, C2	39.36
A1, B2, C1	28.59
A2, B2, C1	30.93
A1, B1, C1	40.45
A2, B1, C2	38.92
A1, B2, C2	36.82
A2, B2, C2	30.62

Tabla N° 3: Valores obtenidos de TCM para el modelo  $2^3$ .



La media del TMC es de 36.00 y obtenemos una desviación estándar de 4.29. Lo cual sugiere que no es necesario un análisis ANOVA, sin embargo, lo realizamos para verificar los resultados obtenidos previamente. A continuación, los niveles de significancia para cada variable:

Variable	Significancia
Drones	0.586
Embarcaciones	0.001
Prob. de pérdida	0.755

Tabla N° 4: Niveles de significancia para el modelo 2<sup>3</sup>.

En este caso, la variable con más efecto sería la cantidad de embarcaciones, seguida de la cantidad de drones y por último, la probabilidad de pérdida. Este orden coincide y verifica el ranking obtenido previamente en la tabla de respuestas del experimento.

#### IV. Capítulo 4: Propuesta de Validación del Modelo:

Durante el desarrollo/diseño del modelo, inicialmente se efectuó el enunciado de una problemática relacionada a la poca eficiencia en las actividades de fiscalización pesquera que realizaba la Armada.

Para resolver lo anterior, se diseñó un modelo basado en un enjambre de drones que permitiera efectuar actividades de inteligencia, reconocimiento y vigilancia (ISR<sup>13</sup>), ante lo cual, se establecieron las limitaciones del modelo y posteriormente las especificaciones.

##### A. Verificación del modelo:

De manera inicial, el enfoque de cada verificación del modelo fue efectuado Bottom-Up, modificando parámetros en forma aleatoria y verificando interacciones entre los agentes y su impacto en el modelo (Análisis de Sensibilidad y Robustez).

De acuerdo al análisis de los resultados obtenidos, estos son obvios y tienen directa relación con la cantidad de embarcaciones a clasificar, lo que se ajusta a la realidad, bajo las limitaciones que durante el estudio se determinaron para el modelo, tratando de mantenerlo simple pero ajustado a ciertas condiciones que permitieran simular la realidad.

<sup>13</sup> ISR: Intelligence, Surveillance, and Reconnaissance.

Siguiendo con la verificación, se efectuaron pruebas de valores límite, donde se determinaron que se producen algunas fallas en el código al aumentar la tasa de fallas del sistema, por lo cual se decidió usar valores más ajustados a los niveles de confiabilidad promedio para drones comerciales.

## **B. Validación del modelo:**

El proceso de validación determina si existe una correspondencia entre el modelo implementado y la realidad, es una actividad compleja puesto que depende de las condiciones de borde y especificaciones del modelo, entendiendo que un modelo es una representación simplificada de la realidad. En todo momento se debe tener presente la interrogante que se quiere resolver, en este caso, podemos plantear la pregunta: ¿es más eficiente la implementación de un modelo de agentes de drones para apoyar las actividades de fiscalización pesquera?, tras los resultados podemos interpretar que si, pero de manera de validar esta afirmación, debemos efectuar dos ejes principales:

- 1) Nivel de ocurrencia del proceso de validación:
  - a) **Microvalidación:** Con respecto a microvalidación, esta permite observar si el modelo logró capturar las propiedades del agente (individuo), efectuando una comparación al más bajo nivel. En este caso particular, el agente (drone) tiene propiedades y características físicas que no fueron simuladas, pero también, podemos observar que dicho agente tiene una ubicación en el mundo, tiene un rumbo y velocidad determinados, tiene limitantes como propiedades de batería, fallas técnicas y la interacción con la estación controladora le permite estar consciente de su entorno (embarcaciones objetivos). La limitación mayor que se logra reconocer es que el componente altura no fue considerado en el modelo, pero sigue ajustándose a la realidad, puesto que podemos asumir una altura de vuelo predefinida para cada drone que sobrepase la altura máxima de cualquier embarcación menor.
  - b) **Macrovalidación:** Con respecto a la macrovalidación, esta permite observar si el modelo logró capturar las partes importantes del sistema. En este caso particular, si analizamos el comportamiento de enjambre y su interacción con la estación controladora, la forma de interacción y limitaciones de detección de radar, los rumbos y velocidades aleatorias que adquieren las embarcaciones a controlar, el proceso y tiempo de carga de batería de cada drone en la unidad

base/despliegue, la cantidad de embarcaciones que potencialmente podemos simular; se puede señalar que el modelo a pesar de sus limitaciones, es adecuado para representar un escenario real.

Del nivel de ocurrencia del proceso de validación se puede señalar que al comienzo señalamos que los agentes (drones), se comportaban como debería ser, de igual manera, identificamos que su interacción es la correcta, para finalmente determinar la validez de su comportamiento como un todo.

2) Nivel de detalle del proceso de validación:

- a) **Face Validation:** Con respecto a Face Validation, esta permite que cualquier persona que vea el modelo (“on face”), pueda distinguir los elementos y componentes del mundo en comparación con el mundo real. En el caso particular de este modelo, se observa que los agentes (drones), se desplazan de manera similar a la realidad, en línea recta, tienen la habilidad de cambiar dirección y tomar decisiones cinemáticas (moverse a una embarcación), en base a información provista por la estación controladora. Cabe destacar que no necesariamente cada agente debe poseer características físicas similares, pero en este caso particular, cada drone tiene un icono de drone que lo hace distinguible, al igual que la estación controladora que simula un OPV (Patrullero de alta mar) y las embarcaciones con el icono distintivo que las caracteriza.

Con respecto al escenario, se utilizó un radio de detección que simula la cobertura del radar, donde cada embarcación que se encuentre fuera de ese radio se transforma en gris o fuera de alcance. Por lo anteriormente mencionado, dentro del nivel de detalle del proceso, el modelo tiene validación facial (“Face Validation”).

- b) **Empirical Validation:** Con respecto a Empirical Validation, esta permite establecer a través de los datos producidos por el modelo si corresponden a un fenómeno real, de manera de efectuar una comparación no tan sólo como las cosas parecen si no que también, como se comporta en el mundo real. Para lo anterior, se desarrollaron diversos experimentos de manera de determinar si las características operacionales (cinemáticas) de los agentes simulados, eran similares al mundo real, como se muestra en la siguientes situaciones:

- i) **Velocidad del agente:** se utilizaron dos variables de salida (distancia recorrida [metros] y tiempo de vuelo [segundos]) para determinar la velocidad de los drones, obteniendo promedio de 6.6 m/s, equivalente a 23.22 km/hr. Efectuando una comparación con una ficha técnica de un drone DJI Mavik Air 2 (civil), esta muestra 68,4km/hr de velocidad máxima y 30km/hr de velocidad promedio lo que es similar a lo obtenido en el modelo, por lo cual, en aspecto de velocidad se encuentra validado empíricamente en base a los promedios de distancia recorrida [metros] y tiempo de vuelo [segundos]
- ii) **Tiempo de vuelo del agente (Autonomía):** se utilizó un promedio de la variable tiempo de vuelo [segundos], de la cual se obtuvo 745 segundos equivalentes a 12' 25'', pero considera un margen de seguridad para retornar a su base de operación de 10% y los drones pueden volar nuevamente con 75% de carga, por lo cual, la autonomía total del drone simulado es de 13' 36''.  
Efectuando una comparación con el mismo drone del punto anterior, este establece una autonomía máxima de 34 minutos, por lo cual es necesario generar ajustes menores en el modelo de manera de alcanzar condiciones más reales.
- iii) **Velocidad de las embarcaciones:** se utilizó una velocidad de embarcación de 4 m/s equivalentes a 7 nudos, estas tienen definidos rumbos aleatorios, tal como se observa en las actividades de pesca. Efectuada una comparación real, las embarcaciones de pesca en actividades extractivas o de tránsito, tienen velocidades promedio de entre 6 - 9 nudos, por lo cual, este aspecto del modelo coincide con el comportamiento real.
- iv) **Radio de cobertura del radar:** se utilizó un radio de cobertura del radar de 2000m equivalentes a 1.07 millas náuticas, pero, los radios de cobertura de radar para las bandas de frecuencia (Sierra y X-ray<sup>14</sup>) utilizadas en los OPV tienen un promedio de 26 millas en superficie, por lo cual, este radio de cobertura está más relacionado al alcance de las señales de control del drone (máximo 5 km) que al alcance real de un radar. Pero, para objeto de comprensión del modelo, sigue siendo una representación de condiciones reales de operación.

---

<sup>14</sup> Corresponde a la designación IEEE (US) de las Bandas de frecuencia, en este caso particular banda S corresponde a 2 - 4 Ghz y X corresponde a 8 - 12 Ghz.

- v) **Tiempo medio de Clasificación [segundos]:** de acuerdo al resultado obtenido de los experimentos y el TMC (Tiempo medio de clasificación) de 35 segundos, podemos relacionarlo con la distancia que debe recorrer cada agente para identificar cada embarcación; con valores promedio, cada agente debe recorrer un promedio de 234 metros para efectuar la clasificación de un drone lo cual es una situación que no se ajusta tanto a la realidad pero que puede ser mitigada para el efecto de reconocer los falsos contactos de radar. Por lo anterior, en un panorama de superficie saturado como el que fue simulado, es válido el modelo.

Habiendo efectuado una validación tanto en niveles de ocurrencia como niveles de detalle, es razonable señalar que en base a los resultados obtenidos de los 4 tipos de validación, el modelo es capaz de replicar potencialmente un escenario real.

Se debe recordar que lo que se busca es demostrar que es más eficiente utilizar un sistema de agentes para efectuar tareas de clasificación y esta versión simplificada de la realidad nos permite resolver ese tipo de interrogante.

Como comentario final, se puede señalar que actualmente no hay aplicaciones del tipo marítimo ni experiencias que utilicen este tipo de sistemas basados en agentes para efectuar tareas de fiscalización, por lo cual, hacer un método de validación cruzada con un sistema real en producción, resulta impracticable.

## V. Conclusiones:

- A. Realizado el modelo, este se centró principalmente en simular una situación de fiscalización pesquera, en la cual, un enjambre de drones efectuara reconocimiento de situaciones anómalas en embarcaciones menores, dentro de un radio de cobertura. Teniendo como objetivo principal, complementar el proceso fiscalizador para hacerlo más eficiente.
- B. Dentro de los análisis que se pueden obtener en base a los experimentos realizados, se señala que la variable asociada a la cantidad de embarcaciones, es el factor más preponderante en el desempeño general del modelo; lo cual, es posible observarlo tanto por sus niveles de significancia en los análisis de varianza, con un intervalo de confianza de 5%; como en la interacción en los experimentos factoriales, donde en cada uno demuestra una reducción del TMC. Lo anterior permite validar de manera cuantitativa el experimento puesto que se acerca a la realidad.
- C. Con respecto a la validez del modelo y su utilidad, en base a lo expuesto previamente, se puede determinar que efectuado los 4 tipos de validación y el correspondiente análisis, se concluye que el modelo desarrollado es capaz de replicar una versión simplificada de la realidad. Esto considera todas las limitaciones planteadas en los puntos precedentes relacionadas a limitaciones geográficas, componentes meteorológicos, alcances de radio frecuencia y todo aquellos imponderables que afecten al problema real.
- D. Por las razones que se mencionaron y la evidencia demostrada durante el desarrollo de este escrito, se permite asegurar que el modelo diseñado es capaz de simular el comportamiento de un sistema basado en agentes del tipo drone, planteándose como una opción útil para cumplir el objetivo de complementar el proceso de fiscalización pesquera.

## VI. Referencias bibliográficas utilizadas:

- [1]. DIRECTEMAR (2022), Ley General de Pesca y Acuicultura, D.S. N° 430. [En línea] Disponible en: <[https://www.directemar.cl/directemar/site/docs/20170125/20170125102709/tm\\_066\\_actualizada\\_10\\_mayo\\_2022.pdf](https://www.directemar.cl/directemar/site/docs/20170125/20170125102709/tm_066_actualizada_10_mayo_2022.pdf)> [Acceso el 21 Jun 2022].
- [2]. DIRECTEMAR (2021), Cuenta Pública de Fiscalización Pesquera del año 2021. [en línea] Disponible en: <[https://www.directemar.cl/directemar/site/docs/20170125/20170125102709/tm\\_066\\_actualizada\\_10\\_mayo\\_2022.pdf](https://www.directemar.cl/directemar/site/docs/20170125/20170125102709/tm_066_actualizada_10_mayo_2022.pdf)> [Acceso el 21 Jun 2022].
- [3]. Gomi, Takashi. (1995). *Subsumption Architecture*. Journal of Japan Society for Fuzzy Theory and Systems. [en línea] Disponible en: <[https://www.researchgate.net/publication/320005961\\_Subsumption\\_Architecture](https://www.researchgate.net/publication/320005961_Subsumption_Architecture)> [Acceso el 15 May 2022].
- [4]. Lindquist, J., (2004). *An analysis of degraded communications in the Army's future force*. [en línea] United States Naval Postgraduate School. Disponible en: <<https://calhoun.nps.edu/handle/10945/1549>> [Acceso el 14 May 2022].
- [5]. Munoz, M., (2011). *Agent-based simulation and analysis of a defensive UAV swarm against an enemy UAV swarm*. [en línea] United States Naval Postgraduate School. Disponible en: <<https://calhoun.nps.edu/handle/10945/5700>> [Acceso el 14 May 2022].
- [6]. Petritoli et al (2018). Reliability and Maintenance Analysis of Unmanned Aerial Vehicles. Sensors, MDPI Journal. [En Línea], Disponible en: <<https://www.mdpi.com/1424-8220/18/9/3171>> [Acceso el 21 Jun 2022].
- [7]. Sanchez, S., Cioppa, T. and Lucas, T., (2004). *Military applications of agent-based simulations*. [en línea] United States Naval Postgraduate School. Disponible en: <<https://calhoun.nps.edu/handle/10945/35333>> [Acceso el 14 May 2022].
- [8]. Sakellariou, Ilias & Kefalas, Petros & Stamatopoulou, Ioanna. (2009). *Multi Agent System Coursework Design in NetLogo*. [en línea] Disponible en: <[https://ccl.northwestern.edu/2009/MAS\\_Coursework\\_Design\\_in\\_NetLogo.pdf](https://ccl.northwestern.edu/2009/MAS_Coursework_Design_in_NetLogo.pdf)> [Acceso el 14 May 2022].
- [9]. Sakellariou, Ilias & Kefalas, Petros & Stamatopoulou, Ioanna. (2022). *Teaching Intelligent Agents using NetLogo*. [en línea] Disponible en: <[https://www.researchgate.net/publication/228667381\\_Teaching\\_Intelligent\\_Agents\\_using\\_NetLogo](https://www.researchgate.net/publication/228667381_Teaching_Intelligent_Agents_using_NetLogo)> [Acceso el 14 May 2022].
- [10]. SERNAPESCA (2021). Anuario Estadístico de Pesca y Acuicultura 2021. [En Línea] Disponible en: <[http://www.sernapesca.cl/sites/default/files/2021\\_030201\\_subsector\\_arterisanal.pdf](http://www.sernapesca.cl/sites/default/files/2021_030201_subsector_arterisanal.pdf)> [Acceso el 21 Jun 2022].

- [11]. Steele, M., (2004). *Agent-based simulation of unmanned surface vehicles: a force in the fleet*. [en línea] United States Naval Postgraduate School. Disponible en: <<https://calhoun.nps.edu/handle/10945/1509>> [Acceso el 14 May 2022].
- [12]. Wilensky & Rand (2016) *An Introduction to Agent-Based Modeling. Modeling Natural, Social and Engineered Complex Systems with Netlogo*. The MIT Press, Massachusetts.



# Anexo “A”

## Código de NetLogo del ABM

Software Netlogo 5.3.1. (February 29, 2016)  
web site [ccl.northwestern.edu/netlogo](http://ccl.northwestern.edu/netlogo)

Wilensky, U. 1999. Netlogo. <http://ccl.northwestern.edu/netlogo/>  
Center for Connected Learning and Computer-Based Modeling,  
Northwestern University. Evanston, IL.

### Código propio:

```
;;to do:
;;[3:15 p. m., 19/5/2022] Gustavo Jara Valdés: 1. ajustar el código para usar la
librería subsumption de canessa
;;[3:15 p. m., 19/5/2022] Gustavo Jara Valdés: 2. cambiar los objetos entrada por
monitores (reflejan mejor que son métricas)
;; - Aplicar breads para especializar por tipo de objeto
;; - Ordenar código para demostrar capas de la arquitectura subsunción y
aplicación reactiva de los drones (Aplicar librería by Canessa)
;; - Calcular trayectoria de la embarcación para que el dron lo intercepte (no lo
siga sino que siga su trayectoria)
;; - Dejar paramétricos los valores de administración de energía
;; - Ajustar el modelo de administración de energía para que siempre haya al
menos 1 dron en vuelo
;; - Hacer que los barcos rojos (de proyección) no pasen por sobre el buque
;; - Ajustar rutina de desviación de embarcaciones para que no intercepten
(choquen) con el buque
;; - Hacer wrap de las embarcaciones sin dejar el mundo como toroide
;; - Revisar rutina de acercamiento de embarcaciones por el este y oeste (se
acercan mucho al buque)

;; velocidades
;; Drones: 8 mt/s = 28.8 Km/h                rango: 6 a 10
;; Embarcaciones: 4 m/s = 14.4 Km/h = 7.cada 8 nudos  rango: 3 a 5
;; Energía:
;; Consumo: 100% es para 800 ticks (13.3 min) -> cada tick (seg) en vuelo
consume 1/8 de almacenamiento
;; Recarga: Cada tick para cada drone aterrizado recarga 1.2 (2 min para carga
total desde 0) y despega cuando alcanza un 75%
;; En vuelo: Regresa a la torre de control con una holgura de 10%

;;
;;
;; Extensiones Netlogo
;;
;;
```

```

extensions [array] ; Para manejo de arregos

;;
;; Breads
;;
directed-link-breed [mi-directed-links mi-directed-link]

;;
;; Variables globales
;;
globals [
  q-size-objeto
  q-size-drone
  id-primer-drone ;
  q-size-torre-control
  q-visibles ; Cantidad de turtles totales (drones + objetos NO nocivos +
objetos nocivos)
  k-lejanía-borde
  k-lejanía-torre-control ; el radio de lejanía a la torre de control
  b-primer-paso-simulacion
  k-perimetro-radar ; el radio de alcance del radar
  x-initial-position-drone
  y-initial-position-drone
  x-initial-position-store
  y-initial-position-store
  x-initial-position-label-store
  y-initial-position-label-store
  k-consumo-por-tick
  k-recarga-por-tick
  k-holgura-regreso-a-torre
  k_almacenamiento-minimo
  b-blink-drone
  k-blink-drone
  n-blink-count

; Para estadísticas
total-dr
prom-dr
dev-est-dr
total-objetivos
prom-objetivos
dev-est-objetivos
total-tpo-vuelo
prom-tpo-vuelo
dev-est-tpo-vuelo
total-consumo
prom-consumo
dev-est-consumo
]

```

```

;;
;;
;; Propiedades para los turtles
;;
turtles-own [
  tipo          ; torre-control, objeto o drone
  subtipo       ; solo para tipo objeto: NO-nocivo o nocivo
  id-origen     ; solo para tipo drone, contiene el id del último objeto asignado
  en la ruta (origen)
  id-destino    ; solo para tipo drone, contiene el id del último objeto asignado
  en la ruta (destino)
  id-drone-asignado ; solo para tipo objeto, contiene el id del drone asignado
  estado        ; Para tipo drone y objeto
  estado-energia ; Para tipo drone
  distancia     ; solo para tipo objetos
  distancia-recorrida ; solo para tipo drone, en metros
  tiempo-vuelo  ; solo para tipo drone, en segundos
  objetivos     ; solo para tipo drone, en cantidad
  consumo       ; solo para tipo drone, en watts
  b-en-perimetro-radar ; solo para tipo objeto
  velocidad     ; Para tipo drone y objeto
  almacenamiento ; 0 a 100%
]

;;
;;
;; Propiedades para los links
;;
links-own [
  id-origen-link
  id-destino-link
]

.....
;;
;;
;;
;;
;; SIMULACIÓN
;;
;;
;;
.....

;;
;;
;; GO !!!
;;
to go
  main-simulation
  tick
end

;;
;;

```

```

;; 1 tick on-demmand
;;
;;
to go-1-tick
  go
end

;;
;;
;; Simulación
;;
;;
to main-simulation
  ; Aplicar probabilidad de perder un drone
  ask turtles with [tipo = "drone" and subtipo = "inspector"] [
    if estado = "en-ruta" [
      ;if (random(100 + 1) * random(aplicar-prob-en-n-ticks + 1) <
probabilidad-perder-drone)[
        if precision ((random 10000 + 1) / 100) 2 <= probabilidad-perder-drone[
          set estado "perdido"
          set color gray
        ]
      ]
    ]
  ]

; Energía
if administracion-energia [
  ask turtles with [tipo = "drone" and subtipo = "inspector"] [
    ifelse estado = "en-ruta" [
      ; consumo
      set almacenamiento precision (almacenamiento - k-consumo-por-tick) 4
      set consumo precision (consumo + k-consumo-por-tick + random 4) 4
    ]
    [
      if estado = "aterrizado" [
        ; recarga
        set almacenamiento precision(almacenamiento + k-recarga-por-tick) 4
        if almacenamiento > 100 [
          set almacenamiento 100
        ]
      ]
    ]
  ]
  F-despliega-almacenamiento who
]

; mostrar/ocultar links de comunicaciones
ask turtles with [tipo = "drone" and subtipo = "inspector"] [
  let tmp-id-drone who
  let tmp-x xcor
  let tmp-y ycor
  let tmp-estado estado

```

```

ask links with [id-origen-link = tmp-id-drone and id-destino-link = 0][
  ;ifelse ((F-distancia-2coordenadas tmp-x tmp-y 0 0) <= k-lejanía-torre-control /
3 * 2)[
    ifelse (tmp-estado = "aterrizado" or tmp-estado = "perdido")[
      hide-link
    ]
    [
      show-link
    ]
  ]
]

```

```

; movimiento de los objetos
ask turtles with [tipo = "object" and estado = "non-inspected"]
;pendown
forward precision (velocidad / 10) 1

```

```

; para non-inspected
if estado = "non-inspected" [
  ; determinar si está en el perimetro del radar
  ifelse (F-dentro-del-perimetro xcor ycor)[
    set b-en-perimetro-radar true
    set color black
  ]
  [
    set b-en-perimetro-radar false
    set color gray
  ]
]

```

```

; determinar si llegó a los bordes para hacerlo rebotar
; superior
if (ycor > max-pycor - k-lejania-borde)[
  set heading (random 180) + 90
]
; inferior
if (ycor * -1 > max-pycor - k-lejania-borde)[
  ifelse (random(100) < 50)[
    set heading (random 90)
  ]
  [
    set heading (random 90) + 270
  ]
]
; derecha
if (xcor > max-pxcor - k-lejania-borde)[
  set heading (random 180) + 180
]
; izquierda
if (xcor * -1 > max-pxcor - k-lejania-borde)[

```

```

    set heading (random 180)
  ]
]
;
; determinar si se acerca al perímetro del buque para cambiarle la dirección
non-insected e inspected-referential
;
if F-distancia-2coordenadas xcor ycor 0 0 <= k-lejanía-torre-control [
  let tmp-menor-distancia 0
  let tmp-menor-cardinal ""
  let tmp-distancia-al-norte F-distancia-2coordenadas xcor ycor 0
  k-lejanía-torre-control
  let tmp-distancia-al-sur F-distancia-2coordenadas xcor ycor 0
  (k-lejanía-torre-control * -1)
  ifelse tmp-distancia-al-norte < tmp-distancia-al-sur
  [
    set tmp-menor-distancia tmp-distancia-al-norte
    set tmp-menor-cardinal "N"
  ]
  [
    set tmp-menor-distancia tmp-distancia-al-sur
    set tmp-menor-cardinal "S"
  ]
  ]
  let tmp-distancia-al-este F-distancia-2coordenadas xcor ycor
  k-lejanía-torre-control 0
  if tmp-distancia-al-este < tmp-menor-distancia
  [
    set tmp-menor-distancia tmp-distancia-al-este
    set tmp-menor-cardinal "E"
  ]
  ]
  let tmp-distancia-al-oeste F-distancia-2coordenadas xcor ycor
  (k-lejanía-torre-control * -1) 0
  if tmp-distancia-al-oeste < tmp-menor-distancia
  [
    set tmp-menor-distancia tmp-distancia-al-oeste
    set tmp-menor-cardinal "O"
  ]
  ]
  let k-holgura 10
  let tmp-b-asignado false
  if (tmp-menor-cardinal = "N") [
    ;pendown
    if (xcor >= 0 - k-holgura and not tmp-b-asignado)[
      ifelse (heading >= 180 and heading <= 180 + 45) [
        set tmp-b-asignado true
        set heading heading - 90
      ]
      [
        if (heading >= 180 + 45 and heading <= 270) [
          set tmp-b-asignado true

```

```

        set heading heading + 90
    ]
]
]
if (xcor <= 0 + k-holgura and not tmp-b-asignado)[
    ifelse (heading >= 90 and heading <= 90 + 45) [
        set tmp-b-asignado true
        set heading heading - 90
    ]
    [
        if (heading >= 90 + 45 and heading <= 180) [
            set tmp-b-asignado true
            set heading heading + 90
        ]
    ]
]
]
]
if (tmp-menor-cardinal = "S") [
    ;pendown
    if (xcor >= 0 - k-holgura and heading >= 270 and heading <= 270 + 45) [
        ;set color red
        set heading heading - 90
    ]
    if (xcor >= 0 - k-holgura and heading >= 270 + 45 and heading <= 360) [
        ;set color violet
        set heading heading + 90
    ]
    if (xcor <= 0 + k-holgura and heading >= 0 and heading <= 45) [
        ;set color red
        set heading heading - 90
    ]
    if (xcor <= 0 + k-holgura and heading >= 45 and heading <= 90) [
        ;set color violet
        set heading heading + 90
    ]
]
]
if (tmp-menor-cardinal = "E") [
    ;pendown
    if (ycor >= 0 - k-holgura and heading >= 180 and heading <= 180 + 45) [
        ;set color red
        set heading heading - 90
    ]
    if (ycor >= 0 - k-holgura and heading >= 180 + 45 and heading <= 270) [
        ;set color violet
        set heading heading + 90
    ]
    if (ycor <= 0 + k-holgura and heading >= 270 and heading <= 270 + 45) [
        ;set color red
        set heading heading - 90
    ]
]
]

```

```

]
if (ycor <= 0 + k-holgura and heading >= 270 + 45 and heading <= 360) [
  ;set color violet
  set heading heading + 90
]
]
]
if (tmp-menor-cardinal = "O") [
  ;pendown
  if (ycor >= 0 - k-holgura and heading >= 90 and heading <= 90 + 45) [
    ;set color red
    set heading heading - 90
  ]
  if (ycor >= 0 - k-holgura and heading >= 90 + 45 and heading <= 180) [
    ;set color violet
    set heading heading + 90
  ]
  if (ycor <= 0 + k-holgura and heading >= 0 and heading <= 45) [
    ;set color red
    set heading heading - 90
  ]
  if (ycor <= 0 + k-holgura and heading >= 45 and heading <= 90) [
    ;set color violet
    set heading heading + 90
  ]
]
]
]
]
; movimiento de los objetos referenciales
ask turtles with [tipo = "object" and estado = "inspected-referential"]
let tmp-velocidad -1
ask turtle (who - q-objetos)[
  set tmp-velocidad velocidad
]
forward precision (tmp-velocidad / 10) 1
]

; recorreremos los drones en ruta para que avancen a su objetivo
let b-actualiza-estadisticas false
ask turtles with [tipo = "drone" and subtipo = "inspector" and estado = "en-ruta"]
;ask turtles with [tipo = "drone" and subtipo = "inspector"]
let tmp-xcor-drone xcor
let tmp-ycor-drone ycor
if estado = "en-ruta" [
  set distancia-recorrida distancia-recorrida + velocidad ; en mt, velocidad en
mt/seg
  set tiempo-vuelo tiempo-vuelo + 1 ; 1 tick = 1 seg
  ;set consumo consumo + (k-consumo-tick)
  set b-actualiza-estadisticas true

```



```

forward precision (velocidad / 10) 1
F-actualizar-xy-almacenamiento who
]

let tmp-id-drone who
let tmp-id-origen id-origen
let tmp-id-destino id-destino
let tmp-color color
let b-atterrizarse false
let tmp-estado "en-ruta"
let b-ajustar-heading false
let tmp-heading -1
let tmp-xcor-destino -1
let tmp-ycor-destino -1
ask turtle id-destino [
  set tmp-xcor-destino xcor
  set tmp-ycor-destino ycor
  ifelse (F-rango round(tmp-xcor-drone) round(tmp-ycor-drone) xcor ycor 2)[ ;
Drone llegó al objeto ?
  ; si el drone llega a su punto origen, lo aterrizamos
  ifelse (tmp-id-drone - q-drones = tmp-id-destino) [ ; regresó a la torre de
control ?
    set b-atterrizarse true
    set tmp-estado "atterrizado"
    set tmp-xcor-drone xcor
    set tmp-ycor-drone ycor

    ; Borrar link del objeto anterior al drone
    ;F-elimina-link tmp-id-origen tmp-id-drone

    ; crear link del objeto actual al drone
    ;F-crea-link tmp-id-origen tmp-id-drone tmp-color 1.5

    ; aterrizarlo mirando al norte
    ask turtles with [who = tmp-id-drone][
      set heading 0 ; Finaliza mirando al norte (todo: no lo deja aterrizado al
norte)
    ]
  ]
  [ ; else, implica que encontró un objeto
  ;
  ; Inspeccionarlo para determinar nivel de nocividad
  ;
  set estado "inspected"
  ask turtle tmp-id-drone [
    set objetivos objetivos + 1
  ]
  ; determinar nocividad en base a una probabilidad
  ifelse (random(100) <= probabilidad-nocividad) [

```

```

        set subtipo "nocive"
        set color red
        F-dibuja-circulo xcor ycor q-size-drone / 2 * 1.1 color 1.6 true ; lo
marcamos con un círculo rojo
    ]
    [
        set subtipo "non-nocive"
        set color green - 1
        set size q-size-drone / 3 * 2
        F-dibuja-circulo xcor ycor q-size-drone / 3 color 1 true ; lo marcamos con
un círculo rojo
    ]
    let tmp-subtipo subtipo

;
; Ajustar links
;

; Crear link desde el objeto anterior al actual
;F-crea-link tmp-id-origen tmp-id-destino tmp-color 1.5

; Borrar link del objeto anterior al drone
;F-elimina-link tmp-id-origen tmp-id-drone

; crear link del objeto actual al drone
;F-crea-link tmp-id-destino tmp-id-drone tmp-color 0.4

; el destino inspeccionado ahora pasa a ser el origen
ask turtles with [who = tmp-id-drone][
    set id-origen tmp-id-destino
]

;
; habilitar objeto referencial asociado que continúe el movimiento
;
if (tmp-subtipo = "nocive")[
    let tmp-heading-al-inspeccionar heading
    ask turtle (who + q-objetos) [
        set estado "inspected-referential"
        set xcor tmp-xcor-destino
        set ycor tmp-ycor-destino
        set heading tmp-heading-al-inspeccionar
        set shape "boat 3"
        ifelse tmp-subtipo = "nocive" [
            set color red
        ]
        [
            set color grey + 1
        ]
    ]
]

```

```

        show-turtle
    ]

    ; crear link entre el objeto inspeccionado y su referencial
    ifelse (subtipo = "nocive")[
        F-crea-link tmp-id-destino tmp-id-destino + q-objetos red + 1 0.1
"discontinuo"
    ]
    [
        F-crea-link tmp-id-destino tmp-id-destino + q-objetos grey + 1 0.1
"discontinuo"
    ]
    ]
    ]
    [ ; como no ha llegado el drone a destino, ajustamos precisión en su dirección
        set b-ajustar-heading true
        set tmp-heading F-direction tmp-xcor-drone tmp-ycor-drone tmp-xcor-destino
tmp-ycor-destino
    ]
    ;
    ; buscar nuevo objetivo, ajustar rutas y calibrar direcciones
    ;
    ;F-asigna-mas-cercano-OBJ
    ]
    if (b-atterrizar)[
        set estado tmp-estado
        set xcor tmp-xcor-drone
        set ycor tmp-ycor-drone
        F-actualizar-xy-almacenamiento tmp-id-drone
    ]
    if (b-ajustar-heading)[
        set estado tmp-estado
        set heading tmp-heading
    ]
    ]

    ; actualiza estadísticas
    if (b-actualiza-estadisticas)[
        let tmp-total-dr 0
        let tmp-total-tpo-vuelo 0
        let tmp-total-objetivos 0
        let tmp-total-consumo 0
        ask turtles with [tipo = "drone" and subtipo = "inspector"]
        set tmp-total-dr tmp-total-dr + distancia-recorrida
        set tmp-total-tpo-vuelo tmp-total-tpo-vuelo + tiempo-vuelo
        set tmp-total-objetivos tmp-total-objetivos + objetivos
        set tmp-total-consumo tmp-total-consumo + consumo
    ]

```

```

set total-dr tmp-total-dr
set total-tpo-vuelo tmp-total-tpo-vuelo
set total-objetivos tmp-total-objetivos
set total-consumo tmp-total-consumo
set prom-dr precision (total-dr / q-drones) 1
set prom-tpo-vuelo precision (total-tpo-vuelo / q-drones) 1
set prom-objetivos precision (total-objetivos / q-drones) 1
set prom-consumo precision (total-consumo / q-drones) 1
ifelse q-drones > 1 [
  set dev-est-dr precision (standard-deviation [distancia-recorrida] of turtles with
[tipo = "drone" and subtipo = "inspector"]) 1
  set dev-est-tpo-vuelo precision (standard-deviation [tiempo-vuelo] of turtles
with [tipo = "drone" and subtipo = "inspector"]) 1
  set dev-est-objetivos precision (standard-deviation [objetivos] of turtles with
[tipo = "drone" and subtipo = "inspector"]) 1
  set dev-est-consumo precision (standard-deviation [consumo] of turtles with
[tipo = "drone" and subtipo = "inspector"]) 1
]
[
  set dev-est-dr 0
  set dev-est-tpo-vuelo 0
  set dev-est-objetivos 0
  set dev-est-consumo 0
]
]
;
; buscar nuevo objetivo, ajustar rutas y calibrar direcciones
;
F-asigna-mas-cercano-OBJ
end

```

```

.....
..
..
..
..
.. CONFIGURACIÓN INICIAL
..
..
..
..
.....
to setup
  clear-all
  setup_globals
  setup_turtles
  setup_drones_to_objetos
  reset-ticks
end

..
..
.. Configuración de variables iniciales
..
..

```

```

to setup_globals
; comportamiento
set q-size-objeto 11
set q-size-drone 10
set q-size-torre-control 10
set id-primer-drone 1 + q-objetos * 2 + q-drones
set k-lejania-borde 5
set k-lejania-torre-control 45
set b-primer-paso-simulacion false
set k-perimetro-radar max-pxcor - k-lejania-borde * 2
set k-consumo-por-tick 1 / 8
set k-recarga-por-tick 1.2 ; estaba en 1 / 12
set k-holgura-regreso-a-torre 10
set k_almacenamiento-minimo 75

; blink a drones durante carga
set b-blink-drone true
set k-blink-drone 20
set n-blink-count 0

; posición inicial de drones
; abscisa (x) posición inicial drones
set x-initial-position-drone array:from-list n-values 5 [0]
;array:set x-initial-position-drone 0 -10
;array:set x-initial-position-drone 1 10
array:set x-initial-position-drone 0 -9
array:set x-initial-position-drone 1 9
;array:set x-initial-position-drone 2 -6
;array:set x-initial-position-drone 3 6
array:set x-initial-position-drone 2 -8
array:set x-initial-position-drone 3 7.8
array:set x-initial-position-drone 4 0
; ordenada (y) posición inicial drones
set y-initial-position-drone array:from-list n-values 5 [0]
array:set y-initial-position-drone 0 q-size-drone * -1
array:set y-initial-position-drone 1 q-size-drone * -1
;array:set y-initial-position-drone 2 q-size-drone * 1.6
;array:set y-initial-position-drone 3 q-size-drone * 1.6
array:set y-initial-position-drone 2 q-size-drone * 1.2
array:set y-initial-position-drone 3 q-size-drone * 1.2
;array:set y-initial-position-drone 4 q-size-drone * -2.5
array:set y-initial-position-drone 4 q-size-drone * -2

; posición inicial de los stores
; abscisa (x) posición inicial stores
set x-initial-position-store array:from-list n-values 5 [0]
array:set x-initial-position-store 0 max-pxcor * -1 + q-size-drone / 2
array:set x-initial-position-store 1 max-pxcor - q-size-drone / 2
array:set x-initial-position-store 2 max-pxcor * -1 + q-size-drone / 2

```

```

array:set x-initial-position-store 3 max-pxcor - q-size-drone / 2
array:set x-initial-position-store 4 0

; ordenada (y) posición inicial stores
set y-initial-position-store array:from-list n-values 5 [0]
array:set y-initial-position-store 0 max-pycor * -1 + q-size-drone / 2
array:set y-initial-position-store 1 max-pycor * -1 + q-size-drone / 2
array:set y-initial-position-store 2 max-pycor - q-size-drone / 2
array:set y-initial-position-store 3 max-pycor - q-size-drone / 2
array:set y-initial-position-store 4 max-pycor * -1 + q-size-drone / 2

; posición inicial de los stores (labels)
; abscisa (x) posición inicial stores (labels)
let tmp-x-label-ajuste 6
let tmp-x-label-ajuste-rigth -4
set x-initial-position-label-store array:from-list n-values 5 [0]
array:set x-initial-position-label-store 0 max-pxcor * -1 + q-size-drone / 2 +
(q-size-drone + tmp-x-label-ajuste)
array:set x-initial-position-label-store 1 max-pxcor - q-size-drone / 2 -
(q-size-drone + tmp-x-label-ajuste + tmp-x-label-ajuste-rigth)
array:set x-initial-position-label-store 2 max-pxcor * -1 + q-size-drone / 2 +
(q-size-drone + tmp-x-label-ajuste)
array:set x-initial-position-label-store 3 max-pxcor - q-size-drone / 2 -
(q-size-drone + tmp-x-label-ajuste + tmp-x-label-ajuste-rigth)
array:set x-initial-position-label-store 4 0 + (q-size-drone + tmp-x-label-ajuste)

; ordenada (y) posición inicial stores (labels)
set y-initial-position-label-store array:from-list n-values 5 [0]
let tmp-y-label-ajuste 2
array:set y-initial-position-label-store 0 max-pycor * -1 + q-size-drone / 2 +
tmp-y-label-ajuste
array:set y-initial-position-label-store 1 max-pycor * -1 + q-size-drone / 2 +
tmp-y-label-ajuste
array:set y-initial-position-label-store 2 max-pycor - q-size-drone / 2 +
tmp-y-label-ajuste
array:set y-initial-position-label-store 3 max-pycor - q-size-drone / 2 +
tmp-y-label-ajuste
array:set y-initial-position-label-store 4 max-pycor * -1 + q-size-drone / 2 +
tmp-y-label-ajuste

; mundo
ask patches [set pcolor white - 3]
;ask patches [set pcolor 86]

; Cantidades de objetos
;set q-objetos-nocivos round(q-objetos-NO-nocivos * porc-objetos-nocivos / 100)
;set q-objetos q-objetos-NO-nocivos + q-objetos-nocivos
set q-visibles 1 + q-drones + q-objetos
end

```

```

;;
;; Configuración de turtles
;;
to setup_turtles
  ;; Torre de control
  create-turtles 1 [
    set tipo "torre-control"
    set subtipo ""
    setxy 0 0
    set color gray - 3
    set size q-size-torre-control
    set shape "torre-control"
  ]

  ;; Objetos
  let rec 0
  create-turtles q-objetos [
    set tipo "object"
    set subtipo ""
    set id-drone-asignado -1
    set size q-size-objeto
    move-to one-of patches
    ; Ubicarlo en posición permitida
    while [not F-coordenada-permitida xcor ycor][
      ;set xcor random max-pxcor * 2 - max-pxcor
      ;set ycor random max-pycor * 2 - max-pycor
      move-to one-of patches
    ]
    set estado "non-inspected"
    set distancia max-pxcor * 4 ; distancia grande para que entre la primera vez
    set shape "boat 3"
    ; determinar si está en el perimetro del radar
    ifelse (F-dentro-del-perimetro xcor ycor)[
      set b-en-perimetro-radar true
      set color black
    ]
    [
      set b-en-perimetro-radar false
      set color gray
    ]
    set velocidad random(3) + 3 ; Entre 3 a 5
    set rec rec + 1
  ]

  ;; Objetos referenciales
  set rec 0
  create-turtles q-objetos [
    set tipo "object"

```

```

set subtipo "referential"
set size q-size-objeto / 2
set estado "non-inspected-referential"
set shape "boat 3"
hide-turtle
]

; crear drones de marca inicial (pistas de aterrizaje)
let q-old-turtles count turtles + 1
create-turtles q-drones [
  set tipo "drone"
  set subtipo "initial"
  set color gray
  set size q-size-drone
  ;set xcor q-drones / -2 * q-size-drone + (who + 1 - q-old-turtles + 0.5) *
q-size-drone
  ;set ycor q-size-drone * -1.5
  set xcor array:item x-initial-position-drone (who - q-objetos * 2 - 1)
  set ycor array:item y-initial-position-drone (who - q-objetos * 2 - 1)

  set heading 0 ; comienza mirando al norte
  set shape "pista-elicoptero2"
]

;; Drones oficiales
create-turtles q-drones [
  set tipo "drone"
  set subtipo "inspector"
  set color F-color-ruta (who + 1 - q-old-turtles - q-drones)
  set size q-size-drone
  ;set xcor q-drones / -2 * q-size-drone + (who + 1 - q-old-turtles - q-drones + 0.5)
* q-size-drone * 3
  ;set ycor q-size-drone * -1.5
  let tmp-xcor -1
  let tmp-ycor -1
  ask turtle (who - q-drones)[
    set tmp-xcor xcor
    set tmp-ycor ycor
  ]
  set xcor tmp-xcor
  set ycor tmp-ycor
  set shape "drone3"

  ; primera asignación
  set id-origen (who - q-drones) ; El initial correspondiente a cada drone, el
destino se determinará por distancia
  set id-destino -1
  set estado "aterrizado"
  set heading 0 ; comienza mirando al norte

```



```

set distancia-recorrida 0
set tiempo-vuelo 0
set objetivos 0
set consumo 0
set velocidad random(3) + 6 ; Entre 6 a 8
set almacenamiento random(51) + 50 ; de 50 a 100
; crea link de comunicaciones
F-crea-link who 0 gray + 4 0.5 "comunicacion3"
let tmp-id-drone who
ask links with [id-origen-link = tmp-id-drone and id-destino-link = 0][
  hide-link
]
pendown
set pen-size 3.5
]

;; Shape almacenamiento para drones
create-turtles q-drones [
  set tipo "drone"
  set subtipo "store"
  set color F-color-ruta (who + 1 - q-old-turtles - q-drones * 2)
  ;set size q-size-drone
  set size q-size-drone / 5 * 3
  set shape "store 000"
  ;set xcor array:item x-initial-position-store (who - q-objetos * 2 - q-drones * 2 - 1)
  ;set ycor array:item y-initial-position-store (who - q-objetos * 2 - q-drones * 2 - 1)
  set xcor [xcor] of turtle (who - q-drones)
  set ycor ([ycor] of turtle (who - q-drones)) - q-size-drone / 5 * 4;
  ifelse administracion-energia [
    show-turtle
  ]
  [
    hide-turtle
  ]
]

;; Labels almacenamiento para drones
create-turtles q-drones [
  set tipo "drone"
  set subtipo "label-store"
  ;set label-color F-color-ruta (who + 1 - q-old-turtles - q-drones * 3)
  set label-color black
  set size q-size-drone
  set shape "nothing"
  ;set xcor array:item x-initial-position-label-store (who - q-objetos * 2 - q-drones *
3 - 1)
  ;set ycor array:item y-initial-position-label-store (who - q-objetos * 2 - q-drones *
3 - 1)
  set xcor ([xcor] of turtle (who - q-drones * 2)) - q-size-drone / 9

```

```

set ycor ([ycor] of turtle (who - q-drones * 2)) - q-size-drone / 8 * 7
ifelse administracion-energia [
  show-turtle
]
[
  hide-turtle
]
]

if administracion-energia [
  ;; Actualiza despliegue de almacenamiento
  ask turtles with [tipo = "drone" and subtipo = "inspector"][
    F-despliega-almacenamiento who
  ]
]

;; Para dibujos
create-turtles 1 [
  set tipo "dibujo"
  set subtipo ""
  ;hide-turtle
]

; dibujar perímetro de la torre de control
;F-dibuja-circulo 0 0 k-lejanía-torre-control - 2 gray - 1 1 false

; dibuja buque
F-dibuja-buque

; dibujar perímetro del radar
F-dibuja-circulo 0 0 k-perimetro-radar - 2 black 1 false
end

;;
;;
;; Asignar drones a objetos - planificación de rutas (el más cerca primero)
;;
;;
to setup_drones_to_objetos
  ; primer link entre inspector e initial
  ; ask turtles with [tipo = "drone" and subtipo = "inspector"][
  ;   F-crea-link id-origen who color 1
  ; ]

  ; asignar el drone más cercano a cada objeto
  F-asigna-mas-cercano-OBJ

  ; inician mirando al norte
  ask turtles with [tipo = "drone" and subtipo = "inspector"][
    set heading 0 ; comienza mirando al norte
  ]

```

```
end
```

```
.....  
;;  
;;  
;;  
;; FUNCIONES  
;;  
;;  
;;  
.....
```

```
;;  
;;  
;; Asigna objeto más cercano  
;;  
;;  
to F-asigna-mas-cercano-OBJ  
  if administracion-energia [  
    ; Para hacer blink a drones descargados cada n ticks  
    set n-blink-count n-blink-count + 1  
    ifelse b-blink-drone [  
      if n-blink-count = k-blink-drone [  
        set b-blink-drone false  
        set n-blink-count 0  
      ]  
    ]  
    [  
      if n-blink-count = k-blink-drone [  
        set b-blink-drone true  
        set n-blink-count 0  
      ]  
    ]  
  ]  
  
  ; dejamos aterrizados los que no han pasado un umbral mínimo de  
  almacenamiento  
  ask turtles with [tipo = "drone" and subtipo = "inspector"] [  
    ifelse estado = "en-ruta" [  
      ifelse estado-energia = "volviendo-a-recargar" [  
        ifelse b-blink-drone [  
          set shape "drone3-border"  
        ]  
        [  
          set shape "drone3"  
        ]  
      ]  
      [  
        ; determinar si le queda energía para volver a la torre de control a recargar  
        let tmp-xcor-origen -1  
        let tmp-ycor-origen -1  
        ask turtle (who - q-drones) [  
          set tmp-xcor-origen xcor  
          set tmp-ycor-origen ycor  
        ]  
      ]  
    ]  
  ]  
]
```



```

; para cada objeto determinamos cual es el dron más cercano
ask turtles with [tipo = "object" and estado = "non-inspected" and
b-en-perimetro-radar][
  let tmp-xcor-objeto xcor
  let tmp-ycor-objeto ycor
  let b-entro false
  let distancia-menor max-pxcor * 4 ; distancia grande para que entre la primera
vez
  let tmp-id-drone-a-asignar -1
  ; recorremos los drones para determinar el más cercano
  ask turtles with [tipo = "drone" and subtipo = "inspector" and estado != "perdido"
and (estado-energia = "ok" or not administracion-energia)][
    let distancia-objeto-a-drone F-distancia-2coordenadas tmp-xcor-objeto
tmp-ycor-objeto xcor ycor ; obtenemos la distancia efectiva entre el objeto y cada
drone
    if distancia-objeto-a-drone < distancia-menor [
      set distancia-menor distancia-objeto-a-drone
      set tmp-id-drone-a-asignar who
      set b-entro true
    ]
  ]
  if (b-entro)[
    ; asignamos al objeto el drone
    set id-drone-asignado tmp-id-drone-a-asignar
    set distancia distancia-menor
  ]
]

; ahora para cada drone determinamos cual es el objeto más cercano
ask turtles with [tipo = "drone" and subtipo = "inspector" and estado != "perdido"]
let tmp-id-drone who
let b-entro false
let tmp-id-objeto-a-asignar -1
if estado-energia = "ok" or not administracion-energia [
  let distancia-menor max-pxcor * 4 ; distancia grande para que entre la primera
vez
  ask turtles with [tipo = "object" and estado = "non-inspected" and
b-en-perimetro-radar and id-drone-asignado = tmp-id-drone][
    if distancia < distancia-menor [
      set distancia-menor distancia
      set tmp-id-objeto-a-asignar who
      set b-entro true
    ]
  ]
  if estado-energia = "volviendo-a-recargar" [
    set tmp-id-objeto-a-asignar who - q-drones
    set b-entro true
  ]
]

```

```

]
ifelse (b-entro)[
  set estado "en-ruta"
  ; asignamos al drone el objeto
  set id-destino tmp-id-objeto-a-asignar
  let tmp-xcor-destino -1
  let tmp-ycor-destino -1
  ask turtles with [who = tmp-id-objeto-a-asignar][ ; todo: no funciona con
id-destino (revisar porque)
  set tmp-xcor-destino xcor
  set tmp-ycor-destino ycor
]

; asignamos el ángulo de recorrido (desde el eje y (norte) hacia la izquierda
(este)
set heading F-direction xcor ycor tmp-xcor-destino tmp-ycor-destino
]
[
; Como no tiene más objetos, lo hacemos volver a la torre de control
set tmp-id-objeto-a-asignar who - q-drones
; asignamos al drone el objeto
set id-destino tmp-id-objeto-a-asignar
let tmp-xcor-destino -1
let tmp-ycor-destino -1
ask turtles with [who = tmp-id-objeto-a-asignar][ ; todo: no funciona con
id-destino (revisar porque)
  set tmp-xcor-destino xcor
  set tmp-ycor-destino ycor
]

; asignamos el ángulo de recorrido (desde el eje y (norte) hacia la izquierda
(este)
set heading F-direction xcor ycor tmp-xcor-destino tmp-ycor-destino
]
]

; poner labels para depuración
if labels [
  ask turtles with [tipo = "drone" and subtipo = "inspector"][
    set label (word who " o:" id-destino " " estado " e:" estado-energia)
  ]
  ask turtles with [tipo = "object" and estado = "non-inspected"][
    set label (word who " d:" id-drone-asignado)
  ]
]
]
end

;;
;;
;; Calcula la dirección en que se debe dirigir un drone

```

```

;;
;;
to-report F-direction [x1 y1 x2 y2]
  let dir-direction -1
  ; excepciones para calcular la dirección en casos en que y1 = y2 o x2 = x1
  ifelse (x1 = x2)[
    ifelse (y1 < y2)[
      set dir-direction 0
    ]
    [
      set dir-direction 180
    ]
  ]
  [
    ifelse (y1 = y2)[
      ifelse (x1 < x2)[
        set dir-direction 90
      ]
      [
        set dir-direction 270
      ]
    ]
  ]
  [
    ; saltadas las excepciones calculamos basado en arcotangente. Para Netlogo
    los ángulos comienzan desde el norte hacia el este
    set dir-direction atan (x2 - x1) (y2 - y1)
  ]
  report dir-direction
end

;;
;;
;; Función para establecer color de ruta para cada drone
;;
to-report F-color-ruta [rec-drone]
  let tmp-color-ruta random 140
  ifelse rec-drone = 0 [
    set tmp-color-ruta 26 ; naranjo
  ]
  [
    ifelse rec-drone = 1 [
      set tmp-color-ruta yellow
    ]
    [
      ifelse rec-drone = 2 [
        set tmp-color-ruta violet
      ]
      [
        ifelse rec-drone = 3 [
          set tmp-color-ruta green
        ]
      ]
    ]
  ]

```

```

    ]
    [
        if rec-drone = 4 [
            set tmp-color-ruta blue
        ]
    ]
]
]
]
report tmp-color-ruta
end

;;
;;
;; Devuelve true si la posición es permitida sino false
;;
to-report F-coordenada-permitida [x y]
    let coordenada-permitida true
    ; abscisas (x) no permitidas cerca del borde
    if (x > max-pxcor - k-lejania-borde or x < max-pxcor * -1 + k-lejania-borde)[
        set coordenada-permitida false
    ]
    ; ordenadas (y) no permitidas cerca del borde
    if (y > max-pycor - k-lejania-borde or y < max-pycor * -1 + k-lejania-borde)[
        set coordenada-permitida false
    ]
    ; coordenadas no permitidas cerca del perímetro del la torre de control
    let tmp-distancia F-distancia-2coordenadas x y 0 0
    ;if (x >= 0 and x < k-lejanía-torre-control and y >= 0 and y <
k-lejanía-torre-control) or (x >= 0 and x < k-lejanía-torre-control and y <= 0 and y >
k-lejanía-torre-control * -1) or (x <= 0 and x > k-lejanía-torre-control * -1 and y >= 0
and y < k-lejanía-torre-control) or (x <= 0 and x > k-lejanía-torre-control * -1 and y
<= 0 and y > k-lejanía-torre-control * -1)[
        if tmp-distancia <= k-lejanía-torre-control [
            set coordenada-permitida false
        ]
    ]
    report coordenada-permitida
end

;;
;;
;; Dibuja un círculo
;;
to F-dibuja-circulo [x y radio color-perimetro size-perimetro linea-continua]
    ; create-turtles 1 [
    ;   set tipo "tmp"
    ;   set xcor x + radio
    ;   set ycor y
    ;   set heading 0
    ; ]
    ask turtles with [tipo = "dibujo"]

```



```

let n-discontinuo 3
penup
set xcor x + radio
set ycor y
set heading 0
pendown
set color color-perimetro
set pen-size size-perimetro
let i 0
let switch-pen true
repeat 360 [
  if (not linea-continua)[
    if (int(i / n-discontinuo) = i / n-discontinuo)[
      set switch-pen not switch-pen
      ifelse (switch-pen)[
        penup
      ]
      [
        pendown
      ]
    ]
  ]
  forward 0.175 * radio / 10
  left 1
  set i i + 1
]
]
ask turtles with[tipo = "tmp"]
die
]
end

```

```

;;
;;
;; Dibuja contorno del buque
;;
to F-dibuja-buque
ask turtles with [tipo = "dibujo"]
set color black
penup

set xcor -7
set ycor -35
set pen-size 2
pendown
set heading 90
forward 14
penup

set xcor 0

```

```

    set ycor 35
    pendown
    set heading 135
    repeat 81 [
        forward 0.95
        right 1
    ]
    penup

    set xcor 0
    set ycor 35
    pendown
    set heading 225
    repeat 80 [
        forward 0.95
        left 1
    ]
    penup

    ;pendown
    ;forward 15
]
end

;;
;;
;; Creación de links
;;
to F-crea-link [Ink-id-origen Ink-id-destino Ink-color Ink-thickness Ink-shape]

    ask turtle Ink-id-origen [
        create-mi-directed-link-to turtle Ink-id-destino [
            set id-origen-link Ink-id-origen
            set id-destino-link Ink-id-destino
            set color Ink-color
            set thickness Ink-thickness
            set shape Ink-shape
        ]
    ]
end

;;
;;
;; Eliminación de links
;;
to F-elimina-link [Ink-id-origen Ink-id-destino]
    ask links with [id-origen-link = Ink-id-origen and id-destino-link = Ink-id-destino][
        die
    ]
end

```

```

..
;;
;; Función distancia entre 2 puntos
;;
to-report F-distancia-2coordenadas [x1 y1 x2 y2]
  ;show (word x1 " " y1 " " x2 " " y2)
  report sqrt(((x2 - x1) ^ 2) + ((y2 - y1) ^ 2))
end
..
;;
;; evalaur siun dron está cerca de un objeto
;;
to-report F-rango [x-drone y-drone x-obj y-obj rango]
  report (x-obj >= x-drone - rango and x-obj <= x-drone + rango) and (y-obj >=
y-drone - rango and y-obj <= y-drone + rango)
end

..
;;
;; Determinar si un obejto está dentro del perímetro del radar
;;
to-report F-dentro-del-perimetro [x y]
  let b-report false
  if (F-distancia-2coordenadas x y 0 0 <= k-perimetro-radar)[
    set b-report true
  ]
  report b-report
end
..
;;
;; Despliega shape y label de almacenamiento
;;
to F-despliega-almacenamiento [id-drone]
  ; label
  let tmp-almacenamiento -1
  ask turtles with [who = id-drone][
    set tmp-almacenamiento almacenamiento
  ]
  ask turtles with [who = id-drone + q-drones * 2][
    set label (word (precision tmp-almacenamiento 0) "%")
  ]
  ; shape
  ask turtles with [who = id-drone + q-drones][
    ifelse tmp-almacenamiento >= 95 [
      set shape "store 100"
    ]
    [
      ifelse tmp-almacenamiento >= 85 [
        set shape "store 090"
      ]
      [
        ifelse tmp-almacenamiento >= 70 [
          set shape "store 075"
        ]
      ]
    ]
  ]

```

```

    ]
    [
        ifelse tmp-almacenamiento >= 45 [
            set shape "store 050"
        ]
        [
            ifelse tmp-almacenamiento >= 20 [
                set shape "store 025"
            ]
            [
                ifelse tmp-almacenamiento >= 7 [
                    set shape "store 010"
                ]
                [
                    set shape "store 000"
                ]
            ]
        ]
    ]
]
end

;;
;; Actuazliar coordenadas del shape y label de almacenamiento
;;
to F-actualizar-xy-almacenamiento [id-drone]
    let tmp-xcor-drone [xcor] of turtle (id-drone)
    let tmp-ycor-drone [ycor] of turtle (id-drone)
    if (administracion-energia)[
        ; Actualizar coordenada del shape de almacenamiento
        ask turtle (id-drone + q-drones) [
            set xcor tmp-xcor-drone;
            set ycor tmp-ycor-drone - q-size-drone / 5 * 4;
        ]

        ; Actualizar coordenada del label de almacenamiento
        ask turtle (id-drone + q-drones * 2) [
            set xcor tmp-xcor-drone - q-size-drone / 9
            set ycor tmp-ycor-drone - q-size-drone / 8 * 7
        ]
    ]
end

```