

# INSTRUCCIONES IMPLEMENTACIÓN PSP-UAVs

Universidad Técnica Federico Santa María - Campus Casa Central  
Departamento de Informática

18 de octubre del 2025

Nicolás Armijo Calquín  
<nicolas.armijoc@usm.cl>

En el presente documento se detallarán las instrucciones del proyecto *Persistent Surveillance Problem with UAVs* a realizar durante el semestre 2025-2.

## 1. Motivación

Los problemas de ingeniería que se enfrentan hoy son cada vez más complejos y no siempre tienen soluciones fáciles o directas. Aquí es donde entran las metaheurísticas: métodos inteligentes que nos permiten buscar soluciones eficientes de manera aproximada, incluso cuando el problema es demasiado grande o complicado para resolverlo de forma exacta.

Aprender a implementar metaheurísticas no solo les dará herramientas poderosas para enfrentar desafíos reales, sino que también desarrollará su creatividad y pensamiento crítico, ya que tendrán que diseñar estrategias, ajustar parámetros y evaluar resultados de manera práctica. Cada experimento que realicen es una oportunidad para descubrir cómo optimizar recursos, mejorar procesos y generar soluciones innovadoras aplicables en la ingeniería moderna.

## 2. Problema, objetivos y restricciones

El problema a resolver es el *Persistent Surveillance Problem with UAVs*. La idea principal de esta parte del proyecto es lograr implementar alguna técnica incompleta de búsqueda, y comparar los resultados obtenidos con los de la literatura y los de sus pares estudiantes.

### 2.1. Objetivo del problema

El objetivo es encontrar una ruta para cada uno de los  $k$  drones utilizados, respetando las restricciones, con tal de **minimizar** la urgencia acumulada de todas las celdas durante todos los ticks de tiempo  $t$ .

### 2.2. Restricciones

Las restricciones a considerar son:

- Cada dron debe despegar desde alguna base habilitada.

- Cada dron, entre 2 instantes de tiempo continuos, sólo puede moverse a una de las 8 celdas vecinas ó quedarse estacionario en la celda que ya se encuentra.
- Ningún dron puede sobrevolar por una celda restringida.
- Dos o más drones no pueden encontrarse sobre la misma celda en un instante de tiempo  $t$ .
- No existe ninguna restricción sobre la autonomía de las baterías ni de la señal.

La cantidad de drones es un parámetro del problema, por lo que es necesario probar con diferentes cantidades de estos.

### 3. Instancias

Las instancias a utilizar se pueden encontrar en Moodle de informática en el curso del proyecto.

#### 3.1. Formato de las instancias

El formato de un archivo de instancia presenta el siguiente formato (en el siguiente orden de aparición):

1. **Grid rows:** La primera línea del archivo contiene “GRID\_ROWS  $m$ ”, donde  $m$  es la cantidad de filas de la grilla.
2. **Grid cols:** La segunda línea del archivo contiene “GRID\_COLS  $n$ ”, donde  $n$  es la cantidad de columnas de la grilla.
3. **N Obstacles:** Para los obstáculos se contiene la línea “N\_OBSTACLES  $o$ ”, donde  $o$  es la cantidad de obstáculos que hay en la grilla. Luego, las siguientes  $o$  líneas corresponden a dos números donde representan “fila columna”.
4. **N Urgencies:** Se tiene la línea “N\_URGENCIES  $u$ ”, donde  $u$  representa la cantidad de celdas de las que su urgencia es distinta de 0. Luego, las siguientes  $u$  líneas corresponden a 3 números donde representan “fila columna urgencia”
5. **N bases:** Para las bases se contiene la línea “N\_BASES  $b$ ”, donde  $b$  es la cantidad de bases (lugares donde puede despegar un dron) que hay en la grilla. Las siguientes  $b$  líneas contienen “id fila columna”

### 3.1.1. Ejemplo

```
GRID_ROWS 10
GRID_COLS 12

N_OBSTACLES 4
1 3
2 3
5 8
7 2

N_URGENCIES 5
0 1 4
0 2 2
3 5 8
4 6 10
8 10 7

N_BASES 2
0 0 0
1 9 11
```

## 3.2. Características de las instancias

Las instancias fueron generadas con especificamente para este proyecto y corresponden a archivos *.txt*. El (0,0) de la grilla se ubica en la esquina superior izquierda, tal como se observa en el Anexo. Se trabaja con 3 tamaños de instancias: pequeño, mediano y grande, donde cada uno tiene una variante de única base (tipo *a*) y multi-base (tipo *b*).

## 3.3. Observaciones

Asegúrese de leer correctamente cada archivo, de tal manera que su programa pueda ser capaz de manejar todos los archivos bajo esta estructura.

Si tiene problemas para leer las instancias, puede modificar la estructura o formato de éstas, sin modificar su contenido, pero debe especificar el cambio (así como también su funcionamiento) en el archivo README.txt y en su presentación.

## 4. Input del programa

El programa debe recibir como input (en línea de comandos) los siguientes parámetros:

1. PSP-UAV, nombre del programa.

2. La ruta del archivo de instancia, por ejemplo : “instancias/PSP-UAV\_01\_a.txt”.
3. Cantidad máxima de drones permitida por operación.
4.  $K$ , la cantidad de iteraciones permitida en una ejecución de su programa (no aplica para técnicas completas).
5.  $T$ , la duración de la operación, es decir, el total de ticks a planificar.

Ejemplo de entrada:

```
./PSP-UAV instancias/PSP-UAV_01_a.txt 5 1000 50
```

Es preciso mencionar que se pueden añadir otros parámetros adicionales en la entrada, si lo hace, debe incorporarlos junto con una descripción en el archivo README.txt, estableciendo al menos un ejemplo de cómo ejecutar su programa con estos parámetros.

## 5. Output del programa

Su programa debe mostrar por pantalla:

1. La urgencia acumulada entre la ventana de tiempo  $T$ .
2. Cantidad de drones utilizados.
3. Tiempo de ejecución de su programa, en segundos.
4. La ruta de cada dron a través de la ventana  $T$ , especificando de qué base despegas.

### 5.1. Ejemplo

```
Urgencia acumulada: 12933.0
Ventana de operación T: 50
Drones utilizados: 2
Tiempo de ejecución: 34.6s
Rutas:
D1: B0 - (0,0) - (1,1) - ... - (3,8) - (4,9)
D2: B1 - (5,8) - (5,8) - ... - (11,3) - (10,3)
```

### 5.2. Consideraciones

Adicionalmente, si usted desea mostrar otros datos que sean importantes (y que puedan visualizarse fácilmente), puede mostrarlos al final del output, especificando en el archivo README.txt qué quieren decir y qué sentido tiene añadirlos a la salida del programa.

Por ejemplo, realizar un script en Python para visualizar las rutas ejecutadas por los drones a modo de video.

## 6. Consideraciones y recomendaciones

1. Recuerde que su programa debe ser realizado en C/C++ con un Makefile para poder compilarlo.
2. Recuerde que debe comentar su código (sin excederse).
3. Si tiene problemas con algunas instancias específicas producto de, por ejemplo, cómputos muy grandes o demasiada memoria ram utilizada que saturó su pc, usted debería hacer lo siguiente:
  - a) Mostrar un resultado parcial (solución que cumple todas las restricciones) con un timeout de a lo MENOS 2 horas.
  - b) Explicar en la presentación cuáles instancias fueron las que originaron el problema.
  - c) Explicar cuál fue el problema que usted detectó: tamaño de la instancia muy grande, error aleatorio, etc.
4. Recuerde que el avance parcial consta de: El código fuente con la implementación desarrollada, Makefile, README y la presentación en formato PDF.

## Anexo

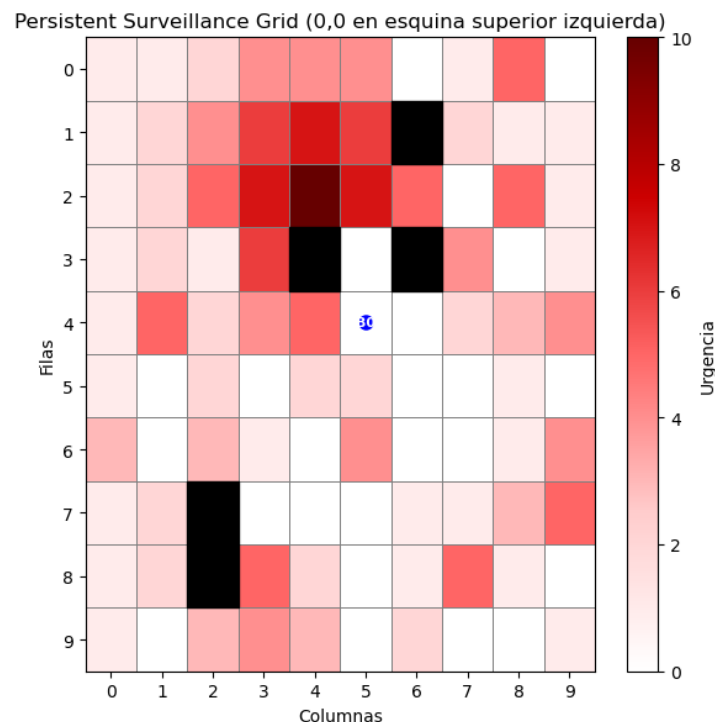


Figura 1: Instancia PSP-UAV\_01\_a

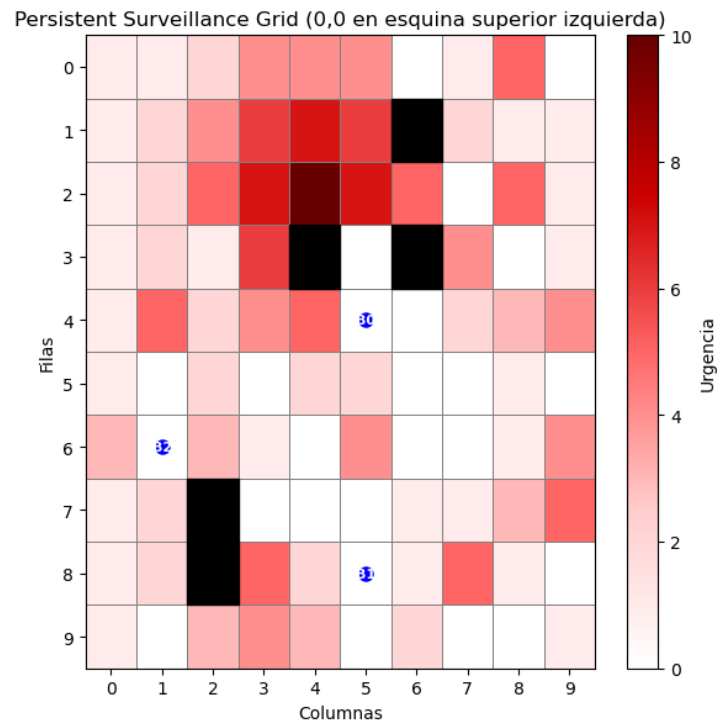


Figura 2: Instancia PSP-UAV\_01\_b

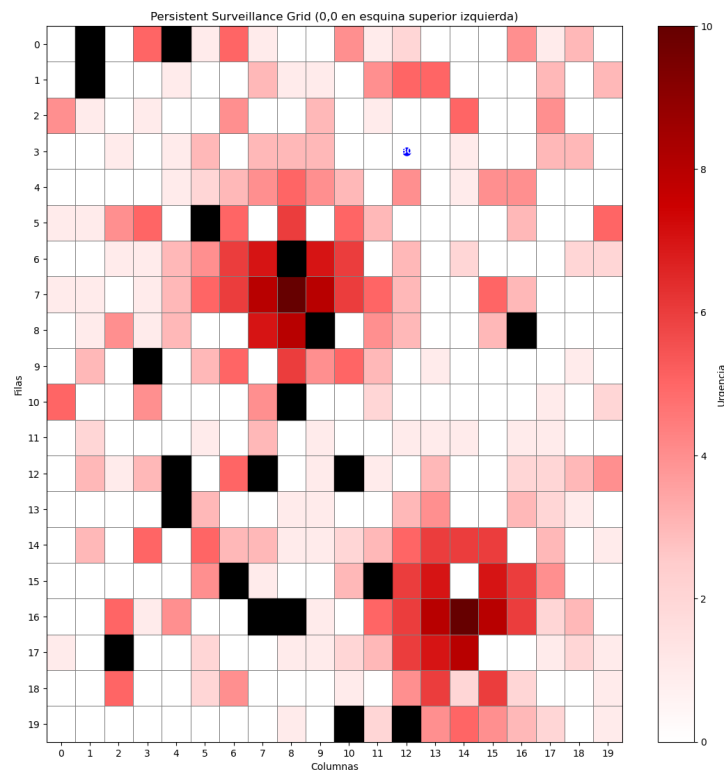


Figura 3: Instancia PSP-UAV\_02\_a

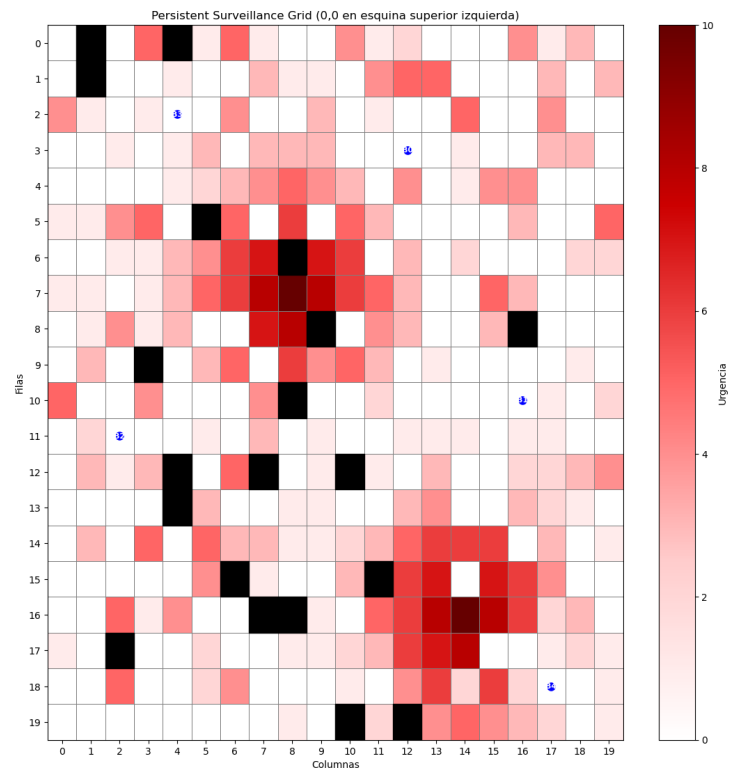


Figura 4: Instancia PSP-UAV\_02\_b

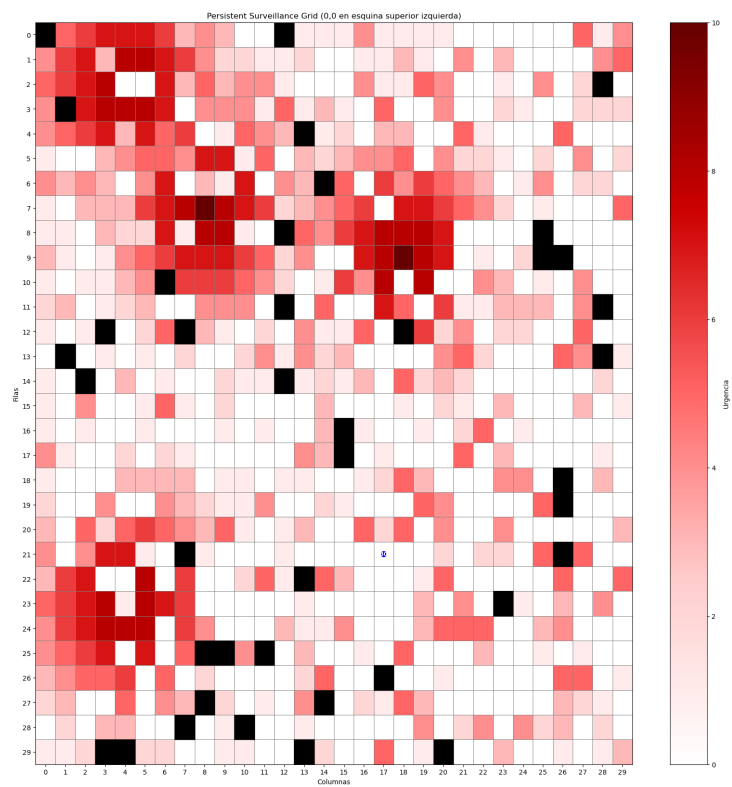


Figura 5: Instancia PSP-UAV\_03\_a

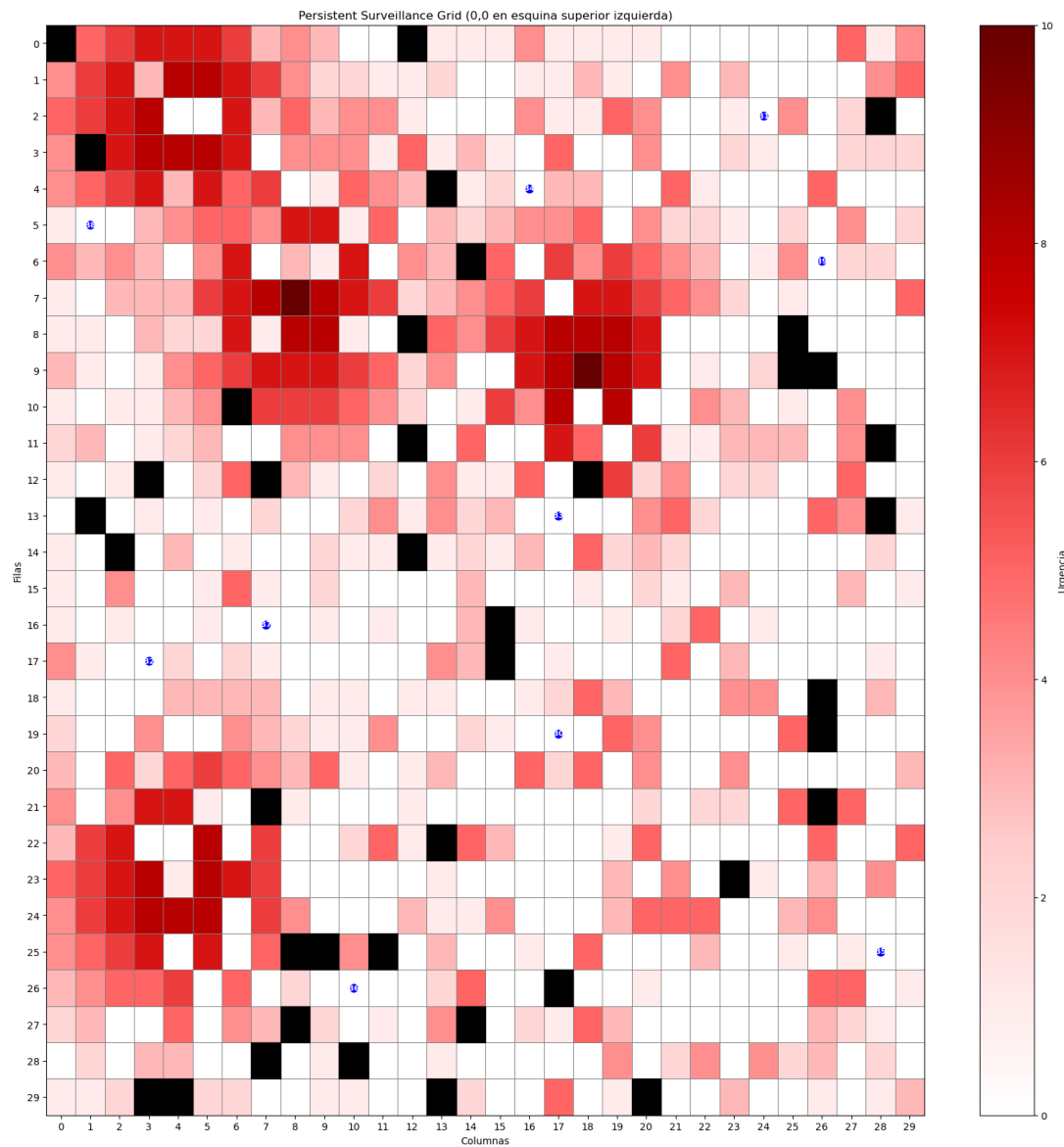


Figura 6: Instancia PSP-UAV\_03\_b