

# Proyecto Integrador: Primer Avance

## Caracterización del Ambiente

Equipo 5

Sebastián Neri                      A01750190

Aldo Sandoval                      A01751137

Emiliano Padilla                      A01658972

### Descripción PEAS

(Performance - Environment - Actuators - Sensors)

<i>Performance</i>	<i>Environment</i>	<i>Actuators</i>	<i>Sensors</i>
El rover tiene la capacidad de recorrer el planeta recolectando muestras. Cuando encuentra una muestra manda una señal con las coordenadas del lugar donde se encuentra el agua.	<u>Zonas rocosas en Marte</u> Es accesible porque todos sus sensores pueden detectar la información disponible del entorno. Es determinista porque el entorno (mapa) es inmutable, está delimitado por bordes y es un agente único el que se encuentra en el mapa. Es no episódico (secuencial) porque las decisiones tomadas por el rover afectan las muestras que tome. Estático porque no hay otro agente junto al rover. Es discreto porque el entorno donde está el rover tiene un número finito de estados.	El extractor de muestras, sistema de movimiento (llantas, motores)	Cámaras, micrófonos, sensores de viento, escáner de entornos, radar

## Programa en Python 3.8.5

Descripción del mapa - Mapa topográfico aleatorio con 6 niveles de altura que son {0, 1, 2, 3, 4, 5}. En el mapa se encuentran lagos representados por el carácter '\_' y obstáculos representados por el carácter '\*' y está delimitado por el carácter '#'. La documentación de la función *crearMapa* se encuentra en el archivo *main.py*.

### Creación del mapa con los bordes

Primero se crea un string vacío al que se le van añadiendo los límites con '#'. Esto pues considerando que en la primera iteración, del for externo, y en la última son puros '#'. Luego consideramos que al inicio se añade y al final de cada ciclo del for anidado se añade un '#'. Para la separación de líneas se concatena un '\n' justo después de que el ciclo anidado haya terminado.

Luego, si 'show\_levels' es True, con la librería de numpy y la representación de nuestros niveles de altura, hacemos una extracción aleatoria de un solo elemento de la lista 'levels' y lo concatena al string del mapa, sino sólo añade un espacio. Además, se añade a la lista 'used\_space' la posición unidimensional de los '#' agregados, esta se determina con la suma del valor  $i + j$  de cada iteración, y no se guarda el espacio usado por la representación de los niveles, dado que esa información para el código no es relevante.

### Insertamos los objetos aleatoriamente en el string

Una vez ya creado el mapa base, se calculan números aleatorios, que fungirán como coordenadas unidimensionales del string del mapa, en este caso va desde 0 hasta la longitud del string. Hay que tomar en cuenta que se genera una coordenada aleatoria que no se encuentra en el espacio ya ocupado que está guardado en la lista 'used\_space'. Después, para añadir el agua, los obstáculos y la nave, se reemplaza el elemento en la posición de la coordenada obtenida con el objeto mediante un slicing del string y se añade la coordenada al grupo de coordenadas del espacio ya usado. Esto es así para el agua, el obstáculo y la nave, en ese orden en específico.

# Tests

## Test 1

### Input:

- n = 2
- m = 2
- Número de obstáculos: 11
- Número de lagos: 18
- Mostrar Niveles: False

### Output:

```
----- TEST 1 -----  
  
No se puede crear ese mapa con ese número de obstáculos y lagos. Intenta otra combinación en la que quepan los lagos, obstáculos y la nave.
```

En este caso se obtuvo este output debido a que las dimensiones mínimas para crear el mapa deben ser de 3x3.

## Test 2

### Input:

- n = 10
- m = 10
- Número de obstáculos: 22
- Número de lagos: 5
- Mostrar Niveles: True

### Output:

```
----- TEST 2 -----  
  
Nave: Q  
Obstáculos: *  
Lagos: _  
  
#####  
#4***1*1#  
#1Q12*1**#  
#4*30_3*0#  
#*5**0*_0#  
#05*3535*#  
#_4*352**#  
#05044***#  
#5_4_3332#  
#####
```

En este caso se observa el mapa generado aleatoriamente con los niveles topográficos, los obstáculos, el agua y la nave, así como bien delimitado y representado en la consola. Se incluye hasta arriba el significado de los caracteres.

## Test 3

Input:

- $n = 0$
- $m = 3000$
- Número de obstáculos: 11012831
- Número de lagos: 181223
- Mostrar Niveles: True

Output:

En este test se obtuvo este output debido a que la combinación de obstáculos, lagos y dimensiones no permiten la creación del mapa, pues no se puede contener todo eso en ese mapa.

## Test 4

Input:

- $n = 100$
- $m = 100$
- Número de obstáculos: 55
- Número de lagos: 722
- Mostrar Niveles: False

Output:

```

----- TEST 4 -----

Nave: Q
Obstáculos: *
Lagos: _

#####
#               *       *           #
# _             _             _     #
# #             Q _           _     #
# # * *               _           #
# # _             _             _     #
# # _           _             _     #
# #               _           _     #
# # _ * _         _           *     #
# # _ _ _         _ _ _         _     #
# # *             _           _ *   #
# #               *           *     #
# # _             _           * *   #
# #               _ _         _     #
# # _             _ *         _ *   #
# # _             _           _ _   #
# # *             _           _     #
# #               _           _     #
# #               _           _     #
# #               _           _     #
#####

```

En este test, podemos ver la correcta generación aleatoria del mapa pero sin los niveles topográficos, una vista más fácil de entender.

## Test 5

### Input:

- n = 15
- m = 30
- Número de obstáculos: 22
- Número de lagos: 45
- Mostrar Niveles: True

### Output:

```
----- TEST 5 -----

Nave: Q
Obstáculos: *
Lagos: _

#####
#452*111_0324210_04_30*13122*#
#232525220445_35_20Q531415_*5#
#2*5003*312103142422*505_0101#
#01503_50_15454255_0251024241#
#_50_033_33201230313*3520*324#
#2*031_4040*4030021__15300443#
#3_45_544231*34135__*2*52424*#
#1031*504441113302325*__00002#
#142351*3__2400_3__*__00223_32#
#5_5_133_*__25340_5003_443014#
#014321413_200322104122_25024#
#542245112_5245_113*553325540#
#15404__2_323044_235554410_*2#
#####
```

Aquí también se representa el mapa correctamente con todos sus elementos bien representados y generados aleatoriamente.

## Referencia

EUROPA PRESS. (2021, 19 febrero). Las increíbles características del rover 'Perseverance' enviado a Marte. *Excélsior*.  
<https://www.excelsior.com.mx/global/las-increibles-caracteristicas-del-rover-perseverance-enviado-a-marte/1433499>