

Descenso al fondo de un cráter en Marte

Marzo de 2024

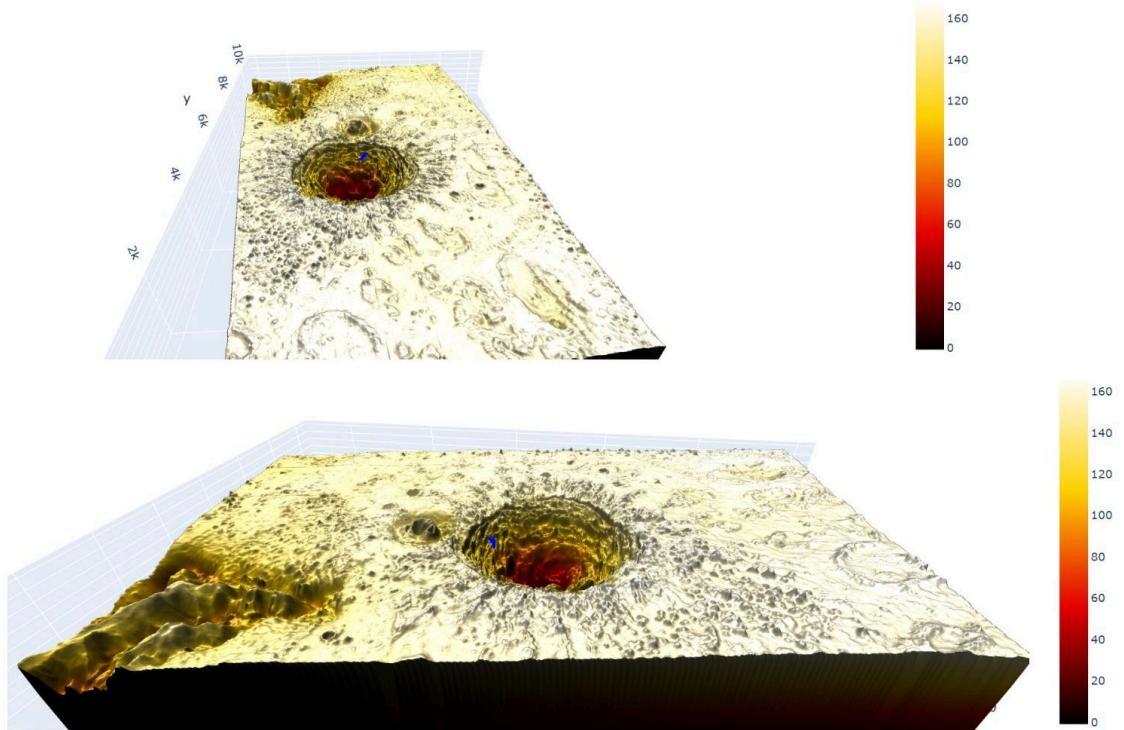
Gabriel Reynoso Escamilla | A01643561

Santiago Mora Cruz | A01369517

Guillermo Villegas Morales | A01637169

Búsqueda Codiciosa

Posición inicial (3500,5800)

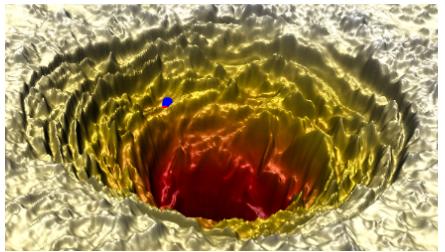
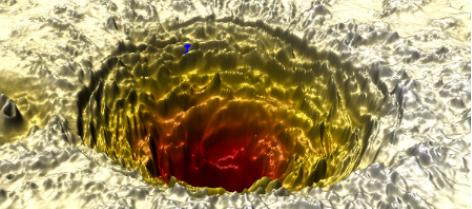
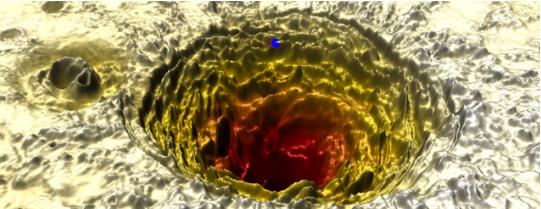
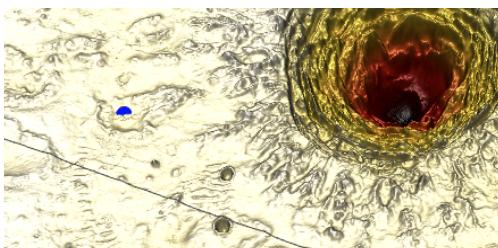
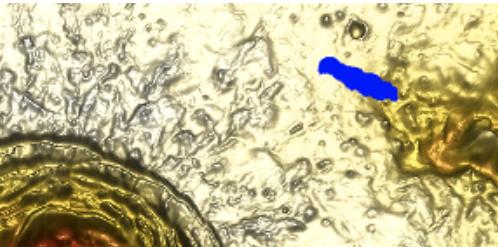


```
3244.535 5735.695
Iteration: 199997    Cost: 105.46416748046897
3244.535 5735.695
Iteration: 199998    Cost: 105.46416748046897
3244.535 5735.695
Iteration: 199999    Cost: 105.46416748046897
3244.535 5735.695
Iteration: 200000    Cost: 105.46416748046897
```

La búsqueda codiciosa no fue capaz de llegar al fondo del cráter, empezó a descender y llegó en el que no encontró un vecino con mejor costo y mejor se quedó en ese punto. Llegó a un mínimo local y se quedó estancado ahí.

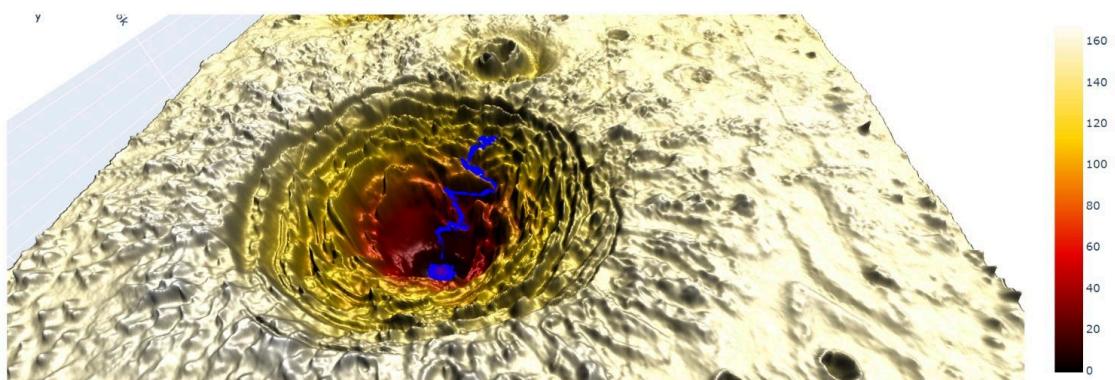
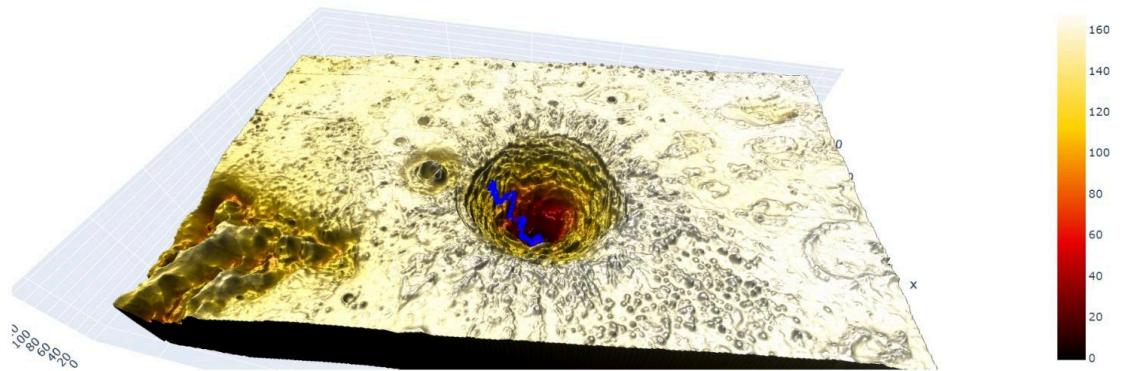
En las pruebas que realizamos, la búsqueda codiciosa nunca llegó al fondo del cráter.

Punto inicial	Ruta
---------------	------

(3425, 4228)	
(4138, 5052)	
(4028, 5065)	
(4731, 2521)	
(1054, 6840)	

Búsqueda con recocido simulado

Posición inicial (3500,5800)

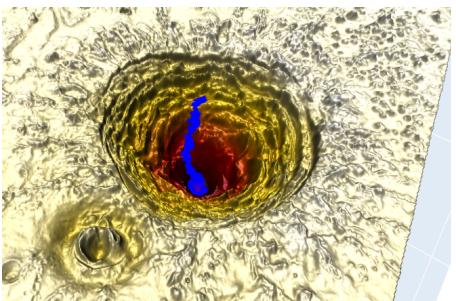


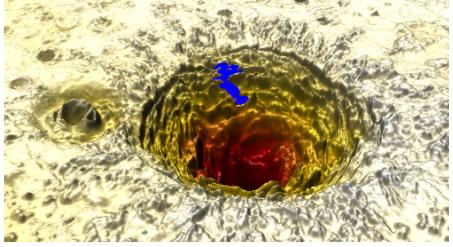
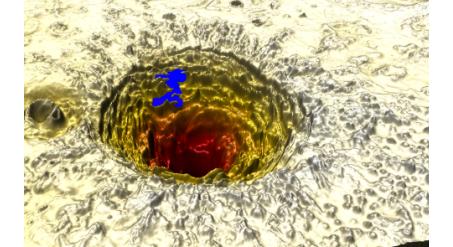
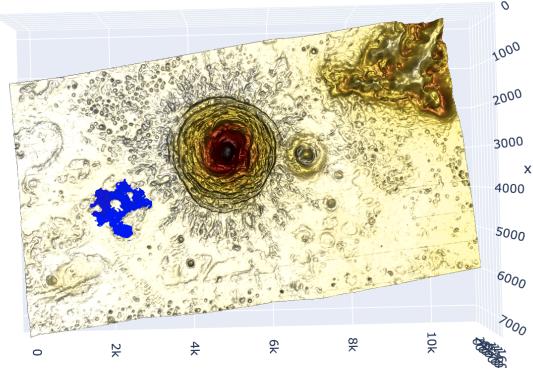
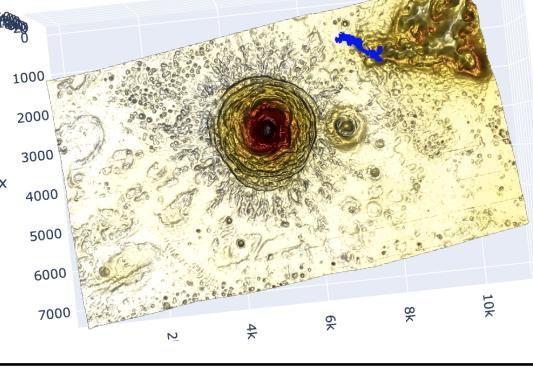
```

2953.23 4952.185
Iteration: 264912    Cost: 0.26782226562521827    Temperature: 0.005000221923106331
2953.23 4952.185
Iteration: 264913    Cost: 0.26782226562521827    Temperature: 0.005000121918667868
2953.23 4952.185
Iteration: 264914    Cost: 0.26782226562521827    Temperature: 0.0050000219162294945
2953.23 4952.185
Iteration: 264915    Cost: 0.26782226562521827    Temperature: 0.00499992191579117

```

Con este algoritmo de búsqueda el rover fue capaz de llegar al punto más bajo del cráter, esto fue posible porque en un principio aceptaba vecinos menos ambiciosos lo que le permite avanzar y mientras se “enfría” busca los mejores vecinos para cambiar de posición.

Punto inicial	Ruta
(3425, 4228)	

(4138, 5052)	
(4028, 5065)	
(4731, 2521)	
(1054, 6840)	

Conclusiones

Gabriel Reynoso:

Esta fase del reto nos permitió explorar los algoritmos de búsqueda local, en un principio fue sencillo porque conocíamos la implementación de algunos problemas de búsqueda local y la búsqueda informada en el otro mapa de marte, las dificultades surgen al definir que es un vecino para el rover en un primer momento intentamos hacerlo con ciclos *If*, similar a los algoritmos de búsqueda informado, esto no fue útil para la búsqueda local sin embargo nos dimos cuenta del error a tiempo y formulamos los vecinos con movimientos aleatorios del

marciano restringidos por la altura que puede escalar. También nos fue difícil ajustar la función que genera el mapa con el camino. Creo que un problema de este tamaño nos permite realmente observar como se comportan estos diferentes algoritmos y en qué situaciones son útiles, también fue muy claro el porqué “Búsqueda local” el algoritmo solo era útil cuando estaba cerca del cráter.

Guillermo Villegas:

El reto resultó relativamente sencillo al ya contar con la solución y lógica del robot explorador pasado, traducir los implementación de un problema de búsqueda a búsqueda local fue cuestión de hacer unos cuantos ajustes a la clase del robot. Nuestra dificultad más grande fue que el camino hacia el fondo del cráter no era visible a simple vista, por lo que no teníamos idea de si el robot iba por buen camino o si estaba completamente perdido. Por buen rato estuvimos corriendo el nuevo código con el mapa anterior, por lo que nos atoramos cuando vimos que el costo del recocido simulado. Esto nos llevó a tener dudas de nuestra implementación, pero al darnos cuenta del error y hacer unos cuantos ajustes, encontramos que el robot llegaba relativamente rápido hasta el fondo con el recocido simulado. No se puede decir lo mismo sobre la búsqueda codiciosa, que en ninguna ocasión logró llegar al fondo. Esto porque el mapa tenía muchos picos, lo que confunden a la búsqueda codiciosa. Con esto aprendimos que la aleatoriedad y el azar son buenas herramientas para la solución de un problema de búsqueda local.

Santiago Mora:

En esta ocasión, la realización del proyecto del robot explorador en marte fue diferente en el sentido en el que ahora se trata como un problema de búsqueda local, lo cual trae consigo sus propias formas de implementación y sus propios retos y dificultades. Para empezar, yo siento que la parte más difícil (pero al mismo tiempo fácil) fue la definición de un vecino en la clase, ya que aunque al tener el mapa era mucho más fácil definir el vecino utilizando la posición del rover, pero al mismo tiempo teníamos que lograr que este cambio de posición fuera aleatorio, pero confirmar que fuera válido. Al final lo logramos utilizando un randomizador de una lista con el -1, 0 y 1 para la posición en x y la posición en y y un if con 5 condiciones and. Después llegó otro reto para el equipo, cuando pensamos que la implementación de los vecinos estaba mal porque la búsqueda voraz tiende a moverse muy poco (o en ocasiones casi no moverse), y pensábamos que los vecinos estaban mal definidos en la clase. Ante esto, probamos la búsqueda aleatoria en el rover y nos dimos cuenta que no era un error de implementación, sino que simplemente la búsqueda voraz puede llegar a ser bastante inútil. Al probar después con el recocido simulado tuvimos instancias donde solo descendía a una altura de unos 50 metros, pero al hacer más pruebas vimos que si había una ruta que descendiera al fondo y que sí se poesía lograr, pero que depende mucho de los movimientos aleatorios que haga el rover al principio, cuando la temperatura aún es alta en el recocido simulado. En general, fue una buena experiencia de aprendizaje que trajo sus propios y únicos retos y dificultades.