



Jorge Eduardo Guijarro Márquez A01563113

Claudio José González Arriaga A00232276

Alan Rojas López A01706146

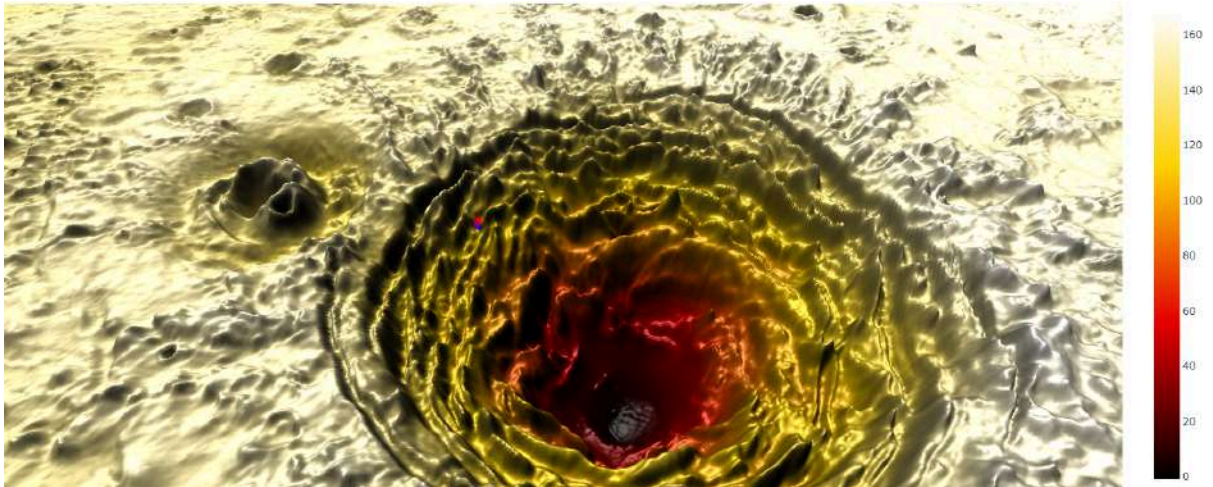
Andrea Ruiz Alvarez A01644031

**Actividad: Descenso al fondo de un cráter en
Marte**

**Diseño de agentes inteligentes
Grupo 103**

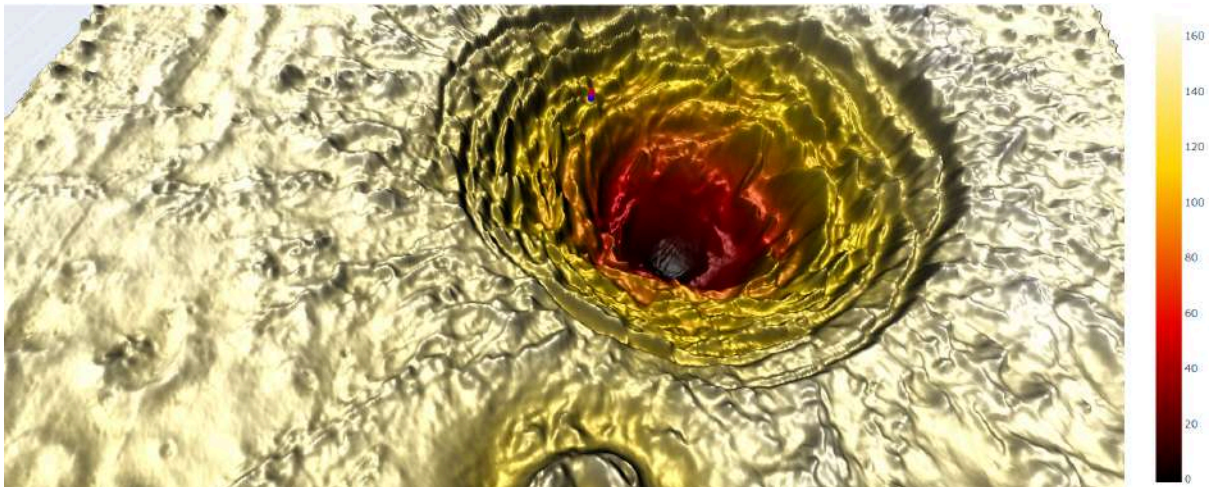
Pruebas con algoritmo de búsqueda voraz

- (3350, 5800)



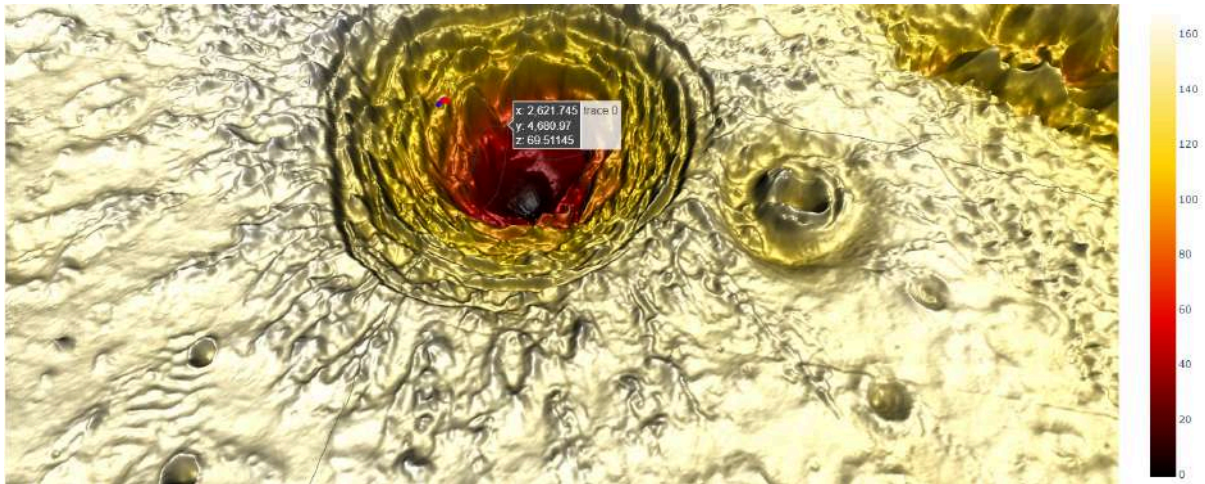
Total distance: 34.529017204943244

- (3676, 5800)



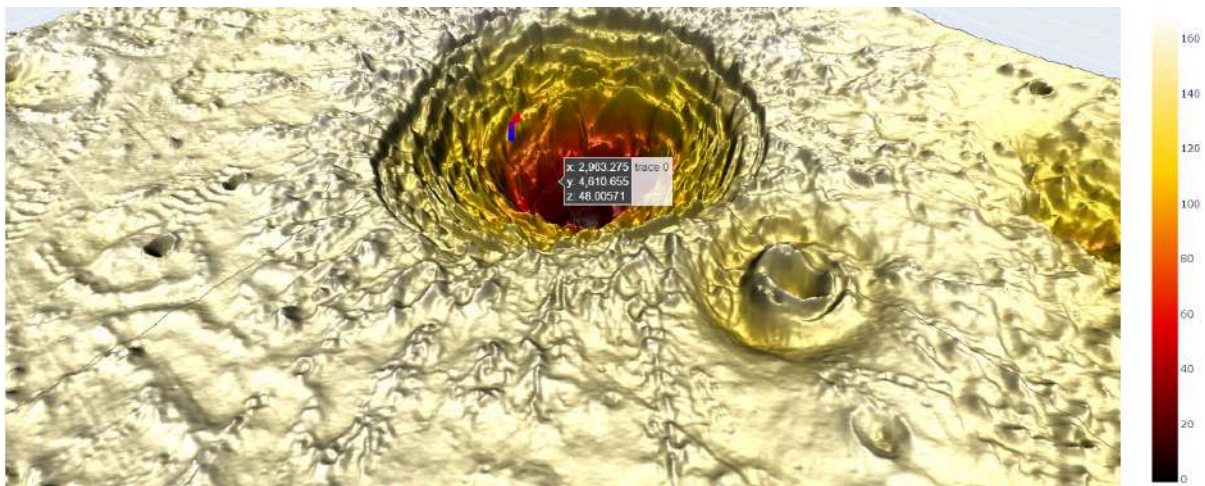
Total distance: 24.268538448285376

- (2802, 4218)



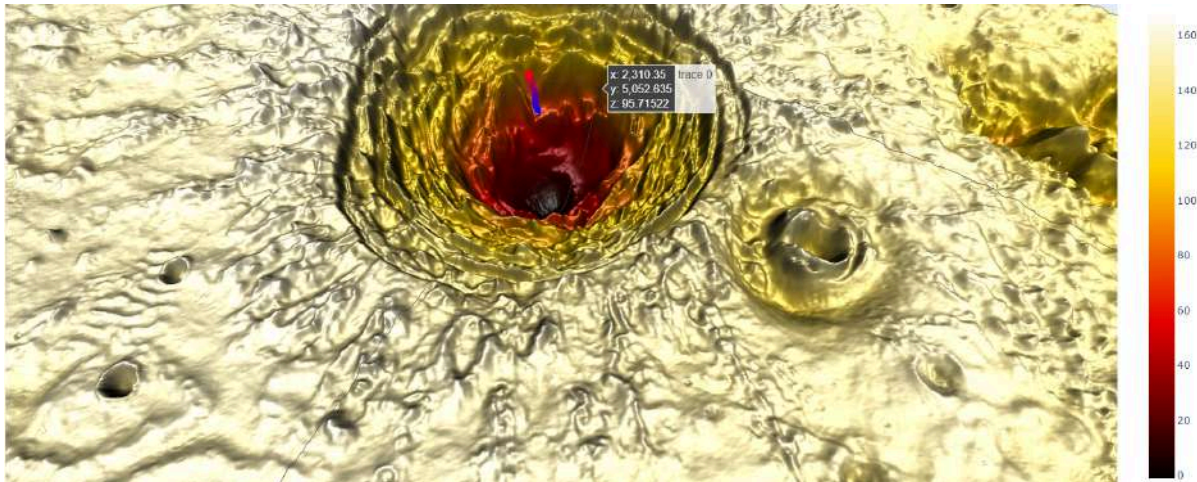
Total distance: 91.4027226115762

- (3184, 4350)



Total distance: 182.59971371264737

- (2591,4640)

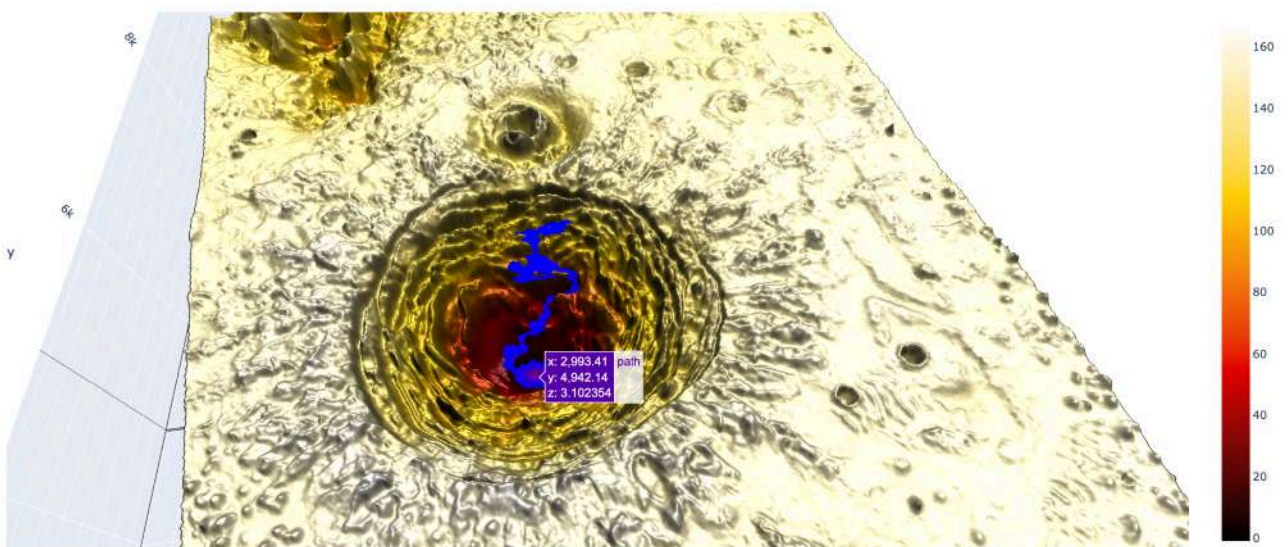


Total distance: 266.3007485110547

Durante las pruebas realizadas con búsqueda voraz, observamos que el algoritmo mostró una capacidad bastante mala para desempeñar la tarea, logrando ejecutar solamente entre 5 y 15 iteraciones en las 5 pruebas realizadas. El algoritmo no fue capaz de encontrar soluciones óptimas o cercanas a él durante los casos probados y el algoritmo no logró cumplir el objetivo de alcanzar el fondo del cráter de manera satisfactoria.

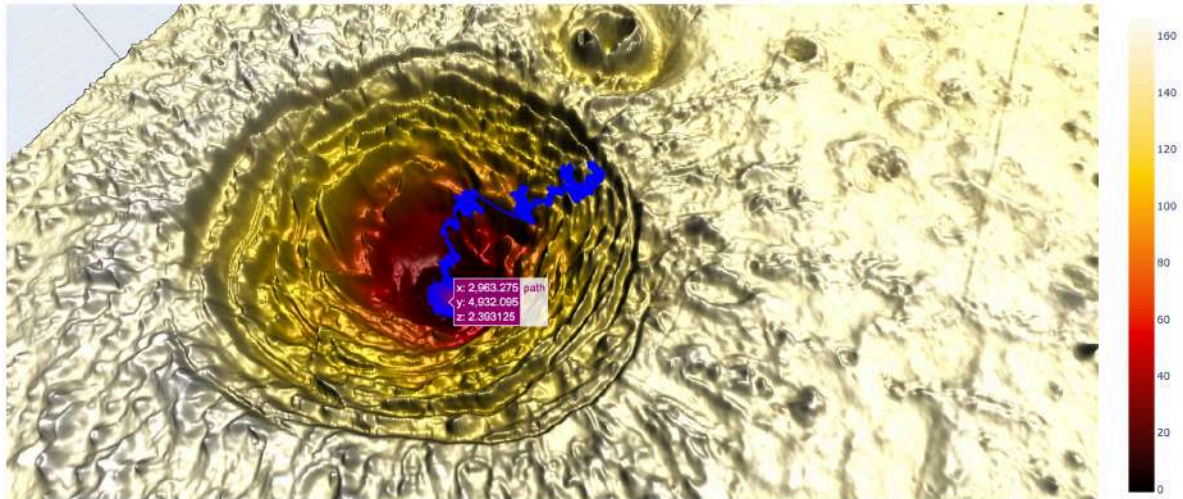
Pruebas con Recocido simulado

- (3350, 5800)



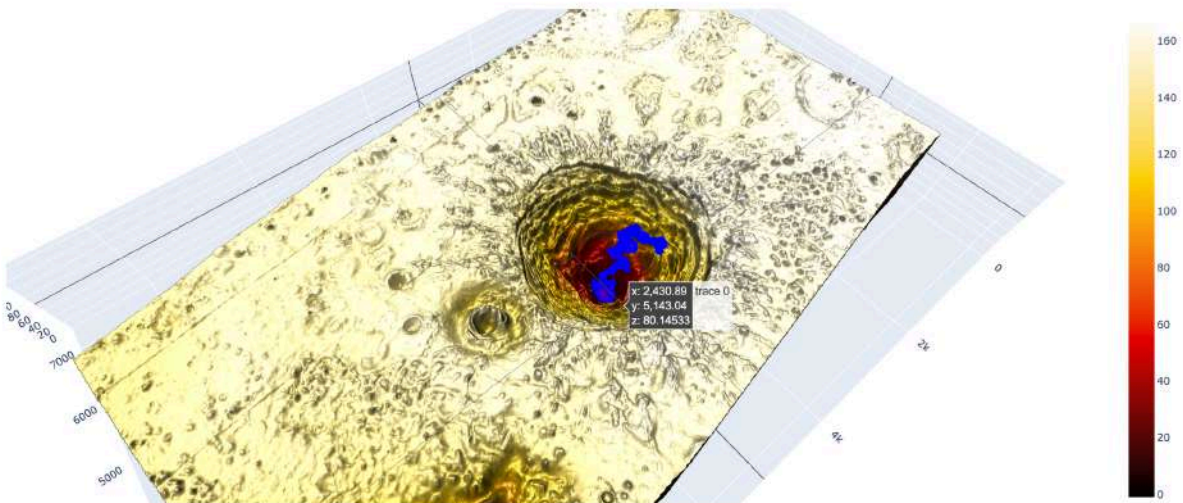
Total distance: 531506.430401794

- (3676, 5800)



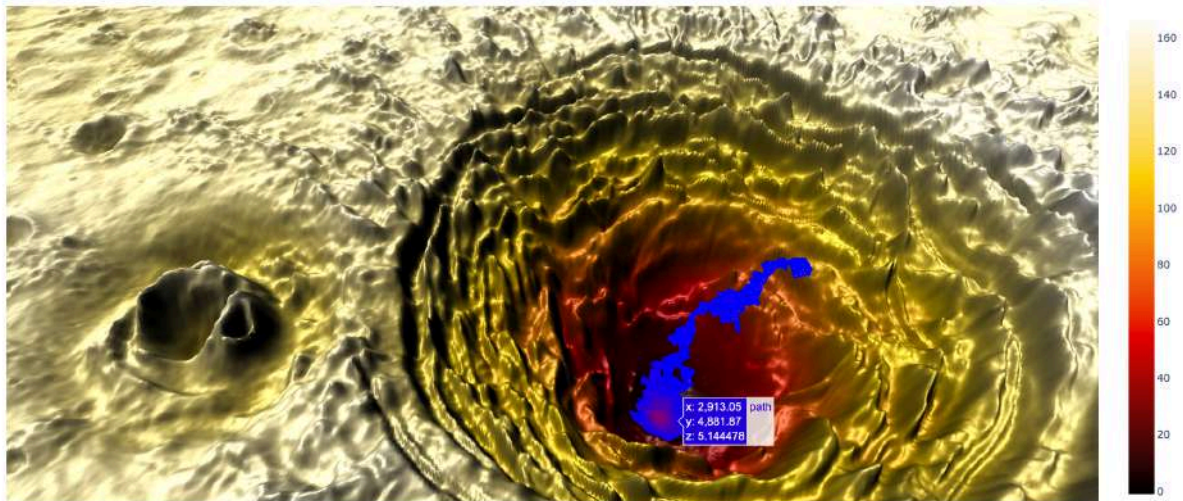
Total distance: 533198.1145631725

- (2802, 4218)



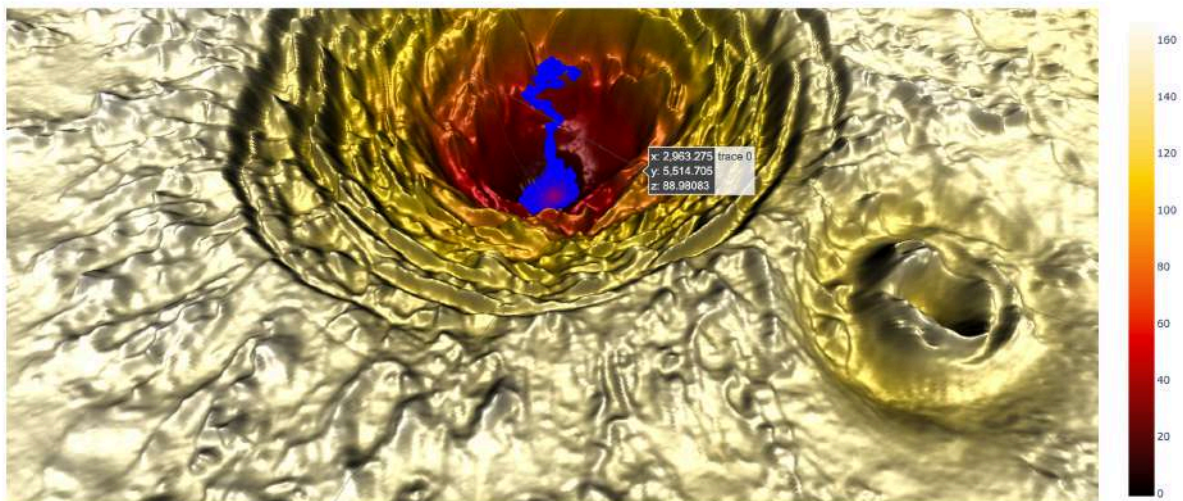
Total distance: 534717.0942650464

- (3184, 4350)



Total distance: 533886.222506087

- (2591,4640)



Total distance: 540063.9460556079

Como podemos observar, el algoritmo de recocido simulado logra recorrer una distancia mucho más largo, esto debido a la naturaleza misma del algoritmo el cual permite emporar localmente para mejorar globalmente.

¿Qué algoritmo logra llegar más profundo en el cráter?

- El algoritmo de recocido simulado llega más profundo al cráter. Posiblemente debido a su función de temperatura, que le permite tomar decisiones que le lleven a escapar de mínimos (o máximos, según el problema) locales, “empeorando” su

estado actual a cambio de una ruta nueva con posibles mejores resultados, mientras que algoritmo de búsqueda codiciosa tiene a quedar atascado en mínimos locales razón por la cual no puede llegar al fondo del cráter. Como vimos en las pruebas, el algoritmo de recocido simulado, por mucho, logra resultados con más distancia recorrida.

¿Recomendarían a los ingenieros del robot utilizar alguno de estos algoritmos?

- Sí, definitivamente el algoritmo de recocido simulado es el candidato a recomendar para los ingenieros del robot. Somos conscientes de que probablemente exista un algoritmo más eficiente en cuanto a distancia recorrida y número de iteraciones, sin embargo, debido a su bajo tiempo de ejecución, y baja demanda de recursos computacionales y financieros, el recocido simulado es una recomendación sólida.

Conclusiones Individuales

Jorge:

Durante el desarrollo de esta actividad, pudimos comparar dos métodos de algoritmos de búsqueda local, probarlos en este contexto, realizar varias pruebas y jugar con los parámetros fue una práctica bastante enriquecedora para conocer no únicamente la teoría de ellos, sino aprender cómo se desempeñan durante la práctica.

Pudimos notar como el algoritmo de búsqueda voraz se desempeñó pésimamente en este contexto, sin embargo, no podemos demeritar tan fácilmente a este algoritmo, simplemente fue peor en este contexto. El algoritmo de recocido simulado, desempeñó un papel sumamente importante para la realización de la actividad y para alcanzar el objetivo deseado, ya que ayudó a demostrar que un agente explorador que implemente este tipo de búsqueda, puede ser muy eficiente y hacer un buen trabajo para este tipo de investigaciones que involucren exploraciones en un futuro.

En resumen, esta experiencia nos ha dejado con una comprensión más profunda de la importancia de elegir el algoritmo adecuado para cada situación específica, y de igual manera la importancia que puede cargar probar distintas técnicas, que incluso, solo se pueden aprender experimentando para lograr que estas soluciones sean las más eficientes posibles y abordar de mejor manera las situaciones.

Alan:

Esta actividad logró integrar bastante bien los algoritmos de búsqueda local, de tal manera que mostraban sus pros y sus contras. Por ejemplo, en el caso del algoritmo de recocido simulado, es interesante cómo es que casi siempre llega a una solución, siendo bastante efectivo con el número de iteraciones. A contraparte, el algoritmo de búsqueda voraz, casi nunca encontraba una solución, aunque en las iteraciones que hacía, era muy efectivo (de hecho ese es su propósito).

Referente al problema en sí, podemos concluir varias cosas. Uno, considero que, al igual que en las actividades anteriores, lo que más me costó fue la programación y planteamiento del problema: no tanto los algoritmos. Dos, el problema en sí es realista para un agente inteligente, por lo que veo que está bien implementado el reto. Y tres, en cuanto al resultado, me parece curioso cómo es que el comportamiento de un agente inteligente, que antes creía complejo e imposible de replicar, es relativamente fácil de aplicar a problemas de la vida real.

En conclusión, con este reto logré poner a prueba mis habilidades de programación, planteamiento de problemas y aplicación de algoritmos, en un escenario realista y de valor.

Claudio:

La búsqueda codiciosa, con su enfoque directo y eficiente, inicialmente parecía prometedora. Sin embargo, su tendencia a estancarse en mínimos locales resultó ser un obstáculo significativo. A pesar de su simplicidad, este enfoque demostró ser insuficiente para guiar al robot explorador hacia el fondo del cráter en muchos casos, dejándolo atrapado en alturas subóptimas.

Por otro lado, el algoritmo de recocido simulado me impresionó con su capacidad para evitar obstáculos y explorar el espacio de búsqueda de manera más efectiva. Al introducir aleatoriedad y permitir soluciones subóptimas, este enfoque demostró ser más adaptable y resistente a los mínimos locales y dado el entorno poco uniforme como lo es la superficie de un cráter de Marte, por lo cual el rover logró descender más profundamente en el cráter, incluso alcanzando el fondo en varias ocasiones.

Fue algo retador el planteamiento de la lógica para implementar la lógica de la selección de los vecinos en torno a la profundidad de estos sin embargo, recomendaría el uso del algoritmo de recocido simulado para la misión del robot explorador. Su capacidad para encontrar soluciones óptimas en entornos desafiantes y su resistencia a quedar atrapado en mínimos locales lo convierten en la opción ideal. Aunque puede requerir más recursos computacionales que la búsqueda codiciosa, su eficacia y fiabilidad son invaluable para garantizar un descenso exitoso y seguro en el cráter. Fue gratificante entender y explorar de primera mano los algoritmos de búsqueda local y sobre todo en el algoritmo de

recocido simulado, ver como algo que podría resultar contradictorio sin un contexto como lo es seleccionar soluciones subestimas localmente llevan a una mejor solución.

Andrea:

Esta situación planteada fue mi retadora en varios aspectos. Tuvimos dificultad al hacer la búsqueda voraz ya que no nos dio un gran resultado o el resultado que esperábamos. El recocido simulado nos dio mejores resultados. Esto es porque con la búsqueda voraz nos estancamos en mínimos locales lo cual no nos permite mejorar mientras avanzaba el algoritmo. Al aplicar el recocido simulado, se escogió una ruta distinta y se escogieron siguientes movimientos de una manera mas óptima, lo cual nos permitió tener una mayor distancia. Esta actividad es un claro ejemplo de cómo no existe un mejor algoritmo de búsqueda, todo depende del contexto de la situación. <para cada situación es muy importante analizarla para escoger el mejor algoritmo posible para ella personalmente. El programar todavía me cuesta un poco y fue retador en este reto. Mos costó trabajo pero esta actividad me ayudó a entender de una manera más amplia los algoritmos de búsqueda y cuando usarlas. Para concluir, esta actividad fue muy enriquecedora en el aspecto de que me enseñó a que tenemos que analizar un problema antes de atacar con un algoritmo de búsqueda.. Esto es para escoger de una mejor manera el mejor algoritmo de búsqueda posible.