Actividad individual. Desarrollo de un problema de planificación

# Objetivo

En esta actividad se implementa las herramientas que dan solución a un problema de planificación, para ello es importante tener presente la definición del problema de planificación, conocimiento de herramientas de desarrollo en contenedores, ejecución de software alojado en contenedores, abstracción de problemas y descripción de problemas en lenguaje PDDL.

# Desarrollo

Para llevar a cabo la práctica, era necesario contar con un equipo basado en plataforma x86, esto era necesario debido a que el compilador de Go requiere de esta plataforma y no existe alguna forma de que funcione en ARM, ya sea virtualizando un sistema operativo o usando un contenedor.

Para asimilar los conceptos de PDDL (Planning Domain Definition Language) se usa el software de Singularity (Singularity, 2020) disponible únicamente en sistemas basados en Linux. Para entender y saber cómo funciona usaremos la página de la competencia de International Conference on Automated Planning and Scheduling y usaremos los recursos del equipo ganador de la edición 2018. (ICAPS Inc, 2024)

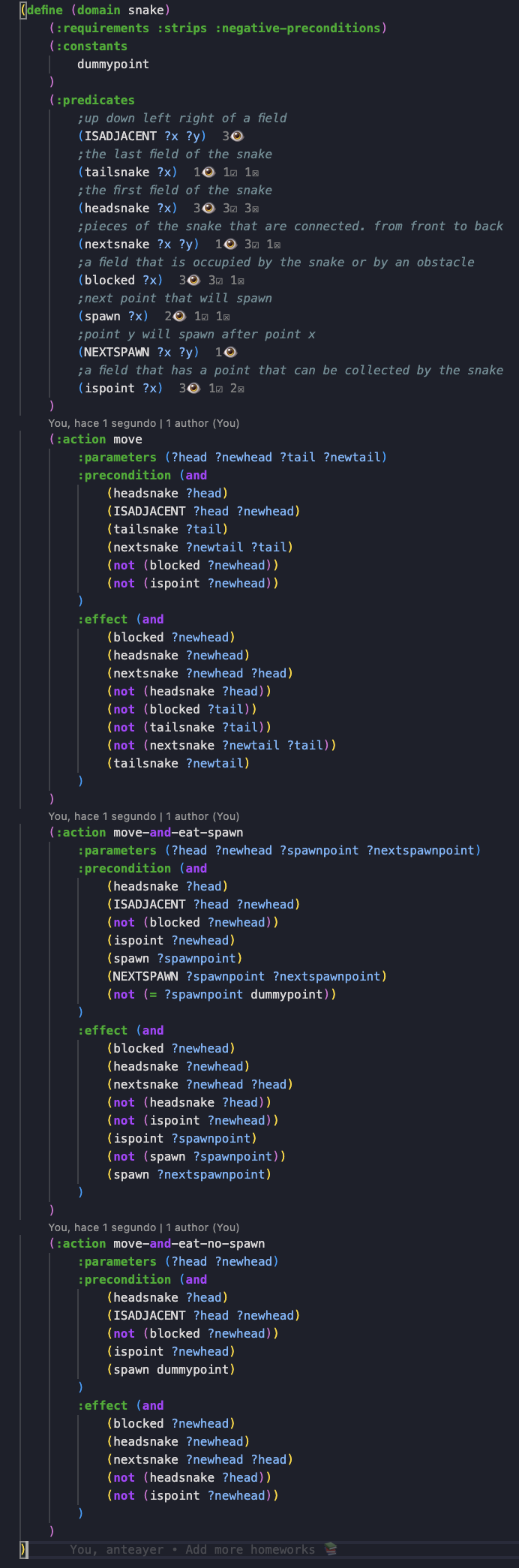
El primer problema para resolver es el Snake, a continuación, se muestran imágenes respecto al código de dominio y problema.  


Ilustración 1PDDL Dominio Snake

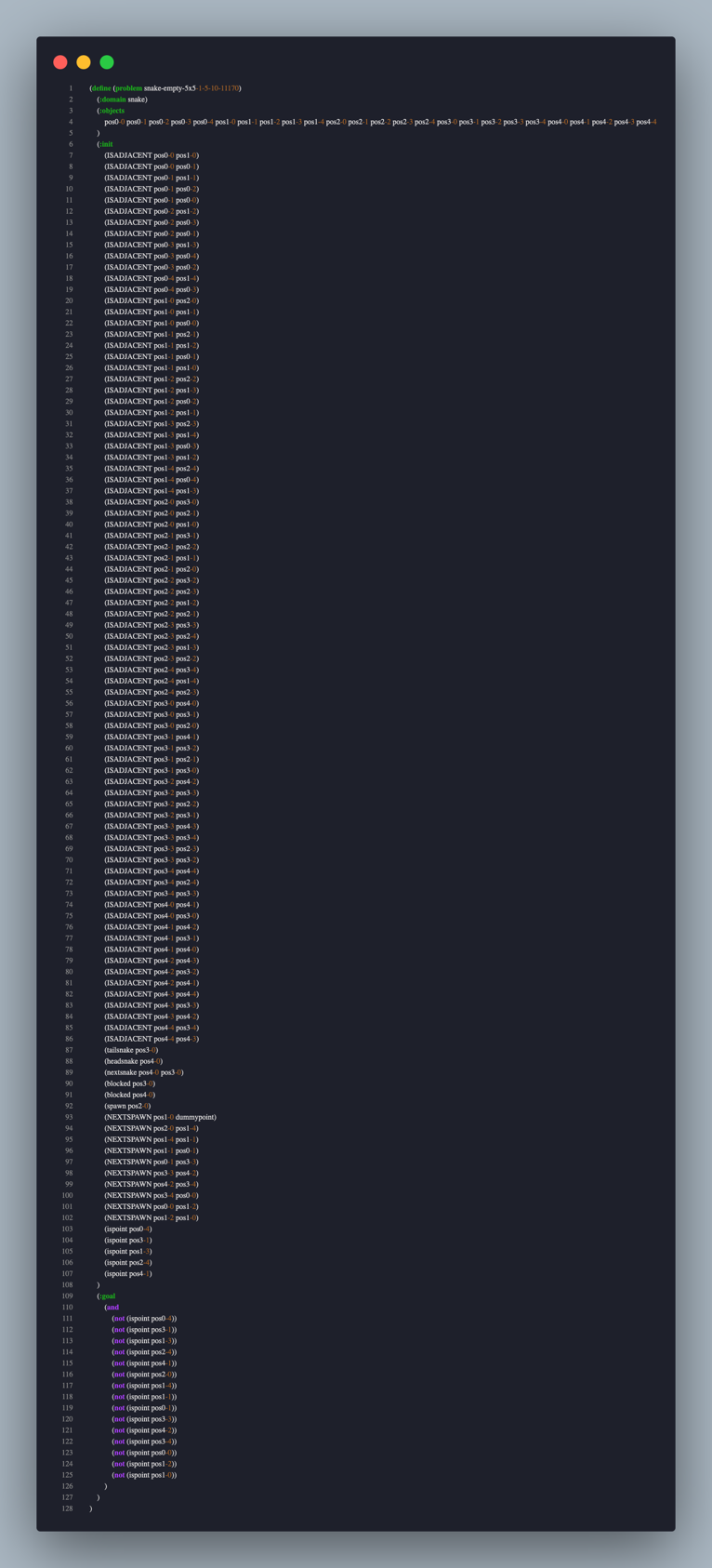


Ilustración 2 PDDL Problema Snake

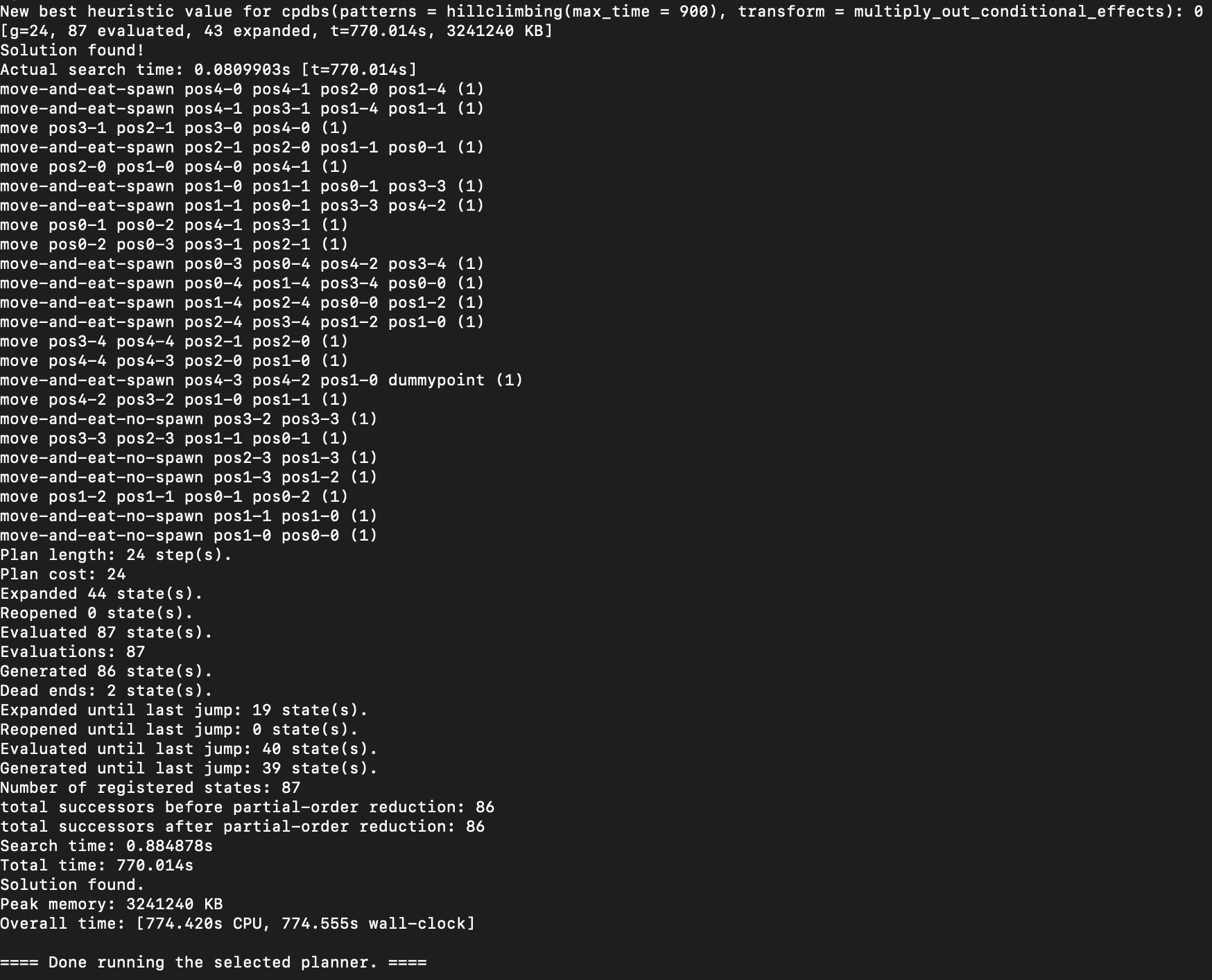
Donde tenemos el siguiente resultado:

Ilustración 3 Solución al problema Snake

Podemos analizar y entender que para resolver problemas usando PDDL, se requieren de dos archivos, uno llamado dominio y otro llamado problema. En los archivos de dominio, contienen las reglas para resolver los problemas, ya sea moverse o realizar una acción y en los archivos de problema, se usa para establecer el problema a resolver, ya sea describiendo un mapa donde se localizan objetos o caminos donde las instrucciones pueden ejecutarse.

Ahora para poder entender el uso, se resuelve el siguiente problema:

Ilustración 4 Problema a resolver

Se tiene un robot tipo *rover* que previamente realizó la excavación de dos rocas localizadas en la localidad 1 y localidad 2. Debido al mal tiempo, no fue posible trasladar las rocas para su análisis. Es por ello por lo que se solicita generar el plan que debe seguir el robot para llevar los minerales al laboratorio de análisis.

Debido al terreno, hay restricciones en la trayectoria: de la localidad 3 a la 1, el camino está libre y existe bidireccionalidad; de la localidad 3 a la 2, el camino solo es de una dirección; de la localidad 2 a la 4 es solo una dirección; de la localidad 3 a 4 y de 4 a 5, el camino es bidireccional.

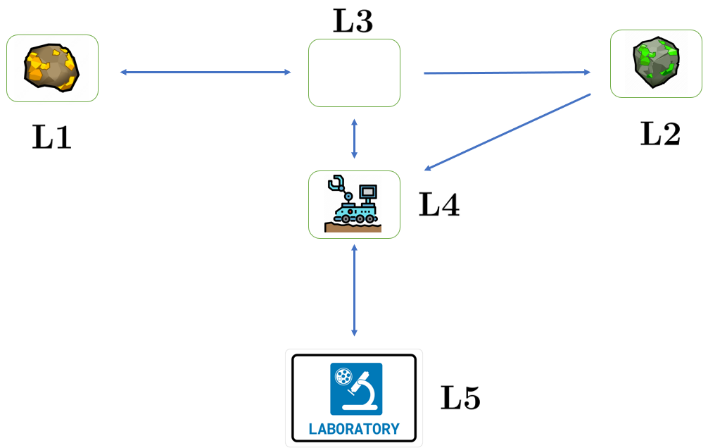
Con la descripción del problema a resolver, se tiene que codificar el dominio y el problema a resolver, usando las configuraciones que realizó el equipo ganador. Las imágenes a continuación describen el código del dominio y del problema:



Ilustración 5 PDDL Dominio Minerales

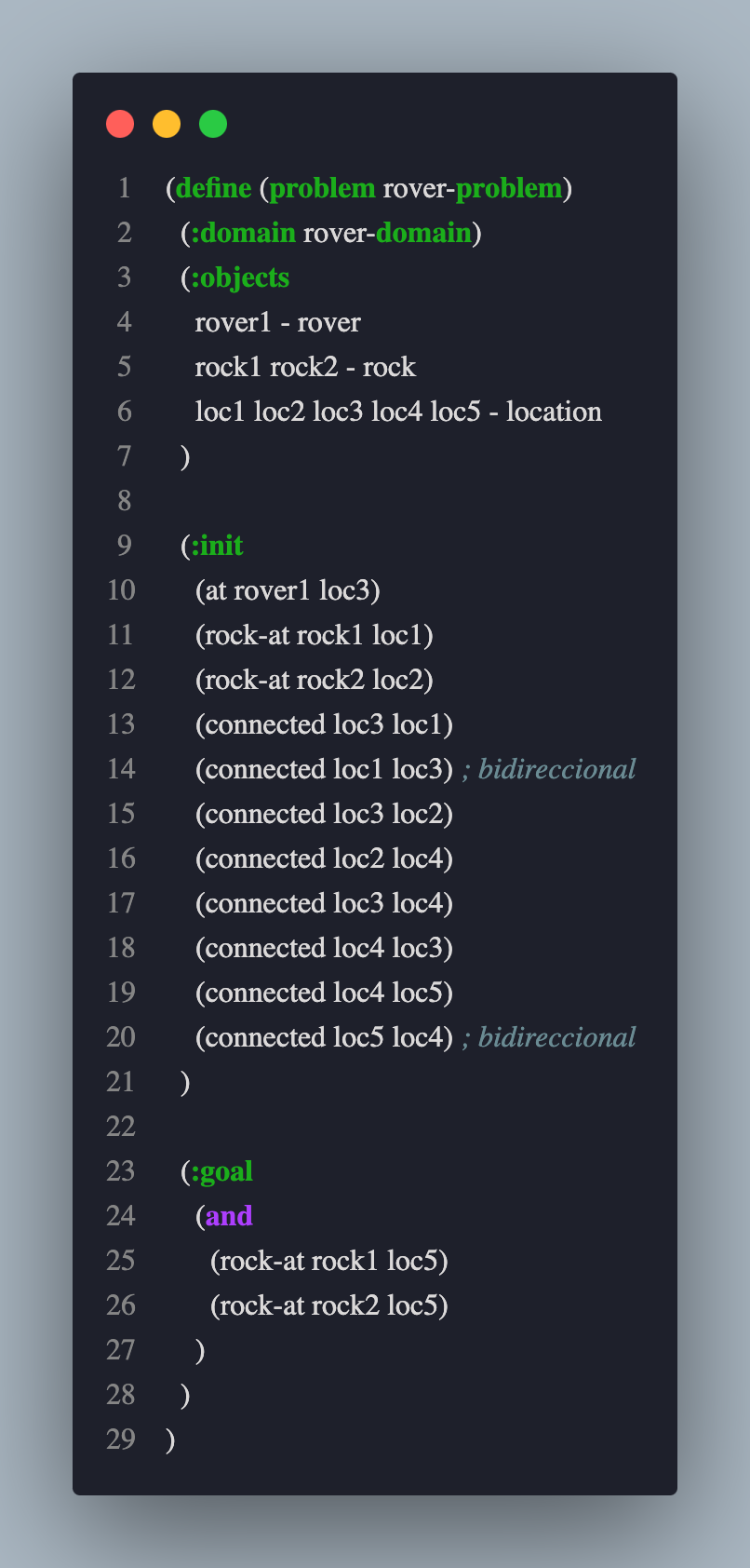


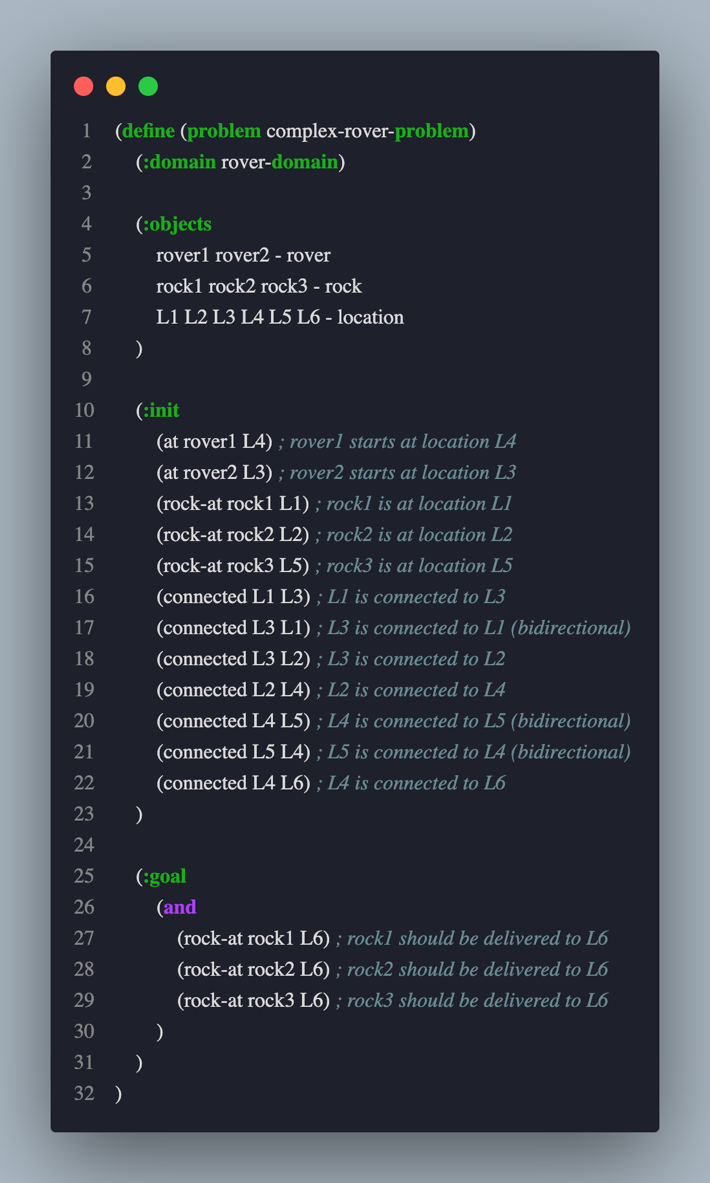
Ilustración 6 PDDL Problema Minerales

Ilustración 7 PDDL Resultado Minerales

 **Explicación:**

* **Tipos**: Definimos tres tipos: location, rover y rock.
* **Predicados**:
  + (at ?r - rover ?l - location): El rover está en una ubicación específica.
  + (rock-at ?rock - rock ?l - location): La roca está en una ubicación específica.
  + (carrying ?r - rover ?rock - rock): El rover está llevando una roca.
  + (connected ?l1 - location ?l2 - location): Dos localidades están conectadas y el rover puede moverse entre ellas.
* **Acciones**:
  + move: Mover el rover entre localidades conectadas.
  + pick-up: Recoger una roca.
  + drop: Dejar una roca.
* **Estado inicial**:
  + El rover (rover1) empieza en loc3.
  + rock1 está en loc1 y rock2 en loc2.
  + Definimos las conexiones entre localidades según las restricciones dadas (bidireccionalidad y unidireccionalidad).
* **Objetivo**: Mover ambas rocas (rock1 y rock2) a la ubicación del laboratorio (loc5).

Este modelo en PDDL permite al planner generar un plan que el rover siga para mover las rocas al laboratorio de acuerdo con las restricciones del terreno.

El siguiente detalle a resolver es modificar el problema ampliando los detalles del anterior problema.

**Explicación:**

1. **Objetos**:
   * **Rovers**: rover1 y rover2 son los dos rovers disponibles en este problema.
   * **Rocas**: Hay tres rocas: roca1, roca2 y roca3.
   * **Localidades**: Hay seis localidades etiquetadas de L1 a L6.
2. **Estado Inicial**:
   * **Posiciones de los Rovers**: rover1 comienza en L4, y rover2 comienza en L3.
   * **Ubicaciones de las Rocas**: roca1 está en L1, roca2 está en L2, y roca3 está en L5.
   * **Conectividad**: Existen caminos bidireccionales y unidireccionales entre ciertas localidades, modelados con el predicado connected.
3. **Estado Objetivo**:
   * El objetivo es entregar roca1, roca2 y roca3 a la ubicación L6.

Este escenario permite simular una situación donde múltiples agentes autónomos (rovers) deben colaborar y planificar sus movimientos y acciones para alcanzar un objetivo común bajo un conjunto de restricciones, representando un desafío realista en robótica y logística.

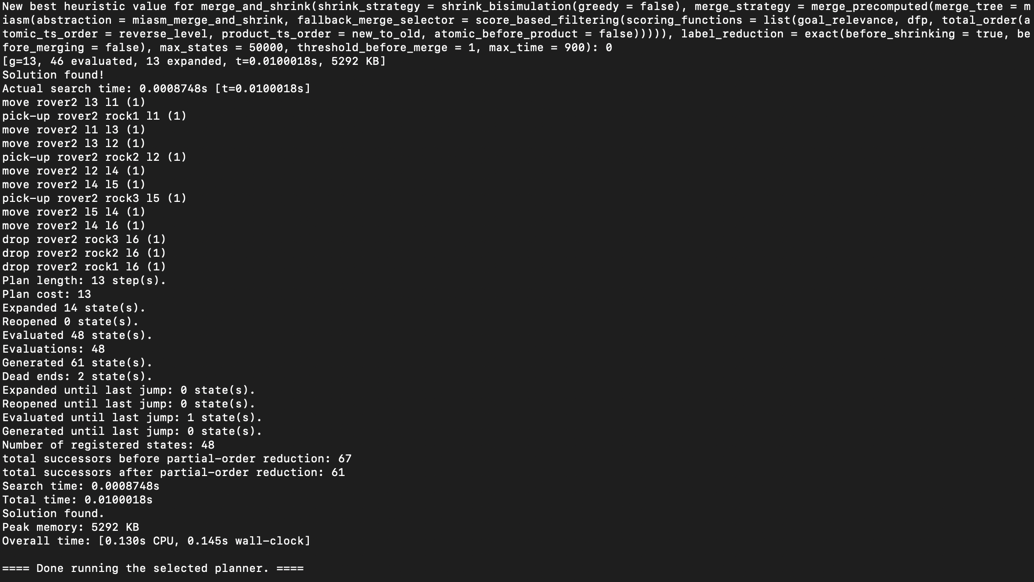
Dando como resultado el siguiente:

Ilustración 8 PDDL Resultado modificado de minerales

Por lo tanto, PDDL (Planning Domain Definition Language) ofrece una estructura clara y formal para modelar problemas de planificación, permitiendo la representación precisa de acciones, precondiciones, efectos, y estados. Esto facilita el diseño de problemas complejos de forma que puedan ser entendidos y procesados por planificadores automáticos.

La capacidad de definir tipos, predicados y acciones en PDDL proporciona flexibilidad para modelar diversos dominios, desde problemas simples de transporte hasta complejas misiones de exploración espacial.

# Tabla de ilustraciones

[Ilustración 1PDDL Dominio Snake 2](#_Toc175779649)

[Ilustración 2 PDDL Problema Snake 3](#_Toc175779650)

[Ilustración 3 Solución al problema Snake 4](#_Toc175779651)

[Ilustración 4 Problema a resolver 5](#_Toc175779652)

[Ilustración 5 PDDL Dominio Minerales 6](#_Toc175779653)

[Ilustración 6 PDDL Problema Minerales 7](#_Toc175779654)

[Ilustración 7 PDDL Resultado Minerales 8](#_Toc175779655)

[Ilustración 8 PDDL Resultado modificado de minerales 10](#_Toc175779656)

# Bibliografía

ICAPS Inc. (16 de Agosto de 2024). *International Conference on Automated Planning and Scheduling*. Obtenido de ICAPS Inc: https://www.icaps-conference.org/competitions/

Singularity. (2020). *Singularity Admin Guide*. Obtenido de Sylabs: https://docs.sylabs.io/guides/3.5/admin-guide/installation.html