

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Hernández Ramírez	27/08/2024
	Nombre: Josué David	

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.

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Hernández Ramírez	27/08/2024
	Nombre: Josué David	

```

(define (domain snake)
  (:requirements :strips :negative-preconditions)
  (:constants
    dummyspoint
  )
  (:predicates
    ;up down left right of a field
    (ISADJACENT ?x ?y) 3
    ;the last field of the snake
    (tailsnake ?x) 1 1 1
    ;the first field of the snake
    (headsnake ?x) 3 3 3
    ;pieces of the snake that are connected, from front to back
    (nextsnake ?x ?y) 1 3 1
    ;a field that is occupied by the snake or by an obstacle
    (blocked ?x) 3 3 1
    ;next point that will spawn
    (spawn ?x) 2 1 1
    ;point y will spawn after point x
    (NEXTSPAWN ?x ?y) 1
    ;a field that has a point that can be collected by the snake
    (ispoint ?x) 3 1 2
  )
)

You, hace 1 segundo | 1 author (You)
(:action move
  :parameters (?head ?newhead ?tail ?newtail)
  :precondition (and
    (headsnake ?head)
    (ISADJACENT ?head ?newhead)
    (tailsnake ?tail)
    (nextsnake ?newtail ?tail)
    (not (blocked ?newhead))
    (not (ispoint ?newhead))
  )
  :effect (and
    (blocked ?newhead)
    (headsnake ?newhead)
    (nextsnake ?newhead ?head)
    (not (headsnake ?head))
    (not (blocked ?tail))
    (not (tailsnake ?tail))
    (not (nextsnake ?newtail ?tail))
    (tailsnake ?newtail)
  )
)

You, hace 1 segundo | 1 author (You)
(:action move-and-eat-spawn
  :parameters (?head ?newhead ?spawnpoint ?nextspawnpoint)
  :precondition (and
    (headsnake ?head)
    (ISADJACENT ?head ?newhead)
    (not (blocked ?newhead))
    (ispoint ?newhead)
    (spawn ?spawnpoint)
    (NEXTSPAWN ?spawnpoint ?nextspawnpoint)
    (not (= ?spawnpoint dummyspoint))
  )
  :effect (and
    (blocked ?newhead)
    (headsnake ?newhead)
    (nextsnake ?newhead ?head)
    (not (headsnake ?head))
    (not (ispoint ?newhead))
    (ispoint ?spawnpoint)
    (not (spawn ?spawnpoint))
    (spawn ?nextspawnpoint)
  )
)

You, hace 1 segundo | 1 author (You)
(:action move-and-eat-no-spawn
  :parameters (?head ?newhead)
  :precondition (and
    (headsnake ?head)
    (ISADJACENT ?head ?newhead)
    (not (blocked ?newhead))
    (ispoint ?newhead)
    (spawn dummyspoint)
  )
  :effect (and
    (blocked ?newhead)
    (headsnake ?newhead)
    (nextsnake ?newhead ?head)
    (not (headsnake ?head))
    (not (ispoint ?newhead))
  )
)

```

Ilustración 1PDDL Dominio Snake

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Hernández Ramírez	27/08/2024
	Nombre: Josué David	

[illegible]

Ilustración 2 PDDL Problema Snake

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Hernández Ramírez	27/08/2024
	Nombre: Josué David	

Donde tenemos el siguiente resultado:

```
New best heuristic value for cpdbs(patterns = hillclimbing(max_time = 900), transform = multiply_out_conditional_effects): 0
[g=24, 87 evaluated, 43 expanded, t=770.014s, 3241240 KB]
Solution found!
Actual search time: 0.0809903s [t=770.014s]
move-and-eat-spawn pos4-0 pos4-1 pos2-0 pos1-4 (1)
move-and-eat-spawn pos4-1 pos3-1 pos1-4 pos1-1 (1)
move pos3-1 pos2-1 pos3-0 pos4-0 (1)
move-and-eat-spawn pos2-1 pos2-0 pos1-1 pos0-1 (1)
move pos2-0 pos1-0 pos4-0 pos4-1 (1)
move-and-eat-spawn pos1-0 pos1-1 pos0-1 pos3-3 (1)
move-and-eat-spawn pos1-1 pos0-1 pos3-3 pos4-2 (1)
move pos0-1 pos0-2 pos4-1 pos3-1 (1)
move pos0-2 pos0-3 pos3-1 pos2-1 (1)
move-and-eat-spawn pos0-3 pos0-4 pos4-2 pos3-4 (1)
move-and-eat-spawn pos0-4 pos1-4 pos3-4 pos0-0 (1)
move-and-eat-spawn pos1-4 pos2-4 pos0-0 pos1-2 (1)
move-and-eat-spawn pos2-4 pos3-4 pos1-2 pos1-0 (1)
move pos3-4 pos4-4 pos2-1 pos2-0 (1)
move pos4-4 pos4-3 pos2-0 pos1-0 (1)
move-and-eat-spawn pos4-3 pos4-2 pos1-0 dummypoint (1)
move pos4-2 pos3-2 pos1-0 pos1-1 (1)
move-and-eat-no-spawn pos3-2 pos3-3 (1)
move pos3-3 pos2-3 pos1-1 pos0-1 (1)
move-and-eat-no-spawn pos2-3 pos1-3 (1)
move-and-eat-no-spawn pos1-3 pos1-2 (1)
move pos1-2 pos1-1 pos0-1 pos0-2 (1)
move-and-eat-no-spawn pos1-1 pos1-0 (1)
move-and-eat-no-spawn pos1-0 pos0-0 (1)
Plan length: 24 step(s).
Plan cost: 24
Expanded 44 state(s).
Reopened 0 state(s).
Evaluated 87 state(s).
Evaluations: 87
Generated 86 state(s).
Dead ends: 2 state(s).
Expanded until last jump: 19 state(s).
Reopened until last jump: 0 state(s).
Evaluated until last jump: 40 state(s).
Generated until last jump: 39 state(s).
Number of registered states: 87
total successors before partial-order reduction: 86
total successors after partial-order reduction: 86
Search time: 0.884878s
Total time: 770.014s
Solution found.
Peak memory: 3241240 KB
Overall time: [774.420s CPU, 774.555s wall-clock]

==== Done running the selected planner. ====
```

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:

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Hernández Ramírez	27/08/2024
	Nombre: Josué David	

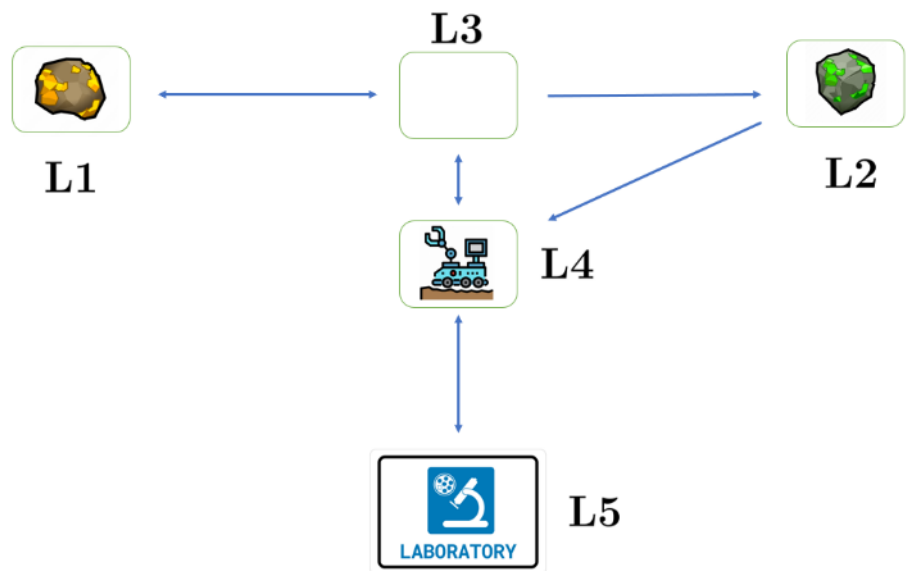


Ilustración 4 Problema a resolver

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:

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Hernández Ramírez	27/08/2024
	Nombre: Josué David	

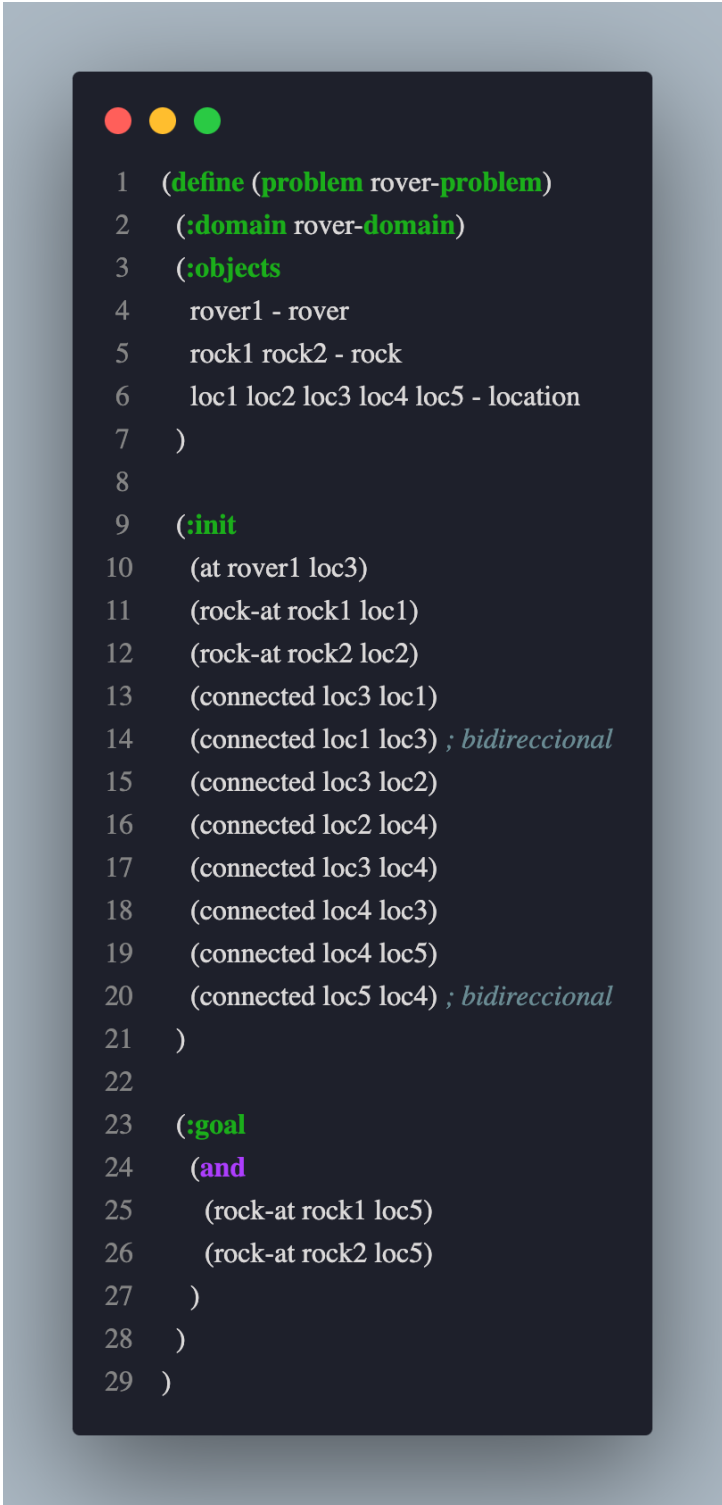
```

1  (define (domain rover-domain)
2    (:requirements :strips :typing)
3    (:types
4      location rover rock
5    )
6
7    (:predicates
8      (at ?r - rover ?l - location) ; El rover está en una ubicación
9      (rock-at ?rock - rock ?l - location) ; Una roca está en una ubicación
10     (carrying ?r - rover ?rock - rock) ; El rover está llevando una roca
11     (connected ?l1 - location ?l2 - location) ; Dos localidades están conectadas
12   )
13
14   (:action move
15     :parameters (?r - rover ?from - location ?to - location)
16     :precondition (and (at ?r ?from) (connected ?from ?to))
17     :effect (and (at ?r ?to) (not (at ?r ?from)))
18   )
19
20   (:action pick-up
21     :parameters (?r - rover ?rock - rock ?l - location)
22     :precondition (and (at ?r ?l) (rock-at ?rock ?l) (not (carrying ?r ?rock)))
23     :effect (and (carrying ?r ?rock) (not (rock-at ?rock ?l)))
24   )
25
26   (:action drop
27     :parameters (?r - rover ?rock - rock ?l - location)
28     :precondition (and (at ?r ?l) (carrying ?r ?rock))
29     :effect (and (rock-at ?rock ?l) (not (carrying ?r ?rock)))
30   )
31 )

```

Ilustración 5 PDDL Dominio Minerales

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Hernández Ramírez	27/08/2024
	Nombre: Josué David	



```

1  (define (problem rover-problem)
2    (:domain rover-domain)
3    (:objects
4      rover1 - rover
5      rock1 rock2 - rock
6      loc1 loc2 loc3 loc4 loc5 - location
7    )
8
9    (:init
10     (at rover1 loc3)
11     (rock-at rock1 loc1)
12     (rock-at rock2 loc2)
13     (connected loc3 loc1)
14     (connected loc1 loc3) ; bidireccional
15     (connected loc3 loc2)
16     (connected loc2 loc4)
17     (connected loc3 loc4)
18     (connected loc4 loc3)
19     (connected loc4 loc5)
20     (connected loc5 loc4) ; bidireccional
21   )
22
23   (:goal
24     (and
25       (rock-at rock1 loc5)
26       (rock-at rock2 loc5)
27     )
28   )
29 )

```

Ilustración 6 PDDL Problema Minerales

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Hernández Ramírez	27/08/2024
	Nombre: Josué David	

```

New best heuristic value for merge_and_shrink(shrink_strategy = shrink_bisimulation(greedy = false), merge_strategy = merge_precomputed(merge_tree = m
iasm(abstraction = miasm_merge_and_shrink, fallback_merge_selector = score_based_filtering(scoring_functions = list(goal_relevance, dfp, total_order(a
tomic_ts_order = reverse_level, product_ts_order = new_to_old, atomic_before_product = false))))), label_reduction = exact(before_shrinking = true, be
fore_merging = false), max_states = 50000, threshold_before_merge = 1, max_time = 900): 0
[g=9, 18 evaluated, 9 expanded, t=0.0040023s, 5292 KB]
Solution found!
Actual search time: 0.0005028s [t=0.0040023s]
move rover1 loc3 loc1 (1)
pick-up rover1 rock1 loc1 (1)
move rover1 loc1 loc3 (1)
move rover1 loc3 loc2 (1)
pick-up rover1 rock2 loc2 (1)
move rover1 loc2 loc4 (1)
move rover1 loc4 loc5 (1)
drop rover1 rock2 loc5 (1)
drop rover1 rock1 loc5 (1)
Plan length: 9 step(s).
Plan cost: 9
Expanded 10 state(s).
Reopened 0 state(s).
Evaluated 18 state(s).
Evaluations: 18
Generated 27 state(s).
Dead ends: 0 state(s).
Expanded until last jump: 0 state(s).
Reopened until last jump: 0 state(s).
Evaluated until last jump: 1 state(s).
Generated until last jump: 0 state(s).
Number of registered states: 18
total successors before partial-order reduction: 27
total successors after partial-order reduction: 27
Search time: 0.0005028s
Total time: 0.0040023s
Solution found.
Peak memory: 5292 KB
Overall time: [0.120s CPU, 0.137s wall-clock]
==== Done running the selected planner. ====

```

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.

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Hernández Ramírez	27/08/2024
	Nombre: Josué David	

- 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.

```

1 (define (problem complex-rover-problem)
2   (:domain rover-domain)
3
4   (:objects
5     rover1 rover2 - rover
6     rock1 rock2 rock3 - rock
7     L1 L2 L3 L4 L5 L6 - location
8   )
9
10  (:init
11    (at rover1 L4) ; rover1 starts at location L4
12    (at rover2 L3) ; rover2 starts at location L3
13    (rock-at rock1 L1) ; rock1 is at location L1
14    (rock-at rock2 L2) ; rock2 is at location L2
15    (rock-at rock3 L5) ; rock3 is at location L5
16    (connected L1 L3) ; L1 is connected to L3
17    (connected L3 L1) ; L3 is connected to L1 (bidirectional)
18    (connected L3 L2) ; L3 is connected to L2
19    (connected L2 L4) ; L2 is connected to L4
20    (connected L4 L5) ; L4 is connected to L5 (bidirectional)
21    (connected L5 L4) ; L5 is connected to L4 (bidirectional)
22    (connected L4 L6) ; L4 is connected to L6
23  )
24
25  (:goal
26    (and
27      (rock-at rock1 L6) ; rock1 should be delivered to L6
28      (rock-at rock2 L6) ; rock2 should be delivered to L6
29      (rock-at rock3 L6) ; rock3 should be delivered to L6
30    )
31  )
32 )

```

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Hernández Ramírez	27/08/2024
	Nombre: Josué David	

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:

```
New best heuristic value for merge and shrink(shrink_strategy = shrink_bisimulation(greedy = false), merge_strategy = merge_precomputed(merge_tree = #
  task(abstraction = miss_merge_and_shrink, fallback_merge_selector = score_based_filtering(scoring_functions = list(goal_relevance, dfp, total_order(a
    tomic_ts_order = reverse_level, product_ts_order = new_to_old, atonic_before_product = false))))), label_reduction = exact(before_shrinking = true, be
    fore_merging = false), max_states = 50000, threshold_before_merge = 1, max_time = 900): 0
(g=13, 46 evaluated, 13 expanded, t=0.0100018s, 5292 KB)
Solution found!
Actual search time: 0.0008748s [t=0.0100018s]
move rover2 13 11 (1)
pick-up rover2 rock1 11 (1)
move rover2 11 13 (1)
move rover2 13 12 (1)
pick-up rover2 rock2 12 (1)
move rover2 12 14 (1)
move rover2 14 15 (1)
pick-up rover2 rock3 15 (1)
move rover2 15 14 (1)
move rover2 14 16 (1)
drop rover2 rock3 16 (1)
drop rover2 rock2 16 (1)
drop rover2 rock1 16 (1)
Plan length: 13 step(s).
Plan cost: 13
Expanded 14 state(s).
Reopened 0 state(s).
Evaluated 48 state(s).
Evaluations: 48
Generated 61 state(s).
Dead ends: 2 state(s).
Expanded until last jump: 0 state(s).
Reopened until last jump: 0 state(s).
Evaluated until last jump: 1 state(s).
Generated until last jump: 0 state(s).
Number of registered states: 48
total successors before partial-order reduction: 67
total successors after partial-order reduction: 61
Search time: 0.0008748s
Total time: 0.0100018s
Solution found.
Peak memory: 5292 KB
Overall time: (0.143s CPU, 0.143s wall-clock)
```

Ilustración 8 PDDL Resultado modificado de minerales

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Hernández Ramírez	27/08/2024
	Nombre: Josué David	

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 1 PDDL Dominio Snake	2
Ilustración 2 PDDL Problema Snake.....	3
Ilustración 3 Solución al problema Snake	4
Ilustración 4 Problema a resolver	5
Ilustración 5 PDDL Dominio Minerales	6
Ilustración 6 PDDL Problema Minerales	7
Ilustración 7 PDDL Resultado Minerales	8
Ilustración 8 PDDL Resultado modificado de minerales.....	10

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>