

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Ponce Proaño	23/06/2020
	Nombre: Miguel Alejandro	

Resolución Laboratorio: Planificación de STRIPS

Contenido

DESCRIPCIÓN DE LA ACTIVIDAD	1
INFORME	1
DESCRIPCIÓN DE LOS ESCENARIOS	3
RESULTADOS PLANES	5
DIFICULTADES ENCONTRADAS	10
BIBLIOGRAFÍA	11
ANEXOS	12

DESCRIPCIÓN DE LA ACTIVIDAD

En una habitación hay un mono, una caja y un plátano, tal como indica la figura (situación inicial). El objetivo del mono es tener el plátano. El mono puede:

- Ir de una posición a otra.
- Empujar la caja de una posición a otra si está en la misma posición que ella y no está sobre ella.
- Subirse a la caja si está en la misma posición que ella.
- Coger el plátano si está encima de la caja.

Opción 2. Realizar el modelado del escenario "habitación-mono-platano" en PDDL (dominio.pddl y problema.pddl - 3 problemas) y ejecutarlo con 3 (tres) planificadores. Realizar el informe detallado y adjuntar el código fuente.

INFORME

Se desarrolló la solución utilizando el lenguaje PDDL, se crearon varios archivos con el código fuente y se describen a continuación:

El archivo domain.pddl contiene la información del dominio cuyo nombre es **dominio-mono**. En la sección requirements, utilizada para la definición del dominio, es en donde se indicará cuales características de PDDL se utilizará en este (Haslum et al., 2019), se incluye los elementos:

- **strips:** El cual indica que el dominio está en la forma simple. Corresponde a un conjunto de entradas y salidas generalmente bien definidas (Haslum et al., 2019).
- **typing:** Para el uso de objetos manejo de herencias y representaciones de clases (Haslum et al., 2019).

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Ponce Proaño	23/06/2020
	Nombre: Miguel Alejandro	

- **negative-preconditions:** Para el uso de predicados tipos falso en las acciones (Haslum et al., 2019).

En la sección types se identificarán los diferentes tipos de clases a los que pertenecen las entidades que se encuentran en el mundo. Se utilizan nombres auto descriptivos como son:

- **posicion:** Utilizada para referenciar la posición donde se encuentran los objetos en el mundo.
- **entidadMono:** Clase utilizada para representar los monos.
- **entidadCaja:** Clase utilizada para representar las cajas.
- **entidadBanana:** Clase utilizada para representar las bananas.

En la sección predicados se realizará la definición del dominio que contienen generalmente la lista de estado del modelo. Se establecerá predicados con una representación de variables binarias, que representan hechos que son verdaderos o falsos (Haslum et al., 2019).

Este caso se utilizarán los siguientes predicados para determinar la posición de las entidades en el mundo con su respectiva posición.

```
(localizacion ?ent - entidadMono ?pos - posicion)
(localizacionCaja ?ent - entidadCaja ?pos - posicion)
(localizacionBanana ?ent - entidadBanana ?pos - posicion)
```

El siguiente predicado representa la navegación entre las diferentes posiciones.

```
(inc ?x - posicion ?y - posicion)
```

Se crean predicados para determinar el estado de las entidades en el mundo al encontrar las cajas y buscar recoger las bananas.

```
(concaja ?ent - entidadMono)
(encimacaja ?ent - entidadMono ?caja - entidadCaja)
(recojebanana ?ent - entidadMono ?banana - entidadBanana)
```

Se crearon las siguientes acciones que serán ejecutadas por las entidades en el mundo.

- **mover-mono:** Acción utilizada para mover el mono de una posición a otra.
- **encontrar-mono-caja:** Acción utilizada para informar cuando el mono ha encontrado una caja en alguna posición.
- **subir-mono-caja:** Acción utilizada para que el mono suba a la caja.
- **bajar-mono-caja:** Acción utilizada para que el mono baje de la caja.
- **recoger-mono-banana:** Acción del mono al encontrar las bananas y estar encima de la caja.
- **empujar-mono-caja:** Acción utilizada para mover el mono en conjunto con la caja de una posición a otra.

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Ponce Proaño	23/06/2020
	Nombre: Miguel Alejandro	

Los archivos con el formato problem-nombre-problema tienen la información de los diferentes problemas que harán uso del dominio(dominio-mono). En la sección objects, se declaran los objetos que interactuarán en el mundo, en donde los nombres son auto descriptivos y haremos uso de posiciones en donde se encontrarán las diferentes entidades que son representadas por monos, banas y cajas.

En la sección init, se inicializan los predicados y condiciones iniciales del mundo, los cuales serán análogos para los diferentes problemas que serán planteados. Por ejemplo:

```
(inc p1 p2)
(inc p2 p3)
(inc p3 p2)
(inc p2 p1)
(localizacion m1 p1)
(localizacionCaja c1 p3)
(localizacionBanana b1 p1)
```

En la sección goal, se inicializan los objetivos meta que se desea alcanzar por las entidades en el mundo, los cuales serán análogos para los diferentes problemas que serán planteados. Por ejemplo:

```
(recojebanana m1 b1)
```

DESCRIPCIÓN DE LOS ESCENARIOS

A continuación, se realiza una descripción gráfica de los diferentes escenarios(problemas), que en adelante serán identificados como escenario1, escenario2 y escenario3.

- ✓ Para el problema problem-base.pddl(escenario1), en este escenario el mono de la posición 3 agarrará las bananas en la posición 2.



Figura situación inicial(escenario1)

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Ponce Proaño	23/06/2020
	Nombre: Miguel Alejandro	

- ✓ Para el problema `problem-multiple-mono-banana.pddl` (escenario2), en este escenario el mono de la posición 1 agarrará las bananas de las posiciones 1 y 4. Por otro lado, el mono de la posición 2 agarrará las bananas de la posición 3.



Figura problema múltiple escenario2

- ✓ Para el problema `problem-multiple-mono-banana-caja.pddl`(escenario3), en el escenario uno de los monos de la posición 1 agarrará las bananas de la posición 4; adicionalmente, otro mono en la misma posición agarrará las bananas de las posiciones 2 y 4. Finalmente, el mono de la posición 6 agarrará las bananas de la posición 1.

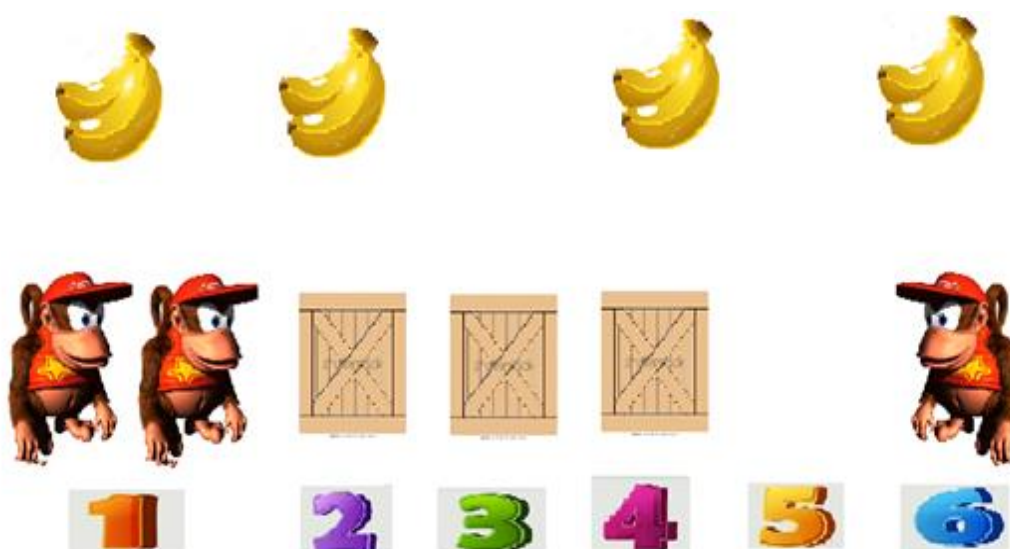


Figura problema múltiple escenario3

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Ponce Proaño	23/06/2020
	Nombre: Miguel Alejandro	

RESULTADOS PLANES

1. **Fast-Forward(FF).** - es un sistema de planificación independiente del dominio desarrollado por Joerg. FF puede manejar tareas de planificación ADL clásicas, así como ADL a escala completa(Joerg, s. f.-a).

Fast-Forward(FF) con JAVA

```

C:\WINDOWS\system32\cmd.exe
Grounding...
Grounding complete
Performing RPG reachability analysis...
Found 9 reachable facts from 9 original facts.
Found 18 applicable actions from 18 original actions
Goal is: (and (recojebanana m1 b1) )
Running FF with EHC...
Performing search using EHC with standard helpful action filter
6 into depth [1]
5 into depth [1][2]
3 into depth [1]
2 into depth [1]
1 into depth [1]
Evaluated 7 states to a max depth of 2
Found EHC plan:
Final plan...
0: (mover-mono m1 p3 p2)
1: (mover-mono m1 p2 p1)
2: (encontrar-mono-caja m1 c1 p1)
3: (empujar-mono-caja m1 c1 p1 p2)
4: (subir-mono-caja m1 c1 p2)
5: (recoger-mono-banana m1 c1 b1 p2)
Final plan length is 6
EHC Plan Time = 0.024644855sec
BFS Plan Time = 0.0sec
Scheduling Time = 0.0sec
Plan written to C:\Users\PC-MIKE\UNIR\UNIR_actividades\RazonamientoPlanificacion\laboratorio\resultado-problem-base.soln
C:\Users\PC-MIKE\UNIR\IA_MachineLearning_Python\RazonamientoPlanificacion\ff-maven>

```

Figura resultados FF escenario1

```

C:\WINDOWS\system32\cmd.exe
Running FF with BFS...
Performing search using BFS
Found BFS plan:
Final plan...
0: (mover-mono m1 p1 p2)
1: (mover-mono m2 p2 p3)
2: (mover-mono m1 p2 p3)
3: (encontrar-mono-caja m1 c1 p3)
4: (subir-mono-caja m1 c1 p3)
5: (encontrar-mono-caja m2 c1 p3)
6: (subir-mono-caja m2 c1 p3)
7: (recoger-mono-banana m2 c1 b3 p3)
8: (bajar-mono-caja m1 c1 p3)
9: (empujar-mono-caja m1 c1 p3 p2)
10: (subir-mono-caja m1 c1 p2)
11: (empujar-mono-caja m1 c1 p2 p1)
12: (recoger-mono-banana m1 c1 b1 p1)
13: (empujar-mono-caja m1 c1 p1 p2)
14: (empujar-mono-caja m1 c1 p2 p3)
15: (empujar-mono-caja m1 c1 p3 p4)
16: (recoger-mono-banana m1 c1 b4 p4)
Final plan length is 17
EHC Plan Time = 0.045485725sec
BFS Plan Time = 0.332459636sec
Scheduling Time = 0.0sec
Plan written to C:\Users\PC-MIKE\UNIR\UNIR_actividades\RazonamientoPlanificacion\laboratorio\resultado-problem-multiple-mono-banana.soln
C:\Users\PC-MIKE\UNIR\IA_MachineLearning_Python\RazonamientoPlanificacion\ff-maven>

```

Figura resultados FF escenario2

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Ponce Proaño	23/06/2020
	Nombre: Miguel Alejandro	

```

C:\WINDOWS\system32\cmd.exe
0: (mover-mono m3 p6 p5)
1: (mover-mono m3 p5 p4)
2: (encontrar-mono-caja m3 c2 p4)
3: (mover-mono m2 p1 p2)
4: (encontrar-mono-caja m2 c3 p2)
5: (subir-mono-caja m2 c3 p2)
6: (mover-mono m1 p1 p2)
7: (encontrar-mono-caja m1 c3 p2)
8: (recoger-mono-banana m2 c3 b2 p2)
9: (subir-mono-caja m1 c3 p2)
10: (empujar-mono-caja m3 c2 p4 p3)
11: (empujar-mono-caja m3 c2 p3 p2)
12: (empujar-mono-caja m3 c2 p2 p1)
13: (subir-mono-caja m3 c2 p1)
14: (recoger-mono-banana m3 c2 b1 p1)
15: (bajar-mono-caja m2 c3 p2)
16: (empujar-mono-caja m2 c3 p2 p3)
17: (empujar-mono-caja m2 c3 p3 p4)
18: (subir-mono-caja m2 c3 p4)
19: (recoger-mono-banana m2 c3 b3 p4)
20: (empujar-mono-caja m2 c3 p4 p5)
21: (empujar-mono-caja m2 c3 p5 p6)
22: (bajar-mono-caja m3 c2 p1)
23: (empujar-mono-caja m3 c2 p1 p2)
24: (empujar-mono-caja m1 c2 p2 p3)
25: (empujar-mono-caja m1 c1 p3 p4)
26: (empujar-mono-caja m1 c1 p4 p5)
27: (empujar-mono-caja m1 c1 p5 p6)
28: (recoger-mono-banana m1 c3 b4 p6)

```

Figura resultados FF escenario3

Fast-Forward(FF) con LINUX

```

General  Transferencia de Archivos por SSH
Ubuntu
problem-base.pddl

ff: parsing domain file
domain 'DOMINIO-MONO' defined
... done.
ff: parsing problem file
problem 'PROBLEMA-MONO-CAJA-BANANA' defined
... done.

Cueing down from goal distance:  6 into depth [1]
                                5          [1][2]
                                3          [1]
                                2          [1]
                                1          [1]
                                0

ff: found legal plan as follows

step  0: MOVER-MONO M1 P3 P2
      1: MOVER-MONO M1 P2 P1
      2: ENCONTRAR-MONO-CAJA M1 C1 P1
      3: EMPUJAR-MONO-CAJA M1 C1 P1 P2
      4: SUBIR-MONO-CAJA M1 C1 P2
      5: RECOGER-MONO-BANANA M1 C1 B1 P2

time spent:  0.00 seconds instantiating 18 easy, 0 hard action templates
            0.00 seconds reachability analysis, yielding 11 facts and 18 actions
            0.00 seconds creating final representation with 11 relevant facts
            0.00 seconds building connectivity graph
            0.00 seconds searching, evaluating 8 states, to a max depth of 2
            0.00 seconds total time

osboxes@osboxes:~/Downloads/FF-v2.3$

```

Figura resultados FF Linux escenario1

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Ponce Proaño	23/06/2020
	Nombre: Miguel Alejandro	

```

General  Transferencia de Archivos por SSH
Ubuntu

Queing down from goal distance:  14 into depth [1][2]
                                12      [1][2]
                                11      [1]
                                10     [1][2][3]
                                8       [1]
                                7      [1][2][3]
                                6      [1][2]
                                5       [1]
                                3       [1]
                                2       [1]
                                1       [1]
                                0

ff: found legal plan as follows

step  0: MOVER-MONO M2 P2 P3
      1: ENCONTRAR-MONO-CAJA M2 C1 P3
      2: SUBIR-MONO-CAJA M2 C1 P3
      3: RECOGER-MONO-BANANA M2 C1 B3 P3
      4: BAJAR-MONO-CAJA M2 C1 P3
      5: MOVER-MONO M1 P1 P2
      6: MOVER-MONO M1 P2 P3
      7: ENCONTRAR-MONO-CAJA M1 C1 P3
      8: EMPUJAR-MONO-CAJA M1 C1 P3 P2
      9: EMPUJAR-MONO-CAJA M1 C1 P2 P1
     10: SUBIR-MONO-CAJA M1 C1 P1
     11: RECOGER-MONO-BANANA M1 C1 B1 P1
     12: BAJAR-MONO-CAJA M1 C1 P1
     13: EMPUJAR-MONO-CAJA M1 C1 P1 P2
     14: EMPUJAR-MONO-CAJA M1 C1 P2 P3
     15: EMPUJAR-MONO-CAJA M1 C1 P3 P4
     16: SUBIR-MONO-CAJA M1 C1 P4
     17: RECOGER-MONO-BANANA M1 C1 B4 P4

```

Figura resultados FF Linux escenario2

```

General  Transferencia de Archivos por SSH
Ubuntu

ff: found legal plan as follows

step  0: MOVER-MONO M1 P1 P2
      1: ENCONTRAR-MONO-CAJA M1 C3 P2
      2: MOVER-MONO M2 P1 P2
      3: ENCONTRAR-MONO-CAJA M2 C3 P2
      4: SUBIR-MONO-CAJA M2 C3 P2
      5: RECOGER-MONO-BANANA M2 C3 B2 P2
      6: MOVER-MONO M3 P6 P5
      7: EMPUJAR-MONO-CAJA M1 C3 P2 P3
      8: EMPUJAR-MONO-CAJA M1 C3 P3 P4
      9: MOVER-MONO M3 P5 P4
     10: MOVER-MONO M3 P4 P3
     11: ENCONTRAR-MONO-CAJA M3 C1 P3
     12: EMPUJAR-MONO-CAJA M1 C2 P4 P5
     13: EMPUJAR-MONO-CAJA M3 C1 P3 P2
     14: EMPUJAR-MONO-CAJA M2 C1 P2 P3
     15: EMPUJAR-MONO-CAJA M1 C2 P5 P6
     16: SUBIR-MONO-CAJA M1 C2 P6
     17: RECOGER-MONO-BANANA M1 C2 B4 P6
     18: EMPUJAR-MONO-CAJA M2 C1 P3 P4
     19: RECOGER-MONO-BANANA M2 C3 B3 P4
     20: EMPUJAR-MONO-CAJA M2 C1 P4 P3
     21: EMPUJAR-MONO-CAJA M2 C1 P3 P2
     22: EMPUJAR-MONO-CAJA M3 C1 P2 P1
     23: SUBIR-MONO-CAJA M3 C1 P1
     24: RECOGER-MONO-BANANA M3 C1 B1 P1

time spent:  0.00 seconds instantiating 318 easy, 0 hard action templates
             0.00 seconds reachability analysis, yielding 72 facts and 318 actions
             0.00 seconds creating final representation with 72 relevant facts
             0.00 seconds building connectivity graph
             0.00 seconds searching, evaluating 96 states, to a max depth of 3
             0.00 seconds total time

```

Figura resultados FF Linux escenario3

2. **LPG-td.-** es una nueva versión de LPG que mejora la versión 1.2, y que participó en la 4ta Competencia Internacional de Planificación, 2004(LPG Team, s. f.).

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Ponce Proaño	23/06/2020
	Nombre: Miguel Alejandro	

LPG-td es una extensión de LPG para manejar las nuevas características de la planificación estándar lenguaje de descripción de dominio PDDL(Gerevini & Serina, 2002).

LPG-td con LINUX

```

General  Transferencia de Archivos por SSH
Ubuntu

Numeric Planning Problem: NO
Problem with Timed Initial Literals: NO
Problem with Derived Predicates: NO

Evaluation function weights:
  Action duration 0.00; Action cost 1.00

Computing mutex... done

Preprocessing total time: 0.01 seconds

Searching ('.' = every 50 search steps):
solution found:
first_solution_cpu_time: 0.02

Plan computed:
  Time: (ACTION) [action Duration; action Cost]
0.0000: (MOVER-MONO M1 P3 P2) [D:1.00; C:1.00]
1.0000: (MOVER-MONO M1 P2 P1) [D:1.00; C:1.00]
2.0000: (ENCONTRAR-MONO-CAJA M1 C1 P1) [D:1.00; C:1.00]
3.0000: (EMPUJAR-MONO-CAJA M1 C1 P1 P2) [D:1.00; C:1.00]
4.0000: (SUBIR-MONO-CAJA M1 C1 P2) [D:1.00; C:1.00]
5.0000: (RECOGER-MONO-BANANA M1 C1 B1 P2) [D:1.00; C:1.00]

Solution number: 1
Total time:      0.02
Search time:     0.01
Actions:         6
Duration:        6.000
Plan quality:    6.000
Total Num Flips: 6
Plan file:       plan_/home/osboxes/Downloads/laboratorio_1.SOL

osboxes@osboxes:~/Downloads/LPG-td-1.4$

```

Figura resultados LPG Linux escenario1

```

General  Transferencia de Archivos por SSH
Ubuntu

solution found:
first_solution_cpu_time: 0.01

Plan computed:
  Time: (ACTION) [action Duration; action Cost]
0.0000: (MOVER-MONO M2 P2 P3) [D:1.00; C:1.00]
0.0000: (MOVER-MONO M1 P1 P2) [D:1.00; C:1.00]
1.0000: (ENCONTRAR-MONO-CAJA M2 C1 P3) [D:1.00; C:1.00]
2.0000: (EMPUJAR-MONO-CAJA M2 C1 P3 P2) [D:1.00; C:1.00]
3.0000: (ENCONTRAR-MONO-CAJA M1 C1 P2) [D:1.00; C:1.00]
4.0000: (EMPUJAR-MONO-CAJA M2 C1 P2 P3) [D:1.00; C:1.00]
5.0000: (SUBIR-MONO-CAJA M2 C1 P3) [D:1.00; C:1.00]
6.0000: (RECOGER-MONO-BANANA M2 C1 B3 P3) [D:1.00; C:1.00]
7.0000: (BAJAR-MONO-CAJA M2 C1 P3) [D:1.00; C:1.00]
8.0000: (EMPUJAR-MONO-CAJA M2 C1 P3 P2) [D:1.00; C:1.00]
9.0000: (EMPUJAR-MONO-CAJA M1 C1 P2 P1) [D:1.00; C:1.00]
10.0000: (SUBIR-MONO-CAJA M1 C1 P1) [D:1.00; C:1.00]
11.0000: (RECOGER-MONO-BANANA M1 C1 B1 P1) [D:1.00; C:1.00]
12.0000: (BAJAR-MONO-CAJA M1 C1 P1) [D:1.00; C:1.00]
13.0000: (EMPUJAR-MONO-CAJA M1 C1 P1 P2) [D:1.00; C:1.00]
14.0000: (EMPUJAR-MONO-CAJA M1 C1 P2 P3) [D:1.00; C:1.00]
15.0000: (EMPUJAR-MONO-CAJA M1 C1 P3 P4) [D:1.00; C:1.00]
16.0000: (SUBIR-MONO-CAJA M1 C1 P4) [D:1.00; C:1.00]
17.0000: (RECOGER-MONO-BANANA M1 C1 B4 P4) [D:1.00; C:1.00]

Solution number: 1
Total time:      0.01
Search time:     0.01
Actions:         19
Duration:        18.000
Plan quality:    19.000
Total Num Flips: 26
Plan file:       plan_/home/osboxes/Downloads/laboratorio_1.SOL

osboxes@osboxes:~/Downloads/LPG-td-1.4$

```

Figura resultados LPG Linux escenario2

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Ponce Proaño	23/06/2020
	Nombre: Miguel Alejandro	

```

General  Transfencia de Archivos por SSH
Ubuntu

Searching ('.' = every 50 search steps):
solution found:
first_solution_cpu_time: 0.03

Plan computed:
Time: (ACTION) [action Duration; action Cost]
0.0000: (MOVER-MONO M2 P1 P2) [D:1.00; C:1.00]
0.0000: (MOVER-MONO M1 P1 P2) [D:1.00; C:1.00]
0.0000: (MOVER-MONO M3 P6 P5) [D:1.00; C:1.00]
1.0000: (MOVER-MONO M2 P2 P3) [D:1.00; C:1.00]
1.0000: (MOVER-MONO M1 P2 P3) [D:1.00; C:1.00]
1.0000: (MOVER-MONO M3 P5 P4) [D:1.00; C:1.00]
2.0000: (MOVER-MONO M2 P3 P2) [D:1.00; C:1.00]
2.0000: (ENCONTRAR-MONO-CAJA M1 C1 P3) [D:1.00; C:1.00]
2.0000: (MOVER-MONO M3 P4 P3) [D:1.00; C:1.00]
3.0000: (ENCONTRAR-MONO-CAJA M2 C3 P2) [D:1.00; C:1.00]
3.0000: (EMPUJAR-MONO-CAJA M1 C1 P3 P4) [D:1.00; C:1.00]
4.0000: (SUBIR-MONO-CAJA M2 C3 P2) [D:1.00; C:1.00]
4.0000: (EMPUJAR-MONO-CAJA M1 C1 P4 P5) [D:1.00; C:1.00]
5.0000: (RECOGER-MONO-BANANA M2 C3 B2 P2) [D:1.00; C:1.00]
5.0000: (EMPUJAR-MONO-CAJA M1 C1 P5 P6) [D:1.00; C:1.00]
6.0000: (BAJAR-MONO-CAJA M2 C3 P2) [D:1.00; C:1.00]
6.0000: (SUBIR-MONO-CAJA M1 C1 P6) [D:1.00; C:1.00]
7.0000: (EMPUJAR-MONO-CAJA M2 C3 P2 P3) [D:1.00; C:1.00]
7.0000: (RECOGER-MONO-BANANA M1 C1 B4 P6) [D:1.00; C:1.00]
8.0000: (EMPUJAR-MONO-CAJA M2 C3 P3 P4) [D:1.00; C:1.00]
9.0000: (SUBIR-MONO-CAJA M2 C3 P4) [D:1.00; C:1.00]
10.0000: (RECOGER-MONO-BANANA M2 C3 B3 P4) [D:1.00; C:1.00]
11.0000: (EMPUJAR-MONO-CAJA M2 C2 P4 P3) [D:1.00; C:1.00]
12.0000: (ENCONTRAR-MONO-CAJA M3 C2 P3) [D:1.00; C:1.00]
13.0000: (EMPUJAR-MONO-CAJA M3 C2 P3 P2) [D:1.00; C:1.00]
14.0000: (EMPUJAR-MONO-CAJA M3 C2 P2 P1) [D:1.00; C:1.00]
15.0000: (SUBIR-MONO-CAJA M3 C2 P1) [D:1.00; C:1.00]
16.0000: (RECOGER-MONO-BANANA M3 C2 B1 P1) [D:1.00; C:1.00]

```

Figura resultados LPG Linux escenario3

3. **Planning domains(PD).** - Es un componente planning.domains que consiste en un planificador y validador automatizado en la nube. Puede invocar el software enviando enlaces a los archivos PDDL o enviando contenido PDDL en formato JSON directamente para recuperar o validar un plan (Andrew Coles, s. f.-b). Se utilizará la conexión a la URI <http://solver.planning.domains/solve>.

VSCoDe PD con Windows

```

File Edit Selection View Go Run Terminal Help
Planner output - laboratorio - Visual Studio Code

Match tree built with 18 nodes.

PDDL problem description loaded:
Domain: DOMINIO-MONO
Problem: PROBLEMA-MONO-CAJA-BANANA
#Actions: 18
#fluents: 11

Landmarks found: 1
Starting search with IW (time budget is 60 secs)...
rel_plan size: 6
#RP_fluents 6
Caption
{#goals, #UNnachieved, #Achieved} -> IW(max_w)

{1/1/0}:IW(1) -> [2][3][4][5][6][7]rel_plan size: 0
#RP_fluents 0Plan found with cost: 6
Total time: 8.9407e-10
Nodes generated during search: 14
Nodes expanded during search: 11
IW search completed
0.00100: (mover-mono m1 p3 p2)
0.00200: (mover-mono m1 p2 p1)
0.00300: (encontrar-mono-caja m1 c1 p1)
0.00400: (empujar-mono-caja m1 c1 p1 p2)
0.00500: (subir-mono-caja m1 c1 p2)
0.00600: (recoger-mono-banana m1 c1 b1 p2)
Planner found 1 plan(s) in 1.559secs.

```

Figura resultados PD escenario1

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Ponce Proaño	23/06/2020
	Nombre: Miguel Alejandro	

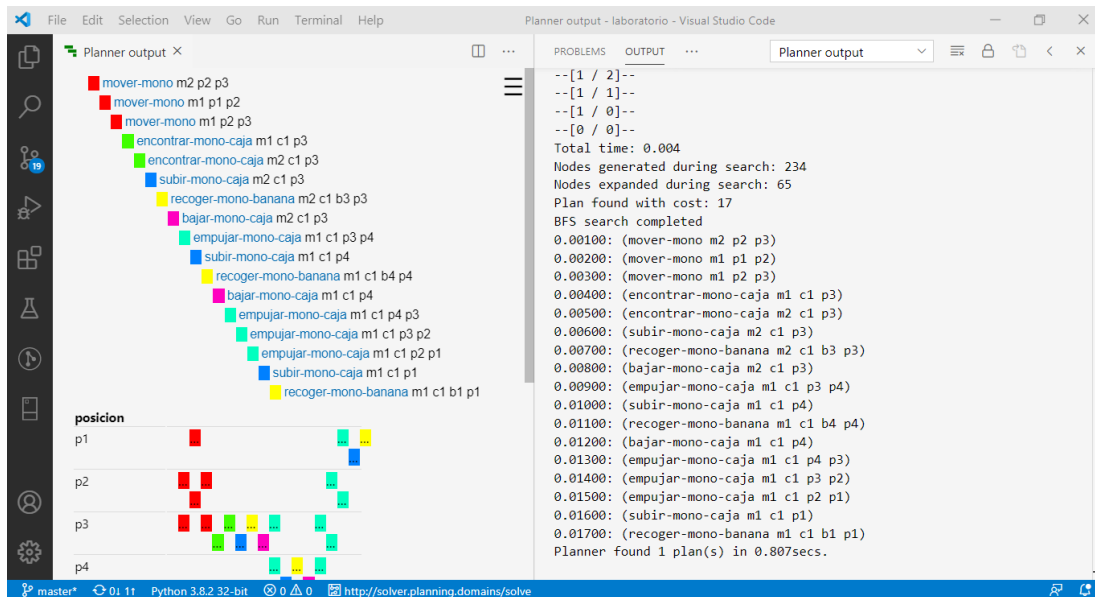


Figura resultados PD escenario2

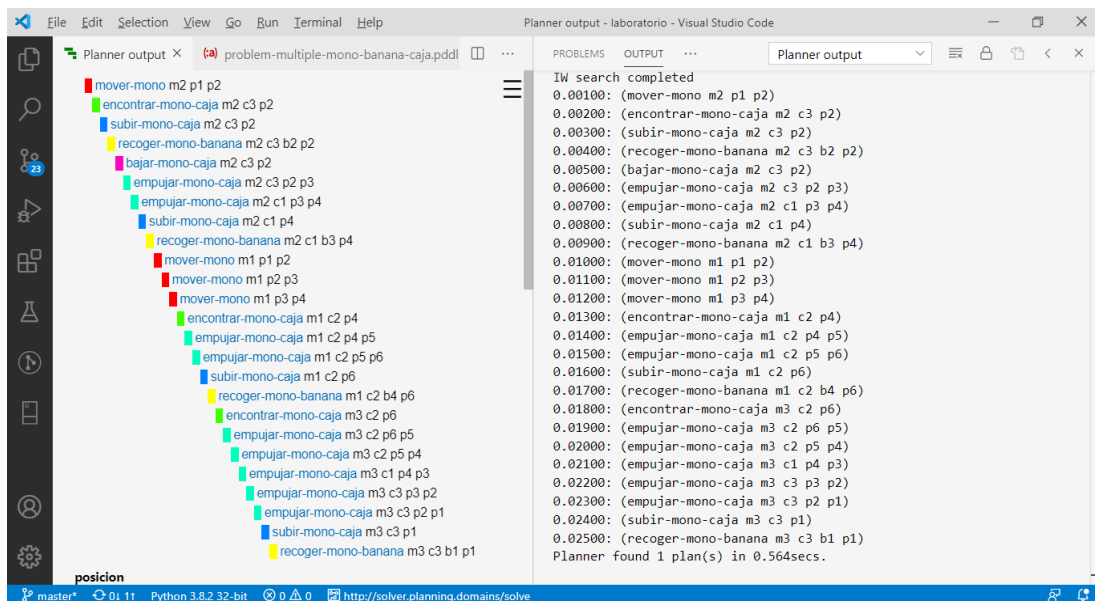


Figura resultados PD escenario3

DIFICULTADES ENCONTRADAS

- Respecto de la codificación en el lenguaje pddl, es un muy diferente de los lenguajes de programación tradicionales. Para el desarrollo del ejercicio se tomó la sugerencia en clase relacionada con el ir creando pequeños micro objetivos e irlos testeando. Por ejemplo, primero probar la acción moverse, luego probar la

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Ponce Proaño	23/06/2020
	Nombre: Miguel Alejandro	

acción encontrar caja, luego probar la acción subir caja y así sucesivamente con la consecución de las diferentes micro-metas se logró realizar un mejor entendimiento del lenguaje. Finalmente se hizo un refuerzo con un libro para entender de forma práctica cómo funciona el lenguaje.

- Respecto del uso de Linux para la compilación de los planificadores fue necesario realizar varias modificaciones en los archivos de rutas de recursos del sistema operativo. Luego de descargar e instalar una máquina de virtual, distribuidas por <https://www.osboxes.org/> para Oracle VM VirtualBox(Umair, s. f.), fue necesario la instalación de varias librerías necesarias para la compilación de los planificadores. Finalmente, se procedió a seguir los pasos para la compilación vistos en clase.
- Respecto del planificador FF en JAVA, el cual hace referencia al trabajo de David Pattison(Pattison, 2015),al navegar hacia los links, al parecer los recursos no existen, entonces se optó por utilizar el código de la siguiente página <https://github.com/dpattiso/javaff> y generar un nuevo jar(dpattiso, 2017/2020). Luego, se creó un proyecto en formato MAVEN, el cual genera un JAR con sus dependencias.

BIBLIOGRAFÍA

- Andrew Coles. (s. f.-b). *Solver.planning.domains*. Recuperado 23 de junio de 2020, de <http://solver.planning.domains/>
- Gerevini, A., & Serina, I. (2002). LPG: A Planner Based on Local Search for Planning Graphs with Action Costs. *AIPS*, 2, 281–290.
- Haslum, P., Lipovetzky, N., Magazzeni, D., & Muise, C. (2019). *An introduction to the planning domain definition language* (Vol. 13). Morgan & Claypool Publishers.
- Joerg. (s. f.-a). *FF Homepage*. Recuperado 23 de junio de 2020, de <https://fai.cs.uni-saarland.de/hoffmann/ff.html>
- Pattison, D. T. (2015). *A new heuristic-based model of goal recognition without libraries* [PhD Thesis]. University of Strathclyde.

Asignatura	Datos del alumno	Fecha
Razonamiento y Planificación Automática	Apellidos: Ponce Proaño	23/06/2020
	Nombre: Miguel Alejandro	

Umair. (s. f.). *OSBoxes—Virtual Machines for VirtualBox & VMware*. OSBoxes - Virtual Machines. Recuperado 23 de junio de 2020, de <https://www.osboxes.org/>

ANEXOS

Se incluye como anexos los siguientes archivos:

Dominio:

- domain.pddl

Problemas:

- problem-base.pddl
- problem-multiple-mono-banana.pddl
- problem-multiple-mono-banana-caja.pddl

Los resultados(resultadosplanificadores.rar) se encuentran en las siguientes carpetas:

Nombre

- FF-JAVA
- FF-LINUX
- LPG-LINUX
- planning.domains-WINDOWS

Resultados:

- resultado-problem-base.soln
- resultado-problem-multiple-mono-banana.soln
- resultado-problem-multiple-mono-banana-caja.soln