

# Razonamiento y Planificación Automática

Adriana Cervantes Castillo

## Clase 6. Sesión explicativa: Laboratorio: Planificación en STRIPS/PDDL

# Contenido

- Objetivos de la actividad
- Descripción de la actividad
- Archivos dominio.pddl, problema.pddl
- Documentos a entregar
- Rubrica de evaluación
- Notas finales

# Objetivo de la actividad 3

- Incentivar el proceso investigativo y utilizar los planificadores del estado de arte para solucionar un problema en STRIPS/PDDL.
- Desarrollar una solución informática a un problema concreto.
- Analizar los resultados obtenidos y ser capaz de expresar las conclusiones en formato de comunicación científica.

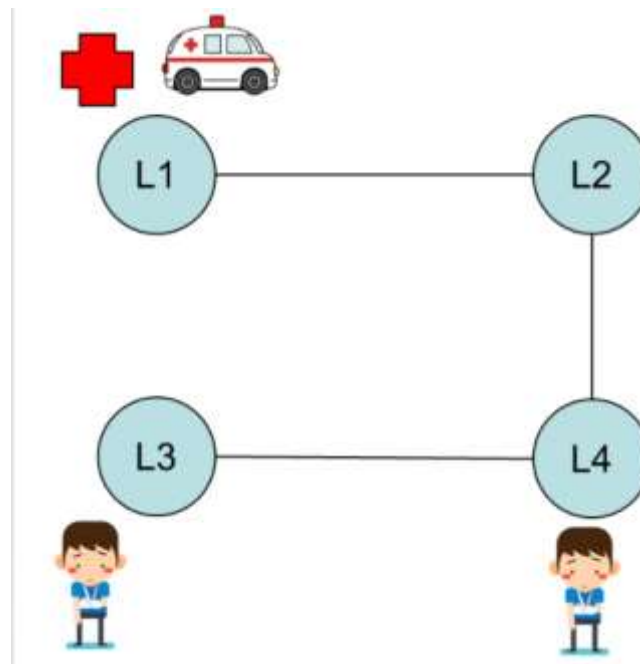
Mostrar con claridad las fases de:

- Diseño
- Desarrollo
- Implementación
- Pruebas
  
- *Análisis de resultados obtenidos*

Mostrar de manera completa y concisa aportando las referencias relevantes y sin aportar información no pertinente.

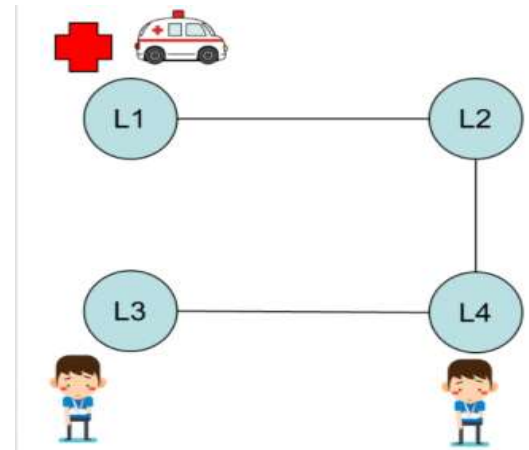
# Descripción de la actividad 3

- Soluciona en STRIPS/PDDL el siguiente problema.
- En una ciudad existen cuatro localizaciones, un hospital, una ambulancia, y dos enfermos. Tal como indica la siguiente figura (situación inicial):

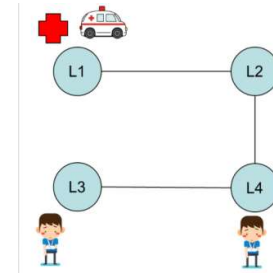


# Descripción de la actividad 3

- El objetivo es trasladar los pacientes al hospital. Las acciones que se pueden realizar son:
- Conducir la ambulancia de una posición a otra.
- Subir un paciente a la ambulancia.
- Bajar a un paciente en una localización.



# Descripción de la actividad 3



- Se deberá modelar en PDDL (dominio.pddl y problema.pddl) el escenario que se plantea en la situación inicial y **resolver el problema con cuatro planificadores del estado del arte.**
- El estudiante debe buscar y configurar por su cuenta los planificadores.

# Restricciones


- Los pacientes son recogidos de sus ubicaciones
- Los pacientes son dejados en el hospital
- Sólo se puede llevar a un paciente cada vez
- Sólo se puede transitar de una ubicación a otra cuando hay una carretera



- Las soluciones ofrecidas por los planificadores se deben **comparar**, indicando cuáles han dado soluciones óptimas y en un menor tiempo de ejecución.
- **Adicionalmente, se deben modelar dos situaciones** (dos problemas) **diferentes a la planteada en esta actividad** (por ejemplo, cambiar las conexiones entre las localizaciones, agregar o eliminar pacientes, colocar otra ambulancia, etc.). Y se deben **resolver los dos problemas con el mejor planificador de los cuatro seleccionados**.

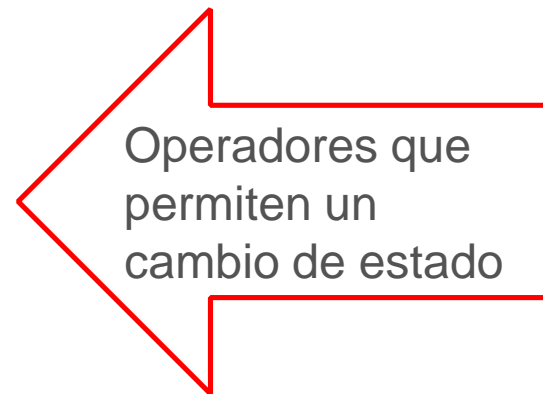
# Archivos domain.pddl y problem.pddl

```
wget https://bitbucket.org/ipc2018-classical/demo-submission/raw/ipc-2018-seq-opt/Singularity
sudo singularity build planner.img Singularity
mkdir rundir
cp path/to/domain.pddl rundir
cp path/to/problem.pddl rundir
RUNDIR="$(pwd)/rundir"
DOMAIN="$RUNDIR/domain.pddl"
PROBLEM="$RUNDIR/problem.pddl"
PLANFILE="$RUNDIR/sas_plan"
COSTBOUND=42 # only in cost-bounded track
ulimit -t 1800
ulimit -v 8388608
singularity run -C -H $RUNDIR planner.img $DOMAIN $PROBLEM $PLANFILE $COSTBOUND
```



# Estructura del archivo dominio.pddl

```
(define (domain nombreProblema)  
  (:requirements : )  
  (:types ) //objetos que participan en el problema  
  (:predicates // situaciones/estados en que pueden estar los  
    objetos  
    (...)  
    ( ...)  
  )  
  (:action nombreAccion  
    :parameters ()  
    :precondition ()  
    :effect ()  
  )  
)
```



# Estructura del archivo problema.pddl

(**define** (**problem** nombreProblema)

(:**domain** nombreDominio)

(:**objects** objetos que participan)

Objetos del problema

(:**init** (estados iniciales del problema)

Estado inicial

)

(:**goal** (and ( estados objetivo)

)

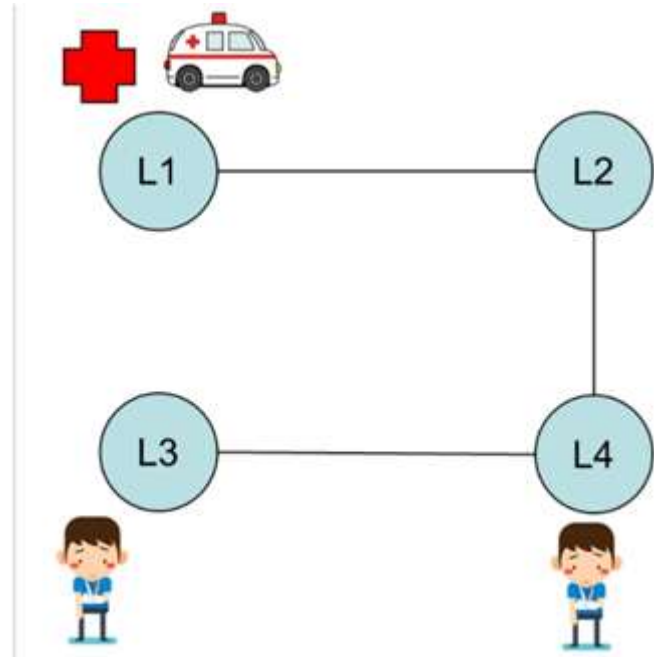
)

)

Estado objetivo

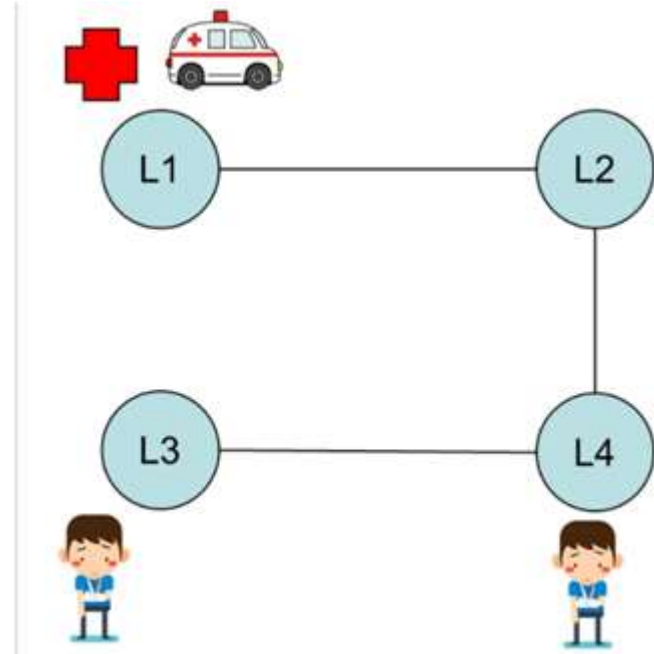
# Dominio.pddl

- Identificar los predicados
  - ¿?
- Identificar los operadores
  - ¿?



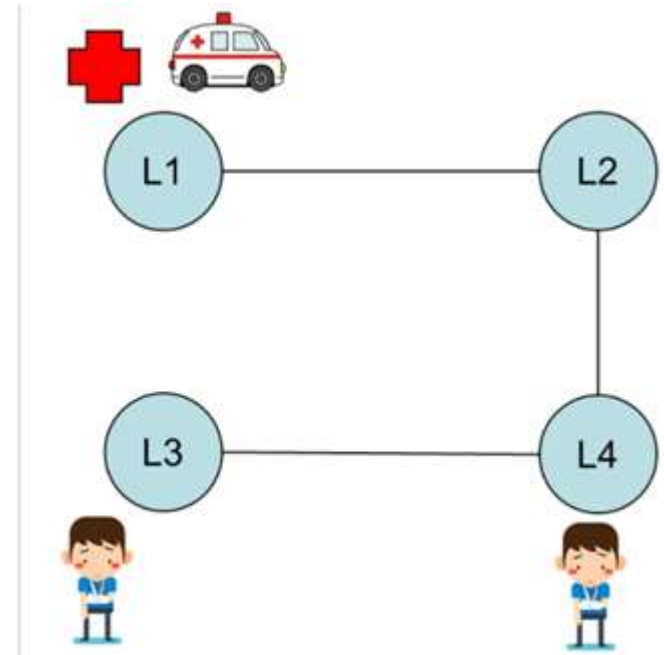
# Dominio.pddl

- Identificar los predicados
  - Vacía-ambulancia
  - Llena-ambulancia
  - Ambulancia I
  - Hospital I
  - Paciente I
  - Ir I1, I2
- Identificar los operadores
  - ¿?



# Dominio.pddl

- Identificar los predicados
  - Vacía-ambulancia
  - Llena-ambulancia
  - Ambulancia l
  - Hospital l
  - Paciente l
  - Ir l1, l2
- Identificar los operadores
  - Mover(ambulancia,x,y)
  - Subir(p,ambulancia)
  - Bajar(p,l)



# Estructura del archivo dominio.pddl

```
(define (domain pacientes)
  (:requirements :typing)
  (:types paciente ambulancia localizacion hospital)
  (:predicates
    (paciente-en-1 ?l - localizacion ?p - paciente )
    (ambulancia-en-1 ?l - localizacion ?a - ambulancia)
    (hospital-en ?l - localizacion)
    (llena-ambulancia ?p - paciente)
    (vacía-ambulancia)
    (ir ?l1 -localizacion ?l2 - localizacion)
  )
```

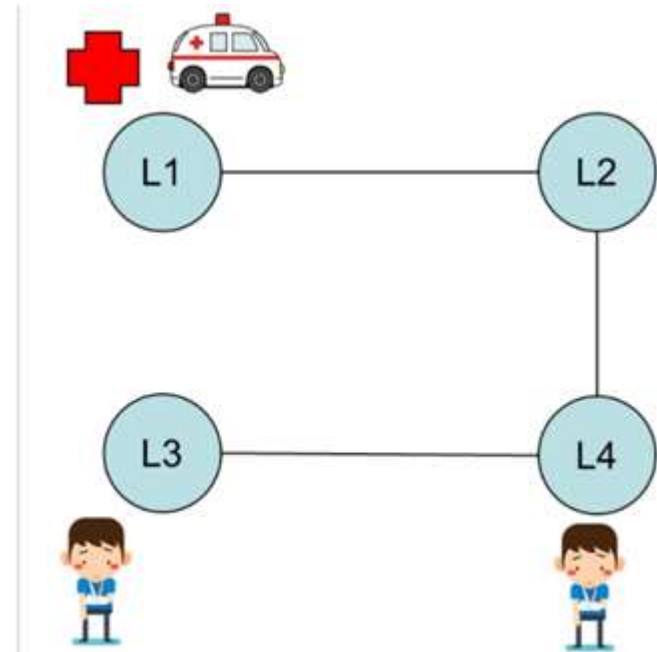


# Acciones

```
(:action mover
  :parameters (?a - ambulancia ?l1 - localizacion ?l2 - localizacion)
  :precondition
    ( and( ambulancia-en-1 ?l1 ?a) (ir ?l1 ?l2 ))
  :effect
    ( and( not( ambulancia-en-1 ?l1 ?a)) (ambulancia-en-1 ?l2 ?a) (ir ?l2 ?l1))
)
;////////////////////////////////////
(:action subir
  :parameters (?p - paciente ?l1 - localizacion ?a - ambulancia)
  :precondition
    (and (vacía-ambulancia) (ambulancia-en-1 ?l1 ?a) (paciente-en-1 ?l1 ?p) )
  :effect
    (and (not(vacía-ambulancia)) (llena-ambulancia ?p))
)
;////////////////////////////////////
(:action bajar
  :parameters (?p - paciente ?l1 - localizacion ?a - ambulancia)
  :precondition
    (and (llena-ambulancia ?p) (ambulancia-en-1 ?l1 ?a))
  :effect
    (and (not(llena-ambulancia ?p)) (vacía-ambulancia)
      (ambulancia-en-1 ?l1 ?a) (paciente-en-1 ?l1 ?p) )
)
```

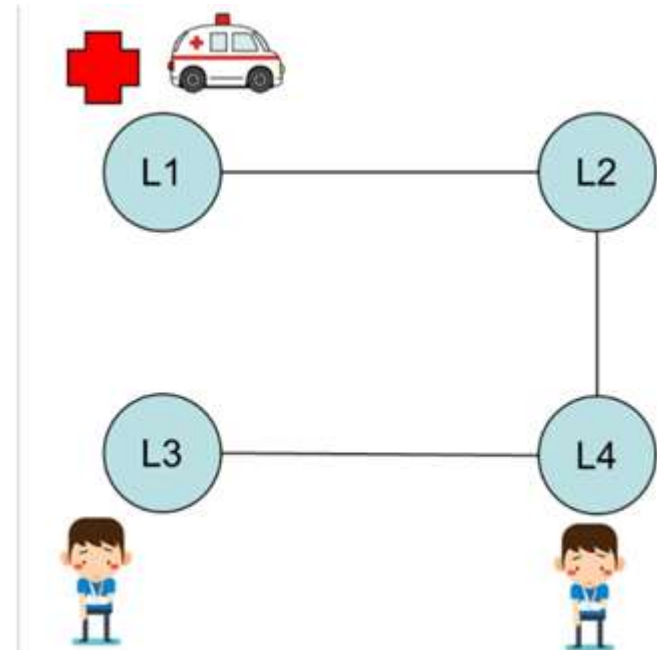
# Acción mover

```
(:action mover
  :parameters (?a - ambulancia ?l1 - localizacion ?l2 - localizacion)
  :precondition
    ( and( ambulancia-en-1 ?l1 ?a) (ir ?l1 ?l2 ))
  :effect
    ( and( not( ambulancia-en-1 ?l1 ?a) (ambulancia-en-1 ?l2 ?a) (ir ?l2 ?l1))
  )
```



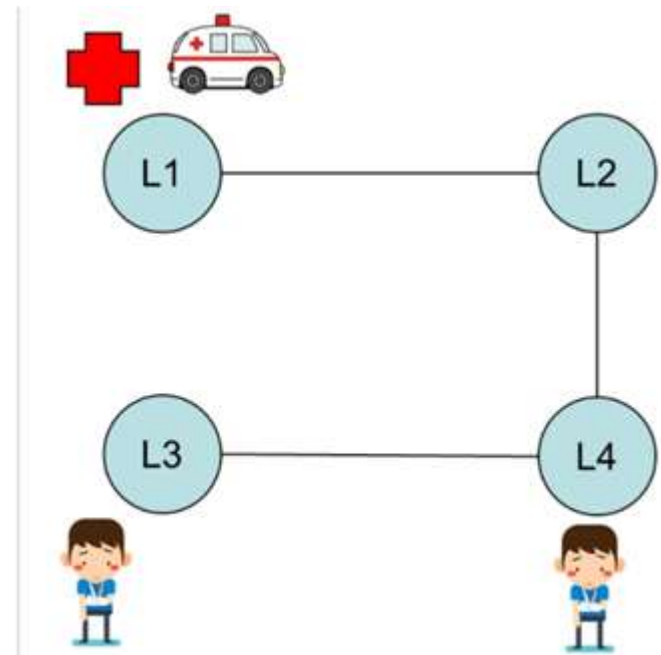
# Acción subir

```
(:action subir
  :parameters (?p - paciente ?l1 - localizacion ?a - ambulancia)
  :precondition
    (and (vacía-ambulancia) (ambulancia-en-l ?l1 ?a) (paciente-en-l ?l1 ?p) )
  :effect
    (and (not(vacía-ambulancia)) (llena-ambulancia ?p))
)
```



# Acción bajar

```
(:action bajar
  :parameters (?p - paciente ?l1 - localizacion ?a - ambulancia)
  :precondition
    (and (llena-ambulancia ?p) (ambulancia-en-1 ?l1 ?a))
  :effect
    (and (not(llena-ambulancia ?p)) (vacía-ambulancia)
      (ambulancia-en-1 ?l1 ?a) (paciente-en-1 ?l1 ?p) )
)
```



```

1 (define (domain pacientes)
2   (:requirements :typing)
3   (:types paciente ambulancia localizacion hospital)
4   (:predicates
5     (paciente-en-1 ?l - localizacion ?p - paciente )
6     (ambulancia-en-1 ?l - localizacion ?a - ambulancia)
7     (hospital-en ?l - localizacion)
8     (llena-ambulancia ?p - paciente)
9     (vacia-ambulancia)
10    (ir ?l1 -localizacion ?l2 - localizacion)
11  )
12  ;////////////////////////////////////
13  (:action mover
14    :parameters (?a - ambulancia ?l1 - localizacion ?l2 - localizacion)
15    :precondition
16      ( and( ambulancia-en-1 ?l1 ?a) (ir ?l1 ?l2 ) )
17    :effect
18      ( and( not( ambulancia-en-1 ?l1 ?a)) (ambulancia-en-1 ?l2 ?a) (ir ?l2 ?l1))
19  )
20  ;////////////////////////////////////
21  (:action subir
22    :parameters (?p - paciente ?l1 - localizacion ?a - ambulancia)
23    :precondition
24      (and (vacia-ambulancia) (ambulancia-en-1 ?l1 ?a) (paciente-en-1 ?l1 ?p) )
25    :effect
26      (and (not(vacia-ambulancia)) (llena-ambulancia ?p))
27  )
28  ;////////////////////////////////////
29  (:action bajar
30    :parameters (?p - paciente ?l1 - localizacion ?a - ambulancia)
31    :precondition
32      (and (llena-ambulancia ?p) (ambulancia-en-1 ?l1 ?a))
33    :effect
34      (and (not(llena-ambulancia ?p)) (vacia-ambulancia)
35        (ambulancia-en-1 ?l1 ?a) (paciente-en-1 ?l1 ?p) )
36  )
37  ;////////////////////////////////////
38
39 )

```

# Problema.pddl

- Identificar objetos en el dominio
  - Hospital, ambulancia, pacientes (p1,p2), localizaciones (L1,L2,L3,L4)
- Identificar estados
  - Inicial
  - objetivo

```
(define (problem trasladoPacientes)
  (:domain pacientes)
  (:objects )

  (:init (...))

  )
  (:goal (and (.....) )
  )
)
```

```

(define (problem trasladoPacientes)
  (:domain pacientes)
  (:objects p1 p2 - paciente l1 l2 l3 l4 – localizacion a – ambulancia h -
  hospital)

  (:init (hospital l1) (ambulancia l1) (vacía-ambulancia)
    (paciente-en l3 p1) (paciente-en l4 p2) (ir l1 l2) (ir l2 l4) (ir l4 l3)
  )
  (:goal (and (paciente-en l1) (paciente-en l1)
    )
  )
)

```



```


1  (define (problem trasladoPacientes)
2      (:domain pacientes)
3      (:objects
4      p1 p2 - paciente
5      l1 l2 l3 l4 - localizacion
6      a - ambulancia
7      h - hospital)
8
9      (:init
10     (hospital-en l1)
11     (paciente-en-l l3 p1)
12     (paciente-en-l l4 p2)
13     (ambulancia-en-l l1 a)
14     (vacía-ambulancia)
15     (ir l1 l2)
16     (ir l2 l4)
17     (ir l4 l3)
18     .....
19
20     )
21     (:goal (and
22     (paciente-en-l l1 p1)
23     (paciente-en-l l1 p2))
24     )
25 )

```

problema.pddl

# Ejecutar planner con archivos dominio.pddl y problema.pddl

```
wget https://bitbucket.org/ipc2018-classical/demo-submission/raw/ipc-2018-seq-opt/Singularity
sudo singularity build planner.img Singularity
mkdir rundir
cp path/to/domain.pddl rundir
cp path/to/problem.pddl rundir
RUNDIR="$(pwd)/rundir"
DOMAIN="$RUNDIR/domain.pddl"
PROBLEM="$RUNDIR/problem.pddl"
PLANFILE="$RUNDIR/sas_plan"
COSTBOUND=42 # only in cost-bounded track
ulimit -t 1800
ulimit -v 8388608
singularity run -C -H $RUNDIR planner.img $DOMAIN $PROBLEM $PLANFILE $COSTBOUND
```



```
ana@ana-Inspiron-N4010: ~/singularity/singularity/pruebas
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
conducting best first search with reopening closed nodes, (recl) bound = 2147483
647
Initializing landmark cut heuristic...
f = 7 [1 evaluated, 0 expanded, t=0s, 4396 KB]
Best heuristic value: 7 [g=0, 1 evaluated, 0 expanded, t=0s, 4396 KB]
f = 8 [2 evaluated, 1 expanded, t=0s, 4396 KB]
f = 9 [4 evaluated, 2 expanded, t=0s, 4396 KB]
Best heuristic value: 6 [g=3, 7 evaluated, 3 expanded, t=0s, 4396 KB]
f = 10 [9 evaluated, 5 expanded, t=0s, 4396 KB]
Best heuristic value: 5 [g=5, 16 evaluated, 9 expanded, t=0s, 4396 KB]
f = 11 [19 evaluated, 11 expanded, t=0s, 4396 KB]
f = 12 [26 evaluated, 17 expanded, t=0s, 4396 KB]
Best heuristic value: 4 [g=8, 27 evaluated, 18 expanded, t=0s, 4396 KB]
f = 13 [53 evaluated, 31 expanded, t=0s, 4396 KB]
f = 14 [79 evaluated, 50 expanded, t=0s, 4396 KB]
Best heuristic value: 3 [g=11, 81 evaluated, 51 expanded, t=0s, 4396 KB]
Best heuristic value: 2 [g=12, 83 evaluated, 52 expanded, t=0s, 4396 KB]
Best heuristic value: 1 [g=13, 84 evaluated, 53 expanded, t=0s, 4396 KB]
Best heuristic value: 0 [g=14, 86 evaluated, 54 expanded, t=0s, 4396 KB]
Solution found!
```

```
ana@ana-Inspiron
Archivo  Editar  Ver  Buscar  Termi
Best heuristic value: 0 [g=14,
Solution found!
Actual search time: 0s [t=0s]
mover a l1 l2 (1)
mover a l2 l4 (1)
mover a l4 l3 (1)
subir p1 l3 a (1)
mover a l3 l4 (1)
mover a l4 l2 (1)
mover a l2 l1 (1)
bajar p1 l1 a (1)
mover a l1 l2 (1)
mover a l2 l4 (1)
subir p2 l4 a (1)
mover a l4 l2 (1)
mover a l2 l1 (1)
bajar p2 l1 a (1)
Plan length: 14 step(s).
Plan cost: 14
Initial state h value: 7.
Expanded 55 state(s).
Reopened 0 state(s).
Evaluated 86 state(s).
Evaluations: 86
```

# Actividad 3. Que se espera del estudiante

- Desde el punto de vista del desarrollo de una solución informática a un problema utilizando PDDL (STRIPS)
  - Capacidad de expresar el dominio del problema correctamente en PDDL
  - Capacidad de expresar el problema concreto que se pretende resolver en PDDL.
  - Capacidad de probar, depurar y completar la solución informática del problema.
- Desde el punto de vista de la comunicación de sus resultados en formato de comunicación científica.
  - Conocimiento de la estructura de un artículo científico.
  - Planificación de la distribución de contenidos en cada uno de los apartados.
  - Expresión técnica correcta de sus conclusiones.

# Comparar rendimiento de los planificadores

- Calidad de la solución.
- Complejidad espacial (si posible) mediante, por ejemplo, número de nodos explorados
- Complejidad temporal (si bien la medición pura del tiempo de ejecución es un indicador bastante impreciso)
- Toma de datos
- Elaboración de estadísticas o grafos
- Extracción y justificación de conclusiones: elección de los mejores.

# Posibles variaciones al problema actual

- Propuesta de nuevas situaciones, pruebas, toma de resultados y extracción de estadísticas y conclusiones en formato de comunicación científica.
- Propuesta de **dos nuevas situaciones**
- Seleccionar sus criterios de relevancia y justificarlos
- **Probar las dos nuevas propuestas con el mejor planificador obtenido.**

# Posibles variaciones al problema actual

- Escalar el tamaño
  - Número de pacientes
  - Número de ambulancias
  - Número de hospitales
  - Número de localizaciones, etc.
- Aumentar la dificultad posible para el planificador
  - Restricciones entre las conexiones
  - Subáreas con o sin hospitales, etc.
- Otros factores.

# Actividad 3. Documentos a entregar

- Códigos fuente de los ficheros (dominio.pddl y problema.pddl).
- Memoria explicativa en Word o PDF de no más de 10 páginas en la que se incluya:
  - Informe detallado de lo presentado y de su funcionamiento.
  - Plan o planes que resuelven el escenario.
  - Capturas de pantalla de la salida de la ejecución de los planificadores.
  - Comparación de los planificadores utilizados.
  - Resultados de las dos situaciones adicionales.
  - Incluir las dificultades encontradas.
  - Referencias con normas APA (se penaliza si no tienen normas APA).
  - Describa el proceso por el que se puede reproducir la ejecución.



# Links de referencia

- <https://www.youtube.com/watch?v=bGv7j-NOYMw>
- <https://www.youtube.com/watch?v=Juh2kMeoC2Y>
- <https://www.youtube.com/watch?v=QrkxIxMeFP0>

//wiki

- <https://planning.wiki/ref/pddl3/domain>
- <https://www.youtube.com/watch?v=Juh2kMeoC2Y&t=386s>
- <http://editor.planning.domains/>

# Rubrica de evaluación

Planificación en STRIPS/PDDL	Descripción	Puntuación máxima (puntos)	Peso %
Criterio 1	Presenta correctamente los códigos del dominio y los problemas en STRIPS/PDDL	3	30 %
Criterio 2	Se emplean cuatro planificadores del estado del arte	2	20 %
Criterio 3	Modela dos situaciones diferentes a la planteada y las resuelve con un planificador	3	30 %
Criterio 4	Entrega un documento final con todo lo solicitado en la actividad	2	20 %
		<b>10</b>	<b>100 %</b>

# Notas Finales

- El programa desarrollado debe ser un trabajo original del estudiante. Cualquier evidencia de o trabajos iguales será calificada con una nota de cero (0).
- Revisar en la plataforma fechas de entrega. (16 de Mayo)
- **Extensión** máxima de la actividad: 10 páginas, fuente Calibri 12 e interlineado 1,5.



[www.unir.net](http://www.unir.net)