

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

Laboratorio – Planificación en STRIPS/PDDL

INTRODUCCIÓN

STRIPS (*Stanford Research Institute Problem Solver*) es un lenguaje de planificación formal desarrollado por el equipo de inteligencia artificial de *Stanford Research Institute* en los años 70. Es utilizado para describir problemas de planificación y generar planes para resolverlos. Es uno de los lenguajes de planificación más antiguos y ha sido ampliamente utilizado en la investigación de inteligencia artificial.

PDDL (*Plan Domain Definition Language*) es un lenguaje de planificación formal utilizado para describir problemas de planificación y generar planes para resolverlos. Fue desarrollado por un equipo de investigadores liderado por profesores de la universidad de California en *Berkeley* en la década de los 90. PDDL es ampliamente utilizado en la investigación de inteligencia artificial y es el lenguaje de planificación utilizado en la competencia de planificación automática de la Asociación Internacional de Inteligencia Artificial (IJCAI).

Un planificador de inteligencia artificial es un sistema o algoritmo diseñado para encontrar un plan o una serie de acciones que permitan alcanzar un objetivo específico en un entorno dado. Los planificadores de inteligencia artificial utilizan técnicas de búsqueda, optimización y lógica para encontrar un plan que maximice la probabilidad de alcanzar el objetivo deseado. Estos sistemas pueden ser utilizados en una variedad de aplicaciones, como la robótica, los sistemas de control automatizado, el diagnóstico médico y la planificación empresarial. Los planificadores de inteligencia artificial pueden trabajar con una representación formal del problema, como una descripción en lenguaje natural o un modelo matemático, y utilizar lenguajes de planificación formal como STRIPS o PDDL para generar planes. Con esta actividad se pretende la resolución autónoma de un caso (transporte, procesamiento y despacho de contenedores) que requiere acceso a los planificadores del estado del arte para solucionar un problema en STRIPS/PDDL.

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

DESCRIPCIÓN DEL PROBLEMA

El problema planteado requiere que se transporten, procesen y se despachen 8 contenedores que están en un puerto. Estos contenedores se pueden procesar en dos fábricas y se almacenan y despachan en un almacén por medio de un tren. Las capacidades y conexiones entre los diferentes espacios se pueden ver en la Figura 1. El objetivo final es que los contenedores (8) sean procesados y despachados (eliminados) en el almacén. Las acciones permitidas del tren son: quedarse parado, cargar, moverse y descargar.

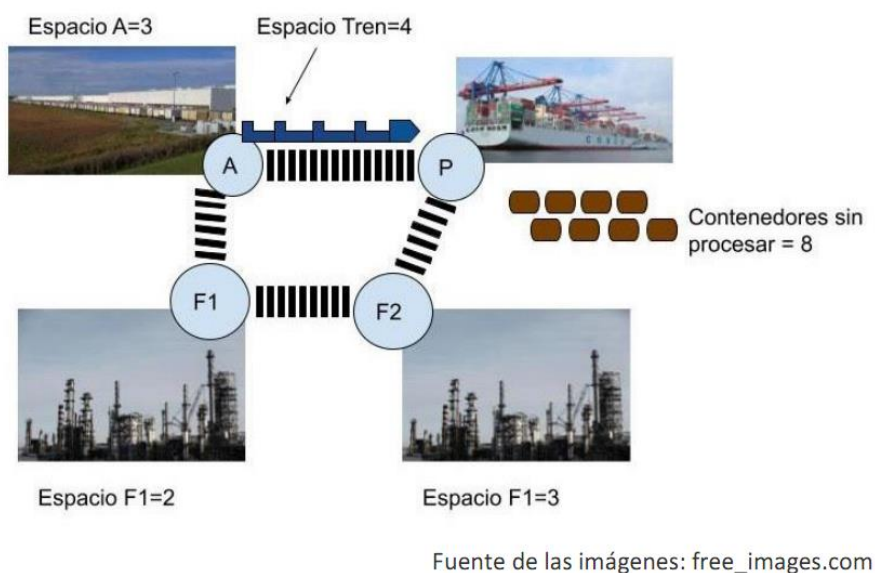


Figura 1. Esquema del problema de planificación.

METODOLOGÍA

Para encontrar el plan que resuelve el problema, se hace uso de la extensión o *plugin* PDDL de entorno de desarrollo (*IDE*) *Visual Studio Code (VSC)*. Esta extensión fue concebida y desarrollada por *Christian Muise* con la contribución de varias personalidades e instituciones relacionadas con problemas de planificación. Esta extensión por defecto usa el desarrollo de <http://planning.domains/> usando la *API* y su *solver*. La *API* contiene tres tipos de objetos:

- *Problems*: Información para cada problema individual, incluyendo su archive, el correspondiente archive de domino y las estadísticas del problema, etc.
- *Domains*: Información para cada dominio individual, incluyendo su descripción y varias estadísticas del dominio.

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

- *Collections*: Conjunto de dominios, incluyendo una colección para cada *International Planning Competitions (IPC)*, colección planificadores, etc.

También, para comparar resultados se hizo uso de un planificador del estado del arte *Soup2018* (Seipp, J., & Röger, G. 2018), el cual corre a través de máquinas virtuales en sistema operativo Linux.

SOLUCIÓN PROPUESTA

La abstracción del problema se planteó de acuerdo con la Figura 2:

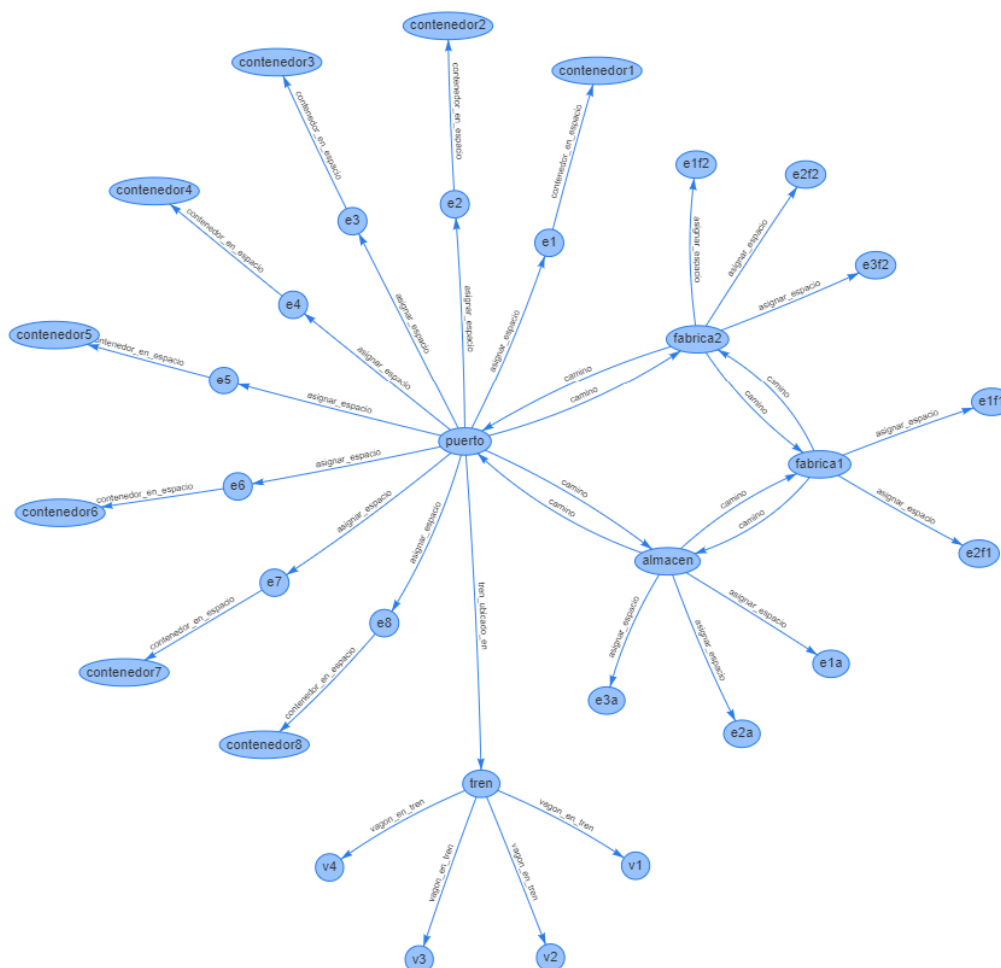


Figura 2. Diagrama de la situación base. Se observan que los 4 objetos de tipo lugar están relacionados con un predicado doble (camino), además, se observa como tren originalmente comienza en puerto y como cada espacio está correctamente asociado a cada lugar.

Situación Base:

Para el caso base, se empieza definiendo en el archivo *domain.pddl* los tipos de objetos:

```
(define (domain transporte_contenedores)
```

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

```
(:requirements :typing :strips :conditional-effects :negative-pre-conditions :disjunctive-preconditions)
```

```
;Se definen los tipos de objetos:
;Lugar: Ubicaciones espaciales de los lugares.
;Contenedor: Objeto a transportar, procesar.
;Espacio: Espacios donde pueden estar los contenedores en los lugares.
;Vagon: Espacios para almacenar y transportar el contenedor en el tren.
;Tren: Objeto que se desplaza y transporta los contenedores en los vagones.
```

```
(:types lugar contenedor espacio vagon tren)
```

Posteriormente, se definen los predicados:

```
(:predicates

  ;ubicacion de lugares en las paradas:
  (almacen_en_lugar ?l - lugar)
  (fabrica_en_lugar ?l - lugar)
  (asignar_espacio ?l - lugar ?es - espacio)

  ;estados de los contenedores:
  (contenedor_eliminado ?con - contenedor)
  (contenedor_procesado ?con - contenedor)

  ;ubicacion de contenedores:
  (contenedor_en_espacio ?es - espacio ?con - contenedor)
  (contenedor_en_vagon ?v - vagon ?con - contenedor)

  ;estado de espacio:
  (libre_espacio ?es - espacio)

  ;estado de vagones:
  (libre_vagon ?v - vagon)

  ;ubicacion del tren:
  (tren_ubicado_en ?l - lugar ?t - tren)
  (vagon_en_tren ?t1 - tren ?v - vagon)

  ;declaramos los caminos:
  (camino ?l1 - lugar ?l2 - lugar)

  ;precondicion dummy para declarar la funcion "quedarse parado":
  (siempre_true ?tren1 - tren)

)
```

Se definen las acciones que puede realizar el tren (cargar, mover, descargar y quedarse parado):

```
;CARGAR:
```

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

```
(:action cargar
  :parameters (?t1 - tren ?l1 - lugar ?v1 - vagon ?con1 - contenedor
    ?es - espacio)
  :precondition (and
    ;El tren debe estar en el lugar l1.
    ;El vagon debe estar libre.
    ;El contenedor debe estar en el espacio.
    ;El espacio debe estar en el lugar l1.
    (tren_ubicado_en ?l1 ?t1)
    (libre_vagon ?v1)
    (contenedor_en_espacio ?es ?con1)
    (not(contenedor_eliminado ?con1))
    (asignar_espacio ?l1 ?es)
  )
  :effect (and
    ;El contenedor deja de estar en el espacio.
    ;El vagon deja de estar libre.
    ;El contenedor esta en el vagon.
    ;El espacio en l1 empieza a estar libre.
    (not (contenedor_en_espacio ?es ?con1))
    (not (libre_vagon ?v1))
    (contenedor_en_vagon ?v1 ?con1)
    (libre_espacio ?es)
  )
)

;MOVER:
(:action mover
  :parameters (?l1 - lugar ?l2 - lugar ?t1 - tren)
  :precondition (and
    ;El tren debe estar en el lugar l1.
    ;Debe haber un camino de conexion entre l1 y l2.
    (tren_ubicado_en ?l1 ?t1)
    (camino ?l1 ?l2)
  )
  :effect (and
    ;El tren deja de estar en l1.
    ;El tren esta en l2.
    (not (tren_ubicado_en ?l1 ?t1))
    (tren_ubicado_en ?l2 ?t1)
  )
)

;DESCARGAR:
(:action descargar
  :parameters (?t1 - tren ?l1 - lugar ?v1 - vagon ?con1 - contenedor
    ?es - espacio)
  :precondition (and
    ;El espacio debe estar en el lugar l1.
    ;el espacio esta libre o el almacen esta en el lugar.
    ;y contenedor esta procesado.
    ;El contenedor debe estar en el vagon.
    ;El tren debe estar en el lugar l1.
    (asignar_espacio ?l1 ?es)
    (or
```

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

```

        (libre_espacio ?es)
        (and
        (almacen_en_lugar ?l1)
        (contenedor_procesado ?con1)
        )
    )
    (contenedor_en_vagon ?v1 ?con1)
    (tren_ubicado_en ?l1 ?t1)
    ;la siguiente es una condicion especial:
    ;en el puerto no podemos descargar nada, solo podemos cargar.
    ;asi que validamos que el lugar donde vamos a descargar no
    tenga puerto.
    (or(almacen_en_lugar ?l1)(fabrica_en_lugar ?l1))
    )
    ;el vagon empieza a estar libre.
    ;es espacio deja de estar libre.
    ;el contenedor deja de estar en el vagon.
    ;el contenedor empieza a estar en el espacio.
    :effect (and
    (not (contenedor_en_vagon ?v1 ?con1))
    (libre_vagon ?v1)
    (not (libre_espacio ?es ))
    ;el siguiente when analiza la situacion en la que desamos eli-
    minar un contenedor
    ;si se nos ocurre dejar un contenedor ya procesado en almacen
    ;este liberara el espacio y quedara como "contenodore_elimi-
    nado"
    (when
    ;Antecedent
    (and
    (almacen_en_lugar ?l1)
    (contenedor_procesado ?con1)
    )
    ;Consequence
    (and
    (contenedor_eliminado ?con1)
    (libre_espacio ?es)
    ;Declaramos al espacio libre porque en los efectos de la
    ;acción, se elimina este predicado lo que seria erroneo
    ;para un contenedor ya procesado en almancen.
    )
    )
    ;El siguiente when comprueba si el contenedor no fue eliminado
    en el anterior when, para asignar el contenedor a un espacio.
    ;Esto se lo realiza porque no se permiten el uso de not en
    ;consecuencias así que antes de intentar eliminar un predicado
    ;es mejor validar cuando si escribirlo.
    (when
    ;Antecedent
    (and
    (not
    (contenedor_eliminado ?con1)
    )
    )
    ;Consequence

```

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

```

        (and
          (contenedor_en_espacio ?es ?con1)
        )
      )
      ;El siguiente when analiza si hemos dejado el contenido en una
      fabrica,
      ;si es asi inmediatamente despues el contenedor se marca como
      "procesado".
      (when
        ;Antecedent
        (and
          (fabrica_en_lugar ?l1)
        )
        ;Consequence
        (and
          (contenedor_procesado ?con1)
        )
      )
    )
  )
)

```

;A continuacion se realizar la accion quedarse parado.
 ;como no se pueden declarar precondiciones y efectos vacios entonces
 ;utilizamos un predicado siempre positivo llamado "siempre_true"

```

(:action quedarse_parado
  :parameters (?tren1 - tren)
  :precondition (and
    (siempre_true ?tren1)
  )
  :effect (and
    (siempre_true ?tren1)
  )
)
)
)

```

Se define el archivo *problem.pddl* donde se establecen el número de lugares, espacios, trenes y contenedores:

```

(define (problem contenedores_almacen)
  (:domain transporte_contenedores)

  (:objects
    ;Hay un tren:
    tren - tren
    ;Hay dos fabricas, un puerto y un almacen:
    fabrica1 fabrica2 puerto almacen - lugar
    ;Ocho espacios asociados al puerto:
    e1 e2 e3 e4 e5 e6 e7 e8 - espacio
    ;Dos espacios asociados almacen fabrica 1:
    e1f1 e2f1 - espacio
    ;Tres espacios asociados almacen fabrica 2:
    e1f2 e2f2 e3f2 - espacio
    ;Tres espacios en almacen:
    e1a e2a e3a - espacio
  )
)

```

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

```

;Ocho contenedores:
contenedor1 contenedor2 contenedor3 contenedor4 contenedor5 conte-
nedor6 contenedor7 contenedor8 - contenedor
;Cuatro vagones en el tren:
v1 v2 v3 v4 - vagon
)

```

Se definen las condiciones iniciales y relaciones entre los objetos:

```

(:init
;Definimos que tenemos en cada lugar:
(almacen_en_lugar almacen)
(fabrica_en_lugar fabrica1)
(fabrica_en_lugar fabrica2)
;Definimos donde esta el tren:
(tren_ubicado_en puerto tren)
;Donde estan los contenedores inicialmente:
(contenedor_en_espacio e1 contenedor1)
(contenedor_en_espacio e2 contenedor2)
(contenedor_en_espacio e3 contenedor3)
(contenedor_en_espacio e4 contenedor4)
(contenedor_en_espacio e5 contenedor5)
(contenedor_en_espacio e6 contenedor6)
(contenedor_en_espacio e7 contenedor7)
(contenedor_en_espacio e8 contenedor8)
;Que espacios estan libres inicialmente:
(libre_espacio e1f1)
(libre_espacio e2f1)
(libre_espacio e1f2)
(libre_espacio e2f2)
(libre_espacio e3f2)
(libre_espacio e1a)
(libre_espacio e2a)
(libre_espacio e3a)
;Donde pertenece cada espacio:
(asignar_espacio puerto e1)
(asignar_espacio puerto e2)
(asignar_espacio puerto e3)
(asignar_espacio puerto e4)
(asignar_espacio puerto e5)
(asignar_espacio puerto e6)
(asignar_espacio puerto e7)
(asignar_espacio puerto e8)
(asignar_espacio fabrica1 e1f1)
(asignar_espacio fabrica1 e2f1)
(asignar_espacio fabrica2 e1f2)
(asignar_espacio fabrica2 e2f2)
(asignar_espacio fabrica2 e3f2)
(asignar_espacio almacen e1a)
(asignar_espacio almacen e2a)
(asignar_espacio almacen e3a)
;Que vagones estan libres inicialmente:
(libre_vagon v1)
(libre_vagon v2)
(libre_vagon v3)

```


Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

```

(libre_vagon v4)
;Conexiones entre los lugares:
(camino fabrica1 fabrica2)
(camino fabrica2 puerto)
(camino puerto almacen)
(camino almacen fabrica1)
(camino fabrica2 fabrica1)
(camino puerto fabrica2)
(camino almacen puerto)
(camino fabrica1 almacen)
;Vagones en tren:
(vagon_en_tren tren v1)
(vagon_en_tren tren v2)
(vagon_en_tren tren v3)
(vagon_en_tren tren v4)
;Condicion para vagon parado:
(siempre_true tren)
)

```

Finalmente, se establece el objetivo:

```

;Los contenedores deben estar todos eliminados:
(:goal (and
  (forall (?c - contenedor)
    (contenedor_eliminado ?c)
  )
)
)
)

```

Las dos situaciones adicionales o problemas modelados fueron:

Situación adicional 1:

Camino bidireccional entre la Fabrica 2 y el Almacén. Para esto, solo se adiciona en el archivo *problem.pddl* en las condiciones iniciales lo siguiente:

```

(:init
;...
;situacion extra
(camino fabrica2 almacen)
(camino almacen fabrica2)
;...
)

```

Situación adicional 2:

8 vagones en el tren. Para esto, se deben adicionar en el archivo *problem.pddl* en los apartados de objetos y condiciones iniciales respectivamente lo siguiente:

```

(:objects
;...

```

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

```

;Ocho vagones en el tren:
v1 v2 v3 v4 v5 v6 v7 v8 - vagon
;...
)

(:init
;...
;Que vagones estan libres inicialmente:
(libre_vagon v1)
(libre_vagon v2)
(libre_vagon v3)
(libre_vagon v4)
(libre_vagon v5)
(libre_vagon v6)
(libre_vagon v7)
(libre_vagon v8)
;Vagones en tren:
(vagon_en_tren tren v1)
(vagon_en_tren tren v2)
(vagon_en_tren tren v3)
(vagon_en_tren tren v4)
(vagon_en_tren tren v5)
(vagon_en_tren tren v6)
(vagon_en_tren tren v7)
(vagon_en_tren tren v8)
;...
)

```

RESULTADOS

Los resultados para la situación o problema base en *VSC*, arrojó los siguientes resultados:

found with cost: 89	0.00600: (descargar tren almacen v4 contenedor5 e1a)
Total time: 0.048	0.00700: (descargar tren almacen v3 contenedor6 e3a)
Nodes generated during search: 4608	0.00800: (mover almacen puerto tren)
Nodes expanded during search: 4284	0.00900: (cargar tren puerto v3 contenedor4 e4)
IW search completed	0.01000: (cargar tren puerto v4 contenedor3 e3)
Plan found:	0.01100: (mover puerto fabrica2 tren)
0.00100: (cargar tren puerto v1 contenedor8 e8)	0.01200: (descargar tren fabrica2 v4 contenedor3 e1f2)
0.00200: (cargar tren puerto v2 contenedor7 e7)	0.01300: (descargar tren fabrica2 v3 contenedor4 e3f2)
0.00300: (cargar tren puerto v3 contenedor6 e6)	0.01400: (mover fabrica2 puerto tren)
0.00400: (cargar tren puerto v4 contenedor5 e5)	0.01500: (cargar tren puerto v3 contenedor2 e2)
0.00500: (mover puerto almacen tren)	

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

0.01600: (cargar tren puerto v4 contenedor1 e1)
0.01700: (mover puerto almacen tren)
0.01800: (descargar tren almacen v4 contenedor1 e2a)
0.01900: (cargar tren almacen v4 contenedor6 e3a)
0.02000: (descargar tren almacen v3 contenedor2 e3a)
0.02100: (cargar tren almacen v3 contenedor1 e2a)
0.02200: (descargar tren almacen v4 contenedor6 e2a)
0.02300: (cargar tren almacen v4 contenedor2 e3a)
0.02400: (descargar tren almacen v2 contenedor7 e3a)
0.02500: (cargar tren almacen v2 contenedor7 e3a)
0.02600: (descargar tren almacen v1 contenedor8 e3a)
0.02700: (mover almacen fabrica1 tren)
0.02800: (descargar tren fabrica1 v3 contenedor1 e1f1)
0.02900: (descargar tren fabrica1 v2 contenedor7 e2f1)
0.03000: (cargar tren fabrica1 v3 contenedor7 e2f1)
0.03100: (mover fabrica1 almacen tren)
0.03200: (descargar tren almacen v3 contenedor7 e3a)
0.03300: (descargar tren almacen v4 contenedor2 e3a)
0.03400: (cargar tren almacen v2 contenedor5 e1a)
0.03500: (descargar tren almacen v2 contenedor5 e1a)
0.03600: (cargar tren almacen v2 contenedor6 e2a)
0.03700: (cargar tren almacen v3 contenedor8 e3a)
0.03800: (descargar tren almacen v3 contenedor8 e3a)
0.03900: (mover almacen fabrica1 tren)
0.04000: (descargar tren fabrica1 v2 contenedor6 e2f1)
0.04100: (cargar tren fabrica1 v3 contenedor1 e1f1)
0.04200: (mover fabrica1 almacen tren)
0.04300: (descargar tren almacen v3 contenedor1 e2a)
0.04400: (cargar tren almacen v3 contenedor2 e3a)
0.04500: (descargar tren almacen v3 contenedor2 e3a)

0.04600: (cargar tren almacen v3 contenedor8 e3a)
0.04700: (cargar tren almacen v4 contenedor5 e1a)
0.04800: (descargar tren almacen v4 contenedor5 e1a)
0.04900: (mover almacen fabrica1 tren)
0.05000: (descargar tren fabrica1 v3 contenedor8 e1f1)
0.05100: (cargar tren fabrica1 v3 contenedor6 e2f1)
0.05200: (mover fabrica1 almacen tren)
0.05300: (descargar tren almacen v3 contenedor6 e3a)
0.05400: (cargar tren almacen v1 contenedor2 e3a)
0.05500: (descargar tren almacen v1 contenedor2 e3a)
0.05600: (cargar tren almacen v4 contenedor5 e1a)
0.05700: (descargar tren almacen v4 contenedor5 e1a)
0.05800: (mover almacen fabrica1 tren)
0.05900: (cargar tren fabrica1 v3 contenedor8 e1f1)
0.06000: (mover fabrica1 almacen tren)
0.06100: (descargar tren almacen v3 contenedor8 e3a)
0.06200: (cargar tren almacen v3 contenedor2 e3a)
0.06300: (descargar tren almacen v3 contenedor2 e3a)
0.06400: (cargar tren almacen v3 contenedor5 e1a)
0.06500: (cargar tren almacen v4 contenedor2 e3a)
0.06600: (descargar tren almacen v3 contenedor5 e3a)
0.06700: (mover almacen fabrica1 tren)
0.06800: (descargar tren fabrica1 v4 contenedor2 e2f1)
0.06900: (mover fabrica1 fabrica2 tren)
0.07000: (cargar tren fabrica2 v4 contenedor3 e1f2)
0.07100: (mover fabrica2 puerto tren)
0.07200: (mover puerto almacen tren)
0.07300: (descargar tren almacen v4 contenedor3 e1a)
0.07400: (cargar tren almacen v4 contenedor5 e3a)
0.07500: (mover almacen fabrica1 tren)
0.07600: (descargar tren fabrica1 v4 contenedor5 e1f1)
0.07700: (cargar tren fabrica1 v1 contenedor2 e2f1)
0.07800: (mover fabrica1 almacen tren)

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

0.07900: (descargar tren almacen v1 contenedor2 e3a)
0.08000: (mover almacen fabrica1 tren)
0.08100: (cargar tren fabrica1 v1 contenedor5 e1f1)
0.08200: (mover fabrica1 almacen tren)
0.08300: (descargar tren almacen v1 contenedor5 e3a)
0.08400: (mover almacen fabrica1 tren)
0.08500: (mover fabrica1 fabrica2 tren)
0.08600: (cargar tren fabrica2 v1 contenedor4 e3f2)

0.08700: (mover fabrica2 puerto tren)
0.08800: (mover puerto almacen tren)
0.08900: (descargar tren almacen v1 contenedor4 e3a)
Metric: 0.089
Makespan: 0.089
States evaluated: undefined
Planner found 1 plan(s) in 0.851secs.

Los resultados para la situación o problema base en el planificador del estado del arte *Soup2018*, arrojó los siguientes resultados:

Solution found!

Actual search time: 0.0600017s
[t=0.0757556s]

cargar tren puerto v1 contenedor1 e1
cargar tren puerto v2 contenedor2 e2
cargar tren puerto v3 contenedor3 e3
cargar tren puerto v4 contenedor4 e4
mover puerto fabrica2 tren
descargar tren fabrica2 v2 contenedor2 e1f2
cargar tren fabrica2 v2 contenedor2 e1f2
descargar tren fabrica2 v3 contenedor3 e1f2
cargar tren fabrica2 v3 contenedor3 e1f2
descargar tren fabrica2 v4 contenedor4 e1f2
cargar tren fabrica2 v4 contenedor4 e1f2
descargar tren fabrica2 v1 contenedor1 e1f2
mover fabrica2 puerto tren
cargar tren puerto v1 contenedor5 e5
mover puerto fabrica2 tren
descargar tren fabrica2 v1 contenedor5 e2f2
mover fabrica2 puerto tren
cargar tren puerto v1 contenedor6 e6
mover puerto fabrica2 tren
descargar tren fabrica2 v1 contenedor6 e3f2
cargar tren fabrica2 v1 contenedor1 e1f2
mover fabrica2 puerto tren
mover puerto almacen tren

descargar tren almacen v1 contenedor1 e1a
descargar tren almacen v2 contenedor2 e1a
descargar tren almacen v3 contenedor3 e1a
descargar tren almacen v4 contenedor4 e1a
mover almacen puerto tren
cargar tren puerto v1 contenedor7 e7
cargar tren puerto v2 contenedor8 e8
mover puerto fabrica2 tren
cargar tren fabrica2 v3 contenedor5 e2f2
cargar tren fabrica2 v4 contenedor6 e3f2
mover fabrica2 fabrica1 tren
mover fabrica1 almacen tren
descargar tren almacen v3 contenedor5 e1a
descargar tren almacen v4 contenedor6 e1a
mover almacen fabrica1 tren
descargar tren fabrica1 v1 contenedor7 e1f1
cargar tren fabrica1 v1 contenedor7 e1f1
mover fabrica1 almacen tren
descargar tren almacen v1 contenedor7 e1a
mover almacen fabrica1 tren
descargar tren fabrica1 v2 contenedor8 e1f1
cargar tren fabrica1 v1 contenedor8 e1f1
mover fabrica1 almacen tren
descargar tren almacen v1 contenedor8 e1a
Plan length: 47 step(s).
Plan cost: 47

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

Expanded 261 state(s).	Search time: 0.0600017s
Reopened 0 state(s).	Total time: 0.0757556s
Evaluated 1681 state(s).	Solution found.
Evaluations: 3882	Peak memory: 6680 KB
Generated 1912 state(s).	exitcode: 0
Dead ends: 0 state(s).	remaining time: 1799.53
Number of registered states: 1681	config 4: relative time 23, remaining 1338
Int hash set load factor: 1681/2048 = 0.820801	g bound: infinity
Int hash set resizes: 11	

Los resultados para la situación o problema adicional 1 en VSC, arrojó los siguientes resultados:

Total time: 0.464	0.01200: (mover puerto fabrica2 tren)
Nodes generated during search: 5464	0.01300: (descargar tren fabrica2 v1 contenedor4 e3f2)
Nodes expanded during search: 552	0.01400: (cargar tren fabrica2 v1 contenedor4 e3f2)
Plan found with cost: 54	0.01500: (mover fabrica2 almacen tren)
BFS search completed	0.01600: (descargar tren almacen v1 contenedor4 e1a)
Plan found:	0.01700: (mover almacen puerto tren)
0.00100: (cargar tren puerto v3 contenedor8 e8)	0.01800: (cargar tren puerto v1 contenedor3 e3)
0.00200: (cargar tren puerto v2 contenedor7 e7)	0.01900: (mover puerto fabrica2 tren)
0.00300: (cargar tren puerto v1 contenedor6 e6)	0.02000: (descargar tren fabrica2 v1 contenedor3 e3f2)
0.00400: (mover puerto fabrica2 tren)	0.02100: (cargar tren fabrica2 v1 contenedor3 e3f2)
0.00500: (descargar tren fabrica2 v1 contenedor6 e1f2)	0.02200: (mover fabrica2 almacen tren)
0.00600: (mover fabrica2 puerto tren)	0.02300: (descargar tren almacen v1 contenedor3 e2a)
0.00700: (cargar tren puerto v1 contenedor5 e5)	0.02400: (mover almacen puerto tren)
0.00800: (mover puerto fabrica2 tren)	0.02500: (cargar tren puerto v1 contenedor2 e2)
0.00900: (descargar tren fabrica2 v1 contenedor5 e2f2)	0.02600: (mover puerto fabrica2 tren)
0.01000: (mover fabrica2 puerto tren)	0.02700: (descargar tren fabrica2 v1 contenedor2 e3f2)
0.01100: (cargar tren puerto v1 contenedor4 e4)	

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

0.02800: (cargar tren fabrica2 v1 contenedor2 e3f2)	0.04300: (mover puerto almacen tren)
0.02900: (mover fabrica2 almacen tren)	0.04400: (descargar tren almacen v2 contenedor6 e3a)
0.03000: (descargar tren almacen v1 contenedor2 e3a)	0.04500: (mover almacen fabrica2 tren)
0.03100: (mover almacen fabrica2 tren)	0.04600: (descargar tren fabrica2 v3 contenedor8 e1f2)
0.03200: (cargar tren fabrica2 v4 contenedor5 e2f2)	0.04700: (cargar tren fabrica2 v2 contenedor8 e1f2)
0.03300: (mover fabrica2 almacen tren)	0.04800: (mover fabrica2 almacen tren)
0.03400: (descargar tren almacen v4 contenedor5 e3a)	0.04900: (descargar tren almacen v2 contenedor8 e2a)
0.03500: (mover almacen fabrica2 tren)	0.05000: (mover almacen fabrica2 tren)
0.03600: (descargar tren fabrica2 v2 contenedor7 e2f2)	0.05100: (descargar tren fabrica2 v1 contenedor1 e1f2)
0.03700: (cargar tren fabrica2 v2 contenedor6 e1f2)	0.05200: (cargar tren fabrica2 v3 contenedor1 e1f2)
0.03800: (cargar tren fabrica2 v4 contenedor7 e2f2)	0.05300: (mover fabrica2 almacen tren)
0.03900: (mover fabrica2 almacen tren)	0.05400: (descargar tren almacen v3 contenedor1 e2a)
0.04000: (descargar tren almacen v4 contenedor7 e3a)	Metric: 0.054
0.04100: (mover almacen puerto tren)	Makespan: 0.054
0.04200: (cargar tren puerto v1 contenedor1 e1)	States evaluated: undefined
	Planner found 1 plan(s) in 1.004secs.

Los resultados para la situación o problema adicional 1 en el planificador del estado del arte, arrojó los siguientes resultados:

Solution found!	cargar tren fabrica2 v2 contenedor2 e1f2
Actual search time: 0.259991s [t=0.27599s]	descargar tren fabrica2 v3 contenedor3 e1f2
cargar tren puerto v1 contenedor1 e1	cargar tren fabrica2 v3 contenedor3 e1f2
cargar tren puerto v2 contenedor2 e2	descargar tren fabrica2 v1 contenedor1 e1f2
cargar tren puerto v3 contenedor3 e3	cargar tren fabrica2 v1 contenedor1 e1f2
cargar tren puerto v4 contenedor7 e7	mover fabrica2 almacen tren
mover puerto fabrica2 tren	descargar tren almacen v1 contenedor1 e3a
descargar tren fabrica2 v4 contenedor7 e3f2	mover almacen puerto tren
cargar tren fabrica2 v4 contenedor7 e3f2	cargar tren puerto v1 contenedor8 e8
descargar tren fabrica2 v2 contenedor2 e1f2	mover puerto almacen tren

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

descargar tren almacen v4 contenedor7 e2a	mover fabrica2 almacen tren
mover almacen puerto tren	descargar tren almacen v1 contenedor5 e1a
cargar tren puerto v4 contenedor4 e4	descargar tren almacen v4 contenedor6 e1a
mover puerto fabrica2 tren	Plan length: 46 step(s).
descargar tren fabrica2 v4 contenedor4 e1f2	Plan cost: 46
cargar tren fabrica2 v4 contenedor4 e1f2	Expanded 940 state(s).
descargar tren fabrica2 v1 contenedor8 e1f2	Reopened 0 state(s).
cargar tren fabrica2 v1 contenedor8 e1f2	Evaluated 7433 state(s).
mover fabrica2 almacen tren	Evaluations: 16744
descargar tren almacen v1 contenedor8 e1a	Generated 9270 state(s).
mover almacen puerto tren	Dead ends: 0 state(s).
cargar tren puerto v1 contenedor5 e5	Number of registered states: 7433
mover puerto fabrica2 tren	Int hash set load factor: 7433/16384 =
descargar tren fabrica2 v1 contenedor5 e1f2	0.453674
mover fabrica2 puerto tren	Int hash set resizes: 14
cargar tren puerto v1 contenedor6 e6	Search time: 0.259991s
mover puerto fabrica2 tren	Total time: 0.27599s
descargar tren fabrica2 v1 contenedor6 e2f2	Solution found.
mover fabrica2 almacen tren	Peak memory: 7576 KB
descargar tren almacen v2 contenedor2 e1a	exitcode: 0
descargar tren almacen v3 contenedor3 e1a	
descargar tren almacen v4 contenedor4 e1a	remaining time: 1636.96
mover almacen fabrica2 tren	config 15: relative time 29, remaining 972
cargar tren fabrica2 v1 contenedor5 e1f2	g bound: infinity
cargar tren fabrica2 v4 contenedor6 e2f2	

Los resultados para la situación o problema adicional 2 en VSC, arrojó los siguientes resultados:

Plan found with cost: 67	0.00200: (cargar tren puerto v7 contenedor7 e7)
Total time: 0.068	0.00300: (cargar tren puerto v8 contenedor6 e6)
Nodes generated during search: 4504	0.00400: (mover puerto almacen tren)
Nodes expanded during search: 3995	0.00500: (descargar tren almacen v8 contenedor6 e1a)
IW search completed	0.00600: (descargar tren almacen v7 contenedor7 e3a)
Plan found:	
0.00100: (cargar tren puerto v1 contenedor8 e8)	

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte
0.00700: (mover almacen puerto tren)	0.03000: (descargar tren fabrica1 v5 contenedor6 e1f1)
0.00800: (cargar tren puerto v7 contenedor5 e5)	0.03100: (cargar tren fabrica1 v8 contenedor7 e2f1)
0.00900: (cargar tren puerto v8 contenedor4 e4)	0.03200: (mover fabrica1 almacen tren)
0.01000: (mover puerto fabrica2 tren)	0.03300: (descargar tren almacen v8 contenedor7 e1a)
0.01100: (descargar tren fabrica2 v8 contenedor4 e1f2)	0.03400: (mover almacen fabrica1 tren)
0.01200: (descargar tren fabrica2 v1 contenedor8 e3f2)	0.03500: (cargar tren fabrica1 v1 contenedor6 e1f1)
0.01300: (cargar tren fabrica2 v8 contenedor8 e3f2)	0.03600: (mover fabrica1 almacen tren)
0.01400: (descargar tren fabrica2 v7 contenedor5 e3f2)	0.03700: (descargar tren almacen v1 contenedor6 e3a)
0.01500: (mover fabrica2 puerto tren)	0.03800: (mover almacen puerto tren)
0.01600: (mover puerto almacen tren)	0.03900: (cargar tren puerto v1 contenedor3 e3)
0.01700: (descargar tren almacen v8 contenedor8 e2a)	0.04000: (cargar tren puerto v3 contenedor2 e2)
0.01800: (cargar tren almacen v6 contenedor6 e1a)	0.04100: (mover puerto fabrica2 tren)
0.01900: (descargar tren almacen v6 contenedor6 e1a)	0.04200: (descargar tren fabrica2 v1 contenedor3 e1f2)
0.02000: (cargar tren almacen v8 contenedor7 e3a)	0.04300: (cargar tren fabrica2 v8 contenedor5 e3f2)
0.02100: (mover almacen fabrica1 tren)	0.04400: (mover fabrica2 fabrica1 tren)
0.02200: (descargar tren fabrica1 v8 contenedor7 e2f1)	0.04500: (descargar tren fabrica1 v3 contenedor2 e2f1)
0.02300: (mover fabrica1 fabrica2 tren)	0.04600: (mover fabrica1 almacen tren)
0.02400: (cargar tren fabrica2 v8 contenedor4 e1f2)	0.04700: (descargar tren almacen v8 contenedor5 e3a)
0.02500: (mover fabrica2 puerto tren)	0.04800: (mover almacen fabrica1 tren)
0.02600: (mover puerto almacen tren)	0.04900: (cargar tren fabrica1 v1 contenedor2 e2f1)
0.02700: (descargar tren almacen v8 contenedor4 e3a)	0.05000: (mover fabrica1 almacen tren)
0.02800: (cargar tren almacen v5 contenedor6 e1a)	0.05100: (descargar tren almacen v1 contenedor2 e3a)
0.02900: (mover almacen fabrica1 tren)	0.05200: (mover almacen puerto tren)

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

0.05300: (cargar tren puerto v1 contenedor1 e1)	0.06200: (mover puerto almacen tren)
0.05400: (mover puerto almacen tren)	0.06300: (descargar tren almacen v8 contenedor3 e3a)
0.05500: (descargar tren almacen v1 contenedor1 e3a)	0.06400: (mover almacen fabrica1 tren)
0.05600: (cargar tren almacen v2 contenedor1 e3a)	0.06500: (cargar tren fabrica1 v1 contenedor1 e2f1)
0.05700: (mover almacen fabrica1 tren)	0.06600: (mover fabrica1 almacen tren)
0.05800: (descargar tren fabrica1 v2 contenedor1 e2f1)	0.06700: (descargar tren almacen v1 contenedor1 e3a)
0.05900: (mover fabrica1 fabrica2 tren)	Metric: 0.067
0.06000: (cargar tren fabrica2 v8 contenedor3 e1f2)	Makespan: 0.067
0.06100: (mover fabrica2 puerto tren)	States evaluated: undefined
	Planner found 1 plan(s) in 0.966secs.

Los resultados para la situación o problema adicional 2 en el planificador del estado del arte, arrojó los siguientes resultados:

Solution found!	descargar tren fabrica1 v3 contenedor3 e1f1
Actual search time: 0.100002s [t=0.113155s]	cargar tren fabrica1 v1 contenedor3 e1f1
cargar tren puerto v1 contenedor1 e1	descargar tren fabrica1 v4 contenedor4 e1f1
cargar tren puerto v2 contenedor2 e2	cargar tren fabrica1 v2 contenedor4 e1f1
cargar tren puerto v3 contenedor3 e3	mover fabrica1 almacen tren
cargar tren puerto v4 contenedor4 e4	descargar tren almacen v1 contenedor3 e1a
cargar tren puerto v5 contenedor5 e5	descargar tren almacen v2 contenedor4 e1a
cargar tren puerto v6 contenedor6 e6	mover almacen fabrica1 tren
cargar tren puerto v7 contenedor7 e7	descargar tren fabrica1 v5 contenedor5 e1f1
cargar tren puerto v8 contenedor8 e8	cargar tren fabrica1 v1 contenedor5 e1f1
mover puerto almacen tren	descargar tren fabrica1 v6 contenedor6 e1f1
mover almacen fabrica1 tren	cargar tren fabrica1 v2 contenedor6 e1f1
descargar tren fabrica1 v1 contenedor1 e1f1	mover fabrica1 almacen tren
cargar tren fabrica1 v1 contenedor1 e1f1	descargar tren almacen v1 contenedor5 e1a
descargar tren fabrica1 v2 contenedor2 e1f1	descargar tren almacen v2 contenedor6 e1a
cargar tren fabrica1 v2 contenedor2 e1f1	mover almacen fabrica1 tren
mover fabrica1 almacen tren	descargar tren fabrica1 v7 contenedor7 e1f1
descargar tren almacen v1 contenedor1 e1a	cargar tren fabrica1 v1 contenedor7 e1f1
descargar tren almacen v2 contenedor2 e1a	descargar tren fabrica1 v8 contenedor8 e1f1
mover almacen fabrica1 tren	cargar tren fabrica1 v2 contenedor8 e1f1

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

mover fabrica1 almacen tren

descargar tren almacen v1 contenedor7 e1a

descargar tren almacen v2 contenedor8 e1a

Plan length: 41 step(s).

Plan cost: 41

Expanded 165 state(s).

Reopened 0 state(s).

Evaluated 1956 state(s).

Evaluations: 2120

Generated 2070 state(s).

Dead ends: 0 state(s).

Number of registered states: 1956

Int hash set load factor: $1956/2048 = 0.955078$

Int hash set resizes: 11

Search time: 0.100002s

Total time: 0.113155s

Solution found.

Peak memory: 7872 KB

exitcode: 0

remaining time: 1790.66

config 22: relative time 28, remaining 724

g bound: infinity

En resumen, los resultados se pueden ver en la Tabla 1:

Tabla 1. Comparativo de costos, tiempo de ejecución y nodos expandidos.

Situación	Visual Studio Code			Planificador Soup2018		
	Costo	Tiempo de ejecución (s)	Nodos expandidos	Costo	Tiempo de ejecución (s)	Nodos expandidos
Situación Base	89	0,851	4284	47	0,076	261
Situación 1 (camino entre Fabrica 2 y Almacen)	54	1,004	552	46	0,276	940
Situación 2 (8 vagones)	67	0,966	3995	41	0,113	165

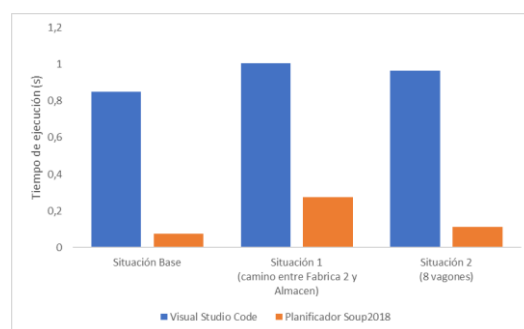
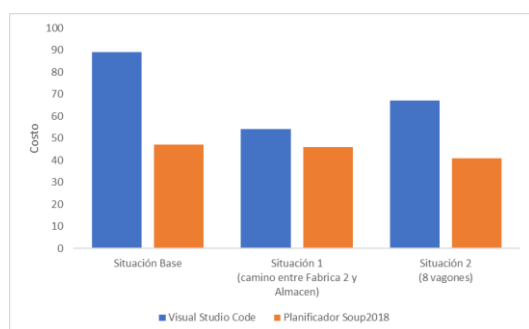


Figura 3. Costos entre las diferentes situaciones.

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

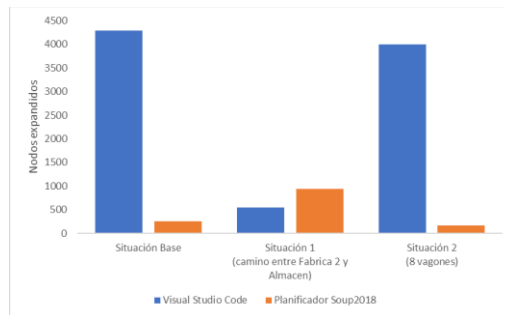


Figura 4. Nodos expandidos en cada una de las situaciones.

DIFICULTADES ENCONTRADAS

- Antes de iniciar la programación del código se debe evaluar qué información requiere la creación de un objeto o si es que sólo usando predicados se puede capturar la situación de una forma suficientemente operable para el objetivo del problema.
- Algunas funcionalidades de *pddl* no son compatibles entre sí, por ejemplo, *when* no acepta *not (delete)* en las consecuencias de estos.
- Con el uso de cláusulas condicionales (*conditional-effects*) con *when* se observaron planes con mayor cantidad de pasos, lo que sugiere que el planificador internamente usa otro paradigma de búsqueda cuando están activados los efectos condicionales que cuando no.

ANÁLISIS DE RESULTADOS Y CONCLUSIONES.

- Se pudo observar que para la situación base, el planificador encuentra una solución, pero no es la óptima. Esto se pudo comprobar en la salida del planificador, cuando obtiene una serie de pasos adicionales intermedios, donde el tren descarga contenedores sin procesar en el almacén y los vuelve a cargar para ser procesados o descargados en la misma u otras posiciones del almacén.
- La comparativa entre el problema base y las situaciones extras nos permite observar que el planificador optimiza el número de movimientos por contenedor, es decir, solo 4 movimientos. Esto sucede cuando se agrega un camino adicional entre la fábrica 2 y al almacén.

Asignatura	Nombres
Razonamiento y Planificación Automática	David Toscano - Edison Giraldo Joel Orellana - Rubén Aponte

- Al adicionar más cantidad de vagones, el planificador a pesar de que realiza su tarea y consigue el objetivo, no necesariamente ocupa los espacios disponibles en los vagones, entre los movimientos. Esto sucede en parte porque las acciones no tienen un costo asociado que permita al planificador identificar si cargar 4 contenedores es mejor que cargar menos o más.
- El uso de cláusulas condicionales (*conditional-effects*) con *when* permite modificar el estado de un contenedor para pasar de no procesado a procesado y finalmente a eliminado, sin la necesidad de creación de nuevas acciones.
- Los mejores planes del planificador *Soup2018* tuvieron mejor resultado en tiempo y en costo que el planificador de *VSC*, pero *Soup2018* debe iterar varias veces generando planes menos óptimos en el proceso.
- El planificador *Soup2018* usa una heurística que reduce la expansión de nodos, frente a los nodos expandidos por *VSC* como se ve en la Figura 4.

ANEXOS

Como parte de este documento, se anexan los *scripts* (*.pddl) para las 3 situaciones analizadas y los planes encontrados tanto para *VSC* (*.htm) como para *Soup2018* (*.txt).

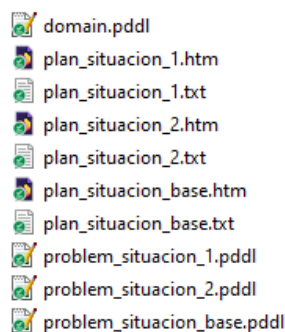


Figura 5. Anexos del documento.

BIBLIOGRAFÍA

- Seipp, J., & Röger, G. (2018). Fast downward stone soup 2018. *IPC2018–Classical Tracks*, 72-74.). Fast downward stone soup 2018. *IPC2018–Classical Tracks*, 72-74.
- PDDL Editor (n.d.). Retrieved January 28, 2023, from <http://editor.planning.domains/>