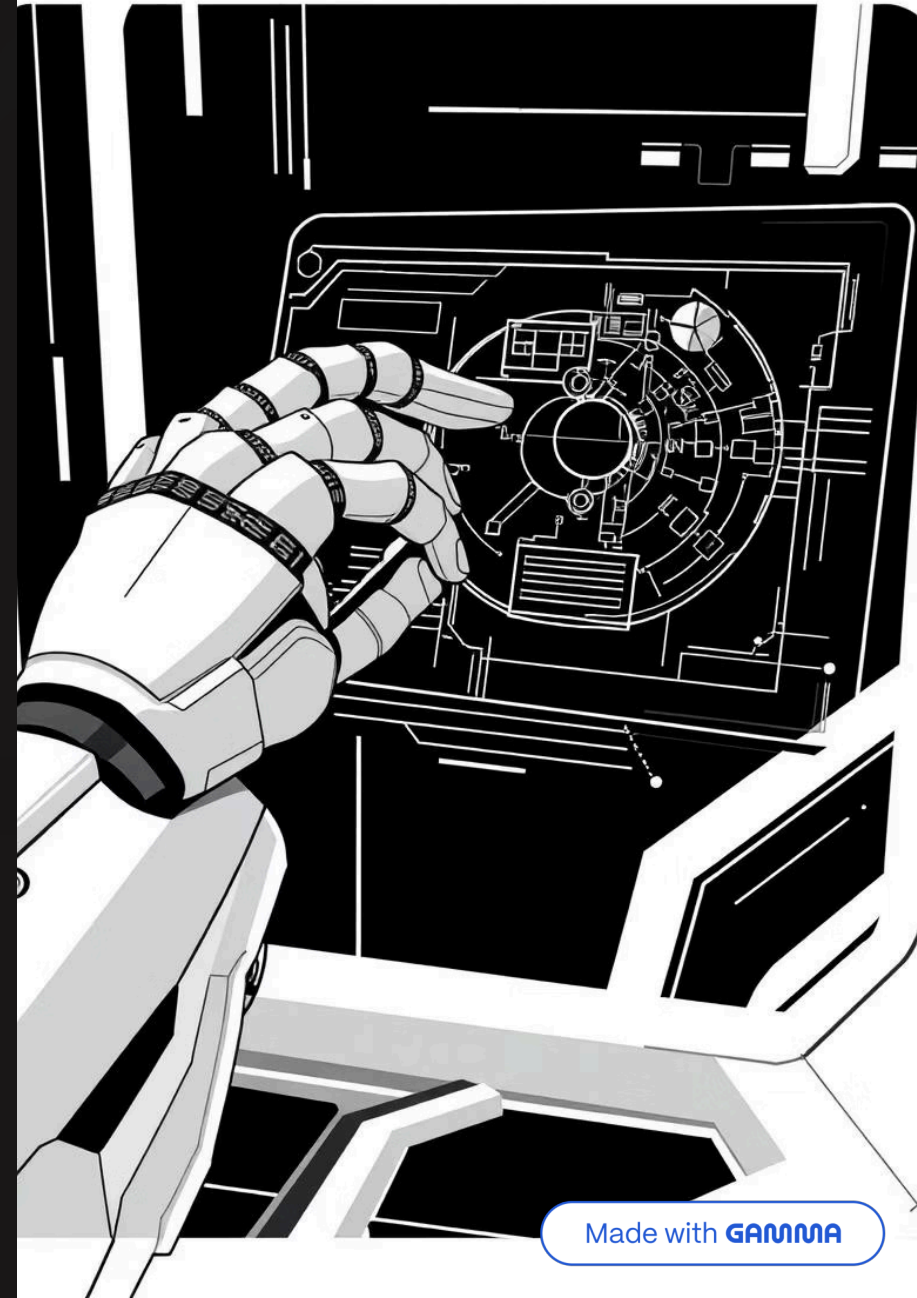


Trabajo Práctico N°6

18 de septiembre de 2025



Temas Clave del Trabajo Práctico 6

1

Modelado de Problemas

Espacios de estado.

2

Algoritmos de Planificación

Hacia adelante y hacia atrás.

3

Lenguaje STRIPS

Representación y solución.

4

GRAPHPLAN

Algoritmo de planificación.

5

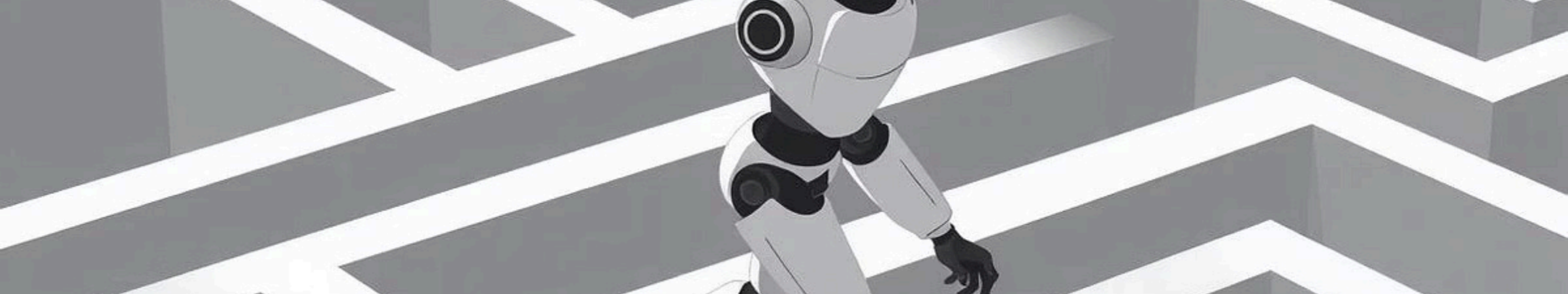
Restricciones

Tiempo y recursos.

6

Caminos Críticos

Tiempos de relajación.



Capítulo 2

Ejercicios Teóricos: Planificación

¿Cómo encuentra un planificador el mejor camino a un estado solución?

Un planificador aplica algoritmos de búsqueda en espacios de estados, explorando configuraciones del mundo (estados) y transiciones (acciones) para conectar el estado inicial con el objetivo. Se comporta como un agente que navega un grafo, donde cada nodo es un estado y cada arista una acción. La búsqueda puede ser ciega o informada, usando heurísticas para optimizar el camino. El objetivo es construir un plan óptimo evaluando costos, restricciones y el orden de las acciones antes de ejecutarlas.

Elementos de una Acción STRIPS



Precondiciones

Lo que debe cumplirse en el estado actual para que la acción sea aplicable.



Efectos

Lo que cambia en el estado cuando se ejecuta la acción (agregar o eliminar condiciones).

Tipos de Planificación

- **Hacia adelante (progresión):** Desde el estado inicial, se aplican acciones hasta el objetivo. Puede ser ineficiente al considerar muchas acciones irrelevantes.
- **Hacia atrás (regresión):** Desde el estado objetivo, se buscan acciones que pudieron haberlo generado. Más eficiente al concentrarse en acciones relevantes.
- **Parcialmente ordenada (POP):** Trabaja con sub-objetivos independientes y los combina, ofreciendo flexibilidad y evitando pasos innecesarios.

Ventajas y Desventajas: Planificación Hacia Adelante vs. Hacia Atrás

Criterio	Planificación Hacia Adelante (Progresión)	Planificación Hacia Atrás (Regresión)
Punto de Partida	Estado inicial	Estado objetivo
Ventajas	Sencilla de implementar y entender. Genera estados alcanzables. Imita la ejecución real.	Se centra en condiciones del objetivo. Evita estados irrelevantes. Más eficiente con objetivo bien definido.
Desventajas	Puede generar muchos estados irrelevantes. Ineficiente sin heurísticas. Riesgo de perderse en espacios grandes.	Difícil garantizar precondiciones alcanzables. Manejo complejo de condiciones negativas. Menos intuitiva.

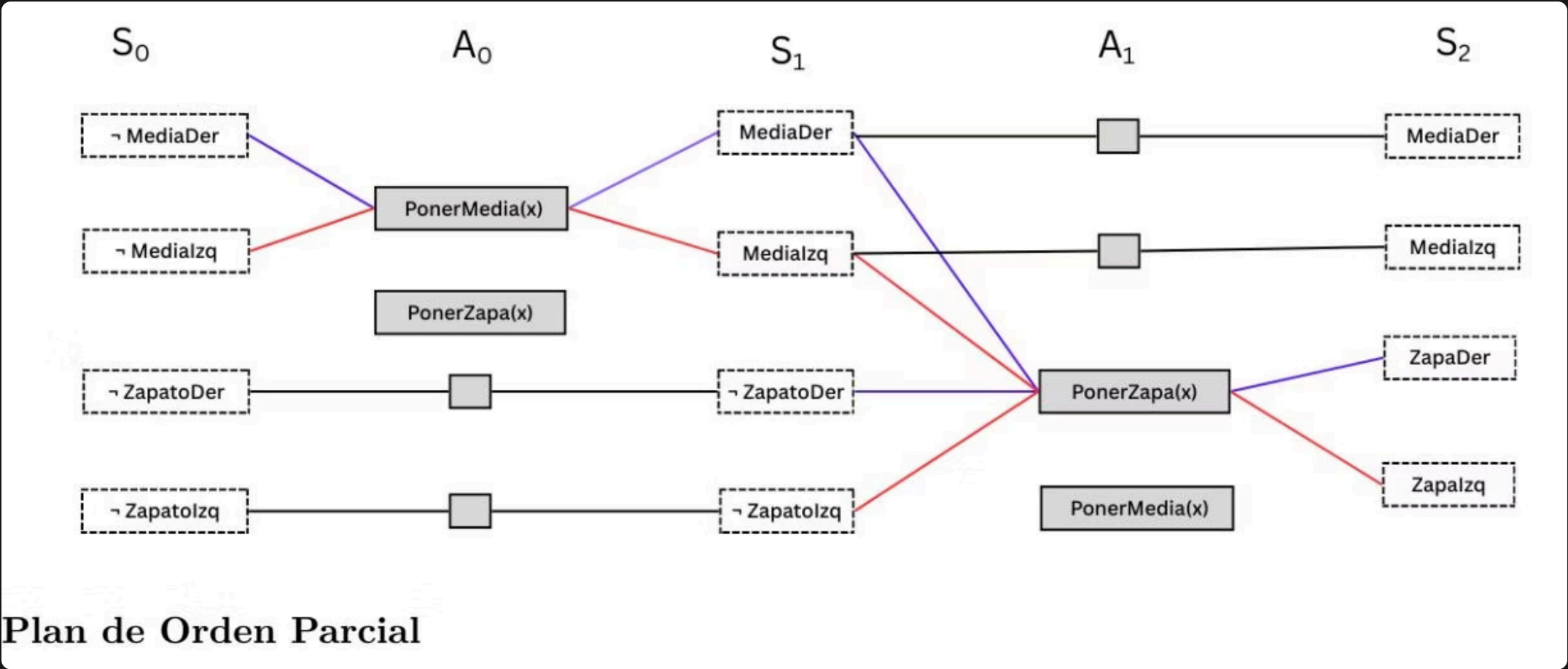
Problema: Ponerse Zapatos y Medias con GRAPHPLAN

Estado inicial: $\neg(\text{medias puestas}), \neg(\text{zapatos puestos}), \neg(\text{medias en pie}), \neg(\text{zapatos en pie})$

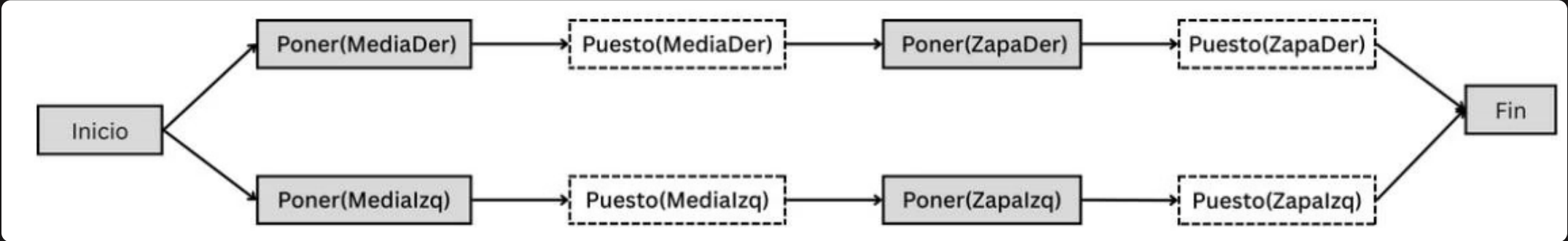
Objetivo: $(\text{medias puestas}) \wedge (\text{zapatos puestos})$

Acciones posibles: Poner medias, Poner zapatos, Quitar medias, Quitar zapatos.

Relaciones de Precedencia: Poner medias < Poner zapatos.



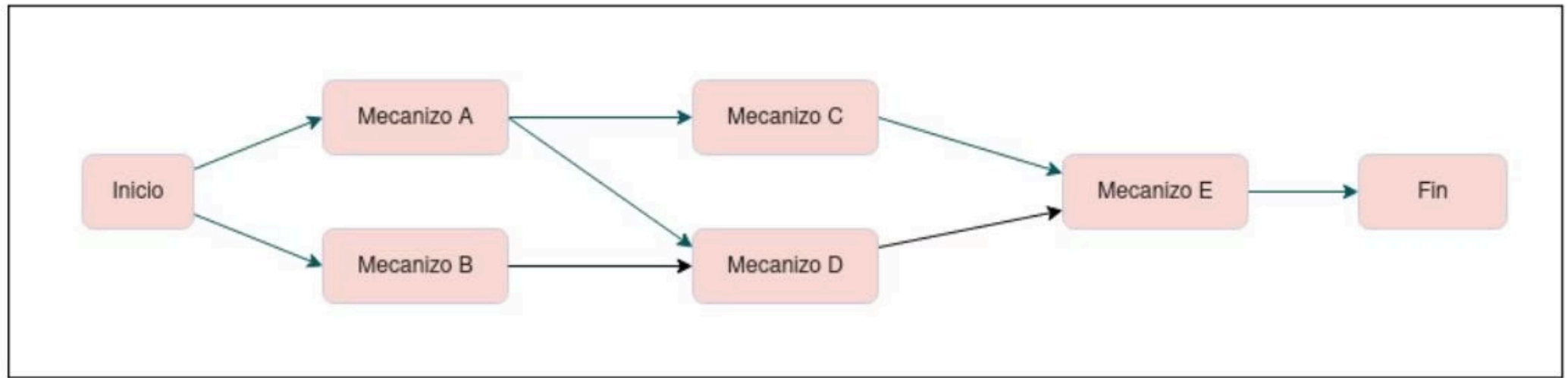
Plan de Orden Parcial



Existen 6 linealizaciones diferentes para el Plan de Orden Parcial.

Ensamblaje de Máquina: Camino Crítico y Tiempos de Relajación

El camino crítico es la secuencia de acciones más larga y restrictiva que determina la duración mínima del proyecto.



Camino Crítico:

Inicio → A → C → E → Fin

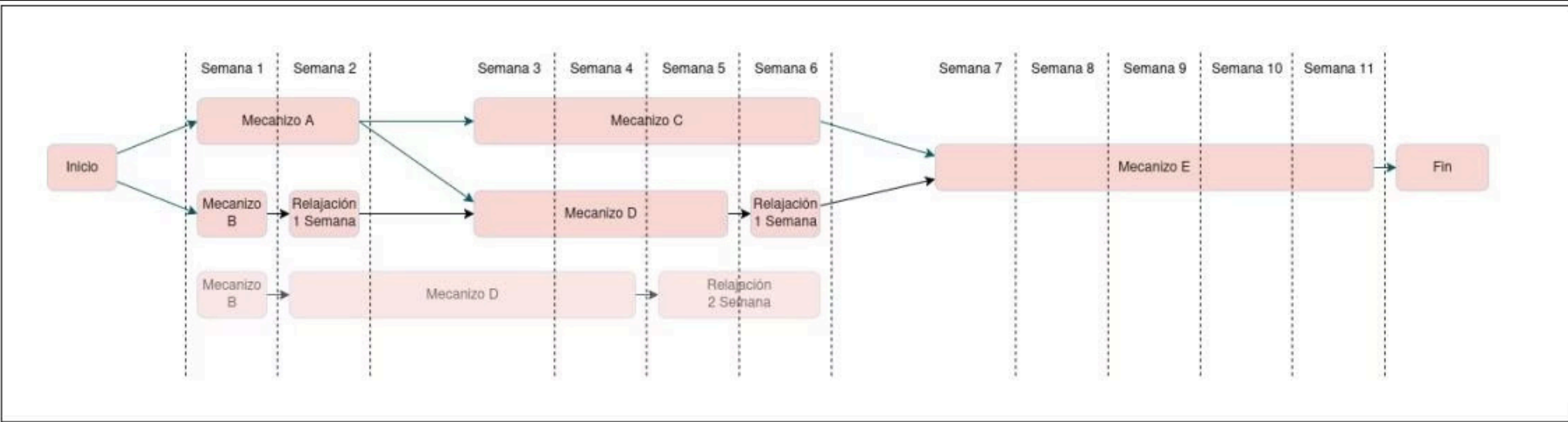
Duración total del proyecto: 11 semanas.

Tiempos de Relajación (Holguras):

- **Holgura total:** Tiempo que una tarea puede retrasarse sin afectar la duración total del proyecto.
- **Holgura libre:** Tiempo que una tarea puede retrasarse sin afectar la fecha de inicio de la siguiente tarea inmediata.

Las tareas en el camino crítico (A, C, E) tienen holgura cero.

Diagrama Temporal y Holguras



Tarea	Predecesoras	Duración	ES	EF	LS	LF	Holgura
A	—	2	0	2	0	2	0
B	—	1	0	1	1	2	1
C	A, B	4	2	6	2	6	0
D	B	3	1	4	3	6	2
E	C, D	5	6	11	6	11	0

Ejercicios de Implementación: Robot de Oficina

Dominio del Robot (STRIPS)

```
(define (domain office-robot)
  (:requirements :strips)
  (:predicates
    (place ?p)
    (thing ?x)
    (box ?b)
    (connected ?a ?b)
    (robot-at ?p)
    (at ?x ?p)
    (carrying-box ?b)
    (in-box ?x ?b)
  )

  (:action move
    :parameters (?from ?to)
    :precondition (and (robot-at ?from) (connected ?from ?to))
    :effect (and (robot-at ?to) (not (robot-at ?from))))

  (:action get-box
    :parameters (?b ?loc)
    :precondition (and (box ?b) (place ?loc) (at ?b ?loc) (robot-at ?loc))
    :effect (and (carrying-box ?b) (not (at ?b ?loc))))

  (:action load-into-box
    :parameters (?x ?loc ?b)
    :precondition (and (thing ?x) (place ?loc) (box ?b) (robot-at ?loc) (carrying-box ?b) (at ?x ?loc))
    :effect (and (in-box ?x ?b) (not (at ?x ?loc))))
)
```

Problema del Robot: Café y Carta

Estado Inicial y Objetivo

```
(define (problem get-coffee-and-mail)
  (:domain office-robot)
  (:objects office store mailbox box coffee mail)
  (:init
    (place office) (place store) (place mailbox)
    (box box) (thing coffee) (thing mail)
    (connected office store) (connected store office)
    (connected office mailbox) (connected mailbox office)
    (robot-at office) (at box office) (at coffee store) (at mail mailbox)
  )
  (:goal (and (robot-at office) (carrying-box box) (in-box coffee box) (in-box mail box)))
)
```

Plan de Acción del Robot:

- get-box box office
- move office mailbox
- load-into-box mail mailbox box
- move mailbox office
- move office store
- load-into-box coffee store box
- move store office

Plan de 7 pasos, costo 7.