

Creo función que reciba el Modo:

```
(:functions
  (tiempo-total-empleado) 4
  (distancia ?p1 ?p2 - posicion) 2
  → (modo_movimiento ?mode - NavMode) 1
  (velocidad-tarea ?t - tarea) 2
  (bateria-total-usada)
  (bateria)
  (umbral)
)
```

Defino en el problema esa función dos veces, una con N0 y otra con N1:

```
View
(:init
  (= (tiempo-total-empleado) 0)
  → (= (modo_movimiento N0) 1)
  → (= (modo_movimiento N1) 2)
  (= (velocidad-tarea fotografia1) 2)
)
```

Creo predicado que indique el modo del vehículo:

```
(:predicates
  (estoy_en ?uv - UV ?p - posicion) 2 1 1
  (conectado ?p1 ?p2 - posicion) 1
  (dockeado ?uv - UV) 1 1 1
  (moviendose ?uv - UV) 3 1 1
  → (modo_navegacion_uv ?uv - UV ?mode - NavMode) 1
)
```

Hasta aquí creo que voy bien y coincido contigo ya que es lo que estaba planteando, ahora empiezan los problemas jeje:

En la creación de la acción que cambie el anterior predicado, ¿No hay ninguna precondition clara no? ¿Así como que no sé si lo estaré haciendo bien, pero en los efectos no puedo hacer más que poner el predicado tal cual no, esto debería darle poder al planificador para introducir en el parámetro *mode* el que desee no?

```
(:durative-action cambiarModoNavegacion
  :parameters (?uv - UV ?mode - NavMode)
  :duration (= ?duration 1)
  :condition (and
    → (at start (moviendose ?uv))
  )
  :effect (and
    (at end (increase (tiempo-total-empleado) 1))
    → (at start (modo_navegacion_uv ?uv ?mode))
  )
)
```

→ con eso se consigue el cambio?

Ahora viene la parte que más confuso estoy:

Fran: "Al cambiar eso estás forzando que cuando se mueve utilice ese modo ya que le pones en precondition que tenga ese modo"

No veo claro cómo forzar a un cierto modo concreto, todo este código actualmente funciona y realiza el movimiento y la foto, pero simplemente aplica el modo N0 sin realizar en ningún momento la acción **cambiarModoNavegación()**.

```
(:durative-action mover
:parameters (?uv - UV ?p1 ?p2 - posicion ?mode - NavMode)
:duration (= ?duration (* (distancia ?p1 ?p2) (modo_movimiento ?mode)))
:condition (and
  (at start (estoy_en ?uv ?p1))
  (at start (not (= ?p1 ?p2)))
  (at start (conectado ?p1 ?p2))
  (at start (moviendose ?uv))
)
:effect (and
  (at end (estoy_en ?uv ?p2))
  (at start (not (estoy_en ?uv ?p1)))
  (at end (increase (tiempo-total-empleado) (distancia ?p1 ?p2)))
)
)
```

A lo mejor es porque en el problema tengo que poner el predicado deseado y/o inicial, en plan que quiero que inicialmente el robot tal tenga el modo N1, por tanto, el planificador tendría que ser inteligente y cambiarlo a N0 puesto que es más rápido. Pero de esto último lo ignora completamente, si por ejemplo añado un tercer modo N2 más rápido que N0 y N1, en la planificación sin hacer ninguna acción de cambiar el modo del vehículo como que directamente le mete en el argumento de la función mover el nuevo modo N2 que es más rápido que los otros dos.

Es decir, parece que no estoy forzando, probablemente la clave de tu frase es:

Fran: "ya que le pones en precondition que tenga ese modo"

Vale, creo que he visto la luz en esa frase anterior:

Habría que hacer algo así:

```
(:durative-action mover
  :parameters (?uv - UV ?p1 ?p2 - posicion ?mode - NavMode)
  :duration (= ?duration (* (distancia ?p1 ?p2) (modo_movimiento ?mode)))
  :condition (and
    (at start (estoy_en ?uv ?p1))
    (at start (not (= ?p1 ?p2)))
    (at start (conectado ?p1 ?p2))
    (at start (moviendose ?uv))
    (at start (modo_navegacion_uv ?uv ?mode))
  )
  :effect (and
    (at end (estoy_en ?uv ?p2))
    (at start (not (estoy_en ?uv ?p1)))
    (at end (increase (tiempo-total-empleado) (distancia ?p1 ?p2)))
  )
)
```

De esa manera estoy haciendo justo lo que tu dices, es decir, forzar que el modo del vehículo corresponda con el que recibo por parámetro:



Clap clap clap. Bárbaro!!