Análisis y diseño de software

dit
UPM

Trenes

José A. Mañas

http://jungla.dit.upm.es/~pepe/doc/adsw/index.html

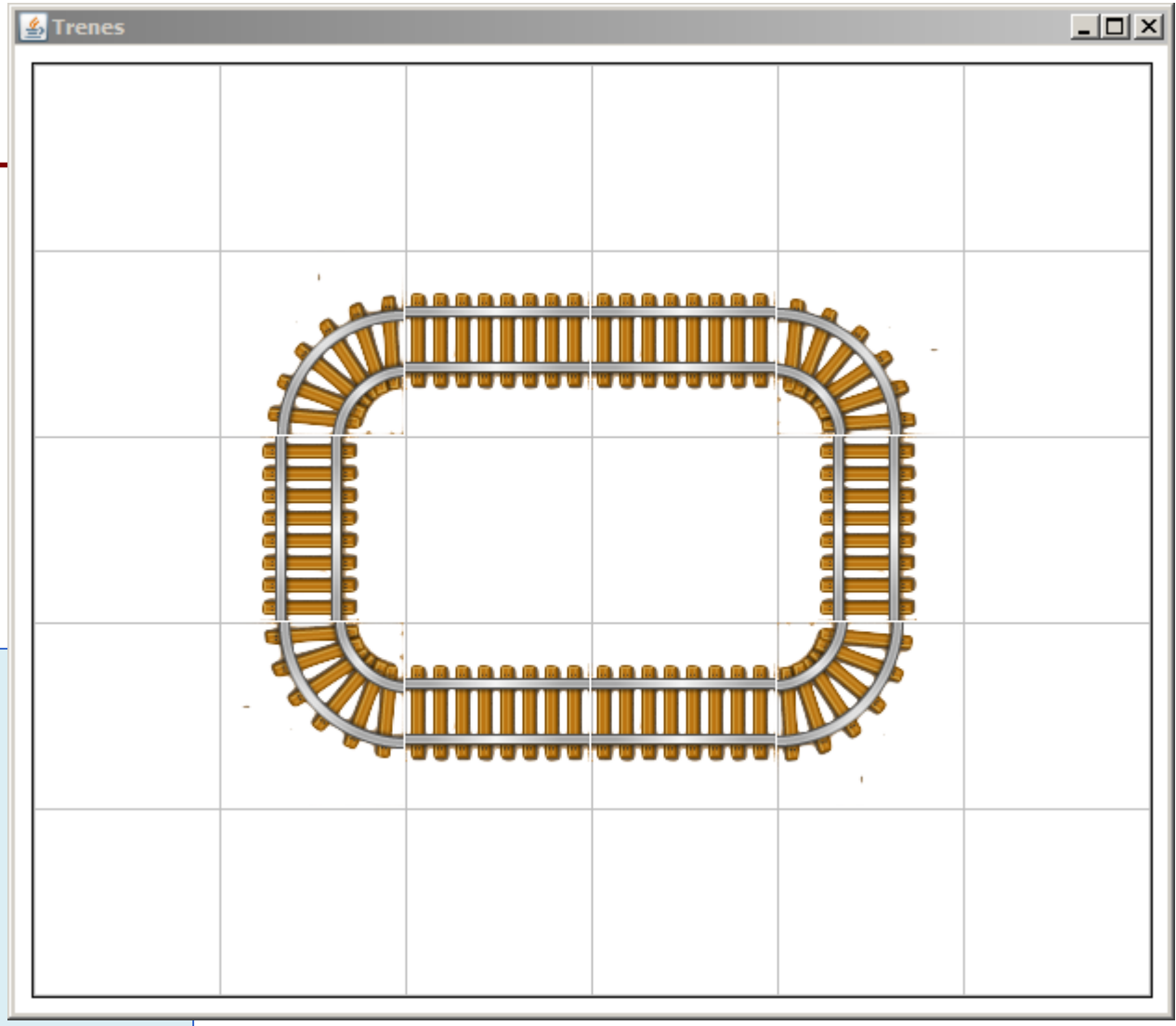
21.3.2017

ETSIT UPM

objetivos

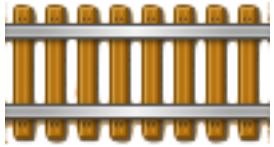
- terreno
  - tramos de vías en una cuadrícula
  - cada tramo tiene hasta 4 enlaces: N, S, E, W
- trenes
- semáforos
- monitores
- ejemplos

ejemplo




```
String[] mapa = {
            "",
            "  - se  h h ws -",
            "  -  v  - -  v -",
            "  - ne  h h nw -",
            ""
      };
Terreno terreno = new Terreno(mapa);
terreno.setVisible();
```

tramos sencillos

h 

v 

en
ne


he 

ve 

es
se


c
+

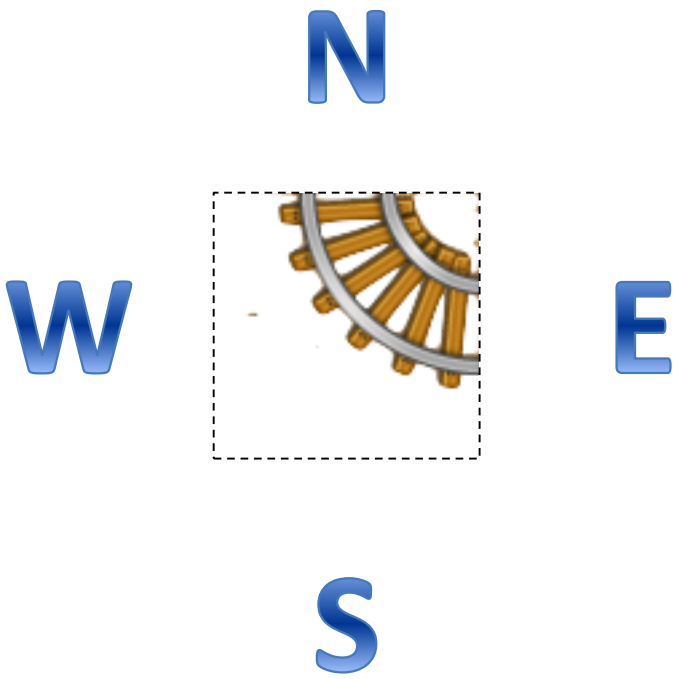

wn
nw


ws
sw


curvas identificadas por sus enlaces

2 formas de nombrarlo:

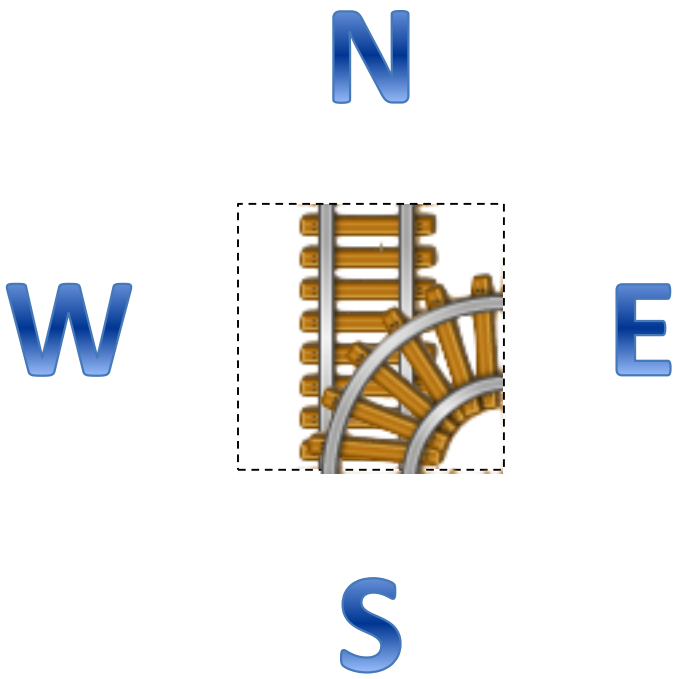
- **en**
- **ne**



tramos compuestos

2 formas de nombrarlo:

- **es_n**
- **se_n**

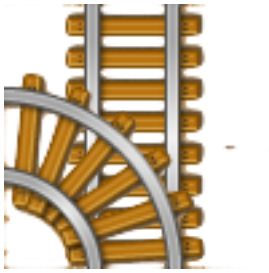


curva sur-este con enlace al norte

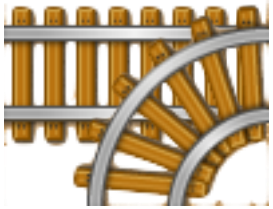
tramos compuestos

es_n
se_n

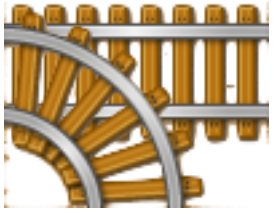
ws_n
sw_n



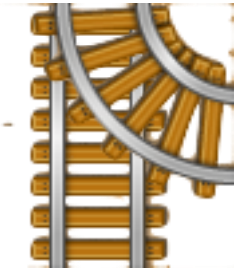
es_w
sw_w



ws_e
sw_e



en_s
ne_s



wn_s
nw_s

en_w
ne_w

wn_e
nw_e

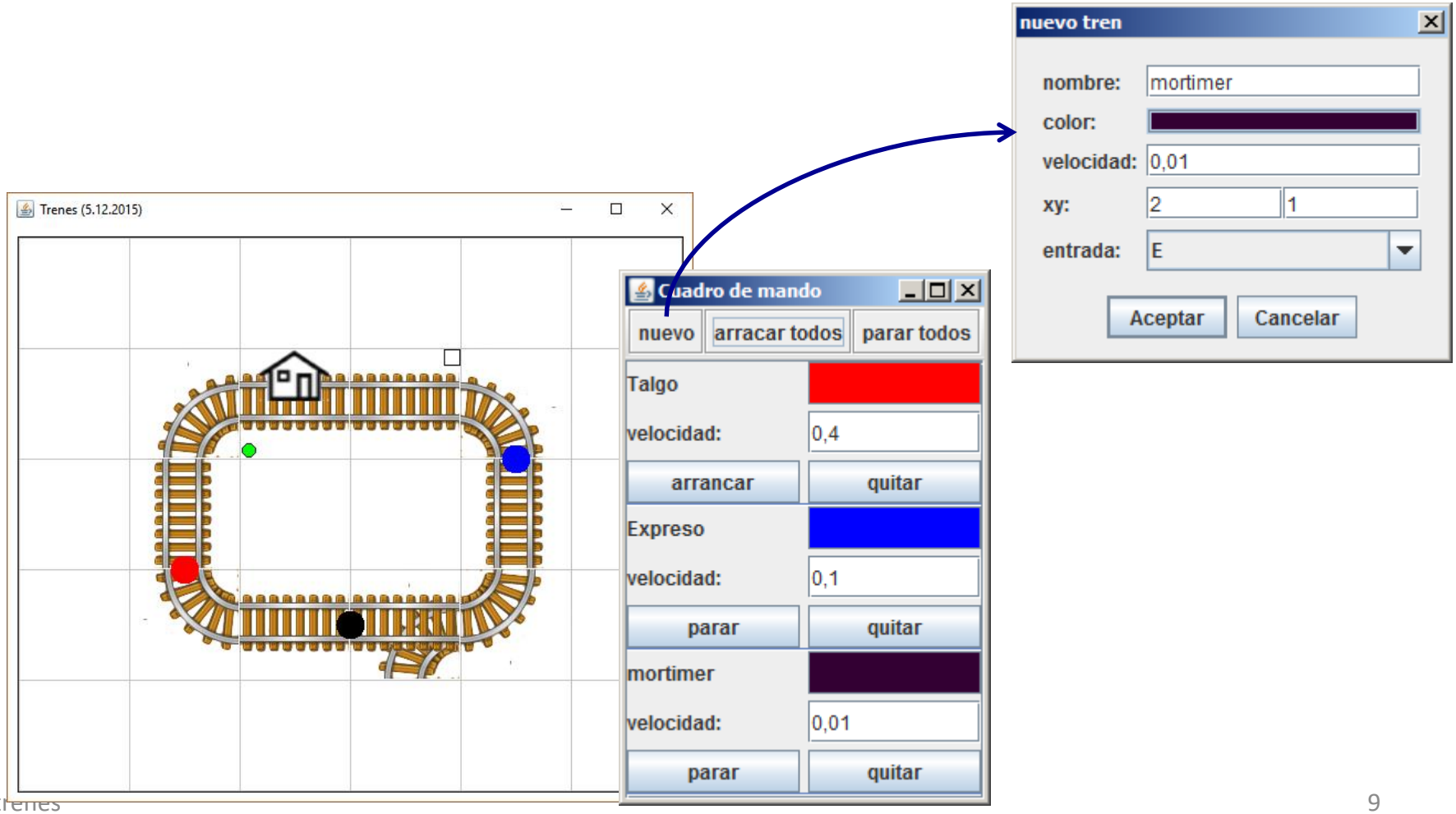
trenes

- se pueden crear y posicionar por programa

```
final Tramo estacion1 = terreno.get(2, 3);
final Tramo estacion2 = terreno.get(3, 3);

Semaphore semaforo = new Semaphore(1);
terreno.ponSemaforoEntrada(estacion1, W, semaforo);
terreno.ponSemaforoSalida(estacion2, E, semaforo);

Tren tren1 = new Tren("Talgo", Color.RED);
tren1.setVelocidad(0.4);
terreno.ponTren(1, 2, S, tren1);

Tren tren2 = new Tren("Expreso", Color.BLUE);
tren2.setVelocidad(0.1);
terreno.ponTren(4, 2, N, tren2);
```

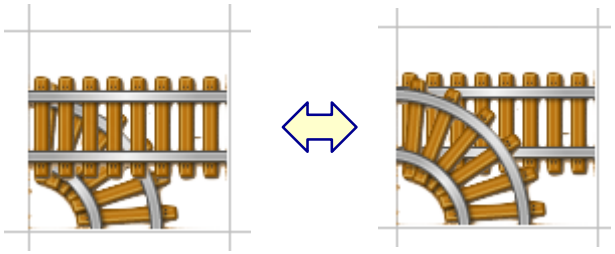
trenes

- se pueden crear y controlar interactivamente



interactividad

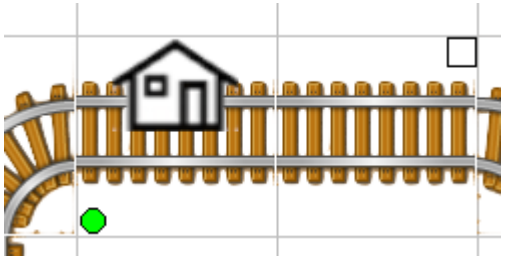
- tramos
  - haga clic para cambiar agujas



- trenes
  - arrancar y parar
  - cambiar la velocidad (siempre $v \geq 0$)
  - cambiar el color
  - quitar el tren
    - ojo con los semáforos y monitores

semáforos

- Se instalan en los enlaces de los tramos
- Aparecen pintados
  - según se entra a la derecha
  - verde si están abiertos
  - rojo si están cerrados



```
Terreno terreno = new Terreno(mapa);
    Tramo estacion1 = terreno.get(2, 3);
    Tramo estacion2 = terreno.get(3, 3);

    Semaphore semaforo = new Semaphore(1);
    terreno.ponSemaforoEntrada(estacion1, W, semaforo);
    terreno.ponSemaforoSalida(estacion2, E, semaforo);
```

semáforos

- Dinámica
  - los crea el usuario con los permisos pertinentes

    Semaphore semaforo = new Semaphore(1);
  - cuando entra un tren, adquiere 1 permiso

    semaforo.acquire();
  - cuando sale un tren, devuelve 1 permiso

    semaforo.release();

monitores

```
public abstract class Monitor {

   /**
    * Metodo para el usuario. Se llama cuando un tren quiere entrar a un tramo monitorizado.
    * El usuario debe programar lo que haya que hacer.
    *
    * @param tag     pista para el monitor.
    * @param tren    tren que entra.
    * @param tramo   tramo al que va a entrar.
    * @param entrada enlace por el que entra.
    */
   public abstract void entro(int tag, Tren tren, Tramo tramo, Enlace entrada);

   ...
```

monitores

```
...

   /**
    * Metodo para el usuario. Se llama cuando un tren sale de un tramo monitorizado.
    * El usuario debe programar lo que haya que hacer.
    *
    * @param tag    pista para el monitor.
    * @param tren   tren que sale.
    * @param tramo  tramo del que va a salir.
    * @param salida enlace por el que sale.
    */
   public abstract void salgo(int tag, Tren tren, Tramo tramo, Enlace salida);
}
```

monitores

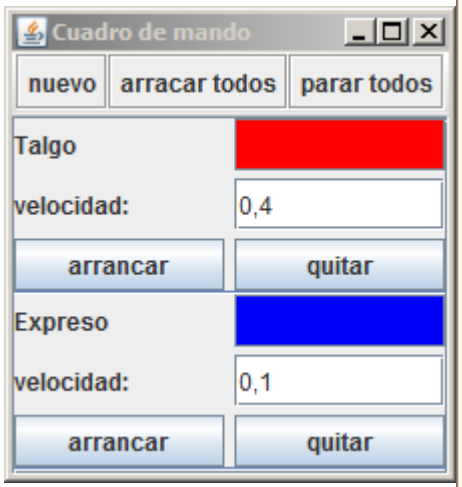
- Se instalan en el terreno
  - monitorizando enlaces

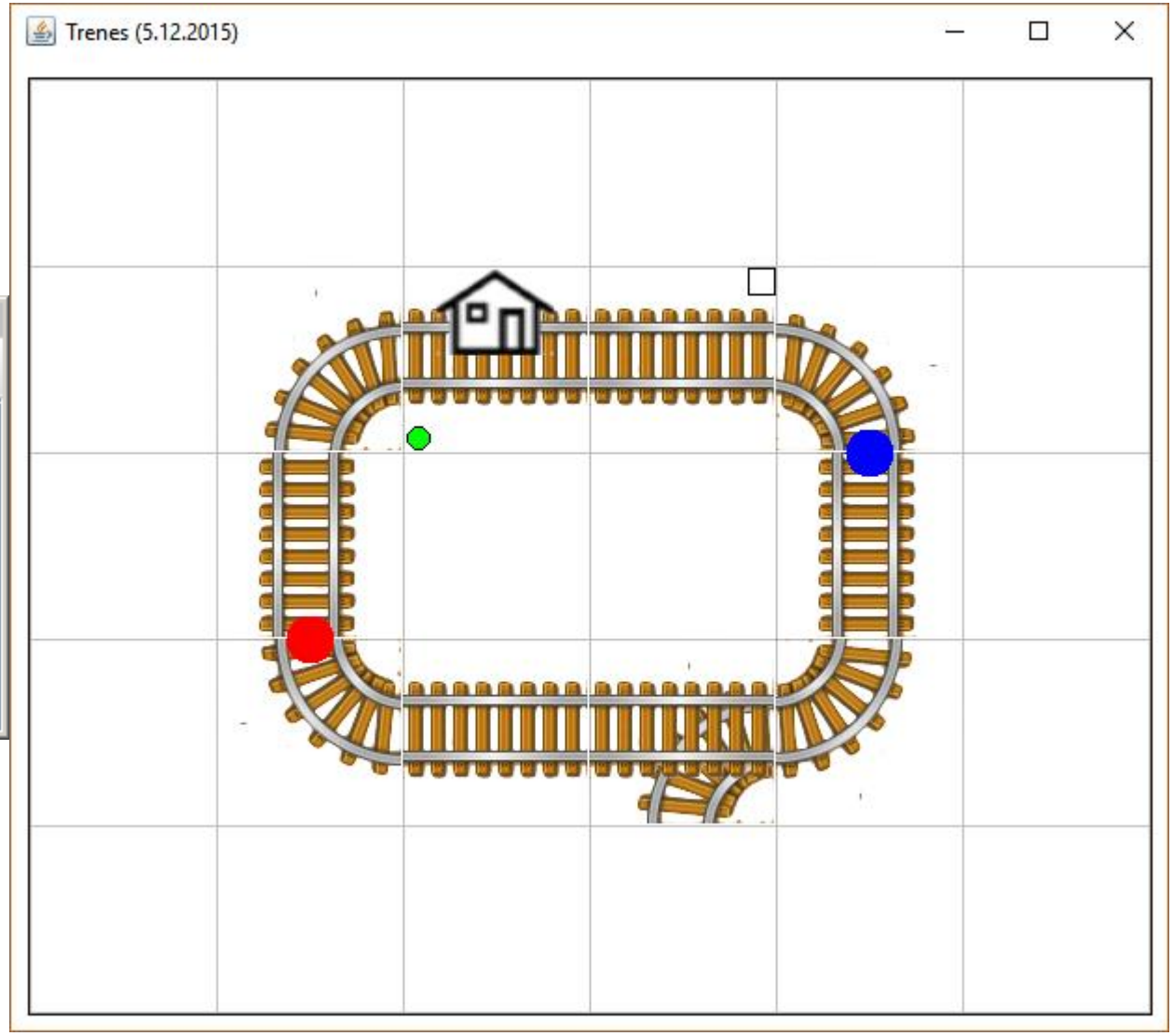
```
Monitor tunel = new MonitorTunel();
terreno.ponMonitorEntrada(cambio23, Enlace.N, tunel, 1);
terreno.ponMonitorEntrada(cambio23, Enlace.W, tunel, 1);
terreno.ponMonitorSalida(cambio43, Enlace.N, tunel, 1);
terreno.ponMonitorSalida(cambio43, Enlace.E, tunel, 1);

terreno.ponMonitorEntrada(cambio43, Enlace.N, tunel, 2);
terreno.ponMonitorEntrada(cambio43, Enlace.E, tunel, 2);
terreno.ponMonitorSalida(cambio23, Enlace.N, tunel, 2);
terreno.ponMonitorSalida(cambio23, Enlace.W, tunel, 2);
```

  - se ponen en un tramo / enlace
  - se marcan con un tag que se le pasará al método de tratamiento

escenario 1

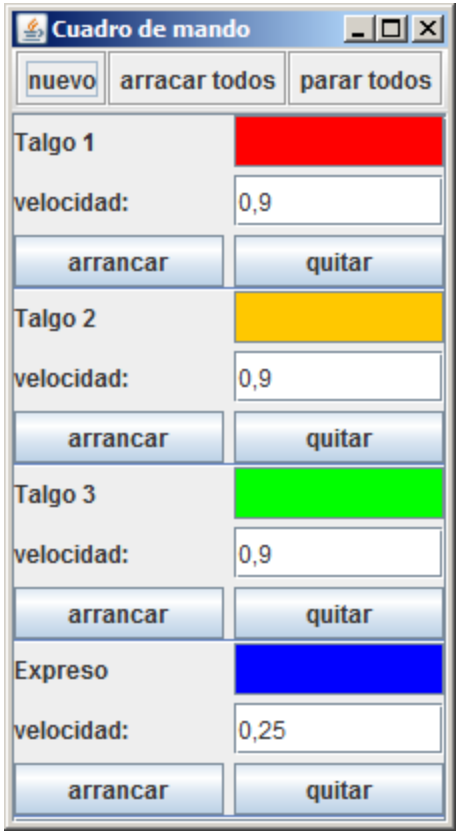
- trenes
- estaciones

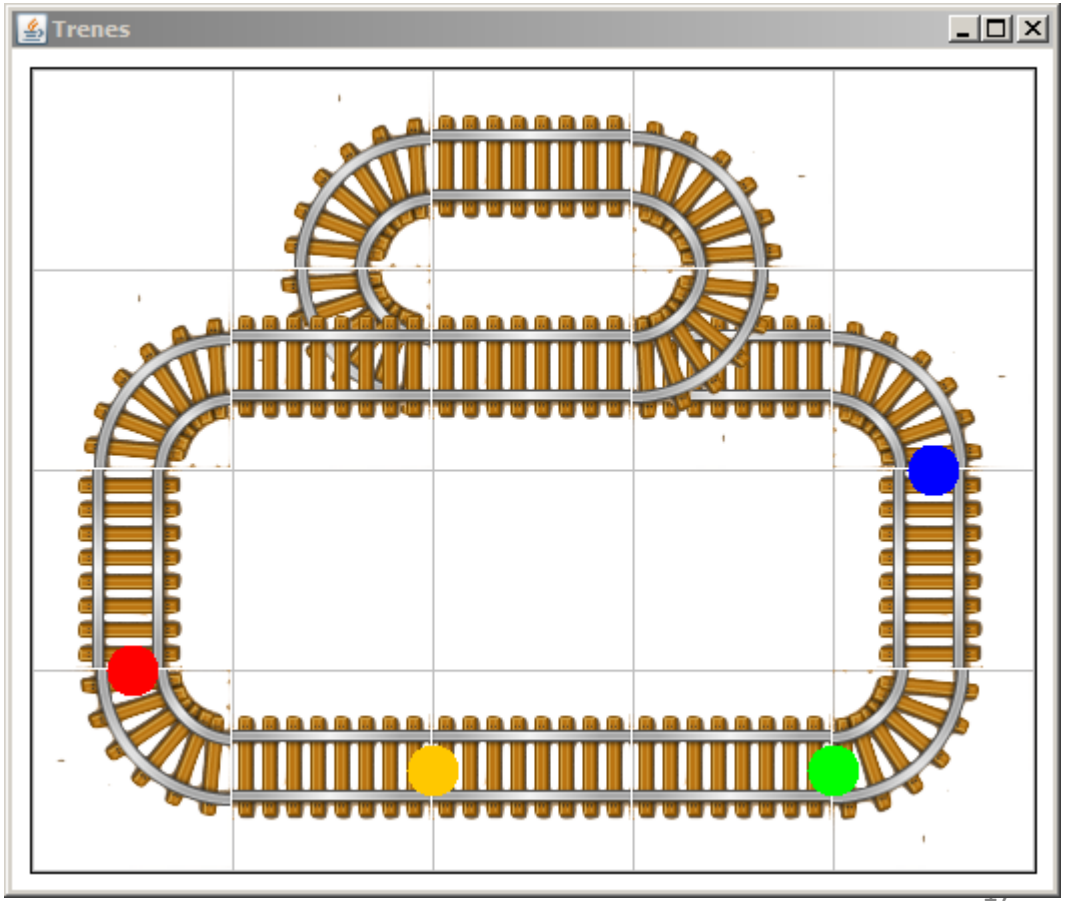




escenario 2

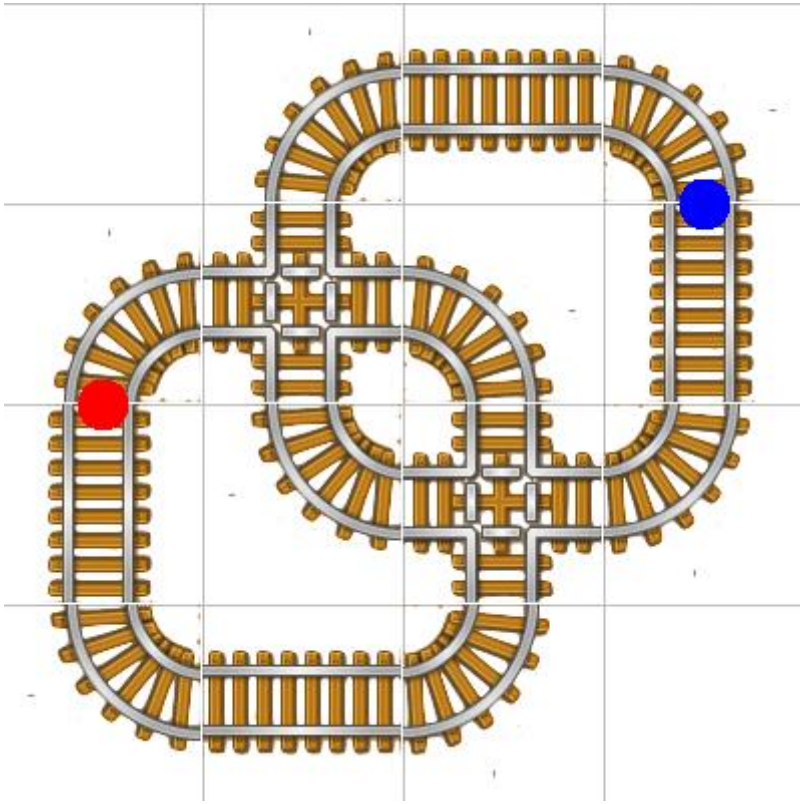
- tramos
  - simples
  - cambios

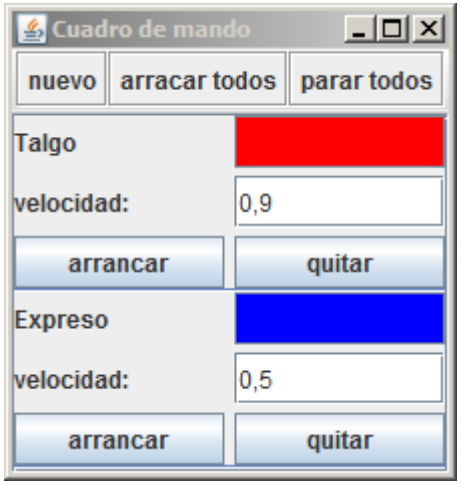




escenario 3

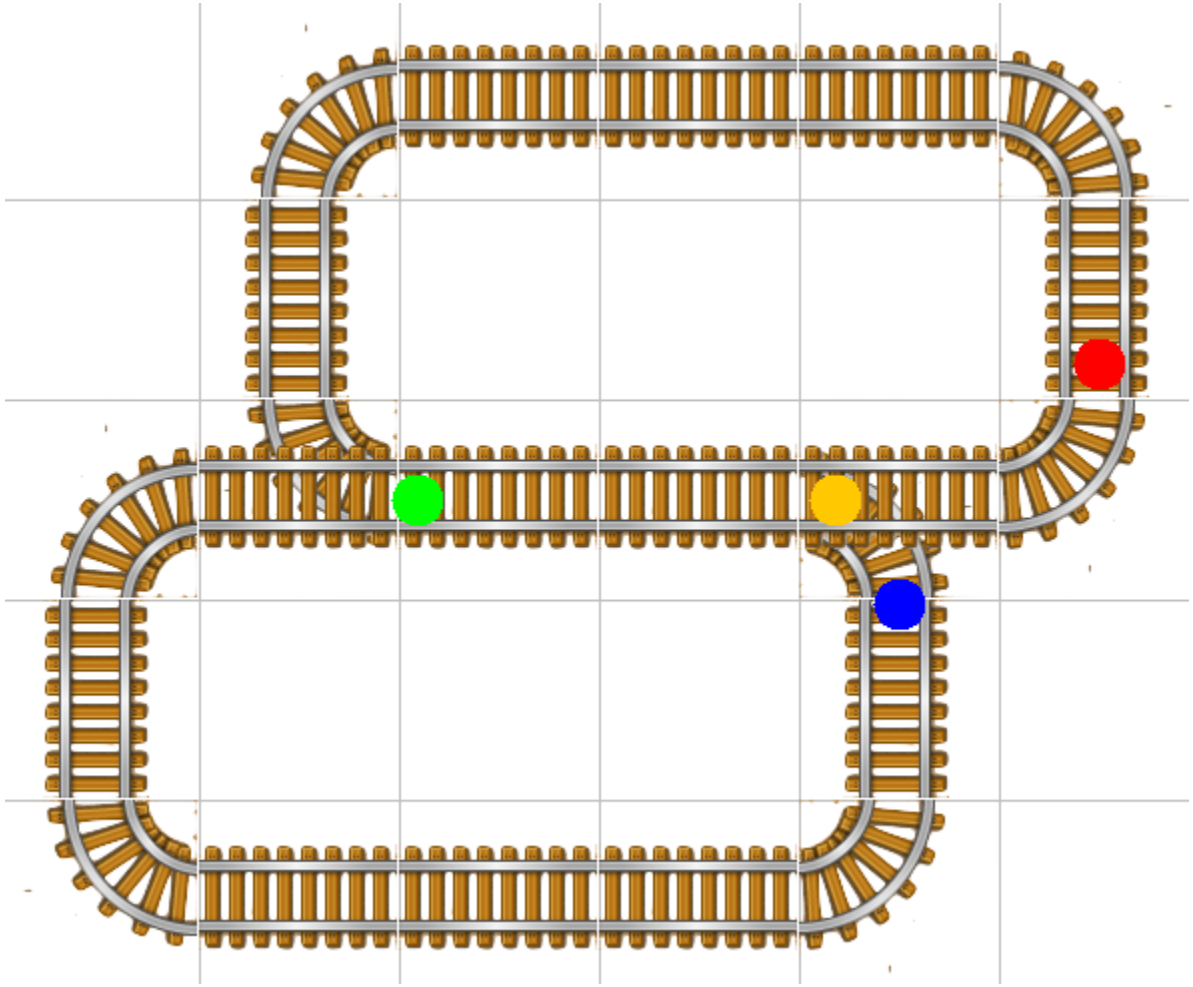
- trenes
- cruces





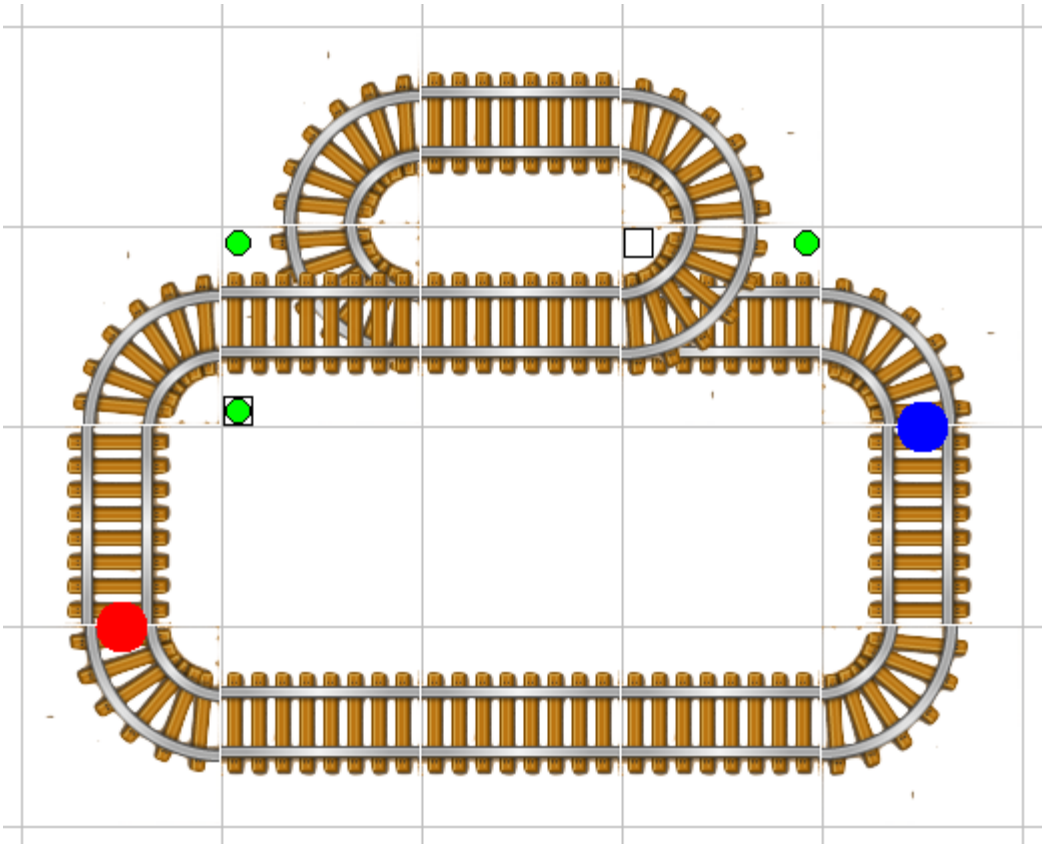
escenario 4

- trenes
- monitores
  (no se pintan)



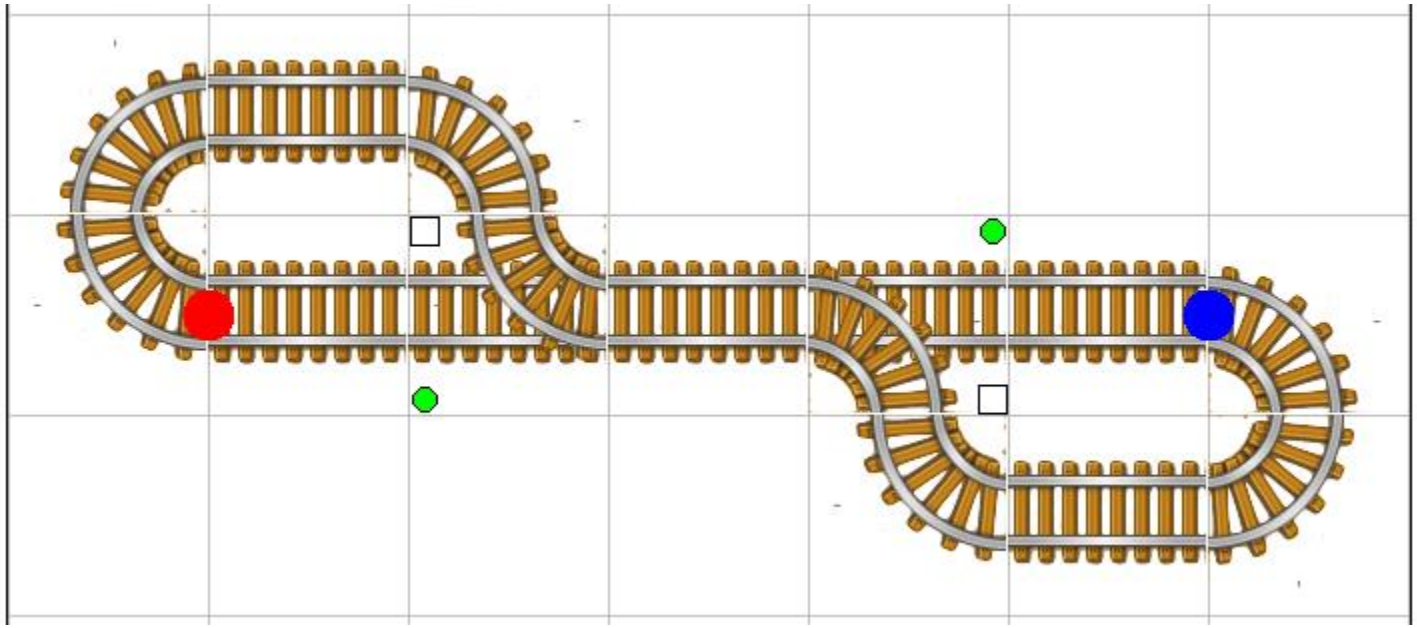
escenario 5

- trenes
- semáforos



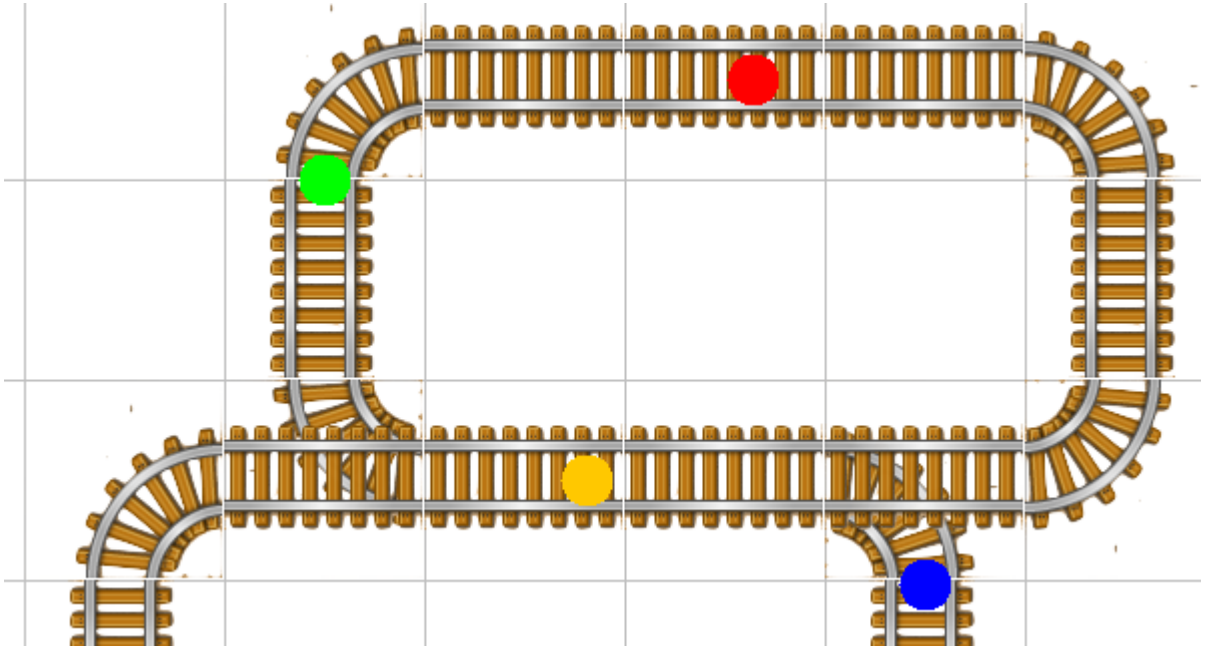
escenario 6

- trenes
- semáforos

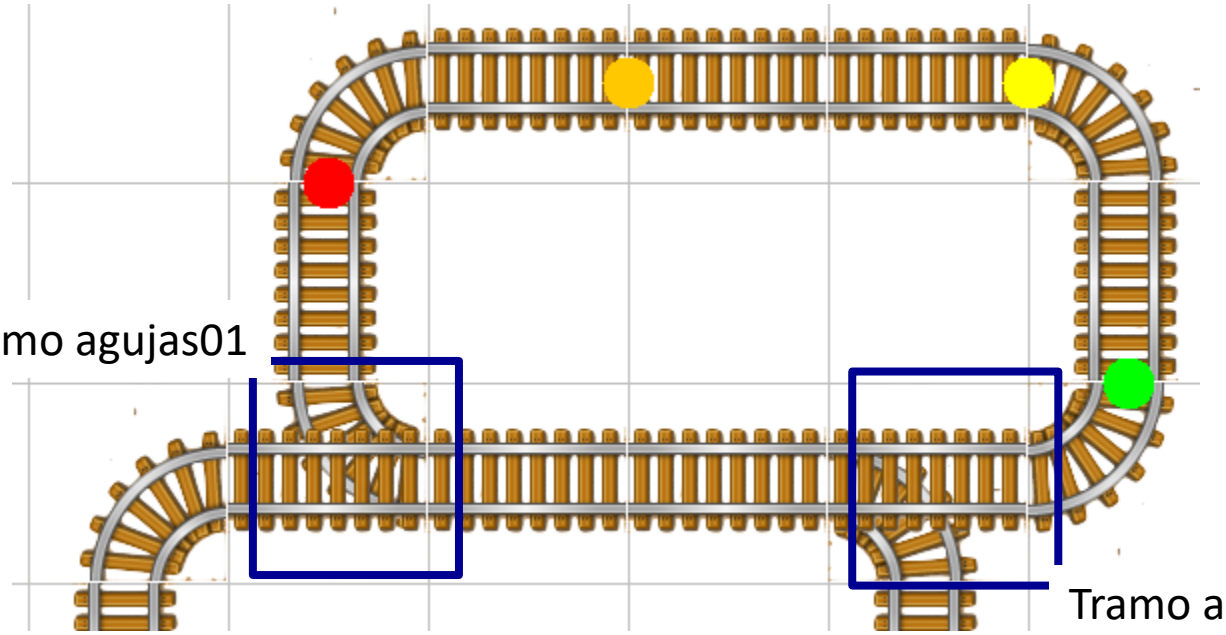


ejercicio 1

- escenario: un túnel con 2 entradas
  - si tu pasas, yo espero



ejercicio 1




```
// tag 1: direccion agujas01 -> agujas02
terreno.ponMonitorEntrada(agujas01, Enlace.N, tunel, 1);
terreno.ponMonitorEntrada(agujas01, Enlace.W, tunel, 1);
terreno.ponMonitorSalida(agujas02, Enlace.S, tunel, 1);
terreno.ponMonitorSalida(agujas02, Enlace.E, tunel, 1);

// tag 2: direccion agujas02 -> agujas01
terreno.ponMonitorEntrada(agujas02, Enlace.S, tunel, 2);
terreno.ponMonitorEntrada(agujas02, Enlace.E, tunel, 2);
terreno.ponMonitorSalida(agujas01, Enlace.N, tunel, 2);
terreno.ponMonitorSalida(agujas01, Enlace.W, tunel, 2);
```

Ejercicio 1

- Programar MonitorTunel

```
public class MonitorTunel
      extends Monitor {

   // la thread se detiene hasta que pueda entrar
   public void entro(int tag, Tren tren, Tramo tramo, Enlace entrada) {
   }

   // la thread sale
   public void salgo(int tag, Tren tren, Tramo tramo, Enlace salida) {
   }

}
```

ejercicio 1.1

FÁCIL

- características
  - sólo pasa un tren en cada momento

ejercicio 1.2

MEDIO

- características
  - sólo pasa un tren en cada momento
  - reparte equitativamente los derechos de paso
  - o sea, si sale un tren por E1, tiene preferencia para entrar un tren esperando en E1
    - equitativo *(fair)*

ejercicio 1.3

MEDIO

- características
  - pueden pasar varios trenes en la misma dirección

ejercicio 1.4

MEDIO

- características
  - pueden pasar varios trenes en la misma dirección
  - pero si hay trenes esperando en dirección contraria, tienen prioridad
    - equitativo *(fair)*