Contexto:

Empezaremos describiendo un poco el concepto de la práctica y su finalidad. Para esta práctica debemos modelar un sistema de trenes, en el que tendremos las respetivas líneas, con sus vías y sus trenes. Cada par de vías compondrán un segmento que será la unión entre dos estaciones. Más tarde deberemos añadir a nuestro modelo el transcurso del tiempo, y para ello deberemos crear una clase reloj que será representada además por una clase abstracta llamada ActiveObject, de esta manera podremos hacer que los trenes se muevan automáticamente con el paso del tiempo.

Diagrama de clases en los diferentes entornos.

Escala de tiempo

Descripción generada automáticamente con confianza media

(Use)

Diagrama

Descripción generada automáticamente

(Visual Paradigm)

Diagrama

Descripción generada automáticamente

(Papyrus)

Cabe mencionar que en estos diagramas ya viene implementado el tiempo, es de mencionar ya que esta es una tarea de uno de los apartados de la práctica.

Clases:

Ahora vamos a seguir describiendo las diferentes clases de nuestro modelo junto con sus atributos y operaciones.

1. Tren: es una clase que representará a un tren, que irá desplazándose de estación a estación y creando nuevas relaciones con estas para representarlo. Sus atributos son
   1. Dirección: nos indica la dirección en la que irá el tren, cuando este llegue al final de la línea su dirección se invertirá.
   2. Time: es un atributo de tipo Integer que nos mostrará el tiempo que lleva moviéndose, o, mejor dicho, “trabajando”.
   3. isMoving: es un atributo de tipo boolean que nos indica si el tren está moviéndose o no.
   4. isOutOfOrder: este atributo de tipo boolean nos indica si el tren esta fuera de servicio. Este se activará cuando nuestro tren se averíe.
   5. move(): operación utilizada para mover el tren de una posición a otra.
   6. stop(): operación utilizada para parar el tren, utilizada para que, cuando llegue a una estación, se pare un minuto.
   7. checkViaEnLinea(v:Via): operación utilizada para comprobar que la siguiente vía pertenezca a la línea a la que pertenece el tren y no acceda a una vía que no pertenezca a dicha línea.
   8. tick(): operación utilizada al heredar de la clase abstracta del reloj. Es utilizada para aumentar los ticks de reloj o segundos en nuestro modelo de tiempo.
   9. irAEstacion(): operación utilizada para indicar la siguiente vía que debe coger el tren.
   10. esperaEstacion(): operación utilizada para indicar al tren que debe esperar 1 minuto en la estación.
   11. reparar(): operación utilizada para reparar el tren .
   12. averiar(): operación utilizada para averiar el tren.
2. Clock: es una clase utilizada para modelar el tiempo, en nuestra implementación cada tick será un segundo, por lo que toda nuestra implementación de instantes y tiempos se basan en minuto, por lo que para cada minuto hacemos 60 ticks. Sus atributos son:
   1. NOW: es un atributo de tipo Integer que nos indica el momento de tiempo en el que estamos.
   2. resolution: es un atributo de tipo Integer que indica en cuantas unidades aumenta cada tick.
   3. tick(): operación utilizada para aumentar los segundos/ticks del tiempo.
   4. run(n : Integer): operación que ejecutará, con un for, una cierta ejecución de ticks que se le pase por parámetro.
3. ActiveObject: clase abstracta de la clase Clock.
4. Estación: clase que representa las estaciones, esta tendrá un único atributo que además será derivado y nos indicará cuando una estación es el final de una linea.
5. Vía: clase que representa las vías. Ésta tendrá también un único atributo derivado que nos dirá cuando la vía esté ocupada y cuando no.
6. Segmento: clase que representa el segmento que hay entre dos estaciones, esta estará compuesta por dos vías.
7. Línea: clase que modela la línea de tren.

Relaciones:

Ahora vamos a exponer nuestras relaciones y explicar que representan y sus multiplicidades.

1. Time: es una relación que relaciona una clase abstracta con la clase. Un reloj puede tener varios objetos activos. Y varios objetos activos pueden pertenecer a un solo reloj.
2. Tren-ActiveObject: esta relación de herencia solo nos indica que el tren es un objeto activo y por tanto un usuario del tiempo.
3. trenEstacion: esta relación representa cuando un tren está en una estación. A medida que el tren vaya moviéndose ira creando y borrando estas relaciones con determinadas instancias. Según nos indica el enunciado puede haber un número indefinido de trenes en una estación, pero un tren solo puede pertenecer en cada instante a una sola estación.
4. Via-Estacion: en este caso hay dos relaciones entre estas dos clases.
   1. estacionViaIda: en este caso esta relación nos indica el sentido que tiene la vía para esa estación, ya que entre dos estaciones habrá dos vías, una para cada sentido.
   2. estacionViaVuelta: de nuevo, esta relación nos indica el sentido que tiene dicha vía para dicha estación.

Como una estación puede ser un nodo entre varias líneas y vías podemos decir que una estación puede tener de una a muchas vías de cada sentido. Pero solo una vía puede pertenecer a una estación.

1. trenVia: esta relación es utilizada para indicarnos cuando un tren está en una vía. Un tren puede estar o no en una vía. Y una vía puede tener o no un tren.
2. viasDelSegmento: como ya hemos dicho antes un segmento es aquello que une dos estaciones entre sí, dicho segmento se compone de dos y solo dos vías.
3. segmentoPerteneceALinea: esta relación es utilizada para saber que segmentos tiene una línea. Cada línea puede tener uno o varios segmentos y viceversa, un mismo segmento puede pertenecer a una sola línea o a varias.
4. trenLinea: con esta relación sabemos a qué línea pertenece cada tren. Un tren puede pertenecer o no a una sola línea, es decir, un tren no puede pertenecer a más vías. Y una línea puede disponer de varios trenes.

Por último, tenemos ciertas observaciones que debemos tener en cuenta.

* Explicar que las líneas son independientes de los trenes, por lo que no tiene sentido hacer composiciones entre líneas y trenes. De hecho, podemos tener 0 trenes en una línea en algunos casos, y los trenes pueden cambiar de línea, aunque en su funcionamiento siempre siguen una.
* Líneas y segmento son agregación porque un segmento puede pertenecer a varias líneas
* Una estación se conecta a otra mediante unas dos vías, las cuales forman un segmento. De esta forma, como no hay estaciones aisladas, una estación siempre tiene dos vías y un segmento. Siempre habrá un número par de vías en una estación.
* No tiene sentido una línea sin segmento ni un segmento que no pertenezca a una línea.
* Un segmento puede pertenecer a más de una línea, y una línea está formada por segmentos.
* En un momento dado un tren está en una estación o en una vía.
* Debido a las multiplicidades, dos trenes no van a estar en la misma vía.

Ahora continuaremos con el diagrama de objetos que utilizaremos para realizar ejecuciones en nuestra práctica.

Diagrama

Descripción generada automáticamente

(Use)

Diagrama

Descripción generada automáticamente

(Papyrus)

Diagrama

Descripción generada automáticamente

(Visual paradigm)

En esta práctica hemos tenido que realizar diferentes ejecuciones, tanto en el apartado d, como es. Trabajaremos con esta instancia de inicio.

Diagrama

Descripción generada automáticamente

Ahora proporcionaremos el .soil y su vista.

Apartado d:

Diagrama

Descripción generada automáticamente

Su .soil es:

*!new* Estacion

*!new* Estacion

*!new* Estacion

*!new* Estacion

*!new* Estacion

*!new* Estacion

*!new* Estacion

*!new* Via

*!new* Via

*!insert* (Estacion1,Via1) into estacionViaIda

*!insert* (Estacion2,Via1) into estacionViaIda

*!insert* (Estacion1,Via2) into estacionViaVuelta

*!insert* (Estacion2,Via2) into estacionViaVuelta

*!new* Via

*!new* Via

*!insert* (Estacion2,Via3) into estacionViaIda

*!insert* (Estacion6,Via3) into estacionViaIda

*!insert* (Estacion2,Via4) into estacionViaVuelta

*!insert* (Estacion6,Via4) into estacionViaVuelta

*!new* Via

*!new* Via

*!insert* (Estacion6,Via5) into estacionViaIda

*!insert* (Estacion7,Via5) into estacionViaIda

*!insert* (Estacion6,Via6) into estacionViaIda

*!insert* (Estacion7,Via6) into estacionViaVuelta

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!insert* (Estacion1,Via7) into estacionViaIda

*!insert* (Estacion4,Via7) into estacionViaIda

*!insert* (Estacion1,Via8) into estacionViaVuelta

*!insert* (Estacion4,Via8) into estacionViaVuelta

*!insert* (Estacion4,Via9) into estacionViaIda

*!insert* (Estacion4,Via10) into estacionViaVuelta

*!insert* (Estacion5,Via9) into estacionViaIda

*!insert* (Estacion5,Via10) into estacionViaVuelta

*!insert* (Estacion2,Via14) into estacionViaVuelta

*!insert* (Estacion2,Via13) into estacionViaIda

*!insert* (Estacion4,Via13) into estacionViaIda

*!insert* (Estacion4,Via14) into estacionViaVuelta

*!insert* (Estacion1,Via11) into estacionViaIda

*!insert* (Estacion1,Via12) into estacionViaVuelta

*!insert* (Estacion3,Via11) into estacionViaIda

*!insert* (Estacion3,Via12) into estacionViaVuelta

*!insert* (Estacion3,Via15) into estacionViaIda

*!insert* (Estacion4,Via15) into estacionViaIda

*!insert* (Estacion4,Via16) into estacionViaVuelta

*!insert* (Estacion3,Via16) into estacionViaVuelta

*!new* Segmento

*!insert* (Segmento1,Via1) into viasDelSegmento

*!insert* (Segmento1,Via2) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento2,Via3) into viasDelSegmento

*!insert* (Segmento2,Via4) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento3,Via5) into viasDelSegmento

*!insert* (Segmento3,Via6) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento4,Via11) into viasDelSegmento

*!insert* (Segmento4,Via12) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento5,Via13) into viasDelSegmento

*!insert* (Segmento5,Via14) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento6,Via15) into viasDelSegmento

*!insert* (Segmento6,Via16) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento7,Via7) into viasDelSegmento

*!insert* (Segmento7,Via8) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento8,Via9) into viasDelSegmento

*!insert* (Segmento8,Via10) into viasDelSegmento

*!new* Linea

*!insert* (Linea1,Segmento1) into segmentoPerteneceALinea

*!insert* (Linea1,Segmento2) into segmentoPerteneceALinea

*!insert* (Linea1,Segmento3) into segmentoPerteneceALinea

*!new* Linea

*!insert* (Linea2,Segmento1) into segmentoPerteneceALinea

*!insert* (Linea2,Segmento5) into segmentoPerteneceALinea

*!insert* (Linea2,Segmento8) into segmentoPerteneceALinea

*!new* Linea

*!insert* (Linea3,Segmento4) into segmentoPerteneceALinea

*!insert* (Linea3,Segmento6) into segmentoPerteneceALinea

*!insert* (Linea3,Segmento8) into segmentoPerteneceALinea

*!new* Tren

*!insert* (Tren1,Linea1) into trenLinea

*!insert* (Tren1,Estacion1) into trenEstacion

*!new* Tren

*!insert* (Tren2,Linea1) into trenLinea

*!insert* (Tren2,Estacion7) into trenEstacion

*!new* Tren

*!insert* (Tren3,Linea2) into trenLinea

*!insert* (Tren3,Estacion1) into trenEstacion

*!new* Tren

*!insert* (Tren4,Linea2) into trenLinea

*!insert* (Tren4,Estacion5) into trenEstacion

*!new* Tren

*!insert* (Tren5,Estacion1) into trenEstacion

*!insert* (Tren5,Linea3) into trenLinea

*!new* Tren

*!insert* (Tren6,Estacion5) into trenEstacion

*!insert* (Tren6,Linea3) into trenLinea

*!new* Clock

*!insert* (Clock1,Tren1) into Time

*!insert* (Clock1,Tren5) into Time

*!insert* (Clock1,Tren6) into Time

*!insert* (Clock1,Tren4) into Time

*!insert* (Clock1,Tren2) into Time

*!insert* (Clock1,Tren3) into Time

*!*Clock1.*run*(20)

*!insert* (Tren1,Via1) into trenVia

*!insert* (Tren2,Via6) into trenVia

*!insert* (Tren5,Via11) into trenVia

*!insert* (Tren4,Via9) into trenVia

*!insert* (Tren6,Via10) into trenVia

*!insert* (Tren3,Via7) into trenVia

*!*destroy Segmento7; destroy Via7; destroy Via8

*!insert* (Tren3,Via1) into trenVia

*!delete* (Tren6,Via10) from trenVia

*!insert* (Tren6,Via9) into trenVia

*!delete* (Tren4,Via9) from trenVia

*!delete* (Tren6,Via9) from trenVia

*!insert* (Tren6,Via10) into trenVia

*!insert* (Tren4,Via10) into trenVia

*!delete* (Linea3,Segmento8) from segmentoPerteneceALinea

*!new* Via

*!new* Segmento

*!new* Via

*!insert* (Estacion4,Via18) into estacionViaIda

*!insert* (Estacion4,Via17) into estacionViaVuelta

*!insert* (Estacion5,Via18) into estacionViaIda

*!insert* (Estacion5,Via17) into estacionViaVuelta

*!insert* (Segmento9,Via18) into viasDelSegmento

*!insert* (Segmento9,Via17) into viasDelSegmento

*!insert* (Linea3,Segmento9) into segmentoPerteneceALinea

*!delete* (Tren6,Via10) from trenVia

*!insert* (Tren6,Via17) into trenVia

*!delete* (Estacion1,Via1) from estacionViaIda

*!insert* (Estacion1,Via1) into estacionViaVuelta

*!delete* (Estacion1,Via2) from estacionViaVuelta

*!insert* (Estacion1,Via2) into estacionViaIda

*!delete* (Estacion1,Via1) from estacionViaVuelta

*!insert* (Estacion1,Via1) into estacionViaIda

*!delete* (Estacion2,Via1) from estacionViaIda

*!insert* (Estacion2,Via1) into estacionViaVuelta

*!delete* (Estacion1,Via2) from estacionViaIda

*!insert* (Estacion1,Via2) into estacionViaVuelta

*!delete* (Estacion2,Via2) from estacionViaVuelta

*!insert* (Estacion2,Via2) into estacionViaIda

*!delete* (Estacion6,Via3) from estacionViaIda

*!insert* (Estacion6,Via3) into estacionViaVuelta

*!delete* (Estacion6,Via4) from estacionViaVuelta

*!insert* (Estacion6,Via4) into estacionViaIda

*!delete* (Estacion7,Via5) from estacionViaIda

*!insert* (Estacion7,Via5) into estacionViaVuelta

*!delete* (Estacion6,Via6) from estacionViaIda

*!insert* (Estacion6,Via6) into estacionViaVuelta

*!delete* (Estacion7,Via6) from estacionViaVuelta

*!insert* (Estacion7,Via6) into estacionViaIda

*!delete* (Estacion4,Via13) from estacionViaIda

*!insert* (Estacion4,Via13) into estacionViaVuelta

*!delete* (Estacion4,Via14) from estacionViaVuelta

*!insert* (Estacion4,Via14) into estacionViaIda

*!delete* (Estacion5,Via9) from estacionViaIda

*!insert* (Estacion5,Via9) into estacionViaVuelta

*!delete* (Estacion5,Via10) from estacionViaVuelta

*!insert* (Estacion5,Via10) into estacionViaIda

*!delete* (Estacion5,Via18) from estacionViaIda

*!insert* (Estacion5,Via18) into estacionViaVuelta

*!delete* (Estacion5,Via17) from estacionViaVuelta

*!insert* (Estacion5,Via17) into estacionViaIda

*!delete* (Estacion3,Via11) from estacionViaIda

*!insert* (Estacion3,Via11) into estacionViaVuelta

*!delete* (Estacion3,Via12) from estacionViaVuelta

*!insert* (Estacion3,Via12) into estacionViaIda

*!delete* (Estacion4,Via15) from estacionViaIda

*!insert* (Estacion4,Via15) into estacionViaVuelta

*!delete* (Estacion4,Via16) from estacionViaVuelta

*!insert* (Estacion4,Via16) into estacionViaIda

*!delete* (Tren3,Via1) from trenVia

*!*Clock1.NOW := 0

*!*Tren1.time := 0

*!*Tren2.time := 0

*!*Tren3.time := 0

*!*Tren4.time := 0

*!*Tren5.time := 0

*!*Tren6.time := 0

*!*Clock1.*run*(12)

*!*Clock1.*run*(108)

Apartado e:

Diagrama

Descripción generada automáticamente

Su .soil es:

*!new* Estacion

*!new* Estacion

*!new* Estacion

*!new* Estacion

*!new* Estacion

*!new* Estacion

*!new* Estacion

*!new* Via

*!new* Via

*!insert* (Estacion1,Via1) into estacionViaIda

*!insert* (Estacion2,Via1) into estacionViaIda

*!insert* (Estacion1,Via2) into estacionViaVuelta

*!insert* (Estacion2,Via2) into estacionViaVuelta

*!new* Via

*!new* Via

*!insert* (Estacion2,Via3) into estacionViaIda

*!insert* (Estacion6,Via3) into estacionViaIda

*!insert* (Estacion2,Via4) into estacionViaVuelta

*!insert* (Estacion6,Via4) into estacionViaVuelta

*!new* Via

*!new* Via

*!insert* (Estacion6,Via5) into estacionViaIda

*!insert* (Estacion7,Via5) into estacionViaIda

*!insert* (Estacion6,Via6) into estacionViaIda

*!insert* (Estacion7,Via6) into estacionViaVuelta

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!new* Via

*!insert* (Estacion1,Via7) into estacionViaIda

*!insert* (Estacion4,Via7) into estacionViaIda

*!insert* (Estacion1,Via8) into estacionViaVuelta

*!insert* (Estacion4,Via8) into estacionViaVuelta

*!insert* (Estacion4,Via9) into estacionViaIda

*!insert* (Estacion4,Via10) into estacionViaVuelta

*!insert* (Estacion5,Via9) into estacionViaIda

*!insert* (Estacion5,Via10) into estacionViaVuelta

*!insert* (Estacion2,Via14) into estacionViaVuelta

*!insert* (Estacion2,Via13) into estacionViaIda

*!insert* (Estacion4,Via13) into estacionViaIda

*!insert* (Estacion4,Via14) into estacionViaVuelta

*!insert* (Estacion1,Via11) into estacionViaIda

*!insert* (Estacion1,Via12) into estacionViaVuelta

*!insert* (Estacion3,Via11) into estacionViaIda

*!insert* (Estacion3,Via12) into estacionViaVuelta

*!insert* (Estacion3,Via15) into estacionViaIda

*!insert* (Estacion4,Via15) into estacionViaIda

*!insert* (Estacion4,Via16) into estacionViaVuelta

*!insert* (Estacion3,Via16) into estacionViaVuelta

*!new* Segmento

*!insert* (Segmento1,Via1) into viasDelSegmento

*!insert* (Segmento1,Via2) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento2,Via3) into viasDelSegmento

*!insert* (Segmento2,Via4) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento3,Via5) into viasDelSegmento

*!insert* (Segmento3,Via6) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento4,Via11) into viasDelSegmento

*!insert* (Segmento4,Via12) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento5,Via13) into viasDelSegmento

*!insert* (Segmento5,Via14) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento6,Via15) into viasDelSegmento

*!insert* (Segmento6,Via16) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento7,Via7) into viasDelSegmento

*!insert* (Segmento7,Via8) into viasDelSegmento

*!new* Segmento

*!insert* (Segmento8,Via9) into viasDelSegmento

*!insert* (Segmento8,Via10) into viasDelSegmento

*!new* Linea

*!insert* (Linea1,Segmento1) into segmentoPerteneceALinea

*!insert* (Linea1,Segmento2) into segmentoPerteneceALinea

*!insert* (Linea1,Segmento3) into segmentoPerteneceALinea

*!new* Linea

*!insert* (Linea2,Segmento1) into segmentoPerteneceALinea

*!insert* (Linea2,Segmento5) into segmentoPerteneceALinea

*!insert* (Linea2,Segmento8) into segmentoPerteneceALinea

*!new* Linea

*!insert* (Linea3,Segmento4) into segmentoPerteneceALinea

*!insert* (Linea3,Segmento6) into segmentoPerteneceALinea

*!insert* (Linea3,Segmento8) into segmentoPerteneceALinea

*!new* Tren

*!insert* (Tren1,Linea1) into trenLinea

*!insert* (Tren1,Estacion1) into trenEstacion

*!new* Tren

*!insert* (Tren2,Linea1) into trenLinea

*!insert* (Tren2,Estacion7) into trenEstacion

*!new* Tren

*!insert* (Tren3,Linea2) into trenLinea

*!insert* (Tren3,Estacion1) into trenEstacion

*!new* Tren

*!insert* (Tren4,Linea2) into trenLinea

*!insert* (Tren4,Estacion5) into trenEstacion

*!new* Tren

*!insert* (Tren5,Estacion1) into trenEstacion

*!insert* (Tren5,Linea3) into trenLinea

*!new* Tren

*!insert* (Tren6,Estacion5) into trenEstacion

*!insert* (Tren6,Linea3) into trenLinea

*!new* Clock

*!insert* (Clock1,Tren1) into Time

*!insert* (Clock1,Tren5) into Time

*!insert* (Clock1,Tren6) into Time

*!insert* (Clock1,Tren4) into Time

*!insert* (Clock1,Tren2) into Time

*!insert* (Clock1,Tren3) into Time

*!*Clock1.*run*(20)

*!insert* (Tren1,Via1) into trenVia

*!insert* (Tren2,Via6) into trenVia

*!insert* (Tren5,Via11) into trenVia

*!insert* (Tren4,Via9) into trenVia

*!insert* (Tren6,Via10) into trenVia

*!insert* (Tren3,Via7) into trenVia

*!*destroy Segmento7; destroy Via7; destroy Via8

*!insert* (Tren3,Via1) into trenVia

*!delete* (Tren6,Via10) from trenVia

*!insert* (Tren6,Via9) into trenVia

*!delete* (Tren4,Via9) from trenVia

*!delete* (Tren6,Via9) from trenVia

*!insert* (Tren6,Via10) into trenVia

*!insert* (Tren4,Via10) into trenVia

*!delete* (Linea3,Segmento8) from segmentoPerteneceALinea

*!new* Via

*!new* Segmento

*!new* Via

*!insert* (Estacion4,Via18) into estacionViaIda

*!insert* (Estacion4,Via17) into estacionViaVuelta

*!insert* (Estacion5,Via18) into estacionViaIda

*!insert* (Estacion5,Via17) into estacionViaVuelta

*!insert* (Segmento9,Via18) into viasDelSegmento

*!insert* (Segmento9,Via17) into viasDelSegmento

*!insert* (Linea3,Segmento9) into segmentoPerteneceALinea

*!delete* (Tren6,Via10) from trenVia

*!insert* (Tren6,Via17) into trenVia

*!delete* (Estacion1,Via1) from estacionViaIda

*!insert* (Estacion1,Via1) into estacionViaVuelta

*!delete* (Estacion1,Via2) from estacionViaVuelta

*!insert* (Estacion1,Via2) into estacionViaIda

*!delete* (Estacion1,Via1) from estacionViaVuelta

*!insert* (Estacion1,Via1) into estacionViaIda

*!delete* (Estacion2,Via1) from estacionViaIda

*!insert* (Estacion2,Via1) into estacionViaVuelta

*!delete* (Estacion1,Via2) from estacionViaIda

*!insert* (Estacion1,Via2) into estacionViaVuelta

*!delete* (Estacion2,Via2) from estacionViaVuelta

*!insert* (Estacion2,Via2) into estacionViaIda

*!delete* (Estacion6,Via3) from estacionViaIda

*!insert* (Estacion6,Via3) into estacionViaVuelta

*!delete* (Estacion6,Via4) from estacionViaVuelta

*!insert* (Estacion6,Via4) into estacionViaIda

*!delete* (Estacion7,Via5) from estacionViaIda

*!insert* (Estacion7,Via5) into estacionViaVuelta

*!delete* (Estacion6,Via6) from estacionViaIda

*!insert* (Estacion6,Via6) into estacionViaVuelta

*!delete* (Estacion7,Via6) from estacionViaVuelta

*!insert* (Estacion7,Via6) into estacionViaIda

*!delete* (Estacion4,Via13) from estacionViaIda

*!insert* (Estacion4,Via13) into estacionViaVuelta

*!delete* (Estacion4,Via14) from estacionViaVuelta

*!insert* (Estacion4,Via14) into estacionViaIda

*!delete* (Estacion5,Via9) from estacionViaIda

*!insert* (Estacion5,Via9) into estacionViaVuelta

*!delete* (Estacion5,Via10) from estacionViaVuelta

*!insert* (Estacion5,Via10) into estacionViaIda

*!delete* (Estacion5,Via18) from estacionViaIda

*!insert* (Estacion5,Via18) into estacionViaVuelta

*!delete* (Estacion5,Via17) from estacionViaVuelta

*!insert* (Estacion5,Via17) into estacionViaIda

*!delete* (Estacion3,Via11) from estacionViaIda

*!insert* (Estacion3,Via11) into estacionViaVuelta

*!delete* (Estacion3,Via12) from estacionViaVuelta

*!insert* (Estacion3,Via12) into estacionViaIda

*!delete* (Estacion4,Via15) from estacionViaIda

*!insert* (Estacion4,Via15) into estacionViaVuelta

*!delete* (Estacion4,Via16) from estacionViaVuelta

*!insert* (Estacion4,Via16) into estacionViaIda

*!delete* (Tren3,Via1) from trenVia

*!*Clock1.NOW := 0

*!*Tren1.time := 0

*!*Tren2.time := 0

*!*Tren3.time := 0

*!*Tren4.time := 0

*!*Tren5.time := 0

*!*Tren6.time := 0

*!*Clock1.*run*(180)

*!*Tren1.*averiar*()

*!*Clock1.*run*(180)

*!*Tren1.*reparar*()

*!*Clock1.*run*(120)

*!*Tren4.*averiar*()

*!*Clock1.*run*(240)

*!*Tren4.*reparar*()

*!*Clock1.*run*(480)

Máquinas de estado de Tren

MÁQUINA DE ESTADO DE UN TREN FUNCIONALDiagrama

Descripción generada automáticamente

En esta máquina de estados podemos destacar cuatro estados diferentes:

* Estado inicial **s**.
* **EstarEnEstacion** 🡪 Indica cuando un tren se encuentra en una estación.
* **Stopped** 🡪 Indica que el tren está parado.
* **CirculandoEnVia** 🡪 Indica que el tren está circulando por una vía.

Y a partir de estos estados, podemos observar las siguientes transiciones:

* **EstarEnEstacion->EstarEnEstacion :** si el tren está averiado, se queda en la propia estación parado. Se ejecuta la operación stop().
* **EstarEnEstacion->Stopped :** si el tren está averiado, o la vía de tren a la que quiere acceder está ocupada, entonces el tren estará parado. Se ejecuta la operación stop().
* **EstarEnEstacion->CirculandoEnVia :** si el tren no está averiado e inicialmente no se está moviendo, entonces pasará a moverse y circular en la vía. Se ejecuta la operación move().
* **Stopped->CirculandoEnVia :** si el tren no está averiado y se mueve, entonces estará circulando en vía. Se ejecuta la operación move().
* **CirculandoEnVia->EstarEnEstacion :** cuando el tren deje de circular en la vía, entonces entrará a la estación y parará allí. Para ello, se debe cumplir que el tren no esté averiado, que esté moviéndose por la vía, que se mantenga la vía ocupada por el mismo tren y que no tenga relación con la estación de llegada (esto se comprobará con la declaración self.estacionDelTren->size() = 0). Se ejecuta la operación stop().

## MÁQUINA DE ESTADOS DE UN TREN AVERIADO

Diagrama

Descripción generada automáticamente

En esta máquina de estados observamos los siguientes tres estados:

* El estado inicial s.
* TrenFuncional.
* Averiado.

A partir de estos estados, podemos destacar las siguientes transiciones:

* **TrenFuncional->Averiado :** Para que un tren se averíe, inicialmente debe ser un tren funcional. Se ejecuta la operación averiar().
* **Averiado->Averiado :** Mientras un tren esté averiado, se mantendrá parado. Se ejecuta la operación stop().
* **Averiado->TrenFuncional :** Para que un tren se repare, inicialmente debe estar averiado. Se ejecuta la operación reparar().