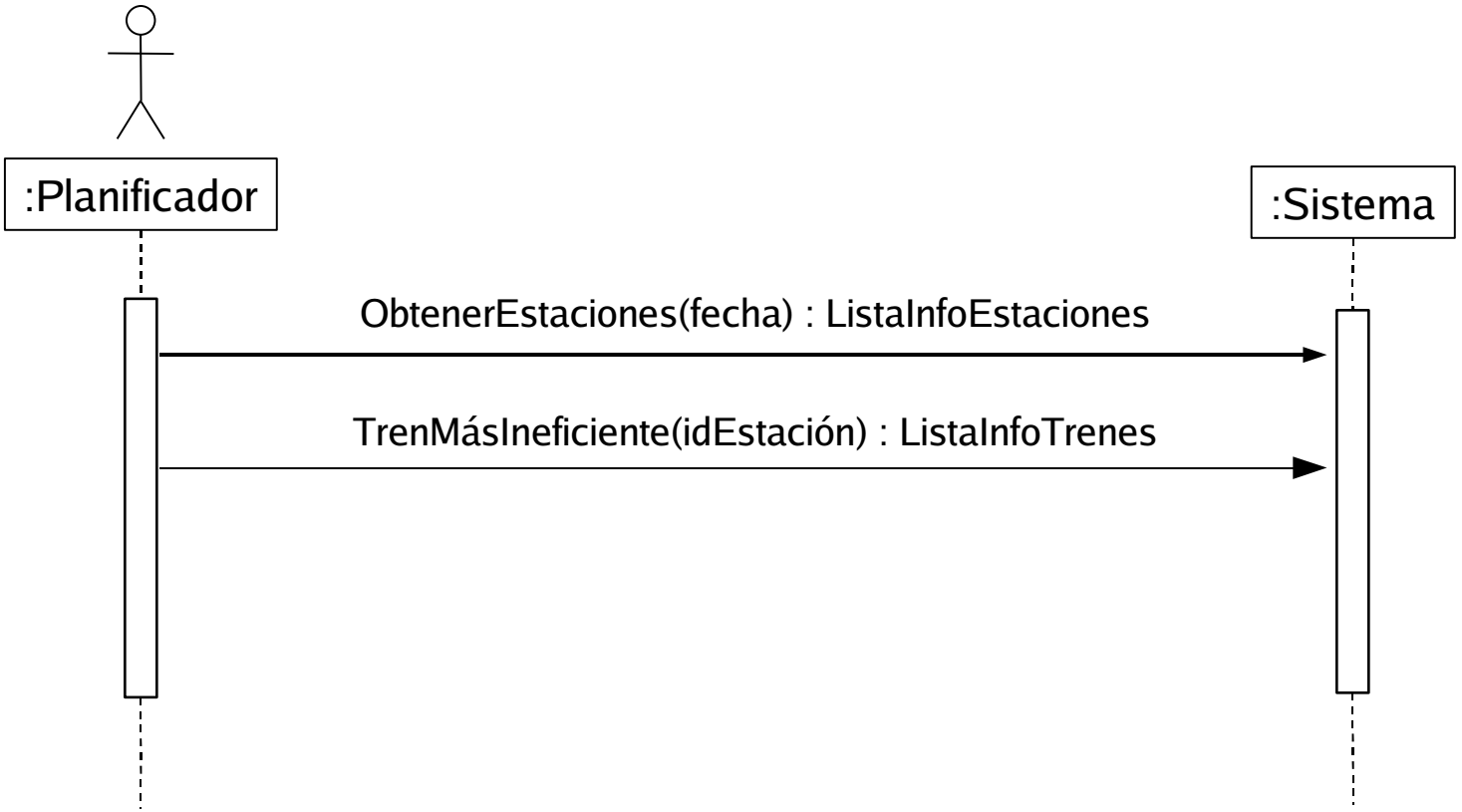
Solución Examen Junio 2006 (b)

- Ejercicio Convoy más Ineficiente (1h 20 min.)
  - Análisis (1,5 puntos): Diagrama Secuencia Sistema + Contratos
  - Diseño (2,5 puntos): Diagramas de Secuencia

Diagrama secuencia sistema



Contrato operación ObtenerEstaciones

- **Name**:ObtenerEstaciones(fecha) : ListaInfoEstaciones
- **Responsabilities**

  Dad una fecha, obtener las estaciones por las que pasan viajes en esa fecha.
- **Preconditions**

  La fecha es válida
- **Postconditions**
- **Salida**

  ListaInfoEstaciones = Lista(idEstacion, nombre, nviajes)

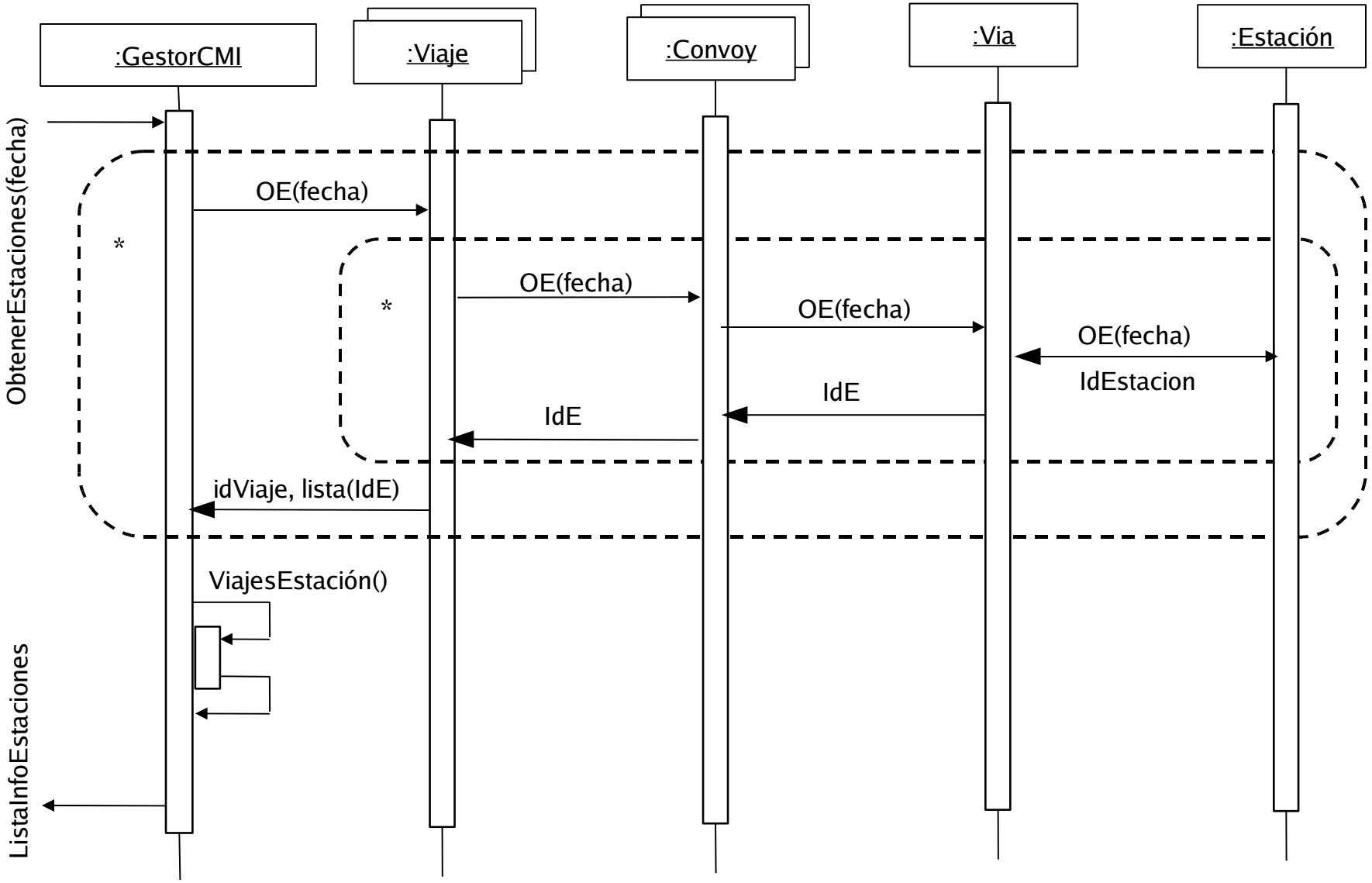
Contrato operación TrenMásIneficiente

- **Name**:TrenMásIneficiente(idEstacion) : ListaInfoTrenes
- **Responsabilities**

  Busca los trenes más ineficientes entre los que en la fecha propuesta pasan por la estación seleccionada.
- **Preconditions**

  idEstación es válido
- **Postconditions**
- **Salida**

  ListaInfoTrenes = Lista(idViaje, FechaSalida, HoraSalida, idEstaciónOrigen, nombreOrigen, idEstacion, nombre, idConvoy, Tmax, Peso, diferencia)

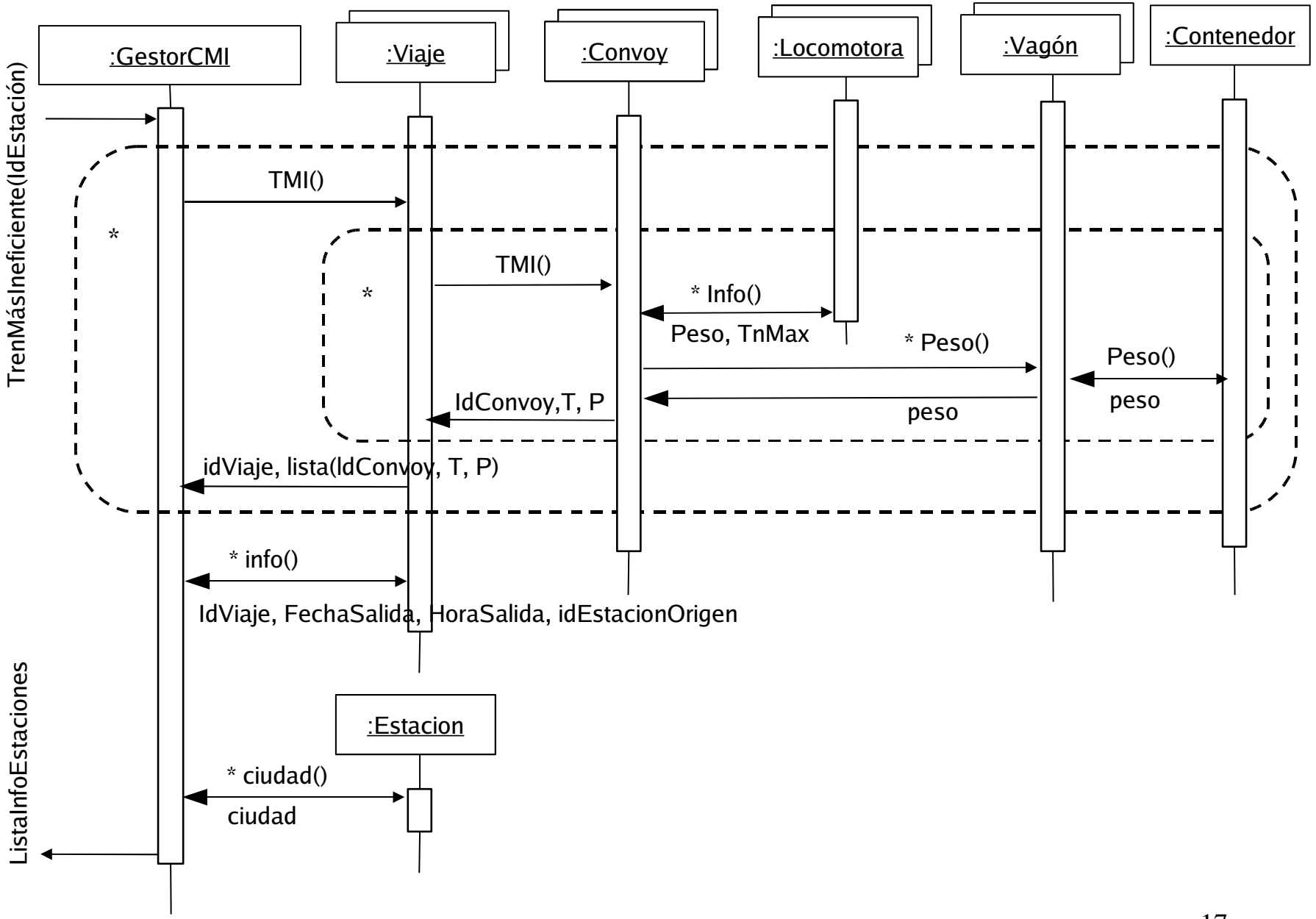

:GestorCMI
:Viaje
:Convoy
:Via
:Estación
ObtenerEstaciones(fecha)
OE(fecha)
*
OE(fecha)
*
OE(fecha)
OE(fecha)
IdEstacion
IdE
IdE
idViaje, lista(IdE)
ViajesEstación()
ListaInfoEstaciones

Escogemos el patrón controlador para gestionar el evento externo ObtenerEstaciones. Aunque otras opciones son posibles, a falta de más información al tratarse de modelar un caso de uso, seleccionamos el controlador de caso de uso: GestorCMI. Además, esta clase artificial agrupa todos los viajes y estaciones y gestiona la estructura auxiliar ViajesEstación. Con ello pretendemos un diseño global con alta cohesión y bajo acoplamiento.

Por el patrón experto, el método ObtenerEstaciones selecciona los viajes cuya FechaSalida sea la proporcionada por el Planificador.

Por el patrón experto, el método OE de Viaje es el encargado de obtener las estaciones (IdEstacion) por las que transcurrirá cada viaje (Viaje conoce la FechaSalida y los Convoyes que lo conforman). Se aplica el mismo patrón experto para los métodos OE de Convoy (conoce las vias por las que pasa), Via (conoce las estaciones por las que pasa) y Estación. Finalmente, el método OE de Estación obtiene el IdEstacion de la estación destino.

Además, el método ViajesEstación de GestorCMI gestiona la estructura de datos ViajesEstación para almacenar para cada estación los IdViaje de viajes que pasan por ellas (eliminando viajes repetidos). Finalmente, este método obtiene el total de viajes que pasan por cada estación.


:GestorCMI
:Viaje
:Convoy
:Locomotora
:Vagón
:Contenedor
TrenMásIneficiente(IdEstación)
TMI()
*
TMI()
*
* Info()
Peso, TnMax
* Peso()
Peso()
peso
IdConvoy,T, P
peso
idViaje, lista(IdConvoy, T, P)
* info()
IdViaje, FechaSalida, HoraSalida, idEstacionOrigen
:Estacion
* ciudad()
ciudad
ListaInfoEstaciones

Escogemos el patrón controlador para gestionar el evento externo TrenMasIneficiente. Aunque otras opciones son posibles, a falta de más información al tratarse de modelar un caso de uso, seleccionamos el mismo el controlador de caso de uso que para la operación anterior: GestorCMI. Además, esta clase artificial agrupa también todos los viajes y estaciones y gestiona la estructura auxiliar ViajesEstación. Con ello pretendemos un diseño global con alta cohesión y bajo acoplamiento.

Por el patrón experto, el método TrebMasIneficiente recorre los viajes de ViajesEstación que corresponden a la Estación seleccionada.

Por el patrón experto, el método TMI de Viaje es el encargado de obtener los convoyes más ineficientes de cada viaje (Viaje conoce los convoyes que lo componen). Se aplica el mismo patrón experto para el método TMI de Convoy (conoce las locomotoras y vagones que lo componen), que debe obtener el peso máximo que pueden transportar las locomotoras del convoy y el peso total entre ambos. Finalmente, el método info de Locomotora obtiene el peso y el tonelaje máximo que puede transportar y los métodos peso de Vagón y Contenedor los pesos de ambos.

Tras obtener los viajes que contienen los convoyes más ineficientes, también por el patrón experto, el método info de Viaje obtiene su idViaje, sus fechas y horas de salida y su estación origen.

Por último, aplicando el patrón experto, el método ciudad de Estación obtiene las ciudades de la estación seleccionada y las estaciones origen de los viajes más ineficientes.