



Ejercicio 1

Se pide crear un lenguaje textual con el que describir turnos (*shifts* en inglés) en una fábrica. Para describir los turnos se deben tener en cuenta 4 elementos:

- Operarios: estos son los trabajadores de la fábrica, que operan máquinas.
- Máquinas: las máquinas disponibles en la fábrica, que tienen que ser operadas por operarios.
- Tareas: una tarea describe el uso de una máquina por un operario, durante una cantidad de tiempo.
- Turnos: un turno es una lista de tareas programadas para un día concreto

A modo de ejemplo, en la Ilustración 1 se muestra un ejemplo de un modelo en su forma textual compatible con el lenguaje que queremos crear. Se asume que las unidades temporales son minutos.

```
operator Juan;
operator Pablo area='Welding';
operator Luis;

machine welder1;
machine welder2;
machine oven1 manufacturer='Bosch';
machine cooler1;

task t1 duration=10 [Juan -> oven1];
task t2 duration=15 [Pablo -> welder1];
task t3 duration=7 [Pablo -> welder2];
task t4 duration=18 [Luis -> cooler1];
task t5 duration=25 [Luis -> oven1];

shift shift1 {
    date : '15032022_morning';
    work : t1,t2,t3,t4;
}

shift shift2 {
    date : '15032022_afternoon';
    work : t2,t3,t5;
}
```

Ilustración 1 Ejemplo de descripción de dos turnos

El lenguaje debe crearse con Xtext, e implementar las siguientes características:

- No se debe permitir que haya dos elementos con el mismo identificador.
- No se debe permitir que las tareas tengan duraciones negativas.
- No se debe permitir que un turno tenga tareas que sumen más de 20 horas.



Ejercicio 2

El fichero *factory.ecore* adjunto describe un metamodelo de fábricas de caramelos. El metamodelo está basado en el creado en la Práctica 1, con la principal diferencia de que las conexiones entre elementos se especifican mediante la clase **Connection**. En la Ilustración 2 se muestra el diagrama de clases de este metamodelo.

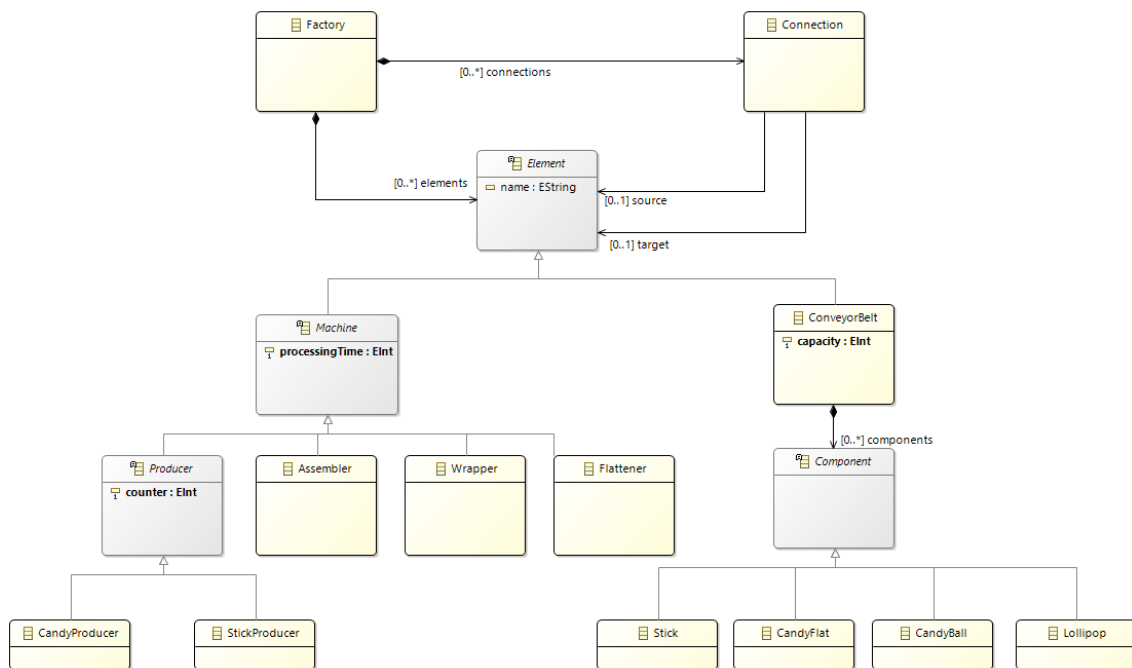


Ilustración 2 Metamodelo de fábrica de caramelos con la clase Connection

Se pide crear una sintaxis concreta textual para este metamodelo utilizando Xtext. Se deben tener en cuenta los siguientes puntos:

- La gramática debe inferirse automáticamente a partir del metamodelo.
- Se aconseja crear un proyecto nuevo (Empty EMF Project) para almacenar este metamodelo nuevo. No olvidar que para generar la gramática se debe tener previamente el fichero genmodel (también adjuntado).
- Se debe modificar la gramática para hacerla compatible con el modelo mostrado en la Ilustración 3. Notar que el **único** cambio con respecto a la gramática que se genera automáticamente es la manera en la que se describen los objetos Connection.



```
Factory {  
  elements {  
    CandyProducer candyProducer1 {  
      processingTime 5 counter 0  
    },  
    StickProducer stickProducer1 {  
      processingTime 8 counter 0  
    },  
    Assembler assembler1 {  
      processingTime 10  
    },  
    Wrapper wrapper1 {  
      processingTime 5  
    },  
    ConveyorBelt belt1 {  
      capacity 10  
    },  
    ConveyorBelt belt2 {  
      capacity 10  
    },  
    ConveyorBelt belt3 {  
      capacity 20  
    },  
    ConveyorBelt lollipopOut {  
      capacity 20  
    }  
  }  
  connections {  
    candyProducer1 -> belt1,  
    stickProducer1 -> belt2,  
    belt1 -> assembler1,  
    belt2 -> assembler1,  
    assembler1 -> belt3,  
    belt3 -> wrapper1,  
    wrapper1 -> lollipopOut  
  }  
}
```

Ilustración 3 Modelo de fábrica de caramelos en forma textual