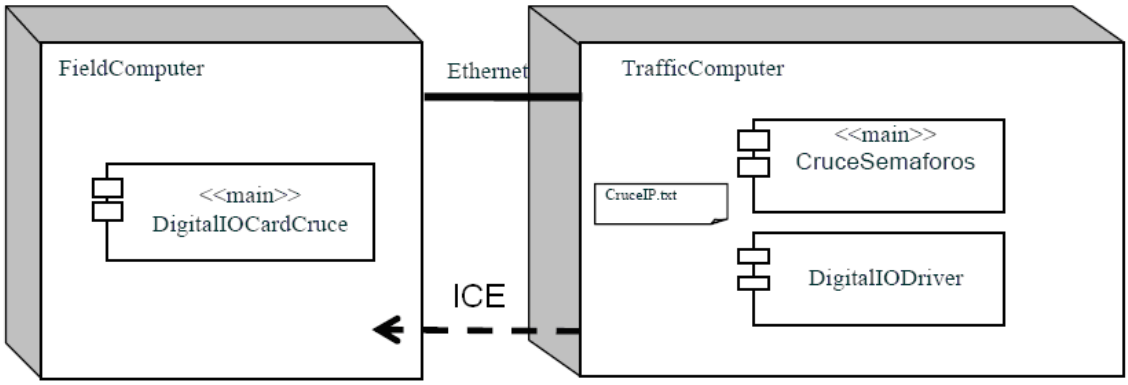
Práctica a realizar: 11-12 Enero 2012

Distribución de la aplicación Cruce Semáforos com ICE

Opción del Miércoles 11-Enero: Distribución del hardware DigitalIOCard.

En el caso se propone la distribución de la aplicación usando Ice, usando el siguiente plan de despliegue:

Criterio de diseño



El sistema debe operar sobre dos computadores:

• FieldComputer: Está localizado en la calle, y dispone de las tarjetas de entrada/salida digital, que controlan las líneas físicas que leen los sensores y encienden y apagan las luces. En él se va a instanciar el objeto de la clase DigitalIOCardCruce que a través de su interfaz DigitalIOCard permite que una aplicación externa pueda leer líneas digitales de entrada, establecer el estado de líneas digitales de salida, y notifica al cliente que espera los cambios.

• TrafficComputer: Está localizado en el centro de control de tráfico de la ciudad, y desde el se controla todo el tráfico de la ciudad, y en particular el cruce que se controla en esta práctica. En el se instancian todos los demás objetos que constituyen la aplicación de control del cruce.

Paso de objeto remoto por parámetros (callback).

A través del método *addInterruptListener*(DigitalIOListener listener) que ofrece el objeto DigitalIOCardDruce, los escuchantes mandan su referencia remota, es decir, los objetos de tipo DigitalListener, interesados en recibir los cambios de la tarjeta digital. De esta forma, el objeto DigitalIOCruce, no necesitará construir el Proxy mediante *stringToProxy* del DigitalIODriver.

Nota: En esta práctica hay que modificar:

• La clase **DigitalIOCruce**: se debe convertir en un servidor ICE que atienda las peticiones del Driver. Al mismo tiempo, es cliente, ya que se declara escuchante ,mediante el método callback *addInterruptListener()*, de los cambios que se producen en la tarjeta digital.

• La clase **Driver**: se debe convertir en un programa que constituya un cliente del servidor ofrecido por la instancia DigitalIOCruce. Esta clase lee el fichero "CrossingIP.txt" que contiene el IP del procesador FieldDriverComputer. Al mismo tiempo, es servidor, ya que, sobre los escuchantes (que mandaron su referencia remota mediante callback) envía la notificación cuando se produzca un cambio sobre la tarjeta (*notifyInterrupt()*).

Tabla 1: Ejemplo de contenido del fichero CrossingIP.txt

127.0.0.1

Véase Anexo.

Distribución de la aplicación Cruce Semáforos con ICE

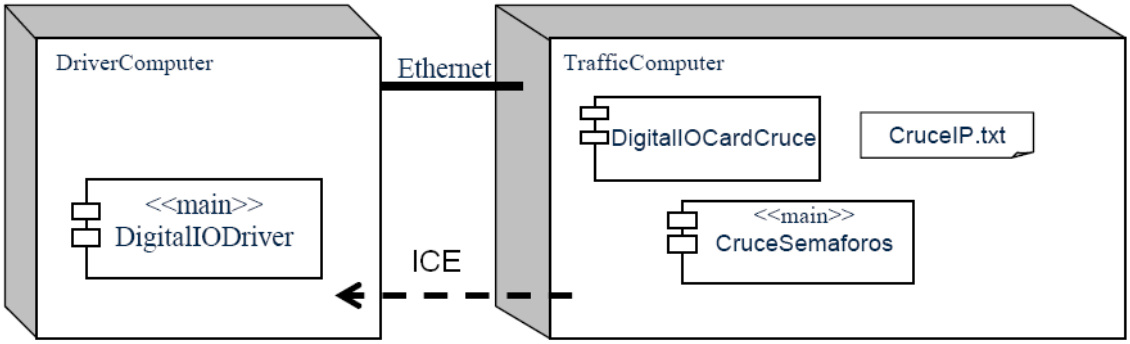
Opción del Jueves 12-Enero: Distribución del driver DigitalIODriver.

En el caso se propone la distribución de la aplicación usando ICE, usando el siguiente plan de despliegue:

Se considera que el driver de la tarjeta se encuentra distribuido, mientras que el resto, se encuentra instalado en un único computador.

Criterio de diseño

El sistema debe operar sobre dos computadores:



• DriverComputer: En el se va a instanciar el objeto de la clase DigitalIODriver que ofrece a una aplicación externa, servicios para que pueda leer líneas digitales de entrada, establecer el estado de líneas digitales de salida, y notificar al cliente que espera los cambios.

• TrafficComputer: Dispone de las tarjetas de entrada/salida digital, que controlan las líneas físicas que leen los sensores y encienden y apagan las luces. Está localizado en el centro de control de tráfico de la ciudad, y desde el se controla todo el tráfico de la ciudad, y en particular el cruce que se controla en esta práctica. En el se instancian todos los demás objetos que constituyen la aplicación de control del cruce.

En este caso, la instancia DriverComputer, actúa como servidor, mientras que el CruceSemaforos, actúa como cliente.

Paso de objeto remoto por parámetros (callback).

A través del método *addInterruptListener*(DigitalIOListener listener) que ofrece el objeto DigitalIOCardCruce, los escuchantes mandan su referencia remota, es decir, los objetos de tipo DigitalListener, interesados en recibir los cambios de la tarjeta digital. De esta forma, el que actúa como servidor en ese momento, no necesita construir el Proxy mediante la operación *stringToProxy*, de la referencia al objeto remoto del DigitalIODriver.

Nota: En esta práctica hay que modificar:

- La clase **DigitalIODriver**: se debe convertir en un servidor Ice que atienda las peticiones de CruceSemaforos. A la vez, actuará como cliente, ya que, se declarará escuchante de los cambios que se produzcan en la tarjeta , y esto lo hará a través del método callback *addInterruptListener()*.
- La clase **DigitalIOCardCruce**: se debe convertir en un programa que constituya un cliente del servidor ofrecido por la instancia DigitalIODriver. Es a la vez servidor, ya que, cuando se produzca un cambio en la tarjeta, notificará a los escuchantes (*notifyInterrupt()*). Esta clase lee el fichero "CrossingIP.txt" que contiene el IP del procesador DriverComputer.
- El programa principal **CruceSemaforo** no debe instanciar el driver.

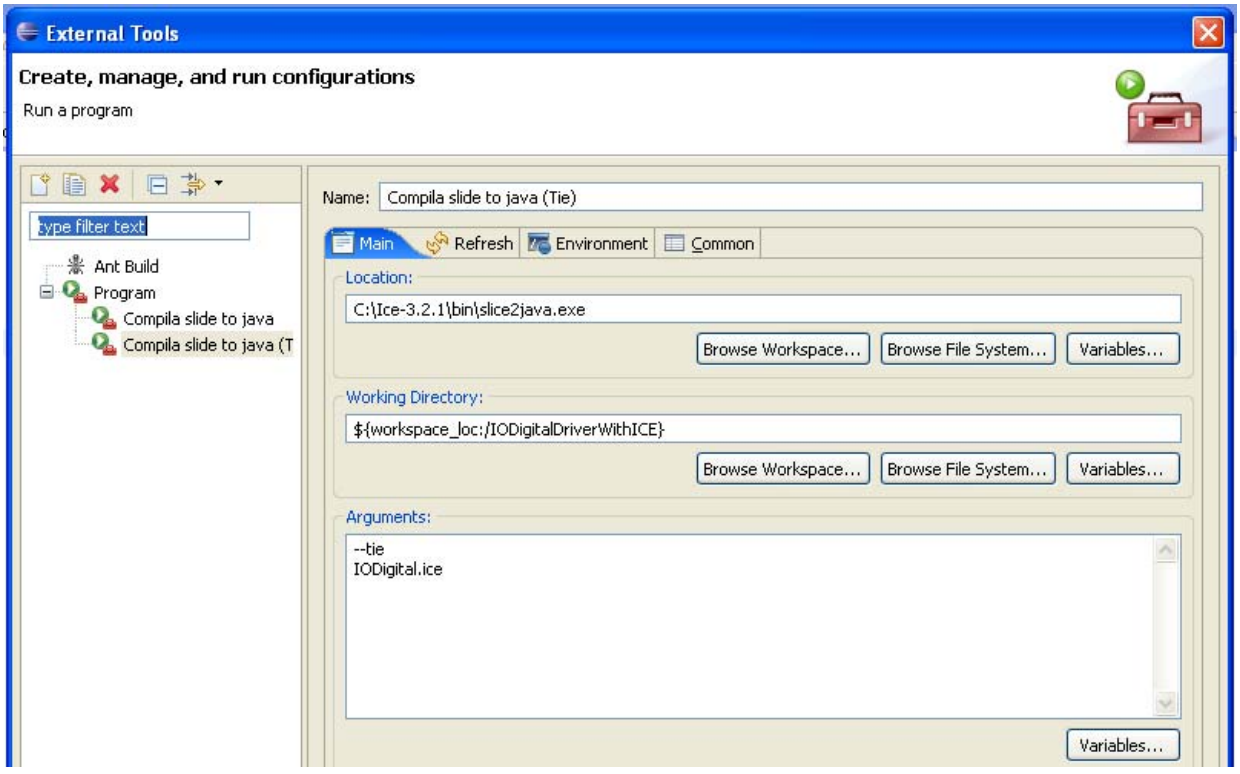
Tabla 1: Ejemplo de contenido del fichero CrossingIP.txt

127.0.0.1

Véase Anexo.

Anexo:

En el caso de que el servant implemente una clase Java, será necesario construirlo implementando la clase _*OperationsNC (servant generado por agregación). Para añadir el servant al adaptador, se hará uso de la clase _*Tie como se explicó en clase (Procodis_8_03.pdf). Para obtenerla, será necesario al compilar el fichero *.ice, añadir la opción de compilación para generarla tal y como se muestra en la figura siguiente.



El resultado de la compilación es el siguiente: