

# Sistemas Distribuidos

## Práctica 2

### Trabajo a realizar

Se proporciona un pequeño ORB implementado, así como un pequeño programa ejemplo (Proceso1.java). Para esta práctica se debe realizar lo siguiente:

#### 1. Leer el código del ORB.

Empezamos por el código del programa ejemplo Proceso1.java. En él se observa cómo crear e inicializar un ORB. De igual forma se observa cómo registrar un objeto en el ORB. Después se debe leer el código del ORB, contenido en el fichero MiniORB.java, se debe entender su función como demultiplexador de invocaciones y como repositorio de objetos. Se deben leer los ficheros de utilidad ParseIn.java, ParseOut.java que implementan el empaquetado y desempaquetado de argumentos. Se debe entender la relación entre referencias a objeto (ObjectRef.java), los proxies (Proxy.java) y esqueletos (Skeleton.java). Finalmente se debe entender el código del objeto A que se proporciona como ejemplo. Ficheros EjemploA.java, ProxyA.java y SkeletonA.java

#### 2. Implementar un programa que utilice el objeto remoto EjemploA que ha registrado el programa Proceso1. Para ello se implementará:

- a. El programa Proceso2.java
- b. El proxy de los objetos de A: ProxyA.java

#### 3. Modificar los programas Proceso1 y Proceso2 para que además de registrar y utilizar el objeto EjemploA, también se registre y se utilice un objeto de tipo EjemploB. Para ello deben implementarse los ficheros ProxyB.java, SkeletonB.java y EjemploBClass.java.

#### 4. Añadir un objeto de tipo EjemploC que tenga un método de esta forma:

```
public void test (EjemploA a);
```

El objeto C al recibir esta invocación, invocará a su vez al objeto de tipo EjemploA que recibe como argumento, invocando primero al método asociar() y después al método obtener(). De esta forma de comprobará que la referencia al objeto recibido es válida.

El objeto C deberá residir en otro proceso, por lo que además de los ficheros `EjemploC.java`, `ProxyC.java`, `SkeletonC.java` y `EjemploCClass.java`, deberá implementarse un nuevo programa, el `Proceso3.java`.

**5. Se deberá realizar una memoria que contenga:**

- a. El código de los ficheros implementados o modificados, resaltando para estos últimos qué partes han sido modificadas.
- b. Una explicación del funcionamiento del ORB.
- c. Una explicación (implementación opcional) de cómo conseguir que el ORB admita objetos de múltiples interfaces.
- d. Una explicación (implementación opcional) de cómo conseguir que el ORB transmita las posibles excepciones que ocurran en objetos remotos hasta el código que los invoca.
- e. Una explicación con todas aquellas mejoras (implementación opcional) que se pueden realizar en el ORB.