

**EJEMPLO 4.17 (cont.)**

```
        System.out.println("Obteniendo el stub del objeto remoto...");
        calc = (RMICalcInterface) registry.lookup("Calculadora");

    } catch (Exception e) {
        e.printStackTrace();
    }

    if (calc != null) {
        System.out.println("Realizando operaciones con el objeto remoto...");

        try {
            System.out.println("2 + 2 = " + calc.suma(2, 2));
            System.out.println("99 - 45 = " + calc.resta(99, 45));
            System.out.println("125 * 3 = " + calc.multip(125, 3));
            System.out.println("1250 / 5 = " + calc.div(1250, 5));
        } catch (RemoteException e) {
            e.printStackTrace();
        }

        System.out.println("Terminado");
    }
}
```

De forma análoga al caso anterior, el Ejemplo 4.17 muestra un código típico de un programa cliente usando Java RMI. En este ejemplo se debe prestar atención a los siguientes aspectos:

- ✓ El método *main* hace uso de las clases *java.rmi.registry.Registry* y *java.rmi.registry LocateRegistry* para localizar el registro de objetos remotos. La llamada al método *lookup()* de la clase *Registry* devuelve el objeto *stub* que el cliente utilizará para invocar remotamente a los métodos del objeto servidor.
- ✓ Para localizar el objeto remoto utiliza los datos de conexión del registro (*host*/dirección IP y número de puerto) y el nombre con el que se inscribió el objeto servidor en el registro (en el Ejemplo 4.17 ese nombre es "Calculadora").
- ✓ Una vez obtenido el *stub*, el cliente invoca los métodos remotos como si se tratase de métodos en un objeto local.

**ACTIVIDADES 4.14**

- Extiende el Ejemplo 4.17 para que utilice el método remoto potencia definido en las actividades anteriores y opere con números reales en lugar de enteros.