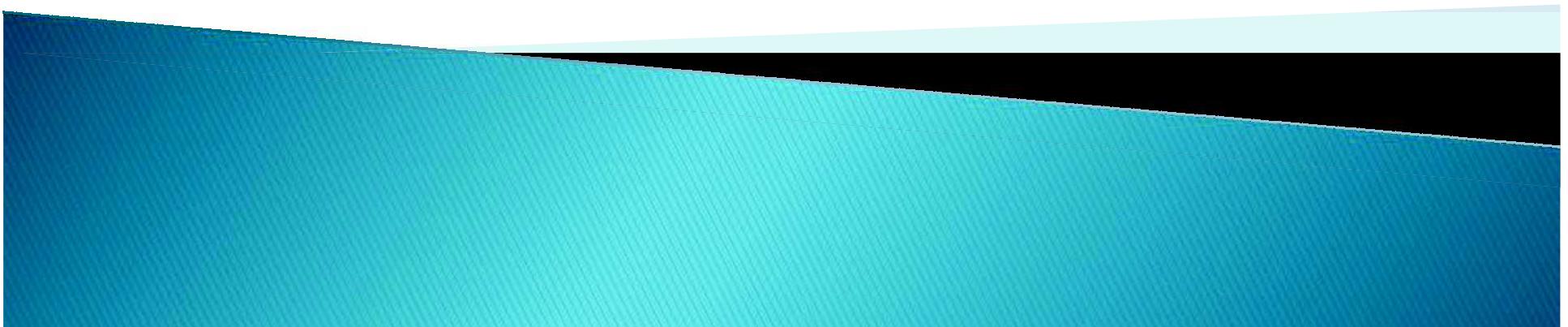


CAPÍTULO 4: GENERACIÓN DE SERVICIOS EN RED

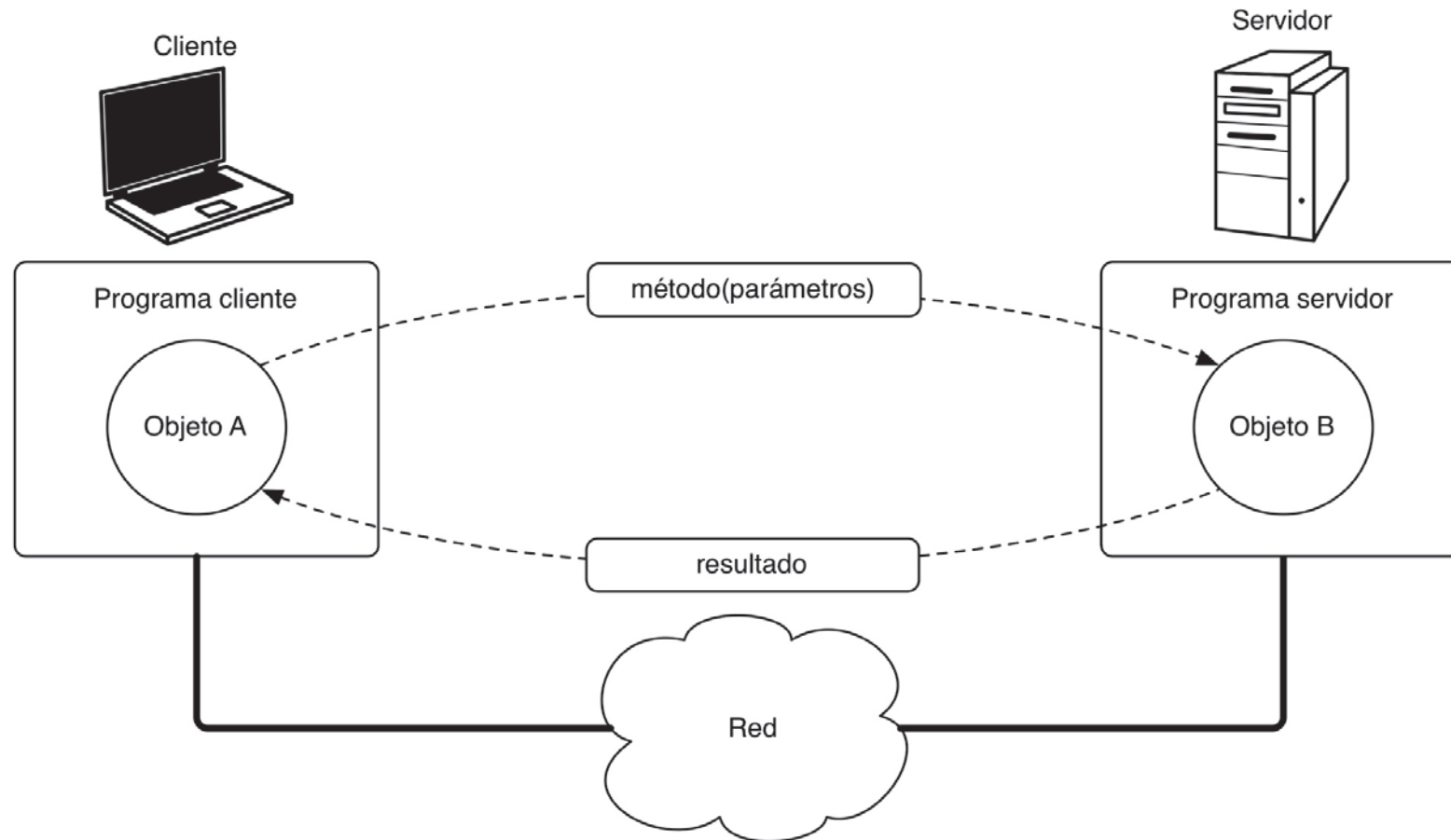
Programación de Servicios y Procesos



Invocación de métodos remotos

- ▶ En un programa orientado a objetos la invocación de un método se puede ver como un proceso de comunicación:
 - El objeto A envía un mensaje al B a modo de petición (invocación del método).
 - El objeto B procesa la petición (ejecución del método) contesta con un mensaje de respuesta (valor de retorno del método).
- ▶ La **invocación de métodos remotos** funciona de la misma forma, pero cada objeto (A y B) se encuentran en máquinas distintas.

Invocación de métodos remotos



Invocación de métodos remotos

- ▶ Objeto servidor: El objeto cuyo método es invocado. También se le llama **objeto remoto**.
- ▶ Objeto cliente: El objeto que realiza la petición.
- ▶ Método invocado: El mensaje de petición, incluyendo los parámetros del método.
- ▶ Valor de retorno: El mensaje de respuesta.

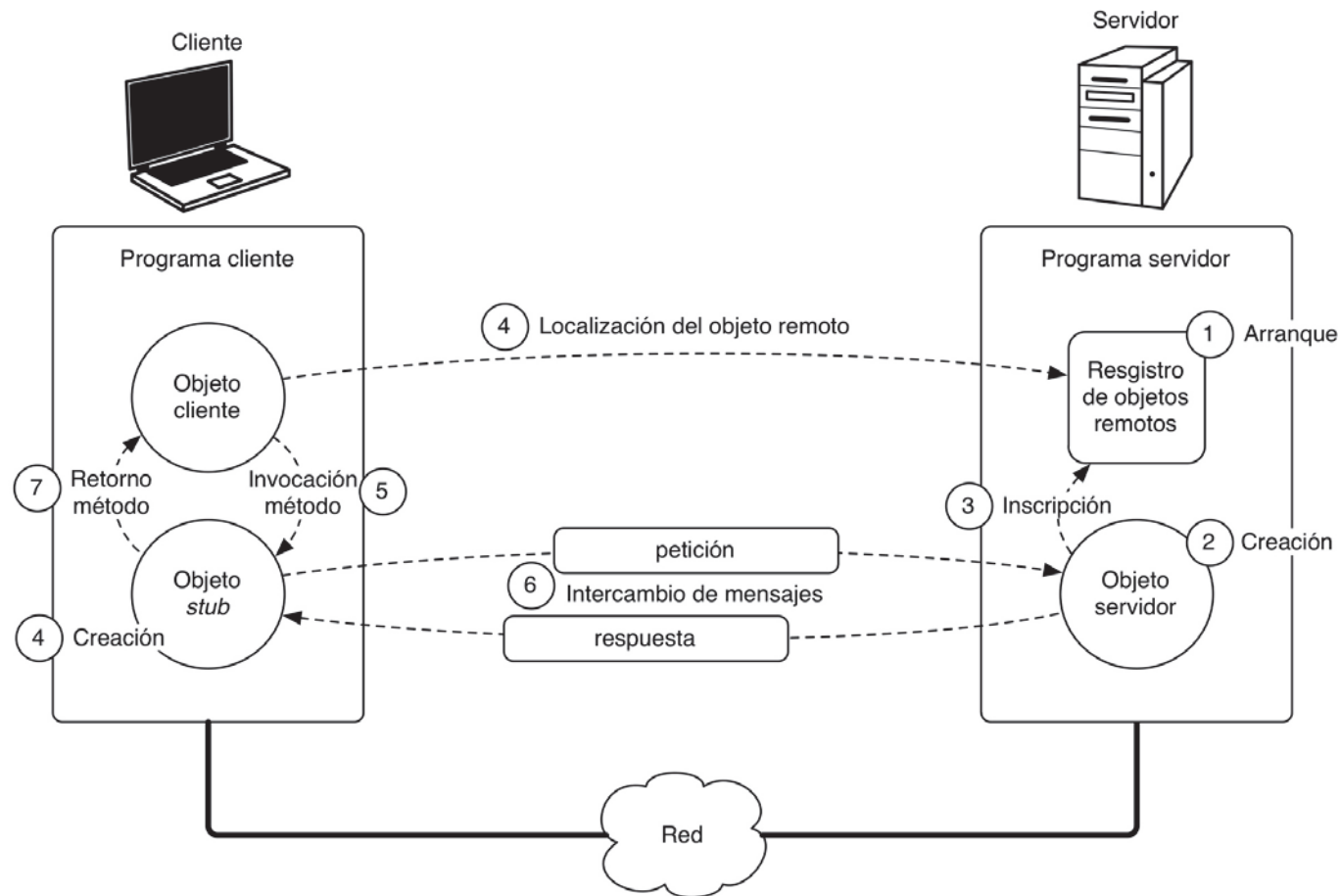
Invocación de métodos remotos

- ▶ Para que puedan llevarse a cabo invocaciones de métodos remotos, debe existir una infraestructura que capture las llamadas y genere el tráfico en la red. Esta infraestructura tiene dos componentes:
 - Objetos *stub*: Sustituyen a los objetos remotos en el programa cliente. Convierten las invocaciones de métodos en mensajes y los transfieren por la red.
 - Registro de objetos remotos: Es un servicio de nivel de aplicación que sirve para poder localizar objetos remotos.

Invocación de métodos remotos

1. (servidor) Arranque del registro de objetos remotos
2. (servidor) Creación del objeto servidor.
3. (servidor) Inscripción del objeto servidor en el registro.
4. (cliente) Localización del objeto remoto y creación del objeto *stub*.
5. (cliente) Invocación del método del objeto *stub*.
6. Intercambio de mensajes entre servidor y cliente.
7. (cliente) Obtención del valor de retorno del método.

Invocación de métodos remotos



Programación de aplicaciones basadas en métodos remotos

- ▶ Como en toda aplicación cliente/servidor, se deben realizar los siguientes pasos:
 1. Definición de las funciones del servidor.
 2. Selección de la tecnología de comunicaciones. En este caso, invocación de métodos remotos.
 3. Definición del protocolo de nivel de aplicación. Para la invocación de métodos remotos en Java esto implica la programación de la **interfaz remota**.
- 1. Una vez hecho esto, se implementan las clases del objeto servidor y el objeto cliente.

Interfaz remota

- ▶ Interfaz Java que contiene los métodos que se ejecutarán de forma remota.
- ▶ Ejemplo:

```
import java.rmi.Remote;  
import java.rmi.RemoteException;
```

```
public interface RMICalcInterface extends Remote {
```

```
    public int suma(int a, int b) throws RemoteException;  
    public int resta(int a, int b) throws RemoteException;  
    public int multip(int a, int b) throws RemoteException;  
    public int div(int a, int b) throws RemoteException;
```

```
}
```

Ejemplo de clase de objetos remotos (1 de 2)

```
import java.rmi.*;
```

```
public class RMICalcServer implements RMICalcInterface {
```

```
    public int suma(int a, int b) throws RemoteException {  
        return (a + b);  
    }
```

```
    public int resta(int a, int b) throws RemoteException {  
        return (a - b);  
    }
```

```
    public int multip(int a, int b) throws RemoteException {  
        return (a * b);  
    }
```

```
    public int div(int a, int b) throws RemoteException {  
        return (a / b);  
    }  
}
```

Ejemplo de clase de objetos remotos (2 de 2)

```
public static void main(String[] args) {  
  
    Registry reg = null;  
    try {  
        reg = LocateRegistry.createRegistry(5555);  
    } catch (Exception e) {  
        System.out.println("ERROR: No se ha podido crear el registro");  
        e.printStackTrace();  
    }  
  
    RMICalcServer serverObject = new RMICalcServer();  
  
    try {  
        reg.rebind("Calculadora",  
                    (RMICalcInterface) UnicastRemoteObject.exportObject(serverObject, 0));  
    } catch (Exception e) {  
        System.out.println("ERROR: No se ha podido inscribir el objeto servidor.");  
        e.printStackTrace();  
    }  
}
```

Ejemplo de clase de cliente de objetos remotos (1 de 2)

```
import java.rmi.*;

public class RMICalcClient {

    public static void main(String[] args) {

        RMICalcInterface calc = null;

        try {

            Registry registry = LocateRegistry.getRegistry("localhost", 5555);
            calc = (RMICalcInterface) registry.lookup("Calculadora");

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