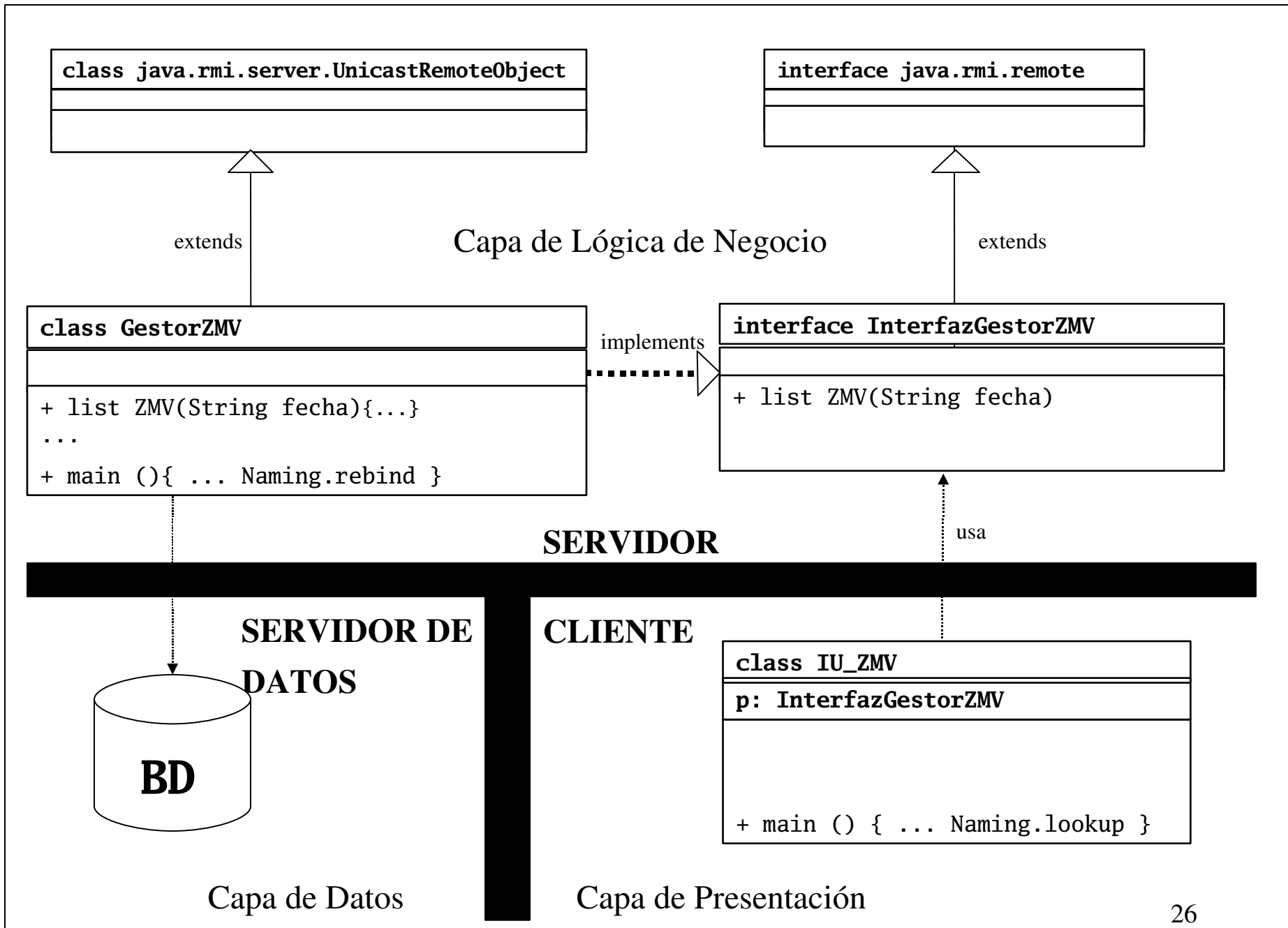
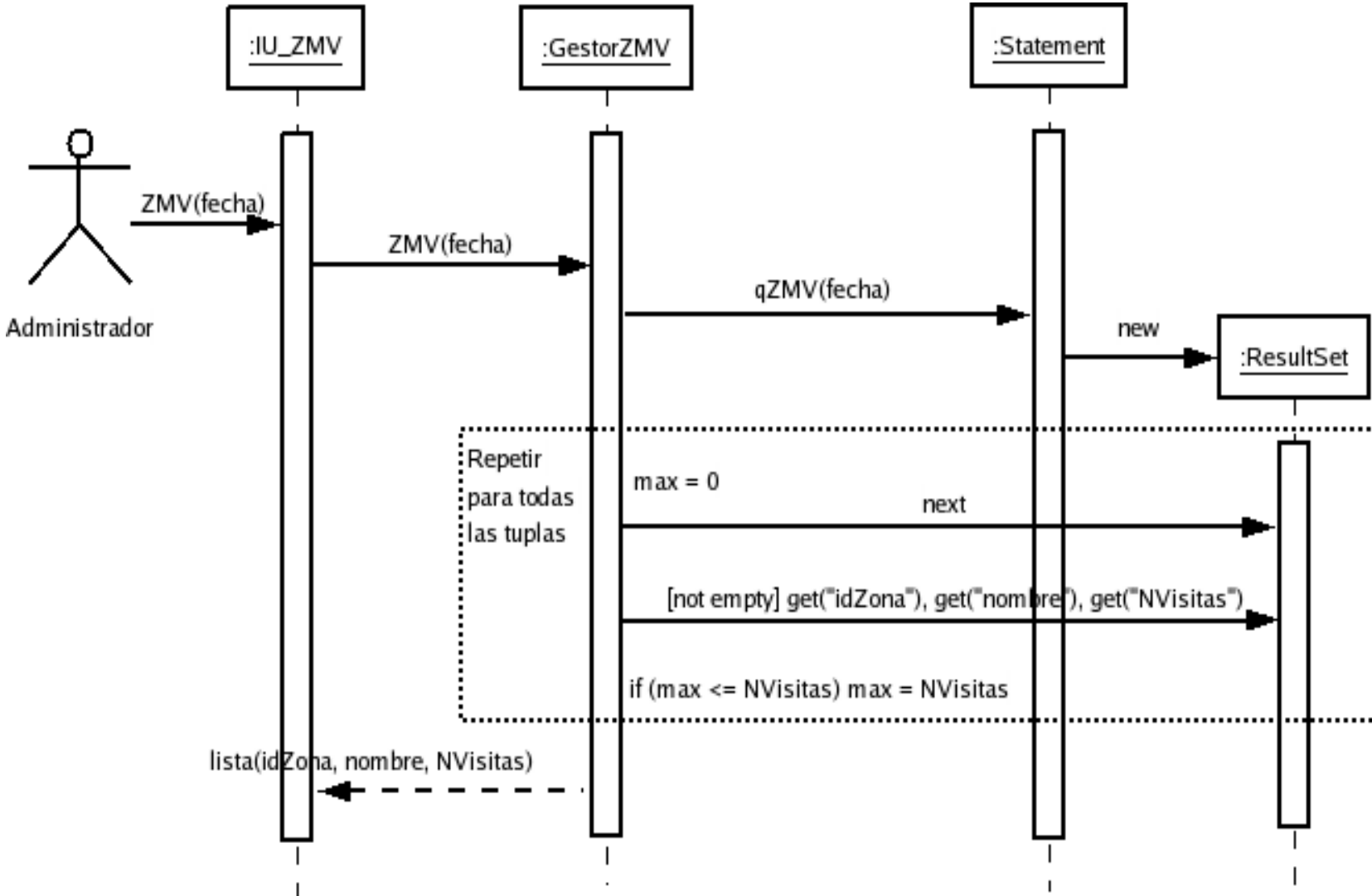


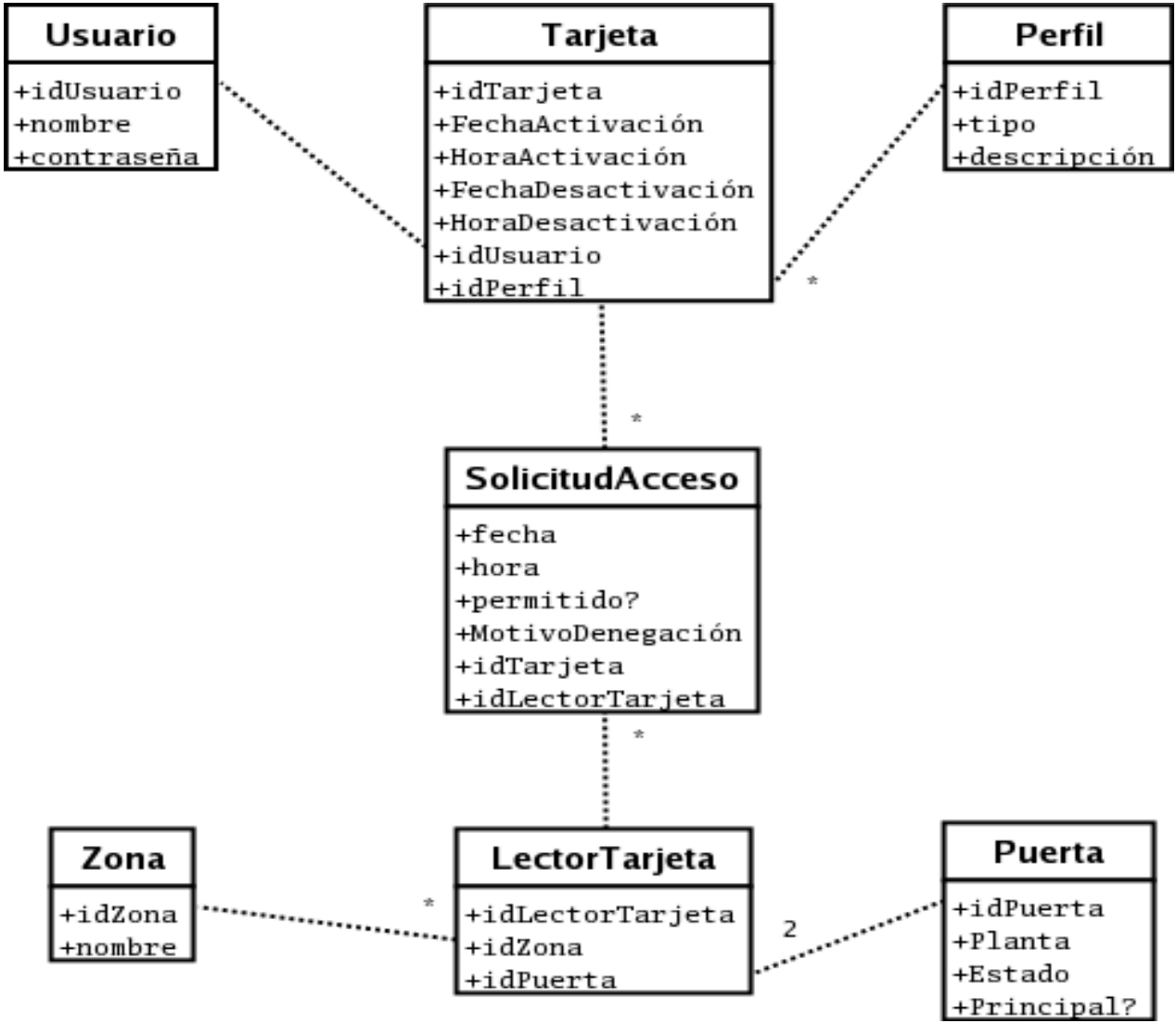
## Solución Examen Junio 2007 (c)

- Ejercicio Implementación (1h.)
  - Realizar el diagrama de clases y la separación física (1 puntos)
  - Completar la consulta SQL qZMV (2 puntos)
  - Implementar el método ZMV de la clase Gestor ZMV (4 puntos)
  - Responder a las preguntas (3 puntos)





# Ingeniería del Software



Esta solución distingue las múltiples visitas de un mismo estudiante a una zona en una misma fecha.

```
SELECT Z.idZona,Z.nombre,COUNT(DISTINCT SA.hora) AS NVisitas
FROM zona AS Z
INNER JOIN LectorTarjeta AS LT ON Z.idZona=LT.idZona
INNER JOIN SolicitudAcceso AS SA ON LT.idLectorTarjeta=SA.idLectorTarjeta
INNER JOIN tarjeta as T ON SA.idTarjeta=T.idTarjeta
INNER JOIN perfil AS P ON T.idPerfil=P.idPerfil
WHERE SA.fecha='+fecha+' AND SA.Permitido=1 AND P.tipo="estudiante"
GROUP BY Z.idZona
ORDER BY NVisitas DESC
```

Esta solución NO distingue las múltiples visitas de un mismo estudiante a una zona en una misma fecha (p.e. dos visitas de un mismo estudiante a la biblioteca en una misma fecha sólo cuenta una vez).

```
SELECT Z.idZona,Z.nombre,COUNT(DISTINCT SA.idTarjeta) AS NVisitas
FROM zona AS Z
INNER JOIN LectorTarjeta AS LT ON Z.idZona=LT.idZona
INNER JOIN SolicitudAcceso AS SA ON LT.idLectorTarjeta=SA.idLectorTarjeta
INNER JOIN tarjeta as T ON SA.idTarjeta=T.idTarjeta
INNER JOIN perfil AS P ON T.idPerfil=P.idPerfil
WHERE SA.fecha='+fecha+' AND SA.Permitido=1 AND P.tipo="estudiante"
GROUP BY Z.idZona
ORDER BY NVisitas DESC
```

```

public list ZMV (String fecha) throws RemoteException {
    String SQL = qZMV;
    list l = new LinkedList();
    int max = 0;
    int NVisitas = 0;
    boolean continue = true;
    try {
        Statement s = c.createStatement(); // c anteriormente definida
        ResultSet r = s.executeQuery(SQL);
        while (r.next() && continue) {
            NVisitas = r.getInt("NVisitas");
            if (max <= NVisitas) {
                list e = new linkedList();
                e.add(r.getString("idZona"));
                e.add(r.getString("nombre"));
                e.add(r.getInt("NVisitas"));
                l.add(e);
                max = NVisitas;
                continue = true
            } else {
                continue = false
            }
        }
    }
    catch (Exception ex) {
        System.out.println(ex.getMessage());
        list e = new linkedList(); e.add("error"); e.add("null"); e.add(0);
        l.add(e);
    }
    return l;
}

```

a) ¿Cómo son las instrucciones para cargar el driver del SGBD? ¿Cómo son las instrucciones que establecen la conexión con la base de datos?

```
Class.forName("org.gjt.mm.mysql.Driver");
```

```
Connection c =
```

```
    DriverManager.getConnection("jdbc:mysql://hiper.controla2.es/controla2");
```

b) ¿Qué instrucciones registran el servicio remoto? ¿En qué método de qué clase se encuentran?

```
GestorZMV g = new GestorZMV();
```

```
Naming.rebind("//localhost:1099/GestorZMV", g);
```

*Se encuentran en el método main de la clase GestorZMV.*

c) ¿Qué instrucciones buscan el objeto remoto? ¿En qué método de qué clase se encuentran?

```
g = (GestorZMV)Naming.lookup("rmi://super.controla2.es:1099/GestorZMV");
```

*Se encuentran en el método main de la clase IU\_ZMV.*