

SEGUNDA PRUEBA

**CUERPO: EJECUTIVO
ESCALA: ADMINISTRATIVA DE INFORMATICA
SISTEMA: LIBRE**

**RESOLUCIÓN DE CONVOCATORIA: 2 DE NOVIEMBRE DE 2005
(D.O.C.M. nº 225, de 9 de noviembre)**

Toledo a 11 de Junio de 2006.

SUPUESTO – 1

Suponga que forma parte de un equipo de desarrollo que está construyendo una aplicación en Java sobre una base de datos Oracle 9i. Se le entregan los siguientes productos:

- Dos diagramas: DIAGRAMA – A y DIAGRAMA -B
- El código fuente de 6 programas escritos en java
- Las instrucciones SQL para la creación de las tablas.

```
PROGRAMA - 1
/*
 * BaseDatos.java
 */

package exp;

import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class BaseDatos
{
    protected String consulta;
    protected ResultSet resultado = null;
    protected PreparedStatement sentencia = null;
    protected Connection conexion;

    public BaseDatos() throws Exception {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            conexion =
DriverManager.getConnection("jdbc:oracle:thin:@10.10.10.10:2020:sid","usuario","clave");
        }
        catch(Exception e) {
            throw new Exception("Error al conectar con la base de datos");
        }
    }

    public void cerrar() throws Exception {
        if(resultado != null) { resultado.close(); }
        if(sentencia != null) { sentencia.close(); }
        if(conexion != null) { conexion.close(); }
    }
}
```

PROGRAMA - 2

```
/*
 * BDPersona.java
 */

package exp;

public class BDPersona extends BaseDatos
{
    public BDPersona() throws Exception{
        super();
    }

    public Persona selectPersona(Persona miPersona) throws Exception {

        consulta = "SELECT EX_NOMBRE, EX_DOMICILIO, EX_TELEFONO "
            + "FROM EX_PERSONA "
            + "WHERE EX_NIF=? ";
        sentencia = conexion.prepareStatement(consulta);
        sentencia.setString(1,miPersona.getNif());
        resultado = sentencia.executeQuery();
        if (resultado.next()) {
            miPersona.setNombre(resultado.getString(1));
            miPersona.setDomicilio(resultado.getString(2));
            miPersona.setTelefono(resultado.getString(3));
        }

        super.cerrar();
        return miPersona;
    }
}
```

PROGRAMA - 3

```
/*
 * BDSolicitud.java
 */

package exp;

public class BDSolicitud extends BaseDatos
{
    public BDSolicitud() throws Exception{
        super();
    }

    public int selectTotalSolicitado(Solicitud miSolicitud) throws Exception {

        int sumaTotal = 0;

        consulta = "SELECT SUM(EX_SOLICITADO) FROM EX_SOLICITUD "
            + "WHERE EX_NIF=? ";
        sentencia = conexion.prepareStatement(consulta);
        sentencia.setString(1,miSolicitud.getNif());
        resultado = sentencia.executeQuery();
        if (resultado.next()) {
            sumaTotal = resultado.getInt(1);
        }

        super.cerrar();
        return sumaTotal;
    }
}
```

PROGRAMA - 4

```
/*
 * Persona.java
 */

package exp;

public class Persona
{

    private String nif;
    private String nombre;
    private String domicilio;
    private String telefono;

    public Persona() { }

    public String getNif()    { return nif; }
    public String getNombre() { return nombre; }
    public String getDomicilio() { return domicilio; }
    public String getTelefono() { return telefono; }

    public void setNif(String newNif)    { nif = newNif; }
    public void setNombre(String newNombre) { nombre = newNombre; }
    public void setDomicilio(String newDomicilio) { domicilio = newDomicilio; }
    public void setTelefono(String newTelefono) { telefono = newTelefono; }

}
```

PROGRAMA - 5

```
/*
 * Proto.java
 */

package exp;

class Proto
{
    public static void main(String[] args)
    {

        Persona miPersona = new Persona();
        Solicitud miSolicitud = new Solicitud();

        BDPersona miBDPersona = new BDPersona();
        BDSolicitud miBDSolicitud = new BDSolicitud();

        miPersona.setNif("123H");
        miSolicitud.setNif("123H");
        miBDPersona.selectPersona(miPersona);
        int totalSolicitado = miBDSolicitud.selectTotalSolicitado(miSolicitud);
        String solicitado = Integer.toString(totalSolicitado);

        System.out.println("Nombre: "+miPersona.getNombre());
        System.out.println("Domicilio: "+miPersona.getDomicilio());
        System.out.println("Telefono: "+miPersona.getTelefono());
        System.out.println("Solicitado:"+solicitado);

    }
}
```

PRGRAMA - 6

```
/*
 * Solicitud.java
 */

package exp;

public class Solicitud
{

    private String nif;
    private int expediente;
    private int solicitado;
    private int concedido;

    public Solicitud() {}

    public String getNif()    { return nif; }
    public int getExpediente() { return expediente; }
    public int getSolicitado() { return solicitado; }
    public int getConcedido() { return concedido; }

    public void setNif(String newNif)    { nif = newNif; }
    public void setExpediente(int newExpediente) { expediente = newExpediente; }
    public void setSolicitado(int newSolicitado) { solicitado = newSolicitado; }
    public void setConcedido(int newConcedido) { concedido = newConcedido; }

}
```

SQL

```
CREATE TABLE EX_PERSONA (
    EX_NIF    VARCHAR2(9) NOT NULL,
    EX_NOMBRE VARCHAR2(30),
    EX_DOMICILIO VARCHAR2(30),
    EX_TELEFONO VARCHAR2(10),
    CONSTRAINT PK_EX_PERSONA PRIMARY KEY (EX_NIF)
    USING INDEX TABLESPACE EXPE_I
);

CREATE TABLE EX_SOLICITUD (
    EX_NIF    VARCHAR2(9) NOT NULL,
    EX_EXPEDIENTE    NUMBER(8) NOT NULL,
    EX_SOLICITADO    NUMBER(8),
    EX_CONCEDIDO    NUMBER(8)
);
```

DIAGRAMA A

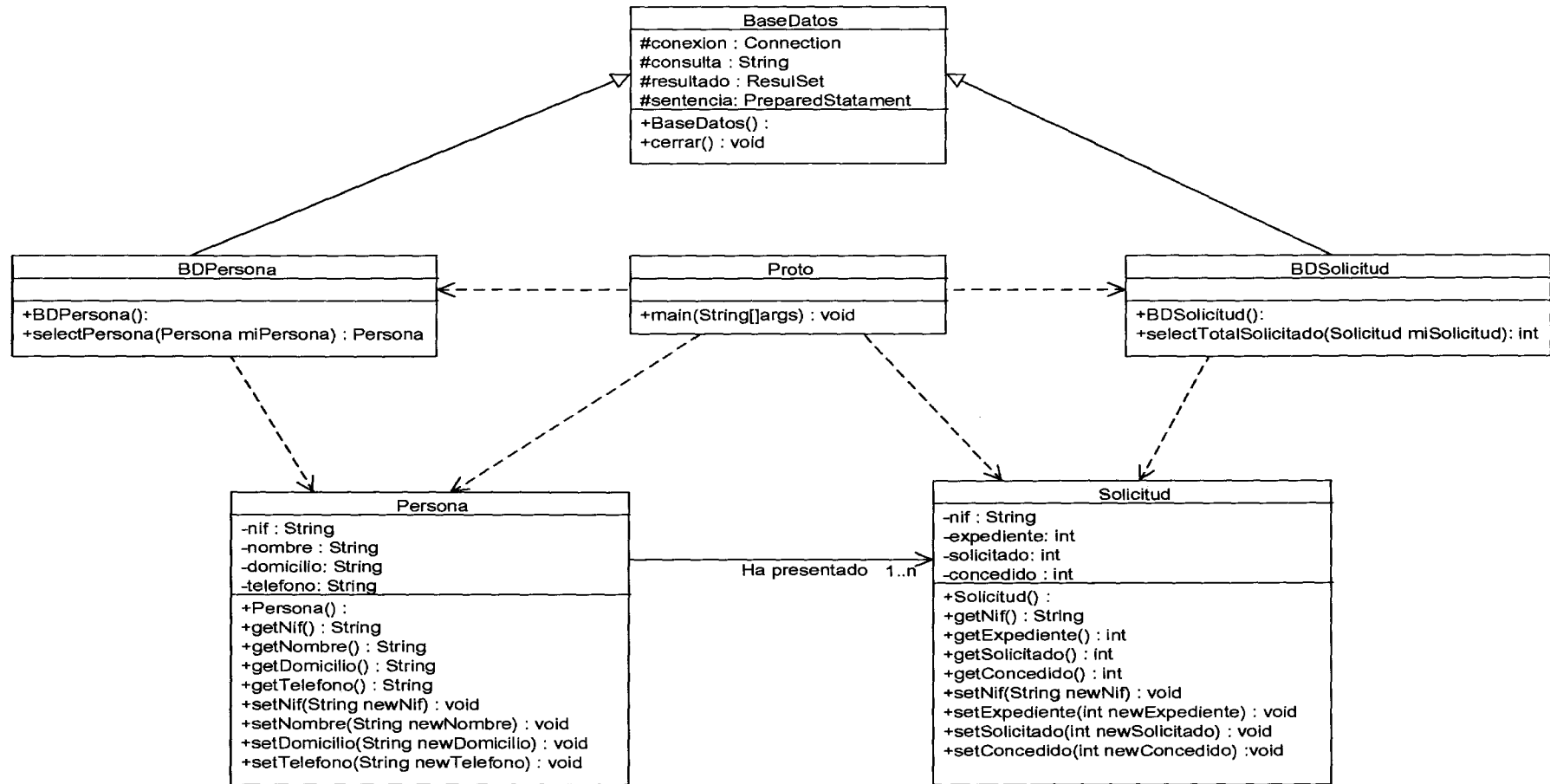
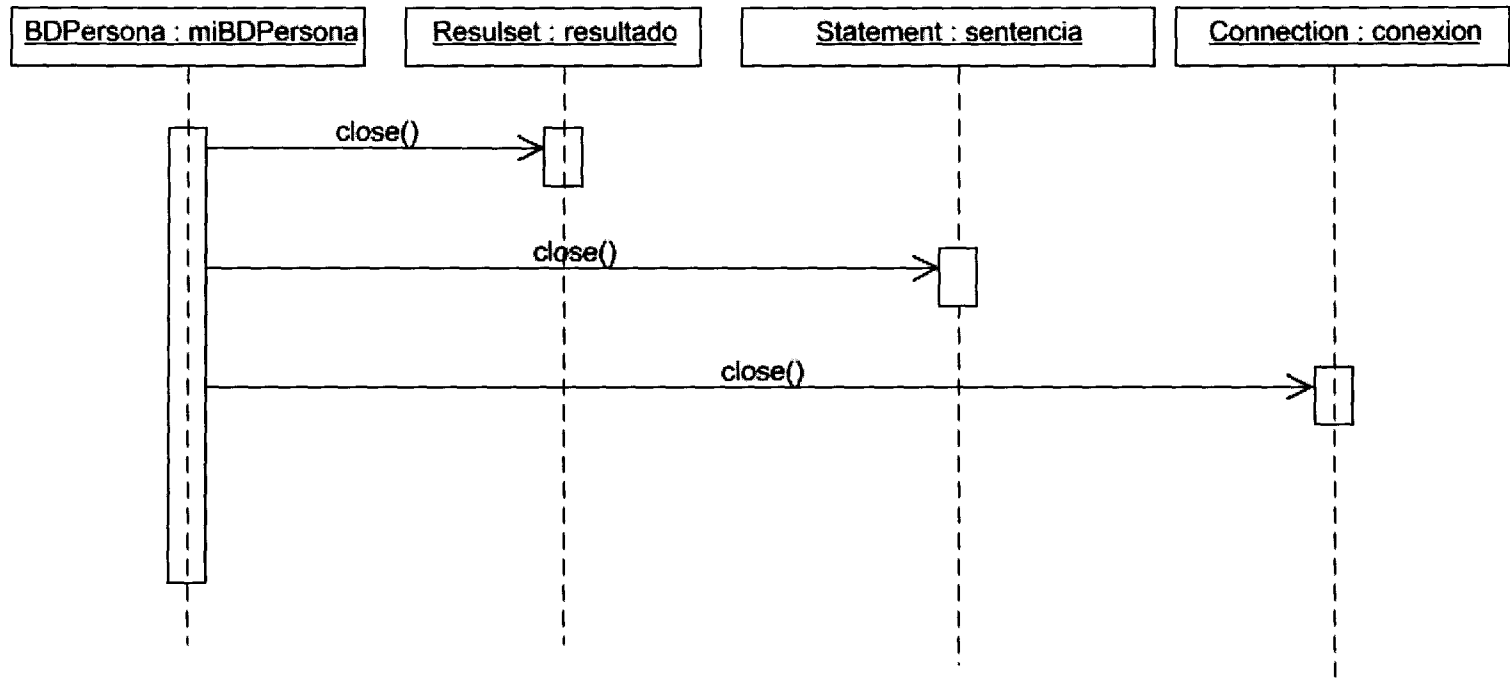


DIAGRAMA B



Con la información proporcionada por el material que le ha sido entregado, conteste por favor a las siguientes preguntas:

1) Métrica 3 permite dos orientaciones distintas para realizar un análisis. A la vista de los documentos entregados para esta práctica, ¿qué orientación se ha utilizado en este proyecto?

2) ¿Qué método de qué clase está descrito en DIAGRAMA-B?

3) ¿Cómo se denomina a DIAGRAMA-A y a DIAGRAMA-B?

4) En DIAGRAMA-A hay tres tipos de relaciones ¿cuáles son?

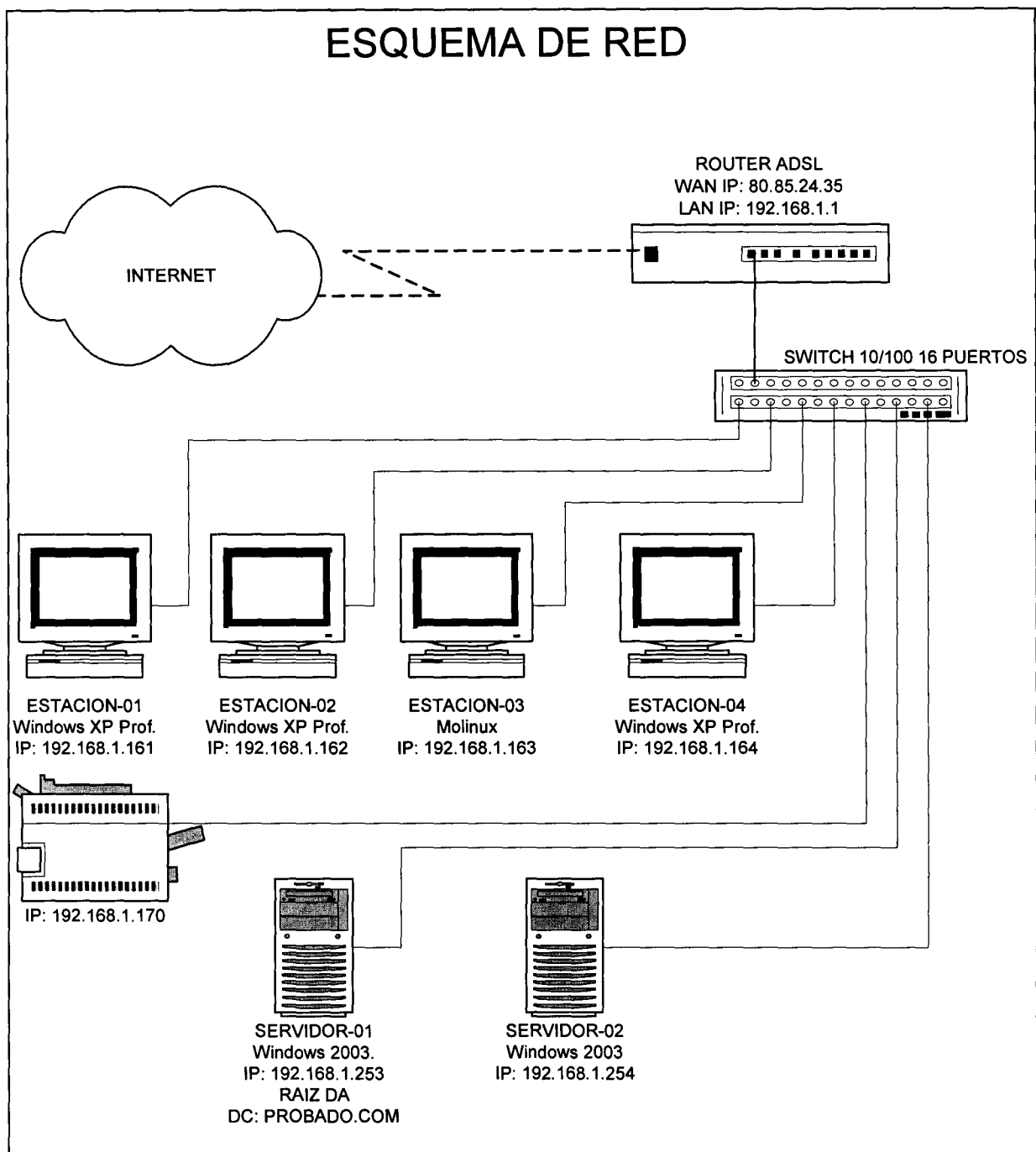
- 5) ¿Qué hace la instrucción `super()`; del método `BDSolicitud()` de la clase `BDSolicitud.java`?
- 6) Si se fija en DIAGRAMA-A verá que los atributos de la clase `BaseDatos` comienzan por una almohadilla "#". ¿Qué quiere decir esto?
- 7) ¿Qué hace la siguiente línea de código de la clase `BaseDatos.java`?
`Class.forName("oracle.jdbc.driver.OracleDriver");`
- 8) El DIAGRAMA-B representa un método de un objeto, ¿dónde se ha instanciado ese objeto en las clases de este ejercicio?

9) Si compilamos la clase Proto.java nos da un error (no es de sintaxis). ¿Por qué da error?

10) Como puede observar en la creación de tablas se ha omitido la creación de una integridad referencial de clave primaria para la tabla EX_SOLICITUD. Si ya tenemos creadas las tablas y no queremos borrarlas. ¿Con qué instrucción se podría crear una integridad referencial de clave primaria sobre la tabla EX_SOLICITUD? La clave primaria está formada por EX_NIF y EX_EXPEDIENTE.

SUPUESTO - 2

En la instalación dispone de un sistema con las características y el esquema de red que se describe a continuación:



- 4) Observando el esquema superior, determine la configuración de las siguientes propiedades del protocolo TCP/IP de las estaciones y servidores:
- a) La máscara de red.

 - b) La puerta de enlace predeterminada.

 - c) ¿Cuál sería la dirección de Broadcast en esta red?
- 5) Si desde una sesión de comando CMD desde ESTACION-01 TECLEAMOS:
- ```
C:\> ping 127.0.0.1
```
- a) ¿Qué equipo de la red responderá?
  
  - b) ¿Cómo se denomina esta dirección de red?
- 6) ¿Para qué sirve y qué problema resuelve el switch?

7) ¿Qué resultado se obtiene al ejecutar esta secuencia de acciones en el caso de que su usuario tenga permisos de administración en SERVIDOR-01?

1. Abrimos sesión con usuarios administrador local en ESTACION-02, la cual pertenece al dominio PROBADO.COM.
2. Pulsamos sobre el botón inicio y opción ejecutar. En la ventana emergente tecleamos: \\SERVIDOR-01\C\$\
3. Pulsamos el botón Aceptar

8) ¿Qué resultado se obtiene al ejecutar esta secuencia de acciones en el caso de que su usuario NO tenga permisos de administración en SERVIDOR-01?

1. Abrimos sesión con usuarios administrador local en ESTACION-02, la cual pertenece al dominio PROBADO.COM.
2. Pulsamos sobre el botón inicio y opción ejecutar. En la ventana emergente tecleamos: \\SERVIDOR-01\C\$\
3. Pulsamos el botón Aceptar

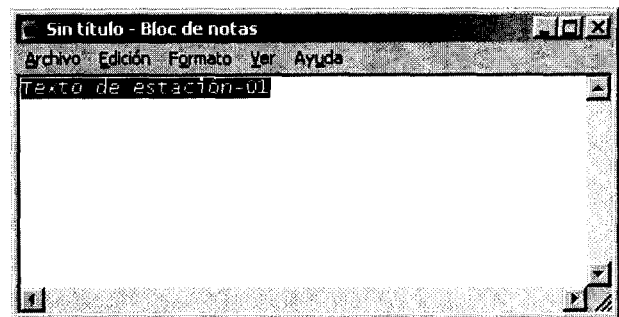


9) Explique que función realiza la siguiente instrucción y sus parámetros:

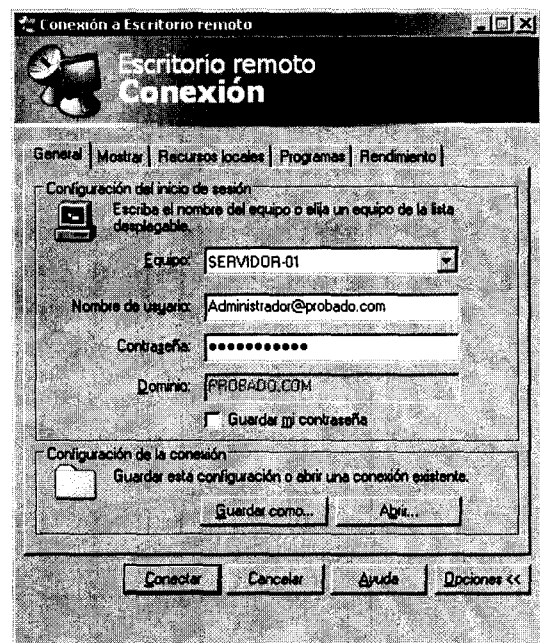
```
ntbackup.exe backup "@C:\explotacion\usr.bks" /n "usr.bkf" /d "Copia diaria de
usr" /r:no /hc:off /m normal /l:s /f "E:\Backup SOL\usr\usr.bkf"
```

10) En su estación de trabajo realiza la secuencia de acciones que se detallan a continuación:

1. Abrir sesión con usuario administrador local en ESTACION-01, la cual pertenece al dominio PROBADO.COM.
2. Pulsa sobre el botón inicio y opción ejecutar. En la ventana emergente teclea: notepad.exe y pulsa el botón Aceptar
3. En el Block de notas escribe "Texto de estación-01".



4. Selecciona el texto y pulsa en el menú edición la opción "Copiar"
5. Abre una conexión a escritorio remoto de SERVIDOR-01



6. Una vez abierta sesión en el servidor, pulsa sobre el botón inicio y opción ejecutar. En la ventana emergente teclea: notepad.exe y pulsa el botón Aceptar
7. En el Block de notas escribe "Texto de SERVIDOR-01".

8. Selecciona el texto y pulsa en el menú edición la opción "Copiar"

a) ¿Qué texto contendrá el portapapeles de ESTACION-01 una vez cerrada la sesión de usuario en la conexión a escritorio remoto de SERVIDOR-01?

b) ¿Qué texto contendrá el portapapeles de ESTACION-01 si minimizamos la conexión a escritorio remoto sin cerrar la sesión de usuario?