



PARLAMENTO DE ANDALUCIA

## SUPUESTO PRÁCTICO NÚMERO 1

El Parlamento de Andalucía utiliza una aplicación, que denominaremos aplicación X, para cuyo funcionamiento se dispone de los siguientes servidores, con sus correspondientes servicios:

- Servidor de base de datos MySQL en Linux Centos 6.5, denominado `servidorBD`. La aplicación X hace uso de dos bases de datos que se encuentran funcionando en este servidor: `bd1` y `bd2`.
- Servidor de aplicaciones en Windows 2012, denominado `servidorAPP`.
- Servidor de almacenamiento en Linux Centos 6.5, denominado `servidorALM`.

La empresa contratada por el Parlamento de Andalucía para el soporte y actualización de la aplicación X ha entregado un paquete de actualización que contiene:

- Un programa escrito en Visual Basic 2015 (VB 14.0), que utiliza el procedimiento `MoverDatosAntiguos` (Figura 1) para mover semanalmente datos en desuso generados por la aplicación al servidor de almacenamiento. Este programa se ejecuta en `servidorAPP`.
- Un script (Figura 2), que debe programarse para que se ejecute diariamente de lunes a viernes en el servidor de base de datos, `servidorBD`, para que realice una exportación local de los datos.
- Un script (Figura 3), que debe programarse para que se ejecute diariamente de lunes a viernes en el servidor de base de datos, `servidorBD`, para copiar los ficheros de la indicada exportación local al servidor de almacenamiento, `servidorALM`.

El Parlamento de Andalucía cuenta con una política de copias de seguridad con las siguientes estrategias de copia:

- Copia de seguridad disco a disco (D2D). Se realiza una copia incremental diaria de lunes a jueves a las 23:00 horas y una copia completa los viernes a las 23:00 horas.
- Copia de seguridad disco a cinta (D2T). Se realizan dos copias de seguridad de este tipo:

- Copia de seguridad completa semanal. Se realiza una copia a cintas todos los domingos a las 00:30 horas. El origen de cada una de estas copias es la última copia a disco que se hizo el viernes inmediatamente anterior. Las cintas empleadas para cada una de estas copias se sacan el lunes inmediatamente posterior para guardarlas en un armario ignífugo. Se mantienen las cintas correspondientes a las últimas cuatro copias realizadas. Estas copias están cifradas, para garantizar el cumplimiento de la normativa de protección de datos.
- Copia de seguridad completa mensual. Se realiza una copia a cintas el primer lunes de cada mes a las 7:00 horas. El origen de cada una de estas copias es la última copia a disco que se hizo el viernes inmediatamente anterior. Las cintas empleadas para cada una de estas copias se sacan inmediatamente, una vez finalizada, para guardarlas en un armario ignífugo. Se mantienen las cintas correspondientes a las veinticuatro últimas copias realizadas. Estas copias están cifradas, para garantizar el cumplimiento de la normativa de protección de datos.
- Copia de seguridad completa anual. Se realiza una copia a cintas el día 1 de enero, a las 12:00 horas. El origen de cada una de estas copias es la última copia completa a disco existente. Las cintas empleadas para cada una de estas copias se sacan inmediatamente, una vez finalizada, para guardarlas en un armario ignífugo. Se mantienen las cintas correspondientes a las últimas cinco copias realizadas. Estas copias están cifradas, para garantizar el cumplimiento de la normativa de protección de datos.

Aproximadamente dos semanas después de haberse instalado el paquete de actualización antes indicado, en concreto el viernes 3 de marzo a las 12:30 horas, se descubre que los ficheros antiguos y las exportaciones de las bases de datos no se han copiado al servidor de almacenamiento, `servidorALM`, y se han eliminado del servidor de origen, `servidorBD` o `servidorAPP` según corresponda.

Sobre la base de estos datos y figuras aportados, que el opositor debe considerar como si fueran reales, en su ejercicio ha de realizar lo que a continuación se indica:

- a) Describa las tareas concretas que realizan el programa y los scripts contenidos en el paquete de actualización (Figuras 1, 2 y 3). Explique cómo podría mejorarse el código del programa y los scripts para hacerlo más robusto, eficiente y legible y, en general, indique cualesquiera otras optimizaciones que se podrían introducir.
- b) Describa los mecanismos con los que se puede automatizar la ejecución periódica del programa y los scripts según el servidor en el que se encuentran, así como cualquier otra tarea que crea necesaria para garantizar su correcto funcionamiento.
- c) Describa las posibles causas de la pérdida de datos y explique cómo realizar una recuperación óptima de los datos perdidos, considerando la política de copias de seguridad del Parlamento de Andalucía descrita.
- d) Indique y analice cualesquiera otras cuestiones que, a su juicio, puede plantear el supuesto descrito.

### **Instrucciones para la realización del ejercicio**

1. Debe considerar que el código de las figuras no tiene ningún error de sintaxis.
2. Los números en rojo no forman parte del código, sino que son números de línea incluidos tan solo para facilitar la realización y lectura del ejercicio.
3. Todos los tipos de datos empleados en el programa de Visual Basic son los que vienen de serie en la FCL (Framework Class Library) de Microsoft.NET Framework 4.6.1, sin que se utilice ninguna librería externa adicional.
4. Las horas están expresadas en formato de 24 horas.

```
01: Sub MoverDatosAntiguos(carpetaOrigen As String, carpetaDestino As String)
02:
03:     Dim fechaTope = DateTime.Now.Subtract(New TimeSpan(365 * 5, 0, 0, 0))
04:
05:     Using memoryStream As New MemoryStream()
06:         Using datosComprimidos As New ZipArchive(memoryStream,
07:             ZipArchiveMode.Create, leaveOpen:=True)
08:
09:             Dim ficherosOrigen = (New DirectoryInfo(carpetaOrigen)).GetFiles().
10:                 Where(Function(f) f.LastAccessTime < fechaTope)
11:
12:             For Each ficheroOrigen In ficherosOrigen
13:                 Dim archivoComprimido = datosComprimidos.CreateEntry(ficheroOrigen.Name)
14:
15:                 Using escritor = New StreamWriter(archivoComprimido.Open())
16:                     escritor.WriteLine(File.ReadAllText(ficheroOrigen.FullName))
17:                 End Using
18:
19:                 File.Delete(ficheroOrigen.FullName)
20:             Next
21:         End Using
22:
23:         memoryStream.Seek(0, SeekOrigin.Begin)
24:
25:         Dim rutaDestino = DateTime.Now.ToString("yyyy-MM-dd_hh-mm-ss")
26:         rutaDestino = Path.Combine(carpetaDestino, $"backup_{rutaDestino}.zip")
27:
28:         Using streamFicheroDestino As New FileStream(rutaDestino, FileMode.CreateNew)
29:             memoryStream.CopyTo(streamFicheroDestino)
30:         End Using
31:     End Using
32: End Sub
```

Figura 1

```
33: function exportar_BD(){
34:     DISCO_BACKUP=$1
35:     BD=$2
36:     FICHERO_EXP=/DISCO_BACKUP/$BD/$BD.sql
37:     FICHERO_LOG=/DISCO_BACKUP/$BD/logs/$BD.log
38:     USUARIO=$3
39:     PASSWORD=$4
40:
41:     mysqldump -u$USUARIO -p$PASSWORD $BD > $FICHERO_EXP 2>$FICHERO_LOG
42:     if !(grep -i "error" $FICHERO_LOG); then
43:         tar -zcvf $FICHERO_EXP.tgz $FICHERO_EXP
44:         rm -f $FICHERO_EXP
45:     fi
46:
47:     if (grep "error:" $FICHERO_LOG); then
48:         mailx -s "PROBLEMA en la exportación de la $BD, ver logs"
49:         informatica@parlamentodeandalucia.es
50:     fi
51: }
52:
53: exportar_BD backup_disk bd1 userbd1 passwd1
54: exportar_BD backup_disk bd2 userbd2 passwd2
```

Figura 2

```
55: function copia_ficheros(){
56:     DST=$1
57:     DISCO_BACKUP=$2
58:     BD=$3
59:     USUARIO=$4
60:
61:     rsync -av $USUARIO@$DST:/$DISCO_BACKUP /$DISCO_BACKUP/$BD
62:     find /$DISCO_BACKUP/$BD -name '*.tgz' -type f -exec rm -f {} \;
63: }
64:
65: copia_ficheros servidorALM backup_disk bd1 root
66: copia_ficheros servidorALM backup_disk bd2 root
```

Figura 3

