

11/07/2019

A108. APLICACIONES INFORMÁTICAS

Puntuación Directa Máxima Alcanzable:	100
Puntuación máxima alcanzable	30

SEGUNDO EJERCICIO

Tiempo máximo: 150 minutos

No abra el cuadernillo hasta que se le indique y lea atentamente las instrucciones de esta portada.

- **Móviles apagados** y, al igual que los relojes, pulseras de actividad y similares, retirados de la mesa. Botellas de agua, estuches y similares pueden tenerse accesibles pero no sobre la mesa.
- Si no hay reloj en la sala, se informará por voz del tiempo que falta para la realizar la prueba: 60-30-15-10-5 y último minuto.
- Sobre la mesa exclusivamente cuadernillo de preguntas, hoja de identificación personal, DNI y bolígrafo (azul o negro). No se permite la utilización de rotuladores de color, si de TIPEX® o similares.
- Utilice en su ejercicio un tipo **de letra que permita su lectura** por el Tribunal.
- Si se le ha facilitado una **hoja de identificación** con una CLAVE rellénela con su DNI, nombre, apellidos y código/denominación de la prueba.
- **Escriba la CLAVE en las hojas de respuesta** que vaya a utilizar. NO escriba su nombre, DNI o firme la prueba ya que es causa de NO CORRECCIÓN.
- La Hoja de Identificación se recogerá transcurridos los primeros minutos de la prueba.
- Si desea un **certificado de asistencia** solicítelo en el momento en el que se le realice el control de presencia.
- Las respuestas deberán ser concretas y precisas. La corrección se realizará conforme a criterios predeterminados.
- La valoración máxima de cada pregunta, en el caso de ser diferentes, viene señalada en el enunciado de la misma.
- Si ha finalizado antes de tiempo levante la mano para que se le recoja la hoja de respuestas
- No se recogen exámenes individualmente en los últimos 3 minutos del ejercicio y si ha finalizado en este plazo permanezca en su sitio, en silencio, hasta la recogida final,
- No olvide indicar en todas sus hojas de respuestas:
 - **Código OPE (A108)**
 - **Clave** identificación
 - **Número de hoja/total** de hojas utilizadas

Gracias por su colaboración

Supuesto 1

Usted, como responsable de proyectos del Departamento de Tecnologías de la Información del Ayuntamiento de Vitoria-Gasteiz, recibe el encargo de adaptar un procedimiento existente mediante el cual las personas, a través de la página web de la Sede Electrónica del Ayuntamiento, podrán solicitar de manera electrónica y con total validez jurídica, siempre que se identifiquen de una manera válida, la inscripción a cursos de formación para personas desempleadas.

Para la tramitación de las inscripciones se requerirá:

- Encontrarse en situación de desempleo en el momento de realizar la solicitud.
- Verificar que reside en el municipio.
- Estar al corriente de pago en la Diputación Foral de Álava.

Actualmente el código de esta aplicación es mantenido por un equipo de varias personas en un repositorio de código SVN. Tanto la aplicación como sus componentes están mavenizados y hasta ahora los artefactos a desplegar son generados en los puestos desarrollo. Por las características del servicio se desarrollaron una serie de componentes de negocio que también son consumidos desde aplicaciones mantenidas por otros equipos de desarrollo. En el desarrollo inicial de esta aplicación y sus componentes se generaron unas pruebas funcionales que sirvieron para validar las entregas.

Los cambios que se nos requieren obligarán al equipo a adaptar notablemente la aplicación web actual de la sede que usan los ciudadanos, y a desarrollar nuevos componentes que podrían ser consumidos a futuro por otras aplicaciones, pero no deberían afectar a los métodos consumidos por otras aplicaciones.

Suponiendo que el personal del Ayuntamiento dispone de las herramientas de interconexión con otras administraciones para realizar dichas solicitudes, y teniendo disponibles todos los servicios que facilita la Administración en materia de interoperabilidad, indique los pasos a seguir para ofrecer y tramitar de manera electrónica estas solicitudes, en concreto:

1. Citar y explicar brevemente las opciones disponibles que permitan: (Puntuación: 20 puntos, 5 cada subapartado)
 - 1.1. La identificación y firma por parte del interesado.
 - 1.2. Notificar al interesado.
 - 1.3. Conseguir los requisitos técnicos y legales de interoperabilidad en las Administraciones de Euskadi.
 - 1.4. Identifique los servicios de interoperabilidad disponibles en NISAE que puedan ser utilizados en la solución para que el ciudadano, si lo desea, no tenga que adjuntar ninguna información que obre en poder de la Administración.
2. Respecto a la seguridad del sistema, y teniendo en cuenta lo siguiente: (Puntuación: 5 puntos)
 - El sistema de información no se considera crítico para la organización, aunque es importante que su funcionamiento sea correcto durante los plazos de presentación de las solicitudes.
 - Los datos de las solicitudes concedidas son públicos, salvo algunos datos de carácter personal como la dirección y el DNI/CIF.
 - 2.1. Determinar razonadamente el nivel requerido de cada una de las 5 dimensiones de seguridad y establecer la categoría del Sistema de Información según lo especificado en el Esquema Nacional de Seguridad.

3. Indique las categorías de amenazas contempladas en el RGPD y principales medidas que se debería adoptar para garantizar la confidencialidad, integridad y disponibilidad de los datos. (Puntuación: 5 puntos)
4. Se decide gestionar el proyecto de manera ágil con SCRUM. (Puntuación: 20 puntos, 4 cada subapartado)
 - 4.1. Describa un equipo de trabajo con diferentes perfiles y asígnele a cada uno de ellos los diferentes Roles.
 - 4.2. Describa el Flujo de Trabajo que definiría para el proyecto.
 - 4.3. Describa los diferentes tipos de Reunión que realizará.
 - 4.4. Describa los principales Documentos que se generarán.
 - 4.5. Enumere y explique brevemente los Beneficios del uso de esta tecnología.
5. Hasta ahora los desarrollos y pruebas se realizaban en el puesto local por lo que se decide utilizar para esta modificación técnicas de Integración Continua. Razona y describe: (Puntuación: 15 pts, 5 cada subapartado)
 - 5.1. Ventajas.
 - 5.2. Posible esquema y productos de una instalación con el objetivo de desplegar artefactos en desarrollo.
 - 5.3. Metodología a usar por el equipo de desarrollo a la hora de gestionar versiones, ramas, fusiones y tests automatizados de pruebas funcionales.

Supuesto 2

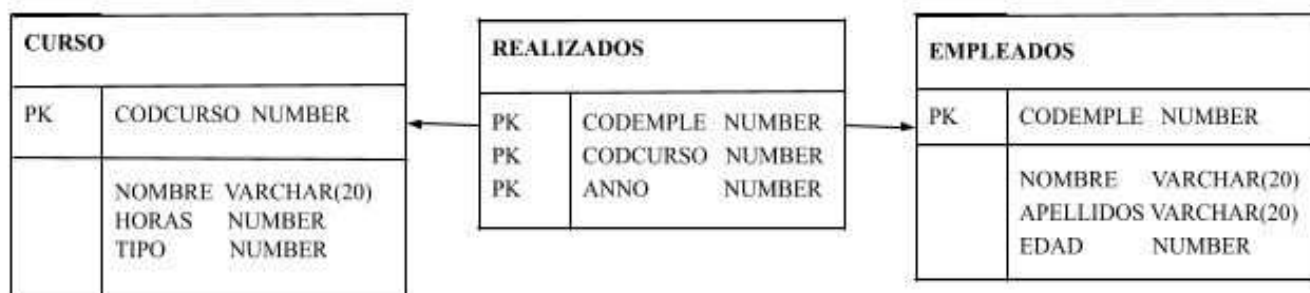
Incluida, en el código de la aplicación, se dispone de la siguiente clase java: ((Puntuación: 6 puntos, 2 cada subapartado)

```
public class CursoBean {  
  
    //Código del curso  
    private Long codCurso;  
  
    //nombre del curso  
    private String nombre;  
    ...  
    public void mostrarDatos() {  
        System.out.println("Curso: "+getNombre());  
    }  
    ...  
}
```

6. Definir dos métodos constructores de la clase, uno sin parámetros que por defecto asigne los valores nombre='curso1' y codCurso='1', y otro que pase por parámetro los valores para asignar a nombre y código del curso.
7. Definir una subclase pública CursoCompletoBean que herede de CursoBean. Debe tener dos atributos privados, uno que se llame 'valoracion' tipo Long y otro llamado 'area' tipo String. Definir los métodos get y set correspondientes al atributo 'valoracion'.
8. Sobrescribir el método mostrarDatos en CursoCompletoBean, de manera que invoque al método mostrarDatos de CursoBean, y después muestre los valores de los atributos 'valoracion' y 'area'.

Supuesto 3

Dado el esquema de Base de Datos siguiente: (Puntuación: 8 puntos, 4 cada subapartado)



9. Definir una única sentencia SQL que seleccione todos los empleados que hayan realizado en total más de 15 horas de cursos entre los años 2015 y 2017 incluidos. Debe mostrarse el código y el apellido del empleado, y el total de horas de cursos realizados en esos años, ordenado por código de empleado ascendente.

10. Definir una única sentencia SQL que seleccione la lista de todos los empleados con todos los cursos que ha realizado en todos los años. Mostrar el código y el apellido del empleado, el nombre del curso, el año de realización y las horas del curso, teniendo en cuenta que las horas de los cursos de tipo 1 se mostrarán multiplicadas por 2 y las de los cursos de tipo 2, multiplicadas por 3. Deberán aparecer solo los empleados registrados que hayan realizado algún curso. Ordenarlo por código de empleado, año del curso y nombre del curso.

Supuesto 4

Se define en java un método que accede a Base de Datos con una sentencia SQL y devuelve el resultado en un vector de objetos de tipo `CursoBean`, definido en el ejercicio 2.

Completar el código sustituyendo los círculos numerados por las partes de código que faltan y que se encuentran entre las opciones ofrecidas (no es necesario usar todas).

```

public List< 1> listar(Long vCodEmple) {
    String sqls =
        "select distinct(c.codcurso), c.nombre" +
        " from realizados r, curso c" +
        " where codemple = " + vCodEmple+
        " and c.codcurso = r.codcurso" ;
    List< 1> resultados = new ArrayList< > 1;
    StringBuffer sql = new StringBuffer(sqls);
    Statement stmt = null;
    ResultSet rs = null;
    try {
        stmt = con.createStatement();
        rs = stmt.executeQuery(sql.toString());
        2(rs. 3()) {
            CursoBean objeto = new CursoBean();
            objeto.setCodCurso(rs. 4codcurso");
            objeto.setNombre(rs.getString("nombre"));
            resultados. 5objeto);
        }
    } 6 (SQLException se) {
        System.out.println("Error");
    } finally {
        try {
            if (rs != null) rs.close();
            stmt.close();
        } 6 (Exception mie) {
            mie.printStackTrace();
        }
    }
    return resultados;
}

```

Opciones ofrecidas:

put	addElement	for	case
haveNext	insert	during	catch
next	append	while	capture
finalize	contains	lastIndexOf	size
long	add	until	throws
getLong	remove	if	throughout
getLongValue	link	toString	CursoBean
getString	while	vCurso	CursoCompletoBean

(Puntuación: 6 puntos, 1 cada etiqueta)

Supuesto 5

En base a la estructura de tablas definidas en el ejercicio 3 con los siguientes datos almacenados:

Empleados:

CODEMPLE	NOMBRE	APELLIDOS	EDAD
1	pepe	garcía	30
2	juan	gómez	35
3	carlos	fernández	40
4	javier	lópez	25

Curso:

CODCURSO	NOMBRE	HORAS	TIPO
1	ofimática	10	1
2	Atención público	20	2
3	Admin. electrónica	30	1

Realizados:

CODEMPLE	CODCURSO	ANNO
1	2	2015
1	1	2017
1	1	2018
1	1	2015
2	1	2016
2	1	2018
3	2	2017

Se dispone del siguiente método java:

```
public void listarCursos(Long vCodEmple) throws MiException {
    DAOFactory f = null;
    try
    {
        f = new DAOFactory(Names.DATA_SOURCE);
        ConexionDAO dao = f.getConexionDAO();
    //-cuerpo del programa
        List<CursoBean> lista = dao.listar(vCodEmple);
        StringBuffer mensaje;
        System.out.println("Cursos del Empleado " + vCodEmple);
        for (CursoBean curso : lista) {
            String texto Formateado = String.format("%-20s",
curso.getCodCurso() + " " + curso.getNombre());
            mensaje = new StringBuffer();
            mensaje.append(textoFormateado + " ");
            List<Long> listaAnnos = dao.listarAnnos(vCodEmple,
curso.getCodCurso());
            for (Long vAnno : listaAnnos) {
                mensaje.append(" " + vAnno);
            }
            System.out.println(mensaje);
        }
    //--fin del cuerpo del programa
    }catch {
        ...tratamiento del error...
    }
}
```

Donde `CursoBean` es la clase descrita en el ejercicio 2.

El método `listar`, es el método descrito en el ejercicio 4.

El método `listarAnnos` se define como:

```
public List<Long> listarAnnos(Long vCodEmple, Long vCurso){..}
```

y devuelve la lista de años que devuelve el select definido en el siguiente String:

```
String sqls="select anno from realizados where codemple=" +
vCodEmple+ " and codcurso=" + vCurso+ " order by anno";
```

Dar por hecho que, aunque no se muestre, la clase contiene el código necesario para que todo funcione correctamente (conexión a BD, importación de librerías, declaración de excepciones, etc.). En la BD solo se guardan datos de los últimos 4 años.

Mostrar la salida por pantalla del método `listarCursos` al invocarlo con la llamada:
`listarCursos(1L);`

(Puntuación: 8 puntos)

Supuesto 6

Dada una página html, con las siguientes etiquetas:

```
<html>
<head>
...
<link rel="stylesheet" href="./styles.css" type="text/css"/>
</head>
<body>
<p>Apartado 1.</p>
<p class="nuevo">Apartado 2.</p>
<div>
    <p class="nuevo">Apartado 3.</p>
    <p>Apartado 4.</p>
</div>
<div class="viejo">
    Apartado 5.
    <p class="nuevo">Apartado 6.</p>
    <p>Apartado 7.</p>
</div>
<p class="nuevo">Apartado 8.</p>
<div id="Viejo">
    <p>Apartado 9.</p>
</div>
<div>
    <p>Apartado 10.</p>
</div>
</body>
</html>
```

y dado el fichero styles.css:

```
p {color:black;}
p.nuevo {color:blue;}
p#nuevo {color:green;}
div {color:#9900FF;}
div.viejo {border:black 3px solid;}
div p.nuevo {color:#FFCC33;}
div.viejo p.nuevo {color:#66FF66;}
div#viejo p {color:#FF00FF;}
```

Indicar el color con el que se mostrará cada apartado (código o nombre de color) y si lleva borde o no.
(Puntuación: 7 puntos)