Responda en forma breve y concisa a un nivel técnico, solo las preguntas que domina y tiene experiencia.

**Java SE (Lenguaje de Programación Java) 2 pts c/p**

1. ¿Que imprime al ejecutar lo siguiente en la línea de comando?

> java SomeClass hello world

public class SomeClass{

public static void main(String args[]){

System.out.println(args[1]);

}

}

1. world
2. hello
3. hello world
4. ArrayIndexOutOfBoundsException is thrown
5. ¿Cuál es la línea de código que genera un error al reemplazar el comentario?

public class SomeClass{

public int i;

public static void main(String argv[]){

SomeClass sc = new SomeClass();

// Comentario

}

}

1. sc.i = 5;
2. int j = sc.i;
3. sc.i = 5.0;
4. System.out.println(sc.i);
5. Dado el siguiente código. ¿Cuál es la **expression** que imprime la palabra **“Equal”**?

String s1="Java";

String s2="java";

if(**expression**) {

System.out.println("Equal");

} else {

System.out.println("Not equal");

}

1. s1==s2
2. s1.matchCase(s2)
3. s1.equalsIgnoreCase(s2)
4. s1.equals(s2)
5. ¿Cuál de la siguiente sentencia puede usarse para determinar si “cat” se encuentra en la lista?

ArrayList<String> list = new ArrayList<String>();

list.add("dog");

list.add("cat");

list.add("frog");

1. list.contains("cat")
2. list.hasObject("cat")
3. list.indexOf("cat")
4. list.indexOf(1)
5. ¿Cuál de la siguiente sentencia imprime cada elemento?

int arr[] = {1,2,3,4,5};

1. for(int n : arr[]) { System.out.println(n); }
2. for(int n : arr) { System.out.println(n); }
3. for(int n=1; n < 6; n++) { System.out.println(arr[n]);}
4. for(int n=1; n <= 5; n++) { System.out.println(arr[n]);}
5. Dado el siguiente código. ¿Cuál de las siguientes sentencias son verdaderas?

public class SomeClass {

public SomeClass(int i, float f) { }

public SomeClass(float f, int i) { }

public SomeClass(float f) { }

public void SomeClass() { }

}

1. Error por qué no puede usarse **void**.
2. Error por qué el primer y segundo no son únicos.
3. La clase no puede tener un constructor por defecto.
4. Compila sin errores.
5. Cuál es la salida del siguiente programa.

class Exc0 extends Exception { }

class Exc1 extends Exc0 { } /\* Line 2 \*/

public class Test {

  public static void main(String args[])   {

    try {

        throw new Exc1(); /\* Line 9 \*/

    }catch (Exc0 e0) {

     System.out.println("Ex0 caught");

    }catch (Exception e){

System.out.println("exception caught");

    }

  }

}

1. Imprime Ex0 caught
2. Imprime exception caught
3. Falla la compilación error en la línea 2.
4. Falla la compilación error en la línea 9.
5. Que imprime al ejecutar el siguiente código.

public class A{

int incrementar (int x) { return x+10; }

}

public class B extends A {

int x = 12;

int incrementar (int x) { return x; }

public static void main(String[] args){

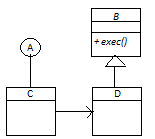
A a = new B();

int y = a.incrementar(24);

System.out.println(y);

}

}

1. 34
2. Error de sintaxis
3. 12
4. 24

**USE EL REVERSO DE LA PÁGINA**

1. Escriba el código Java del siguiente diagrama de clases.
2. En qué casos usa los siguientes componentes (Socket, RMI, JDBC, Thread, Stream I/O)

**Java EE – Programación Web – Patrones ( 2pts c/p)**

1. Cuál es la línea de código a reemplazar en el comentario para imprimir Hola Mundo usando JQuery.

$("#b1").click(function(e) {

//comentario

});

<input id="b1" type="button" value="OK" />

<div class="r1"></div>

1. $("#r1").html("Hola Mundo");
2. $(".r1").html("Hola Mundo");
3. $("#div").html("Hola Mundo");
4. Ninguno
5. Cuál de los siguientes tipos son partes de HTML5.
6. <input type="email" … />
7. <input type="number" … />
8. <input type="string" … />
9. <input type="date" … />
10. Que componentes forman parte de Java EE.
11. RMI
12. Swing - AWT
13. JSP/Servlet
14. JAX-RS
15. La etiqueta <c:if … /> forma parte de la librería:
16. Struts
17. Spring MVC
18. JSTL
19. JSP
20. La configuración del tiempo de duración de una sesión en la aplicación web se define en:
21. Web-config.xml
22. Struts-config.xml
23. Web.xml
24. Faces-config.xml
25. Una aplicación Web se empaqueta en un:
26. jar
27. ear
28. war
29. zip
30. Para definir las dependencias Maven se utiliza:
31. web.xml
32. pom.xml
33. build.xml
34. maven.xml
35. Que patrón de software permite crear una única instancia.
36. Factory
37. SingleInstance
38. Singleton
39. MVC
40. El patrón de software para centralizar el CRUD (mantenimiento de tablas) de las base de datos es:
41. DataSource
42. DAO
43. AbstractFactory
44. Facade
45. Mencione diferencias entre un Web Services SOAP y RESTFul

**Framework Spring - 2 ptos c/p**

1. El módulo que permite incluir código transversal para validaciones, seguridad, logging, excepciones etc. es:
2. AOP
3. ORM Integration
4. Spring MVC
5. Spring JDBC
6. Una de las anotaciones para inyectar objetos es:
7. @Injection
8. @Dependecy
9. @Autowided
10. @Aspect
11. La implementación de Spring para enviar sentencias sql es:
12. SpringDAO
13. SpringJDBC
14. JdbcTemplate
15. SqlTemplate
16. La clase que debe atender las peticiones Web se define con:
17. @Web
18. @Controller
19. @Services
20. @WebController
21. La anotación para controlar las transacciones es:
22. @JdbcTransactional
23. @TransactionalJdbc
24. @Transactional
25. @TransactionalTemplate
26. La anotación para representar una clase que incluye el CRUD de la BD es:
27. @Services
28. @Transaction
29. @JDBC
30. @Repository
31. Dado la siguiente configuración. ¿Cuál es la respuesta correcta?

<bean id="bankService" class="services.BankServiceImpl" p:bankName="NationalBank" />

1. Debe declararse el namespace p
2. Se está inyectando el objeto NationalBank
3. El uso de p: no es parte de una configuración de Spring.
4. N.A.
5. Para cargar un archivo de propiedades, debemos configurar en Spring.
6. Usar <util:properties />
7. Usar <properties load=”…” />
8. Usar <context:property-placeholder />
9. Usar <bean load-properties=”…” />
10. Complete la configuración de Spring para inyectar a service.ServiceImpl un dao.DaoImpl.

<bean >

<property />

</bean>

<bean />

1. Explique para que se utiliza las siguientes anotaciones:

@RequestMapping:

@ResponseBody:

**Oracle PL/SQL – 2ptos c/p**

1. Que sentencia filtra los job con el valor clerks para cada departamento?

DEPTNO MIN(SAL) MAX(SAL)

10 1300 1300

20 800 1100

1. Select deptno, MIN(sal), MAX(sal)

WHERE job = 'CLERK' GROUP BY deptno;

1. SELECT deptno, MIN(sal), MAX(sal)

GROUP BY deptno;

1. Select deptno, MIN(sal), MAX(sal)

FROM emp WHERE job = 'CLERK'

1. SELECT deptno, MIN(sal), MAX(sal)

FROM emp WHERE job = 'CLERK' GROUP BY deptno;

1. SELECT deptno, MIN, MAX

FROM emp WHERE job = 'CLERK' GROUP BY deptno;

1. Se tiene una tabla EMP con tres columnas EMP\_NUM ,EMP\_NAME and EMP\_ADDR. Cuál es la sintaxis para crear un índice en la columna EMP\_NUM.
2. Create index EMP(EMP\_NUM);
3. Create index emp\_ind on EMP;
4. Create index emp\_ind on EMP(EMP\_NUM);
5. Create index EMP(EMP\_NUM) emp\_ind;
6. Determine que provoca un error.

SELECT group\_id, isotope, AVG(atomic\_weight)

FROM char\_n

WHERE AVG(atomic\_weight) > 89.00

GROUP BY group\_id, isotope

ORDER BY AVG(atomic\_weight);

1. SELECT group\_id, isotope, AVG(atomic\_weight)
2. WHERE AVG(atomic\_weight) > 89.00
3. GROUP BY group\_id, isotope
4. ORDER BY AVG(atomic\_weight);
5. Se requiere filtrar las 100 primeras filas.
6. SELECT TOP 100 EMPNO, ENAME FROM EMP WHERE STATE = 'FL';
7. SELECT FIRST 100 EMPNO, ENAME FROM EMP WHERE STATE= 'FL';
8. SELECT EMPNO, ENAME FROM EMP WHERE ROWID <= 100;
9. SELECT EMPNO, ENAME FROM EMP WHERE ROWNUM <= 100;
10. En siguiente código que línea produce un error

cursor action\_cursor is

select name, rate, action

into action\_record

from action\_table;

1. cursor action\_cursor is
2. select name, rate, action
3. into action\_record
4. from action\_table;
5. No hay errores
6. Que sucede cuando usamos FETCH en un cursor.
7. Abre el cursor automáticamente
8. Cierra el cursor automáticamente
9. Carga el contenido del registro actual en la variable.
10. Crea una variable con el valor del registro actual.
11. En el siguiente código, Que debería escribir en lugar de **{modo}** para el parámetro v\_cost\_per\_ticket.

CREATE OR REPLACE PROCEDURE find\_cpt

(v\_movie\_id NUMBER, v\_cost\_per\_ticket **{modo}** NUMBER) IS

BEGIN

IF v\_cost\_per\_ticket  > 8.5 THEN

SELECT  cost\_per\_ticket

INTO            v\_cost\_per\_ticket

FROM            gross\_receipt

WHERE   movie\_id = v\_movie\_id;

END IF;

END;

1. IN
2. OUT
3. RETURN
4. IN OUT
5. Que se debe usar para generar un mensaje de error personalizado.
6. RAISE\_ERROR
7. SQLERRM
8. RAISE\_APPLICATION\_ERROR
9. RAISE\_SERVER\_ERROR
10. En la siguiente función, que debe agregarse para compilar sin errores.

CREATE OR REPLACE FUNCTION validar

(v\_param IN NUMBER) IS

BEGIN

 IF v\_param > 1 THEN

  RETURN TRUE;

 ELSE

RETURN FALSE;

 END IF;

END;

1. RETURN antes del IS
2. RETURN number después del IS
3. RETURN boolean antes del IS
4. RETURN boolean después del IS
5. En casos usas estas sentencias.
6. %TYPE
7. EXECUTE INMEDIATE

**Análisis de Sistemas – UML (20 minutos – 4 puntos c/p)**

Se requiere desarrollar un software para una agencia de viajes que permita gestionar el proceso de venta de los servicios y paquetes turísticos que se solicitan en sus oficinas. La agencia tiene diversas oficinas donde acuden los clientes a solicitar presupuesto de viajes a diversos países, que consta de varios servicios hoteles, tours, pasajes de aerolíneas, entre otros. Los vendedores que trabajan en cada oficina cuentan con un catálogo de tours de diversos países con los precios de cada servicio. La agencia de viaje ofrece paquetes de viajes que constan de servicios y precios establecidos para una cantidad mínima de  personas. El sistema debe considerar crear el presupuesto por el mismo cliente desde la web en modo on line y a su vez que los vendedores puedan crear su presupuesto de los clientes en sus propias oficinas en un aplicativo web interno.

Debemos considerar que al crear el presupuesto se pueda imprimir y enviar por mail la información solicitada. Si el cliente acepta el presupuesto se genera un comprobante de pago esto puede ser en efectivo dentro de las oficinas o usando tarjetas de créditos, tanto en la aplicación on line y punto de venta de cada oficina. Es importante que los servicios que se ofrecen puedan integrarse con los sistemas que son socios de negocios de la agencia como hoteles, líneas aéreas, etc, para mostrar precios y registros de reservas. Así mismo considerar que el administrador de la oficina solo podrá actualizar información de los servicios y paquetes que se ofrecen en el catálogo, como también generar reportes de ventas y administrar los vendedores de cada oficina.

La aplicación debe estar diseñada en alta disponibilidad, es decir se debe tomar las consideraciones en la arquitectura para mantener funcionamiento por algún percance de falla que ocurra en los componentes o dispositivos que lo conformen. Todo el desarrollo debe estar con tecnología web usando Java y servidor de aplicaciones JBoss conectándose a base de datos Oracle.

Se pide:

* Identificar los requerimientos funcionales y no funcionales (4 ptos)
* UML: Diseñe el diagrama de caso de uso (4 ptos)
* UML: Diseñe el diagrama de clases (4 ptos)
* UML: Diseñe el diagrama de componentes (4 ptos)
* UML: Diseñe el diagrama de despliegue (4 ptos)