



Cuestionario Departamento de Desarrollo

Título:	Cuestionario Departamento Desarrollo
Autor:	Konecta
Fecha:	06/07/2015
Versión:	4.0

Control de Cambios

Fecha	Ver.	Autor	Observaciones/Cambios
12/04/2012	1.0	Konecta	Creación del documento.
06/07/2015	3.0	Konecta	Versión 2015
02/05/2017	4.0	Konecta	Inclusión bloque básico

Cuestionario

Programación web (JAVA)

1) Define los conceptos de herencia y polimorfismo

Herencia: Como su nombre lo indica arroja el concepto de clases y subclases donde una subclase(Clase hijo) puede heredar propiedades o atributos y métodos de una clase de nivel superior o padre.

Private Class Persona{

Atributos:

Nombre
Apellido
Edad
Sexo

Metodos

Caminar()
Correr()
Comer()
Beber()
Dormir()

}

Public Class:Hombre extends Persona{

Nombre
Apellido
Edad
Sexo
Color_cabello
Raza

Metodos

Caminar()
Correr()
Beber()
Dormir()
Estudiar()
Trabajar()
Jugar()
Llorar()

}

Polimorfismo: Se entiende por polimorfismos las formas o comportamientos que pueden adquirir o alcanzar una clase u objeto sin perder su identificación Ejemplo: El signo + puede tomar un comportamiento de operador suma o un comportamiento de concatenar cadenas de caracteres

2) Explica la diferencia entre un Interface y una clase Abstracta

Una clase abstracta es una clase normal común y corriente con la diferencia de que posee un método vacío o abstracto para definir qué se debe hacer pero no como se hace.

Una interfaz posee clases abstractas y por si solas no hacen nada o no ejecutan nada y al igual que las clases abstractas dicen que se debe hacer pero no como, esto ultimo se define en las clases que implementen la interface. La implementación también diferencia una interfaz de una clase abstracta.

3) ¿Cuál de estas declaraciones no es correcta?

- a) `public class MiClase implements Cloneable`
- b) `private protected class MiClase`
- c) `public abstract class MiClase`
- d) `class MiClase extends Thread`

La que no es correcta es la b) porque esta hacienda referencia al alcance privado y protegido a la vez se recuerda que los alcances son (Public, Private, Protected)

4) ¿Cuál es el efecto de añadir final a la declaración de una clase?

El efecto es que una vez agregada la declaración final no permite crear clases derivadas de la clase declarada. Por ejemplo si se crea una clase la clase cuadrado no se espera generar una clase derivada con los mismos patrones de la clase cuadrado.

5) ¿Cuál es el efecto de añadir final a la declaración de una variable?

No es posible reasignar valor o cambiarlo lo que indica que su uso esta limitado a los hechos en los cuales necesitamos que permanezcan durante la ejecución de un programa algunos valores constantes.

6) ¿Cuál es el efecto de añadir final a la declaración de un método?

Por el contrario una vez declarado final un método se pretende crear una clase derivada y redefinir o traer con ella algunos métodos de la clase padre.

a. ¿Cuál es la diferencia principal entre un Map y un HashMap?

Map: Es una interfaz que define el uso de algún componente o proceso en Java.

HashMap: Define cuando se ejecuta dicha interfaz o Map, es decir es la clase que la implementa a Map

7) ¿Qué es un deadlock?

En sistemas operativos se conoce como interbloqueo y es aquella actividad en la que dos o mas procesos esperan un recurso cualquiera que nunca va a ser otorgado por el procesador. La única forma de que este ocurra es cuando se liberen los recursos que e esperan por parte de cada proceso.

8) ¿A qué corresponden las siglas MVC?

MVC: Corresponde a un modelo o patrón de diseño llamado modelo vista controlador.

Spring 3

¿Conoce Spring (ha usado Spring 3)? Comente brevemente su experiencia. ¿Qué IDE usó para trabajar con Spring?

Es un framework implementación y diseño para aplicaciones Java EE es decir a nivel empresarial ya que permite usar servidores tomcat para el despliegue de aplicaciones.

Los IDE que he usado son Eclipse, Net Bean, Spring boot.

¿Conoce el concepto de “Inyección de dependencias” (o “Inversión de Control”)? Defínalo brevemente.

Es la implementación de módulos de control independientes que permiten realizar cambios en el comportamiento de nuestras aplicaciones ya que sus componentes hacen uso de las funcionalidades

Comente alguna anotación que permita realizarlo con Spring.

@value puede utilizar esta variable instancia de una clase y reutilizarla en sus clases secundarias

¿En qué consisten las anotaciones @Component, @Repository, @Controller y @Service?

@Component es el estereotipo principal e indica que una clase con esta anotación es un component o Bean de Spring.

@Repository, @Controller y @Service son especificaciones o usos específicos de la clase @Component dependiendo del propósito guardar o almacenar datos, presentación, entre otros

¿Cuáles son las diferencias entre ellas?

@Repository: Uso de clases en persistencia o repositorio para bases de dato.

@Controller: Uso de clases para definir el controlador o controladores a utilizar

@Service: Uso de clases en la clase de servicio

¿En qué consiste la anotación @Qualifier?

Consiste en una situación en la que cree más de un bean del mismo tipo y desee conectar solo uno de ellos con una propiedad

Servicios Web

¿En qué consisten los servicios Web? Describalo brevemente.

Son servicios que se encuentran publicados en la nube administrados como librerías o paquetes de Google que tienen distintas funcionalidades, es lo que en muchos casos conocemos como código reutilizable con un potencial de aplicación constante.

¿Ha trabajado con algún marco de trabajo (o Framework) de Servicios Web con Java? ¿Cuales? ¿Qué herramientas utilizó?

API REST

¿Conoce el protocolo SOAP? ¿y la filosofía REST? Explíquelo brevemente.

Protocolo SOAP: Es un protocolo entre consumidor y proveedor o aportante de la información basado en XML pero desde el WSDL para interprete XML y entrega información no comprensible para el ser humano. Usa invocación a través de RPC Remote Process Call y esta sobre http, smtp, ftp, entre otros.

REST: Solo funciona en http y entrega información de en XMLy JSON llamando los servicios a través de URL con estilo de arquitectura

JPA

¿En qué consiste JPA? ¿Conoce algún marco de trabajo similar?

JPA es el gestor de entidades para bases de datos a nivel de código, traduce Java Persistence que indica gestión de almacen o guardado de datos.

Funciona similar a la gestión de Django para Python

¿Cómo se configuran las entidades en JPA? ¿qué elemento de configuración se necesita?

Para que JPA pueda persistir esta entidad, necesita:

- Un archivo de configuración XML llamado *persistence.xml*, el cual debe estar ubicado en el directorio *META-INF* de nuestra aplicación.
- Una instancia de EntityManager a través de la clase factoría EntityManagerFactory

Maven

¿Conoce Maven? ¿Lo ha utilizado? Comente su experiencia

Es elemento generado por líneas de comando que nos ilustra cómo crear nuestros directorios o carpetas para nuestros proyectos que manejan tareas habituales que se realizan (compilado, generación del jar, documentar, distribuir dependencias con archivos .jar, entre otros).

Lo he utilizado a nivel educativo y en proyectos propios de la Universidad

SQL I.

Tenemos una base de datos formada por dos tablas, una llamada "Clientes" donde existe información referente a los datos personales de los clientes de un banco, además de su cuenta corriente.

En otra tabla llamada "Adeudos", tenemos los importes de las compras de los clientes en comercios, pagadas con su tarjeta de crédito. La estructura de ambas tablas es:

CLIENTES

Id_cliente (PK)

Nombre

Apellido1

Apellido02

Telefono

Direccion

Cpostal

Saldo

ADEUDOS

Id_cliente

Importe_adeudo

Se pide:

- 1) Escribe la sentencia SQL para obtener la relación de todos los clientes que sean de la comunidad de Madrid (Códigos postales desde el 28000 al 28999 ambos incluidos).

```
Select * from CLIENTES where Cpostal between 28000 and 289999;
```

```
Select * from CLIENTES where Cpostal >= 28000 and Copstal <= 289999;
```

- 2) Escribe la sentencia SQL para eliminar todos los clientes que tengan saldo negativo (solamente de la tabla "Clientes").

```
Delete from CLIENTES where Saldo < 0
```

- 3) Escribe la sentencia SQL para registrarte como cliente con un saldo inicial de 1.000 € a tu favor.

```
Insert into CLIENTES (Id_cliente, Nombre, Apellido1, Apellido02, Telefono, Direccion, Cpostal, Saldo ) values (200, John Jairo, Zuñiga, Puerta, Kara 0 # 39 .137, 57, 1000)
```

- 4) Por pago de intereses, queremos incrementar en un 4% el saldo de todos nuestros clientes. Escribe la sentencia SQL para que el saldo quede actualizado.

```
Update CLIENTES set Saldo = (Saldo*0,04)+Saldo
```

- 5) Queremos actualizar el saldo de los clientes, descontándole el importe de sus compras (Importe_adeudo). Escribe la sentencia SQL para que el saldo quede actualizado.

Update CLIENTES set Saldo = Saldo – Importe_adeudo where CLIENTE.Id_cliente = ADEUDOS.Id_cliente

- 6) Queremos crear una nueva tabla llamada "Ricos", que contenga los registros de aquellos clientes que han quedado con un saldo mayor de 1.000 €. Escribe la sentencia SQL, que lleve a cabo esta operación.

Create table Ricos AS (select * from CLIENTES where Saldo>1.000);

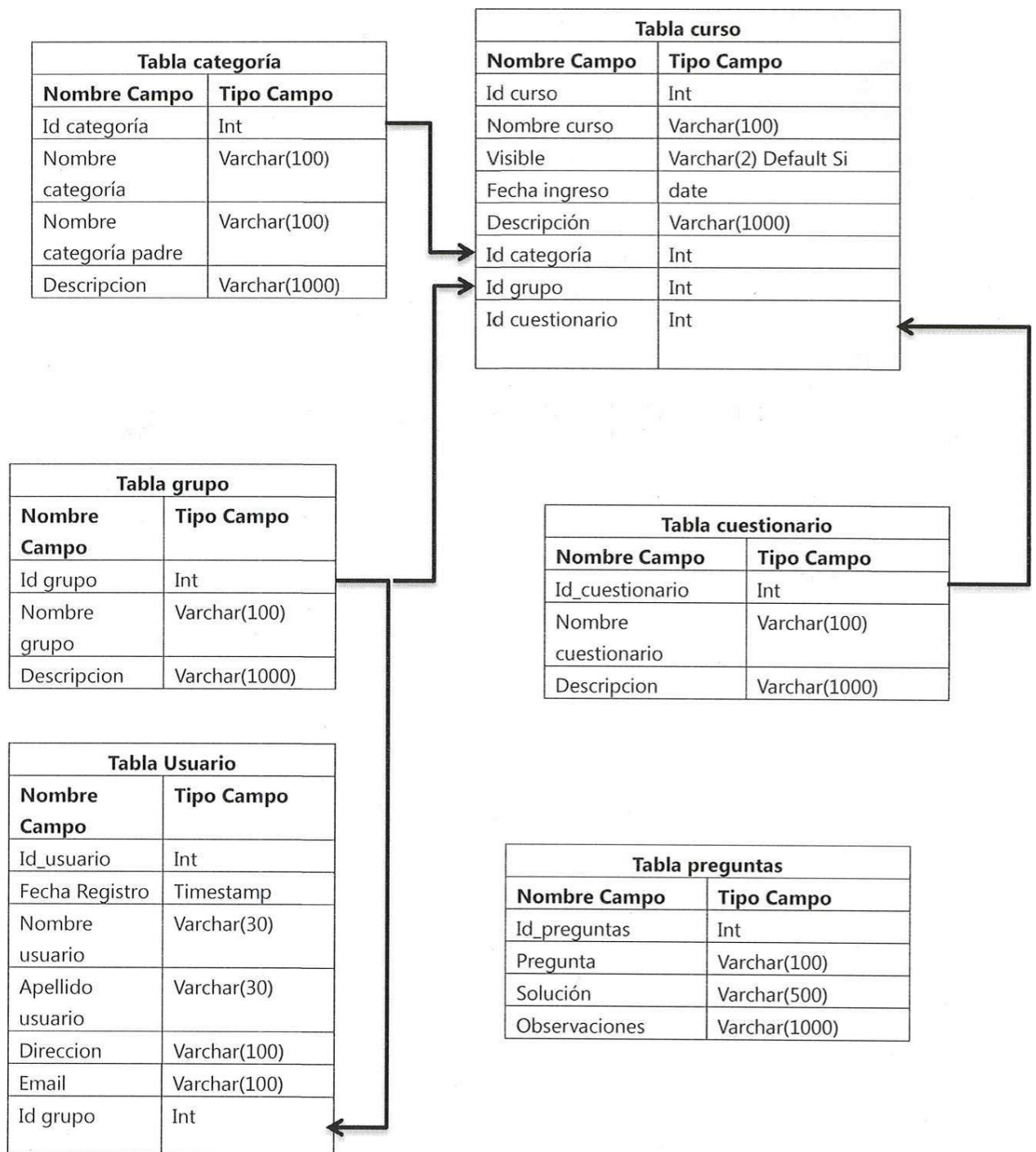
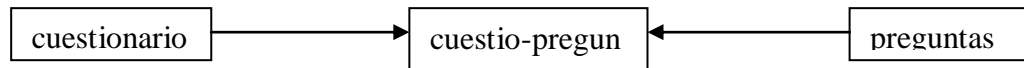


Diagrama SQL I.

Según el diagrama SQL 1:

- 1) ¿Cuál es la relación entre la tabla cuestionario y la tabla preguntas? Teniendo en cuenta que un cuestionario puede tener muchas preguntas y una pregunta puede pertenecer a muchos cuestionarios. Grafique su respuesta.



La relación dada entre las dos tablas es una relación varios a varios por lo que se necesita acudir a una tabla intermedia.

- 2) Realice una consulta donde me devuelva el nombre del grupo y la cantidad de cursos por cada uno de ellos.

```
Select curso.nombre, count(*) from grupo, curso group by id_curso
```

- 3) Realice una consulta en donde esta me devuelva todos los cursos que no tienen una categoría asociada. Realícelo con una subconsulta y relacionando las tablas.

```
Select * from cursos where Id_categoria NOT IN (select Id_categoria from categoría)
```

- 4) Realice una consulta que me devuelva los campos categoría (nombre categoría), curso(nombre curso), cuestionario (Nombre cuestionario) grupo (nombre grupo) de los cursos registrados.

- 5) Teniendo en cuenta la siguiente imagen, realice una consulta que me devuelva todos los usuarios que deben presentar el cuestionario Test 1 y que la fecha de ingreso este entre el 1 de enero de 2017 al día en curso.

Id cuestionario	Nombre cuestionario	Descripcion
1	Test 1	Des 1
2	Test 2	Des 2