PRUEBA # 1
CONCEPTOS DE PROGRAMACIÓN

Fecha: 22/12/21
Nombre y apellidos: Luisa Fernanda Arboleda Segura

CC: 1.045.516.627

Nota: No se permite el uso de teléfonos celulares y/o computadores.

1. (1 Pto) ¿En programación orientada a objetos (POO), a que se refiere el término clase?

R// Una clase en POO es donde se declara, describe o se definen las propiedades y comportamientos de un objeto, es decir como van a ser estos objetos, una clase contiene además métodos y atributos que reúnen las características comunes de estos.

2. (1 Pto) ¿En programación orientada a objetos (POO), a que se refiere el término objeto?

R// En POO un objeto es una entidad que encapsula los atributos y comportamiento de esto, es decir una instancia de una clase.

3. (1 Pto) ¿Cuál es la diferencia entre una clase y un objeto?

R// La clase es un modelo a partir del cual se puede crear la instancia, es decir, objetos. mientras que el objeto es una entidad, una instancia de la clase
a partir del cual puede crear la instancia, es decir, objetos.

4. (1 Pto) ¿Qué es una arquitectura basada en capas?, defina cada uno de sus niveles o capas.

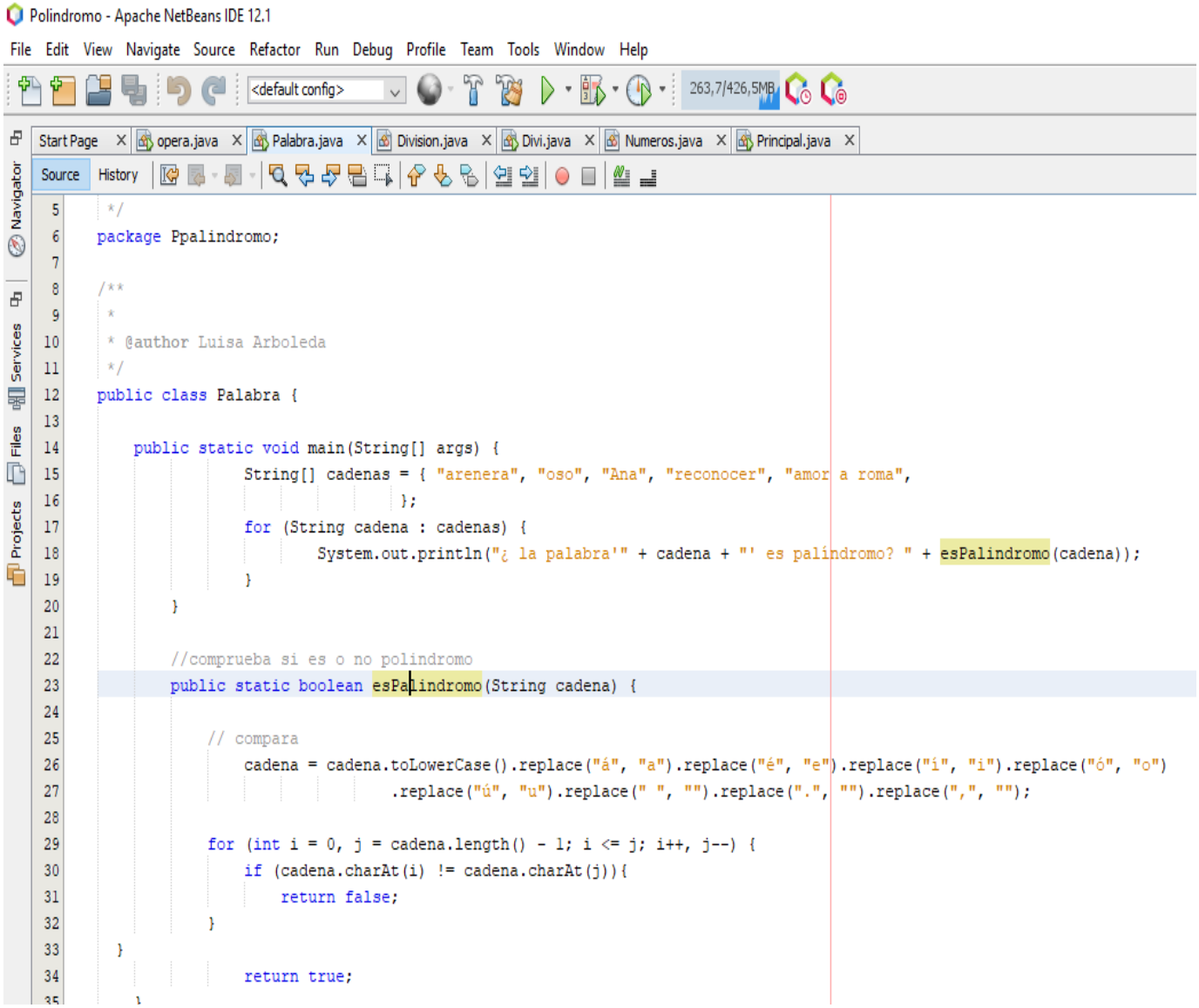
R// Una arquitectura basada en capas es un patrón de arquitectura de software, está divide la aplicación en capas, con el objetivo de que cada capa tenga su rol definido, cada capa tiene una responsabilidad, además las capas deben respetar una estructura jerárquica, lo que quiere decir que cada capa solo puede comunicarse con la que está debajo suyo, la arquitectura basada en capas contiene las siguientes capas: .

- Capa de presentación: Esta capa es la que ve el usuario (interfaz gráfica), la que presenta la información y captura la información del usuario, esta capa se comunica con la capa de negocio.
- Capa de negocio: Es donde se encuentran las funciones que se ejecutan, es decir las

peticiones de usuarios, donde se procesa y se envían respuestas tras su proceso, en esta capa se establecen todas las reglas que deben cumplirse, esta capa se comunica con la capa de presentación.

- Capa de datos: Esta capa se encarga de almacenar los datos del sistema y del usuarios, donde su función principal es almacenar y devolver datos a la capa de negocios.

5. (2 Pto) Una palabra o frase palíndroma es aquella que se lee y escribe de igual forma al derecho y al revés, por ejemplo (arenera, oso, Ana, reconocer, amor a roma), diseñe un algoritmo (**en pseudo-codigo o en el lenguaje programación de su preferencia**) que reciba como parámetro una palabra o frase y retorne un mensaje informando al usuario si la palabra ingresada es palíndroma o no.



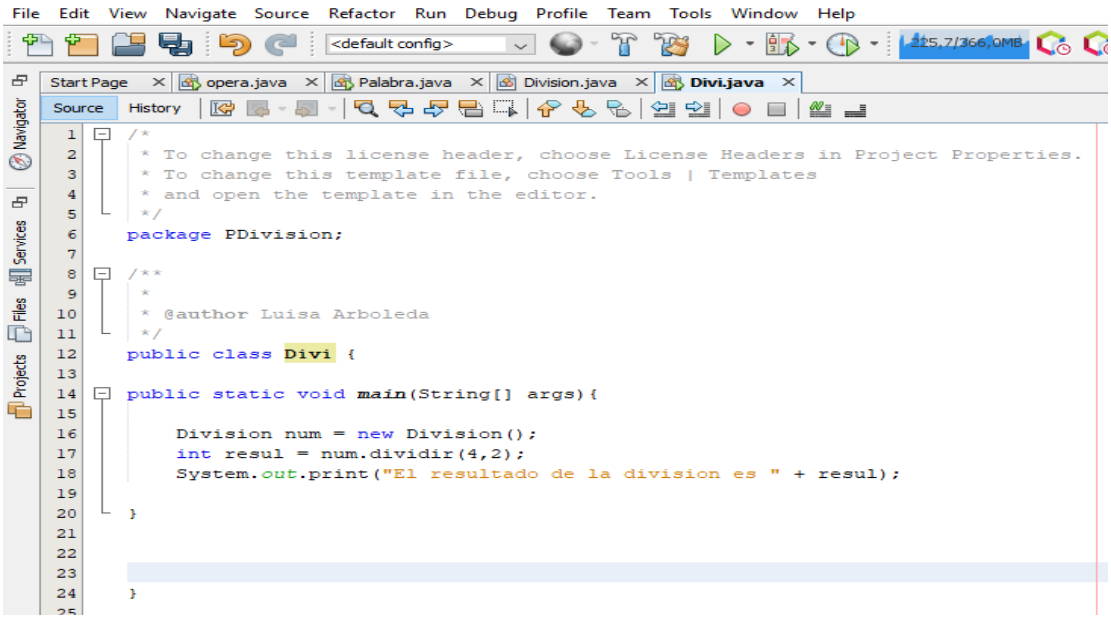
Nota: No se permite el uso de funciones avanzadas como reverse de java u otras. 6. (1 Pto)

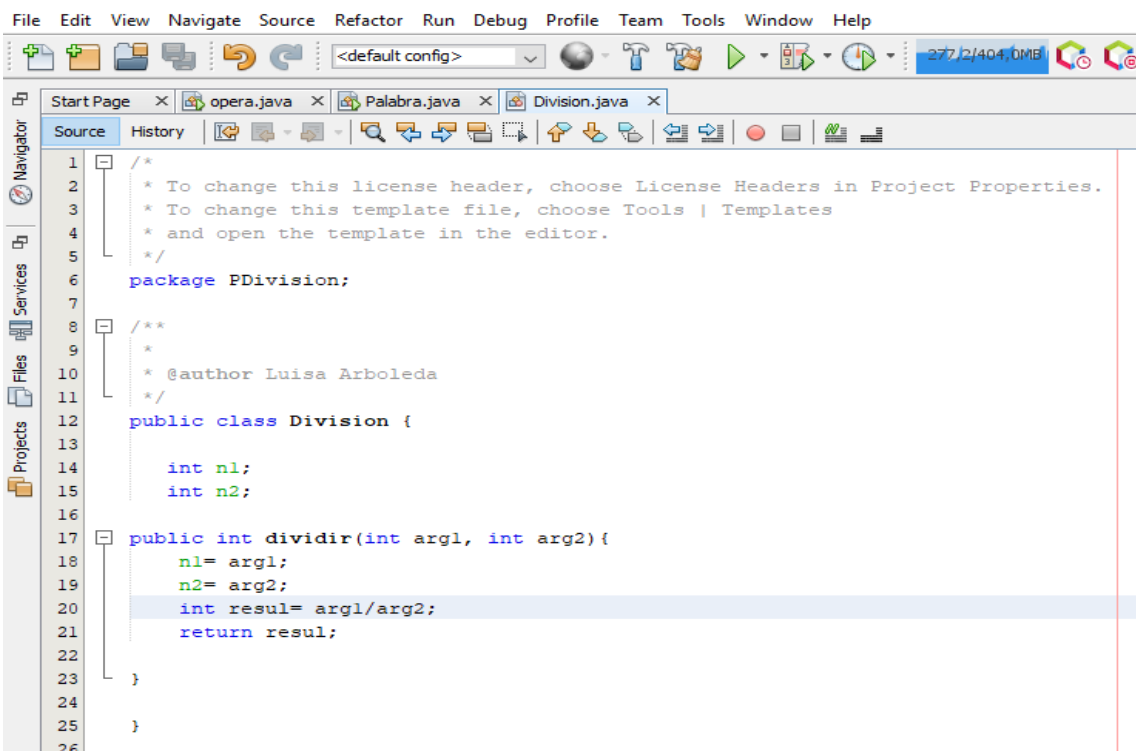
¿Cuál es el resultado de ejecutar el siguiente pseudo-código con el valor “5”?

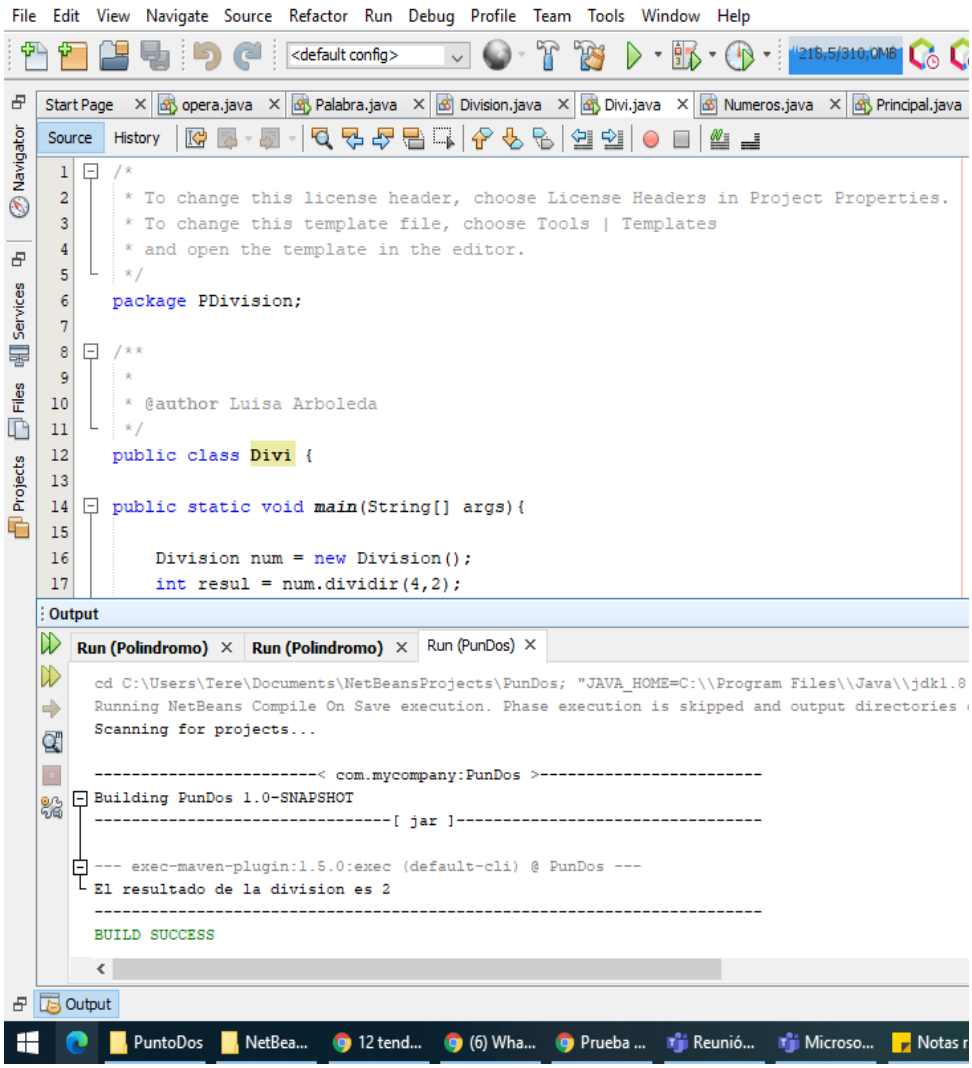
```
funcion metodoQueHaceAlgo(Entero valor)

      si(valor < 3)entonces
            retornar valor;
      fin si

      retornar metodoQueHaceAlgo(valor-1)*metodoQueHaceAlgo(valor-2);

fin funcion
```

R// EL resultado es 8

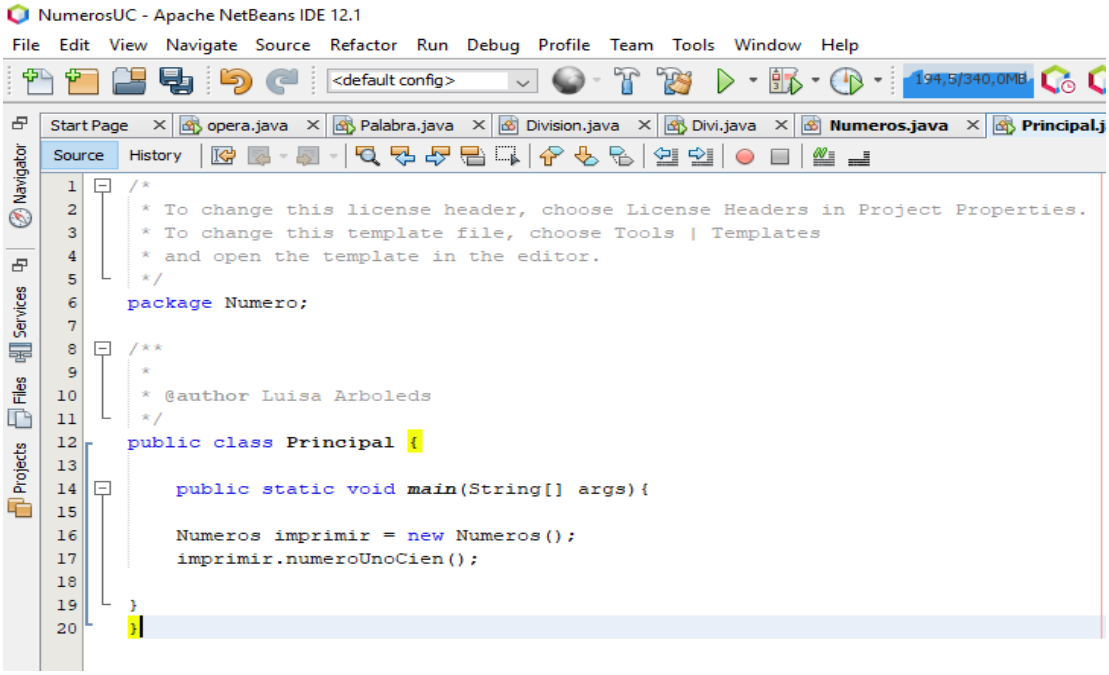
7. (1 Pto) Realice un algoritmo que reciba como parámetro dos números enteros y retorne la división de ambos números.

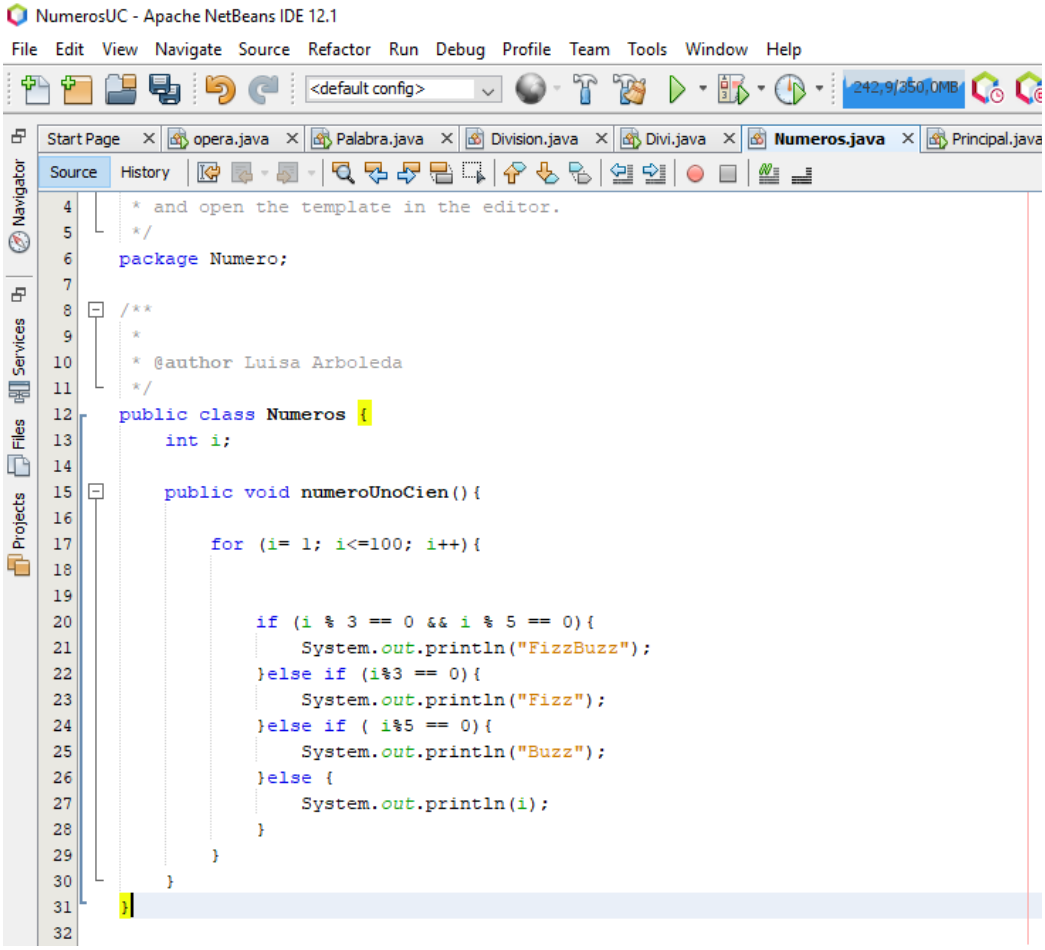


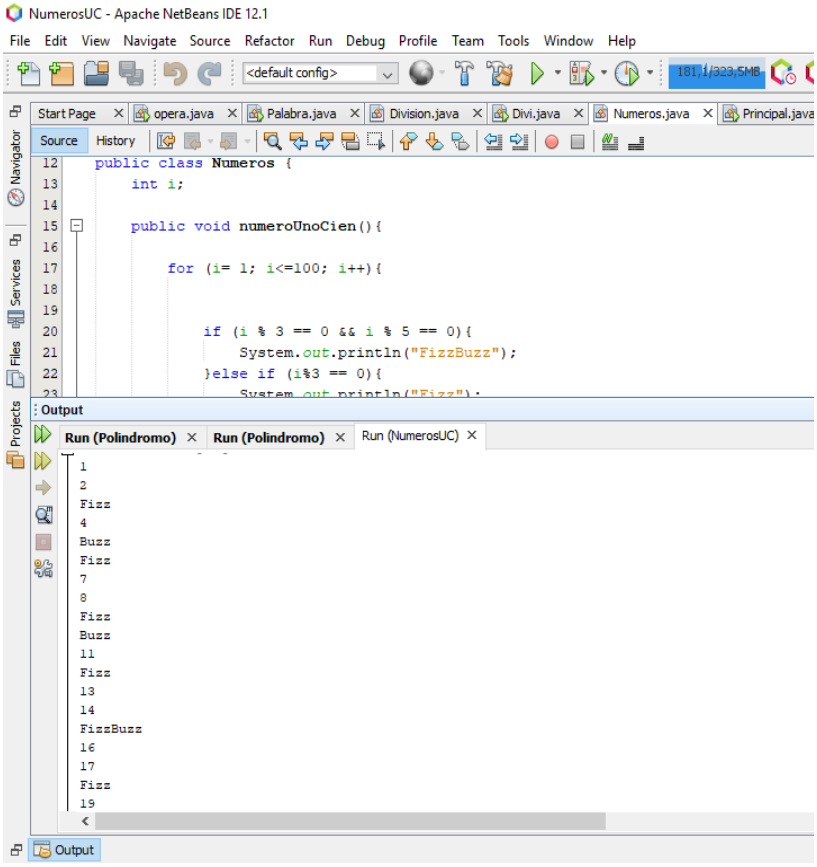

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Start Page
opera.java
Palabra.java
Division.java
Source
History
Navigator
Services
Files
Projects
/*
* To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.
*/
package PDivision;
/**
*
* @author Luisa Arboleda
*/
public class Division {
int n1;
int n2;
public int dividir(int arg1, int arg2){
n1= arg1;
n2= arg2;
int resul= arg1/arg2;
return resul;
}
}



8. (2 Pto) Escriba un algoritmo que imprima los números del 1 al 100. Pero para los múltiplos de 3 imprima "Fizz" en lugar del número y para los múltiplos de 5 imprima "Buzz". Para los números que son múltiplos de ambos imprima "FizzBuzz".


NumerosUC - Apache NetBeans IDE 12.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
194,5/340,0MB
Start Page
opera.java
Palabra.java
Division.java
Divi.java
Numeros.java
Principal.j
Source
History
Navigator
Services
Files
Projects
1 /*
2 * To change this license header, choose License Headers in Project Properties.
3 * To change this template file, choose Tools | Templates
4 * and open the template in the editor.
5 */
6 package Numero;
7
8 /**
9 *
10 * @author Luisa Arboleds
11 */
12 public class Principal {
13
14 public static void main(String[] args){
15
16 Numeros imprimir = new Numeros();
17 imprimir.numeroUnoCien();
18
19 }
20 }


NumerosUC - Apache NetBeans IDE 12.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
242,9/350,0MB
Start Page
opera.java
Palabra.java
Division.java
Divi.java
Numeros.java
Principal.java
Source
History
Navigator
Services
Files
Projects
4 * and open the template in the editor.
5 */
6 package Numero;
7
8 /**
9 *
10 * @author Luisa Arboleda
11 */
12 public class Numeros {
13 int i;
14
15 public void numeroUnoCien(){
16
17 for (i= 1; i<=100; i++){
18
19
20 if (i % 3 == 0 && i % 5 == 0){
21 System.out.println("FizzBuzz");
22 }else if (i%3 == 0){
23 System.out.println("Fizz");
24 }else if (i%5 == 0){
25 System.out.println("Buzz");
26 }else {
27 System.out.println(i);
28 }
29 }
30 }
31 }
32



9. Nombre tres tendencias actuales en el área de software.

R//

1. Inteligencia Artificial (IA)
2. Blockchain
3. Realidad aumentada

PRUEBA # 2
BASES DE DATOS

Fecha: 22-12-2021
Nombre y apellidos: Luisa Fernanda Arboleda Segura

CC: 1.045.516.627

Nota: No se permite el uso de teléfonos celulares y/o computadores.

1. (1 Pto) ¿Qué es una “primary key” o clave principal o primaria?

R// Es la clave primaria que se asigna a un dato de la tabla y se identifica de manera única dentro de la tabla.

2. (1 Pto) ¿Qué son las “foreign keys” o claves externas o foráneas?

R//Un foreign key es una clave foránea que sirve para relacionar dos tablas.

3. (1 Pto) ¿Qué es un stored procedure o procedimiento almacenado? Es un conjunto de instrucciones o programas que se encuentra almacenado en la base de datos, el cual puede ser ejecutado cuando se desee.

La empresa “Mascotas & Mascotas”, rescata animales que han sido abandonados, los rehabilitan y cuando están totalmente sanos, los ponen a disposición para que estos sean adoptados, se tienen las siguientes entidades:

Tabla mascotas
Código
Nombre
Tipo mascota

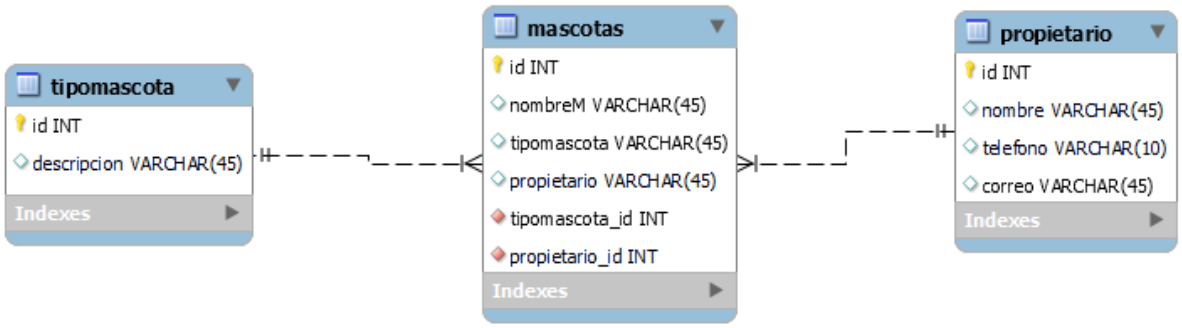
Código propietario

Tabla Propietarios
Código
Nombre

Tabla Tipo mascota
Código
Descripción

4. (2 Pto) Elabore el modelo entidad relación lo más detallado posible (relaciones, tipos de datos, llaves primarias y foráneas, etc…), si ud. lo considera necesario, puede adicionar más entidades o tablas.

R//



De acuerdo al modelo ER que Ud. acaba de elaborar, escriba las siguientes instrucciones SQL:

5. (1 Pto) Listar todas las mascotas.

```sql
/*Listar todas las mascotas*/
SELECT  mascota.id, nombreM, tipomascota.descripcion
 FROM dbmascotas.mascota
INNER JOIN dbmascotas.tipomascota
ON  mascota.tipomascota_id = tipomascota.id ;
```

6. (1 Pto) Listar las mascotas que no han sido adoptadas.

```sql
/* Listar las mascotas que no han sido adoptadas.*/
SELECT  mascota.id, nombreM, tipomascota.descripcion
 FROM dbmascotas.mascota
 LEFT JOIN dbmascotas.tipomascota
ON  mascota.tipomascota_id = tipomascota.id
 LEFT JOIN dbmascotas.propietario
ON  mascota.propietario_id = propietario.id
WHERE mascota.propietario_id  IS NULL;
```

7. (1 Pto) Listar el número de mascotas por cada tipo de mascota.

```sql
/*Listar el número de mascotas por cada tipo de mascota*/
SELECT DISTINCT tipomascota.*, COUNT(*) AS mascotas_cnt
 FROM dbmascotas.mascota m
  INNER JOIN dbmascotas.tipomascota
ON m.tipomascota_id = tipomascota.id
GROUP BY tipomascota.id;
```

8. (1 Pto) Listar los propietarios que tengan más de una mascota.

```sql
--
/*Listar los usuarios que tengan mas de una mascota*/
SELECT DISTINCT propietario.*, COUNT(*) AS mascotas_cnt
FROM dbmascotas.mascota m
INNER JOIN dbmascotas.propietario
ON m.propietario_id = propietario.id
GROUP BY propietario.id
HAVING COUNT(mascotas_cnt) > 1;
--
```

9. (1 Pto) Listar el número de mascotas por cada tipo de mascota y por cada propietario.

```sql
/*Listar el número de mascotas por cada tipo de mascota y por cada propietario.*/

SELECT DISTINCT tipomascota.*, propietario.*, COUNT(*) AS mascotas_cnt
 FROM dbmascotas.mascota m
  LEFT JOIN dbmascotas.tipomascota
ON m.tipomascota_id = tipomascota.id
 LEFT JOIN dbmascotas.propietario
ON  m.propietario_id = propietario.id
WHERE  m.propietario_id = propietario.id
GROUP BY  tipomascota.id
ORDER BY tipomascota.id ASC ;
```

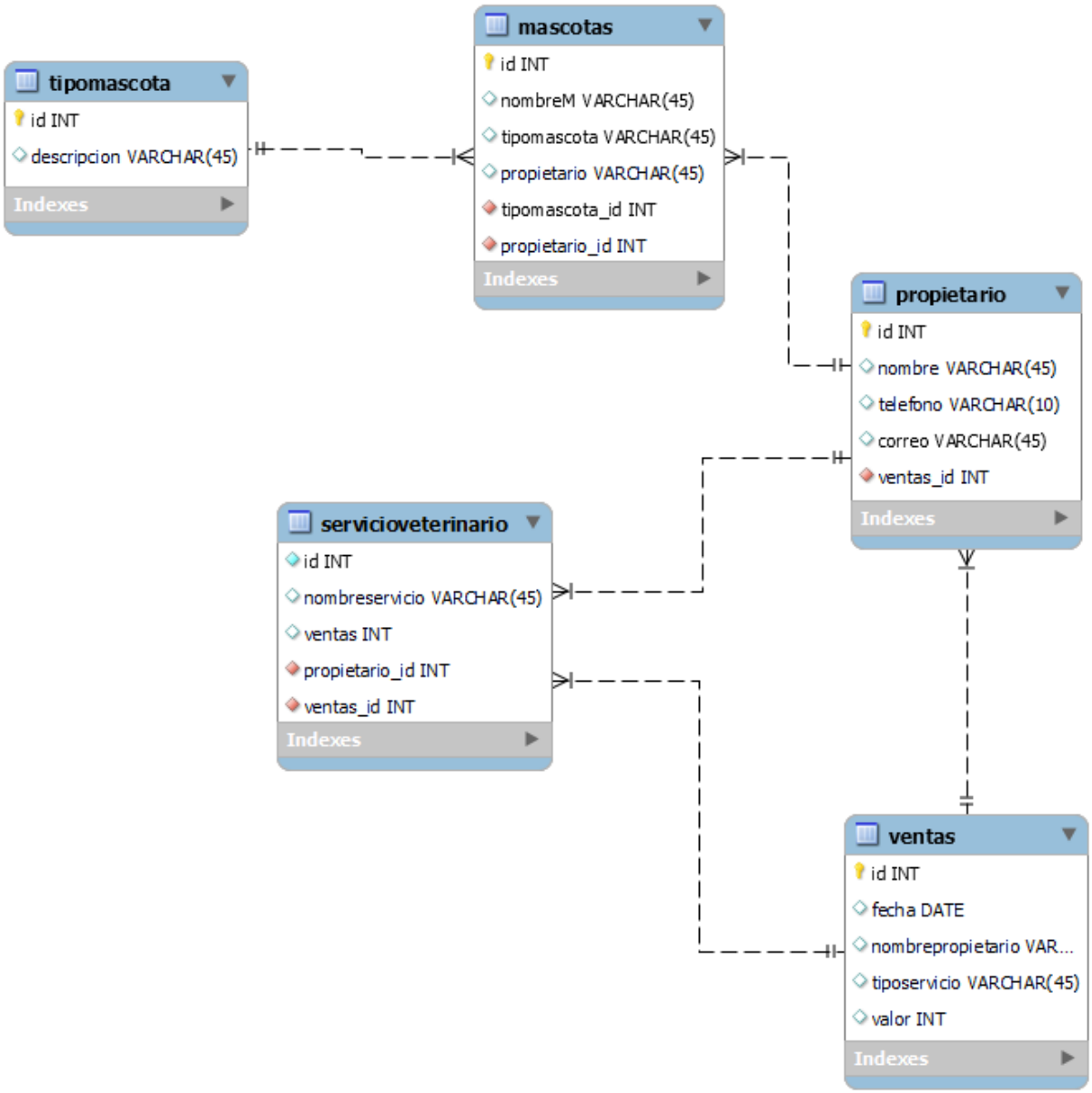
10. (1 Pto) Listas los propietarios que no tienen mascotas.

```sql
/*Listar los propietarios que no tienen mascotas.*/

SELECT DISTINCT propietario.id, propietario.nombre, propietario.comentarios
 FROM dbmascotas.mascota  JOIN dbmascotas.propietario
ON mascota.propietario_id  != propietario.id
ORDER BY propietario.id ASC;
```

11. (1 Pto) Cree un stored procedure o procedimiento almacenado con cualquiera de las sentencias anteriores.

```
155     --
156     /* Procedimiento almacenado*/
157     drop procedure if exists listar_mascotas;
158     delimiter //
159     CREATE PROCEDURE listar_mascotas()
160     BEGIN
161     SELECT  mascota.id, nombreM, tipomascota.descripcion
162      FROM dbmascotas.mascota
163     INNER JOIN dbmascotas.tipomascota
164     ON  mascota.tipomascota_id = tipomascota.id ;
165     END //
166     delimiter ;
167     CALL listar_mascotas();
168
```

12. (2 Pto) La empresa “Mascotas & Mascotas”, además de prestar el servicio de adopción, también presta el servicio de veterinaria al público, la empresa desea conocer cuáles son las ventas mensuales por cada propietario. ¿Qué cambios le realizaría al modelo entidad relación para cumplir con este requisito?. Realice la sentencia SQL para obtener esta información.



13. (1 Pto) Escriba una sentencia SQL que actualice la columna “Nombre” de la tabla mascota, de modo que los nombres de todas las mascotas queden en mayúscula

```
168
169     /* Actualizar columna nombre en mayuscula*/
170 •   UPDATE dbmascotas.mascota
171     set nombreM = UPPER(nombreM)
172
```

14. (1 Pto) Escriba una sentencia SQL que elimine a los propietarios que no tienen mascotas.

```
160     /*eliminar a los propietarios que no tienen mascotas.*/
161 •   DELETE FROM dbmascotas.propietario
162     WHERE mascota.propietario_id  != propietario.id
163
```

15. (1 Pto) Explique las siguientes funciones de agregación:

a. COUNT() : Esta función nos da el total de filas seleccionadas.
b. MAX() : Esta función nos da el valor máximo de la columna
c. MIN(): Esta función nos da el valor mínimo de la columna