

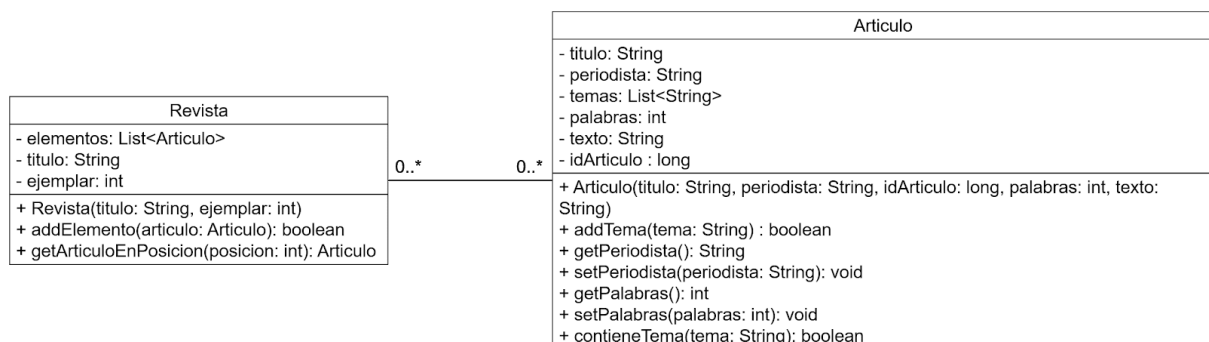
Examen 111Mil – Administrador de Revistas y Artículos

Luis es el editor de la revista “The Post”. Al principio Luis podía administrar todos los artículos de forma manual, pero debido al éxito que la revista ha adquirido en los últimos tiempos, Luis necesita una forma fácil de mantener ordenados los artículos y contribuciones de sus periodistas, para saber cuántos artículos tiene en un determinado tema, cuántos artículos hizo un determinado periodista. Por esta razón, Luis necesita desarrollar un sistema que lo ayude en su negocio.

Como sabe que cursamos el programa 111Mil se puso en contacto con nosotros para que lo ayudemos a construir el sistema, que en esta primera etapa del proyecto deberá registrar los artículos que le envían los periodistas y los ejemplares de revistas.

Ejercicio 1. Implementar desde el diagrama de clases

Dados el siguiente diagrama y código, escriba en Java SOLO los fragmentos de código faltantes que se indican:



En el siguiente código, completar los fragmentos de código faltantes en donde está indicado para:

- Declarar los atributos elementos, título y ejemplar de la clase Revista.
- Implementar el método `getArticuloEnPosicion` de la clase Revista. Tener en cuenta que si la posición pasada por parámetro no existe (la posición excede la cantidad de elementos o es menor que cero), se debe retornar `null`.
- Implementar los métodos `getPalabras`, `setPalabras`, `getPeriodista` y `setPeriodista` de la clase Articulo.

```
import java.util.ArrayList;
import java.util.List;

public class Revista {

    private List<Articulo> elementos;
    private String titulo;
    private int ejemplar;

    public Revista(String titulo, int ejemplar) {
        this.titulo = titulo;
        this.ejemplar = ejemplar;
        this.elementos = new ArrayList<Articulo>();
    }
}
```

```

    }

    public boolean addElemento(Articulo articulo){
        return elementos.add(articulo);
    }

    public Articulo getArticuloEnPosicion(int posicion){
        if(posicion >= elementos.size() || posicion < 0)
            return null;
        return elementos.get(posicion);
    }
}

```

```

import java.util.ArrayList;
import java.util.List;

public class Articulo {

    //NO complete las declaraciones de atributos

    public Articulo(String titulo, String periodista, int palabras,String texto,
long idArticulo) {
        this.titulo = titulo;
        this.periodista = periodista;
        this.palabras = palabras;
        this.texto = texto;
        this.idArticulo = idArticulo;
        this.temas = new ArrayList<String>();
    }

    public boolean addTema(String tema){
        return temas.add(tema);
    }

    public boolean contieneTema(String tema){
        return temas.contains(tema);
    }

    public String getPeriodista() {
        return periodista;
    }

    public void setPeriodista(String periodista) {
        this.periodista = periodista;
    }

    public int getPalabras() {
        return palabras;
    }

    public void setPalabras(int palabras) {
        this.palabras = palabras;
    }
}

```

Ejercicio 2. Implementar un método a partir de un enunciado

A Luis le gustó la primera implementación del programa que le resuelve sus problemas. Pero se le ocurrió otra funcionalidad que quiere agregar al sistema:

- Determinar la cantidad de artículos de un determinado tema en la revista. El tema debe ser un String pasado por parámetro y la cantidad debe retornarse como un int.

Codifique en Java todos los métodos necesarios de la clase Revista y de la clase Artículo para satisfacer el nuevo requisito de Luis. Por cada método, indique en qué clase debe implementarse el mismo.

En la clase Revista.

```
public int getCantidadArticulosTema(String tema){
    int cant = 0;
    for(Articulo a : elementos)
        if(a.contieneTema(tema))
            cant++;
    return cant;
}
```

Ejercicio 3. Determine las opciones correctas.

Finalmente, considerando todo lo implementado, Luis empezó a armar su colección y escribió el siguiente programa, pero no usó nombres representativos. Indique cuál de las opciones indica lo que hace el código:

```
public float metodoX(String tema){
    float cant = 0;
    float sumaPalabras = 0;

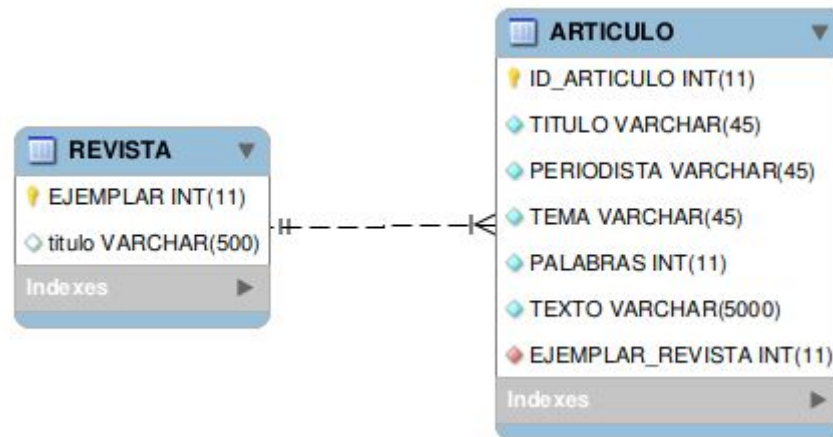
    for(Articulo a : elementos)
        if(a.contieneTema(tema)){
            cant++;
            sumaPalabras += a.getPalabras();
        }

    if(cant == 0)
        return 0;

    return sumaPalabras / cant;
}
```

	Cantidad promedio de palabras de todos los artículos de una revista.
X	Cantidad promedio de palabras de los artículos de una revista de un tema particular.
	Porcentaje de palabras de los artículos de una revista de un tema particular.
	Ninguna de las anteriores es correcta.

Ejercicio 4. Interpretación de Esquema de Tablas.



Dado el siguiente DER, determine si las siguientes afirmaciones son verdaderas (V) o falsas (F). Asigne a CADA afirmación un valor V o F. Todas las afirmaciones deben tener un valor asignado. Puede haber varios V y varios F.

F	Entre las tablas REVISTA y ARTICULO existe una relación 1 a 1.
V	Entre las tablas REVISTA y ARTICULO existe una relación 1 a N.
F	Entre las tablas REVISTA y ARTICULO existe una relación N a N.
V	La clave primaria de la tabla ARTICULO es idArticulo.
F	La clave foránea de la tabla ARTICULO es tema.
F	La tabla REVISTA no tiene clave primaria.
V	La tabla REVISTA no tiene clave foránea.

Ejercicio 5. Consultas SQL.

Considerando el DER anterior, se desean obtener los nombres de artículos de política y el número de ejemplar en el que fue publicado, si dicho artículo tiene al menos 200 palabras. Los artículos deben ser ordenados de acuerdo al ejemplar en el que fueron publicados de forma descendente. Escriba la consulta SQL de acuerdo a lo especificado.

```
select articulo.ejemplar_revista, articulo.titulo
from articulo
where articulo.palabras >= 200 and articulo.tema = 'politica'
order by articulo.ejemplar_revista desc
```