# Programación 2

# Tecnicatura en Desarrollo de Aplicaciones Informáticas



#### Programación 2 TUDAI - Parcial 26-10-2022

Extra, extra! Teniendo en cuenta todos los principios de la Programación Orientada a Objetos, defina las clases necesarias para modelar el siguiente problema e implemente en Java.

a) Modelar un sistema que permita la organización de noticias de un portal electrónico. De cada noticia se guarda: el contenido (texto), el título, el autor, la categoría ("policial", "espectáculos", etc, solo tiene una única categoría) y una lista de palabras claves. Las noticias del portal se organizan en una estructura de secciones, subsecciones, grupos y subgrupos. Una noticia puede estar contenida dentro un subgrupo, grupo, subsección o sección. La categoría de cada sección / subsección se determina por la categoría del elemento en la posición 4, mientras que para los grupos y subgrupos es la posición 5 (estas posiciones no son fijas sino que pueden cambiar, y si no hay elementos suficientes siempre retornan "Sin



definir"). Por otro lado, las palabras claves de una sección/subsección/grupo/subgrupo, se calculan como la

## Punto a)

Modelar un sistema que permita la organización de noticias de un portal electrónico. De cada noticia se guarda: el contenido (texto), el título, el autor, la categoría ("policial", "espectáculos", etc, solo tiene una única categoría) y una lista de palabras claves. Las noticias del portal se organizan en una estructura de secciones, subsecciones, grupos y subgrupos. Una noticia puede estar contenida dentro un subgrupo, grupo, subsección o sección. La categoría de cada sección / subsección se determina por la categoría del elemento en la posición 4, mientras que para los grupos y subgrupos es la posición 5 (estas posiciones no son fijas sino que pueden cambiar, y si no hay elementos suficientes siempre retornan "Sin definir"). Por otro lado, las palabras claves de una sección /subsección/ grupo/subgrupo, se calculan como la unión de todas las palabras claves de los elementos que contiene, sin incluir palabras repetidas

### Punto a)

Modelar un sistema que permita la organización de noticias de un portal electrónico. De cada noticia se guarda: el contenido (texto), el título, el autor, la categoría ("policial", "espectáculos", etc, solo tiene una única categoría) y una lista de palabras claves. Las noticias del portal se organizan en una estructura de secciones, subsecciones, grupos y subgrupos. Una noticia puede estar contenida dentro un subgrupo, grupo, subsección o sección. La categoría de cada sección / subsección se determina por la categoría del elemento en la posición 4, mientras que para los grupos y subgrupos es la posición 5 (estas posiciones no son fijas sino que pueden cambiar, y si no hay elementos suficientes siempre retornan "Sin definir"). Por otro lado, las palabras claves de una sección /subsección/ grupo/subgrupo, se calculan como la unión de todas las palabras claves de los elementos que contiene, sin incluir palabras repetidas

# Punto a)

Modelar un sistema que permita la organización electrónico. De cada **noticia** se guarda: **el cont autor**, **la categoría** ("policial", "espectáculos" **categoría**) y una **lista de palabras claves**. L organizan en una estructura de secciones, subse Una noticia puede estar contenida dentro un su sección. La categoría de cada sección / subs

(C) Noticia ArrayList<String> palabrasClaves String contenido String titulo String autor String categoria String getContenido() void setContenido(String s) String getTitulo() void setTitulo(String s) String getCategoria() void setCoategoria(String s) String getAutor() void setAutor(String s)

rtal

ica

se

)OS.

n o

el

sección. La categoría de cada sección / subs la categoría del elemento en la posición 4, mientras que para los grupos y subgrupos es la posición 5 (estas posiciones no son fijas sino que pueden cambiar, y si no hay elementos suficientes siempre retornan "Sin definir"). Por otro lado, las palabras claves de una sección /subsección/ grupo/subgrupo, se calculan como la unión de todas las palabras claves de los elementos que contiene, sin incluir palabras repetidas

# Punto a)

Modelar un sistema que permita la organización electrónico. De cada **noticia** se guarda: **el contautor**, **la categoría** ("policial", "espectáculos" **categoría**) y una **lista de palabras claves**. Lorganizan en una estructura de secciones, subse Una noticia puede estar contenida dentro un su sección la categoría de cada sección / subs

ncluir palabras repetidas

(C) Noticia ArrayList<String> palabrasClaves String contenido String titulo String autor String categoria String getContenido() void setContenido(String s) String getTitulo() void setTitulo(String s) String getCategoria() void setCoategoria(String s) String getAutor() void setAutor(String s)

rtal

ica

se

)OS.

n o

la

el

elemento en la posición 4, mientras que para los grupos y a posición 5 (estas posiciones no son fijas sino que pueden no hay las pa ¿Qué método faltó? pre retornan "Sin definir"). /subsección/ grupo/subgrupo, no la unión de todas las palabras claves de los elementos que

# Punto a)

Modelar un sistema que permita la organizació electrónico. De cada **noticia** se guarda: **el cont** autor, la categoría ("policial", "espectáculos" categoría) y una lista de palabras claves. L organizan en una estructura de secciones, subse Una noticia puede estar contenida dentro un su sección. La categoría de cada sección / subs categoría del elemento en la posición 4, mientras que para los grupos y

(C) Noticia ArrayList<String> palabrasClaves String contenido String titulo String autor rtal String categoria String getContenido() void setContenido(String s) ica String getTitulo() void setTitulo(String s) String getCategoria() )OS. void setCoategoria(String s) String getAutor() n o void setAutor(String s)

el

se

la

subgrupos es la posición 5 (estas posiciones no son fijas sino que pueden cambiar, y si no siempre retornan "Sin definir"). ¡Las palabras claves! ción /subsección/ grupo/subgrupo, como la unión de todas las palabras claves de los elementos que n incluir palabras repetidas

#### Manejo de Arraylist palabrasClaves

- \_\_\_\_
- No hay que romper el encapsulamiento!
- El ArrayList se crea en el constructor
  - Nadie fuera de la clase lo conoce

```
public ArrayNist<String> getPalabrasClaves() {
    return palabrasClaves;
}
public void setPalabrasClaves(ArrayList(String> pals) {
    return palabrasClaves;
```

#### C Noticia

- Nocicia
- ArrayList<String> palabrasClaves
- String contenido
- String titulo
- String autor
- String categoria
- String getContenido()
- void setContenido(String s)
- String getTitulo()
- void setTitulo(String s)
- String getCategoria()
- void setCoategoria(String s)
- String getAutor()
- void setAutor(String s)



**ROTURA DE ENCAPSULAMIENTO** 

Cualquiera desde afuera puede manipular el ArrayList sin pasar por la clase

#### Manejo de Arraylist palabrasClaves

- \_\_\_\_
- No hay que romper el encapsulamiento!
- El ArrayList se crea en el constructor
  - Nadie fuera de la clase lo conoce

#### Devolver el ArrayList

```
public ArrayList<String> getPalabrasClaves() {
    return new ArrayList<String> (palabrasClaves);
}

public ArrayList<String> getPalabrasClaves() {
    ArrayList<String> aux = new ArrayList<String>();
    aux.addAll(palabrasClaves);
    return aux;
}

public ArrayList<String> getPalabrasClaves() {
    Ar ayxist<String> aux = palabrasClaves;
    return aux;
}
```

#### c) Noticia

- ArrayList<String> palabrasClaves
- String contenido
- String titulo
- String autor
- String categoria
- String getContenido()
- void setContenido(String s)
- String getTitulo()
- void setTitulo(String s)
- String getCategoria()
- void setCoategoria(String s)
- String getAutor()
- void setAutor(String s)

Crea un Arreglo nuevo y le agrega las palabras claves

Crea una Variable nueva y "APUNTA" al mismo arreglo

#### Manejo de Arraylist palabrasClaves

- \_\_\_
- No hay que romper el encapsulamiento!
- El ArrayList se crea en el constructor
  - Nadie fuera de la clase lo conoce

Agregar al ArrayList

```
palabrasClaves.add(s)
```

public void addPalabrasClaves(String s) {
 if (!palabrasClaves.contains(s) ) {

- C Noticia
- ArrayList<String> palabrasClaves
- String contenido
- String titulo
- String autor
- String categoria
- String getContenido()
- void setContenido(String s)
- String getTitulo()
- void setTitulo(String s)
- String getCategoria()
- void setCoategoria(String s)
- String getAutor()
- void setAutor(String s)

- + Controlo repetidos
- + No implementó equals porque se usa el de String

# Punto a)

Modelar un sistema que permita la organizació electrónico. De cada **noticia** se guarda: **el cont autor**, **la categoría** ("policial", "espectáculos" **categoría**) y una **lista de palabras claves**. I organizan en una estructura de secciones, subse Una noticia puede estar contenida dentro un su sección. La categoría de cada sección / subsecategoría del elemento en la posición / mien

#### C Noticia

- ArrayList<String> palabrasClaves
- String contenido
- String titulo
- String autor
- String categoria
- String getContenido()
- void setContenido(String s)
- String getTitulo()
- void setTitulo(String s)
- String getCategoria()
- void setCoategoria(String s)
- String getAutor()
- void setAutor(String s)
- ArrayList<String> getPalabrasClaves()
- void addPlabraClave(String s)

categoría del elemento en la posición 4, miencias que para cos grupos y subgrupos es la posición 5 (estas posiciones no son fijas sino que pueden cambiar, y si no hay elementos suficientes siempre retornan "Sin definir"). Por otro lado, las palabras claves de una sección /subsección/ grupo/subgrupo, se calculan como la unión de todas las palabras claves de los elementos que contiene, sin incluir palabras repetidas

### Punto a)

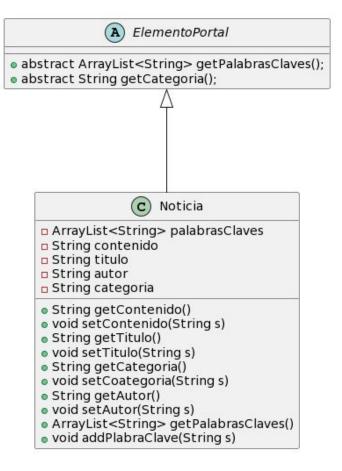
Modelar un sistema que permita la organización de noticias de un portal electrónico. De cada noticia se guarda: el contenido (texto), el título, el autor, la categoría ("policial", "espectáculos", etc, solo tiene una única categoría) y una lista de palabras claves. Las noticias del portal se organizan en una estructura de secciones, subsecciones, grupos y subgrupos. Una noticia puede estar contenida dentro un subgrupo, grupo, subsección o sección. La categoría de cada sección / subsección se determina por la categoría del elemento en la posición 4, mientras que para los grupos y subgrupos es la posición 5 (estas posiciones no son fijas sino que pueden cambiar, y si no hay elementos suficientes siempre retornan "Sin definir"). Por otro lado, las palabras claves de una sección /subsección/ grupo/subgrupo, se calculan como la unión de todas las palabras claves de los elementos que contiene, sin incluir palabras repetidas

## Abstracción principal

Elemento Común

Clase abstracta

**ElementoPortal** 

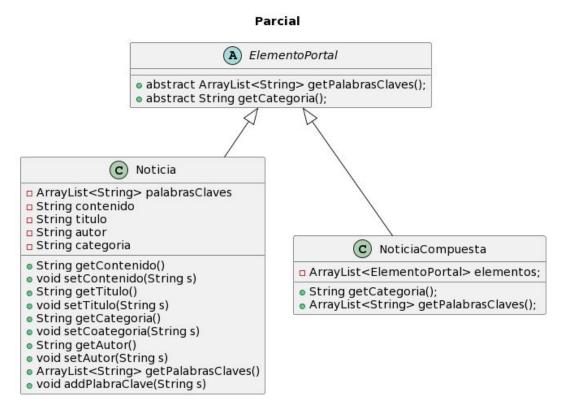


### Abstracción principal

Elemento Común

Clase abstracta

**ElementoPortal** 



## Abstracción principal

```
public abstract class ElementoPortal{

Elemento Común

public abstract String getCategoria();
public abstract ArrayList<String> getpalabrasClaves();

Clase abstracta

ElementoPortal

}
```

#### **Grandes NO**

 No hay atributos en común hasta el momento! solo dos métodos

```
public abstract class ElementoPortal{
   String categoria;
   public abstract String getCategoria();
   public abstract ArrayList<String> getpalabrasClaves();
}
```

#### **Grandes NO**

 No hay atributos en común hasta el momento! solo dos métodos

```
public abstract class ElementoPortal{
   String categoria;
   Stirng titulo;
   String autor;
   ....
   public abstract String getCategoria();
   public abstract ArrayList<String> getpalabrasClaves();
}
```

#### **Grandes NO**

 No hay atributos en común hasta el momento! solo dos métodos

```
public abstract class ElementoPortal{
   String categoria;
   Stirng titulo;
   String autor;
   ArrayList<String> palabrasClaves;
   ....
   public abstract String getCategoria();
   public abstract ArrayList<String> getpalabrasClaves();
}
```



## Abstraer atributos que no son comunes

- Solo abstraen los atributos y no los métodos. ¿Qué sentido tiene?
- Si esta el atributo normalmente puede estar el get y el set
- Atributos y dejar los métodos abstractos? no



## **NoticiaCompuesta**

Las noticias del portal se organizan en una estructura de secciones, subsecciones, grupos y subgrupos. Una noticia puede estar contenida dentro un subgrupo, grupo, subsección o sección. La categoría de cada sección / subsección se determina por la categoría del elemento en la posición 4, mientras que para los grupos y subgrupos es la posición 5 (estas posiciones no son fijas sino que pueden cambiar, y si no hay elementos suficientes siempre retornan "Sin definir")

```
public class NoticiaCompuesta extends ElementoPortal{
                 private ArrayList<ElementoPortal> elementos;
Las noticia:
                 public String getCategoria(){
                                                                          secciones,
                     if (4 < elementos.size())
subsecciones
                                                                        da dentro un
                         return elementos.get
                                                 .getCategoria();
                                                                          sección
subgrupo, g
                     } else {
subsección s
                         return "Sin Definir"
                                                                        posición 4,
                                                                        osiciones no
mientras que
son fijas sino que pueden cambiar, y si no hay elementos suficientes siempre
retornan "Sin definir")
            public class NoticiaCompuesta extends ElementoPortal{
              private ArrayList<ElementoPortal> elementos;
                                                                              NO
              public String getCategoria() {
                  if (5 < elementos.size())
                      return elementos.get
                                                    tegoria();
                  } else {
                      return "Sin Definir"
```

```
public class NoticiaCompuesta extends ElementoPortal{
  private ArrayList<ElementoPortal> elementos;
  private int posicion;
  private String defecto;

public String getCategoria() {
    if (posicion < elementos.size()) {
        return elementos.get(posicion).getCategoria();
    } else {
        return defecto;
    }
}</pre>
```

Cada instancia puede devolver algo distinto

```
public class NoticiaCompuesta extends ElementoPortal{
  private ArrayList<ElementoPortal> elementos;
  private int posicion;
  private static String defecto;

public String getCategoria() {
    if (posicion < elementos.size()) {
        return elementos.get(posicion).getCategoria();
    } else {
        return defecto;
    }
}</pre>
```

Todas las instancias devuelven el mismo valor y si cambia todas cambian

```
public class NoticiaCompuesta extends ElementoPortal{
 private ArrayList<ElementoPortal> elementos;
 private int posicion;
 private static final String defecto="Sin Definir";
 public String getCategoria() {
     if (posicion < elementos.size()) {
         return elementos.get(posicion).getCategoria();
     } else {
        return defecto;
Sigue constante en código, no se puede cambiar nunca, a
menos de recompilar el código.
Esta forma es útil para cosas que nunca Cambian por
ejemplo el número PI
```

### NoticiaCompuesta - Obtener Palabras claves

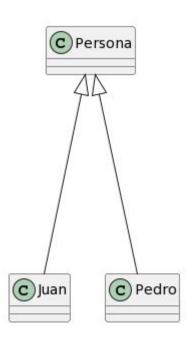
Por otro lado, las palabras claves de una sección /subsección/ grupo/subgrupo, se calculan como la unión de todas las palabras claves de los elementos que contiene, sin incluir palabras repetidas

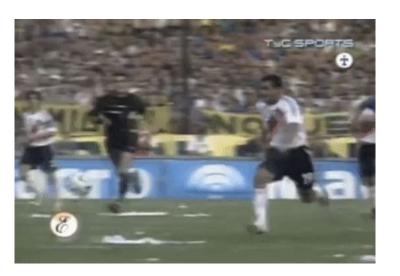
```
public ArrayList<String> getPalabrasClaves() {
    ArrayList<String> result = new ArrayList<String>();
    for(int i = 0; i<elementos.size(); i++) {
        ArrayList<String> aux = elementos.get(i).getPalabrasClaves();
        for(int j = 0; j<aux.size(); j++) {
            if (!result.contains(aux.get(i)) {
                result.add(aux.get(i);
            }
        }
    }
    return result;
}</pre>
```

# No tan grandes momentos

\_\_\_\_

Clase Vs Instancia

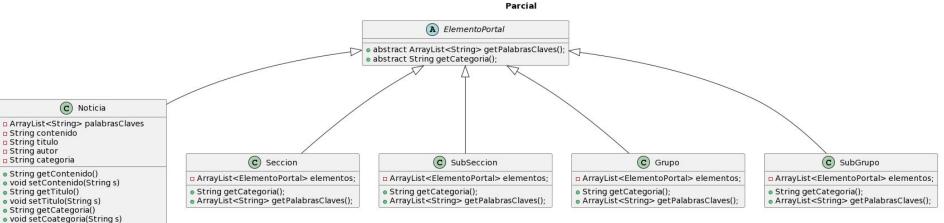




#### Todas las clases todas

String getAutor()void setAutor(String s)

ArrayList<String> getPalabrasClaves()
 void addPlabraClave(String s)



Clase Vs Instancia

#### Parcial (A) ElementoPortal abstract ArrayList<String> getPalabrasClaves(); abstract String getCategoria(); (c) Noticia ArrayList<String> palabrasClaves □ String contenido □ String titulo String autor String categoria (A) NoticiaCompuesta String getContenido() void setContenido(String s) ArrayList<ElementoPortal> elementos; String getTitulo() ArrayList<String> getPalabrasClaves(); void setTitulo(String s) String getCategoria() void setCoategoria(String s) String getAutor() void setAutor(String s) ArrayList<String> getPalabrasClaves() void addPlabraClave(String s) C Seccion C Grupo

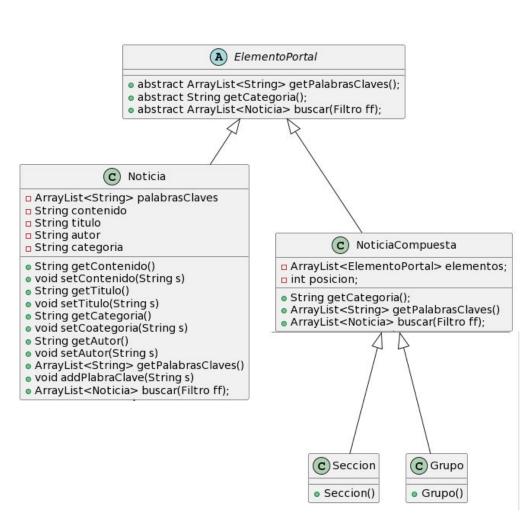
String getCategoria();

String getCategoria();



Clase Vs Instancia

Intermedia era abstracta, abajo usan 5 o 4





Clase Vs Instancia

Las sub clases con solo constructores y por ejemplo llaman a super(4)

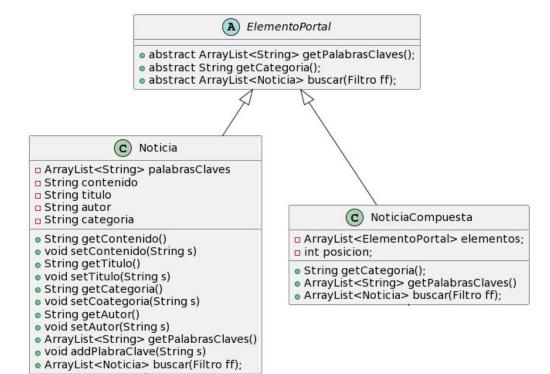
#### Busquedas

El sistema de noticias posee un motor de búsqueda el cual permite listar noticias por los siguientes criterios:

- Todas las noticias cuyo título contenga la palabra "Tandil"
- Todas las noticias que en el contenido de la misma se encuentre la frase "en 1930"
- Todas las noticias de la categoría "Sociales"
- Todas las noticias cuyo autor sea "Ricardo Ruben"
- Cualquier combinación lógica de los anteriores

#### Búsquedas

Se abstrae la búsqueda en elemento portal y la implementan los dos hijos



#### Busquedas

#### Noticia

```
public ArrayList<Noticia> buscar(Filtro ff) {
 ArrayList<Noticia> salida = new ArrayList<Noticia>();
 if (ff.cumple(this)) {
     salida.add(this);
 return salida;
                          public ArrayList<Noticia> buscar(Filtro ff) {
                           ArrayList<Noticia> salida = new ArrayList<Noticia>();
                           for(int i =0; i<elementos.size();i++){</pre>
                               salida.addAll(elementos.get(i).buscar(ff));
                           return salida:
                                             NoticiaCompuesta
                                             La busqueda es la suma de
                                             las busquedas sobre los
                                             elementos
```

#### Busquedas, error comun

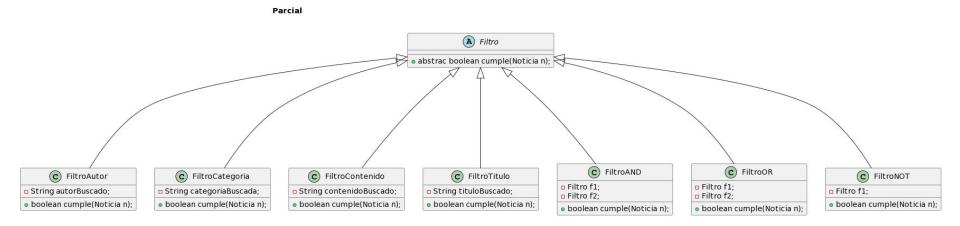
Provaca que:

-Solo un nivel de búsqueda, nunca se propaga. (MAL)

El cumple de filtro queda sobre ElementoPortal, que no tiene muchos metodos

```
public ArrayList<Noticia> buscar(Filtro ff) {
   ArrayList<Noticia> salida = new ArrayList<Noticia>();
   for(int i =0; i<elementos.size();i++) {
      if (ff.cumple(elementos.get(i)) {
            salida.add(elementos.get(i));
      }
   return salida;
}</pre>
```

#### **Filtros**





¿Son clases Nuevas? ¿hacen algo distinto? tienen atributos distintos?

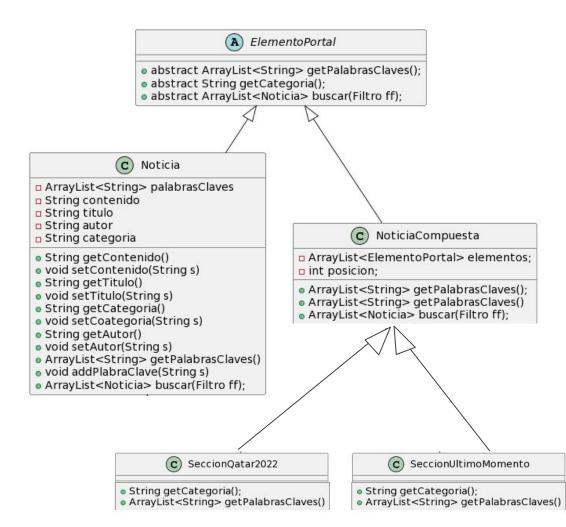
- Una sección "último momento" que posee un conjunto de noticias, secciones, subsecciones, grupos y subgrupos, pero su categoría no depende de las noticias que contiene, sino que es "último momento". Por otro lado, si le solicitan las palabras claves, realiza una unión de todas las palabras claves de las noticias que contiene (eliminando repetidas), las ordena alfabéticamente y sólo retorna las 5 primeras palabras claves de la unión.
- Una sección "Qatar 2022" que posee un conjunto de noticias, secciones, subsecciones, grupos y subgrupos, y posee las siguientes características: las palabras claves de la misma se calculan como las 3 primeras palabras claves (ordenadas alfabéticamente) de la unión de todas las palabras claves de las noticias que contiene (sin repetidos), y la categoría de la sección va a ser "Qatar 2022", independientemente de las noticias que tenga.

- Una sección **"último momento"** que posee un conjunto de noticias, secciones, subsecciones, grupos y subgrupos, pero su categoría no depende de las noticias que contiene, sino que es **"último momento"**. Por otro lado, si le solicitan las palabras claves, realiza una unión de todas las palabras claves de las noticias que contiene (eliminando repetidas), las ordena alfabéticamente y sólo retorna las **5** primeras palabras claves de la unión.
- Una sección "Qatar 2022" que posee un conjunto de noticias, secciones, subsecciones, grupos y subgrupos, y posee las siguientes características: las palabras claves de la misma se calculan como las 3 primeras palabras claves (ordenadas alfabéticamente) de la unión de todas las palabras claves de las noticias que contiene (sin repetidos), y la categoría de la sección va a ser "Qatar 2022", independientemente de las noticias que tenga.

### Clase vs Instancia

NO!



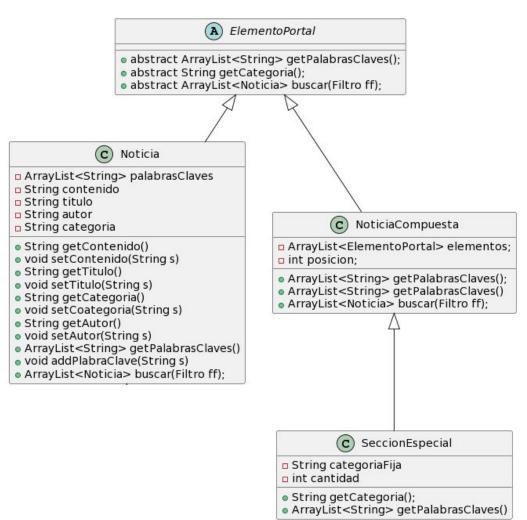


- Una sección **"último momento"** que posee un conjunto de noticias, secciones, subsecciones, grupos y subgrupos, pero su categoría no depende de las noticias que contiene, sino que es **"último momento"**. Por otro lado, si le solicitan las palabras claves, realiza una unión de todas las palabras claves de las noticias que contiene (eliminando repetidas), las ordena alfabéticamente y sólo retorna las **5** primeras palabras claves de la unión.
- Una sección "Qatar 2022" que posee un conjunto de noticias, secciones, subsecciones, grupos y subgrupos, y posee las siguientes características: las palabras claves de la misma se calculan como las 3 primeras palabras claves (ordenadas alfabéticamente) de la unión de todas las palabras claves de las noticias que contiene (sin repetidos), y la categoría de la sección va a ser "Qatar 2022", independientemente de las noticias que tenga.

- Una sección **NOMBRE** que posee un conjunto de noticias, secciones, subsecciones, grupos y subgrupos, pero su categoría no depende de las noticias que contiene, sino que es **VALOR**. Por otro lado, si le solicitan las palabras claves, realiza una unión de todas las palabras claves de las noticias que contiene (eliminando repetidas), las ordena alfabéticamente y sólo retorna las **CANTIDAD** primeras palabras claves de la unión.
- Una sección **NOMBRE** que posee un conjunto de noticias, secciones, subsecciones, grupos y subgrupos, y posee las siguientes características: las palabras claves de la misma se calculan como las **CANTIDAD** primeras palabras claves (ordenadas alfabéticamente) de la unión de todas las palabras claves de las noticias que contiene (sin repetidos), y la categoría de la sección va a ser **VALOR**, independientemente de las noticias que tenga.

# NuevaClase SeccionEspecia

```
public String getCategoria() {
    return categoriaFija;
}
```



# NuevaClase SeccionEspecia

```
abstract String getCategoria();
                                                           abstract ArrayList<Noticia> buscar(Filtro ff):
                                                        (c) Noticia
public String getCategoria() {
                                              ArrayList<String> palabrasClaves
     return categoriaFija;
                                              String contenido
                                              String titulo
                                              - String autor
                                                                                             Compuesta
                                                                                             )Portal> elementos:
    public ArrayList<String> getPalabrasClaves() {
          ArrayList<String> aux = super.getPalabrasClaves();
                                                                                             etPalabrasClaves():
                                                                                             etPalabrasClaves()
          Collection.sort(aux);
                                                                                              buscar(Filtro ff):
          ArrayList<String> result = new ArrayList<String>();
          for (int i = 0; (i<aux.size() && (i<cantidad)); i++){
               result.add(aux.get(i));
          return result;
                                                                                             nEspecial
                                                                               u oriniy caregoriai ija
                                                                               n int cantidad
                                                                               String getCategoria();
                                                                               ArrayList<String> getPalabrasClaves()
```

A) ElementoPortal

abstract ArrayList<String> getPalabrasClaves();

# Noticias "Especiales"

¿Son clases Nuevas? ¿hacen algo distinto? tienen atributos distintos?

- Una noticia Patrocinada, esta noticia es paga y tiene la particularidad de que cuando se buscan noticias en el sitio siempre se incluye en la búsqueda.
- Una noticia Deportiva, la cual posee un título, un contenido, palabras claves, el autor es "Ernesto Cherq", y la categoría es "Deportes"

### **Noticias "Especiales"**

¿Son clases Nuevas? ¿hacen algo distinto? tienen atributos distintos?

- Una noticia Patrocinada, esta noticia es paga y tiene la particularidad de que cuando se buscan noticias en el sitio siempre se incluye en la búsqueda.
- Una noticia Deportiva la sual posee un título, un contenido, patalras claves, el autor es "Ernesto Cherq", y la categoría es "Deportes"

SOLO SON VALORES DE LOS ATRIBUTOS, es una INSTANCIA

### **Noticias "Especiales"**

¿Son clases Nuevas? ¿hacen algo distinto? tienen atributos distintos?

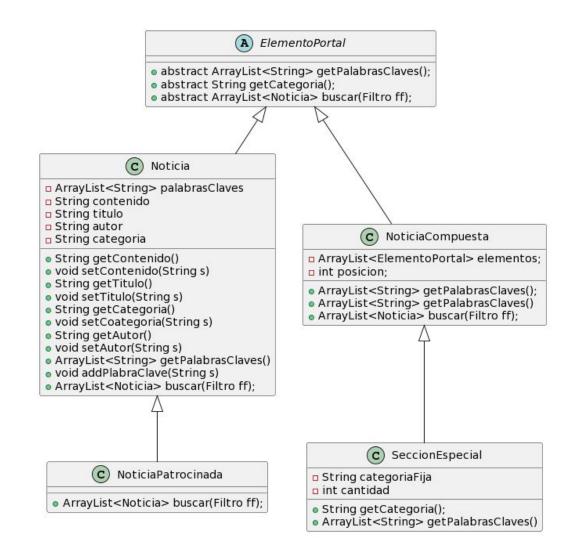
- Una noticia Patrocinada, esta noticia es paga y tiene la particularidad de que cuando se buscan noticias en el sitio siempre se incluye en la búsqueda.
- Una noticia Deport Es una Noticia, pero que contenido, atarca cambio algo del y la categoría es comportamiento. Es otra clase nueva

Cherq",

SOLO SON VALORES DE LOS ATRIBUTOS, es una INSTANCIA

### **Noticia**Patrocinada

\_\_\_\_



### Noticia Patrocinada

salida.add(this); return salida;

Siempre se agrega al buscar

```
ElementoPortal
                                                                         abstract ArrayList<String> getPalabrasClaves();
                                                                         abstract String getCategoria();
                                                                         • abstract ArrayList<Noticia> buscar(Filtro ff):
                                                                         Noticia
                                                         ArrayList<String> palabrasClaves
                                                         String contenido
                                                         String titulo
                                                         String autor
                                                                                                              NoticiaCompuesta
                                                         String categoria
                                                          Ctring actContonida/
                                                                                                  ArrayList<ElementoPortal> elementos;
                                                                                                  int posicion;
public ArrayList<Noticia> buscar(Filtro ff) {
                                                                                                  • ArrayList<String> getPalabrasClaves();
   ArrayList<Noticia> salida = new ArrayList<Noticia>();
                                                                                                  ArrayList<String> getPalabrasClaves()
                                                                                                  ArrayList<Noticia> buscar(Filtro ff);
                                                                                         es()
                                                         void addriablaciave(String s)
                                                         • ArrayList<Noticia> buscar(Filtro ff);
                                                                                                           (c) SeccionEspecial
                                                                    NoticiaPatrocinada
                                                                                                   String categoriaFija
                                                                                                   n int cantidad
                                                          ArrayList<Noticia> buscar(Filtro ff);
                                                                                                   String getCategoria();
                                                                                                   ArrayList<String> getPalabrasClaves()
```

### Clasificador palabras claves

Siempre hubo problemas en determinar las palabras claves de una noticia. Para evitar esto, se debe proveer un servicio que, dada una noticia, permita completar sus palabras claves de acuerdo a diferentes condiciones. Por ejemplo:

- Si el texto de la noticia contiene "Fútbol" se le agrega la palabra clave "deportes".
- Si el texto contiene "iphone" o "android" se le agrega la palabra clave "móviles".
- Si el título posee la palabra "heridos" se le agrega la palabra clave "policiales" y la palabra clave "urgente".

Tener en cuenta que estos son solo algunos ejemplos, y que en el portal pueden existir varias condiciones al mismo tiempo. Por ejemplo, dada una noticia con el contenido "jugador de Fútbol es la nueva imagen de android", siguiendo las condiciones detalladas anteriormente se le agregaría la palabra clave "deportes" (por la primera condición) y también la palabra clave "móviles" (por la segunda condición).

# Clasificador palabras claves

#### ¿Quien Hace esto? ¿Hay algo similar?

Siempre hubo problemas en determinar las palabras claves de una noticia. Para evitar esto, se debe proveer un servicio que, dada una noticia, permita completar sus palabras claves de acuerdo a diferentes condiciones. Por ejemplo:

- Si el texto de la noticia contiene "Fútbol" se le agrega la palabra clave "deportes".
- Si el texto contiene "iphone" o "android" se le agrega la palabra clave "móviles".
- Si el título posee la palabra "heridos" se le agrega la palabra clave "policiales" y la palabra clave "urgente".

Tener en cuenta que estos son solo algunos ejemplos, y que en el portal pueden existir varias condiciones al mismo tiempo. Por ejemplo, dada una noticia con el contenido "jugador de Fútbol es la nueva imagen de android", siguiendo las condiciones detalladas anteriormente se le agregaría la palabra clave "deportes" (por la primera condición) y también la palabra clave "móviles" (por la segunda condición).

### **Portal Noticias**

¿Qué hace?

public void completarNoticia(Noticia n) {
}

¿Qué tiene?

Cómo controla todo?

Siempre hubo problemas en determinar las palabras claves de una noticia. Para evitar esto, se debe proveer un servicio que, dada una noticia, permita completar sus p<del>alabras claves de acuerdo a diferentes co</del>ndiciones. Por ejemplo:

- Si el texto de la noticia contiene "Fútbol" se le agrega la palabra clave "deportes".
- Si el texto contiene "iphone" o "android" se le agrega la palabra clave "móviles".
- Si el título posee la palabra "heridos" se le agrega la palabra clave "policiales" y la palabra clave "urgente".

#### Puedo Usar Filtros!

### Clasificador

```
c Clasificador
```

- ArrayList<String> palabras;
- Filtro cond;
- void clasificar(Noticia n);

Si cumple "cond" le agrego palabras

```
public void clasificar(Noticia nn) {
    if (cond.cumple(nn) ) {
        for (int i= 0; i<palabras.size(); i++) {
            nn.addPalabraClave(palabras.get(i));
        }
}</pre>
```

### **PortalNoticias**

Recorro mis clasificadores y le pido que clasifiquen la noticia dada

```
public class PortalNoticias {
   ArrayList<Clasificador> clasificadores;
   ...
   public void clasificarNoticia(Noticia nn) {
        for(int i =0; i<clasificadores.size();i++) {
            clasificadores.get(i).clasificar(nn);
        }
   }
}</pre>
```

