

# Examen - Unidad 5

## Diccionario de la RAE

Índice de contenido:

<b>Introducción.....</b>	<b>2</b>
<b>1. Tipos.....</b>	<b>3</b>
<b>2. Aceptación.....</b>	<b>3</b>
Constructor.....	4
Getters / Setters.....	4
Métodos.....	4
<b>3. Palabra.....</b>	<b>5</b>
Constructores.....	5
Getters / Setters.....	6
Métodos.....	6
<b>4. Diccionario.....</b>	<b>7</b>
Constructores.....	7
Getters / Setters.....	7
Métodos.....	7
<b>4. Main.....</b>	<b>8</b>

# Introducción

El diccionario de la RAE permite buscar los significados o acepciones de una palabra, que tiene la **palabra** en sí, la **etimología** y puede tener una o varias **acepciones** o significados:

The diagram illustrates the components of a dictionary entry for the word "gato".

- Palabra**: Points to the word "gato<sup>1</sup>, ta" in a blue oval.
- Etimología**: Points to "Del lat. tardío *cattus*." in a green oval.
- Acepciones o significados**: Points to the numbered list of meanings.
  1. m. y f. Mamífero de la familia de los félidos, digitigrado, doméstico, de unos 50 cm de largo desde la cabeza hasta el arranque de la cola, que por sí sola mide unos 20 cm, de cabeza redonda, lengua muy áspera, patas cortas y generalmente pelaje suave y espeso, de color blanco, gris, pardo, rojizo o negro, empleado en algunos lugares para cazar ratones. U. en m. ref. a la especie.  
Sin.: minino, michino, cucho<sup>2</sup>, micho, morrongo, morroño, mozo<sup>3</sup>.
  2. m. y f. coloq. Persona nacida en Madrid.  
Sin.: madrileño.

Cada acepción se compone de un **número**, un **tipo y género** y el **significado** en sí.

Además, cada acepción puede llegar a tener una lista de sinónimos y/o antónimos:

The diagram provides a more detailed view of the dictionary entry for "gato", highlighting specific parts.

- Tipo y género**: Points to the circled "1." and "2." in the numbered list.
- Sinónimos (los antónimos saldrían debajo)**: Points to the "Sin.:" line for the first meaning: "Sin.: minino, michino, cucho<sup>2</sup>, micho, morrongo, morroño, mozo<sup>3</sup>."

Veamos cómo vamos a programar esto. Por lo pronto, **todo se implementará en el paquete diccionario salvo el Main**, que se hará fuera de él.

# 1. Tipos

Para el tipo, género y demás literales necesarios, vamos a usar un enumerado:

<code>&lt;&lt;enumeration&gt;&gt;</code> <b>Tipos</b>
- tipo: String
- Tipos(String) + getNombreTipo(): String

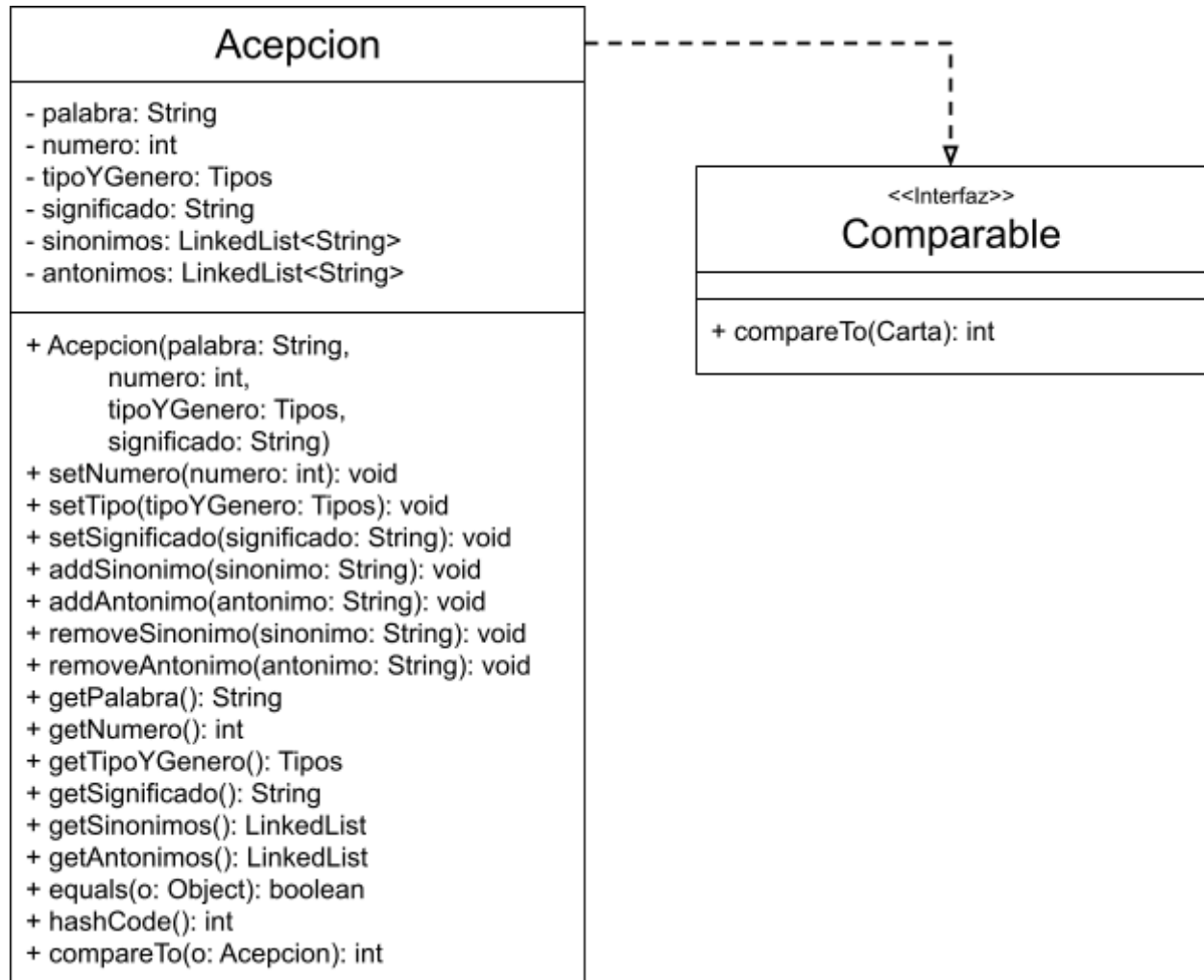
Cuyo contenido será el siguiente:

Término	Abreviatura
NOMBRE_MASCULINO	m.
NOMBRE_FEMENINO	f.
NOMBRE_MASCULINO_Y_FEMENINO	m. y f.
COLOQUIAL	coloq.
ADJETIVO	adj.
ADVERBIO	adv.
CONJUNCION_COPULATIVA	conj. copulat.
PREPOSICION	prep.
PREFIJO	pref.
SUFIJO	suf.
SINONIMO	sin.
ANTONIMO	ant.

## 2. Aceptación

Esta clase representa un significado de la palabra. Recordemos que cada aceptación o significado tiene un **número**, un **tipo y género** y el **significado** en sí. Además **es posible** que tenga una **lista** de **sinónimos** y otra de **antónimos**.

Vamos a abstraer el concepto de significado implementando la siguiente clase para nuestro programa:



Esta clase implementará la interfaz **Comparable** para comparar este significado con otro mediante su número.

## Constructor

Establece los valores de los atributos.

## Getters / Setters

Deben implementarse los getters y setters para los atributos.

## Métodos

Deben implementarse los getters para los atributos.

- **addSinonimo(String sinonimo)**

Añade un sinónimo a la lista de sinónimos de esta acepción. Al hacerlo, **reordena todos los sinónimos alfabéticamente**.

- **addAntonimo(String antonimo)**  
Hace lo mismo que el método anterior, pero para la lista de antónimos.
- **removeSinonimo(String sinonimo)**  
Borra el sinónimo indicado.
- **removeAntonimo(String antonimo)**  
Borra el antónimo indicado.
- **equals(Object): boolean**  
Has las validaciones necesarias, incluyendo además que serán iguales si el **palabra** y el **significado** son los mismos.
- **hashCode(): int**  
Crea el código hash.

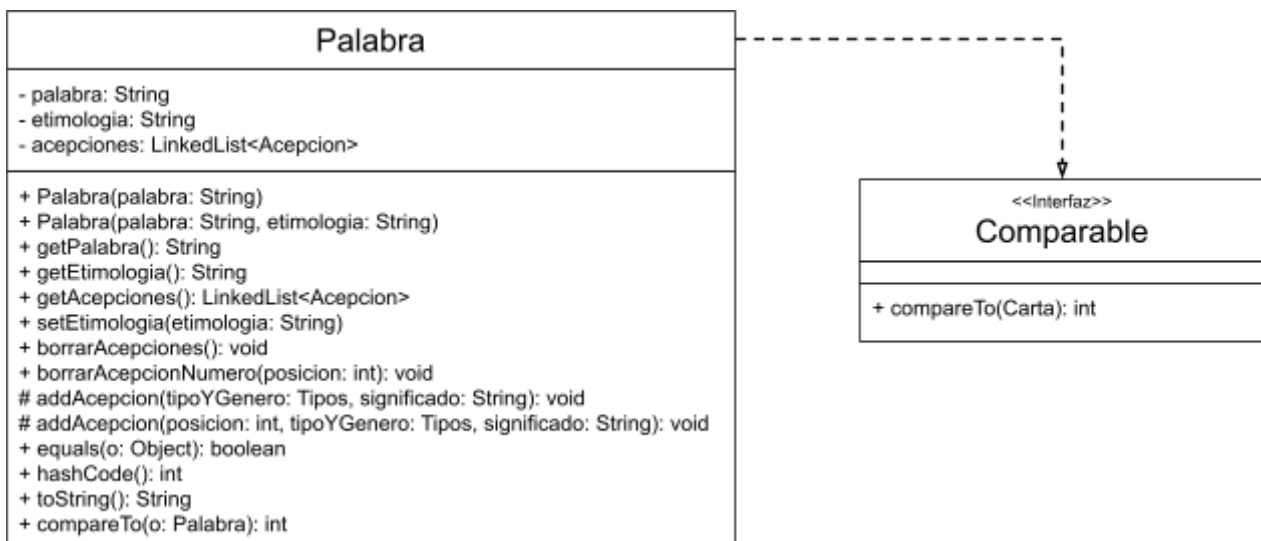


Recuerda que debes usar los mismos datos que has usado en el equals.

- **compareTo(Carta): int**  
Permite comparar dos acepciones según su **número**.

### 3. Palabra

Cada palabra en el diccionario se compone de la **palabra** en sí, su **etimología** y tiene una o varias **acepciones** o significados:



### Constructores

Los constructores inicializan los valores de la palabra. El constructor `Palabra(palabra: String)` poner el atributo etimología a nulo.

# Getters / Setters

Deben implementarse los getters y setters para los atributos.

## Métodos

- **borrarAcepciones()**  
Elimina todas las acepciones de la palabra.
- **borrarAcepcionNumero(int posicion)**  
Elimina la acepción de la posición indicada.
- **addAcepcion(int posicion, Tipos tipoYGenero, String significado)**  
Añade una acepción en la posición indicada. Si la posición es negativa, la añade al principio. Si excede el límite de la lista, se añade al final.  
  
**Si la acepción ya existe**, no debe añadirse.
- **addAcepcion(Tipos tipoYGenero, String significado)**  
Con la misma mecánica que el método anterior, añade una nueva acepción al final de la lista de acepciones de la palabra.
- **equals(Object): boolean**  
Has las validaciones necesarias, incluyendo además que serán iguales si la **palabra** lo es.
- **hashCode(): int**  
Crea el código hash.
- **toString(): String**  
Devuelve la palabra con todas sus acepciones en el mismo formato que el diccionario que se muestra en las figuras anteriores. Esto es:

```
gato:
1.- m. y f. Mamífero de la familia de los félidos, dig
sin.: [minino, michino, micho, morrongo, morroño]
ant.: [perro]
2.- m. y f. Persona nacida en Madrid.
3.- m. Máquina que sirve para levantar grandes pesos a
4.- coloq. Hombre sagaz, astuto.
sin.: [zorro, pícaro, traicionero]
```

- **compareTo(Carta): int**  
Permite comparar dos acepciones por orden alfabético. Es decir, según la **palabra**.

## 4. Diccionario

Esta clase engloba todas las palabras del diccionario, con sus respectivas acepciones.

Diccionario
- diccionario: LinkedList<Palabra>
+ Diccionario()   + addPalabra(palabra: Palabra, tipoYGenero: Tipos, significado: String): void + addPalabra(palabra: Palabra, numeroSignificado: int, tipoYGenero: Tipos, significado: String): void + addSinonimo(palabra: Palabra, numeroAcepcion: int, sinonimo: String): void + addAntonimo(palabra: Palabra, numeroAcepcion: int, antonimo: String): void + toString(): String

### Constructores

El constructor inicializa los valores del diccionario.

### Getters / Setters

No debe tener getters ni setters.

### Métodos

- **addPalabra(palabra: Palabra, numeroSignificado: int, tipoYGenero: Tipos, significado: String): void**  
Añade un nuevo significado de esa palabra con el número de acepción indicado. Si la palabra no está incluida en el diccionario, la crea y luego le asigna esa acepción o significado.
- **addPalabra(palabra: Palabra, tipoYGenero: Tipos, significado: String): void**  
Añade un nuevo significado de esa palabra como **primer significado**. Si la palabra no está incluida en el diccionario, la crea y luego le asigna esa acepción o significado.

- **addSinonimo(palabra: Palabra, numeroAcepcion: int, sinonimo: String): void**

Si la palabra no existe, muestra un error por pantalla y finaliza.

Si existe, añade el sinónimo al significado indicado. **Importante**: no se pueden repetir sinónimos.

- **addAntonimo(palabra: Palabra, numeroAcepcion: int, antonimo: String): void**

Hace lo mismo que el método anterior, pero para los antónimos.

- **toString(): String**

Muestra cada palabra con todas sus acepciones y la separa de la siguiente con una línea de guiones:

```
gato:
1.- m. y f. Mamífero de la familia de los félidos, digitígrado, domo
sin.: [minino, michino, micho, morrongo, morroño]
ant.: [perro]
2.- m. y f. Persona nacida en Madrid.
3.- m. Máquina que sirve para levantar grandes pesos a poca altura,
4.- coloq. Hombre sagaz, astuto.
sin.: [zorro, pícaro, traicionero]
-----
perro:
1.- m. y f. Mamífero doméstico de la familia de los cánidos, de tama
sin.: [can, chucho, tuso, pulguinto]
-----
```

## 4. Main

Se te proporciona un main para el ejercicio. El resultado de su ejecución debe ser el de la pantalla anterior.