línea horizontal

**Urtzi Arana Santamaria**

Bases de Datos (GMC)

Laboratorio 1

**25 de Abril, 2021**

# VISIÓN GENERAL

Laboratorio cuyo objetivo principal es familiarizarse con el lenguaje de programación Java y el uso de listas dinámicas doblemente enlazadas.

# ESPECIFICACIONES

1. Debe ser una lista tipada para almacenar elementos de tipo *String*, es decir, sólo se almacenarán cadenas de caracteres.
2. La funcionalidad que se debe incluir es:
   * Contar el número de elementos que tiene la lista.
   * Mostrar el elemento que hay en una posición concreta de la lista.
   * Comprobar si un elemento está en la lista.
   * Imprimir los elementos que contiene la lista.
   * Insertar un elemento nuevo.
   * Sacar un elemento concreto de la lista.
   * Sacar el elemento que ocupa una posición en la lista.
   * Concatenar dos listas.
   * Reemplazar un elemento de la lista.
3. incluir una clase Principal que use la estructura que has implementado y la aplique sobre algún problema real que pueda resolverse con una lista doblemente enlazada. Deberás utilizar todos los métodos implementados y el funcionamiento correcto debe hacerse explícito mostrando mensajes de resultados de ejecución por consola

# HITOS

## Creación del Proyecto en el IDE Visual Studio Code

Utilizamos la herramienta Java Tools para arrancar el proyecto de CountryList, en este proyecto se incorporan las carpetas .dist , lib y la principal src. Dentro de la carpeta de src, se crean tres ficheros java; App.java, List.java y Node.java.

## Definición del fichero Node.java

Para crear una lista dinámicas doblemente enlazada, necesitamos un archivo que recoja las especificaciones de los nodos que van a ser punto de referencias en la lista dinámica.

Primero creamos la clase pública Node, definiendo como variables privadas las siguientes :

* countryName (tipada como String); que recoge el nombre del país
* nextElement (tipada como Node); es el puntero que utilizaremos para acceder al siguiente elemento dentro de la lista
* preElement (tipada como Node); es el puntero que utilizaremos para acceder al anterior elemento dentro de la lista

Lo siguiente será ir creando los métodos que se utilizaran para acceder (get) a los elementos previos y posteriores de la lista, además de definirlos (set).

Finalmente incorporamos un método que (showList()) que nos retornará el nombre del país y que utilizaremos en las iteraciones sobre la lista en vez de getCountry() finalmente principalmente para que sea un método más entendible.

**Definición del fichero List.java**

En este archivo van a ir definidos los métodos principales que utilizaremos para interactuar con la lista y el usuario.

Primero creamos la clase pública List, definiendo como variables privadas las siguientes :

* first (tipada como Node); primero posición del puntero en el nodo
* last (tipada como Node); último posición del puntero en el nodo

Adicionalmente se definen globalmente algunas variables que serán utilizadas posteriormente por los métodos:

* size(tipada como integer): para cualquier el tamaño de la lista
* position(tipada como integer): para obtener la posición del elemento
* n(tipada como integer): variable para el iterador

Los dos primeros métodos que definimos serán addFisrt() y addLast(), que nos servirán para añadir elementos en la lista. Ambos usan un condicional para cuando la lista este vacia (ambos null) se definan las posiciones first y last como node.

El método printList() utilizar un bucle for para iterar sobre la lista e imprimir por consola los elementos, adicionalmente se incorpora un contador size para poder utilizarlo posteriormente en el método countList().

Los métodos printCountry() y validateCountry() tambien hacen uso del bucle for para iterar pero adicionalmente y de cara a poder interactuar con el usuario hacen uso del paquete JOptionPane de Java para mostar una ventana emergente donde el usuario incorpora el dato solicitado.

Los métodos extractCountry() y extractPosition(), requieren de una variable para ser ejecutados, como además iteractuan con el objeto (la lista de elementos) debemos declararlas como tal.

**Definición del fichero App.java**

Este es el ejecutable principal sobre el que iremos llamando a los diferentes métodos de cara a cumplir los objetivos del laboratorio.

Primero incorporaremos una lista de cuatro países para interactuar sobre ella, la imprimimos por consola, contamos los países que hay, imprimimos un país que el usuario nos solicita por posición y validamos que el que nos introduce por pantalla existe. Importante que hayamos imprimido por consola previamente los países para que el usuario sepa que introducir.

Para terminar con la primera parte del ejercicio, sacamos de la lista el elemento en la posición 3 y el elemento llamado Andorra.

En la segunda parte del ejercicio vamos a hacer uso del paquete de Java util.LinkedList ya que resulta más directo y fácil que a través de otras formas más complejas que requerirían de mayor líneas de código.

Instanciamos dos nuevas listas enlazadas, con dos países cada una, luego utilizamos el método addAll() para añadir en una tercera instancia con una nueva lista enlazada llamada concat para incorporar los elementos de cada una de las lista.

Finalmente para reemplazar un elemento de esta tercera lista enlazada, utilizamos el método set donde el primer argumento es la posición que queremos reemplazar y el segundo el nombre del país. Y luego imprimimos por pantalla.

**Fin**