

Nombre \_\_\_\_\_

ED-Ex2A-20180207

Haz un proyecto con la siguiente funcionalidad:

(2 puntos) En un TextArea a la izquierda de la ventana el usuario podrá escribir palabras (una por renglón). Cuando se pulse un botón situado debajo (con el texto Leer), revisará el contenido del TextArea quedándose de cada línea con las 20 primeras letras, eliminando las líneas vacías y las líneas repetidas (refrescando el TextArea con los valores depurados).

(2 puntos) Después, en un panel situado a la derecha del TextArea, pondrá una lista con tantos botones como palabras correctas tengamos, con el texto de cada una de ellas en cada botón, y se limpiará de contenido el TextArea de la derecha. Al pulsar un botón de la lista se añadirá su texto en el TextArea de la derecha (cada palabra en una línea).

(2 puntos) Cuando pulsemos un botón situado debajo de este TextArea derecho, el programa comprobará que no hay repeticiones, que están todas las palabras de la izquierda y que están en el orden impuesto por uno de los dos botones de radio situados debajo (alfabético descendente y por longitud de la palabra ascendente). Indicará el error (o el éxito) en una etiqueta situada más abajo.

(2 puntos) Para trabajar el “modelo o negocio” te apoyarás en una clase que has de construir: Lista que lleva todo el control de un ArrayList de String: cuántos elementos tiene (longitud), añadir nuevos elementos (añadir) depurándolos (si vacíos no lo añade y devuelve -2, si está repetido no lo añade y devuelve -1, si supera las 20 letras añade sólo las 20 primeras y devuelve 0, en otro caso añade y devuelve 0), vaciar el ArrayList(limpiar), coger el elemento situado en una posición del Array, comprobar si está ordenado alfabéticamente descendente, comprobar si está ordenada por la longitud de las cadenas.

(2 puntos) Se harán las pruebas de la clase Lista usando La clase ListaTest generada con JUnit que se adjunta.

AYUDAS: el TextArea se usa como un TextField que permite más de una línea (separador de líneas: System.lineSeparator() )

La cadena que devuelve el método getText() podemos trocearla con el método split(System.lineSeparator())

```
public class ListaTest {
    Lista i1, i2, i4, i5, i6;
    public ListaTest() {
        i1= new Lista(); i1.añadir("uno"); i1.añadir("dos"); i1.añadir("tres");
        i2= new Lista(); i2.añadir("cuatro"); i2.añadir("dos"); i2.añadir("tres");
        i4= new Lista(); i4.añadir("uno");
        i5= new Lista();
        i6= new Lista(); i6.añadir("tres"); i6.añadir("cuatro");
            i6.añadir("cinco"); i6.añadir("algo");
    }
}
```

```
}
```

```
@Test
```

```
public void testLongitud() {  
    System.out.println("longitud");  
    assertEquals(3, i1.longitud());  
    assertEquals(1, i4.longitud());  
    assertEquals(0, i5.longitud());  
}
```

```
@Test
```

```
public void testAñadir() {  
    System.out.println("añadir");  
    Lista i3= new Lista(); i3.añadir("cinco"); i3.añadir("tres"); i3.añadir("dos");  
    assertEquals(-1, i3.añadir("dos"));  
    assertEquals(-2, i3.añadir(""));  
    assertEquals(0, i3.añadir("cuatro"));  
    assertEquals(0, i3.añadir("01234567890123456789012345"));  
    assertEquals("01234567890123456789", i3.coger(4));  
    assertEquals("cuatro", i3.coger(3));  
}
```

```
@Test
```

```
public void testCoger() {  
    System.out.println("coger");  
    assertEquals("uno", i1.coger(0));  
    assertEquals("tres", i1.coger(2));  
    assertEquals("", i1.coger(3));  
    assertEquals("", i1.coger(-8));  
}
```

```
@Test
```

```
public void testOrdenadoAlfabeticoDescendente() {  
    System.out.println("ordenadoAlfabeticoDescendente");  
    assertEquals(false, i1.ordenadoAlfabeticoDescendente());  
    assertEquals(true, i6.ordenadoAlfabeticoDescendente());  
    assertEquals(true, i4.ordenadoAlfabeticoDescendente());  
}
```

```
@Test
```

```
public void testOrdenadoLongitudAscendente() {  
    System.out.println("ordenadoLongitudAscendente");  
    assertEquals(true, i1.ordenadoLongitudAscendente());  
    assertEquals(false, i2.ordenadoLongitudAscendente());  
    assertEquals(true, i5.ordenadoLongitudAscendente());  
}
```

```
@Test
```

```
public void testLimpiar() {  
    System.out.println("limpiar");  
    Lista i3= new Lista(); i3.añadir("cinco"); i3.añadir("tres"); i3.añadir("dos");  
    assertEquals(3, i3.longitud());  
    i3.limpiar();  
    assertEquals(0, i3.longitud());  
}  
  
}
```