



**Universidad Autónoma de Zacatecas**

**Unidad Académica de Ingeniería Eléctrica**

**Programa Académico de Ingeniería de Software.**

**Nombre de la Práctica** Pilas.

**Numero de Práctica** 13

**Nombre de la carrera** Ingeniería de Software

**Nombre de la materia** Lab. Estructuras de Datos

**Nombre del alumno** Jesús Manuel Juárez Pasillas

**Nombre del docente** Aldonso Becerra Sánchez

**Fecha:** 20/09/2021

## Práctica 13: Pilas.

### Introducción:

Las pilas al igual que los arreglos nos permiten almacenar una gran cantidad de elementos donde su capacidad es estática. Aunque estos se parecen bastante las pilas nos permiten ir agregando elementos y para sacarlos se tendrá que comenzar desde el último que se ingreso al primero, de esta forma nos permite dar un orden a los datos que se almacenan en ella y con esto dar solución a una gran variedad de problemáticas.

### Desarrollo:

Para esta práctica se harán varias actividades usando pilas para el manejo de los datos. Los ejercicios consisten en:

1. Encriptar un mensaje.
2. Desencriptar un archivo el cual contiene texto.
3. Evaluar las palabras que tenga un archivo para ver si son palíndromas.

La forma de encriptar el mensaje será haciendo agrupaciones con el carácter “Ç”, el cual definirá cual será el inicio de la encriptación y cuál será su final en dicha agrupación, dentro de esta agrupación el mensaje cambiara de dirección siendo que si la agrupación contiene “Hola” quede “aloH”, y esta frase quedara dentro del mensaje encapsulado entre el carácter mencionado.

La encriptación del mensaje se hará de forma aleatoria, siendo que el número de agrupaciones sea aleatorio, el inicio y su longitud de cada agrupación también será aleatorio, por lo que se deberá validar cada dimensión para que no queden agrupaciones empalmadas

Para realizar este ejercicio se hizo un nuevo paquete (src/herramientas/texto) y en el se agregó una nueva clase.

**Encriptacion:** A esta clase se le agrego un método estático en el cual solo se le pasa el mensaje que se quiere encriptar y el método regresa el mensaje encriptado.

```
Encripta un mensaje donde el número de partes a encriptar, las posiciones  
donde encriptara y sus longitudes son aleatorias.  
public static String encriptarMensaje(String mensaje);
```

Para desencriptar el archivo se hizo escribió un nuevo archivo (encriptado.txt) en el cual se tiene el mensaje encriptado del ejercicio anterior y también se agrego un nuevo elemento (oÇosÇ) el cual es un palíndromo que servirá para el siguiente ejercicio. Y con estos elementos almacenados en el archivo se desencriptaron y se almacenaron automáticamente dentro del archivo (desencriptado.txt), donde ambos archivos están dentro de la carpeta “src” dentro del proyecto. Para esto se utilizo la misma clase que el ejercicio anterior.

**Encriptacion:** Se agregaron dos métodos nuevos uno el cual recibe el nombre del archivo que se quiere desencriptar y otro el cual recibe un mensaje el cual desencripta, el primer método hace uso las veces que sea necesario del segundo método.

*Desencripta un archivo de texto y escribe un archivo nuevo.*

```
public static boolean desencriptarArchivo(String archivo)
```

*Desencripta un mensaje dado.*

```
public static String desencriptar(String mensaje)
```

Para el último ejercicio se hizo uso del archivo escrito en el ejercicio anterior, el cual tiene los mensajes desencriptados y en el cual buscara que palabras dentro del archivo son palíndromas (Un palíndromo es una palabra la cual se lee igual al derecho que al revés).

Para esto se hizo una nueva clase dentro del paquete (src/herramientas/texto) llamada Palindromo.

**Palindromo:** Dentro de esta clase se agregaron dos métodos, uno el cual recibe el nombre de un archivo el cual lo abre y obtiene los textos en un Arreglo y este hace uso del segundo método el cual recibe una palabra y verifica que sea un palíndromo y regresa true o false, si regresa true el primer método imprime la palabra.

*Imprime las palabras dentro del archivo que son palíndromas.*

```
public static void palindromosEnArchivo(String archivo)
```

*Mete la cadena en dos pilas, una de forma normal y otra en forma invertida y evalúa los datos para ver si son iguales.*

```
public static boolean esPalindroma(String cadena)
```

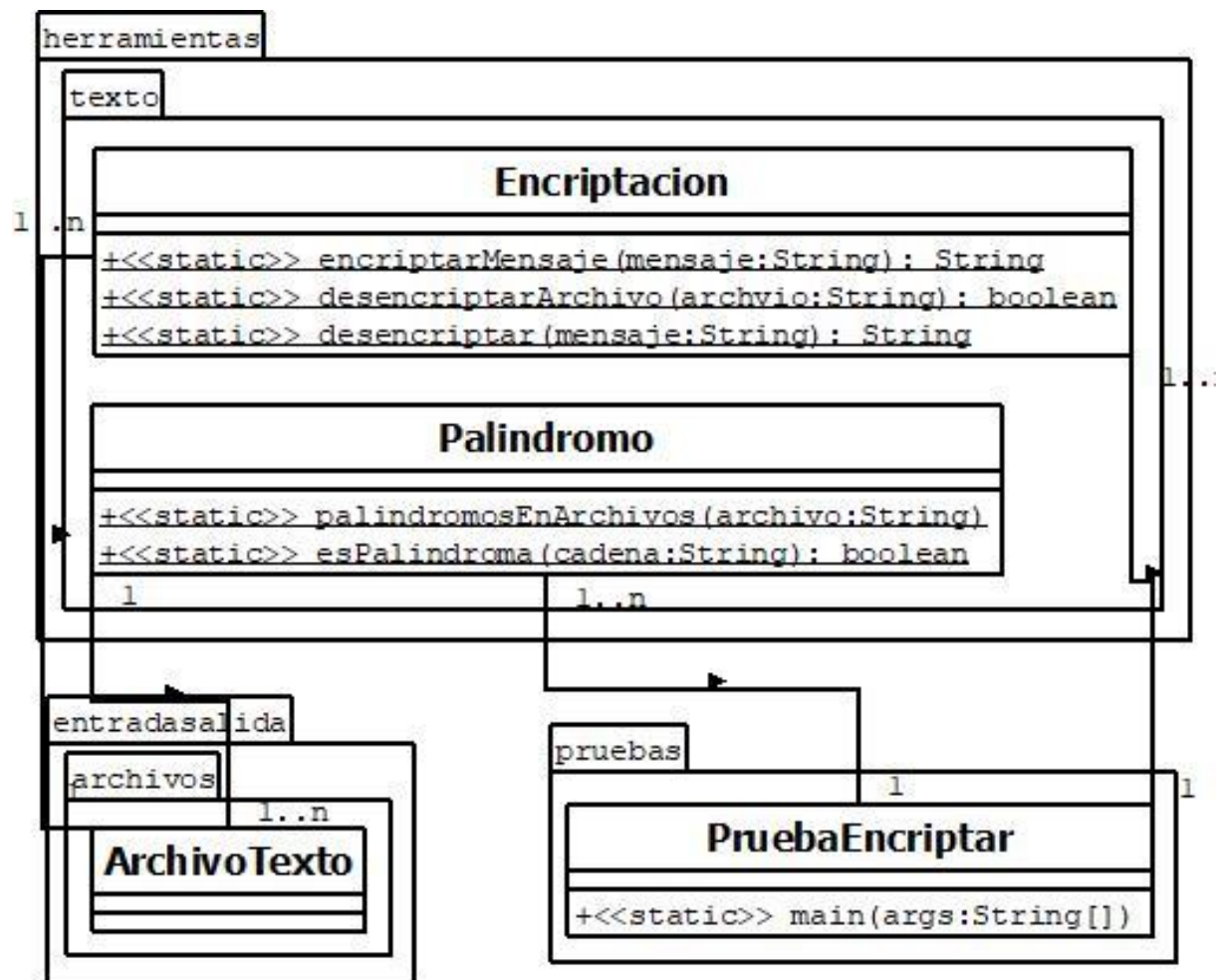
**Nota:** Toda la documentación esta agregada en la carpeta “doc” dentro de la carpeta del proyecto (“edylab\_2021\_13/doc”).

## Capturas del programa funcionando:

```
Estructuras dçsotaD eç
Desencriptar archivo de texto: true
Estructuras de Datos
0S0
Palabras palindromas en el archivo desencriptado:
0S0
```

## Código agregado:

Todas las clases y métodos que se ven en el diagrama son nuevos a excepción de la clase ArchivoTexto.



## Pre-evaluación:

Pre-Evaluación para prácticas de Laboratorio de Estructuras de Datos	PRE-EVALUACIÓN DEL ALUMNO
CUMPLE CON LA FUNCIONALIDAD SOLICITADA.	Sí
DISPONE DE CÓDIGO AUTO-DOCUMENTADO.	Sí
DISPONE DE CÓDIGO DOCUMENTADO A NIVEL DE CLASE Y MÉTODO.	Sí
DISPONE DE INDENTACIÓN CORRECTA.	Sí
CUMPLE LA POO.	Sí
DISPONE DE UNA FORMA FÁCIL DE UTILIZAR EL PROGRAMA PARA EL USUARIO.	Sí
DISPONE DE UN REPORTE CON FORMATO IDC.	Sí
LA INFORMACIÓN DEL REPORTE ESTÁ LIBRE DE ERRORES DE ORTOGRAFÍA.	Sí
SE ENTREGÓ EN TIEMPO Y FORMA LA PRÁCTICA.	Si
INCLUYE LA DOCUMENTACIÓN GENERADA CON JAVADOC.	Sí
INCLUYE EL CÓDIGO AGREGADO EN FORMATO UML.	Sí
INCLUYE LAS CAPTURAS DE PANTALLA DEL PROGRAMA FUNCIONANDO.	Sí
LA PRÁCTICA ESTÁ TOTALMENTE REALIZADA (ESPECIFIQUE EL PORCENTAJE COMPLETADO).	100%
Observaciones:	

## Conclusión:

El uso de las pilas facilitó la elaboración de esta práctica, ya que se usó para guardar algunos elementos de los mensajes y hacerle las operaciones correspondientes para así, obtener los resultados que queríamos.

En el caso de los palíndromos el hacer dos pilas, una conteniendo el mensaje tal cual, y otra que contuviera el mensaje al revés para poder evaluar las posiciones de cada pila y ver si todos los datos de estas posiciones son iguales, conlleva a que no existan errores al comparar cada posición.