

**Análisis y diseño de Algoritmos**

Proyecto Nº1

Campus Santa Fe

Enrique Lira Martínez A01023351

Profesor: Dr. Víctor Manuel de la Cueva H

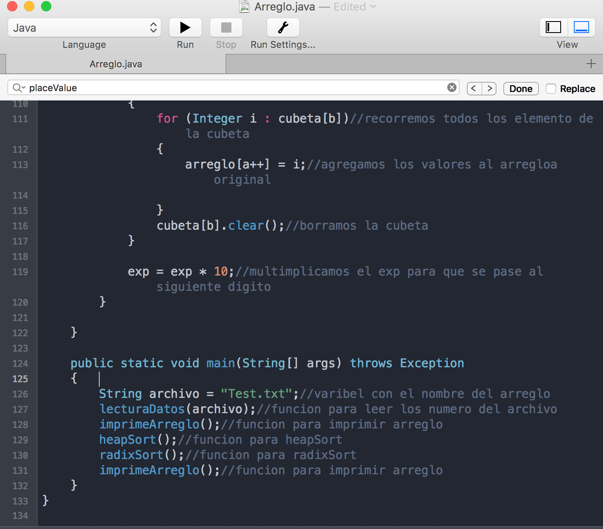
12 de febrero de 2018

**MANUAL DEL USUARIO**

**CONFIGURACIÓN**

1. El usuario debe contar con un IDE o un editor de texto en caso de que el usuario tenga un IDE.

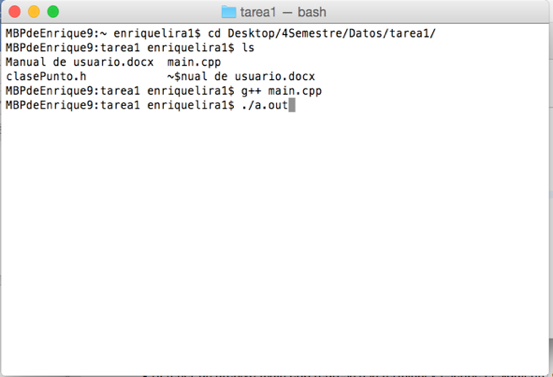
* Si el usuario tiene un IDE deberá de hacer los siguientes pasos, el usuario deberá solamente abrir el archivo en su IDE y correr el programa con el arreglo.java descrito a continuación.



***Imagen 1.1***

***Ejemplificación de un IDE***

* Si el usuario tiene un editor de texto deberá de hacer los siguientes pasos, el usuario deberá solamente su terminal y deberá de buscar el archivo (Ej. cd Desktop/capeta……) el usuario deberá de saber dónde se ubica el archivo.



***Imagen 1.2***

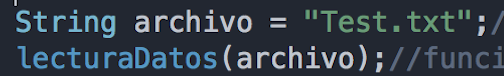
***Ejemplificación de la terminal***

* Después de haber hecho el programa main usted deberá de correr el programa en la terminal con el comando g++ main.cpp y dará enter, si no hay ningún error usted deberá escribir e comando ./a.out

**ESTRUCTURA GENERAL**

**Estructura del main propuesto**

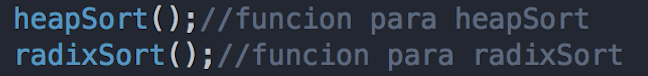
1. Primero el usuario tendrá que definirá el nombre del archivo lo podrá poner en una variable como se muestra a continuación o directamente en la función lecturaDatos



***Imagen 2.1***

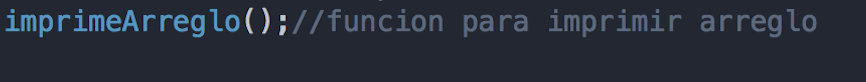
***Ejemplificación de la declaración para la funcionlecturaDatos()***

1. A continuación, se podrán hacer uso de la función radi sort o heap sort de la siguientes maneras

****

***Imagen 2.4***

***Ejemplificación de las funciones radixsort y heapsort***

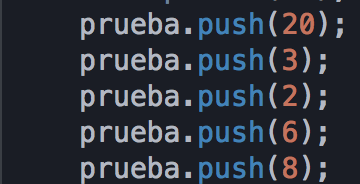
1. Si se quiere imprimirá el arreglo se deber usar la siguiente función****

***Imagen 2.3***

***Ejemplificación de la función imprimir***

**FUNCIÓN RADIXSORT ()**

A partir de una pila ya creada, podemos añadir elementos a esta mientras no sobre pasemos el tamaño máximo

****

***Imagen 2.6***

***Ejemplificación de la declaración de la función push***

**FUNCIÓN IMPRIMEPILA()**

Muestra en pantalla el valor actual para el tope de la pila y los elementos que se encuentran en esta.

**../../../Captura%20de%20pantalla%202017-03-13%20a%20la(s)%2018.52.53.png**

***Imagen 2.6***

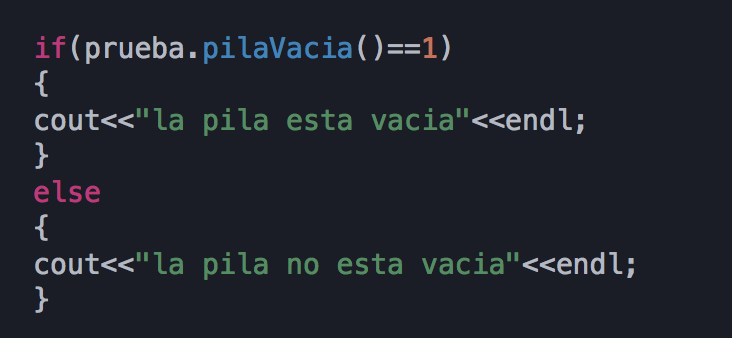
***Resultado de la función imprimirPila***

**FUNCIÓN PILAVACÍA() Y PILALLENA()**

* **pilaVacía():**

Lo que hace esta función es verificar si la pila está vacía verificando el

valor del tope de la pila Si ese valor es -1, quiere decir que no contiene elementos y nos regresa un valor true, caso contrario devuelve un false.

../../../Captura%20de%20pantalla%202017-03-13%20a%20la(s)%2018.54.34.png

***Imagen 2.7***

***Resultado de la función pilavacia***

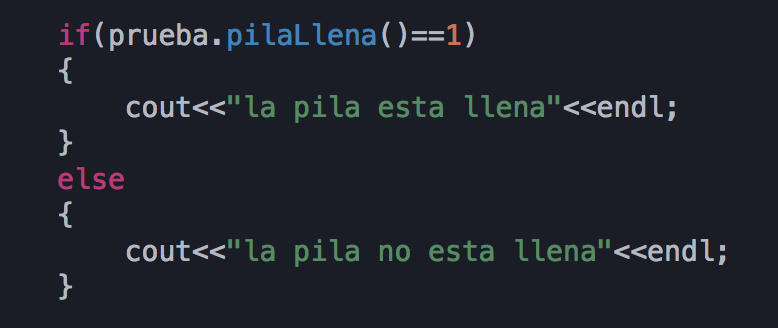
* **pilaLlena():**

Esta función verifica si la pila está llena comparando el valor del tope con

el tamaño de esta. Si el valor del tope coincide con el tamaño del arreglo,

quiere decir que ya no hay más espacios para almacenar elementos y

arroja un true . En caso de que existan más espacios arroja un false

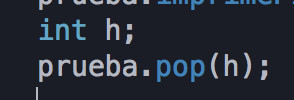
../../../Captura%20de%20pantalla%202017-03-13%20a%20la(s)%2018.55.19.png

***Imagen 2.7***

***Resultado de la función pilaLlena***

**FUNCIÓN POP()**

Esta función nos ayudará a sacar los elementos que estén en el tope mientras la pila no esté vacía. Devolverá un true en caso de haber hecho la operación de manera exitosa o un false .

../../../Captura%20de%20pantalla%202017-03-13%20a%20la(s)%2018.56.02.png

***Imagen 2.8***

***Resultado de la función pop***

**ALGORITMO**

Se crearon siguientes la funciones:

* Pila()
* Bool Push(dato)
* bool Pop(&dato)
* bool pilaLlena()
* bool pilaVacia()
* bool imprimir()

cada función tiene un funcionamiento diferente que a continuación se describirá su algoritmo

* Pila()

Paso 1: Inicio

Paso 2: se inicializa la variable tamaño con el valor ingresado en el método Tam = 0

Paso 3: Fin

* bool Push(dato)

Paso 1:inicio

Paso 2:si(top >= maximo)

Entonces regresa falso

Si (top < maximo)

Entonces pila[++top]=dato regresa verdadero

Paso 3:Final

* bool Pop(&dato)

paso 1:incio

Paso 2:si(top == -1)

Entonces regresa falso

Si no

Entonces datos=pila[top--] regresa verdadero

Paso 3:Final

* bool pilaLlena()

paso 1:inicio

Paso 2:si(top == maximo-1)

Entonces regresa verdadero

Si no

regresa falso

Paso 3:Final

* bool pilaVacia()

paso 1:inicio

Paso 2:si(top == -1)

Entonces regresa verdadero

Si no

regresa falso

Paso 3:Final

* bool imprimir()

paso 1:inicio

Paso 2:hasta que(i == top)

Imprime pila[i]

Paso 3:Final

**Descripción Técnica**

**Función push**

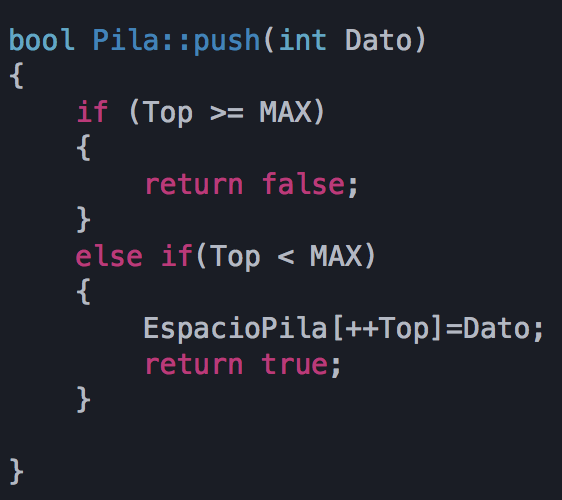
**Argumentos:** la función tiene como parámetro a ‘dato’ que sirve para el ingreso del valor de la pila

**Variables:**

* top: Es la posición del ultimo valor de la pila.
* EspacioPila[ ]: Arreglo de la pila.

**Lógica:**

La función recibe el valor que el usuario desea ingresar. Primero se verifica que la pila no esté llena mediante la condición. Si esta se cumple entonces devuelve un falso, entonces termina la función . si no entonces aumenta el valor de top en 1 para ser capaz de almacenar un nuevo elemento en la pila. Agregamos al arreglo el valor de dato en la posición de top. Devolvemos un valor true para indicar que la operación push se realizó con éxito.

****

**Función pop**

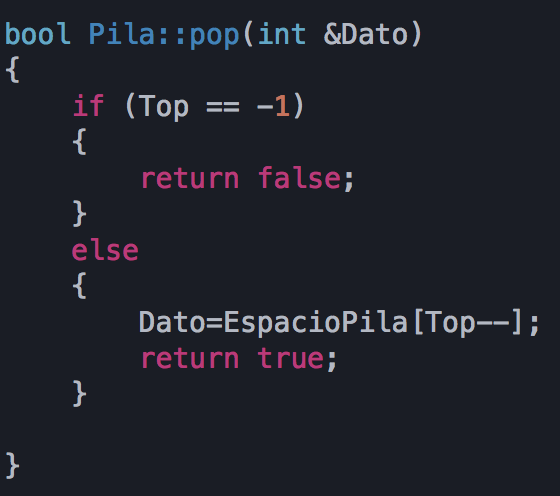
**Argumentos:** la función recibe la variable dato referenciado que sirve para eliminar valores que están en el tope de la pila.

**Variables:**

* top: Es la posición del ultimo valor de la pila.
* EspacioPila[ ]: Arreglo de la pila.

**Lógica:**

La función recibe el valor por referencia para poder eliminar el valor que se encuentra al final de nuestra pila. Para ello, primero validaremos que la pila no se encuentre vacía. Si se encuentra vacía se regresara un false si no entonces el dato será el valor de pila en la posición top y al finalizar esta accion se disminuirá el valor de top.

****

**Función pilaLlena**

**Argumentos:** No recibe valores

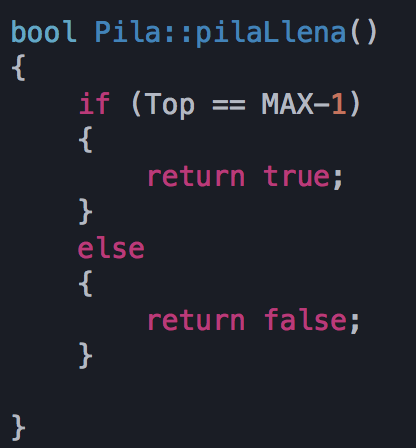
**Variables:**

* top: Es la posición del ultimo valor de la pila.
* MAX : Tamaño máximo de la pila

**Lógica:**

Lo que se evaluará esta función es si el arreglo de la pila se encuentra llena, verificando que el valor de nuestro tope sea igual al tamaño

de nuestro arreglo menos 1.si esto se cumple entonces se regresará un true pero sino entonces será un false.

****

**Función pilaVacia**

**Argumentos:** No recibe valores

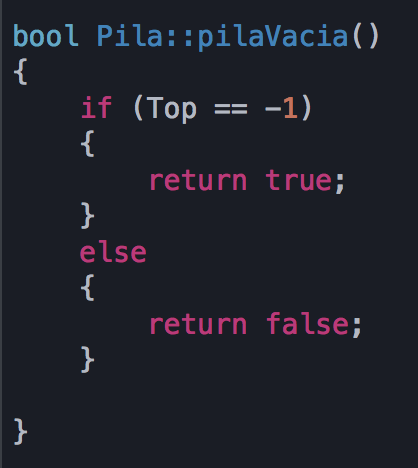
**Variables:**

top: Es la posición del ultimo valor de la pila

**Lógica:**

Lo que se evaluará esta función es si el arreglo de la pila se encuentra vacía. Por lo tanto, lo único que verificará es si el valor de nuestro tope es -1 .En caso de que esto se cumpla se devolverá un valor true y en caso contrario se

devolverá un valor false

****

**Función imprimrirPila**

**Argumentos:** No recibe valores

**Variables:**

* top: Es la posición del ultimo valor de la pila

**Lógica:**

Lo que se hace esta función es imprimir el valor actual que contiene el tope de

nuestro arreglo.

