Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

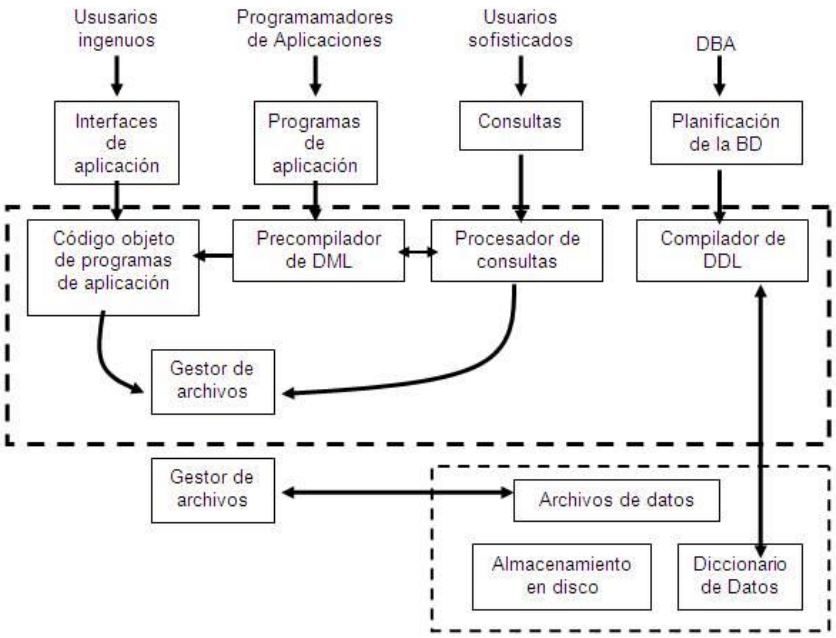
División de electrónica y computación

Ingeniería en computación

“Actividad 4: Aplicación de Pila y Cola”

Estructura de base de datos

Sergio Uribe Eusebio



Resumen personal del trabajo realizado, y forma en que fue abordado el problema

En este trabajo me sirvió mucho porque fue un gran aprendizaje para mí porque ha sido en el cual ya me ha quedado un poco más claro todo lo que aprendo en el aula de clases ya que ahí me quedan muchas dudas y solo las resuelvo programando en mi casa.

Una pila es una estructura de datos a la cual se puede acceder solo por un extremo de la misma. Las operaciones de inserción y extracción se realizan a través del tope, por lo cual no se puede acceder a cualquier elemento de la pila. Se la suele llamar estructura L.I.F.O. como acrónimo de las palabras inglesas "last in, first out" (último en entrar, primero en salir). La pila se considera un grupo ordenado de elementos, teniendo en cuenta que el orden de los mismos depende del tiempo que lleven "dentro" de la estructura.

Una cola es una colección de elementos homogéneos (almacenados en dicha estructura), en la misma se pueden insertar elementos por uno de los extremos, llamado frente, y retirar los mismos por el otro extremo, denominado final.

Es importante aclarar que, tanto el frente como el final de la cola, son los únicos indicados para retirar e insertar elementos, respectivamente. Esto nos indica que no podemos acceder directamente a cualquier elemento de la cola, sino solo al primero, o sea el que está o se encuentra en el frente, y no se pueden insertar elementos en cualquier posición sino solo por el final, así el elemento insertado queda como último.

Código

```
#include <iostream>

#include <string>

#include <cstring>

#include <stdlib.h>

#include "pila.h"
```

```
#include "cola.h"
using namespace std;



int main()
{
   Cola<char,50> myCol;
   Pila<char,50> myPila;
   myPila.inicializa();
   myCol.inicializa();
   int opcion;
   string a;
   cout << "Ingrese la cadena para convertirla" << endl;
   cin >> a;
   char *y= strdup(a.c_str());
   int x = a.length();
   for(int contador = 0; contador<=x; contador++)
   {
      char e = y[contador];
      myPila.push(e);
      myCol.enqueue(e);
   }
   do
   {
      cout << "Como la desea convertir" << endl;
      cout << "1. Notacion posfija (Pila)" << endl;
```

```
    cout << "2. Notacion infija (Cola)" << endl;
    cout << "3. Salir" << endl;
    cin >> opcion;
    switch (opcion)
    {
    case 1:
       system("cls");
       myPila.separar();
       myPila.imprimir();
       system ("pause");
       return 0;
       break;
    case 2:
       system("cls");
       myCol.imprimir();
       system ("pause");
       return 0;
       break;
    case 3:
       return 0;
       break;

    }
}
while (opcion !=3);
return 0;
```

```
}
#ifndef COLA_H_INCLUDED
#define COLA_H_INCLUDED
#include <iostream>

using namespace std;
template <class T, int arraySize = 1024>
class Cola
{
public:
   int frente;
   int final;
   T datos [arraySize];
   void inicializa();
   void enqueue(T e);
   void dequeue();
   void imprimir ();
   bool vacia();
   bool llena();
   T front();
};

template <class T, int arraySize>
void Cola<T, arraySize>::imprimir()
{
   for(int x = 0; x <final; x++)
```

```
    {
        cout << datos[x];
    }
    cout << endl;
}

template <class T, int arraySize>
void Cola<T, arraySize>::inicializa()
{
    frente = 0;
    final = 0;
}

template <class T, int arraySize>
bool Cola<T, arraySize>::vacia()
{
    if (frente = final+1)
    {
        return true;
    }
    else return false;
}

template <class T, int arraySize>
bool Cola<T, arraySize>::llena()
{
```

```
   if (frente = final + 2 or frente = 0 and final = 47 or frente = 1 and final =
48)
   {
      return true;
   }
   else return false;
}

template <class T, int arraySize>
void Cola<T, arraySize>::enqueue(T e)
{
   datos [final] = e;
   final ++;
}

template <class T, int arraySize>
void Cola<T, arraySize>::dequeue()
{
   frente = frente+1;
   if (frente = 49)
   {
      final = 0;
   };
}

template <class T, int arraySize>
T Cola<T, arraySize>::front()
```

```
{
   if(vacia())
   {
   } return datos[frente];
}
#endif // COLA_H_INCLUDED

#ifndef PILA_H_INCLUDED
#define PILA_H_INCLUDED
#include <iostream>

using namespace std;
class PilaException : std::exception     //las class ListException "hereda" parametros de std::exception
{
private:
   std::string msg;

public:
   explicit PilaException(const char* message) : msg(message) {}
   explicit PilaException(const std::string& message) : msg(message) {}
   virtual ~PilaException() throw () {}
   virtual const char* what() const throw ()
   {
      return msg.c_str();
   }
```

```
};

template <class T, int arraySize = 1024>
class Pila
{
public:
   T data[arraySize];
   int r [50];
   int inde [50];
   int tope;
   void inicializa ();
   void imprimir ();
   void push(T e);
   bool vacia();
   bool llena();
   T top();
   void pop();
   void separar();
private:
};

template <class T, int arraySize>
void Pila<T, arraySize>::separar()
{
   for(int i = 0; i <tope; i++)
   {
```

```
        if (data[i] == '+' or data[i] == '-')
        {
            r [i]= 1;
        }
        else if (data[i] == '*' or data[i] == '/')
        {
            r [i]= 2;
        }
        else if (data[i] == '^')
        {
            r [i]= 3;
        }
        else r[i] = 0;
    }
}

template <class T, int arraySize>
void Pila<T, arraySize>::imprimir()
{
    for(int i = 0; i <= tope; i++)
    {
        if(r[i] == 0)
        {
            cout << data[i];
        }
        else if (r[i]==3)
```

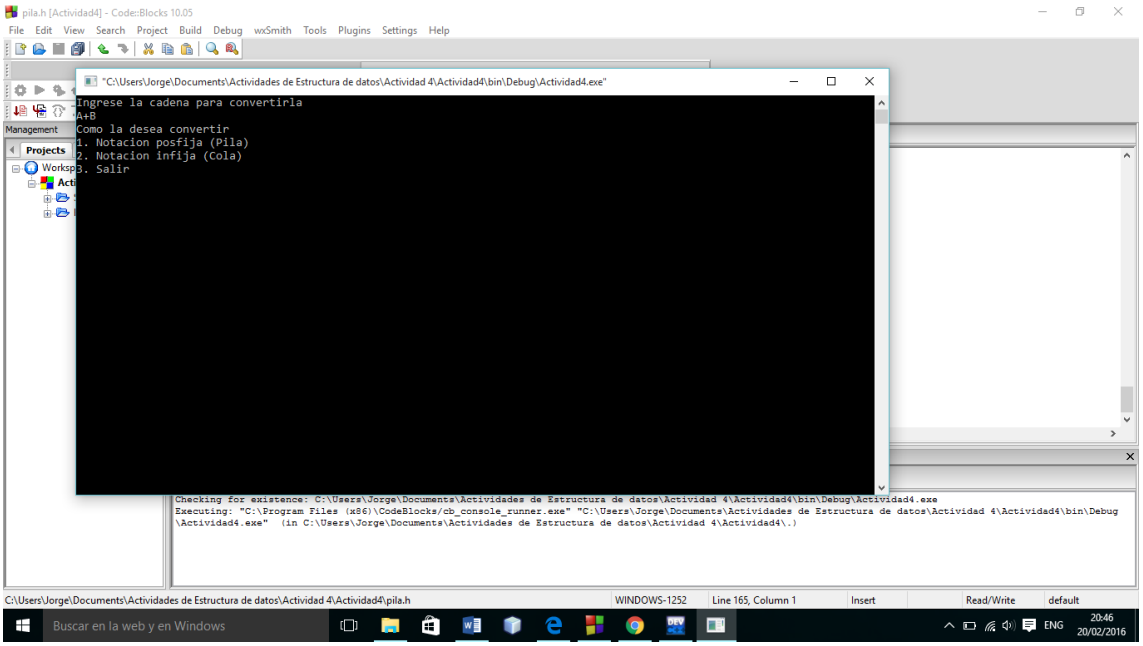
```
{
   inde[i]=i;
}
else if (r[i]==2)
{
   for(int y = 0; y<=tope; y++)
   {
      if(inde[y]!= -1)
      {
         cout << data[y];
         inde[y]=-1;
      }
   }

}
else if(r[i]==1)
{
   for(int p = 0; p<=tope; p++)
   {
      if(inde[p]!= -1)
      {
         cout << data[p];
         inde[p]=-1;
      }
   }
```

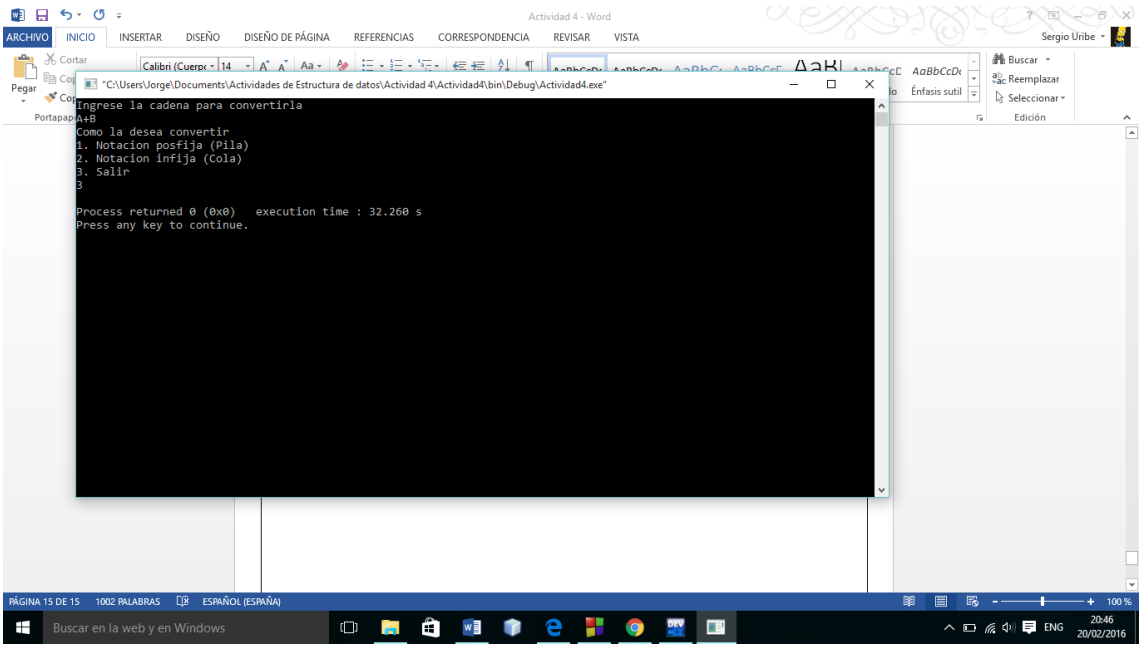
```
        }


    }
    cout << endl;
}

template <class T, int arraySize>
void Pila<T, arraySize>::inicializa()
{
    tope = -1;
    for(int g=0; g<50; g++)
    {
        inde [g]=-1;
    }
}

template <class T, int arraySize>
bool Pila<T, arraySize>::vacia()
{
    if (tope = -1)
    {
        return true;
    }
    else return false;
}
```

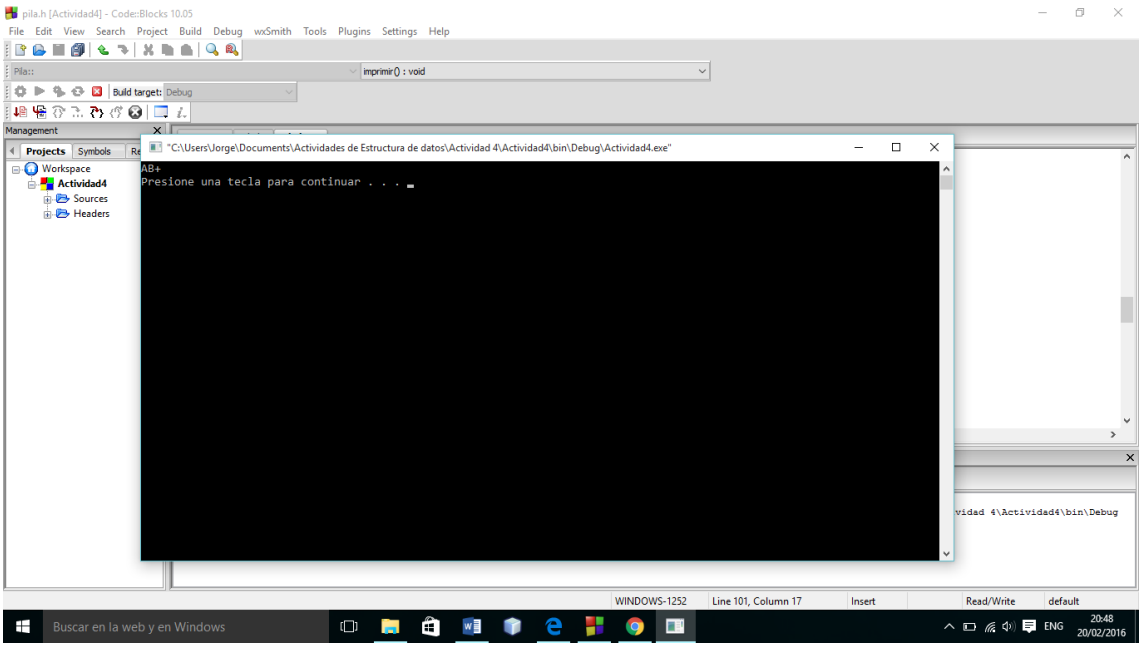
```
template <class T, int arraySize>
bool Pila<T, arraySize>::llena()
{
   if (tope = 49)
   {
      return true;
   }
   else return false;
}

template <class T, int arraySize>
void Pila<T, arraySize>::push(T e)
{
   tope ++;
   data[tope] = e;

}

template <class T, int arraySize>
void Pila<T, arraySize>::pop()
{
   if (vacia())
   {
      throw PilaException ("No se encuentra ningun dato");
   }
   else(top--);
```

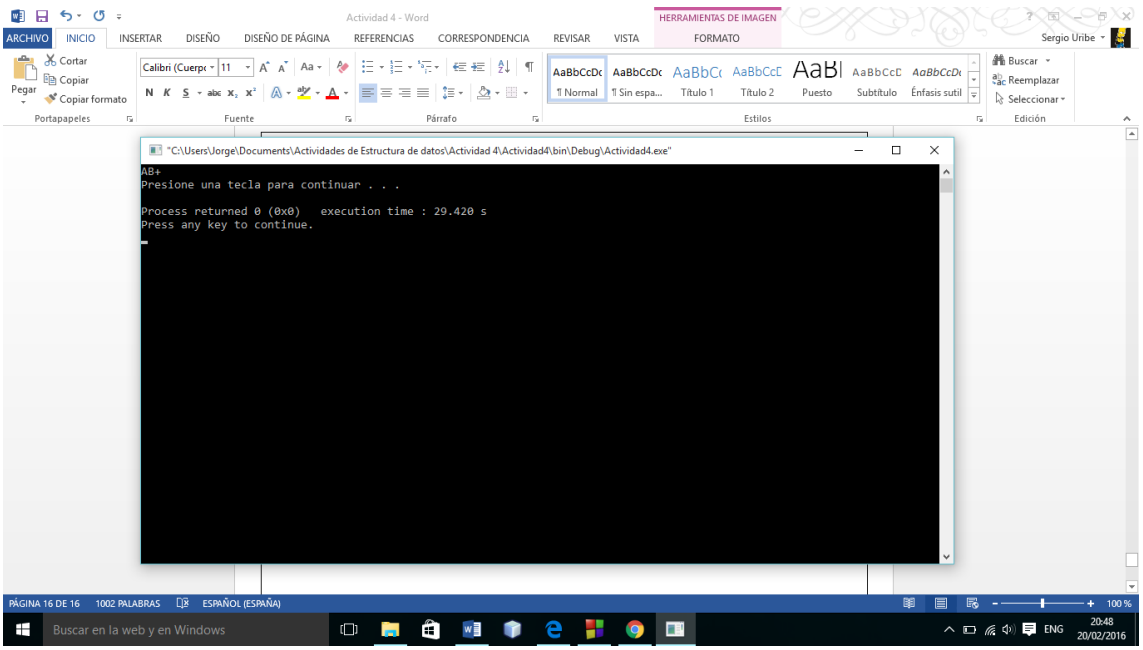
```
}

template <class T, int arraySize>
T Pila<T, arraySize>::top()
{
   if (vacia())
   {
      throw PilaException ("No se encuentra ningun dato");
   }
   else return data[tope];
}
#endif // PILA_H_INCLUDED
```

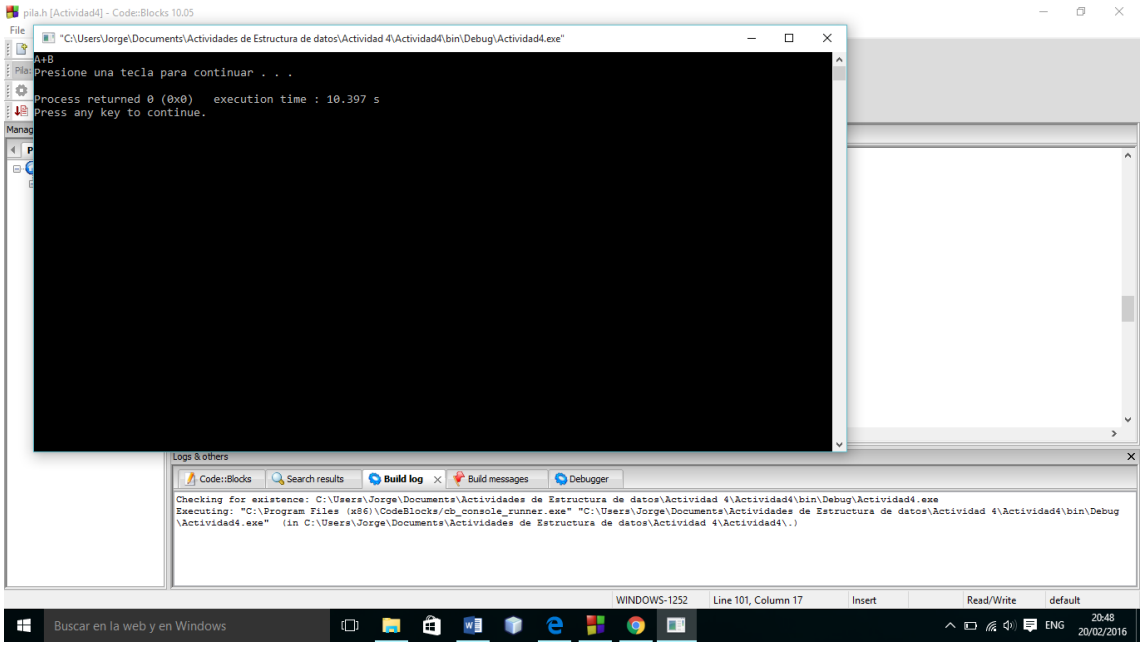
Capturas de pantalla






pila.h [Actividad4] - Code::Blocks 10.05
File Edit View Search Project Build Debug wxSmith Tools Plugins Settings Help
imprimir() : void
Build target: Debug
Management
Projects Symbols
Workspace
Actividad4
Sources
Headers
"C:\Users\Jorge\Documents\Actividades de Estructura de datos\Actividad 4\Actividad4\bin\Debug\Actividad4.exe"
AB+
Presione una tecla para continuar . . .
WINDOWS-1252 Line 101, Column 17 Insert Read/Write default
Buscar en la web y en Windows
ENG 20:48 20/02/2016


Actividad 4 - Word
ARCHIVO INICIO INSERTAR DISEÑO DISEÑO DE PÁGINA REFERENCIAS CORRESPONDENCIA REVISAR VISTA HERRAMIENTAS DE IMAGEN FORMATO
Sergio Uribe
Pegar Cortar Copiar Copiar formato
Portapapeles Fuente Párrafo Estilos Edición
Buscar Reemplazar Seleccionar
"C:\Users\Jorge\Documents\Actividades de Estructura de datos\Actividad 4\Actividad4\bin\Debug\Actividad4.exe"
AB+
Presione una tecla para continuar . . .
Process returned 0 (0x0) execution time : 29.420 s
Press any key to continue.
PÁGINA 16 DE 16 1002 PALABRAS ESPAÑOL (ESPAÑA) 100 %
Buscar en la web y en Windows
ENG 20:48 20/02/2016


pila.h [Actividad4] - Code::Blocks 10.05
"C:\Users\Jorge\Documents\Actividades de Estructura de datos\Actividad 4\Actividad4\bin\Debug\Actividad4.exe"
A+B
Presione una tecla para continuar . . .
Process returned 0 (0x0) execution time : 10.397 s
Press any key to continue.
Logs & others
Code::Blocks
Search results
Build log
Build messages
Debugger
Checking for existence: C:\Users\Jorge\Documents\Actividades de Estructura de datos\Actividad 4\Actividad4\bin\Debug\Actividad4.exe
Executing: "C:\Program Files (x86)\CodeBlocks/cb_console_runner.exe" "C:\Users\Jorge\Documents\Actividades de Estructura de datos\Actividad 4\Actividad4\bin\Debug\Actividad4.exe" (in C:\Users\Jorge\Documents\Actividades de Estructura de datos\Actividad 4\Actividad4\.)
WINDOWS-1252
Line 101, Column 17
Insert
Read/Write
default
Buscar en la web y en Windows
ENG
20:48
20/02/2016