

Tecnológico Nacional de México
Campus Querétaro.

Implemente Estructura de Colas (FIFO) en la simulación de atención de procesos.

Alumno:

Arellano Ochoa Daniel Ignacio

Carrera:

Ingeniera en sistemas computacionales

Docente:

Felipe Estrada Rojas

Fecha: 24/10/2020

Ejecución:

```
Proceso 21 Porcentaje: 77
Proceso 22 Porcentaje: 77
Proceso 23 Porcentaje: 97
Proceso 24 Porcentaje: 75
Proceso 25 Porcentaje: 77
Proceso 26 Porcentaje: 77
Proceso 27 Porcentaje: 22
Proceso 28 Porcentaje: 77
Proceso 29 Porcentaje: 55
Proceso 30 Porcentaje: 22
Proceso 31 Porcentaje: 77
Proceso 32 Porcentaje: 44
Proceso 33 Porcentaje: 77
Proceso 34 Porcentaje: 77
Proceso 35 Porcentaje: 55
Proceso 36 Porcentaje: 22
Proceso 37 Porcentaje: 99
Proceso 38 Porcentaje: 55
Proceso 39 Porcentaje: 22
Proceso 40 Porcentaje: 75
Proceso 41 Porcentaje: 77
Proceso 42 Porcentaje: 75
Proceso 43 Porcentaje: 77
Proceso 44 Porcentaje: 77
Proceso 45 Porcentaje: 22
Proceso 46 Porcentaje: 22
Proceso 47 Porcentaje: 75
Proceso 48 Porcentaje: 77
Proceso 49 Porcentaje: 77
Proceso 50 Porcentaje: 44
Proceso 51 Porcentaje: 77
Proceso 52 Porcentaje: 77
Proceso 53 Porcentaje: 55
Proceso 54 Porcentaje: 99
Proceso 55 Porcentaje: 75
Procesos con 0%: 67
PS C:\Users\Daniel\Desktop\cola>
```

En la ejecución anterior se generaron 203 procesos de los cuales 81 fueron completados al 100%, y de los 122 restantes 67 no fueron atendidos y 55 fueron atendidos por menos del 100%

```
Bienvenido al programa, se ejecutaran 100 minutos de procesamiento simulado
Se generaron: 497 procesos
Se atendieron al 100%: 47 procesos
En la cola hay: 450 procesos
Lista de procesos incompletos:
Proceso 1 Porcentaje: 77
Proceso 2 Porcentaje: 66
Proceso 3 Porcentaje: 77
Proceso 4 Porcentaje: 66
Proceso 5 Porcentaje: 77
Proceso 6 Porcentaje: 77
Proceso 7 Porcentaje: 66
Proceso 8 Porcentaje: 77
Proceso 9 Porcentaje: 66
Proceso 10 Porcentaje: 77
Proceso 11 Porcentaje: 77
Proceso 12 Porcentaje: 66
Proceso 13 Porcentaje: 77
Proceso 14 Porcentaje: 66
Proceso 15 Porcentaje: 77
Proceso 16 Porcentaje: 66
Proceso 17 Porcentaje: 77
Proceso 18 Porcentaje: 66
Proceso 19 Porcentaje: 77
Proceso 20 Porcentaje: 66
Proceso 21 Porcentaje: 77
Proceso 22 Porcentaje: 66
Proceso 23 Porcentaje: 77
Proceso 24 Porcentaje: 66
Proceso 25 Porcentaje: 77
Proceso 26 Porcentaje: 66
Proceso 27 Porcentaje: 77
Proceso 28 Porcentaje: 66
Proceso 29 Porcentaje: 77
Proceso 30 Porcentaje: 66
Proceso 31 Porcentaje: 77
Proceso 32 Porcentaje: 66
```

```
Proceso 31 Porcentaje: 77
Proceso 32 Porcentaje: 66
Proceso 33 Porcentaje: 77
Proceso 34 Porcentaje: 66
Proceso 35 Porcentaje: 77
Proceso 36 Porcentaje: 66
Proceso 37 Porcentaje: 77
Proceso 38 Porcentaje: 66
Proceso 39 Porcentaje: 77
Proceso 40 Porcentaje: 66
Proceso 41 Porcentaje: 66
Proceso 42 Porcentaje: 77
Proceso 43 Porcentaje: 66
Proceso 44 Porcentaje: 77
Proceso 45 Porcentaje: 66
Proceso 46 Porcentaje: 66
Proceso 47 Porcentaje: 77
Proceso 48 Porcentaje: 66
Proceso 49 Porcentaje: 77
Proceso 50 Porcentaje: 66
Proceso 51 Porcentaje: 77
Proceso 52 Porcentaje: 66
Proceso 53 Porcentaje: 77
Proceso 54 Porcentaje: 66
Proceso 55 Porcentaje: 77
Proceso 56 Porcentaje: 66
Proceso 57 Porcentaje: 77
Proceso 58 Porcentaje: 66
Proceso 59 Porcentaje: 77
Proceso 60 Porcentaje: 66
Proceso 61 Porcentaje: 77
Proceso 62 Porcentaje: 66
Proceso 63 Porcentaje: 77
Proceso 64 Porcentaje: 66
Proceso 65 Porcentaje: 77
Proceso 66 Porcentaje: 66
Proceso 67 Porcentaje: 77
```

```
Proceso 66 Porcentaje: 66
Proceso 67 Porcentaje: 77
Proceso 68 Porcentaje: 66
Proceso 69 Porcentaje: 77
Proceso 70 Porcentaje: 66
Proceso 71 Porcentaje: 77
Proceso 72 Porcentaje: 66
Proceso 73 Porcentaje: 77
Proceso 74 Porcentaje: 66
Proceso 75 Porcentaje: 77
Proceso 76 Porcentaje: 66
Proceso 77 Porcentaje: 77
Proceso 78 Porcentaje: 66
Proceso 79 Porcentaje: 77
Proceso 80 Porcentaje: 66
Proceso 81 Porcentaje: 77
Proceso 82 Porcentaje: 66
Proceso 83 Porcentaje: 77
Proceso 84 Porcentaje: 66
Proceso 85 Porcentaje: 77
Proceso 86 Porcentaje: 66
Proceso 87 Porcentaje: 77
Proceso 88 Porcentaje: 66
Proceso 89 Porcentaje: 77
Proceso 90 Porcentaje: 66
Proceso 91 Porcentaje: 77
Proceso 92 Porcentaje: 66
Proceso 93 Porcentaje: 77
Proceso 94 Porcentaje: 66
Proceso 95 Porcentaje: 77
Proceso 96 Porcentaje: 66
Proceso 97 Porcentaje: 77
Proceso 98 Porcentaje: 66
Proceso 99 Porcentaje: 77
Proceso 100 Porcentaje: 66
Proceso 101 Porcentaje: 77
Proceso 102 Porcentaje: 66
```

```
Proceso 72 Porcentaje: 66
Proceso 73 Porcentaje: 77
Proceso 74 Porcentaje: 66
Proceso 75 Porcentaje: 77
Proceso 76 Porcentaje: 66
Proceso 77 Porcentaje: 77
Proceso 78 Porcentaje: 66
Proceso 79 Porcentaje: 77
Proceso 80 Porcentaje: 66
Proceso 81 Porcentaje: 77
Proceso 82 Porcentaje: 66
Proceso 83 Porcentaje: 77
Proceso 84 Porcentaje: 66
Proceso 85 Porcentaje: 77
Proceso 86 Porcentaje: 66
Proceso 87 Porcentaje: 77
Proceso 88 Porcentaje: 66
Proceso 89 Porcentaje: 77
Proceso 90 Porcentaje: 66
Proceso 91 Porcentaje: 77
Proceso 92 Porcentaje: 66
Proceso 93 Porcentaje: 77
Proceso 94 Porcentaje: 66
Proceso 95 Porcentaje: 77
Proceso 96 Porcentaje: 66
Proceso 97 Porcentaje: 77
Proceso 98 Porcentaje: 66
Proceso 99 Porcentaje: 77
Proceso 100 Porcentaje: 66
Proceso 101 Porcentaje: 77
Proceso 102 Porcentaje: 66
Proceso 103 Porcentaje: 77
Proceso 104 Porcentaje: 66
Proceso 105 Porcentaje: 77
Proceso 106 Porcentaje: 77
Procesos con 0%: 344
PS C:\Users\Daniel\Desktop\col
```

En la ejecución anterior se generaron 497 procesos de los cuales 47 fueron completados al 100%, y de los 450 restantes, 344 no fueron atendidos y 106 fueron atendidos por menos del 100%

Conclusión:

El uso de las pilas puede llegar a ser muy útil como herramienta al momento de realizar algunos programas específicos que aunque pueden ser solucionados ocupando otras herramientas como pilas o listas, es importante saber en qué momentos se ocupan las colas pues este tipo de estructuras manejan ciertos tipos de funciones que la hacen óptima para solucionar algunos problemas.

Código:

```
/*Nombre: Arellano Ochoa Daniel Ignacio
No. de control: 19141118
Programa que simula el procesamiento durante 100 minutos en los cuales por cada minutos se genraran n procesos y se atenderan m procesos donde m y n son valores aletarios de 0 a 5
*/
#include <string>
#include <iostream>
#include "Procesador.hpp"
#include "Proceso.hpp"

using namespace std;

//Menu de inicio del programa
int main(int argc, const char * argv[]){

    cout<<"Bienvenido al programa, se ejecutaran 100 minutos de procesamiento simulado"<<endl;
    Procesador* intel=new Procesador();
    intel->procesar();

}
```

```
#ifndef Procesador_hpp
#define Procesador_hpp
#include "Proceso.hpp"

class Procesador
{
private:
    Proceso* inicio;
    Proceso* final;
    int proGenerados;
```

```

    int proAtendidos;
    int proCola;
    int proCero;

public:
    Procesador();

    void insertar(Proceso*);
    void sacar();
    void generarProcesos(int*);
    void atenderProcesos(int*);
    void procesar();
    void procesosColaInc();

};

#endif /*Procesador_hpp*/

```

```

#include "Procesador.hpp"
#include "Proceso.hpp"
#include <string>
#include <iostream>
#include <stdlib.h>
#include <time.h>
//clase de procesador
using namespace std;
//constructor del procesador, con datos inicializados
Procesador::Procesador(){
    this->inicio=NULL;//proceso que se encuentra al inicio de la cola
    this->final=NULL;//procesos que se encuentnra al final de la cola
    this->proGenerados=0;//procesos generados
    this->proAtendidos=0;//procesos atendido al 100%
    this->proCola=0;//procesos atendidos de 0% a 99%
    this->proCero=0;//procesos no atendido
}

//metodo para insertar un procesos dentro de la cola
void Procesador::insertar(Proceso* p){
    if(this->proCola==0){
        this->inicio=p;
        this->final=p;
    }else{
        this->final->setAnt(p);
    }
}

```

```

        this->final=p;
    }
    this->proCola++;
}
//metodo para sacar el proceso que se encuentra al inicio de la cola
void Procesador::sacar(){
    this->inicio=this->inicio->getAnt();
    this->proCola--;
}
//metodo que generar N procesos, insertandolos a la cola
void Procesador::generarProcesos(int* N){
    Proceso* pro;
    for (size_t i = 0; i < *N; i++)
    {
        pro=new Proceso();
        this->insertar(pro);
        this->proGenerados++;
    }
}
//metodo que atiende M procesos, verificando si se logro atender al 100%, si
//es el caso contrario se volvera a formar dentro de la cola
void Procesador::atenderProcesos(int* M){
    srand(time(NULL));
    int pAux=0;
    Proceso* proAux;
    for (size_t i = 0; i < *M; i++)
    {
        if(this->proCola>0){
            proAux=this->inicio;
            this->sacar();
            pAux=rand()%100 + 1;
            pAux= pAux + proAux->getPorcentaje();
            proAux->setPorcentaje(&pAux);

            if(proAux->getPorcentaje()<100){
                this->insertar(proAux);
            }else{
                this->proAtendidos++;
            }
        }
    }
}
}

```

```

}
//metodo que generar y atiende N y M procesos respectivamente, durante 100
ciclos (100 minutos simulados), dando informacion al final
void Procesador::procesar(){
    srand(time(NULL));
    int M=0;//proceso a atender
    int N=0;//procesos a generar

    for (size_t i = 0; i < 100; i++)
    {
        //cout<<"minuto"<<i<<endl;
        M= rand()%6;
        N= rand()%6;
        //cout<<"M= "<<M<<" N= "<<N<<endl;
        this->generarProcesos(&N);
        if(M>this->proCola){
            M=this->proCola;
            this->atenderProcesos(&M);
        }else{
            this->atenderProcesos(&M);
        }
    }

    cout<<"Se generaron: "<<this->proGenerados<<" procesos"<<endl;
    cout<<"Se atendieron al 100%: "<<this->proAtendidos<<" procesos"<<endl;
    cout<<"En la cola hay: "<<this->proCola<<" procesos"<<endl;
    cout<<"Lista de procesos incompletos:"<<endl;
    this->procesosColaInc();
}

//metodo que cuenta los procesos que quedaron sin atender y enumero e imprim
e los procesos que fueron atendidos pero no al 100%
void Procesador::procesosColaInc(){
    Proceso* auxPro;
    int i=1;
    int aux= this->proCola;
    while (aux>0)
    {
        auxPro=this->inicio;
        this->sacar();
        if(auxPro->getPorcentaje()>0){
            cout<<"Proceso "<<i<<" Porcentaje: "<<auxPro-
>getPorcentaje()<<endl;
            i++;
        }else{

```

```

        this->proCero++;
    }
    this->insertar(auxPro);
    aux--;
}
cout<<"Procesos con 0%: "<<this->proCero<<endl;
}

```

```

#ifndef Proceso_hpp
#define Proceso_hpp

using namespace std;
class Proceso
{
private:
    int porcentaje;
    Proceso* ant;
public:
    Proceso();
    int getPorcentaje();
    Proceso* getAnt();
    void setPorcentaje(int*);
    void setAnt(Proceso*);
};

#endif /*Proceso_hpp*/

```

```

#include "Proceso.hpp"
#include <iostream>
#include <string>
//clase de proceso
using namespace std;
//constructor con los atributos de porcentaje y ant inicializados
Proceso::Proceso(){
    this->porcentaje=0;//porcentaje atendido
    this->ant=NULL;//procesos atras de este dentro de una cola
}
//metodo que regresa el porcentaje que ha sido atendido este proceso
int Proceso::getPorcentaje(){
    return this->porcentaje;
}

```



```
//metodo que refresca el proceso que se encuentra anterior a este dentro de una cola
Proceso* Proceso::getAnt(){
    return this->ant;
}
//metodo que permite modificar el proceso que se encuentra atras de este dentro de una cola
void Proceso::setAnt(Proceso* p){
    this->ant=p;
}
//metodo que permite modificar el porcentaje atendido de este proceso
void Proceso::setPorcentaje(int* n){
    this->porcentaje=*n;
}
```