Tecnológico Nacional de México Campus Querétaro.

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

Alumno: Arellano Ochoa Daniel Ignacio

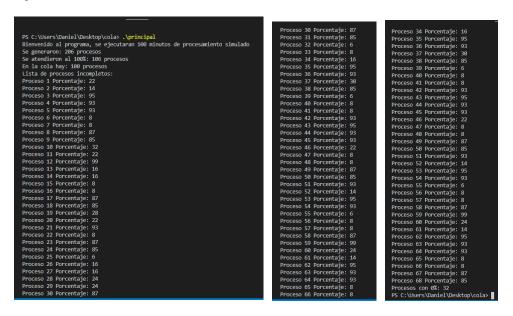
Carrera: Ingeniera en sistemas computacionales

> Docente: Felipe Estrada Rojas

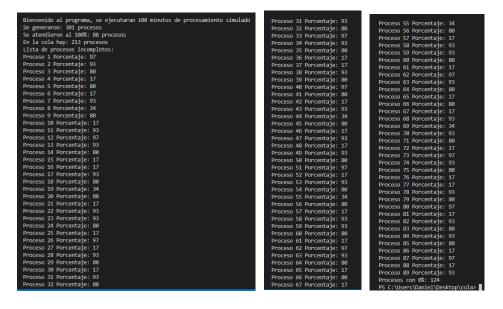
Fecha: 24/10/2020

Código: El archivo cpp se anexo junto con este trabajo, de igual manera el código se encuentra hasta el final de este trabajo.

Ejecución:



En la ejecución anterior se generaron 206 procesos de los cuales 106 fueron completados al 100%, y de los 100 restantes 32 no fueron atendidos y 68 fueron atendidos por menos del 100%



En la ejecución anterior se generaron 301 procesos de los cuales 88 fueron completados al 100%, y de los 213 restantes, 124 no fueron atendidos y 89 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 c
ada 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 procesamient</pre>
o simulado"<<endl;</pre>
    Procesador* intel=new Procesador();
    intel->procesar();
```

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

```
#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 <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 encuentra al final de la cola
    this->proGenerados=0;//procesos generados
    this->proAtendidos=0;//procesos atendido al 100%
    this->proCola=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 encuetra 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;
   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){</pre>
                this->insertar(proAux);
            }else{
```

```
this->proAtendidos++;
    }
//metodo que generar y ateiende N y M procesos respectivamente, durante 100
ciclos (100 minutos simulados), dando informacion al final
void Procesador::procesar(){
    srand(time(NULL));
   int M;//atender
   int N;//generar
    for (size_t i = 0; i < 100; i++)
        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);
            this->atenderProcesos(&M);
    cout<<"Se generaron: "<<this->proGenerados<<" procesos"<<endl;</pre>
    cout<<"Se atendieron al 100%: "<<this->proAtendidos<<" procesos"<<endl;</pre>
    cout<<"En la cola hay: "<<this->proCola<<" procesos"<<endl;</pre>
    cout<<"Lista de procesos incompletos:"<<endl;</pre>
    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;
   auxPro=this->inicio;
   int i=1;
   bool auxF=true;
   while (auxPro!=NULL && auxF==true)
```

```
{
    if(auxPro == this->final)
        auxF=false;
    if(auxPro->getPorcentaje()>0){
        cout<<"Proceso "<<i<" Porcentaje: "<<auxPro-
>getPorcentaje()<<endl;
        i++;
    }else{
        this->procero++;
    }
    auxPro=auxPro->getAnt();
}
cout<<"Procesos con 0%: "<<this->procero<<endl;
}</pre>
```