**Proyecto Sistema Distribuido de Servicios de Taquilla Virtual**

Memoria Grupo C

PCCD - 2023/2024

Por:

Bedriñana Cárdenas, Renato

Blanco Demelo, Hugo

Blanco Pumar, Pablo

Esparis Matanza, Santiago

**Índice**

[**1. Pseudocódigo final. 3**](#_9gd4lv9f1cxa)

[1.1. Main. 3](#_tq5uhzomh3e5)

[1.2. Receiver. 4](#_kpvx6wu17c2j)

[1.3. Lectores - Consultas 5](#_1uax5nsbjmxz)

[1.4. Escritores - Reservas, Pagos, Administración y Anulaciones. 6](#_pdg5w17gkfsk)

[**2. Pruebas funcionales. 7**](#_rwd8g0jkgwmt)

[2.1. Evolución 4 Nodos. 7](#_r6jqk0wbi1wm)

[2.2. Evolución 6 Nodos. 7](#_q443t32s86c7)

[2.3. Evolución 10 Nodos - No funcional. 7](#_5ppyoratu8ah)

[**3. Métricas. 8**](#_a3ivfz47xac3)

[3.1. Tiempo que pasa un nodo fuera de la SC. 8](#_foaqg1w4ems1)

[3.2. Tiempo que pasa un proceso hasta que entra en la SC. 9](#_vnsagg1c446c)

[3.3. Tiempo medio de vida de un proceso. 10](#_l7p1b82dee7b)

[3.4. Número medio de mensajes por nodo. 11](#_wgdkf3p8ftp5)

# Pseudocódigo final.

En este apartado representaremos el pseudocódigo de los procesos principales necesarios para desarrollar este proyecto.

## 1.1. Main.

int main() {

obtenerBuzón ();

inicializarSemaforos ();

crearProcesosRandom();

while (1) {

impirmeMenu ();

switch (opcion) {

case 1:

sem\_post(procesoConcreto);

break;

case 2:

sem\_post(secuenciaProcesos);

break;

case 3:

exit();

break;

}

}

}

## 1.2. Receiver.

confirmaciones=0, num\_nodos, num\_procesos, estoy\_SC=0, max\_ticket=0, mi\_ticket, mi\_prioridad=0, quiero=0, SC\_consultas, vector\_peticiones[NUM\_PRIORIDADES]={0}, semaforos\_de\_paso[num\_procesos], flag\_pedir\_otra\_vez[NUM\_PRIORIDADES]={0}

-------------------------------------------------------------------------------------------------------------------

Proceso Receive{

receive(tipo\_origen,{id\_nodo\_origen,ticket\_origen,prioridad\_origen,flag\_consulta})

max\_ticket=MAX(max\_ticket,ticket\_origen)

if(tipo\_origen==confirmacion AND ticket\_origen==mi\_ticket){

confirmaciones++

if(confirmaciones==num\_nodos){

mi\_prioridad=0

confirmaciones=0

dar\_SC(ticket\_origen) // nos confirman a la solicitud con el ticket

}

}

else if (tipo\_origen==peticion AND NOT quiero){

send(confirmacion,id\_nodo\_origen,{mi\_id,ticket\_origen})

}

else if (tipo\_origen==peticion AND NOT estoy\_SC){

if(prioridad\_origen>mi\_prioridad){

send(confirmacion,id\_nodo\_origen,{mi\_id,ticket\_origen})

cancelar\_mi\_solicitud();

}

else if (prioridad\_origen==mi\_prioridad AND (ticket\_origen<mi\_ticket OR

(ticket\_origen==mi\_ticket AND id\_nodo\_origen<mi\_id))){

send(confirmacion,id\_nodo\_origen,{mi\_id,ticket\_origen})

}

else {

id\_nodos\_pend[++nodos\_pend] = id\_nodo\_origen

ticket\_nodos\_pend[nodos\_pend] = ticket\_origen

}

}

else if (estoy\_SC AND flag\_consulta AND SC\_consultas){

send(confirmacion,id\_nodo\_origen,{mi\_id,ticket\_origen})

}

else{

id\_nodos\_pend[++nodos\_pend] = id\_nodo\_origen

ticket\_nodos\_pend[nodos\_pend] = ticket\_origen

}

}

## 1.3. Lectores - Consultas

void lector () {

while(1){

sem\_wait(&solicitar\_SC]);

solicitar\_SC(nro\_proceso,prioridad,1);

sem\_wait(&sem\_ProtegeLectores);

contadorLectores ++;

if (contadorLectores == 1){

sem\_wait(&sem\_exclusionMutuaEscritor);

SC\_consultas = 1;

}

procesos\_permitidos\_en\_SC++;

sem\_post(&sem\_ProtegeLectores);

Sección\_Critica();

sem\_wait(&sem\_ProtegeLectores);

contadorLectores--;

procesos\_permitidos\_en\_SC--;

if (contadorLectores == 0) {

liberar\_SC();

SC\_consultas = 0;

sem\_post (&sem\_exclusionMutuaEscritor);

} else sem\_post(&sem\_exclusionMutuaEscritor);

sem\_post(&sem\_ProtegeLectores);

}

}

## 1.4. Escritores - Reservas, Pagos, Administración y Anulaciones.

void escritor () {

while (1) {

sem\_wait (&solicitar\_SC)

solicitar\_SC ();

sem\_wait (&sem\_procesos\_permitidos\_en\_SC);

sem\_procesos\_permitidos\_en\_SC++;

sem\_post (&procesos\_permitidos\_SC);

sem\_wait (&exclusionMutua\_Escritores);

– – SC – –

sem\_wait (&sem\_procesos\_permitidos\_en\_SC);

procesos\_permitidos\_en\_SC–;

if (procesos\_permitidos\_en\_SC == 0 ) {

liberar\_SC();

sem\_post (&sem\_exclusionMutuaEscritor);

}

else {

sem\_post (&sem\_procesos\_permitidos\_en\_SC);

sem\_post (&sem\_exclusionMutuaEscritor);

}

}

}

# Pruebas funcionales.

Existen tres versiones diferentes del código implementado, cada uno para las diferentes evoluciones de 4, 6 y 10 nodos con sus respectivos procesos máximos.

## 2.1. Evolución 4 Nodos.

Se debe ejecutar *./tickets.c idNodo* en cuatro terminales diferentes. Una vez inicializados, se presiona *Enter* en todas las terminales.

De esta manera, cada nodo genera 10 procesos aleatorios entre Consultas y Pagos, después procederá a imprimir el menú. En este menú se puede introducir procesos de uno en uno en la sección crítica o se puede introducir una secuencia de procesos para que puedan disputar la sección crítica,

## 2.2. Evolución 6 Nodos.

Se debe ejecutar *./tickets.c idNodo* en seis terminales diferentes. Una vez inicializados, se presiona *Enter* en todas las terminales.

De esta manera, cada nodo genera 20 procesos aleatorios y procede a imprimir el menú. En este menú se puede introducir procesos de uno en uno en la sección crítica o se puede introducir una secuencia de procesos para que puedan disputar la sección crítica,

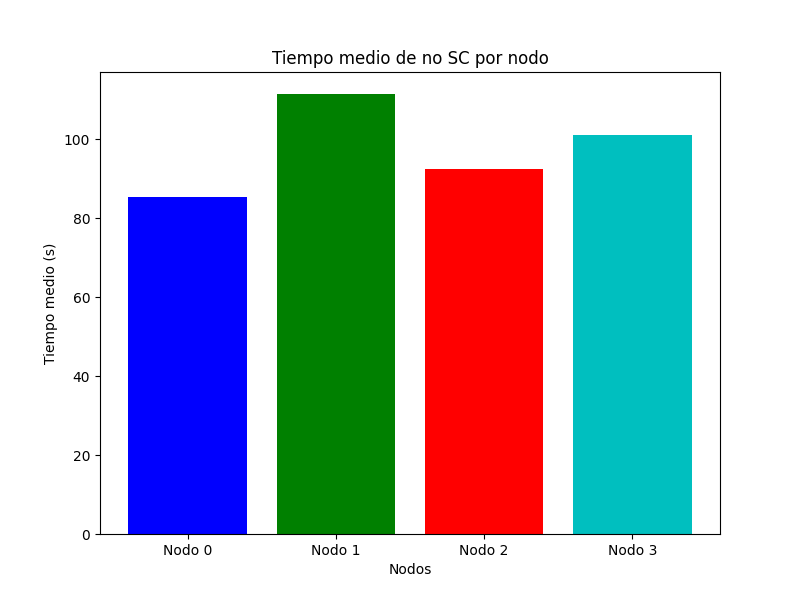
## 2.3. Evolución 10 Nodos - No funcional.

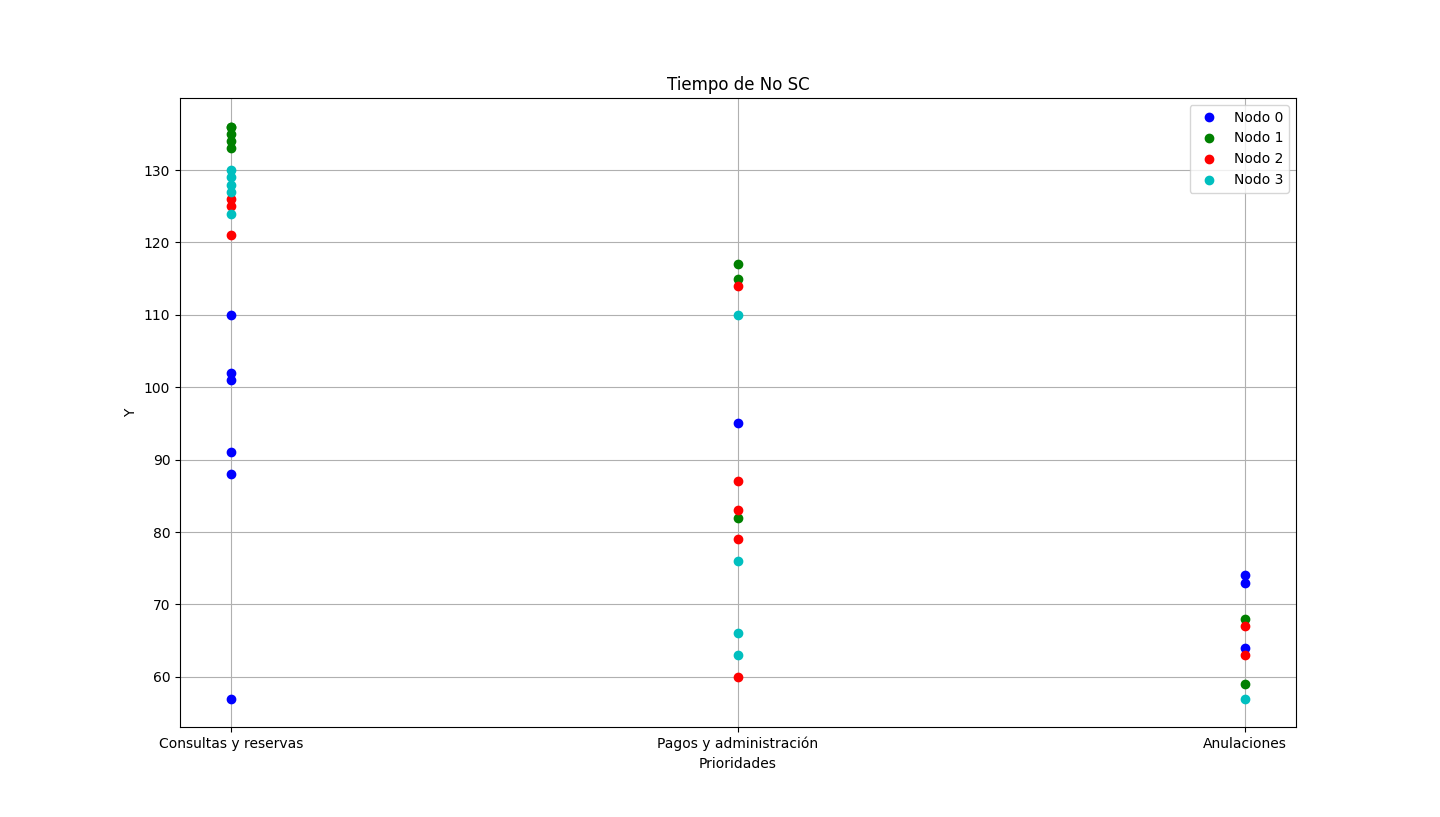
Se debe ejecutar *./tickets.c idNodo* en diez terminales diferentes. Una vez inicializados, se presiona *Enter* en todas las terminales.

De esta manera, cada nodo genera 100 procesos aleatorios y procede a imprimir el menú. En este menú se puede introducir procesos de uno en uno en la sección crítica o se puede introducir una secuencia de procesos para que puedan disputar la sección crítica,

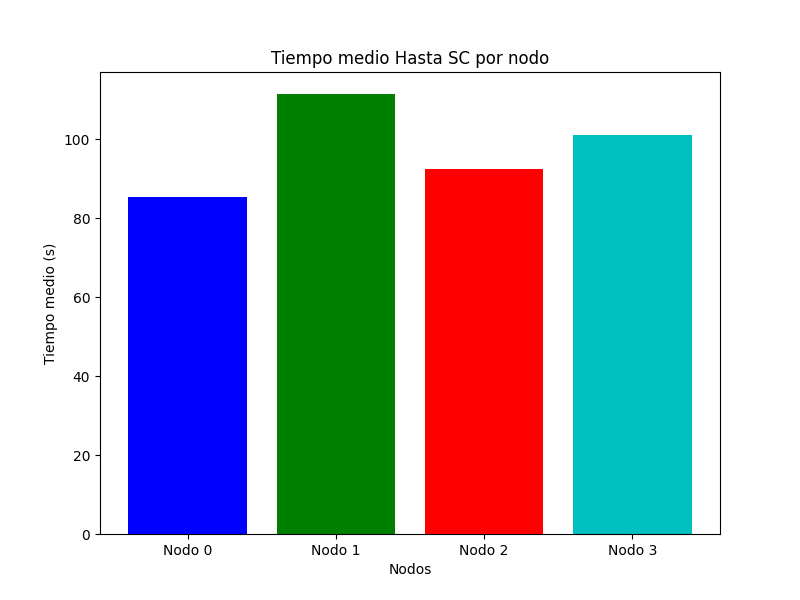
# Métricas.

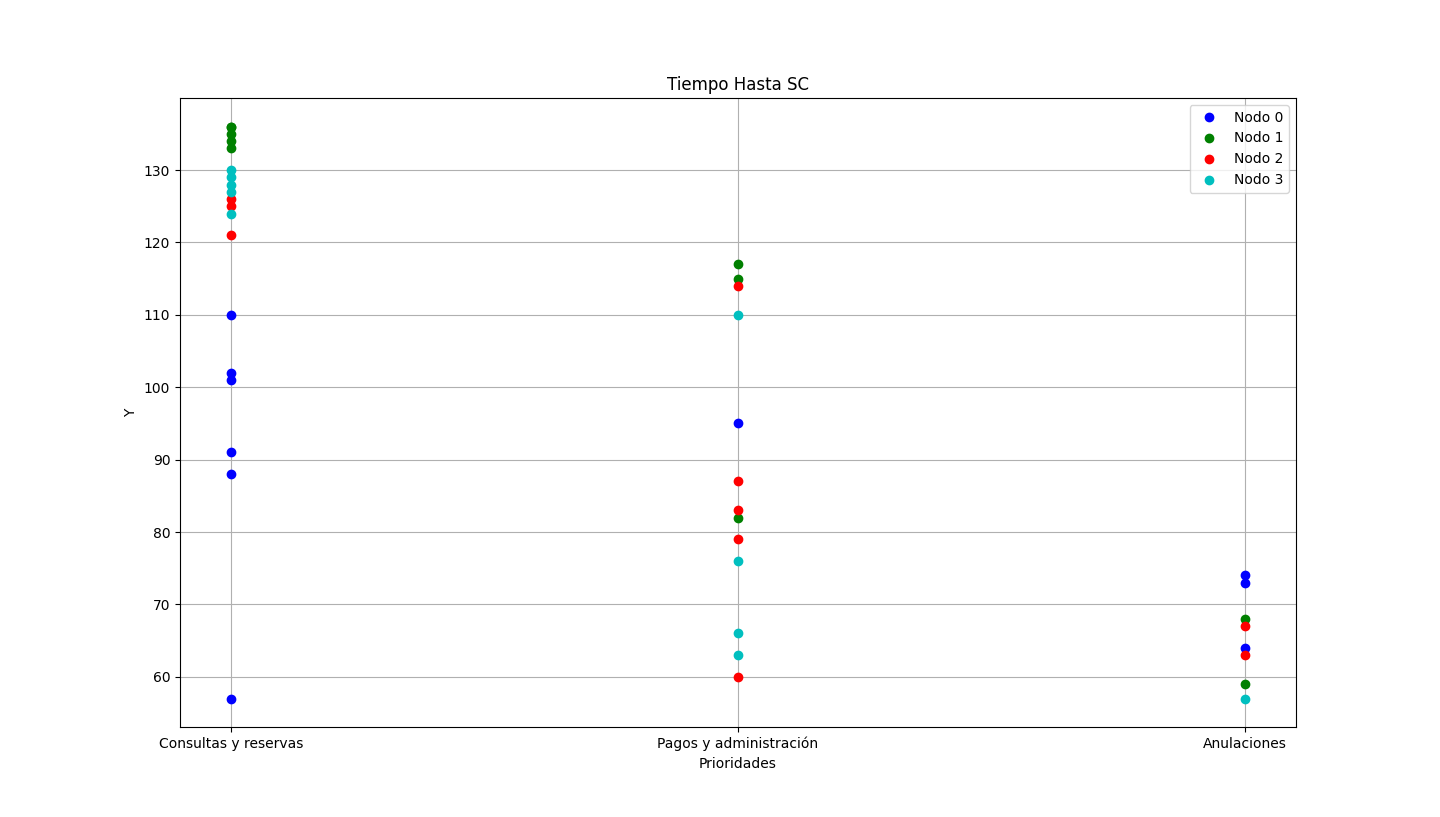
## 3.1. Tiempo que pasa un nodo fuera de la SC.



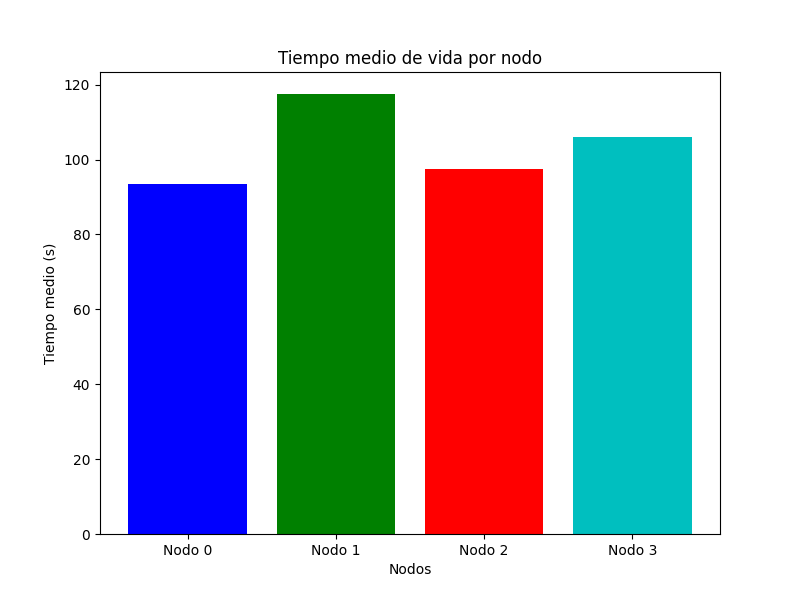


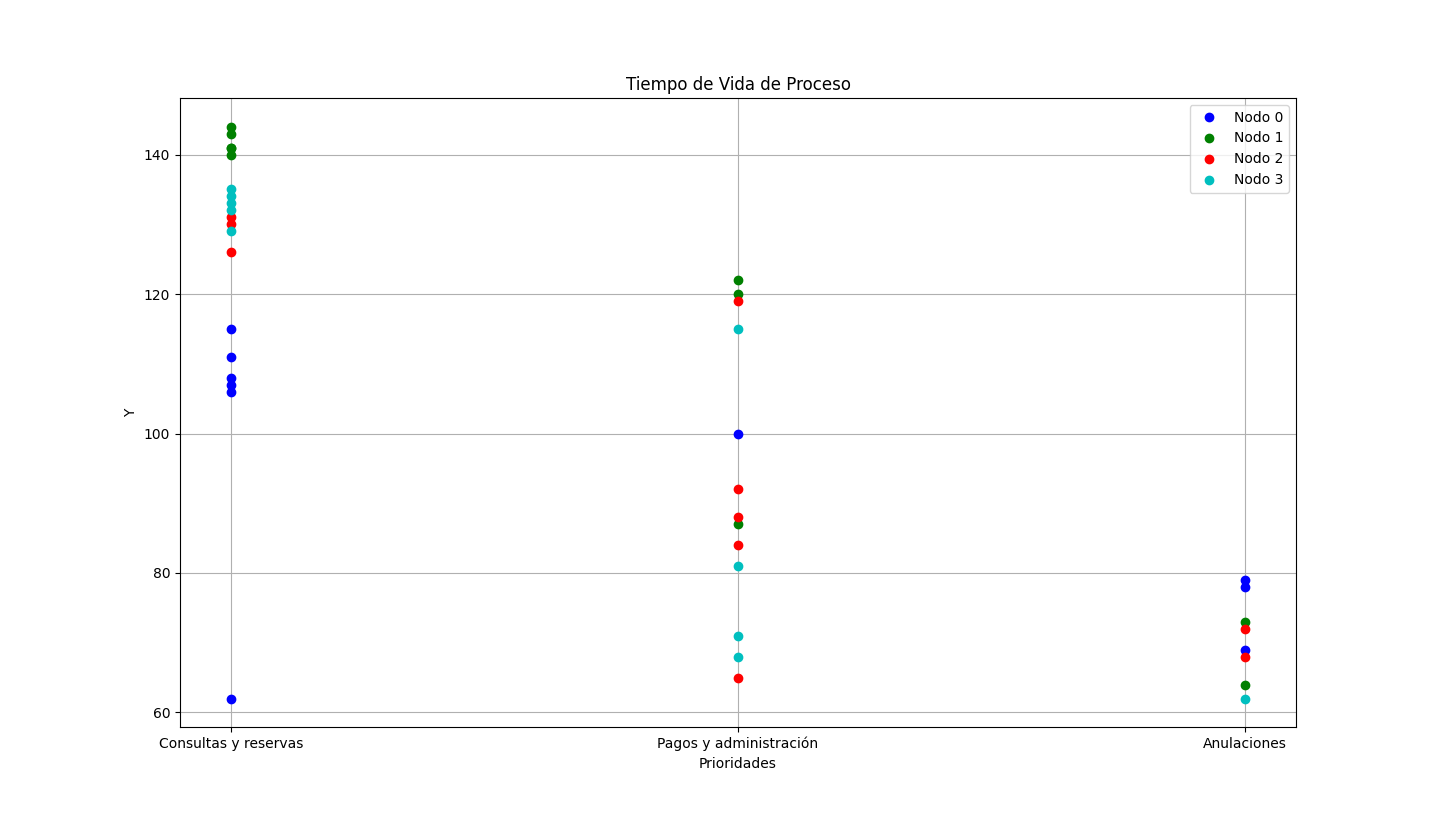
## 3.2. Tiempo que pasa un proceso hasta que entra en la SC.





## 3.3. Tiempo medio de vida de un proceso.





## 3.4. Número medio de mensajes por nodo.

