**Universidad de Costa Rica**

**Proyecto de simulación**

**Curso: CI-0130**

**Profesora: Ileana Alpizar**

**Estudiantes:**

**María Peraza Durán B85929**

**José Ignacio Víquez Rojas B78451**

**Fecha de entrega : Martes 10 de noviembre**

**Diagramas de los eventos:**

Se incluyen en un archivo SVG externo, con el nombre de “Diagrama-Eventos”.

**Fórmulas para la generación de números aleatorios:**

1. Distribución normal parámetros y – método directo.

Solución:

Para poder generar un valor usando el método directo, se debe de aplicar una de las siguientes formulas:

Como podemos ver aparecen dos valores r1 , r2 estos son dos valores generados de entre 0 y 1 con distribución uniforme de manera aleatoria. Es importante mencionar que se puede usar cualquiera de las dos fórmulas, en nuestro caso se decidió usar la de coseno. Esto genera un valor “z” que después se utiliza en la fórmula:

Este “x” es nuestro valor aleatorio.

1. Distribución normal parámetros y – método convolucional (TLC).

Solución:

Para poder generar un valor aleatorio para una distribución normal se hace de la siguiente manera:

Y si usamos el método de TLC, para generar un valor a “z”; a continuación, se muestra la fórmula para esto:

En el libro de texto del curso, se recomienda usar “n = 12” de esta manera la formula se simplifica a la siguiente:

En nuestra simulación se usará la “n = 12”, como se recomienda, para el cálculo de los valores de la “z” en la ecuación de “x”. Con esto ya se puede generar un valor aleatorio usando TLC.

1. Distribución uniforme en (a,b).

Solución:

Para generar un valor aleatorio con una distribución uniforme en el rango de [a,b] se usa la siguiente formula, en donde “x” representa el valor generado:

Como podemos apreciar se utiliza un valor “r” que se genera de manera aleatoria en el rango de [0, 1].

1. Distribución exponencial parámetro λ.

Solución:

Para generar un valor aleatorio con distribución exponencial con parámetro λ. Se debe de utilizar una de las siguientes formulas:

Como podemos apreciar se utiliza un valor “r” que se genera de manera aleatoria en el rango de [0, 1].

1. Distribución con función de densidad f(x) = kx a ≤ x ≤ b con k constante.

Solución:

Primero hemos de integrar la función f(x) en el intervalo de [a, x] para así obtener la función de distribución acumulada o F(x). Se escoge este intervalo ya que representa el área en la que el valor podría estar. Veamos:

Ahora ocupamos calcular el valor de la inversa de F(x). Esto se realiza igualando F(x) = r, en donde “r” es un valor tomado de ente [0, 1]. Veamos:

Con esta fórmula podemos calcular un valor aleatorio para “x”.

**Manual de instalación:**

Para poder compilar y ejecutar el programa se deben de ejecutar en consola los siguientes comandos desde la carpeta en donde se encuentran:

|  |
| --- |
| g++ main.cc Mensaje.cc Evento.cc -o Simulacion  ./Simulacion |

**Notas:**

Cuando se le solicita al usuario los valores para X1, X2, X3 estos deben ir en notación decimal. Ej: 0.3. Lo mismo aplica los valores que sean “fraccionarios”.

Cuando se le solicita valores al usuario estos deben de ser los que se especifican, no se hizo manejo de errores para ninguna parte, ya que esto vas más allá del propósito de este proyecto. Por ejemplo, si se solicita que se ingrese un valor del menú inicial y se ingresa un “hola” el programa no funcionara de la manera esperada, esto aplica para cualquier parte en la que se requiera interacción del usuario.

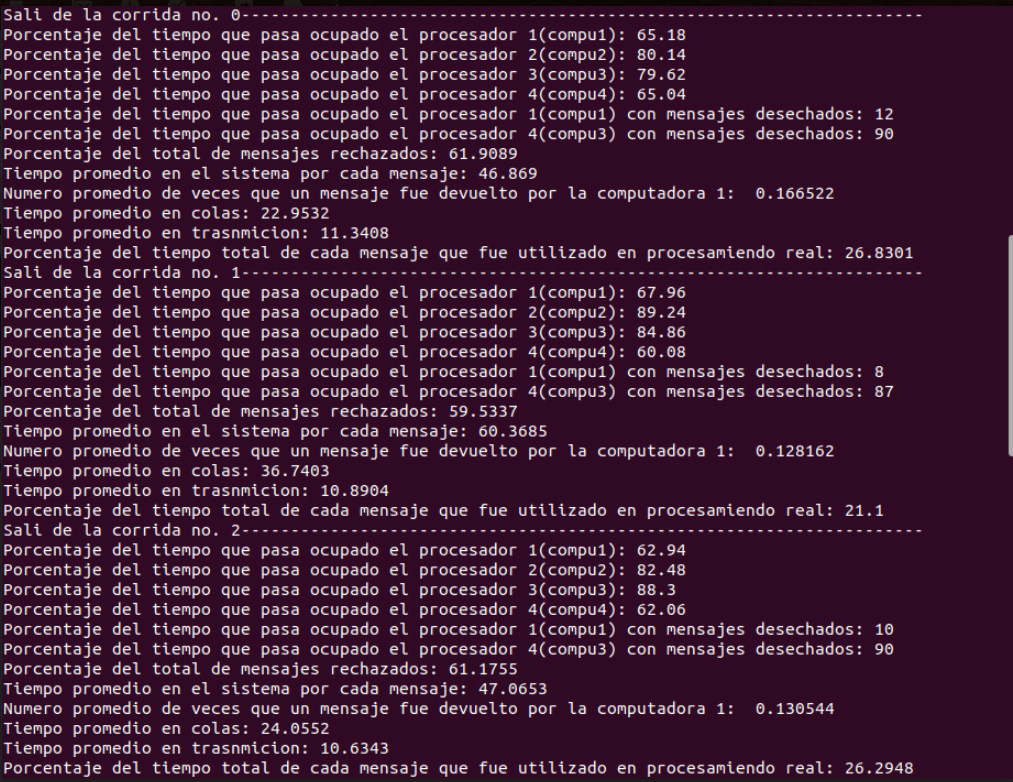
El dato del intervalo de confianza solo se despliega si se realizan 10 corridas.

El programa fue creado, compilado, ejecutado y probado en Ubuntu.

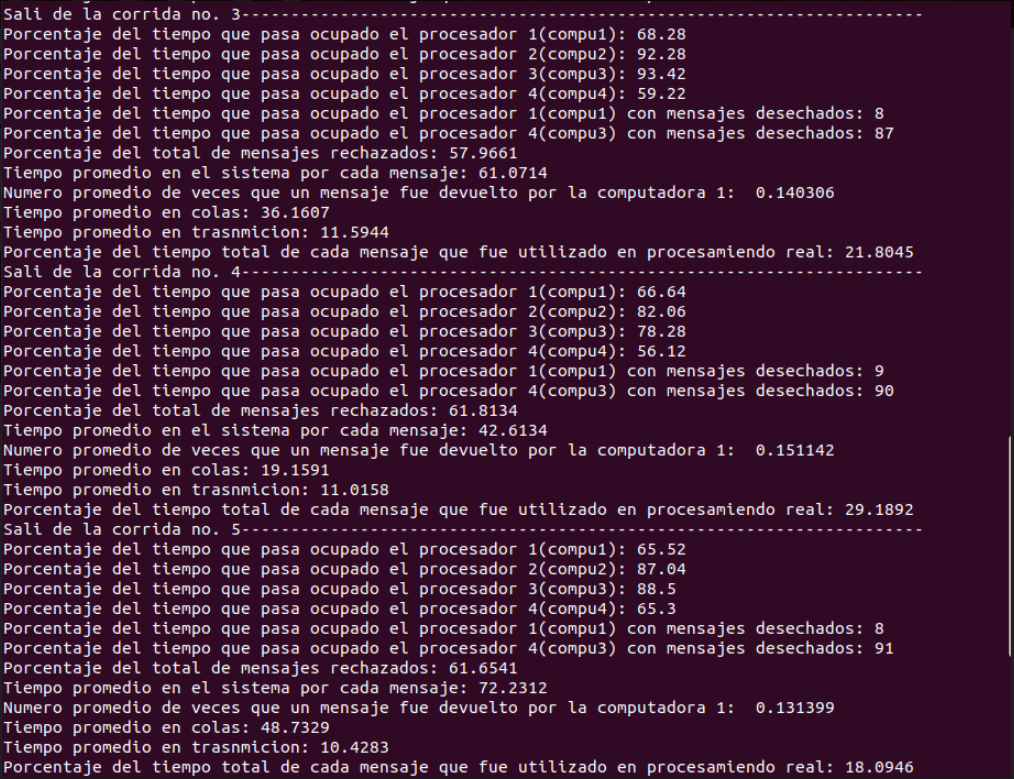
**Estadísticas obtenidas (10 corridas):**

Se adjuntan capturas de una simulación con 10 ejecuciones o “corridas”:

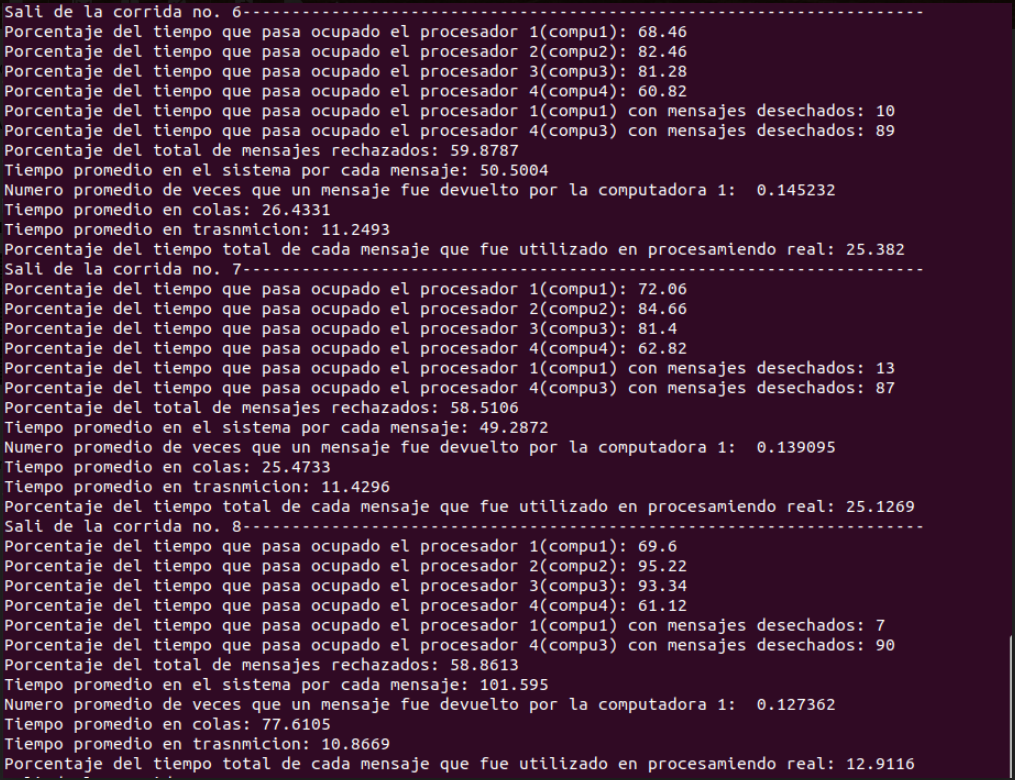
Captura de las simulaciones 0-2:



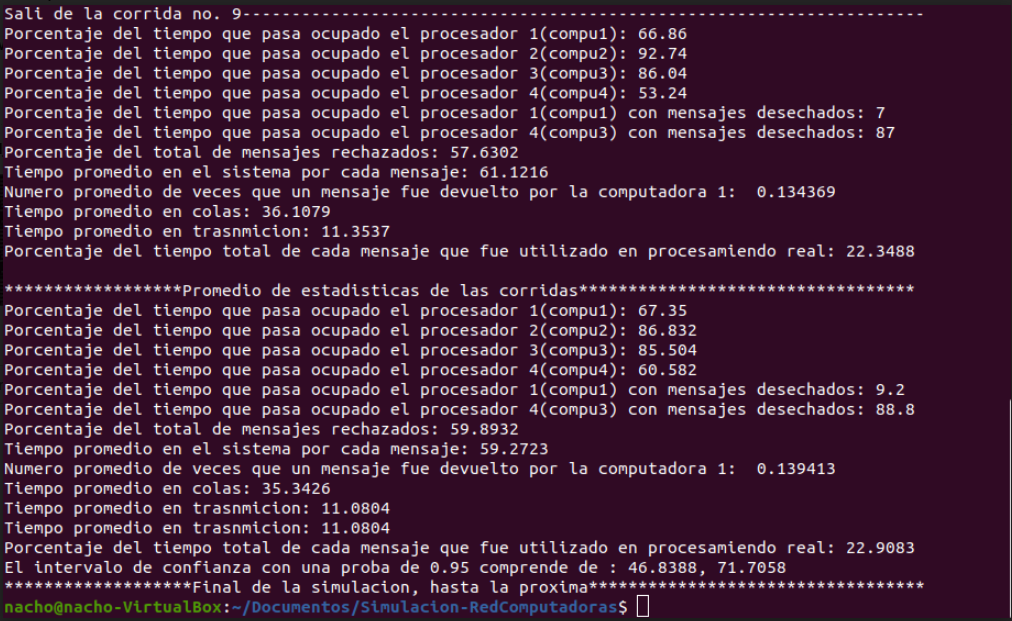
Captura de las simulaciones 3 – 5:



Captura de las simulaciones 6 – 8:

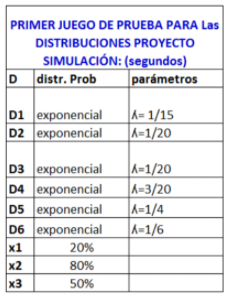


Captura de las simulaciones 8 – 9 y estadísticas finales:



**Análisis del sistema:**

Los resultados que se muestran en las corridas fueron los obtenidos con la siguiente información:



Se usaron los valores con dos decimales y redondeando.

Concluimos que:

* Los procesadores de la computadora 2 pasan en promedio más tiempo ocupados que los de las otras computadoras.
* La computadora 3 pasa una gran mayoría de tiempo de trabajo con mensajes que será desechados, eso ya que la probabilidad de rechazo es de 80% lo que hace que la mayoría de las veces se elimine el mensaje. Esto se evidencia también en el porcentaje de mensaje rechazados del sistema.
* La computadora 1, de los mensajes que procesa, la mayoría son enviados a su destino ya que se ve que el numero promedio de veces que fue devuelto por la computadora 1 es muy bajo. \*
* El sistema es ineficiente ya que si observamos el tiempo promedio que pasa un mensaje en el sistema (59.27) y el tiempo promedio que pasa un mensaje en cola (35.34) nos damos cuenta que en promedio un mensaje pasa más de la mitad del tiempo haciendo cola.
* Se puede notar que el porcentaje de tiempo total, refiriéndose a la suma de los tiempos de los mensajes que llegaron a su destino y que se desecharon, que un mensaje fue utilizado en tiempo real de procesamiento es de apenas un 22.90%. Esto indica que hubo aproximadamente un 78 % del tiempo que estuvo en el sistema, un mensaje estuvo) haciendo otras tareas. Apoyando a lo que se concluyó en el punto anterior.

**Problemas no resueltos:**

Todo se completó.

**Notas de los integrantes:**

|  |  |  |
| --- | --- | --- |
| Nombre | María Peraza Durán b85929 | José ignacio Víquez Rojas b78451 |
| Nota | **100** | **100** |