Proyecto de Simulación

# Manual de Usuario

## Iniciar la aplicación

La simulación fue implementada utilizando HTML y Javascript. Para abrir el sitio web es necesario utilizar un servidor web, de lo contrario abrir directamente el HTML con un navegador puede dar errores de presentación.

Si se tiene instalado Python 3 se puede crear fácilmente un servidor usando los siguientes comandos desde una terminal de Linux:

cd directorioProyecto

python3 –m http.server

o bien en una terminal de Windows (como PowerShell):

cd directorioProyecto

python.exe –m http.server

Luego se puede abrir un navegador con la dirección que indicó el script anterior, normalmente <http://localhost:8000/>.

Además el proyecto se encuentra disponible en las siguientes direcciones:

* <http://proyectoio2013.estebarb.tk>
* <http://pacific-lowlands-9599.herokuapp.com>

El programa descarga Angular.js, jQuery y Twitter Bootstrap, por lo que requiere conexión a internet. Si no se utiliza un servidor para proveer el HTML al navegador entonces se requiere utilizar Firefox, ya ni Chrome ni Internet Explorer permiten que los scripts accedan a archivos en file://.

El funcionamiento del programa (utilizando las URLs anteriores) ha sido comprobado los siguientes navegadores:

1. Google Chrome (Ubuntu 13.04 y Windows 8)
2. Mozilla Firefox (Ubuntu 13.04 y Windows 8)
3. Internet Explorer 10 (Windows 8)

El código fuente también está disponible en <https://github.com/estebarb/pio13>.

## Uso de la Simulación

El programa permite configurar la cantidad de veces que se ejecutará la simulación, la cantidad de minutos a simular y la pausa (en milisegundos) entre cada paso de tiempo. También permite especificar si se desea que se actualice la GUI en tiempo real o bien si se actualiza hasta el final (esto a veces es útil por motivos de rendimiento).

Para iniciar la simulación hay dos opciones:

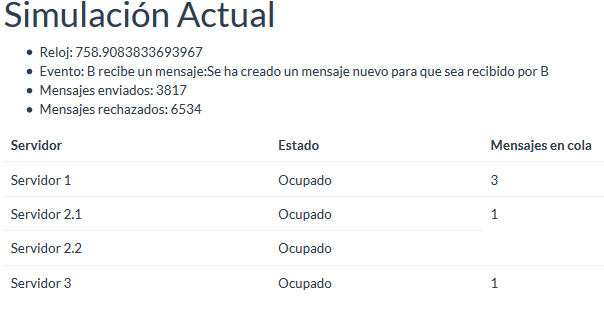
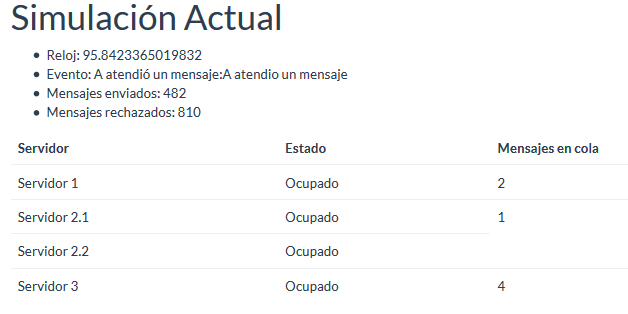
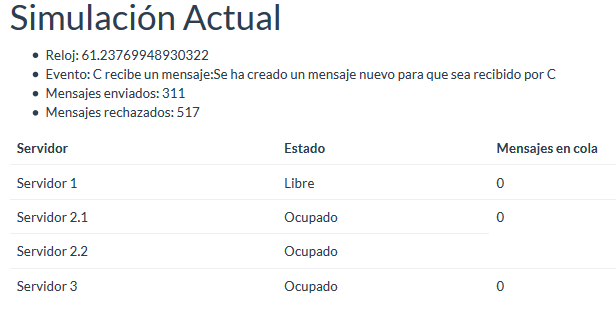
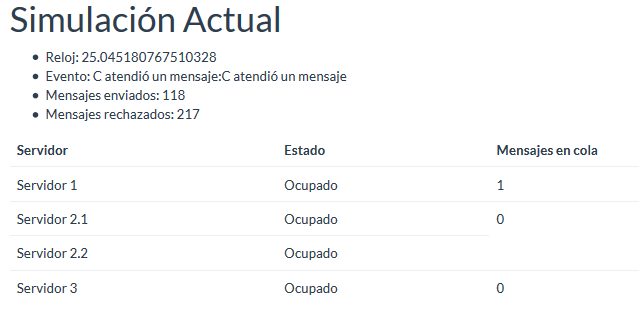
* **Iniciar Simulación**: En este modo la simulación será ejecutaba paso a paso, con una pausa entre cada paso de la simulación. El estado de la simulación y las estadísticas acumuladas serán actualizadas instantáneamente en la pantalla.
* **Ejecución Rápida**: En este modo el sistema no hará pausas entre cada paso de la simulación, por lo que es el modo ideal para correr simulaciones que en el otro modo tardarían demasiado. Sin embargo, las estadísticas y los datos de la simulación actual solo se actualizan al final de cada simulación.

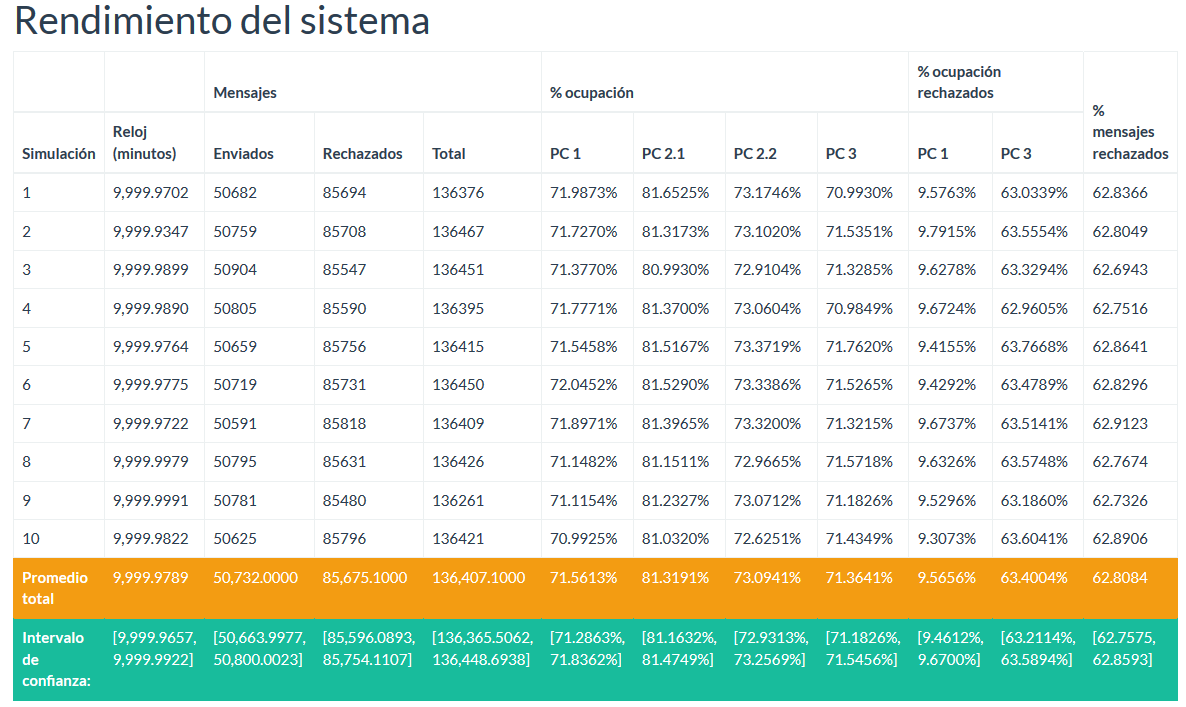
Para borrar los datos acumulados o detener una simulación se puede presionar el botón **Reiniciar**.

Una vez iniciada la simulación se pueden ver los datos de la simulación actual:

* El reloj actual
* Nombre y descripción del evento que se está procesando
* Cantidad de mensajes enviados
* Cantidad de mensajes rechazados
* Estado de ocupación de cada servidor y cantidad de mensajes en cola

Estos datos son actualizados en de forma instantánea según se va dando la simulación (pero no en el modo de ejecución rápida).



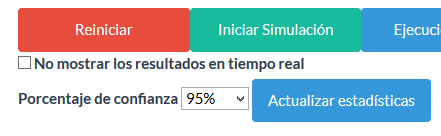
Además, de forma simultánea a la simulación también se actualizan las estadísticas acumuladas del sistema. Estas estadísticas incluyen:

* Reloj del último evento simulado (en minutos)
* Cantidad de mensajes enviados, rechazados y procesados en total
* Porcentaje de ocupación en cada CPU
* Porcentaje de ocupación con mensajes que al final son rechazados
* Porcentaje de mensajes rechazados

Además, se incluye las siguientes estadísticas, para los mensajes rechazados, aceptados y ambos:

* Promedio de devoluciones desde la computadora 1 hasta la 2 o la 3.
* Tiempo promedio por mensaje en el sistema, en colas o en transmisión
* El porcentaje de tiempo en procesamiento por mensaje (no en colas o en transmisión).



En todas las tablas es posible modificar el porcentaje de confianza con que se desea calcular el intervalo de confianza. Basta con seleccionarlo de la lista y hacer clic en “Actualizar Estadísticas”.

# Descripción del sistema a simular

El enunciado del proyecto, incluidas las modificaciones, es el siguiente:

Suponga que hay una red de 3 computadoras cuyo único trabajo es el de encargarse de la recepción, distribución y envío de los mensajes electrónicos en determinada empresa. Esta red funciona así:

* La computadora No.1 es la única encargada de enviar los mensajes a su destino, pero ésta no los recibe directamente de los usuarios sino desde las otras 2 computadoras.
* La computadora No.2, que tiene 2 procesadores, recibe mensajes con una distribución para el tiempo entre arribos normal con una media de 15 segundos y una varianza de 1 segundo cuadrado. Esta computadora analiza y prepara cada uno de estos mensajes con alguno de sus dos procesadores (cada procesador tarda en promedio un tiempo con distribución uniforme entre 12 y 25 segundos, una vez analizado y preparado un mensaje, éste lo envía siempre a la computadora No.1. La computadora No. 1 usualmente le devuelve a esta computadora (la Núm 2) el 20% de los mensajes que recibe de ella para que los analice y prepare de nuevo pues llegaron con algún error. (Esto ocurre igual con los mensajes que llegan por primera vez a la computadora 1 desde la 2 o con los que llegan por n-ésima vez).
* La computadora No. 3 recibe mensajes con distribución de tiempo entre arribos con densidad igual a f(x) = x/24 con x entre 4 y 8 segundos. Esta máquina solo cuenta con 1 procesador y tiene el mismo trabajo que se describe para la computadora 2: analiza y prepara cada mensaje para ser enviado a la computadora 1, sin embargo se da el caso especial acá de que en promedio, el 80% de todos los mensajes que aquí llegan son rechazados totalmente como resultado del análisis por no cumplir con alguna especificación obligatoria. Los que no son rechazados son enviados a la computadora No.1. La computadora No. 1 usualmente le devuelve a esta computadora el 50% de todos los mensajes que recibe de ella (igual probabilidad para los mensajes nuevos y para lo que ya habían sido devueltos 1 o más veces). Estos mensajes devueltos se vuelven a analizar y preparar en la computadora No. 3, y corren igual suerte como si fueran nuevos: puede ser rechazado con la misma probabilidad. El procesador de esta máquina puede analizar y preparar (o rechazar) un mensaje en un promedio de 4 segundos, tiempo exponencial.
* La computadora No. 1, con un procesador, puede procesar un promedio de 10 mensajes por minuto (t. exponencial), ya sea enviándolos a su destino, o devolviéndolos a su computadora de origen por llegar con error. Cuando se devuelve un mensaje, ya sea a la computadora 2 o a la 3 el tiempo que tarda en llegar tiene distribución constante de 3 segundos.
* El tiempo que tarda en llegar un mensaje a la computadora 1 cuando es enviado desde la computadora 2 o desde la 3 siempre son 20 segundos (Suponga que no se usan las mismas líneas de transmisión y que las colisiones de arribos no son problema).
* En cualquier computadora el tiempo que se tarda en colocar un mensaje que llega en la cola, o en asignárselo a un procesador si éste está libre es 0 para efectos prácticos.

# Diseño del Simulador

El simulador representa la simulación por medio de eventos que modifican el estado del sistema. El siguiente diagrama representa los tipos de objetos que representan la simulación:



Los conceptos fundamentales del simulador son los siguientes:

1. El estado del sistema está totalmente contenido en un objeto Estado.
2. Los eventos son representados por un objeto Evento que contiene los siguientes datos:
   1. Información sobre la instancia actual del evento (como la hora de ocurrencia o el mensaje asociado al evento).
   2. Una función lambda que transforma un Estado en el siguiente estado. Debido a que es una función pura (en el ámbito del simulador) se facilita enormemente ejecutar la simulación, pues cada evento puede ejecutar cualquier tipo de lógica, sin afectar la lógica del simulador.
3. Los eventos a ejecutar están en una lista, ordenada por el tiempo de ocurrencia del evento. Esto ofrece las siguientes ventajas:
   1. Como la lista está ordenada el próximo evento siempre será la cabeza de la lista.
   2. Programar nuevos eventos es trivial: simplemente se insertan a la lista de eventos. Como la lista se puede expandir para contener la cantidad de eventos que sea nunca van a suceder problemas como que no se pueda programar un evento porque hay otro del mismo tipo programado (o que le “caiga encima”).
   3. Los eventos pueden programar nuevos eventos de una forma funcionalmente pura. Nunca modifican el estado del simulador. En particular, nunca hay mutación del estado de eventos previamente programados.
4. El simulador básicamente ejecuta pasos de la simulación hasta que la simulación se termine ().

# Estadísticas y análisis del sistema

A continuación se muestran los resultados de ejecutar la simulación diez veces durante 1000 minutos, usando un porcentaje de confianza del 95%.

## Rendimiento del sistema

| **Simulación** | **Reloj (minutos)** | **Mensajes** | | | **% ocupación** | | | | **% ocupación rechazados** | | **% mensajes rechazados** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Enviados** | **Rechazados** | **Total** | **PC 1** | **PC 2.1** | **PC 2.2** | **PC 3** | **PC 1** | **PC 3** |
| **1** | 999.9703 | 5038 | 8623 | 13661 | 70.9801% | 81.5587% | 73.1686% | 71.6409% | 9.4162% | 63.8840% | 63.1213 |
| **2** | 999.9898 | 5031 | 8591 | 13622 | 73.1505% | 81.3144% | 73.6530% | 71.6734% | 9.8018% | 63.7108% | 63.0671 |
| **3** | 999.9875 | 5056 | 8596 | 13652 | 68.5993% | 81.2862% | 73.2299% | 70.9265% | 8.4329% | 63.1635% | 62.9651 |
| **4** | 999.9593 | 5101 | 8527 | 13628 | 71.9728% | 81.5598% | 73.4986% | 71.5196% | 9.5489% | 63.7778% | 62.5697 |
| **5** | 999.9950 | 5047 | 8588 | 13635 | 71.7094% | 81.5126% | 73.0429% | 71.8082% | 10.1713% | 64.0695% | 62.9850 |
| **6** | 999.9559 | 4992 | 8655 | 13647 | 71.6318% | 81.9290% | 73.9681% | 70.9380% | 9.6381% | 63.8078% | 63.4205 |
| **7** | 999.9733 | 5016 | 8656 | 13672 | 70.1209% | 81.3812% | 72.4284% | 72.7521% | 9.8455% | 65.3881% | 63.3119 |
| **8** | 999.9964 | 5044 | 8615 | 13659 | 71.2851% | 80.4276% | 72.9695% | 72.8444% | 9.3089% | 64.7451% | 63.0720 |
| **9** | 999.9969 | 5038 | 8612 | 13650 | 70.6901% | 82.0284% | 73.8787% | 71.2798% | 9.3791% | 63.3384% | 63.0916 |
| **10** | 999.9701 | 5058 | 8589 | 13647 | 72.2071% | 80.7476% | 73.0652% | 70.5710% | 9.7471% | 62.5475% | 62.9369 |
|  | 999.9794 | 5,042.1000 | 8,605.2000 | 13,647.3000 | 71.2347% | 81.3746% | 73.2903% | 71.5954% | 9.5290% | 63.8433% | 63.0541 |
| **Intervalo de Confianza** | [999.9683, 999.9906] | [5,021.7407, 5,062.4593] | [8,578.5638, 8,631.8362] | [13,636.3006, 13,658.2994] | [70.3396%, 72.1298%] | [81.0271%, 81.7220%] | [72.9568%, 73.6238%] | [71.0630%, 72.1278%] | [9.1976%, 9.8604%] | [63.2754%, 64.4111%] | [62.8910, 63.2173] |

## Mensajes rechazados

| **Simulación** | **Media rebotes en PC1** | **T. promedio por mensaje en** | | | **% tiempo cómputo** |
| --- | --- | --- | --- | --- | --- |
| **Sistema** | **Colas** | **Trans.** |
| **1** | 0.10843 | 13.5158s | 5.9217s | 2.4939s | 63.9107% |
| **2** | 0.11675 | 14.3981s | 6.5787s | 2.6853s | 62.7924% |
| **3** | 0.09981 | 13.3081s | 6.0150s | 2.2957s | 64.5208% |
| **4** | 0.10930 | 14.0094s | 6.3361s | 2.5139s | 63.5875% |
| **5** | 0.11667 | 14.2520s | 6.3816s | 2.6835s | 62.7714% |
| **6** | 0.11473 | 14.0283s | 6.2981s | 2.6388s | 63.8058% |
| **7** | 0.11229 | 14.4022s | 6.6047s | 2.5827s | 61.9804% |
| **8** | 0.11120 | 14.1960s | 6.4808s | 2.5576s | 62.1034% |
| **9** | 0.10996 | 13.7611s | 6.1657s | 2.5291s | 63.5780% |
| **10** | 0.11259 | 13.8883s | 6.2487s | 2.5895s | 64.1780% |
| **Promedio total** | **0.11117** | **13.9759s** | **6.3031s** | **2.5570s** | **63.3228%** |
| **Intervalo de confianza** | **[0.10765, 0.11470]** | **[13.7147s, 14.2371s]** | **[6.1421s, 6.4641s]** | **[2.4759s, 2.6381s]** | **[62.7031%, 63.9426%]** |

## Mensajes enviados

| **Simulación** | **Media rebotes en PC1** | **T. promedio por mensaje en** | | | **% tiempo cómputo** |
| --- | --- | --- | --- | --- | --- |
| **Sistema** | **Colas** | **Trans.** |
| **1** | 0.22787 | 66.3629s | 14.4468s | 25.2410s | 40.5315% |
| **2** | 0.22739 | 68.0751s | 15.8714s | 25.2300s | 40.3132% |
| **3** | 0.21479 | 64.5414s | 13.2127s | 24.9403s | 41.0811% |
| **4** | 0.21819 | 66.2477s | 14.7602s | 25.0184s | 40.3519% |
| **5** | 0.21914 | 66.4889s | 14.8530s | 25.0402s | 40.2901% |
| **6** | 0.23137 | 68.3179s | 15.9733s | 25.3215s | 40.1874% |
| **7** | 0.21910 | 64.7784s | 13.2666s | 25.0393s | 41.0255% |
| **8** | 0.21372 | 65.2064s | 13.7564s | 24.9155s | 40.9286% |
| **9** | 0.23065 | 66.3063s | 14.1952s | 25.3049s | 40.5080% |
| **10** | 0.21945 | 67.3621s | 15.7197s | 25.0474s | 40.1095% |
| **Promedio total** | 0.22217 | 66.3687s | 14.6055s | 25.1099s | 40.5327% |
| **Intervalo de confianza:** | [0.21750, 0.22684] | [65.4468s, 67.2906s] | [13.8727s, 15.3384s] | [25.0024s, 25.2173s] | [40.2782%, 40.7872%] |

## Todos los mensajes

| **Simulación** | **Media rebotes en PC1** | **T. promedio por mensaje en** | | | **% tiempo cómputo** |
| --- | --- | --- | --- | --- | --- |
| **Sistema** | **Colas** | **Trans.** |
| **1** | 0.15248 | 33.0051s | 9.0657s | 10.8827s | 55.2888% |
| **2** | 0.15761 | 34.2226s | 10.0108s | 11.0117s | 54.4902% |
| **3** | 0.14240 | 32.2823s | 8.6806s | 10.6821s | 55.8399% |
| **4** | 0.15006 | 33.5624s | 9.4893s | 10.9374s | 54.8904% |
| **5** | 0.15460 | 33.5875s | 9.5173s | 10.9589s | 54.4499% |
| **6** | 0.15740 | 33.8871s | 9.8372s | 10.9360s | 55.1663% |
| **7** | 0.15148 | 32.8843s | 9.0488s | 10.8216s | 54.2924% |
| **8** | 0.14906 | 33.0331s | 9.1676s | 10.8140s | 54.2840% |
| **9** | 0.15451 | 33.1547s | 9.1293s | 10.9353s | 55.0632% |
| **10** | 0.15219 | 33.7073s | 9.7589s | 10.9131s | 55.2575% |
| **Promedio total** | 0.15218 | 33.3326s | 9.3706s | 10.8893s | 54.9023% |
| **Intervalo de confianza:** | [0.14899, 0.15536] | [32.9273s, 33.7380s] | [9.0703s, 9.6708s] | [10.8217s, 10.9568s] | [54.5345%, 55.2700%] |

## Análisis del sistema