

Generador de Conjuntos de Tareas para Simulación en Sistemas de Tiempo Real

Gabriela Olguín, Laura Biscayart, José M. Urriza

Universidad Nacional de la Patagonia San Juan Bosco, Puerto Madryn, Argentina.
golguin@unpata.edu.ar, lalibiscayart@gmail.com,
josemurriza@unp.edu.ar

Resumen. En la disciplina Sistemas de Tiempo Real, una forma de comprobar las nuevas teorías o modelos es realizar simulaciones. El objetivo de estas simulaciones es validar los resultados que se obtendrían de estos modelos o teorías en la práctica. Para esto, es necesario generar miles o millones de sistemas de tiempo real sintéticos automáticamente. Estos sistemas deben ser generados bajo ciertas premisas y formatos para que el simulador pueda utilizarlos y al mismo tiempo validar los resultados. En este trabajo, se presenta un software para generar conjuntos de sistemas de tiempo real sintéticos. El software se encuentra escrito en el lenguaje Ada 2005 y es de código abierto.

Palabras Clave: Generación de tareas, Simulación de Sistemas, Tiempo Real.

1 Introducción

La simulación en la disciplina Sistemas de Tiempo Real permite comprobar la eficacia de los algoritmos surgidos de nuevas teorías o modelos. Sean algoritmos para determinar la planificabilidad de sistemas, para planificación de tareas heterogéneas, ahorro de energía o tolerancia a las fallas, la simulación permite comparar sus rendimientos entre sí y, sobretodo, evaluarlos en ambientes o plataformas con diferentes características.

Para validar estos algoritmos, se realizan pruebas sobre miles o millones de sistemas sintéticos. Es necesaria una herramienta que permita generar automáticamente estos sistemas y para ello se requiere la construcción de una aplicación específica.

Por otro lado, en grupos de desarrollo donde se investigan varias sub-disciplinas, contar con una aplicación que permita generar conjuntos de sistemas sintéticos, y brindar la posibilidad de que los mismos sean configurables, simplifica la generación de datos de entrada para cada una de las simulaciones, las cuales suelen tener requerimientos diferentes. Por esto, la flexibilidad de la aplicación es una de las principales características a tener en cuenta.

Después de una búsqueda exhaustiva, no fue posible encontrar aplicaciones para generación de conjuntos de sistemas de tiempo real de código abierto. Es por ello, que teniendo como principal objetivo la flexibilización, en vez de seguir creando

aplicaciones que generen conjuntos de tareas *ad hoc* adaptadas para cada tipo de simulación, se optó por construir un software de generación adaptable.

El software es configurable, de uso libre y de código abierto, programado en Ada 2005 [1] es sencillo de utilizar y genera millones de sistemas en un tiempo razonablemente pequeño. Para facilitar la integración con cualquier simulador, la salida puede guardarse en archivos con formato *.xml* o en archivos de texto plano *.txt*. El software se puede encontrar en <http://www.rtsg.unp.edu.ar/>.

En las siguientes secciones se presentará cómo el software fue desarrollado y qué posibles conjuntos pueden ser generados con el mismo, a partir de la configuración del archivo *setup.txt*. En la sección 2 se presenta el Modelo utilizado para la generación, donde se describe cómo está compuesto cada sistema, cómo se obtienen cada uno de sus componentes, el cálculo del hiperperíodo, la verificación del factor de utilización obtenido y el cálculo de las diferencias entre los factores de utilización de cada sistema. En la sección 3 se describe el generador implementado, explicando sus entradas y salidas. En la sección 4 se muestra en gráficos los resultados obtenidos de varias ejecuciones del generador para evaluar la disparidad de los factores de utilización generados. Por último en la sección 5 se presentan las conclusiones y se mencionan posibles trabajos futuros para mejorar el software desarrollado.

2 Modelo Utilizado para la Generación

A continuación se detalla el modelo de sistemas utilizado. El mismo es similar al presentado en los trabajos de [2, 3]. En la Tabla 1 se define la terminología utilizada para la definición del sistema.

Tabla 1. Terminología

<i>Tiempo de ejecución (C_i)</i>	Máximo tiempo que podría tomar la ejecución de una tarea. Normalmente se define como el <i>peor caso de tiempo de ejecución (WCET)</i> .
<i>Periodo (T_i)</i>	Intervalo de tiempo entre arribos de una tarea periódica.
<i>Vencimiento (D_i)</i>	Instante de tiempo máximo, relativo a la instanciación de la tarea, en que la misma debe completar su ejecución.
<i>Tiempo de bloqueo (B_i)</i>	Máximo tiempo en que una tarea podría esperar por tareas de menor prioridad con las que comparte recursos.
<i>Retardo de instanciación (Jitter) (J_i)</i>	Diferencia entre el tiempo de arribo de una tarea y el tiempo en que está lista para su ejecución.
<i>Offset (Of_i)</i>	Máximo desplazamiento del tiempo de arribo de una tarea respecto del instante crítico.
<i>Tiempo opcional de ejecución (Co_i)</i>	Máximo tiempo que podría ejecutarse una instancia opcional de una tarea luego de completar su tiempo de ejecución.
<i>Hiperperíodo</i>	Es el mínimo común múltiplo de los periodos de las tareas.

Cada sistema S generado contiene n tareas periódicas, con tiempo de ejecución (C_i), período (T_i), vencimiento (D_i), tiempo de bloqueo (B_i), retardo de instanciación (J_i), tiempo opcional de ejecución (Co_i) y offset (Of_i) dentro de un rango configurable. En la Tabla 2 se listan las variables utilizadas.

Tabla 2. Notación

$S(n)$	Sistema de n tareas
U	Factor de utilización de cada sistema (en porcentaje)
n	Cantidad de tareas en cada sistema
i	Una tarea ($i = 1, \dots, n$)
C_i	Peor tiempo de ejecución de la tarea i
T_i	Período de la tarea i
T_{\min}	Valor mínimo para generar el T_i según el rango asignado
T_{\max}	Valor máximo para generar el T_i según el rango asignado
D_i	Vencimiento de la tarea i
D_{rel}	Relación de D_i respecto de T_i (\leq , \geq , $=$, \Leftrightarrow)
D_{\min}	Valor mínimo del porcentaje de variación de D_i respecto de T_i
D_{\max}	Valor máximo del porcentaje de variación de D_i respecto de T_i
B_i	Máximo tiempo de bloqueo de la tarea i
B_{\min}	Valor mínimo del porcentaje de variación de B_i respecto de C_i
B_{\max}	Valor máximo del porcentaje de variación de B_i respecto de C_i
J_i	Retardo de instanciación de la tarea i
J_{\min}	Valor mínimo del porcentaje de variación de J_i respecto de un valor en el orden de magnitud de T_i
J_{\max}	Valor máximo del porcentaje de variación de J_i respecto de un valor en el orden de magnitud de T_i
Of_i	Máximo tiempo de offset de la tarea i respecto del instante crítico
Of_{\min}	Valor mínimo del porcentaje de variación de Of_i respecto de T_i
Of_{\max}	Valor máximo del porcentaje de variación de Of_i respecto de T_i
Co_i	Tiempo opcional de ejecución para la tarea i
M	Orden de magnitud del período de una tarea
R	Cantidad de rangos para valores de los T_i
$R_{(i \bmod R)\min}$	Valor mínimo del rango al que pertenecerá el T_i de la tarea i
$R_{(i \bmod R)\max}$	Valor máximo del rango al que pertenecerá el T_i de la tarea i
U_i	Factor de utilización de la tarea i
U_{final}	Factor de utilización del sistema generado
<i>hiperperíodo</i>	El máximo común divisor entre los T_i del sistema generado

<i>DU</i>	Función de distribución de probabilidad uniforme
<i>random</i>	Función para generar un valor aleatorio con una <i>DU</i> entre 0 y 1
<i>random(min,max)</i>	Función para generar un valor aleatorio con una <i>DU</i> entre los valores min y max
<i>mcd</i>	Función para calcular el máximo común divisor entre dos números naturales
<i>mcm</i>	Función para calcular el mínimo común múltiplo entre dos números naturales
<i>mod</i>	Función para calcular el resto de la división entre dos números naturales
α, β	VARIABLES AUXILIARES
λ	Diferencia entre los factores de utilización de las tareas de un sistema (llamado <i>U-difference</i> en [4])
ε	Error porcentual admitido para cada <i>U</i>

En cada una de las sub secciones siguientes se expone la forma en que se obtiene cada uno de los componentes de las tareas. Luego se agregan otras funciones de validación y observación de los sistemas obtenidos.

2.1 Generación de C_i, T_i, U_i

Para comenzar con cada conjunto de tareas, se generan valores para todos los T_i y U_i , dentro de los límites establecidos en la configuración. A partir de ellos se calculan los C_i .

Los T_i pueden pertenecer a diferentes rangos (R), para casos en que se necesita simular tareas con períodos en rangos muy diferentes, por ejemplo algunas tareas con período entre 10 y 10^2 , y otras entre 10^3 y 10^4 . La cantidad de rangos, y el valor mínimo y el máximo para cada uno, se especifican en el archivo de configuración. La cantidad de tareas en cada rango será n/R , quedando mayor cantidad de tareas en los rangos inferiores si n no fuera exactamente divisible por R .

Los valores para T_i se generan aleatoriamente con una distribución de probabilidad uniforme (*DU*) dentro de cada rango. Enumerando los rangos de 0 a $R-1$, para generar cada T_i se sigue el algoritmo:

```
Tmin = R(i mod R)min
Tmax = R(i mod R)max
Ti = random( Tmin, Tmax )
```

El factor de utilización de cada tarea, U_i , es presentado en la siguiente ecuación:

$$U_i = \frac{C_i}{T_i}$$

El factor de utilización de cada sistema, U , es:

$$U = \sum_{i=1}^n U_i = \sum_{i=1}^n \frac{C_i}{T_i}$$

Para generar los U_i se utiliza el algoritmo *UUniFast* propuesto por Bini en [5]. Con esto se obtienen factores de utilización para cada tarea distribuidos uniformemente. El algoritmo que se presenta fue extraído del trabajo presentado en [6]. En pseudocódigo es:

```

Utotal = U
for i = 1 to n - 1
    sgteU = Utotal * random(1/(n-i))
    Ui = Utotal - sgteU
    Utotal = sgteU
end for
Un = Utotal

```

Una vez generados U_i y T_i , cada C_i se calcula como:

$$C_i = U_i \cdot T_i$$

2.2 Generación de los D_i

El vencimiento de cada tarea, D_i , se genera como un porcentaje de T_i . Son configurables los porcentajes mínimo y máximo, y la relación entre D_i y T_i (si es igual, menor o igual, mayor o igual, o arbitrario). En los modelos utilizados en [2] D_i se considera igual a T_i , pero en otros contextos, como en [7], esta relación podría ser arbitraria. Los cálculos se realizan siguiendo el algoritmo:

```

α = random( Dmin, Dmax )
case Drel
    si "<=>" => seleccionar aleatoriamente con una DU una de
    las otras opciones
    si "=" => Di = Ti
    si "<=" => Di = Ti - α * Ti
    si ">=" => Di = Ti + α * Ti

```

2.3 Generación de los B_i

El peor tiempo de bloqueo para cada tarea, B_i , se genera como un porcentaje de C_i . Puede configurarse el valor mínimo y el máximo de ese porcentaje, aunque se espera que el máximo no supere el 30 por ciento. Se genera un valor α con una DU entre el mínimo y el máximo configurados, y luego se calcula B_i como:

$$\alpha = \text{random}(B_{\min}, B_{\max}) \Rightarrow B_i = \alpha \cdot C_i$$

2.4 Generación de los J_i

El tiempo de retardo de instanciación (jitter), J_i , es el tiempo en el cual es sistema no realiza ninguna clase de tarea esperando que la tarea i se encuentre disponible o se instancie. Se calcula como un porcentaje de algún valor del orden de magnitud de T_i (M). En este caso también se puede configurar el porcentaje mínimo y el máximo. Se genera un valor α con una DU entre el mínimo y el máximo porcentaje configurado, luego se genera un valor β con una DU en el orden de magnitud de T_i (M) y a partir de ellos se calcula J_i . Todo esto se puede resumir en la siguiente línea:

$$M = \log_{10}(T_i), \alpha = \text{random}(J_{\min}, J_{\max}), \beta = \text{random}(10^M, 10^{M+1} - 1) \Rightarrow J_i = \alpha \cdot \beta$$

2.5 Generación del Of_i

El tiempo de offset (Of_i), al igual que el tiempo de ejecución opcional (Co_i), es un parámetro de generación optativa. Se utiliza en casos en que se desea simular el arribo de tareas después del instante crítico.

El Of_i se genera como un porcentaje de T_i . Primero se genera un valor α aleatoriamente con una DU entre el valor mínimo y el máximo configurados, luego se obtiene Of_i :

$$\alpha = \text{random}(Of_{\min}, Of_{\max}) \Rightarrow Of_i = \alpha \cdot T_i$$

2.6 Generación del Co_i

El tiempo de ejecución opcional (Co_i) es un valor de generación optativa que se utiliza en el estudio de modelos de computación imprecisa como los presentados en [8, 9]. Se genera aleatoriamente con una DU entre 0 y C_i :

$$Co_i = \text{random}(0, C_i)$$

2.7 Cálculo del hiperperíodo

Dada la periodicidad de las tareas, el estado de las tareas en cualquier instante de ejecución, se repite nuevamente transcurrido el hiperperíodo del sistema. Para calcular el hiperperíodo de cada sistema generado, se crea una lista de los T_i , ordenada de menor a mayor, y se calcula el mínimo común múltiplo (mcm) entre todos los valores, siguiendo el algoritmo:

```
mcm_parcial = T1
for i = T2 to Tn
    mcm_parcial = mcm( Ti, mcm_parcial )
end for
hiperperiodo = mcm_parcial
```

El mínimo común múltiplo (*mcm*) se calcula como:

$$mcm(a,b) = \frac{a \cdot b}{mcd(a,b)}$$

Donde $a > b$ y *mcd* es el máximo común divisor entre *a* y *b*, expresado como:

$$mcd(a,b) = a \bmod b \mid b \bmod (a \bmod b) = 0$$

2.8 Verificación del *U*

Al terminar de generar todos los valores, se verifica que el factor de utilización del sistema obtenido se acerque al objetivo, considerando un margen de error ε configurable. Para esto se vuelve a calcular como:

$$U_{final} = \sum_{i=1}^n \frac{C_i}{T_i}$$

Y el resultado obtenido se valida mediante:

$$U \cdot (100 - \varepsilon) \leq U_{final} \leq U \cdot (100 + \varepsilon)$$

2.9 Cálculo de la Disparidad de los *U_i*

Según lo presentado por Bini en [5], las diferencias entre los factores de utilización de las tareas de un sistema deben ser pequeñas. Esto garantiza una distribución pareja de la utilización del recurso entre las tareas. Cuanto mayor es la cantidad de tareas en el sistema, mejor distribuidos estarán los factores de utilización.

Para observar la distribución de los factores de utilización de los conjuntos generados puede calcularse la función de densidad de probabilidad de las diferencias entre el máximo y el mínimo *U_i* en cada conjunto. Sea

$$\lambda = \frac{\max_{i=1}^n U_i - \min_{j=1}^n U_j}{U} \quad \text{con } i \neq j \quad (1)$$

En la sección de resultados, se grafica la función de densidad de probabilidad de λ , de manera que pueda observar donde se centra la mayor parte de la generación de los sistemas.

3 Descripción del Generador

El generador comienza leyendo el archivo de configuración para tomar los valores de los parámetros a utilizar en la generación de tareas. Luego genera todos los valores de salida para un sistema, los guarda en el archivo de respaldo, verifica si el sistema generado responde a las características solicitadas, y en ese caso lo guarda en el

archivo principal de salida. Estos pasos se repiten hasta completar la cantidad de sistemas válidos solicitados.

Los parámetros se toman de un archivo de configuración con datos por defecto. Este archivo puede modificarse con un editor de textos para indicar las características generales de la salida, de cada conjunto de sistemas, de cada sistema y de las variables particulares de las tareas. Previo a cada ejecución del generador se debe indicar esos valores.

La salida se guarda en dos archivos: uno contiene todos los sistemas generados durante la ejecución del programa (el archivo de respaldo); el otro (el archivo principal de salida) contiene los sistemas cuyo factor de utilización coincide con el solicitado, considerando un margen de error porcentual configurable. El formato de ambos archivos puede ser *xml* o *txt*. También puede generarse un archivo con los λ obtenidos, pero se encuentran en el formato *txt*.

En las sub secciones siguientes se describen los archivos de configuración y de salida.

3.1 El Archivo de Configuración

El archivo de configuración permite determinar el comportamiento del generador. Contiene todos los valores que podrían modificarse, debidamente comentados para facilitar su seguimiento y adaptación. Describen las características generales de la salida, de los conjuntos, y las propiedades de cada variable. Ellos son:

- Parámetros generales
 - La cantidad máxima de sistemas a generar
 - El formato de salida (*xml* o *txt*)
 - El margen de error porcentual (ϵ)
 - Si se debe realizar o no el cálculo de la diferencia de los $U_i(\lambda)$ en cada sistema
 - Precisión de λ en el caso que se genere
- Parámetros que describen cada sistema
 - El factor de utilización deseado (U)
 - La cantidad de tareas (n)
 - La forma de ordenar las tareas en el sistema. Puede ser según los períodos (*Rate Monotonic*) o según los vencimientos (*Deadline Monotonic*)
- Parámetros que acotan los valores de los períodos (T_i)
 - La cantidad de rangos a considerar (R)
 - Para cada rango el valor mínimo y el máximo
- Parámetros que afectan los tiempos de ejecución (C_i)
 - Precisión de salida (cantidad de dígitos)
- Parámetros que describen las características de los vencimientos (D_i)
 - Su relación respecto de T_i (\leq , \geq , $=$, $\leq \Rightarrow$)

- El valor mínimo del porcentaje de variación respecto de T_i
- El valor máximo del porcentaje de variación respecto de T_i
- La precisión de salida (cantidad de dígitos)
- Parámetros que describen las características de los tiempos de bloqueo (B_i)
 - El valor mínimo del porcentaje respecto de T_i
 - El valor máximo del porcentaje respecto de T_i
 - La precisión de salida (cantidad de dígitos)
- Parámetros que describen las características de los retardos de instanciación (J_i)
 - El valor mínimo del porcentaje respecto de un valor del orden de T_i
 - El valor máximo del porcentaje respecto de un valor del orden de T_i
 - La precisión de salida (cantidad de dígitos)
- Parámetros que afectan la generación de tiempos de offset (Of_i)
 - Indicación de si se generan o no
 - El valor mínimo del porcentaje respecto de T_i
 - El valor máximo del porcentaje respecto de T_i
- Parámetros que afectan la generación de tiempos de ejecución opcional (Co_i)
 - Si se generan o no
 - La precisión de salida

Para facilitar la lectura y mantenimiento del archivo de configuración se permite el uso de comentarios de línea, indicados con un carácter # al comienzo de la línea, y se ignoran los espacios y líneas en blanco.

Cada parámetro tiene un nombre fijo, seguido de un signo = y su valor en números naturales. Se define un parámetro por línea y se ignora todo lo que se escriba a continuación del valor. Puede modificarse el orden de los parámetros, agregarse o quitarse, aunque se ignoran también los parámetros nuevos, y los valores que no se hayan asignado se inicializan en cero.

3.2 Los Archivos de Salida

Cada archivo *xml* tiene el siguiente formato:

```
<?xml version="1.0" ?>
<Set size="cantidad_de_sistemas" n="n">
  <S count="contador_de_sistemas" U="U" mcm="hiperperíodo">
    <i nro="i" C="Ci" T="Ti" D="Di" B="Bi" J="Ji"
      Of="Ofi" CO="Coi" />
    ...
  </S>
  ...
</Set>
```

En el encabezado se muestra la cantidad de sistemas solicitados (*size*) y la cantidad de tareas en cada sistema (*n*). Luego se listan los sistemas, con las tareas que los componen. De cada sistema (*S*) se muestra su número de orden dentro del archivo (*count*), su factor de utilización (*U*) y su hiperperíodo (*mcm*).

En cada línea dentro de la etiqueta `<S> ... </S>` se muestran los valores de los parámetros para cada tarea. Los parámetros de generación opcional que no hayan sido solicitados se muestran con valor cero.

Dentro de cada sistema las tareas están ordenadas de menor a mayor por períodos o por vencimientos, según lo solicitado en la configuración. Para contemplar los casos de conjuntos de tareas donde el D_i es arbitrario respecto del T_i , si se pide ordenar por vencimiento, se toma $D_i - J_i$ (como lo realizado en [7]).

La cantidad de sistemas en el archivo de respaldo puede ser mayor a la solicitada debido al descarte de sistemas cuyo factor de utilización no se acerca al solicitado.

Ejemplo de archivo de salida:

```
<?xml version="1.0" ?>
<Set size=" 10" n=" 3" >
  <S count=" 1" U=" 80" mcm=" 394016040" >
    <i nro=" 1" C=" 10.89" T=" 94" D=" 82.11" B=" 0.48" J="
0.40" Of=" 0" Co=" 0" />
    <i nro=" 2" C="288.30" T=" 792" D="689.57" B=" 8.48" J="
1.71" Of=" 0" Co=" 0" />
    <i nro=" 3" C="3387.45" T=" 10585" D="9562.67" B=" 92.14"
J="300.23" Of=" 0" Co=" 0" />
  </S>
```

El tipo de archivo texto muestra los mismos datos, pero reemplazando las etiquetas *xml* por el carácter ‘.’. A continuación se incluye un fragmento del archivo principal de salida en formato *txt* para 10 sistemas de 3 tareas con factor de utilización del 80%:

```
10: 3
1: 80: 394016040
  1: 10.89: 94: 82.11: 0.48: 0.40: 0: 0
  2: 288.30: 792: 689.57: 8.48: 1.71: 0: 0
  3: 3387.45: 10585: 9562.67: 92.14: 300.23: 0: 0
```

El nombre de estos archivos se construye con el prefijo *rtts_* seguido del factor de utilización solicitado, un guión bajo y la cantidad de tareas en cada sistema. El nombre del archivo de respaldo tiene además el sufijo *-bkp*. La nomenclatura elegida tiene por finalidad facilitar la programación de un paquete de pruebas, cuya salida quedaría en archivos separados.

4 Resultados Obtenidos

Para evaluar la performance desde el punto de vista de la calidad de los conjuntos de tareas, de la misma manera que en el trabajo de Bini ([5]), se realizaron generaciones

de conjuntos de distintos tamaños y distintos factores de utilización. En este caso, se realizaron exhaustivas corridas del generador para cien mil sistemas para cada factor de utilización en conjuntos de 10, 20 y 50 tareas, variando el factor de utilización de diez en diez. En total fueron 30 ejecuciones del generador logrando así tres millones de sistemas distintos.

La calidad del conjunto de tareas está determinada por el valor de λ (ecuación (1)). Este valor precisa la diferencia existente entre el mayor y menor factor de utilización de las tareas de cada sistema. Consecuentemente, para cada sistema generado se obtuvo el valor de λ y se calculó su función de densidad de probabilidad.

En los siguientes gráficos se observará que a medida que se incrementa el tamaño de cada conjunto, los valores de λ son menores y menos dispersos, lo cual indica una mejor distribución del factor de utilización entre las tareas de cada sistema.

A continuación se presentan los gráficos para cada factor de utilización. En el eje x se representan los λ calculados, y en el eje y su función de densidad de probabilidad.

En los gráficos para conjuntos de diez tareas (figuras 1 a 10), λ se encuentra alrededor de 0,23 lo cual indica que existe una diferencia centrada en el 23%.

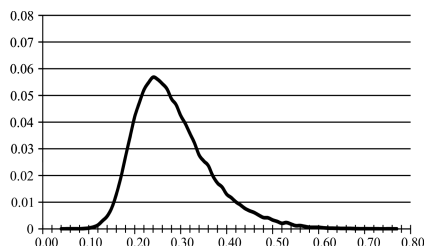


Figura 1. Conjunto de 10 tareas $U= 10\%$.

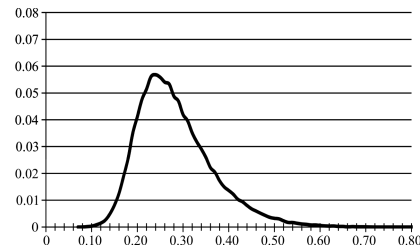


Figura 4. Conjunto de 10 tareas $U= 40\%$.

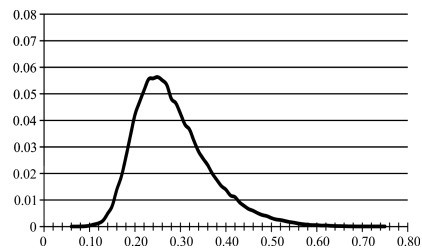


Figura 2. Conjunto de 10 tareas $U= 20\%$.

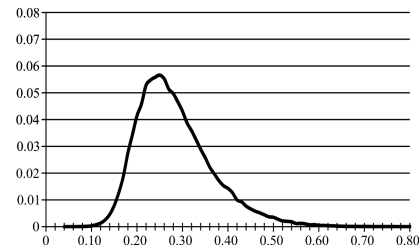


Figura 5. Conjunto de 10 tareas $U= 50\%$.

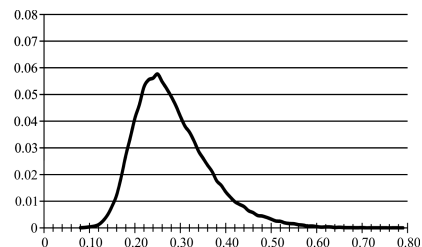


Figura 3. Conjunto de 10 tareas $U= 30\%$.

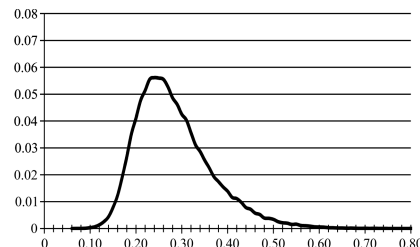


Figura 6. Conjunto de 10 tareas $U= 60\%$.

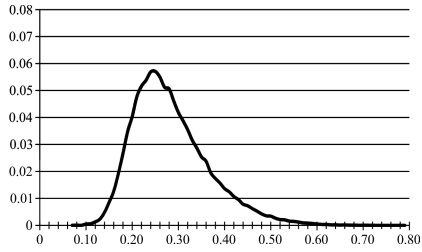


Figura 7. Conjunto de 10 tareas $U= 70\%$.

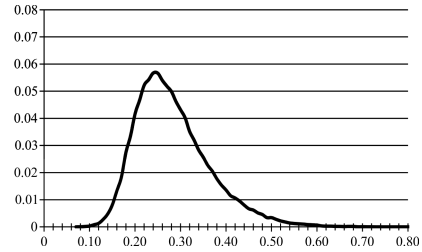


Figura 9. Conjunto de 10 tareas $U= 90\%$.

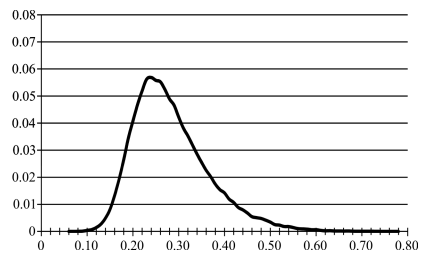


Figura 8. Conjunto de 10 tareas $U= 80\%$.

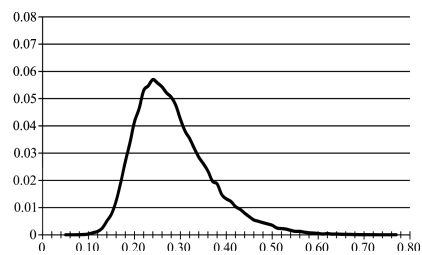


Figura 10. Conjunto de 10 tareas $U= 100\%$.

Para conjuntos de veinte tareas (figuras 11 a 20), λ tiene valores centrados en 0,15.

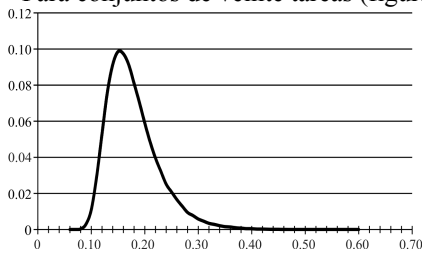


Figura 11. Conjunto de 20 tareas $U= 10\%$.

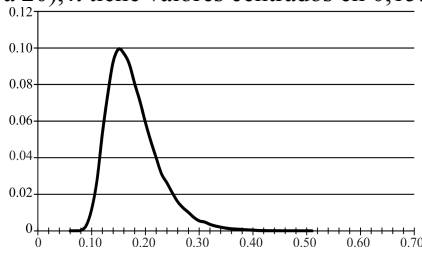


Figura 14. Conjunto de 20 tareas $U= 40\%$.

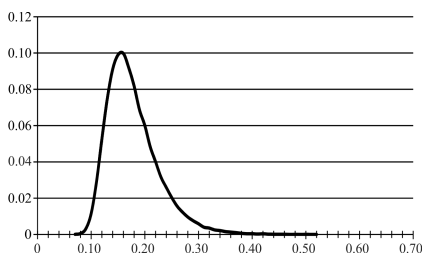


Figura 12. Conjunto de 20 tareas $U= 20\%$.

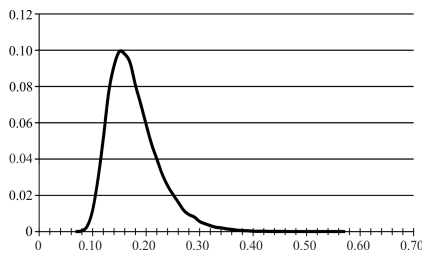


Figura 15. Conjunto de 20 tareas $U= 50\%$.

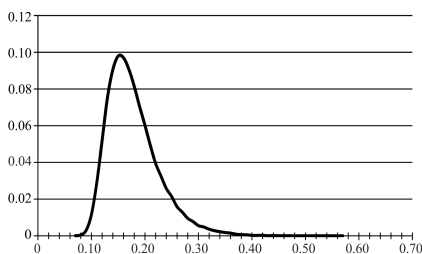


Figura 13. Conjunto de 20 tareas $U= 30\%$.

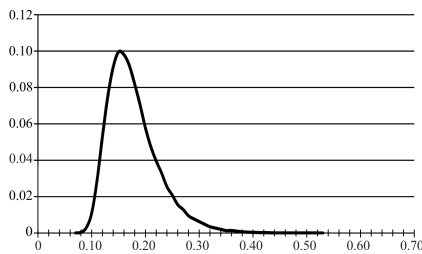


Figura 16. Conjunto de 20 tareas $U= 60\%$.

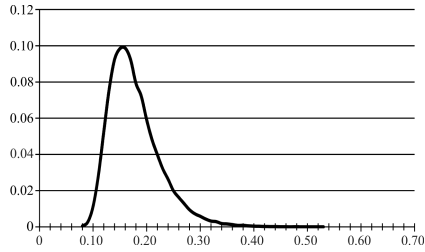


Figura 17. Conjunto de 20 tareas $U= 70\%$.

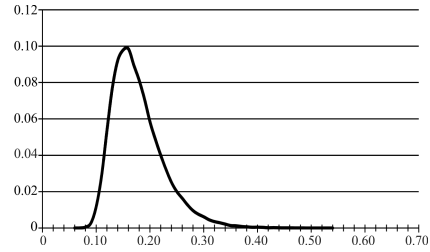


Figura 19. Conjunto de 20 tareas $U= 90\%$.

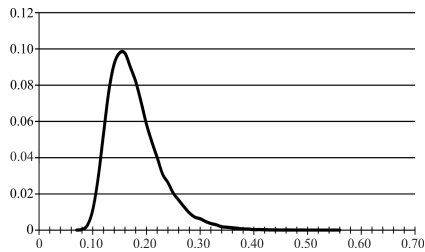


Figura 18. Conjunto de 20 tareas $U= 80\%$.

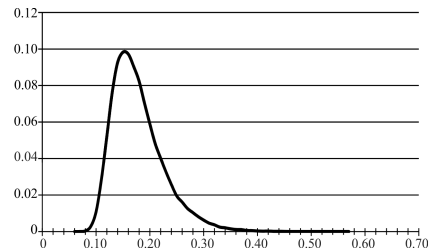


Figura 20. Conjunto de 20 tareas $U= 100\%$.

Para conjuntos de 50 tareas (figuras 21 a 30), λ está centrada en 0,078.

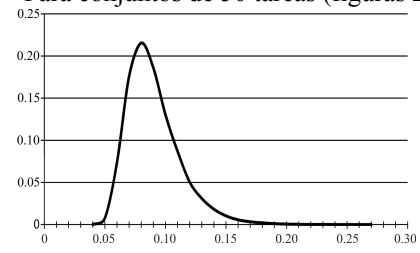


Figura 21. Conjunto de 50 tareas $U= 10\%$.

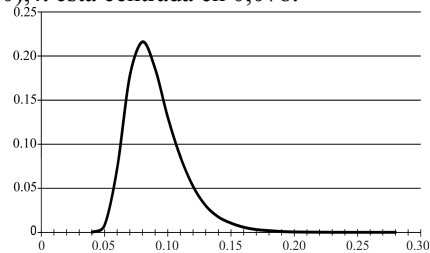


Figura 24. Conjunto de 50 tareas $U= 40\%$.

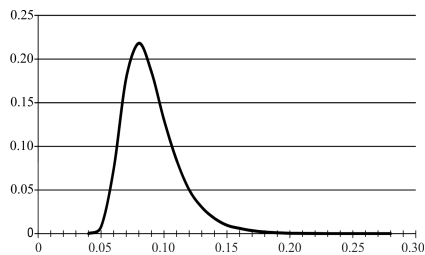


Figura 22. Conjunto de 50 tareas $U= 20\%$.

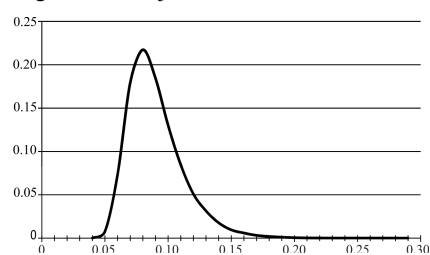


Figura 25. Conjunto de 50 tareas $U= 50\%$.

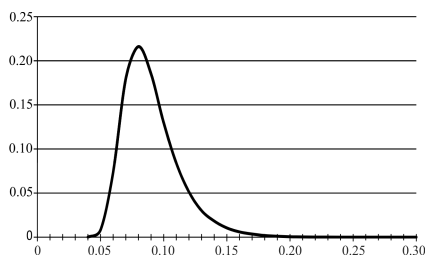


Figura 23. Conjunto de 50 tareas $U= 30\%$.

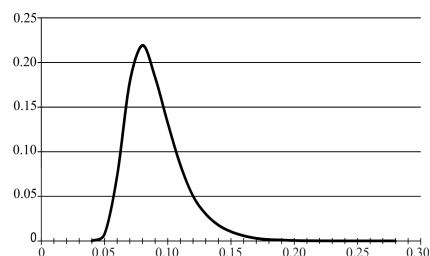


Figura 26. Conjunto de 50 tareas $U= 60\%$.

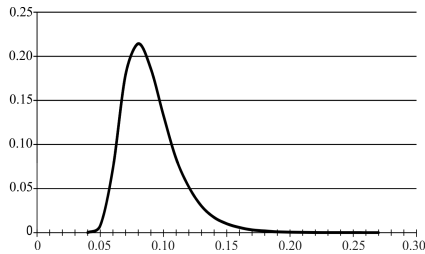


Figura 27. Conjunto de 50 tareas $U= 70\%$.

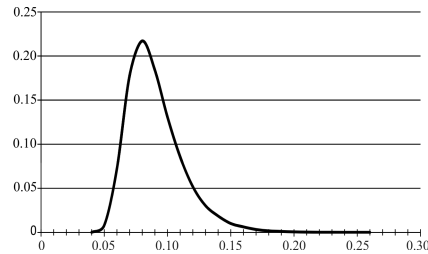


Figura 29. Conjunto de 50 tareas $U= 90\%$.

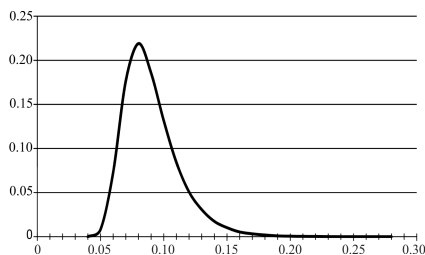


Figura 28. Conjunto de 50 tareas $U= 80\%$.

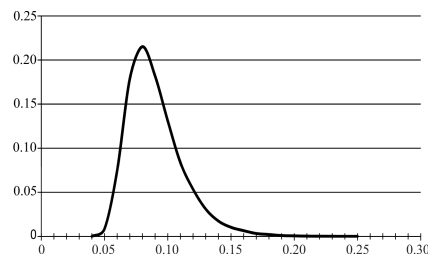


Figura 30. Conjunto de 50 tareas $U= 100\%$.

5 Conclusiones y Trabajos Futuros

Se construyó un software de generación de tareas teniendo como principal objetivo la flexibilidad, para que se pueda adaptar a distintas disciplinas de *Sistemas de Tiempo Real*. Como se mencionó en la introducción, para comprobar la eficacia de los algoritmos es necesario validarlos, y es mediante este software de generación de tareas que se podrán realizar pruebas sobre miles o millones de sistemas sintéticos.

Su flexibilidad se basa en la posibilidad de definir su configuración según la aplicación que lo va a utilizar, dado que cada una de ellas suele tener requerimientos diferentes.

Este software también garantiza una distribución aleatoria homogénea en la asignación del factor de utilización entre las tareas, independiente del factor de utilización solicitado para los sistemas, como evidencian los gráficos de la sección 4.

Es sencillo de utilizar, genera millones de sistemas en un tiempo razonablemente pequeño y fue escrito en ADA 2005.

En trabajos futuros, se realizarán algunas mejoras para facilitar su uso, como: agregar la interfaz gráfica para poder ingresar la configuración del sistema sin tener que cambiar el archivo de configuración, generar valores con diferentes distribuciones de probabilidad (como exponencial o de *poisson*), verificar si los sistemas generados son planificables para distintos tipos de disciplinas de prioridades y finalmente agregar otros formatos de archivos de salida.

Referencias

- [1] AdaCore, "Ada 2005 - <http://libre.adacore.com/libre/>," 2005 ed, 2010.
- [2] C. L. Liu and J. W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment," *Journal of the ACM*, vol. 20, pp. 46-61, 1973.
- [3] N. C. Audsley, A. Burns, M. F. Richardson, K. Tindell, and A. J. Wellings, "Applying New Scheduling Theory to Static Priority Preemptive Scheduling," *Software Engineering Journal*, vol. 8, pp. 284-292, 1993.
- [4] E. Bini and C. B. Giorgio, "Schedulability Analysis of Periodic Fixed Priority Systems," *IEEE Trans. on Computers*, vol. 53, pp. 1462-1473, November 2004.
- [5] E. Bini and G. C. Buttazzo, "Biasing Effects in Schedulability Measures," in *Proceedings of the 16th Euromicro Conference on Real-Time Systems*; IEEE Computer Society, 2004, pp. 196-203.
- [6] R. I. Davis and A. Burns, "Priority Assignment for Global Fixed Priority Pre-emptive Scheduling in Multiprocessor Real -Time Systems.," University of York, York 2009.
- [7] R. Davis and A. Burns, "Response Time Upper Bounds for Fixed Priority Real-Time Systems," in *The 29th IEEE Real-Time Systems Symposium*, IEEE, Ed. Barcelona, Spain; IEEE Computer Society, 2008, pp. 407-418.
- [8] R. M. Santos, J. M. Urriza, J. Santos, and J. D. Orozco, "Heuristic use of Singularities for On-Line Scheduling of Real-Time Mandatory/Reward-Based Optional Systems," in *14th Euromicro Conference on Real-Time Systems*, Vienna, Austria, 2002, pp. 103-110.
- [9] R. M. Santos, J. M. Urriza, J. Santos, and J. D. Orozco, "Diagramación on-line de sistemas de tareas de tiempo real periódicas mandatorias duras/opcionales basadas en recompensas con factor de depreciación," in *31 JAIIO AST2002*, Santa Fe, Argentina, 2002, p. 148.