**Universidad Tecnológica de Panamá**

**Centro Regional de Azuero**

**Facultad de Ingeniería de Sistemas Computacionales**

**Ingeniería de Sistemas y Computación**

**Modelado y Simulación**

**“Investigación sobre la Maquina de Ernie, la bajara española y los generadores de números aleatorios en los simuladores GPSS, Arenas y lenguajes vb, c++ y c”**

**Profesora: Gloria Villalaz**

**Estudiantes:**

**Miguel Batista 6-719-1146**

**Joe Torres 6-719-2055**

**II SEMESTRE**

**2017**

**Generadores de Números Aleatorios**

**GPSS**

La secuencia de números aleatorios en el GPSS se obtiene a través de un generador congruencial multiplicativo que tiene un período máximo de 32 bits, este período excluye el 0. Los generadores en el GPSS no necesitan declararse. La semilla inicial del generador de números aleatorios es igual al número de entidad RN, por ejemplo, si elegimos RN2, la semilla inicial es 2.

La semilla puede cambiarse a través de la sentencia RMULT.

Sólo podemos controlar con RMULT las semillas de los 8 generadores que tiene el lenguaje.

El generador Pseudo-random de GPSS World se basa en el algoritmo multiplicativo-congruencial de Lehmer con un período máximo. El algoritmo produce números pseudo-aleatorios en el intervalo 0 a y genera números aleatorios diferentes antes de repetirse.

A menos que sea cambiada por un RMULT, la semilla inicial es igual al número que identifica el número aleatorio elegido. RN6 comienza con una semilla 6.

Usos por parte del sistema: el GPSS World usa los generadores de números aleatorios en la programación en caso de que los tiempos de ocurrencia de eventos estén empatados. En los bloques TRANSFER en modo fraccional y para la generación de números aleatorios para los bloques GENERATE y ADVANCE.

**ARENAS SIMULATOR**

ARENA ofrece una amplia gama de funciones de distribución para modelar tiempos de llegada, tiempos de proceso, etc. Para utilizar cualquiera de las funciones sólo es necesario introducir el nombre (o su forma abreviada) y luego proporcionar los parámetros requeridos. ARENA genera entonces un número aleatorio uniformemente distribuido en el rango de 0 a 1 que, a su vez, se utiliza para seleccionar un valor de la distribución especificada. ARENA proporciona 10 flujos de números aleatorios para seleccionar o puede, si lo desea, aceptar el flujo predeterminado (10). Para utilizar, por ejemplo, la distribución exponencial con un valor medio de 4, puede introducir el nombre completo y el parámetro UNIFORM (4) o el nombre abreviado (cuatro primeras letras) y el parámetro UNIF (4). Puede seleccionar un flujo de números aleatorios en particular agregando un parámetro extra a la sentencia ARENA y esto será automáticamente entendido por el compilador. Por ejemplo, para utilizar una distribución exponencial con una media de 4, pero al seleccionar un flujo de números aleatorios 8, modifique la declaración anterior a UNIF (4,8). El Apéndice B contiene todos los detalles sobre las funciones disponibles.

Todas las distribuciones de ARENA se basan en un generador congruencial multiplicativo para números uniformemente distribuidos entre 0 y 1. La constante multiplicativa es 16,807 y el módulo es 2 31 - 1 Esto da una longitud de ciclo de aproximadamente 2,10 9 y no tenemos conocimiento de ninguna problemas con este generador.

Si, por cualquier razón, los 10 flujos de números aleatorios no son suficientes para sus necesidades, es posible ampliar el rango utilizando el elemento ` SEEDS '. El elemento de SEEDS le permite especificar el valor inicial de semilla para cada flujo o definir un nuevo flujo al que se puede hacer referencia por nombre o número. También puede decidir cuándo replicar varias ejecuciones de una simulación para restablecer la semilla entre las ejecuciones o permitir que la secuencia de números continúe entre las ejecuciones (la predeterminada). El restablecimiento, por supuesto, generará exactamente la misma secuencia para cada ejecución que a veces puede ser útil si, por ejemplo, cambia un parámetro entre las ejecuciones.

**La Baraja Española**

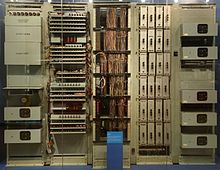
La baraja española consiste en un mazo de 48 naipes o cartas, clasificados en 4 "palos" y numerados del 1 al 12 (en la de 40 naipes, faltan los ochos y los nueves). Ciertos mazos incluyen además 2 comodines. Las figuras de la baraja española correspondientes a los números 10, 11 y 12, y se llaman "sota", "caballo" y "rey" respectivamente y, a diferencia de la baraja francesa, se representan de cuerpo entero.

Los cuatro palos son: oros, espadas, copas y bastos (en su respectivo orden ascendente). Para ciertos juegos se dividen en palos cortos (oros y copas) y largos (bastos y espadas).

El recuadro que rodea la figura tiene una señal para distinguir el palo de que se trate sin desplegar del todo las cartas: las copas una interrupción, las espadas dos, los bastos tres y los oros ninguna. Esta señal se llama la pinta y ha dejado la expresión: le conocí por la pinta (en la baraja francesa, esta señal son las figuritas que hay debajo del número).

Esta baraja es la base de las cartas del Tarot, salvo que se trata de 56 cartas (es lo que se llaman los arcanos menores), con los mismos palos, y que llegan hasta el 10 y además sota, caballo, reina y rey. En el tarot se añaden, además, los arcanos mayores, que son los que, generalmente, se usan para la adivinación.

**ERNIE**



ERNIE es un generador de números aleatorios de hardware . El primer ERNIE fue construido en la estación de investigación de la oficina de correos por un equipo llevado por Sidney Broadhurst. Los diseñadores fueron Tommy Flowers y Harry Fensom y se basa en Colossus , la primera computadora digital del mundo.Fue introducido en 1957, y generó números de enlace basados ​​en el ruido de señal creado por los tubos de neón . ERNIE 1 está en las colecciones del museo de la ciencia en Londres y estaba en la exhibición entre 2008 y 2015.

(ERNIE1)

ERNIE es siglas para el "equipo electrónico del indicador del número al azar". ERNIE 2 sustituyó al primer ERNIE en 1972. ERNIE 3 en 1988 tenía el tamaño de una computadora personal; al final de su vida tomó cinco horas y media para completar su sorteo mensual.

En agosto de 2004 ERNIE 4 se puso en servicio en previsión de un aumento de premios cada mes a partir de septiembre de 2004. Desarrollado por LogicaCMG , es 500 veces más rápido que el original y genera un millón de números por hora; estos se comprueban contra una lista de bonos válidos. En comparación, el ERNIE original generó 2.000 números por hora y era del tamaño de una furgoneta.



ERNIE 4 utiliza el ruido térmico en los transistores como su fuente de aleatoriedad para generar números aleatorios verdaderos; el ERNIE original utilizó un diodo de neón de gas. Por el contrario, los números pseudoaleatorios , aunque a veces simplemente se denominan aleatorios, se producen de forma determinista por el algoritmo utilizado para generarlos. La aleatoriedad de los números de ERNIE deriva de fluctuaciones estadísticas aleatorias en los procesos físicos involucrados. La producción de ERNIE es evaluada de forma independiente cada mes por el Departamento de Actuarios del Gobierno , y el sorteo sólo es válido si se certifica que es estadísticamente consistente con la aleatoriedad.

(Caja de dinero ERNIE)

**Generadores de números aleatorios**

**Lenguaje C y C++**

Lenguaje C y C++ usa las funciones:

**Srand**

Inicializa el generador de números aleatorios.

El generador de números pseudo-aleatorios se inicializa utilizando el argumento pasado como semilla.

Para cada valor de semilla diferente usado en una llamada srand, se puede esperar que el generador de números pseudo-aleatorios genere una sucesión de resultados diferente en las llamadas posteriores a rand.

Dos inicializaciones diferentes con la misma semilla generarán la misma sucesión de resultados en las llamadas posteriores a rand.

Si se establece la semilla 1, el generador se reinicializa a su valor inicial y produce los mismos valores que antes de cualquier llamada a rand o srand. Con el fin de generar números al azar,

srand se inicializa normalmente a un valor de tiempo de ejecución distintivo, como el valor devuelto por la función hora. Esto es bastante distintivo para las necesidades de asignación al azar más triviales.

**Rand**

Devuelve un número pseudo-aleatorio integral en el rango entre 0 y RAND\_MAX. Este número es generado por un algoritmo que devuelve una secuencia de números aparentemente no relacionados cada vez que se llama. Este algoritmo utiliza una semilla para generar la serie, que debe ser inicializado a un valor distintivo utilizando la función srand.

RAND\_MAX es una constante definida en <cstdlib>.

Una forma típica de generar números pseudo-aleatorios triviales en un rango determinado usando “rand” es usar el módulo del valor devuelto por el intervalo de rango y agregar el valor inicial del rango, ejemplo:

v1 = rand() % 100; // v1 in the range 0 to 99

v2 = rand() % 100 + 1; // v2 in the range 1 to 100

v3 = rand() % 30 + 1985; // v3 in the range 1985-2014

Observe sin embargo que esta operación modulo no genera números aleatorios uniformemente distribuidos en el span (ya que en la mayoría de los casos esta operación hace que los números menores sean más probables).

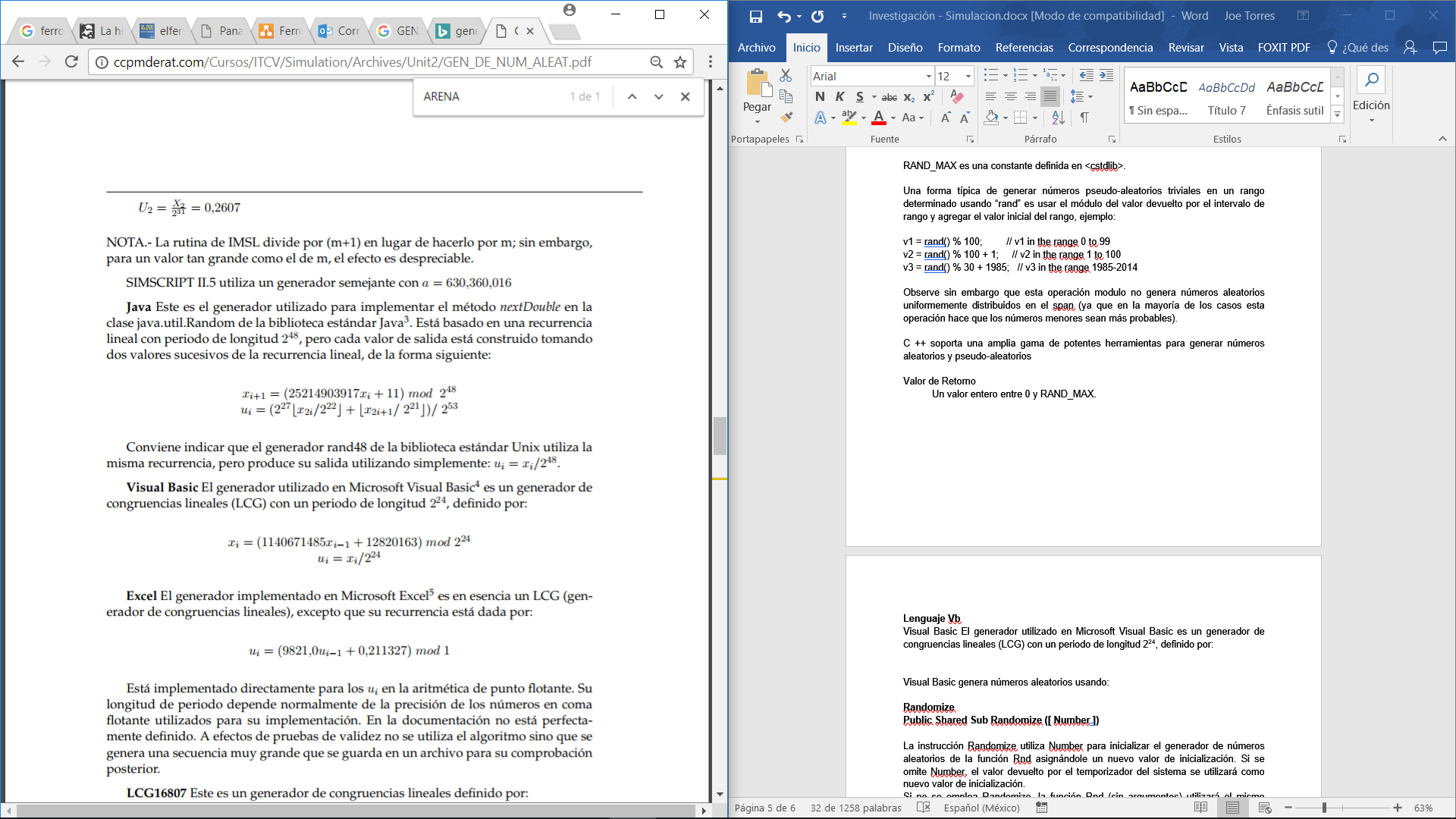
C ++ soporta una amplia gama de potentes herramientas para generar números aleatorios y pseudo-aleatorios

Valor de Retorno

Un valor entero entre 0 y RAND\_MAX.

**Lenguaje Vb**

El generador utilizado en Microsoft Visual Basic 4 es un generador de congruencias lineales (LCG) con un periodo de longitud 224, definido por:



Visual Basic genera números aleatorios usando:

**Randomize**

**Public Shared Sub Randomize ([ Number ])**

La instrucción Randomize utiliza Number para inicializar el generador de números aleatorios de la función Rnd asignándole un nuevo valor de inicialización. Si se omite Number, el valor devuelto por el temporizador del sistema se utilizará como nuevo valor de inicialización.

Si no se emplea Randomize, la función Rnd (sin argumentos) utilizará el mismo número como valor de inicialización la primera vez que se le llame y más adelante utilizará como valor de inicialización el último número generado.

**Rnd**

**Public Shared Function Rnd[(Number)] As Single**

La función Rnd devuelve un valor menor que 1, pero mayor o igual a cero.

El valor de Number determina la forma en que Rnd genera un número aleatorio.

Para cualquier valor de inicialización dado, se genera la misma secuencia de números ya que cada llamada sucesiva que se hace a la función Rnd utiliza el número anteriormente generado como inicialización para el siguiente número de la secuencia.

Antes de llamar a la función Rnd, utilice la instrucción Randomize sin argumento para inicializar el generador de números aleatorios con un valor de inicialización basado en el temporizador del sistema.

|  |  |
| --- | --- |
| **Si Number es** | **Rnd genera** |
| Menor que cero | Siempre el mismo número, utilizando *Number* como valor de inicialización. |
| Mayor que cero | El siguiente número aleatorio en la secuencia. |
| Igual a cero | El número generado por última vez. |
| Omitido | El siguiente número aleatorio en la secuencia. |

Para producir enteros aleatorios en un intervalo dado, utilice la siguiente fórmula.

VB

randomValue = CInt(Math.Floor((upperbound - lowerbound + 1) \* Rnd())) + lowerbound

Aquí, upperbound es el número mayor en el intervalo y lowerbound es el número menor.

Ejemplo:

Randomize()

Dim value As Integer = CInt(Int((6 \* Rnd()) + 1))