

UNIVERSIDAD AUTÓNOMA DE TAMAULIPAS

SIMULACIÓN DE SISTEMAS ING. RAMÓN V. ROQUE HERNÁNDEZ, M.C.

ramonroque@yahoo.com ramonroque@hotmail.com
<http://www.geocities.com/ramonroque>

Descripción general de la materia: La materia introduce al alumno en los conceptos y técnicas de Simulación por computadora mediante ejemplos prácticos aplicables al mundo real.

Contenido:

Unidad I.- Teoría de modelos y simulación

- Simulación
 - Introducción y Conceptos
 - Usos, limitaciones y Áreas de aplicación de la simulación
 - La simulación en la historia
 - Simulador por computadora
 - Simulación como proceso experimental
 - Resolución analítica y simulación
 - Simulación de montecarlo
 - Etapas de una simulación
- Sistemas
 - Conceptos
 - Clasificación de sistemas
- Modelos
 - Utilidad de los modelos
 - Pasos para realizar un modelo
 - Características y clasificación de los modelos
 - La computadora en la simulación de modelos

Unidad II.- Técnicas de Simulación en Lenguajes de Programación (C#)

- Nociones del lenguaje
- Uso de números aleatorios para experimentos de simulación
- Implementación de la Simulación de Montecarlo
- Diversos Proyectos de simulación aplicada

Unidad III.- Técnicas de Simulación en Paquetes Computacionales

- Paquetes Administrativos de uso General (Excel)
 - Fundamentos de Excel
 - Técnicas de Simulación utilizadas en Excel
 - Proyectos de simulación en Excel
- Paquetes Especializados

Unidad IV.- Generadores de Números pseudo-aleatorios

- Métodos primitivos
- Congruencial Mixto y Congruencial Multiplicativo

Unidad V.- Pruebas Estadísticas para los números pseudo-aleatorios

- Pruebas de los promedios y frecuencias.

Forma de Evaluar: Asistencia, Tareas, Participación, Prácticas, Proyectos, Exámenes Parciales y Ordinario.

Alguna Bibliografía Recomendada:

- Simulación: Un enfoque práctico, Coss Bu, Ed. Limusa ISBN: 968-18-1506-8
- Simulación: Métodos y Aplicaciones, David Rios Insua, Ed. AlfaOmega ISBN: 970-15-0509-3
- Simulación, Sheldon M.Ross, Pearson Educación ISBN : 970-17-0259-X

Caso de estudio

POLAROID: Simulación de sistemas para alcanzar el éxito

Cuando Polaroid Corporation, fabricante de la conocida cámara instantánea, diseña un nuevo proceso de manufactura, los ingenieros prueban el nuevo diseño por simulación. Al simular cómo operará la línea de ensamble, los ingenieros de Polaroid pueden obtener respuestas a preguntas importantes antes de que la compañía invierta tiempo, esfuerzo y recursos financieros instalando computadoras, cintas transportadoras, robots y otros equipos que forman la línea de ensamble.

La cámara Polaroid fue desarrollada en la década de los cuarentas por Edwin Land. Desde entonces, la investigación y el desarrollo condujeron a la creación de modelos innovadores que cada vez tenían más características. Pero los productos revolucionarios, junto con los avances en tecnología de producción, requieren de nuevos procedimientos de fabricación y ensamble.

La simulación es equivalente a probar la línea de producción antes que ésta exista. Se puede verificar la velocidad de los procesos y vigilar el intercambio entre componentes clave. Durante la simulación pueden aparecer desarrollos y resultados inesperados o no deseables que requieren de correcciones. Por ejemplo, si bajo ciertas condiciones existe el riesgo de que los transportes controlados por computadora choquen entre sí, la simulación descubrirá la posibilidad de que esto ocurra. Asimismo, también se hará evidente a través de la simulación el riesgo de que los productos terminados sufran retrasos en la sección de empaque.

En Polaroid la información es un recurso muy poderoso. La industria con la que compite Polaroid se aboca a proporcionar el producto fotográfico adecuado al menor costo. Aunque los productos de Polaroid están protegidos por patentes, la compañía debe vender sus cámaras con precios competitivos al mismo tiempo que asegura un buen margen de ganancias para la firma. Por ejemplo, se pueden ajustar los niveles excesivos de inventario o la acumulación de partes para obtener una mejor planificación y control, pero esto solo es posible si se cuenta con la información correcta. En una industria tan competitiva como ésta, cualquier oportunidad para mejorar la eficiencia en la manufactura aumentará el éxito de Polaroid en el mercado.

Hace tiempo, la simulación significaba pilas de papel que contenían números y más números generados por una computadora, que indicaban promedios y desviaciones de los promedios entre otras cosas. En Polaroid todo esto ha cambiado. Los ingenieros utilizan computadoras de escritorio que muestran en forma visual los procesos y actividades de fabricación. Los niveles de inventario mostrados sobre la pantalla cambian a medida que se van terminando los productos. Las actividades de ensamble son representadas por medio de técnicas de animación, no por los listados que caracterizaron en el pasado a la simulación. Y ahora los gerentes pueden obtener con rapidez una idea con respecto a la forma en que operará la línea de producción, sin necesidad de asesoría sobre cómo interpretar los datos estadísticos que describen el comportamiento de un modelo matemático en proceso.

En el futuro la simulación se trasladará a la planta de fabricación. Herramientas fáciles de emplear permitirán a los técnicos de la planta hacer la simulación correcta de lo que ocurre a su alrededor.

El tipo de simulación que ahora se emplea en Polaroid Corporation tendrá cada vez mayor auge en la medida en que las computadoras y métodos de análisis aumenten su capacidad y sofisticación. El color y la animación junto con software poderoso, permitirán describir de manera más completa la información sobre los sistemas antes que estos sean desarrollados, hecho que trae beneficios para todos.

UNIDAD I: Teoría de Modelos y Simulación

Simulación

Cuando alguien tiene la responsabilidad de conducir un sistema dado, como por ejemplo: un banco, una ciudad, un sistema de transporte, etc., debe tomar continuamente decisiones acerca de las acciones que ejecutará sobre el sistema. Estas decisiones deben ser tales que la conducta resultante del sistema satisfaga de la mejor manera posible los objetivos planteados.

Para poder decidir correctamente es necesario saber cómo responderá el sistema ante una determinada acción. Esto podría hacerse por experimentación con el sistema mismo; pero factores de costos, seguridad y otros hacen que esta opción generalmente no sea viable. A fin de superar estos inconvenientes, se reemplaza el sistema real por otro sistema que en la mayoría de los casos es una versión simplificada. Este último sistema es el modelo a utilizar para llevar a cabo las experiencias necesarias sin los inconvenientes planteados anteriormente.

Al proceso de experimentar con un modelo se denomina *simulación*. Al proceso de diseñar el plan de experimentación para adoptar la mejor decisión se denomina *optimización*. Si el plan de experimentación se lleva a cabo con el solo objeto de aprender a conducir el sistema, entonces se denomina *entrenamiento* o *capacitación*.

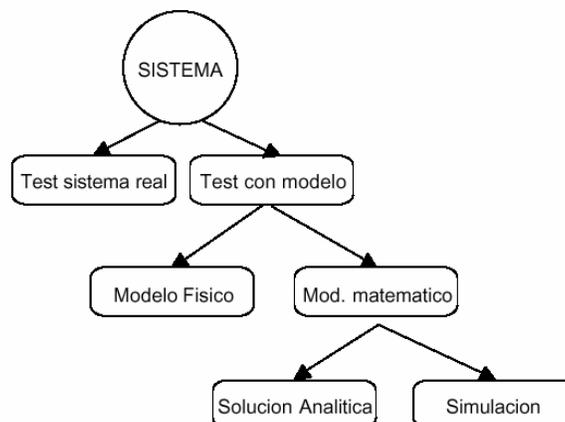


Figura: Maneras de estudiar un sistema

En este punto, es conveniente plantear las siguientes definiciones:

- **Sistema:** Conjunto de objetos o ideas que están interrelacionados entre sí como una unidad para la consecución de un fin (Shannon, 1988). También se puede definir como la porción del Universo que será objeto de la simulación.
- **Modelo:** Un objeto X es un modelo del sistema Y para el observador Z , si Z puede emplear X para responder cuestiones que le interesan acerca de Y (Minsky).

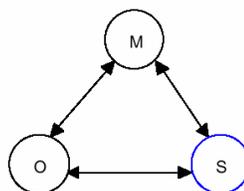


Figura: Modelo, Sistema, Observador

- **Simulación:** Simulación es el proceso de diseñar un modelo de un sistema real y llevar a cabo experimentos con él, con la finalidad de aprender el comportamiento del sistema o de evaluar diversas estrategias para el funcionamiento del sistema (Shannon, 1988).

Uso de la simulación

La simulación es conveniente cuando:

- No existe una formulación matemática analíticamente resoluble. Muchos sistemas reales no pueden ser modelados matemáticamente con las herramientas actualmente disponibles, por ejemplo la conducta de un cliente de un banco.
- Existe una formulación matemática, pero es difícil obtener una solución analítica. Los modelos matemáticos utilizados para modelar un reactor nuclear o una planta química son imposibles de resolver en forma analítica sin realizar serias simplificaciones.
- No existe el sistema real. Es problema del ingeniero que tiene que diseñar un sistema nuevo. El diseño del sistema mejorará notablemente si se cuenta con un modelo adecuado para realizar experimentos.
- Los experimentos son imposibles debido a impedimentos económicos, de seguridad, de calidad o éticos. En este caso el sistema real esta disponible para realizar experimentos, pero la dificultad de los mismos hace que se descarte esta opción. Un ejemplo de esto es la imposibilidad de provocar fallas en un avión real para evaluar la conducta del piloto, tampoco se puede variar el valor de un impuesto a para evaluar la reacción del mercado.
- El sistema evoluciona muy lentamente o muy rápidamente. Un ejemplo de dinámica lenta es el problema de los científicos que estudian la evolución del clima. Ellos deben predecir la conducta futura del clima dadas las condiciones actuales, no pueden esperar a que un tornado arrase una ciudad para luego dar el mensaje de alerta. Por el contrario, existen fenómenos muy rápidos que deben ser simulados para poder observarlos en detalles, por ejemplo una explosión.

Limitaciones en la aplicación de la simulación

Entre las posibles desventajas de la simulación se pueden citar:

- El desarrollo de un modelo puede ser costoso, laborioso y lento.
- Existe la posibilidad de cometer errores. No se debe olvidar que la experimentación se lleva a cabo con un modelo y no con el sistema real; entonces, si el modelo está mal o se cometen errores en su manejo, los resultados también serán incorrectos.
- No se puede conocer el grado de imprecisión de los resultados. Por lo general el modelo se utiliza para experimentar situaciones nunca planteadas en el sistema real, por lo tanto no existe información previa para estimar el grado de correspondencia entre la respuesta del modelo y la del sistema real.

Áreas en que se aplica la simulación

Actualmente la simulación presta un invaluable servicio en casi todas las áreas posibles, algunas de ellas son:

- **Procesos de manufacturas:** Ayuda a detectar cuellos de botellas, a distribuir personal, determinar la política de producción.
- **Plantas industriales:** Brinda información para establecer las condiciones óptimas de operación, y para la elaboración de procedimientos de operación y de emergencias.
- **Sistemas públicos:** Predice la demanda de energía durante las diferentes épocas del año, anticipa el comportamiento del clima, predice la forma de propagación de enfermedades.
- **Sistemas de transportes:** Detecta zonas de posible congestionamiento, zonas con mayor riesgo de accidentes, predice la demanda para cada hora del día.
- **Construcción:** Predice el efecto de los vientos y temblores sobre la estabilidad de los edificios, provee información sobre las condiciones de iluminación y condiciones ambientales en el interior de los mismos, detecta las partes de las estructuras que deben ser reforzadas.
- **Diseño:** Permite la selección adecuada de materiales y formas. Posibilita estudiar la sensibilidad del diseño con respecto a parámetros no controlables.
- **Educación:** Es una excelente herramienta para ayudar a comprender un sistema real debido a que puede expandir, comprimir o detener el tiempo, y además es capaz de brindar información sobre variables que no pueden ser medidas en el sistema real.
- **Capacitación:** Dado que el riesgo y los costos son casi nulos, una persona puede utilizar el simulador para aprender por sí misma utilizando el método más natural para aprender: el de prueba y error.
- **Optimización:** Se emplea para determinar situaciones del proceso en las que se logra una mejoría de rendimiento.
- **Además en:** Siniestros, catastrofes, diagnosis, física, química, matemáticas, biología, medicina.

La Simulación en la Historia

La importancia de la Simulación es evidente al considerar el impacto que tuvieron algunos eventos como:

- **La Perestroyka:** Estudios de simulación efectuados en Rusia en las décadas del 70 y 80 convencieron a los dirigentes de la necesidad de plantear un fuerte cambio en la economía de ese país.
- **La caída de la bolsa de New York en 1988:** La utilización de programas de simulación por parte de los corredores de la bolsa causó una falsa inestabilidad que provocó la caída.
- **El regreso del Apolo 13:** La simulación jugó un rol fundamental en la determinación del plan de emergencia. La nave retornó con éxito a pesar de las graves averías.
- **Los Voyagers:** Gracias a la simulación se pudieron establecer los itinerarios óptimos para estas naves con un mínimo consumo de energía aprovechando la atracción gravitacional de los planetas.
- **Proyecto Monte Carlo:** Von Newman y Ulam (1945) emplearon simulación para estudiar reacciones nucleares.
- **Los modelos del planeta:** Algunos plantean la posibilidad de un calentamiento global debido al efecto invernadero. Otros plantean la posibilidad de un enfriamiento y predicen una nueva era glacial.

- **Capacitación de tropas:** En el operativo "Tormenta del desierto" llevado a cabo en la guerra contra Irak, las tropas de todas las fuerzas estadounidenses que participaron (fuerza aérea, marina y ejército) fueron entrenadas con simuladores.
- **Capacitación de policías:** Se utiliza entornos virtuales para que el policía aprenda a conducirse en situaciones de riesgo.
- **Simuladores de vuelos:** Fue una de las primeras aplicaciones de los simuladores. Actualmente se utilizan para entrenar pilotos de aviones comerciales y de combate.

El simulador por computadora

Un simulador por computadora está compuesto por las siguientes partes:

- **Un modelo:** Es un modelo simbólico. Puede ser un conjunto de ecuaciones, reglas lógicas o un modelo estadístico.
- **El evaluador:** Es el conjunto de procedimientos que procesarán el modelo para obtener los resultados de la simulación. Puede contener rutinas para la resolución de sistemas de ecuaciones, generadores de números aleatorios, rutinas estadísticas, etc.
- **La interfaz:** Es la parte dedicada a interactuar con el usuario, recibe las acciones del mismo y presenta los resultados de la simulación en una forma adecuada. Esta unidad puede ser tan compleja como la cabina utilizada en los simuladores de vuelos profesionales.

La simulación como proceso experimental

La simulación de sistemas por computadora está basada en una generalización del concepto de experimentación del método científico, según el cual en lugar de realizar los experimentos sobre el sistema real, se realizan sobre un modelo dinámico que lo representa, de manera que si el modelo es una representación válida del sistema entonces los resultados de la experimentación con el modelo pueden transferirse al propio sistema.

El modelo se debe entender como un instrumento de investigación sometido a revisión continua para conseguir un refinamiento progresivo en la comprensión del sistema.



Figura: Esquema del proceso experimental de la simulación

La simulación y los experimentos de simulación son una herramienta de análisis de sistemas que responden a preguntas del tipo "¿que pasaría si...?". La simulación por computadora es por lo tanto una técnica en la que el investigador construye un modelo del sistema y realiza experimentos en una computadora sobre el modelo, interpretando los resultados en términos del comportamiento del sistema objeto del estudio.

Resolución analítica vs. Simulación

Algunos modelos simbólicos pueden resolverse analíticamente. La ventaja de una solución analítica es que da una visión integral sobre la conducta del sistema. Variando sus parámetros es posible identificar fácilmente cambios importantes en el comportamiento, detectar puntos críticos y sacar conclusiones generales para el tipo de sistema analizado. Por ejemplo, El área EXACTA de un círculo de radio "r" puede ser fácilmente calculada por la siguiente ecuación:

$$Area = \pi r^2$$

Sin embargo, no siempre es posible obtener una solución analítica, ya sea por la naturaleza del modelo o de los experimentos que se desean realizar. En este caso, el modelo deberá ser tratado por algún tipo de método numérico. Debido a esto, el análisis de los resultados es más complejo que el requerido por una solución analítica. A continuación se da un ejemplo de cómo obtener un valor APROXIMADO al área real de un círculo de radio "r" utilizando la simulación de MonteCarlo.

Simulación de Monte Carlo

La simulación de Monte Carlo es un método que emplea números aleatorios uniformemente distribuidos en el intervalo [0,1] que es utilizado para resolver problemas donde la evolución con el tiempo no es de importancia. A continuación, se analizará un ejemplo para comparar una solución analítica con una solución obtenida por simulación.

Determinación del área de un círculo

Cuando se desea calcular el área de un círculo de radio $r = 10$ cm no existen mayores problemas, ya que tanto el área a como su perímetro p pueden evaluarse analíticamente con las siguientes fórmulas:

$$a = \pi r^2$$

$$p = 2\pi r$$

En este caso la solución es $a = 314.16$ cm² y $p = 62.83$ cm.

Sin embargo, cuando se desea determinar el área de una forma irregular, por ejemplo la superficie plana de México, el problema debe necesariamente ser resuelto con un método numérico; es decir, simulación.

La determinación del área del círculo utilizando la simulación de Monte Carlo implica la siguiente secuencia:

1. Crear un cuadrado de lado $2.r$ que encierre al círculo.
2. Colocar n puntos al azar dentro del cuadrado.
3. Asignar a c el número de puntos que quedaron dentro del círculo.
4. Como la probabilidad de colocar un punto dentro del círculo es igual al cociente del área del círculo dividida el área del cuadrado, el área del círculo se puede estimar en función del área del cuadrado (fácilmente calculable) con:

$$A_{\text{circulo}} = \frac{c}{n} A_{\text{cuadrado}} = \frac{c}{n} (4r^2)$$

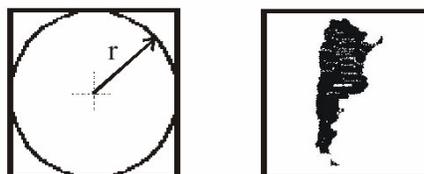


Figura: Determinación del Área de un círculo y Determinación del área de una figura compleja.

Es importante notar que para un determinado n , el resultado será distinto cada vez que se realice la simulación. Es decir, que el resultado será un número aleatorio. A medida que n aumente, la varianza del resultado disminuirá y el valor medio se aproximará a la solución analítica. Para un $n = 100$, el resultado de una simulación es 320 cm^2 ; mientras que para $n = 10000$, un resultado es 313 cm^2 .

El mismo principio se puede aplicar para figuras complejas como se muestra en la Figura 2. Conociendo la escala, se puede fijar un cuadrado arbitrario y calcular el área de México. Sin embargo, la determinación del perímetro de la figura es un problema de mayor magnitud para el cual se necesita recurrir a la teoría de fractales.

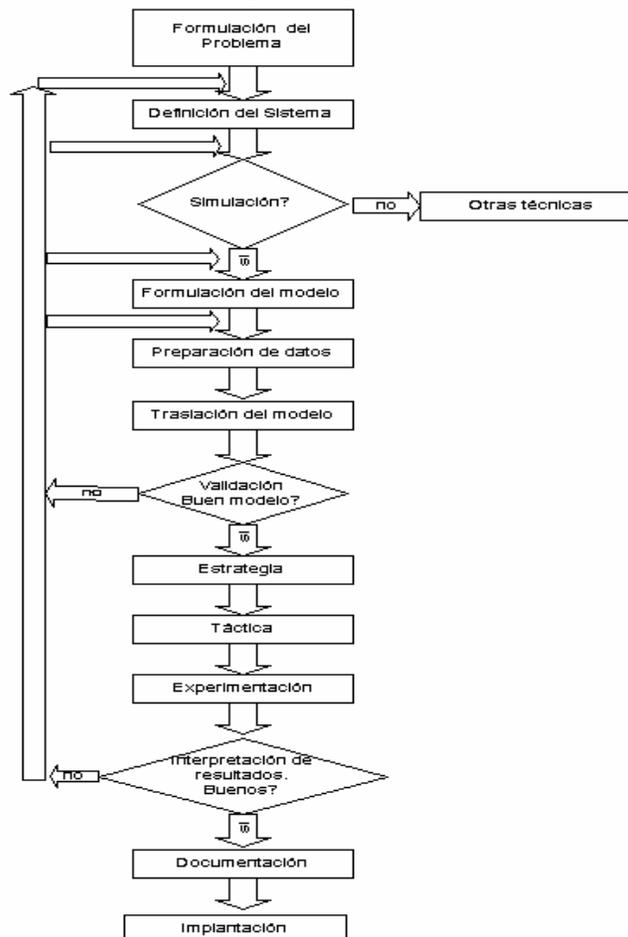
Etapas de una simulación

En el desarrollo de una simulación se pueden distinguir las siguientes etapas:

- **Formulación del problema:** En este paso debe quedar perfectamente establecido el objeto de la simulación. El cliente y el desarrollador deben acordar lo más detalladamente posible los siguientes factores: los resultados que se esperan del simulador, el plan de experimentación, el tiempo disponible, las variables de interés, el tipo de perturbaciones a estudiar, el tratamiento estadístico de los resultados, la complejidad de la interfaz del simulador, etc. Se debe establecer si el simulador será operado por el usuario o si el usuario sólo recibirá los resultados. Finalmente, se debe establecer si el usuario solicita un trabajo de simulación o un trabajo de optimización.
- **Definición del sistema:** El sistema a simular debe estar perfectamente definido. El cliente y el desarrollador deben acordar dónde estará la frontera del sistema a estudiar y las interacciones con el medioambiente que serán consideradas.
- **Formulación del modelo:** Comienza con el desarrollo de un modelo simple que captura los aspectos relevantes del sistema real. Los aspectos relevantes del sistema real dependen de la formulación del problema; para un ingeniero de seguridad los aspectos relevantes de un automóvil son diferentes de los aspectos considerados por un ingeniero mecánico para el mismo sistema. Este modelo simple se irá enriqueciendo como resultado de varias iteraciones.
- **Colección de datos:** La naturaleza y cantidad de datos necesarios están determinadas por la formulación del problema y del modelo. Los datos pueden ser provistos por registros históricos, experimentos de laboratorios o mediciones realizadas en el sistema real. Los mismos deberán ser procesados adecuadamente para darles el formato exigido por el modelo.
- **Implementación del modelo en la computadora:** El modelo es implementado utilizando algún lenguaje de computación. Existen lenguajes específicos de simulación que facilitan esta tarea; también, existen programas que ya cuentan con modelos implementados para casos especiales.
- **Verificación:** En esta etapa se comprueba que no se hayan cometido errores durante la implementación del modelo. Para ello, se utilizan las herramientas de *debugging* provistas por el entorno de programación.
- **Validación:** En esta etapa se comprueba la exactitud del modelo desarrollado. Esto se lleva a cabo comparando las predicciones del modelo con: mediciones realizadas en el

sistema real, datos históricos o datos de sistemas similares. Como resultado de esta etapa puede surgir la necesidad de modificar el modelo o recolectar datos adicionales.

- **Diseño de experimentos:** En esta etapa se decide las características de los experimentos a realizar: el tiempo de arranque, el tiempo de simulación y el número de simulaciones. No se debe incluir aquí la elaboración del conjunto de alternativas a probar para seleccionar la mejor, la elaboración de esta lista y su manejo es tarea de la optimización y no de la simulación. Debe quedar claro cuando se formula el problema si lo que el cliente desea es un estudio de simulación o de optimización.
- **Experimentación:** En esta etapa se realizan las simulaciones de acuerdo el diseño previo. Los resultados obtenidos son debidamente recolectados y procesados.
- **Interpretación:** Se analiza la sensibilidad del modelo con respecto a los parámetros que tienen asociados la mayor incertidumbre. Si es necesario, se deberán recolectar datos adicionales para refinar la estimación de los parámetros críticos.
- **Implementación:** Conviene acompañar al cliente en la etapa de implementación para evitar el mal manejo del simulador o el mal empleo de los resultados del mismo.
- **Documentación:** Incluye la elaboración de la documentación técnica y manuales de uso. La documentación técnica debe contar con una descripción detallada del modelo y de los datos; también, se debe incluir la evolución histórica de las distintas etapas del desarrollo. Esta documentación será de utilidad para el posterior perfeccionamiento del simulador.



Sistemas

Un **sistema** es una sección de la realidad que es el foco primario de un estudio y está compuesto de componentes que interactúan con otros de acuerdo a ciertas reglas dentro de una frontera identificada para el propósito del estudio. Un sistema puede realizar una función que no es realizable por sus componentes individuales. La **frontera del sistema** son sus límites, lo que lo separa del medio ambiente. El **medio ambiente** es todo lo que está fuera del sistema.

Los objetos o componentes que forman parte del sistema se denominan **entidades (Elementos)**, por ejemplo: un auto está compuesto por un motor, ruedas, carrocería, etc. Estas entidades poseen propiedades denominadas **atributos (Propiedades, parámetros)**, que se fijaron durante el diseño del sistema ya sea por el diseñador o por la naturaleza del sistema, y se relacionan entre sí a través de **relaciones o funciones**. Estas relaciones pueden ser:

- **Estáticas o estructurales:** un auto posee cuatro ruedas.
- **Dinámicas o funcionales:** un auto consume combustible si se enciende el motor.

Los valores asumidos por los atributos de las entidades en un momento dado determinan el **estado del sistema**. El estado puede ser:

- **Estático (estacionario).**- Si se mantiene constante en el tiempo
- **Dinámico (transitorio).**- Si evoluciona con el tiempo.

Un sistema puede presentar los dos tipos de conductas en diferentes instantes de tiempo.

Las variables involucradas en un sistema se clasifican en:

- **Variables de entrada:** Pueden ser:
 - **Manipulables (U).**- Las que se fijan a voluntad. Por ejemplo, la posición del pedal de un acelerador
 - **No manipulables (D).**- Cuando dependen del medio ambiente del sistema. Por ejemplo, la velocidad del viento.
- **Variables de salida (Y):** Son las variables que son medidas o traspasan la frontera del sistema.
- **Variables internas:** Son las variables del sistema que no son ni de entrada, ni de salida, ni parámetros, pero se utilizan en procesos internos.

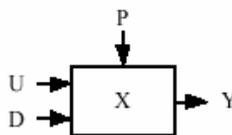


Figura : Variables de un sistema.

Clasificación de sistemas

En función de su aleatoriedad, se clasifican en Determinísticos y Estocásticos:

- **Determinístico:** Si el sistema no contiene ningún elemento aleatorio es un sistema determinístico. En este tipo de sistema, las variables de salida e internas quedan perfectamente determinadas al especificar las variables de entrada, los parámetros y las variables de estado. Es decir, las relaciones funcionales entre las variables del sistema están SIEMPRE perfectamente definidas.

- **Estocástico:** En este caso algún elemento del sistema tiene una conducta aleatoria. Entonces, para entradas conocidas no es posible asegurar los valores de salida. Un ejemplo de sistema estocástico es una máquina de un casino en la cual una misma acción (tirar la palanca) genera un resultado incierto (ganar o perder). Cuando un sistema determinístico es alimentado con entradas estocásticas, la respuesta del sistema es también estocástica. Por ejemplo, la temperatura ambiente es una variable estocástica que afecta la respuesta de un calentador eléctrico. En el mundo real, los sistemas siempre tienen elementos estocásticos ya sea por su propia naturaleza o porque son fenómenos no comprendidos actualmente; por ejemplo, a un cavernícola le podía parecer que las eclipses eran fenómenos aleatorios, hoy se pueden predecir. Sin embargo, se puede considerar a un sistema real como un sistema determinístico si su incertidumbre es menor que un valor aceptado.

En función del tiempo se clasifican en continuos y discretos:

- **Continuo:** Se tiene un sistema continuo cuando las relaciones funcionales entre las variables del sistema sólo permiten que el estado evolucione en el tiempo en forma continua (basta que una variable evolucione continuamente). Matemáticamente, el estado cambia en infinitos puntos de tiempo.
- **Discreto:** Se tiene un sistema discreto cuando las relaciones funcionales del sistema sólo permiten que el estado *varíe* en un conjunto finito (contable) de puntos temporales. Las causas instantáneas de los cambios de estados se denominan *eventos*..

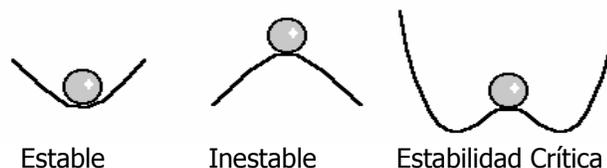
En función de su interacción con el medio ambiente se clasifican en abiertos y cerrados:

- **Abierto:** El sistema interactúa con el medio ambiente y se ve afectado por él.
- **Cerrado:** La conducta del sistema no es función del medio en que esté. Este sistema puede ser trasladado a otro medio ambiente sin que se vea afectado.

En función de su estabilidad, los sistemas se clasifican en estables, inestables o de estabilidad crítica:

- **Estable:** Si al sistema se le aplica una perturbación momentánea (se introduce una "molestia" y luego se elimina) y el sistema luego de cierto tiempo vuelve a su estado original, será estable.
- **Inestable:** Si al sistema se le aplica una perturbación momentánea, y el sistema sufre cambios que lo alejan cada vez más de su estado original, se dice que es inestable.
- **De estabilidad crítica:** Si al sistema se le aplica una pequeña perturbación momentánea y el sistema sufre cambios que hacen que ni vuelva a su estado original, ni que se aleje cada vez más de él, se dice que tiene estabilidad crítica o límite.

Ejemplos:



Un ejemplo de sistema en estado estable, es un péndulo en su posición de reposo; en cambio, el péndulo invertido es un ejemplo de estado naturalmente inestable.

Modelado de sistemas

Modelado es el proceso de construcción de un modelo. Un **modelo** es una representación de un objeto, sistema, o idea, es una abstracción de la realidad que captura la esencia funcional del sistema, con el detalle suficiente como para que pueda utilizarse en la investigación y la experimentación en lugar del sistema real.

Un modelo es un conjunto formado por otros dos conjuntos:

- Un conjunto de variables
- Un conjunto de relaciones entre las variables

Usualmente, el propósito de un modelo es ayudar explicar, entender, describir, predecir o mejorar un sistema. Los modelos son útiles para:

- **El pensamiento:** Al construir un modelo necesariamente se debe ordenar y completar el conocimiento que del sistema real se posee.
- **La comunicación:** Un modelo elimina la ambigüedad del lenguaje para comunicarse con expertos.
- **El entrenamiento y la instrucción:** Un modelo puede ser utilizado para entrenar con costo y riesgo casi nulos. Por ejemplo, los submarinos a escala utilizados por la marina alemana para entrenar en secreto antes de la segunda guerra mundial; o también, el sistema de barcos a escalas utilizados actualmente en Francia para entrenar a los capitanes de barcos petroleros.
- **La predicción:** Un modelo sirve para predecir la conducta del sistema real. Es el caso de los modelos utilizados para predecir, mediante simulación, la evolución del clima mundial. El modelo de la teoría de la relatividad predice, sin hacer una simulación, que no es posible superar la velocidad de la luz.
- **La experimentación:** La experimentación con un modelo es barata y segura. Se emplea frecuentemente en el diseño de un sistema.

El modelado es un arte. Cualquier conjunto de reglas para desarrollar modelos tiene una utilidad limitada y sólo puede servir como una guía sugerida. El arte de modelar consiste en la habilidad para analizar un problema, resumir sus características esenciales, seleccionar y modificar las suposiciones básicas que caracterizan al sistema, y luego enriquecer y elaborar el modelo hasta obtener una aproximación útil.

Pasos sugeridos para realizar un modelo

1. Establecer una definición clara de los objetivos.
2. Analizar el sistema real.
3. Dividir el problema del sistema en problemas simples.
4. Buscar analogías.
5. Determinar las variables de interés.
6. Escribir los datos obvios.
7. Escribir las ecuaciones teóricas o empíricas que describen los fenómenos presentes y relacionan las variables de interés.
8. Si se tiene un modelo manejable, enriquecerlo. De otra manera, simplificarlo.

Simplificar un modelo implica:

- Convertir variables en constantes.
- Eliminar o combinar variables.
- Agregar suposiciones más potentes y restricciones.
- Restringir los límites del sistema.

Para enriquecerlo se procede de la forma contraria. Durante el proceso de modelado se debe alcanzar un equilibrio entre el grado de detalle y el riesgo de falta de exactitud. El mejor modelo, es el modelo más simple que puede resolver el problema con el grado de exactitud requerido.

Características de un buen modelo

- Fácil de entender por parte del usuario.
- Dirigido a metas u objetivos.
- Sensato, en cuanto no de respuestas absurdas.
- Fácil de manipular y controlar por parte del usuario. Es decir, debe ser sencillo comunicarse con el modelo.
- Completo, en lo referente a asuntos importantes.
- Adaptable, con un sencillo procedimiento para modificar o actualizar el modelo.
- Evolutivo, debe ser sencillo al principio y volverse más complejo en el tiempo.

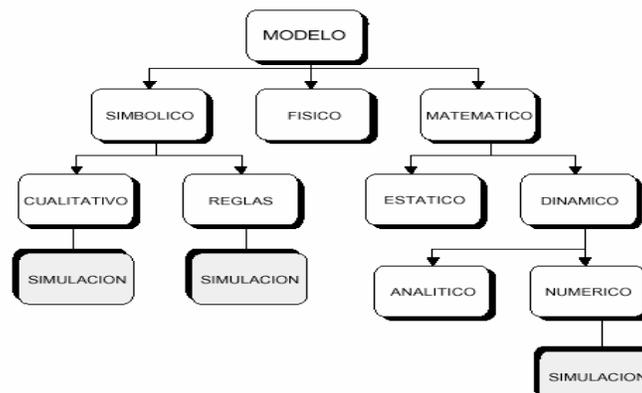
Clasificación de modelos

Como ya fue planteado, un modelo es también un sistema; por lo tanto valen todas las definiciones y clasificaciones que se establecieron anteriormente. Debido a que un modelo es una simplificación o abstracción de un sistema real, no es necesario, salvo en los aspectos relevantes, que el modelo guarde una total correspondencia con el sistema real.

Es posible realizar distintos tipos de clasificaciones de modelos. Por ejemplo según la naturaleza del sistema o del uso que se vaya a dar al modelo. Un ejemplo de clasificación de modelos es:

- **Físico.** En algunos casos se puede construir un sistema físico cuyo comportamiento represente el del sistema en estudio, por ejemplo un modelo a escala de un barco.
- **Mental.** Para mantener un vaso en equilibrio, el cerebro no precisa conocer la formulación matemática de la ley de la gravitación universal, sino que a partir de una cierta formulación intuitiva se sirve de ella para efectuar el control del sistema.
- **Gráfico.** Del mismo modo pueden ser útiles gráficos que representen el comportamiento del sistema ante distintas situaciones.
- **Matemático.** En muchas situaciones, el comportamiento de los sistemas permite hacer uso de las leyes físicas, químicas, etc. que los gobiernan, con las que se puede elaborar el modelo del sistema preciso.
- **Computarizado.** Utilizando un lenguaje o paquete computacional ya sea de Propósito General o específico de Simulación.

Existe una gran cantidad de técnicas de modelado, y, por ello es posible construir una gran cantidad de modelos para un sistema dado. En la siguiente figura se muestra un diagrama con algunas de las posibilidades de modelado más comunes:



Los modelos también se pueden clasificar en:

- **Experimentales (De caja negra).**- Si el modelo que se construye sólo se orienta a reproducir las salidas del sistema real sin intentar modelar su comportamiento interno; entonces, será un modelo experimental o de caja negra.
- **Modelos de base teórica.**- Si el modelo también intenta reproducir las relaciones funcionales del sistema será un modelo con base teórica.

Un modelo de caja negra requiere una gran cantidad de datos para poder calibrarlo o ajustarlo correctamente, y su rango de validez está limitado a este conjunto de datos. En contraposición, un modelo teórico requiere una cantidad menor de datos y puede ser utilizado fuera del rango de los mismos ya que el rango de validez del modelo está dado por la teoría utilizada y no por los datos.

Un ejemplo simple se presenta cuando se desea determinar el espacio recorrido por un cuerpo móvil a velocidad constante en función del tiempo. Para una velocidad dada, se puede construir una gráfica de espacio vs. tiempo a partir de medidas experimentales. Esta gráfica sólo podrá emplearse para el caso en que la velocidad del móvil sea idéntica a aquella que se utilizó en los experimentos. Además, sólo proveerá información para los tiempos que pertenezcan al intervalo de experimentación. Estos problemas no se presentan si se utiliza un modelo teórico simple: $e = v \cdot t$; ni siquiera son necesarios los datos experimentales.

La computadora en la simulación de modelos de sistemas

La masiva utilización de la informática en la enseñanza y en el entorno industrial , la sorprendente y revolucionaria evolución de los ordenadores personales en cuanto a tamaño, coste, velocidad, software, etc. han ayudado sin lugar a dudas a que la simulación por computadora sea hoy en día la herramienta más utilizada para realizar experimentos de simulación de sistemas.

Un ***programa de simulación de computadora*** se puede definir como una secuencia de instrucciones que el usuario define para resolver un problema que puede estar plasmado en unas ecuaciones que describen a un sistema previamente modelizado mediante dichas ecuaciones.

La construcción de un modelo de simulación ha pasado, de ser una labor reservada a especialistas en programación, de difícil y costosa realización, basada en procesos batch y en una interpretación de tediosos listados, a ser un ejercicio estructurado alrededor de la utilización de entornos cada vez mas amables y flexibles que permiten aprovechar la característica mas destacable de la simulación: la posibilidad de estudiar la evolución dinámica de los sistemas a lo largo del tiempo.

Hoy en día existe un amplio abanico de posibilidades para resolver estos problemas con los distintos lenguajes que podemos utilizar para traducir nuestros modelos en una computadora y posteriormente resolverlos para obtener la simulación del comportamiento del sistema modelado. Podemos utilizar lenguajes de programación general, lenguajes específicos para simulación o paquetes de software de simulación especialmente preparados para la misma.

A la hora de elegir una herramienta u otra hay que tener en cuenta primeramente la velocidad de ejecución de los programas y la utilización de recursos necesaria (memoria, procesadores, etc.).

Lenguajes de programación

Hay distintos niveles de lenguajes, en el más bajo nivel se encuentra el lenguaje máquina cuyas instrucciones se escriben en notación binaria. Este lenguaje es sin duda el más tedioso y menos práctico de utilizar. En un nivel superior se encuentran el lenguaje ensamblador que utiliza instrucciones mnemónicas para representar dichas funciones.

Los lenguajes de alto nivel o lenguajes de propósito general tales como C, Fortran, Basic, Cobol, Lisp, Algol, Pascal, etc. normalmente alejan al programador de las tareas de bajo nivel del computador y suelen ir apoyados en un conjunto de librerías que en el caso de la simulación facilitan mucho la tarea de modelizar los sistemas y reducen el tiempo de ejecución del programa.

Lenguajes de simulación

Los lenguajes de simulación son similares a los lenguajes de programación de alto nivel pero están especialmente preparados para determinadas aplicaciones de la simulación. Suelen venir acompañados de una metodología de programación apoyada por un sistema de símbolos propios para la descripción del modelo por ejemplo mediante diagramas u otras herramientas que simplifican notablemente la modelización y facilitan la posterior depuración del modelo. Entre estos lenguajes específicos podemos nombrar los siguientes: MIDAS, DYSAC, DSL, GASP, MIMIC, DYNAMO, GPSS, SIMULA, CSSL, CSMP, ACSL, DARE-P, C-Simscript, SLAM, SIMAN, SIMNON, SIMSCRIPT-II-5, ADA, GASP IV, SDL. Muchos de estos lenguajes dependen de los lenguajes de propósito general como es el caso de Slam o Siman que dependen de Fortran para las subrutinas.

Ventajas de usar un lenguaje de simulación:

- Menor esfuerzo requerido para programar el modelo.
- Los bloques básicos de construcción del lenguaje son mucho más simples de manipular.
- Facilitan la detección de errores, generación de escenarios, manipulación del modelo, etc.
- El usuario sólo precisa conocer detalles de programación relativos al producto.

Desventajas de usar Lenguajes de Simulación:

- Un lenguaje de Simulación no permite desarrollar proyectos más complejos, es decir, está limitado a realizar algunas aplicaciones solamente.

Ventajas de usar un lenguaje de propósito general:

- Un lenguaje de propósito general apoyado con librerías, permite afrontar problemas de la máxima complejidad y tamaño pudiendo ser el programa altamente portable.

Desventajas de usar Lenguajes de propósito general:

- Se requiere conocer de programación, así como saber Interpretar los mensajes de error del entorno de desarrollo.

En los años sesenta se realizaban estudios de simulación cuyos costos se medían en años-hombre y su duración en meses. En los setenta aparecieron diversos lenguajes específicamente orientados a la simulación tales como SIMSCRIPT, etc. La década de los ochenta supuso la adaptación sobre PC de productos ya existentes y la aparición de nuevos productos como SIMAN.

Los noventa han protagonizado hasta ahora una explosión de nuevos productos de manejo más intuitivo bajo entornos gráficos como Windows. Es el caso de Simfactory o Simvox, por ejemplo.

La evolución de los ordenadores y del software comercial se dirige hacia sistemas que puedan ser manejados por personas no-especialistas, con máquinas cada vez más potentes a menor coste. Las técnicas orientadas al objeto conducen a programas de utilización más intuitiva. Todo ello nos sugiere un incremento considerable de la aplicación de las técnicas de simulación.

Sin embargo, a pesar de todo, se estima que en el mercado norteamericano, que es el más desarrollado, sólo se tiene en cuenta la aplicación de técnicas de simulación en un 30% de los casos en los que podría aplicarse, y de este porcentaje, sólo en el 10% de los casos se utiliza regularmente. En Europa las cifras son menores, situándose en torno al 3%, a excepción de Inglaterra donde dicho porcentaje se eleva al 15%.

UNIDAD II

SIMULACION DE SISTEMAS POR COMPUTADORA EN LENGUAJES DE PROGRAMACION

Tipos de sistemas a Simular.-

- **Determinísticos.-** Se dice que un sistema es determinístico si su respuesta ante una acción exterior está determinada siempre de manera unívoca. Un sistema determinístico tiene un comportamiento predecible exactamente, ya que se conocen sus parámetros y el comportamiento de sus elementos.
- **Estocásticos.-** Un sistema estocástico es aquel cuyos parámetros que lo constituyen poseen una variabilidad significativa, comportándose aleatoriamente. En este caso, dichos parámetros o variables aleatorias se modelan en términos probabilísticos y se asumen ciertas condiciones sobre las funciones de probabilidad que describen su comportamiento.

Dependiendo de la manera como se aborde el proceso de construcción, el modelo de un sistema determinístico puede ser determinístico o estocástico.

Modelos determinísticos.-

Un modelo determinístico queda reducido a una ecuación o un conjunto de ecuaciones que describen el comportamiento del sistema de manera precisa.

Generalmente los modelos determinísticos contienen ecuaciones diferenciales, y se utilizan para describir sistemas que cambian a través del tiempo. Sin embargo, no es necesario que un modelo contenga ecuaciones diferenciales para considerarlo determinístico.

La simulación de modelos determinísticos puede resultar trivial para muchos estudiosos de la simulación, llegando incluso a denominarlos "evaluadores" en lugar de "simuladores", sin embargo, su utilidad es muy reconocida por las personas involucradas en el estudio de sistemas físicos como trayectorias, movimientos, etc.

Modelos estocásticos.-

La simulación de modelos estocásticos se apoya en la generación de números aleatorios. Cada número generado sirve para simular un evento y tomar una decisión.

Un número aleatorio entre 0 y 1 debe ser:

- Uniformemente distribuido (con la misma probabilidad de que todos los números sean elegidos)
- Estadísticamente independiente (que un número no tenga relación con otro)
- Reproducible (que se pueda volver a generar).

Los números aleatorios pueden ser generados por computadora mediante sencillos algoritmos. Incluso la mayoría de los lenguajes de programación poseen funciones integradas para proporcionarlos con una simple llamada.

Cuando los números aleatorios se generan con un algoritmo, muchos autores los denominan pseudoaleatorios por ser producto de una regla puramente determinística. Sin embargo, esta objeción puede ser superada si los números satisfacen ciertas pruebas de aleatoriedad.

NOTAS IMPORTANTES ACERCA DE LOS PROYECTOS DE SIMULACION POR COMPUTADORA EN LENGUAJES DE PROGRAMACION

- Se Pueden simular sistemas determinísticos y estocásticos utilizando lenguajes de programación.
- La simulación computarizada de un sistema se lleva a cabo primero entendiendo el sistema a simular, luego realizando un modelo y posteriormente, implementándolo en un lenguaje de programación.
- Una vez escrito el programa en un lenguaje de programación, se llevan a cabo experimentos, mediante los cuales se conocen los resultados.
- Los "Experimentos de simulación" se refieren a las ejecuciones (corridas) que se realizan con el programa en la computadora.
- El proceso de simulación estocástica arroja estimaciones y resultados aproximados a la realidad.
- Los resultados del experimento de simulación y los resultados del sistema real tienden a ser muy cercanos conforme el número de experimentos de simulación sea mayor (Es decir, a mayor N, menor Error).
- Para simular un sistema estocástico se utilizan números aleatorios.
- Un número aleatorio generado por computadora debe ser:
 - Uniformemente distribuido (con la misma probabilidad de que todos los números sean elegidos) .
 - Estadísticamente independiente (que un numero no tenga relación con otro).
 - Reproducible (que se pueda volver a generar).
- Un número aleatorio generado, corresponde al resultado de algún evento o suceso.
- En **C#.NET** Una variable tipo "Random" permite generar números aleatorios. Para declararla (una sola vez en el programa) se hace lo siguiente:

```
System.Random r= new Random();
```
- Para generar un número aleatorio entre 0 y 0.99999 (Aproximadamente 1), se hace lo siguiente:

```
double numero = r.NextDouble();
```
- Se puede generar un num aleatorio entero entre dos rangos diferentes de 0 y 1 utilizando la formula
$$\text{NumAleatorio} = \text{int}((\text{RangoSuperior} - \text{RangoInferior} + 1) * \text{numero} + \text{RangoInferior})$$

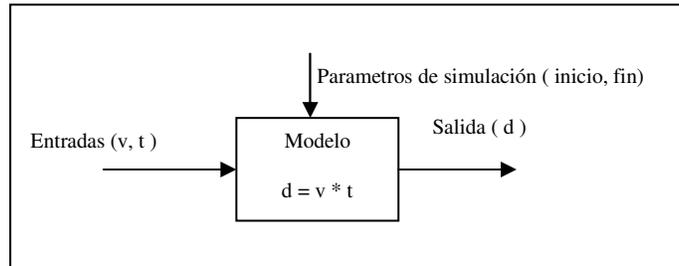
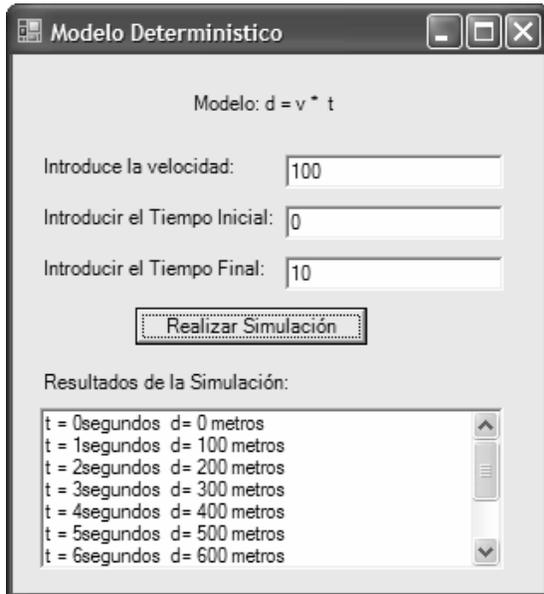
C#.NET simplifica esta fórmula con la siguiente instrucción (siempre y cuando se haya declarado la variable "r" como se indicó anteriormente):

```
int NumAleatorio = r.Next(RangoInferior, RangoSuperior+1);
```
- Se debe seguir la metodología de un proyecto de simulación para todos los proyectos a realizar, y ser muy cuidadoso al momento de realizar el modelo y la lógica del programa.
- También es muy importante saber interpretar los resultados de los experimentos de simulación. Es posible que una serie de experimentos bien realizados arroje resultados correctos pero mal interpretados que conduzcan a tomar decisiones equivocadas.

PRACTICAS DE LA MATERIA DE SIMULACION

- Pueden hacerse individualmente o en equipos de dos personas.
- Deben revisarse el avance en el laboratorio por el maestro y anotadas en la lista para tomarse en cuenta para la calificación.
- Posterior a la realización de cada práctica se debe entregar al maestro un reporte impreso (O en medios electrónicos) que contenga:
 - Portada.
 - Descripción de la práctica (Brevemente en que consistió).
 - Teoría (Conceptos, métodos, importantes para la realización de la práctica).
 - Imagen del programa ejecutándose.
 - Diagrama de flujo.
 - Listado del programa.
- Los reportes pueden entregarse cualquier día después de la realización de la práctica, pero siempre antes de presentar el examen parcial al que pertenecen.
- Las prácticas pueden realizarse utilizando cualquier lenguaje de programación (Preferentemente C#.NET).

Proyecto numero 1.- Modelo determinístico



▪ **Completar la línea en blanco:**

Sistema a modelar _____

Estado (respuesta) del sistema (Estático / Dinámico) _____

Variables del modelo

Variables de Entrada _____

Variables de salida _____

Variables internas _____

Tipo de modelo

(Determinístico/Estocástico) _____

(Caja negra / Base teórica) _____

(Físico/Mental/Gráfico/Matemático/Computarizado) _____

Elementos del simulador

Modelo _____

Evaluador _____

Interfaz _____

Herramienta usada para simulación

(Lenguaje de programación/ Paquete) _____

(Uso General / Específico de simulación) _____

Etapas de la simulación _____

▪ **Realizar Diagrama de Flujo y Programa**

PROYECTO DE SIMULACION NUMERO 2: MODELO ESTOCASTICO ESTIMACION DE LAS PROBABILIDADES EN LANZAMIENTOS DE MONEDAS

Introducción

De acuerdo a la teoría de la probabilidad, sabemos que en el lanzamiento de una moneda sin cargar (es decir, con la misma probabilidad de que caiga un lado u otro) existe una probabilidad del 50% de que caiga cara y una probabilidad de 50% de que caiga cruz (.5 y .5).

Este resultado se obtiene dividiendo $1 / 2$ (Un lado que nos interesa obtener entre el total de lados que es "2").

Si quisiéramos verificar esta probabilidad con 10 lanzamientos, esperaríamos que el resultado de aproximadamente 5 lanzamientos sean cara y 5 sean cruz. Sin embargo, 10 es un número pequeño de lanzamientos para verificarlo con poco margen de error; es decir, en 10 lanzamientos tal vez obtengamos 5 caras y 5 cruces, o bien 8 caras y 2 cruces, o 9 caras y una cruz, o cualquier otra combinación.

Sin embargo, conforme aumente considerablemente el número de lanzamientos, se esperará obtener cara en la mitad de ellos y cruz en la otra mitad. Esto significa que la probabilidad de obtener cara o cruz será muy cercana a .5 conforme el número de lanzamientos se incremente.

"A mayor número de iteraciones realizadas en un experimento con un modelo estocástico, el error de estimación al usar simulaciones tiende a ser cero".

Descripción del programa

Realizar un programa que permita simular "N" lanzamientos de una moneda y determinar la probabilidad de obtener cara o cruz en ese experimento de simulación.

El número N será introducido por el usuario. Cada lanzamiento equivale a generar un número aleatorio entre 0 y 1. Si el número está entre 0 y 0.5 considerar "Cara" como resultado, de lo contrario, considerar "Cruz".

Probabilidad de obtener cara = Numero de veces que se obtuvo "cara" / N

Probabilidad de obtener cruz = Numero de veces que se obtuvo "cruz" / N

Realizar pruebas con el programa y anotar los resultados obtenidos para N = 10, 100, 1000, 10000, 100000 lanzamientos.

Resultados de la Simulación	
Numero de Veces que cayó CARA:	5113
Numero de Veces que cayó CRUZ:	4887
Probabilidad de que caiga CARA:	0.5113
Probabilidad de que caiga CRUZ:	0.4887

Procedimiento sugerido:

1. Obtener el Numero de lanzamientos N
2. Inicializar semilla aleatoria y contadores de cara y cruz en cero
3. Hacer un ciclo de 1 a N con los pasos 3a y 3b
 - a. Generar un número aleatorio.
 - b. Si el número aleatorio es mayor o igual a 0.5, considerar CARA (Sumar uno al contador de Cara); de lo contrario considerar CRUZ (Sumar uno al contador de Cruz).
4. Calcular la probabilidad de obtener cara y la probabilidad de obtener cruz.
5. Imprimir resultados

PROYECTO DE SIMULACION NUMERO 3: SIMULACION DE UN JUEGO DE AZAR CON REGLAS PRE-ESTABLECIDAS (APUESTAS Y VOLADOS)

Se Desea simular un juego de azar que consiste en lanzar una moneda y apostar dinero en cada lanzamiento. Si se gana el volado, se gana la cantidad apostada, pero si se pierde el volado, se paga esa misma cantidad (de la apuesta).

Antes de hacer el primer lanzamiento, se determina:

- *Una cantidad inicial* (con la que se comienza el juego).
- *Una cantidad para apostar* (Si se gana el volado, esta cantidad se suma a lo que se tiene, pero si se pierde el volado, esta cantidad se resta).
- *La cantidad que se desea tener* (Si se llega a esta cantidad, se deja de jugar).

El juego termina cuando se llega a la cantidad deseada, en cuyo caso, se considera "triunfo"; sin embargo, el juego también puede terminar cuando el jugador pierde todo su dinero (Es decir, llega a ceros), en este caso se considera "Bancarrota".

Realizar un programa que permita simular este juego. Utilizar números aleatorios para determinar el evento del lanzamiento. Si el número generado es menor que .5 se gana el volado, de lo contrario se pierde.

Mostrar la siguiente información en cada lanzamiento hasta que el juego termine: El número aleatorio generado, Cuanto dinero se tiene y si ganó o perdió en cada lanzamiento. Al final del programa especificar el numero de lanzamientos realizados, y si se llegó a la meta ("Triunfo"), o bien si quedó en ceros ("Bancarrota").



PROYECTO DE SIMULACION NUMERO 4 USO DE NÚMEROS ALEATORIOS PARA ESTIMAR LA PROBABILIDAD DE GANAR EL LUCKY 7 POR PRIMERA VEZ

El Lucky 7 es uno de los juegos clásicos que se encuentran en los casinos. Consiste en una máquina con tres rodillos con números impresos del 0 al 9 (cada uno), que giran de manera aleatoria e independiente cuando una palanca es activada por el jugador. Al detenerse, cada rodillo muestra un número por una ventanita. El jugador gana cuando los 3 rodillos muestran el número "7" por su respectiva ventanita.

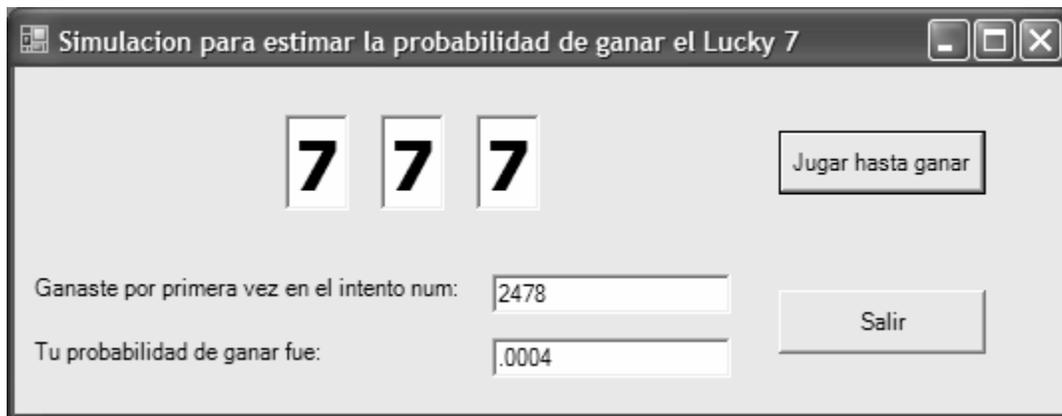
Suponga que cada número tiene la misma probabilidad de aparecer por la ventanita al accionar la palanca.

Realice un programa que muestre cuantos intentos son necesarios para ganar por primera vez un juego en el lucky 7. Use tres números aleatorios independientes (Entre 0 y 9) para simular cada rodillo. Calcular también la probabilidad de ganar en cada simulación: $1 / \text{numero de intentos}$.

Números aleatorios en el rango de 0 a 9.- Para las ocasiones en que se necesita generar números aleatorios en un rango distinto al de 0 a 1, como en este caso, que se requiere números entre 0 y 9, en C# se puede hacer lo siguiente:

```
System.Random r = new System.Random(); //Se Declara variable tipo Random  
int NA = r.Next(0,10); //Genera un num aleatorio entre 0 y 9
```

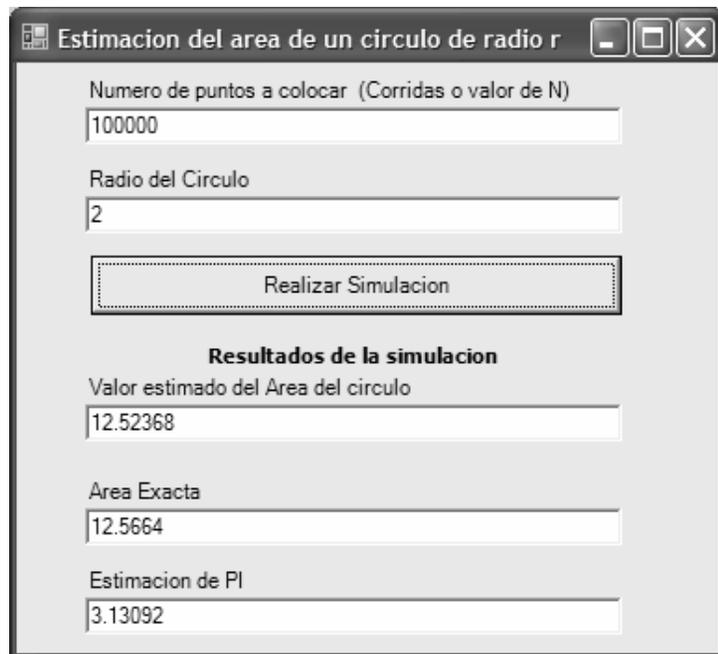
↑ ↑
RangoInferior RangoSuperior + 1



El algoritmo para hacer el programa sería:

1. Inicializar el Número de Intentos en cero.
2. Hacer un ciclo hasta que $X=7$, $Y = 7$ y $Z=7$
 - a. Generar 3 números aleatorios diferentes X,Y,Z cada uno entre 0 y 9.
 - b. Incrementar 1 al Numero de Intentos
 - c. Imprimir los números en los 3 cuadros de texto
 - d. "Refrescar" los 3 cuadros de Texto de los números aleatorios (Solo para efecto Visual). Esto puede hacerse con el método REFRESH Ej.: `txtX.Refresh();`
3. Cerrar el ciclo (Regresar al paso 2)
4. Calcular la Probabilidad de Ganar = $1 / \text{Numero de Intentos}$
5. Imprimir el Número de Intentos y la Probabilidad de Ganar en los cuadros de texto correspondientes.

PROYECTO DE SIMULACIÓN NUMERO 5
ESTIMACIÓN DEL AREA DE UN CIRCULO DE RADIO "R" MEDIANTE EL METODO DE MONTE CARLO

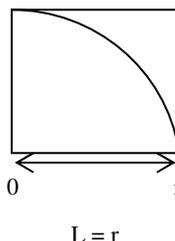


La elaboración del programa consiste en:

1. Pedir dos valores: **N** (Numero de puntos a colocar en el cuadrado) y **R** (Radio del Circulo).
2. Inicializar "C" en 0
3. Hacer un ciclo de 1 a N.
4. Generar dos coordenadas aleatorias X,Y dentro del cuadrado (Es decir, dos números aleatorios entre 0 y r).
5. Si se cumple la condición $RAIZCUADRADA(x^2 + y^2) \leq R$, el punto generado cae dentro del círculo; por lo que hay que incrementarle uno a "C". Si la condición no se cumple, no se hace nada.
6. Completar el ciclo (Regresar al paso 3).
7. Calcular el Area Estimada = $(c/n) * (4 * r^2)$
8. Calcular Pi Estimada $(c/n) * 4$
9. Imprimir Area Estimada , Pi Estimada.
10. Fin del proceso

El procedimiento matemático consiste en:

1. Dividir al círculo en 4 partes y tomar solo una de ellas para calcular su área. Al final multiplicar este resultado por 4.
2. Generar "n" puntos (coordenadas) X,Y aleatoriamente dentro del **cuadrado**. (Números aleatorios entre 0 y r).
3. Determinar el número "c" de puntos (X,Y) que quedaron dentro del **semicírculo**, mediante la siguiente evaluación:



Si $\sqrt{x^2 + y^2} \leq r$, la coordenada (X,Y) está dentro del círculo

$$\frac{A_{semicirc}}{A_{cuadrado}} = \frac{c}{n}$$

$$A_{semicirc} = \left(\frac{c}{n}\right)(A_{cuadrado})$$

$$A_{semicirc} = \left(\frac{c}{n}\right)(L^2) = \left(\frac{c}{n}\right)(r^2)$$

Entonces el área completa del círculo es cuatro veces el área del semicírculo:

$$AreaEstimadaDelCirculo = \left(\frac{c}{n}\right)(4r^2)$$

Análíticamente, El área del círculo es:

$$AreaExactaDelCirculo = \pi r^2$$

Como el área estimada es muy cercana al área real del círculo (con un margen de error pequeño si el número "n" de puntos generados es grande),

$$AreaEstimadaDelCirculo \approx AreaExactaDelCirculo$$

entonces, una buena estimación del número PI sería:

$$\left(\frac{c}{n}\right)(4) \approx \pi$$

COMO OBTENER LA RAIZ CUADRADA DE UN NUMERO EN C#.NET

`raizCuadrada = System.Math.Sqrt (Numero);`

REFLEXIONES IMPORTANTES SOBRE LA SIMULACION DE MONTE CARLO

- **Elemento Clave: ALEATORIEDAD:** Se asume que algunas entradas del modelo son variables aleatorias. Se simulan las entradas al modelo generando variables aleatorias provenientes de sus distribuciones de probabilidad. Como las entradas del modelo son aleatorias, las salidas del modelo son aleatorias también.
- **Proceso de Simulación de Monte Carlo.-** Se desarrolla un modelo básico que se “comporta” como el sistema o problema real, con una consideración especial de azar o variables de entrada aleatorias. Se realiza una serie de ejecuciones en la computadora (Denominadas *corridas*) con la finalidad de aprender acerca de la conducta del modelo simulado. Se Calcula la estimación, y estadísticas de salida y se hacen inferencias acerca del problema real. *A mayor número de experimentos (a mayor tamaño de la muestra), mayor será el grado de confiabilidad del resultado.*
- **Variable Aleatoria:** Es una variable que tiene muchos posibles valores, pero no es posible saber con exactitud cual de todos tomará. Las variables aleatorias pueden ser discretas o continuas.
- **Variable Aleatoria Discreta:** Una variable aleatoria Discreta se caracteriza por tener como función de probabilidad una lista de posibles valores y sus probabilidades asociadas.

Ejemplo: Número de carros vendidos en una agencia.

Variable Discreta (<u>Num carros vendidos</u>)	Probabilidad (<u>Frec. Relativa</u>)	Prob. Acumulada (<u>Frec. Rel. Acum.</u>)
0	0.10	0.10
1	0.20	0.30
2	0.30	0.60
3	0.20	0.80
4	0.10	0.90
5	0.10	1.00
	1.00	

- **Variable Aleatoria Continua:** Se caracteriza por tener una función de probabilidad representada por el área bajo la curva de una función (ecuación) $f(x)$
- **Distribución de Probabilidad:** Mecanismo matemático usado para caracterizar (representar) el patrón o conducta probabilística de una variable aleatoria. Ejemplos de distribuciones de Probabilidad Discreta: Distribución binomial, de Poisson. Ejemplos de distribuciones de Probabilidad Continua: Distribución Uniforme, Normal, Triangular, exponencial.

SIMULACION DE MONTECARLO DE VARIABLES DISCRETAS

La simulación de Monte Carlo es una técnica cuantitativa que combina conceptos estadísticos (muestreo aleatorio) con la capacidad que tienen las computadoras para generar números pseudo-aleatorios y automatizar cálculos con la finalidad de imitar, mediante modelos matemáticos, el comportamiento aleatorio de sistemas reales no dinámicos (por lo general, cuando se trata de sistemas cuyo estado va cambiando con el paso del tiempo, se recurre a otros tipos de simulación).

Los orígenes de esta técnica están ligados al trabajo desarrollado por Stan Ulam y John Von Neumann a finales de los 40 en el laboratorio de Los Alamos, cuando investigaban el movimiento aleatorio de los neutrones. En años posteriores, la simulación de Monte Carlo se ha venido aplicando a una infinidad de ámbitos como alternativa a los modelos matemáticos exactos o incluso como único medio de estimar soluciones para problemas complejos. Así, en la actualidad es posible encontrar modelos que hacen uso de simulación Monte Carlo en las áreas informática, empresarial, económica, industrial e incluso social. En otras palabras, la simulación de Monte Carlo está presente en todos aquellos ámbitos en los que el comportamiento aleatorio o probabilística desempeña un papel fundamental -precisamente, el nombre de Monte Carlo proviene de la famosa ciudad de Mónaco, donde abundan los casinos de juego y donde el azar, la probabilidad y el comportamiento aleatorio conforman todo un estilo de vida.

La simulación MonteCarlo consiste en crear un modelo matemático del sistema, proceso o actividad que se quiere analizar, identificando las variables (*inputs* del modelo) cuyo comportamiento aleatorio determina el comportamiento global del sistema. Una vez identificadas, se lleva a cabo un experimento consistente en: (1) **generar** –en la computadora- **muestras aleatorias** (valores concretos) para dichos *inputs*, y (2) analizar el comportamiento del sistema ante los valores generados. Tras repetir n veces este experimento, se tendrán n observaciones sobre el comportamiento del sistema, lo cual será de utilidad para entender el funcionamiento del mismo –obviamente, el análisis será más preciso cuanto mayor sea el número n de experimentos llevados a cabo. Ejemplo:

En la tabla se muestra un análisis histórico de 200 días sobre el número de consultas diarias realizadas a una página WEB residente en un servidor. La tabla incluye el número de consultas diarias (0 a 5) junto con las frecuencias absolutas (número de días que se producen 0, 1,..., 5 consultas), las frecuencias relativas (10/200 = 0,05,...), y las frecuencias relativas acumuladas.

Numero de consultas	Frecuencia Absoluta (Numero de Días)	Frecuencia Relativa (Num Días / Total)	Frecuencia Relativa acumulada
0	10	0.05	0.05
1	20	0.10	0.15
2	40	0.20	0.35
3	60	0.30	0.65
4	40	0.20	0.85
5	30	0.15	1.00
TOTAL:	200	1.00	

Se puede interpretar la **frecuencia relativa** como *la probabilidad de que ocurra el suceso asociado* (en este caso se refiere a la probabilidad de que se realicen cierto número de consultas) (por ej. la probabilidad de que se den 3 consultas en un día es de 0.30), por lo que la tabla anterior proporciona la distribución de probabilidad asociada a una variable aleatoria discreta (la variable aleatoria es el número de consultas a la página WEB, que sólo puede tomar valores enteros entre 0 y 5).

Supóngase que se desea conocer el número esperado (o medio) de consultas por día. La respuesta a esta pregunta es fácil al recurrir a la teoría de la probabilidad: Denotando por X a la variable aleatoria que representa el número diario de consultas a la pagina, se sabe que:

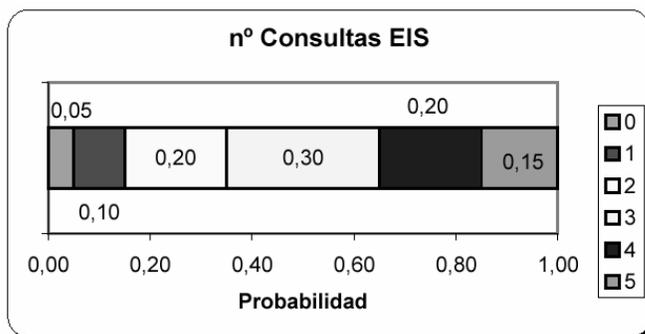
$$E[X] = \sum_{i=0}^5 x_i \cdot P(X = x_i) = 0 \cdot 0,05 + 1 \cdot 0,10 + \dots + 5 \cdot 0,15 = 2,95$$

Por otra parte, también se puede usar la simulación de Monte Carlo para estimar el número esperado de consultas diarias (en este caso se ha podido obtener el valor exacto usando teoría de probabilidad, pero ello no siempre será factible).

Cuando se conozca la distribución de probabilidad asociada a una variable aleatoria discreta, será posible usar la columna de frecuencias relativas acumuladas para obtener los llamados **intervalos de números aleatorios** asociados a cada suceso. En este caso, los intervalos obtenidos son:

- [0,00 , 0,05) para el suceso 0 (0 consultas)
- [0,05 , 0,15) para el suceso 1 (1 consulta)
- [0,15 , 0,35) para el suceso 2 (2 consultas)
- [0,35 , 0,65) para el suceso 3 (3 consultas)
- [0,65 , 0,85) para el suceso 4 (4 consultas)
- [0,85 , 1,00) para el suceso 5 (5 consultas)

El gráfico siguiente muestra cada una de las probabilidades sobre el número de consultas. En él, se aprecia claramente la relación existente entre probabilidad de cada suceso y el área que éste ocupa.



Esto significa que, al generar un número pseudo-aleatorio con el ordenador (proveniente de una distribución uniforme entre 0 y 1), estaremos llevando a cabo un experimento cuyo resultado, obtenido de forma aleatoria y según la distribución de probabilidad anterior, estará asociado a un suceso. Así por ejemplo, si el ordenador nos proporciona el número pseudo-aleatorio 0.2567, podremos suponer que ese día se han producido 2 consultas al servidor.

**PROYECTO DE SIMULACION NUMERO 6
SIMULACION DE MONTECARLO DE VARIABLES DISCRETAS
"ESTIMACION DEL NUMERO DE CONSULTAS A UN SERVIDOR WEB"**

Una empresa tiene un servidor WEB, y está analizando la posibilidad de incrementar su ancho de banda para ofrecer mayor velocidad de acceso.

Se tiene disponible un análisis histórico de 200 días sobre el número de consultas diarias realizadas a ese servidor, y se muestra a continuación:

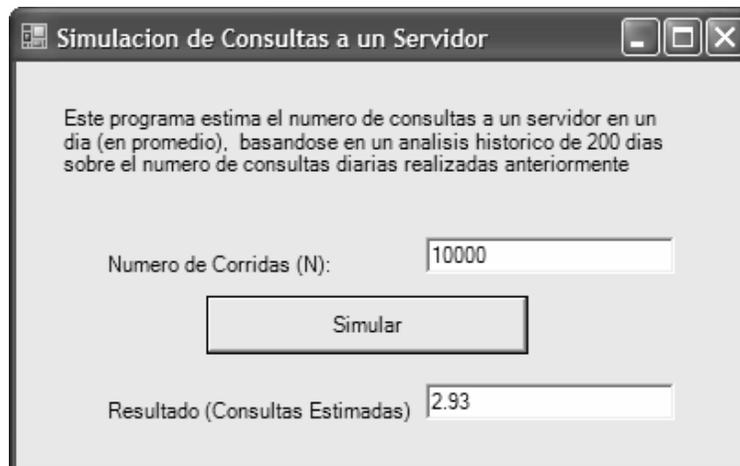
Numero de consultas	Frecuencia Absoluta (Numero de Dias)	Frecuencia Relativa (Num Dias / Total)	Frecuencia Relativa acumulada
0	10		
1	20		
2	40		
3	60		
4	40		
5	30		
TOTAL:	200	1.00	

▪ **Procedimiento sugerido para resolver este problema utilizando Simulación de Montecarlo**

1. Completar la tabla con los datos que faltan, calculando frecuencia relativa y frecuencia relativa acumulada.
2. Obtener los rangos de números aleatorios y los eventos asociados:
 - De _____ a _____ Equivale a 0 Consultas
 - De _____ a _____ Equivale a 1 Consultas
 - De _____ a _____ Equivale a 2 Consultas
 - De _____ a _____ Equivale a 3 Consultas
 - De _____ a _____ Equivale a 4 Consultas
 - De _____ a _____ Equivale a 5 Consultas
3. Utilizar el Algoritmo sugerido y realizar el diagrama de flujo correspondiente.
4. Implementar el programa en un lenguaje de programación.

▪ **Algoritmo Sugerido**

1. Pedir N (Número de corridas)
2. Inicializar variable "suma" en ceros.
3. Repetir n veces los pasos a,b:
 - a. Generar un numero Aleatorio entre 0 y 1
 - b. Identificar a cuantas consultas al servidor corresponde ese numero (de acuerdo a la tabla), se puede utilizar "CASE" y agregar ese numero de consultas a la variable "suma"
4. Al terminar el ciclo, Calcular el promedio = Suma / n
5. Desplegar el promedio estimado de consultas



PROYECTO DE SIMULACION NUMERO 7 ESTIMACION DE INGRESOS EN VENTA DE LAPTOPS

Un negocio local vende un modelo de computadoras portátiles (Lap Tops) a 2,000 dolares cada uno. El dueño del negocio necesita un pronóstico acerca de cuantas computadoras venderá esta semana. Este pronóstico le permitirá estimar sus ingresos y planificar los pedidos para su proveedor de computadoras, y así mantener siempre ese modelo en existencia dentro de la bodega. El dueño del negocio ha llevado un registro de las ventas de ese modelo de lap tops en las pasadas 100 semanas:

Numero de Laptops vendidas por semana	Frecuencia Absoluta (Num de Semanas)	Frecuencia Relativa (Num de Semanas/Total)	Frecuencia Relativa Acumulada
0	20		
1	40		
2	20		
3	10		
4	10		
TOTAL:	100	1.0	

"ListBox" para mostrar los resultados de cada ejecución del programa.

Para imprimir en el listBox:
`listBox1.Items.Add ("mensaje");`

Para limpiar el listBox:
`listBox1.Items.Clear ();`

Estimar el número de laptops a vender en esta semana, basados en el registro de las 100 semanas anteriores, y determinar también el ingreso esperado. Mostrar el número de computadoras vendidas en cada corrida del programa.

Procedimiento Sugerido:

1. Completar la tabla de Frecuencias relativas y Frecuencias Relativas Acumuladas.
2. Determinar con las Frecuencias Relativas Acumuladas, el rango para cada evento (Ejemplo: De 0 a 0.20 equivale a vender 0 laptops, de 0.20 a 0.60 equivale a vender 1 laptop, etc.)
3. Realizar el diagrama de flujo y el programa basados en los rangos obtenidos en el paso 2.
 - a. Pedir el valor de N (Numero de corridas o ejecuciones)
 - b. Hacer un ciclo que se repita N veces
 - Generar el numero aleatorio
 - Determinar a cual evento corresponde el numero generado (a cuantas laptops vendidas equivale)
 - Ir sumando el numero de laptops vendidas en cada ejecución del ciclo
 - También imprimir en cada ejecución el número de laptops vendidas.
 - c. Al salir del ciclo, calcular el promedio de las laptops vendidas = Sumatoria / N
 - d. Calcular el ingreso esperado = promedio de las laptops * 2000 dlls
 - e. Imprimir los resultados

PROYECTO DE SIMULACION NUMERO 8 ESTIMACION DE PEDIDOS EN UNA PASTERERIA

Una pastelería se especializa en la elaboración de Pays con ingredientes importados. Diariamente se elaboran siempre 14 pays, de los cuales, no siempre se venden todos; sin embargo, hay días en que los pedidos exceden la cantidad de pays disponibles. Se ha observado en los últimos 10 días la siguiente distribución de Pedidos:

Dia 1 – 13 pays pedidos	Dia 6 - 10 pays pedidos
Dia 2 – 15 pays pedidos	Dia 7 - 12 pays pedidos
Dia 3 – 12 pays pedidos	Dia 8 - 13 pays pedidos
Dia 4 – 11 pays pedidos	Dia 9 - 16 pays pedidos
Dia 5 - 14 pays pedidos	Dia 10 – 14 pays pedidos

Estimar el número de pays que se espera vender basados en esta información, y el número de pays que pueden sobrar al final del día, tomando en cuenta que diariamente se elaboran siempre 14 pays.

Simulación de ventas de pays

Este programa estima los pays pedidos y no vendidos en una pastelería basados en una observación de 100 días anteriores. Considerando que diariamente siempre se elaboran 14 pays para vender.

Numero de corridas (n) : 100000

Simular Salir

Estimaciones: Resultados de la Simulación

Numero de pays pedidos (vendidos): 12.99856

Numero de pays no vendidos (14-vendidos): 1.00144

Procedimiento Sugerido:

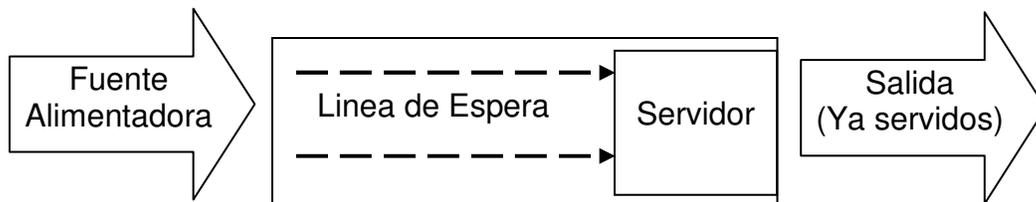
1. Completar la tabla de Frecuencias relativas y Frecuencias Relativas Acumuladas.
2. Determinar con las Frecuencias Relativas Acumuladas, el rango para cada evento (Ejemplo: De 0 a 0.1 equivale a vender 10 pays, de 0.1 a 0.2 equivale a vender 11 pays, etc.)
3. Realizar el diagrama de flujo y el programa basados en los rangos obtenidos en el paso 2.
 - a. Inicializar Variables (Sumatoria en Cero)
 - b. Pedir el valor de N (Numero de corridas o ejecuciones)
 - c. Hacer un ciclo que se repita N veces
 1. Generar el numero aleatorio
 2. Determinar a cual evento corresponde el numero generado (a cuantos pays vendidos equivale)
 3. Ir sumando el numero de pays vendidos en cada ejecución del ciclo
 - d. Al salir del ciclo, calcular el promedio de pays vendidos = Sumatoria / N
 - e. Calcular los Pays No Vendidos = 14 – Pays vendidos
 - f. Imprimir los resultados

SIMULACION DE LINEAS DE ESPERA (SISTEMAS DE COLAS)

El análisis de sistemas de colas frecuentemente ilustra la dificultad de construir un modelo matemático que contenga todos los elementos presentes en el sistema. Generalmente, los modelos de colas asumen que:

- La fuente que alimenta al sistema es de tamaño infinito.
- El sistema ha alcanzado un estado estable.
- Filosofía de Atención: FIFO (First Input – First Output).
- Los clientes que llegan al sistema se unen a la cola.
- Los clientes que están en la cola, permanecerán en ella hasta ser atendidos.
- Los clientes llegan en forma individual.
- Las salidas del sistema son en forma individual.

Ejemplo: La representación de un sistema típico de colas de un servidor se muestra en la figura, donde se puede apreciar que los clientes son generados por una fuente alimentadora, la cual genera X número de clientes por unidad de tiempo. Si un cliente que llega al sistema encuentra desocupado al servidor, entonces este cliente empieza a ser inmediatamente servido y su salida del sistema empieza inmediatamente después de que su servicio ha terminado. Por el contrario, si un cliente al arribar encuentra al servidor ocupado, entonces este cliente pasa a una línea de espera, donde esperará su turno para ser servido. Finalmente, los clientes que permanecen en la cola, son servidos en base a la filosofía FIFO.



En muchas situaciones, sin embargo, el sistema de colas puede tener características especiales importantes. Puede tener etapas de transición y de estado estable, puede ser dependiente, o formar parte de un grupo de facilidades independientes. Puede tener prioridades de servicio, o algunos clientes que llegan al sistema cuando la cola es excesivamente grande pueden rehusar entrar al sistema. También, puede ser que la fuente que alimenta al sistema sea finita. La exclusión de tales características especiales puede distorsionar la naturaleza verdadera del sistema, por lo cual deberán ser considerados en la modelación del mismo.

Si hay varias características especiales importantes, una modelación matemática completa resultaría difícil o casi imposible. En tales situaciones, el análisis completo del sistema puede ser logrado a través del uso de la técnica de simulación.

La simulación de un sistema de colas puede ser realizada de dos maneras:

- **Incrementos a tiempo fijo.-** El tiempo es adelantado en intervalos fijos de tiempo. Por ejemplo, el tiempo puede ser adelantado en intervalos de 30 minutos. Cuando el reloj de la simulación es adelantado en esta unidad de tiempo, el sistema es actualizado al considerar todos los eventos que ocurrieron durante ese intervalo de tiempo. El sistema ahora está en un nuevo estado y listo para que el proceso sea repetido.
- **Incrementos al próximo evento.-** El reloj de la simulación es adelantado por el tiempo entre eventos. Por ejemplo, si el tiempo entre el cuarto y el quinto evento es de 10 minutos, el reloj de la simulación es adelantado por ese lapso de tiempo y el sistema es actualizado para el quinto evento.

Ambos procedimientos actualizan el estado del sistema de acuerdo a los eventos que ocurren en un intervalo específico de tiempo. La diferencia entre estos procedimientos es la manera en la cual los incrementos son especificados.

PROYECTO DE SIMULACION NUMERO 9 SIMULACION DE COLAS EN INCREMENTOS DE TIEMPO FIJO

Simular el comportamiento de la llegada de personas a una linea de espera de registro para un gimnasio con áreas de ejercicio divididas especialmente para hombres y para mujeres. Se desea saber a cada instante de tiempo si llega o no llega una persona para registrarse, de acuerdo a la siguiente distribución de probabilidad:

Probabilidad de que llegue una persona: 20%
Probabilidad de que no llegue una persona: 80%

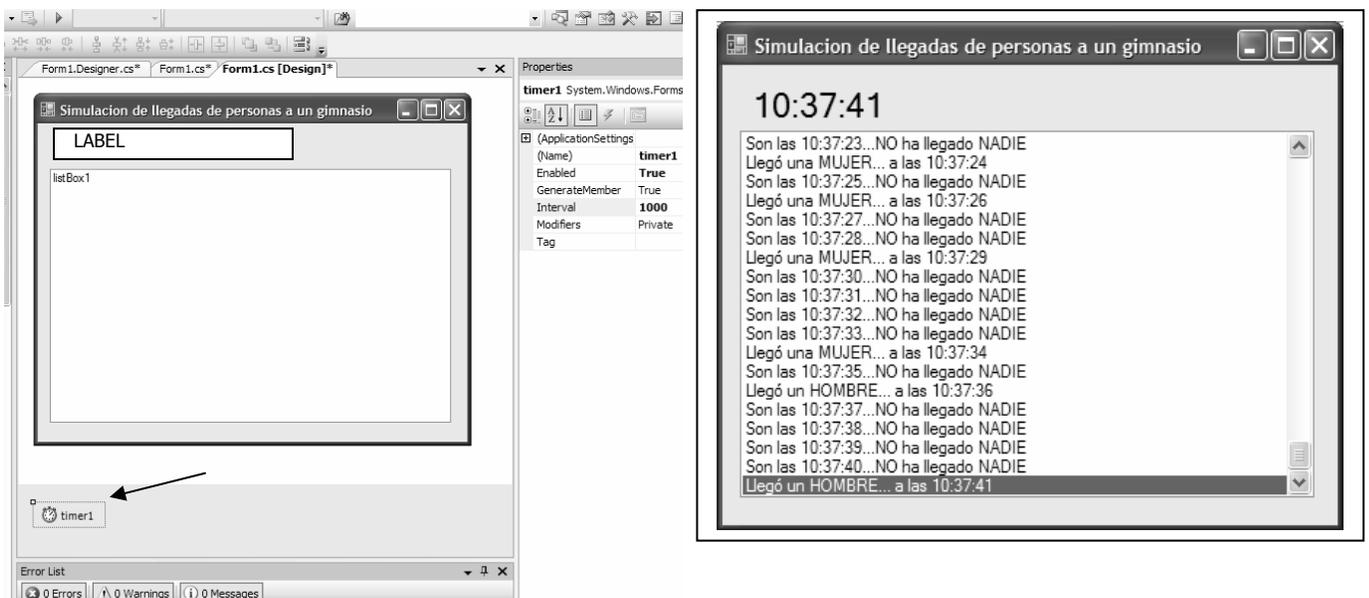
En caso de que llegue una persona, determinar si es hombre o mujer de acuerdo a la siguiente distribución de probabilidad observada con anterioridad:

Probabilidad de que llegue un Hombre: 30%
Probabilidad de que llegue una Mujer: 70%

El programa realizará una simulación continua, es decir, al momento de cargar el programa, se inicia una simulación infinita, hasta que el usuario termine el programa (cierre la ventana)

Recomendaciones para realizar esta simulación:

- Utilizar un control "TIMER" en la forma, y en la propiedad "INTERVAL" de este control, colocar la cantidad de tiempo de incremento que será utilizada para la simulación en milisegundos (1000 milisegundos = 1 segundo). NOTA: Se recomienda usar intervalos cortos (entre 1 y 5 segundos por ejemplo) para observar los resultados de manera mas rápida.
- Usar una lista para mostrar los resultados y una Etiqueta (LABEL) para mostrar la hora actual durante la simulación.
- Todo el código del programa se tecleará en el control "Timer" dentro del único evento que este control posee: "Timer". NOTA: El código tecleado se ejecuta automáticamente cada numero de milisegundos indicado en el intervalo de tiempo (del timer).



PROYECTO DE SIMULACION NUMERO 10 SIMULACION DE COLAS EN INCREMENTOS AL PROXIMO EVENTO

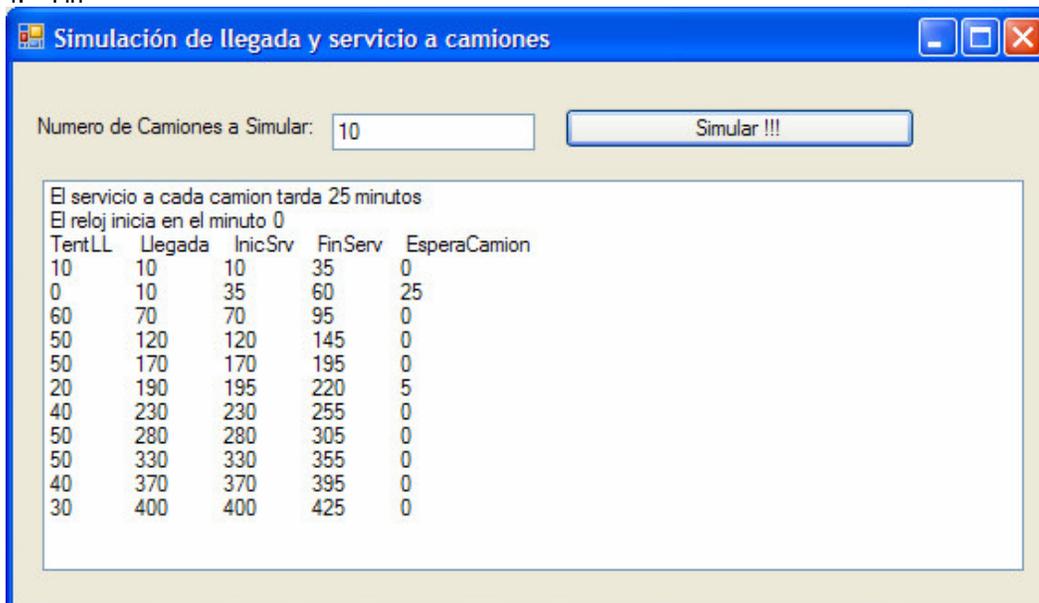
La mercancía llega a un almacén durante la noche. Hay una sola persona que atiende y descarga los camiones. Este empleado se tarda 25 minutos en dar servicio a cada camión que llega. El tiempo entre llegadas de cada camión varía de acuerdo a la siguiente tabla de probabilidades que se han observado:

Tiempo entre llegadas	Frec. Relativa (Probabilidad)	Frec. Rel. Acumulada
0 minutos	0.05	
10 minutos	0.05	
20 minutos	0.10	
30 minutos	0.05	
40 minutos	0.20	
50 minutos	0.35	
60 minutos	0.20	

Simular la llegada de "N" camiones. Y mostrar los siguientes datos para cada camión: Tiempo entre llegadas, Tiempo de Llegada, Inicio Servicio, Fin de Servicio y Tiempo de Espera

Algoritmo Sugerido:

1. Preguntar el valor de "N"
2. Inicializar Tllegada = 0, Terminacionservicio=0, Inicioservicio=0, Tesperacamion=0, Semilla Aleatoria
3. Hacer un ciclo de 1 a N
 - a. Generar un Numero Aleatorio entre 0 y 1
 - b. Determinar a cuantos minutos de "Tiempo entre llegadas" equivale. (Usar un CASE y Poner la cantidad de minutos en la variable Tentrellegadas)
 - c. Tllegada = Tllegada + Tentrellegadas
 - d. Si Tllegada >= Terminacionservicio
Inicioservicio = Tllegada
Si no,
Inicioservicio = Terminacionservicio
 - e. Terminacionservicio = Inicioservicio + 25
 - f. Tesperacamion = Inicioservicio - Tllegada
 - g. Imprimir Tentrellegadas, Tllegada, Inicioservicio, Terminacionservicio, Tesperacamion
4. Fin



EJERCICIO DE REPASO

Estimación de probabilidades en el lanzamiento de dos dados

Introducción

Si se lanzaran dos dados, todas las posibles combinaciones de números obtenidas en el lanzamiento de ambos dados serían 36 en total:

(1,1), (1,2), (1,3), (1,4), (1,5), (1,6),
(2,1), (2,2), (2,3), (2,4), (2,5), (2,6),
(3,1), (3,2), (3,3), (3,4), (3,5), (3,6),
(4,1), (4,2), (4,3), (4,4), (4,5), (4,6),
(5,1), (5,2), (5,3), (5,4), (5,5), (5,6),
(6,1), (6,2), (6,3), (6,4), (6,5), (6,6)

Supongamos que estamos interesados en conocer la probabilidad de obtener 2 números iguales (cualquiera) en el lanzamiento, es decir (1,1) (2,2) (3,3) (4,4) (5,5) o (6,6). Si ambos dados no están cargados (tienen la misma probabilidad de que caiga cualquiera de sus lados), la probabilidad sería de $6 / 36 = 0.1666$

Y la probabilidad de que la suma de los dos números obtenidos en el lanzamiento sea siete, también sería de $6 / 36 = 0.1666$ ya que las posibles combinaciones para obtener un 7 serían (1,6) (2,5) (3,4) (4,3) (5,2) y (6,1) es decir, también 6 en total.

Descripción del programa

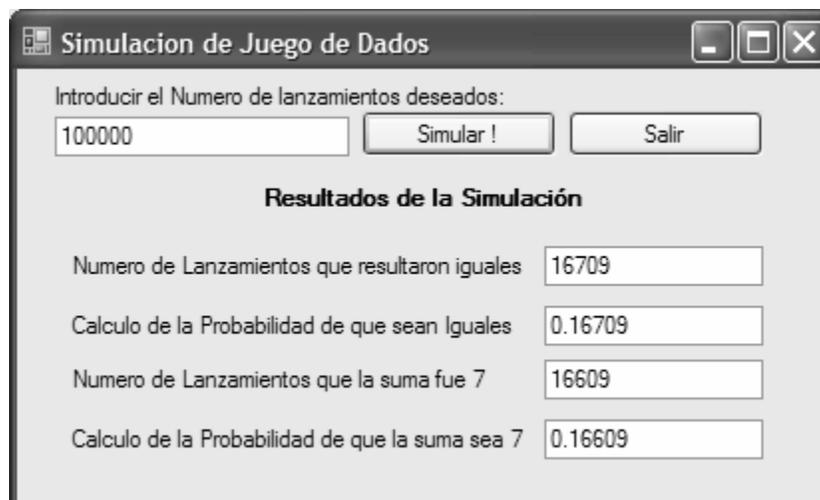
Realizar un programa que permita simular "N" lanzamientos de dos dados, y determinar:

- El número de lanzamientos y la probabilidad de que los números obtenidos sean iguales.
- El número de lanzamientos y la probabilidad de que la suma de ambos números sea 7

El valor de "N" será introducido por el usuario, y se usarán dos números aleatorios entre 1 y 6 (uno para el número obtenido por cada dado).

Probabilidad de que Ambos Dados caigan igual = (Número de veces que fueron iguales) / N
Probabilidad de que la suma de ambos sea 7 = (Número de veces que la suma fue 7) / N

Realizar pruebas con el programa y anotar la probabilidad obtenida en 10, 100, 1000, 10000, 100000, 1000000 lanzamientos. Verificar que a mayor número de lanzamientos, la probabilidad obtenida tiende a ser .1666 en ambos casos (que los números sean iguales, o que su suma sea siete).



Resultados de la Simulación	
Numero de Lanzamientos que resultaron iguales	16709
Calculo de la Probabilidad de que sean Iguales	0.16709
Numero de Lanzamientos que la suma fue 7	16609
Calculo de la Probabilidad de que la suma sea 7	0.16609

UNIDAD III SIMULACION DE SISTEMAS POR COMPUTADORA EN PAQUETES COMPUTACIONALES

Usando Hojas de cálculo comerciales para realizar simulaciones.

Son muchos los autores que han apostado por utilizar hojas de cálculo para realizar simulaciones. La potencia de las hojas de cálculo reside en su universalidad, en su facilidad de uso, en su capacidad para recalculer valores y, sobre todo, en las posibilidades que ofrece con respecto al análisis de escenarios ("*what-if analysis*").

Conceptos básicos para la simulación de MODELOS DETERMINISTICOS en EXCEL.

Usar Excel para la simulación de modelos determinísticos implica el uso y manejo de celdas y formulas dentro de una hoja de cálculo.

Las formulas en EXCEL se inician con el signo de igual y un parentesis, seguidos por cualquier operacion matemática permitida por EXCEL.

Las formulas pueden contener números constantes, simbolos, parentesis, funciones especiales, etc.

Conceptos básicos para la simulación de MODELOS ESTOCASTICOS en EXCEL.

Excel es un paquete comercial muy popular, que posee diferentes funciones integradas que pueden utilizarse para proyectos de simulación. Las últimas versiones de Excel incorporan además, un lenguaje de programación propio: el *Visual Basic for Applications*, con el cual es posible crear auténticas aplicaciones de simulación destinadas al usuario final.

En el mercado existen de hecho varios complementos de Excel (*Add-Ins*) específicamente diseñados para realizar simulación de Monte Carlo, siendo los más conocidos: @Risk, Crystall Ball, Insight.xla, SimTools.xla, etc.

▪ La función =ALEATORIO()

Las hojas de cálculo como Excel (y cualquier lenguaje de programación estándar) son capaces de generar números pseudo-aleatorios provenientes de una distribución uniforme entre el 0 y el 1. Este tipo de números pseudo-aleatorios son los elementos básicos a partir de los cuales se desarrolla cualquier simulación estocástica por computadora.

En Excel, es posible obtener un número pseudo-aleatorio -proveniente de una distribución uniforme entre el 0 y el 1- usando la función =ALEATORIO()

Los números generados mediante la función ALEATORIO tienen dos propiedades que los hacen equiparables a números completamente aleatorios:

1. Cada vez que se usa la función ALEATORIO, cualquier número real entre 0 y 1 tiene la misma probabilidad de ser generado (de ahí el nombre de distribución uniforme).
2. Los diferentes números generados son estadísticamente independientes unos de otros (es decir, el valor del número generado en un momento dado no depende de los generados con anterioridad).

La función =ALEATORIO() es una función **volátil** de Excel. Esto significa que cada vez que pulsamos la tecla **F9** o cambiamos alguno de los *inputs* del modelo, todas las celdas donde aparezca la función ALEATORIO() serán recalculadas de forma automática. Se pueden encontrar ejemplos del uso de ALEATORIO en el propio menú de ayuda de Excel.

▪ **La función BUSCARV**

Busca un valor específico en una columna determinada más a la izquierda de un conjunto de celdas y devuelve el valor en el mismo renglón de una columna especificada en la tabla.

=**BUSCARV**(valor_buscado ; rango_en_donde_buscar ; columna_a_imprimir_si_se_encuentra)

▪ **La función SI**

Devuelve un valor si la condición especificada es VERDADERO y otro valor si dicho argumento es FALSO.

=**SI**(condición ; valor_si_verdadero ; valor_si_falso)

▪ **La Función CONTAR.SI**

Realiza la suma de las celdas dentro de un rango que cumplan con cierta condición especificada.

=**CONTAR.SI** (rango ; condicion)

por ejemplo, la instrucción =CONTAR.SI (B1:B20 ; "Cara") cuenta todas las celdas entre el rango B1 y B20 que contengan la palabra "Cara". El resultado es un número entero.

PROYECTO DE SIMULACION NUMERO 11: SIMULACION DE UN MODELO DETERMINISTICO USANDO EXCEL

The screenshot shows a Microsoft Excel spreadsheet titled "Determinístico". The spreadsheet contains the following data:

	A	B	C	D	E	F	G	H	I
1	Simulación de un modelo Determinístico Usando EXCEL								
2	MODELO: Distancia = Velocidad * Tiempo								
4		Velocidad (Metros / Seg)	Tiempo (Seg)	Distancia (Metros)					
6		20	0	0	=(\$B\$6 * C6)				
7			1	20					
8			2	40					
9			3	60					
10			4	80					
11			5	100					
12			6	120					
13			7	140					
14			8	160					
15			9	180					
16			10	200					
17			11	220					
18			12	240					
19			13	260					
20			14	280					
21			15	300					
22			16	320					
23			17	340					
24			18	360					
25			19	380					
26			20	400					

Utilizar Excel para realizar la simulación del modelo descrito en la práctica #1:

$$Distancia = Velocidad * Tiempo$$

Procedimiento Sugerido.-

- Mantener 3 columnas de datos: Una para Velocidad, otra para Tiempo y una mas para Distancia.
- En la columna de "Velocidad", solamente utilizar la primera celda, pues este dato se mantendrá constante durante la simulación (evaluación).
- Llenar la columna de "Tiempo" con valores.
- En la columna "Distancia" introducir la formula correcta que indique una multiplicacion entre la Velocidad y el valor de Tiempo correspondiente.

NOTA: Se utilizan signos de pesos en los nombres de las celdas para indicar a EXCEL que esa celda es FIJA, y que al copiar o mover su contenido, el nombre de esa celda no debe cambiarse.

- Copiar la formula a las demas celdas.
- Observar el comportamiento del modelo al cambiar el valor de la Velocidad.

**EJEMPLO DEL PROCEDIMIENTO PARA LA SIMULACION DE MONTECARLO
DE VARIABLES DISCRETAS EN EXCEL
PROYECTO 12 : REALIZAR EN EXCEL EL PROYECTO DE LAS CONSULTAS AL SERVIDOR**

- Introducir la tabla del problema (Variable Aleatoria, Frecuencia absoluta, Frecuencia Relativa, Frecuencia Relativa Acumulada). Los totales pueden introducirse manualmente, o calcularse con la función =SUMA(rango)
- Escribir a un lado, los rangos obtenidos y el número de consultas que corresponde a cada rango.
- En otra parte de la hoja de calculo, escribir dos columnas seguidas, una para el número aleatorio y otra para el número de consultas que equivale a ese número aleatorio.
- Introducir la función =ALEATORIO() en la columna de Numero Aleatorio
- Copiar el contenido de la celda "N" veces hacia abajo. Puede "extender" con el mouse, la parte inferior izquierda de la celda hacia abajo, y se copiará el contenido automáticamente.
- Introducir la fórmula =BUSCARV() En la columna "Equivale a" como se muestra en el dibujo, y copiar el contenido de las celdas hacia abajo "N" veces.
- NOTA: Como el rango de celdas usadas para la comparación son siempre las mismas, se utiliza el signo \$ para indicarle a EXCEL que no cambie ese rango cuando se copia la fórmula a otras celdas.
- El resultado de la simulación es el promedio de la columna que contiene el número de consultas que corresponden a cada número aleatorio.

	A	B	C	D	E	F	G	H	I
1									
2	CONSULTAS	FREC ABS	FREC REL	F REL AC	DE...	A...	CONSULTAS	0.57186116	EQUIVALE A
3	0	10	0.05	0.05	0.00	0.05	0	0.45982828	3
4	1	20	0.10	0.15	0.05	0.15	1	0.1848285	2
5	2	40	0.20	0.35	0.15	0.35	2	0.4120446	3
6	3	60	0.30	0.65	0.35	0.65	3	0.90137181	5
7	4	40	0.20	0.85	0.65	0.85	4	0.58384447	3
8	5	30	0.15	1.00	0.85	1.00	5	0.14781943	1
9	Total	200	1.00					0.94592507	5
10								0.14141467	1
11								0.04484085	0
12	RESULTADO DE LA SIMULACION								
13	PROMEDIO: 2.8947368								
14								0.59330513	3
15								0.06096429	1
16								0.94160656	5
17								0.7293992	4
18								0.64086769	3
19								0.17293004	2
20								0.55720697	3
21								0.54102134	3
22								0.9653152	5
23									

Estas columnas pueden calcularse por formulas
 Formula para la celda C3: =(B3/B9)
 Formula para la celda D4: =(C4+D3)

Formula para la primera celda =BUSCARV(H2,\$E\$3:\$G\$8,3)

=PROMEDIO(I2:I20)

=ALEATORIO()

**PROYECTO DE SIMULACION NUMERO 13:
SIMULACION DEL LANZAMIENTO DE UNA MONEDA
PARA ESTIMAR PROBABILIDADES UTILIZANDO EXCEL**

The screenshot shows a Microsoft Excel spreadsheet titled "Microsoft Excel - lanzamiento". The spreadsheet is used for simulating a coin toss. It features a table of random numbers in column A and their corresponding outcomes (Cara or Cruz) in column B. A summary table in columns D-F shows the results of 20 trials, including the number of heads and tails, and the estimated probabilities. The formula bar shows the formula for the first cell in column B: `=BUSCARV(A1,D2:F3,3)`. A callout box explains the formula: "Fórmula para la primera celda de la columna B" and shows the formula `=BUSCARV(A1,D2:F3,3)`. Another callout box shows the formula for the first cell in column C: `=ALEATORIO()`. A third callout box shows the formulas for the summary table: `=CONTAR.SI(B1:B20,"Cara")`, `=CONTAR.SI(B1:B20,"Cruz")`, `=(F9/E7)`, and `=(F10/E7)`.

	A	B	C	D	E	F	G	H
1	0.1065638	Cara		De...	A...	Cae		Simulación del lanzamiento de una moneda.
2	0.6045103	Cruz		0	0.5	Cara		
3	0.1143868	Cara		0.5	1	Cruz		
4	0.9519299	Cruz						
5	0.2628796	Cara						
6	0.6830314	Cruz						
7	0.4021456	Cara						
8	0.9732401	Cruz						
9	0.7233237	Cruz						
10	0.5178202	Cruz						
11	0.6694131	Cruz						
12	0.0382301	Cara						
13	0.2172719	Cara						
14	0.0908264	Cara						
15	0.2026148	Cara						
16	0.8766919	Cruz						
17	0.5277126	Cruz						
18	0.6121898	Cruz						
19	0.1532839	Cara						
20	0.1185156	Cara						
21								
22								

Valor de N: 20

Resultados de la simulación	
NUMERO DE CARAS:	10
NUMERO DE CRUCES:	10
Prob. De que caiga cara:	0.5
Prob. De que caiga cruz:	0.5

Fórmula para la primera celda de la columna B
`=BUSCARV(A1,D2:F3,3)`

`=ALEATORIO()`

`=CONTAR.SI(B1:B20,"Cara")`
`=CONTAR.SI(B1:B20,"Cruz")`
`=(F9/E7)`
`=(F10/E7)`

**PROYECTO DE SIMULACION NUMERO 14:
SIMULACION DEL COMPORTAMIENTO DE UN SEMAFORO FISCAL USANDO EXCEL**

The screenshot shows an Excel spreadsheet with the following data and formulas:

	A	B	C	D	E	F	G	H
1	Numero Aleatorio	Rojo o Verde						
2	0.71442693	ROJO		=SI (A2>=0.4, "ROJO", "VERDE")			SIMULACION DE UN SEMAFORO FISCAL CON 40% DE PROB. DE VERDE Y 60% DE PROB. DE ROJO	
3	0.226097454	VERDE						
4	0.936362363	ROJO						
5	0.045792184	VERDE						
6	0.832077636	ROJO						
7	0.261747353	VERDE						
8	0.018231384	VERDE						
9	0.839813349	ROJO						
10	0.904758017	ROJO						
11	0.972985789	ROJO						
12	0.16788124	VERDE						
13	0.782604272	ROJO						
14	0.890904315	ROJO		NUMERO DE ROJOS:	14			=CONTAR.SI(B2:B31, "ROJO")
15	0.508673208	ROJO		NUMERO DE VERDES:	16			=CONTAR.SI(B2:B31, "VERDE")
16	0.896171199	ROJO		PROBABILIDAD DE ROJO:	0.466667		=(E14/30)	
17	0.926044684	ROJO		PROBABILIDAD DE VERDE:	0.533333		=(E15/30)	
18	0.176531384	VERDE						
19	0.22003515	VERDE						
20	0.229203616	VERDE						
21	0.62722919	ROJO						
22	0.90972568	ROJO						
23	0.067681336	VERDE						
24	0.096920513	VERDE						
25	0.171050361	VERDE						
26	0.323995492	VERDE						
27	0.072000068	VERDE						
28	0.399435971	VERDE						
29	0.749840964	ROJO						
30	0.314483099	VERDE						
31	0.002861537	VERDE						
32								
33								

EJERCICIO DEMOSTRATIVO (REPASO)
SIMULACION DE LINEA DE ESPERA DE ATENCION A CAMIONES CON INCREMENTOS AL PROXIMO EVENTO

Microsoft Excel - Camiones

File Edit View Insert Format Tools Data Window Help

Type a question for help

Arial 10 B I U

E10

	A	B	C	D	E	F	G	H	I
1									
2		De...	A...	Tiempo entre llegadas					
3		0	0.05	0					
4		0.05	0.1	10					
5		0.1	0.2	20					
6		0.2	0.25	30					
7		0.25	0.45	40					
8		0.45	0.8	50					
9		0.8	1	60					
10									
11									
12	=aleatorio()	=buscarv(A16,\$B\$3:\$D\$9,3)	=(C15+B16)	=SI(C16>=E15,C16,E15)	=(D16+25)	=(D16-C16)			
13	Simulación de Servicio a Camiones								
14	NumeroAleatorio	Tiempo entre llegadas	Llegada	InicioServicio	FinServicio	EsperaCamion			
15		0	0	0	0	0			
16	0.233301589	30	30	30	55	0			
17	0.466231237	50	80	80	105	0			
18	0.06831034	10	90	105	130	15			
19	0.087179922	10	100	130	155	30			
20	0.405344531	40	140	155	180	15			
21	0.099970094	10	150	180	205	30			
22	0.017621232	0	150	205	230	55			
23	0.269450461	40	190	230	255	40			
24	0.484301474	50	240	255	280	15			
25	0.536103713	50	290	290	315	0			
26	0.285896599	40	330	330	355	0			
27	0.001830038	0	330	355	380	25			
28	0.184133568	20	350	380	405	30			
29	0.587094898	50	400	405	430	5			
30	0.070323583	10	410	430	455	20			
31	TentrellegadasProm:	25.625			Espera Promedio:	18.66666667			
32									
33									

Sheet1 / Sheet2 / Sheet3

Draw AutoShapes

Ready CAPS NUM

UNIDAD IV GENERACIÓN DE NUMEROS ALEATORIOS

La importancia de los números aleatorios rectangulares (distribución uniforme) radica en su uso para los experimentos de simulación. Algunos autores como Toucher han sugerido tres formas para obtener los números rectangulares:

1. **Provisión externa.**- Implica tener los números aleatorios en un dispositivo de almacenamiento secundario (disco duro, disquete, cinta, etc.) guardados en archivos o bases de datos, y tomarlos como entrada para el problema que se está simulando.
2. **Generación interna a partir de un proceso físico al azar.**- Implica utilizar algún aditamento especial de la computadora digital capaz de registrar los resultados de un proceso aleatorio y además, reducir esos resultados a sucesiones de dígitos.
3. **Generadores matemáticos.**- Es uno de los más aceptados, implica la generación de números a través de una relación de recurrencia.

Es importante mencionar que históricamente, en los inicios de simulación se trabajó con los siguientes métodos para generar números aleatorios:

- **Métodos manuales.**- Consistían en utilizar dispositivos físicos tales como arrojar dados, cartas, monedas, etc. o algún tipo de dispositivo físico o eléctrico que sea capaz de dar resultados aleatorios con igualdad de probabilidades para que los números aleatorios resultantes tengan una distribución uniforme.
- **Tablas de biblioteca.**- Mediante alguno de los métodos descritos antes, la corporación Rand construyó tablas de números aleatorios. La ventaja de las tablas era que las secuencias de números podían reproducirse, cosa que no ocurría con los métodos manuales.
- **Métodos de computación analógica.**- Se utilizaron dispositivos eléctricos capaces de generar pulsos de secuencia aleatoria. De hecho, las tablas de la Rand fueron generadas así, e incluso vendían las tablas grabadas en cinta magnética para alimentar la computadora digital, lo que evidentemente hacía muy lenta la simulación por el problema de los tiempos de acceso a la información de cinta.
- **Métodos de computación digital.**- Generación interna por procesos físicos. Con aditamentos especiales se medía, por ejemplo, el ruido térmico de un circuito en un circuito de válvulas electrónicas de vacío, los resultados obtenidos se registraban y traducían a sucesiones de dígitos. Pero en este método los resultados tampoco podían reproducirse y no era posible comprobar los cálculos efectuados; para ver si cumplían con los requisitos de aleatoriedad y distribución uniforme; de manera que de una experiencia debían sacarse gran cantidad de datos, los que eran almacenados en la memoria de la computadora y se la limitaba mucho.

La alternativa fue la generación de números pseudoaleatorios. Estos, en realidad son generados por métodos matemáticos determinísticos, por consiguiente con un método determinístico.

Se obtienen números que pasan todas las pruebas estadísticas concernientes a la aleatoriedad y a la distribución pero en realidad son determinísticos, por eso el prefijo pseudo (=falso).

El determinismo permite obtener la misma sucesión de números, si cada vez que se inicia la secuencia se lo hace con las mismas entradas y constantes. Tiene la ventaja de que pueden repetirse los experimentos pero también significa que debe cuidarse no obtener la misma secuencia cuando no se lo desee.

Independientemente del método que se utilice, los números rectangulares deben poseer ciertas características deseables que aseguren o aumenten la confiabilidad de los resultados obtenidos en la simulación. Tales características son:

- Uniformemente distribuidos
- Estadísticamente independientes
- Reproducibles
- Periodo largo (Sin repetición dentro de una longitud determinada de la sucesión)
- Generados a través de un método rápido
- Generados a través de un método que no requiera mucha capacidad de almacenamiento de la computadora.

Finalmente, es necesario señalar que algunos autores califican a los números rectangulares generados a través de relaciones de recurrencia como "números pseudoaleatorios", por ser una sucesión de dígitos generada mediante una regla puramente determinística. Sin embargo, esta objeción puede superarse, al menos parcialmente, al tomar el punto de vista un tanto práctico de que una sucesión puede considerarse aleatoria si satisface un cierto conjunto de pruebas estadísticas de aleatoriedad.

Generadores de números aleatorios uniformes

Un algoritmo de computadora que sirve para generar una secuencia de números aleatorios con las características citadas anteriormente es llamado *generador de números aleatorios*. Cualquier algoritmo generador de números aleatorios puede generar solamente un número finito de enteros diferentes, después de lo cual la secuencia vuelve a repetirse (De ahí que también se llamen generadores de números pseudoaleatorios).

Método de los medios cuadrados

En la actualidad dejó de utilizarse, pero históricamente fue muy interesante. Es un método sencillo para usar cuando se quiere generar manualmente alguna secuencia. El método consiste en tomar un número de 4 dígitos, elevarlo al cuadrado y tomar sus cuatro números centrales como el siguiente número de la sucesión.

Ejemplo:

$$x_0 = 2152 \qquad (x_0)^2 = 04631104$$

$$x_1 = 6311 \qquad (x_1)^2 = 39828721$$

$$x_2 = 8287 \qquad (x_2)^2 = \dots\dots\dots$$

Este método se descartó porque estadísticamente era poco satisfactorio. Hágase la prueba con el número 2500.

Generadores Congruenciales Lineales

Varios esquemas han sido propuestos para la generación de los números pseudoaleatorios a través de relaciones matemáticas de recurrencia. Actualmente casi todas las computadoras incluyen en sus programas de biblioteca alguna variante de los métodos congruenciales sugeridos por Lehmer en 1951. Los dos métodos congruenciales mas populares son el congruencial mixto y el congruencial multiplicativo.

▪ **Congruencial Mixto**

Genera una secuencia de números pseudoaleatorios en la cual el próximo número pseudoaleatorio es determinado a partir del último número generado, es decir, el número pseudoaleatorio X_{n+1} es derivado a partir del número pseudoaleatorio X_n

La relación de recurrencia para el generador congruencial mixto es:

$$X_{n+1} = (aX_n + c) \text{ mod } m$$

Donde:

- X₀** es la semilla ($X_0 > 0$)
- a** es el multiplicador ($a > 0$)
- c** es la constante aditiva ($c > 0$)
- m** es el módulo o residuo de la división ($m > X_0, m > a, m > c$)
"m" representa el número de valores diferentes que pueden ser generados.

Los parámetros a, c, m deben ser elegidos cuidadosamente para asegurar un largo periodo y que la secuencia generada tenga buena propiedad de aleatoriedad y uniformidad.

Para empezar el algoritmo se requiere de un valor inicial X_0 llamado semilla.

El algoritmo generará un entero entre 0 y m

Nótese que si se necesita obtener un número entre 0 y 1, basta dividir cada número generado entre m (En caso de un periodo completo).

Para obtener buenos resultados, tomar en cuenta las siguientes consideraciones en la selección de los parámetros para el generador:

- X₀** El valor de la semilla es irrelevante en términos generales.
- a** Entero Impar, no divisible entre 3 o 5.
Para garantizar un periodo completo: Generalmente $2^K + 1$ (elegir $K \geq 2$)
O También $10^K + 1$ (elegir $K \geq 2$)
- c** Puede ser cualquier constante, pero para asegurar buenos resultados se debe seleccionar un Entero impar, relativamente primo a m
- m** Numero primo mas grande posible pero menor que 2^{16} o 10^{16}

Ejemplo : Generador congruencial mixto

Considerar un generador congruencial mixto con $X_0 = 5, a = 21, c = 3, m = 8$

$$X_{n+1} = (aX_n + c) \text{ mod } m$$

- $X_0 = 5$
- $X_1 = (21(5)+3) \text{ mod } 8 = 4$
- $X_2 = (21(4)+3) \text{ mod } 8 = 7$
- $X_3 = (21(7)+3) \text{ mod } 8 = 6$
- $X_4 = (21(6)+3) \text{ mod } 8 = 1$
- $X_5 = (21(1)+3) \text{ mod } 8 = 0$
- $X_6 = (21(0)+3) \text{ mod } 8 = 3$
- $X_7 = (21(3)+3) \text{ mod } 8 = 2$
- $X_8 = (21(2)+3) \text{ mod } 8 = 5$
- $X_9 = (21(5)+3) \text{ mod } 8 = 4$
- $X_{10} = (21(4)+3) \text{ mod } 8 = 7$



Notar que el periodo del generador = m = 8
A partir de la iteración número 9 se obtienen números repetidos

n	0	1	2	3	4	5	6	7	8	9	10
X_n	5	4	7	6	1	0	3	2	5	4	7
	0.625	0.500	0.875	0.750	0.125	0.000	0.375	0.250	0.625	0.500	0.875

**Algunas recetas numéricas eficientes
de Generadores Congruenciales Mixtos de ciclo completo**

Limite en	"m"	"a"	"c"
2 ²⁰	6075	106	1283
2 ²¹	7875	211	1663
2 ²²	7875	421	1663
2 ²³	6075	1366	1283
	6655	936	1399
	11979	430	2531
2 ²⁴	14406	967	3041
	29282	419	6173
	53125	171	11213
2 ²⁵	12960	1741	2731
	14000	1541	2957
	21870	1291	4621
	31104	625	6571
	139968	205	29573
2 ²⁶	29282	1255	6173
	81000	421	17117
	134456	281	28411

¿ Cómo Generaba Visual Studio 6 los números Aleatorios con la función RND ?

Al invocar la función de generación de numeros aleatorios, Visual Studio 6 utiliza internamente un generador congruencial mixto de números aleatorios de la forma

$$X_{n+1} = (aX_n+c) \text{ mod } m$$

Con los siguientes valores:

$$X_0 = 32780, \quad a = 1140671485, \quad c = 12820163, \quad m = 2^{24}$$

Este proceso es totalmente transparente para el programador, y puede pensarse en las funciones de generación de números aleatorios como si fueran funcionalidades encapsuladas de manera eficiente para generar números aleatorios útiles en experimentos sencillos de simulación.

▪ Congruencial Multiplicativo

El generador congruencial multiplicativo nace del generador congruencial mixto con $c = 0$. Tiene la forma

$$X_{n+1} = aX_n \text{ mod } m$$
$$X_n = aX_{n-1} \text{ mod } m$$

O bien,

Estos generadores son mas eficientes (computacionalmente hablando) que los congruenciales mixtos, y seleccionando adecuadamente los parámetros "a", "m" y "x₀" se puede obtener también un periodo completo.

Para obtener buenos resultados, tomar en cuenta las siguientes consideraciones en la selección de los parámetros para el generador:

- X₀** Impar, mayor que cero, de preferencia relativamente primo a "m". Puede usarse 1.
- a** Raiz primitiva de "m", es decir: $a_n \text{ mod } m \neq 1$ para $n = 1, 2, \dots, m-2$
Algunos autores recomiendan seleccionar "a" como $a = 8t \pm 3$ "t" es cualquier entero.
- m** Numero primo mas grande posible y siempre diferente de 2^k "k" es cualquier entero.

Ejemplos:

Un generador muy usado es $X_n = 7^5 X_{n-1} \text{ mod } (2^{31} - 1)$ tiene buenas propiedades aleatorias y es recomendado como un estándar mínimo.

Un estudio de generadores multiplicativos con $m = 2^{31} - 1$ que comparó su eficiencia y aleatoriedad, recomienda los dos siguientes como mejores:

$$X_n = 48271 X_{n-1} \text{ mod } (2^{31} - 1)$$
$$X_n = 69621 X_{n-1} \text{ mod } (2^{31} - 1)$$

▪ Generadores Combinados

1. Si se tienen dos secuencias X_n, Y_n de números aleatorios entre 0 y $m-1$, ambas pueden ser combinadas para obtener una tercera secuencia de la siguiente manera:

$$W_n = (X_n + Y_n) \text{ mod } m$$

- Si las dos secuencias tienen diferentes periodos y son obtenidas por diferentes algoritmos, el periodo y la aleatoriedad pueden incrementar considerablemente.
2. Otra técnica que se puede aplicar a estas dos secuencias X_n, Y_n es el OR-Exclusivo. Esta técnica es similar a la anterior, excepto que la suma es reemplazada por un Or-Exclusivo Bit por Bit. Se ha demostrado que esta técnica aplicada a numeros ligeramente aleatorios puede ser usada para generar números con mayor aleatoriedad.

Recomendaciones al implementar cualquier generador de numeros aleatorios

- No usar valores "cero" para la semilla. Aunque puede funcionar para un generador congruencial mixto, hace que los generadores multiplicativos tengan problemas.
- No use semillas aleatorias. De hacerlo, la simulación no puede ser reproducida y no pueden estudiarse adecuadamente sus características.
- No "inventar" algoritmos propios para generar números aleatorios. Recordar que resultados difíciles de predecir no necesariamente son aleatorios. Es mejor usar operaciones simples cuya aleatoriedad pueda ser evaluada analíticamente.

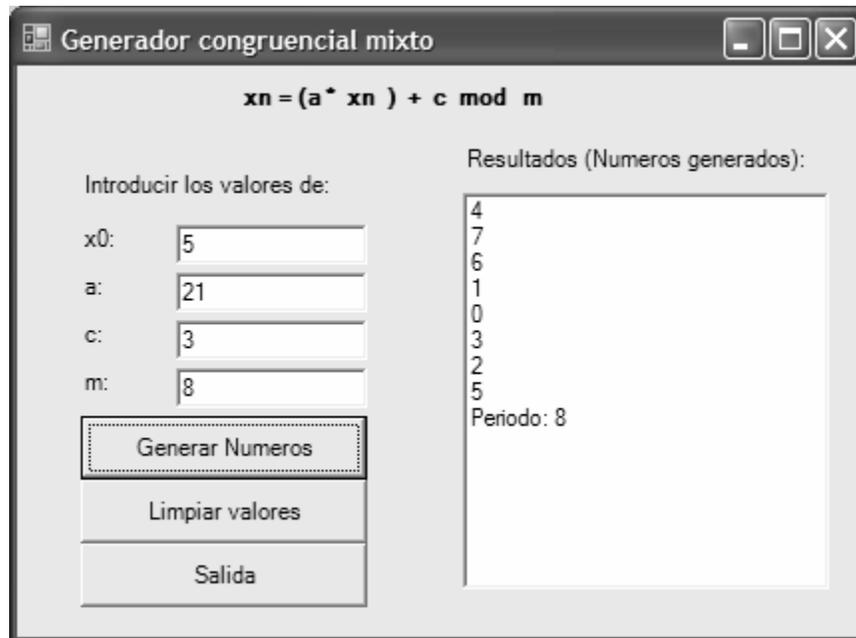
Reflexiones Finales

Diseñar nuevos generadores parece una tarea muy simple; sin embargo, muchos generadores propuestos por expertos estadísticos han sido encontrados deficientes. Por lo tanto, es mejor usar un generador que ya haya sido extensamente probado en lugar de inventar uno nuevo.

PROYECTO DE SIMULACION NUMERO 15: IMPLEMENTACION DE UN GENERADOR CONGRUENCIAL MIXTO DE NUMEROS PSEUDOALEATORIOS

Consideraciones generales para el programa:

- El usuario introducirá los valores de X_0 , a , c , m
- El programa desplegará los números generados en una lista
- Al final de la lista debe mostrarse el periodo del generador
- Incluir los botones de "Generar números aleatorios" para iniciar con el proceso, "Limpiar Valores" para quitar todos los datos de la pantalla y el botón "Salir" para abandonar el programa.



Procedimiento Sugerido (Botón "Generar numeros aleatorios")

1. Obtener los valores de X_0 , a , c , m
2. Inicializar Periodo y NumAleatorio en ceros
3. NumAnterior = X_0
4. Repetir mientras NumAleatorio $\neq X_0$ y Periodo $\leq m$
 - Calcular numAleatorio = $(a * \text{numAnterior} + c) \text{ Mod } m$
 - Cambiar numAnterior = numAleatorio
 - Incrementar Periodo
 - Imprimir en la lista el numero aleatorio
5. Al salir del ciclo, imprimir Periodo en la misma lista o en un cuadro de texto.

Verificar el programa con los siguientes parámetros y observar los resultados. Determinar si son buenos o malos generadores.

1. $X_0 = 154$ $a = 250$ $c=124$ $m=896$
2. $X_0 = 5$ $a = 21$ $c=3$ $m=8$
3. Utilizar combinaciones de la tabla de "recetas numéricas" y demostrar que son generadores de ciclo completo.

AGREGANDO CARACTERISTICAS DE ALMACENAMIENTO AL PROYECTO 15

Los números aleatorios (como cualquier otra información) generados por un programa, pueden ser almacenados en archivos de diferentes formatos desde el mismo programa por medio de código escrito con la finalidad de guardar los datos para uso posterior.

Modificar el proyecto 15 de la siguiente manera:

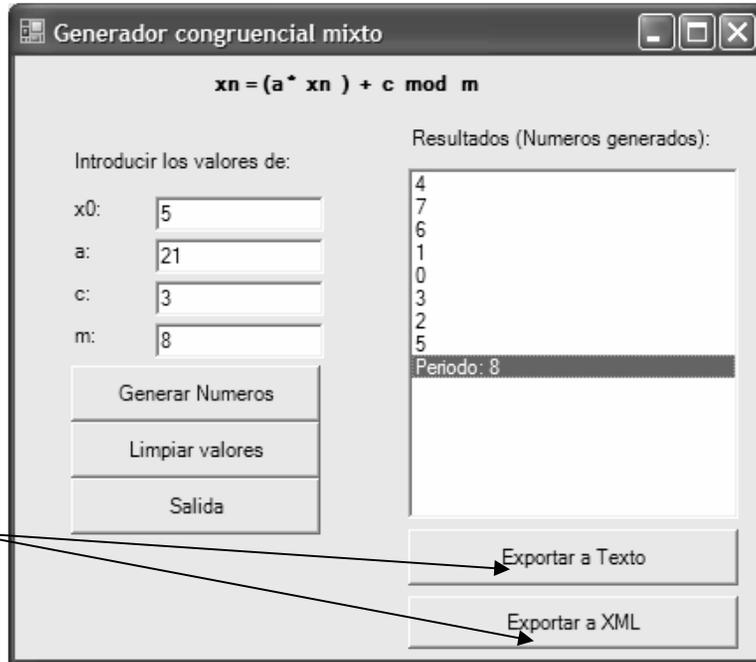


Figura 1.
Agregar
estos
botones al
proyecto 15

NOTA:

XML Significa eXtensible Markup Language y es un formato de representación de datos basado en texto plano muy utilizado actualmente en el intercambio de información en el Web, así como entre plataformas, sistemas operativos y aplicaciones con formatos diferentes o incompatibles.

XML es un estándar abierto (no es propietario de ninguna compañía).

Código para el botón "Exportar a Archivo de Texto"

```
int x;  
try  
{  
    System.IO.FileStream miArchivo = new  
        System.IO.FileStream  
            ("C:\\Ejemplo.txt", System.IO.FileMode.Create,  
                System.IO.FileAccess.ReadWrite);  
  
    System.IO.StreamWriter FlujoEscritura =  
        new System.IO.StreamWriter(miArchivo);  
  
    for (x=0;x<=listBox1.Items.Count-1;x++)  
    {  
        listBox1.SelectedIndex = x;  
        FlujoEscritura.WriteLine (listBox1.Text);  
    }  
    FlujoEscritura.Flush();  
    FlujoEscritura.Close();  
    miArchivo.Close();  
    MessageBox.Show  
        (" El archivo ha sido creado exitosamente..");  
}  
catch (Exception ee)  
{  
    MessageBox.Show (" Hubo un error : " + ee.Message);  
}
```

Código para el botón "Exportar a Archivo XML"

```
try  
{  
    using (System.Xml.XmlWriter writer =  
        System.Xml.XmlWriter.Create(@"c:\ejemplo.xml"))  
    {  
        writer.WriteStartElement("NumerosAleatoriosGenerados");  
        for (int x = 0; x <= listBox1.Items.Count - 2; x++)  
        {  
            listBox1.SelectedIndex = x;  
            writer.WriteStartElement ("Numero");  
            writer.WriteElementString  
                ("ValorEnteroSolamente", listBox1.Text);  
  
            writer.WriteElementString ("ValorEntre0y1",  
                System.Convert.ToString  
                    (double.Parse(listBox1.Text)  
                    / double.Parse(txtm.Text) ) );  
            writer.WriteEndElement();  
        }  
        writer.WriteEndElement();  
        writer.Flush();  
    }  
    MessageBox.Show("Archivo XML creado exitosamente");  
}  
catch (Exception ex)  
{  
    MessageBox.Show("Error..." + ex.Message);  
}
```

PROYECTO DE SIMULACION NUMERO 16 IMPLEMENTACION EN EXCEL DE UN GENERADOR CONGRUENCIAL MIXTO DE NUMEROS PSEUDOALEATORIOS

	A	B	C	D	E	F	G	H	I	J	K
4	Generador de Nums. Aleatorios Congruencial Mixto										
5	$x_{n+1} = (a x_n + c) \text{ mod } m$										
6		a	c	m	x	Num Aleat generado					
7		21	3	8	5	0.625					
8					4	0.5					
9		=RESIDUO(\$B\$7 * E7 + \$C\$7, \$D\$7)			7	0.875					
10					6	0.75					
11					1	0.125					
12					0	0					
13					3	0.375					
14					2	0.25					
15					5	0.625					
16					4	0.5					
17					7	0.875					
18					6	0.75					
19					1	0.125					
20											

En las celdas con color gris se teclea solamente el valor
En las celdas con color rojo se debe teclear la formula
En las celdas con color amarillo se copia la formula en rojo

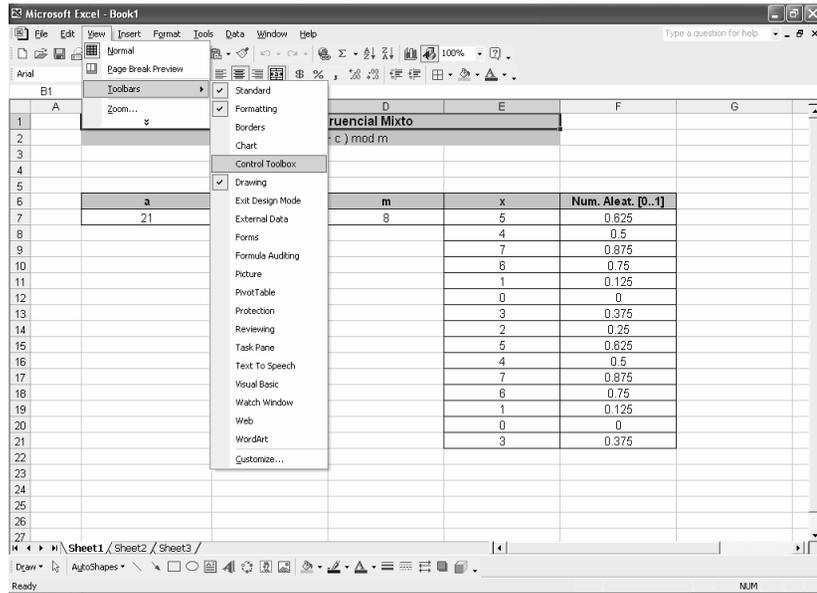
Probar el generador con los valores propuestos $a = 21$, $c = 3$, $m = 8$ y $x_0 = 5$
Luego, cambiar los valores por otros, y verificar el cambio de los números aleatorios generados.

PROYECTO 17

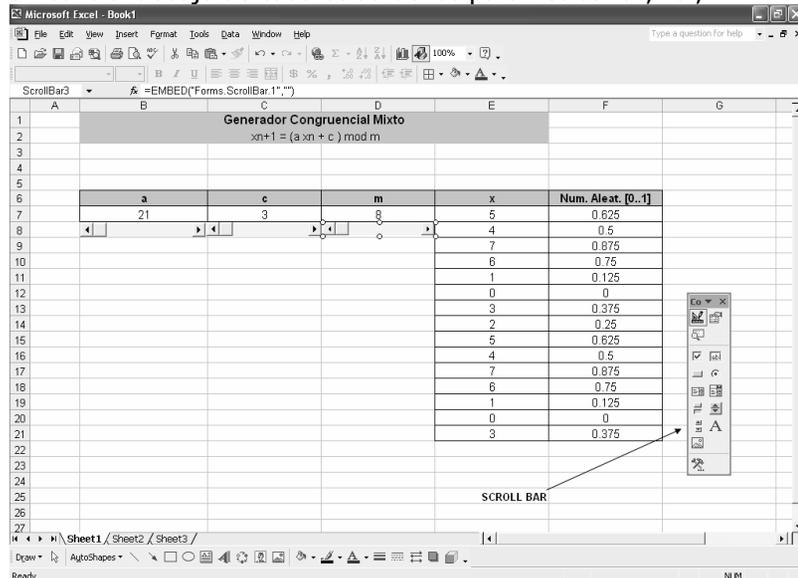
Agregando "BARRAS ACTIVAS" al generador Congruencial Mixto

PASO 1.- Crear la hoja de trabajo como se describe en la práctica 16

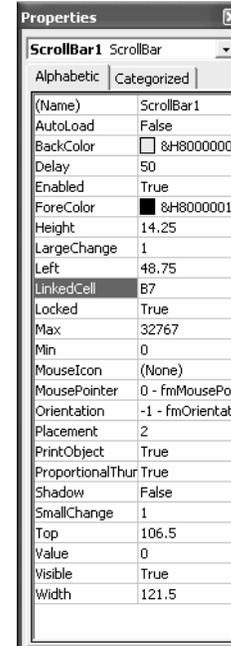
PASO 2.- Del menú de EXCEL seleccionar VIEW – ToolBars – Control Toolbox



PASO 3.- De esa barra de herramientas, seleccionar el "SCROLL BAR" y poner un control debajo de cada celda de los parametros "a", "c", "m"



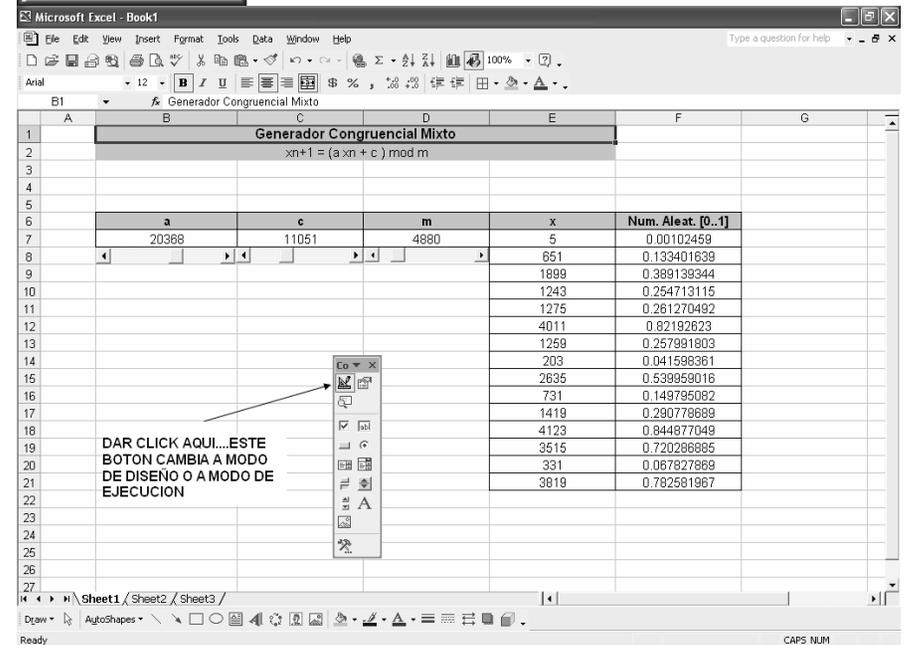
PASO 4.- Dar Click derecho en el SCROLLBAR debajo del parámetro "a" y seleccionar PROPIEDADES. En la propiedad "LIKED CELL" teclear "B7"



PASO 5.- Repetir el paso 4 ahora para el SCROLLBAR debajo del parámetro "c", y teclear "C7" en la propiedad LINKED CELL

PASO 6.- Repetir el paso 4 ahora para el SCROLLBAR debajo del parámetro "m", y teclear "D7" en la propiedad LINKED CELL

PASO 7.- Oprimir el boton con la figura de "una escuadra y un lapiz". Este boton cambia de modo "diseño" a modo "ejecución" y viceversa. Hacer pruebas moviendo el SCROLL de las barras. Observar los cambios en los valores.

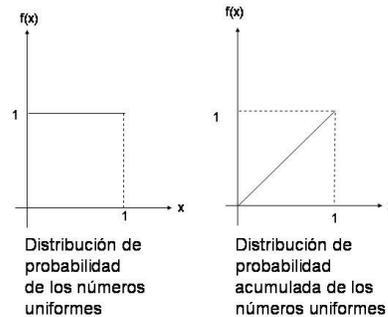


UNIDAD V.- PRUEBAS ESTADÍSTICAS PARA NUMEROS PSEUDOALEATORIOS

Una vez que se han generado números aleatorios, es necesario asegurarse de que el generador usado haya producido una secuencia de números suficientemente aleatoria. Para esto se someten los números generados a pruebas estadísticas. Si los números no pasan la prueba, se puede asumir que el generador es malo. Pasar una prueba es una condición necesaria pero no suficiente, ya que un generador puede pasar una prueba y luego no pasarla si se usa otra semilla u otro segmento del ciclo.

▪ Prueba de los promedios

Esta prueba se basa en el hecho de que la distribución de probabilidad y la distribución acumulada de los números uniformes en un intervalo (0:1) son de la siguiente manera:



Conociendo estos parámetros, es posible plantear una prueba de hipótesis de promedios, con la cual se trata de probar que los números pseudoaleatorios generados provienen de un universo uniforme con media de 0.5. Una prueba de hipótesis de promedios puede ser planteada de la siguiente forma:

$$\begin{aligned} \text{Hipótesis Nula} & \quad H_0 : u = 1/2 \\ \text{Hipótesis alternativa} & \quad H_1 : u <> 1/2 \end{aligned}$$

Pasos para realizar la prueba de los promedios:

1. Se requiere tener N números pseudoaleatorios para realizarles la prueba.
2. En seguida se obtiene el promedio aritmético de estos números de la siguiente manera:

$$prom = \frac{Num_1 + Num_2 + \dots + Num_N}{N}$$

3. Ahora se obtiene Z_0 que es un estadístico que se calcula de la siguiente manera:

$$z_0 = \frac{\left(prom - \frac{1}{2} \right) \sqrt{N}}{\sqrt{1/12}}$$

4. El valor obtenido de Z_0 Se compara con la Z de tablas. Si $ABS(Z_0) < Z$ de tablas entonces se Acepta la hipótesis nula y los números pasan la prueba, de lo contrario, se rechaza la hipótesis nula y los números no pasan la prueba. A un nivel de confiabilidad del 95% el valor de Z de tablas es de 1.96

Entonces, puede generalizarse que, para un grado de confiabilidad de 95%, si el valor absoluto obtenido de Z_0 es menor que 1.96 los números han pasado la prueba.

Ejemplo: Obtener la prueba de los promedios de los siguientes números pseudoaleatorios:

0.625 ,0.5, 0.875, 0.75, 0.125, 0, 0.375, 0.25

Se Obtiene el promedio Aritmético:

$$prom = \frac{0.625 + 0.5 + 0.875 + 0.75 + 0.125 + 0 + 0.375 + 0.25}{8} = 0.4375$$

Se Obtiene el estadístico:

$$z_0 = \frac{\left(prom - \frac{1}{2} \right) \sqrt{N}}{\sqrt{1/12}} = \frac{(0.4375 - 0.5) \sqrt{8}}{\sqrt{1/12}} = -.6123$$

¿Es $ABS(z_0) < 1.96$?
 $ABS(-.6123) = .6123 < 1.96$
 SI. Los números han pasado la prueba

▪ **Prueba de las Frecuencias (CHI-Cuadrado)**

Esta es probablemente una de las mas importantes pruebas sobre aleatoriedad. Consiste en dividir el intervalo (0:1) en n subintervalos para luego comparar para cada subintervalo la frecuencia esperada con la frecuencia observada. Si estas frecuencias son bastante parecidas, entonces los números provienen de una distribución uniforme.

Pasos para realizar la Prueba de las Frecuencias

1. Se requiere tener N números pseudoaleatorios para realizarles la prueba.
2. Luego, dividir el intervalo (0:1) en varios subintervalos de la misma longitud.
3. Determinar cuantos numeros de los "N" a realizarles la prueba **se encuentran** en cada intervalo (Frecuencia Obseveda FO) y cuantos **deberían encontrarse** en cada intervalo (Frecuencia Esperada FE). Por ejemplo, para 5 intervalos iguales, esto puede hacerse en una tabla de la siguiente manera:

Frecuencia Esperada(FE)	FE ₁	FE ₂	FE ₃	FE ₄	FE ₅
Frecuencia Observada(FO)	FO ₁	FO ₂	FO ₃	FO ₄	FO ₅
	De 0 a 0.2	De 0.2 a 0.4	De 0.4 a 0.6	De 0.6 a 0.8	De 0.8 a 1.0

4. Calcular x_0^2 que es un estadístico que se obtiene de la siguiente manera:

$$x_0^2 = \frac{1}{FE} \left[(FO_1 - FE_1)^2 + (FO_2 - FE_2)^2 + \dots + (FO_n - FE_n)^2 \right]$$

5. Se compara este estadístico con x_0^2 de tablas con parámetros de porcentaje de confiabilidad y (n-1) grados de libertad. Si x_0^2 es menor que x_0^2 de tablas, se Acepta la hipótesis nula y los números pasan la prueba, de lo contrario, se rechaza la hipótesis nula y los números no pasan la prueba.

NOTA: Si se usan 5 intervalos y un nivel de confiabilidad del 95%, x_0^2 de tablas = 9.49

Ejemplo:

Realizar la prueba de las frecuencias a 100 números pseudoaleatorios generados por un programa implementado en C# utilizando 5 intervalos iguales de (0.20) cada uno

Obtener los 100 números aleatorios sin la intervención del usuario (Es decir, generados directamente dentro del programa).

Se divide el intervalo (0:1) en 5 subintervalos iguales (de 0.2) y se determina la frecuencia esperada y observada. La frecuencia Esperada = INT(N/5) = INT (100/5) = 20

Frecuencia Esperada(FE)	20	20	20	20	20
Frecuencia Observada(FO)	21	22	19	23	15
	De 0 a 0.2	De 0.2 a 0.4	De 0.4 a 0.6	De 0.6 a 0.8	De 0.8 a 1.0

4. Se Calcula x_0^2

$$x_0^2 = \frac{1}{20} \left[(21 - 20)^2 + (22 - 20)^2 + (19 - 20)^2 + (23 - 20)^2 + (15 - 20)^2 \right] = 2$$

5. Se compara el estadístico obtenido con valores de tablas a un nivel de confianza del 95%:

¿Es $2 < 9.49$? SI. Los números PASAN la prueba

PROYECTO DE SIMULACION NUMERO 18 PRUEBA DE LOS PROMEDIOS PARA NUMEROS PSEUDOALEATORIOS

Implementar un algoritmo que realice la prueba de los promedios a "N" números pseudoaleatorios generados dentro de un programa en C#. El usuario introducirá el valor de "N" y el programa debe dar como resultado el valor del estadístico z_0 y un mensaje que indique si la prueba fue superada o no. Realizar la comparación del estadístico con 1.96 que es el valor de Z de tablas para un nivel confiabilidad del 95%.

NOTA: La función para obtener el Valor absoluto de un número en C# es: `System.Math.Abs(numero)`
La función para obtener la raíz cuadrada de un número en C# es: `System.Math.Sqrt(numero)`

Prueba de los promedios para números pseudo-aleatorios

Este programa realiza la prueba de los promedios a "N" numeros pseudoaleatorios generados desde C#

Cantidad de Números a generar: (Valor de N) 1000 Generar Numeros y Realizar Prueba

Resultados de la prueba

Estadistico Z0 Calculado 0.711438644522388

Resultado: Si ABS(Z0 < 1.96 se pasa la prueba con 95% de SI PASAN LA PRUEBA

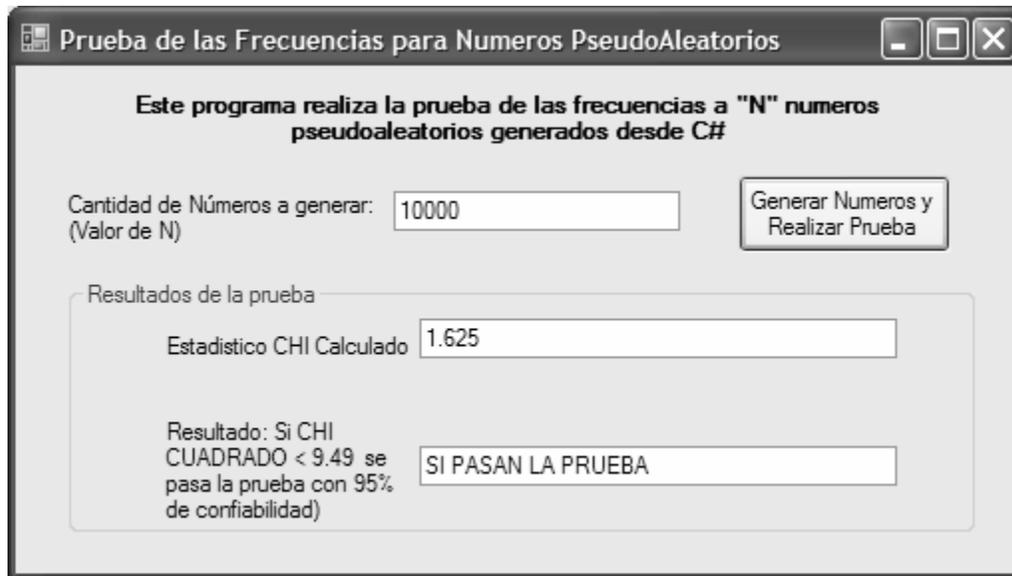
PROCEDIMIENTO SUGERIDO.-

1. Obtener el valor de N
2. Declarar las variables e inicializar suma=0
3. Hacer un ciclo de 1 a N
 - 3.1. suma = suma + Nuevo numero aleatorio generado
4. Calcular promedio = suma/N
5. Calcular $Z_0 = ((\text{promedio} - 0.5) * \text{raiz}(N)) / \text{raiz}(1.0/12)$
6. Calcular $\text{abs}Z_0 = \text{ValorAbsoluto}(Z_0)$
7. Si $\text{abs}z_0 < 1.96$
Imprimir "Si pasan la prueba"
Si no,
Imprimir "No pasan la prueba"
8. Fin del programa

PROYECTO DE SIMULACION NUMERO 19 PRUEBA DE LAS FRECUENCIAS PARA NUMEROS PSEUDOALEATORIOS

Implementar un algoritmo que realice la prueba de las frecuencias a "N" números pseudoaleatorios generados con la instrucción RND. El usuario introducirá el valor de "N" y el programa debe dar como resultado el valor del estadístico χ_0^2 y un mensaje que indique si la prueba fue superada o no. Utilizar 5 intervalos de tamaño 0.20 (Frecuencia Esperada = $\text{int}(N/5)$) y realizar la comparacion del estadístico χ_0^2 con 9.45 que es el valor de tablas para un nivel confiabilidad del 95%.

NOTA: Para elevar un número a cualquier potencia en C# se utiliza: `System.Math.Pow(numero, exponente)`
Por ejemplo, para elevar un número al cuadrado se utilizaría: `System.Math.Pow(numero, 2)`



PROCEDIMIENTO SUGERIDO

1. Obtener el valor de "N"
2. Inicializar 5 variables en ceros para llevar la cuenta de la Frecuencia Observada (FO1=0, FO2=0, FO3=0, FO4=0, FO5=0)
3. $FE = \text{Int}(N / 5)$
4. Hacer un ciclo de 1 a N
 - 4.1 Aleatorio = Num aleatorio
 - 4.2 Verificar en que rango se encuentra la variable "Aleatorio":
 - De 0 a 0.20 : FO1 = FO1 +1
 - De 0.20 a 0.40 : FO2 = FO2 +1
 - De 0.40 a 0.60 : FO3 = FO3 +1
 - De 0.60 a 0.80 : FO4 = FO4 +1
 - De 0.80 a 1.00: FO5 = FO5 +1
5. Calcular CHI = Formula en apuntes
6. Imprimir el valor obtenido de CHI
7. Si $CHI < 9.45$,
 - imprimir "SI PASAN LA PRUEBA",
 - de lo contrario
 - imprimir "NO PASAN LA PRUEBA"