



Integrator Project I - Taller Diseño de Experimentos

2020-1

Christian Flor - A00355624, Daniel Fernández - A00354694, Felipe Sanchez - A00355727

Taller Diseño de Experimentos

DISEÑO DE EXPERIMENTOS

1. Planeación y realización

Este experimento pretende evidenciar los efectos que tienen algunos factores sobre proceso de ordenamiento de los distintos datos contenidos en un arreglo. Encontrar posibles mejoras para este tipo de operaciones es fundamental hoy en día, pues los datos son una parte esencial de cómo vivimos. Desde la información de los alumnos de un colegio hasta los datos financieros de alguna empresa multinacional; es muy probable, por no decir seguro, que se requiera que estos datos estén en algún orden específico, ya sea para su visualización o para llevar a cabo operaciones complejas. Dicho esto, la eficiencia con que este proceso sea llevado a cabo tendrá impacto en muchos escenarios de la vida cotidiana.

Unidad experimental:

Para este experimento la unidad experimental son los grupos de repeticiones de cada tratamiento, es decir, cada tratamiento pasará por n repeticiones y se calculará el promedio de ese tratamiento, el cual será utilizado para realizar los cálculos estadísticos.

Factores controlables:

Para este experimento se definieron los siguientes factores controlables basados en su impacto en el experimento y en la facilidad con que estos pueden ser modificados:

- Algoritmo implementado.
- Tamaño de la entrada.
- Orden de los elementos en el arreglo.

Factores no controlables:

Las siguientes condiciones se definieron como no controlables al presentar características como alta dificultad para su manipulación, altos costos o al requerir conocimientos que van más allá de los que actualmente se manejan, entre otras:

- Procesadores del equipo de computo.
- Ram del equipo.

- Otros procesos que realiza el equipo (algunos se pueden controlar, pero otros son de vital importancia para el funcionamiento del dispositivo por lo que no se pueden intervenir).

Variable de respuesta: para este experimento la variable de respuesta seleccionada fue el tiempo que se emplea en el ordenamiento de los elementos del arreglo, ya que interesa conocer qué tan eficientes son estos procesos en relación al tamaño del arreglo.

Para medir el tiempo se hará uso de una función que retorne la cantidad de milisegundos que han transcurrido desde un momento de referencia hasta el momento en que empieza a correr el algoritmo y el momento en que termina, valores con los que se puede obtener el tiempo de ejecución del algoritmo, de esta manera no se incluye el tiempo que se pasa configurando el escenario y haciendo las pruebas unitarias.

Variables de estudio: se tendrán en cuentas algunos de los factores que tienen gran impacto en este proceso, se harán variar para observar cómo afectan estos a las variables de respuesta. Las variables de estudio son las siguientes:

- Algoritmo de ordenamiento usado: es decir, la secuencia lógica de pasos que se realizarán para llevar el arreglo al estado que se desea.
- Tamaño del arreglo: cuántos elementos está almacenando la estructura de datos.
- Estado de los valores en el arreglo: como están organizados los elementos inicialmente, esto puede ser, ascendente o descendentemente, así como no presentar ningún patrón.

Niveles de cada factor: los factores anteriores tendrán los siguientes niveles:

- Algoritmo de ordenamiento:
 - Burbuja.
 - Inserción.
- Tamaño de la entrada:
 - 10^1
 - 10^2
 - 10^3
 - 10^4
- Estado de los valores en el arreglo:
 - Ordenados ascendentemente.
 - Ordenados descendentemente.
 - Aleatoriamente.

La combinación de los anteriores niveles da como resultado 24 diferentes tratamientos a la hora de evaluar el proceso de ordenamiento.

Planear y organizar el trabajo experimental:

Ya con toda la información requerida, se procederá de la siguiente manera:

Un equipo de computo ejecutará todos los veinticuatro tratamientos definidos, cada uno contará con mil repeticiones. Para cada ejecución de un tratamiento, se tomará el tiempo que este tarde en terminar y se guardará dicho valor en un archivo .csv con el fin de realizar un análisis estadístico de todos los datos una vez concluidas todas las pruebas.

Durante el tiempo de ejecución de las pruebas, el equipo de cómputo no realizará ninguna otra operación en primer plano para así dedicar la mayor cantidad de recursos posibles a la recolección de los datos que se necesitan. Operaciones en segundo plano y otras que se requieran para el funcionamiento normal de dispositivo escapen al control que se puede ejercer en este experimento, por lo tanto no se verán detenidas.

Realizar el experimento:

El plan se llevó a cabalidad en el computador seleccionado y se siguió al pie de la letra. Llegado al caso de que sucediera algún problema en la ejecución de las pruebas nuestro compañero que estaba llevando a cabo las pruebas nos reportaría lo sucedido para así saber que teníamos que modificar del plan.

Debido a que llevamos todo el plan al pie de la letra todo funcionó correctamente y hasta en un menor tiempo al que habíamos calculado. Las 1000 repeticiones tomaron un total de 2 horas y 11 minutos, empezando a las 6:23PM y terminando a las 8:34PM del día sábado 15 de febrero de 2020.

2. Análisis(ANOVA)

[Click aquí para ver la hoja de cálculo con los datos recolectados y el análisis ANOVA.](#)

3. Interpretación

El análisis de varianza ANOVA se utiliza para contrastar parámetros en más de dos grupos. En este caso nos interesa saber si el tiempo promedio de ejecución de cada tratamiento es igual o diferente al de los demás dado un nivel de significancia o error tipo I del 5%.

Para este propósito definimos dos hipótesis, la hipótesis nula (H_0) y la hipótesis alterna (H_1):

H_0 : El tiempo promedio de todos los tratamientos es el mismo.

H_1 : Al menos uno de los tratamientos tiene un tiempo promedio distinto a los demás.

Para tomar una decisión debemos comparar el estadístico **Sig** que arroja el análisis ANOVA; si este es menor que el nivel de significancia, en este caso 5%, se rechaza la hipótesis nula.

Este experimento dio en sus resultados que el valor **Sig** es igual a 0, como es menor que el nivel de significancia (0,05) entonces se asume la hipótesis alterna que dice que al menos uno de los tratamientos tiene un promedio de tiempo diferente al resto.

Cuando la hipótesis nula es rechazada, es conveniente realizar un post ANOVA o prueba post hoc, ya que esta analiza todas las posibles parejas de tratamientos para saber cuales son diferentes o iguales dado el nivel de significancia previo. Por ejemplo:

(I) Tratamiento		Diferencia de medias (I-J)	Desv. Error	Sig.	Intervalo de confianza al 95%	
					Límite inferior	Límite superior
ba1	ia1	-16	1,297	1,000	-4,73	4,70
	bi1	1	1,297	1,000	-4,72	4,72
	ii1	3	1,297	1,000	-4,72	4,72
	bd1	1	1,297	1,000	-4,72	4,72
	id1	2	1,297	1,000	-4,72	4,72

En la anterior imagen la prueba post hoc compara el tratamiento ba1 (algoritmo burbuja, orden aleatorio, tamaño de entrada 10^1) con otros 5 tratamientos. Las medias de cualquier par de tratamientos serán iguales si el límite inferior y el límite superior tiene signos opuestos; así, con un nivel de significancia del 5%, vemos que el tratamiento ba1 tiene una media de tiempo igual al de los tratamientos ia1, bi1, ii1, bd1 e id1.

Ahora un ejemplo de dos tratamientos con medias diferentes:

ii3	ba3	-21,938*	1,297	0	-26,66	-17,22
-----	-----	----------	-------	---	--------	--------

Aquí se comparan el tratamiento ii3 (algoritmo inserción, orden ascendente, tamaño de entrada 10^3) con el tratamiento ba3 (algoritmo burbuja, orden aleatorio, tamaño de entrada 10^3); como se puede ver, en las columnas correspondientes a ambos límites se encuentran valores negativos, esto significa que la media del tratamiento ba3 es mayor que la del ii3, ya que en esencia estos límites se calculan al restar los diferentes valores que podrían tomar las medias de ambos tratamientos, así:

límite inferior = minvp_ii3 - minvp_ba3

límite superior = maxvp_ii3 - maxvp_ba3

minvp_ii3 => menor valor posible de ii3

minvp_ba3 => menor valor posible de ba3

maxvp_ii3 => mayor valor posible de ii3

maxvp_ba3 => mayor valor posible de ba3

4. Control y conclusiones finales

Este experimentos permite concluir lo siguiente:

- El ordenamiento con el algoritmo inserción es más eficiente a medida que el tamaño de la entrada aumenta.
- El ordenamiento con el algoritmo inserción tiene un mejor comportamiento cuando la lista ya está ordenada, su forma cuadrática en este caso solo se nota para entradas muy grandes.

Análisis de complejidad temporal y espacial de los algoritmos escogidos

Complejidad temporal BubbleSort		
#	public static void BubbleSort<T>(List<T> items) where T : IComparable{	#veces que se repite
1	for (int i = 0; i < items.Count; i++) {	$n + 1$
2	for (int j = 0; j < items.Count - 1; j++) {	$n(n + 1) = n^2 + n$
3	if (items[j].CompareTo(items[j + 1]) > 0) {	n^2
4	T temp = items[j + 1];	n^2
5	items[j + 1] = items[j];	n^2
6	items[j] = temp;	n^2
7	}	0
8	}	0
9	}	0
10	}	0
		$O(5n^2 + 2n + 1) = O(n^2)$

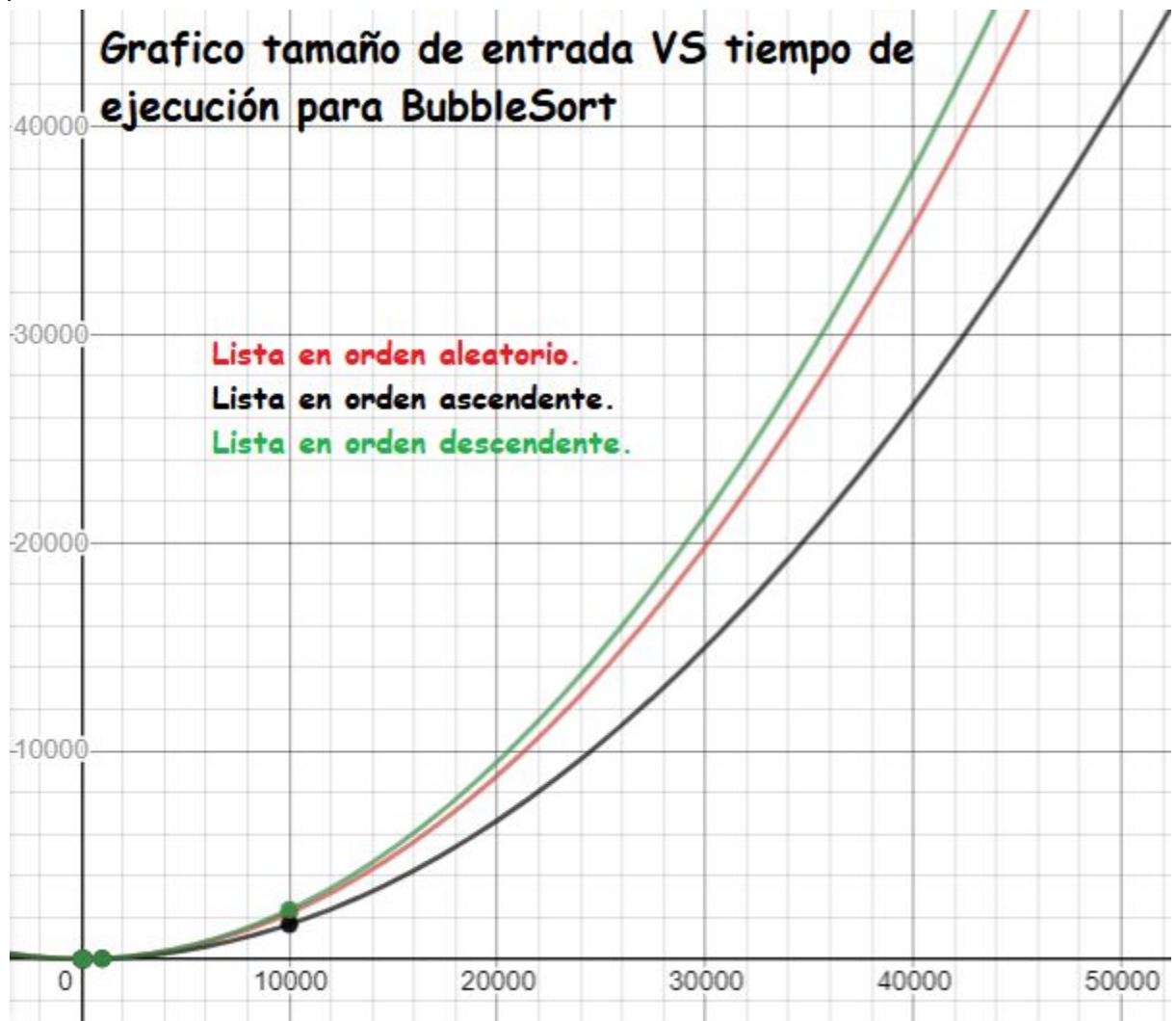
Complejidad espacial BubbleSort		
	Variable	Cantidad valores atómicos
Entrada	items	n
Salida		
Auxiliar	temp	1
	i	1
	j	1
		$O(n + 3)$

Complejidad temporal InsertionSort		
#	public static void InsertionSort<T>(List<T> items) where T : IComparable{	#veces que se repite
1	for (int i = 1; i < items.Count; ++i) {	n
2	T it = items[i];	$n - 1$
3	int j = i - 1;	$n - 1$
4	while (j >= 0 && it.CompareTo(items[j]) < 0){	$\frac{3n*(n+1)}{2} + 1$
5	items[j + 1] = items[j];	
6	--j;	
7	}items[j+1] = it;	$n - 1$
8	}	0
9	}	0
		$O(\frac{11n+3n^2}{2} - 2) = O(n^2)$

Complejidad espacial InsertionSort		
	Variable	Cantidad valores atómicos
Entrada	items	n
Salida		
Auxiliar	i	1
	it	1
	j	1
		$O(n + 3)$

Comparación de los resultados obtenidos en el experimento con el análisis de complejidades.

Los siguientes son gráficos de tamaño de entrada VS tiempo (La variable de respuesta medida en milisegundos) tanto para BubbleSort como para InsertionSort. Para construirlos sacamos la media de cada tratamiento y la usamos como dato representativo para un algoritmo y orden de elementos fijos, luego utilizamos la calculadora gráfica en línea [Desmos](#) para realizar la regresión por mínimos cuadrados. En ellos podemos notar la naturaleza cuadrática de estos dos algoritmos de ordenamiento, lo cual concuerda con los análisis de complejidad temporal presentados anteriormente.



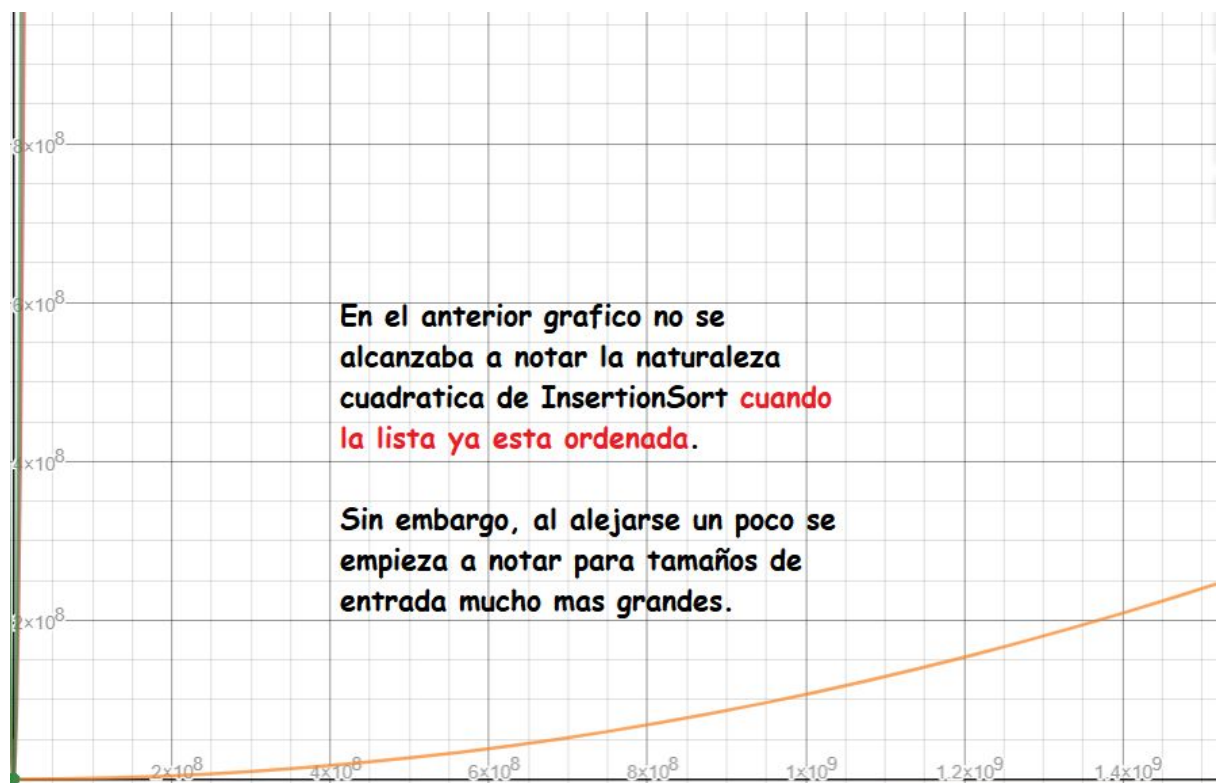
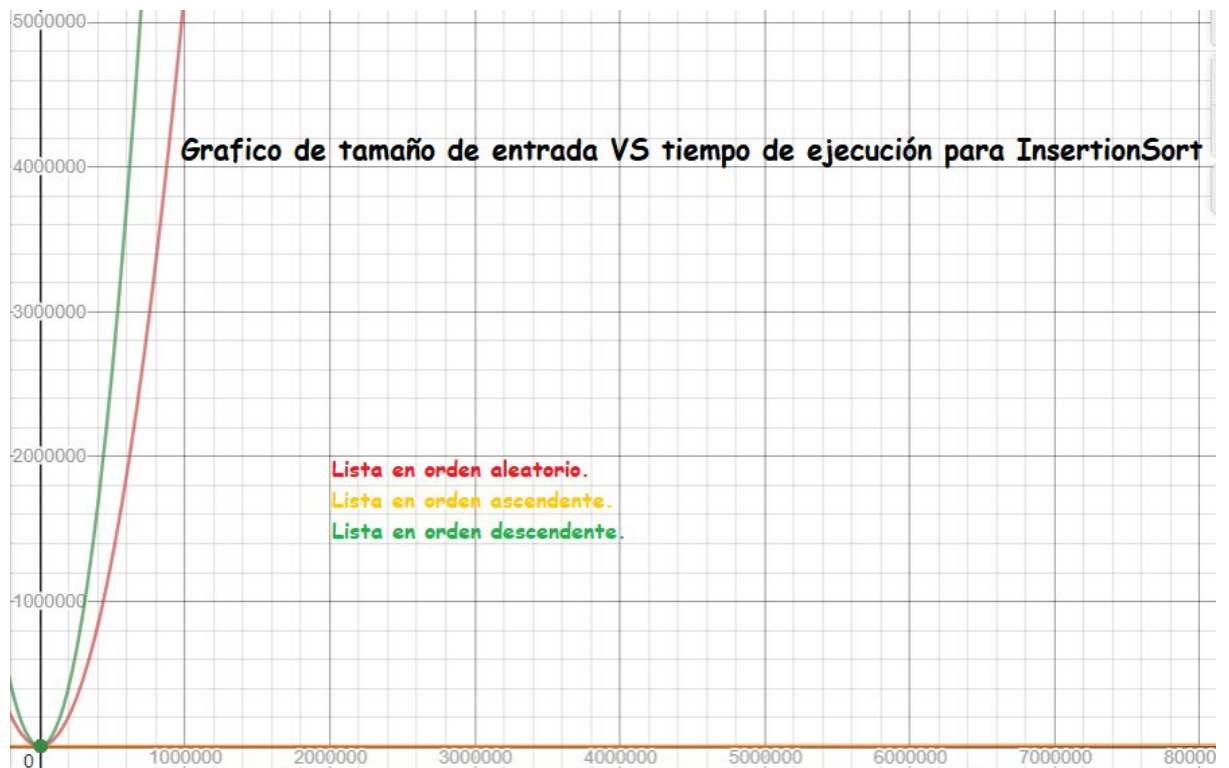


Diagrama de Clases



Diseño de las pruebas

Configuración de los Escenarios

Nombre	Clase	Escenario
SetupStageAleatoryOrder10toThePowerOf1	SorterTest	Una lista de 10 enteros en orden aleatorio: ints = { 6, 8, 21, 5, 1, 4, 10, 23, 12, 21 }
SetupStageAscendingOrder10toThePowerOf1	SorterTest	Una lista de 10 enteros en orden ascendente: ints = { 1, 4, 5, 6, 8, 10, 12, 21, 21, 23 }
SetupStageDecreasingOrder10toThePowerOf1	SorterTest	Una lista de 10 enteros en orden descendente: ints = { 23, 21, 21, 12, 10, 8, 6, 5, 4, 1 }
SetupStageAleatoryOrder10toThePowerOf2	SorterTest	Una lista de 100 enteros en orden aleatorio.
SetupStageAscendingOrder10toThePowerOf2	SorterTest	Una lista de 100 enteros en orden ascendente.
SetupStageDecreasingOrder10toThePowerOf2	SorterTest	Una lista de 100 enteros en orden descendente.
SetupStageAleatoryOrder10toThePowerOf3	SorterTest	Una lista de 1000 enteros en orden aleatorio.
SetupStageAscendingOrder10toThePowerOf3	SorterTest	Una lista de 1000 enteros en orden ascendente.
SetupStageDecreasingOrder10toThePowerOf3	SorterTest	Una lista de 1000 enteros en orden descendente.
SetupStageAleatoryOrder10toThePowerOf4	SorterTest	Una lista de 10.000 enteros en orden aleatorio.

SetupStageAscendingOrder10toThePowerOf4	SorterTest	Una lista de 10.000 enteros en orden ascendente.
SetupStageDecreasingOrder10toThePowerOf4	SorterTest	Una lista de 10.000 enteros en orden descendente.

Diseño de Casos de Prueba

Objetivo de la Prueba: Verificar que los métodos de burbuja e inserción están ordenando de manera correcta, con una lista de 10 enteros en orden aleatorio

Clase	Método	Escenario	Valores de Entrada	Resultado
SorterTest	BubbleSortTestAleatoryOrder10toThePowerOf1	SetupStageAleatoryOrder10toThePowerOf1	ninguno	True Después de haber verificado todos los valores en orden aleatorio de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de burbuja.
SorterTest	InsertionSortTestAleatoryOrder10toThePowerOf1	SetupStageAleatoryOrder10toThePowerOf1	ninguno	True Después de haber verificado todos los valores en orden aleatorio de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de inserción.

Objetivo de la Prueba: Verificar que los métodos de burbuja e inserción están ordenando de manera correcta, con una lista de 10 enteros en orden ascendente

Clase	Método	Escenario	Valores de Entrada	Resultado
SorterTest	BubbleSortTestAscendingOrder10toThePowerOf1	SetupStageAscendingOrder10toThePowerOf1	ninguno	True Después de haber verificado todos los valores en orden ascendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de burbuja.
SorterTest	InsertionSortTestAscendingOrder10toThePowerOf1	SetupStageAscendingOrder10toThePowerOf1	ninguno	True Después de haber verificado todos los valores en orden ascendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de inserción.

Objetivo de la Prueba: Verificar que los métodos de burbuja e inserción están ordenando de manera correcta, con una lista de 10 enteros en orden descendente

Clase	Método	Escenario	Valores de Entrada	Resultado
SorterTest	BubbleSort TestDecreasingOrder10toThePowerOf1	SetupStage Decreasing Order10toThePowerOf1	ninguno	True Después de haber verificado todos los valores en orden descendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de burbuja.
SorterTest	InsertionSort TestDecreasingOrder10toThePowerOf1	SetupStage Decreasing Order10toThePowerOf1	ninguno	True Después de haber verificado todos los valores en orden descendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de inserción.

Objetivo de la Prueba: Verificar que los métodos de burbuja e inserción están ordenando de manera correcta, con una lista de 100 enteros en orden aleatorio

Clase	Método	Escenario	Valores de Entrada	Resultado
SorterTest	BubbleSort TestAleatoryOrder10toThePowerOf2	SetupStage AleatoryOrder10toThePowerOf2	ninguno	True Después de haber verificado todos los valores en orden aleatorio de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de burbuja.
SorterTest	InsertionSort TestAleatoryOrder10toThePowerOf2	SetupStage AleatoryOrder10toThePowerOf2	ninguno	True Después de haber verificado todos los valores en orden aleatorio de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de inserción.

Objetivo de la Prueba: Verificar que los métodos de burbuja e inserción están ordenando de manera correcta, con una lista de 100 enteros en orden ascendente

Clase	Método	Escenario	Valores de Entrada	Resultado
SorterTest	BubbleSort TestAscendingOrder10toThePowerOf2	SetupStage AscendingOrder10toThePowerOf2	ninguno	True Después de haber verificado todos los valores en orden ascendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de burbuja.
SorterTest	InsertionSort TestAscendingOrder10toThePowerOf2	SetupStage AscendingOrder10toThePowerOf2	ninguno	True Después de haber verificado todos los valores en orden ascendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de inserción.

Objetivo de la Prueba: Verificar que los métodos de burbuja e inserción están ordenando de manera correcta, con una lista de 100 enteros en orden descendente

Clase	Método	Escenario	Valores de Entrada	Resultado
SorterTest	BubbleSort TestDecreasingOrder10toThePowerOf2	SetupStage Decreasing Order10toThePowerOf2	ninguno	True Después de haber verificado todos los valores en orden descendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de burbuja.
SorterTest	InsertionSort TestDecreasingOrder10toThePowerOf2	SetupStage Decreasing Order10toThePowerOf2	ninguno	True Después de haber verificado todos los valores en orden descendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de inserción.

Objetivo de la Prueba: Verificar que los métodos de burbuja e inserción están ordenando de manera correcta, con una lista de 1000 enteros en orden aleatorio

Clase	Método	Escenario	Valores de Entrada	Resultado
SorterTest	BubbleSort TestAleatoryOrder10toThePowerOf3	SetupStage AleatoryOrder10toThePowerOf3	ninguno	True Después de haber verificado todos los valores en orden aleatorio de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de burbuja.
SorterTest	InsertionSort TestAleatoryOrder10toThePowerOf3	SetupStage AleatoryOrder10toThePowerOf3	ninguno	True Después de haber verificado todos los valores en orden aleatorio de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de inserción.

Objetivo de la Prueba: Verificar que los métodos de burbuja e inserción están ordenando de manera correcta, con una lista de 1000 enteros en orden ascendente

Clase	Método	Escenario	Valores de Entrada	Resultado
SorterTest	BubbleSort TestAscendingOrder10toThePowerOf3	SetupStage AscendingOrder10toThePowerOf3	ninguno	True Después de haber verificado todos los valores en orden ascendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de burbuja.
SorterTest	InsertionSort TestAscendingOrder10toThePowerOf3	SetupStage AscendingOrder10toThePowerOf3	ninguno	True Después de haber verificado todos los valores en orden ascendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de inserción.

Objetivo de la Prueba: Verificar que los métodos de burbuja e inserción están ordenando de manera correcta, con una lista de 1000 enteros en orden descendente

Clase	Método	Escenario	Valores de Entrada	Resultado
SorterTest	BubbleSort TestDecreasingOrder10toThePowerOf3	SetupStage Decreasing Order10toThePowerOf3	ninguno	True Después de haber verificado todos los valores en orden descendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de burbuja.
SorterTest	InsertionSort TestDecreasingOrder10toThePowerOf3	SetupStage Decreasing Order10toThePowerOf3	ninguno	True Después de haber verificado todos los valores en orden descendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de inserción.

Objetivo de la Prueba: Verificar que los métodos de burbuja e inserción están ordenando de manera correcta, con una lista de 10000 enteros en orden aleatorio

Clase	Método	Escenario	Valores de Entrada	Resultado
SorterTest	BubbleSort TestAleatoryOrder10toThePowerOf4	SetupStage AleatoryOrder10toThePowerOf4	ninguno	True Después de haber verificado todos los valores en orden aleatorio de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de burbuja.
SorterTest	InsertionSort TestAleatoryOrder10toThePowerOf4	SetupStage AleatoryOrder10toThePowerOf4	ninguno	True Después de haber verificado todos los valores en orden aleatorio de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de inserción.

Objetivo de la Prueba: Verificar que los métodos de burbuja e inserción están ordenando de manera correcta, con una lista de 10000 enteros en orden ascendente

Clase	Método	Escenario	Valores de Entrada	Resultado
SorterTest	BubbleSort TestAscendingOrder10toThePowerOf4	SetupStage AscendingOrder10toThePowerOf4	ninguno	True Después de haber verificado todos los valores en orden ascendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de burbuja.
SorterTest	InsertionSort TestAscendingOrder10toThePowerOf4	SetupStage AscendingOrder10toThePowerOf4	ninguno	True Después de haber verificado todos los valores en orden ascendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de inserción.

Objetivo de la Prueba: Verificar que los métodos de burbuja e inserción están ordenando de manera correcta, con una lista de 10000 enteros en orden descendente

Clase	Método	Escenario	Valores de Entrada	Resultado
SorterTest	BubbleSort TestDecreasingOrder10toThePowerOf4	SetupStage Decreasing Order10toThePowerOf4	ninguno	True Después de haber verificado todos los valores en orden descendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de burbuja.
SorterTest	InsertionSortTestDecreasingOrder10toThePowerOf4	SetupStage Decreasing Order10toThePowerOf4	ninguno	True Después de haber verificado todos los valores en orden descendente de la lista, se encontró que esta ordenando de manera correcta de acuerdo con el método de inserción.