UNIVERSIDAD

Panamericana

Proyecto 2do Parcial

Estructuras de Datos y Algoritmos

Maestro: Ricardo Tachiquín Gutiérrez

Dahlia Divana Jimenez Saldaña 026872

Andre NIcasio Romo 0247456

Aguascalientes,Aguascalientes, a 21 de abril de 2024

Documentación del uso de la GUI

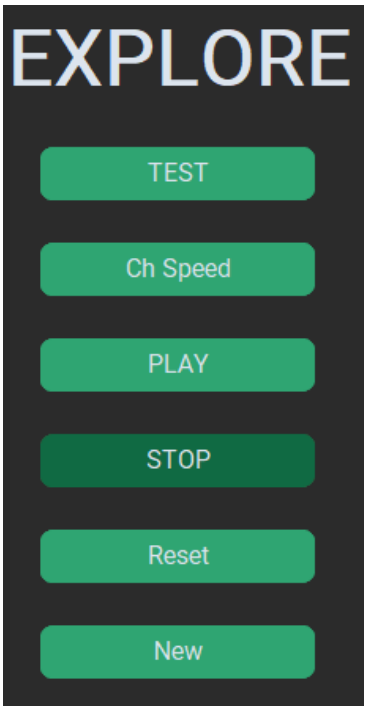
La aplicación que generamos es una interfaz gráfica que nos permite observar el comportamiento de ordenamiento en tiempo real de una lista de valores de enteros positivos generados aleatoriamente sin repetirlos, donde el usuario puede ingresar la cantidad de valores que quiere ingresar desde 10 a 1000.

La interfaz gráfica muestra en el lado izquierdo un menú donde aparece la opción de TEST, CH SPEED, PLAY, STOP, RESET, NEW, a un lado del menu aparecera la pagina general de EXPLORE donde podrá observar 2 sliders uno que es la CANTIDAD DE VALORES que quiere ingresar y el otro para asignar la VELOCIDAD , en la esquina superior derecha se encuentra el grupo de Radiobuttons donde podrá escoger cual METODO DE ORDENAMIENTO quiere escoger, podrá elegir entre "Bubble", "InsertionSort", "Selection Sort", "Counting Sort", "Shell Sort", "Heap Sort", "QuickSort", "MergeSort", "Radix Sort", "TimSort", "Cocktail Shaker". Bajo los sliders podrás observar la gráfica que se generará al ingresar los datos ingresados.

La ventana TEST se podrá abrir al presionar este botón , aquí podrás elegir entre generar una lista de 2000 valores enteros positivos aleatorios con el botón RANDOM y la opción de generar los mismos 2000 valores enteros positivos ordenados inversamente con el botón INVORD. Además encontrarás 2 botones uno para INICIAR y otro para CERRAR la ventana de TEST, al presionar INICIAR aparecerá una caja de texto donde se muestra el análisis de cada uno de los algoritmos, para que tu puedas saber el tipo de algoritmo la complejidad que tiene y cuanto tarda en hacer el ordenamiento de los 2000 valores ingresados. De igual manera se podrá observar en una curva de rendimiento cada uno de los métodos.

EXPLORE

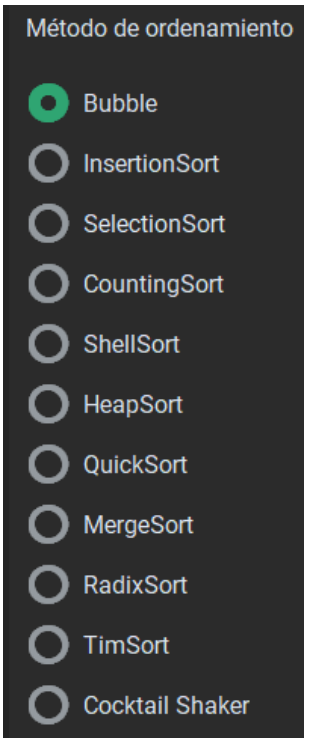
En la pestaña principal se mostrará un menú del lado izquierdo, donde se muestra hasta arriba el título EXPLORE, esto nos ayudará a identificar en qué pestaña estamos, abajo del título se mostrarán una serie de botones que tienen distintas funciones, se



muestra el botón TEST el cual nos dirigirá a una nueva ventana que posteriormente explicaremos, despues esta CH Speed esta nos permitirá cambiar la velocidad con la que se ordena el algoritmo sin pausar o resetear o generar un lista de valores nuevos, despues esta el PLAY esta nos permite que comience el ordenamiento de los valores,pero antes de esto tendras que presionar NEW para que el programa genere la nueva lista de los valores aleatorios, abajo de este botón está el STOP con el cual podrás parar el ordenamiento para asi poder, si así lo deseas, cambiar el método de ordenamiento o generar otra lista, o simplemente pausar y volver a reanudarlo otra vez, después está RESET con este reseteamos la lista generada para volver a establecer todos los parámetros después , claramente primero se tiene que poner en pausa el programa para poder resetearlo, y por último NEW con este podrás generar una nueva lista de valores aleatorios, primero tendrás que pausar el ordenamiento para posteriormente presionar NEW para generar una nueva lista.

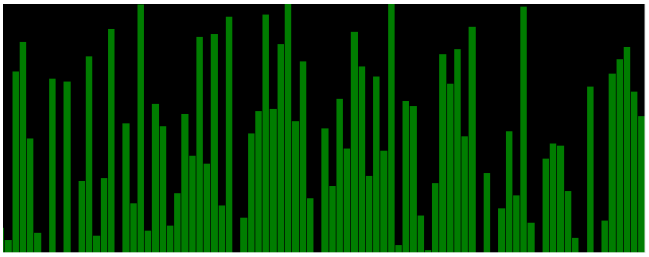
Al lado del menú se mostrarán todas las funciones que tiene EXPLORE, se muestran 2 sliders uno para cambiar la cantidad de valores que quieres ordenar , puedes ordenar desde 1 a 1000 valores y el otro es para ajustar la velocidad.

En el lado derecho se muestran una serie de radiobuttons por medio de los cuales puedes seleccionar un método de ordenamiento entre los 11 que se muestran ahí, son los siguientes:

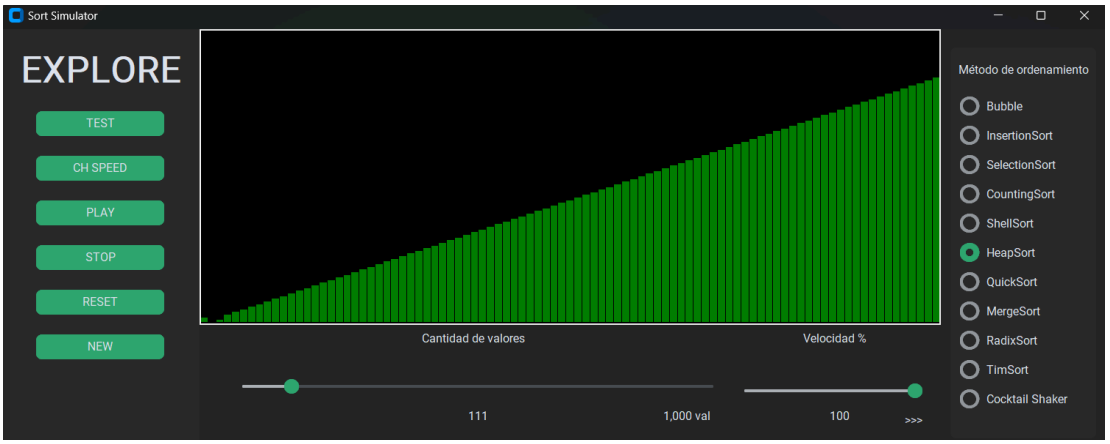


- **Bubble Sort**: Recorre repetidamente la lista, comparando elementos adyacentes y los intercambia si están en el orden incorrecto. Continúa este proceso hasta que no se requieran más intercambios.
- **Insertion Sort**: Recorre la lista y toma un elemento cada vez, colocándolo en la posición correcta respecto a los elementos que ya están ordenados.
- **Selection Sort**: Busca el elemento más pequeño de la lista y lo intercambia con el primer elemento. Luego busca el segundo elemento más pequeño y lo intercambia con el segundo lugar, y así sucesivamente.
- **Counting Sort**: Cuenta el número de elementos distintos en la lista y utiliza esa información para ordenar los elementos en tiempo lineal.
- **Shell Sort**: Es una extensión del algoritmo de inserción que divide la lista en sublistas más pequeñas y ordena esas sublistas utilizando inserción. Gradualmente, reduce el tamaño de las sublistas hasta que toda la lista está ordenada.
- **Heap Sort**: Convierte la lista en un montículo (heap) y luego extrae repetidamente el elemento máximo (para ordenar de manera ascendente) del montículo, reconstruyendo el montículo después de cada extracción.
- **Quick Sort:** Elige un elemento como pivote y divide la lista alrededor del pivote de manera que los elementos menores que el pivote estén a su izquierda y los mayores a su derecha. Luego, repite este proceso recursivamente para las sublistas generadas.
- **Merge Sort:** Divide la lista en sublistas más pequeñas, las ordena recursivamente y luego combina las sub listas ordenadas para obtener la lista completa ordenada.

- **Radix Sort**: Clasifica los elementos comparando los dígitos de las representaciones numéricas o alfabéticas. Comienza clasificando los elementos menos significativos y avanza hacia los más significativos.
- **Tim Sort:** Es una combinación del mergesort y el insertion sort. Divide la lista en subarreglos pequeños, los ordena usando insertion sort y luego los fusiona usando merge sort.
- **Cocktail Shaker Sort:** Es una variante del Bubble Sort que también se conoce como Shaker Sort o Bidirectional Bubble Sort. Funciona de manera similar al Bubble Sort, pero en cada pasada recorre la lista en ambas direcciones, intercambiando elementos si es necesario.

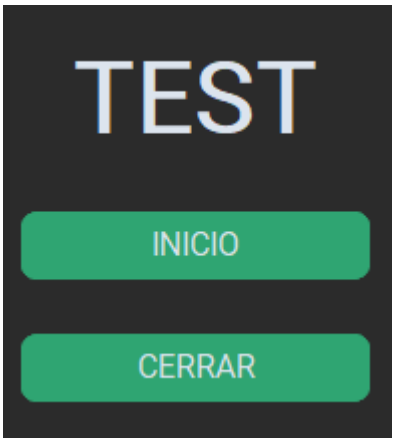


Al lado de esto se mostrará una gráfica donde se reflejarán todos los valores desordenados y podrás observar cómo se van ordenando. Para esto primero tienes que seleccionar la cantidad de valores que deseas generar, esto lo podras hacer el el slider que dice “Cantidad de valores” una vez que tengas esto puedes establecer la velocidad con la que se quieren ordenar los datos en el slider “Velocidad”, y por último selecciona uno de los 11 métodos, para finalizar y comenzar a ver el ordenamiento de los valores debes presionar PLAY y podrás comenzar a ver cómo se ordenan en tiempo real. Para poder observar todos los datos deberás de ampliar la ventana a pantalla completa.

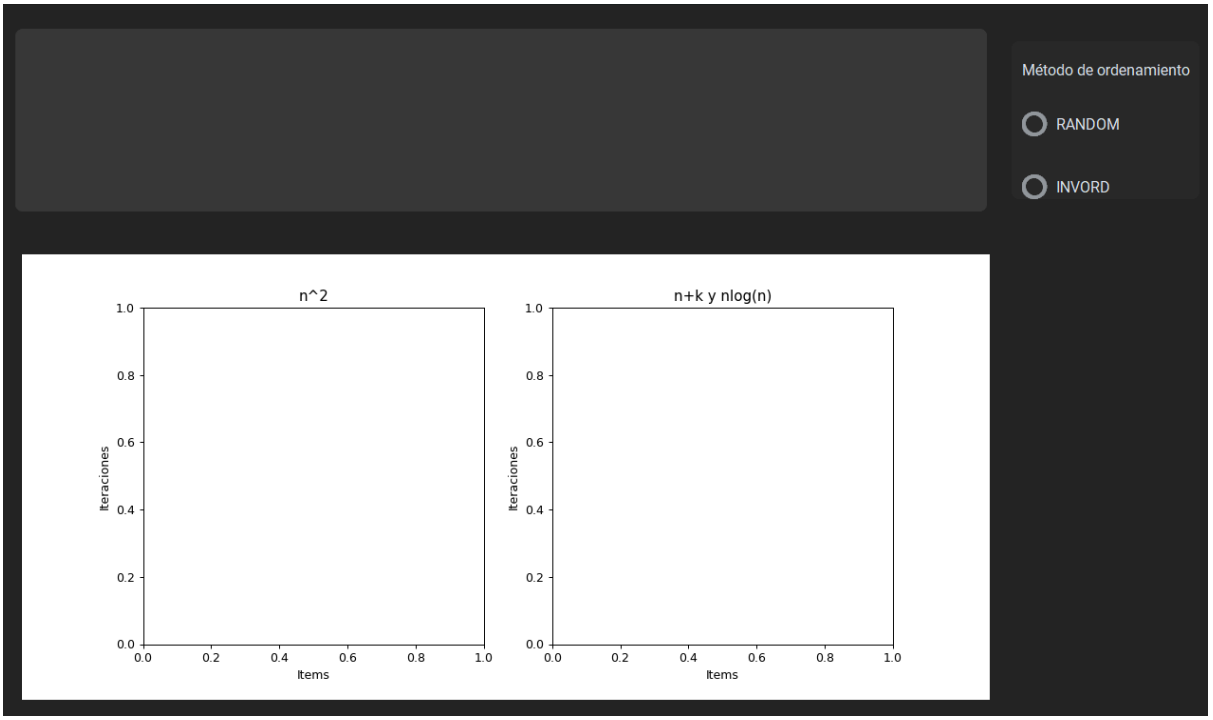


TEST

Como habíamos comentado anteriormente se encontraba un botón TEST en el menú de la

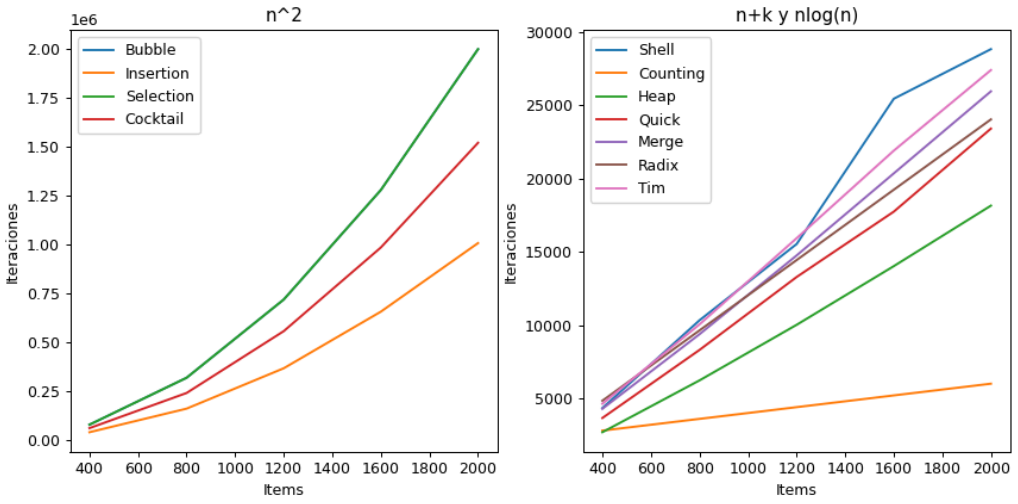


ventana principal, al presionar dicho botón el programa desplegará una nueva ventana donde se podrán observar otros componentes y podrás realizar distintas funciones. se mostrará del lado izquierdo el título de la ventana TEST y abajo de esto se mostrarán 2 botones, INICIO y CERRAR. Al lado derecho se muestra un cuadro de texto y hasta la derecha se muestran 2 radiobuttons, RANDOM que con este podrás generar una lista de 2000 valores aleatorios enteros positivos y con el otro InvOrd que este generará 2000 valores enteros positivos ordenados de manera inversa (2000,1999,1998…). y abajo se mostrará una gráfica. Se verá de la siguiente manera…

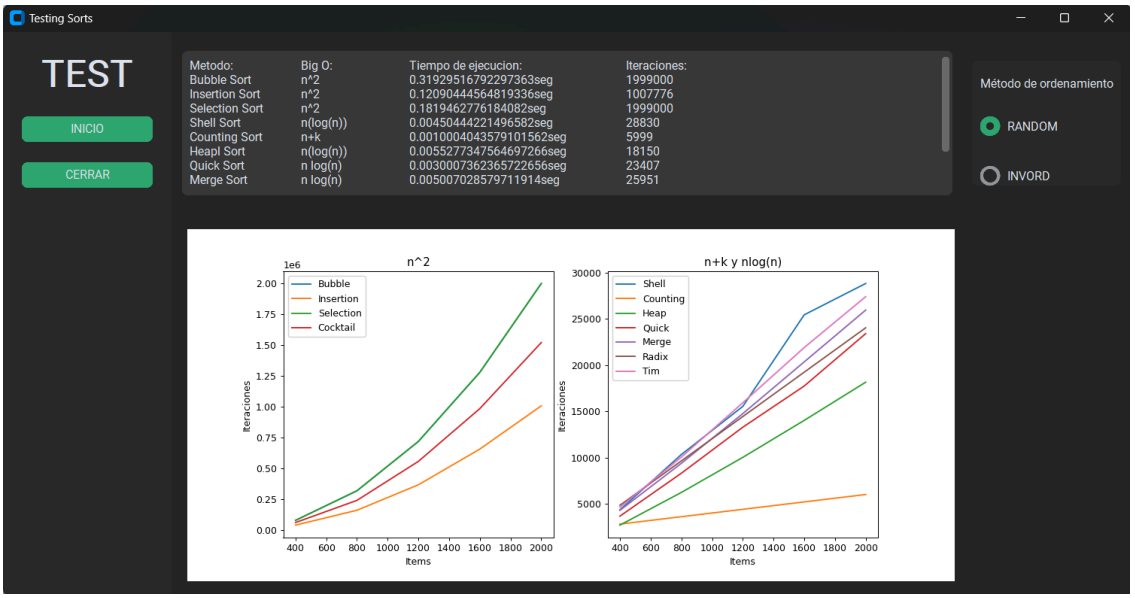


Para que funcione esta ventana tendremos que seleccionar una opción ya sea RANDOM o InvOrd una vez que seleccione una deberá presionar INICIAR para que se muestran los valores en la caja de texto y la curva de rendimiento de todos los métodos.

Metodo:	Big O:	Tiempo de ejecucion:
Bubble Sort	n^2	0.3317198753356933 6seg
Insertion Sort	n^2	0.1178205013275146 5seg
Selection Sort	n^2	0.1796698570251464 8seg
Shell Sort	n(log(n))	0.0040011405944824 22seg
Counting Sort	n+k	0.0010004043579101 562seg
Heapl Sort	n(log(n))	0.0070013999938964 84seg



Una vez que acabes de usar esta ventana podrás presionar CERRAR y automáticamente se cerrará y solo podrás ver las ventana de EXPLORE. De igual manera para terminar el programa podrás presionar la cruz que se encuentra en la esquina superior derecha de la pantalla para cerrar todo.



Características de los algoritmos

1. **Bubble Sort:**

   Descripción: Este algoritmo de ordenamiento compara pares de elementos adyacentes y los intercambia si están en el orden incorrecto. Este proceso se repite hasta que no se necesiten más intercambios.

   Implementación: Cuando se selecciona "Bubble" en la interfaz gráfica y se presiona el botón "PLAY", se activa este algoritmo.

2. **Insertion Sort:**

   Descripción: Insertion Sort es un algoritmo de ordenamiento simple que construye una matriz ordenada uno por uno, comparando cada elemento con los elementos adyacentes y colocándolos en su lugar adecuado.

   Implementación: Cuando se selecciona "InsertionSort" en la interfaz gráfica y se presiona el botón "PLAY", se activa este algoritmo.

3. **Selection Sort:**

   Descripción: Selection Sort divide la lista en dos partes: una parte ordenada y una parte desordenada. Encuentra el elemento más pequeño de la parte desordenada y lo coloca al principio de la parte ordenada.

   Implementación: Cuando se selecciona "SelectionSort" en la interfaz gráfica y se presiona el botón "PLAY", se activa este algoritmo.

4. **Counting Sort:**

   Descripción: Counting Sort es un algoritmo de ordenamiento eficiente para clasificar elementos cuyo rango es conocido previamente. Cuenta el número de elementos con valores distintos y calcula la posición de cada elemento en el resultado.

   Implementación: Cuando se selecciona "CountingSort" en la interfaz gráfica y se presiona el botón "PLAY", se activa este algoritmo.

5. **Shell Sort:**

   Descripción: Shell Sort mejora Insertion Sort al comparar elementos distantes entre sí en lugar de elementos adyacentes. Utiliza una secuencia de brecha decreciente para realizar múltiples pasadas de ordenamiento.

   Implementación: Cuando se selecciona "ShellSort" en la interfaz gráfica y se presiona el botón "PLAY", se activa este algoritmo.

6. **Heap Sort:**

   Descripción: Heap Sort convierte la lista en un montículo (heap), luego extrae el elemento máximo (o mínimo) repetidamente para obtener una lista ordenada. Es eficiente y no requiere de espacio adicional.

   Implementación: Cuando se selecciona "HeapSort" en la interfaz gráfica y se presiona el botón "PLAY", se activa este algoritmo.

7. **Quick Sort:**

   Descripción: Quick Sort utiliza el método de partición para dividir la lista en subconjuntos, luego ordena recursivamente cada subconjunto. Es rápido y eficiente para listas grandes.

   Implementación: Cuando se selecciona "QuickSort" en la interfaz gráfica y se presiona el botón "PLAY", se activa este algoritmo.

8. **Merge Sort:**

   Descripción: Merge Sort divide la lista en mitades, ordena cada mitad por separado y luego fusiona las mitades ordenadas para obtener la lista final ordenada. Es eficiente y estable.

   Implementación: Cuando se selecciona "MergeSort" en la interfaz gráfica y se presiona el botón "PLAY", se activa este algoritmo.

9. **Radix Sort:**

   Descripción: Radix Sort clasifica los elementos basándose en los dígitos individuales de los números. Es especialmente eficiente para clasificar números enteros.

   Implementación: Cuando se selecciona "RadixSort" en la interfaz gráfica y se presiona el botón "PLAY", se activa este algoritmo.

10. **Tim Sort:**

    Descripción: Tim Sort es una variante híbrida de Merge Sort y Insertion Sort. Utiliza la estrategia de "ordenamiento por inserción" en subconjuntos pequeños y luego aplica "ordenamiento por mezcla" para combinar estos subconjuntos ordenados.

    Implementación: Cuando se selecciona "TimSort" en la interfaz gráfica y se presiona el botón "PLAY", se activa este algoritmo.

11. **Cocktail Shaker Sort:**

    Descripción: Cocktail Shaker Sort (o Bidirectional Bubble Sort) es una variante de Bubble Sort que permite el ordenamiento en ambas direcciones, reduciendo el número de pasadas.

    Implementación: Cuando se selecciona "CocktailShaker" en la interfaz gráfica y se presiona el botón "PLAY", se activa este algoritmo.

Consideraciones

1. Interfaz Gráfica (GUI): La aplicación proporciona una interfaz gráfica intuitiva que permite a los usuarios observar el comportamiento del ordenamiento en tiempo real de una lista de valores enteros positivos generados aleatoriamente. La GUI incluye elementos como menús, botones, sliders y gráficos para facilitar la interacción y visualización de los datos.

2. Funcionalidades principales:

- Exploración (EXPLORE): Permite ajustar la cantidad de valores a ordenar, la velocidad del proceso y la selección del algoritmo de ordenamiento. Los usuarios pueden iniciar, detener y reiniciar el ordenamiento, así como generar nuevas listas de valores.
- Pruebas (TEST): Ofrece la opción de generar una lista de valores aleatorios o inversamente ordenados para evaluar el rendimiento de los algoritmos de ordenamiento. Proporciona análisis detallado y curvas de rendimiento.

3. Selección de algoritmos: La aplicación ofrece una amplia gama de algoritmos de ordenamiento, desde los clásicos como Bubble Sort hasta opciones más avanzadas como Tim Sort. Esto permite a los usuarios comparar y analizar el rendimiento de diferentes enfoques de ordenamiento.

4. Análisis de complejidad y rendimiento: Se proporciona información detallada sobre cada algoritmo, incluyendo su descripción, implementación y características. Además, se ofrece un análisis de la complejidad temporal y espacial de cada algoritmo, así como su rendimiento en la clasificación de una lista de 2000 valores.

5. Facilidad de uso: La interfaz gráfica está diseñada de manera que sea fácil de entender y utilizar para usuarios de diferentes niveles de habilidad. Se proporciona retroalimentación visual en tiempo real durante el proceso de ordenamiento para una experiencia interactiva.

Conclusiones

La aplicación ofrece una amplia variedad de opciones y configuraciones que permiten a los usuarios explorar y comprender el funcionamiento de diferentes algoritmos de ordenamiento en tiempo real. Esto se complementa con la inclusión de descripciones completas de los algoritmos, junto con análisis detallados de su complejidad y rendimiento. Esta información proporciona a los usuarios una comprensión profunda de cada método de ordenamiento, lo que la convierte en una herramienta útil tanto para estudiantes como para profesionales.

Además de su utilidad educativa en entornos académicos, la aplicación también se destaca como una herramienta de evaluación. La capacidad de generar listas de valores aleatorios e inversamente ordenados para pruebas, junto con el análisis de rendimiento proporcionado, permite a los usuarios evaluar y comparar diferentes algoritmos de ordenamiento en condiciones controladas. Esta funcionalidad es invaluable para aquellos que desean comprender mejor las fortalezas y debilidades de cada método de ordenamiento en diferentes escenarios.

Aunque la aplicación ya ofrece una amplia funcionalidad, hay potencial para mejoras adicionales. Por ejemplo, se podrían incluir más opciones de configuración para adaptarse a las necesidades específicas de los usuarios, y se podría trabajar en la optimización del rendimiento para manejar conjuntos de datos más grandes de manera eficiente. Estas mejoras podrían hacer que la aplicación sea aún más versátil y útil para una variedad de propósitos educativos y profesionales.

Participación

Ambos participamos en la idea del diseño gráfico de la aplicación , al igual que las decisiones de los tipos de gráficas y estructuras para el programa, Dahlia se enfocó principalmente en el desarrollo de la interfaz gráfica y en el acomodo de los componentes además del desarrollo de todo el documento y la explicación de cada uno de los algoritmos que se utilizaron en el programa de ordenamiento, mientras que André desarrollo las funciones de cada uno de estos componentes para que el código fuese funcional.