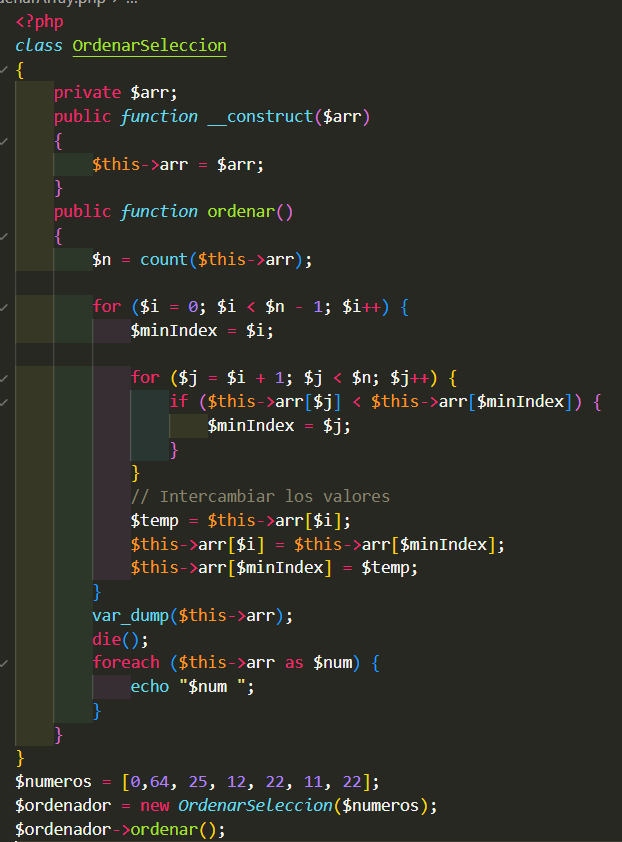
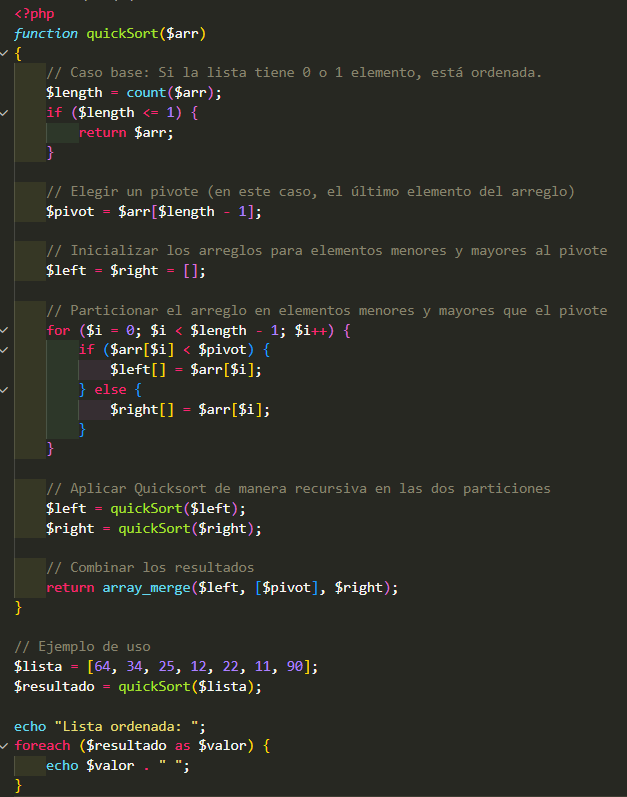
¿Qué es un algoritmo de selección?

Los algoritmos de selección son un tipo de algoritmo de ordenamiento utilizado en programación para organizar un conjunto de elementos en un orden específico. Su objetivo principal es encontrar el elemento más pequeño (o más grande) en una lista no ordenada y colocarlo en la posición correcta, de modo que los elementos restantes estén en orden ascendente (o descendente) después de cada iteración. Estos algoritmos son útiles cuando se necesita ordenar un conjunto de datos o encontrar el elemento más extremo en una lista.

Hay varios algoritmos de selección comunes, pero el más básico es el llamado "Selección por mínimos" (Selection Sort en inglés), que sigue un proceso iterativo para encontrar y colocar el elemento más pequeño en la posición adecuada en cada iteración. Aquí hay una descripción general de cómo funciona el algoritmo de selección:

* Inicialmente, considera toda la lista como no ordenada.
* Encuentra el elemento mínimo en la lista no ordenada.
* Intercambia el elemento mínimo con el primer elemento de la lista no ordenada.
* Ahora, la primera posición de la lista está en orden.

Repite los pasos 1-4 para la lista restante (sin incluir el elemento ya ordenado) hasta que toda la lista esté ordenada.

El algoritmo de selección es simple de implementar, pero no es muy eficiente en términos de tiempo de ejecución en comparación con otros algoritmos de ordenamiento como Quicksort o Mergesort. Tiene una complejidad de tiempo promedio de O(n^2), lo que significa que su rendimiento puede ser lento en listas grandes. Sin embargo, puede ser útil en situaciones en las que la simplicidad de la implementación es más importante que la eficiencia, o cuando se trabaja con listas pequeñas.

Quicksort es un algoritmo de ordenamiento eficiente y ampliamente utilizado en programación. Funciona dividiendo una lista en subconjuntos más pequeños y ordenándolos de manera recursiva. El algoritmo se basa en el principio de "dividir y conquistar", donde divide la lista en partes más pequeñas, ordena esas partes y luego las combina para obtener la lista ordenada completa.

La función **quickSort** implementa el algoritmo de Quicksort de manera recursiva. Aquí está cómo funciona:

* Se verifica si la longitud del arreglo es igual a 0 o 1. Si es así, el arreglo se considera ordenado, y se devuelve tal cual.
* Se elige un pivote, que en este caso es el último elemento del arreglo.
* Se crean dos arreglos, $left y $right, para almacenar los elementos menores y mayores que el pivote.
* Se recorre el arreglo original y se distribuyen los elementos en los arreglos $left y $right según si son menores o mayores que el pivote.
* Se aplica Quicksort de manera recursiva a los arreglos $left y $right.
* Finalmente, se combina el resultado: se une el arreglo ordenado $left, el pivote y el arreglo ordenado $right para obtener la lista ordenada completa.

El desarrollo de los algoritmos se deben hacer usando POO

1. Implemente los algoritmos mostrados anteriormente.

|  |
| --- |
|  |

1. Crea una clase que calcule el Máximo Común Divisor (MCD) de dos números enteros utilizando el algoritmo de Euclides. Luego, muestra el resultado en pantalla.  
   [¿Qué es el MCD?](https://www.youtube.com/watch?v=WD4rGWCRBYY) – [Algoritmo de Euclides](https://www.youtube.com/watch?v=rG9nQe-7is8)

|  |
| --- |
|  |

1. Suba en Git y anexe los proyectos de la guía en un archivo .zip documentos requeridos.

|  |
| --- |
|  |

1. Subir la evidencia al LMS