**PRÁC****TICA DE LABORATORIO Nro. 3**

*Carrera Computación*

|  |  |  |
| --- | --- | --- |
| **A. DATOS INFORMATIVOS** | | |
| **Asignatura:**  Estructura de datos | **Ciclo / Semestre:**  Tercero | **Paralelo:**  A |
| **Docente:**  José Oswaldo Guamán Quinche | **Período Académico:**  Octubre 2023 - Abril 2024 | |
| **Integrantes:** | | |

|  |  |
| --- | --- |
| **B. INFORMACIÓN GENERAL** | |
| **Unidad:**  ALGORITMOS DE BÚSQUEDA, ORDENACIÓN | |
| **Tema:**  Implementación de método de búsqueda por listas | |
| **Fecha:** Loja, \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 | **Nro. horas:** 6 |
| **Objetivos:** | |
| * Implementar el método MergeSort con la clase lista usando objetos * Implementar el método quickshort con la clase lista usada objetos | |
| **Corresponde al resultado de aprendizaje:** | |
| R3. Aplica los métodos de ordenación y búsqueda en la resolución de problemas, bajo los principios de solidaridad, transparencia, responsabilidad y honestidad. | |
| **Recursos y/o materiales:** | |
| * Computador. * IDE de programación. * Lenguaje de programación java * Material bibliográfico o recurso indicado en el EVA. | |

|  |
| --- |
| **C. DESARROLLO** |
| **Instrucciones:** |
| **Primera parte**  Implementar el algoritmo de ordenación tanta de MergeSort como de Quicksort con Listas usando objetos (tal como se vio en clases), no usar librerías externas ni las clases propias del lenguaje de programación. Se debe usar la práctica 2, luego comprobar. Luego hacer lo siguiente:  1.- Tener guardado mínimo 20 registros  2.- Mostrar los datos en la ventana. (Haga uso de modelo de tablas)  3.- Permitir ordenar por método quickshort y MergeSort  4.- El sistema debe permitir escoger si el ordenamiento es ascendente o descendente.  5.- El sistema debe permitir escoger por el criterio de ordenamiento (minimo 4).  6.- Mostrar en la tabla la lista ordenada y medir el tiempo de cada algoritmo e ir documentando  **Nota:** Todos los códigos deben guardarse en un repositorio git (gitlab, github o bitbucket) |
| **Resolución:** |
| Toda lo relacionado con la parte de instrucciones, se debe ubicar fragmentos de código y demostraciones en caso que sea necesaria. |
| **Conclusiones:** |
|  |

|  |  |
| --- | --- |
| **D. RÚBRICA DE EVALUACIÓN** | |
| **Nota: En caso de no cumplir con alguno de los parámetros establecidos se calificará la nota igual a 0**  **Si se encuentra copia con algún compañero o prácticas realizadas de otros años, o bajados del internet, se aplicará el reglamento de deshonestidad estudiantil y se calificará sobre 0.**  **No se aceptará trabajos atrasados ni incompletos, se calificará sobre 0.**  **Todo acerca de deshonestidad académica que no diga este documento.** | |
| **Informe de trabajo:**   * Contenido: pertinente y concreto; (obligatorio) * Estructura y organización: Elementos vinculados y estructurados coherentemente. (obligatorio) * Originalidad y creatividad: trabajo inédito, presentación de nuevas ideas. | 2 ptos |
| **Resolución de Ejercicios:**   * Proceso de resolución de ejercicios: con originalidad y creatividad usando lo aprendido en clases (obligatorio) * Ejecución de programas y solución de la aplicación (debe estar todo solucionado al 100% obligatorio) | 6 ptos |
| **Conclusiones:**   * Redacción * Originalidad y creatividad: conclusiones inéditas en base a su experiencia y objetivos planteados (obligatorio). | 1 ptos |
| **Video**   * Se debe entregar un video con la demostración y explicando el código de su solución (obligatorio). | 1 ptos |
| **Total**  **(Ponderado en calificación final 25%)** | 10 ptos |

|  |  |
| --- | --- |
| **E. FIRMAS DE RESPONSABILIDAD DE LO ACTUADO** | |
| **Estudiante(s):** | **Firma** |
|  |  |