

**UNIVERSIDAD DE LIMA**

**CARRERA DE INGENIERÍA DE SISTEMAS**

**PLAN DE TRABAJO**

**Curso: 1492 - ESTRUCTURA DE DATOS Y ALGORITMOS** **Semestre: 2023-2**

**Outcome 3: Habilidad para comunicarse efectivamente con una variedad de audiencias**

**Indicador 3.1: Comunicar efectivamente en forma escrita para una variedad de audiencias, respetando las normas aplicables al a elaboración de documentos académicos**

| **Semana** | **Avance de la asignatura**  **(Sílabo)** | **Actividades** | **Evaluaciones (Prácticas, Exámenes, Trabajos)** | **Criterio ABET** |
| --- | --- | --- | --- | --- |
| **01** | Concepto de algoritmo. Notación algorítmica. Clases, objetos y métodos. Estructuras de datos. | Actividades:   * Presentar el sílabo, plan y sistema de evaluación * Explicar conceptos de algoritmos. * Explicar la instalación de JDK 8 y Netbeans 8.2 * Explicar la notación algorítmica con ejemplos. * Explicar los fundamentos básicos de un TDA (Tipo de Dato Abstracto) * Explicar los conceptos de Clases, Métodos y Objetos   Tareas   * Ejercicios propuestos sobre notación algorítmica y POO - Semana 1   Recursos:   * Instaladores de JDK 8 y Netbeans 8.2 * Presentación de diapositivas - Semana-01 * Desarrollo de ejercicios de la guía N02. |  |  |
| **02** | Análisis de algoritmos de forma teórica y experimental. Complejidad del peor caso y del mejor caso. Complejidad en tiempo y en espacio. Notación O grande. | Actividades:   * Captar y motivar a los estudiantes sobre el análisis de complejidad de algoritmos. * Explicar las diferencias entre el análisis de algoritmo teórica y empírica. * Identificar el tamaño del problema y operaciones fundamentales en un algoritmo. * Explicar la necesidad de conocer el peor, promedio y mejor caso en el análisis de complejidad de un algoritmo. * Identificar la notación asintótica de la complejidad de un algoritmo.   Tareas:   * Ejercicios propuestos sobre análisis de complejidad de algoritmos   Recursos:   * Presentación de diapositivas - Semana-02 * Desarrollo de ejercicios de la guía N02. |  |  |
| **03** | Algoritmos elementales para arreglos unidimensionales: inserción, búsqueda, eliminación.  Algoritmos de búsqueda: búsqueda secuencial y binaria: Análisis de complejidad | Actividades:   * Conocer saberes previos sobre el uso de arreglos * Desarrollar la implementación de la estructura de datos arreglo con sus operaciones básicas (agregar, insertar, eliminar, i-ésimo, ubicación, mostrar) y extendidas (max, min. moda). * Explicar la diferencia entre el algoritmo de búsqueda secuencial y binaria. * Identificar la diferencia de la complejidad de algoritmos de búsqueda secuencial y binaria. * Implementar una estructura de datos arreglo con datos ordenados   Tareas:   * Ejercicios propuestos sobre uso de la estructura de datos arreglo.   Recursos:   * Presentación de diapositivas - Semana-03 * Desarrollo de ejercicios de la guía N03 |  |  |
| **04** | Algoritmos de ordenamiento y su análisis: Método de la Burbuja, Inserción y Selección. Prueba de corrección. Arreglos bidimensionales. | Actividades:   * Conocer saberes previos sobre ordenar elementos de un arreglo. * Explicar los algoritmos de ordenacion iterativos Burbuja, Inserción y Selección. * Analizar la complejidad de algoritmos iterativos de ordenación. * Implementar algoritmos basado en la estructura de datos Matriz o Arreglo Bidimensional. Operaciones de recorrido por filas y columnas.   Tareas:   * Ejercicios propuestos sobre uso de la estructura de datos arreglo bidimensional.   Recursos:   * Presentación de diapositivas - Semana-04 * Desarrollo de ejercicios de la guía N04 |  |  |
| **05** | Listas Enlazadas: operaciones de recorrido, búsqueda, inserción y eliminación. | Actividades:   * Captar la atención de los estudiantes sobre el uso del TDA Lista en aplicaciones de cómputo. * Explicar el TDA Lista enlazada como estructura dinámica y su diferencia con una estructura estática arreglo. * Implementar el TDA Lista enlazada usando clases y objetos o referencias. * Implementar las operaciones básicas de un TDA Lista enlazada (Agregar, i-ésimo, ubicación, insertar, eliminar, recorrido).   Recursos:   * Presentación de diapositivas - Semana-05 * Desarrollo de ejercicios de la guía N05 | Nota 1: Evaluación en clase / Ejercicio individual |  |
| **06** | Listas Enlazadas. Implementación y análisis. | Actividades:   * Captar la atención sobre conceptos básicos del TDA Lista enlazada. * Realizar el análisis de complejidad de las operaciones básicas de un TDA Lista enlazada y sus posibles mejoras. * Desarrollar aplicaciones de mediana dificultad del uso del TDA Lista enlazada.   Tareas:   * Ejercicios propuestos sobre extensión del TDA Lista enlazada y su uso en el desarrolla de aplicaciones de mediana dificultad.   Recursos:   * Presentación de diapositivas - Semana-06 * Desarrollo de ejercicios de la guía N06 |  |  |
| **07** | Cola: operaciones de encolar y desencolar. Representación. Colas prioridad. | Actividades:   * Captar la atención de los estudiantes sobre el uso del TDA Cola en aplicaciones de cómputo. * Implementar las operaciones básicas del TDA Cola y realizar su análisis de complejidad de dichas operaciones. * Desarrollar aplicaciones del uso del TDA Cola.   Tareas:   * Ejercicios propuestos sobre el uso del TDA Cola en el desarrollo de aplicaciones de mediana dificultad.   Recursos:   * Presentación de diapositivas - Semana-07 * Desarrollo de ejercicios de la guía N07 |  |  |
| **08** | Pila: operaciones de apilar y desapilar. Representación. Aplicaciones de pilas. | Actividades:   * Captar la atención de los estudiantes sobre el uso del TDA Pila en aplicaciones de cómputo. * Implementar las operaciones básicas del TDA Pila y realizar su análisis de complejidad. * Desarrollar aplicaciones del uso del TDA Pila. Evaluación de expresiones y sus notaciones.   Tareas:   * Ejercicios propuestos sobre el uso del TDA Pila en el desarrollo de aplicaciones de mediana dificultad.   Recursos:   * Presentación de diapositivas - Semana-08 * Desarrollo de ejercicios de la guía N08 |  |  |
| **09** | Recursividad. Algoritmos recursivos. Caso base y caso recurrente. Llamadas recursivas. | Actividades:   * Captar la atención de los estudiantes sobre la importancia de la recursividad en la solución de problemas de cómputo. * Explicar el concepto de recursividad como estrategia de diseño de algoritmo divide y vencerás. * Desarrollar algoritmos recursivos para identificar el caso base y caso recurrente. * Explicar la complejidad de algoritmos recursivos.   Tareas:   * Ejercicios propuestos sobre algoritmos recursivos.   Recursos:   * Presentación de diapositivas - Semana-09 * Desarrollo de ejercicios de la guía N09 | Nota 2: Evaluación en clase / Ejercicio individual |  |
| **10** | Algoritmos recursivos de ordenamiento: Quicksort y MergeSort. Aplicaciones de recursividad. | Actividades:   * Captar la atención de los estudiantes sobre la aplicación de algoritmos de ordenación interna y externa en aplicaciones de cómputo. * Desarrollar algoritmos recursivos en aplicaciones de mediana dificultad que usan arreglos. * Explicar y analizar los algoritmos recursivos de ordenación Quicksort y MergeSort.   Recursos:   * Presentación de diapositivas - Semana-10. * Desarrollo de ejercicios de la guía N10. |  |  |
| **11** | Grafos, definición. Representación. Recorrido en profundidad y en anchura. | Actividades:   * Captar la atención de los estudiantes sobre la aplicación de Grafos en el desarrollo de aplicaciones de cómputo. * Explicar el TDA grafos y sus formas de implementación * Implementar un TDA Grafos con matriz de adyacencia. * Implementar operaciones de recorrido de grafos.   Recursos:   * Presentación de diapositivas - Semana-11. * Desarrollo de ejercicios de la guía N11. |  |  |
| **12** | Problema de árboles de expansión de costo mínimo. Algoritmo de Kruskal. | Actividades:   * Captar la atención de los estudiantes sobre la importancia de aplicar algoritmos Kruskal. * Explicar el Problema de árboles de expansión de costo mínimo. * Implementar el Algoritmo de Kruskal.   Tareas:   * Ejercicios propuestos sobre TDA Grafo y sus aplicaciones.   Recursos:   * Presentación de diapositivas - Semana-12. * Desarrollo de ejercicios de la guía N12. |  |  |
| **13** | Algoritmo de Prim. Algoritmo de Dijkstra. | Actividades:   * Captar la atención de los estudiantes sobre la importancia de aplicar algoritmos de Prim y Dijkstra. * Explicar el Problema del camino más corto. * Implementar el Algoritmo de Dijkstra.   Recursos:   * Presentación de diapositivas - Semana-13. * Desarrollo de ejercicios de la guía N13. | Nota 3: Entrega de trabajo / Proyecto |  |
| **14** | Árbol Binario. operaciones de búsqueda, inserción y eliminación. Recorridos de árboles. | Actividades:   * Captar la atención de los estudiantes sobre la importancia del uso de estructuras de datos no lineales Árboles. * Implementar el TDA Árbol Binario y sus principales operaciones (Crear, insertar, buscar, eliminar y recorridos). * Aplicar el TDA Árbol Binario para desarrollar aplicaciones de mediana dificultad.   Tareas:   * Ejercicios propuestos sobre TDA Árboles y sus aplicaciones.   Recursos:   * Presentación de diapositivas - Semana-14. * Desarrollo de ejercicios de la guía N14. |  |  |
| **15** | Árbol Binario de Búsqueda: operaciones de búsqueda, inserción y eliminación. Representación. | Actividades:   * Captar la atención de los estudiantes sobre la importancia del uso de estructuras de datos no lineales Árboles binarios de búsqueda. * Implementar el TDA Árbol Binario Búsqueda y sus principales operaciones (Crear, insertar, buscar y eliminar). * Aplicar el TDA Árbol Binario Búsqueda para desarrollar aplicaciones de mediana dificultad.   Recursos:   * Presentación de diapositivas - Semana-15. * Desarrollo de ejercicios de la guía N15. | Nota 4: Evaluación en clase / Ejercicio individual |  |
| **16** | Retroalimentación del aprendizaje. |  |  |  |
| **17** | Entrega final de notas |  |  |  |