

COMPETENCIAS POR UNIDAD		ESTRUCTURA DE DATOS , AGO-DIC2019
UNIDAD 1	Representar y aplicar los tipos de datos abstractos por medio de un lenguaje de programaciOn.	1.- Conocer conceptos fundamentales sobre los tipos de datos abstractos TDAs y su uso. 2.- Conocer la gestion de memoria estatica. Arreglos estaticos. 3.- Conocer la gestion de memoria dinamica. Arreglos dinamicos.
UNIDAD 2	Comprender y aplicar la recursividad como herramienta de programacion para el manejo de las estructuras de datos.	1.- Conocer los fundamentos de la mecanica de recursion: rama recursiva y no recursiva. 2.- Solucion de problemas repetitivos usando metodos recursivos.
UNIDAD 3	Conocer, identificar y aplicar las estructuras lineales en la solucion de problemas del mundo real.	1.- Conocer los conceptos fundamentales de la estructura de datos: Lista enlazada. 2.- Implementacion de las clases Lista, Nodo. Atributos, propiedades y metodos fundamentales: Constructor, inicializacion, InsBegin, InsEnd, Cab, InsBefore, InsAfter, DelBefore, DelAfter 3.- Definicion de otras propiedades y metodos en la clase Lista. 4.- Solucion de problemas usando listas. 5.- Conocer los conceptos fundamentales de la estructura de datos Pila. 6.- Implementacion de la clase Pila. Atributos, propiedades y metodos fundamentales: Constructor, inicializacion, Push, Pop, Empty, Full, Length, Tope. 7.- Definicion de otras propiedades y metodos en la clase Pila. 8.- Solucion de problemas usando pilas, arreglos y archivos. 9.- Conocer los conceptos fundamentales de la estructura de datos Cola. 10.- Implementacion de la clase Cola. Atributos, propiedades y metodos fundamentales: Constructor, inicializacion, Insert, Remove, Empty, Full, Length, Front, Rear. 11.- Definicion de otras propiedades y metodos en la clase Cola. 12.- Solucion de problemas usando colas, pilas, arreglos y archivos.

UNIDAD 4	Conocer, identificar y aplicar las estructuras no lineales en la solución de problemas del mundo real.	1.- Conocer los conceptos fundamentales de la estructura de datos: árbol binario. 2.- Implementación de las clases Árbol, Nodo. Atributos, propiedades y métodos fundamentales: Constructor, inicialización, Agregar, Buscar, Recorrer por orden, Inorden, Posorden, Eliminar. 3.- Definición de otras propiedades y métodos en la clase Árbol. 4.- Solución de problemas usando árboles binarios.
UNIDAD 5	Aplicar el método de ordenamiento pertinente en la solución de un problema real.	1.- Conocer los algoritmos de ordenación. 2.- Implementar en un lenguaje de programación los algoritmos de ordenación. 3.- Programar la medición de tiempo para conocer la eficiencia de los algoritmos de ordenación.
UNIDAD 6	Aplicar el método de búsqueda pertinente en la solución de un problema real.	1.- Conocer los algoritmos de búsqueda. 2.- Implementar en un lenguaje de programación los algoritmos de búsqueda. 3.- Programar la medición de tiempo para conocer la eficiencia de los algoritmos de búsqueda.
UNIDAD 7	Comprender la complejidad de los algoritmos e identificar la eficiencia de los mismos.	1.- Conocer conceptos fundamentales del análisis de complejidad de algoritmos. 2.- Conocer y dominar los conceptos de complejidad en tiempo de ejecución y espacio de memoria. Asintotas. 3.- Conocer y dominar los órdenes de complejidad de algoritmos.