|  |  |
| --- | --- |
|  |  |
| ALGORITMOS |  |
|  |  |
|  |  |
|  |  |

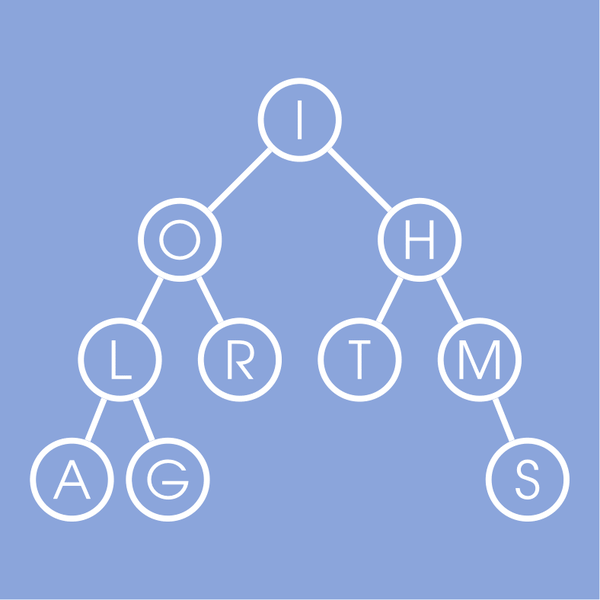
**INTEGRANTES:**

* Ariel Calderon Cueva
* Xihovik Darío Moreta
* Jhonathan Aucatoma
* Edison Ariel Arias
* Kevin Fernández Rivadeneira
* Hilda Cando Cunalata

**CURSO:** SOFTWARE

**Tipos de datos, estructuras de datos y tipos de datos abstractos**

En el mundo de la programación, la manipulación y representación de datos son aspectos cruciales que definen la eficiencia y funcionalidad de un software. En este contexto, nos adentramos en la comprensión profunda de tres conceptos fundamentales: tipo de datos (TD), estructura de datos (ED) y tipo de datos abstracto (TDA). Aunque estos términos comparten el objetivo de gestionar información, suelen generar confusión debido a sus matices específicos.



**Tipo de Datos (TD):** Se refiere al conjunto de valores que una variable puede tomar en un lenguaje de programación y las operaciones que pueden realizarse con dicha variable. Se ejemplifica con el tipo boolean en Pascal y se destaca la importancia de comprender el conjunto de valores y operaciones asociadas.

**Estructura de Datos (ED):** Consiste en una colección de variables organizadas de manera específica. La célula se presenta como la unidad básica de una estructura de datos. Se menciona la posibilidad de enlazar células usando apuntadores, que son valores que dirigen a una célula específica. Estos apuntadores pueden implementarse como cursores o punteros, según el contexto.

**Tipo de Datos Abstracto (TDA**): Se define por un modelo matemático y un conjunto de operaciones sobre ese modelo. Se distingue entre la definición y la implementación de un TDA. La definición implica establecer el modelo y las operaciones de manera clara y sin ambigüedad. La implementación implica encontrar una estructura de datos adecuada y escribir procedimientos que cumplan con las operaciones propuestas.

**Definición del TDA:** Se presenta un ejemplo con el TDA Número Complejo, especificando modelo (par ordenado de números reales) y conjunto de operaciones (suma, diferencia, producto, módulo, argumento, leer número, imprimir número) con descripciones precisas de cada operación.

**Implementación del TDA:** Implica encontrar una estructura de datos adecuada y escribir procedimientos que ejecuten las operaciones definidas. Se destaca la importancia de la concordancia entre la definición y la implementación para garantizar el correcto funcionamiento del programa.

**Uso del TDA**: Se señala que, si el lenguaje de programación ya tiene un tipo estándar que coincide con la definición del TDA, puede utilizarse sin necesidad de una implementación especial.