1/22/23 - **Encapsulamiento**

|  |  |
| --- | --- |
| Puntos clave | Notas |
| se puede encapsular:  Elementos o atributos, clases o incluso métodos.  El encapsulamiento nos permite tener dicho elemento inalterable en la POO. | El **encapsulamiento** es hacer que un dato sea inviolable, inalterable cuando se le asigne un modificador de acceso (no se trata solo de ocultar el dato sino también de protegerlo).  Un modificador de acceso define el alcance y visibilidad de un miembro de clase.  La **encapsulación** es también llamada ocultamiento de información.  **Algunos beneficios de encapsulación son**:   * Controlar la manera en que los datos son accedidos o modificados. * El código es más flexible y fácil de cambiar a partir de nuevos requerimientos. * Poder modificar una parte del código sin afectar otras partes de este. * Ayuda a mantener la integridad de los datos.   **Modificadores de Acceso**:   1. **Public**: Es el más permisivos de todos, Accede a todo. 2. **Protected**: Podrá ser accedido por la clase, paquetes y subclases. 3. **Default**: Permite el acceso a nivel de clases de internas y paquetes (No podremos ver las herencias si ha detener (osea subclases) 4. **Private**: Solo podrá ser modificado dentro de la clase. |
| Sumario: El **Encapsulamiento** es hacer que un dato sea inviolable, inalterable cuando se le asigne un modificador de acceso. | |

1/22/23

|  |  |
| --- | --- |
| Puntos clave | Notas |
|  | A picture containing chart  Description automatically generated |
| Sumario: | |