1/4/23 – **Que es un UML**

|  |  |
| --- | --- |
| Puntos clave | Notas |
|  | Como ya viste **UML significa Unified Modeling Language** el cual es un lenguaje estándar de modelado de sistemas orientados a objetos.  Esto significa que tendremos una manera gráfica de representar una situación, justo como hemos venido viendo.  A continuación te voy a presentar los elementos que puedes utilizar para hacer estas representaciones.  **Las clases se representan así:**  Table  Description automatically generated  **En la parte superior se colocan los atributos o propiedades, y debajo las operaciones de la clase**. Notarás que el **primer carácter con el que empiezan es un símbolo**. Este denotará la **visibilidad del atributo o método**, esto es un término que tiene que ver con **Encapsulamiento** y veremos más adelante a detalle.  **Estos son los niveles de visibilidad que puedes tener:**   * (- private) * (+ public) * (# protected) * (~ default)   Una forma de representar las relaciones que tendrá un elemento con otro es a través de las **flechas en UML**, y aquí tenemos varios tipos, estos son los más comunes: |
| Sumario: | |

1/4/23

|  |  |
| --- | --- |
| Puntos clave | Notas |
|  | **Asociación**    Como su nombre lo dice, notarás que cada vez que esté referenciada este tipo de flecha significará que **ese elemento contiene al otro en su definición**. La flecha apuntará hacia la dependencia.  Diagram  Description automatically generated with low confidence  Con esto vemos que la **ClaseA** está asociada y depende de la **ClaseB**.  **Herencia**    Siempre que veamos este tipo de flecha se estará expresando **la herencia**.  **La dirección de la flecha irá desde el hijo hasta el padre.**  Diagram  Description automatically generated  Con esto vemos que la **ClaseB** hereda de la **ClaseA** |
| Sumario: | |

1/18/23

|  |  |
| --- | --- |
| Puntos clave | Notas |
|  | **Agregación**    Este se parece a la asociación en que un elemento dependerá del otro, pero en este caso será: **Un elemento dependerá de muchos otros**. Aquí tomamos como referencia la multiplicidad del elemento. Lo que comúnmente conocerías en **Bases de Datos como Relaciones** uno a muchos.  A picture containing chart  Description automatically generated  Con esto decimos que la **ClaseA** contiene varios elementos de la **ClaseB**. Estos últimos son comúnmente **representados con listas o colecciones de datos**.  **Composición**    Este es similar al anterior solo que su **relación es totalmente compenetrada de tal modo que conceptualmente una de estas clases no podría vivir si no existiera la otra**.  A picture containing diagram  Description automatically generated |
| Sumario: | |