1. **¿Qué es un TAD?**

Un TAD es un Tipo Abstracto de Datos. Es un concepto de programación que se refiere a una abstracción de un tipo de dato que especifica un conjunto de valores y operaciones permitidas

1. **¿Dónde se produce el encapsulamiento?**

El encapsulamiento se produce en la programación orientada a objetos, donde se agrupan datos y los métodos que operan en esos datos en una única unidad llamada objeto.

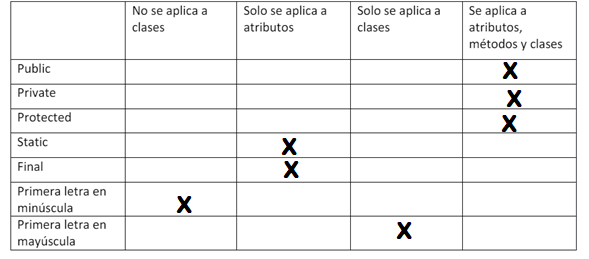
1. **¿Cuáles son las semejanzas y las diferencias entre funciones, procedimientos y métodos?**

Funciones: Son bloques de código que realizan una tarea específica y devuelven un valor. Pueden recibir argumentos de entrada y devolver un resultado.

Procedimientos: Son similares a las funciones pero no devuelven un valor. Realizan una tarea específica pero no retornan ningún resultado.

Métodos: Son funciones o procedimientos que están asociados con un objeto en la programación orientada a objetos. Operan en los datos del objeto y pueden acceder a ellos a través de la palabra clave "this" o "self".

1. **¿Qué es UML? ¿Y cómo se representa una clase en ese lenguaje?**es un lenguaje de modelado estándar utilizado en el desarrollo de software para visualizar, especificar, construir y documentar los artefactos de un sistema
2. **Marcar con una cruz**



6. **Verdadero o Falso**

- Un constructor…

o Es el método principal para ejecutar un programa. **Falso**

o Crea instancias. **Verdadero**

o Devuelve el valor de un atributo privado. **Falso**

o Tiene sentencia return. **Falso**

o Siempre existe uno por defecto, sin parámetros ni inicializaciones de atributos. **Verdadero**

o Se puede sobreescribir. **Verdadero**

o Se puede sobrecargar. **Falso**

o Su nombre se escribe con mayúscula. **Verdadero**

o Su calificador de acceso es static.**Falso**

o Su tipo de devolución no se indica y corresponde a la clase.**Verdadero**

- Un método…

o Puede tener múltiples parámetros con el mismo nombre, siempre y cuando tengan tipos diferentes. **Falso**

o Puede sobrecargarse. **Verdadero**

o Puede sobreescribirse. **Verdadero**

o Puede ser static. **Verdadero**

o Puede ser tanto public como protected, pero no private. **Falso**

o Un método puede tener un modificador de acceso final. **Verdadero**

Calificadores de acceso. Completa.

1. Se necesita que cualquiera pueda acceder al color de un vehículo. Entonces, declaro color como: Public
2. Se necesita que color se pueda acceder a través no sólo de vehículo, sí no ahora también de Buses, y como todos sabemos un bus es un tipo de vehículo, entonces también deberá tener acceso a color. Entonces, declaro color como: Protected
3. Se necesita que color se pueda acceder solamente para vehículo. Entonces, declaro color como: Private