

**GA4-220501095-AA2-EV01 - Taller de conceptos y principios de
programación orientada a objetos**

SOR JUNNY LONDOÑO RIVERA

FABIÁN MONTEJO ZABALA

Donaldo Andrés Beltrán Prieto

Instructor

Servicio Nacional de Aprendizaje-SENA

ANALISIS Y DESARROLLO DE SOFTWARE (2627038)

Regional Quindío.

2023

INTRODUCCIÓN.

La programación diseñada a objetos (POO) es de programación que se basa en el concepto de objetos. Utilizado para desarrollar aplicaciones complejas porque nos permite desarrollar y trabajar con estructuras de datos grandes y complejas. La POO es una de las técnicas de programación más utilizadas en el mundo y amplia estructura de aplicaciones basadas en software.

OBJETIVOS:

Teniendo en cuenta las características del software a construir seleccione una de las herramientas para captura de requisitos y defina sus funcionalidades. Identificar los diferentes conceptos definidos dentro de la programación orientada a objetos (POO) con el fin de proporcionar una forma de diseño y estructuras para programas complejos de manera más organizada.

TERMINOLOGÍAS COMUNES UTILIZADAS EN EL PARADIGMA DE POO

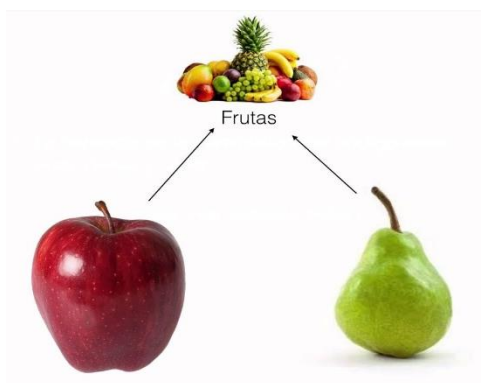
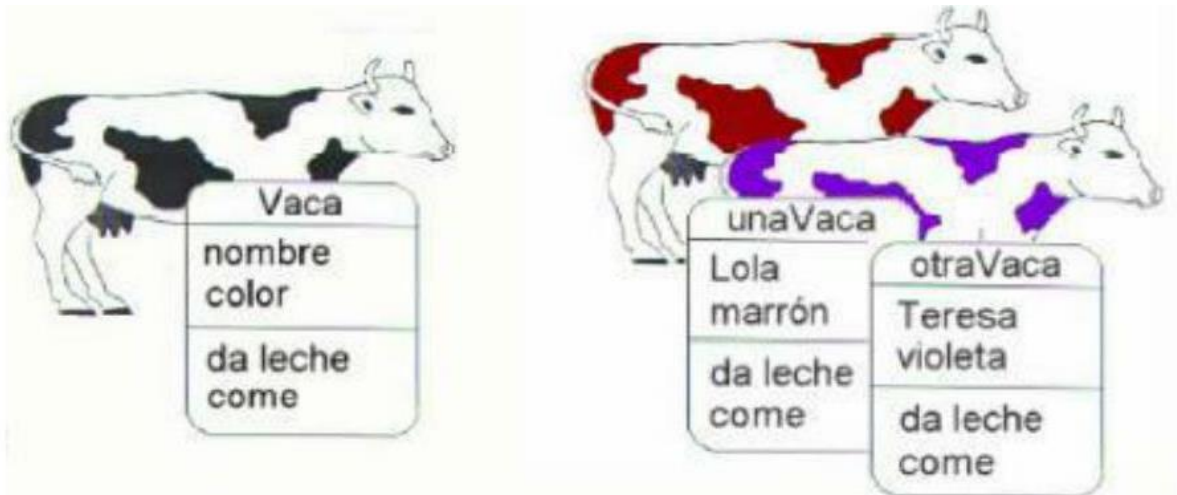
Estas son algunas de las terminologías más comunes utilizadas en la programación orientada a objetos. Existen muchas más que pueden cambiarse de acuerdo al tipo de lenguaje de programación que se utilice dentro de cada software a diseñar.

PROGRAMACIÓN DIRIGIDA A LOS OBJETOS (POO)

Deriva de sus siglas POO (Programación Orientada a Objetos): Es un paradigma que se basa en el uso de objetos y clases, para organizar y estructurar el código. Los objetos son instancias de una clase, que es una plantilla o modelos que describen los atributos y comportamientos de un conjunto de objetos relacionados. en **(POO)** básicamente nos permite organizar y estructurar mejor el código en términos de objetos y cómo interactúan entre sí, para que de esta manera podamos entenderlo de una mejor manera permitiendo que el diseño de software sea más modular, escalable y sostenible

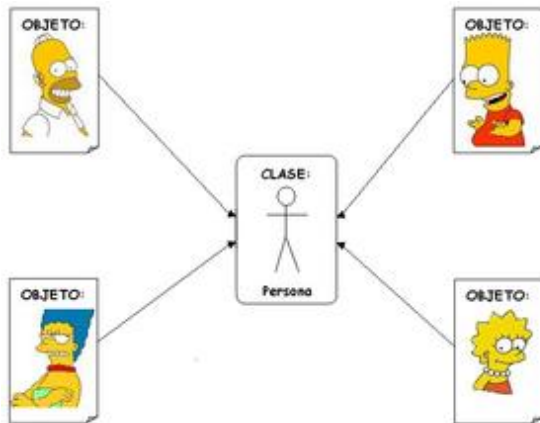
CLASES

Conjunto de objetos relacionados. Una clase es como un diseño que define cómo atributos (propiedades) y los métodos (funciones) que tienen los objetos de esa clase las plantillas o modelos que definen las características y comportamientos



Herencias: Se habla de herencia, cuando se heredan las características y comportamientos de la superclase, lo que reutilizar código y organizar de manera jerárquica las clases

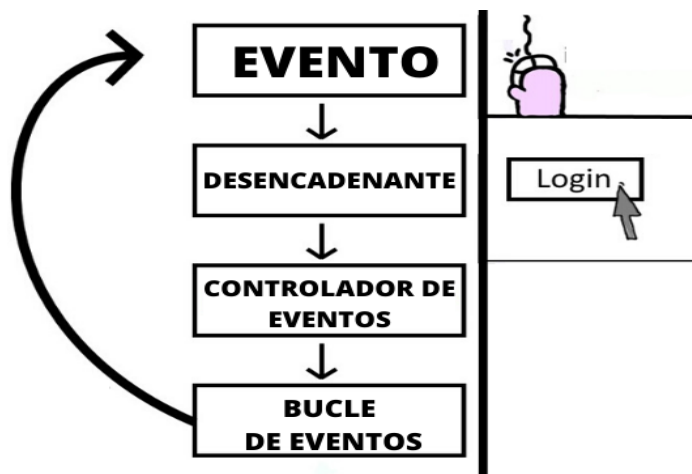
Objetos: Los objetos son una copia de una clase, pero con algunas variaciones y adiciones. Un objeto es una representación en código de un concepto o entidad que existe realmente. Los objetos pueden tener atributos que almacenan datos y funciones que realizan acciones o procesan datos.



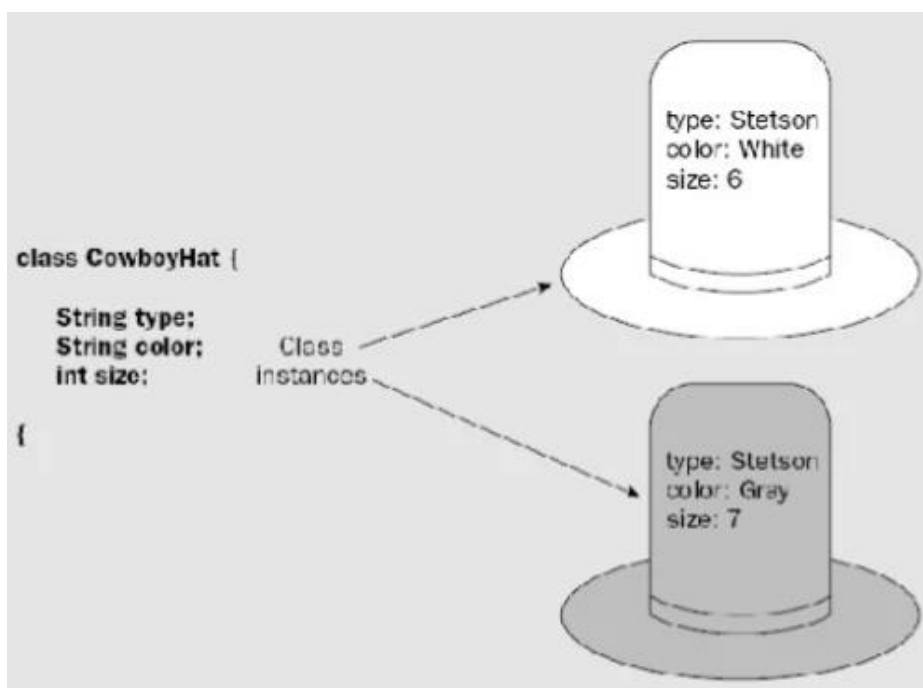
Métodos: Es una función vinculada a una clase o un objeto. Los métodos definen el comportamiento de un objeto y permiten realizar operaciones sobre él. Además, acceden y manipulan las propiedades de la clase u objeto al que pertenecen.



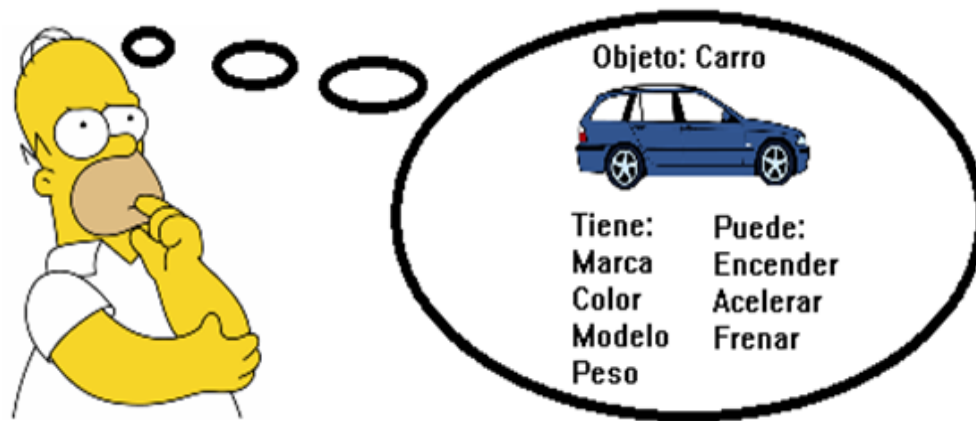
Eventos: Son acciones que se producen en el sistema o en la interfaz de usuario y que pueden ser identificadas y gestionadas por el código. Los eventos pueden ser generados por el usuario, el sistema operativo, el hardware u otros componentes de software.



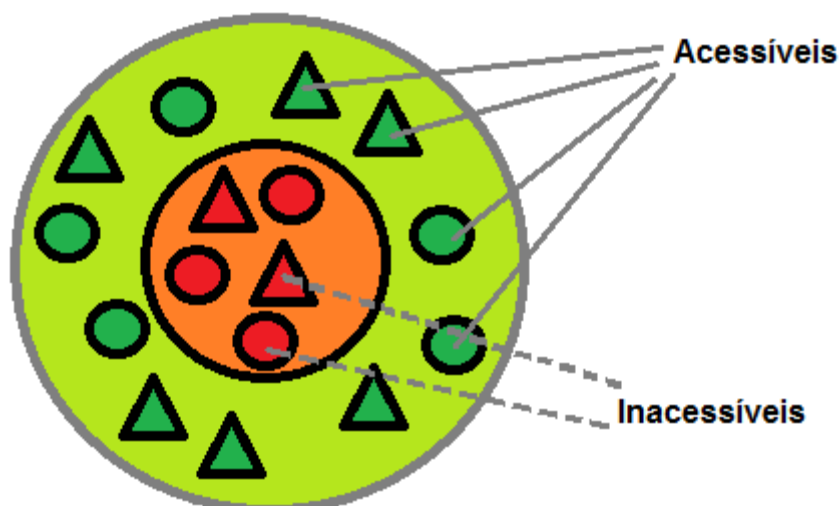
Atributos: Son las propiedades o características de una clase o de un objeto. Los atributos definen el estado o las características de los objetos y pueden almacenar datos. Por ejemplo, si tenemos una clase "PASEO", los atributos podrían ser lugar, días de permanencia, reservaciones, transporte, alimentación, entre otras.



Abstracción: Es el proceso de simplificar la complejidad del mundo real para modelar en el código. Programación orientada a objetos, la abstracción se logra mediante el uso de clases y objetos.



Encapsulamiento: Es un principio de Programación orientada a objetos que consiste en ocultar los detalles internos de una clase y exponer sólo lo necesario para su uso externo. Esto se logra definiendo los atributos y métodos de una clase con diferentes niveles de acceso, lo que permite proteger el estado interno de los objetos



Polimorfismo: Es la capacidad de una clase o de un objeto para responder a distintos mensajes o llamadas a métodos de manera similar, independientemente de su tipo o clase concreta.



CONCLUSIÓN

La POO es un paradigma que se centra en la creación de clases y objetos. Las clases son la "forma" y atributos que tendrán los objetos, y gracias a sus diferentes propiedades nos permite tener un código más ordenado. Los fundamentos de la POO son: La abstracción, el encapsulamiento, la herencia y el polimorfismo. Orientada a objetos nos da una forma organizada y modular de diseñar y estructurar programas complejos mediante el uso de objetos, clases, herencia, métodos, atributos, abstracción, encapsulamiento y polimorfismo.