

# Deber 1

David Pucó  
Nrc 1323

## Consulta:

### • Tipos de datos primitivos

Son valores básicos y esenciales para el lenguaje de la programación, cumplen con un propósito específico y están diseñados para ser procesados eficientemente por el hardware. Entre los tipos de datos se encuentran:

**Enteros:** Son números sin decimales, pueden ser positivos o negativos. Los enteros se usan para contar objetos o representar el número de elementos de una lista.

**Flotantes:** Representan números con decimales, el tamaño depende del lenguaje de programación usado. Sirven para obtener valores decimales precisos de cálculos matemáticos complejos.

**Caracteres:** Son letras, símbolos o números del teclado. Son usados para representar texto o valores alfanuméricos.

**Cadena de caracteres:** Está representado por un conjunto de caracteres que son fundamentales en la generación de texto, son esenciales para mostrar la información en la programación.

**Booleanos:** Estos valores pueden ser verdaderos o falsos. Sirven para realizar comparaciones y operaciones lógicas.

**Byte:** Cumple con la función de almacenamiento que representa una cantidad específica de datos.

**Short:** Es un tipo de dato para guardar valores numéricos pequeños. Consume menor espacio en la memoria.

**Long:** Usado para cálculos matemáticos complejos, contiene un rango más amplio para valores.

### • Tipos de referencias

Estos datos proporcionan almacenamiento a través de las referencias que se encargan de ubicar en la memoria, donde se encuentra el valor real de la información. Los tipos de referencias permiten manipular estructuras de datos más complejas y compartir información sin que haya duplicados.

**Arreglos:** Son estructuras de datos que poseen la capacidad de almacenar un conjunto de elementos relacionadas bajo un solo nombre.

**Cadena de caracteres:** Este tipo de cadenas al manipularlas, el trabajo que se desarrolla es hacia la referencia de su ubicación en la memoria.

**Clases y objetos:** La clase es una estructura definida por el usuario y el objeto es un tipo de dato que se origina a partir de la clase de programación. La variable clase es una referencia del objeto que se crea en memoria. El objeto de referencia que pasa a función se encarga de modificar al objeto original.

### • Paradigma de programación orientada a objetos

Los POO a parte de otorgar una estructura clara al programa, ayuda a la reutilización de código y el aislamiento de componentes, mejorando así la escalabilidad y mantenibilidad de las aplicaciones de software. Se basa en el concepto de objetos como parte de un elemento fundamental de la organización y estructura del código, permitiendo así crear programas. Los objetos son parte de una clase, la cual define los datos que contiene y las operaciones que se llevan a cabo en esos datos.

Los objetos se definen como una parte específica de una clase. La clase actúa como un modelo que agrupa las propiedades comunes de todos los elementos derivados de esa clase. Estos objetos son variables que almacenan datos. El comportamiento de un objeto se define por medio de métodos, lo que permite a los objetos realizar tareas internas, manipular datos, entre otros.

**Abstracción:** Permite concentrarse en aspectos importantes, sin la necesidad de entrar en profundidad sobre cada detalle del comportamiento del objeto en concreto. Con la reducción de la complejidad logra que se mejore en el aspecto de la comprensibilidad del código.

**Encapsulamiento:** Promueve la protección del estado interno de cambios no autorizados y otorga solo una interfaz pública para su interacción, reduciendo así las dependencias entre diferentes partes del código, posibilitando así un mejor mantenimiento y modificación.

**Herencia:** Proporciona que una clase herede propiedades y comportamientos de otra clase, esto permite que se ahorren en reescribir líneas de código, además de permitir la reutilización del mismo.

**Polimorfismo:** Se refiere a la capacidad de un objeto para que en determinado contexto se forme y se comporte de maneras distintas usando la misma interfaz. Permite tratar de manera uniforme a objetos de diferentes clases, también hace que el software sea adaptable y flexible ante nuevas situaciones.

• **Clase:** Es una plantilla para crear objetos, las clases definen un conjunto de métodos y atributos

• **Objeto:** Es una entidad que consta con un comportamiento o estado, y almacena datos de lo que representa.

• **Atributo:** Es un conjunto de características que se asigna a una variable, objeto o elemento del código. Su uso sirve para describir un objeto en particular que dependiendo del lenguaje de programación adoptara una u otra forma.

• **Método:** Representa el comportamiento de un objeto, esta función se encuentra definida dentro de una clase. Se utiliza para realizar tareas específicas en un objeto.



## • Sistema de control de versionamiento

Es un sistema que permite registrar cambios realizados sobre uno o varios archivos, con el propósito de poder recuperar diferentes versiones del archivo durante los periodos de modificaciones. Estos sistemas cuentan con una clasificación de tres tipos:

### Sistema de control de versiones locales

Hacen uso del almacenamiento en una base de datos, cuando se requiere revisar una versión anterior de un archivo, se usa el sistema de control de versiones para que de esta manera se reduzca la posibilidad de eliminar versiones del archivo, ya que solo se tiene una copia del archivo a la disposición de las modificaciones pertinentes.

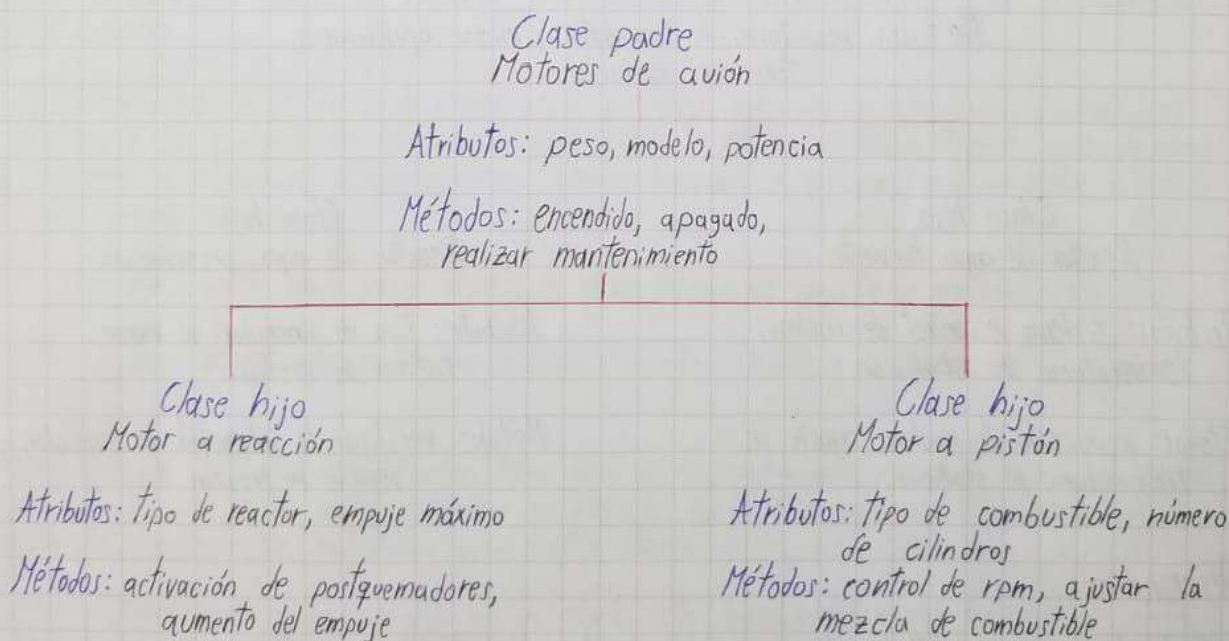
### Sistema de control de versiones centralizadas

En este sistema la información pasa de ser almacenada de los discos duros a los servidores, esto permite que los colaboradores puedan hacer cambios y revisiones de archivos de una manera más fácil.

### Sistema de control de versiones distribuidas

Se utiliza una red de repositorios para que cada desarrollador pueda trabajar de manera aislada, con esto se evita los conflictos de editar varios desarrolladores al mismo tiempo un archivo. Permite trabajar a cada desarrollador a su propio ritmo, almacenando los cambios a nivel local. Lo que hace poco probable de perder toda la información, ya que cada desarrollador cuenta con el archivo completo en sus repositorios locales que a su vez forman parte de un repositorio de red.

## • Ejemplos de diagramas de UML



Clase padre  
Misil balístico

Atributos: nombre, carga explosiva, peso, alcance,  
radio de explosión

Métodos: dirección, detonar, inicio de lanzamiento

Clase hijo  
Misil intercontinental

Atributos: capacidad de llevar múltiples ojivas  
nucleares, número de etapas del misil

Métodos: activación de etapas

Clase hijo  
Misil de corto alcance

Atributos: sistema de guiado, velocidad  
máxima

Métodos: ajuste de trayectoria, uso de  
contramedidas

Clase padre  
Reactor nuclear

Atributos: material de combustión, nivel de seguridad,  
potencia térmica

Métodos: monitoreo de temperatura, iniciar operaciones,  
terminar operaciones

Clase hijo  
Reactor de agua hirviente

Atributos: sistema de control de nucleos,  
temperatura de ebullición

Métodos: inspección de nucleos, ajuste de  
temperatura de ebullición

Clase hijo  
Reactor de agua presurizada

Atributos: tipo de generador de vapor,  
sistema de presión

Métodos: verificar la integridad del generador,  
regular la presión

### Referencias:

- CodersLinks. (2024). ¿Qué es la programación orientada a objetos (POO) y cuáles son sus principios fundamentales? <https://coderslink.com/talento/blog/qué-es-la-programación-orientada-a-objetos>
- Medium. (2017). Sistema de control de versiones: qué son y por qué amarlos. <https://medium.com/@jointdeveloper/sistema-de-control-de-versiones>
- Web Design Cusco. (S.F). Los 4 principios fundamentales de la programación orientada a objetos (POO). <https://webdesigncusco.com/los-4-principios-fundamentales-de-la-programación-orientada-a-objetos>