

# Tarea #1.



**PONDERACIÓN: 1 pts**

 **Tiempo estimado: 2 hrs**

Universidad San Carlos de Guatemala

Facultad de ingeniería.

Ingeniería en ciencias y sistemas

## 1. MARCO FORMATIVO

### 1.1. Valores

Nombre del valor	¿Cómo se aplica en tu laboratorio?
<b>Disciplina</b>	<b>Fomenta el respeto por normas y estándares que garantizan claridad en el código.</b>

### 1.2. Competencia(s)

Tipo de Competencia	
Competencia General	Aplica principios básicos de ingeniería, ciencias de computación y sistemas de información y comunicación, en la formulación y resolución adecuada de problemas complejos.
Competencia Específica	Comprender el funcionamiento del lenguaje de programación a través de su aplicación para la resolución de problemas relacionados con las fases del compilador.

### 1.3. Objetivo SMART

SMART	Definición	Objetivo redactado
<b>Específico (¿Qué?)</b>	El objetivo es concreto y tangible.	Diseñar e implementar una clase en Java que modele un objeto de la vida real con al menos 3 atributos, 1 constructor y 2 métodos funcionales.
<b>Medible (¿Cuánto?)</b>	El objetivo tiene una medida objetiva de éxito.	Crear exactamente 2 instancias con valores diferentes y ejecutar todos los métodos mostrando resultados en consola.
<b>Alcanzable (¿Cómo?)</b>	El objetivo debe ser posible con los recursos disponibles.	Utilizar conocimientos básicos de sintaxis Java y POO de nivel introductorio.

<b>Realista (¿Para qué?)</b>	El objetivo contribuye a metas más amplias.	Aplicar el principio de abstracción para comprender cómo modelar entidades reales en código, cumpliendo con la competencia del curso.
<b>A Tiempo (¿Cuándo?)</b>	El objetivo tiene fecha límite o mejor aún un cronograma de hitos de progreso	Completar toda la implementación, pruebas y documentación en el plazo establecido de una semana.

## 2. Actividad

Elegir un objeto (ej: Libro, Coche, Celular, Perro, etc.) y pensar en al menos 3 atributos que lo definan (ej: título, autor, páginas para un Libro) y crear una clase en el lenguaje C++. Debe contar con:

- Atributos públicos o privados
- Un constructor que inicialice los atributos.
- Al menos dos métodos adicionales que representen acciones del objeto (ej: para Celular, llamar(), cargarBateria()).

Posteriormente deberá crear dos instancias (objetos) de la clase con valores diferentes. y probar los métodos desde el método main, mostrando su funcionamiento.

Deberá crear un repositorio en GitHub llamado 1S2026\_TAREAS\_#CARNET y crear una carpeta llamada Tarea 1 que contenga lo siguiente:

- Código fuente.
- Markdown o PDF con capturas de pantalla que muestran la salida en consola de la implementación de los métodos.

DEBERÁ ENTREGAR EL LINK EN LA PLATAFORMA UEDI Y AGREGAR AL AUXILIAR COMO COLABORADOR.

Usuario de Github: **kevinmp2**

### 3. Cronograma

Asignación de tarea	03/02/2026
Entrega de Tarea	08/02/2026 antes de las 23:59

### 4. Rúbrica de calificación

CRITERIO	DESCRIPCIÓN	PUNTEO
Definición de clase	Definición de la clase con al menos 3 atributos y 2 métodos	30 pts
Creación de 2 instancias de la clase	Creación de al menos 2 instancias (objetos) de la clase con datos distintos,	30 pts
Implementación de al menos 2 métodos de la clase en ambas instancias	Implementación de al menos 2 métodos de la clase mostrados en el documento PDF	30 pts
Cumplimiento de ambos entregables solicitados	Entrega de código fuente y PDF con capturas	10 pts
<b>TOTAL</b>		<b>100 pts</b>

## Reflexión y aplicación práctica

¿Qué aprendí con esta tarea?

Esta tarea permitió comprender los fundamentos de la abstracción en la programación orientada a objetos utilizando el lenguaje C++. Se logró entender cómo modelar objetos del mundo real mediante clases, identificando sus atributos y definiendo sus comportamientos a través de métodos. Además, se comprendió el uso de constructores, el concepto de encapsulamiento y la forma en que se crean instancias de una clase, donde cada objeto mantiene su propio estado independiente. Este ejercicio reforzó el uso correcto de clases, objetos, modificadores de acceso y la importancia de estructurar correctamente el código para lograr programas más organizados y legibles.

**¿Cómo crees que aplicarás este conocimiento en otro curso o en la vida profesional?**

Este conocimiento será fundamental en cursos posteriores relacionados con programación avanzada, estructuras de datos, desarrollo de software o ingeniería de sistemas. En C++, el uso de clases es esencial para modelar entidades como usuarios, productos, cuentas o procesos, especialmente en aplicaciones de escritorio, sistemas embebidos o videojuegos. En el ámbito profesional, la abstracción permite diseñar sistemas más escalables, mantenibles y fáciles de entender, ya que divide problemas complejos en componentes más simples y reutilizables. Por ejemplo, en un sistema empresarial se pueden crear clases para representar empleados, inventarios o facturación, facilitando el mantenimiento del código y el trabajo colaborativo.

## Conclusión

La práctica de la abstracción mediante la creación de clases en C++ representa un pilar fundamental de la programación orientada a objetos. A través de este ejercicio se evidenció cómo es posible representar entidades del mundo real de forma estructurada, definiendo claramente sus atributos y comportamientos. La posibilidad de crear múltiples instancias de una misma clase demuestra la flexibilidad y potencia de este paradigma, permitiendo desarrollar soluciones más organizadas, reutilizables y eficientes. Este aprendizaje no solo fortalece las bases en C++, sino que también proporciona una forma de pensar aplicable a cualquier lenguaje orientado a objetos, preparando al estudiante para enfrentar proyectos más complejos en el ámbito profesional.

