

## Práctico 2

### Paradigma Orientado a Objetos - C++

1. Modelar e implementar el objeto Perro, determinar sus atributos y métodos útiles. Realizar un programita que permita cargar 6 perros y las edades de cada uno, y determinar cual es el mayor de ellos, y la edad promedio de todos ellos.
2. Modelar e implementar la clase Vector de 2 componentes, y la clase Vector de 3 componentes. ¿Qué atributos deberían tener? ¿Qué métodos deberían tener?
3. Modelar e implementar el objeto Vehiculo. Tanto sus atributos como sus métodos. Realizar un programita que luego permita cargar los autos favoritos y los autos más odiados de su grupo de amigos.
4. Crear un Objeto Temperatura. Deberá tener como métodos, la posibilidad de conversión a diferentes temperaturas, además deberá permitir definir cuál será la temperatura por defecto que utilizará. Grados Celcius, Farenheit, Kelvins, etc.
5. Crear el objeto Empleado, deberá determinar cuáles serían sus atributos y métodos, considerando que se necesitará calcular su salario, y por lo tanto, información como horas trabajadas y tarifa por hora, serán importantes.
6. Modelar el objeto Celular. Luego, realizar un programa que permita ingresar la cantidad de dinero que el cliente dispone y determinar cuales modelos de la tienda podría comprarse, suponiendo que la tienda tiene 6 modelos a la venta.

### Paradigma Orientado a Objetos - Python

7. Modelar e implementar la clase Estudiante. Qué atributos y métodos considera debería tener?
8. Implemente un programa que permita cargar 6 estudiantes, donde cada uno tiene las notas de sus exámenes finales. Pueden ser diferentes entre un estudiante y otro, ya que cada uno ha rendido o promocionado diferente cantidad de materias. Modifique la clase Estudiante si fuese necesario. Realice un programita que le permita cargar 3 estudiantes y determine cual es el alumno que será el abanderado, si lo hubiese.
9. Modelar e implementar una clase para un personaje de un juego de aventura. Qué atributos debería tener? Qué métodos podría tener?
10. Crear una clase Texto que contenga métodos que le permitan invertir una frase, obtener la primer palabra, obtener la i-ésima palabra, chequear si es palíndrome o no. Contar la cantidad de una determinada letra pasada como argumento. *(No utilizar comandos chetos como reverse, count, split de los str ni de las listas. Desarrollar los métodos en base a razonamiento y las estructuras básicas, fundamentales. Crear al menos un método propio que usted encuentre interesante para un texto. ¿Qué podría ser divertido o útil de realizar con un texto?*

11. Crear una clase Carrito de compras, que permita agregar items, quitar items, determinar el costo total, y determinar si tiene envío gratis o no en base al monto total.
12. Modelar los objetos *TarjetaCredito* y *Cliente*. Además de determinar los atributos y posibles métodos de cada uno de esos objetos, debe prestar atención a cómo podrían estar relacionados entre si. Podría suponer que un cliente tiene solo una tarjeta de crédito por ejemplo, dentro de mi tienda, o que le permitiré tener todas las que necesite y de las cuales disponga. ¿Cómo deberá modelar dicha situación?
13. Implemente una clase llamada *MensajeCripto* con métodos *encriptar* y *desencriptar*. Deberá utilizar un esquema para encriptación básico de *Cifrado por Sustitución*, como por ejemplo el *Cifrado César*, de asignar caracteres de un conjunto específico al alfabeto español, para poder re escribir dicho texto. Y luego poder descifrarlo.  
Deberá desarrollar los algoritmos Usted, de modo artesanal, sin caer en la tentación de utilizar librerías para criptografía ya existentes.  
Sería ideal, que antes de investigar bien y mirar el material compartido por la cátedra, intente pensar por sí mismo en qué consistiría el encriptamiento o codificación de un mensaje, y su correspondiente desencriptado o decodificación del mismo.
14. ¿Cómo haría para modelar el objeto etiqueta *HTML*? Investigue, cuál es el conjunto de etiquetas de HTML puro y básico. Piense entonces, como implementar un objeto *EtiquetaHTML* que permita construir etiquetas HTML, que sean válidas y que tengan el contenido que va dentro de ellas. Simplifique tanto la cantidad de etiquetas de contemplará como los valores y opciones que pueden tomar dentro de ellas.
15. Crear la clase *Binario*. Esta clase debe permitir trabajar con números binarios, sumarlos, restarlos, etc., además de permitir la conversión hacia y desde números en base 10.