

# Ayudantía 2

Alexander Inostroza – [alexander.inostroza@usm.cl](mailto:alexander.inostroza@usm.cl)

## 1. Modelando perros

Una veterinaria desea implementar un sistema de registros digital para las mascotas que reciben, para ello, quieren empezar modelando la información de los perros, por lo que la veterinaria le pide a usted implementar una (muy básica) clase que represente un perro.

Para ello le entregan la siguiente plantilla:

```
public class Perro {  
    // Constructor vacío, puede settear valores genéricos como nombre = ""  
    public Perro(){}  
  
    // Constructor  
    public Perro(String nombre, String raza, String dueño, int edad){}  
  
    // métodos setters  
    public void setNombre(String nombre){}  
    ...  
  
    // métodos getters  
    public String getNombre(){return "";}  
    ...  
}
```

## 2. La piscina más barata

Pedrito desea construir una piscina en su patio, para ello cotiza con 2 empresas que ofrecen el servicio de construcción de piscinas con forma de prisma rectangular. La empresa A le cobra 10.000 pesos por cada metro cúbico de piscina, mientras que la empresa B le cobra 15.000 pesos por cada metro cuadrado de pared que tenga la piscina.

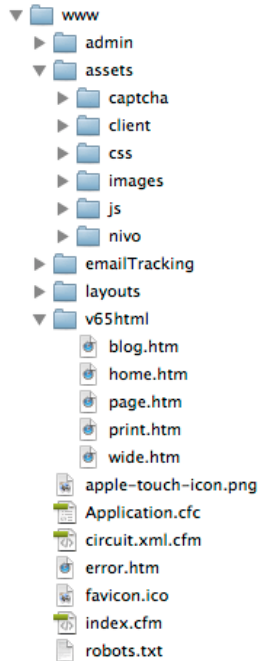
Para calcular el precio bajo cada oferta, Pedrito quiere un código en Java que le permita guardar las dimensiones de la piscina y obtener el precio que le cobraría cada empresa. Para ello, complete la clase **CalculadoraDePiscinas** y escriba un programa **main** que le permita ingresar las dimensiones de una piscina y obtener la cotización bajo el modelo de costos de cada empresa.

```
public class CalculadoraDePiscinas {  
    // Constructor vacío  
    public CalculadoraDePiscinas(){}  
  
    // Constructor con dimensiones  
    public CalculadoraDePiscinas(int x, int y, int z){}  
  
    // Precio bajo la cotización de la empresa A  
    public int precioEmpresaA(){ return 0;}  
  
    // Precio bajo la cotización de la empresa B  
    public int precioEmpresaB(){ return 0;}  
}
```

¿Para qué dimensiones es mejor la oferta de la empresa A? ¿Para qué dimensiones es mejor la oferta de la empresa B?

### 3. Directorios en un sistema operativo

Los directorios (o carpetas) en un sistema operativo son capaces de almacenar dentro de si más directorios y archivos.



Para simular directorios usando Java, escriba una clase **Directory**.

Esta clase debe ser capaz de almacenar muchos objetos de clase **Directory**, además de archivos (pueden ser Strings como “*nombre\_archivo.txt*”), para ello puede utilizar la clase **ArrayList**. Cada directorio debe saber cuál es su directorio padre. Debe además permitir agregar archivos y directorios nuevos, eliminar archivos y directorios, modificar el nombre de archivos y directorios que contenga dicho directorio.

Adicionalmente, implemente la clase **JavaFileBrowser**, que le permita navegar por los directorios creados. Esta clase debe:

- Almacenar un directorio *root*, que es el directorio que no tiene directorio padre.
- Almacenar un directorio actual.
- Navegar entre directorios (ver comando **cd**).
- Listar los contenidos del directorio actual (ver comando **ls** en Linux, **dir** en Windows).