# Guía Número 1 - Intro

**Obligatorios:**

**Ejercicio 1**

Diseñe e implemente la clase **SubeCard**, donde a la misma se le debe poder depositar dinero, cobrar un viaje y solicitarle el balance.

**Ejercicio 2**

Diseñe e implemente una clase **Semaphore**, a la cual se le pueda cambiar el tiempo que dura la señal de Verde, la de Amarillo y la de Rojo. Adicionalmente la misma debe poder responder a la pregunta ¿Es válido desplazarse en sentido del semáforo?, y también a la pregunta mutuamente excluyente ¿Es válido cruzar el semáforo como peatón?

**Adicionales:**

**Ejercicio 3**

Diseñe e implemente la clase **Point2D**, el cual debe representar un punto en el espacio cartesiano de dos dimensiones. Debe contar con los siguientes métodos:

* getDistance(Point2D point): que indique la distancia que existe entre el punto destino.
* add(Point2D point): que devuelva un nuevo punto la suma del punto original y el argumento.
* getX() : devuelve la coordenada en X
* getY(): devuelve la coordenada en Y

Ayuda: Puede utilizar las funciones: (más adelante veremos que son)

Math.pow(x, y) : eleva el número *x* a la potencia y

Math.sqrt(x): devuelve la raíz cuadrada de *x*

**Ejercicio 4**

Implemente la clase círculo, que esté definida cómo un centro y un radio. De tal manera que permita consultar su centro, su área e indicar su un punto está incluido en su área:

* center() : obtiene el centro del círculo
* area(): obtiene el área del círculo
* contains(Point2D point): devuelve True/False si el punto está contenido está contenido.
* perimeter(): devuelve el perímetro del círculo.

Ayuda: Puede utilizar la constante π haciendo uso de:

Math.PI

**Ejercicio 5**

Implemente la clase **Interval**, que representa una secuencia igualmente espaciados de números desde el comienzo (inclusive) y hasta el final(exclusive). Debe contener los siguientes métodos:

* first() : devuelve el primer elemento del intervalo
* last() : devuelve el último elemento del intervalo
* at (int i): devuelve el i-ésimo elemento del intervalo.
* size(): devuelve la cantidad de elementos del intervalo

**Ejercicio 6**

Implemente la clase **Condor**, **Chicken** y **Parrot**, donde todas sepan:

* eat(int gramsOfFood) => Por cada gramo puede volar una cantidad distinta de metros, y otra cantidad distinta de metros para caminar
* canFly() => retorna boolean, ya que pueden existir aves
* fly() => retorna los metros volados
* walk() => retorna los metros volados
* isHungry() => retorna boolean