

30/10/2023

Estructura de datos



# Instituto Tecnológico de Chetumal

Tema - Tipo Abstracto de Datos (TAD)

“Ejercicio 2 - Escala”



Grupo I3B

Integrantes del equipo:

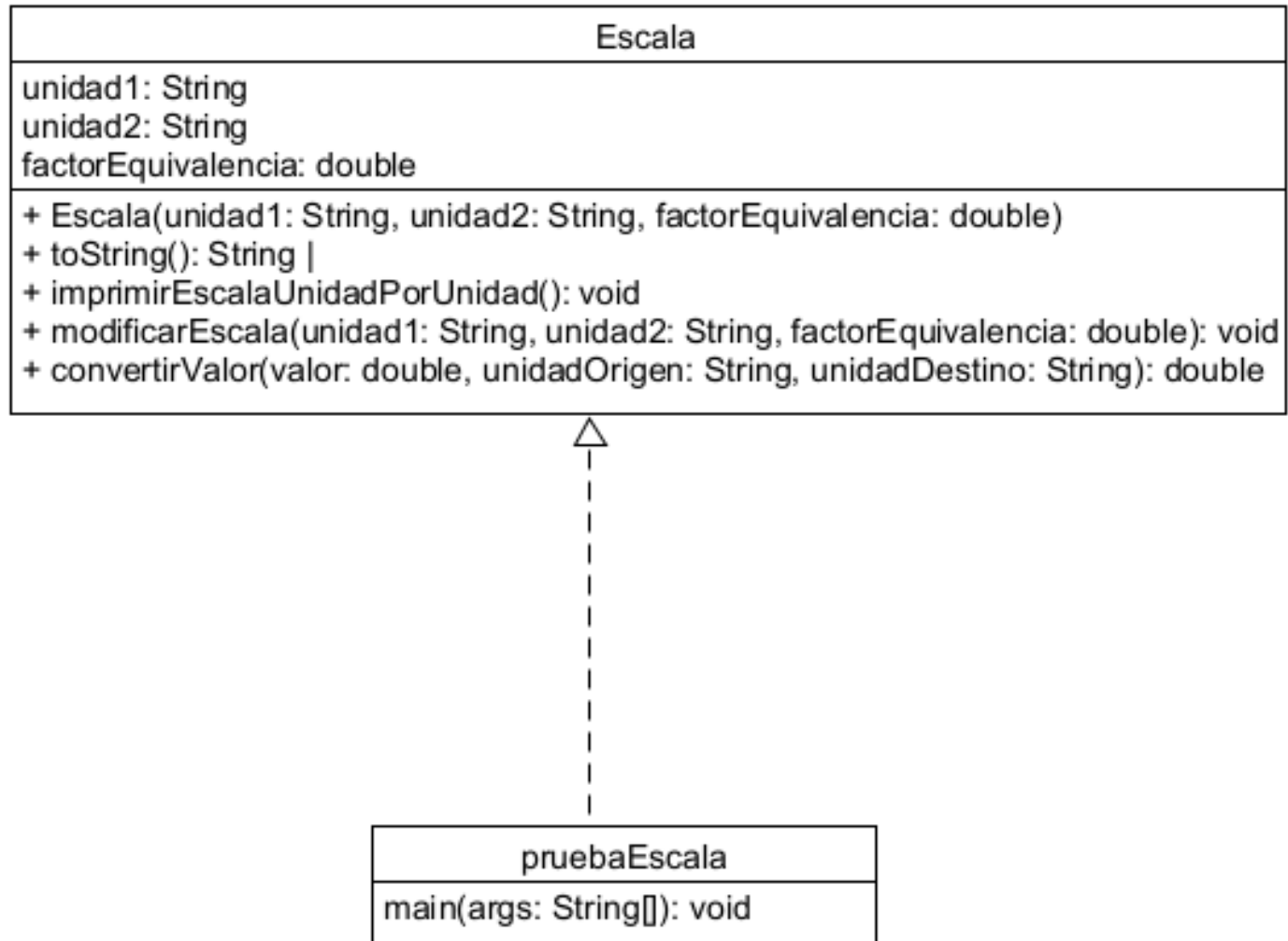
- Andrés Oswaldo Mukul Cab
- Edwin Antonio Flota Cano

Docente:

➤ Mtro. Agustín Pat Esquivel



## Diagrama UML



## Descripción de los atributos y métodos de la clase que representa el TAD.

### Ejercicio a resolver:

Elabore la especificación e implementación de un TDA en Java que represente una escala.

El TDA deberá permitir crear una escala con unidades de medida a elección del usuario y un tercer valor que representa el factor de equivalencia de la segunda unidad respecto a la primera.

El TDA deberá permitir manejar escalas de reducción o de ampliación.

Además se permitirá modificar los valores y unidades de la escala.

Cuando un objeto escala sea tratado como cadena (método `__str__` para Python o `toString` para Java) se deberá permitir imprimirla en la forma más adecuada como una escala numérica.

Se necesitará un método para imprimirla como una escala unidad por unidad

Además es necesario tener un método para devolver el valor real de una dimensión usando la escala creada.

### Clases, atributos y métodos que usamos para resolverlo:

#### Clase Escala

La primera clase que creamos fue la clase “Escala” y esta es la que representa una escala de conversión entre las dos unidades de medida, nos permite crear escalas con unidades de medida personalizadas por la persona que este de usuario y otorga un factor de equivalencia entre ellas cómo nos indicó la parte del ejercicio “deberá permitir crear una escala con unidades de medida a elección del usuario y un tercer valor que representa el factor de equivalencia de la segunda unidad respecto a la primera”.

Esta clase también nos permite manejar la gestión de escalas de reducción y ampliación, modificar los valores y unidades de la escala, y la conversión de valores de una unidad a otra, como lo indica el tercer párrafo del ejercicio.

Atributos:

- `unidad1 (String)`: primera unidad de medida en la escala.
- `unidad2 (String)`: segunda unidad de medida en la escala.
- `factorEquivalencia (double)`: el factor de equivalencia entre las dos unidades. este factor nos ayuda a determinar cómo se convierten valores de una unidad a la otra.

Métodos:

- `Escala(unidad1: String, unidad2: String, factorEquivalencia: double)`: el método constructor de la clase para crear una instancia de Escala con unidades y el factor de equivalencia que se pueden modificar.
- `toString(): String`: este método lo utilizamos para representar la escala en forma de cadena de caracteres como indica el quinto párrafo del ejercicio.

- `imprimirEscalaUnidadPorUnidad()`: método que imprime los detalles de la escala (las unidades y el factor de equivalencia).
- `modificarEscala(unidad1: String, unidad2: String, factorEquivalencia: double)`: es el método que nos permite modificar las unidades y el factor de equivalencia de la escala.
- `convertirValor(valor: double, unidadOrigen: String, unidadDestino: String): double`: este convierte un valor de una unidad de origen a una unidad de destino, utilizando el factor de equivalencia que le pongamos a la escala.

### **Clase PruebaEscala**

La clase `PruebaEscala` es nuestra clase para pruebas que tiene el método `main`. funciona para demostrar el uso de la clase `Escala` y probar que funcione correctamente los métodos y atributos que mencionamos anteriormente. En el método `main`, creamos una instancia de `Escala`. También cuenta con un `Scanner` que permite pedir tanto la unidad 1, la unidad y el factor de equivalencia a través del teclado.

Métodos:

- `main(args: String[])`: se usa e método principal que se ejecuta cuando se inicia el programa. este método, realiza las pruebas de la clase `Escala` para verificar que funcione bien.