

# Trabajo Práctico 1 — Smalltalk

[7507/9502] Algoritmos y Programación III  
Curso 2  
Segundo cuatrimestre de 2024

Alumno:	Ceniceros, Valentino
Número de padrón:	111054
Email:	vceniceros@fi.uba.ar

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Supuestos</b>	<b>2</b>
<b>3. Modelo de dominio</b>	<b>2</b>
<b>4. Diagramas de clase</b>	<b>3</b>
<b>5. Detalles de implementación</b>	<b>6</b>
5.1. CalcularPesoMaximo . . . . .	6
5.2. compararConPesoEnLaLuna . . . . .	6
<b>6. Excepciones</b>	<b>6</b>
<b>7. Diagramas de secuencia</b>	<b>7</b>

## 1. Introducción

El presente informe reúne la documentación de la solución del primer trabajo práctico de la materia Algoritmos y Programación III que consiste en desarrollar una aplicación de un sistema un calculador de pesos máximos de unos juegos olímpicos en Pharo utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

## 2. Supuestos

**dos atletas no pueden empatar en peso máximo:**

esto es debido a que en los test eso no se ve y porque a nivel lógico del mundo real, no tendría mucho sentido que eso suceda, qué probabilidad hay de que te encuentres 2 atletas con el mismo peso, edad y altura?

**la edad mínima es 16 y la máxima 55:**

esto más que un supuesto son las reglas de las olimpiadas en el caso de la edad mínima, y tome la máxima basada en la edad máxima de desarrollo físico.

**un peso no puede ser negativo ni nulo:**

esto también debido a una razón lógica y es que si bien en la luna puedo tener pesos muy pequeños, estos jamás pueden ser menores o iguales a 0.

**la altura mínima es 1.25:** esto es debido a que los atletas más cortos de estatura en la historia de las olimpiadas empataron con la altura de 1.27, fijando el límite en 1.25 dejó margen para que alguien deje asentado un precedente.

## 3. Modelo de dominio

El Cómo encare el modelado empezó por una algoOlimpícos como una fachada central después implementa primero una clase peso que se encarga de toda lógica de multiplicar y comparar, una clase nacionalidad que tenga por estados local, limítrofe y lejano delegándoles a estas el cálculo de porcentaje en base a la nacionalidad, una clase edad que tiene por estados a veterano y joven haciendo lo mismo para el porcentaje según edad y todas estas están relacionadas con atleta siendo este el núcleo de todo el cálculo de los pesos

**AlgoOlimpícos:** Esta clase central gestiona una colección de atletas y un criterio de selección. Tiene métodos para crear instancias de esta clase, registrar nuevos atletas, obtener un atleta según un criterio, y calcular el peso máximo de un atleta específico.

**Atleta:** Representa a un atleta, con atributos como peso, altura, nacionalidad y edad. Incluye métodos para obtener el nombre del atleta, comparar nombres y calcular el peso máximo que puede levantar.

**Criterio:** Es una clase abstracta que define un método para obtener un atleta basado en un criterio específico. Esta clase será extendida por otras implementaciones que definen criterios concretos, siendo este fuerte y débil cada uno obteniendo el atleta de mayor y menor peso máximo respectivamente.

**Peso:** Esta clase representa el peso, con un valor numérico y métodos para realizar conversiones y comparaciones con otros pesos.

**CalculadorDePorcentajes:** Es una interfaz que define un método para calcular porcentajes sobre un peso, siendo implementada por las clases Nacionalidad y Edad.

**Nacionalidad y Edad:** Ambas clases tienen atributos específicos y métodos para calcular un porcentaje que afecta al peso máximo de un atleta en base a la condición de su nacionalidad o su edad.

**SeleccionadorDeCriterio, SeleccionadorDeEdad, y SeleccionadorDeNacion:** Estas clases son responsables de seleccionar criterios, edades y nacionalidades específicas, respectivamente.

## 4. Diagramas de clase

Los diagramas de clases son los siguientes: Diagrama general, de condición, de plenitud, y de excepciones.



Figura 1: Diagrama de clases

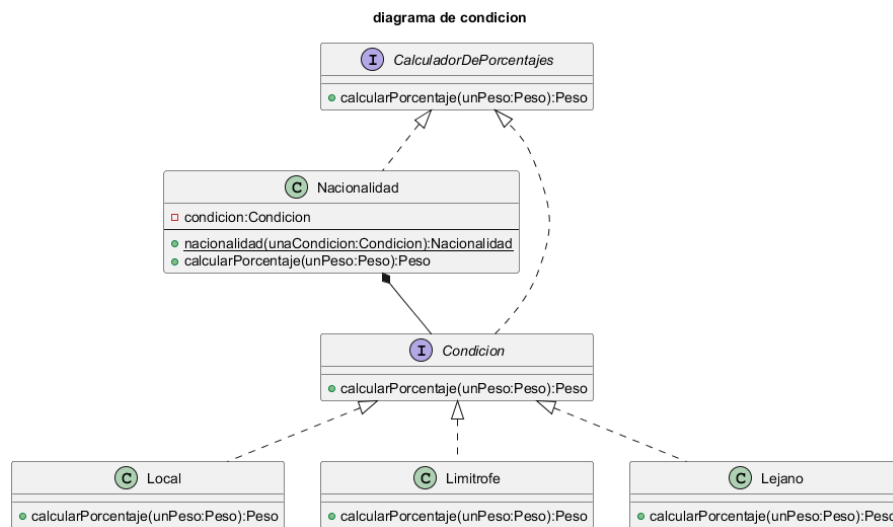


Figura 2: Diagrama de condicion

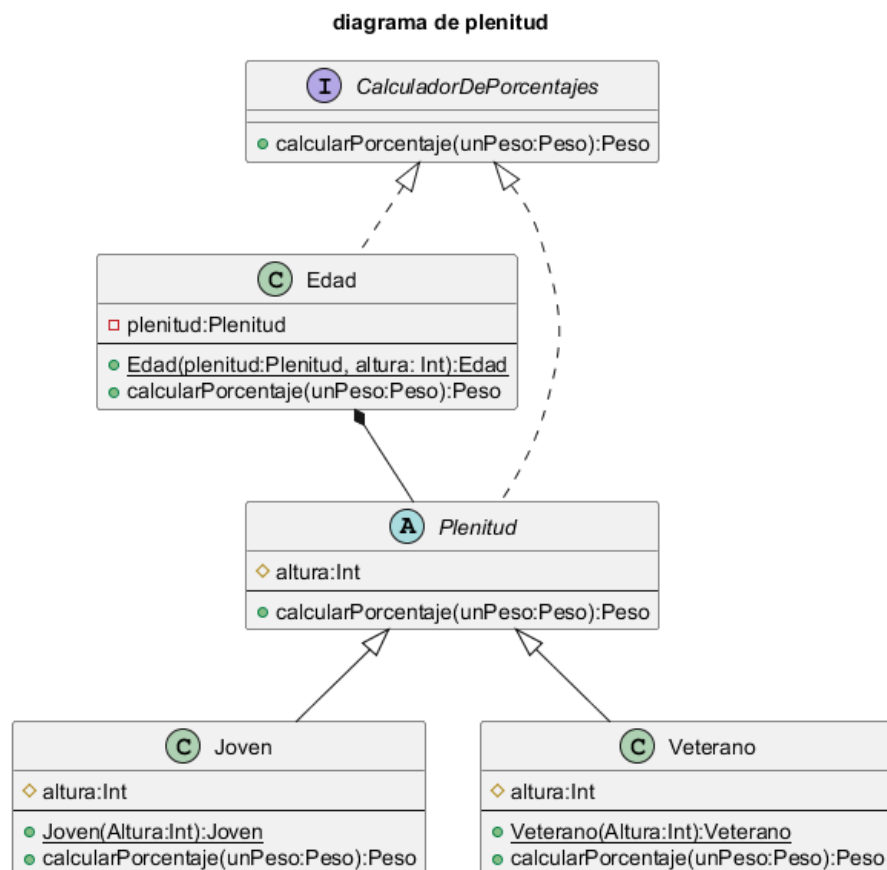


Figura 3: Diagrama de plenitud

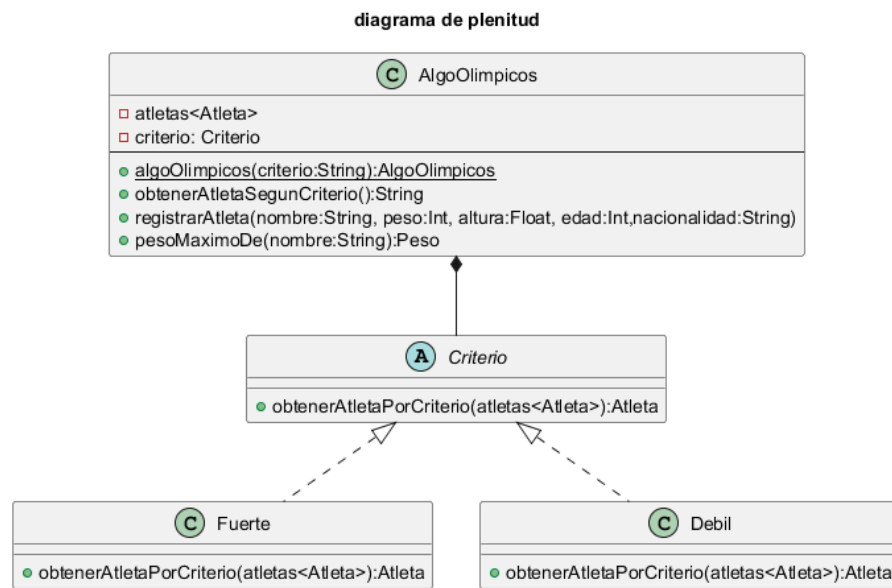


Figura 4: Diagrama de criterio

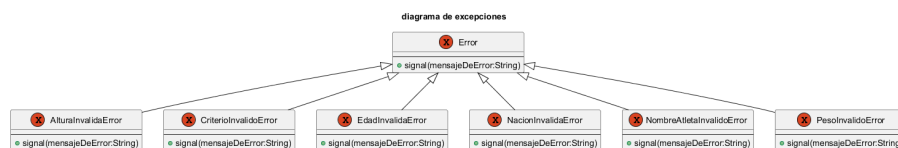


Figura 5: Diagrama de excepciones

## 5. Detalles de implementación

### 5.1. CalcularPesoMaximo

destaco primero el cómo el atleta calcula el peso máximo que puede levantar, esto debido a que en 2 líneas esta cuenta resume la relación y colaboración entre los distintos objetos que se encargan de calcular peso

```
calcularPesoMaximo
```

```
  ^edad calcularPorcentaje: (nacionalidad calcularPorcentaje: pesoAtleta).
```

### 5.2. compararConPesoEnLaLuna

Dote de un comportamiento a peso que le permite convertirse a sí mismo en un peso lunar que luego es comparable con otro peso en la luna.

```
calcularPesoMaximo
```

```
  ^edad calcularPorcentaje: (nacionalidad calcularPorcentaje: pesoAtleta).
```

## 6. Excepciones

**AlturaInvalidaError** Este error se acciona si el usuario y/o test ingresa una altura menor a 1.25 con el fin de respetar el supuesto de la altura.

**CriterioInvalidoError** este se acciona en caso de que el criterio que se quiera seleccionar no sea 'Fuerte' o 'Débil' esto para no generar errores a la hora de instanciar nuevos criterios.

**EdadInvalidaError** Este error se activa al recibir una edad menor a 16 y mayor a 55, esto debido también al supuesto de la edad

**NacionInvalidaError** Este error se acciona en seleccionar nacionalidad si es que él se le envía algo que no sea un string y que no sea mayor o igual a 3 caracteres, este error es debido a que quizás en algún futuro se pueda llegar a pasar las naciones como sus acrónimos (3 caracteres que lo representen) menos que eso no tendría sentido.

**NombreAtletaInvalido** Mismo caso que con nacionalidad este error salta cuando el valor ingresado no es un string mayor o igual a 3 caracteres.

**PesoInvalidoError** esta excepción ocurre cuando se carga un peso negativo o igual a 0, esto debido a que rompería la lógica del ejercicio.

## 7. Diagramas de secuencia

Los diagramas de secuencia que quiero destacar son el de constructores, el cual se encarga de mostrar con casos más genéricos el cómo se crean e instancian las distintas clases en el proyecto, y los diagramas del test 03 y el test 06.

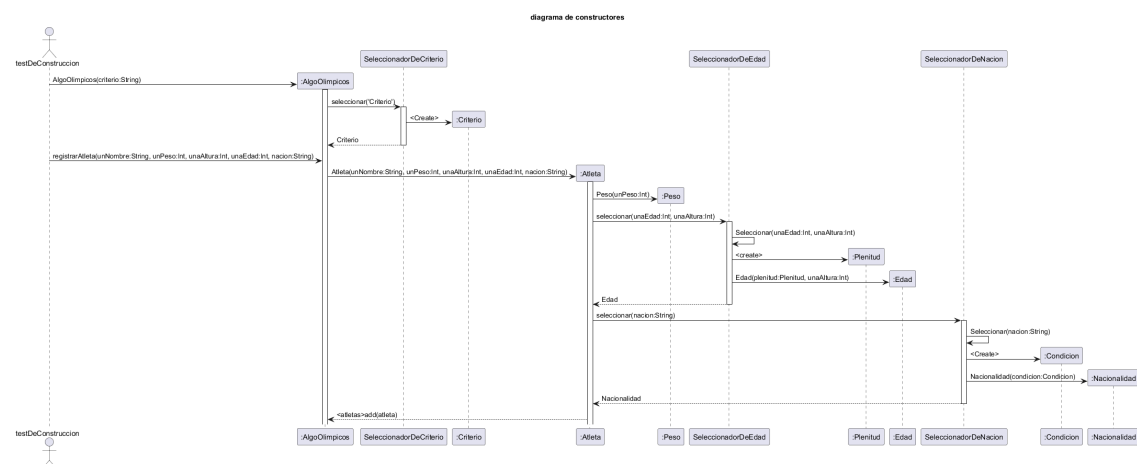


Figura 6: Diagrama de constructores.



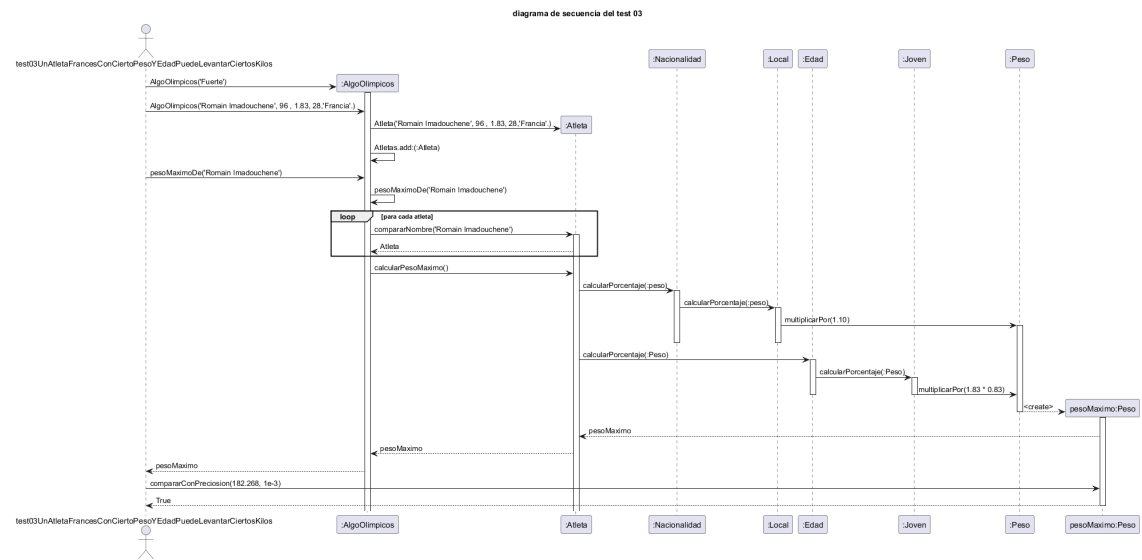


Figura 7: diagrama de secuencia del test 03.

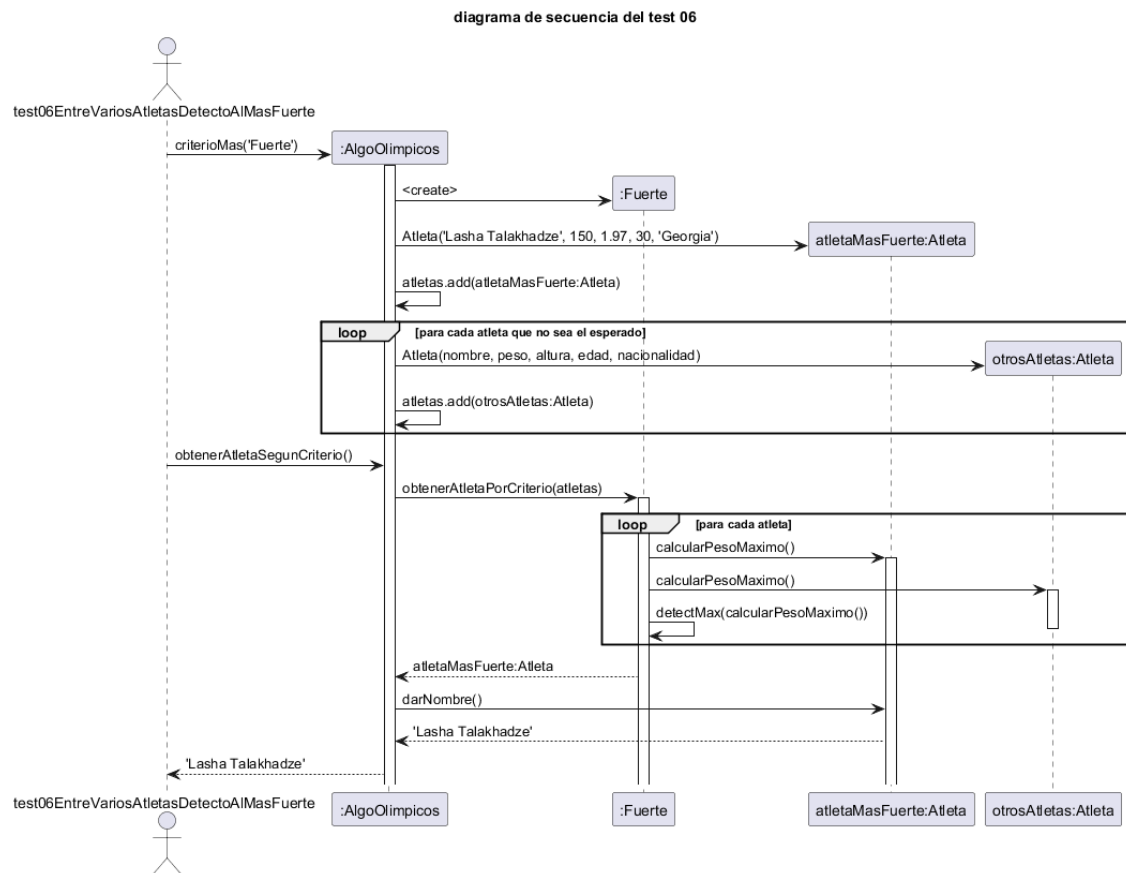


Figura 8: diagrama de secuencia del test 06.