

# Trabajo Práctico 1 — Smalltalk

[7507/9502] Algoritmos y Programación III  
Curso 1  
Primer cuatrimestre de 2023

Alumno:	OVIEDO, Ignacio
Número de padrón:	109821
Email:	ioviedo@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>2</b>
<b>5. Detalles de implementación</b>	<b>3</b>
5.1. Pilares del paradigma empleados . . . . .	3
5.2. Herencia y delegacion . . . . .	4
<b>6. Excepciones</b>	<b>4</b>
<b>7. Diagramas de secuencia</b>	<b>4</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 de una agencia de viajes en Pharo utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

## 2. Supuestos

El unico supuesto que tome fue lo que hacian los dos tipos de criterios, tome que, en caso de cargar mas de un viaje con el mismo nombre, criterio maximo pedia que guarde el viaje con la tarifa mas alta y criterio minimo pedia que guarde el viaje con la tarifa mas baja.

## 3. Modelo de dominio

AlgoViajes contiene 2 atributos, por un lado un diccionario que usa como claves los nombres y como valor un objeto ViajeCalculado, que sera el devuelto cuando piden el viaje indicado segun el nombre y por el otro lado un comprobador que comprueba si los datos ingresados son correctos y ademas si se ingresan 2 o mas viajes con el mismo nombre comprueba si debe guardarse o no segun el criterio.

ViajeCalculado seria el objeto principal se podria decir, que tiene como atributos 2 objetos, CiudadEsperada que contiene todos los datos relacionados al destino y tarifa que contiene el precio del viaje y es la que le envia a CalculadorDeExtras para que le aplique todos los extras.

## 4. Diagramas de clase

A Tarifa no le inclui todos sus metodos en el grafico ya que me parecian metodos simples e internos que no afectan por fuera de si mismos.

En ComprobadoresDeTarifa el programa con el que hice el grafico no me permitia colocar italica en metodos, lo señale con una nota pero lo vuelvo a resaltar, el metodo abstracto es el ComprobarSiSobreescribir.

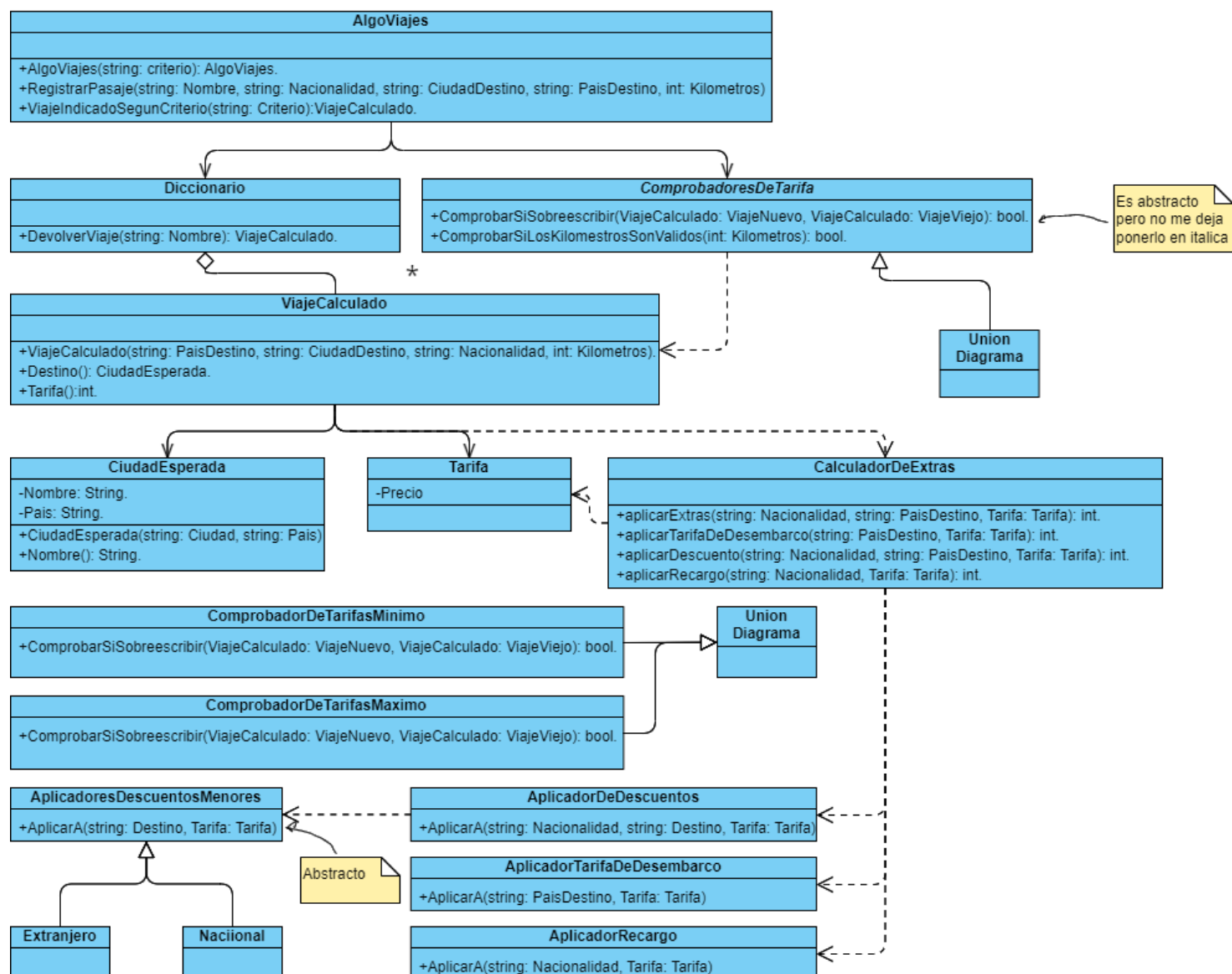


Figura 1: Diagrama de Clases.

## 5. Detalles de implementación

### 5.1. Pilares del paradigma empleados

Tarifa lo hice como un objeto aparte para tener abstracción respecto de `ViajeCalculado` y mantener su encapsulamiento, ya que este se la envía a `CalculadorDeExtras` y nada más. Además utilice un diccionario para tener abstracción respecto a `AlgoViajes`, ya que solamente le pido el `ViajeCalculado` que tiene guardado en ese nombre y listo en vez de recorrer un `OrderedCollection` o otro método de almacenamiento.

También utilice polimorfismo en los `ComprobadoresDeTarifas` (Maximo y Minimo) para poder ahorrar tener dos clases distintas de Viajes y poder tener una sola que ejecuta el mismo método solamente que utiliza un `ComprobadorDeTarifa` distinto según el criterio.

Utilice herencia y polimorfismo en el caso de `AplicadoresDescuentosMenores` ya que comparten un metodo solamente que lo ejecutan de distinta forma.

No utilice herencia en los distintos tipos de aplicadores ya que aunque su funcion es muy parecida no comparten ningun metodo, ya que todos necesitan distintos parametros.

## 5.2. Herencia y delegacion

Utilice herencia en los `Comprobadores` ya que comparten su utilidad solamente que dan distintos resultados.

Por otro lado use delegacion en: `ViajeCalculado` que le delega a `CalculadorDeExtras` la aplicacion de todos los extras y recargos sobre el precio de la tarifa, `AlgoViajes` le delega a `Diccionario` el guardado de todos los `ViajesCalculado` y tambien le delega a los `ComprobadoresDeTarifa` si el `ViajeCalculado` debe ser guardado o no.

## 6. Excepciones

**CriterioInvalido** Esta excepcion devuelve error cuando el criterio enviado por el usuario es invalido, osea si no es Maximo ni Minimo.

**KilometrosInvalidos** Esta excepcion devuelve error cuando los kilometros ingresados son invalidos, cuando son un texto o son un numero inferior a 0.

**ViajeInexistente** Esta excepcion devuelve error cuando el usuario pide la devolucion de un viaje con un nombre que no fue ingresado previamente.

## 7. Diagramas de secuencia

Hice dos diagramas ya que sino me quedaba muy ancho el grafico, el primero muestra el proceso que se lleva a cabo para registrar un pasaje y el segundo muestra el proceso que se realiza para decidir si guardar el pasaje creado o no.

Aclaraciones:

\*El primer mensaje del segundo diagrama deberia formar parte del primer diagrama pero no supe donde hacer bien el corte de los graficos y decidi que hacerlo asi era la mejor manera.

\*Cuando se registra por segunda vez un pasaje despues de preguntarle al `ComprobadorDeTarifas` si sobrescribir el viaje este devuelve true o false, dependiendo lo que devuelva `Viajes` le dice al `Diccionario` que sobrescriba o no, no devuelve true y despues false, sino que devuelve uno solo, lo señale con un recuadro y lo especifique en una nota, pero no estoy seguro que haya quedado lo suficientemente claro, asi que lo vuelvo a resaltar.

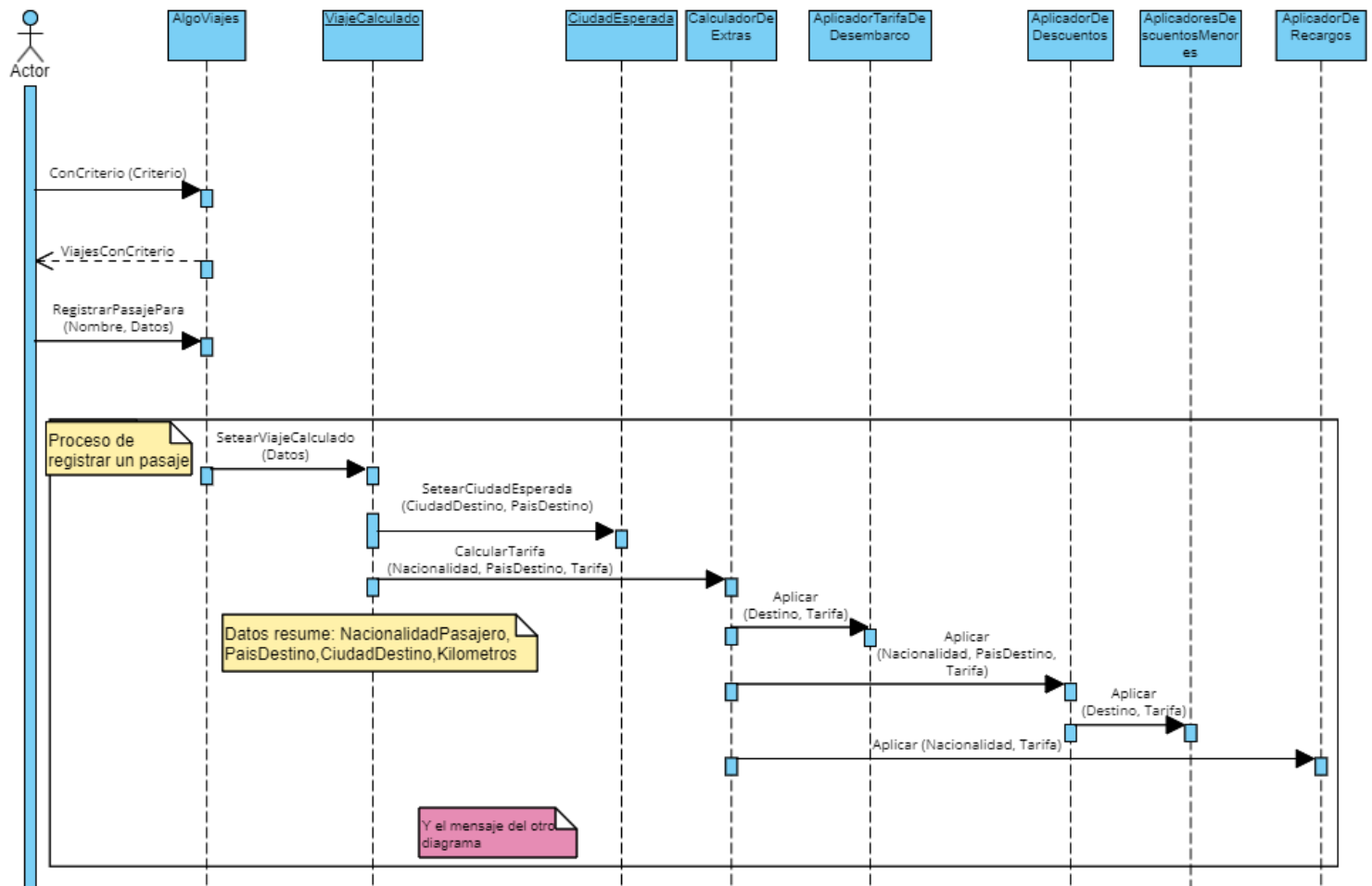


Figura 2: Diagrama de Secuencia 1, proceso de registrar pasaje.

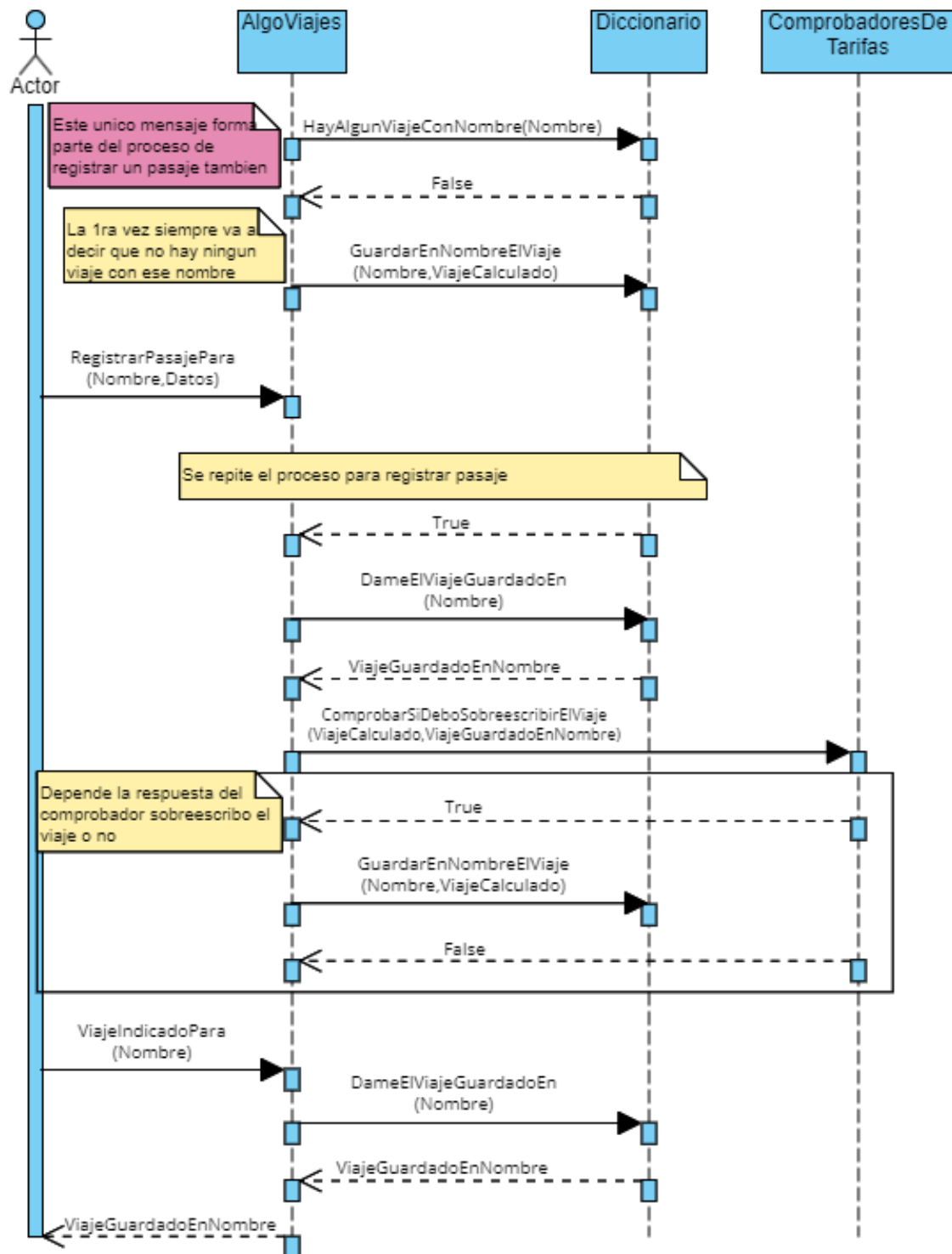


Figura 3: Diagrama de Secuencia 2, proceso de verificación de guardado.