

# Trabajo Práctico 1 — Smalltalk

[7507/9502] Algoritmos y Programación III  
Curso 01  
Primer cuatrimestre de 2022

Alumno:	GANDOLFO MARIET, Joaquín
Email:	jgandolfo@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>4</b>
5.1. Explicación de la clase TeleAlgo . . . . .	4
5.2. Explicación de la clase Llamada . . . . .	4
5.3. Explicación de la clase Horario . . . . .	4
5.4. Explicación de la clase Descuento . . . . .	4
<b>6. Excepciones</b>	<b>5</b>
<b>7. Diagramas de secuencia</b>	<b>5</b>
<b>8. MuTalk y code coverage</b>	<b>10</b>

## 1. Introducción

El presente informe reúne la documentación de la solución del primer trabajo práctico de la materia Algoritmos y Programación III. El trabajo práctico consiste en realizar una serie de clases con su paquete de tests. Se realizó a partir de ver en unas pruebas propuestas por la cátedra, los resultados que se esperaba que devuelvan las instancias de ciertas clases al recibir determinados mensajes.

## 2. Supuestos

- No hay diferencia de costos entre llamadas internacionales realizadas en horario hábil y en horario no hábil.
- Se usa el sistema horario de 24 horas. Desde las 8 hasta las 19 se considera hábil incluyendo ambos extremos. Los demás números que pertenezcan al rango de 0 a 23 serán considerados como no hábiles. Se lanzará el error `HorarioInvalidoError` en el caso que haya un ingreso no válido.
- Si a una instancia de `TeleAlgo` no se le aplica una tarifa al minuto de llamada internacional, las comunicaciones de esta índole serán gratuitas. Si se le aplica una tarifa negativa, también serán gratuitas.
- Si se inicializa un `TeleAlgo` sin parámetro (`abonoMensual`), al igual que con el caso de las llamadas internacionales sin aplicar tarifa, este tendrá valor nulo y al momento de calcular el gasto final, no habrá una reflexión de este sobre dicho valor.

## 3. Modelo de dominio

El modelo que realicé para la implementación de la solución del trabajo, se diferencia fuertemente con la primera entrega, dado que rompí con esa herencia tan fuerte que había propuesto para las llamadas. En esta nueva iteración, planteé el problema casi desde cero.

Mi nuevo modelo, tiene las clases:

`TeleAlgo`: dentro de ella, tiene todos los métodos con los que había que cumplir con las pruebas dadas por la cátedra y también tiene asociada una `OrderedCollection` de llamadas.

`Llamada`: a diferencia de la entrega anterior, ahora cuento con una sola clase `Llamada`, que tiene asociada a ella las clases `Horario` y `Descuento`.

`Horario`: es una clase abstracta que tiene como hijas las clases `HorarioInternacional`, `HorarioNacionalHabil` y `HorarioNacionalNoHabil`. `Horario` tiene el método abstracto `calcularGastoLlamadaConDuracion(unaDuracion)` que me permitió hacer uso del polimorfismo y así evitar repetir código.

`Descuento`: es una clase abstracta que si no fuera por el constructor con argumento que tiene implementado, sería una interfaz. Tiene como hijas a las clases `ConDescuento` y `SinDescuento`. Nuevamente, el uso del polimorfismo en estas a través del método abstracto de `Descuento`, `calcularElDescuentoDe(unValor)`, me permitió tener un código prolijo y sin repetición.

## 4. Diagramas de clase

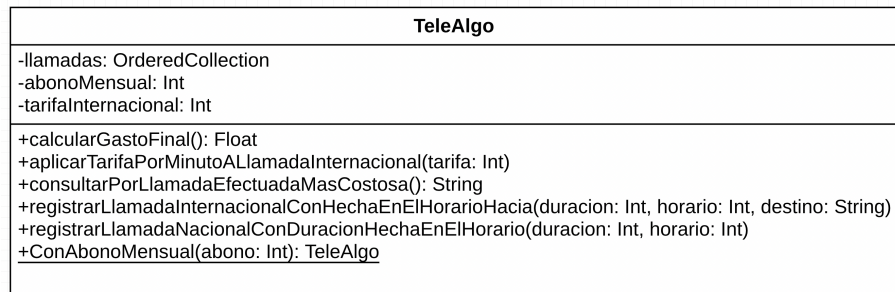


Figura 1: Diagrama de la clase TeleAlgo.

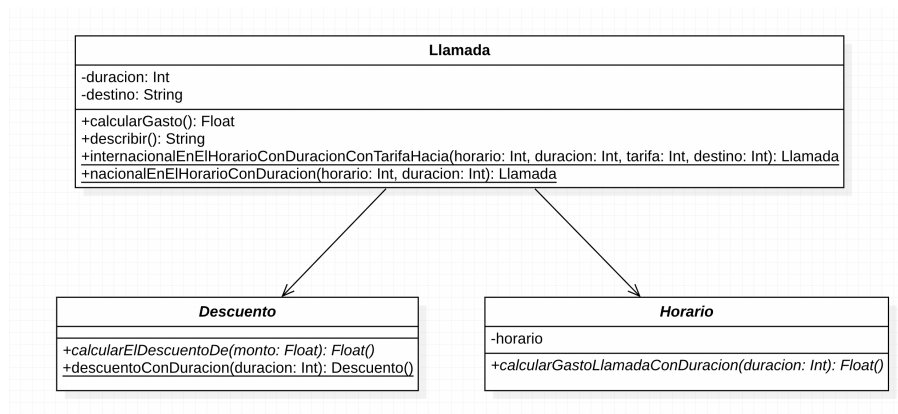


Figura 2: Diagrama de la clase Llamada.

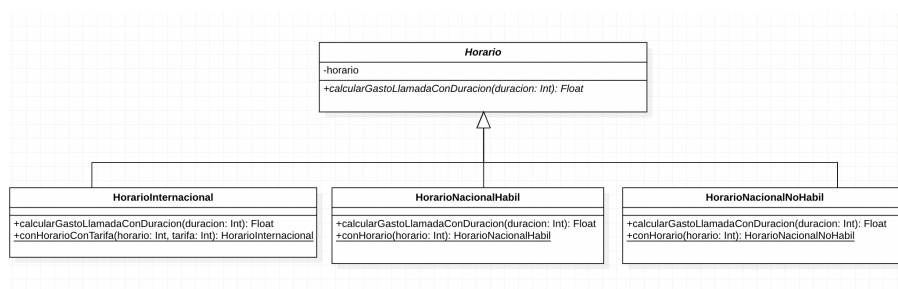


Figura 3: Diagrama de la clase abstracta Horario.

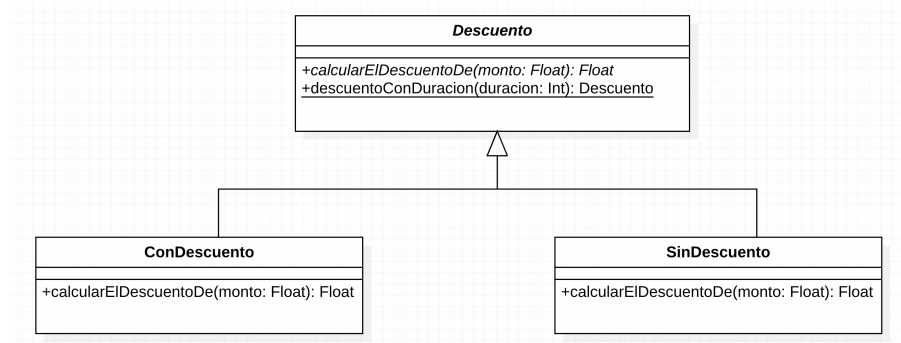


Figura 4: Diagrama de la clase abstracta Descuento.

Dejé explícitos en los diagramas los constructores con parámetros en todas las clases, porque creo que es lo correcto para la visualización del código y su fácil uso. Además, de no ser por ellos, clases como Descuento, serían interfaz.

## 5. Detalles de implementación

### 5.1. Explicación de la clase TeleAlgo

Clase principal del modelo, con la cual interactúa el usuario en los diagramas de secuencia, y no puede interactuar con otra que no sea esta para no romper el encapsulamiento. Usa los atributos (todos privados): llamadas (una OrderedCollection de Llamada), un entero que representa el abono mensual que paga este servicio, y por último, un entero tarifaInternacional, que es el valor que se tiene en cuenta para el cálculo del costo de las llamadas internacionales. Sus métodos son: AplicarTarifaPorMinutoALlamadaInternacional, calcularGastoFinal (calcula el gasto de cada una de las llamadas que tiene registradas más el abonoMensual). ConsultarPorLlamadaEfectuadaMasCostosa me da la descripción de la llamada mas cara que se haya realizado, y si no tiene ninguna llamada registrada, me lo informa. Luego tengo los métodos que sirven para registrar llamadas, y que como están diferenciados entre Internacional y Nacional, eso me dio la pauta de como elegir mi modelo de Horario.

### 5.2. Explicación de la clase Llamada

La clase Llamada tiene como atributos privados, un entero que representa la duración de la llamada, y un String que respresenta el destino ('Nacional' en caso de ser registrada como nacional). Tiene tambien asociados a ella, un Horario y un Descuento. Sus métodos principales son el de calcularGasto y describir. Para calcular gasto, hace uso tanto del atributo Horario como del Descuento. Describir simplemente da la información de sus atributos. para calcularGasto, le delega a Horario, y esta responde al llamado polimórfico.

### 5.3. Explicación de la clase Horario

Es una clase clave con respecto al modelo, ya que en ella mediante la delegación, se calcula el gasto de la llamada (sin tener en cuenta el descuento). Hay tres clases hijas de la clase abstracta horario: HorarioInternacional, HorarioNacionalHabil y HorarioNacionalNoHabil. Las últimas dos diferencian el mensaje del constructor con argumento de la primera.

### 5.4. Explicación de la clase Descuento

Es una clase abstracta que responde, dependiendo si es ConDescuento o SinDescuento de una forma o de otra al mensaje calcularDescuento(monto).

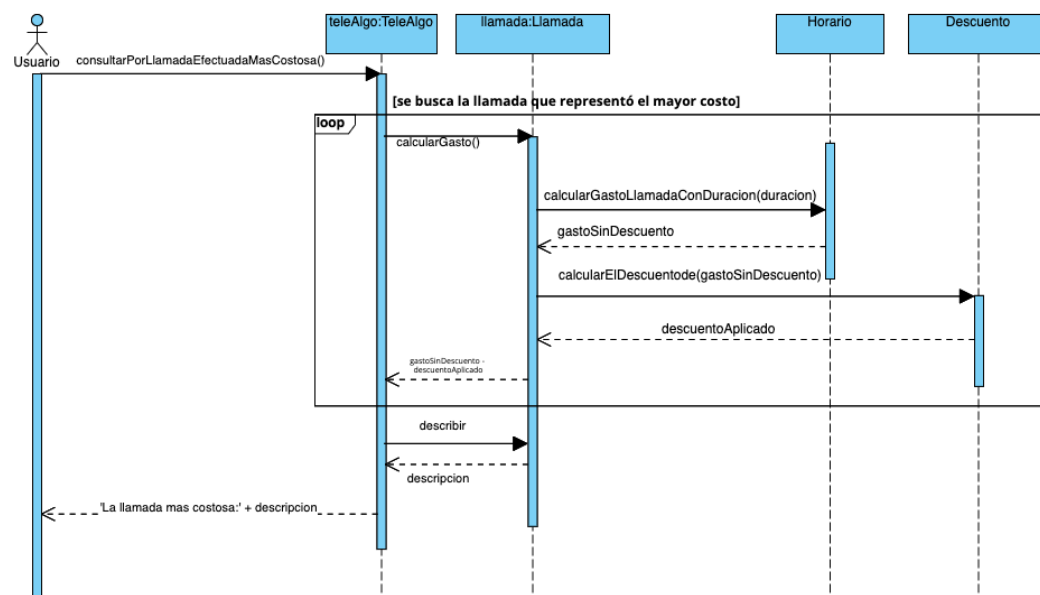
## 6. Excepciones

**HorarioInvalidoError** Esta excepción tiene como finalidad, que al momento de registrar una llamada, ya sea Nacional o Internacional, se haga siguiendo el sistema de 24 hs diarias. En el caso que al momento de que un TeleAlgo quiera registrar una llamada con un horario fuera de rango, se arrojará esta excepción. Fue la única excepción que vi necesaria su realización, porque por ejemplo para el caso de una tarifa con valor negativo, encuentre más sentido que el valor de la llamada sea gratuito (en el apartado de supuestos se explica).

## 7. Diagramas de secuencia

Visual Paradigm Online Free Edition

Consultar por la llamada más costosa, en el caso que se hayan agregado llamadas



Visual Paradigm Online Free Edition

Figura 5: Consultar por llamada más costosa en el caso que haya llamadas registradas.

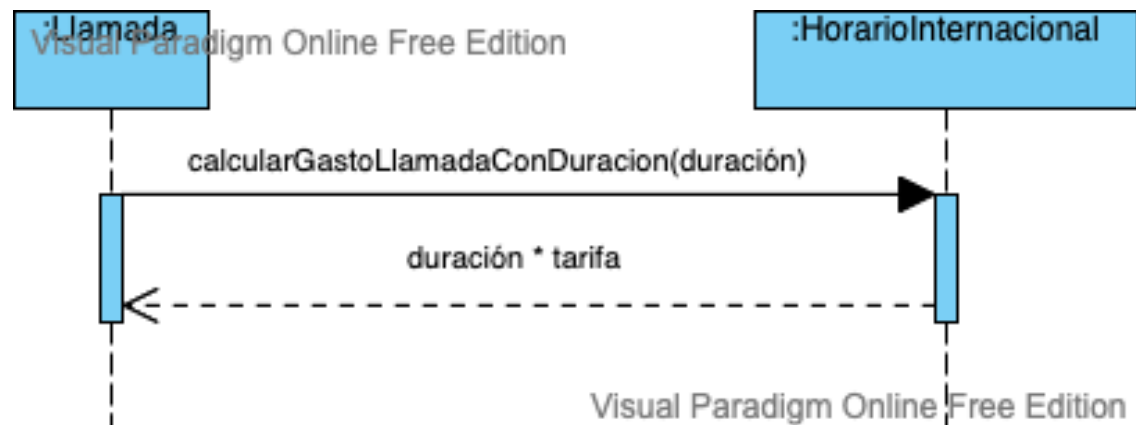


Figura 6: abstracción Horario

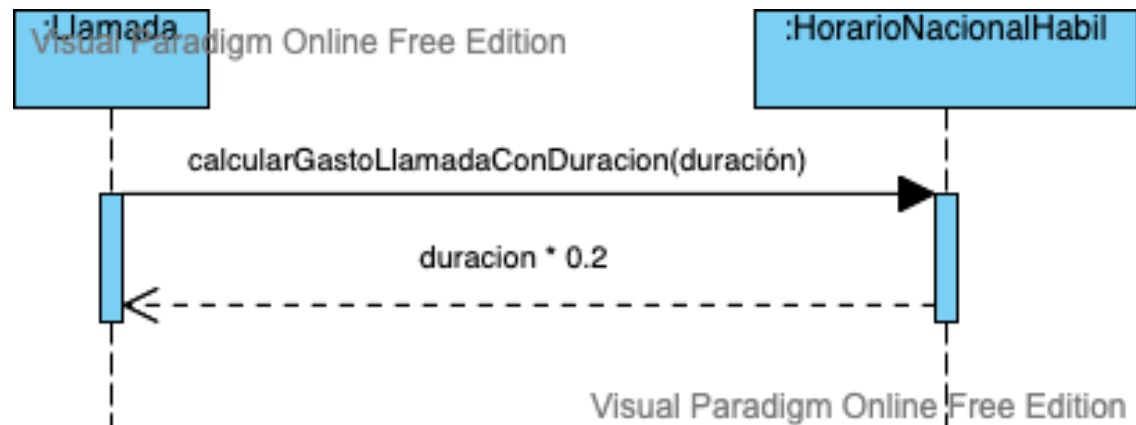


Figura 7: abstracción Horario

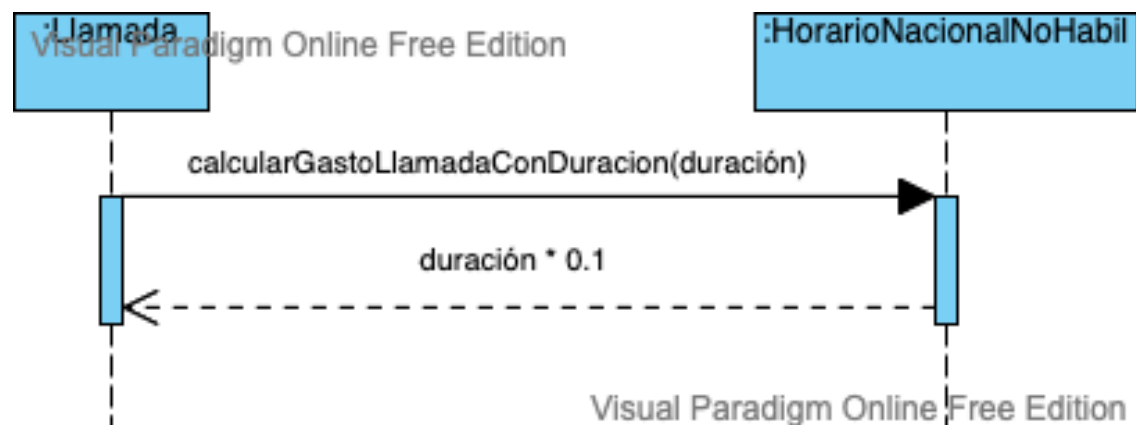


Figura 8: abstracción Horario

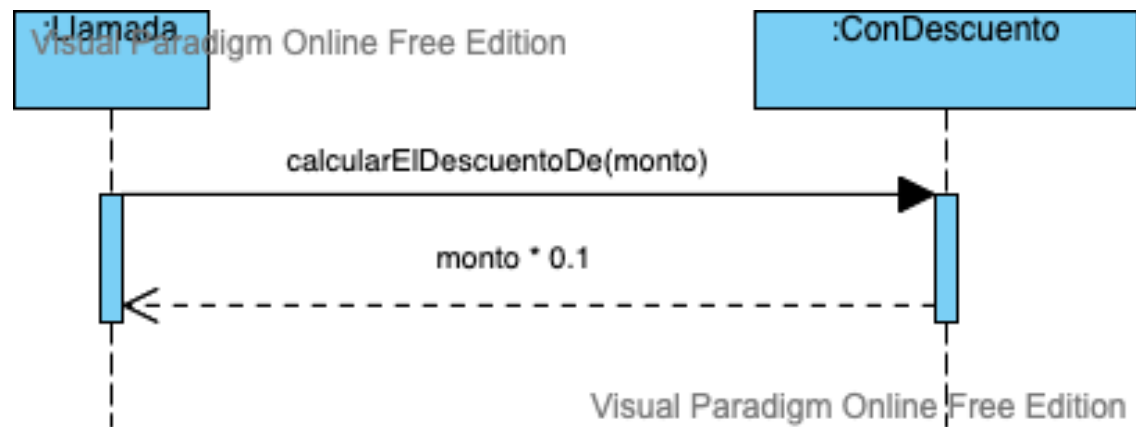


Figura 9: detenimiento en la abstracción Descuento

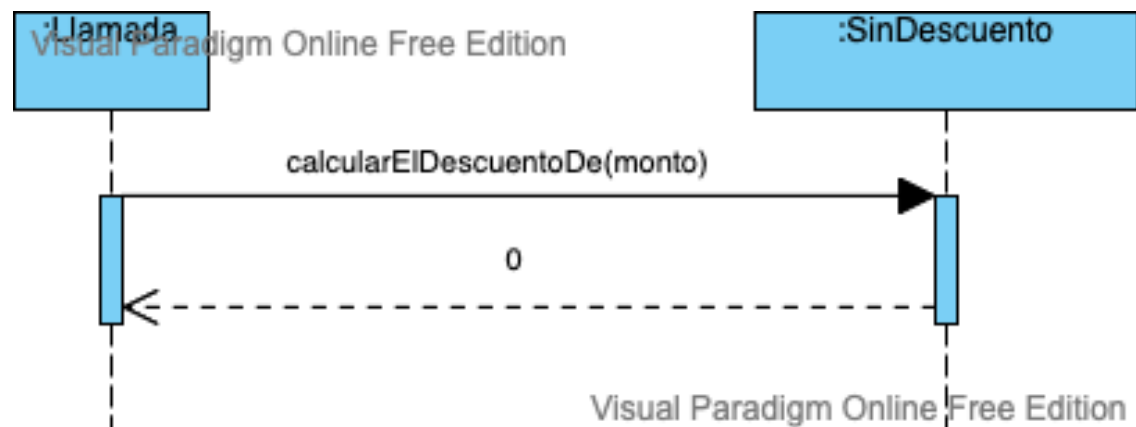
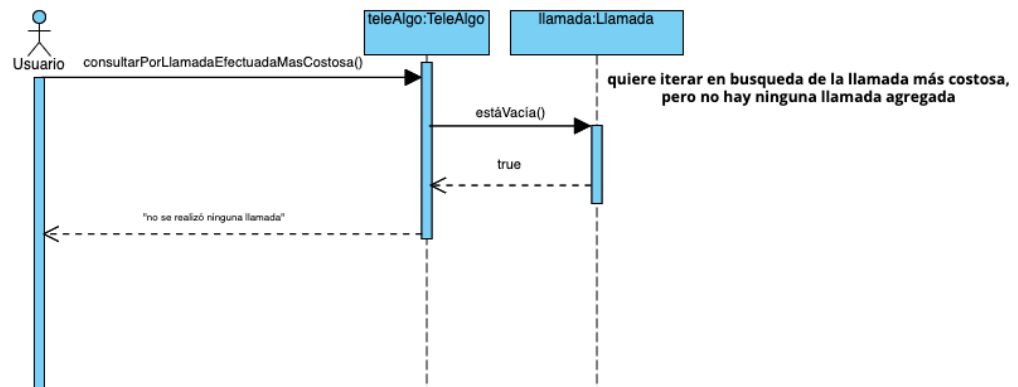


Figura 10: detenimiento en la abstracción Descuento



Visual Paradigm Online Free Edition

Consultar por la llamada más costosa, en el caso contrario, cuando no se hayan agregado llamadas

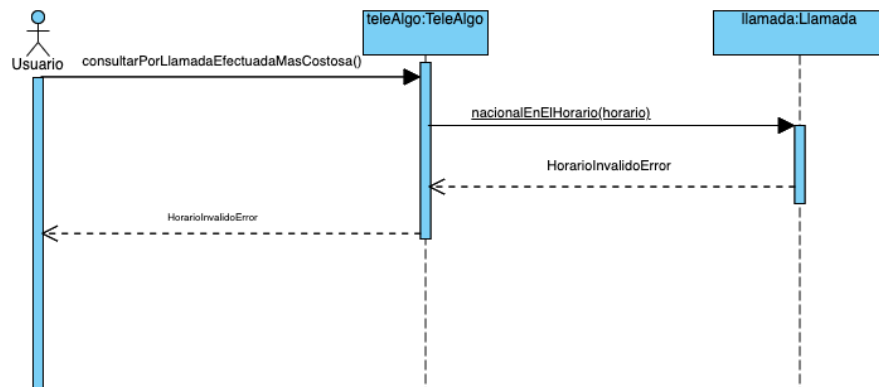


Visual Paradigm Online Free Edition

Figura 11: Consultar por llamada más costosa en el caso que NO haya llamadas registradas

Visual Paradigm Online Free Edition

Se quiere registrar una llamada Nacional en un horario invalido, se levanta HorarioInvalidoError



Visual Paradigm Online Free Edition

Figura 12: HorarioInvalidoError

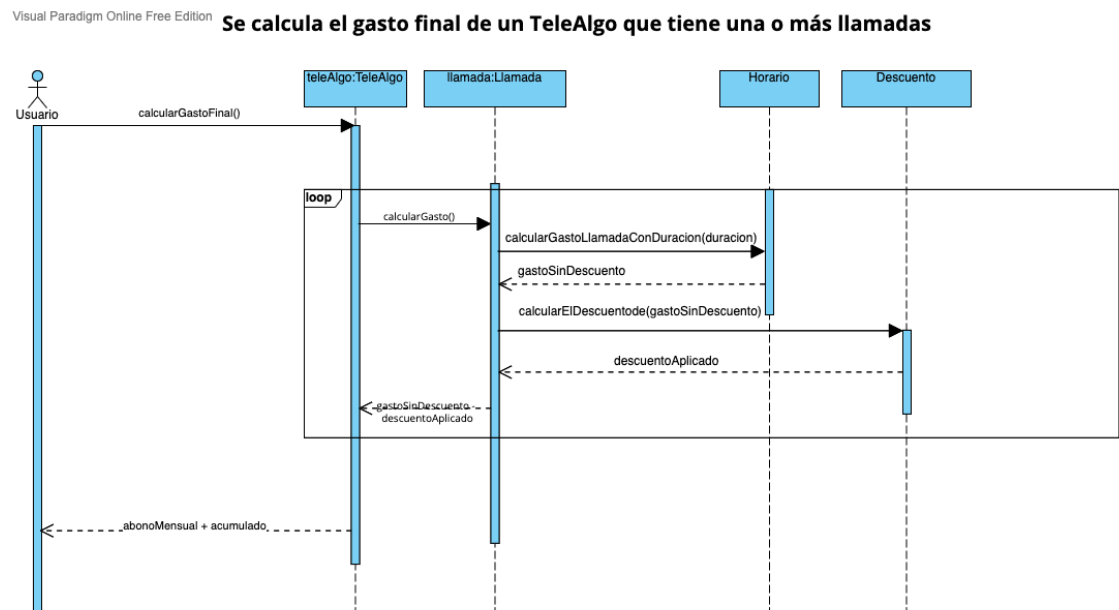


Figura 13: GastoFinal

## 8. MuTalk y code coverage

Con respecto a la entrega anterior, pude resolver los problemas que tenía en Mutalk.

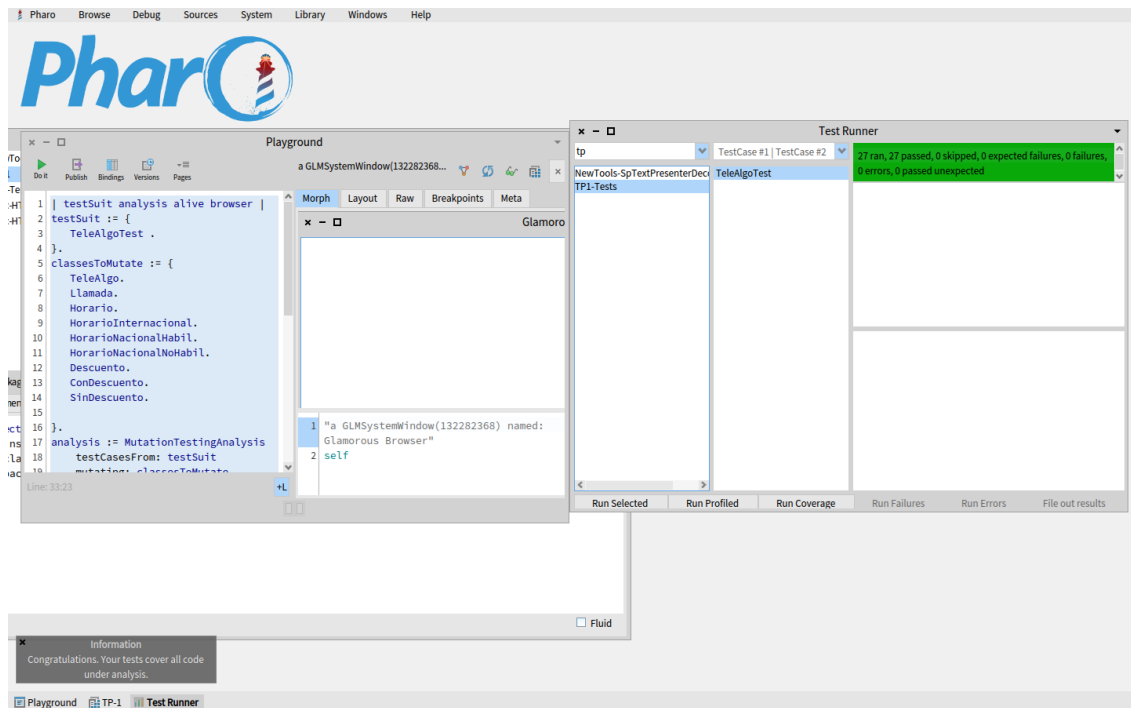


Figura 14: No sobrevivió ningún mutante y la cobertura del código me dio máxima.