

Trabajo Práctico 1 — Smalltalk

[7507/9502] Algoritmos y Programación III

Curso 1

Primer cuatrimestre de 2022

Alumno:	FÁBREGAS, Alejo Valentín
Número de padrón:	106160
Email:	afabregas@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
3. Modelo de dominio	2
4. Diagramas de clase	3
5. Detalles de implementación	4
5.1. Constructores de las subclases de Llamada	4
5.2. Implementación de otras clases adicionales	4
6. Excepciones	5
7. Diagramas de secuencia	5

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 telefonía en Pharo utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

2. Supuestos

Debido a que las pruebas implementadas por la cátedra pueden ser ambiguas, y estas definen los requisitos del sistema TeleAlgo, se optó por realizar ciertos supuestos en la implementación, que se enuncian a continuación:

- Se supone que el sistema TeleAlgo es único para cada usuario, por lo que allí se registrarán todas sus llamadas nacionales e internacionales, incluyendo un abono mensual fijo.
- Se supone que los tipos de datos ingresados al sistema TeleAlgo serán del tipo correcto. Esto no se verificará, sin embargo, si son del tipo correcto, se va a chequear que los datos ingresados sean válidos en el contexto en el que se van a utilizar.
- Se supone que las llamadas internacionales no varían su costo según el horario en el que se hayan efectuado. Se van a facturar según el costo de la tarifa internacional sin importar si fueron realizadas en horario hábil o no hábil.
- Se supone que el descuento del 10 por ciento a llamadas de más de 30 minutos se aplica a ambos tipos de llamadas, nacionales e internacionales.
- Se supone que las llamadas nacionales tienen predefinido como país destino a Argentina, por lo que si la llamada más costosa de un usuario es nacional, se indica que fue una llamada hacia Argentina.
- Se supone que los horarios serán únicamente números enteros de 1 a 24.
- Se supone que el horario hábil no incluye las 20 horas.
- Se supone que si un usuario quiere chequear cuál es la llamada más costosa, pero no hay llamadas en el registro, se notificará esto con un mensaje, y no se devolverá información de una llamada.

3. Modelo de dominio

De acuerdo con los requerimientos definidos por las pruebas de la cátedra, se desarrolló un modelo de dominio que pueda resolver los problemas requeridos.

La clase TeleAlgo representa abstractamente el sistema de telefonía TeleAlgo, que se otorga a cada usuario. Allí, cada usuario va a poder realizar dos tipos de llamadas: nacionales e internacionales, cada una con sus particularidades. Los usuarios tendrán que pagar un abono mensual fijo, y además tendrán una tarifa internacional de acuerdo al servicio que hayan contratado con la empresa de telefonía. El servicio TeleAlgo también cuenta con un registro de llamadas, donde se van almacenando todas las llamadas efectuadas por el usuario, quien puede consultar cuál fue la llamada más costosa que realizó.

La clase TeleAlgo engloba todas las funcionalidades del sistema, y se encarga de delegar la mayor cantidad de responsabilidades posibles a otras clases, a la hora de realizar llamados o consultar el registro de llamadas, por ejemplo.

Las clases LlamadaNacional y LlamadaInternacional heredan la mayor parte de sus métodos y atributos de su clase madre abstracta Llamada. Éstas modelan las llamadas nacionales e internacionales, respectivamente, cada una con sus propios requisitos, como su facturación, su destino, etc. Por esto, son capaces de responder con comportamientos distintos a los mismos mensajes, por lo que aquí aplicamos otro de los pilares de la programación orientada a objetos, el polimorfismo.

Como cada clase tiene sus responsabilidades bien definidas, y se aplica la delegación acordeamente, se asegura el encapsulamiento, por lo que no revelamos detalles de implementación a clases

que no deberían saberlos. Por ejemplo, TeleAlgo no sabe como RegistroLlamadas implementa ese historial de llamadas, sin embargo cumple con todas las responsabilidades que TeleAlgo le delega.

4. Diagramas de clase

A continuación se adjuntan los diagramas de clases que representan el modelo desarrollado. El primer diagrama muestra las principales clases del modelo mientras que el segundo muestra las excepciones junto con las clases con las que se relacionan. Por motivos de claridad se omitieron los getters y setters que no aportan información a los diagramas, algunos nombres fueron acortados y algunos métodos fueron omitidos ya que no eran muy relevantes.

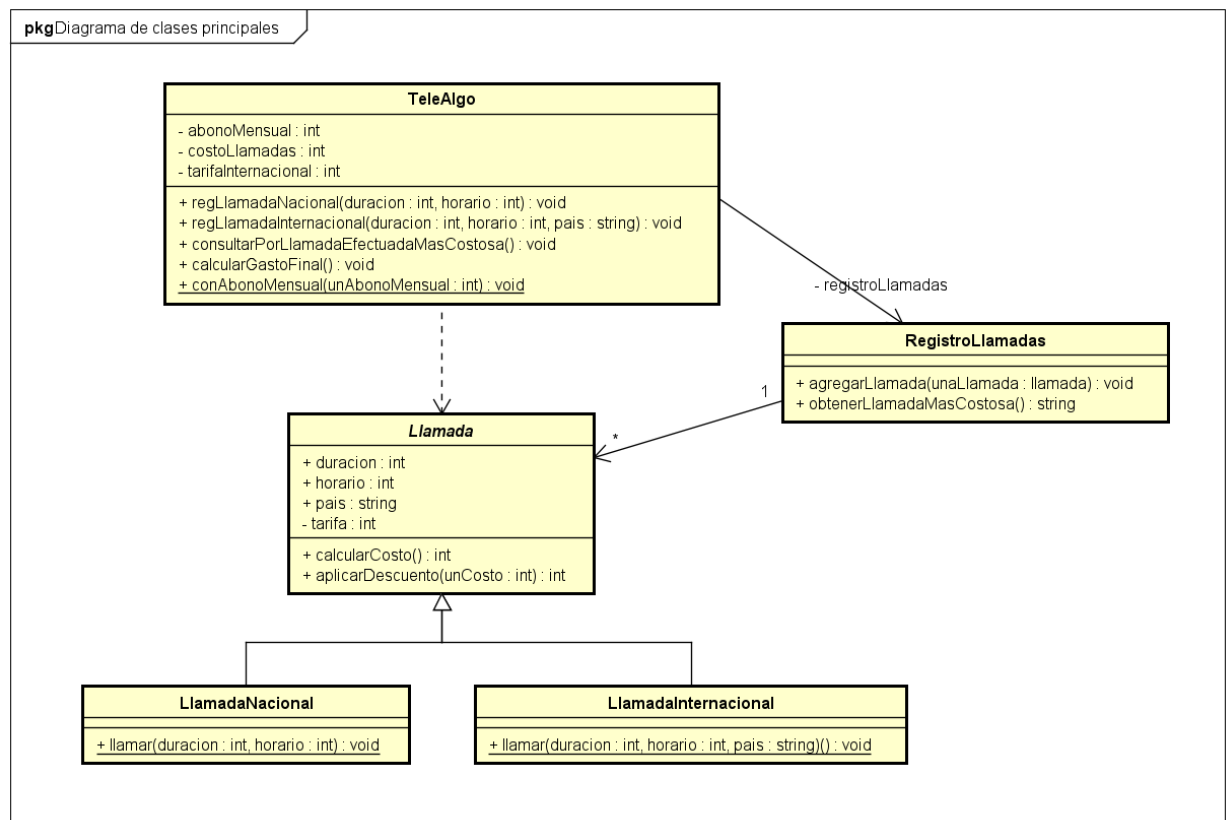


Figura 1: Diagrama de clases principales.

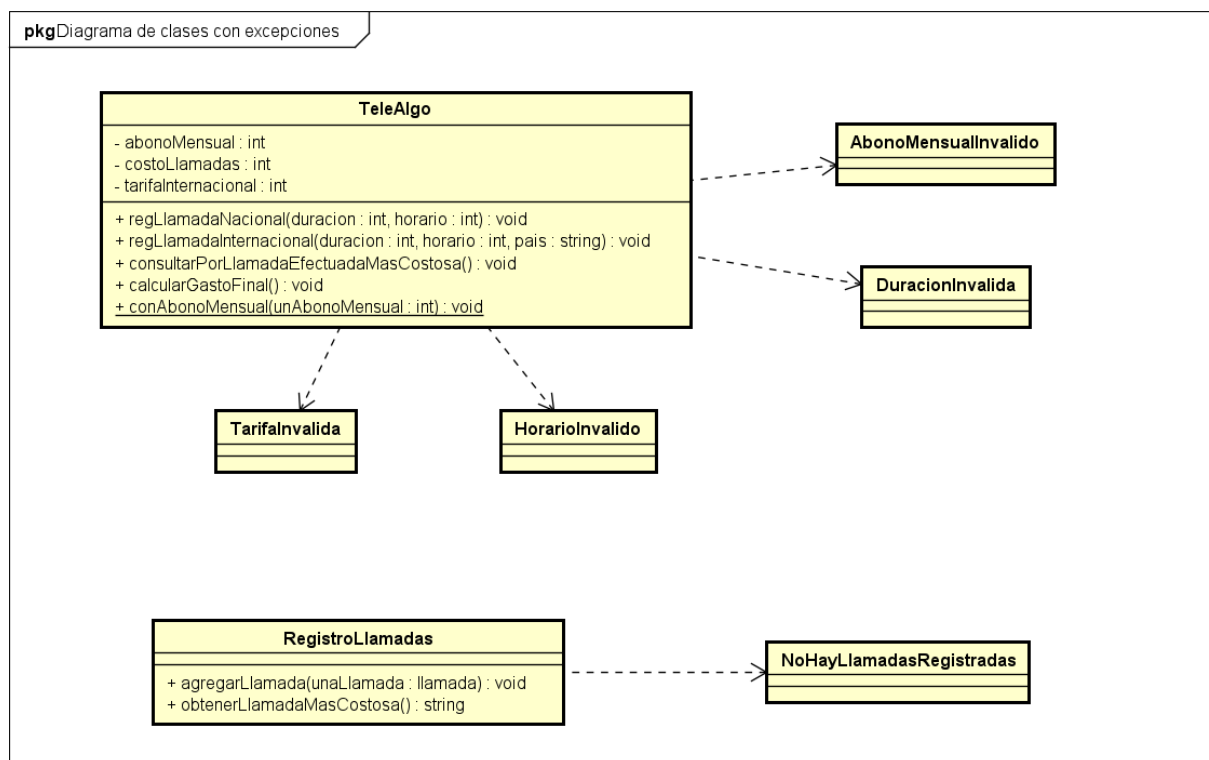


Figura 2: Diagrama de clases con excepciones.

5. Detalles de implementación

5.1. Constructores de las subclases de Llamada

Para cumplir con los requisitos de cada tipo de llamada, se implementaron dos constructores de las clases LlamadaNacional y LlamadaInternacional, y no uno genérico de Llamada, y es lo que más diferencia estas dos clases. Sus constructores son la principal diferencia entre las dos clases, ya que definen sus atributos de acuerdo a los requerimientos de cada tipo de llamada, por ejemplo, su tarifa o lugar de destino. A continuación se adjunta el código del constructor de LlamadaInternacional, un método de clase, a modo de ejemplo.

```

llamarConDuracion: unaDuracion EnElHorario: unHorario Hacia: unPais
| unaLlamadaInternacional |
unaLlamadaInternacional := self new.
unaLlamadaInternacional establecerDuracion: unaDuracion.
unaLlamadaInternacional establecerHorario: unHorario.
unaLlamadaInternacional establecerPais: unPais.
~ unaLlamadaInternacional.
  
```

5.2. Implementación de otras clases adicionales

La principal razón por la que no se implementaron otras clases además de las ya descritas, fue principalmente para no crear clases anémicas. Por ejemplo, una opción fue crear las clases FacturaciónNacional y FacturaciónInternacional para calcular los costos de las llamadas nacionales e internacionales respectivamente, pero descubrí que al hacer eso se implementaban clases que

recibían valores de las llamadas y simplemente realizaban una cuenta, sin tener comportamiento propio, y con un tiempo de vida demasiado corto. Por esto, se decidió excluirlas de la solución. Algo similar se podría pensar con una clase Descuento.

6. Excepciones

AbonoMensualInvalido Se lanza esta excepción cuando se ingresa un abono mensual de TeleAlgo que es negativo.

DuracionInvalida Se lanza esta excepción cuando se ingresa una duracion de una llamada que es negativa.

HorarioInvalido Se lanza esta excepción cuando se ingresa un horario de una llamada que no está dentro del rango aceptable definido en las pruebas (entre 1 y 24 horas).

TarifaInvalida Se lanza esta excepción cuando se ingresa una tarifa de llamada internacional que es negativa.

NoHayLlamadasRegistradas Esta exepción ocurre cuando queremos obtener la llamada más costosa de TeleAlgo pero no hay llamadas guardadas en el registro de llamadas. En lugar de informar sobre la llamada más costosa, el tratamiento es dar un mensaje que indica que no hay llamadas en el registro.

7. Diagramas de secuencia

El siguiente diagrama de secuencia está inspirado en la prueba 2 de TeleAlgoTest, y consiste en registrar una llamada nacional en TeleAlgo, con todo lo que eso implica. Dicha llamada se va a guardar en el registro, y además le vamos a pedir el costo total a TeleAlgo.

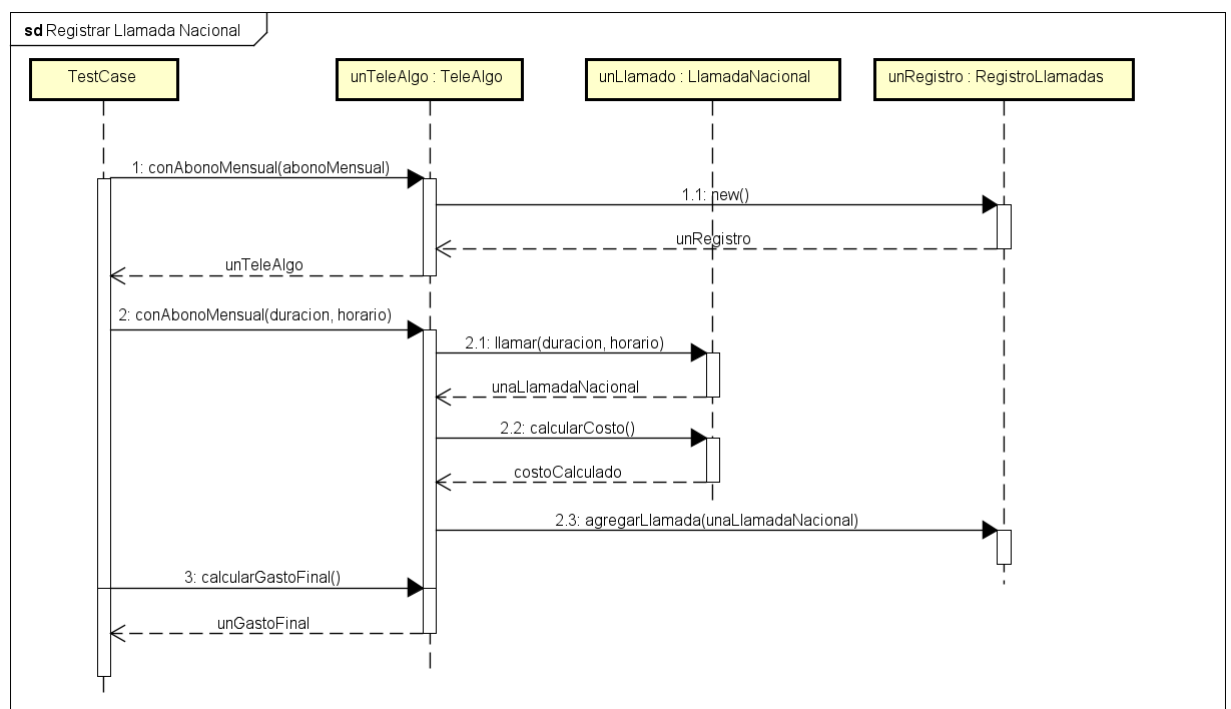


Figura 3: Diagrama de secuencia de realizar un llamado nacional.

El siguiente diagrama de secuencia está inspirado en la prueba 3 de RegistroLlamadasTest, y consiste en instanciar algunas llamadas y guardarlas en un registro de llamadas, para luego pedirle que nos dé una descripción de la llamada más costosa del registro.

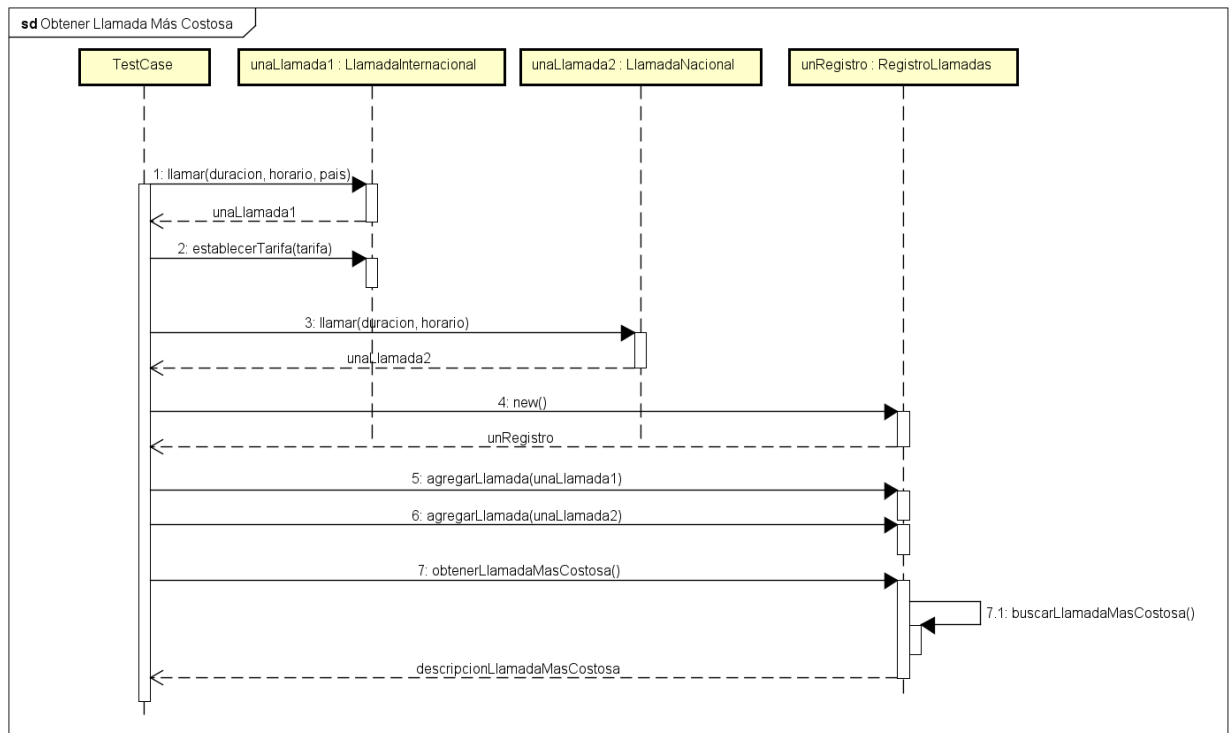


Figura 4: Diagrama de secuencia de obtener la llamada más costosa del registro.