

Trabajo Práctico 1 — Smalltalk

[75.07/95.02] Algoritmos y Programación III

Curso 1

Primer cuatrimestre de 2022

Alumno:	CAMPILLAY, Edgar Matías
Número de padrón:	106691
Email:	ecampillay@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
3. Modelo de dominio	3
3.1. TeleAlgo	3
3.2. GestorDeLlamadas	3
3.3. Registro	3
3.4. Llamada	3
4. Diagramas de clase	4
5. Detalles de implementación	5
5.1. Gestores	5
5.2. Llamada	5
6. Excepciones	6
7. Diagramas de secuencia	7

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 para simular el uso de una operadora telefónica utilizando en Pharo los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

2. Supuestos

Supuestos generados a partir de situaciones no contempladas en la especificación de pruebas iniciales de la cátedra:

- **Solo hay dos tipos de abonos mensuales: \$100 y \$200.**
- El abono de \$200 ofrece descuentos sobre la duración original de las llamadas e incluye de manera acumulativa los descuentos del plan más básico, \$100.
- La operadora puede cambiar las tarifas internacionales pero si la llamada ya había realizado, el precio de la misma nunca va a cambiar.
- La duración de las llamadas están medidas en minutos, el horario de la misma en el formato de 24 horas.
- precio de las llamadas para el plan de \$200 no se da en función de su duración, sino, en función de la **cantidad de minutos a cobrar**:

- Minutos cobrados llamada internacional:

$$\frac{2}{3}duracin \tag{1}$$

- Minutos cobrados llamada local:

$$\frac{1}{2}duracin \tag{2}$$

- Llamadas del tipo internacional tienen una tarifa base igual a \$1 que puede ser modificada a futuro.
- Llamadas internacionales también poseen descuentos a partir de los 30 minutos de duración (no inclusive).
- Dos usuarios no pueden tener diferentes tarifas a llamadas internacionales, **si se actualiza la tarifa de un solo usuario, se actualizarán las tarifas de todo el sistema.**
- Así como puedo calcular la llamada más costosa, puedo calcular la llamada más barata.
- Si la llamada más costosa/barata es local, la descripción de la misma no indicará que fue a “Argentina”, sino, hacia “destino local”.
- **Si no hay un abono válido propuesto, entonces el plan dado de alta cuenta con el abono más básico.**
- Una llamada no puede durar mas de 300 minutos (5 horas), ni tener duración negativa.
- Si no se efectuaron llamadas, al realizar una consulta, la misma avisará al usuario.
- **No puede haber tarifas internacionales nulas, ni negativas.**
- Si una llamada tiene duración 0, se cobra el equivalente a 1 minuto.

3. Modelo de dominio

El modelo consta de cuatro clases principales:

3.1. TeleAlgo

La clase principal, cuyas responsabilidades son:

- Permitir que el usuario interactúe con el programa,
- Simular el uso de una operadora móvil,
- Presentar los datos de manera clara y legible,
- Permitir al usuario acceder a información sobre sus consumos y deudas.

Para lograr las tareas mencionadas la clase hace uso de un gestor dado en función del tipo de plan elegido por el usuario, \$100 ó \$200.

3.2. GestorDeLlamadas

Clase abstracta, sus hijas -GestorPlan100 y GestorPlan200- son utilizadas únicamente por la clase TeleAlgo, las responsabilidades de la clase padre e hijas son:

- Puente entre la interfaz del operador, TeleAlgo, y el registro de llamadas,
- Realización de llamadas,
- Aplicación de descuentos propios del plan seleccionado.

Para llevar a cabo sus tareas esta clase utiliza un registro.

3.3. Registro

La clase que almacena y maneja las llamadas, las responsabilidades son:

- Agregar las llamadas que les manda el gesto al historial,
- Buscar la llamada más costosa/barata,
- Obtener un total del precio de todas las llamadas en el historial.

3.4. Llamada

Si bien no es clase principal, la realización de llamadas es prácticamente el objetivo del modelo. Esta clase posee tarifa, duración, un horario de emisión, cantidad de minutos a pagar y un destino (el cual define el tipo de llamada a instanciar). Las responsabilidades de esta clase son:

- Realizar un calculo en función de los minutos a pagar y la tarifa definida,
- Verificar si se puede dar un descuento en función de su duración,
- Instanciar la llamada correcta según su tipo y horario,
- Devolver una descripción detallada de si misma.

4. Diagramas de clase

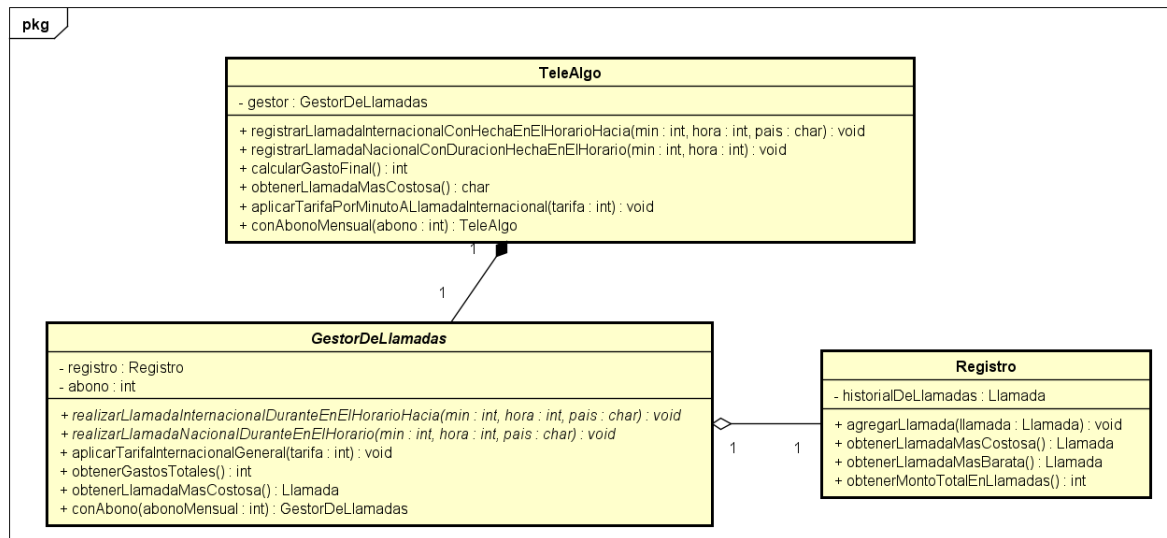


Figura 1: Diagrama principal de la clase TeleAlgo.

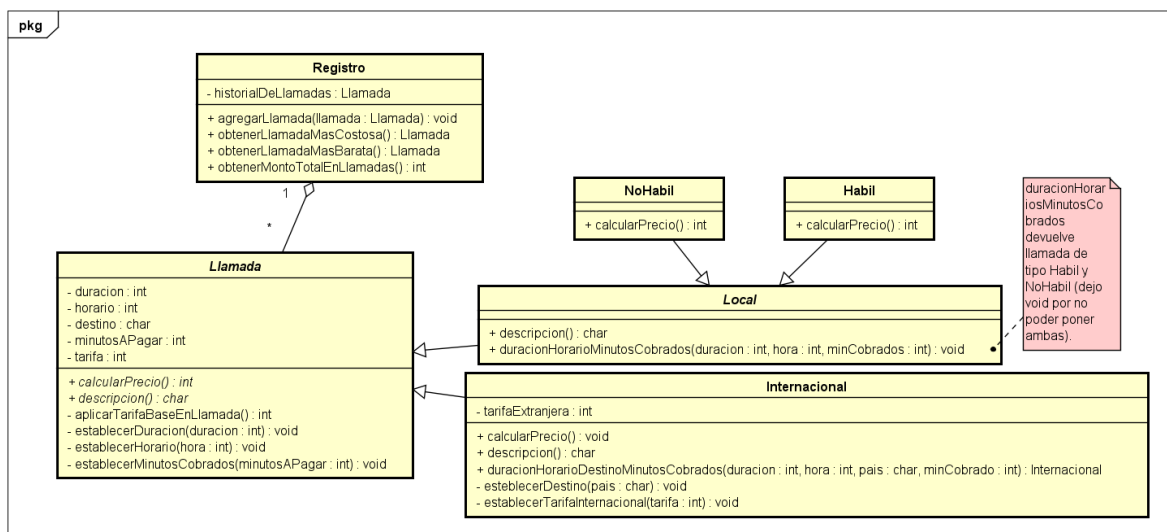


Figura 2: Diagrama de Llamadas y sus tipos.

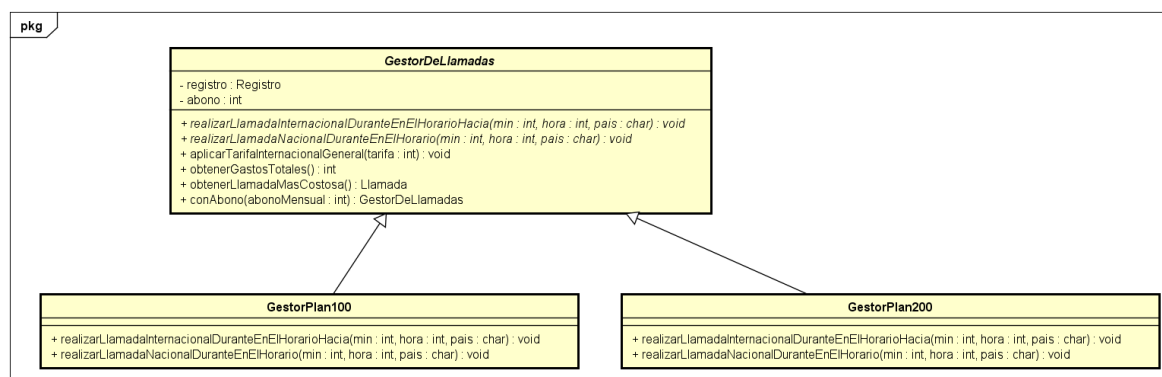


Figura 3: Diagrama de los tipos de gestores según el plan contratado.

5. Detalles de implementación

5.1. Gestores

Los gestores son creados según el plan seleccionado. Utilizando herencia se permitió que exista un plan mayor, de \$200 que permita el usuario acceder a descuentos por sobre los minutos que habla, sin importar la duración de las llamadas (acumulable con el beneficio original del %10 de descuento a llamadas de larga duración, mayores a 30 minutos). Para este descuentos se implementaron los minutos a ser cobrados.

```

realizarLlamadaInternacionalDurante: unosMinutos EnElHorario: unaHora Hacia: unPais
    registro agregarLlamada: (Internacional duracion: unosMinutos Horario: unaHora
Destino: unPais MinutosCobrados: (unosMinutos-(unosMinutos/3))).
  
```

Como se puede ver en la parte de minutos cobrados el plan de \$200 da un descuento del %33 aproximadamente para cada llamada internacional. Esta implementación abre la posibilidad de agregar futuras implementaciones como tener X cantidad de minutos libres o agregar más planes con distintos niveles de descuentos de una manera relativamente simple, sin perder la duración original de cada llamada.

5.2. Llamada

La clase abstracta **Llamada** cuenta con dos tipos principales de llamadas, **Local** e **Internacional**, de llamadas donde se utilizó herencia para evitar repetir código y marcar las diferencias entre los comportamientos de los distintos tipos. La clase **Internacional**, es una clase **no abstracta**, esta posee una tarifa base de \$1 que luego puede ser cambiada por la operadora, tiene una descripción ligeramente diferente a la llamada de tipo Local, y tener un destino establecido y diferente del país de origen, Argentina. La clase **Local** es una clase **abstracta** padre de **Hábil** y **NoHábil** la misma posee un método de instancia que permite a través de una condición en función del horario, devolver una instancia de una de las hijas. Estas clases nacieron con la necesidad de eliminar un condicional que analizaba el horario de la llamada cada vez que se debía calcular su precio. La clase **NoHábil** tiene una tarifa de \$0.1 por minuto mientras que la clase **Hábil** mantiene el valor duplicado (\$0.2).

Código para cálculo previo:

```

calcularPrecio
    |precioAuxiliar|
    precioAuxiliar := self aplicarTarifaBaseEnLlamada.
    (horario between: 8 and: 20) ifTrue: [ precioAuxiliar := precioAuxiliar * 2. ].
  
```

```
        ^ precioAuxiliar.  
Código actual:  
calcularPrecio  
    "caso NoHabil."  
    ^ self aplicarTarifaBaseEnLlamada.  
calcularPrecio  
    "caso Habil."  
    ^ self aplicarTarifaBaseEnLlamada * 2.
```

6. Excepciones

CampoInvalido Padre de 7 tipos de excepciones diferentes, las cuales cuentan con un único método de clase que indica la validez ó invalidez del parámetro asignado.

AbonoInvalido Tiene la responsabilidad de revisar un abono, indicar si este es válido y alertar al usuario en caso negativo. Si bien la alerta se ve claramente desde el gestor, en TeleAlgo no se puede apreciar la misma porque en caso de saltar, el sistema la trata asignándole el plan más básico a la línea a instanciar.

DestinoInvalido Permite al usuario saber si al momento de realizar una llamada internacional, envió un destino incorrecto. Destinos incorrectos son considerados en el caso de cadenas de texto, aquellos que cuenten con símbolos o números dentro de la misma, y aquellos objetos diferentes a una cadena de texto.

GestorInvalido Revisa que el momento de asignar un gestor, este efectivamente se trate de una clase hija de GestorDeLlamadas, y no cualquier objeto.

HorarioInvalido Dado un horario, se alertará al usuario sobre si este es incorrecto o no, basándose en el formato de 24 horas, además de considerar invalido a todos aquellos objetos que no sean un número válido entre 1 y 24.

MinutosACobrarInvalidos Realiza una comprobación sobre la cantidad de minutos a ser vendidos por la operadora, solo revisa que estos no sean negativos y que efectivamente se trate de un número y no otro objeto aleatorio.

MinutosInvalidos Al igual que el ítem anterior revisa que no se estén instanciando llamadas con duraciones negativas, y que estas efectivamente estén recibiendo parámetros correctos.

TarifaIternacionalInvalida Revisa que la potencial tarifa internacional sea correcta, evitando de esa manera, que hayan llamadas gratuitas ó que se este generando deuda negativa.

DuracionExcedida Comprueba que las llamadas no sean infinitas, dándole un límite de 300 minutos por llamada, 5 horas.

RegistroAceptaSoloLlamadas Sirve para asegurar que solo instancias de la clase Llamada lleguen al historial.

NoSeRealizaronLlamadas Alerta al usuario sobre el intento de buscar una llamada en un registro que está vacío.

7. Diagramas de secuencia

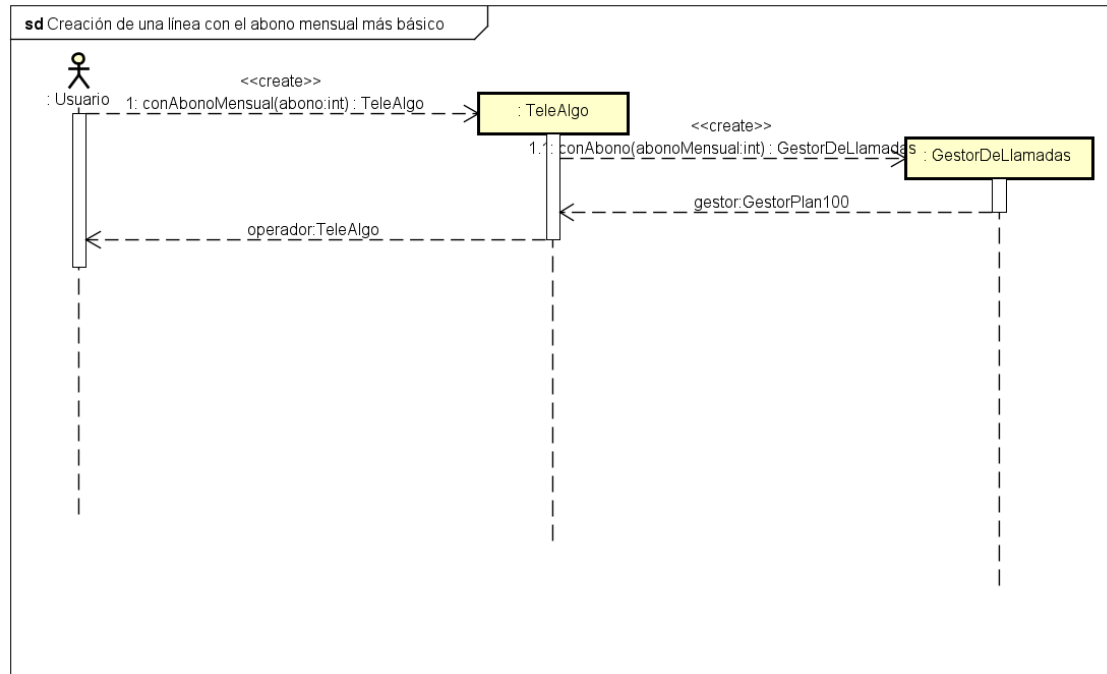


Figura 4: Creación de una línea con el abono mensual más básico, \$100.

De manera análoga la clase Gestor a través de su método de instancia puede devolver el gestor del plan de \$200, GestorPlan200, para el plan más costoso.

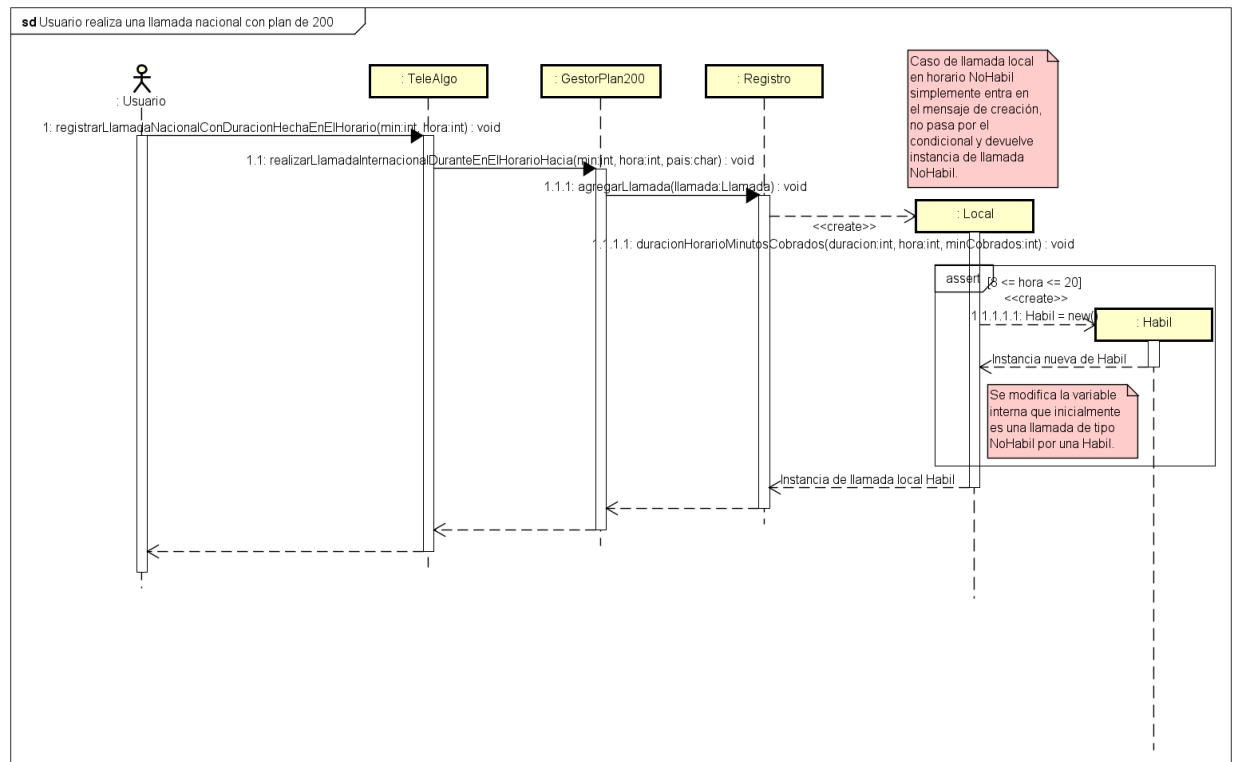


Figura 5: Realización de llamada con plan de \$200.

También de manera análoga con el plan de \$100 sucede exactamente casi la misma secuencia, con la diferencia que cuando el gestor decide realizar la llamada en el plan básico no realiza descuentos sobre los minutos a cobrar, mientras que en el plan *premium*. Y si decidiera realizar una llamada en un horario no hábil, entonces no pasaría nunca por el condicional.

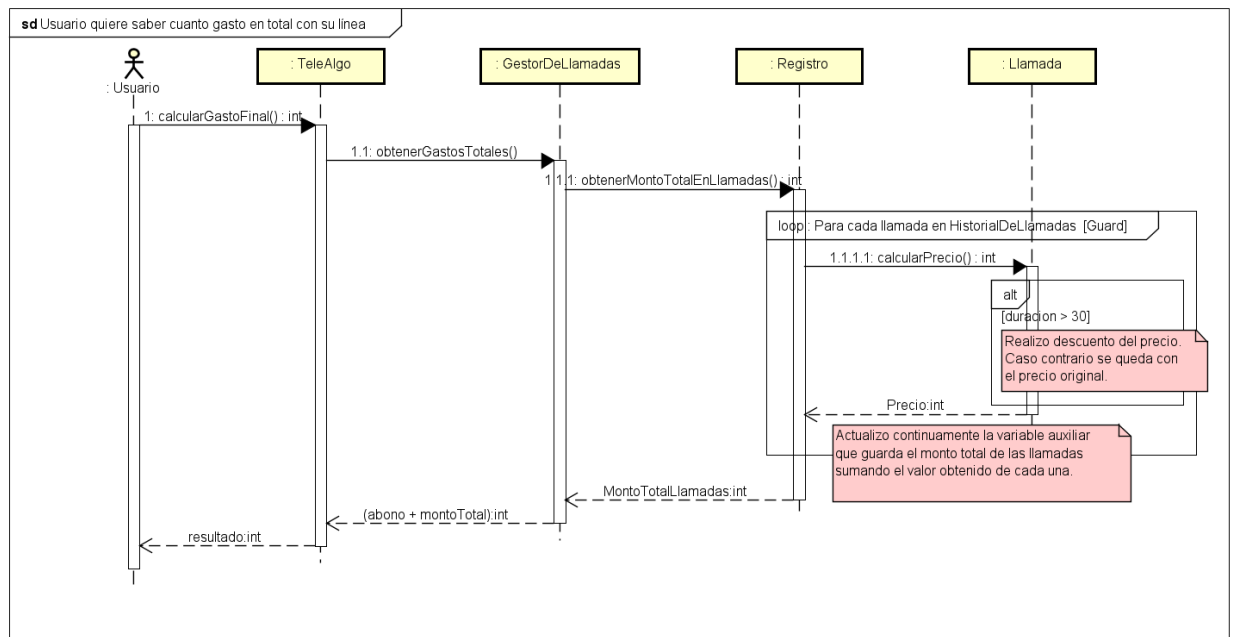


Figura 6: Usuario intenta conocer sus gastos totales en la línea.

Aclaración, donde se habla de GestorDeLlamadas y Llamadas se hace referencia a las clases hijas que sean parte del modelo (ya que las clases padres al ser abstractas nunca se irán a instanciar). En el caso del gestor este puede ser el GestorPlan100 ó GestorPlan200. Y en el caso de Llamada, dentro del bucle para cada elemento del historial del registro, una instancia de *OrderedCollection*, se envía el mismo mensaje a todas la llamadas (hijas de la clase llamada), independientemente de que estas sean, Habil, NoHabil ó Internacional.