FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Trabajo Práctico 1 — Smalltalk

[7507/9502] Algoritmos y Programación III
Curso 2
Primer cuatrimestre de 2021

Alumno:	Cañón Bello Cristian Felipe
Número de padrón:	100168
Email:	ccanon@fi.uba.ar

Índice

1. Introducción

El presente informe reúne la documentación de la solución del primer trabajo práctico de la materia Algoritmos y Programación III de la Universida de Buenos aires. Consiste en desarrollar una aplicación de un sistema para la determinación de la posible circulación de una persona y manejo de grupos basado en elconfinamiento por la pandemia del Covid-19. Esta aplicación esta desarrolada en el framework Pharo con el lenguaje Smalltalk, utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

2. Supuestos

Para la realización de nuestra aplicación los supuestos que tuvimos en cuenta para se desarollo fueron los siguientes:

2.1. Delegación:

Nuestro clase Algovid tiene tres colecciones en las cuales guarda personas, burbujas y colegios referidos por el nombre de cada uno, Algovid no sabe como estan implementadas estas clases por dentro y para todos los métodos les delega a estos tres el comportamiento para la respuesta.

2.2. Carga de Datos:

La buena carga de datos por parte del usuario ya que al recibir solo Strings, para nuestro programa va a ser distinto, una Persona Juan a una juan y analogamente para los sintomas, por lo que en los Sintomas habituales es muy importante.

2.3. ¿Herencia?:

Burbujas y Colegios son grupos de personas y de burbujas respectivamente, por lo que podríamos haber creado un objeto del cual estos dos heredaran comportamiento, pero como estos tienen muy poco comportamiento no lo vi como algo necesario de implementar.
Por el contrario en diagnóstico tenemos un *estado* que puede adoptar tres valores distintos los cuales son: Sospechoso, Positivo y Sano. Estos últimos se comportan de igual manera por lo que en este caso al ser los tres un estado de diagnóstico implícitamente se crea una clase Estado que es abtracta y se refleja en mis diagramas de clases.

3. Modelo de dominio

En nuestra aplicación de algovid, delegamos todo el trabajo a los objetos que guardamos en nuestras colecciones.
Cada persona tiene una clase Síntomas que se hace cargo de los síntomas, también tiene una clase Diagnóstico que tiene un estado que va cambiando a medida de que agregamos Síntomas a la persona o cambiamos sus estados PersonaDeRiesgo y PersonalEsencial o interactuamos con persona.
Burbuja y colegio interactuan con sus integrantes(burbuja con personas y colegios con burbujas) para resonder si estaPinchada y hayClasesPresenciales, respectivamente.

4. Diagramas de clase

En nuestro diagrama podemos observar como se relacionan las clases con las otras, algunos de los métodos mas importantes y sus multiplicidades.

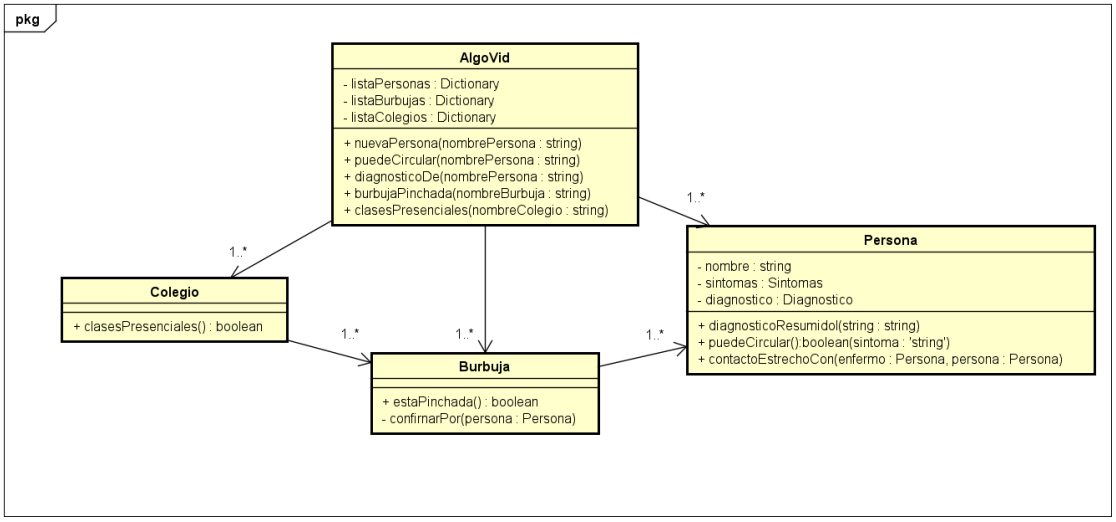

Figura 1: Diagrama de clases de Algovid.

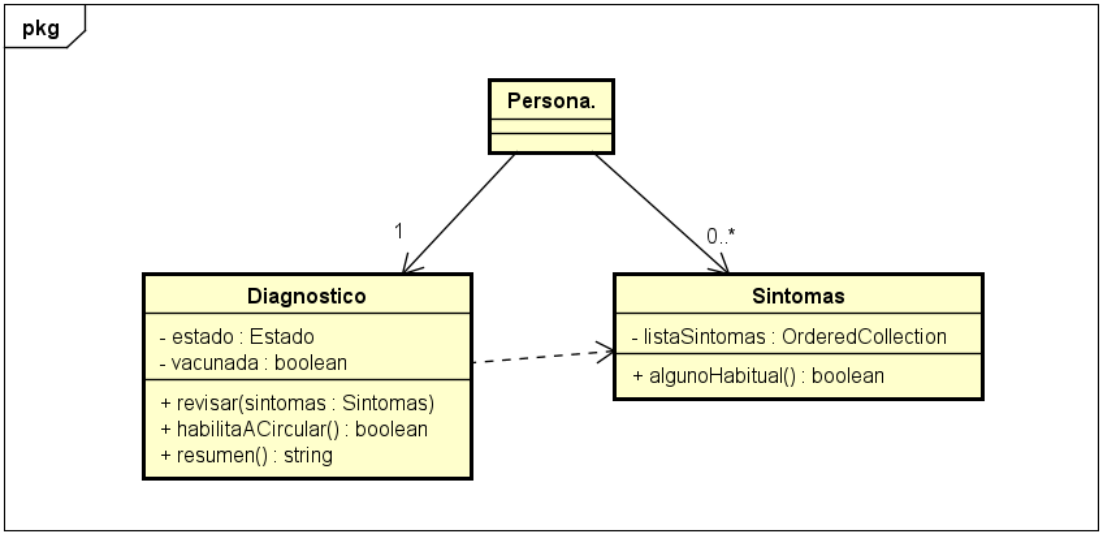

Figura 2: Diagrama de clases de Persona.

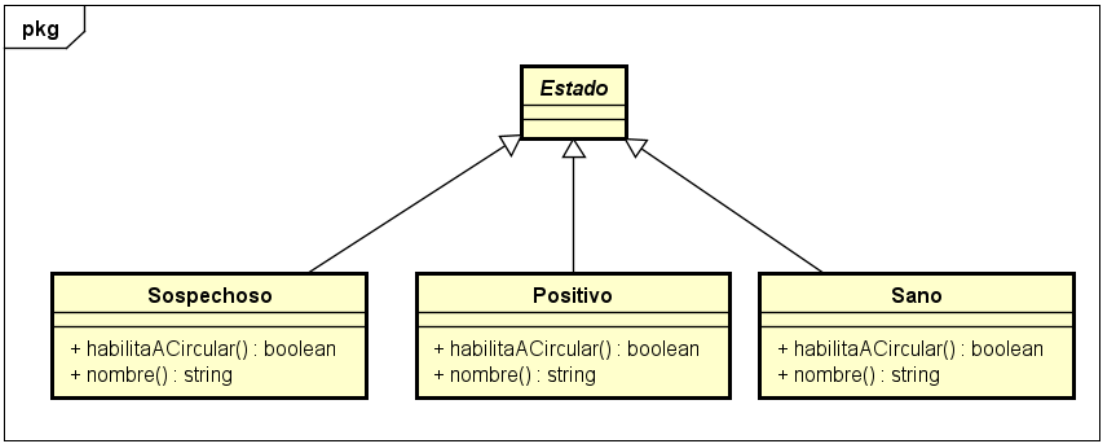

Figura 3: Diagrama de clases de *Estado*.

5. Detalles de implementación

5.1. Clase Diagnóstico

La clase diagnóstico es la más interesante de nuestra aplicación ya que es la responsable de las respuestas de la misma. En especial su método revisar, cuando a Persona se le agrega un nuevo síntoma, esta le dice a diagnóstico que revise su estado usando la lista de sintomas, ya que pudo cambiar por agregar un nuevo síntoma.

```
estado := Sospechoso new.
((listaSintomas algunoHabitual and:(listaSintomas cantidad  >= 3)) or:
(listaSintomas cantidad  >= 4)) ifTrue: [estado := Positivo new].
```

5.2. Clase Burbuja

La clase burbuja tiene dos métodos bastantes interesantes, estaPinchada y confinarPor, estos métodos son los encargados de revisar el estado de sus integrantes y en dado caso de estar pinchada(alguno positivo) cambiar el estado de todos los demas para que no puedan circular

6. Excepciones

Excepción NombreInexistente Para la validación de nombre que el programa recibe de parte del usuario, tenemos esta excepción que lo que hace es que si no existe una persona, burbuja o colegio con el nombre que se paso como parámetro devuelve el error para que el usuario sepa que el nombre que paso esta mal o no existe en la aplicación.

7. Diagramas de secuencia

En los diagramas de secuencias mostramos un caso básico y tres caso mas interesantes para ver como van interuactuando las intancias de las clases entre ellas para resolver un llamado de servicio por parte del usuario.

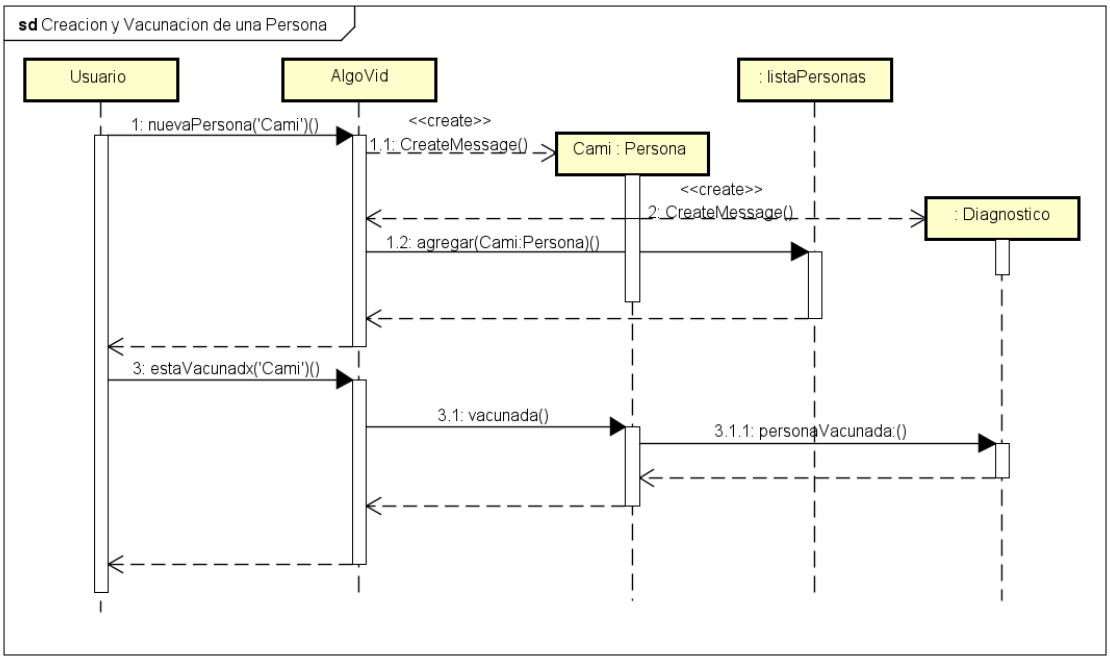

Figura 4: Creación y Vacunación de una Persona.

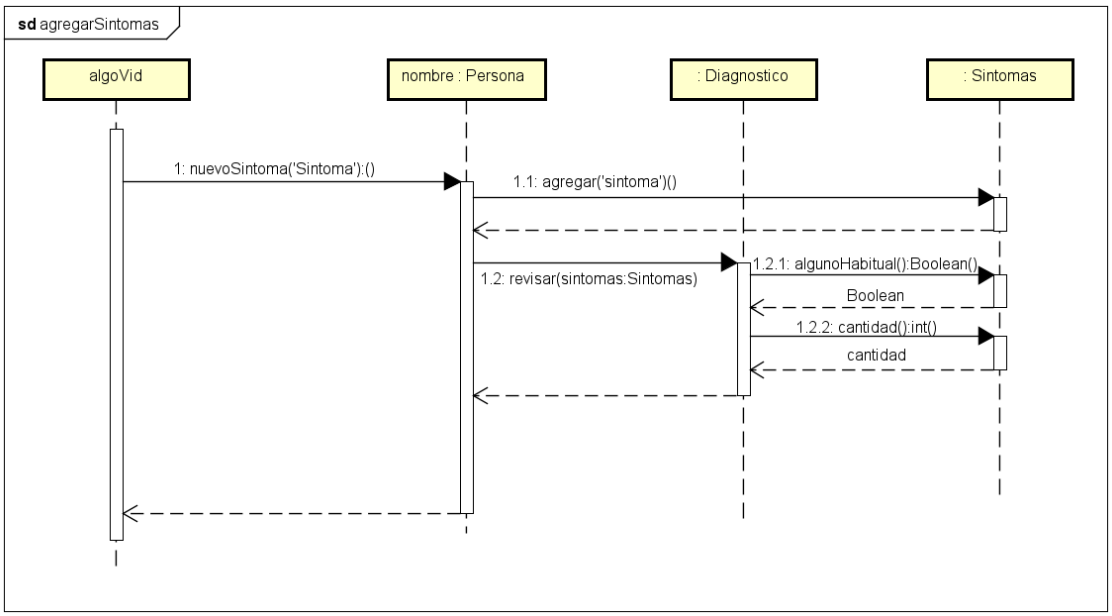

Figura 5: Agregar Síntoma a Persona.

En este diagrama se evidencia como el diagnóstico de cada persona va cambiando a medida de que agregamos algún Síntoma ya que el diagnóstico depende de la cantidad de síntomas o de si hay alguno habitual(Tos, Fiebre, Cansancio). Diagnóstico evalúa estas dos variables y modifica su estado.

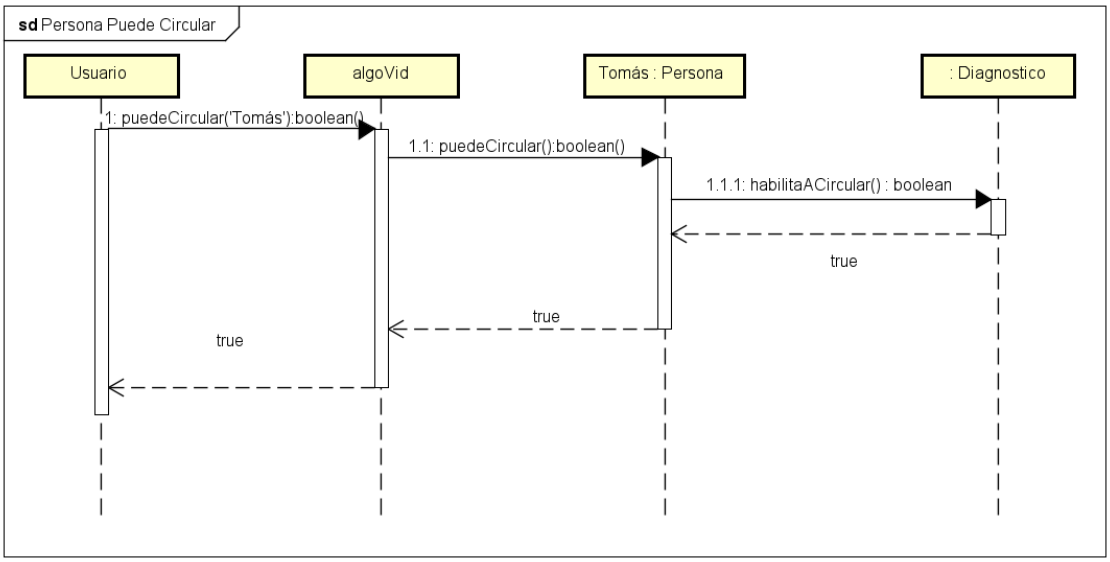

Figura 6: Persona puede Circular(Caso True).

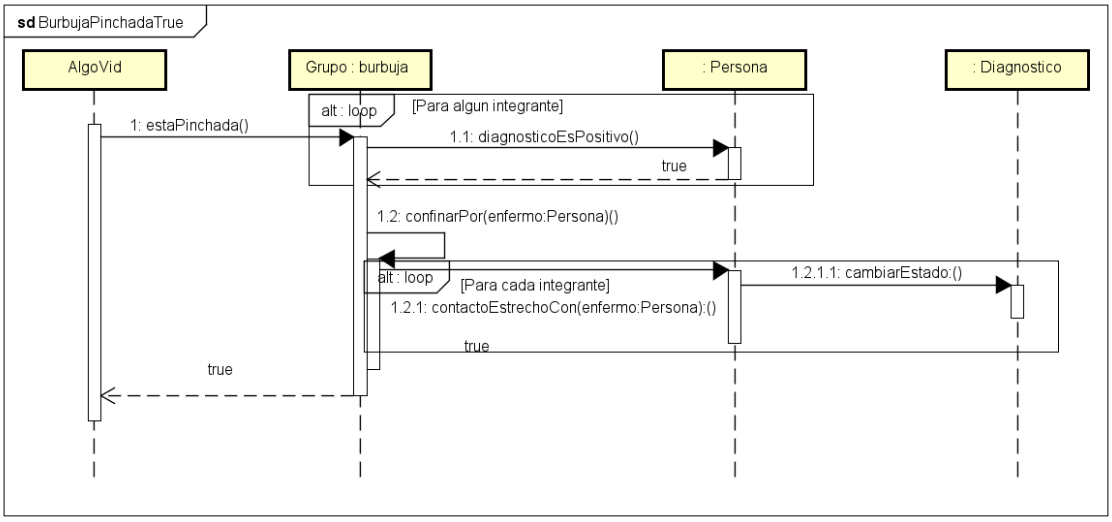

Figura 7: Burbuja Pinchada(Caso True).

En este diagrama podemos observar el comportamiento de bubuja para el método estaPinchada. Primero hace en loop con todos sus integrantes viendo si hay por lo menos uno Positivo y en caso de ser True, hace otro loop en el que cambia el diagnóstico de estos ya que tienen contacto estrecho con un caso positivo y termina devolviendo el boolean del primer loop como respuesta.