

Trabajo Práctico 1 — Smalltalk

[7507/9502] Algoritmos y Programación II
Primer cuatrimestre de 2021

Alumno:	Marchesini, Sofia
Ppadrón:	105994
Email:	smarchesini@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
3. Modelo de dominio	2
4. Diagramas de clase	3
5. Detalles de implementación	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 general de casos de Covid utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso

2. Supuestos

Para el desarrollo del modelo asumí las siguientes faltas a partir de que no hubo especificaciones sobre ellas

- **Una persona esencial puede circular en una burbuja pinchada:** una persona esencial no puede circular pero una persona confirmada o sospechosa y esencial no puede circular, entonces si una persona esencial esta en una burbuja que se pincha entonces esta no puede circular.

- **El contacto estrecho entre dos personas es un momento :** el contacto estrecho entre dos personas se puede considerar como un instante de encuentro. Entonces si una persona da covid positivo antes del encuentro, una vez que haya contacto estrecho con una persona esa otra persona no puede circular. Si hay contacto estrecho entre dos personas, y luego del encuentro una de esas dos personas da covid positivo, la otra persona puede circular tranquilamente.

- **Una persona no tiene mas de una burbuja :** Como se sabe una burbuja esta compuesta por personas , yo deduje que una persona no podía estar en mas de una burbuja a la vez por las pruebas que se nos otorgaron.

- **Una persona no puede volver a estar sana :** una persona no puede cambiar su diagnostico una vez que ya fue asignado ya que las pruebas de la catedras no otorgan mensajes como sacar síntomas, o cambiar diagnostico, ya que el modelo representa un momento.

- **Un colegio no puede tener clases no presenciales y luego presenciales :** nuevamente, ya que no se sigue la historia de un objeto en si en el tiempo , se ve algo que no se repite una cantidad de veces.

- **Una Burbuja no puede cambiar su estado :** nuevamente, una burbuja no tiene la capacidad de estar pinchada y luego no.

- **Una Burbuja no existe sin Personas :** no tendría sentido que la burbuja existiera sin personas que la compongan porque el funcionamiento de una burbuja se basa en personas, o sea esta pinchada o no esta pinchada dependiendo del estado de la persona.

- **Un colegio no existe sin Burbujas:** no tendria sentido que el colegio existiera sin burbujas ya que nuevamente el funcionamiento de un colegio depende del estado de las burbujas.

- **Algovid no existe sin Personas, Burbujas Y colegios :** Algovid por consigna no debe tener ni lógica ni comportamiento, entonces delega todo su comportamiento a estos tres objetos que hacen todo el trabajo, por consecuencia sin estos objetos Algovid no serviría para nada.

3. Modelo de dominio

El diseño de mi Trabajo Practico consiste en una clase Algovid que delega a otros objetos su comportamiento ya que no puede hacer nada a través de ella misma. Esto es con el único objetivo de crear un sistema que analice la situación de las personas en pandemia por covid.

Algovid guarda tres colecciones, una de objetos Burbuja, una de objetos Colegio y una de objetos Persona , ya que los nombres de lo que define la palabra son transferidos inicialmente a Algovid, y es conveniente guardarlos alli para facilitar el manejo del sistema a la hora de delegar los métodos.

Algovid delega sus métodos a tres principales clases, Persona, Burbuja y Colegio, las cuales se ayudan de clases abstractas para facilitar realización de sus metodos. A través de Persona se maneja todo el Diagnostico de la persona y sus diferentes estados, como por ejemplo si es una persona de riesgo o si es personal esencial. Burbuja maneja el estado de las burbujas y devuelve si esta esta pinchada o no. Y finalmente la clase Colegio se encarga de averiguar si las clases son presenciales o no, entre otras cosas. Todas estas clases le devuelven a Algovid lo que los mensajes iniciales piden.

Las ventajas de haberlo implementado asi, es que aporta mucha claridad, ya que hay un objeto para cada cosa especifica y no hace cosas de mas o que no estan definidas por su nombre. Por otro lado nunca se rompe el encapsulamiento. Y no hay metodos que ocupen mas de dos lineas de codigo ya que para casi todo se uso polimorfismo y se intento repetir lo menos posible de codigo.

4. Diagramas de clase

Dada la complejidad del modelo decidí presentar más de un diagrama. Se muestra un diagrama de Algovid y las clases mas importantes a quienes delega sus mensajes. Y luego un diagrama para cada una de estas tres clases.

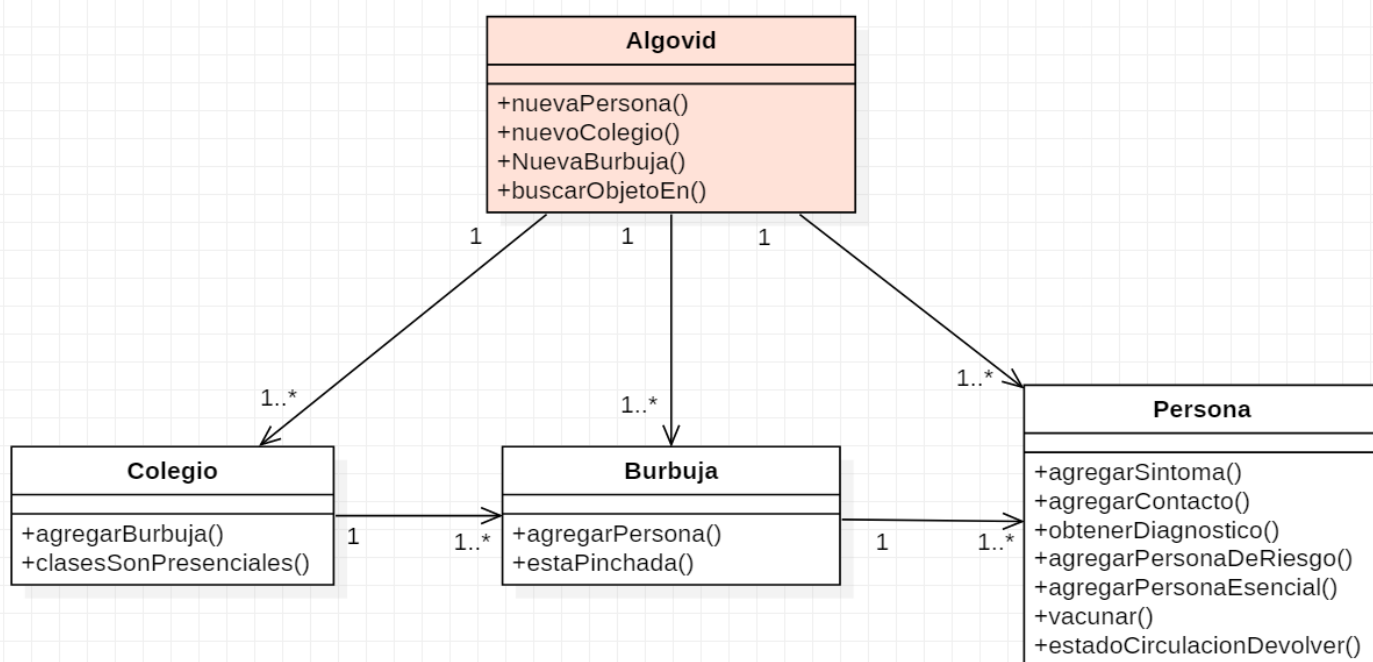


Figura 1: Diagrama de Algovid.

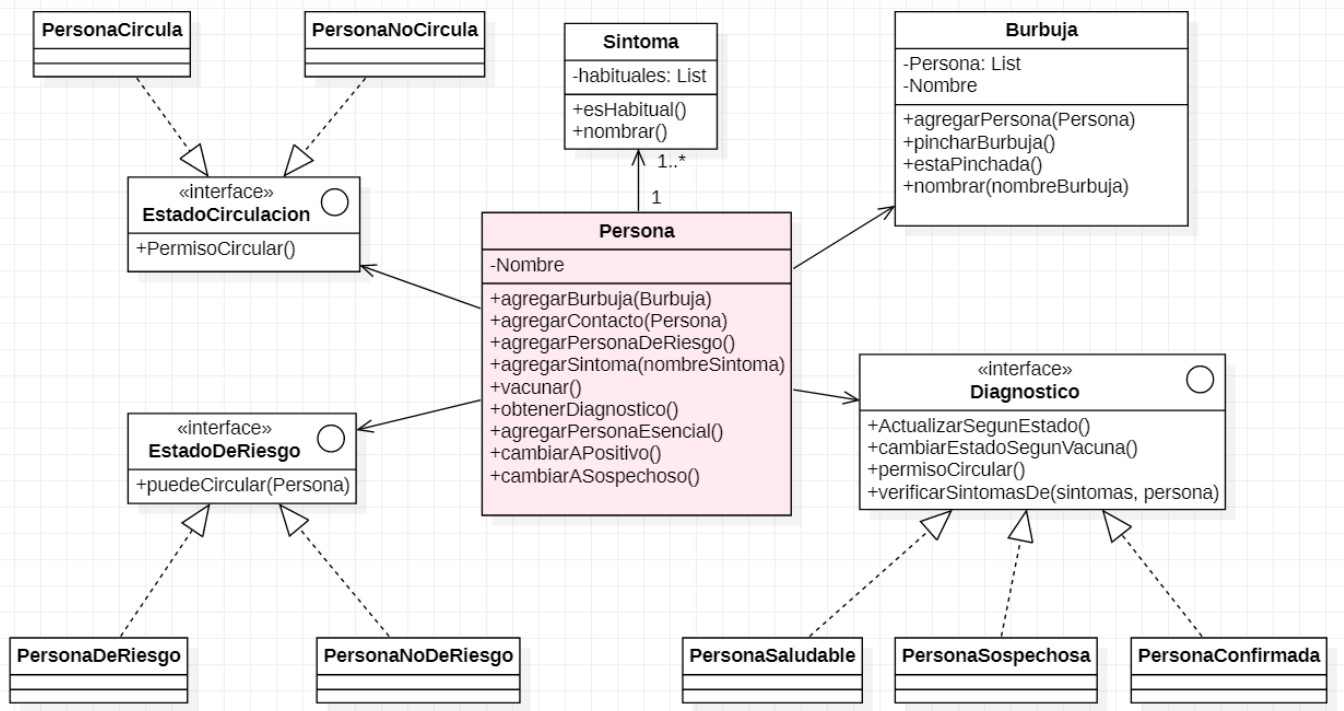


Figura 2: Diagrama de Persona.

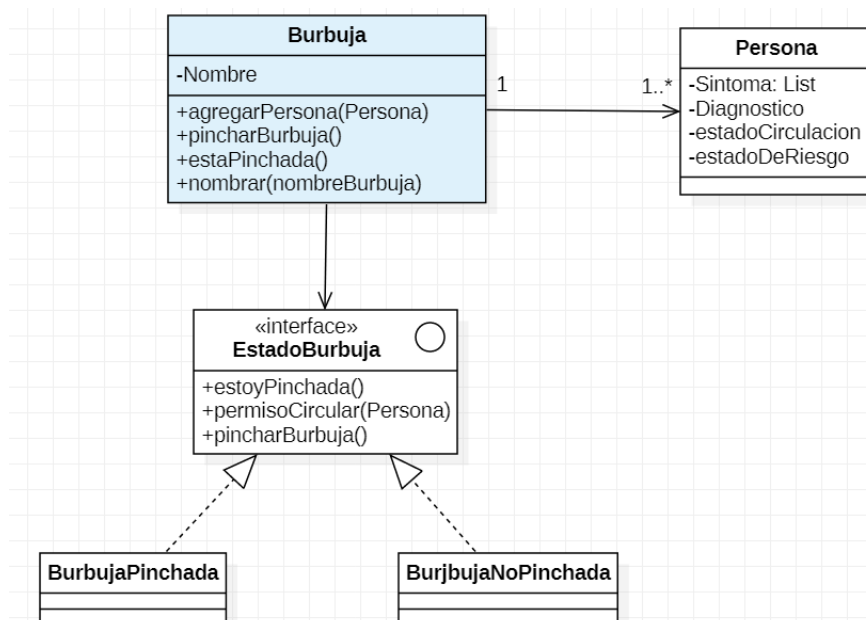


Figura 3: Diagrama de Burbuja.

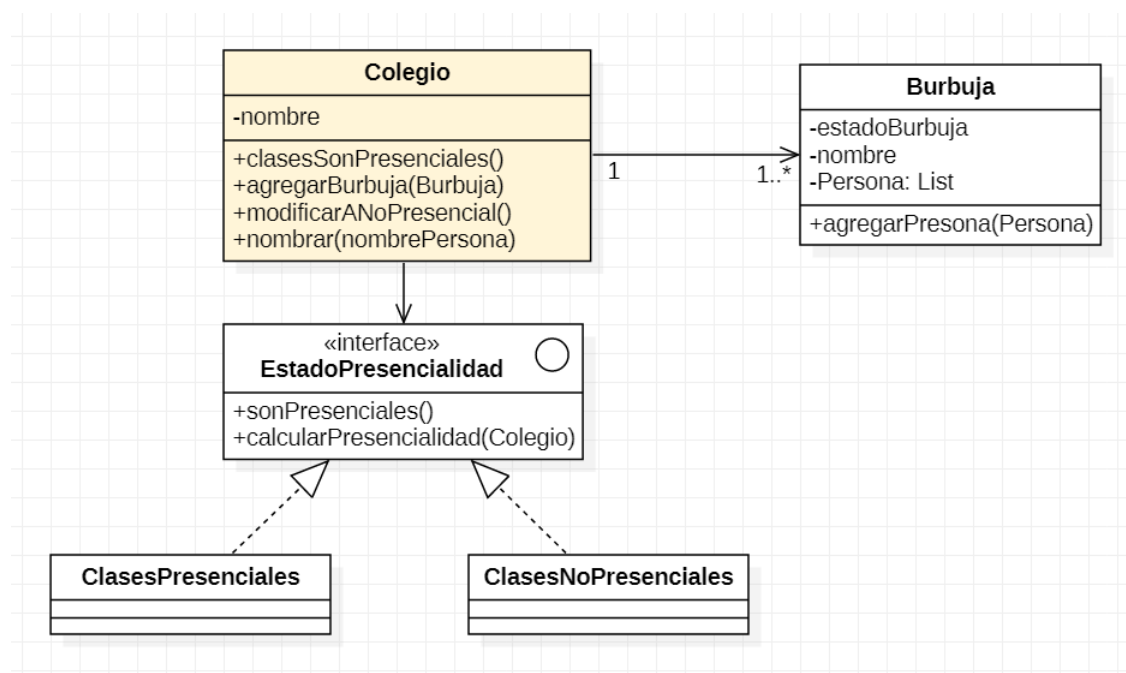


Figura 4: Diagrama de Colegio.

5. Detalles de implementación

Paradigma

El paradigma del trabajo es el paradigma orientado a objetos. Esto de por si da muchas ventajas, como la reusabilidad de codigo. La claridad, es muy sencillo leer y comprender lo que se quiere transmitir. Para mi tambien es facil la modificacion de las cosas y la refactorizacion continua se hace sencilla por esto. Por ultimo se puede considerar que es sencillo testear tambien , pues se pueden identificar los problemas de base pudiendo testear todo independientemente.

Intente que el trabajo sea lo mas claro posible dividiendo los metodos a clases especificas logrando asi tener la mayor cantidad de objetos posibles, pero sin que haya objetos de mas que no tengan comportamiento.

Use donde fue posible polimorfismo , que en el paradigma de objetos se define como la capacidad de objeto de poder tratar indistintamente a otros que sean potencialmente distintos. O sea distintos objetos entienden y responden a mismos mensajes de distinta forma. Para esto use Herencia, use distintas clases abstractas que entienden una cantidad reducida de mensajes para que algunas clases hereden de ella y se pueda realizar ordenadamente el polimorfismo.

Delegacion Y Herencia

- Algoritmo delega los metodos que un usuario le da a otros objetos , ya que este no puede tener comportamiento ni logica, como previamente explique en Modelo de Dominio.
- Persona es inicializada con diferentes estados compuestos por la informacion de la persona. Guarda su diagnostico, su estado de riesgo, su Burbuja, si es que pertenece a una, sus sintomas y su estado de circulacion. Como ya dije utiliza varias clases abstractas. Entre ellas : EstadoDeRiesgo, EstadoCirculacion, Y Diagnostico . En los objetos que heredan de ellas se encuentran las diferentes posibilidades que puede definir a la Persona. Esto da la ventaja de que aporta mucha claridad al

lector. Parecio necesario hacer estas clases porque Persona no podia encargarse de todo.

- o Clase Abstracta : EstadoDeRiesgo

- Clases que hereda : PersonaDeRiesgo y PersonaNoDeRiesgo

- o Clase Abstracta : Estado Circulacion

- Clases que hereda : PersonaCircula y PersonaNoCircula

- o Clase Abstrata : Diagnostico

- Clases que hereda : Persona Saludable, PersonaSospechosa y PersonaConfirmada

- Burbuja es una clase que es inicializada con una coleccion de personas, las que la componen, y por otra parte se inicializa como una burbuja no pinchada hasta que alguna de las personas en la burbuja sea positiva. El estado de la burbuja es una clase abstracta, que puede ser o pinchada o no pinchada. De esta manera se puede hacer polimorfismo facilmente, y todos los mensajes de burbuja quedan practicamente de una linea. Esta Burbuja ademas se encarga de enviar el mensaje de actualizar el estado de las personas de la burbuja si esta se pincha.

- o Clase Abstracta : EstadoBurbuja

- Clases que hereda : BurbujaNoPinchada y BurbujaPinchada

- Colegio es inicializada con la burbujas que las componen y se encarga de transmitirle a Algovid si las clases son presenciales o no lo son. Para esto colegio usa una clase abstracta llamada EstadoPresencialidad que en principio esta inicializada con clases presenciales, uno de los objetos que heredan de ella. De nuevo use esta clase abstracta porque vi necesario su uso con tantos estados que almacenar.

- o Clase Abstracta : EstadoPresencialidad

- Clases que herada : ClasesPresenciales y ClasesNoPresenciales

- Sintoma es una clase que es utilizada principal y unicamente para devolver si un sintoma es habitual o no lo es. No usa otras clases, si no que Persona al agregar sus sintomas crea un objeto sintoma asignandole un nombre y para ver el diagnostico de una persona se fija a traves de metodos de Sintoma si el sintoma en el que estoy parado es habitual o no.

6. Excepciones

1. **NoExisteElNombreError** : Este error se lanza cuando queremos utilizar metodos que requieren que se haya almacenado una Persona, un Colegio o una Burbuja previamente y no fueron creados.

2. **NombresRepetidosError** : Este error lanza cuando queremos guardar dos nombres de Burbuja , de Persona y de Colegio que son iguales. Este error es porque basicamente no se podria diferenciar solo por un string de nombre que son distintas personas.

3. **ColeccionVacuaError** : Este error solo se lanza cuando intentamos recorrer una coleccion que esta vacia, por ejemplo cuando intentamos utilizar los mensajes de colegio sin que se haya guardado una burbuja dentro.

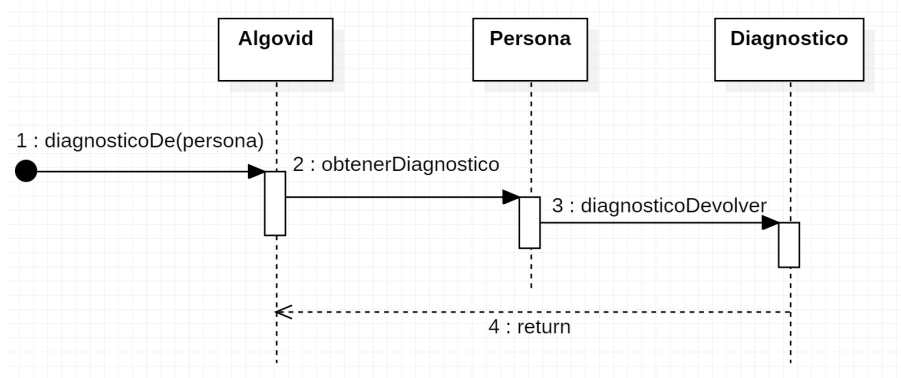
4. **NombreVacio** : Este error se lanza cuando queremos asignarle nombre a Burbuja, Persona o Colegio con un nombre vacio.

5. **ParametrosInvalidos** : Este error se lanza cuando un parametro es de un tipo no esperado, en este caso siempre se esperan de tipos String, entonces cuando por ejemplo se quiere asignar un nombre de tipo entero, se lanza.

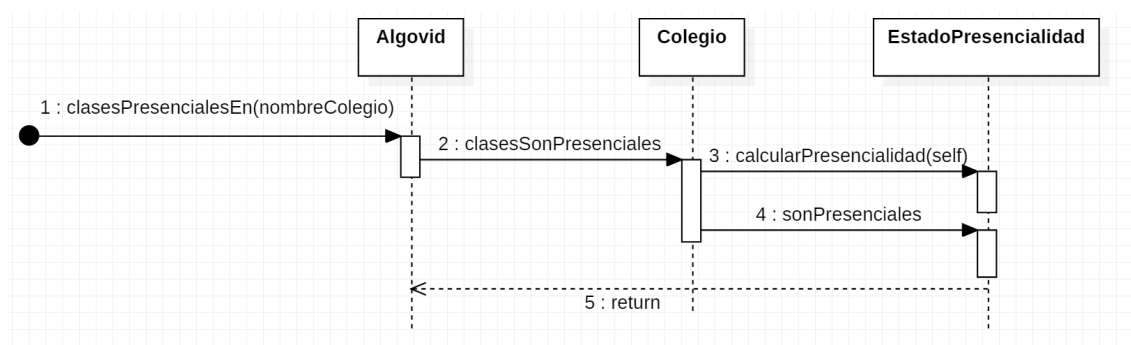
7. Diagramas de secuencia

Diagramas de Metodos basicos y generales de la implementacion

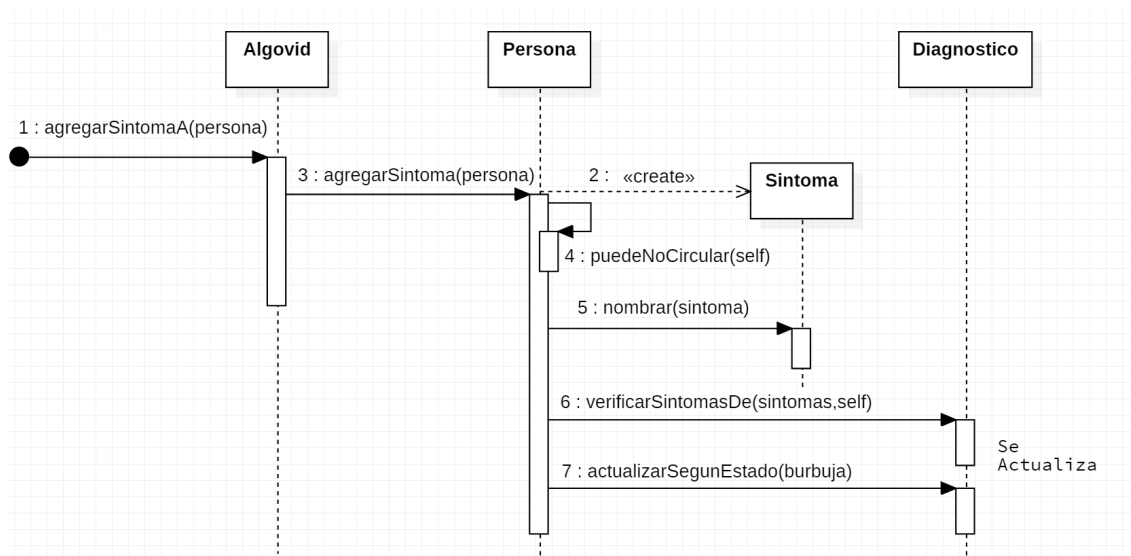
Metodo DiagnosticoDe: dependiendo del diagnostico se devolvera el nombre del diagnostico que sea.



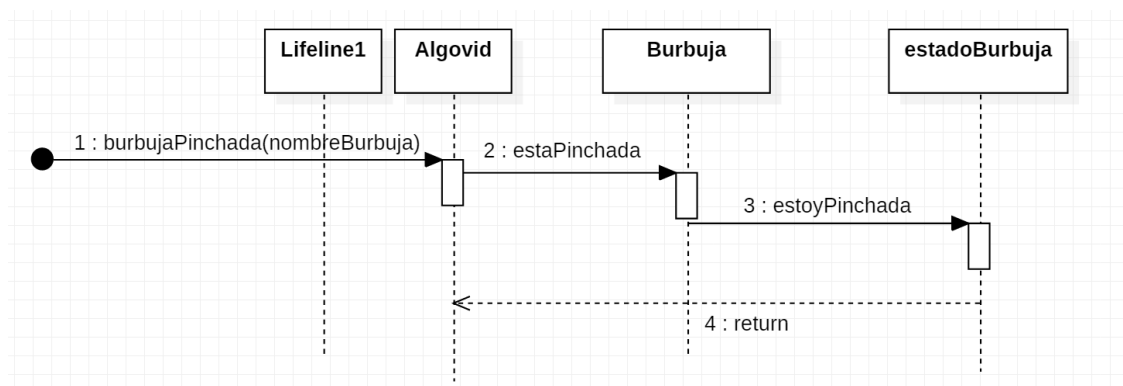
Metodo ClasesPresencialesEn: dependiendo el estado de presencialidad se devolvera el correspondiente.



Metodo AgregarSintomaA: dependiendo el diagnostico de la persona pasaran distintas cosas.



Metodo burbujaPinchada: dependiendo el estado de la burbuja se devolvera lo correspondiente.



Diagramas de Pruebas

Diagrama de la prueba : una persona confirmada entonces la burbuja esta pinchada desde la clase Burbuja

Caso donde previamente fue agregada a la burbuja una persona que es sospechosa entonces todavia la burbuja no esta pinchada. Agrego un cuarto sintoma y la persona se convierte en positiva, y se pincha la burbuja.

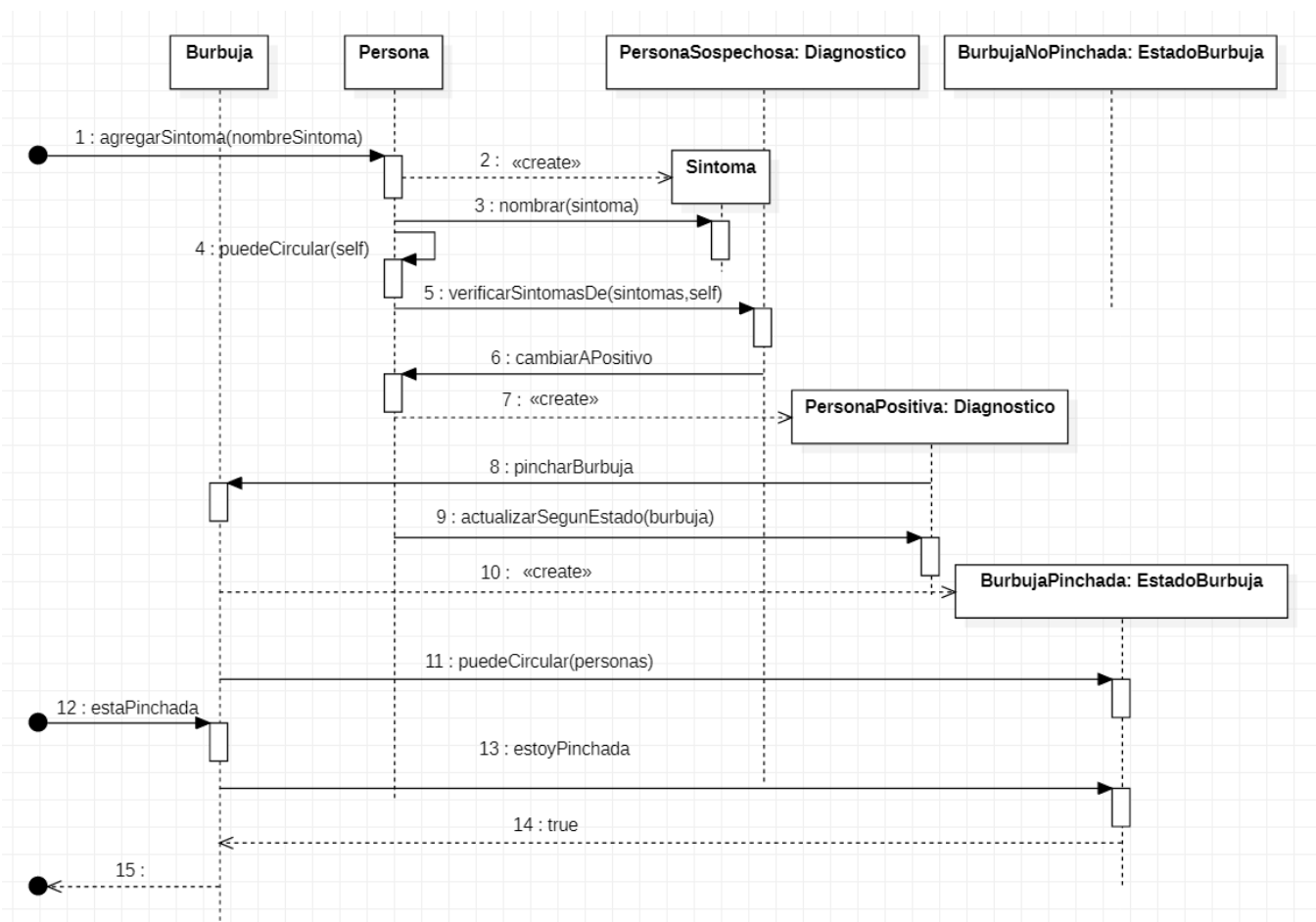


Diagrama de la prueba : una persona sospechosa y vacunada no puede circular

Caso donde una persona ya fue agregada a Algoovid previamente y es saludable, no de riesgo y no puede circular por defecto. La vacuno y le agrego un sintoma entonces es sospechosa , entonces no puede circular.

