

Trabajo Práctico 1 — Smalltalk

[7507/9502] Algoritmos y Programación III

Curso 2

Primer cuatrimestre de 2021

Alumno:	RIZZO EHRENBLOCK, Gonzalo Daniel
Número de padrón:	106475
Email:	grizzoe@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
3. Diagramas de clase	2
4. Detalles de implementación	5
4.1. Clase AlgoVid	5
4.2. Clase Persona	6
4.3. Clase EstadoSalud	6
4.4. Paradigmas	6
4.5. Delegacion	6
5. Excepciones	6
6. Diagramas de secuencia	6

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 un sistema de detección y registro de Coronavirus (COVID-19) utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

2. Supuestos

1. No pueden haber personas, colegios ni burbujas con el mismo nombre. Para evitar problemas si el sistema es usado con una cantidad de datos muy grande (propensa a repeticiones) lo que se puede hacer, por ejemplo, es agregar el dni seguido del nombre al registrar una nueva persona.
2. Supuse que todas las personas harán lo esperado en un contexto de pandemia. Es decir, al poseer al menos un sintoma no circulan. Esto no quita que una persona "sana" no pueda tener contacto estrecho con un positivo (pueden vivir en la misma casa por ejemplo).
3. Se detecta que una persona es positiva basandose en su cantidad de sintomas y si estos son habituales o no. Por lo tanto, no se contempla la posibilidad de que una persona sea asintomatica y positiva

3. Diagramas de clase

Estos diagramas muestran las clases utilizadas y las relaciones entre ellas. Se omiten algunas para no complicar la lectura de los diagramas.

En el primero se muestran las cuatro principales clases del trabajo.

En el segundo se puede ver la relacion de la clase persona con su salud, el hecho de que pueda presentar sintomas o ser positivo de Coronavirus y la conexion de esto con su circulacion.

En el ultimo diagrama se pueden ver las relaciones entre la clase Algovid y sus excepciones.

aradigm Online Free Edition

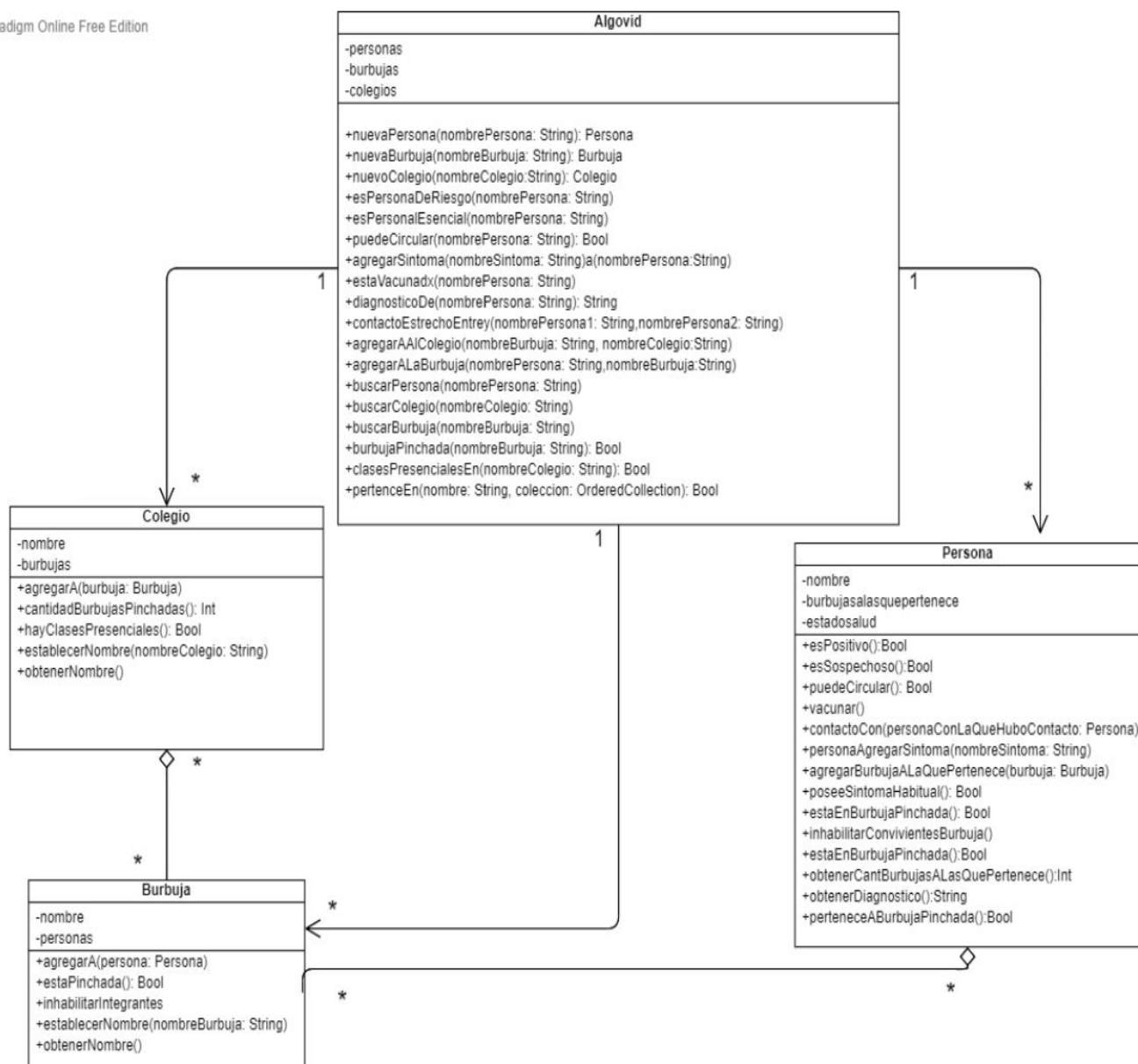


Figura 1: Diagrama 1.

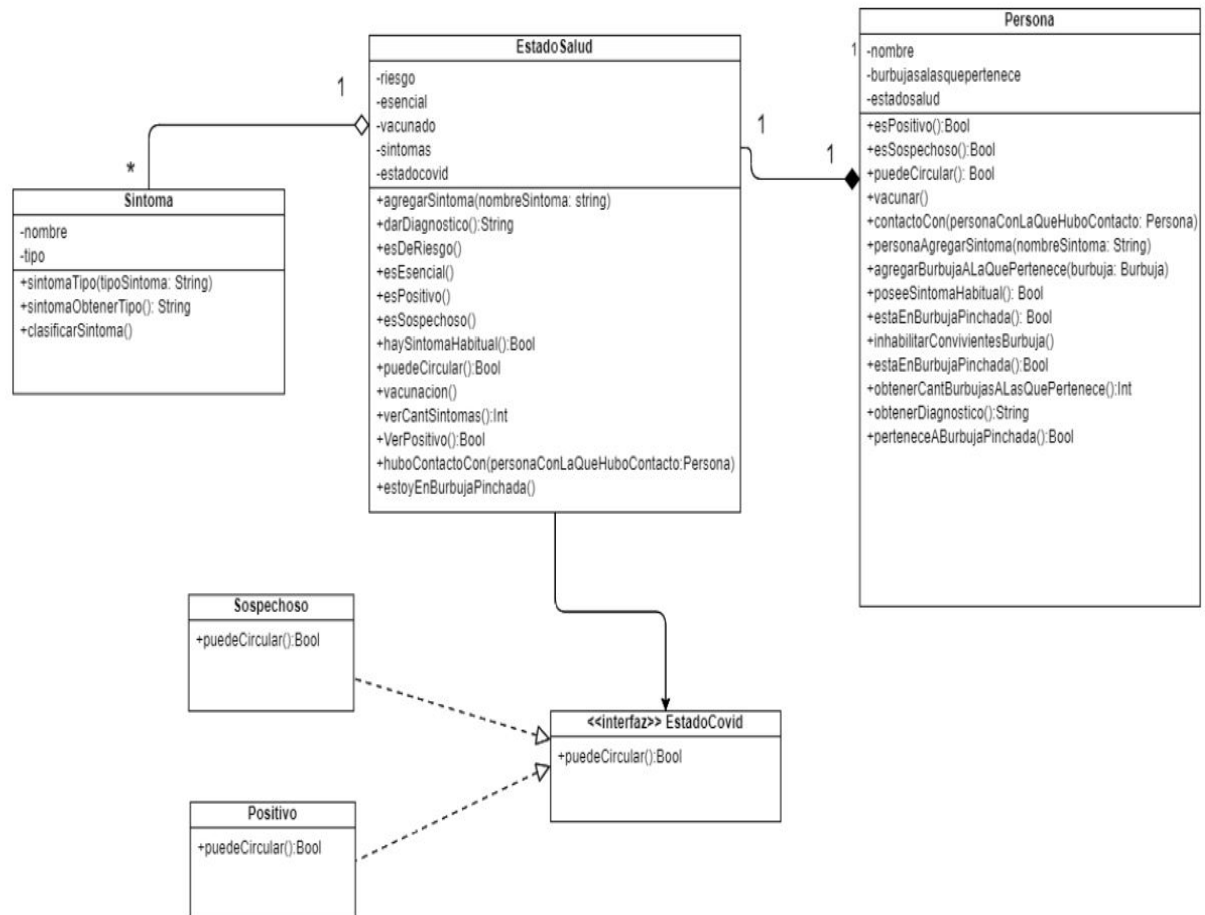


Figura 2: Diagrama 2.

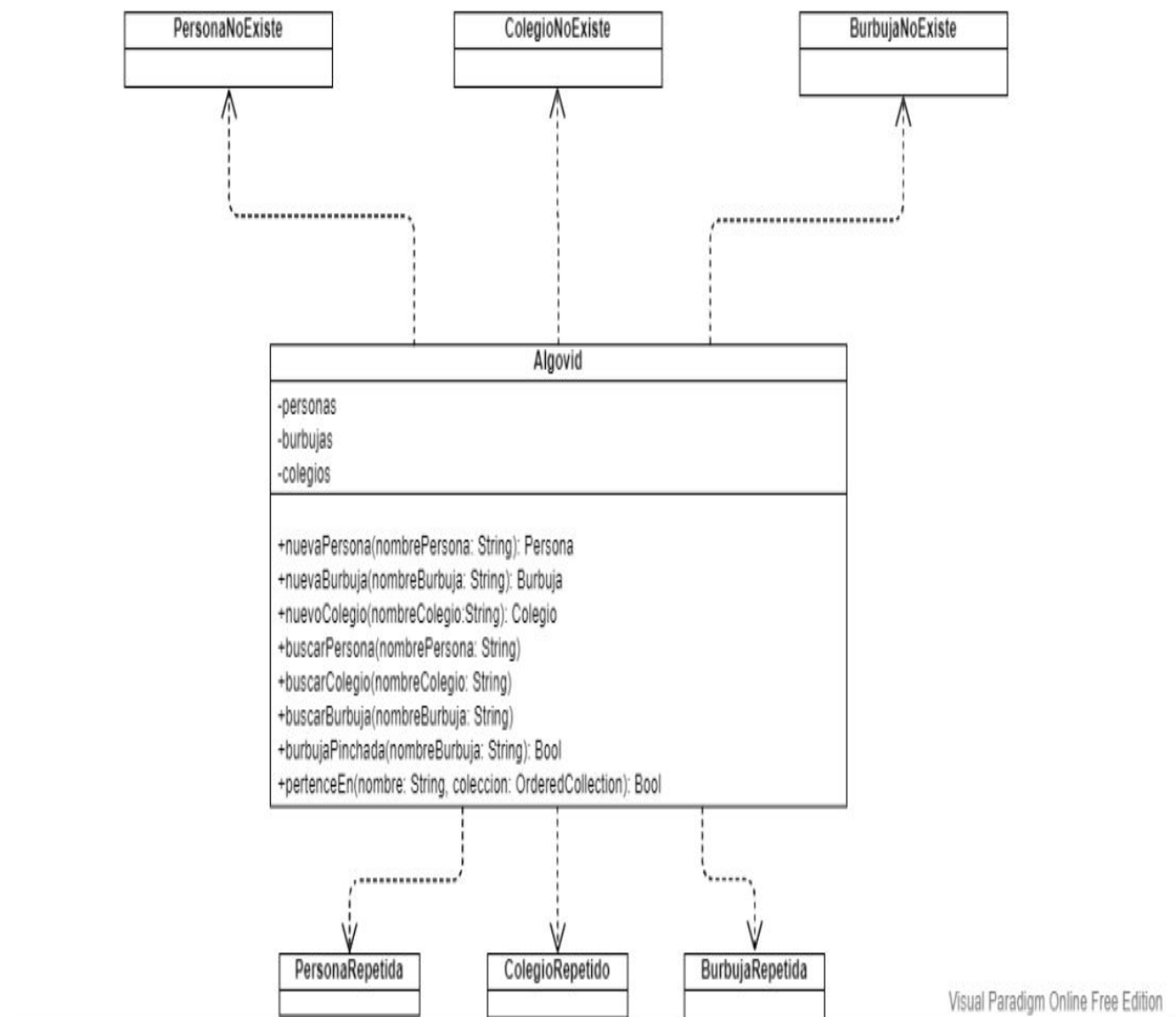


Figura 3: Diagrama 3.

4. Detalles de implementación

4.1. Clase AlgoVid

Tiene una `orderedCollection` de personas, otra de colegios y otra de burbujas. Delega todas las responsabilidades a las clases `Persona`, `Colegio` y `Burbuja`, no tiene más lógica que buscar dentro de las colecciones y delegar. Con esta clase podemos ver los conceptos de delegación, abstracción y `tell don't ask` ya que `AlgoVid` solo da instrucciones para que se ejecuten los métodos de las demás clases pero no ve cómo estas están implementadas (tampoco debería importarle).

4.2. Clase Persona

Para respetar las ideas de poliformismo, decidí que los atributos de esta clase no guarde booleanos en ningún caso.

Decidí hacerlo de esta manera por que si el proyecto escala, sería bueno tener objetos para los diferentes casos y no solo booleanos que limitan las posibilidades de polimorfismo.

Contemplé la idea de solo tener 2 tipos de persona, personaHabilitada y personaNoHabilitada, con un método "motivo", pero luego me di cuenta que esta solución no era la más optima y me limitaba el rango de posibilidades de acción que si me da la solución actual.

Los metodos relacionados con su salud se los delega a la clase EstadoSalud.

4.3. Clase EstadoSalud

Esta clase trata todas las operaciones sobre la salud del paciente, como agregar sintomas, Covid Positivo, Sospechoso, riesgo.

El metodo puedeCircular se extiende en las clases EstadoCovid, Sospechoso y Positivo, ya que cada una (Sospechoso y Positivo)de estas entiende (y actua) de manera diferente sobre lo que significa "poder circular", pero ambas devuelven la posibilidad de circular o no. Aplicando el concepto del Patrón Strategy.

4.4. Paradigmas

Polimorfismo: Sus ventajas son que permite reutilizar código según sea necesario, usar una sola variable para almacenar múltiples tipos de datos, facilita el debugging cuando el proyecto es muy grande.

Abstracción: La ventaja es que las clases no necesitan saber el funcionamiento de las demás entonces se puede aplicar facilmente el principio tell don't ask. Presente en AlgoVid (explicado arriba).

Encapsulamiento: Protege la integridad y seguridad de cada clase.

4.5. Delegacion

Las ventajas de la delegacion es que evita que una clase se encargue de todas las operaciones, teniendo varias clases que realizan tareas especificas, al trabajar en conforman un "todo". Además al usar delegacion el codigo queda mucho mas modularizado. Se puede ver en la clase AlgoVid como delega todas las operaciones a las demas clases, permitiendo que las operaciones de AlgoVid se entiendan facilmente.

5. Excepciones

PersonaNoExiste, ColegioNoExiste y BurbujaNoExiste fueron creadas para informar que la persona/colegio/burbuja ingresada no se encuentra en el sistema, por lo que no se realiza la operacion de pretendía hacer el usuario.

PersonaRepetida, ColegioRepetido y BurbujaRepetida fueron creadas para informar que ya existe una persona/colegio/burbuja con el nombre que intenta introducir el usuario, por lo que no se crea la entidad.

6. Diagramas de secuencia

A continuación se detallan los diagramas de secuencia de las principales y más relevantes acciones. Se omiten algunas para no complicar la lectura de los diagramas.

En el primer diagrama se ejemplifica lo que sucede con una persona sospechosa, la interacción con su sintoma y circulación. El usuario crea AlgoVid y envía los mensajes para llegar a esta situación.

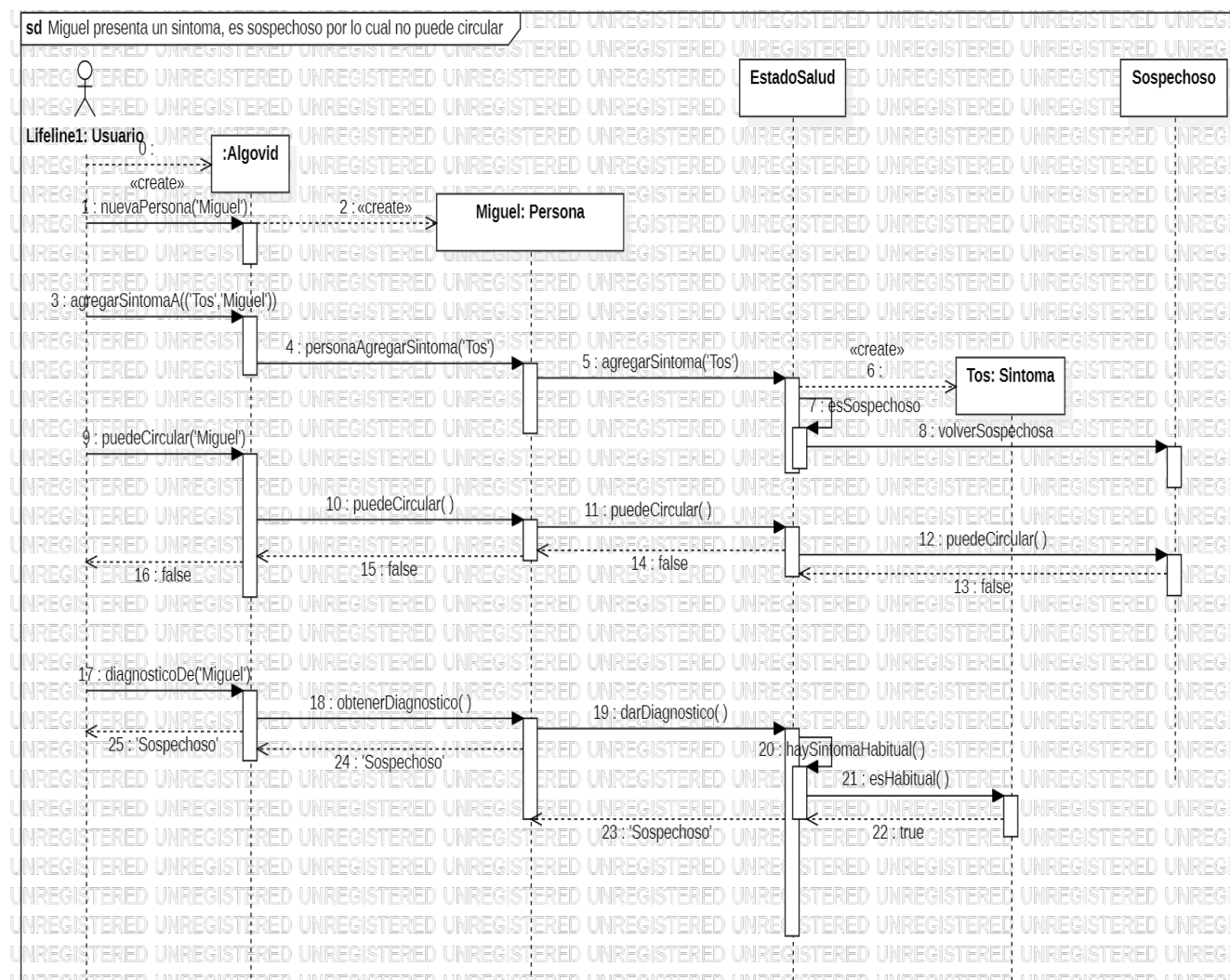


Figura 4: 1

En el segundo diagrama se ejemplifica lo que sucede cuando una persona que no es positiva entra en contacto estrecho con una persona positiva. El usuario crea AlgoVid y envía los mensajes para llegar a esta situación.

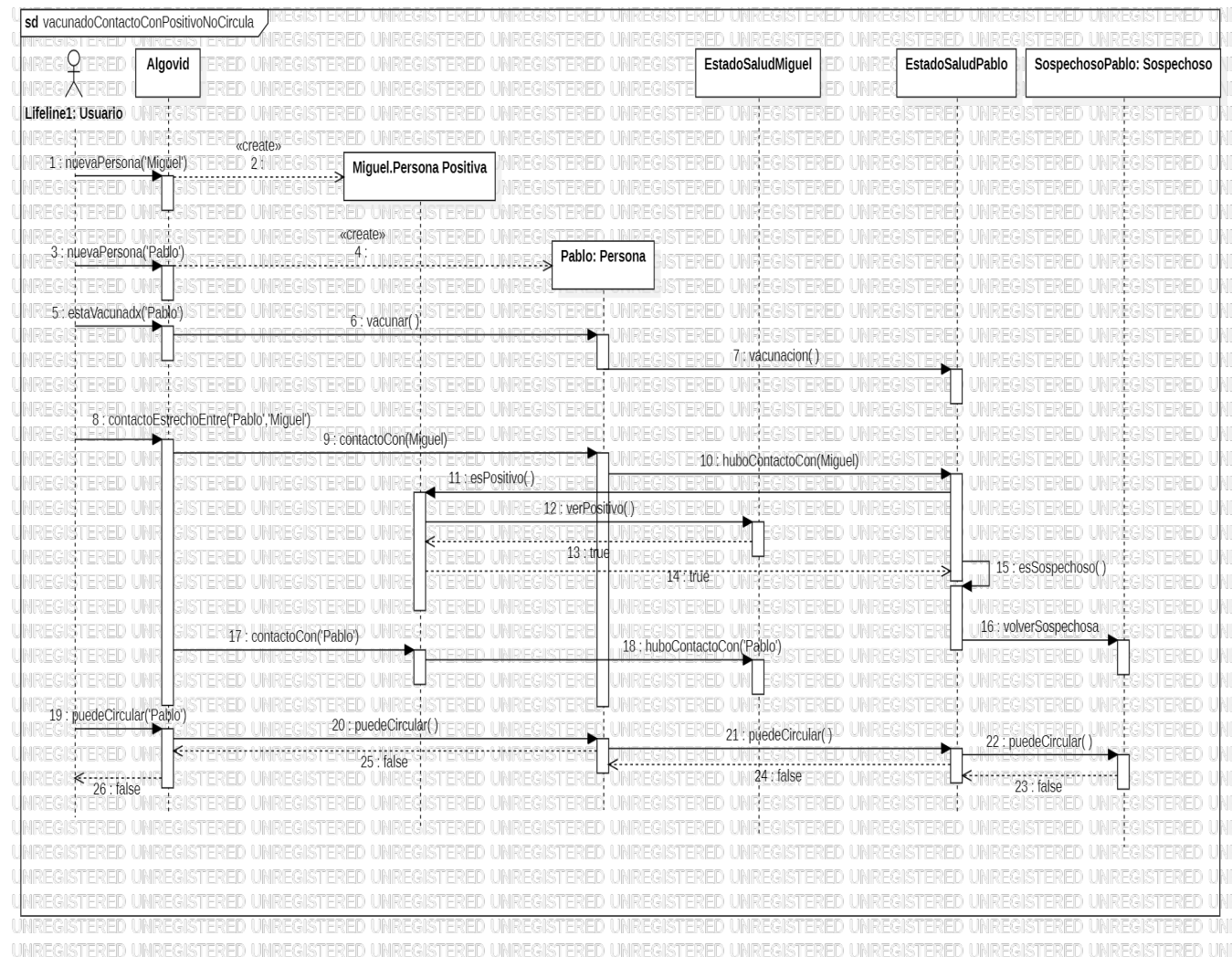


Figura 5: 2

En el tercer y último diagrama se ejemplifica lo que sucede cuando una persona positiva pincha una burbuja (en la que había una persona no positiva). El usuario crea AlgoVid y envía los mensajes para llegar a esta situación.

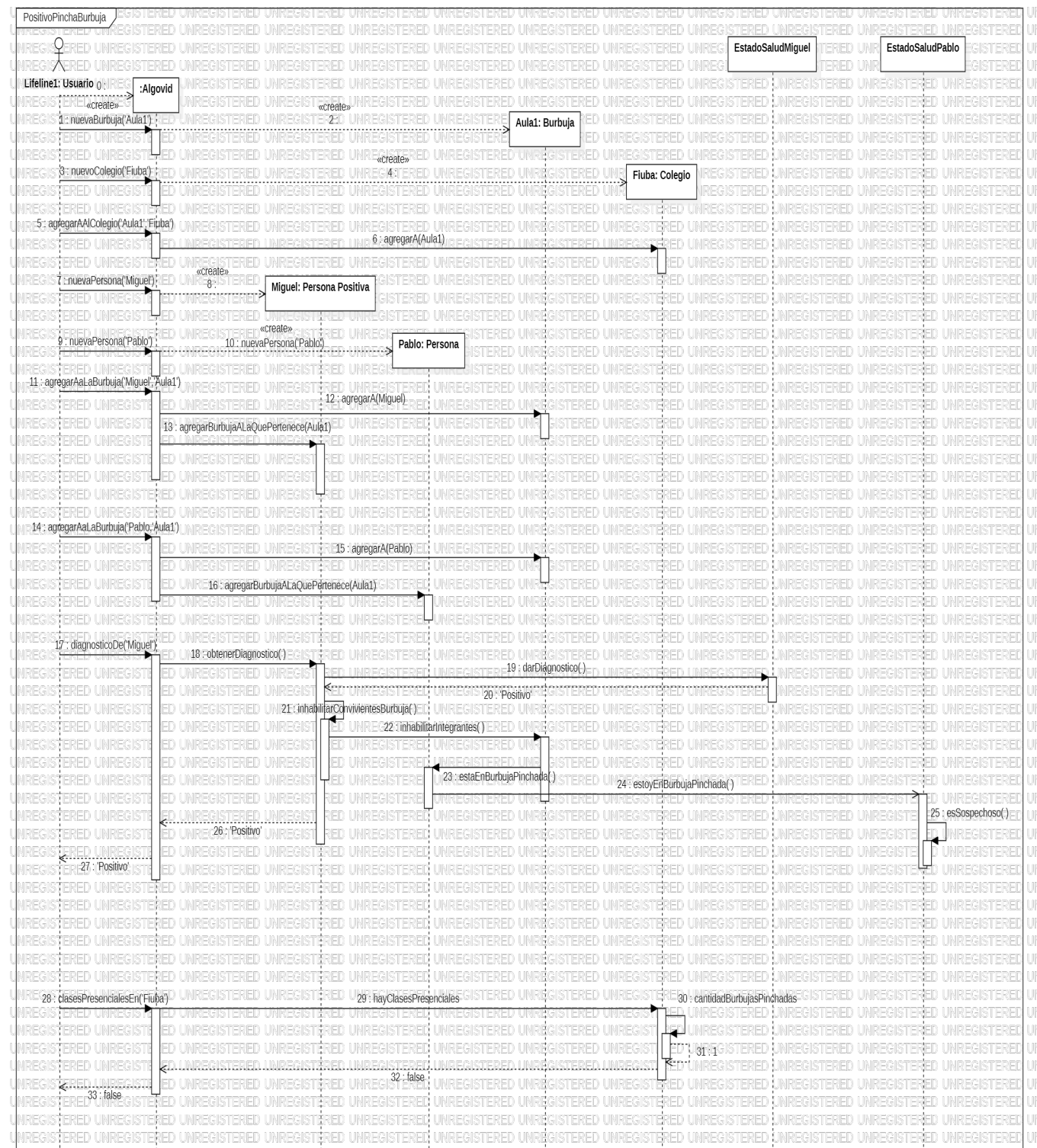


Figura 6: 3