

# Trabajo Práctico 1 — Smalltalk

[7507/9502] Algoritmos y Programación III  
Curso 1  
Primer cuatrimestre de 2021

Alumno:	BALMACEDA, Fernando
Número de padrón:	105525
Email:	fbalmaceda@fi.uba.ar

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Supuestos</b>	<b>2</b>
<b>3. Diagramas de clase</b>	<b>3</b>
<b>4. Detalles de implementación</b>	<b>7</b>
4.1. Pilares del paradigma de programación orientada a objetos . . . . .	7
4.2. Utilización de herencia y delegación . . . . .	7
<b>5. Excepciones</b>	<b>7</b>
<b>6. Diagramas de secuencia</b>	<b>8</b>

## 1. Introducción

En el siguiente informe se detallará la documentación correspondiente a la solución del primer trabajo practica de Algoritmos y Programación 3. Este trabajo consiste en el desarrollo de una aplicación donde se registran personas a las cuales se les designa un diagnostico y la posibilidad de circular según sus síntomas y/o su condición. También se pueden registrar burbujas y colegios donde se pueden agregar personas dentro. Todo esto se realiza basándose en los conceptos de la programación orientada a objetos que se han visto en el curso.

## 2. Supuestos

**Diagnostico de la persona por defecto:** Se consideró que por defecto el diagnostico de la persona sea "Negativo", y no "Sospechoso".

**Mismo nombre de personas:** Se considero que no pueden existir 2 o más personas con el mismo nombre. Debido a esto se realizó una excepción.

**Mismo nombre de burbujas:** Se consideró que no pueden existir 2 o más burbujas con el mismo nombre. Debido a esto se realizó una excepción.

**Mismo nombre de colegios:** Se consideró que no pueden existir 2 o más colegios con el mismo de nombre. Debido a esto se realizó una excepción.

**Mismo nombre de síntomas:** Se consideró que una persona no puede tener 2 o más síntomas con el mismo nombre. Debido a esto se realizó una excepción.

**Una persona en dos burbujas distintas:** Se consideró que una persona no puede estar en 2 burbujas al mismo tiempo. Esto está más explicito en el diagrama de clases.

### 3. Diagramas de clase

A continuación se muestran los diagramas de clases de las distintas clase que se usaron para la implementación de todo el trabajo.

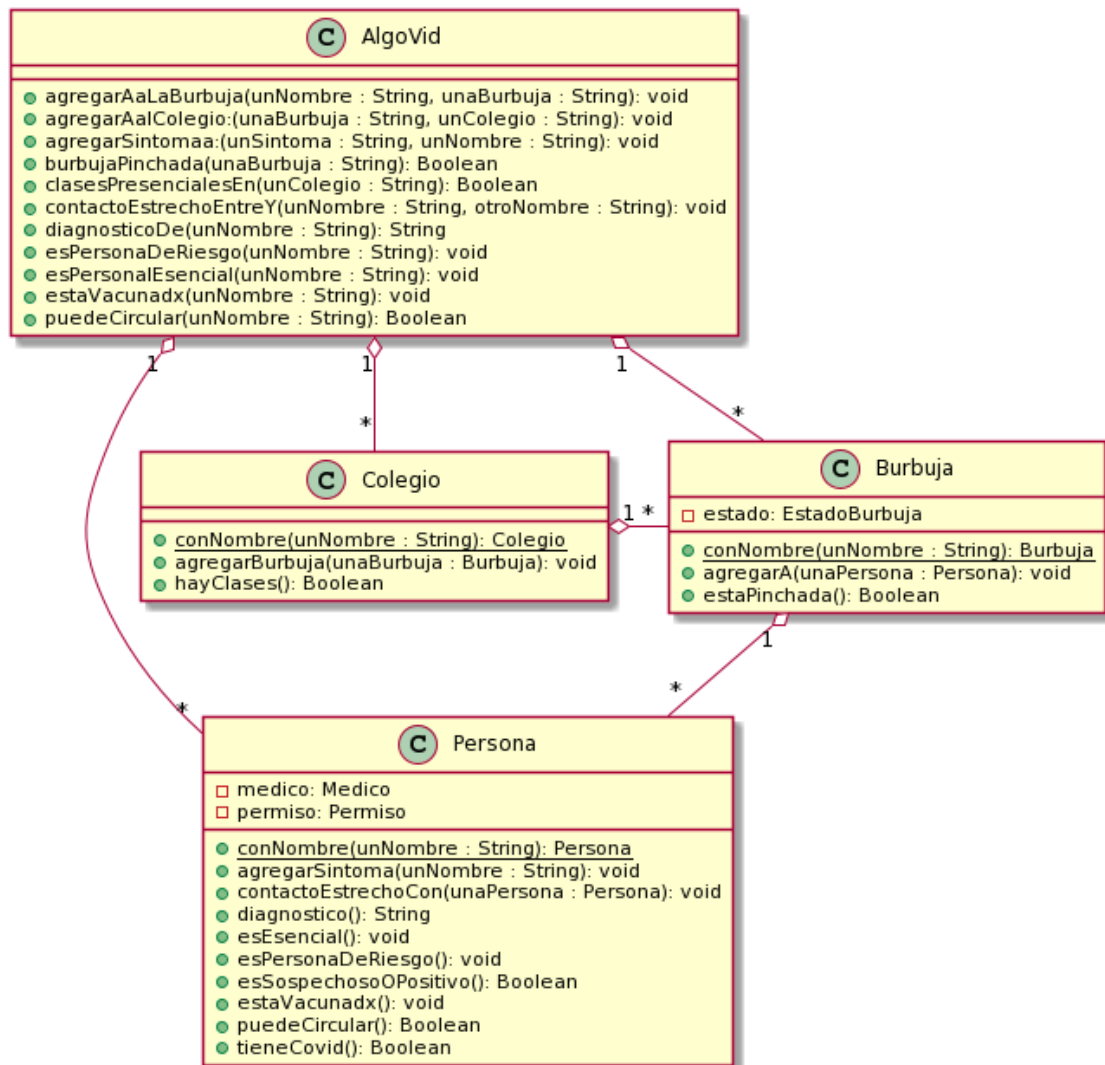


Figura 1: Diagrama de clase de AlgoVid.

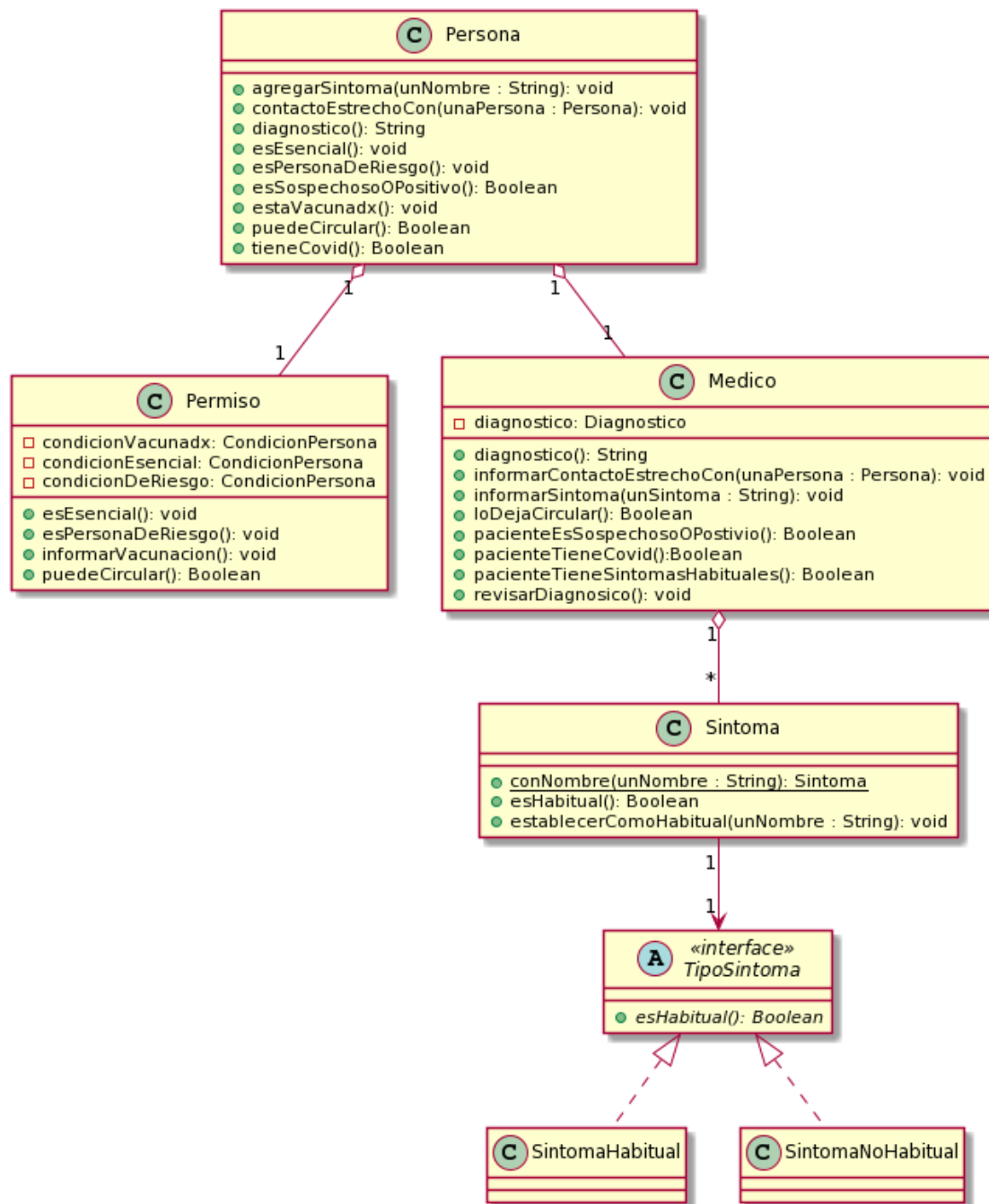


Figura 2: Diagrama de clase de Persona.

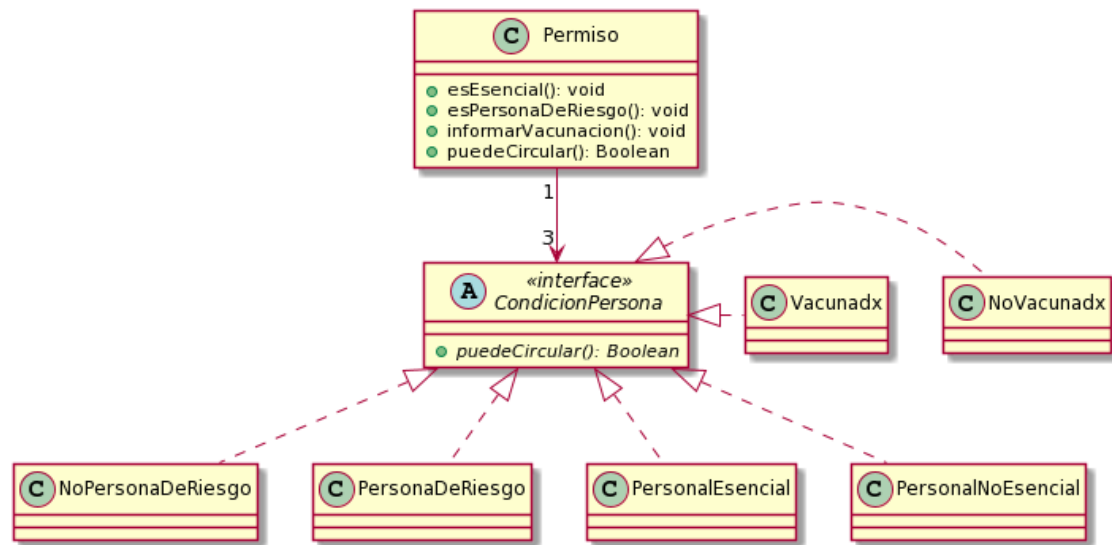


Figura 3: Diagrama de clase de Permiso.

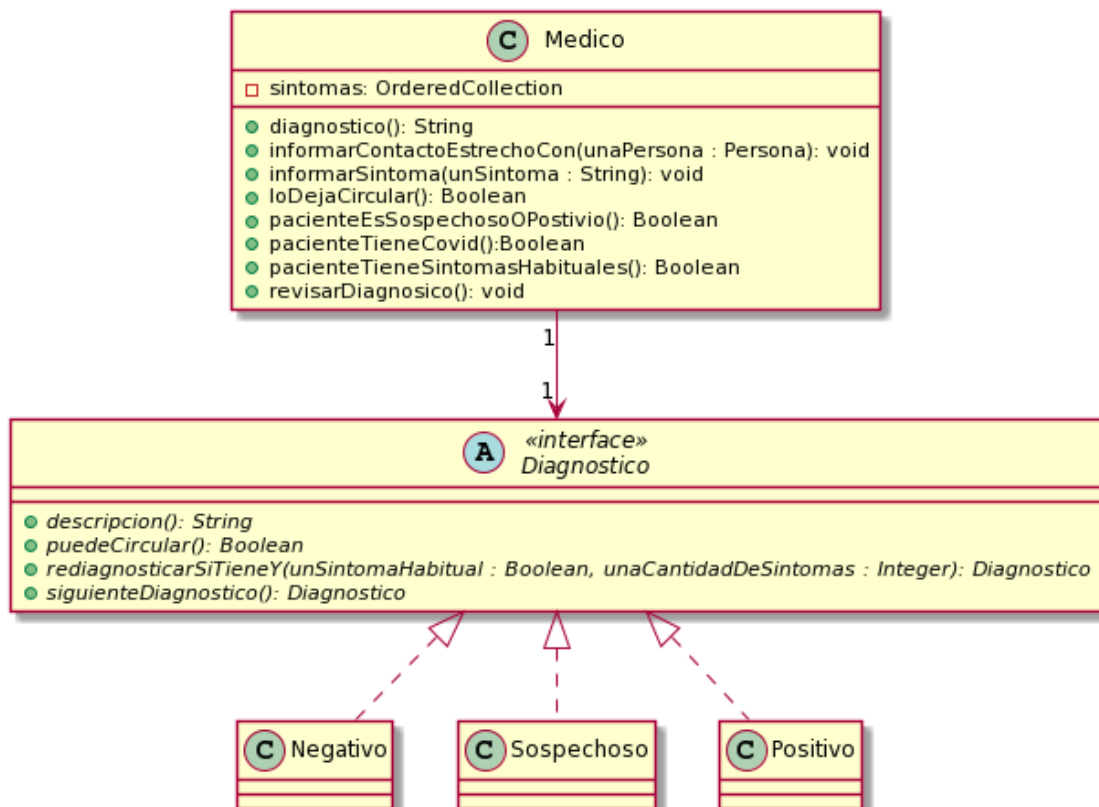


Figura 4: Diagrama de clase de Médico.

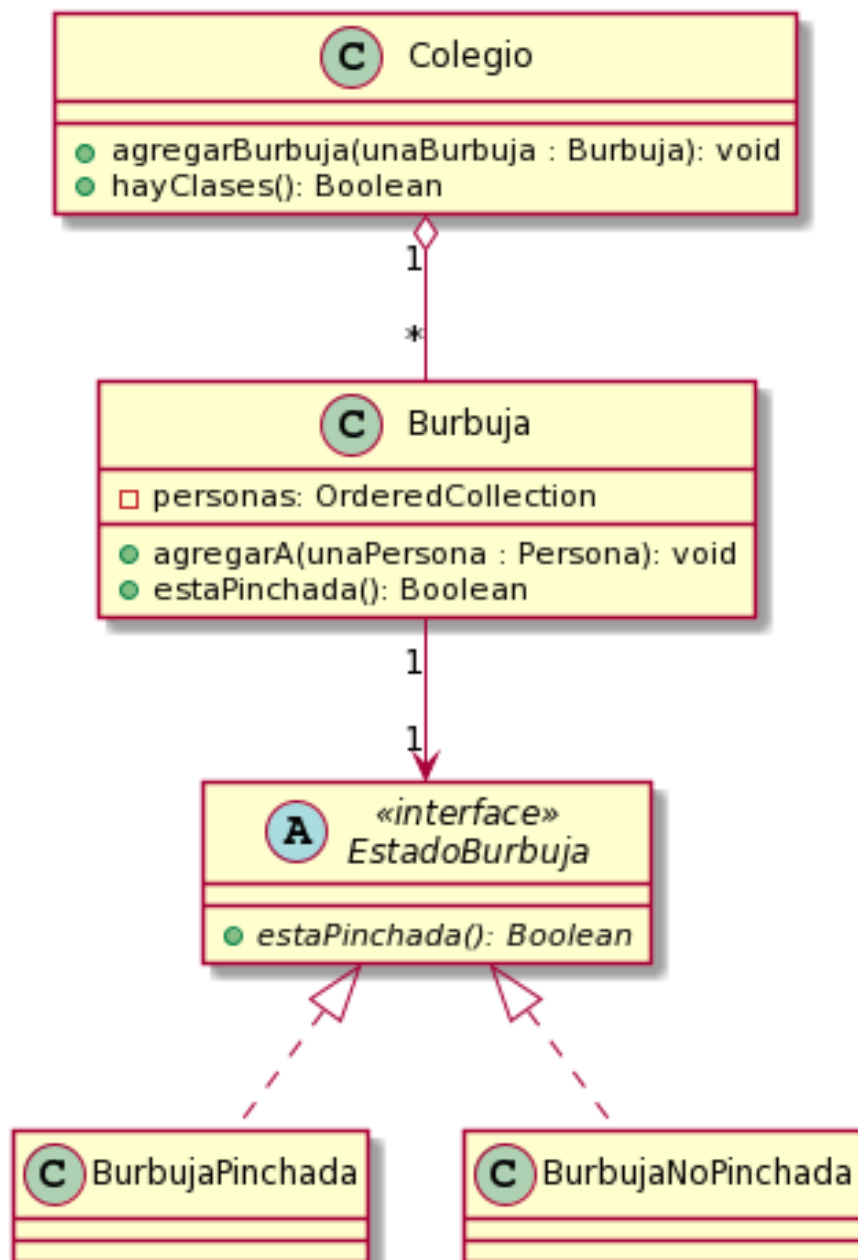


Figura 5: Diagrama de clase de Colegio.

## 4. Detalles de implementación

### 4.1. Pilares del paradigma de programación orientada a objetos

Para la realización del trabajo utilice herencia, encapsulamiento y polimorfismo.

**Ejemplo de herencia:** Un caso de herencia lo implemento por ejemplo con la interfaz TipoSintoma, la cual tiene como clases hija a SintomaHabitual y SintomaNoHabitual. En este caso las clases SintomaHabitual y SintomaNoHabitual heredan los métodos de TipoSintoma.

**Ejemplo de polimorfismo:** Un caso de polimorfismo lo implemento cuando por ejemplo le tengo que asignar un diagnostico a la persona, le delego el trabajo de deteminarlo a la clase Médico, el cual luego le consulta a las subclases de Diagnostico si permanece igual o cambia de estado.

**Ejemplo de encapsulamiento:** Un caso de encapsulamiento lo implemento cuando por ejemplo tengo que chequear si la burbuja esta pinchada, en vez de comparar el diagnostico de la persona con el string 'Positivo' le envio un mensaje 'tieneCovid'. De esta manera es encapsulo evito que burbuja conozca de alguna forma el diagnostico de la persona en especifico.

### 4.2. Utilización de herencia y delegación

Utilice herencia porque es la mejor forma de organizar las clases por su tipo sin tenerlos mezclados por toda la implementación. Utilice delegación para evitar que una clase posea mucha responsabilidad, y pasarle la tarea correspondiente a otra clase. De esta forma la implementación queda mejor organizada.

## 5. Excepciones

**BurbujaNoEncontradaError:** En caso de que se utilice el método "buscarBurbujaz la iteración devuelva 'None' se le informa al usuario que la burbuja que especificó no fue encontrada, por lo que el nombre está mal o no fue anteriormente creada.

**ColegioNoEncontradoError:** En caso de que se utilice el método "buscarColegioz la iteración devuelva 'None' se le informa al usuario que el colegio que especificó no fue encontrada, por lo que el nombre está mal o no fue anteriormente creado.

**PersonaNoEncontradError:** En caso de que se utilice el método "buscarPersonaz la iteración devuelva 'None' se le informa al usuario que la persona que especificó no fue encontrada, por lo que el nombre está mal o no fue anteriormente creada.

**MismoNombreBurbujaError:** En caso de que se utilice el metodo "nuevaBurbujaz ya exista una burbuja con el mismo nombre se le informa al usuario que no la puede agregar.

**MismoNombreColegioError:** En caso de que se utilice el metodo "nuevaColegioz ya exista un colegio con el mismo nombre se le informa al usuario que no lo puede agregar.

**MismoNombrePersonaError:** En caso de que se utilice el metodo "nuevaPersonaz ya exista una persona con el mismo nombre se le informa al usuario que no la puede agregar.

**NoHayBurbujasError:** En caso de que se quiera buscar una burbuja en una coleccion vacia se le avisa al usuario que se la puede buscar. Además evita realizar un ZeroDivide cuando se utiliza el método çlasesPresencialesEn".

**NoHayColegiosError:** En caso de que se quiera buscar un colegio en una coleccion vacia se le avisa al usuario que se lo puede buscar.

**NoHayPersonasError:** En caso de que se quiera buscar una persona en una coleccion vacia se le avisa al usuario que se la puede buscar.



**SintomaYaAgregadoError:** En caso de que se quiera agregar un síntoma a la persona que ya fue registrado anteriormente se le informara al usuario que no lo puede hacer.

## 6. Diagramas de secuencia

El siguiente diagrama representa la comunicación entre clases cuando se utiliza el método de AlgoVid 'burbujaPinchada'. Se lo dividió en dos partes para no dificultar la visualización del mismo.

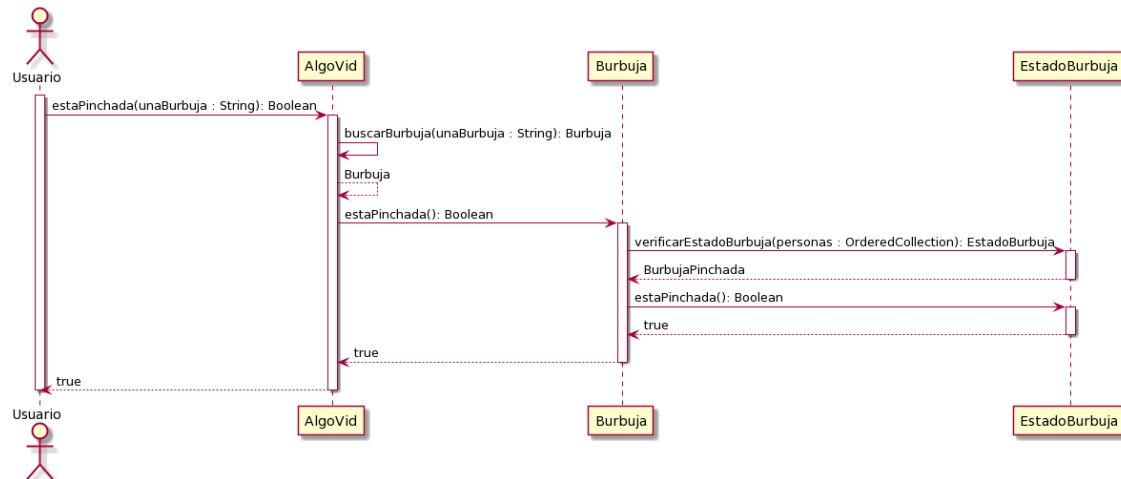


Figura 6: Primera parte del diagrama de secuencia de burbujaPinchada

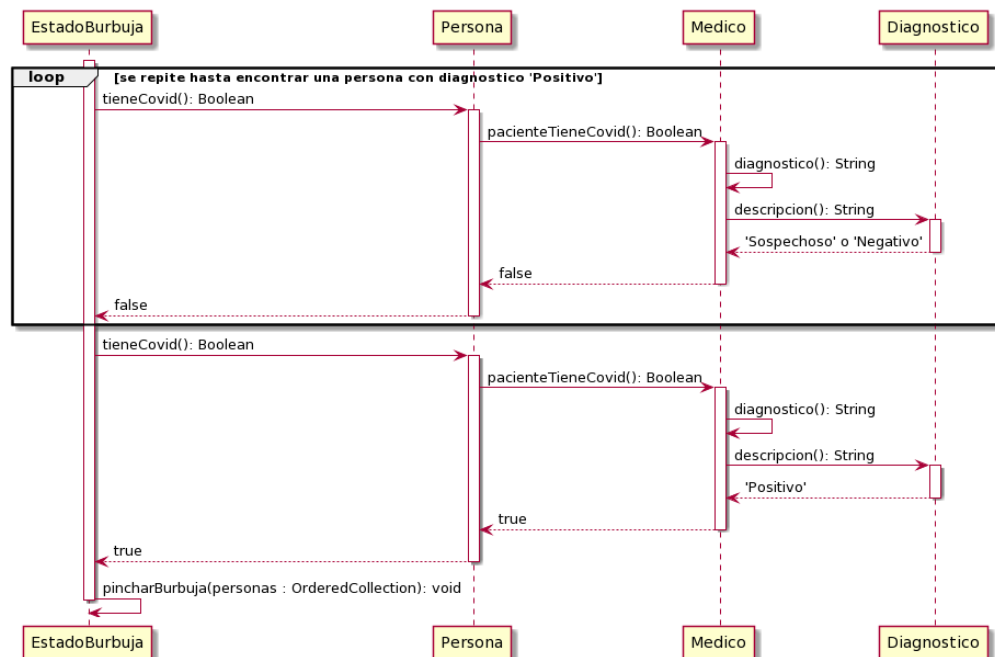


Figura 7: Segunda parte del diagrama de secuencia de burbujaPinchada

El siguiente diagrama de secuencia representa la comunicación entre clases cuando se le pide a AlgoVid el diagnostico de una persona recién agregada, el cual sera 'Negativo'.

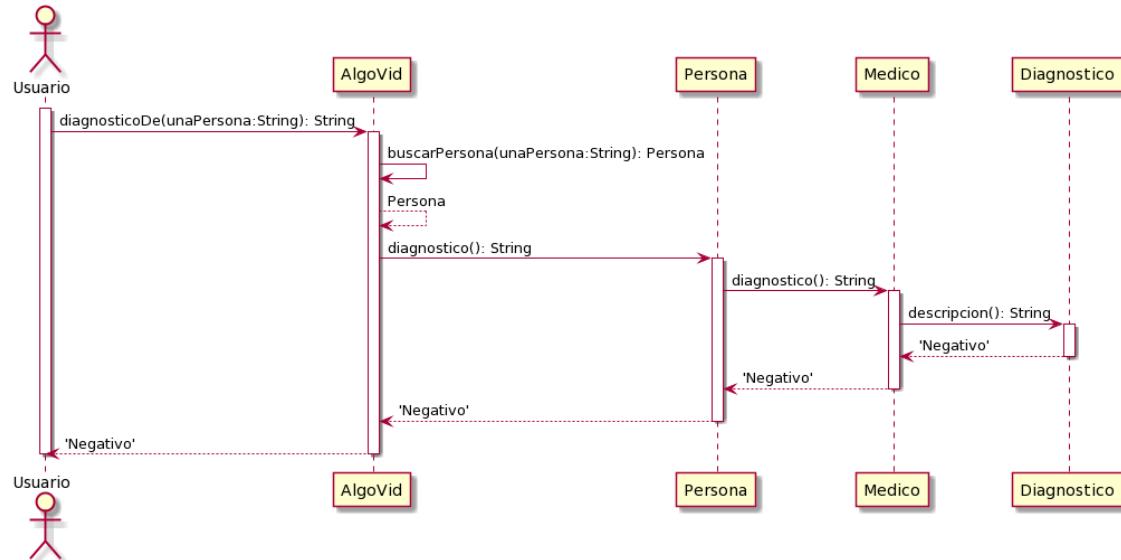


Figura 8: Diagrama de secuencia de diagnosticDe

El siguiente diagrama de secuencia representa la comunicación entre clases cuando se le pide a AlgoVid que informe la vacunacion de una persona.

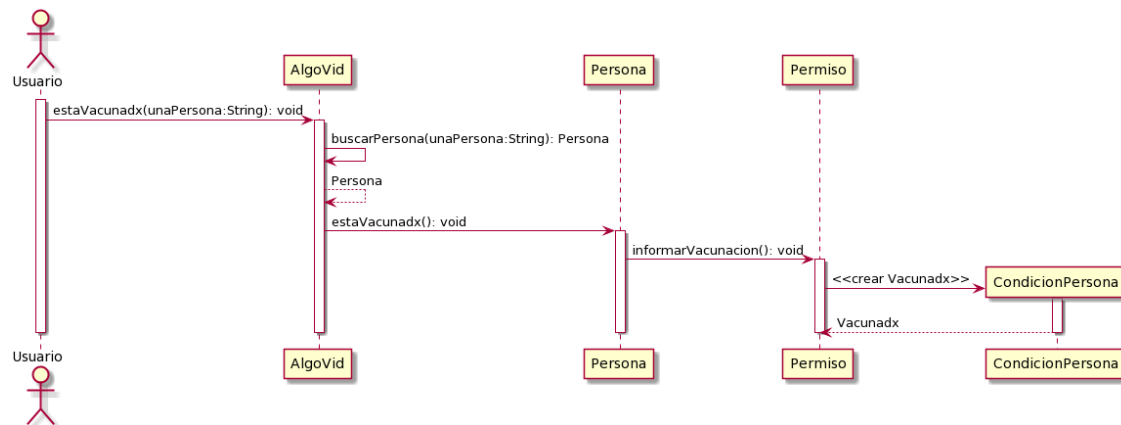


Figura 9: Diagrama de secuencia de estaVacunadx