

Trabajo Práctico 1 — Smalltalk

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

Alumno:	BIACH, Santiago
Número de padrón:	105793
Email:	sbiach@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
3. Diagramas de clase	3
4. Detalles de implementación	8
4.1. Pilares del paradigma	8
4.2. Herencia vs. Delegacion	9
5. Excepciones	9
6. Diagramas de secuencia	10

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 que permite diagnosticar personas, burbujas y colegios para determinar su estado en lo que a salud respecta. El programa fue creado en Pharo utilizando los conceptos del paradigma de la orientación a objetos vistos hasta ahora en el curso.

2. Supuestos

Como supuestos sobre las situaciones que no estan especificadas en el enunciado, he tomado que:

Supuesto 1 Los nombres de las personas, burbujas y colegios son case sensitive. Esto quiere decir que Juan y jUAn son dos personas distintas.

Supuesto 2 Los sintomas son case insensitive. Si alguien tiene Tos o tOS, hace referencia al mismo sintoma, y por ende son lo mismo.

Supuesto 3 Una persona puede tener una cantidad indefinida de contactos estrechos.

Supuesto 4 Una persona puede pertenecer a una cantidad indefinida de burbujas.

Supuesto 5 Si una persona tiene contacto estrecho con otra, cuyo diagnostico es negativo, y luego esta presenta sintomas y es diagnosticado positivo, la primera persona no puede circular.

Supuesto 6 No se puede tener dos personas, burbujas o colegios con el mismo nombre.

Supuesto 7 Las burbujas pueden tener una cantidad indeterminada de personas, y los colegios una cantidad ilimitada de burbujas.

Supuesto 8 Un colegio tiene clases presenciales si la relacion entre la cantidad de burbujas pinchadas respecto del total es menor o igual a $0,4(2/5)$.

Supuesto 9 Si un colegio no tiene burbujas, no se le puede preguntar si tiene clases.

Supuesto 10 Una persona no puede tener dos veces el mismo sintoma.

3. Diagramas de clase

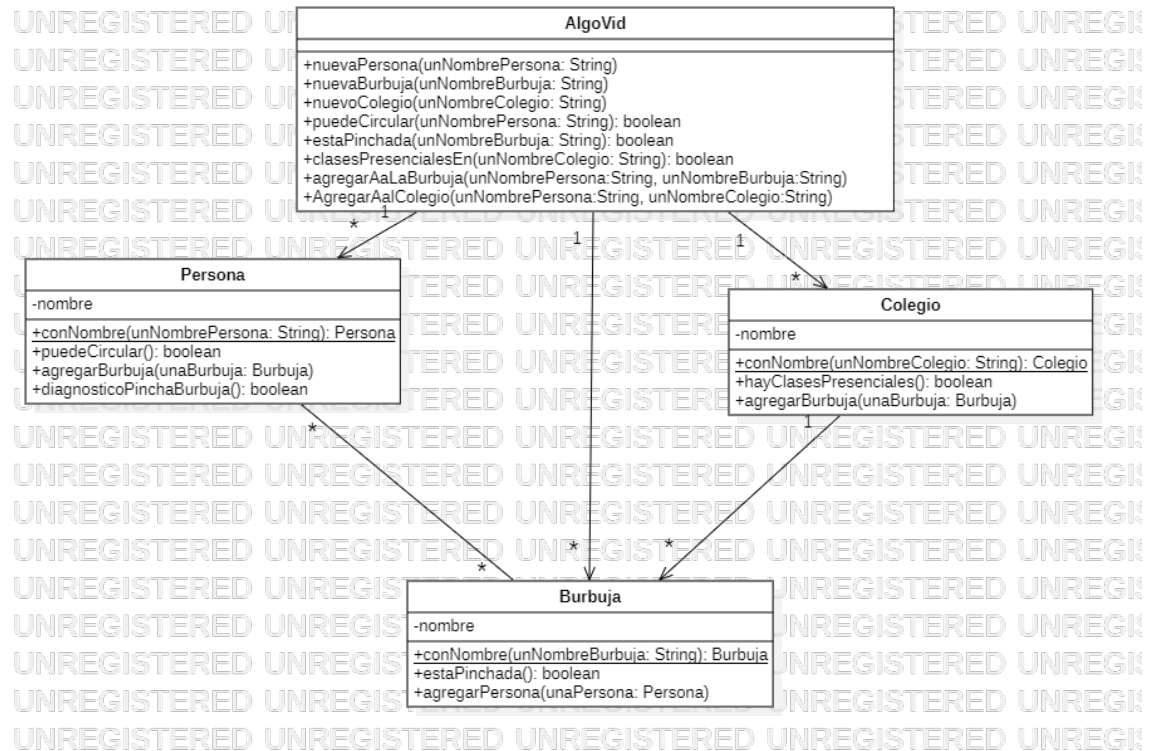


Figura 1: Diagrama de la estructura principal.

En el primer diagrama se muestra la estructura principal de la clase `algovid`, que cuenta con colecciones de `Personas`, `Burbujas` y `Colegios`. A su vez, los `colegios` tienen `burbujas`, y una `burbuja` puede tener varias `personas`. También una `persona` puede tener varias `burbujas`.

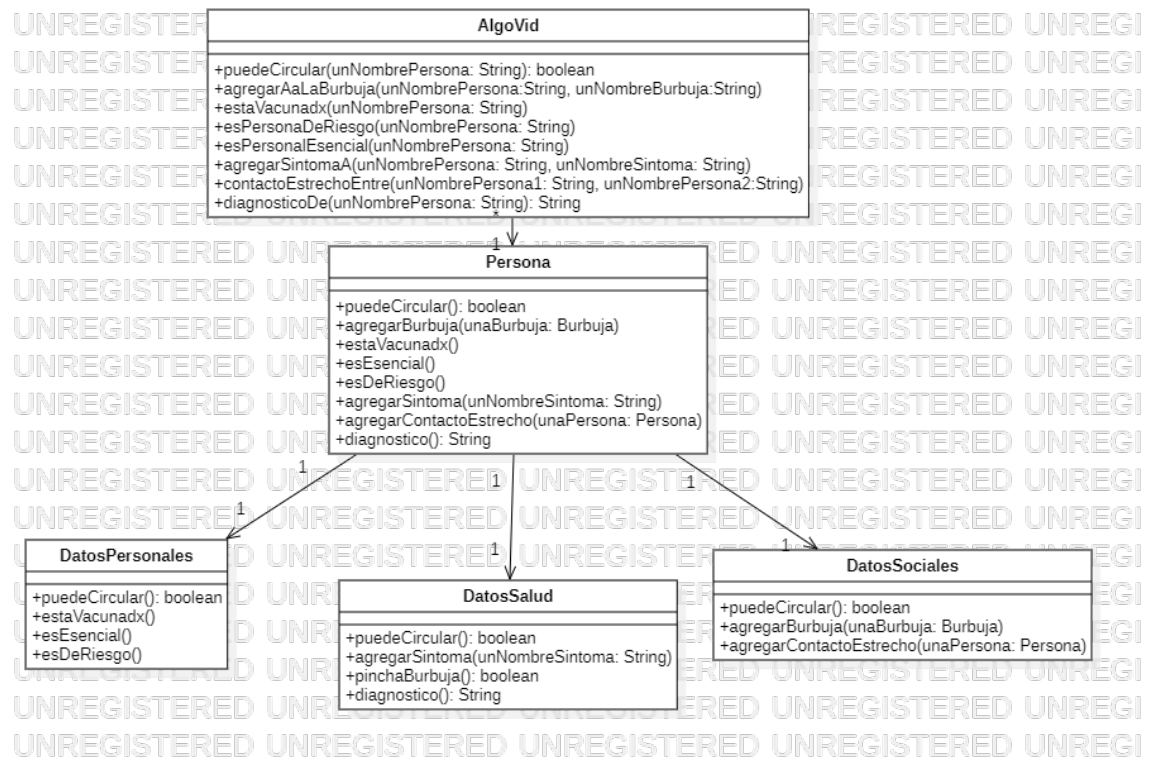


Figura 2: Diagrama de la estructura de persona.

En el diagrama de arriba se muestra la estructura de la clase persona. La clase persona delega su comportamiento a objetos intermedios, como lo son DatosSalud, DatosPersonales y DatosSociales.

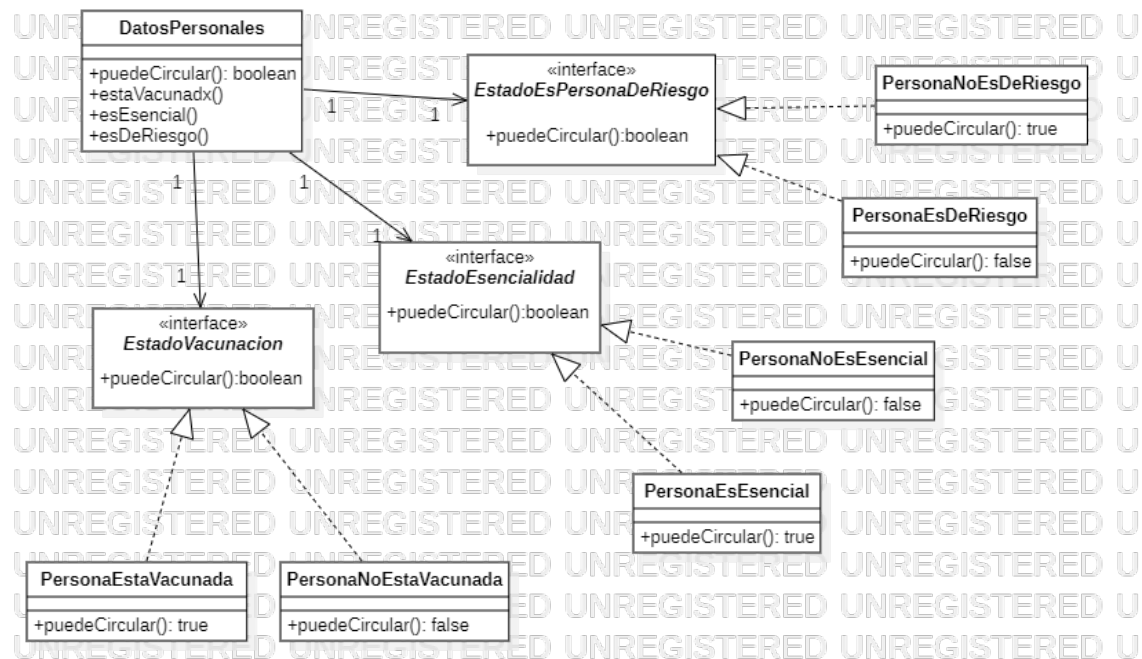


Figura 3: Diagrama de la estructura de DatosPersonales.

En este diagrama mostramos como es la estructura de la clase `DatosPersonales`. Esta tiene tres atributos, `EstadoVacunacion`, `EstadoEsPersonaDeRiesgo` y `EstadoEsencialidad`. Dependiendo en que estado este estos atributos, respondera de una forma u otra cuando le pregunten si puede circular.

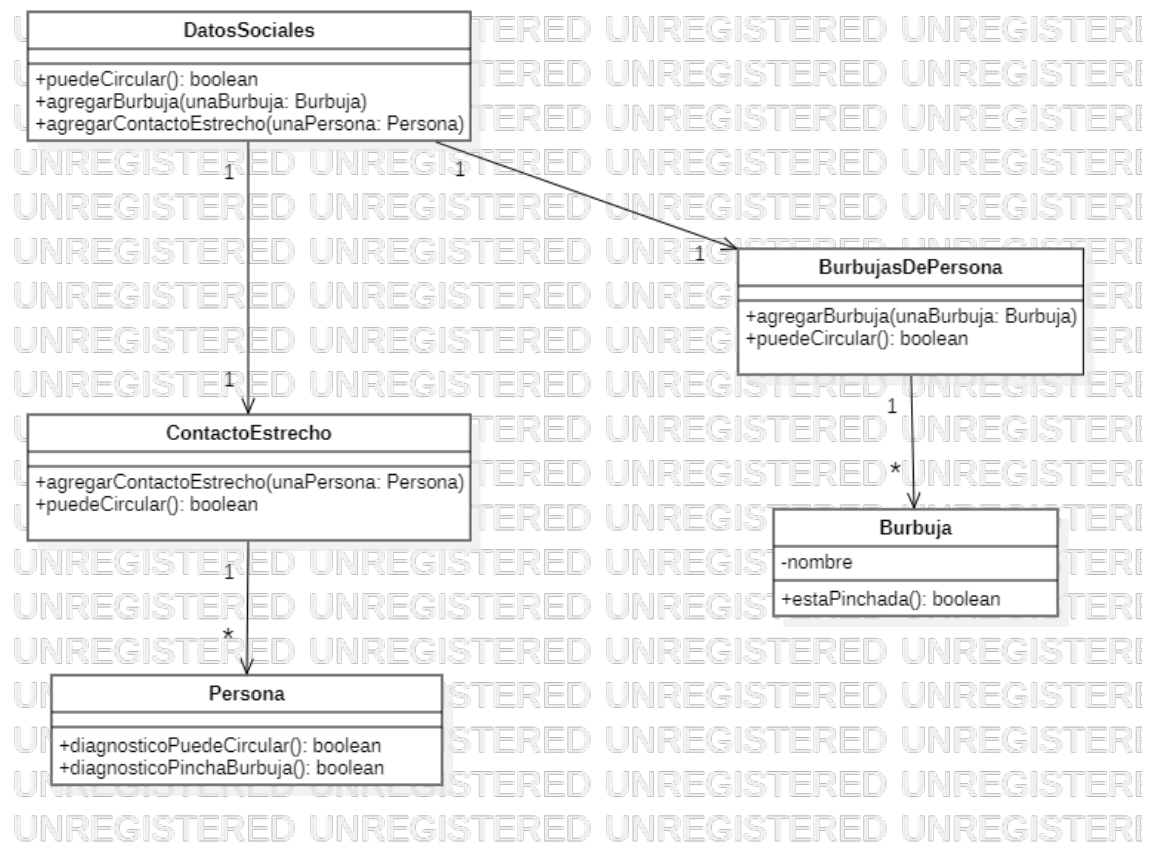


Figura 4: Diagrama de la estructura de DatosSociales.

En este diagrama mostramos como es la estructura de la clase `DatosSociales`. Cuenta de dos objetos en los que delega responsabilidad, `contactoEstrecho` y `BurbujasDePersona`. `ContactoEstrecho` posee una coleccion de personas, que usara para determinar si no puede circular. `BurbujasDePersona` tiene una coleccion de burbujas, que usara para determinar si no puede circular.

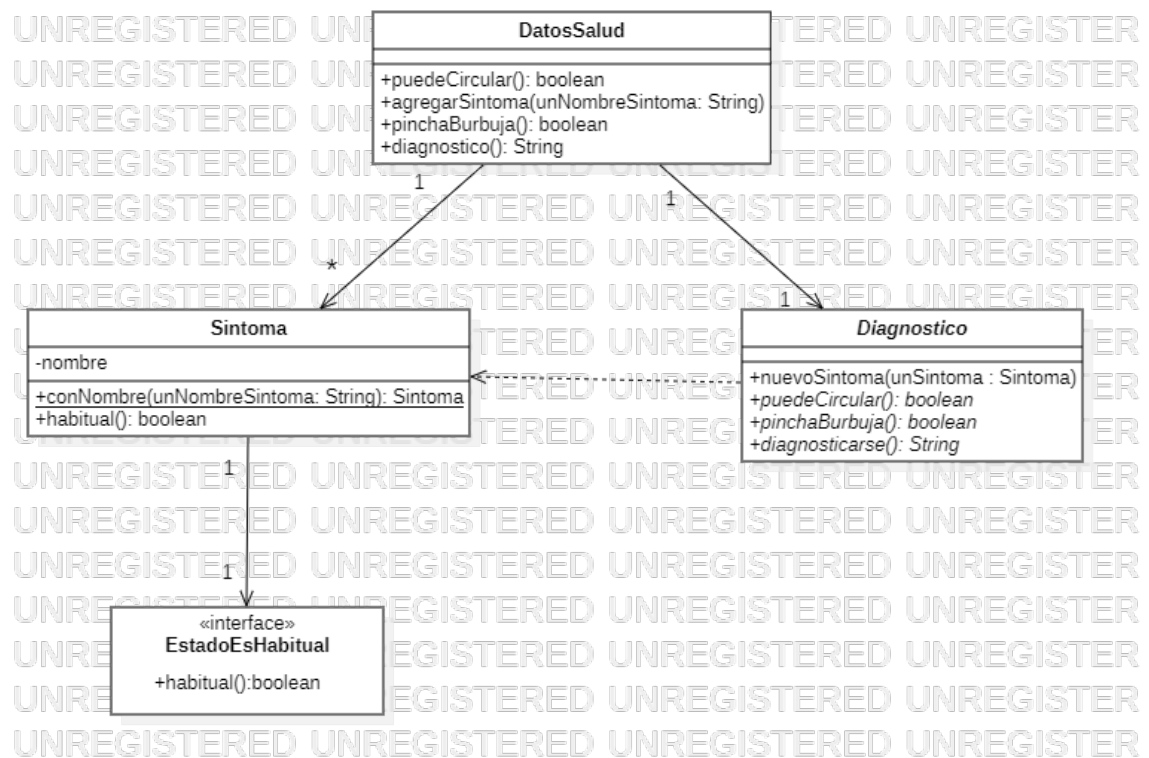


Figura 5: Diagrama de la estructura de DatosSalud.

En este diagrama mostramos la estructura de DatosSalud. Posee un diagnostico y una coleccion de sintomas. Estos sintomas pueden ser habituales o no habituales dependiendo de su estado. El diagnostico utiliza al sintoma nuevo cada vez que se agrega uno para evaluar que hacer.

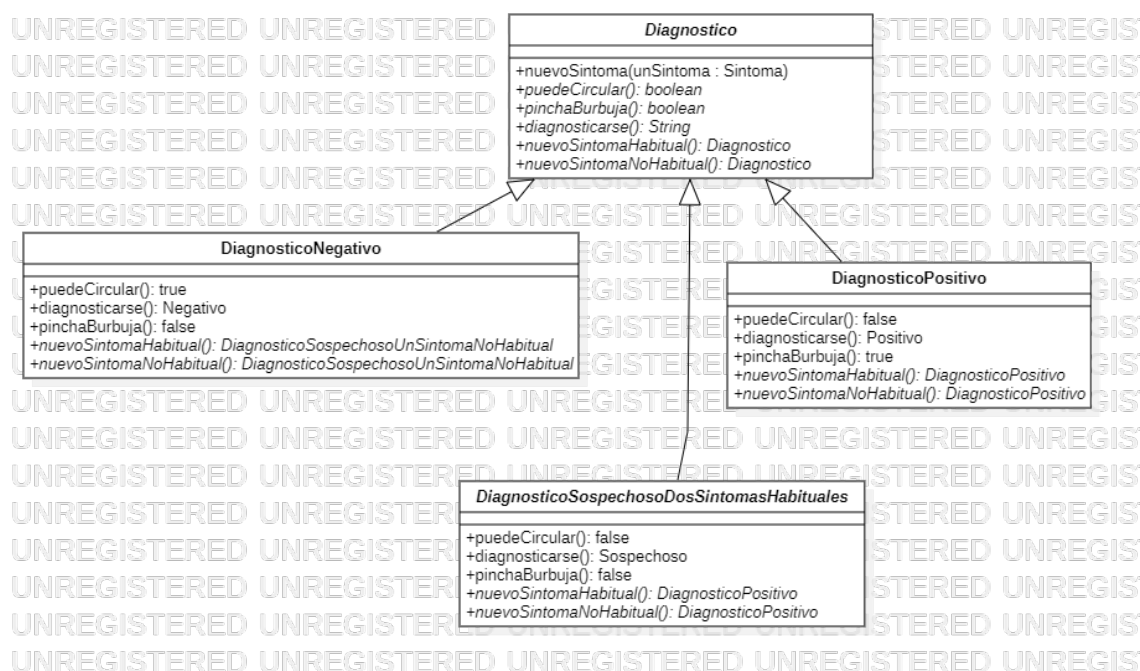


Figura 6: Diagrama de la estructura de Diagnostico y sus hijos.

En este ultimo diagrama mostramos a la clase Diagnostico con sus hijos. No estan todos, ya que se pierde legibilidad. Al llegar un nuevo sintoma, el diagnostico actual (por ejemplo negativo) Se pregunta si este es habitual o no, y decide que hacer. Si es no habitual, el diagnostico pasa a ser SospechosoConUnSintomaNoHabitual, si es habitual, pasa a ser SospechosoConUnSintomaHabitual. Este proceso se repite cada vez que llega un sintoma nuevo.

4. Detalles de implementación

4.1. Pilares del paradigma

En esta subseccion explicare un poco que pilares del paradigma utilice, explicandolos y proveyendo ejemplos.

Polimorfismo El polimorfismo, en palabras de Carlos Fontela, es la posibilidad de que distintos objetos respondan de manera diferente a la llegada del mismo mensaje. Esto lo utilice para, por ejemplo, diagnosticar a la persona y eliminar la mayor cantidad de ifs posibles, haciendo uso de de una clase diagnostico de donde heredan diagnosticos mas especificos.

```

diagnosticarse: unNuevoSintoma
  diagnostico := diagnostico nuevoSintoma: unSintoma.
  
```

El mensaje nuevo sintoma lo entienden todos los hijos de la clase diagnostico, lo que ayudo a eliminar ifs.

```

nuevoSintoma: unSintoma
  (unSintoma habitual)
  ifTrue:[ ^self nuevoSintomaHabitual]
  ifFalse:[ ^self nuevoSintomaNoHabitual]
  
```

Luego, dependiendo de si el sintoma es habitual o no, se le manda un mensaje al diagnostico y este responde.

Encapsulamiento Este concepto hace referencia a que cada objeto es responsable de responder a las mensajes que recibe, sin que quien envia el mensaje sepa como lo hace. Una razon es que puede haber infinitas implementaciones para una misma operacion, y si se quiere cambiar una implementacion por otra, este cambio no tiene que afectar al cliente de dicha operacion. Tuve este pilar presente, o al menos lo intente, durante todo el desarrollo del programa. Un posible ejemplo de esto es la clase Colegio.

```
CantidadBurbujasPinchadas
    ~ burbujas count: [:burbuja | burbuja estaPinchada ].
```

La clase colegio tiene el metodo cantidad burbujas pinchadas, que itera sobre su coleccion de burbujas y le pregunta a cada una si esta pinchada. Al colegio no le interesa saber que es lo que hace la burbuja para saber si esta pinchada, solo le interesa saber si lo esta o no.

4.2. Herencia vs. Delegacion

En palabras de Carlos Fontela, la herencia es una relacion entre clases, en la que una es un caso particular de la otra. Un ejemplo de donde utilice herencia es en la clase Diagnostico, de la que heredan Diagnostico Negativo, Positivo y Sospechoso. La delegacion es cuando un objeto, para responder un mensaje, envia mensajes a otros objetos. En el modelo que implemente, un claro ejemplo es el mensaje de persona puedeCircular

```
puedeCircular
    ~ datosPersonales puedeCircular & datosSalud puedeCircular &
      datosSociales puedeCircular
```

Aqui, para responder al mensaje puedeCircular delega en sus datos para responder.

5. Excepciones

PersonaInexistenteError, BurbujaInexistenteError, ColegioInexistenteError Agrupo estas tres excepciones en el mismo item, ya que a sus efectos hacen lo mismo. Estan excepciones son lanzadas cuando se intenta operar sobre una persona, burbuja o colegio que no existe. Fueron necesarias ya que, debido al uso de OrderedCollections, al intentar utilizar un objeto inexistente, lanzaba errorNotFound. Para que el usuario entienda mejor cual fue el error, capturo esa excepcion y lanzo una de las tres ya mencionadas.

PersonaYaExistenteError, BurbujaYaExistenteError, ColegioYaExistenteError Al igual que en el item anterior, estas tres excepciones estan agrupadas debido a su similitud. Son lanzadas cuando se intenta crear una nueva persona, colegio o burbuja con el nombre de una persona, colegio o burbuja que ya existe. Esto pasa ya que debido a como funciona el metodo detect de las OrderedCollection, si se agrega un elemento nuevo con el mismo nombre que uno ya existente, detect solo devuelve lo primero que encuentra. Por ende, nunca se podria usar al elemento nuevo. Por eso, al intentar crear una persona, burbuja o colegio nuevo con un nombre ya usado, lanza una de las tres excepciones.

NoHayBurbujasEnElColegioError Esta excepcion es lanzada cuando se intenta preguntar si hay clases presenciales en un colegio sin burbujas. Es necesaria ya que, debido a como fue implementado el metodo de hayClasesPresenciales, si no hay burbujas se lanza un ZeroDivide. Para que el usuario entienda mejor porque sucedio el error, capturo el ZeroDivide y lanzo esta excepcion.

SintomaYaExistenteError EstaExcepcion es lanzada cuando se intenta agregar dos veces el mismo sintoma a una persona. Decidi lanzar esta excepcion ya que, se podria dar el caso de que una persona tenga tres veces el sintoma Tos, y el diagnostico seria positivo. Para evitar este tipo de dilemas, al agregar un sintoma ya existente se lanza esta excepcion.

6. Diagramas de secuencia

Por ultimo, mostraremos un par de diagramas de secuencia.

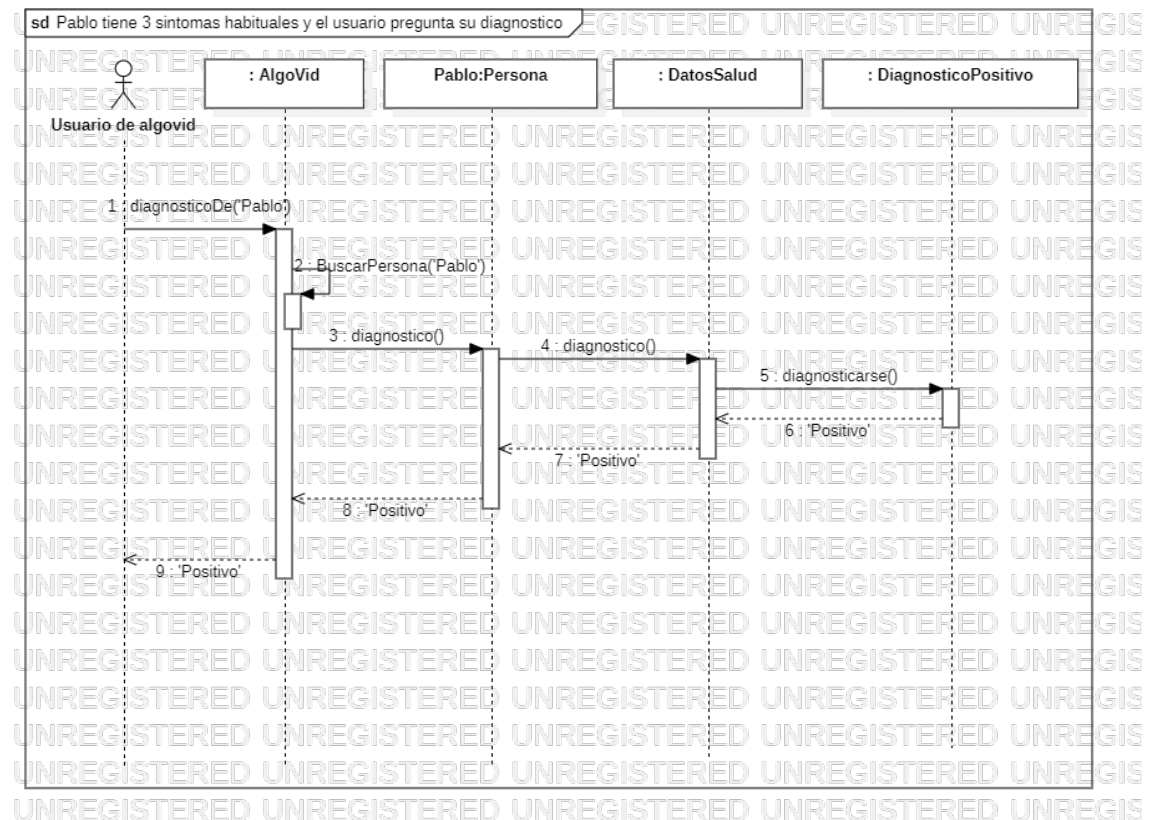


Figura 7: Pablo tiene 3 sintomas habituales y el usuario pregunta su diagnostico.

En el diagrama de secuencia de arriba vemos como un usuario pregunta sobre el diagnostico de una persona. AlgoVid busca a la persona y al encontrarla le pregunta su diagnostico. La persona le pregunta al objeto intermedio datos salud y este le pregunta a su diagnostico. Su diagnostico devuelve positivo.

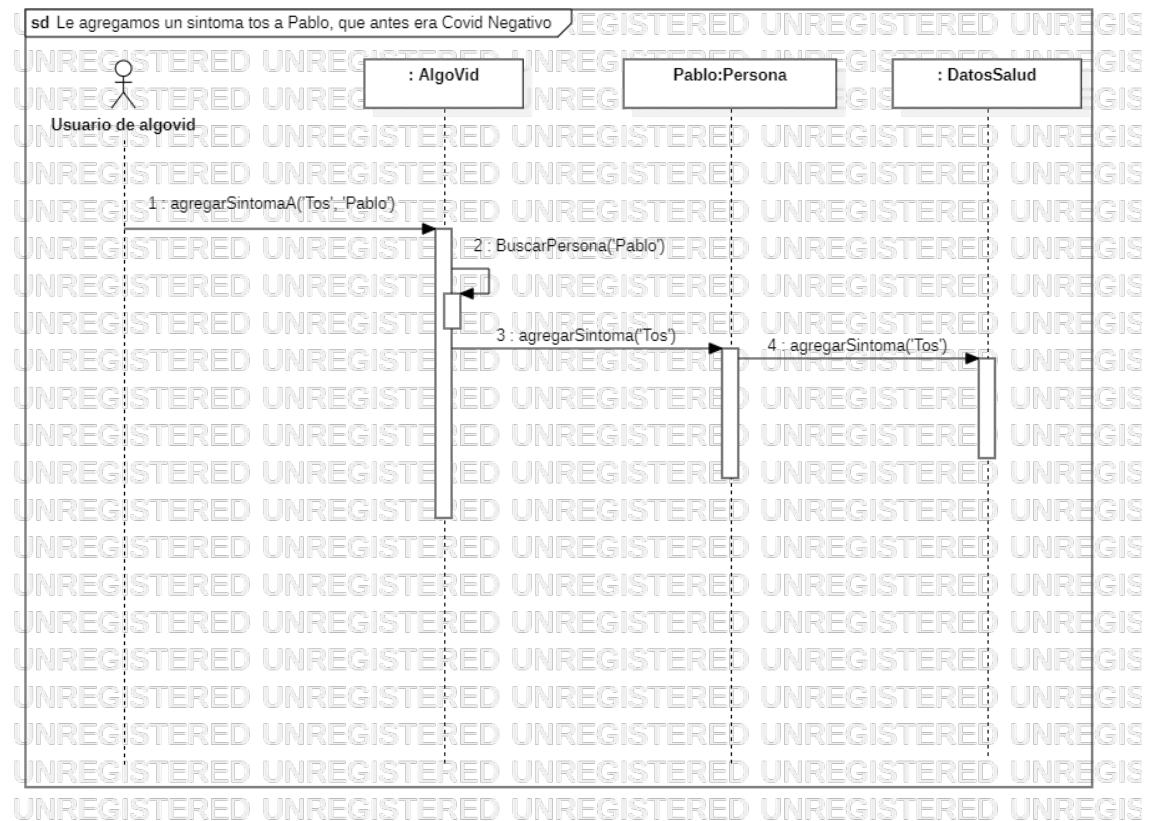


Figura 8: Le agregamos un sintoma tos a Pablo, que antes era Covid Negativo.

En este diagrama vemos como se le agrega un sintoma a Pablo, que antes era covid negativo. Algovid busca a la perosna y al encontrarla le agrega un sintoma Tos. A su vez, la persona llama a DatosSalud. (El diagrama sigue en la proxima figura)

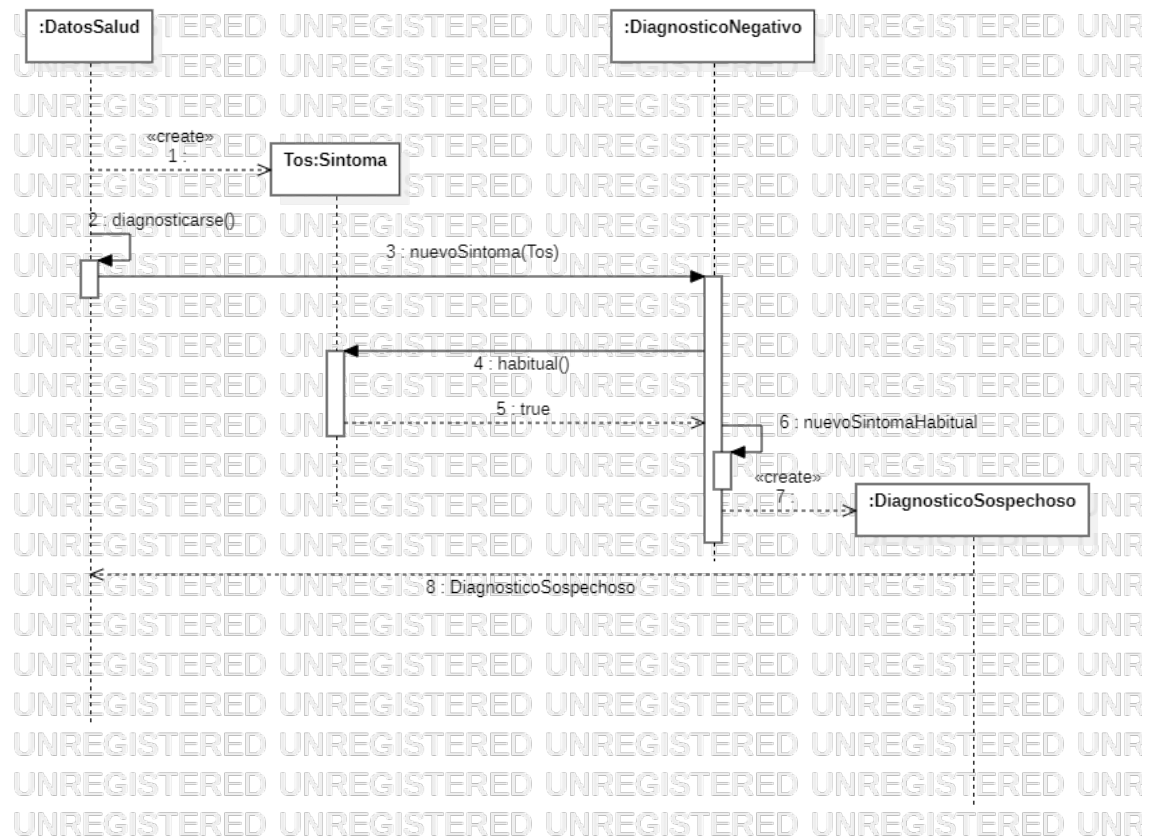


Figura 9: Seguimos con el diagrama anterior.

`DatosSalud` recibe el nombre del sintoma a agregar y crea un objeto `Sintoma`. Luego, actualiza su estado. Para eso le manda un mensaje a su diagnostico con el nuevo sintoma como parametro. Alli, el diagnostico le pregunta si es un `Sintoma Habitual`, a lo que `Tos` respondo `true`. Entonces, el diagnostico llama al mensaje `nuevoSintomaHabitual`, que crea un objeto `DiagnosticoSospechosoUnSintomaHabitual(No entraba en el diagrama)`, y lo devuelve. Entonces, el diagnostico de `DatosSalud` queda actualizado a `DiagnosticoSospechoso`.