

# Trabajo Practico I - SmallTalk

[7507/9502] Algoritmos y Programación III  
Curso 01  
Primer cuatrimestre de 2021

Alumno	Kübler, Andrés Tomás
Número de padrón	105238
Email	akubler@fi.uba.ar

## Indice

I. Introduccion.....	3
II. Supuestos.....	3
III. Diagramas de Clase.....	3
IV. Detalles de implementacion.....	5
i. AlgoVid	
ii. Persona	
iii. Burbuja	
iv. Colegio	
V. Excepciones.....	5
VI. Diagrama de secuencia.....	6

## I. 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 de un sistema de salud utilizando los conceptos del paradigma de la orientación a objetos visto hasta ahora en el curso.

## II. Supuestos

El presente trabajo fue diseñado e implementado a raíz de los siguientes supuestos

- ◆ AlgoVid no existe sin las clases Persona, Burbuja y Colegio, ya que esta clase se encarga de delegar los mensajes que se le envían y sin estas tres clases sería inútil.
- ◆ La clase Burbuja puede ser inicializada vacía, es decir, sin poseer personas.
- ◆ La clase Colegio puede ser inicializada vacía, es decir, sin poseer burbujas.
- ◆ Si las clases se suspenden, por tener 40% o más de burbujas pinchadas, estas no podrán volver a la normalidad.
- ◆ No se pueden crear y/o agregar dos objetos Persona, Burbuja o Colegio con el mismo nombre para una misma colección.
- ◆ Una Persona no puede pertenecer a más de una burbuja.
- ◆ Una vez que la Persona pertenece a una burbuja, no puede cambiarse/trasladarse a otra burbuja.
- ◆ Una vez que la Persona posee un nuevo síntoma este no se borra, es decir, la persona no puede volver a estar sana (diagnóstico “Saludable”).
- ◆ Una vez que una Burbuja está pinchada no puede volver a no estar pinchada.

## III. Diagramas de clase

Un diagrama de clase es un diagrama estático en el cual se representa la estructura de un sistema compuesto por clases, reflejando así sus atributos, métodos y las relaciones con otros objetos. A continuación se presentan algunos diagramas de clase correspondientes al trabajo práctico.

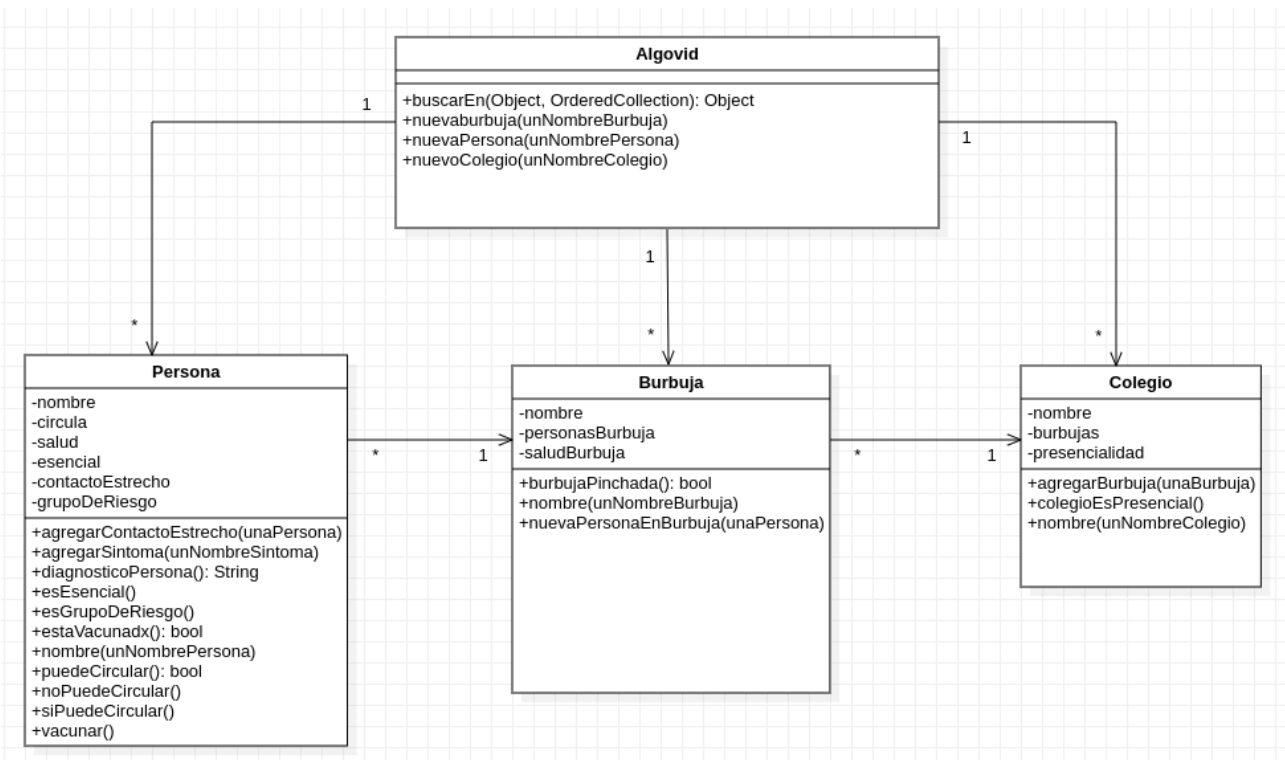


Figura 1: Diagrama mostrando las relaciones directas de la clase AlgoVid.

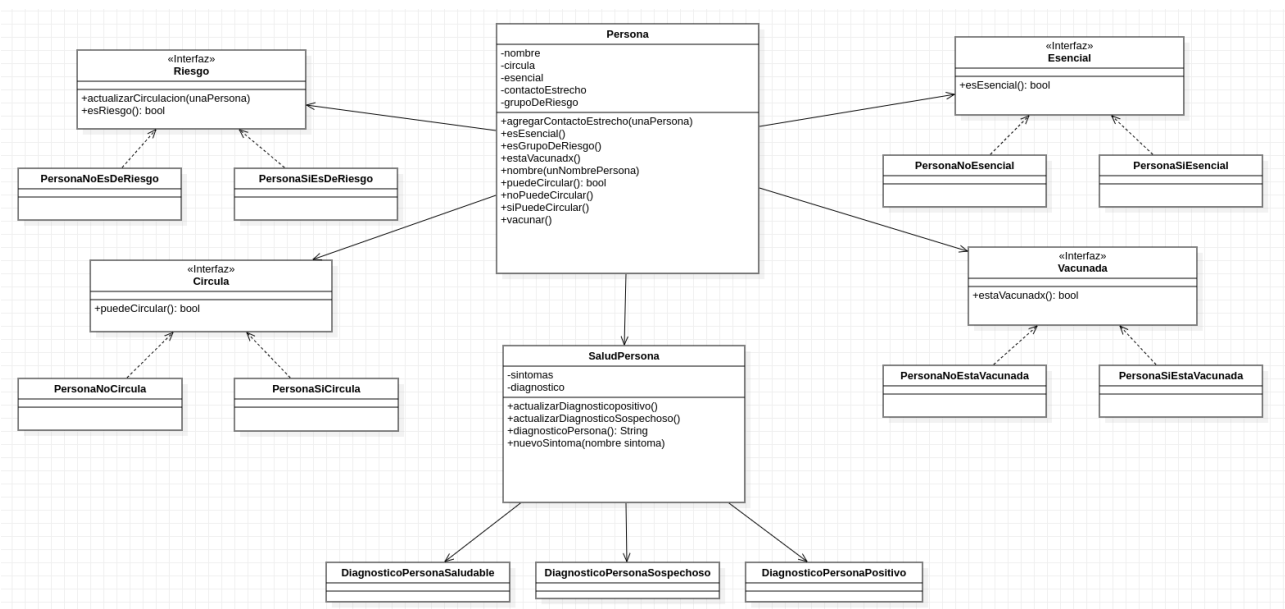


Figura 2: Diagrama mostrando la clase Persona con sus metodos y atributos.

## IV. Detalles de implementacion

A lo largo de la implementacion del trabajo practico, surgió la necesidad de refactorizar muchas veces debido a la repeticion de codigo, logrando asi la creacion de varias clases abstractas que son utilizadas en las clases principales del programa. Las clases que heredan de la clase abstracta cumplen con la relacion “es un” , pilar fundamental de la POO.

### i. AlgoVid

Esta es la clase principal del trabajo practico. Se encarga de recibir mensajes del Actor y realizar operaciones y delegar mensajes a otras clases. Internamente esta compuesto por tres colecciones: una colección para almacenar Personas, otra para almacenar Burbujas y finalmente otra para almacenar Colegios.

### ii. Persona

Esta clase es un pilar del programa ya que con ella se pueden realizar operaciones en otras clases del programa, ampliando asi su funcionalidad. Como variables de instancia posee circula, vacunado, salud, esencial, contactoEstrecho y grupoDeRiesgo.

La razon de la creacion de esta clase fue la simplificacion y manejo de informacion a traves de objetos., facilitando asi las operaciones de cada Persona individualmente en otras clases del programa.

### iii. Burbuja

Esta clase es la encargada de administrar un conjunto de objetos Persona compartiendo asi características grupales entre ellos, tales como *BurbujaPinchada* la cual impide que todos las personas de la burbuja puedan circular. Como variables de instancia posee personasBurbuja y saludBurbuja.

### iv. Colegio

El colegio es una clase que engloba una n cantidad de burbujas, es por eso que cumple un rol muy importante en el momento de manejar perfectamente las clases Burbuja y Persona para que estan puedan utilizarse mediante herencia .Como variables de instancia posee burbujas y presencialidad.

## V. Excepciones

**ColeccionNoEncontradaError** Esta excepcion tiene como finalidad evitar problemas en el momento que el usuario quiera buscar un objeto, mediante la operación de *AlgoVid buscar:En:* , en una colección inexistente.

**ObjetoBuscadoNoExisteError** Esta excepcion tiene como finalidad evitar problemas en el momento que el usuario quiera buscar un objeto, mediante la operación de *AlgoVid buscar:En:* , en una colección a la cual no pertenece o si el objeto no existe.

**ObjetoYaExisteEnColeccionError** Esta excepcion tiene como finalidad evitar problemas en el momento que el usuario quiera agregar un objeto con el mismo nombre en una coleccion. Para eso creamos el metodo de AlgoVid *existe:En:* el cual busca al objeto y si lo encuentra lanza la excepcion.

## VI. Diagramas de secuencia

Los diagramas de secuencia son un tipo de diagrama de interaccion cuyo objetivo es describir el comportamiento dinamico de una operación especifica y/o general. A continuacion se presentan algunos diagramas de secuencia correspondientes al trabajo practico.

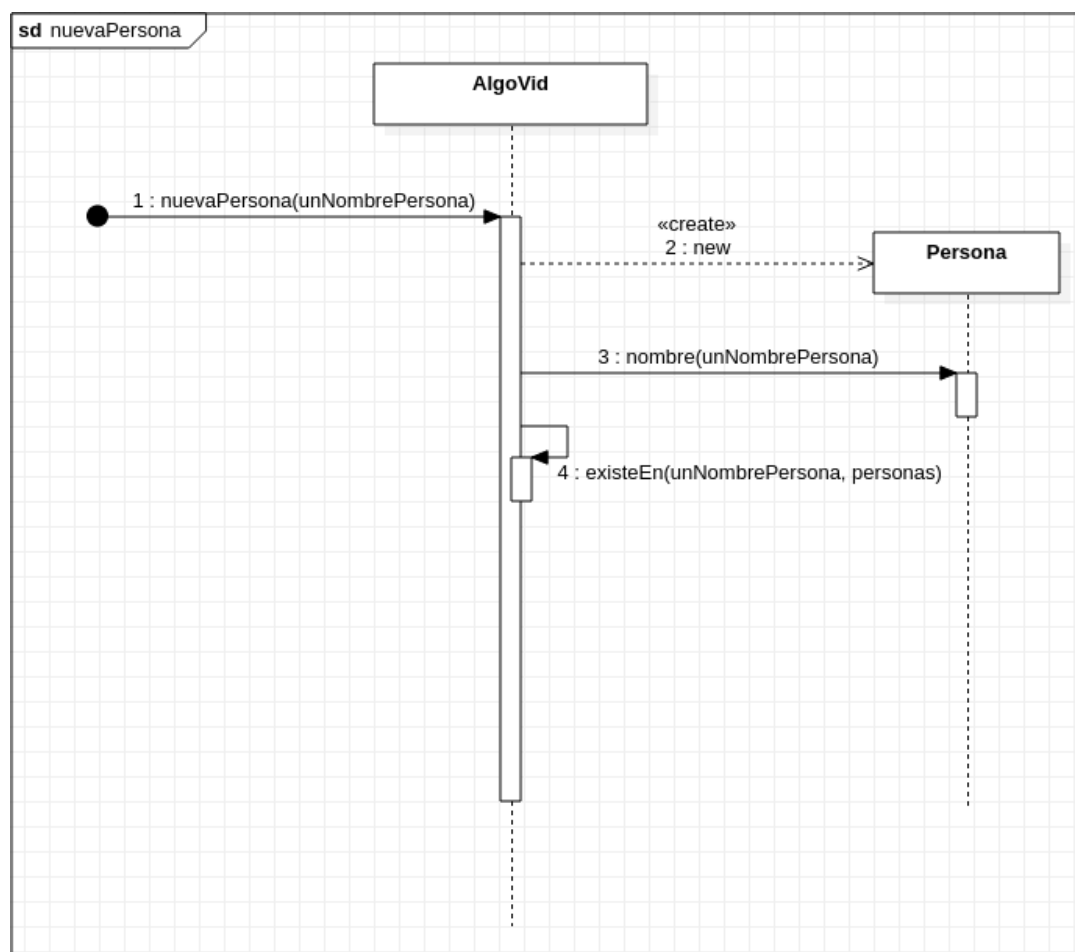


Figura 3: Creacion de una Persona

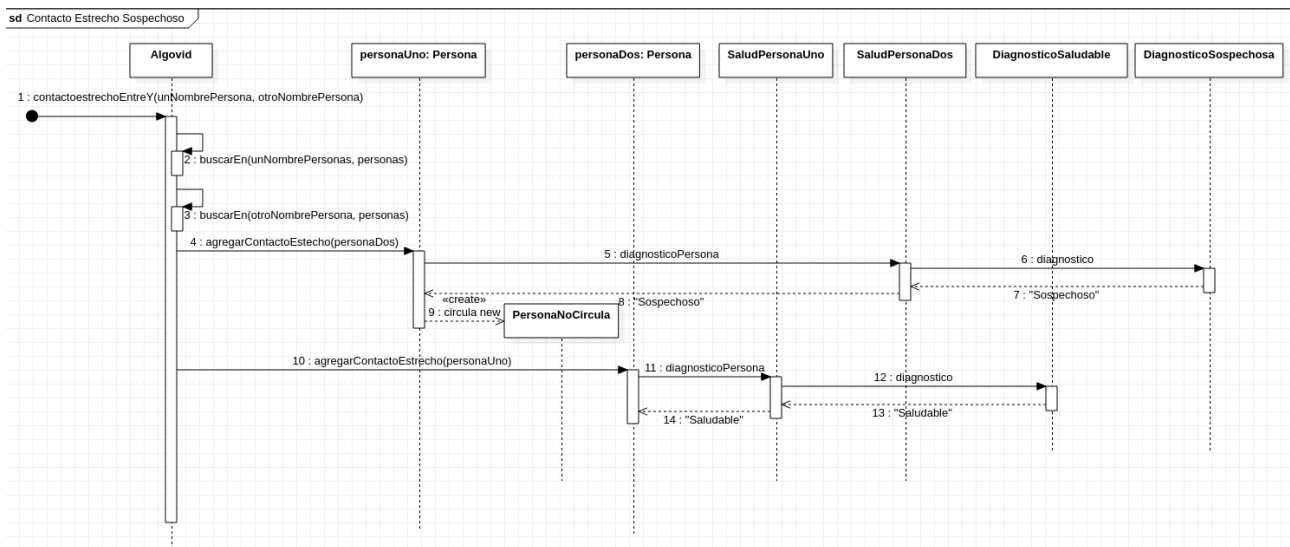


Figura 4: Contacto estrecho entre una persona saludable y una sospechosa

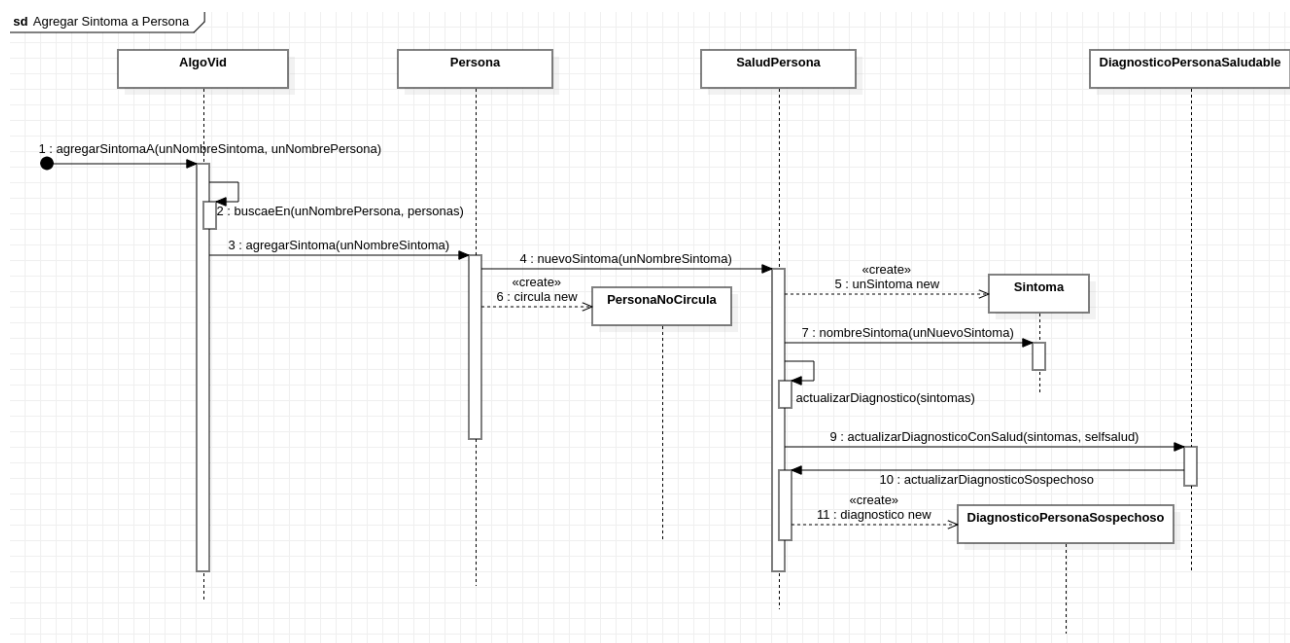


Figura 5: Agregar sintoma a una persona saludable