

# Entrega 3: Codificación

PADSOF

Grupo 5

David del Val,  
Junco de las Heras,  
Jorge Fernandez

Doble grado Informatica-Matematicas

Grupo 2201

UAM, Marzo 2020

# Diagrama de clases actualizado

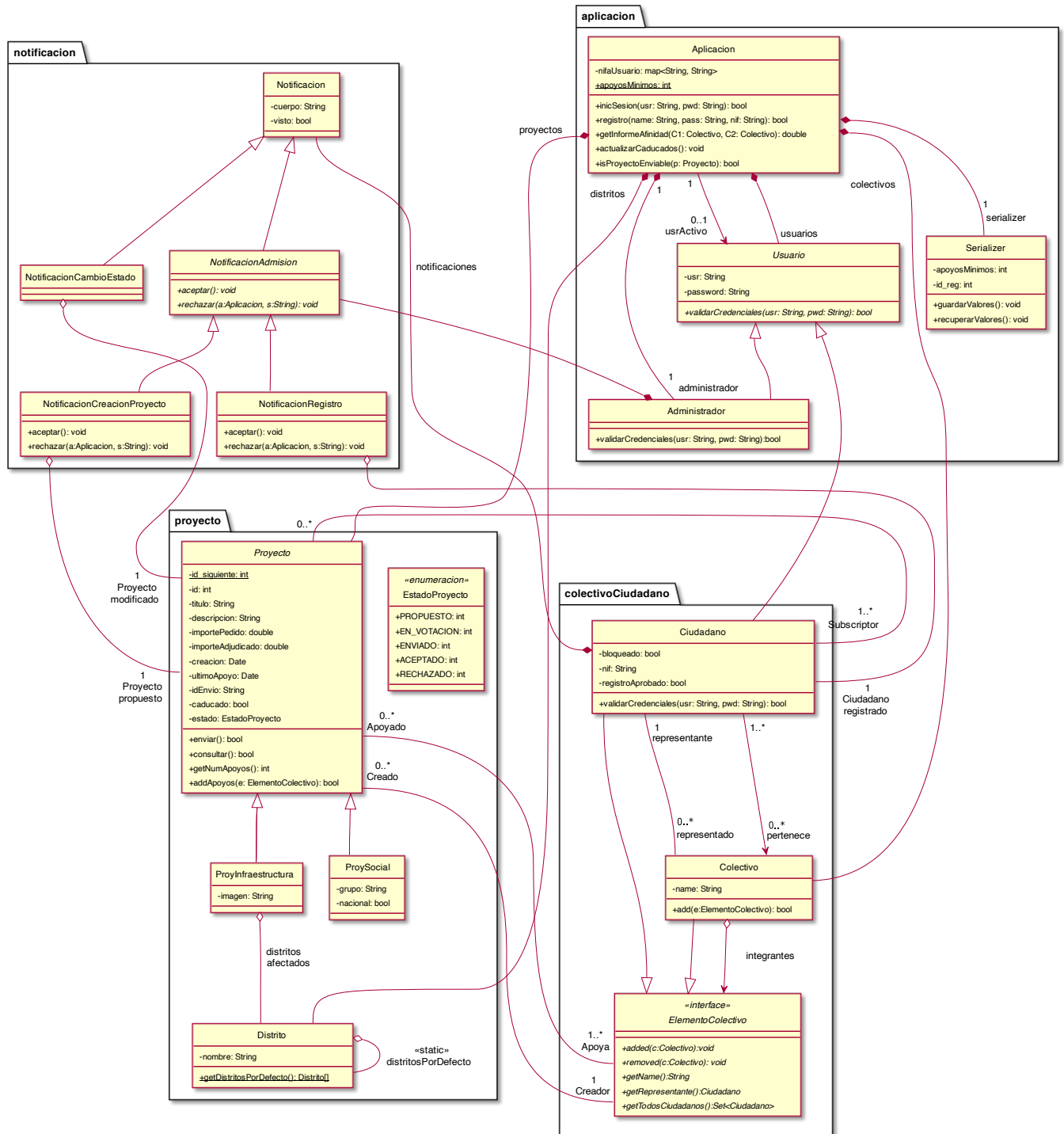


Figura 1: Diagrama de Clases

En la página siguiente hemos incluido unos comentarios sobre los cambios al diagrama de clases.

Para esta entrega, hemos efectuado una serie de cambios al diagrama de clases. Dejando a parte los debidos a errores previos que han sido corregidos, la mayoría de ellos se han debido a nuevos factores que no se habían previsto en la fase de diseño y que han aparecido al llevar el diseño anterior a la práctica.

En primer lugar, hemos separado las clases en distintos paquetes. Esto ha sido necesario para poder trabajar cómodamente con tantos archivos de código en un mismo proyecto. Respecto al criterio que se ha tomado para esta división, en algunos aspectos es claro (como el paquete de notificaciones) mientras que en otros se han tomado decisiones ligeramente menos motivadas (como la inclusión del ciudadano en el paquete **ColectivoCiudadano** cuando hereda de *Usuario*). Aún con todo esto, esta división facilita enormemente el manejo de los archivos y ayuda a separar conceptualmente el diseño en varios bloques.

Otro de los grandes cambios ha sido la creación de diversos tipos de notificaciones. Si bien inicialmente consideramos que sería suficiente con un campo de texto en la notificación, a la hora de implementarlo hemos optado por incluir diversos tipos para poder guardar, por ejemplo, una referencia al ciudadano que se ha registrado en la notificación correspondiente al registro.

Por otra parte, hemos añadido más métodos al diagrama si bien no están todos incluidos. Como norma general, hemos añadido los métodos que no eran setters, getters, constructores o para añadir o eliminar elementos. A excepción, claro está, de algunos que hemos considerado sumamente importantes, como es el caso de **addApoyos** en **Proyecto**. Además, se han añadido tanto el método de generar informes de afinidad (**Aplicacion.getInformeAfinidad**) como el de generar informes de popularidad (**Proyecto.getNumApoyos**).

Otra diferencia importante son los métodos en la interfaz **ElementoColectivo**. En el diseño inicial no habíamos considerado ninguno de ellos y estos han surgido según hemos encontrado situaciones en las que su uso nos simplificaba la implementación o mejoraba el diseño respecto a futuras expansiones.

Por último, queremos mencionar, aunque sea brevemente, la adición de la clase **Serializer**. Esta clase no guarda información relevante durante la ejecución de la aplicación, sino que asiste en su serialización y deserialización. En particular, utilizamos una instancia de **Serializer** para guardar los valores de las variables de clase que utilizamos en el programa ya que estas no se guardan en la serialización de los objetos de sus respectivas clases.