Documentación TPA - 1era entrega

Decisiones de diseño tomadas y su justificación:

Como primera aproximación decidimos modelar la clase **Servicios Públicos** en la que podemos establecer las líneas existentes y el tipo de transporte del servicio en cuestión, a su vez el sistema deberá permitirnos agregar, eliminar y modificar el estado en que se encuentran las líneas de los diferentes servicios

Luego preferimos modelar la clase **Línea** (ya que un servicio puede poseer muchas líneas), las mismas deben poseer un nombre que las identifique así como también su estación de origen, destino, el conjunto de estaciones que componen a la misma y el estado. También se deben poder agregar y eliminar estaciones a la línea.

Decidimos hacer un enum para los **Estados** ya que es una cantidad de estados finita la cual no se debe modificar frecuentemente, y además para poder cambiar entre estados fácilmente como se requiere para Servicio y Línea.

Elegimos crear la clase **Comunidad** con atributos descripción, la cual incluye de qué trata la comunidad en cuestión y cualquier detalle que los administradores deseen agregar, una lista de los miembros y una lista de los administradores. Además cuenta con atributos para administrar los miembros, saber si un miembro determinado es administrador, y agregar un servicio a una determinada estación, función clave que deben poder cumplir todas las comunidades.

La clase de **Miembro** contiene una lista con las comunidades a las que pertenece, y métodos para unirse, o irse de comunidades específicas, además de otro método que le permite saber si es administrador de alguna comunidad en particular.

También creamos la clase **Servicio** a partir de la cual se podrá definir el estado del mismo y una descripción en la que se podrá aclarar si el servicio es un baño, que contenga al de mujeres, hombres, discapacitados, etc.

Generamos además una clase **Estación** la cual cuenta con un nombre, su ubicación y los servicios que posee. En la misma podremos agregar, eliminar o modificar el estado de cada servicio que contenga.

Por último, la **Ubicación**, que está compuesta por una latitud y una longitud, decidimos modelar de esta forma para garantizar la unicidad de la ubicación de cada estación.

En lo que respecta a los **requerimientos de seguridad**, decidimos guiarnos por un patrón strategy, ya que nos permite mantener un conjunto de algoritmos o requerimientos para conformar una contraseña y que los mismos sean modulares y extensibles respecto de las necesidades del cliente.