

Decisiones de diseño

Creación del buscador: Para solucionar el tema de la búsqueda decidimos crear una clase externa a los puntos de interés del dominio con el objetivo de que el Buscador pueda contener una lista de puntos de interés y pueda manejar la búsqueda sobre esta lista. De esta forma el Buscador maneja el criterio de semejanza en base al texto ingresado y por ello puede filtrar y ordenar la *lista de puntos* requiriéndoles a cada uno únicamente sus campos propios.

Herencia de punto de interés: La herencia con la que se modelaron los puntos de interés permite no repetir lógica en cuanto a ciertos métodos que se comportan similar como es el caso de *estasCercaDe(Point)* y *listaDeTags()* (utilizado por el buscador). Los cambios sobre estos métodos simplemente se redefinen en la subclase puntual (como ocurre con *estasCercaDe(Point)* de los CGP). Análogamente decidimos modelar a los locales comerciales ya que el método de *estasDisponible(DateTime)* se comporta idénticamente para todos (solo varía el horario de atención).

Dirección: La abstracción de la dirección fue más que nada para sacar tantas variables de estado interno dentro del punto. Podrían haber quedado dentro del POI pero nos pareció cómodo extraer la idea.

Comuna: La comuna se creó más que nada para poder aportar el nombre de la “comuna” en la búsqueda y hacerla más eficiente (Sin ello podría haber sido tranquilamente un Polygon)

Horario: Se toma la decisión de crear esta clase, que aloja una *lista de días hábiles* y los *turnos* de cada Punto de Interés (en el caso de los CGP, de los Servicios), para delegar tareas que no son propias de los Puntos de Interés.

Turno: La clase que se comporta como un *Intervalo De Tiempo*, entre una hora de inicio y una de fin, son los turnos disponibles en cada día hábil que se creó debido a no encontrar un tipo de dato para intervalos en **Joda-Time** (se encontró pero con intervalo mínimo de 1 día). Se encuentran en una lista dentro del horario ya que hay algunos Puntos de Interés con distinta cantidad de turnos, por ejemplo la librería Carrousel tiene 2 turnos distintos separados por un intervalo

Servicios: Decidimos representar los servicios de un CGP mediante una lista de la clase servicio que se encargará de brindar su horario de atención cuando se consulte por ese servicio en particular. De esa manera se puede trabajar con varios servicios a la vez y se pueden cambiar los mismos en el caso que se desee (cosa que nos pareció razonable ya que la idea de agregar o quitar servicios es algo factible).