

Entrega 4 TP: Persistencia

Decisiones generales:

- La solicitud de alta que una persona hace para ser parte de un área tiene un estado que puede ser ACEPTADA, PENDIENTE o RECHAZADA.
- Cada parada tiene dos campos que hacen referencia a la parada siguiente y a la parada anterior respectivamente, ya que eso es necesario para realizar el cálculo de distancias.
- Si dos paradas de diferente línea están en la misma dirección, las guardaremos como paradas diferentes. Esto es necesario, ya que en cada parada guardamos la distancia hasta la parada anterior y a la siguiente, y esto es único de cada parada de cada línea.
- **SentidoRecorrido** no lo persistimos porque es stateless, los resultados de sus métodos dependen de los parámetros enviados.
- Toda provincia o municipio que se dé de alta debe tener su registro en la tabla de **Sector** para que los agentes sectoriales puedan relacionarse con esta. Las cargamos al iniciar la aplicación.
- La tabla de localidades utiliza de id el número que viene de la API. Por esto, es una id natural, no un autogenerado.
- El tipo de las líneas de transportes públicos es un atributo de la clase "Línea", cuyo tipo será un enumerado con los valores "Colectivo", "Subte" o "Tren". Decidimos esto porque el comportamiento no cambia dependiendo del tipo, por lo que utilizar una herencia sería una mala decisión. Subte, Colectivo y Tren serán instancias de la clase **tipoLinea**.

Desnormalización:

- Decidimos persistir la **huella de carbono** de cada **persona** (para no realizar constantemente el cálculo con los tramos y trayectos que puede ser muy pesado). Para ello necesitamos guardar el valor como un DOUBLE y la fecha del cálculo para saber si debemos recalcularla o no. De esta manera estamos mejorando la performance del sistema, haciendo también que la experiencia que tengan las personas con la aplicación sea mucho mejor. Como solo nos interesa guardar la última huella calculada de las personas, decidimos incluirlo en la misma tabla de personas.
- Decidimos persistir la **huella de carbono** de cada **organización**. Como en los requerimientos se nos pide guardar la evolución de la misma, una organización tendrá muchas huellas de carbono que corresponden a cada período de tiempo. Es por esto mismo que decidimos crear una tabla llamada "huella" en la que almacenaremos:
 - id de la huella.
 - id de la organización.
 - fecha del cálculo (DATE).
 - valor de la huella (DOUBLE).

De esta manera, el discriminar por provincia o municipio es muy sencillo, y además tenemos registro de las fechas de cada una.

- Decidimos que la dirección conozca directamente a su localidad, municipio y provincia, ya que si bien podría conocerlos haciendo Joins solo sabiendo la localidad, sería poco performante consultar 5 tablas para saber la provincia de una

organización, ya que debería consultar a la organización, a la dirección, a la localidad, al municipio y a la provincia.

Mapeo de herencia:

- Los **tramos** los vamos a persistir con la estrategia de mapeo de herencia JOINED, teniendo un campo "tramo_concreto_id" que apuntaría a cada una de las tablas hijas y agregando un campo discriminante que indique a qué tabla hija pertenece. Esto resultaría en 5 clases:
 - tramo: la clase padre con los atributos en común para todos los tramos.
 - tramo_contratado.
 - tramo_particular.
 - tramo_limpio.
 - tramo_publico.

Esto implicaría que usemos la estrategia TABLE PER CLASS con los tramos privados. Desaparece tramo privado y quedan sus hijas:

- tramo_contratado.
- tramo_particular.
- tramo_limpio.
- Decidimos que para persistir las **actividades** vamos a utilizar la estrategia SINGLE TABLE, guardando todos los atributos de todos los tipos de actividad en una sola tabla.