

Trabajo en duplas: Final Cuidándonos 27/02/2019.

Integrantes:

-Federico Leonel Bietti 175663-1.

-Cosme Fulanito 00000-0.

Contexto general

Punto 2 - Modelo de Dominio (40 puntos)

Diagrama de dominio:

Justificaciones y decisiones de diseño:

Contexto general

El Gobierno de la Ciudad nos ha contratado para la realización de un sistema de seguridad personal, el cual debe asegurar que los vecinos de todas las comunas puedan caminar de un destino a otro sin peligro. Será una aplicación Mobile en donde una persona, el transeúnte, podrá escoger quienes serán sus cuidadores personales durante todo el trayecto que recorra.

Siendo este el dominio general, se deberán tener en cuenta las siguientes especificaciones:

1. De las personas nos interesa saber: nombre y apellido, dirección, edad y sexo.
2. Para que una persona sea cuidadora de otra deberá tener instalada la aplicación; o sea que al menos deberá ser un usuario pasivo. Se considera usuario activo a aquel que solicita los acompañamientos.
3. Cada vez que un usuario quiera ir hacia un destino, deberá especificar la dirección exacta donde se encuentra actualmente y la del destino final; además de escoger quiénes serán sus cuidadores (puede haber un solo cuidador). Una vez especificados estos datos, se deberá presionar el botón de confirmar cuidadores. Los cuidadores seleccionados por el transeúnte serán notificados y deberán aceptar o rechazar el cuidado.
4. Si al menos un cuidador acepta la responsabilidad durante el trayecto, al transeúnte se le habilitará el botón de "comenzar". Al ser presionado este botón, el sistema deberá calcular el tiempo de demora aproximado y volverle a notificar a sus cuidadores. La distancia (en metros) entre dos direcciones será calculada por "Distance Matrix API" de Google, cuyo sistema nos brinda una interface REST.
5. Durante todo el recorrido, el sistema no deberá enviar notificaciones al transeúnte (por motivos de seguridad), ya que el mismo estará en movimiento.
6. Una vez que el transeúnte llegue a su destino, deberá presionar el botón "llegué bien!". El sistema deberá volver a habilitar las notificaciones, ya que se considera que no hay peligro alguno, y se deberá volver a notificar a sus cuidadores con esta situación.
7. Si algo malo sucede, el sistema deberá darse cuenta de esta situación por el tiempo aproximado que calculó. El mismo va a reaccionar frente a este incidente según lo que haya configurado el usuario:
 - Enviar un mensaje de alerta a sus cuidadores

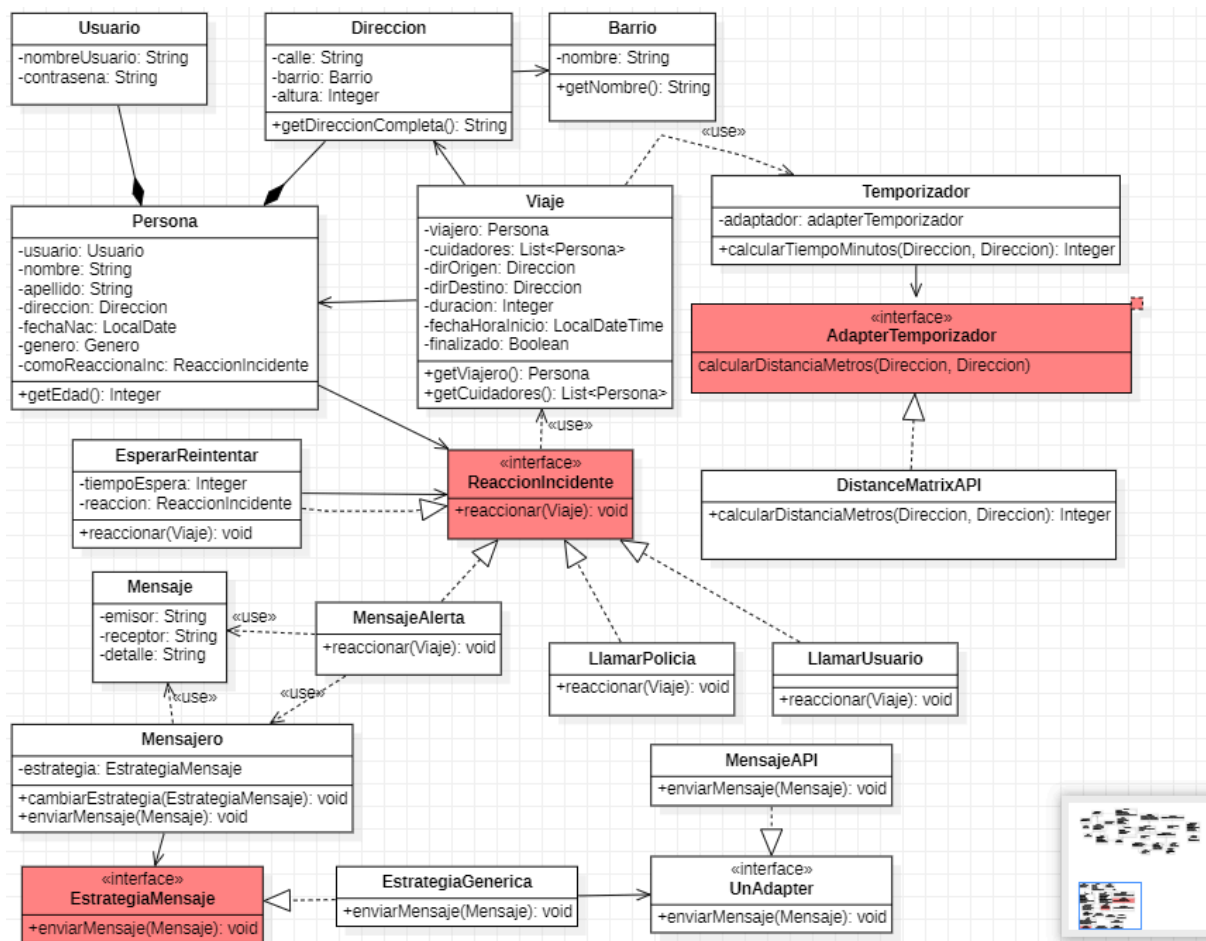
- Realizar una llamada automática a la policía
- Realizar una llamada al celular del usuario
- Esperar N minutos para ver si es una falsa alarma (los minutos deben ser parametrizables)

Se debe considerar que pueden surgir nuevas formas de reaccionar frente a un incidente y que el usuario puede cambiar esta configuración cuantas veces quiera.

Punto 2 - Modelo de Dominio (40 puntos)

1. Modelar el dominio presentado utilizando el paradigma orientado a objetos, comunicando su solución mediante un diagrama de clases. Si utiliza patrones de diseño, indíquelos y justifique su uso. NOTA: Puede ayudarse para comunicar, además, con código, pseudo-código, prosa u otros diagramas (diagrama de secuencia, de estados, entre otros). (25 puntos)

Diagrama de dominio:



Justificaciones y decisiones de diseño:

- Clase Persona: Se opta por una sola clase para representar a las personas ya que si hacía una para el viajero y otra para el cuidador, en runtime, un viajero no podría ser cuidador y viceversa.
- EstrategiaMensaje: Se opta por un patrón strategy para dar extensibilidad y un adapter para que sea mas flexible en cuanto a la API que se usa. Así logro que el envío de mensajes me quede desacoplado del resto del sistema.
- Temporizador: Como el DistanceMatrixAPI solo me da la distancia en metros, tuve que poner una clase para calcular el tiempo en recorrer esa distancia. Con esto se me ocurre usar un valor promedio de velocidad o en un futuro agregar una funcionalidad para calcular la velocidad promedio de cada persona por cualquier motivo (discapacidad u otros).
- ReaccionIncidente: Se modela una clase por variante que implementan esta interfaz, esto me da la extensibilidad que requería el contexto. La persona elige una forma de reaccion y va a reaccionar así en todos los viajes, si quiere cambiar la forma en que reacciona se puede hacer un método, me pareció engorroso del punto de vista del usuario tener que elegir una forma para cada viaje.
- Direccion: Como es pedido del gobierno de la ciudad, se supone alcance CABA y la dirección consta de calle altura y el barrio.