



Apellido y Nombre: Legajo:

Cuidándonos

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 1 – Arquitectura (35 puntos)

1. Para el envío de mensajes de alerta a los cuidadores se propusieron dos alternativas:
 - a) **Cola de mensajes:** el transeúnte tomará el papel de “publish”, quien dejará un mensaje en el topic “cuidandonos/{id_transeunte}/{id_viaje}”; y los cuidadores tomarán el papel de “subscriber”. En particular, la implementación se hará mediante Firebase (software de uso libre de Google), quien nos obliga a implementar el método “onMessageReceive(Message m)” en el código que correrá sobre los smartphones.
 - b) **Proceso propio:** proceso que correrá en el código de los smartphones, el cual hace una llamada REST al servidor central cada N segundos para que el mismo verifique si se produjeron cambios en la base de datos sobre el viaje del transeúnte cuidado.



Compare ambas soluciones con respecto a la **mantenibilidad** y **disponibilidad**.

2.

a) Siendo Cuidándonos una aplicación que se utilizará desde smartphones entonces (seleccione las opciones que cree que son correctas):

- ☐ La aplicación deberá ser nativa en su totalidad.
- ☐ La aplicación deberá tener una capa de visualización que corra sobre el sistema operativo del smartphone y la lógica de negocio implementada en un servidor en la nube.
- ☐ El cálculo de la distancia deberá hacerla la propia aplicación.
- ☐ El cálculo de la distancia deberá ser delegada a un componente de terceros (tal cual lo presenta el dominio actualmente).
- ☐ El dominio podría ser implementado en una base de datos no relacional
- ☐ La aplicación podría tener persistencia polígota

b) Justifique sus elecciones del ítem anterior según el atributo de calidad **performance** y cualquiera de los siguientes que elija que crea que aplican al dominio:

- | | |
|--|-------------------------------------|
| <input type="checkbox"/> Seguridad | <input type="checkbox"/> Usabilidad |
| <input type="checkbox"/> Portabilidad | <input type="checkbox"/> Eficiencia |
| <input type="checkbox"/> Funcionalidad | <input type="checkbox"/> Madurez |

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)**

2. Ahora un transeúnte también podrá escoger un destino con varias paradas; esto es:

Posición actual -> primer destino -> segundo destino -> ... -> destino final.

Para esto, se deberá especificar la dirección exacta de cada destino y el orden en el que se recorrerán. Además, el usuario deberá especificar si se detendrá N minutos en cada parada, o si irá avisando punto a punto su estado de "salud" (si llegó bien).

Si se especifica que se va a detener en cada parada, entonces el sistema deberá ir calculando las demoras aproximadas por secciones (demora de A->B, demora B->C, etc.); caso contrario, se deberá hacer un cálculo aproximado total.

Extienda su solución para que soporte este nuevo requerimiento. Además, muestre mediante código o pseudocódigo cómo implementaría el cálculo de demora aproximado. **(15 puntos)**

Punto 3 – Persistencia (25 puntos)

Utilizando un DER, explique cómo persistirá el modelo del punto anterior (dominio) indicando claramente:

- Qué elementos del modelo es necesario persistir.
- Las claves primarias, las foráneas y las restricciones según corresponda.
- Si fueran necesarias, estrategias de mapeo de herencia utilizadas. Explicar por qué fueron elegidas y compararlas con alguna otra.
- Justificaciones sobre las decisiones de diseño tomadas anteriormente.

NOTA: Explicar supuestos y justificar decisiones de diseño.

Condiciones de aprobación: Para aprobar debe sumar como mínimo 60 puntos y no menos del 50 % en cada sección.