

## **Justificaciones de diseño:**

1. En primera instancia utilizamos el patrón state para poder modelar a las personas que interactúan con la aplicación, puesto que contemplamos que un “cuidador” pueda ser un “transeúnte” y viceversa. Ambos interactúan de manera diferente dependiendo del rol que tengan asignado al momento de usar la aplicación y nada impide que al terminar de realizar una actividad no realice otra.
2. Se decidió la creación de una clase “Viaje” que englobe a todo aquello relacionado a su uso específico, de esta manera se puede disponer de los datos necesarios para los cálculos que se deban realizar y se pueda gestionar a las personas que interactúan sin sobrecargar a las mismas clases. Además de permitir que clases cuyos tiempos de vida dependan del propio “viaje” puedan interactuar y ver su utilización finalizada al terminar.
3. Se le asignó un método “preferencias” al transeúnte para que este pueda modificar y/o pasar datos necesarios a futuro a las diferentes “formas de respuesta” ante alguna situación.
4. Para los cálculos de las distancias se optó por el patrón adapter puesto que no se especifica ni se conoce como ha de interactuar la API de Google con la aplicación, pero sí sabemos los parámetros que en teoría debería recibir para funcionar. Por ello delegamos el comportamiento al adapter, aumentando la cohesión a futuro y le proporciona mayor mantenibilidad a nuestra solución puesto que se produce desacoplamiento de funciones entre la aplicación (el cliente) y la API (clase adaptada).
5. Para las formas de respuesta, optamos por el patrón strategy, pues el propio “transeúnte” debe poder modificar su funcionamiento pero a la vez entendiendo que existen muchas maneras de resolver la misma situación (no llegar a destino en el tiempo estipulado) por lo que debe poder optar por las diferentes alternativas que configuró en el momento. Este patrón de diseño aporta la flexibilidad a la hora de agregar nuevas respuestas posibles a futuro, junto a la mantenibilidad de las ya existentes formas de resolver los problemas.