

TP Cuidándonos

Alumnas	<ul style="list-style-type: none">• Jimena Cori• Melani R Lorenzo
Curso k3001 k3101	Martes - Mañana
Fecha entrega	07/05/2024

Punto 2 - Modelo de Dominio

1)

Se utilizaron adapters para calcular la distancia para priorizar la mantenibilidad y extensibilidad del código en caso que en un futuro la API para calcular distancias cambie. En el caso del notificador push, es útil para abstraer y encapsular la lógica de envío de notificaciones push a través de diferentes proveedores de servicios de mensajería, lo que haría el sistema ser compatible con múltiples plataformas de notificación push.

Para modelar las reacciones se pedía considerar que pueden surgir nuevas formas de reaccionar frente a un incidente y que el usuario puede cambiar esta configuración cuantas veces quiera. Entonces por eso optamos por un patrón strategy con interfaz para asegurar extensibilidad si se necesitan incorporar nuevas formas.

2)

```
public class Trayecto {  
    ...  
    public double calcularTiempoDemoraAproxIncluyendoParadas(calculador:  
CalculadorDeTiempoDemora){  
        double demoraEnMinutos;  
  
        if(paradas.all(parada -> parada.seDemora()))  
            demoraEnMinutos = paradas.sum(parada ->  
parada.calcularTiempoDemoraAprox(calculador);  
        else  
            demoraEnMinutos = this.calcularTiempoDemoraAprox(calculador);  
  
        return demoraEnMinutos;  
    }  
}
```

```
public class Parada {  
    ...  
    public double calcularTiempoDemoraAprox(calculador:  
CalculadorDeTiempoDemora){  
        double demoraEnMinutos;
```

```
        demoraEnMinutos = calculador
            .calculadorDeMinutosDemora(this.origen.obtenerDireccion(),
            this.destino.obtenerDireccion())
            + this.demoraMinutos;

    return demoraEnMinutos;
}
}
```