

2. Explicación del Flujo

A. Cómo el sistema decide qué canal usar

1. El cliente del sistema invoca:

```
NotificationService.SendByEvent(eventName, notification)
```

2. El NotificationService consulta su tabla de mapeos:

```
eventToChannelModel[eventName]
```

3. Por cada canal configurado, el NotificationService consulta su registry.

```
registry[channelName]
```

4. Si el evento tiene varios canales, el NotificationService crea un CompositeNotifier y agrega todas las estrategias encontradas.

5. Finalmente, ejecuta:

```
notifier.Send(notification)
```

B. Cómo se agregaría un nuevo canal sin romper el código

1. Crear una nueva clase que implemente INotifier.

2. Registrar el canal en el NotificationService con Register.

3. Mapear un evento al nuevo canal con MapEventToChannel.

El sistema no requiere modificar código existente, cumpliendo el principio Open/Closed.