Uno de los patrones de diseño vistos en clase es el patrón *Observer*, el cual le permite a una clase *Subject* notificar cambios a una clase *Observer* sin necesidad de aumentar el nivel de acoplamiento entre ambas. Sin embargo, una debilidad de la implementación vista en clases de este patrón es que, para realizar notificaciones, es necesario agregar a la clase *Subject* un atributo adicional a su definición que contenga todos los *Observers* que tendría que notificar. Se podría argumentar que lo anterior no debiese ser responsabilidad de la clase *Subject*, por lo que dicha implementación disminuiría levemente su cohesión.

Una implementación alternativa del patrón *Observer* provista por algunas plataformas es la posibilidad de canalizar todas las observaciones y notificaciones a través de una clase única que llamaremos *NotificationCenter*. Esta clase define los siguientes tres métodos:

- `register_listener(event_id, listener)`
- `delete_listener(event_id, listener)`
- `notify(event_id, *arguments)`

El parámetro `event_id` es un identificador (que en Ruby puede representarse por un símbolo) que permite identificar un evento específico. Al invocar el método `notify` del *NotificationCenter*, se invoca subsecuentemente el método `update(…)` de todos los *Observers* que se hallan registrado con el respectivo `event_id`. Para que esto funcione, dichos *Observers* deben efectivamente implementar ese método. Con lo anterior en mente, el flujo para observar eventos y notificar cambios es el siguiente:

- Observer implementa método update
- Observer consigue una referencia al NotificationCenter
- Observer se registra como listener para un evento X
- Subject realiza una operación de interés
- Subject consigue una referencia al NotificationCenter
- Subject invoca el método notify para el evento X
- NotificationCenter llama al método update de los Observers registrados con el evento X

Escriba una implementación para la clase NotificationCenter aquí descrita en Ruby e ilustre su funcionamiento escribiendo un breve script de ejemplo que reproduzca el flujo anteriormente descrito. (6 puntos)