Trabajo practico final

A la caza de las vinchucas 2022

Integrantes

Nicolas Cervino

Lautaro Martin Villagra

Emails de Contacto

nicolascervino2@gmail.com

lautarovillagram@gmail.com

-Actualizador de categorias

Para esta solucion decidimos utilizar la estrategia de observer y observable, donde el observable son los usuarios regulares(ya que son los unicos que pueden cambiar de categoria, los usuarios expertos tienen una categoria fija) y el observer es la clase ActualizadorDeCategoria. Cuando el usuario regular realiza una opinion o agrega una muestra, se ejecuta la funcion actualizar en el observable, que verifica las muestras enviadas y las opiniones enviadas del usuario y actualiza su categoria dependiendo del resultado de las mismas, esto se realiza automaticamente y no es necesario hacerlo manualmente.

-Opinar las Muestras

Para agregar opiniones a la muestra, primero se verifica que el usuario puede opinar en esta muestra y que el parametro String utilizado para indicar la especie pertenece a una de las categorias asignadas, esto se realiza creando una variable local de la clase Especies, donde la misma incluye una lista con todas las categorias que se pueden utilizar (hasta el momento).
La logica de las opiniones aplica la estrategia template method pattern, donde puedeOpinarEn y opinar son abstractas y se definen en las subclases que lo utilizan, ya que tanto el usuarioRegular como el usuarioExperto tienen diferentes comportamientos en estas funciones. Esto permite que, en el caso de tener que modificar los comportamientos de las mismas en una subclase particular, no sea necesario modificar el codigo en la otra u otras subclases que se deseen agregar en un futuro.

-Organizaciones y Zonas de Cobertura

Esta solución también implementa el patrón observador, sin embargo, en este caso se requiere de updates específicos que le indiquen a una organización que fue lo que cambio en una ZonaDeCobertura.
Las organizaciones observan a las zonas de cobertura para saber si las mismas reciben una nueva muestra o una de sus muestras se verifica, para eso implementan una interfaz OrganizacionObserver que posee 2 updates para estos casos.
Además, se inicializan con 2 Funcionalidades Externas que se utilizan según el tipo de update que se vaya a realizar, pero las mismas pueden ser modificadas mediante un método setter.
Las zonas de cobertura son subclases de una clase abstracta ZonaObservable. Esta clase abstracta conoce a todos sus observadores y define los métodos para añadir y eliminar a los mismos, así como también un método notify que actúa según un String que recibe por parámetro. De esta forma, la zona de cobertura no necesita definir estos métodos que realmente no forman parte de la lógica de la misma.
Un detalle que falto implementar para terminar de automatizar este proceso es que cuando una muestra se crea, si su ubicación pertenece a alguna zonaDeCobertura, la

misma se agregue a la misma, lo mismo que cuando la muestra se verifica. Sin embargo, para que esto sea posible sería necesaria otra estructura que conozca a todas las zonas de cobertura.

-Buscador

En el buscador no pudimos encontrar ningún patrón de diseño que se adecuara, se pensó en un principio en aplicar un patrón Strategy pero al final se descartó y se optó por implementarlo de manera convencional. Un buscador conoce a una lista de muestras, y tiene métodos para aplicar los distintos filtros de búsqueda. Además, posee 2 métodos que reciben el resultado de 2 de estas búsquedas y permiten combinarlos con operadores lógicos AND y OR respectivamente.