



Universidad
Nacional
de Quilmes

Informe del trabajo Práctico 'A la caza de las vinchucas'

Profesores:

- * Diego Cano
- * Diego Torres
- * Matías Butti

Integrantes:

- * Salas Marcos Nahuel
- * Villarroel Julián Esteban
- * Justiniano Sofía Agustina

E-mails:

nahuop@gmail.com

villarroel.julian.esteban@gmail.com

sofia.agustina.justiniano@gmail.com

12/06/2022

Decisiones de diseño:

Diseño de **tipoDeOrganizacion** y **Calificacion**:

- Para la implementación de los tipos de Organización y la Calificación de la muestra, utilizamos Enumerativos, inicialmente habíamos utilizado String. Por lo tanto, para la opinión utilizamos el enumerativo Calificación.

Diseño de la **Muestra**:

- Muestra tendrá un atributo que representa a la “fotoVinchuca”. Inicialmente fue pensada como un String, pero luego nos percatamos que podría ser tipada como un BufferedImage para poder representarla.
- La Muestra siempre podrá conocer su estado actual, donde siempre que una muestra sea instanciada, por default su Estado será ser una muestra votada.
- Contendrá, además, un historial de opiniones, en la cual cada opinión estará asociada a un usuario en cuestión.
- Podrá conocer en todo momento su resultado actual, obteniendo la opinión con más votos adquiridos de acuerdo al tipo de Calificación que obtuvo dicha muestra.
- Podrá obtener las muestras cercanas entre una muestra y otra, dependiendo del kilometraje pretendido. De forma implementativa, esto se lo delegará a la ubicación que conoce la muestra en cuestión.

Diseño de la **Ubicación**:

- Se utilizó la fórmula de Haversine extraída de la web, para resolver la distancia entre dos ubicaciones geográficas conociendo su latitud y longitud.
- Obtiene su distancia respecto de otra ubicación.
- Obtiene las muestras cercanas de una muestra.

Diseño de la **AplicaciónWeb**:

- La aplicación web fue lo que casi a último momento decidimos implementar, ya que desde la misma deben de pasar todas (o la mayoría), de las funcionalidades que tienen las muestras, ya anteriormente implementadas y testeadas.

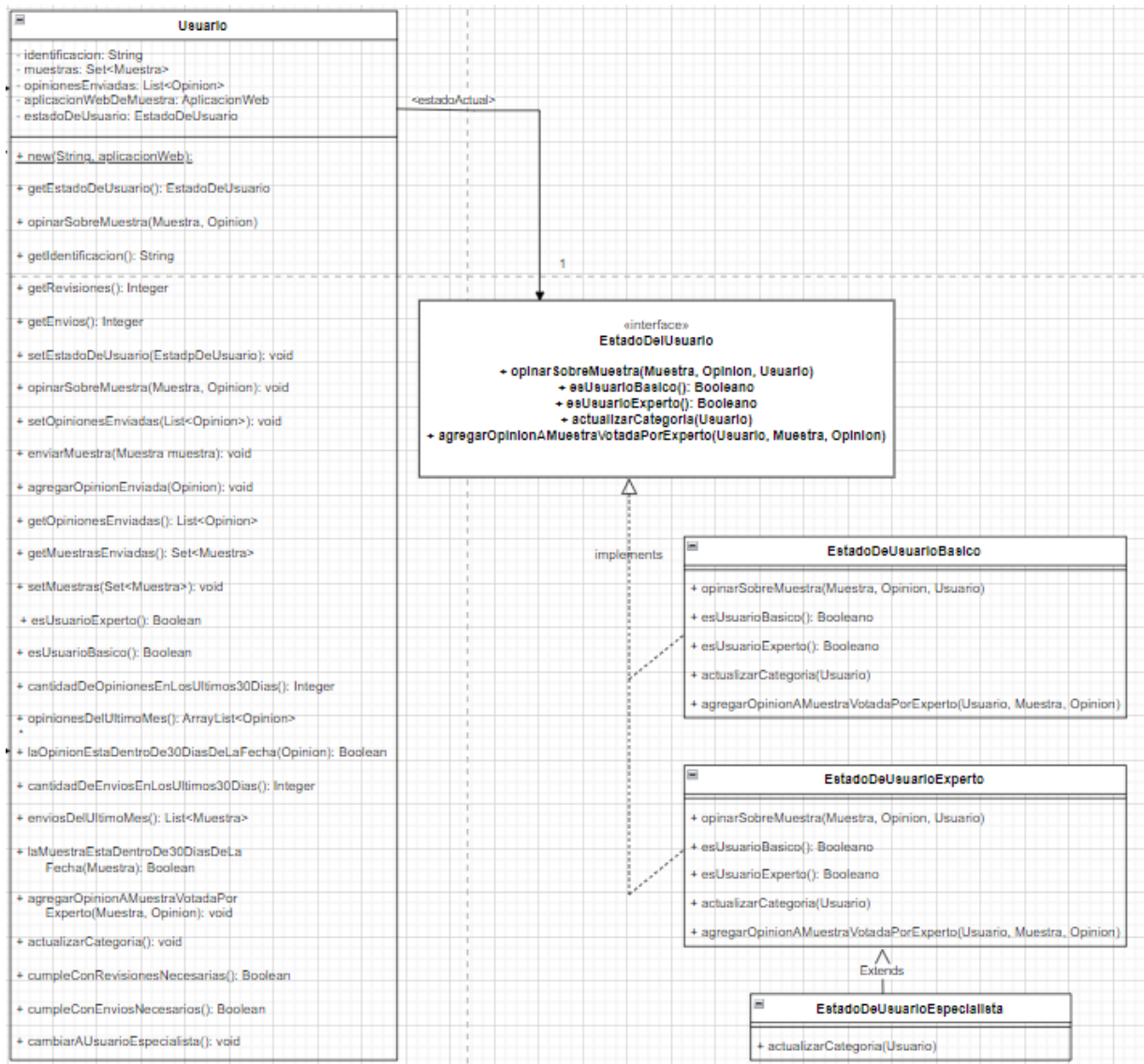
Patrones de diseño utilizados:

Para mejor visualización de las imágenes, las mismas están descritas en el UML.

Design Patterns: State

Para el desarrollo de los Estados del usuario(Básico, Experto y Especialista) y Estados de muestra(verificada, votada, votada por experto) utilizamos el patrón de diseño State.

Roles del estado del usuario:



Usuario: Context.

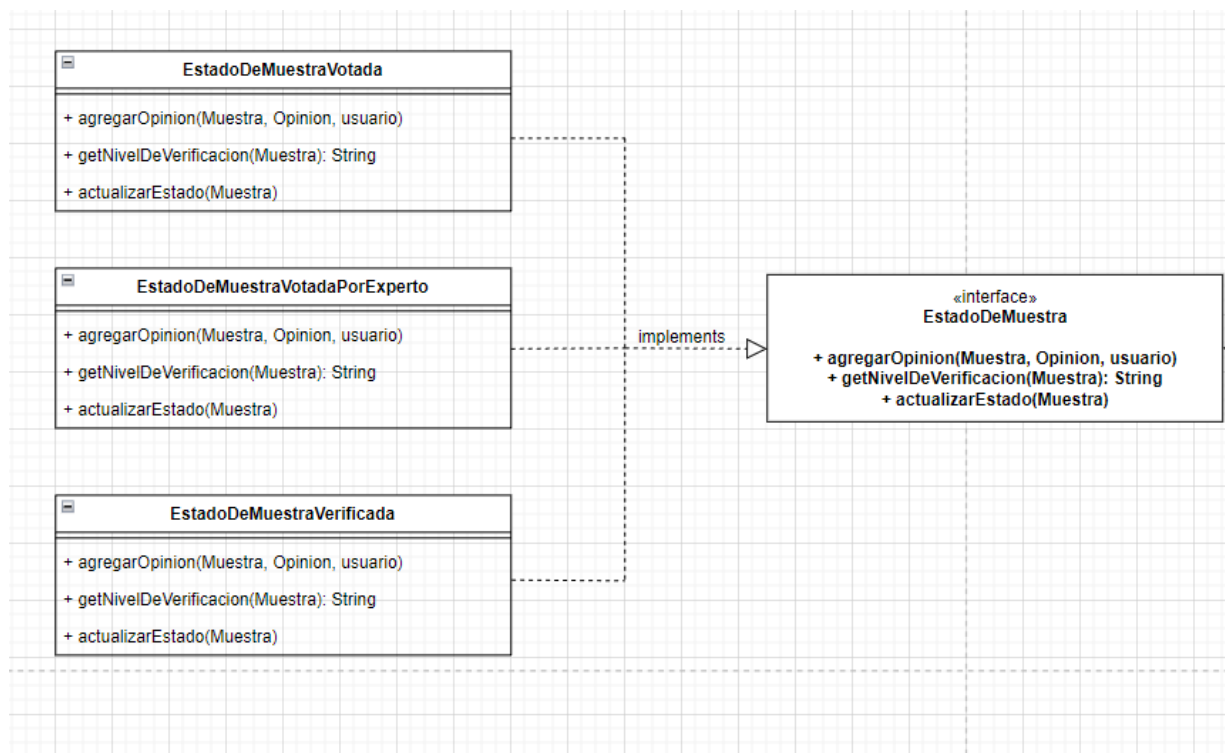
EstadoDelUsuario: Cumple el rol de State.

EstadoDeUsuarioBasico: Cumple el rol de ConcreteState.

EstadoDeUsuarioExperto: Cumple el rol de ConcreteState.

EstadoDeUsuarioEspecialista: Cumple el rol de ConcreteState.

Roles de los estados de muestras:



Muestra: Context.

EstadoDeMuestra: Cumple el rol de State.

EstadoDeMuestraVotada: Cumple el rol de ConcreteState.

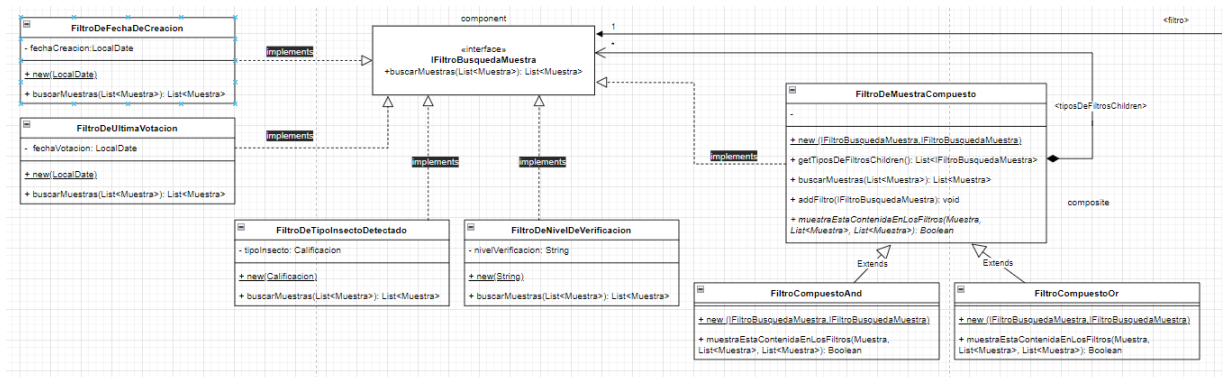
EstadoDeMuestraVotadaPorExperto: Cumple el rol de ConcreteState.

EstadoDeMuestraVerificada: Cumple el rol de ConcreteState.

Design Patterns: Composite y Template

Para la implementación del filtro de búsqueda, identificamos el patrón Composite, en el cual una muestra puede ser filtrada con un tipo de filtro o compuesto de más de un

filtro. Además, dentro de éste diseño hemos podido identificar el patrón Template Method a la hora de componer a más de un filtro.



FiltroDeMuestraCompuesto: Cumple el rol del composite

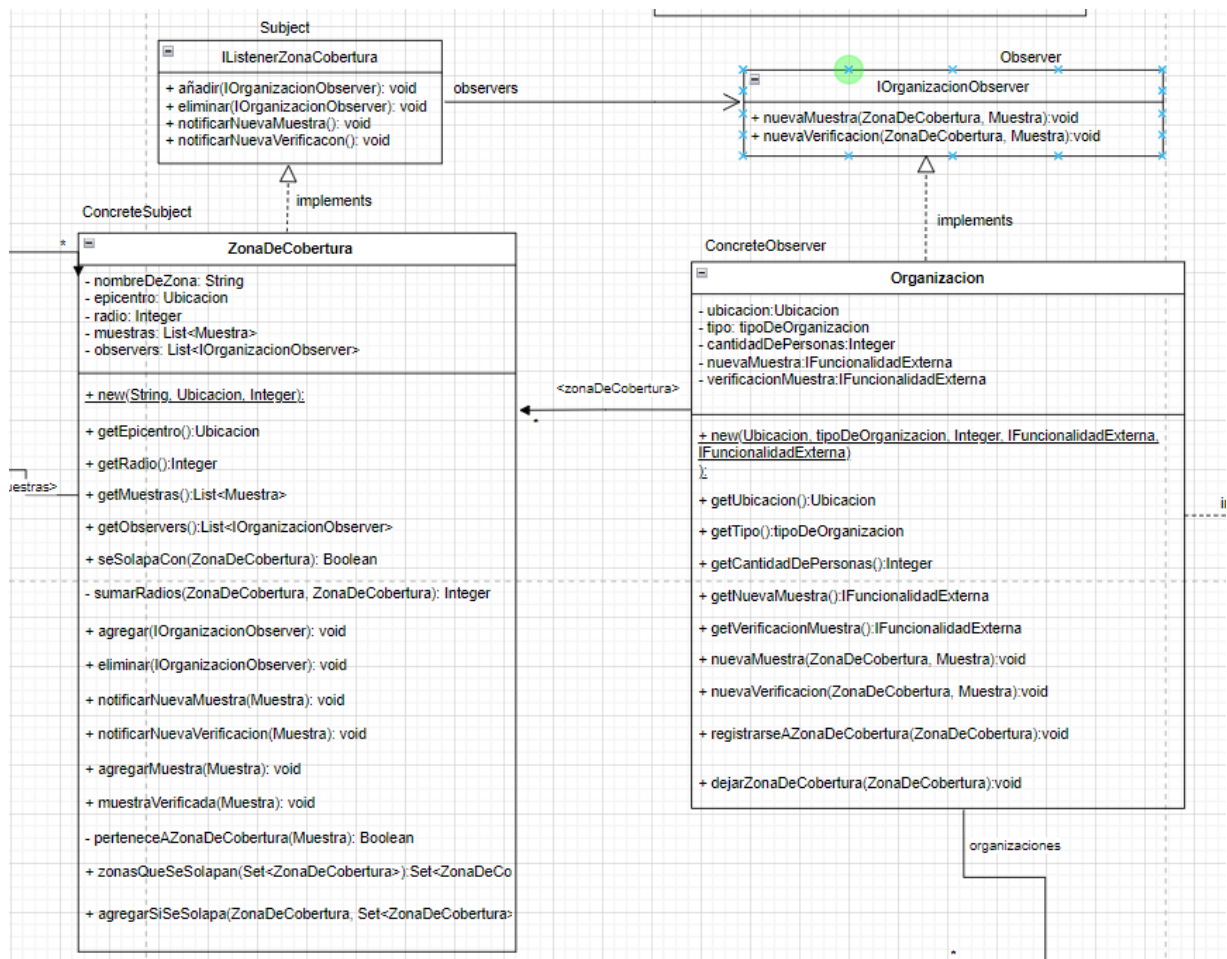
FiltroCompuestoAnd, FiltroCompuestoOr: ambos extienden del composite y, a la vez, están conformando el patrón de diseño Template Method.

IFiltroDeBusquedaMuestra: Cumple el rol component.

FiltroDeFechaDeCreacion, FiltroDeUltimaVotacion, FiltroDeTipoDeInsectoDetectado, FiltroDeNivelDeVerificacion: Cumplen con el rol de leaf indicado en el libro Gamma et. al.

Design Patterns: Observer

Para la parte de la zona de cobertura y organización implementamos el patrón de diseño Observer, para que las organizaciones puedan suscribirse y desuscribirse a las zonas de cobertura y las mismas puedan ser notificadas cuando suben y/o se verifica una muestra.



ZonaDeCobertura: Cumple el rol ConcreteSubject. El mismo envía la notificación a sus observadores cuando cambia su estado(se agrega y/o verifica una muestra en la zona).

IListenerZonaCobertura: Cumple el rol Subject.

Organizacion: Cumple con el rol ConcreteObserver. Implementa la interfaz de actualización para mantener su estado consistente con el del sujeto

IOrganizacionObserver: Cumple con el rol Observer. Define la interfaz de actualización para los observadores.