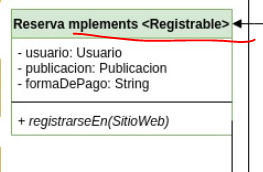
**Programación con Objetos II - TP Final 1er. Sem 2021**

Grupo Objetosos:

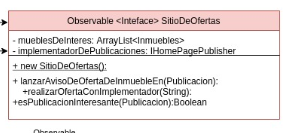
* Sofía Justiniano
* Rodrigo Caceres Jaramillo
* Bianca Osores

Devolución:

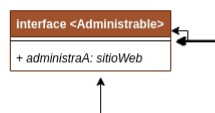
* No usar “mis” en los nombres de mensajes (ej: “misReservas”) ya que no cumple la nomenclatura estandard y además es confuso (quien envía el mensaje no pide “mis” reservas sino “sus” reservas; y usando “sus” de daría la confusión inversa).
* A su vez los nombres de mensajes pueden ser más cortos. Ej: “elUsuarioPuedeCancelarReserva” puede reducirse a “cancelable” o “isCancellable” (se puede asumir que es por parte del usuario).
* Política de Cancelación es un Strategy, esto es correcto.
* PolíticaDeCancelación tiene como atributos al usuario y la reserva que se desea cancelar. No hace falta que los conozca de forma contínua (atributos) sino que se les pueden pasar como parámetros en el momento de cancelación.
* No mezclar notación UML con sintaxis Java:



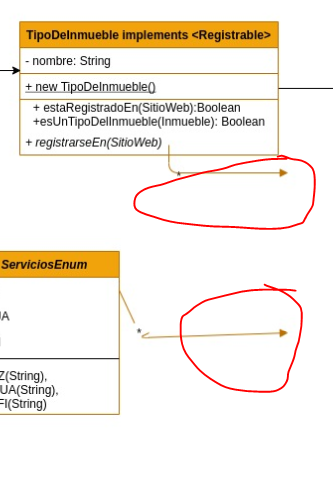
* Nombres de clases como este no cumplen las pautas mínimas de la notación UML (repasar UML):



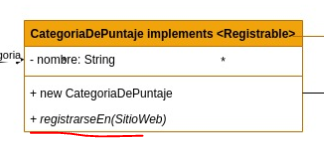
* Allí hay además un error conceptual: una interface no debe tener atributos (sé que Java lo permite, pero en otro contexto y con otro sentido).
* Similar problema en notación UML. En este caso, además, la interfaz no tiene sentido ¿Cuál es el aspecto que abstrae y para qué?:



* Las siguientes flechas no quedan apuntando a nada. Y tampoco tienen nombre. Se desconoce su destino:

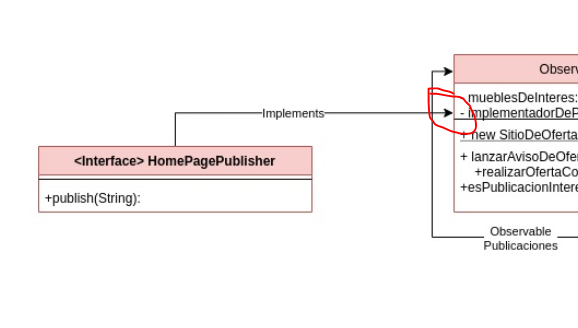


* Poner nombres de relaciones que definan directamente a la clase destino o, mejor aún, el rol que ella juega para quien la conoce. “conFormaDePago” debería ser “formaDePago”, mientras que “esDeCategoría” podría ser simplemente “categoría”. Funcionalmente no cambia nada, pero es importante manejar una nomenclatura standard para mejorar la legibilidad cuando se trabaja en equipo con otras personas (y en este caso, los docentes también somos lectores del código).
* La relación “realizaUnaBúsquedaDeInmuebleCon”, es otro ejemplo en este sentido.
* En la misma línea, el mensaje “elInmuebleEstáOcupado(inmueble)” puede reducirse a estáOcupado(inmueble). El parámetro también ayuda a definir de qué estoy hablando, lo cual me permite acortar el nombre del mensaje.
* En el modelo hay muchos “Administradores”: de Publicación, de Usuario, de Reserva, de OcupaciónDeInmueble, e incluso un “Administrador” general, además del SitioWeb. Se debe intentar no tener esos administradores y distribuir esas responsabilidades en las clases del modelo que representan conceptos propios del dominio del problema.
* La búsqueda se define con una única clase. Podría usar una composición para definir primeros los parámetros básicos y luego poder sumar otros. La clase ParametrosDeBusqueda es totalmente estática (sólo contiene información). Debería contener comportamiento que le permita colaborar en la ejecución de la búsqueda (lo cual, as su vez, obliga a rever su nombre).
* El el sitio el que registra a la categoría, no es necesario pedirle a ella que se registre. No es su responsabilidad:



* No tiene sentido crear una interface cuyo único sentido sea registrarse en el sitio web:



* Mucho del comportamiento de las reservas depende de su estado, y esto no se está reflejando mediante el patrón de diseño correspondiente.
* No se gestionan reservas condicionales.
* El patrón observer propuesto no tiene un nivel suficiente de abstracción, y tampoco aparece el rol de Observer en ninguna parte del modelo.
* Una interface nunca implementa una clase, sino que es al revés. La flecha utilizada para el “implements” no es la correcta según UML, y finalmente la clase Observer no debería implementar una interfaz del dominio del problema (si esa fue la intención):
* 

**Comentarios Generales:**

El modelo entregado contiene errores de mucha consideración y no resuelve mínimamente el problema del enunciado según los principios de diseño OO presentados en la materia. Debido a esto, no es posible hacer una revisión del código Java entregado.

La cátedra considera la ausencia temporal de uno de los miembros del equipo por COVID-19 y les propone una nueva instancia de entrega.

**Nota del Trabajo**: Reentregar.