

Hola, estuve mirando su entrega de laboratorio. En general lo que entregaron está bien. Van comentarios de los diagramas de comunicación de cuatro casos de uso principales como muestra de lo que encontré. También comento aspectos del DCD, los patrones y los criterios GRASP aplicados. No es necesario que reentreguen nada, solo tengan en cuenta los siguientes comentarios para la implementación.

- Alta Usuario: En `confirmarAlta()` deberían indicar con una nota que se obtiene el `DataUsuario` y, además, deberían definir qué tipo de usuario se tiene que crear. El `create()` no debería ir a `Usuario`, sino a `Empleado` o `Huésped`, según lo que sea que se debe construir. Lo que muestran como dos diagramas se hace para el caso de polimorfismo, cosa que no existe en este caso ya que se debe crear una instancia concreta y no invocar un comportamiento polimórfico.
- Realizar Reserva:
  - En `confirmarReserva()`, `dtReserva.esGrupal()` no es un mensaje al Controlador, sino un acceso al datatype, eso es lo que debería aparecer en la guarda en lugar de `[esGrupal]`
  - ¿Qué sucede después de los `create()`? Porque la reserva se asocia con huéspedes y una habitación, así que deberían mostrar la creación de links con esos elementos, ya que no es solo inicializar atributos primitivos.
- Registrar Estadía:
  - Al iniciar un diagrama de comunicación, se reinicia la numeración. En los diagramas de `esReservaHostalHuesped` deberían comenzar de cero y no seguir la numeración del diagrama anterior.
  - Al crear un objeto como Estadía se pueden pasar parámetros en el `create` y no necesariamente usar otras operaciones `set` para inicializar la información.
  - Ojo que en `agregarEstadiaAHuesped`, le están haciendo `agregarEstadia` a `hostal` cuando debería ser `huésped`
- Baja de Reserva:
  - En `seleccionarReservaAEliminar()` si lo que almacenan es el código, entonces no es necesario buscar la reserva. Además, si recuerdan el código, en `confirmarBaja()` deberían buscar la reserva y no asumir que ya la tienen. Estos dos diagramas les quedaron inconsistentes.
  - En `confirmarBaja()` no quedó bien la numeración. El `destroy()` a Estadía debería ser 1.2 y no 2. Además, ¿qué sucede en esa destrucción? Lo que está en `notificarEstadia()` podría ser parte del destructor y no ser una operación aparte. En esa misma operación, ¿cómo que la estadía le habla a otra estadía para que remueva una estadía de su colección de estadías? Algo quedó mal, creo que quisieron hablar del `ControladorEstadias`.
  - Al eliminar algo hay que eliminar todo link que otras cosas tengan con ese algo. Al `confirmarBaja()` hacen `destroy` en `Calificación`, pero no eliminan los Comentarios asociados, ni desvinculan los Empleados que conocen los comentarios ni los Hostal que conocen las calificaciones. Si fueran a implementar esto, dejarían muchísima memoria colgada y links apuntando a nada.

- Otros: En consultar top 3 a la estadía le están pidiendo la calificación para luego pedirle a la calificación su DT. Acá están hablando con extraños ya que si lo que necesitan es solo el DT, podrían pedirle directamente a la estadía que devolviera el DT de la calificación y así eliminar la dependencia entre ControladorEstadias y Calificación, al menos en este diagrama. Además, si DTHostal tiene un DTHabitacion y éste, a su vez, tiene un DTEstadía que tiene un DTCalificación, entonces podrían iterar sobre los hostales y a partir de ello calcular el promedio de cada uno. De lo contrario, no queda claro cómo hacen para vincular los DTCalificación con los hostales si el DTCalificación solo tiene el puntaje, comentario y fecha (según su DCD).
- DCD: Sin entrar en detalles, el DCD aparenta estar bien elaborado. Un comentario es que no es necesario tener una fábrica por interfaz, alcanza con tener una fábrica para todas las interfaces.
- Patrones: La aplicación de los patrones es correcta. Una observación sobre Observer: el concepto Notificación es un tanto ficticio y tendría sentido tenerlo si lo que quieren almacenar es un DT con información y no generar una dependencia con Calificación. Sin embargo, en el diagrama de comunicación de calificar, le pasan al Empleado la Calificación para generar Notificación, por lo que ya tiene esta dependencia y no parece necesario generar una nueva clase que agrega una clase nueva.

Saludos

--

Dr. Ing. Daniel Calegari García  
Profesor Agregado  
Instituto de Computación  
Facultad de Ingeniería  
Universidad de la República  
J. Herrera y Reissig 565  
Montevideo 11300, Uruguay

mail :: dcalegar [at] [fing.edu.uy](mailto:dcalegar@fing.edu.uy)  
web :: [www.fing.edu.uy/~dcalegar](http://www.fing.edu.uy/~dcalegar)  
fax :: (+ 598) 2711 0469  
tel :: (+ 598) 2711 4244 ext. 1125