

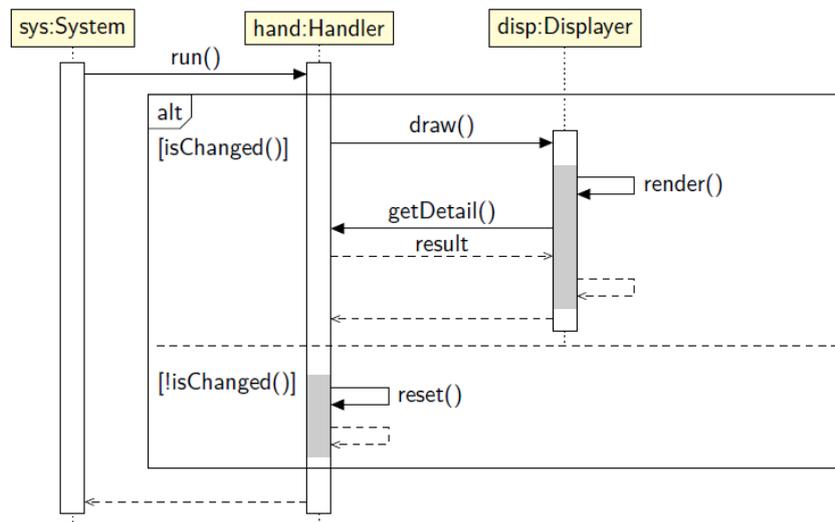
**Nombre:** \_\_\_\_\_ **Matrícula:** \_\_\_\_\_

**Sección A**

1. Identifique las técnicas para validación de requerimientos e ilústrelas por medio de un ejemplo aplicable a algún requerimiento no funcional. Indique la categoría de requerimiento no funcional de acuerdo con la clasificación de Sommerville. **[15%]**
2. Defina Ingeniería de Software. **[05%]**
3. Con base en el código de ética de ingeniería de software de la ACM, proponga un ejemplo que ilustre cada principio al que un ingeniero de software debe adherirse. **[16%]**

**Sección B**

4. Escriba el mínimo código fuente en Java para las clases *System*, *Handler* y *Displayer* que capture todo el comportamiento en el siguiente diagrama de secuencia UML. Los cuerpos de todos los métodos deberían ser dejados vacíos a menos que las llamadas sean indicadas en el diagrama. Los constructores y propiedades (es decir, métodos *get* y *set*) de las clases no necesitan ser mostrados. **[15%]**

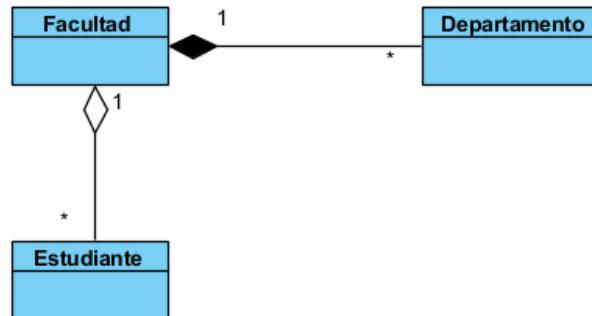


5. Se requiere un sistema para rastreo de errores en software de código abierto que pueda grabar reportes de errores y manejar los avances en su corrección. Escriba un diagrama de estado UML que capture los estados y transiciones para reportes de errores.

*El sistema puede ser usado por usuarios normales (es decir, el público en general) y desarrolladores, quienes son usuarios confiables. Reportes de errores entrantes son abiertos como No Confirmados si vienen de un usuario normal, pero directamente como Confirmados, si vienen desde un desarrollador. Un desarrollador puede hacerse propietario de un error Confirmado si nadie lo ha hecho aún. El propietario de un error puede liberar su propiedad en cualquier momento. Un error puede ser marcado como Resuelto por su propietario cuando el error ha sido corregido. Sin embargo, el error no puede ser marcado como Cerrado hasta que haya sido examinado y pasado un chequeo de calidad por un desarrollador diferente al propietario. Si falla el chequeo de calidad, es devuelto al propietario para que se encargue de ello. Después que un error ha sido Cerrado, cualquier usuario puede reabrirlo si piensa que la solución no estuvo completa, aunque*

solo los desarrolladores pueden reabrirlo como un error Confirmado. [5 puntos por los estados, 5 puntos por las transiciones, 5 puntos por los detalles de las transiciones.] [15%]

6. Escriba el mínimo código fuente en Java que satisfaga la representación en el diagrama de clases a continuación. [10%]



7. Considere un sistema para reserva de vuelos en línea en el que los usuarios pueden registrarse en un servicio para ver vuelos disponibles, reservar vuelos, ver y adquirir ofertas promocionales, reservar hoteles en el destino de vuelo, pagar en línea por reservas de vuelos/hoteles usando tarjetas de débito/crédito, ver historial de transacciones, actualizar sus perfiles, recibir notificaciones de reservas por e-mails y celulares, imprimir recibos para todas las transacciones efectuadas. El sistema está conectado a un consorcio representando a las compañías de tarjetas de débito/crédito para verificar tarjetas y manejar los pagos. El sistema está diseñado para tener una disponibilidad 24/7. El sistema puede lidiar hasta con 300 usuarios en línea durante las horas de poca actividad. Los perfiles de usuario, itinerarios y transacciones son almacenados y respaldados hasta en dos diferentes ubicaciones (Una en Ecuador y otra en Estados Unidos) por seguridad. Toda información transaccional pasada por el sistema para reserva de vuelos en línea al consorcio de tarjetas de crédito/débito es segura y está encriptada usando certificados SSL de 128 bits. El procesamiento de pagos en línea no debe exceder los 40 segundos.
- Especifique **cinco requerimientos funcionales** del sistema. Haga uso de **MoSCoW** en su especificación. [12%]
  - Especifique **cinco requerimientos no funcionales** del sistema. Especifique el tipo de requerimiento no funcional usando la categorización de Sommerville para requerimientos no funcionales. Haga uso de **MoSCoW** en su especificación. Describa el mecanismo mediante el cual **validaría** cada uno de ellos. [12%]