



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada
1^{er} semestre 2016

Actividad 08

Manejo de excepciones - Testing con unittest

Instrucciones

Cierto alumno de una universidad no tan lejana se ofreció para crear la página web **www.progralotodo.com** que consiste de un sistema de venta y compra online. Sin embargo, este alumno nunca aprendió manejo de excepciones y testing por lo que su programa no tiene ningún manejo de errores y se cae varias veces por diversos motivos :(

Debido a su alto prestigio entre los alumnos de IIC2233, se le ha pedido que modifique la clase Sistema y Usuario para que Progralotodo no lance excepciones. Además, debe ser capaz de diferenciar las distintas excepciones y corregir el código **una vez lanzada** de la excepción en vez de printear el motivo del error cuando sea necesario.

Además, con el fin de impresionar a los dueños de la página, usted decide implementar un programa para testear todas las funciones que se encuentran en el main.py, de tal forma que Progralotodo pueda asegurarse de que los métodos creados están funcionando de manera correcta.

Requerimientos

- Modificar los métodos de la clase Sistema y Usuario utilizando manejo de excepciones para que el programa no falle.
- Para cada error, usted debe diferenciar entre notificar al usuario indicando el tipo de error o bien **una vez lanzada** la excepción, solucionar el problema. Por ejemplo, si la excepción se genera porque se ingresa un rut inválido, se debe notificar al usuario que se generó una excepción de tipo X dado que se ingresó un rut inválido. Sin embargo, si se está realizando un cálculo y hay valores que no corresponden al tipo necesario para este, se debe corregir y utilizar los valores que sí son válidos para el cálculo correcto.
- No modificar el main.py desde la línea 166.
- Crear un nuevo archivo con un programa para testear las funciones de la clase Sistema y Usuario.
- Crear los métodos setUp y tearDown de manera de que al testear el programa, no se generen archivos innecesarios.

Notas

- En la sentencia `except` debe incluir el tipo de excepción. No se considerará correcto si no lo incluye.
- Si un usuario no ha sido calificado, su promedio de calificaciones es cero.
- El id de cada producto se genera al momento de instanciarlo. El primer producto instanciado tendrá un `id = 0`, el segundo un `id = 1` y así sucesivamente.
- Para el testeo del método `generar_orden_compra`, debe chequear que nombre de usuario y rut de la orden generada concuerde con el usuario que compró el producto y además, que el precio indicado en la orden corresponda al precio final pagado.

To - DO

- (2.50 pts) Agregue código de manejo de excepciones en las partes correspondientes del programa para manejar todas las excepciones provocadas por el main existente. Procure no repetir código innecesario.
- (0.80 pts) Por cada excepción capturada explique al usuario qué es lo que no se pudo hacer y por qué no se pudo o cuando sea pueda corregir, luego de lanzada la excepción corrija la falla.
- (1.00 pts) Método `SetUp` y `TearDown`.
- (0.20 pts) Test de login
- (0.20 pts) Test de `rut_valido`
- (0.20 pts) Test de `revisar_usuario`
- (0.30 pts) Test de búsqueda
- (0.30 pts) Test de `calificacion`
- (0.40 pts) Test de `vender_producto`
- (0.35 pts) Test de `comprar_producto`
- (0.75 pts) Test de `generar_orden_compra`

Tips

- Si un usuario califica a otro con algo distinto a un número, se debe calcular el promedio sin este tipo de calificaciones.
- Al momento de hacer los test, procure utilizar casos límite, como que al momento de comprar, el usuario tenga el dinero justo para adquirir el producto con el descuento ya aplicado.