LABORATORIO DE INGENIERÍA DE SOFTWARE

TAREA 3

OBJETIVOS GENERALES

Formalizar y ejercitar el conocimiento teórico-práctico de técnicas de programación por pares, desarrollo dirigido por pruebas (TDD) y su relación con la técnica de análisis de fronteras.

Introducir al estudiante a un *web application framework.*

ACTIVIDADES ESPECÍFICAS A REALIZAR

1) Programar utilizando las siguientes herramientas de apoyo del curso: un lenguaje de programación orientado a objeto (Python 3.x), un IDE (Eclipse/PyDev), una herramienta de automatización de pruebas unitarias (PyUnit), una herramienta de control de versiones (EGit).
2) Aplicar TDD, la técnica de análisis de fronteras y malicia para elaborar una suite de casos de prueba unitaria en PyUnit.
3) Instalar el *web application framework* Django en el ambiente de desarrollo integrado Eclipse de cada estudiante y familiarizarse con él.

ENUNCIADO

Desarrolle la funcionalidad básica de una billetera electrónica utilizando las técnicas de programación en pares, de desarrollo dirigido por casos de pruebas (TDD) y de análisis de fronteras enriquecido con malicia. Programe una clase "BilleteraElectronica" que contenga:

1) Un atributo *identificador*;

2) Atributos de datos personales del dueño (nombres, apellidos, CI) así como un número PIN (*Personal Identification Number*) que sirve como código secreto a la hora de autentificar la identidad del usuario. Recuerde que los nombres y apellidos pueden tener los caracteres especiales correspondientes al Castellano (acentos, diéresis, eñes, guiones…)

3) Una estructura que registre todos los créditos (recargas) relacionados a la billetera electrónica, donde se almacene el monto, fecha de la transacción y un id del establecimiento donde se realizó la recarga (puede ser un identificador numérico).

4) Una estructura que registre todos los débitos (consumos) relacionados a la billetera electrónica, donde de igual manera se almacene el monto, la fecha de la transacción y un id del local donde se realizó el consumo (igualmente puede ser un identificador numérico).

5) Un método *saldo* que devuelva el balance (saldo) actual de la billetera electrónica.

6) Un método *recargar* que permita, dados los datos de una recarga, registrar un crédito a la billetera electrónica.

7) Un método *consumir* que permita, dados los datos de un consumo, registrar un débito asociado a la billetera. Es importante destacar que son necesarias dos validaciones a la hora de registrar un consumo. En primer lugar, que el PIN ingresado por el usuario corresponde al PIN registrado, y en segundo lugar, que el cliente cuenta con balance suficiente para cubrir el coste de la operación.

Utilice la técnica de programación por pares y TDD. El conductor debe hacer un *commit* a su repositorio local Git al culminar un incremento de definición de pruebas, desarrollo y refactorización y se intercambia el rol con el navegador. Una vez culminado el TDD, agregue casos de prueba de frontera, esquinas y malicia, haciendo los *commit* correspondientes. Recuerde hacer *push* a su repositorio remoto Github cada vez que considere conveniente respaldar su trabajo y al culminar esta programación para que el profesor pueda accederla y corregirla. Ambos miembros del par deben turnarse los *push*. Etiquete cada caso de prueba como "caso interior", "caso frontera", "caso esquina", y "caso malicioso".

Hint: Identifique los dominios de este problema y sus fronteras.

Consideraciones Adicionales:

Los identificadores, tanto de las billeteras electrónicas, los puntos o establecimientos de recarga y los puntos de consumo no requieren ser validados.

En esta etapa del curso, no se abarcará la persistencia de datos, es decir, todas las funcionalidades que se desarrollen utilizarán objetos creados por su equipo y cuyo alcance se limitan a los casos de prueba definidos. La persistencia de dichos objetos en la base de datos, serán objeto del alcance en fases más avanzadas del curso.

ESPECIFICACIONES DE LA ENTREGA

La actividad debe realizarse en pares cumpliendo con las siguientes condiciones:

1) Ambos integrantes deben pertenecer al mismo equipo

2) Las parejas debe ser diferentes a las conformadas para la Tarea 2.

La entrega consistirá de los siguientes productos:

1) Acceso al repositorio Github de la tarea para que los profesores puedan clonar su trabajo de par en Eclipse, revisar su código fuente, compilarlo y ejecutar las suites correspondientes de prueba en Eclipse/PyDev.
2) Un informe que incluya el número de horas que trabajó cada miembro del par, cómo se dividieron las labores y un breve resumen de sus experiencias con las herramientas de apoyo. Casos de prueba y breve descripción del software.
3) Agregue, en un apéndice de su informe como constancia de haber instalado Django en cada máquina de trabajo un *printScreen* de la perspectiva PyDev de un proyecto Django cuyo nombre corresponda a la concatenación de los apellidos del par.
4) Los informes deben seguir los "Lineamientos sobre cómo escribir informes técnicos" de la profesora Soraya Abad que está publicado en la misma página web del curso.

Todos los productos se deben enviar vía email al profesor Reinoza a más tardar el día jueves 19 de octubre del 2017 antes de las 18:00.

Carnet	Nombre y apellido		Observaciones
CRITERIOS			
	Informe	**/1,5**	**Observaciones**
Introducción	Afirmaciones generales	0.17	
	Meollo del trabajo	0.17	
	Resumen del contenido	0.17	
	Descripción de los capítulos	0.17	
Cuerpo	Programacion por pares	0.17	
	Experiencias con las herramientas de apoyo	0.17	
Conclusiones	Objetivo	0.17	
	Descripción del trabajo realizado	0.17	
	Recomendaciones	0.17	
Total		**1.50**	
El enfoque TDD en las pruebas (Código)		**/1**	**Observaciones**
Casos de prueba TDD		0.17	
Fronteras		0.17	
Esquinas		0.17	
Malicia		0.17	
Orden de las pruebas		0.17	
Comentarios en código de prueba		0.17	
Total		**1.00**	
Sobre la implementación		**/1**	**Observaciones**
Billetera Electrónica			
-Funcionalidad.		0.33	
-Legibilidad		0.33	
-SOLID		0.33	
Total		**1.00**	
Sobre la consistencia entre la documentación y el código		**/1**	**Observaciones**
Clases		0.33	
Estructuras		0.33	
Métodos		0.33	
Total		**1.00**	
Otros		**/0,5**	**Observaciones**
Instalacion de Django		0.5	
Total		**0.5**	

Total Par Tarea 3	5.00	**/5**