12 de Mayo de 2016
Universidad Simón Bolívar
Ingeniería de Software

Tarea 3:
Programación en pareja
Github y PyUnit

Gabriel Gimenez Carnet: 12-11006
Erick Silva Carnet: 11-10969

Índice.

Introducción.

Para este trabajo se usó principalmente la técnica de programación en pares para realizar una billetera electrónica. Esta técnica consiste en que dos programadores participen en el desarrollo “cooperativo“ en un solo proyecto. Al aplicar esta técnica cada uno de los programadores escoge un rol inicial entre el controlador y el navegador, estos roles son intercambiados entre los dos programadores cada cierto tiempo para evitar fatiga.

El rol de controlador se encarga de la codificación como tal, de tener una visión a corto plazo del código, es decir, este rol se encarga de pensar en las proximas lineas de código, como se van a escribir y qué función se usará en ellas.

El rol del navegador es uno de visión a más largo plazo, como es el de pensar en un caso de prueba para el código que el controlador está escribiendo en el momento, este rol también se encarga de supervisar el código que está escribiendo el controlador, para asi capturar errores comunes que este pueda cometer.

Las ventajas principales de esta tecnica de programacion son un código con menos errores, mayor calidad y un mayor aprendizaje de parte de ambos programadores.

Además, se utilizaron las herramientas de PyDev en Eclipse y de Github que han sido explicadas anteriormente.

Desarrollo.

En esta tarea se pedía usar la técnica de programación en pares para programar una clase “BilleteraElectronica” que sería usada por un restaurante.

Primeramente desarrollamos la clase Billetera Electrónica siguiendo la especificación dada en el enunciado, es decir:

La clase BilleteraElectronica contiene:

- El nombre, apellido y CI del dueño de la Billetera.
- El pin e identificador de la billetera electrónica.
- Una estructura para almacenar todas las recargas.
- Una estructura para almacenar todos los consumos.

Luego se procedió a crear la clase Transaccion, que es una estructura que permite almacenar la información de las recargas y consumos.

La clase Transaccion contiene:

- Fecha de la transacción (Se toma como la fecha en que se crea el objeto)
- Monto de la transacción

Id del establecimiento
Tipo (recarga o consumo)

Durante la programación inicial, el navegante estuvo atento del código del conductor, y estuvo pensando en las diversas pruebas que debían surgir para continuar con el desarrollo dirigido por pruebas.

Una vez se dio por terminada la generación del código, se procedió a generar más casos de prueba: casos borde, casos esquina y casos maliciosos que podrían afectar nuestro sistema. Con cada nuevo caso se probaba el programa contra los casos dados y en caso de fallas se arreglaba el programa y se volvia a probar contra todos los casos.

Las actividades de cada uno de los miembros fueron distribuidas igualmente ya que la mayoria del codigo fue programado con la tecnica de programacion en pares, asi, cada uno programaba y hacia casos de prueba por un tiempo determinado antes de cambiar de rol.

Los tiempo usado por cada uno de los miembros del equipo para la realización del proyecto fueron los siguientes.

Erick Silva:

1 hora y 30 minutos	Escribiendo Informe
2 horas	Haciendo Casos de Prueba
30 minutos	Desarrollando/Arreglando programas principales

Gabriel Gimenez:

1 hora	Escribiendo Informe
1 hora	Desarrollando las clases principales
2 horas	Desarrollando casos de Prueba

Conclusión.

Durante la ejecución de esta asignación, fue importante el utilizo de la función de merge de git, la cual se pudo aprovechar en los momentos que cada integrante trabajaban en sus respectivos ordenadores.

Por otro lado, se indagó aún más en la documentación de PyUnit así como de unittest, para agregar funcionalidades que antes no habíamos experimentado, como las aserciones de lanzado de errores, que se probaron bastante útiles para esta entrega.

En cuanto a la programación en parejas, fue útil para reducir los errores ya que el que tuviera el rol del navegador podía darse cuenta de ellos mientras que el conductor programaba, en lugar de tener que revisar el codigo despues de esto.

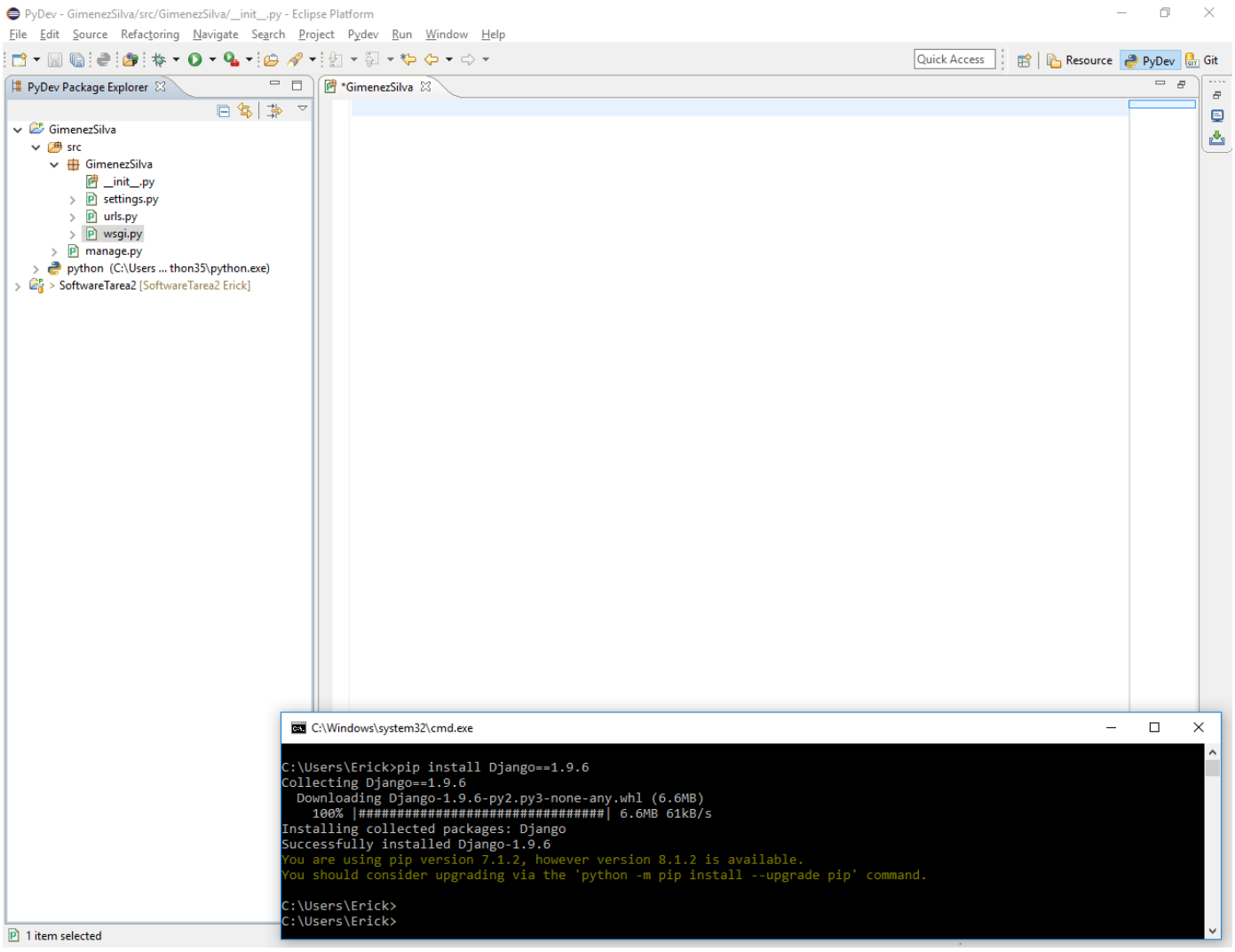
Referencias.

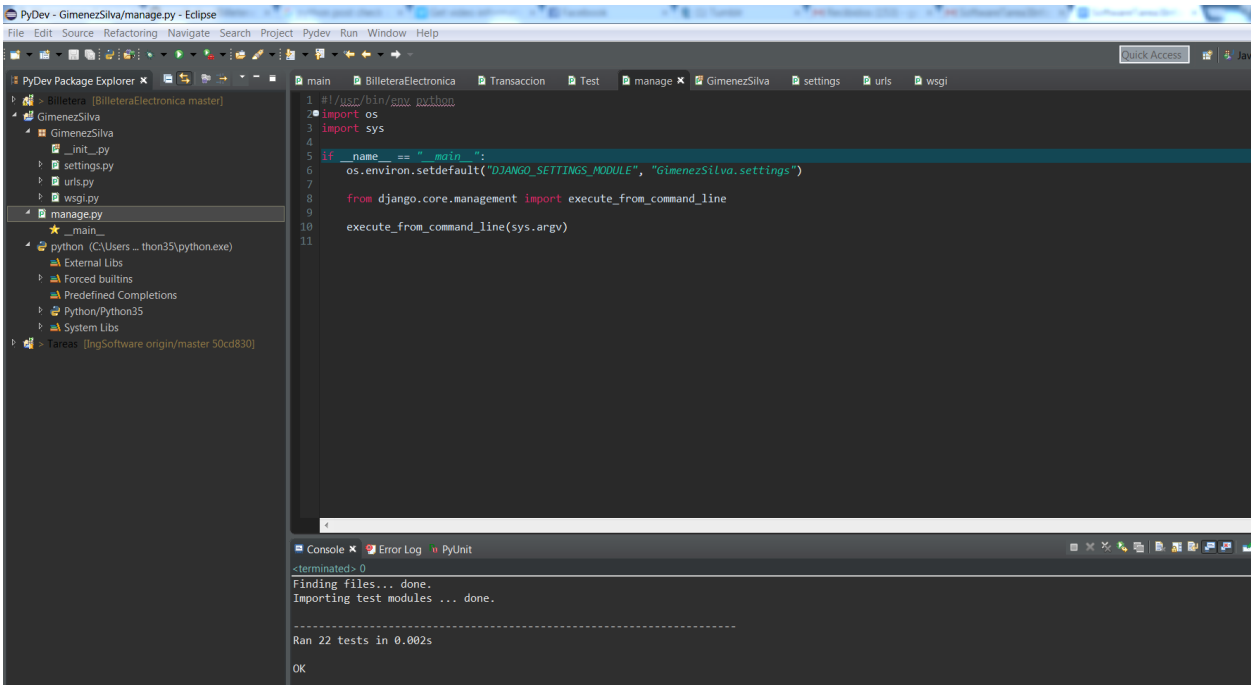
https://es.wikipedia.org/wiki/Programaci%C3%B3n_en_pareja

https://es.wikipedia.org/wiki/GitHub

Apéndice.

Print Screen de proyecto Django como prueba de que todo está instalado.




PyDev - GimenezSilva/manage.py - Eclipse
File Edit Source Refactoring Navigate Search Project Pydev Run Window Help
Quick Access
PyDev Package Explorer
Billetera [BilleteraElectronica master]
GimenezSilva
GimenezSilva
__init__.py
settings.py
urls.py
wsgi.py
manage.py
__main__
python (C:\Users ... thon35\python.exe)
External Libs
Forced builtins
Predefined Completions
Python/Python35
System Libs
Tareas [IngSoftware origin/master 50cd830]
main
BilleteraElectronica
Transaccion
Test
manage
GimenezSilva
settings
urls
wsgi
#!/usr/bin/env python
import os
import sys
if __name__ == "__main__":
os.environ.setdefault("DJANGO_SETTINGS_MODULE", "GimenezSilva.settings")
from django.core.management import execute_from_command_line
execute_from_command_line(sys.argv)
Console
Error Log
PyUnit
<terminated> 0
Finding files... done.
Importing test modules ... done.
Ran 22 tests in 0.002s
OK