

**Universidad Simón Bolívar**

**Ingeniería de computación**

laboratorio de   
INGENIERÍA DE SOFTWARE

**Tarea 3**

**Integrantes:**

* Nabil Márquez 11-10683
* Roberto Rinaldi 10-10605

**Índice general**

1. Introducción…………………………………………….3
2. Planteamiento del problema………………………………………………...4
3. Recursos y tiempo empleado………………………………………………..6
4. Problemas Encontrados y Conclusión…………………………………………......7
5. Repositorio y Bibliografía………………………………………………8

**Introducción**

Este documento tiene como finalidad informar a los lectores sobre el desarrolle de la funcionalidad del sub-sistema de control de acceso utilizando las técnicas de programación en pares, de desarrollo dirigido por casos de pruebas (TDD) y de análisis de fronteras enriquecido con malicia. Los objetivos de la tercera tarea de la materia “Ingeniería del Software I” son:

* 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 ambiente *web* integrado y al problema de la seguridad en el control de acceso.

El informe está dividido en la introducción, el planteamiento del problema donde se explica las actividades realizadas, recursos y tiempo empleado el cual muestra las herramientas tanto teóricas como prácticas usadas para poder efectuar la tarea, además del tiempo que le tomó a cada integrante completar su contribución, le sigue problemas encontrados donde se especifica todos los inconvenientes y deficiencias encontradas durante el desarrollo, conjuntamente con la conclusión. Finalmente damos la dirección de nuestro repositorio con las soluciones, y las referencias.

**Planteamiento del problema**

Para esta asignación se requiere organizar el proyecto en el explorador de proyectos eclipse/pydev de la forma que se indica en el enunciado de la tarea.

La interfaz del usuario correspondiente a la *presentation layer* (presentation),debe contener mínimo cuatro páginasenhtml5 utilizando AngularJS, con sus diferentes funcionalidades, tales como:

* user.html: administra los usuarios del sistema (insertar, buscar, modificar y eliminar usuarios).
* dpt.html: administra los departamentos a los cuales están adscritos los usuarios del sistema (insertar, buscar, modificar y eliminar departamentos).
* role.html: administra los roles que juegan los usuarios del sistema (insertar, buscar, modificar y eliminar roles).
* login.html solicita el Username y el Passwork para el inicio de sesión (como en gmail).

Cada clase debe ser codificada en módulos de pydev a nivel de capa de negocios o *business layer* (business):

* user.py: contiene la clase clsUser y los métodos insertar, buscar, modificar y eliminar usuarios.
* dpt.py: contiene la clase clsDpt y los métodos insertar, buscar, modificar y eliminar departamentos.
* Role.py: contiene la clase clsRole y los métodos insertar, buscar, modificar y eliminar roles.
* Login.py: contiene la clase clsLoginy los métodos longitud, encriptar y check\_password.

Los nombres de las clases deben contener el prefijo clsy deben ser desarrollas utilizando flask en el ambiente eclipse/pydev (ver presentación del taller III).

Cada clase debe contener las diferentes funcionalidades de persistencia del repositorio obase de datos utilizando el enfoque de ORM, tales como: inserción, búsqueda, eliminación y modificación de registros. Se recomienda utilizar como RDBMS a PostgreSQL.

Se debe codificar las diferentes clases que conforman el modelo (model.py) de control de acceso a nivel como capa de datos o data layer (data).

**Recursos y tiempo empleado**

Esta tarea requirió un tiempo excesivo, ya que tuvimos problemas con la instalación de las herramientas utilizadas (Eclipse/PyDev, AngularJS, Flask y sus extensiones Flask-Script, Flask-SqlAlchemyy Flask-Migrate), incluyendo la necesidad de instalación de Ubuntu, lo que represento una demora para empezar a trabajar.

Utilizamos la interfaz de usuario disponible en la página del profesor Ascander Suarez.

División de labores y tiempo empleado en la tarea:

Nabil Márquez Clases, casos de pruebas, model, interfaz. 19 horas aprox.

1. Roberto Rinaldi Casos de pruebas, Informe, clases. 18 horas aprox.

**Problemas Encontrados y Conclusión**

En esta tarea se requiere el desarrollo completo del sub-sistema de control de acceso, para lo cual no estábamos preparados, sin embargo, hicimos todo lo posible. Como ya mencionamos tuvimos problemas en la instalación de todas las herramientas necesarias, para las cuales no se dieron indicaciones completas de como instalarlas. Además, tuvimos un problema con git, que no sabemos por qué ocurre, que se trata de que una de las ramas al hacer push aparece como si fuera la rama principal, y tuvimos un problema a última hora también para hacer merge, por lo que nos parece un poco confuso usar egit y consideramos más fácil de usar git desde el terminal

La interfaz utilizada solamente loguea y tuvimos problemas para integrar la interfaz con los módulos, tuvimos que cambiar el nombre del directorio Access-control, sustituir el – por un \_, lo cual no sabemos por qué no funcionaba con -.

Tuvimos problemas importando módulos de otros directorios, lo cual no sabíamos cómo hacer ya que estando en diferentes directorios no funcionaba, y perdimos tiempo en eso.

Tuvimos que investigar sobre sql alchemy porque no fue explicado.

El desarrollo de esta entrega fue muy agitado, y nos pareció que no aprendimos bien algunas cosas, por ejemplo copiamos algunas cosas de internet, por mencionar algunas, en model.py copiamos los imports y la parte de create\_engine, los cuales no sabemos por qué ni cómo funcionan. Tampoco aprendimos correctamente la configuración del servidor postgresql, y todo lo referente a la interfaz y su integración. Por lo que necesitamos un mejor aprendizaje de los conocimientos necesarios para el desarrollo del subsistema, los cuales se deben dar con el tiempo y claridad necesarios, para poder aplicarlos en el proyecto con la calidad y excelencia que nos caracteriza en la Universidad Simón Bolívar.

Como punto positivo de esta tarea, debemos decir que nos ayudó bastante a familiarizarnos con lo que es la estructura completa de una aplicación web.

**Soluciones Propuestas**

El repositorio en el que se encuentra la tarea es el siguiente:

<https://github.com/nabiljesus/tarea3>

**Referencias**

[1] 23 pags. Publicado en octubre de 2010, por la Soraya Abad Mota. Título: “Lineamientos sobre cómo escribir informes técnicos”. https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxpbmdzb2Z0d2FyZTF0ZXJ1ZWx8Z3g6NmI5NDkwMzFlZjI5OGIyNA