

Autoevaluación del proyecto final “Sistema de evaluación para tesis de maestría”

Juan José Restrepo Toro

Luisa Rincón

Programación orientada a objetos

Pontificia Universidad Javeriana Cali

Santiago de Cali

Mayo de 2022

Retos y complicaciones en el proyecto

En el proyecto final se tuvieron diversos aprendizajes y retos para poner en práctica lo aprendido en las clases de “Programación orientada a objetos”.

En primer lugar, se mejoró el manejo en herramientas como Git y Github, lo que conllevó a un mejor uso de los commits, de las ramas y del entorno. Además, fue posible implementar una mejor organización de la estructura lógica que debe tener un programa de software que interactúa con un usuario; de igual manera, se implementó el modelo MVC con el fin de distribuir de mejor manera el código para obtener un programa mejor organizado y cuyo mantenimiento se facilita al usar esta clase de esquemas de diseño. Además, se logró trabajar con una Framework llamado “Streamlit”, el cual permite crear interfaces Front-END de una manera simplificada y que permite crear aplicaciones web de manera rápida.

Por otro lado, se implementaron nuevas prácticas de la ingeniería de software tales como las excepciones y las pruebas unitarias, las cuales permitieron obtener un programa de mejor calidad y cuyas interacciones permiten que a pesar de que haya fallas por parte de los usuarios al utilizar el sistema, este siga funcionando.

Para este caso, se trabajó con el paradigma POO en el lenguaje de programación Python, lo cual conllevó un cambio en la forma en la que se implementan los conceptos vistos en el curso de “Programación Orientada a Objetos”. Sin embargo, se logró abstraer los temas y nociones vistas previamente e implementarlas con buena medida en Python.

En este orden de ideas, se entendió la diferencia de programar mediante POO en C++ y Python, tales como la forma en que se guardan las instancias en memoria, la forma en que trabajan los constructores, etc. También, fue posible ver las implicaciones y utilidades que tienen los tanto paquetes en Python como el archivo “requirements.txt”

Por otro lado, se encontró cierta complejidad para entender los comandos e instrucciones con las que trabaja “Streamlit”. Ejemplo de ello, fue el error en cuanto al manejo de las sesiones en este Framework y al uso de llaves únicas para cada entrada gráfica que se tiene.

Igualmente, se encontró dificultad para obtener el archivo PDF correspondiente a un acta exportada.

Cabe mencionar que en algún momento se intentó implementar un sistema de login al sistema, de tal manera que dependiendo del usuario que desee ingresar y una contraseña, la persona ingresara a las funciones que tiene acceso como dicho usuario en particular.

Finalmente, un aspecto que se tuvo que revisar fue el diagrama UML, ya que, a medida que se trabajó en el proyecto, fue necesario implementar nuevos métodos y atributos que no estaban originalmente planteados en el UML. Por consiguiente, se tuvieron que realizar actualizaciones en el UML, los cuales debían dar cuenta de las relaciones existentes entre las clases, los atributos y los métodos de cada una.

Mi percepción del proyecto

Vi este proyecto como algo que buscaba retornos a cada uno de nosotros. Para mí, fue muy interesante partir de una idea de diseño para dar solución a un problema y luego llegar a un sistema con interfaz gráfica e interacciones entre cada una de las clases. Así mismo, pienso que se nos planteó la tarea de ponernos en la posición del usuario y pensar no solo como programadores, sino como personas que van a acceder a ese sistema y que les gustaría que funcionara de cierta manera.

También, personalmente, me sentí a gusto al desarrollar este proyecto, ya que sentía que tenía claro los conceptos y nociones vistas en clase, de tal manera que contaba con la capacidad de abstraer esos temas e implementarlos en un problema de la vida real.

Trabajo en equipo

En cuanto al trabajo en equipo, se priorizó la planeación y el diseño colectivo del programa. En ese orden de ideas, la construcción del primer esquema de UML, se realizó en conjunto con el fin de que cada uno de los miembros lográramos comprender los métodos y los atributos de la clase y así, pudiéramos hablar el mismo “lenguaje”.

De tal manera, que se planteó un trabajo donde cada miembro trabaja individualmente, pero mantenía “actualizado” a la otra persona de los cambios en el programa que iban teniendo lugar. Así mismo, se realizaron sesiones de trabajo en equipo en las cuales uno de los integrantes codificaba y la otra persona aportaba ideas y correcciones.

Este equipo estaba conformado por **Francesco Totti** y mi persona, **Juan José Restrepo**. A continuación, se expone el trabajo realizado por cada uno de los miembros:

- Juan José Restrepo Toro:
 - Ambos trabajamos en todas las partes que involucraba la construcción del sistema (Modelo, Vista, Controlador). Yo me encargué de la parte de pruebas unitarias y de la definición del controlador. También, hice el manual técnico y los cambios finales al UML.
- Francesco Totti:
 - Francesco se encargó del manejo de excepciones en el programa y de revisar cómo exportar y crear los archivos PDF. También, escribió el readme y trabajó en la construcción del sistema MVC.

Continuando con la reflexión, siento que pudimos haber hecho ciertas cosas de una manera más organizada. No obstante, considero que en este trabajo pudimos ver reflejado nuestro aprendizaje y que logramos obtener un sistema bueno y funcional.

Finalmente, por todos los aspectos mencionados anteriormente, considero que ambos integrantes del equipo, podríamos contar con una nota relacionada al trabajo en el proyecto de **4.8**.

Para concluir, mencionar me gustó este proyecto y la posibilidad de aplicar todos los conocimientos visto en este. Al mismo tiempo, logré desarrollar cierto nivel de experticia en el uso de herramientas como Git y GitHub. Además, logré conocer cómo se maneja el paradigma

POO en 2 lenguajes de programación y aprender buenas prácticas que se manejan en la ingeniería de software.

