

Primer proyecto

Víctor Orozco

19 de septiembre de 2017

1. OBJETIVOS

El objetivo de este proyecto es que el estudiante mejore sus habilidades primaras de desarrollo de software, tales como:

- Elaboración de **documentación técnica**
- Buenas practicas de **programación por pares**
- Implementación de software mediante **programación orientada a objetos**
- **Manejo e implementación** de estructuras de datos básicas

2. DESCRIPCIÓN DEL PROYECTO

El congreso de la república de Goathemala se ha percatado que una de las mayores deficiencias al momento de emitir los votos en el hemiciclo parlamentario es el desconocimiento de las leyes del país.

Por tal motivo, ha solicitado su apoyo como futuro ingeniero de software para el diseño y desarrollo del sistema denominado:

Sistema de control y préstamo de leyes y reglamentos de la republica - SisLey

El objetivo de este sistema es que cualquier diputado y/o asesor de diputado tenga a su disposición una biblioteca completa de leyes y reglamentos del país bajo la premisa de que mientras mejor acceso tengan a las leyes, sus votos mejoraran.

2.1. FUNCIONALIDAD ESPERADA

Para el funcionamiento de SisLey, el mismo debe permitir las siguientes operaciones de administración de información:

- **Usuarios:** Creación, modificación y eliminación de grupos de usuarios. Entendiendo un grupo de usuarios como un conjunto de usuarios conformados por un parlamentario y hasta un máximo de ocho asesores.
- **Leyes:** Creación, modificación y eliminación de leyes. Así como administración de múltiples copias hasta un máximo de cinco.
- **Reglamentos:** Creación, modificación y eliminación de reglamentos. Así como administración de múltiples copias hasta un máximo de cinco. Los reglamentos a su vez deben de estar relacionados con las leyes ya que una ley puede tener un número variable de reglamentos

Adicionalmente el sistema debe permitir las siguientes operaciones:

- **Préstamo:** Una ley puede ser asignada a un usuario de un grupo y un grupo puede tener como máximo dos copias de la misma ley y/o reglamento. Las solicitudes deben funcionar como una lista de espera en caso de que las leyes se agoten.
- **Préstamo por lotes:** A solicitud de cualquier interesado el sistema debe ser capaz de asignar un lote de ley+reglamentos a un usuario.
- **Devolución:** Las copias de las leyes y reglamentos deben rotarse adecuadamente, por lo tanto a devolverse una ley o reglamento debe ser la última a ser prestada en posteriores ocasiones.
- **Informe por ley:** El sistema debe permitir visualizar el estado actual de las leyes y reglamentos en préstamo y en reserva. Y mostrar sus relaciones (es decir si se consulta una ley, debe mostrar el estado actual de la ley y sus reglamentos, y si se consulta un reglamento, debe mostrar el estado de sus copias así como la ley a la que pertenece).
- **Informe por grupo:** El sistema debe permitir visualizar el estado actual de las leyes y reglamentos asignados a un grupo, así como los usuarios responsables por cada préstamo.

3. REQUERIMIENTOS TÉCNICOS

Para ser considerada para calificación su solución debe cumplir con los siguientes requerimientos técnicos:

- Para su implementación puede utilizar Java o C# utilizando componentes gráficos (se sugiere Swing o JavaFX para Java y Winforms para C#).

- Las estructuras de datos a ser utilizadas pueden ser seleccionadas a discreción del estudiante, sin embargo la implementación de las mismas es manual quedando prohibido el uso de bibliotecas de funciones.
- El proyecto debe ser documentado adecuadamente tanto a nivel de código (Java-JavaDoc, C#-XmlDoc), así como a nivel de proyecto (documentación con Markdown).
- El proyecto debe ser implementado en parejas a través de un repositorio en GitHub, el historial del proyecto debe reflejar la colaboración de ambos miembros de la dupla.
- Proyectos con menos de 5 commits, sin documentación, con copias de otro proyecto quedan automáticamente descartados.

4. PRESENTACIÓN

El grupo debe elaborar una presentación tipo Lightning Talk la cual debe ser entregada en formato PDF y cuya presentación sera cronometrada (5 minutos) el día designado en clase. Algunos ejemplos:

- Top 10 Things Android Game Developers Should Know
- Scala vs. LINQ

Adicionalmente se seleccionara otra dupla al azar para evaluar la calidad de la implementación en un proceso de revisión de código.

5. CONDICIONES DE TRABAJO

Ultimo commit permitido: **8 de octubre de 2017, 23:55 PM**

Fecha de presentación: **9 de octubre de 2017**