

**Proyecto final**  
**Programación orientada**  
**A objetos**

**Presentado por:**  
**Juan camilo hernandez saavedra**  
**Código:**  
**8931911**

**Presentado al profesor:**  
**Gerardo sarria**

**Junio 6 2018**  
**Semestre 2018-1**

## 1. Análisis:

- **Objetivo general:** se desea generar un software, que sea capaz de dar apoyo a la administración y manejo de versiones de archivos, sean código o simplemente texto, este programa deberá tener un sistema que permita tener un registro de los cambios de los archivos que se manejen en el software, para poder tener control sobre los cambios que se le realicen sobre estos. Además, este software debe permitir además de usar el manejo de versiones lineal, poder usar el manejo “ramas” y así poder permitir mejorar, corregir versiones viejas.
- **Objetivos específicos:** se busca que este nuevo software sea capaz de:  
Poder obtener versiones y dependiendo de la versión obtenida generar un delta.

poder modificar la versión ya obtenida y guardarla en el registro de versiones respectivo.

poder crear el archivo de control a los archivos que no posean uno, para poder así, tener control de sus futuras versiones, estos archivos deberán llamarse de la forma: s\_nombreArchivo.

el software podrá además de tener la posibilidad de manejar versiones, dejar trabajar como un editor de texto normal, se podrá tanto abrir archivos y guardar archivos, estas opciones estarán deshabilitadas mientras se haya obtenido una versión y no se haya modificado.

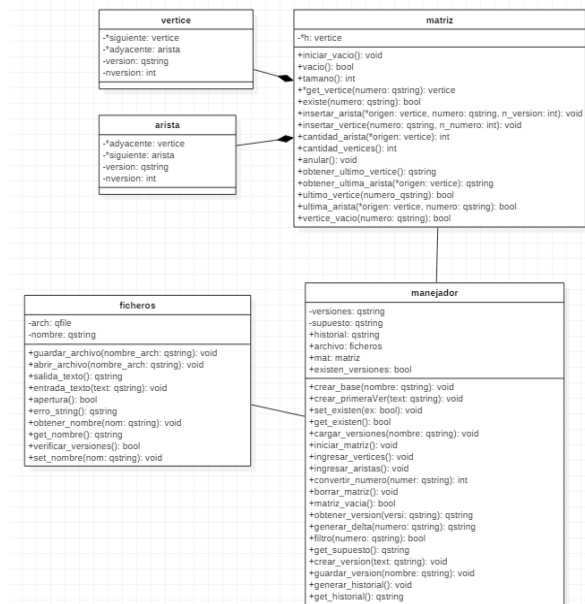
- **Diseño:** este software es realizado en el lenguaje c++, ayudado con la herramienta de desarrollo grafico de Qt creator, donde proveerá manejo de versiones sobre los archivos de gran importancia como pueden ser código, o textos. Este software permitirá usarse como editor normal, y además permitirá tener control sobre las versiones, teniendo como funciones importantes obtener, crear, modificar, mostrar historial.
1. **Función crear:** esta función se encargará de crear el archivo de versiones, en el caso de que el archivo en cuestión no posea uno. El caso de que el archivo no esté guardado o ya posea un archivo de versiones no permitirá crear uno. El software al abrir un archivo

se encargará de advertirle en el caso de que no posea archivo de versiones.

2. **Función obtener:** esta función se encargara de pedirle al usuario la versión que desea modificar, para ello existe la posibilidad de pedir una versión en concreto, es decir, introduciendo la versión exacta que desea obtener, o introduciendo ciertos valores como puede ser; un numero entero mayor al número de versión de la rama principal, que obtendrá la última versión de la rama principal, o bien introducir un numero mayor a la rama secundaria, donde se obtendrá la ultima versión de esa subrama. Al obtener una versión se generará un delta, que será de ayuda para modificar la versión, pues este delta será el “supuesto” de número de versión que tendrá esa modificación, esto se generara dependiendo que entrada o que versión se obtuvo. Al obtener una versión no se podrá guardar, abrir, o crear un archivo de versiones.
3. **Función modificar:** esta función se encarga de guardar la versión modificada, que se haya obtenido anteriormente, usando el delta como su nuevo numero de versión, al usar esta función se podrá volver a usar las funciones como guardar, abrir, etc.
4. **Función mostrar historial:** esta función se encarga de mostrar el historial de versiones, muestra las versiones en su orden de creación, esto solo se podrá usar si el archivo que se este usando en ese momento posea versiones o mínimo un archivo de versiones.

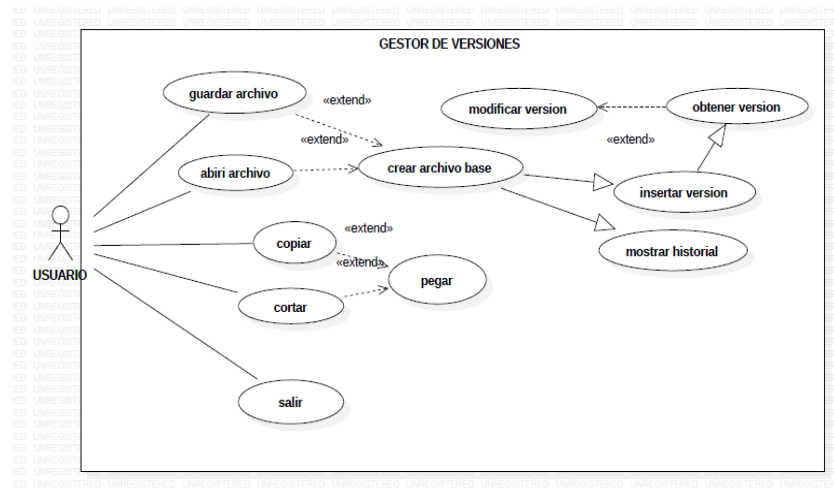
## 2. Diagramas:

- Diagrama de clases:



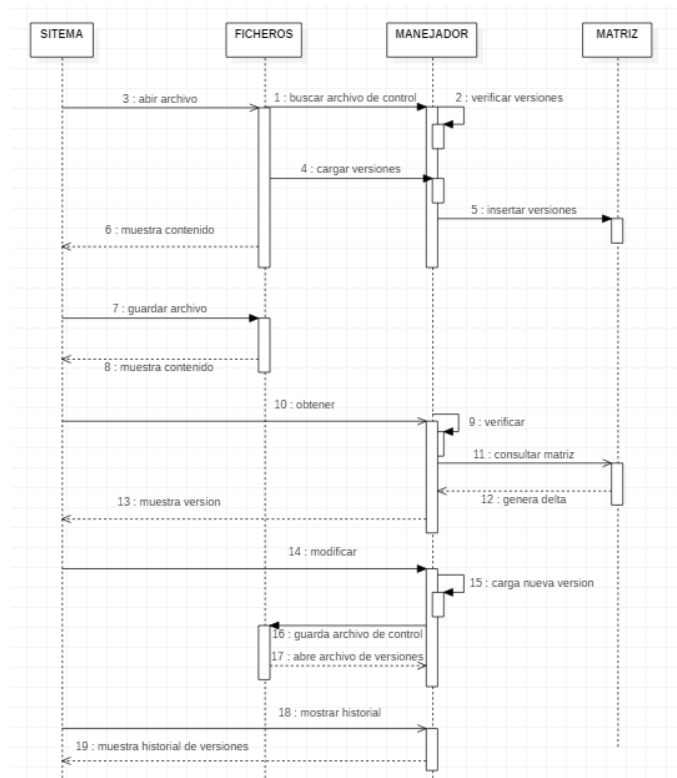
(los diagramas van a estar con mejor vista en los archivos pdf correspondientes)

- Diagrama de casos de usos:



(los diagramas van a estar con mejor vista en los archivos pdf correspondientes)

- Diagrama de secuencia:



(los diagramas van a estar con mejor vista en los archivos pdf correspondientes)

### 3. Bitácora de desarrollo:

- **Primera fase del proyecto (comienzo):**

**Establecer la justificación:** Este proyecto tiene como objetivo principal aplicar las metodologías y herramientas vistas en clase en la materia objetos y programación a gran escala dictado por el ingeniero Gerardo Sarria, persona que nos introdujo en la ingeniería de software que es un campo amplio y de suma importancia para nuestra carrera (ingeniería de sistemas y computación). El proyecto del gestor de versiones será una forma útil de manifestar la interacción entre ingeniero – cliente, además, de esta forma se evalúan los conocimientos adquiridos durante estos tres semestres que llevamos en nuestra carrera y nos da una perspectiva de cómo podría ser nuestro campo laboral si optamos por el camino de ingeniería de software.

Como objetivos más específicos, se pueden resaltar algunos tales como:

- ✓ Conocer y aplicar el paradigma de la programación orientada a objetos para el desarrollo de software.
- ✓ La utilización de estándares de codificación y programación segura que nos ayudan a desarrollar sistemas robustos y legibles.
- ✓ El uso adecuado del análisis y diseño de sistemas complejos mediante herramientas como UML.
- ✓ Profundizar en aspectos relevantes como las interfaces gráficas de usuario, excepciones en los sistemas y metodologías de desarrollo.

**Dibujar los diagramas correspondientes:** los diagramas están en los documentos pdf correspondientes, y están en las imágenes anteriores.

**Identificar riesgos:**

Tiempo límite.

Fallas en el sistema.

Desconocimiento sobre temas no abordado en clase que se incluyen en el proyecto.

Sistema debe ser compacto y seguro.

- **Segunda fase del proyecto(elaboración):** se deben capturar todos los requisitos del sistema:

El usuario podrá obtener las versiones que desee y así mismo poder modificarlas, para posteriormente ser guardadas en el archivo de control.

El usuario podrá crear archivo de control a todos los documentos que no posean uno con el fin de poder empezar a manejar y controlar las versiones de su archivo o documento.

El usuario podrá ver el historial de versiones, si es que el archivo posee versiones, en el orden en el que fueron creadas.

El usuario podrá ver las diferencias que posee cierta versión que el desee, con respecto a la versión inicial o versión base.

Los archivos de control solo deberán guardar el numero de versiones y las diferencias que tienen con respecto a la versión que se modificó.

Como extra el usuario podrá guardar, abrir, editar sus archivos como un editor de texto normal.

- **Establecer los riesgos y sus posibles mitigaciones:**

**Tiempo límite:** para manejar este aspecto opte por crear horarios específicos en los cuales poder dedicarle al proyecto, además como estoy solo en proyecto, poder de manera eficiente desarrollar el proyecto completamente, en el tiempo específico que tenemos.

**Fallas en el sistema:** cada función desarrollada será probada para encontrar posibles errores, fallas, etc. esto se realizará en un programa de solo texto, pues facilita la depuración a la hora de encontrar errores. Además, se prevé que el programa sea probado por agentes externos, con el fin de poder encontrar fallos a la hora de la ejecución.



**Desconocimientos de temas no visto en clase:** sobre temas no abordados en clase, como herramientas que nos ayuden a desarrollar el software, se buscara opinión de ingenieros de sistemas y compañeros de semestres avanzados, temas como las interfaces gráficas, también se investigara por medio de otras fuentes para facilitar los requerimientos de sistema.

**El programa debe ser compacto y seguro:** gracias a las bases acerca de la programación segura que se dio en el curso, se conocen métodos como el manejo de excepciones, las validaciones de entradas, entre otros, los cuales ayudan a que la aplicación haga frente a cualquier escenario que se le presente. Sin embargo, se consultarán fuentes como el internet y libros de programación para garantizar la seguridad en la información.