Manual Técnico: “Creación de un generador de actas de posgrado haciendo uso de la programación orientada a objetos”.

Proyecto 1

Carlos Felipe Palacio Lozano

Pontificia Universidad Javeriana Cali

Facultad de ingeniería y ciencias

Ingeniería en sistemas y computación

Cali, Valle del Cauca

Contenido

[Presentación. 3](#_Toc67389797)

[Principales funcionalidades. 4](#_Toc67389798)

[Crear acta. 4](#_Toc67389799)

[Crear persona 5](#_Toc67389800)

[Crear jurado. 6](#_Toc67389801)

[Diligenciar comentarios, calificaciones y criterios. 6](#_Toc67389802)

[Crear archivo de texto con los datos de un acto 7](#_Toc67389803)

[Funcionalidades de mostrar y listar. 7](#_Toc67389804)

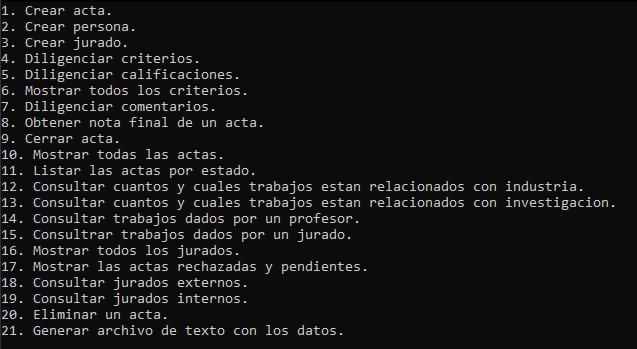
[Diagrama de clases 8](#_Toc67389805)

# Presentación.

En el presente documento se mostrarán las principales características del proyecto de la materia de Programación Orientada a Objetos denominado como: “Creación de actas de grado” que busca aplicar las enseñanzas que se nos han impartido en clase.

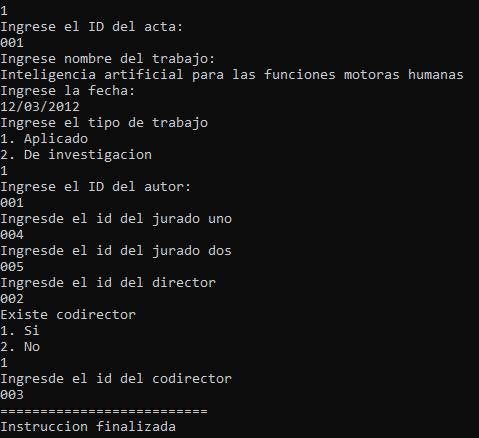
# Principales funcionalidades.

El programa cuenta con 21 funcionalidades principales que están incluidas en el menú del proyecto en el archivo main.cpp, todas ellas son métodos de la clase maestría, y se almacenan los valores en listas de dicha clase.



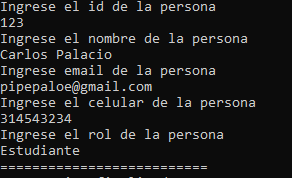
## Crear acta.

La finalidad de este método es crear un acta e ingresarla en la lista de actas que se tiene en la clase universidad (ver diagrama de clases). En esta función pide ingresar el id de personas que deberían estar dentro de las demás listas del sistema (ser previamente creadas), pero en caso de que no lo sean, el método redirigirá al usuario para que pueda crear la persona que necesita.



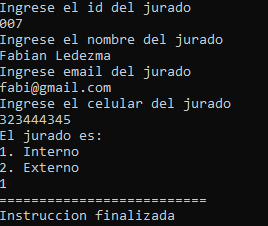
## Crear persona

Al igual que el método de crear acta, crear persona permite al usuario crear una persona de tipo estudiante, director o codirector que será añadido a la lista de personas para que puedan ser utilizadas en las demás funcionalidades del programa. Importante: este método no puede crear personas de rol jurado.



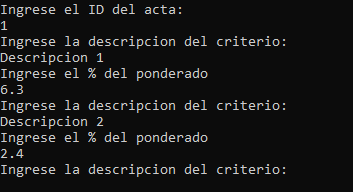
## Crear jurado.

Al igual que la funcionalidad anterior, los jurados tienen una lista, y una clase, exclusiva para ellos, ya que en la creación del programa se contempló que tendrían métodos únicos a los que solo ellos podrían acceder, en este método solo es posible crear personas de rol jurado.



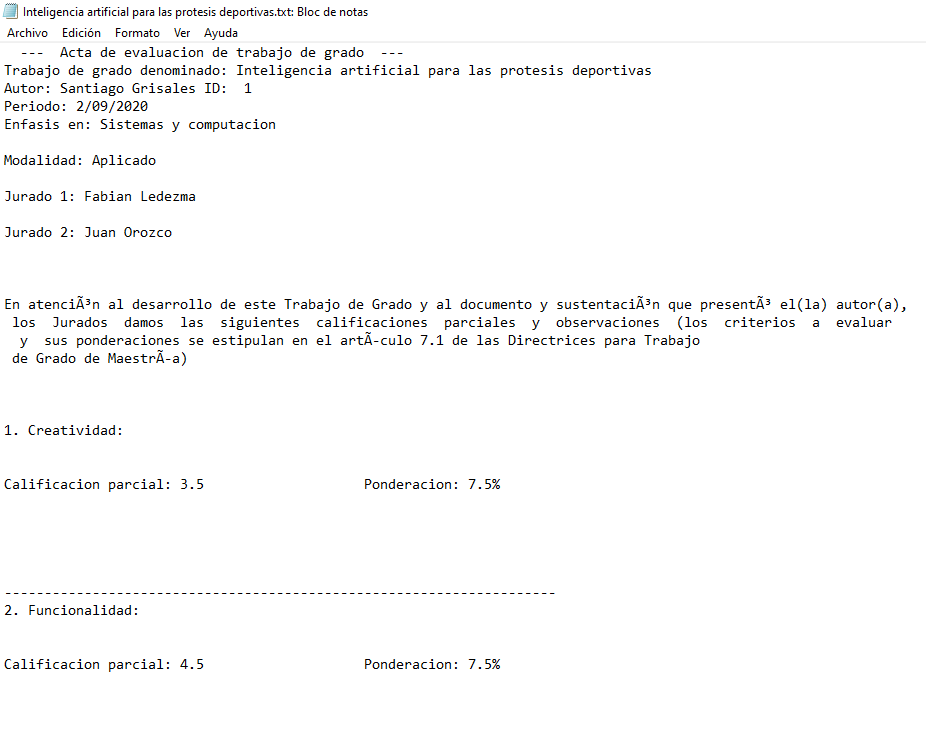
## Diligenciar comentarios, calificaciones y criterios.

Las funcionalidades #4, 5 y 7 están relacionadas, todas hacen referencia o uso de otras dos clases llamadas Detalle Criterio y Criterio. La funcionalidad #4 permite crear los criterios de cada acta (por defecto establecido en 8), haber utilizado la funcionalidad #4 es necesario para poder hacer uso de las dos siguientes. La #5, permite diligenciar las calificaciones de los criterios ya creados previamente, al igual que la #7, pero con los comentarios.



## Crear archivo de texto con los datos de un acto

Esta funcionalidad es de las más llamativas del programa. Permite generar un archivo de texto con los datos del acta que el usuario haya introducido. A continuación, un fragmento del archivo de texto que genera el programa.



## Funcionalidades de mostrar y listar.

Las funcionalidades de mostrar y listar son las que más abundan en el programa. Todas ellas, se encargan de mostrar una serie de datos o mostrar un dato que el usuario desee de algún acta, persona, etc, en específico.

# Generalidades

Dentro del programa, encontramos distintas funciones cuyas características parecen repetirse. Esto es debido, porque la mayor parte de las funcionalidades (al menos de que creen algo), necesitan buscar en las listas contenidas en el programa un dato que vincule el objeto que se quiere modificar con los datos del programa. En la gran mayoría de casos, este dato es el ID, tanto del acta, como del usuario, como de los jurados. Al estar trabajando con listas por nodos en C++, no podemos acceder directamente a una posición de memoria dado que no son contiguas, es por ello, que se crean los Iterator, que son apuntadores a la posición de memoria del objeto que queremos buscar. La función encarga de devolver estos apuntadores en el caso de las actas es la siguiente.

list<Acta>::iterator Maestria::obtenerActa(int idActa){

    for(list<Acta>::iterator it = listaActas.begin(); it != listaActas.end(); it++){

        if(it->getNumero() == idActa){

            return it; // Función que devuelve un apuntador al acta cuando la encuentra

        }

    }

}

Este método es de los pilares en el funcionamiento de la clase actas y en general de todo el programa.

# Diagrama de clases

