**Universidad Nacional Campus Sarapiquí**

**Estudiante:**

**Axel Andrade Villalobos**

**Carrera:**

**Ingeniería en sistema**

**Curso:**

**Programación 1**

**Proyecto 3**

**Profesor:**

**Steven Cruz**

**Año: 2016**

**Introducción**

El presente documento presenta las diferentes descripciones sobre aspectos que se llevaron a cabo durante el desarrollo del proyecto. En este sentido, es importante especificarlos y analizarlos, ya que apunta a un mejor entendimiento del mismo.

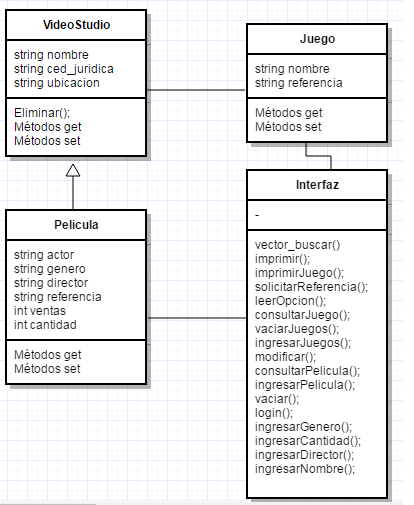
Durante el reconocimiento del problema se seleccionaron los elementos necesarios para terminar con éxito lo planteado, entre ellos la investigación por internet en sitios con información útil y con análisis de diferentes temas sobre el lenguaje de programación C++.

Dentro de este marco, el documento esta pensado para mostrar el siguiente orden de aspectos:

* UML
* Mencionar las clases usadas y sus relaciones.
* Explicar los métodos utilizados.
* Funcionamiento

Sobre las bases de las ideas expuestas, a continuación se procede a desarrollar cada punto.

**UML**

****

**Clases usadas**

* VideoStudio: Clase principal.
* Pelicula: Clase heredada de VideoStudio.
* Juego: Clase que se relaciona con VideoStudio.
* Interfaz: Contiene todos los métodos que interactúan con las clases Pelicula y Juego.

**Métodos**

**vector\_buscar():***Al recibir el vector y el objeto, recorre el vector y verifica que si el objeto pasado por referencia es igual al del arreglo, lo devuelve.*

**Imprimir():***Recibe el parámetro e imprime las características de la película.*

**imprimirJuego():***Recibe el parámetro e imprime las características del juego.*

**solicitarReferencia():***Ingresa referencia según usuario.*

**leerOpcion():***Ingresa opción según usuario.*

**ingresarNombre():***Ingresa nombre según usuario.*

**ingresarDirector():***Ingresa dirección según usuario*

**ingresarActor():** *Ingresa actor según usuario.*

**ingresarCantidad():***Ingresa cantidad según usuario.*

**ingresarGenero():***Ingresa genero según usuario.*

**Login():***Verifica si el usuario y password establecidos son iguales a lo establecido por el usuario, si es igual devuelve true sino, llama recursivamente al metodo otra vez para ingresar los datos.*

**vaciar():** *Recibe el puntero que apunta a película y lo setea.*

**ingresarPelicula():***Guardar la película por referencia según el número que ingrese el usuario, si la película existe, no se guarda, sino empieza a ingresar datos.*

**consultarPelicula():***Buscara la película por referencia según el número que ingrese el usuario, si la película existe, se procederá a imprimir, sino no muestra registro alguno.*

**modificar():***Recibe el puntero que apunta a película y lo setea.*

**ingresarJuego():***Solicita un numero de referencia donde se guardara una película, si existe, retorna puntero sin datos, ya que si se deja así, el puntero lo devuelve con datos y se guarda en el vector, cosa que no debe ser así, ya que el vector solo guarda objetos con sus respectivos datos. En caso contrario se procede a ingresar datos y retorna el juego contrario.*

**vaciarJuegos():***Recibe el puntero que apunta al juego y lo restablece.*

**consultarJuego():***Busca el juego en el arreglo, si esta, imprime los datos del mismo, de lo contrario muestra que no existe.*

**Funcionalidad**

El programa inicia solicitando un usuario y clave, luego estos datos son comparados, si son correctos, el proceso salta a un menú; en caso contrario se repite el proceso de ingresar datos hasta que sean correctos. De lo expuesto anteriormente, una vez en el menú el usuario puede elegir las opciones según desee, desde agregar una película, eliminarla, alquilar, etc.

Todas estas opciones fueron implementadas con vectores de tal manera que se pudiera acceder a una película guardada en los mismos y así mismo aplicarle diferentes funciones como las mencionadas.

Otra de las herramientas útiles fueron los punteros, ya que acceden directamente a los datos guardados en los vectores. Además se utiliza la interfaz que separa los métodos de la implementación y de la misma manera hace fácil transmitirle datos a los objetos que se declaren.

**Conclusión**

De lo anterior, es posible concluir que muchas de las herramientas y la investigación que se llevó a cabo, produjo un nuevo aprendizaje y además abre la posibilidad de resolver diferentes problemas con más agilidad. Aunque hay muchas herramientas por descubrir, el conocimiento adquirido ha sido bueno y puede dirigirse a una mejor comprensión de los lenguajes de programación. En este sentido, es importante llevar a la práctica lo aprendido y sobre todo seguir ese camino para desarrollar una mejor forma de resolver los diferentes problemas.