

|  |
| --- |
| Ingeniería de Software |
| Trabajo Práctico N° 1 |
| Grupo: RDM  Integrantes: •Abratte, Diego José  •Moral, Juan Ramiro  •Rodriguez, Manuel  Docentes: •Jorge, Javier  •Nonino, Julián  •Micelli, Martín |

ÍNDICE

[Introducción 2](#_Toc483843744)

[Plan de Gestión de las Configuraciones 3](#_Toc483843745)

[Documento de Requerimientos 8](#_Toc483843746)

[Arquitectura del Sistema 23](#_Toc483843747)

# Introducción

En el presente informe se detallan las metodologías de Ingeniería de Software utilizadas para la implementación de un nuevo producto llamado Don’t Forget It. Dicho producto inicialmente consta de una aplicación para PC que permite al usuario cargar una serie o un programa de TV con sus respectivas temporadas y capítulos para luego poder señalar qué temporadas y qué capítulos fueron vistos. Para la realización del software se utilizó el entorno de desarrollo de Java IntelliJ.

# Plan de Gestión de las Configuraciones

**Dirección y forma de acceso a la herramienta de control de versiones:**

Para el control de versiones se utilizará GitHub que es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git (sistema distribuido de control de código o SCM). Dicha herramienta permite el seguimiento del historial de archivos y directorios, pudiendo identificar fácilmente cambios realizados y quiénes realizaron esos cambios .

En GitHub, creamos un repositorio público en el cual nos encontramos los tres miembros del equipo. De esta manera podemos visualizar el estado del proyecto y realizar las modificaciones que se crean necesarias. La dirección del repositorio nombrado anteriormente es:

<https://github.com/rmoral45/2017-UNC-IngSoft-RDM>

Las operaciones relacionadas al control de versiones son realizadas en la consola de Windows para los integrantes del grupo que usan este sistema operativo y en la consola de Linux para los integrantes del grupo que usan este sistema operativo.

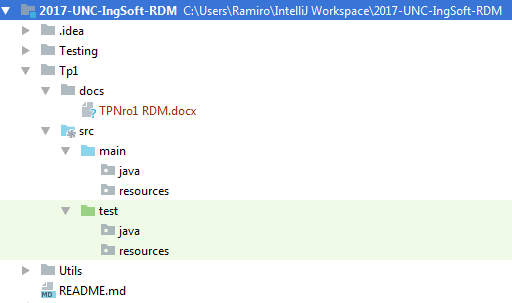
**Dirección y forma de acceso a la herramienta de integración continua:**

Para la integración continua se utilizará Travis CI, que es un servicio de integración continua distribuido utilizado para construir y proyectos de prueba alojados en GitHub. Travis CI detecta automáticamente cuando se realiza un commit y push a un repositorio GitHub que está utilizando Travis CI, y cada vez que esto sucede, se tratará de construir el proyecto y ejecutar pruebas. Esto incluye commit en todas las ramas, no sólo a la rama principal.

El link para acceder a la herramienta de integración continua es:

<https://travis-ci.org/rmoral45/2017-UNC-IngSoft-RDM>

**Esquema de directorios y propósito de cada uno:**



En **src** se encuentran los archivos fuente del proyecto, del tipo **\*.java**.

En **Documentos** se encuentran los archivos de texto, del tipo **\*.pdf**.

En **2017-UNC-IngSoft-RDM**, se encuentran archivos autogenerados por GitHub y por IntelliJ (IDE utilizada).

**Normas de etiquetado y nombramiento de los archivos:**

Para un mejor seguimiento del historial de versiones se optó por una nomenclatura **v.A.B.C** para nombrar los archivos que se suban al repositorio. Estos deben cumplir con las siguientes condiciones:

* Siempre se comienza con la letra **“v”**.
* El primer número **(A)** indica los cambios mayores en el diseño del código (Indica la versión mayor). Estas versiones son estables. Debe incrementarse a medida que se realicen modificaciones significativas en el código. Se comienza con el número 1.
* El segundo número **(B)** indica cambios menores en el diseño del código (Indica la versión menor). Éste se modifica cuando, por ejemplo, se lleva a cabo la implementación de uno o más requerimientos como así también la implementación de nuevas funcionalidades. Se comienza con el número 1. Cada vez que se incrementa el primer dígito **(A)**, éste **(B)** vuelve a ser 1.
* El tercer número **(C)** representa correcciones de bugs o errores encontrados. También se incluyen cambios no funcionales (correcciones ortográficas, comentarios, etc.).

La versión inicial es la **“v1.1.0”**. Luego, a medida que se van realizando modificaciones en el sistema, la versión irá cambiando de números según corresponda.

Esto es implementado de esta manera ya que ayuda a mejorar la comunicación entre los miembros del equipo, manejo de los lanzamientos y administración de fallos.

**Plan del esquema de ramas a utilizar**:

Se creará una rama cada vez que un colaborador del proyecto quiera modificar el mismo, o agregar una funcionalidad. La rama creada deberá especificar en su nombre la versión que se está modificando, junto con el nombre del colaborador, más un número que indique su propia versión de modificación. Se debe cumplir con el siguiente formato:

**versión-nombre-versiónPropia**

Por ejemplo: **v1.1.0-Juan-0** para indicar que se modifica la versión 1.1.0 del proyecto por el colaborador Juan en su primera versión de cambio.

Cuando el desarrollador considere que su versión está lista para una nueva entrega, deberá fusionar la rama principal con su propia rama (merge) para que sus cambios queden registrados en la rama principal.

**Forma de entrega de los “releases”, instrucciones mínimas de instalación y formato de entrega:**

Los releases serán generados cada vez que se avance en una versión mayor del sistema. Éstas son las únicas versiones potencialmente entregables, las versiones internas serán utilizadas para control y gestión interna. Las distintas versiones del release del software se entregarán en formato ejecutable.

Las instrucciones básicas para instalar el software son:

1. La computadora en la que se desee instalar el software debe contar con el sistema operativo Windows 7 (32 0 64 bits) en adelante.
2. Se deberá ejecutar el archivo de instalación.

**Change Control Board (CCB). Listado y rol de los integrantes del equipo:**

Los integrantes y sus respectivos roles en la CCB son:

► Abratte, Diego José: Desarrollador y Gestor del producto.

► Moral, Juan Ramiro: Desarrollador y Tester.

► Rodríguez, Manuel: Desarrollador y Diseñador de Interfaces.

A modo de aclaración:

El desarrollador es el encargado de llevar a cabo la codificación del software.

El gestor del producto es el encargado de administrar y organizar el proyecto (manejo de tiempos, reparto de tareas, etc.).

El diseñador de interfaces es el encargado de realizar la interfaz gráfica del programa, diseño de íconos y cualquier imagen que se utilice en el mismo.

El tester es el encargado de llevar a cabo e implementar las pruebas sobre el software programado con el objetivo de encontrar bugs en caso de que éstos existan.

Si bien el equipo tiene integrantes con roles definidos, al ser un grupo reducido de personas, el proceso de construcción del sistema es colaborativo. Todos los miembros de la CCB se involucran en todas las actividades con el objetivo de que el producto pueda ser entregado en el menor tiempo posible, siempre manteniendo la jerarquía antes listada en lo que administración de las actividades respecta

Los medios de comunicación utilizados para el contacto entre los miembros del equipo fueron las redes sociales Facebook y Whatsapp. A través de éstas se organizaron los horarios de las reuniones y se realizaron comentarios sobre la evolución del proyecto.

En la primera reunión se estableció que el equipo se reunirá tres veces por semana. La duración de las reuniones es de 4 horas.

Al inicio de cada reunión se establecen las metas diarias de la misma y se revisan las metas generales del proyecto para ver el grado de progreso.

# Documento de Requerimientos

**Requerimientos Funcionales:**

1. El sistema debe permitir al usuario ingresar una serie.
2. Cuando el usuario cargue una nueva serie, el sistema deberá generar una nueva ventana con el nombre de la nueva serie agregada.
3. El sistema no deberá permitir al usuario cargar una serie sin nombre.
4. El sistema debe contar con un botón que despliegue un listado de números para que el usuario señale la cantidad de temporadas de la serie que quiere cargar. Dicho listado de números no debe superar los 50 números.
5. El sistema debe contar con un botón que despliegue un listado de números para que el usuario señale la cantidad de capítulos de la serie que quiere cargar. Dicho listado de números no debe superar los 50 números.
6. El listado de capítulos generado por el usuario deberá contar con un campo en el que se le pueda ingresar el nombre. Por defecto, el programa generará una lista con nombres genéricos según el número de capítulo que corresponda en ese campo (Por ejemplo, para el primer capítulo, como nombre del capítulo figurará *capítulo I)*.
7. El sistema permitirá al usuario cambiar el nombre por defecto del capítulo haciendo doble click en él.
8. El sistema permitirá al usuario marcar en color verde oscuro los capítulos que ya ha visto de la serie mediante un botón que diga *Visto*.
9. El sistema marcará automáticamente en color verde oscuro aquellas temporadas en las cuales el usuario ha visto todos sus capítulos.
10. Cada vez que el programa se ejecute, se deberá abrir una ventana que muestre una lista de las series cargadas.
11. Si el programa se ejecuta y no hay series cargadas, se deberá abrir una ventana que muestre un mensaje diciendo “*Todavía no se han cargado series”*.
12. Si el usuario desea ver una serie que ya está cargada en el programa, lo hará mediante un botón que diga *Mostrar serie*.
13. Si el usuario desea ver la información que se tiene guardada sobre una serie, el programa deberá mostrar un listado con las temporadas de dicha serie en una nueva ventana con el nombre de la serie seleccionada.
14. Si el usuario desea ver una temporada de una serie que ya se encontraba cargada cuando se abrió el programa, lo hará mediante un botón que diga *Mostrar temporada.*
15. Si el usuario hace click en el botón *Mostrar temporada,* el programa deberá mostrar un listado con los capítulos de dicha temporada en la misma ventana.
16. Si el usuario desea agregar una nueva temporada a una serie ya cargada, lo hará mediante un botón que diga *Agregar temporada* en la ventana de una serie en particular.
17. Las nuevas temporadas se agregarán al final del listado de temporadas de una serie respetando la numeración de las temporadas previamente cargadas (si la última temporada cargada anteriormente fue la número 19, la nueva temporada que se agregará será la número 20).
18. Si el usuario desea agregar un nuevo capítulo a una temporada de una serie ya cargada, lo hará mediante un botón que diga *Agregar capítulo* en el listado de una temporada en particular de una serie.
19. Los nuevos capítulos se agregarán al final del listado de capítulos de una temporada respetando la numeración de los capítulos previamente cargados (si el último capítulo cargado anteriormente fue el número 20, el nuevo capítulo que se agregará será el número 20 y lo hará con el nombre por defecto).
20. El programa deberá permitir al usuario eliminar una serie ya cargada mediante un botón que diga *Eliminar serie* en la ventana que se abre cuando se ejecuta el programa.
21. Si el usuario hace click en el botón *Eliminar serie*, se deberá generar un mensaje de confirmación de esta acción en una nueva ventana diciendo *“¿Está seguro de eliminar esta serie?”.*
22. Si el usuario elimina una serie, será redirigido a la ventana que se muestra cuando se ejecuta el programa.

**Requerimientos No Funcionales:**

1. Una temporada marcada en color verde oscuro, es una temporada vista o una temporada en la que todos sus capítulos fueron vistos.
2. Cuando el usuario cargue una serie, el tiempo en que se muestra la nueva ventana no deberá superar a un (1) segundo.
3. El software no debe presentar complicaciones a la hora de su instalación.
4. El tamaño de la fuente utilizada en el software será de 11 puntos. El tipo de la misma deberá ser Verdana y en color negro.
5. Los colores en el fondo de las ventanas serán de colores claros.
6. El programa no deberá pesar más de 100 Mb.

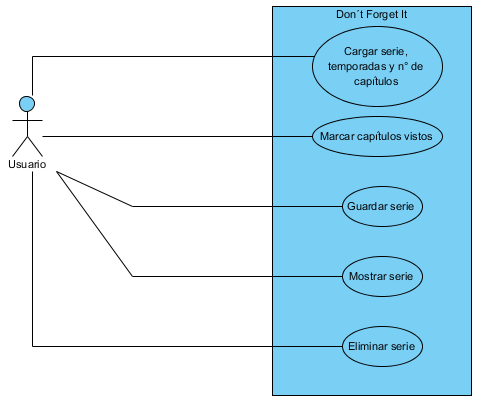
**Restricciones Generales:**

1. El software del sistema será desarrollado en Java
2. El software del sistema deberá ser soportado por Windows 7 en sus versiones de 32 y 64 bits.
3. El testing del software se realizará bajo el framework JUnit.
4. Los datos actualizados en el programa, serán almacenados en un archivo de configuración que será leído cada vez que se ejecute el programa.

**Diagrama de casos de uso:**

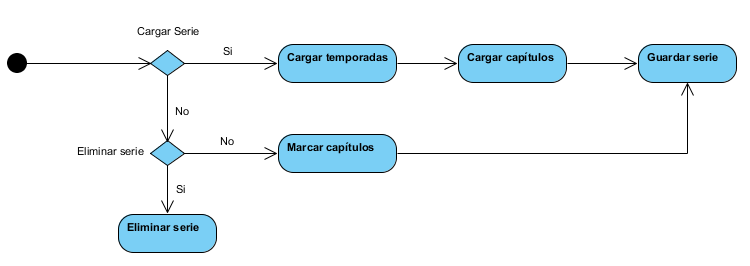
El diagrama de casos de uso es el utilizado para representar la o las diferentes interacciones entre el sistema y algún agente externo, donde cada caso de uso representa una acción discreta que implica interacción externa con el sistema.

En el siguiente diagrama, como agente externo tenemos al Usuario, que es el encargado de cargar la serie, temporadas y n° de capítulos, marcar los capítulos vistos y eliminar la información que considere inútil.



**Diagrama de Actividades:**

El diagrama de actividad es el encargado de mostrar cómo un sistema implementa su funcionalidad, en nuestro sistema cuando se quiere cargar una serie, se abrirá una ventana con los campos y opciones correspondientes para hacerlo.



**Matriz de trazabilidad:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requerimientos** | **Cargar serie, temporadas y capítulos** | **Eliminar serie** | **Marcar capítulos vistos** | **Guardar Serie** | **Mostrar Serie** |
| **RF1** | x |  |  |  |  |
| **RF2** | x |  |  |  | x |
| **RF3** | x |  |  |  |  |
| **RF4** | x |  |  |  |  |
| **RF5** | x |  |  |  |  |
| **RF6** | x |  |  |  |  |
| **RF7** | x |  |  |  |  |
| **RF8** |  |  | x |  | x |
| **RF9** |  |  | x |  |  |
| **RF10** |  |  |  |  | x |
| **RF11** |  |  |  |  | x |
| **RF12** |  |  |  |  | x |
| **RF13** |  |  |  |  | x |
| **RF14** |  |  |  |  | x |
| **RF15** |  |  |  |  | x |
| **RF16** | x |  |  |  |  |
| **RF17** | x |  |  |  |  |
| **RF18** | x |  |  |  |  |
| **RF19** | x |  |  |  |  |
| **RF20** |  | x |  |  |  |
| **RF21** |  | x |  |  |  |
| **RF22** |  | x |  |  | x |
| **RNF1** |  |  |  |  |  |
| **RNF2** |  |  |  |  |  |
| **RNF3** |  |  |  |  |  |
| **RNF4** |  |  |  |  |  |
| **RNF5** |  |  |  |  |  |

**Casos de prueba del sistema:**

**Requerimientos Funcionales**:

**RF 1:** El sistema debe permitir al usuario ingresar el nombre de una serie.

Prueba 1: a) Ejecutar el programa.

b) Hacer click en el botón *Cargar serie*.

c) Ingresar el nombre de la serie.

d) Hacer click en el botón *Listo.*

Resultado esperado: Luego de esto, el usuario podrá ver una nueva serie cargada en el programa.

**RF 2:** Cuando el usuario cargue una nueva serie, el sistema deberá generar una nueva ventana con el nombre de la nueva serie agregada.

Prueba 2: a) Ejecutar el programa.

b) Hacer click en el botón *Cargar serie.*

c) Ingresar el nombre de la serie.

d) Hacer click en el botón *Listo.*

Resultado esperado: Luego de esto, aparecerá una nueva ventana titulada con el nombre de la serie.

**RF 3:** El sistema no deberá permitir al usuario cargar una serie sin nombre.

Prueba 3: a) Ejecutar el programa.

b) Hacer click en el botón *Cargar serie.*

c) Hacer click en el botón *Listo*.

Resultado esperado: Luego de esto, aparecerá un mensaje de error que diga *“No se puede cargar una serie sin nombre, ingrese el nombre de la serie e intente nuevamente”.*

**RF 4:** El sistema debe contar con un botón que despliegue un listado de números para que el usuario señale la cantidad de temporadas de la serie que quiere cargar. Dicho listado de números no debe superar los 50 números.

Prueba 4: a) Ejecutar el programa.

b) Hacer click en el botón *Cargar serie*.

c) Ingresar el nombre de la serie.

d) Hacer click en el botón *Listo.*

e) En la nueva ventana, desplegar el menú *Número de temporadas*.

Resultado esperado: Luego de esto, aparecerá un listado de números empezando por el 1 y terminando en el 50 en orden creciente, en donde el usuario podrá seleccionar algún número.

**RF 5:** El sistema debe contar con un botón que despliegue un listado de números para que el usuario señale la cantidad de capítulos de la serie que quiere cargar. Dicho listado de números no debe superar los 50 números.

Prueba 5: a) Ejecutar el programa.

b) Hacer click en el botón *Cargar serie*.

c) Ingresar el nombre de la serie.

d) Hacer click en el botón *Listo.*

e) En la nueva ventana, desplegar el menú *Número de temporadas* y seleccionar un número de temporadas.

d) Desplegar el menú *Número de capítulos por temporada*.

Resultado esperado: Luego de esto, aparecerá en la misma ventana un listado de números empezado por el 1 y terminando en el 50 en orden creciente, en donde el usuario podrá seleccionar algún número.

**RF 6:** El listado de capítulos generado por el usuario deberá contar con un campo en el que se le pueda ingresar el nombre. Por defecto, el programa generará una lista con nombres genéricos según el número de capítulo que corresponda en ese campo (Por ejemplo, para el primer capítulo, como nombre del capítulo figurará *capítulo I)*.

Prueba 6: a) Ejecutar el programa.

b) Hacer click en el botón *Cargar serie*.

c) Ingresar el nombre de la serie.

d) Hacer click en el botón *Listo.*

e) En la nueva ventana, desplegar el menú *Número de temporadas* y seleccionar un número de temporadas.

d) Desplegar el menú *Número de capítulos por temporada*.

f) Seleccionar un número de capítulos por temporada.

Resultado esperado: Luego de esto, aparecerá en la misma ventana el listado de temporadas con sus capítulos según el número que el usuario haya señalado por cada temporada, el nombre de cada capítulo será el determinado por defecto por el sistema. Para el capítulo 1 será *“Capítulo I”*, para el capítulo 2 será “*Capítulo II”* y así sucesivamente.

**RF 7:** El sistema permitirá al usuario cambiar el nombre por defecto del capítulo haciendo doble click en él.

Prueba 7: a) Ejecutar el programa.

b) Hacer click en el botón *Cargar serie*.

c) Ingresar el nombre de la serie.

d) Hacer click en el botón *Listo.*

e) En la nueva ventana, desplegar el menú *Número de temporadas* y seleccionar un número de temporadas.

d) Desplegar el menú *Número de capítulos por temporada*.

f) Seleccionar un número de capítulos por temporada.

e) Hacer doble click en el nombre del capítulo por defecto.

g) Escribir el nombre del capítulo.

Resultado esperado: Luego de esto, se reemplazará el nombre del capítulo por defecto por el texto que el usuario haya ingresado.

**RF 8:** El sistema permitirá al usuario marcar en color verde oscuro los capítulos que ya ha visto de la serie mediante un botón que diga *Visto*.

Prueba 8: a) Ejecutar el programa.

b) Hacer click en el botón *Cargar serie*.

c) Ingresar el nombre de la serie.

d) Hacer click en el botón *Listo.*

e) En la nueva ventana, desplegar el menú *Número de temporadas* y seleccionar un número de temporadas.

d) Desplegar el menú *Número de capítulos por temporada*.

f) Seleccionar un número de capítulos por temporada.

g) Hacer click en el botón *Visto* de algún capítulo.

Resultado esperado: Luego de esto, se marcará en color verde oscuro el capítulo que se ha seleccionado como visto.

**RF 9:** El sistema marcará automáticamente en color verde oscuro aquellas temporadas en las cuales el usuario ha visto todos sus capítulos.

Prueba 9: a) Ejecutar el programa.

b) Hacer click en el botón *Cargar serie*.

c) Ingresar el nombre de la serie.

d) Hacer click en el botón *Listo.*

e) En la nueva ventana, desplegar el menú *Número de temporadas* y seleccionar un número de temporadas.

d) Desplegar el menú *Número de capítulos por temporada*.

f) Seleccionar un número de capítulos por temporada.

g) Hacer click en el botón *Visto* de todos los capítulos.

Resultado esperado: Luego de esto, el usuario podrá observar la temporada en la que se han marcado todos sus capítulos como vistos marcada en color verde oscuro.

**RF 10:** Cada vez que el programa se ejecute, se deberá abrir una ventana que muestre una lista de las series cargadas.

Prueba 10: a) Ejecutar el programa.

Resultado esperado: Luego de esto, el usuario podrá observar un listado con las series cargadas en una ventana.

**RF 11:** Si el programa se ejecuta y no hay series cargadas, se deberá abrir una ventana que muestre un mensaje diciendo “*Todavía no se han cargado series”*.

Prueba 11: a) Ejecutar el programa.

Resultado esperado: Luego de esto, el usuario podrá observar un mensaje diciendo “*Todavía no se han cargado series”* en una ventana.

**RF 12:** Si el usuario desea ver una serie que ya está cargada en el programa, lo hará mediante un botón que diga *Mostrar serie* y en una nueva ventana.

Prueba 12: a) Ejecutar el programa.

b) Hacer click en el botón *Mostrar serie* de alguna de las series cargadas.

Resultado esperado: Luego de esto, en una ventana nueva el usuario podrá ver la información que se ha guardado sobre esa serie.

**RF 13:** Si el usuario desea ver la información que se tiene guardada sobre una serie, el sistema deberá mostrar un listado con las temporadas de dicha serie en una nueva ventana con el nombre de la serie seleccionada.

Prueba 13: a) Ejecutar el programa.

b) Hacer click en el botón *Mostrar serie* de alguna de las series cargadas.

Resultado esperado: Luego de esto, en una ventana nueva el usuario podrá ver una lista con las temporadas de la serie seleccionada para ver, y en el caso de existir, las marcas correspondientes a capítulos y temporadas vistas.

**RF 14:** Si el usuario desea ver una temporada de una serie que ya se encontraba cargada cuando se abrió el programa, lo hará mediante un botón que diga *Mostrar temporada.*

Prueba 14: a) Ejecutar el programa.

b) Hacer click en el botón *Mostrar serie* de alguna de las series cargadas.

c) Hacer click en el botón *Mostrar temporada* de alguna de las temporadas de la serie que se ha seleccionado para mostrar.

Resultado esperado: Luego de esto, el usuario podrá ver la información de la temporada que se ha seleccionado para mostrar.

**RF 15:** Si el usuario hace click en el botón *Mostrar temporada,* el programa deberá mostrar un listado con los capítulos de dicha temporada en la misma ventana.

Prueba 15: a) Ejecutar el programa.

b) Hacer click en el botón *Mostrar serie*  de alguna de las series cargadas.

c) Hacer click en el botón *Mostrar temporada* de alguna de las temporadas de la serie que se ha seleccionado para mostrar.

Resultado esperado: Luego de esto, el usuario podrá ver un listado con los capítulos y su respectivo estado de la temporada seleccionada para mostrar en la misma ventana.

**RF 16:** Si el usuario desea agregar una nueva temporada a una serie ya cargada, lo hará mediante un botón que diga *Agregar temporada* en la ventana de una serie en particular.

Prueba 16: a) Ejecutar el programa.

b) Hacer click en el botón *Mostrar serie* de alguna de las series cargadas.

c) Hacer click en el botón *Agregar temporada.*

Resultado esperado: Luego de esto, el usuario podrá ver una nueva temporada vacía agregada al listado de temporadas previamente cargadas.

**RF 17:** Las nuevas temporadas se agregarán al final del listado de temporadas de una serie respetando la numeración de las temporadas previamente cargadas (si la última temporada cargada anteriormente fue la número 19, la nueva temporada que se agregará será la número 20).

Prueba 17: a) Ejecutar el programa.

b) Hacer click en el botón *Mostrar serie* de alguna de las series cargadas.

c) Hacer click en el botón *Agregar temporada.*

Resultado esperado: Luego de esto, el usuario podrá ver una nueva temporada vacía agregada al final del listado de temporadas previamente cargadas. Si la última temporada previamente cargada fue la número 19, la nueva temporada cargada será la número 20.

**RF 18:** Si el usuario desea agregar un nuevo capítulo a una temporada de una serie ya cargada, lo hará mediante un botón que diga *Agregar capítulo* en el listado de una temporada en particular de una serie.

Prueba 18: a) Ejecutar el programa.

b) Hacer click en el botón *Mostrar serie* de alguna de las series cargadas.

c) Hacer click en el botón *Mostrar temporada*.

d) Hacer click en el botón *Agregar capítulo.*

Resultado esperado: Luego de esto, el usuario podrá ver un nuevo capítulo agregado al listado de capítulos previamente cargados.

**RF 19:** Los nuevos capítulos se agregarán al final del listado de capítulos de una temporada respetando la numeración de los capítulos previamente cargados (si el último capítulo cargado anteriormente fue el número 20, el nuevo capítulo que se agregará será el número 20 y lo hará con el nombre por defecto).

Prueba 19: a) Ejecutar el programa.

b) Hacer click en el botón *Mostrar serie* de alguna de las series cargadas.

c) Hacer click en el botón *Mostrar temporada*.

d) Hacer click en el botón *Agregar capítulo.*

Resultado esperado: Luego de esto, el usuario podrá ver un nuevo capítulo con el nombre por defecto agregado al final del listado de capítulos previamente cargados. Si el último capítulo previamente cargado fue el número 19, el nuevo capítulo cargado será el número 20.

**RF 20:** El programa deberá permitir al usuario eliminar una serie ya cargada mediante un botón que diga *Eliminar serie.*

Prueba 20: a) Ejecutar el programa.

b) Hacer click en el botón *Mostrar serie* de alguna de las series cargadas.

c) Hacer click en el botón *Eliminar serie*.

Resultado esperado: Luego de esto, el usuario podrá ver que la serie ha sido eliminada del listado de series cargadas en el programa.

**RF 21:** Si el usuario hace click en el botón *Eliminar serie*, se deberá generar un mensaje de confirmación de esta acción en una nueva ventana diciendo *“¿Está seguro de eliminar esta serie?”.*

Prueba 21: a) Ejecutar el programa.

b) Hacer click en el botón *Mostrar serie* de alguna de las series cargadas.

c) Hacer click en el botón *Eliminar serie*.

Resultado esperado: Luego de esto, el usuario podrá ver una nueva ventana en donde le preguntarán si confirma la acción de eliminar la serie o no.

**RF 22:** Si el usuario elimina una serie, será redirigido a la ventana que se muestra cuando se ejecuta el programa.

Prueba 22: a) Ejecutar el programa.

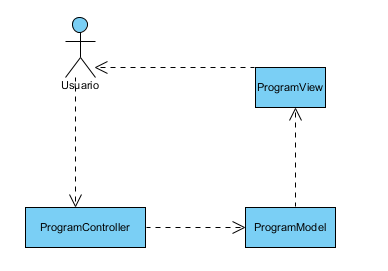
b) Hacer click en el botón *Mostrar serie* de alguna de las series cargadas.

c) Hacer click en el botón *Eliminar serie*.

d) Hacer click en el botón *Sí* en la confirmación de la acción.

Resultado esperado: Luego de esto, el usuario podrá ver que será redirigido a la ventana que aparece cuando ejecuta el programa.

**Diagrama de Arquitectura preliminar:**



# Arquitectura del Sistema

El patrón de arquitectura utilizado en esta situación fue le Model View Cotroller (MVC) el cual separa la aplicación en 3 componentes, Vistas, Modelos y Controladores permitiéndonos así que cuando hacemos un cambio en alguna parte de nuestro código o agregar alguna funcionalidad nueva, esto no afecte otra parte del mismo por lo tanto el resto de la aplicación debería permanecer intacta, este modelo nos permite cumplir de manera sencilla con los requerimientos no funcionales como ser el mantenimiento.

Decidimos optar por aplicar esta arquitectura debido a que recibimos una aplicación ya implementada y debíamos agregarle una nueva funcionalidad a la misma.

Podemos ver con la siguiente imagen como funciona este patrón de arquitectura:

