



Detección y Control de Puntero Mediante la Cámara de un Smartphone para Pantallas LED

Autor:

Bioing. Zenklusen Kevin Nicolás

Director:

PhD. Ms. Argoty Pulido Camilo Enrique (A confirmar) (FIUBA)

*Esta planificación fue realizada en el curso de Gestión de proyectos
entre el 15 de octubre de 2024 y el 3 de diciembre de 2024.*

Índice

| | |
|---|----|
| 1. Descripción técnica-conceptual del proyecto a realizar | 5 |
| 2. Identificación y análisis de los interesados | 7 |
| 3. Propósito del proyecto | 7 |
| 4. Alcance del proyecto | 7 |
| 5. Supuestos del proyecto. | 8 |
| 6. Requerimientos | 9 |
| 7. Historias de usuarios (<i>Product backlog</i>). | 9 |
| 8. Entregables principales del proyecto | 10 |
| 9. Desglose del trabajo en tareas | 10 |
| 10. Diagrama de Activity On Node. | 11 |
| 11. Diagrama de Gantt | 11 |
| 12. Presupuesto detallado del proyecto | 15 |
| 13. Gestión de riesgos | 15 |
| 14. Gestión de la calidad | 16 |
| 15. Procesos de cierre | 17 |

Registros de cambios

| Revisión | Detalles de los cambios realizados | Fecha |
|----------|--|-----------------------|
| 0 | Creación del documento | 15 de octubre de 2024 |
| 1 | Se completa hasta el punto 5 inclusive | 29 de octubre de 2024 |

Acta de constitución del proyecto

Buenos Aires, 15 de octubre de 2024

Por medio de la presente se acuerda con el Bioing. Zenklusen Kevin Nicolás que su Trabajo Final de la Carrera de Especialización en Sistemas Embebidos se titulará “Detección y Control de Puntero Mediante la Cámara de un Smartphone para Pantallas LED” y consistirá en el desarrollo de una app para smartphones que permite controlar el cursor del mouse en PC con SO Windows 10/11 en fomra inalámbrica. El trabajo tendrá un presupuesto preliminar estimado de 600 horas, con fecha de inicio el 15 de octubre de 2024 y fecha de presentación pública el 15 de mayo de 2025.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

-
Emprendimiento personal

PhD. Ms. Argoty Pulido Camilo Enrique (A confirmar)
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

El proyecto propone el desarrollo de un sistema que permite a los usuarios interactuar con pantallas LED utilizando la cámara de un smartphone para apuntar y controlar el puntero del mouse. Con aplicaciones orientadas principalmente al ámbito de videojuegos de disparos, como Duck Hunt (Figura 1) o House of the Dead, este sistema busca emular la funcionalidad de las clásicas lightguns (Figura 2), adaptándola a tecnologías de visualización modernas, específicamente las pantallas LED. Aprovechando algoritmos de visión por computadora en los dispositivos móviles, el sistema permitirá identificar la ubicación de la pantalla y realizar interacciones precisas en la misma mediante gestos en el smartphone.



Figura 1. Pantalla del videojuego *Duck Hunt* (1984), un clásico juego de disparos que utilizaba la tecnología de lightgun.



Figura 2. La NES Zapper, una lightgun usada para interactuar con juegos de disparo en pantallas CRT.

Las "lightguns" fueron dispositivos creados para ampliar la interacción con los videojuegos, permitiendo una experiencia de disparo inmersiva en pantallas CRT al detectar de manera precisa la dirección en que el usuario apuntaba un periférico con forma de pistola. Sin embargo, el principio de funcionamiento de las pantallas LED difiere del de las CRT, por lo que la tecnología de las "lightguns" no es directamente aplicable. Actualmente, existen algunos dispositivos que utilizan sensores de movimiento o cámaras externas para realizar tareas de detección de puntería,

aunque estos suelen ser costosos o requieren configuraciones complejas. Frente a este panorama, el proyecto propone una solución más accesible y fácil de configurar, integrando el uso de smartphones y los sensores que ya poseen, lo que elimina la necesidad de hardware adicional.

Para el funcionamiento inicial del sistema, se propone desarrollar una aplicación para smartphone que ejecute la detección de la pantalla LED y procese la información de ubicación del punto de puntería. Los datos obtenidos serán enviados a una computadora con sistema operativo Windows 10/11, la cual interpretará esta información y permitirá controlar el movimiento del puntero del mouse en la pantalla de la PC. Adicionalmente, se implementarán funciones para realizar clic derecho e izquierdo mediante toques en la pantalla del teléfono, proporcionando una experiencia de interacción completa.

El sistema propuesto destaca al ofrecer una alternativa económica y adaptable a distintos tamaños y resoluciones de pantallas LED, utilizando un smartphone como controlador interactivo en lugar de hardware especializado. Los principales desafíos del proyecto incluyen la precisión en la detección de la pantalla, el cálculo de posición relativo en tiempo real y la transmisión inalámbrica de datos para que el sistema responda de manera fluida y sin retrasos notables. Estos desafíos se abordarán mediante la implementación de algoritmos de visión por computadora para identificar la pantalla y se considerará el uso de sensores de movimiento del smartphone para mejorar la exactitud de los movimientos del puntero.

Los principales componentes y funcionalidades del sistema incluyen:

- Detección de pantalla LED: El sistema debe reconocer la pantalla en la imagen capturada por la cámara del smartphone mediante técnicas de visión por computadora y delimitarla con un "bounding box".
- Detección de puntería: Se evaluará si el centro de la imagen capturada por la cámara está dentro del "bounding box", lo que indicará si el usuario está apuntando hacia la pantalla.
- Cálculo de posición relativa: Cuando el smartphone esté apuntando a la pantalla, se calculará la posición exacta donde el usuario está mirando dentro de la misma mediante proporcionalidad.
- Interacción adicional: Las interacciones con la pantalla táctil del teléfono generarán información adicional que será utilizada para el control de la PC. Se considerará el uso de los sensores de movimiento del smartphone para mejorar la precisión.
- Envío de datos: La información de la ubicación del punto y los datos agregados serán enviados a una computadora mediante una conexión inalámbrica, pudiendo ser una conexión WiFi o Bluetooth. Serán recibidos por un software específico en el PC que moverá el puntero del mouse en la pantalla en la posición calculada.
- Funcionalidad de click: El sistema permitirá simular el 'click' del mouse mediante el toque en la pantalla del smartphone.

En la figura 3 se presenta el diagrama en bloques del sistema. Se observa que los datos de posición obtenidos por la cámara se procesan para identificar la ubicación de la pantalla y, posteriormente, para transmitir la información a la computadora.

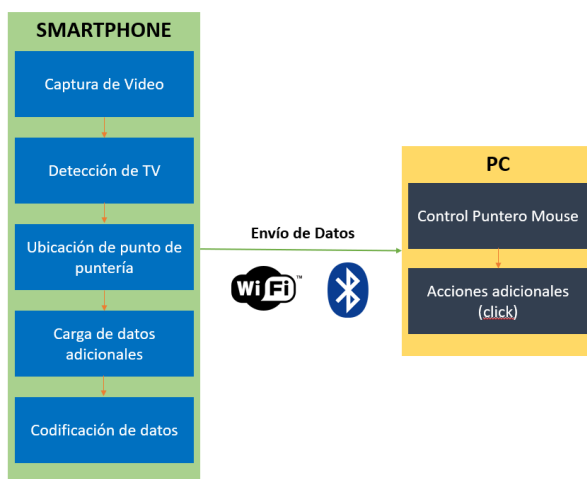


Figura 3. Diagrama de bloques del sistema.

2. Identificación y análisis de los interesados

| Rol | Nombre y Apellido | Organización | Puesto |
|---------------|---|-------------------------|----------------------------|
| Cliente | - | Emprendimiento personal | - |
| Responsable | Bioing. Zenklusen Kevin Nicolás | FIUBA | Alumno |
| Orientador | PhD. Ms. Argoty Pulido Camilo Enrique (A confirmar) | FIUBA | Director del Trabajo Final |
| Usuario final | Usuario de App | - | - |

- Orientador: El PhD. Ms. Argoty Pulido Camilo Enrique (A confirmar) es docente del posgrado de especialización en inteligencia artificial, posee conocimientos en la temática y va a ayudar con la definición de los requerimientos y el desarrollo del software.

3. Propósito del proyecto

Desarrollar un sistema innovador que permita a los usuarios interactuar con pantallas LED utilizando la cámara de un smartphone para apuntar y controlar el puntero del mouse, emulando la experiencia de videojuegos clásicos de disparo. A través de algoritmos de visión por computadora el sistema busca ofrecer una forma precisa y entretenida de interactuar con la pantalla, adaptando la funcionalidad de las lightguns tradicionales a tecnologías modernas y accesibles, y brindando una nueva experiencia de juego que combina nostalgia y avance tecnológico.

4. Alcance del proyecto

El proyecto incluye:

- Desarrollo de una aplicación móvil para smartphones (Android y/o iOS [Sujeto a disponibilidad]) que permita detectar pantallas LED.
- Implementación de algoritmos de visión por computadora para identificar la ubicación de la pantalla.
- Control del puntero del mouse a través del smartphone, incluyendo clics izquierdo y derecho.
- Pruebas de usabilidad para asegurar la interacción precisa con las pantallas LED.
- Documentación técnica del sistema.
- Aplicación para sistemas operativos Windows 10/11 para la comunicación de datos y el control del cursor.

El presente proyecto no incluye:

- Desarrollo de hardware especializado para la detección de pantallas.
- Compatibilidad con sistemas operativos distintos a Windows.
- Desarrollo de videojuegos específicos o contenido adicional más allá de la funcionalidad principal.
- Mantenimiento o actualizaciones de la aplicación.
- Garantía de rendimiento en todos los tipos de pantallas LED existentes en el mercado.
- Garantía de rendimiento en todas las PCs existentes en el mercado.
- Garantía de funcionamiento en todos los smartphones existentes en el mercado.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- Se dispondrá de un desarrollador con las habilidades necesarias en programación, visión por computadora y diseño de interfaces.
- Habrá acceso a recursos tecnológicos adecuados, como smartphones compatibles y computadoras con sistemas operativos Windows para pruebas y desarrollo.
- El tiempo asignado para el desarrollo y pruebas del proyecto será suficiente para cumplir con los plazos establecidos sin comprometer la calidad del producto.
- Se garantizará la disponibilidad de datos y pantallas LED para realizar pruebas de funcionalidad y usabilidad del sistema.
- Los requisitos técnicos de las plataformas y dispositivos serán accesibles y manejables, sin impedimentos significativos para la implementación del software.
- La legislación y normativas relacionadas con el uso de tecnologías móviles y videojuegos permitirán la implementación del proyecto sin restricciones.

6. Requerimientos

Los requerimientos deben enumerarse y de ser posible estar agrupados por afinidad, por ejemplo:

1. Requerimientos funcionales:
 - 1.1. El sistema debe...
 - 1.2. Tal componente debe...
 - 1.3. El usuario debe poder...
2. Requerimientos de documentación:
 - 2.1. Requerimiento 1.
 - 2.2. Requerimiento 2 (prioridad menor)
3. Requerimiento de testing...
4. Requerimientos de la interfaz...
5. Requerimientos interoperabilidad...
6. etc...

Leyendo los requerimientos se debe poder interpretar cómo será el proyecto y su funcionalidad.

Indicar claramente cuál es la prioridad entre los distintos requerimientos y si hay requerimientos opcionales.

!!!No olvidarse de que los requerimientos incluyen a las regulaciones y normas vigentes!!!

Y al escribirlos seguir las siguientes reglas:

- Ser breve y conciso (nadie lee cosas largas).
- Ser específico: no dejar lugar a confusiones.
- Expresar los requerimientos en términos que sean cuantificables y medibles.

7. Historias de usuarios (*Product backlog*)

Descripción: en esta sección se deben incluir las historias de usuarios y su ponderación (*history points*). Recordar que las historias de usuarios son descripciones cortas y simples de una característica contada desde la perspectiva de la persona que desea la nueva capacidad, generalmente un usuario o cliente del sistema. La ponderación es un número entero que representa el tamaño de la historia comparada con otras historias de similar tipo.

Se debe indicar explícitamente el criterio para calcular los *story points* de cada historia.

El formato propuesto es:

1. “Como [rol] quiero [tal cosa] para [tal otra cosa].”

Story points: 8 (complejidad: 3, dificultad: 2, incertidumbre: 3)

8. Entregables principales del proyecto

Los entregables del proyecto son (ejemplo):

- Manual de usuario.
- Diagrama de circuitos esquemáticos.
- Código fuente del firmware.
- Diagrama de instalación.
- Memoria del trabajo final.
- etc...

9. Desglose del trabajo en tareas

El WBS debe tener relación directa o indirecta con los requerimientos. Son todas las actividades que se harán en el proyecto para dar cumplimiento a los requerimientos. Se recomienda mostrar el WBS mediante una lista indexada:

1. Grupo de tareas 1 (suma h)

1.1. Tarea 1 (tantas h)

1.2. Tarea 2 (tantas h)

1.3. Tarea 3 (tantas h)

2. Grupo de tareas 2 (suma h)

2.1. Tarea 1 (tantas h)

2.2. Tarea 2 (tantas h)

2.3. Tarea 3 (tantas h)

3. Grupo de tareas 3 (suma h)

3.1. Tarea 1 (tantas h)

3.2. Tarea 2 (tantas h)

3.3. Tarea 3 (tantas h)

3.4. Tarea 4 (tantas h)

3.5. Tarea 5 (tantas h)

Cantidad total de horas: tantas.

¡Importante! la unidad de horas es h y va separada por espacio del número. Es incorrecto escribir “23hs”.

Se recomienda que no haya ninguna tarea que lleve más de 40 h. De ser así se recomienda dividirla en tareas de menor duración.

10. Diagrama de Activity On Node

Armaz el AoN a partir del WBS definido en la etapa anterior.

Una herramienta simple para desarrollar los diagramas es el Draw.io (<https://app.diagrams.net/>). Draw.io

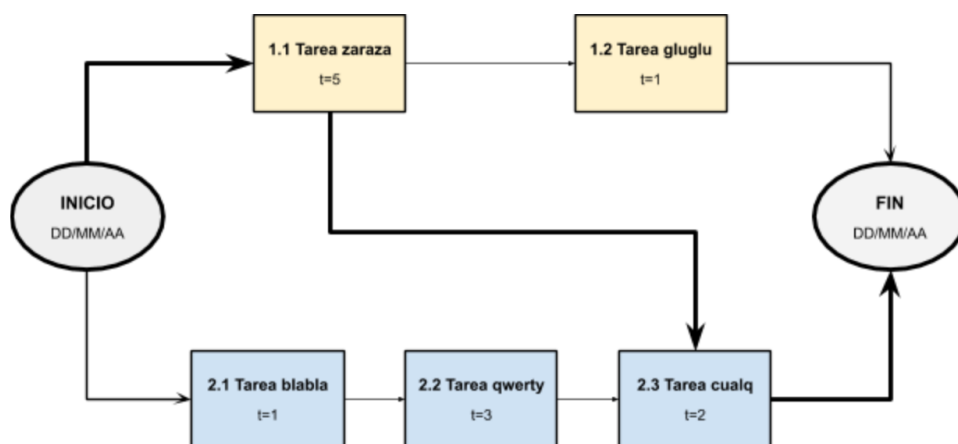


Figura 4. Diagrama de *Activity on Node*.

Indicar claramente en qué unidades están expresados los tiempos. De ser necesario indicar los caminos semi críticos y analizar sus tiempos mediante un cuadro. Es recomendable usar colores y un cuadro indicativo describiendo qué representa cada color.

11. Diagrama de Gantt

Existen muchos programas y recursos *online* para hacer diagramas de Gantt, entre los cuales destacamos:

- Planner
- GanttProject
- Trello + *plugins*. En el siguiente link hay un tutorial oficial:
<https://blog.trello.com/es/diagrama-de-gantt-de-un-proyecto>
- Creately, herramienta online colaborativa.
<https://creately.com/diagram/example/ieb3p3ml/LaTeX>

- Se puede hacer en latex con el paquete *pgfgantt*
<http://ctan.dcc.uchile.cl/graphics/pgf/contrib/pgfgantt/pgfgantt.pdf>

Pegar acá una captura de pantalla del diagrama de Gantt, cuidando que la letra sea suficientemente grande como para ser legible. Si el diagrama queda demasiado ancho, se puede pegar primero la “tabla” del Gantt y luego pegar la parte del diagrama de barras del diagrama de Gantt.

Configurar el software para que en la parte de la tabla muestre los códigos del EDT (WBS).
Configurar el software para que al lado de cada barra muestre el nombre de cada tarea.
Revisar que la fecha de finalización coincida con lo indicado en el Acta Constitutiva.

En la figura 5, se muestra un ejemplo de diagrama de gantt realizado con el paquete de *pgfgantt*. En la plantilla pueden ver el código que lo genera y usarlo de base para construir el propio.

Las fechas pueden ser calculadas utilizando alguna de las herramientas antes citadas. Sin embargo, el siguiente ejemplo fue elaborado utilizando [esta hoja de cálculo](#).

Es importante destacar que el ancho del diagrama estará dado por la longitud del texto utilizado para las tareas (Ejemplo: tarea 1, tarea 2, etcétera) y el valor $x\ unit$. Para mejorar la apariencia del diagrama, es necesario ajustar este valor y, quizás, acortar los nombres de las tareas.

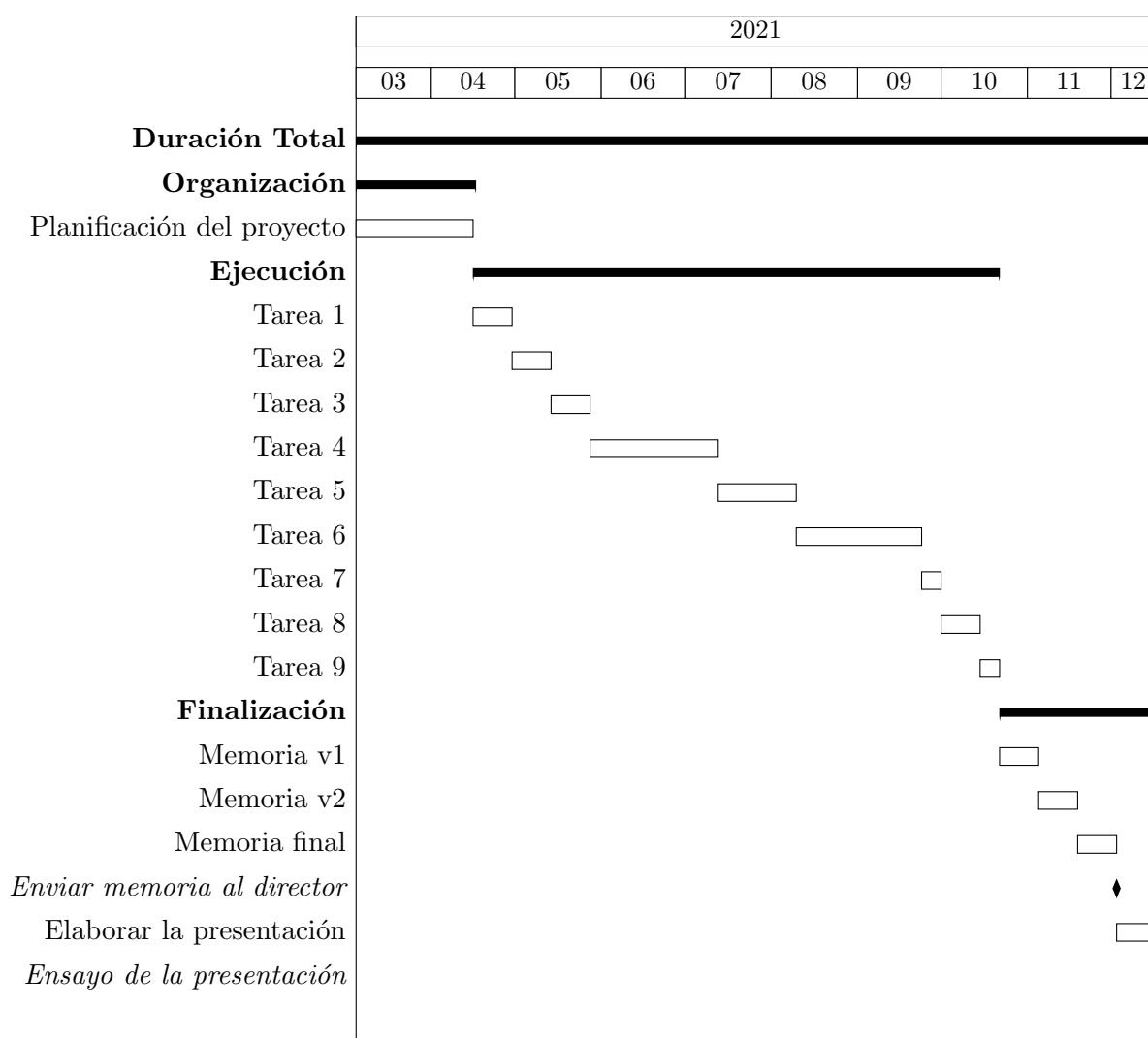


Figura 5. Diagrama de gantt de ejemplo



Figura 6. Ejemplo de diagrama de Gantt (apaisado).

12. Presupuesto detallado del proyecto

Si el proyecto es complejo entonces separarlo en partes:

- Un total global, indicando el subtotal acumulado por cada una de las áreas.
- El desglose detallado del subtotal de cada una de las áreas.

IMPORTANTE: No olvidarse de considerar los **COSTOS INDIRECTOS**.

Incluir la aclaración de si se emplea como moneda el peso argentino (ARS) o si se usa moneda extranjera (USD, EUR, etc). Si es en moneda extranjera se debe indicar la tasa de conversión respecto a la moneda local en una fecha dada.

| COSTOS DIRECTOS | | | |
|-------------------|----------|----------------|-------------|
| Descripción | Cantidad | Valor unitario | Valor total |
| | | | |
| | | | |
| | | | |
| | | | |
| SUBTOTAL | | | |
| COSTOS INDIRECTOS | | | |
| Descripción | Cantidad | Valor unitario | Valor total |
| | | | |
| | | | |
| | | | |
| | | | |
| SUBTOTAL | | | |
| TOTAL | | | |

13. Gestión de riesgos

a) Identificación de los riesgos (al menos cinco) y estimación de sus consecuencias:

Riesgo 1: detallar el riesgo (riesgo es algo que si ocurre altera los planes previstos de forma negativa)

- Severidad (S): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2:

- Severidad (S): X.
Justificación...

- Ocurriencia (O): Y.
Justificación...

Riesgo 3:

- Severidad (S): X.
Justificación...
- Ocurriencia (O): Y.
Justificación...

b) Tabla de gestión de riesgos: (El RPN se calcula como $RPN=S \times O$)

| Riesgo | S | O | RPN | S* | O* | RPN* |
|--------|---|---|-----|----|----|------|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

Criterio adoptado:

Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a...

Nota: los valores marcados con (*) en la tabla corresponden luego de haber aplicado la mitigación.

c) Plan de mitigación de los riesgos que originalmente excedían el RPN máximo establecido:

Riesgo 1: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).
Nueva asignación de S y O, con su respectiva justificación:

- Severidad (S*): mientras más severo, más alto es el número (usar números del 1 al 10). Justificar el motivo por el cual se asigna determinado número de severidad (S).
- Probabilidad de ocurrencia (O*): mientras más probable, más alto es el número (usar del 1 al 10). Justificar el motivo por el cual se asigna determinado número de (O).

Riesgo 2: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

Riesgo 3: plan de mitigación (si por el RPN fuera necesario elaborar un plan de mitigación).

14. Gestión de la calidad

Elija al menos diez requerimientos que a su criterio sean los más importantes/críticos/que aportan más valor y para cada uno de ellos indique las acciones de verificación y validación que permitan asegurar su cumplimiento.

- Req #1: copiar acá el requerimiento con su correspondiente número.

- Verificación para confirmar si se cumplió con lo requerido antes de mostrar el sistema al cliente. Detallar.
- Validación con el cliente para confirmar que está de acuerdo en que se cumplió con lo requerido. Detallar.

Tener en cuenta que en este contexto se pueden mencionar simulaciones, cálculos, revisión de hojas de datos, consulta con expertos, mediciones, etc.

Las acciones de verificación suelen considerar al entregable como “caja blanca”, es decir se conoce en profundidad su funcionamiento interno.

En cambio, las acciones de validación suelen considerar al entregable como “caja negra”, es decir, que no se conocen los detalles de su funcionamiento interno.

15. Procesos de cierre

Establecer las pautas de trabajo para realizar una reunión final de evaluación del proyecto, tal que contemple las siguientes actividades:

- Pautas de trabajo que se seguirán para analizar si se respetó el Plan de Proyecto original:
 - Indicar quién se ocupará de hacer esto y cuál será el procedimiento a aplicar.
- Identificación de las técnicas y procedimientos útiles e inútiles que se emplearon, los problemas que surgieron y cómo se solucionaron:
 - Indicar quién se ocupará de hacer esto y cuál será el procedimiento para dejar registro.
- Indicar quién organizará el acto de agradecimiento a todos los interesados, y en especial al equipo de trabajo y colaboradores:
 - Indicar esto y quién financiará los gastos correspondientes.