



Visión computacional estereoscópica

Autor:

Ing. Ariel Salassa

Director:

Esp. Ing. Hernán Contigiani (Globant)

*Esta planificación fue realizada en el curso de Gestión de proyectos
entre el 26 de abril de 2022 y el 14 de junio de 2022.*

Índice

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

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0.0	Creación del documento	26/04/2022
1.0	Se completa hasta la sección 5 inclusive	08/05/2022
1.1	Correcciones de la versión 1.0	20/05/2022
2.0	Se completa hasta la sección 9 inclusive	22/05/2022
2.1	Correcciones de la versión 2.0	26/05/2022
3.0	Se completa hasta sección 12 inclusive	29/05/2022
3.1	Correcciones de la versión 3.0	03/06/2022
4.0	Se completa hasta sección 12 inclusive	04/06/2022
4.1	Correcciones de la versión 4.0	06/06/2022

Acta de constitución del proyecto

Buenos Aires, 26 de abril de 2022

Por medio de la presente se acuerda con el Ing. Ariel Salassa que su Trabajo Final de la Carrera de Especialización en Inteligencia Artificial se titulará “Visión computacional estereoscópica” y consistirá en investigar el estado del arte de la visión computacional realizada con más de una cámara como entrada, de modo que se puedan tomar decisiones en un ambiente 3D simulado, con fecha de inicio 26 de abril de 2022 y fecha de presentación pública diciembre de 2022. Se estima que el valor aproximado del proyecto es de *15405 USD* y que tendrá una duración de 588 hs aproximadamente.

Se adjunta a esta acta la planificación inicial.

Ariel Lutenberg
Director posgrado FIUBA

Esp. Ing. Hernán Contigiani
Globan

Esp. Ing. Hernán Contigiani
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

La visión computacional en pocas palabras consta de cámaras que entregan fotogramas a un sistema que los toma y los procesa teniendo en cuenta una calibración previa.

La idea del proyecto se basa en el uso de al menos dos cámaras conectadas a un mismo sistema de visión que sea capaz de procesar las imágenes de cada una de las cámaras permitiendo estimar atributos de objetos tales como altura, profundidad y volumen.

Actualmente existen cámaras estereoscópicas integradas en un único dispositivo (Figura 1) que realizan dichas tareas, permitiendo agregar funcionalidades como la navegación de robots o asistencia a conductores. Las cámaras estereoscópicas integradas tienen la ventaja de que el mismo hardware y firmware realiza la inferencia, pero tienen la desventaja de que la distancia entre cámaras es fija y esto no permite tomar espectros 3D en infraestructuras preexistentes.

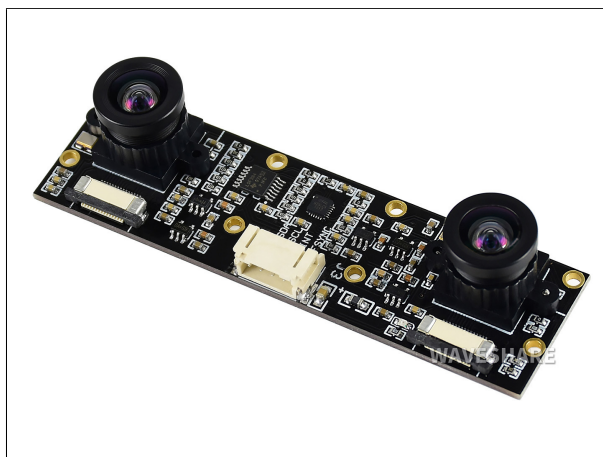


Figura 1. Cámara estereoscópica.

Tener un sistema de cámaras conectado a un servidor puede otorgar mucha más precisión como así también, mejor alcance de la solución. Incluso se podría evaluar la posibilidad de hacer visión computacional estereoscópica con cámaras que no están fijas.

Las partes que conforman un sistema de visión computacional se pueden observar en el siguiente diagrama de la Figura 2.

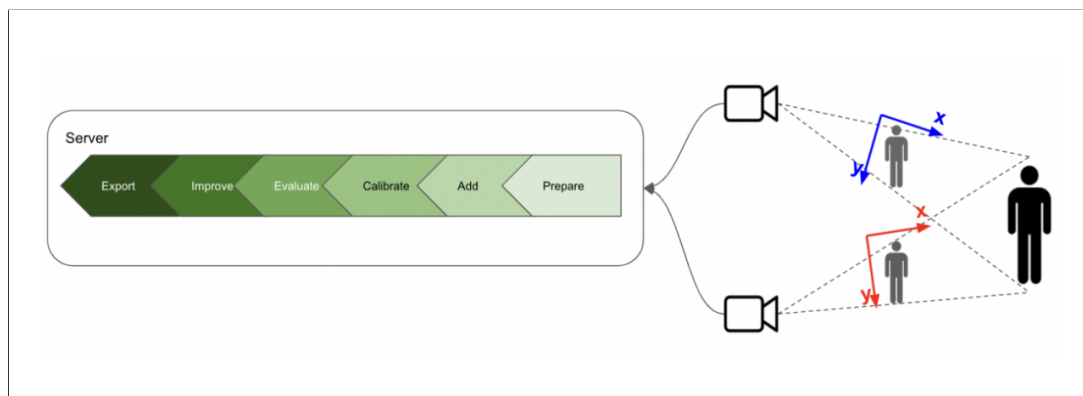


Figura 2. Partes de un sistema de visión computacional.

2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Responsable	Ing. Ariel Salassa	Autónomo	Sr. Software Engineer Alumno
Cliente	Ing. Juan Pablo Pizarro	Globant	Software Development Specialist
Orientador	Esp. Ing. Hernán Contigiani	Globant	Sr. Software Engineer Director Trabajo final
Usuario final	Desarrolladores de Software	-	Software developers

Cuadro 1. Identificación de los interesados.

- Responsable: Ing. Ariel Salassa, es la persona que desarrollará el proyecto.
- Cliente: El Ing. Juan Pablo Pizarro, es líder y referente técnico del equipo de desarrollo del departamento de IOT de Globant. Él es el principal interesado en que su departamento cuente con alguna solución de visión estereoscópica. Por sus altos grados de conocimientos técnicos también puede ser capaz de orientar en el desarrollo.
- Orientador: Hernán Contigiani es el director del presente proyecto y líder técnico en Globant. Su función será orientar al responsable a lo largo de la realización del proyecto.
- Desarrolladores de Software: son los usuarios finales que podrán hacer uso del sistema para enriquecer sus aplicaciones.

3. Propósito del proyecto

El propósito del proyecto es investigar el estado del arte de la visión computacional con más de una cámara como entrada de video de modo de que se puedan tomar decisiones en un ambiente 3D simulado. Con ello se pretende llegar a un sistema de visión estereoscópica más versátil que aquellos integrados en un dispositivo.

4. Alcance del proyecto

El proyecto comprenderá las siguientes etapas:

- Planificación de tareas.
- Formación en visión computacional.
- Investigación de sistemas utilizados en cámaras estereoscópicas.
- Formación en protocolos de comunicación utilizados para *straming*.
- Configuración y calibración de cámaras.
- Comunicación remota entre las cámaras y el servidor.

- Preparación de los datos.
- Extracción de características.
- Prototipado conceptual funcional.
- Reporte de resultados.

El presente proyecto no incluye:

- Desarrollo fino de redes neuronales que permitan detectar y clasificar objetos.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- El responsable dispondrá de suficiente cantidad de tiempo para resolver los problemas que se presenten en el desarrollo del proyecto.
- El responsable tendrá a su disposición a su director y/o colaboradores cuando sea pertinente.
- Se puede alcanzar un prototipo del sistema utilizando *webcams*.
- Se dispondrá de suficiente potencia de cómputo para trabajar con *streaming* de imágenes.
- Es posible realizar una composición de dos o más imágenes en una única imagen de salida que permita estimar profundidad y volumen de los objetos presentes en la imagen.
- Se puede aplicar el protocolo RTMP (*Real Time Messaging Protocol*) o RTSP (*Real Time Streaming Protocol*) para la adquisición remota de las imágenes.
- Se puede integrar al sistema una red neuronal pre entrenada para detectar objetos.

6. Requerimientos

1. Requerimientos de documentación

- 1.1. El trabajo debe ser continuamente documentado y se presentarán informes mensuales de avance al director.
- 1.2. Los informes de avance pueden ser presentados como código correctamente documentado con los resultados correspondientes.
- 1.3. El código final será de tipo abierto y deberá quedar ordenado en un repositorio público.

2. Requerimientos de lenguajes y frameworks

- 2.1. La biblioteca para el tratamiento de las imágenes será *OpenCV*.
- 2.2. El código para el desarrollo deberá ser escrito en Python. Si se requiriese una optimización de hardware, entonces deberá utilizarse C++.

- 2.3. Se deberá priorizar el uso de bibliotecas estándar (NumPy, Pandas, Matplotlib, Seaborn, etcétera).

3. Requerimientos funcionales

- 3.1. El sistema de visión debe estar compuesto por al menos dos cámaras.
3.2. Cada cámara debe estar correctamente calibrada.
3.3. El sistema de visión debe poder comandarse de manera remota.
3.4. La transmisión de imágenes desde el sistema de visión hacia el resto del sistema de clasificación y evaluación debe hacerse usando protocolos de transmisión en tiempo real.
3.5. El sistema debe ser capaz de detectar objetos y determinar profundidad, volumen y/o distancia de los mismos.
3.6. El error del sistema no deberá ser superior al 15 %.

4. Requerimientos de testing y evaluación

- 4.1. Los resultados del sistema deberán ser corroborados previamente en un entorno local y con objetos de dimensiones conocidas.

7. Historias de usuarios (*Product backlog*)

La medida del trabajo a efectuar para cumplir con cada una de las historias de usuarios estará dada por *story points*. Para ponderar los esfuerzos se utilizará la serie de Fibonacci con valores: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, etc. Cuando el esfuerzo se considere alto los *story points* tomarán valores entre 0 y 3 inclusive. Cuando se considere medio, entre 5 y 13 inclusive. Cuando el esfuerzo se considere alto el valor de los *story points* será igual o mayor a 21.

- Como desarrollador quiero un prototipo de visión estereoscópica que pueda extraer características como distancia, profundidad y volumen de objetos.

Dificultad: alta (34) – Complejidad: alta (34) – Incertidumbre: alta (34)

Story Points: $34 + 34 + 34 = 102 \rightarrow 89$

- Como cliente quiero tener un sistema de visión realizado con tecnologías estándar y *open source* para poder mantenerlo a futuro con el menor esfuerzo posible.

Dificultad: media (8) – Complejidad: media (8) – Incertidumbre: baja (1)

Story Points: $8 + 8 + 1 = 17 \rightarrow 21$

- Como desarrollador quiero poder acceder remotamente al sistema de visión.

Dificultad: alta (21) – Complejidad: alta (21) – Incertidumbre: alta (21)

Story Points: $21 + 21 + 21 = 63 \rightarrow 55$

- Como cliente quiero tener el prototipo de un sistema de visión que permita futuras iteraciones.

Dificultad: media (8) – Complejidad: media (8) – Incertidumbre: alta (21)

Story Points: $8 + 8 + 21 = 37 \rightarrow 34$

- Como cliente quiero recibir la documentación del trabajo realizado para que pueda servir como base de futuros desarrollos de la empresa.

Dificultad: media (13) – Complejidad: baja (2) – Incertidumbre: baja (1)

Story Points: $13 + 2 + 1 = 16 \rightarrow 13$

- Como cliente quiero tener una presentación resumida del trabajo para mostrar sus resultados y su potencial a todos los interesados.

Dificultad: baja (3) – Complejidad: baja (2) – Incertidumbre: baja (2)

Story Points: $3 + 2 + 2 = 7 \rightarrow 8$

8. Entregables principales del proyecto

Los entregables del proyecto son:

- Informe final.
- Presentación final.
- Diagrama de funcionamiento.
- Prototipo conceptual funcional.
- Repositorio de código del sistema.
- Informe de avance.

9. Desglose del trabajo en tareas

1. Planificación. (60 hs)

- 1.1. Estudio de necesidades. (9 hs)
- 1.2. Análisis de factibilidad. (3 hs)
- 1.3. Definición de requerimientos. (9 hs)
- 1.4. Confección de documento de planificación. (39 hs)

2. Investigación y capacitación. (114 hs)

- 2.1. Capacitación en conceptos básicos de visión estereoscópica y sus aplicaciones. (9 hs)
- 2.2. Capacitación en geometría de la formación de imágenes. (6 hs)
- 2.3. Capacitación en geometría epipolar. (6 hs)
- 2.4. Capacitación en distorsión de lentes. (6 hs)
- 2.5. Capacitación en calibración de cámaras. (6 hs)
- 2.6. Capacitación en *OpenCV*. (21 hs)
- 2.7. Capacitación en protocolos de *streaming* en tiempo real. (15 hs)
- 2.8. Elaboración de código de ejemplo de visión estereoscópica. (30 hs)
- 2.9. Elaboración de código de ejemplo de uso de protocolos en tiempo real. (15 hs)

3. *Setup* de cámaras. (45 hs)
 - 3.1. Conexión de cámaras y acceso al *stream* de video mediante ordenador. (15 hs)
 - 3.2. Calibración de cámaras individuales. (15 hs)
 - 3.3. Montaje de cámaras y calibración conjunta. (15 hs)
4. Construcción del *pipeline* de preparación de datos. (51 hs)
 - 4.1. Suavizado y ajuste de tamaño. (15 hs)
 - 4.2. Filtro de distorsión de lentes. (21 hs)
 - 4.3. Evaluación y selección de FPS (*frames per second*) del sistema. (15 hs)
5. Evaluación estéreo. (105 hs)
 - 5.1. Detección de objetos de cada imagen. (15 hs)
 - 5.2. Implementación de *template matching*. (15 hs)
 - 5.3. Implementación de geometría epipolar. (15 hs)
 - 5.4. Implementación de disparidad estéreo. (15 hs)
 - 5.5. Cálculo de distancias y profundidades. (15 hs)
 - 5.6. Estimación volumétrica. (15hs)
 - 5.7. Prototipado y preparación de resultados de salida. (15 hs)
6. Pruebas de validación. (63 hs)
 - 6.1. Pruebas locales en tiempo real. (21 hs)
 - 6.2. Pruebas remotas en tiempo real. (21 hs)
 - 6.3. Ajustes finales. (21 hs)
7. Documentación y presentación final. (150 hs)
 - 7.1. Elaborar informe de avance proyecto. (39 hs)
 - 7.2. Elaborar informe final del proyecto. (90 hs)
 - 1) Redacción y corrección capítulo I. (18 hs)
 - 2) Redacción y corrección capítulo II. (18 hs)
 - 3) Redacción y corrección capítulo III. (18 hs)
 - 4) Redacción y corrección capítulo IV. (18 hs)
 - 5) Redacción y corrección capítulo V. (18 hs)
 - 7.3. Preparación de presentación final. (21 hs)

Cantidad total de horas: (588 hs)

10. Diagrama de Activity On Node

En la Figura 3 se muestra el diagrama de *Activity on Node* del proyecto. Las flechas resaltadas en negro ilustran el camino crítico del proyecto. También se puede ver que los hitos marcan el fin de las distintas fases del proyecto que, a su vez, están representadas en distintos colores.

La primera fase está comprendida por tareas previas al desarrollo en código del proyecto. En esta fase se considera la planificación y la capacitación.

La segunda fase involucra todo el *pipeline* necesario para tratar las imágenes y obtener resultados de salida.

La tercera fase consiste en tomar como punto de partida un sistema ya prototipado para someterlo a distintas pruebas de interés.

Finalmente, la última fase, involucra toda la realización de toda la documentación del trabajo realizado.

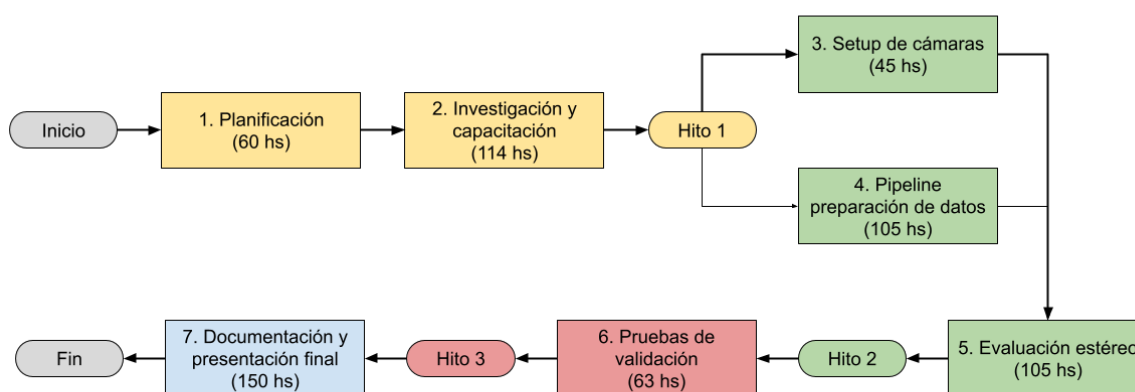


Figura 3. Diagrama en *Activity on Node*.

11. Diagrama de Gantt

A continuación se muestra el diagrama de Gantt del presente proyecto. Se consideró la jornada laboral de 3 horas de trabajo desde la fecha de inicio del curso hasta finales de año. En la figura 4 se muestra el diagrama de Gantt, tal como se enumeró en la sección 9.

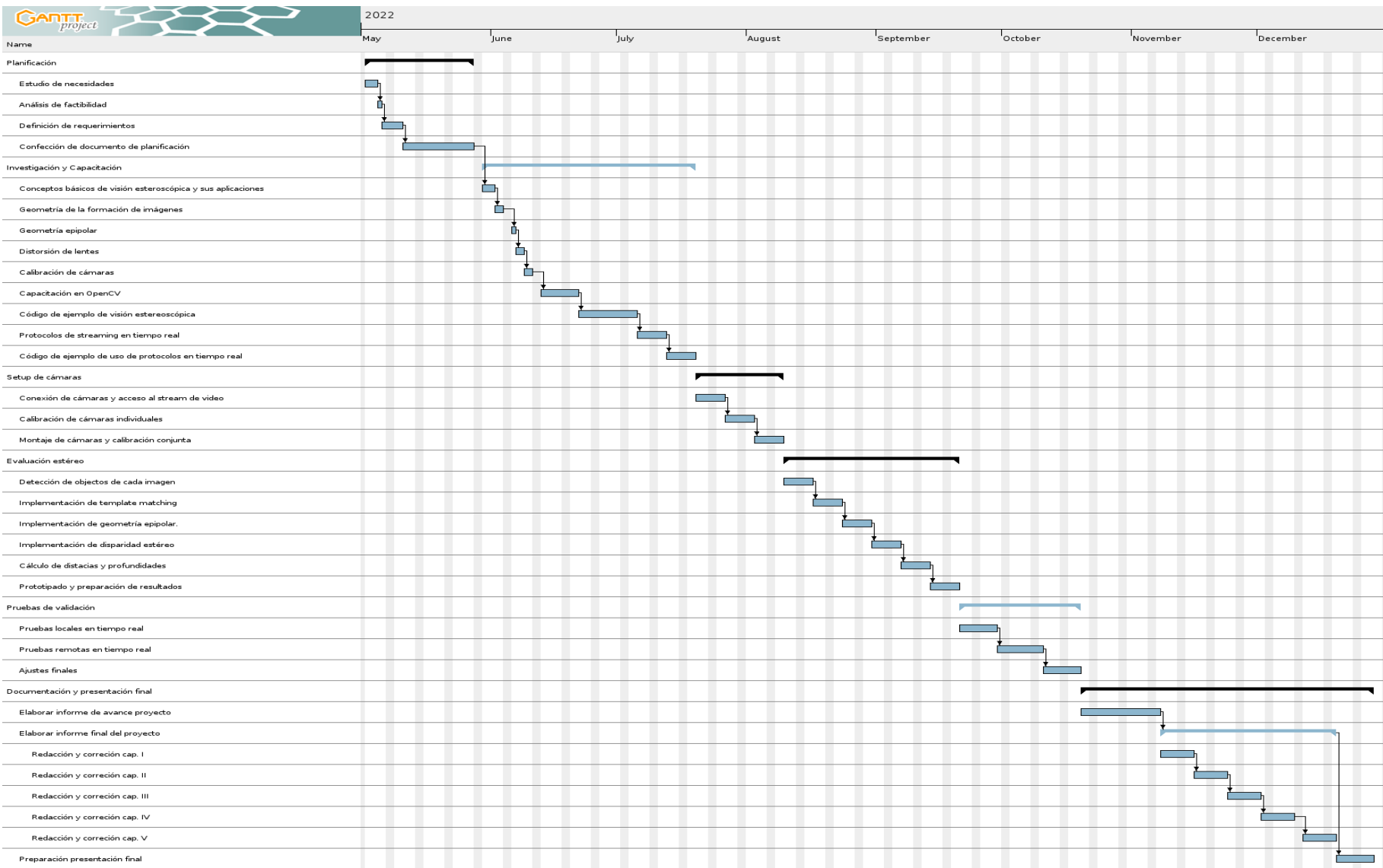


Figura 4. Diagrama de Gantt.

12. Presupuesto detallado del proyecto

En esta sección se detallan los gastos del proyecto. Las unidades del valor unitario están dadas en dólares estadounidenses. Al momento de redactar este informe la cotización oficial es de $1USD = 121ARS$.

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total [USD]
Mano de obra	590 horas	20 USD/hora	11800
WebCam	2 unidades	25 USD	50
SUBTOTAL			11850
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total [USD]
30 % del costo directo	-	-	3555
SUBTOTAL			3555
TOTAL			15405

13. Gestión de riesgos

a) Identificación de los riesgos del proyecto:

Riesgo 1: falta de tiempo del responsable.

- Severidad (5): se atrasarían las tareas, no se cumpliría con la planificación y, en el peor de los casos, no se lograría completar el proyecto para la fecha de presentación establecida.
- Probabilidad de ocurrencia (5): el responsable no trabaja en la misma empresa que ha propuesto el proyecto y, lógicamente, el trabajo es ponderante sobre el proyecto.

Riesgo 2: no contar con suficiente capacidad de cómputo.

- Severidad (8): trabajar con procesamiento de imágenes siempre consume una parte considerable de los recursos de hardware. No contar con el equipo adecuado implicaría que las tareas se harían más largas de lo estimado o que se deberían contratar servicios en la nube que encarecerían el costo del proyecto.
- Probabilidad de ocurrencia (2): el responsable posee un equipo adecuado y posee un equipo aceptable de *back up* ante cualquier eventualidad.

Riesgo 3: no contar con las cámaras necesarias.

- Severidad (7): toda prueba debería ser en base a dispositivos simulados que no suelen encontrarse fácilmente. El mero hecho de buscar un dispositivo sería una tarea imprevista que alargaría los tiempos del proyecto.
- Probabilidad de ocurrencia (3): la oferta de cámaras de calidad aceptable es amplia.

Riesgo 4: mala calibración de cámaras.

- Severidad (6): la correcta calibración de las cámaras es un requerimiento para las etapas posteriores del proyecto. Una mala calibración impediría el avance del proyecto y, por lo tanto, alargaría los tiempos estimados.
- Probabilidad de ocurrencia (4): la calibración es un proceso iterativo donde se usan patrones definidos por el desarrollador. Hay una serie de posibles errores sistemáticos en el proceso que podrían causar problemas.

Riesgo 5: comportamiento inesperado del sistema a la hora de evaluar disparidad.

- Severidad (8): el riesgo implicaría adentrarse en profundidad en el comportamiento de la cámara seleccionada, en la implementación de código de *OpenCV* o volver al paso de calibración, dilatando considerablemente los tiempos. Además, sin un correcto mapa de disparidad, nunca se podría estimar la distancia de un objeto a las cámaras o el volumen de un objeto detectado.
- Probabilidad de ocurrencia (3): la biblioteca seleccionada es conocida por su robustez y una buena calibración debería evitar este riesgo.

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

Riesgo	S	O	RPN	S*	O*	RPN*
Falta de tiempo del responsable	5	4	20	-	-	20
No contar con suficiente capacidad de cómputo	8	2	16	-	-	16
No contar con las cámaras necesarias	7	3	21	-	-	21
Mala calibración de cámaras	6	4	24	4	1	4
Comportamiento inesperado del sistema a la hora de evaluar disparidad	8	3	24	5	1	5

Criterio adoptado: se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a 21.

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 4: mala calibración de cámaras.

- Plan de mitigación: en primer lugar se repetirá el proceso y se comprobará si los parámetros de calibración varían. Si no varían se procederá a buscar patrones alternativos para la calibración de las mismas o se investigarán otros métodos.
- Severidad (4): cada uno de los planes de mitigación conlleva un tiempo adicional no menor por lo que los tiempos del proyecto se dilatarían.
- Probabilidad de ocurrencia (1): haciendo todos los pasos con cuidado debería alcanzarse la calibración deseada. Es un proceso que está relativamente estandarizado.

Riesgo 5: comportamiento inesperado del sistema a la hora de evaluar disparidad.

- Plan de mitigación: se buscaría el origen del problema. A priori la fuente de error podría venir de un error en la biblioteca utilizada, una pobre calibración de las cámaras o un problema de las mismas cámaras.
- Severidad (5): encontrar el origen del problema puede llegar a ser una tarea tediosa y repetitiva. Dilataría los tiempo estimados considerablemente.
- Probabilidad de ocurrencia (1): la principal biblioteca de visión es bastante estándar y es ampliamente utilizada. Las cámaras que se van a utilizar en este proyecto tienen *datasheets* con suficiente documentación. Si se fue cuidadoso en el proceso de calibración es poco probable la disparidad no sea correctamente calculada.

14. Gestión de la calidad

- Req 1.1: el trabajo debe ser continuamente documentado y se presentarán informes de avance una vez cada tres semanas al director.
 - Verificación: se controlará que cada avance tenga la documentación necesaria de manera que el director pueda entender el avance sin entrar en detalles de implementación.
 - Validación: revisión y conformidad por parte del director.
- Req 1.2: los informes de avance pueden ser presentados como código correctamente documentado con los resultados correspondientes.
 - Verificación: se controlará que cada avance tenga la documentación necesaria de manera que el director pueda entender el avance sin entrar en detalles de implementación.
 - Validación: revisión y conformidad por parte del director.
- Req 1.3: el código final será de tipo abierto y deberá quedar ordenado en un repositorio público.
 - Verificación: desde el principio se utilizará un repositorio público, cuya propiedad será compartida entre el responsable, el director y el cliente.
 - Validación: revisión y conformidad por parte del director y del cliente.
- Req 2.1: la biblioteca para el tratamiento de las imágenes será OpenCV.
 - Verificación: todo el código escrito se volcará en un entorno virtual provisto de un *Jupyter Lab* con *OpenCV* y todas sus dependencias pre instaladas.
 - Validación: N/A (requerimiento interno).
- Req 2.2: el código para el desarrollo deberá ser escrito en Python. Si se requiriese una optimización de hardware, entonces deberá utilizarse C++..
 - Verificación: Se prestará especial atención a los tiempos de ejecución de código en cada uno de los pasos y se evaluarán aquellos cuyos tiempos ejecución tomen más tiempo o recursos de cómputo del esperado. Se intentará reescribir la funcionalidad en C++ de las partes del código que resulten inaceptables en cuanto a tiempos de ejecución y recursos consumidos.
 - Validación: revisión y conformidad por parte del director.

- Req 2.3: se deberá priorizar el uso de bibliotecas estándar (NumPy, Pandas, Matplotlib, Seaborn, etcétera).
 - Verificación: todo el código escrito se volcará en un entorno virtual provisto de un *Jupyter Lab* con todas las mencionadas bibliotecas pre instaladas.
 - Validación: N/A (requerimiento interno).
- Req 3.1: el sistema de visión debe estar compuesto por al menos dos cámaras.
 - Verificación: N/A (requerimiento interno).
 - Validación: El prototipo entregable final tendrá al menos dos cámaras.
- Req 3.2: cada cámara debe estar correctamente calibrada.
 - Verificación: se utilizará como patrón una cuadrilla de ajedrez de dimensiones conocidas y se tomarán varias imágenes con distintas posiciones y orientaciones y se utilizarán las funciones de calibración de cámara de *OpenCV* para determinar los parámetros de una cámara. El procedimiento se repetirá para cada una de las cámaras (por más que sean idénticas).
 - Validación: Se evaluará que los errores de reproyección (distancias entre los puntos de las esquinas detectados en la imagen y los puntos del mundo ideal correspondientes proyectados en la imagen) sean razonablemente pequeños.
- Req 3.3: el sistema de visión debe poder comandarse de manera remota.
 - Verificación: el sistema ejecutará un servidor que podrá ser consultado remotamente por protocolo HTTP para ejecutar un comando y retornar una respuesta.
 - Validación: con un primer prototipo se dará acceso remoto del sistema al cliente y/o director para su validación.
- Req 3.4: la transmisión de imágenes desde el sistema de visión hacia el resto del sistema de clasificación y evaluación debe hacerse usando protocolos de transmisión en tiempo real.
 - Verificación: se estudiará con detalle el *datasheet* de las cámaras y se accederá a parámetros de bajo nivel mediante código. Se utilizará un protocolo de *streaming* en tiempo real para transmitir la imagen de video de la cámara a otro dispositivo conectado al sistema de manera inalámbrica.
 - Validación: se colocará la cámara en un lugar y el responsable en otro lugar con un monitor y el responsable deberá ser capaz de ver en tiempo real las imágenes que transmite la cámara.
- Req 3.5: el sistema debe ser capaz de detectar objetos y determinar profundidad, volumen y/o distancia de los mismos.
 - Verificación: para la detección de objetos se utilizará una red convolucional pre entrenada y se verificarán las clases predichas por la misma. Para determinar profundidad, volumen y/o distancia se utilizarán objetos de dimensiones conocidas y se compararán las mismas con los datos de salida del sistema.
 - Validación: se utilizarán objetos de dimensiones conocidas y se validarán las mismas con los datos de salida del sistema.
- Req 3.6: el error del sistema no deberá ser superior al 15 %.
 - Verificación: se guardarán los resultados de salida del sistema para objetos conocidos para consultar y verificar.

- Validación: se compararán los datos de salida con las dimensiones conocidas de los objetos de validación.
- Req 4.1: Requerimientos de testing y evaluación.
 - Verificación: se contará con una serie de objetos de dimensiones conocidas.
 - Validación: revisión y conformidad por parte del director.

15. Procesos de cierre

Una vez finalizado el proyecto, se procederá a su cierre para lo cual se contemplarán las siguientes actividades, cada una de ellas a cargo del responsable Ariel Salassa:

- Análisis del cumplimiento del Plan de Proyecto original:
 - Se compararán las horas dedicadas a cada tarea con las horas planificadas en el inicio.
 - Se analizará el porcentaje de requerimientos cumplidos.
 - Se evaluará el desempeño de la solución y la satisfacción del equipo de trabajo.
- Identificación de las técnicas y procedimientos útiles e inútiles que se utilizaron, así como los problemas que surgieron y cómo se solucionaron.
 - Se dejará asentado en los informes al director qué fue lo que se probó, resaltando lo que dio resultado y lo que no, los inconvenientes que surgieron y cómo se solucionaron.
 - Se dejarán escritas las posibles oportunidades de mejora que presente la solución.
- Presentación oral virtual del proyecto y acto de agradecimiento a todos los interesados.
 - Se organizará una sesión virtual donde se presentará el proyecto, su utilidad y su estado al concluir la especialización.