



Mejorando la precisión de un sistema de checkout autónomo mediante técnicas de visión por computadora

Autor:
Ing. Rodrigo Pazos

Director:
Ing. Maxim Dorogov (FIUBA)

Codirector:
John Doe (FIUBA)

Esta planificación fue realizada en el curso de Gestión de proyectos entre el 20 de junio de 2023 y el 15 de agosto de 2023.



Uso de técnicas de visión por computadora para mejorar la precisión de un sistema autónomo de checkout

Autor:
Ing. Rodrigo Pazos

Director:
Ing. Maxim Dorogov (FIUBA)

Esta planificación fue realizada en el curso de Gestión de proyectos entre el 20 de junio de 2023 y el 15 de agosto de 2023.

Índice

1. Descripción técnica-conceptual del proyecto a realizar.	5
2. Identificación y análisis de los interesados	6
3. Propósito del proyecto.	7
4. Alcance del proyecto	7
5. Supuestos del proyecto.	7
6. Requerimientos	8
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	14
13. Gestión de riesgos.	14
14. Gestión de la calidad	15
15. Procesos de cierre.	16

Índice

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

Acta de constitución del proyecto

Buenos Aires, 20 de junio de 2023

Por medio de la presente se acuerda con el Ing. Ing. Rodrigo Pazos que su Trabajo Final de la Carrera de Especialización en Inteligencia Artificial se titulará "Mejorando la precisión de un sistema de **checkout autónomo mediante técnicas de visión por computadora**", consistirá esencialmente en la mejora de un sistema existente para mejorar su capacidad predictiva, y tendrá un presupuesto preliminar estimado de 600 h de trabajo, con fecha de inicio 20 de junio de 2023 y fecha de presentación pública 15 de **mayo de 2022**.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

Dirección de la carrera de CEIA
FIUBA

Ing. Maxim Dorogov
Director del Trabajo Final

Acta de constitución del proyecto

Buenos Aires, 20 de junio de 2023

Por medio de la presente se acuerda con el Ing. Ing. Rodrigo Pazos que su Trabajo Final de la Carrera de Especialización en Inteligencia Artificial se titulará "**Uso de técnicas de visión por computadora para mejorar** la precisión de un sistema **autónomo de checkout**", consistirá esencialmente en la mejora de un sistema existente para mejorar su capacidad predictiva, y tendrá un presupuesto preliminar estimado de 600 h de trabajo, con fecha de inicio 20 de junio de 2023 y fecha de presentación pública 31 de **enero 2023**.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

Dirección de la carrera de CEIA
FIUBA

Ing. Maxim Dorogov
Director del Trabajo Final

- **Usuario final:** En el equipo de desarrollo y mantenimiento técnico, se incluyen tanto los desarrolladores como los encargados de *devops*. Estos profesionales requerirán herramientas y procesos ajustados a las necesidades del negocio para llevar a cabo eficientemente sus tareas.

3. Propósito del proyecto

El propósito de este proyecto es contribuir al avance del conocimiento en el uso de cámaras en sistemas de checkout autónomo, con el objetivo de mejorar la única aproximación de código abierto disponible. En simultáneo, se busca profundizar sobre las diferentes técnicas de visión de computadora y explorar cómo utilizarlas y combinarlas de manera efectiva para desarrollar un predictor adecuado para este problema específico.

4. Alcance del proyecto

El presente proyecto incluye:

- Actualización y refactorización del código ya existente para mejorar su ergonomía y usabilidad.
- una investigación sobre las técnicas disponibles y sobre cómo utilizarlas y combinarlas para lograr un predictor adecuado para este problema
- Una actualización pertinente de la documentación, para que sea fácil usar y expandir el proyecto, de forma que otros puedan construir sobre este.
- Implementación de un predictor basado en los imágenes de video del dataset para resolver el problema presentado.

El presente proyecto no incluye:

- Una implementación concreta con la infraestructura que pueda usarse con el software desarrollado.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- El dataset (videos y datos de los sensores) sigue completamente disponible.
- El proyecto ganador sigue disponible en GitHub.
- Los videos disponibles deben tener un formato procesable.
- Debe existir un dataset que contenga **imágenes** de los productos que se observan en los videos para poder entrenar **predictores**
- **No deben haber inconsistencias en el dataset (falta de relación entre el los datos de sensores y lo que se observa en los vides, por ejemplo).**

- **Usuario final:** En el equipo de desarrollo y mantenimiento técnico, se incluyen tanto los desarrolladores como los encargados de *devops*. Estos profesionales requerirán herramientas y procesos ajustados a las necesidades del negocio para llevar a cabo eficientemente sus tareas.

3. Propósito del proyecto

El propósito de este proyecto es contribuir al avance del conocimiento en el uso de cámaras en sistemas de checkout autónomo, con el objetivo de mejorar la única aproximación de código abierto disponible. En simultáneo, se busca profundizar sobre las diferentes técnicas de visión de computadora y explorar cómo utilizarlas y combinarlas de manera efectiva para desarrollar un predictor adecuado para este problema específico.

4. Alcance del proyecto

El presente proyecto incluye:

- Actualización y refactorización del código ya existente para mejorar su ergonomía y usabilidad.
- una investigación sobre las técnicas disponibles y sobre cómo utilizarlas y combinarlas para lograr un predictor adecuado para este problema
- Una actualización pertinente de la documentación, para que sea fácil usar y expandir el proyecto, de forma que otros puedan construir sobre este.
- Implementación de un predictor basado en los imágenes de video del dataset para resolver el problema presentado.

El presente proyecto no incluye:

- Una implementación concreta con la infraestructura que pueda usarse con el software desarrollado.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- El dataset (videos y datos de los sensores) sigue completamente disponible.
- El proyecto ganador sigue disponible en GitHub.
- Los videos disponibles deben tener un formato procesable.
- Debe existir un dataset que contenga **imágenes** de los productos que se observan en los videos para poder entrenar **predictores**.
- **El dataset debe ser consistente, evitando la falta de relación entre los datos de sensores y las observaciones en los videos, garantizando que no haya inconsistencias.**

6. Requerimientos

1. Requerimientos funcionales

- 1.1. El sistema debe ser capaz de procesar videos de compras y extraer información relevante de las imágenes, incluyendo la detección de productos.
- 1.2. El sistema debe utilizar técnicas de visión por computadora para identificar con precisión los productos presentes en el video, incluso en casos de escenarios complejos con varios clientes, movimientos de productos entre estantes y productos fuera de los lugares designados.
- 1.3. El sistema debe ser capaz de distinguir entre diferentes versiones de un mismo producto, **como versiones grandes o chicas**, evitando confusiones y garantizando una identificación precisa.
- 1.4. El sistema debe mantener actualizado el inventario de productos a medida que se realizan las compras, registrando las cantidades vendidas y actualizando los niveles de stock en tiempo real.
- 1.5. El sistema debe generar un ticket de compra detallado y preciso, incluyendo los productos adquiridos, sus precios y el total a pagar por el cliente.
- 1.6. El sistema debe ser escalable y flexible, permitiendo la incorporación de nuevos productos

2. Requerimientos no funcionales

- 2.1. El sistema debe **lograr una precisión mínima del 96.7 % en la identificación de productos, igualando o superando la precisión alcanzada por el equipo ganador de la competencia de AiFi.**
- 2.2. El sistema debe ser fácil de mantener y actualizar, con una estructura de código limpia y modular que facilite la incorporación de mejoras y la corrección de errores.
- 2.3. **El sistema debe ser fácil de mantener y actualizar, con una estructura de código limpia y modular que facilite la incorporación de mejoras y la corrección de errores.**
- 2.4. El sistema debe ser robusto y capaz de manejar diferentes tipos de productos, envases y etiquetas.

3. Requerimientos de documentación

- 3.1. Se debe proporcionar un informe técnico que describa la arquitectura del sistema, incluyendo diagramas y explicaciones detalladas de los componentes, módulos y su interacción.
- 3.2. El proyecto debe contar con una documentación clara y detallada del código fuente, incluyendo comentarios, explicaciones de las funciones y métodos utilizados, y guías de uso de las diferentes partes del sistema
- 3.3. Se deben documentar las técnicas de visión por computadora utilizadas, incluyendo algoritmos, metodologías y herramientas específicas empleadas en el procesamiento de imágenes y la identificación de productos.
- 3.4. Se debe documentar el proceso de entrenamiento del modelo de visión por computadora, detallando el conjunto de datos utilizado, la preparación de los datos, los algoritmos de aprendizaje automático empleados y los resultados obtenidos.

6. Requerimientos

1. Requerimientos funcionales

- 1.1. El sistema debe ser capaz de procesar videos de compras **en tiempo real** y extraer información relevante de las imágenes, incluyendo la detección de productos. **En este contexto tiempo real contempla una demora de hasta 5 minutos luego de finalizada la compra.**
- 1.2. El sistema debe utilizar técnicas de visión por computadora para identificar con precisión los productos presentes en el video, incluso en casos de escenarios complejos con varios clientes, movimientos de productos entre estantes y productos fuera de los lugares designados.
- 1.3. El sistema debe ser capaz de distinguir entre diferentes versiones de un mismo producto, evitando confusiones y garantizando una identificación precisa.
- 1.4. El sistema debe mantener actualizado el inventario de productos a medida que se realizan las compras, registrando las cantidades vendidas y actualizando los niveles de stock en tiempo real.
- 1.5. El sistema debe generar un ticket de compra detallado y preciso, incluyendo los productos adquiridos, sus precios y el total a pagar por el cliente.
- 1.6. El sistema debe ser escalable y flexible, permitiendo la incorporación de nuevos productos

2. Requerimientos no funcionales

- 2.1. El sistema debe **permitir la integración de nuevos modelos de forma sencilla.**
- 2.2. El sistema debe ser fácil de mantener y actualizar, con una estructura de código limpia y modular que facilite la incorporación de mejoras y la corrección de errores.
- 2.3. El sistema debe ser robusto y capaz de manejar diferentes tipos de productos, envases y etiquetas.

3. Requerimientos de documentación

- 3.1. Se debe proporcionar un informe técnico que describa la arquitectura del sistema, incluyendo diagramas y explicaciones detalladas de los componentes, módulos y su interacción.
- 3.2. El proyecto debe contar con una documentación clara y detallada del código fuente, incluyendo comentarios, explicaciones de las funciones y métodos utilizados, y guías de uso de las diferentes partes del sistema
- 3.3. Se deben documentar las técnicas de visión por computadora utilizadas, incluyendo algoritmos, metodologías y herramientas específicas empleadas en el procesamiento de imágenes y la identificación de productos.
- 3.4. Se debe documentar el proceso de entrenamiento del modelo de visión por computadora, detallando el conjunto de datos utilizado, la preparación de los datos, los algoritmos de aprendizaje automático empleados y los resultados obtenidos.

7. Historias de usuarios (*Product backlog*)

Para la estimación de *story points* se utilizara el siguiente esquema, basado en la complejidad y la dificultad de cada historia de usuario.

7. Historias de usuarios (*Product backlog*)

Para la estimación de *story points* se utilizara el siguiente esquema, basado en la complejidad y la dificultad de cada historia de usuario.

Cuadro 1. Tabla de Complejidad y Dificultad

Complejidad		Dificultad	
Baja	1	Baja	1
Media	3	Media	2
Alta	8	Alta	5

Cuadro 2. Tabla de asignación de Story Points

Complejidad	Dificultad	Story Points
1	1	1
1	2	2
1	5	3
3	1	3
3	2	5
3	5	8
8	1	8
8	2	13
8	5	13

- Como encargado del local, quiero que el sistema genere predicciones con muy alta precisión para mantener el stock actualizado y generar los cobros pertinentes.
 - Complejidad: 1
 - Dificultad: 5
 - Story Points: 3
- Como usuario, quiero que el sistema genere un ticket de compra detallado y preciso, incluyendo los productos adquiridos, sus precios y el total a pagar, para facilitar y garantizar un correcto proceso de cobro.
 - Complejidad: 3
 - Dificultad: 5
 - Story Points: 8
- Como encargado de mantener el sistema, quiero que el sistema sea fácil de mantener y actualizar, con una estructura de código modular y documentación clara, para facilitar las futuras mejoras y correcciones.
 - Complejidad: 3
 - Dificultad: 2
 - Story Points: 5
- Como encargado de mantener el sistema, quiero que el sistema sea escalable y flexible, permitiendo la incorporación de nuevos productos y la adaptación a diferentes configuraciones de cámaras y entornos de tienda, para que pueda crecer y adaptarse a mis necesidades cambiantes.

Cuadro 1. Tabla de Complejidad y Dificultad

Complejidad		Dificultad	
Baja	1	Baja	1
Media	3	Media	2
Alta	8	Alta	5

En función a la estimación de complejidad y dificultad se calculará el promedio. Luego se redondeará hacia arriba para conseguir el valor de Fibonacci correspondiente. Este valor será el que corresponda a los *story points* de la *user story*.

Cuadro 2. Tabla de asignación de Story Points

Complejidad	Dificultad	Promedio	Story Points
1	1	1	1
1	2	1,5	2
1	5	3	3
3	1	2	2
3	2	2,5	3
3	5	4	5
8	1	4,5	5
8	2	5	5
8	5	6,5	8

- Como encargado del local, quiero que el sistema genere predicciones con muy alta precisión para mantener el stock actualizado y generar los cobros pertinentes.
 - Complejidad: 1
 - Dificultad: 5
 - Story Points: 3
- Como usuario, quiero que el sistema genere un ticket de compra detallado y preciso, incluyendo los productos adquiridos, sus precios y el total a pagar, para facilitar y garantizar un correcto proceso de cobro.
 - Complejidad: 3
 - Dificultad: 5
 - Story Points: 5
- Como encargado de mantener el sistema, quiero que el sistema sea fácil de mantener y actualizar, con una estructura de código modular y documentación clara, para facilitar las futuras mejoras y correcciones.
 - Complejidad: 3
 - Dificultad: 2
 - Story Points: 3
- Como encargado de mantener el sistema, quiero que el sistema sea escalable y flexible, permitiendo la incorporación de nuevos productos y la adaptación a diferentes configuraciones de cámaras y entornos de tienda, para que pueda crecer y adaptarse a mis necesidades cambiantes.

- Complejidad: 3
- Dificultad: 5
- Story Points: 8

8. Entregables principales del proyecto

Los entregables del proyecto son (ejemplo):

- Manual de uso.
 - Guía de **instalación**
 - Guía de **integración**
 - Diagrama de **arquitectura**
 - Detalles de la entrada y salida del **sistema**
 - Detalles de los modelos **utilizados**
- Código **fuentes**
 - Código actualizado con la nueva funcionalidad de visión por **computadora**
 - Uno o más modelos de visión por computadora **implementados**
- Informe **final**

9. Desglose del trabajo en tareas

- Investigación de técnicas de visión por **computadora**
 - Investigar técnicas disponibles generalmente para la detección de **productos** (40 hs)
 - Investigar técnicas utilizadas por empresas de checkout autónomo para la detección de **productos** (30 hs)
 - Investigar técnicas que buscan resolver el problemas de deteccion de productos **similares** (40 hs)
 - Investigar técnicas que buscan reducir el tiempo de entrenamiento del **modelo** (40 hs)
 - Desarrollar un reporte de las técnicas investigadas y definir una o mas soluciones a probar e implementar (20 hs)
- Implementar y probar las mejores **soluciones**
 - Implementar las mejores **soluciones** (70 hs)
 - Realizar pruebas de las soluciones **implementadas** (70 hs)
 - Desarrollar un informe de las pruebas (20 hs)
- Refactorización y limpieza del código
 - Reconocer la estructura del código (40 hs)
 - Refactorizarlo de forma de mejorar la claridad y ergonomía (40 hs)

- Complejidad: 3
- Dificultad: 5
- Story Points: 5

8. Entregables principales del proyecto

Los entregables del proyecto son (ejemplo):

- Manual de uso.
 - Guía de **instalación**
 - Guía de **integración**
 - Diagrama de **arquitectura**
 - Detalles de la entrada y salida del **sistema**
 - Detalles de los modelos **utilizados**
- Código **fuentes**
 - Código actualizado con la nueva funcionalidad de visión por **computadora**
 - Uno o más modelos de visión por computadora **implementados**
- Informe **final**

9. Desglose del trabajo en tareas

- Investigación de técnicas de visión por **computadora**. (150 h)
 - Investigar técnicas disponibles generalmente para la detección de **productos**. (40 h)
 - Investigar técnicas utilizadas por empresas de checkout autónomo para la detección de **productos**. (30 h)
 - Investigar técnicas que buscan resolver el problemas de deteccion de productos **similares**. (40 h)
 - Investigar técnicas que buscan reducir el tiempo de entrenamiento del **modelo**. (40 h)
- Implementar y probar las mejores **soluciones**. (150 h)
 - Implementar las mejores 3 **soluciones**. (75 h)
 - Implementar la mejor solución. (25 h)
 - Implementar la segunda mejor solución. (25 h)
 - Implementar la tercera mejor solución. (25 h)
 - Realizar pruebas de las soluciones **implementadas**. (75 h)
 - Realizar pruebas de la mejor solución. (25 h)
 - Realizar pruebas de la segunda mejor solución. (25 h)
 - Realizar pruebas de la tercera mejor solución. (25 h)

- 3.3. Implementar una interfaz para poder desarrollar varios modelos predictores de **productos** que formen parte del nuevo **proceso** (20 h)
- 3.4. Integrar los predictores al código **refactorizado** (30 hs)
4. Documentación e informe **final**
 - 4.1. Desarrollar **manual** de uso (50 hs)
 - 4.2. **Documentar** código (40 hs)
 - 4.3. **Redactar** el informe final (50 hs)

Cantidad total de horas: 600 **hs**

10. Diagrama de Activity On Node

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

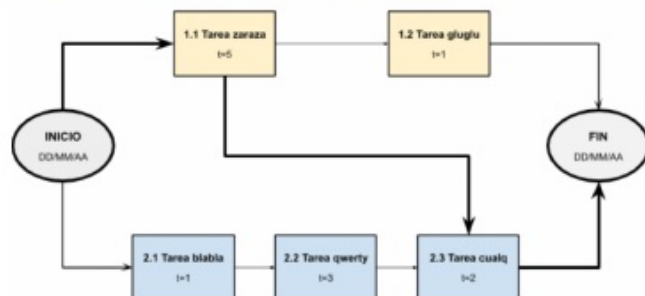


Figura 2. Diagrama de *Activity on Node*.

Indicar claramente en qué unidades están expresados los tiempos. De ser necesario indicar los caminos semicríticos y analizar sus tiempos mediante un cuadro. Es recomendable usar colores y un cuadro indicativo describiendo qué representa cada color, como se muestra en el siguiente ejemplo:

11. Diagrama de Gantt

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

- **Planner**
- **GanttProject**

3. Refactorización y limpieza del código. (140 h)

- 3.1. Reconocer la estructura del código. (40 h).
- 3.2. Refactorizarlo de forma de mejorar la claridad y ergonomía. (40 h)
- 3.3. Implementar una interfaz para poder desarrollar varios modelos predictores de **productos** que formen parte del nuevo **proceso**. (30 h)
- 3.4. Integrar los predictores al código **refactorizado**. (30 h)

4. Documentación e informe final. (160 h)

- 4.1. **Inicio** elaboración memoria técnica - Taller de Trabajo Final A. (100 h)
 - 1) Desarrollar un reporte de las técnicas investigadas y definir al menos tres soluciones a probar e implementar. (20 h)
 - 2) Desarrollar un informe de las pruebas realizadas. (20 h)
 - 3) Documentación del código. (20 h)
 - 4) Elaboración manuales de uso y reparación. (40 h)
- 4.2. Revisión y correcciones de la memoria. (5 h)
- 4.3. **Fin** de elaboración memoria técnica - Taller de Trabajo Final B. (40 h)
- 4.4. Revisión y correcciones de la memoria. (5 h)
- 4.5. Elaboración presentación final. (8 h)
- 4.6. Revisión y correcciones de la presentación final. (2 h)

Cantidad total de horas: 600 **h**

10. Diagrama de Activity On Node

En la Figura 2 se observa el Activity on Node del proyecto. Caben destacar las siguientes referencias y aclaraciones:

- Cada nodo contiene una referencia a una tarea del WBS presentado en el punto anterior.
- Cada grupo de tareas fue agrupado por colores: lila para el trabajo de implementación, rojo para el trabajo relacionado a implementación y prueba de modelos, amarillo para el trabajo relacionado a la refactorización del código y naranja para el trabajo de documentación.
- Se detallan el camino crítico y semicrítico, con su referencia correspondiente en la figura.
- El camino crítico del proyecto es de 360 horas. Es decir, que no podría terminarse antes de las 360 horas de trabajo.

- 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 3, 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.

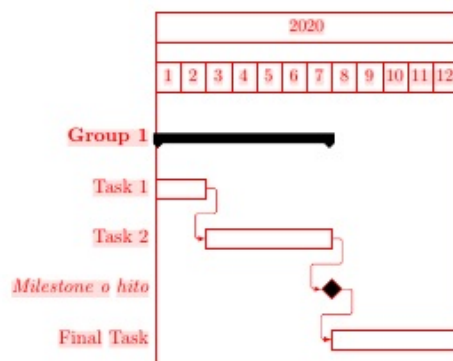


Figura 3. Diagrama de Gantt de ejemplo

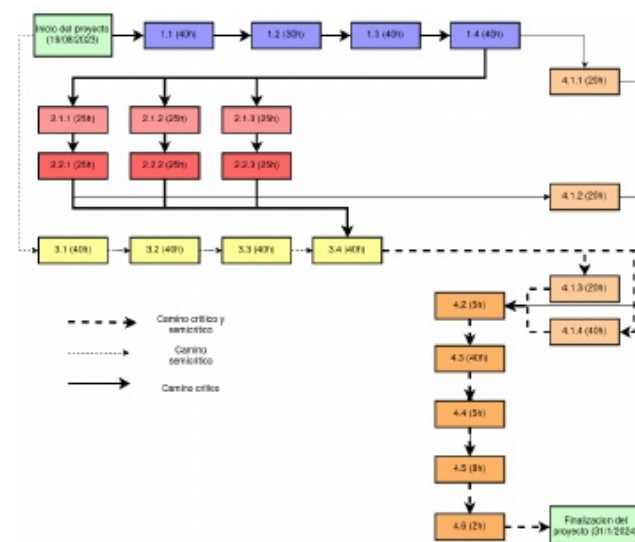
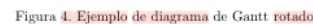


Figura 2. Diagrama de Activity on Node.

11. Diagrama de Gantt



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**.

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):
- Ocurrencia (O):

Riesgo 3:

- Severidad (S):

El diagrama de Gantt que se muestra en la Figura 3 contempla 31 horas de trabajo semanal.

12. Presupuesto detallado del proyecto

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
Sueldo ingeniero	600 h	\$ 5.000 / h	\$ 3.000.000
PC Laptop	600 h	\$ 530 / h	\$ 318.000
PC con placa de video	600 h	\$ 1060 / h	\$ 636.000
SUBTOTAL			\$3.954.000
COSTOS INDIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
Espacio de coworking (individual)	6 meses	\$ 20.000 / mes	\$ 120.000
SUBTOTAL			\$ 120.000
TOTAL			\$ 4.074.000

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):
- Ocurrencia (O):

Riesgo 3:

- Severidad (S):
- Ocurrecia (O):

■ **Ocurrencia (O):**

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.

- 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.

Riesgo	S	O	RPN	S*	O*	RPN*

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

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.

- 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.