



Solución IoT para robot de exploración ambiental de datos críticos con almacenamiento en blockchain

Autor:

Ing. Gonzalo F. Carreño

Director:

Esp. Ing. Sergio Alberino (UTN/FRBA)

*Esta planificación fue realizada en el curso de Gestión de proyectos
entre el 22 de junio de 2024 y el 17 de agosto de 2024.*

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>).	10
8. Entregables principales del proyecto	11
9. Desglose del trabajo en tareas	12
10. Diagrama de Activity On Node.	13
11. Diagrama de Gantt	14
12. Presupuesto detallado del proyecto	16
13. Gestión de riesgos	16
14. Gestión de la calidad	18
15. Procesos de cierre	19

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento.	22 de junio de 2024
1	Se aplican correcciones y completan capítulos restantes	9 de agosto de 2024

Acta de constitución del proyecto

Buenos Aires, 22 de junio de 2024

Por medio de la presente se acuerda con el Ing. Gonzalo F. Carreño que su Trabajo Final de la Carrera de Maestría en Internet de las Cosas se titulará “Solución IoT para robot de exploración ambiental de datos críticos con almacenamiento en blockchain”, consistirá en la implementación de un sistema embebido de robot de exploración ambiental conectado a un sistema back-end en la nube pública capaz de persistir los datos en una red blockchain, y tendrá un presupuesto preliminar estimado de \$ 160 dólares estadounidenses y 615 horas de trabajo, con fecha de inicio 22 de junio de 2024 y fecha de presentación pública a definir entre octubre y diciembre de 2025.

Se adjunta a esta acta la planificación inicial.

Dr. Ing. Ariel Lutenberg
Director posgrado FIUBA

Esp. Lic. Mariano Landini
FCE UBA - Cliente

Esp. Ing. Sergio Alberino
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

El presente proyecto es un emprendimiento personal que busca integrar el dispositivo robótico de exploración ambiental controlable a distancia desarrollado en el marco de la Carrera de Especialización en Sistemas Embebidos con un backend para el análisis y explotación de datos en la nube pública y una red blockchain para el almacenamiento inmutable de datos críticos.

Estado del arte: los robots exploradores son los dispositivos robotizados que han sido creados con el fin de reconocer y explorar un lugar o terreno siendo capaces de moverse de forma autónoma o controlados por personas a control remoto. Su objetivo es evitar poner en riesgo la vida de los humanos, ya sea debido a que el lugar es inaccesible o porque se encuentra en una zona contaminada. Algunos de los tipos de robots exploradores más conocidos son los espaciales, de minas, de rescate en catástrofes, de tuberías, acuáticos y/o submarinos y de suelos.

Estos dispositivos, forman parte de una solución arquitectónica IoT más amplia, cumpliendo el rol de nodos “edge”. Una vez implementados y funcionando envían lecturas del reconocimiento y operaciones realizadas a sistemas de procesamiento y almacenamiento de datos que conforman el backend, generalmente en la nube, desde donde se puede visualizar los resultados obtenidos, sus métricas derivadas, realizar trazabilidad de las operaciones así como almacenar y gobernar los datos generados.

En situaciones en las que el robot es utilizado para explorar y monitorear un área ambientalmente sensible, como una reserva natural o un sitio afectado por un desastre ecológico y su misión es recopilar datos críticos tales como niveles de contaminación, temperatura, humedad, y calidad del aire. Es de gran importancia almacenar estas mediciones de una manera en la que se pueda asegurar la integridad y transparencia de los datos, como por ejemplo, en una cadena de bloques (blockchain).

Una arquitectura blockchain se basa en el procesamiento y almacenamiento de transacciones agrupadas en bloques encadenados e inmutables de forma distribuida entre los nodos de una red en lo que se conoce como un *distributed ledger*. De esta manera se puede asegurar la integridad de los datos ya que los registros generados no se pueden modificar una vez creados. Además, como la red es accesible entre los actores intervinientes en el caso de uso, ya sea pública o privada y con o sin permisos, se puede garantizar la transparencia de los datos. La mayoría de las redes blockchain constan de la tecnología para la implementación de código backend ejecutable en la red, que aunque su nombre puede cambiar dependiendo de la red en la cual se implementan, usualmente se los conoce como *Smart Contracts*. La ejecución de estos componentes es realizada por los nodos de la red en el proceso que se conoce como minería, y como tal es una actividad que requiere el pago de un *fee* (comisión) conocido como *gas* medido en diferentes unidades dependiendo de la red y normalmente pagable desde una cuenta nominada en el token de la red asociada a la aplicación o propietario de los *smart contracts*. La forma de interactuar con los *smart contracts* en un caso de uso interactivo desde afuera de la red, se realiza a través de otro componente conocido como dApps (*de-centralized applications*) que haciendo uso de ciertas bibliotecas de Web3 invocan a estos para almacenar y obtener datos en y desde el *ledger*.

En la figura 1 se puede apreciar una posible arquitectura del sistema final.

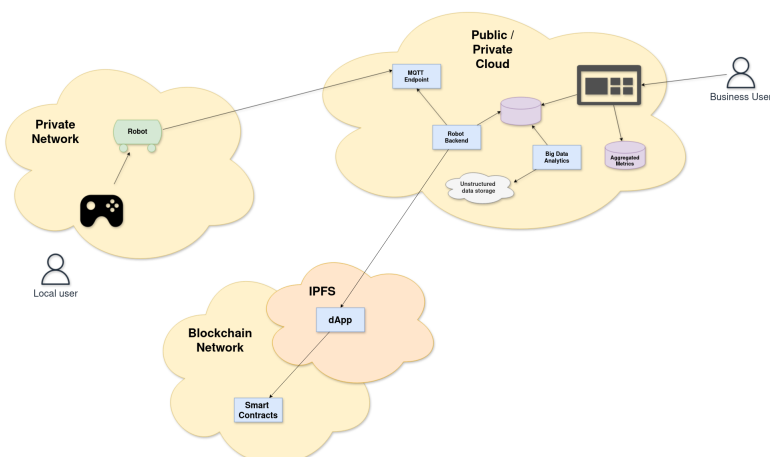


Figura 1. Arquitectura del sistema

El hardware utilizado para la solución de IoT propuesta es un robot de exploración ambiental de control inalámbrico, desarrollado en la Carrera de Especialización en Sistemas Embebidos de la Universidad de Buenos Aires. En la figura 2 puede apreciarse una fotografía del mismo.

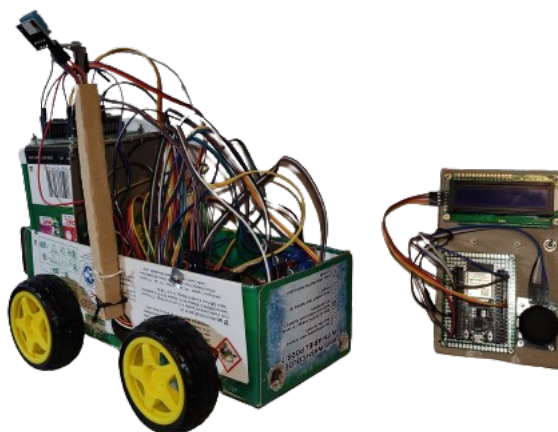


Figura 2. Robot de exploración ambiental

2. Identificación y análisis de los interesados

A continuación, se enumeran los diferentes roles e individuos que participarán en el proyecto.

Rol	Nombre y Apellido	Organización	Puesto
Cliente	Esp. Lic. Mariano Landini	FCE UBA - Cliente	-
Responsable	Ing. Gonzalo F. Carreño	UTN/FRBA	Alumno
Orientador	Esp. Ing. Sergio Alberino	UTN/FRBA	Director Trabajo final

3. Propósito del proyecto

El propósito de este proyecto es desarrollar una solución IoT para un robot de exploración ambiental para casos de uso de datos críticos en los que sea necesario contar con almacenamiento inmutable y transparente para todas las partes en una arquitectura blockchain. Por otra parte, se pretende volcar en un desarrollo concreto y de aplicación industrial los conocimientos adquiridos durante la cursada de la Maestría en Internet de las Cosas.

4. Alcance del proyecto

Contando con el hardware mencionado anteriormente que auspicia de componente edge en una arquitectura IoT, se propone la implementación de la plataforma tecnológica que alcanza:

- La publicación del endpoint MQTT para la recepción de los datos enviados por el robot.
- La adaptación del sistema embebido del robot de exploración ambiental para la conexión segura con el backend vía MQTT.
- La arquitectura e implementación de los sistemas backend y el modelo de datos necesario para el almacenamiento de las mediciones enviadas por el robot.
- La arquitectura, implementación y despliegue de la dApp y *smart contracts* necesarios para el almacenamiento de las mediciones en una red Blockchain (a definir).
- La definición de métricas agregadas de valor y posterior arquitectura e implementación de los sistemas analíticos para procesar de forma *batch* y/o *real-time* (dependiendo de las métricas a definir) utilizando herramientas de procesamiento paralelo basadas en big data.
- La implementación de la interfaz gráfica para poder visualizar los datos enviados y analíticas calculadas.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- será posible tener acceso a redes de desarrollo y testing de forma gratuita mediante la obtención *tokens* de prueba por medio de *Faucets*,
- será posible experimentar, desarrollar y hacer testing del backend cloud con el presupuesto estimado,
- será posible contar con la capa gratuita en el servicio cloud utilizado,
- arquitectónicamente resultará viable implementar la solución propuesta,
- se dispondrá del conjunto de bibliotecas, drivers y APIs de bajo nivel para el desarrollo de las funcionalidades planteadas en el alcance sin ser necesario el desarrollo de drivers y dichos componentes de bajo nivel. Además, tanto estos componentes de software como los *open source* de la comunidad de software libre utilizados durante el desarrollo del producto, se encontrará estable para que su integración en el proyecto no resulte en desvíos,

- tanto el prototipado de los componentes de software del sistema embebido como el ensamblado de los componentes de hardware del dispositivo no producirán desvíos considerables en el plan,
- no habrá desvíos no contemplados en el plan que impidan o demoren entregas en el proyecto,
- el comité académico encargado de la corrección tendrá disponibilidad para realizar la evaluación en las fechas planificadas de entrega,
- el director asignado tendrá la disponibilidad de tiempo para darle seguimiento al proyecto,
- el alumno contará con una disponibilidad de entre 3 y 5 horas diarias (incluyendo fines de semana) para el desarrollo del proyecto en el tiempo convenido,
- los materiales, componentes, software de terceros y servicios cloud utilizados funcionaran de forma óptima y de acuerdo a lo esperado,
- el robot desarrollado seguirá funcionando de forma estable sin ser necesario su ajuste, reparacion o modificacion a nivel hardware o sistema base (fuera de lo planificado para la integración con MQTT),
- los recursos no directamente relacionados con el desarrollo del proyecto, pero utilizados durante el mismo, funcionaran adecuadamente y en caso de falta de suministro (por ejemplo el servicio de Internet) o avería (por ejemplo en el caso de la computadora utilizada) la resolución será expeditiva no suponiendo un desvío en el plan,
- no sucederán nuevos eventos de impacto global (pandemia, guerras, etc) durante el desarrollo del proyecto que impliquen una demora o imposibilidad en la entrega.

6. Requerimientos

A continuación, se listan los requerimientos del producto:

1. Requerimientos funcionales

- 1.1. El robot de exploración ambiental debe poder enviar a la plataforma datos de mediciones de parámetros ambientales, incluyendo los datos de fecha, hora, localización geográfica (que puede ser implementada como un mock inicialmente) y la categorización si es o no un valor crítico.
- 1.2. El robot de exploración ambiental debe incorporar lógica para categorizar los valores medidos de cada parámetro ambiental como valores críticos si:
 - 1) representan un maximo o minimo global sensado hasta el momento,
 - 2) representan un máximo o mínimo local durante el último día.
- 1.3. La plataforma debe poder recibir y almacenar las mediciones de parámetros ambientales enviadas por el robot.
- 1.4. Los datos considerados críticos deben ser almacenados en un sistema inmutable.
- 1.5. La plataforma debe poder procesar las mediciones de parámetros ambientales enviadas por el robot para generar métricas de valor para el usuario de negocio.
- 1.6. El la plataforma proveer dos *front-end* con interfaz web:

- 1) el *front-end* para el usuario de negocio,
 - 2) el *front-end* para el usuario administrador.
- 1.7. El *front-end* para el usuario de negocio debe proveer métricas para visualizar:
- 1) las lecturas históricas almacenadas,
 - 2) agregaciones (máximo, mínimo, promedio, etc) de cada parámetro ambiental agrupado por frecuencias (ventanas de tiempo) y coordenadas geográficas,
 - 3) las referencias a los datos persistidos en blockchain.
- 1.8. El *front-end* para el usuario de administración debe permitir:
- 1) acceder a los diferentes recursos utilizados por la herramienta (topics MQTT, *smart contracts*, *buckets*, etc),
 - 2) resetear valores y estado.
2. Requerimientos no funcionales
- 2.1. La plataforma debe contar con al menos un *back-end* de procesamiento y acceso a datos operacionales para la lógica de negocio.
- 2.2. La plataforma debe contar con al menos un *back-end* de acceso, procesamiento, almacenamiento de datos analíticos para la generación de métricas.
- 2.3. El envío de los valores ambientales censados al *back-end* debe ser mediante MQTT.
- 2.4. Las lecturas ambientales categorizadas como críticas deben ser almacenadas en blockchain para garantizar fiabilidad e inmutabilidad.
- 2.5. La gestión de datos almacenados en blockchain debe ser implementada mediante *smart contracts* desplegados en la red.
- 2.6. La interacción con los *smart contracts* debe realizarse desde una dApp.
- 2.7. Los sistemas de transferencia y almacenamiento de datos utilizados deben contar con seguridad, permitiendo encriptación, autenticación y autorización.
3. Requerimientos de documentación
- 3.1. Video demostrativo.
- 3.2. Documentación de arquitectura técnica del diseño del sistema.
- 3.3. Manual de usuario.
- 3.4. Memoria final.
4. Requerimiento de testing
- 4.1. Se debe incluir tests de unitarios de componentes.
- 4.2. Se debe incluir tests funcionales (*smoke test*) del producto general.
5. Requerimientos opcionales
- 5.1. De infraestructura y despliegue:
- 1) Se permite realizar el despliegue de la dApp en un IPFS (preferentemente) o en la nube.
 - 2) Se permite agregar hardware al robot para la captura de datos adicionales.
 - 3) Se permite agregar automatización para la creación de la infraestructura como código.
- 5.2. De datos:
- 1) Se permite almacenar cualquier otro dato adicional sensado o derivado.
 - 2) Se permite agregar cualquier implementación de gobierno de datos.
 - 3) Se permite almacenar cualquier otra métrica o gráfico de explotación de datos adicional.

7. Historias de usuarios (*Product backlog*)

A continuación, se listan las historias de usuario. La ponderación de *story points* se realiza considerando 1 punto = 1 día:

1. Integración MQTT

- Detalle: Como arquitecto quiero que el envío de los datos desde el robot al *back-end* se realice mediante MQTT.
- Esfuerzo: 5 puntos
- Criterio de aceptación: funcionalidad verificada y documentación.

2. Arquitectura operacional

- Detalle: Como arquitecto quiero que la arquitectura de tanto los *front-ends* como el *back-end* sean web exponiendo APIs vía servicios REST. La interfaz gráfica debe ser implementada en Angular u otro framework MVC/MVVC de Javascript.
- Esfuerzo: 37 puntos
- Criterio de aceptación: funcionalidad verificada y documentación.

3. Arquitectura analítica

- Detalle: Como arquitecto quiero que las métricas de valor sean generadas de forma *batch* mediante un *pipeline* de datos.
- Esfuerzo: 27 puntos
- Criterio de aceptación: funcionalidad verificada y documentación.

4. Arquitectura blockchain

- Detalle: Como arquitecto blockchain quiero que la infraestructura seleccionada permita el despliegue de dApps y *smart contracts* de manera escalable, simple y que el desarrollo sea fácil de mantener y mejorar.
- Esfuerzo: 37 puntos
- Criterio de aceptación: funcionalidad verificada y documentación.

5. Métricas agregadas

- Detalle: Como usuario quiero una métrica que me indique los valores agregados de mínimo, máximo y promedio de cada parámetro ambiental agrupado por fecha y coordenadas.
- Esfuerzo: 15 puntos
- Criterio de aceptación: funcionalidad verificada y documentación.

6. Métricas históricas

- Detalle: Como usuario quiero poder ver en series de tiempo los valores medidos para los diferentes parámetros ambientales.
- Esfuerzo: 18 puntos
- Criterio de aceptación: funcionalidad verificada y documentación.

7. Gráficos de explotación

- Detalle: Como usuario quiero contar con métricas gráficas (*heatmap*, *pie*, etc) para poder visualizar más fácilmente los agregados.
- Esfuerzo: 20 puntos
- Criterio de aceptación: funcionalidad verificada y documentación.

8. Smart contracts

- Detalle: Como arquitecto quiero que el acceso a los datos en blockchain se realice por medio de la implementación de *smart contracts*.
- Esfuerzo: 20 puntos
- Criterio de aceptación: funcionalidad verificada, tests y documentación.

9. dApp

- Detalle: Como arquitecto quiero que el acceso a los *smart contracts* se realice desde una dApp utilizando una biblioteca Web3.
- Esfuerzo: 20 puntos
- Criterio de aceptación: funcionalidad verificada, tests y documentación.

10. Integración continua

- Detalle: Como arquitecto quiero garantizar la calidad del producto incluyendo un pipeline de integración continua como parte del proceso de construcción.
- Esfuerzo: 3 puntos
- Criterio de aceptación: funcionalidad verificada y documentación.

11. Encriptación punto a punto

- Detalle: Como ingeniero de seguridad quiero que los datos transmitidos en cada segmento viajen encriptados mediante certificados.
- Esfuerzo: 2 puntos
- Criterio de aceptación: funcionalidad verificada, tests y documentación.

8. Entregables principales del proyecto

Los entregables del proyecto son:

- Video demostrativo del uso de la plataforma.
- Código fuente del firmware.
- Código fuente del software.
- Documentación:
 1. Manual de usuario.
 2. Memoria final.
 3. Documentación de arquitectura técnica del sistema (hardware y software).

9. Desglose del trabajo en tareas

A continuación, se puede apreciar el conjunto de actividades y tareas que se realizarán durante el desarrollo del proyecto.

1. Desarrollo cloud (214 hs)
 - 1.1. Análisis y selección de arquitectura cloud (18 hs).
 - 1.2. Setup de arquitectura cloud (18 hs).
 - 1.3. POC de backend cloud (16 hs).
 - 1.4. POC front-end negocio (16 hs).
 - 1.5. POC front-end admin (16 hs).
 - 1.6. Desarrollo de funcionalidad cloud (39 hs).
 - 1.7. Testing front-back cloud (5 hs).
 - 1.8. POC data pipeline analítico (16 hs).
 - 1.9. Desarrollo de data pipeline analítico (45 hs).
 - 1.10. Testing data pipeline analítico (5 hs).
 - 1.11. POC integración MQTT robot-cloud (5 hs).
 - 1.12. Integración MQTT robot-cloud (15 hs).
2. Desarrollo blockchain (240 hs)
 - 2.1. Analisis y seleccion de arquitectura blockchain (34 hs).
 - 2.2. Setup de arquitectura blockchain (35 hs).
 - 2.3. POC smart contract (30 hs).
 - 2.4. Desarrollo smart contracts (50 hs).
 - 2.5. POC dApp (30 hs).
 - 2.6. Desarrollo dApp (45 hs).
 - 2.7. Testing blockchain (16 hs).
3. Set-up ambiente de integración continua (actividad opcional) (16 hs)
 - 3.1. Set-up imagen Docker con código de proyecto (10 hs).
 - 3.2. Set-up servicio de integración continua cloud (4 hs).
 - 3.3. Configuración con Github para tomar los commits y ejecución de builds (2 hs).
4. Documentación (100 hs)
 - 4.1. Escritura de manual de usuario (24 hs).
 - 4.2. Escritura de memoria final (40 hs).
 - 4.3. Escritura de documentación de arquitectura técnica del sistema (24 hs).
 - 4.4. Creación del video demostrativo de uso (12 hs).
5. Gestión del proyecto (45 hs)
 - 5.1. Definición de alcance, funcionalidades e historias de usuario (3 hs).
 - 5.2. Armado del plan de actividades y tareas (3 hs).

- 5.3. Reconocimiento de riesgos (3 hs).
- 5.4. Definición de proceso de calidad (2 hs).
- 5.5. Confección de documentación de planificación de proyecto (10 hs).
- 5.6. Seguimiento y control de hitos, desvíos y riesgos(24 hs).

Cantidad total de horas: (615 hs)

10. Diagrama de Activity On Node

A continuación, se detalla la lista de actividades que se realizarán durante el proyecto con los tiempos expresados en días y horas, considerando que habrá solamente un recurso asignado con una dedicación de 3 horas diarias solamente días hábiles.

Id	Tarea	Duración	Dependencia	Predecesora
1	Set-up integración continua (opcional)	5 d / 16 h	-	-
2	Desarrollo cloud	71 d / 214 h	-	-
3	Desarrollo blockchain	80 d / 240 h	-	-
4	Gestión del proyecto	15 d / 45 h	-	-
5	Documentación	33 d / 100 h	-	-
TOTAL		205 días / 615 horas		

Como se puede apreciar la estimación es de 205 días, siendo en total 41 semanas o 10 meses de trabajo con las restricciones planteadas.

A continuación, se puede apreciar en la figura 3, el camino crítico está formado por la tarea [3] (*desarrollo blockchain*) y su duración es 80 días.

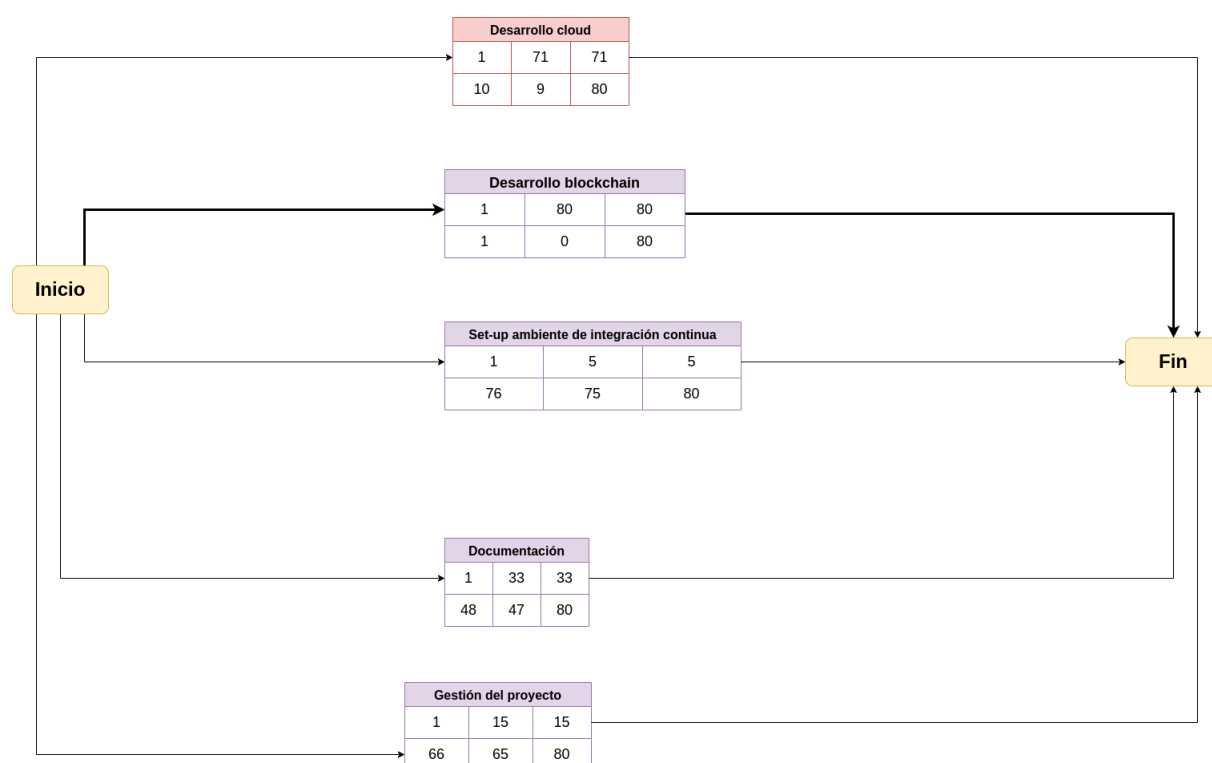


Figura 3. Diagrama de *Activity-On-Node*.

11. Diagrama de Gantt

A continuación, se puede apreciar el diagrama de Gantt general en la figura 4, y los detalles de sus tareas en las figuras 5, 6 y 7.

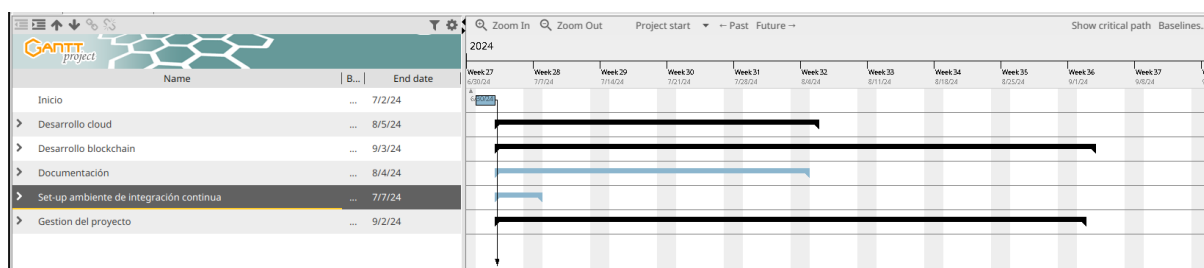


Figura 4. Diagrama de Gantt general.

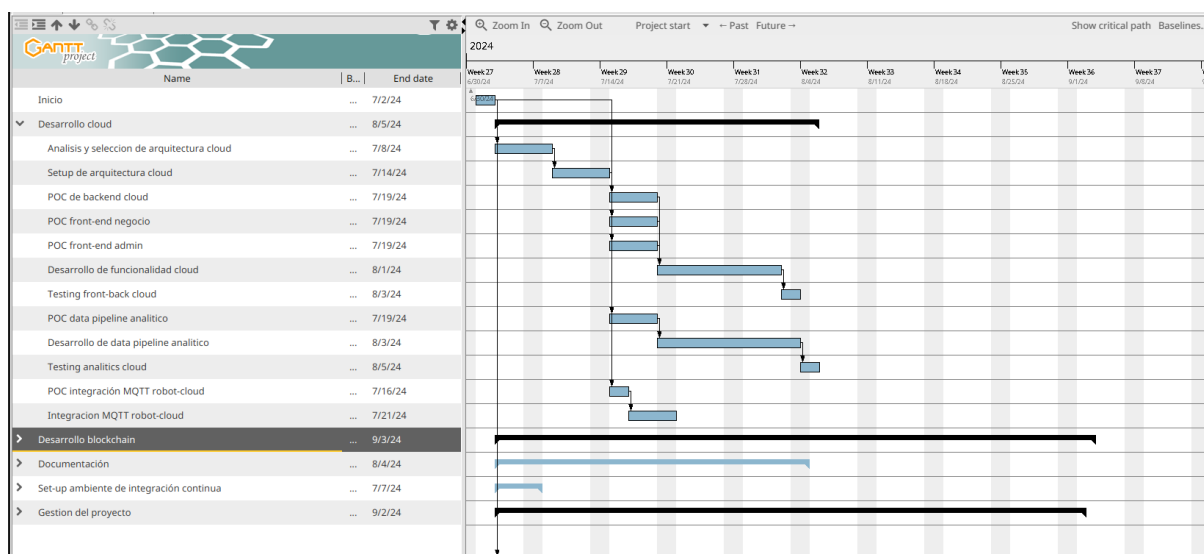


Figura 5. Diagrama de Gantt detallado (parte 1).

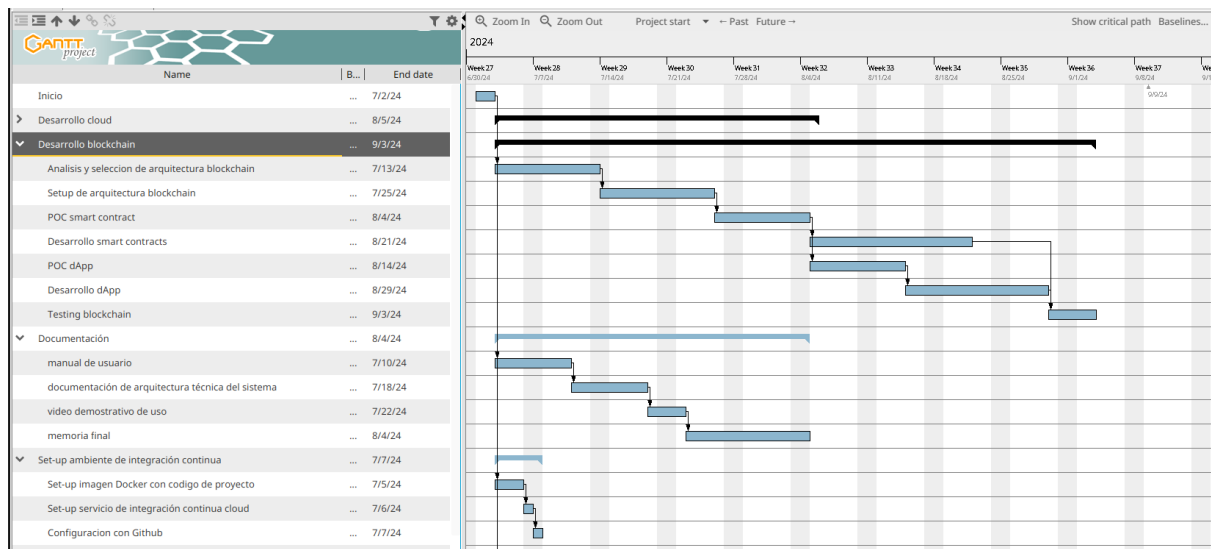


Figura 6. Diagrama de Gantt detallado (parte 2).

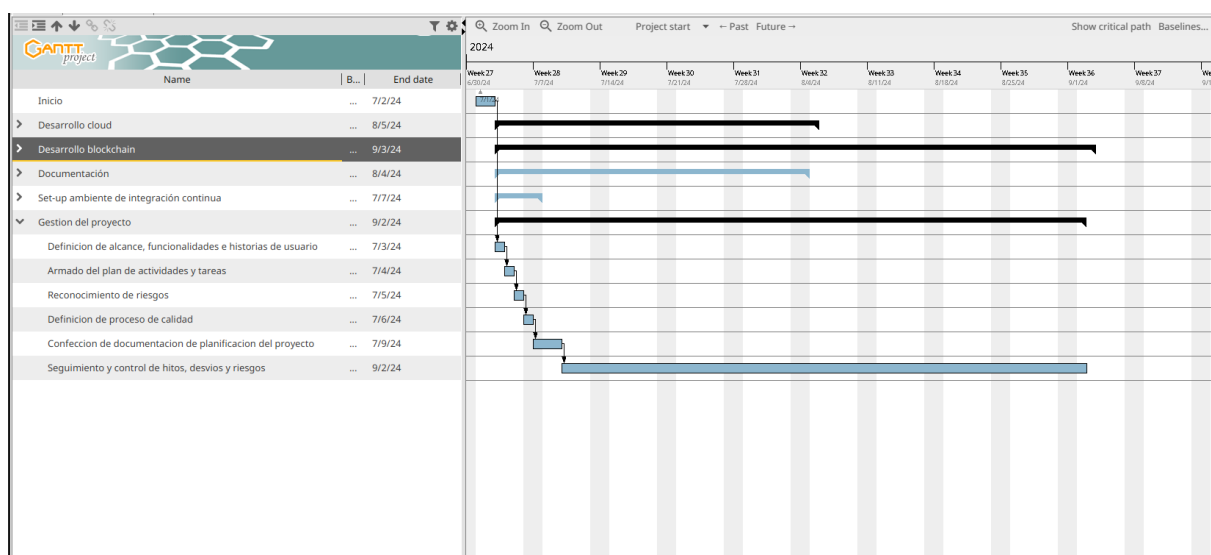


Figura 7. Diagrama de Gantt detallado (parte 3).

12. Presupuesto detallado del proyecto

El siguiente cuadro presenta los costos en dólares estadounidenses estimados para el proyecto:

COSTOS DIRECTOS			
Descripción	Cantidad	Valor unitario	Valor total
Procesamiento cloud	1	\$ 50,00	\$ 50,00
Almacenamiento cloud	1	\$ 20,00	\$ 20,00
Transferencia de datos cloud	1	\$ 50,00	\$ 50,00
Alojamiento IPFS	1	\$ 20,00	\$ 20,00
Varios	1	\$ 20,00	\$ 20,00
TOTAL			\$ 160,00

13. Gestión de riesgos

1. Riesgo de demora

- Severidad (S): 9 - Teniendo en cuenta que solo habrá un recurso (el alumno) asignado al proyecto desarrollando el producto, una demora en cualquier tarea puede implicar demora en la fecha de entrega.
- Ocurrencia (O): 5 - Dado el desafío de innovación planteado por el alcance el la probabilidad de demora es media.

2. Riesgo de no contar con toda la funcionalidad deseada

- Severidad (S): 9 - El no cumplimiento con la funcionalidad deseada pone en riesgo el éxito del proyecto.
- Ocurrencia (O): 5 - Dado el desafío de innovación planteado por el alcance, la probabilidad de no contar con toda la funcionalidad es media.

3. Riesgo de calidad insuficiente

- Severidad (S): 4 - La calidad insuficiente no pone en riesgo el cumplimiento con la funcionalidad pero si compromete la estabilidad y resistencia a fallas del producto, por lo que puede desencadenar en un producto poco o menos confiable.
- Ocurrencia (O): 4 - Se estima que con las técnicas empleadas durante el desarrollo del producto, este riesgo tiene una baja probabilidad de ocurrencia.

4. Riesgo de desvío en costos

- Severidad (S): 5 - La ocurrencia de este riesgo impacta en los costos del proyecto, pero no imposibilita ni demora la entrega del producto.
- Ocurrencia (O): 8 - Teniendo en cuenta que los precios son estimados sin haber cerrado la definición de la arquitectura final y la inflación argentina, es muy probable que exista un desvío en costos.

5. Riesgo de indisponibilidad de recursos

- Severidad (S): 5 - Al momento de realizar el presente plan se identifican ciertos recursos y se asume que será posible disponer de ellos. No obstante, existe el riesgo de que esto no suceda así, y sea más difícil por ejemplo adquirir *tokens* de prueba por medio de *Faucets* para el acceso a redes de desarrollo blockchain; realizar el despliegue de la dApp; acceder a servicios cloud que soporten todas las prestaciones necesarias por la arquitectura; o haya recursos no directamente asociados al proyecto pero cuya ausencia lo afectan, como por ejemplo fallas en el acceso a internet, el mal funcionamiento de la computadora utilizada para su desarrollo, fallas en el hardware del robot que hagan necesario su reparación, etc.
- Ocurrencia (O): 2 - Se espera que la probabilidad de ocurrencia de este riesgo sea realmente baja.

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

Riesgo	S	O	RPN	S*	O*	RPN*
Riesgo de demora en la entrega	9	5	45	9	1	9
Riesgo de no contar con toda la funcionalidad deseada	9	5	45	10	1	10
Riesgo de calidad insuficiente	6	4	24	4	2	8
Riesgo de desvío en costos	5	8	40	5	3	15
Riesgo de indisponibilidad de recursos	5	2	10	-	-	-

Criterio adoptado: Se tomarán medidas de mitigación en los riesgos cuyos números de RPN sean mayores a 15.

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:

1. Riesgo de demora en la entrega: las posibles causas del evento asociado están vinculadas a situaciones no controlables ni predecibles que impactan de alguna manera en la disponibilidad de tiempo o alguno de los recursos necesarios para la realización del proyecto. Con el fin de cumplir con la entrega de la funcionalidad en la fecha acordada se consideran como posibles acciones de mitigación la eliminación (o no realización) de otras tareas dentro del plan, como por ejemplo, tareas de documentación, de testing, y de ser necesario, de desarrollo de funcionalidad. Se asume que la eliminación de estas tareas ponen en riesgo la calidad del producto y/o contar con toda la funcionalidad esperada.
 - Nueva Severidad (S*): 9 - No cambia.
 - Nueva Ocurrencia (O*): 1.
2. Riesgo de no contar con toda la funcionalidad deseada: las posibles causas del evento asociado están vinculadas a situaciones que impactan de alguna manera en la viabilidad o desarrollo de alguna de las funcionalidades en el tiempo planificado. Por este motivo se considera como herramientas de mitigación: sacrificar algún otro entregable (como por ejemplo la cobertura de testing y/o documentación, lo cual puede implicar sacrificar calidad, mantenibilidad y/o usabilidad respectivamente), o bien redefinir el alcance y funcionalidad en base a lo que es posible desarrollar.
 - Nueva Severidad (S*): 10 - Dado que tras la mitigación se incrementa el impacto por pérdida de calidad.

- Nueva Ocurrencia (O*): 1 - Se reduce mucho la probabilidad de ocurrencia dado que se agrega tiempo para el desarrollo de funcionalidad eliminando el tiempo empenado para el desarrollo de tests y/o documentación.
- 3. Riesgo de calidad insuficiente: las posibles causas del evento asociado están vinculadas a la falta de estabilidad del producto, sea por una arquitectura demasiado compleja o ineficiente, o la presencia de bugs en el desarrollo. Para mitigar este problema se plantea emplear más tiempo en el desarrollo de pruebas de concepto y definición de la arquitectura, además incrementar las prácticas de testing y CI/CD (como actividad opcional) siempre que esto no dispare el riesgo 1, el cual podría generar una demora una demora en el proyecto.
 - Nueva Severidad (S*): 6 - No cambia.
 - Nueva Ocurrencia (O*): 2 - Se reduce la probabilidad de que esto suceda.
- 4. Riesgo de desvío en costos: Las posibles causas del evento que dispara este riesgo están vinculadas a la dificultad de estimar los costos de procesamiento, almacenamiento y transferencia de datos tanto en la nube pública como en la red blockchain a utilizar, debido a que la arquitectura no ha sido definida al momento de la escritura del presente plan y los costos vinculados a su implementación varían dependiendo del proveedor, red y tecnologías utilizadas. Además, indirectamente asociado a esto, es posible olvidar estimar algún componente. Finalmente el impacto de la inflación en Argentina puede ser otro causal del desvío en costos. Para mitigar el primer factor se agrega el item *Varios / Imprevistos* a la tabla de materiales con el fin de proveer holgura en el caso de no contemplar algún componente adicional. Para mitigar el factor inflación se plantean los precios en dólares americanos.
 - Nueva Severidad (S*): 5 - Esto no varía.
 - Nueva Ocurrencia (O*): 3 - Se reduce mucho la probabilidad de ocurrencia dado que la inflación del dólar estadounidense es menor que la del peso argentino.

14. Gestión de la calidad

1. Funcionalidad del *front-end* negocio: Métricas y gráficos de explotación.
 - 1.1. Verificación previo a la entrega: se verificará mediante la ejecución de tests de integración para esta funcionalidad.
 - 1.2. Validación: el cliente validará la funcionalidad en el producto final.
2. Funcionalidad del *front-end* admin: Control de la plataforma.
 - 2.1. Verificación previo a la entrega: se verificará mediante la ejecución de tests de integración para esta funcionalidad.
 - 2.2. Validación: el cliente validará la funcionalidad en el producto final.
3. Funcionalidad de la DApp, almacenamiento en blockchain y ejecución de *smart contracts*.
 - 3.1. Verificación previo a la entrega: se verificará mediante la ejecución de tests de integración para esta funcionalidad.

- 3.2. Validación: el cliente validará la funcionalidad en el producto final.
- 4. Documentación técnica, manual de usuario y memoria final.
 - 4.1. Verificación previo a la entrega: se verificará mediante la revisión de los documentos.
 - 4.2. Validación: el cliente validará los documentos.
- 5. Testing.
 - 5.1. Verificación previo a la entrega: Se verificará el cumplimiento con los tests por funcionalidad previo la integración de cada componente en el prototipo final.
 - 5.2. Validación: el cliente validará el reporte de los tests de integración.

15. Procesos de cierre

- Pautas de trabajo que se seguirán para analizar si se respetó el Plan de Proyecto original:
 - Responsable: Ing. Gonzalo F. Carreño:
 - Se evaluarán los requerimientos y los objetivos alcanzados frente a los planteados en el plan.
 - Se pondrá especial interés en verificar si se cumplieron los objetivos de tiempo y funcionalidad propuestos.
- Identificación de las técnicas y procedimientos útiles e inútiles que se emplearon, y los problemas que surgieron y cómo se solucionaron: - Responsable: Ing. Gonzalo F. Carreño:
 - Se evaluará cuál fue la configuración que mejores resultados arrojó para los objetivos planteados en el plan.
 - Se identificarán nuevas herramientas o procedimientos, en caso que corresponda.
- Indicar quién organizará el acto de agradecimiento a todos los interesados, y en especial al equipo de trabajo y colaboradores - Responsable: Ing. Gonzalo F. Carreño:
 - Luego de la presentación del proyecto mediante la defensa pública, se procederá a agradecer a todas las personas que participaron del desarrollo del proyecto, al director y a las autoridades de la CEIoT.