

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

Autor

Esp. Ing. Gonzalo Carreno

Director del trabajo

Esp. Ing. Sergio Alberino

Este plan de trabajo ha sido realizado en el marco de la asignatura TTF-A entre marzo y abril de 2025.

Tabla de contenido

1. Breve resumen del trabajo realizado hasta la fecha	2
2. Avance en las tareas	4
3. Cumplimiento de los requerimientos	5
4. Gestión de riesgos	6

IMPORTANTE: No borrar las consignas en cada una de las cuatro secciones de este documento, de forma tal que el jurado tenga claro qué es lo solicitado en cada caso, así como el significado de los símbolos y colores utilizados.

Revisión	Cambios realizados	Fecha
1.0	Creación del documento	28/03/2025

1. Breve resumen del trabajo realizado hasta la fecha

Elabore un detalle del estado del proyecto a la fecha. **Utilicé como mínimo dos páginas completas y como máximo tres páginas.** Explique muy brevemente en qué consiste su Trabajo Final, **aunque esa información esté más detallada en el Plan de Trabajo al cual su Jurado también tiene acceso.** Incluya imágenes y tablas según considere apropiado. **Indique con claridad por qué estima que podrá completar todos los faltantes (o al menos la gran mayoría) antes del inicio del Taller de Trabajo Final.**

El trabajo final consiste en la implementación de una solución IoT para un robot de exploración ambiental en un caso de uso en el que sea necesario almacenar datos críticos de una forma transparente e inmutable.

La arquitectura de la solución se basa en un sistema embebido, desarrollado previamente como proyecto final de la CESE, integrado a un back-end en la nube pública y una red blockchain para el almacenamiento de los datos críticos.

Al día de la fecha se cuenta con el siguiente grado de avance para las tareas detalladas en el plan de proyecto desarrollado en el marco de GTI durante el 2024:

- Desarrollo cloud: 30%
- Desarrollo blockchain: 70%
- Set-up CI/CD: 45%
- Documentación: 5%
- Gestión de proyecto (proceso continuo): 35-40% del avance total

Con respecto al desarrollo cloud, se han realizado las tareas de arquitectura y diseño de la solución en la nube pública AWS casi en su totalidad, habiéndolo desplegado algunos de los servicios necesarios para el funcionamiento de la solución.

Se implementó la prueba de concepto y posterior publicación del tópico MQTT utilizando el servicio AWS IoT Core, necesario para la comunicación de lecturas de exploración ambiental. Se implementó la seguridad del mismo y se configuró e integró el dispositivo robótico, con los correspondientes certificados de seguridad, para el envío, recepción y almacenamiento de las lecturas en AWS S3.

Además se hicieron pruebas de concepto para el procesamiento de datos usando AWS Athena, se configuró AWS Glue para poder reconocer los datos de las lecturas enviadas, y se hizo una prueba de concepto de la orquestación del data pipeline con AWS Step Functions.

Desde el punto de vista de los backends operacionales se hizo una prueba de concepto de un servicio Node.js desplegado como un contenedor Docker en el servicio AWS App Runner para estudiar su performance, flexibilidad y costos asociados.

Quedan pendiente varias tareas como el prototipo e implementación del modelo de datos, el sistema analítico, el desarrollo completo del sistema operacional, el sistema de visualización de resultados, la interfaz de administración de la plataforma, y el testing de cada uno de estos componentes .

Desde el punto de vista del desarrollo blockchain se realizaron las actividades de arquitectura y diseño de la solución, y prototipado de varios de los componentes.

Se decidió basar la solución en la red de Ethereum y se desarrollaron prototipos para los componentes smart contracts y dApp. Posteriormente se desarrolló gran parte de la funcionalidad de los mismos, optimizando el almacenamiento de los datos en la red. Los smart contracts fueron desplegados inicialmente de manera local como contenedores Docker utilizando Ganache, y posteriormente, tras la obtención de tokens ETH mediante faucets, y la integración con Metamask, se realizaron despliegues en la red de pruebas de Sepolia.

Utilizando el servicio Alchemy se integró la dApp (actualmente corriendo en un contenedor Docker de forma local) con los smart contracts desplegados en Sepolia. Se desarrollaron además, ciertos endpoints en el backend de la dApp para administrar los smart contracts y analizar sus costos operativos.

Queda aún pendiente el despliegue de la dApp en AWS App Runner, su integración con el tópico MQTT y AWS IoT Core para la redirección de las lecturas y su almacenamiento en la red blockchain. También queda pendiente el desarrollo del testing de todos los componentes blockchain.

Con respecto a las actividades incluidas en la metodología CI/CD, todo el desarrollo se basó en el uso de imágenes Docker y se encuentra visionado en Github. Queda pendiente la configuración del servicio para el lanzamiento de la compilación automática y versión de imágenes Docker en un repositorio central en AWS.

Desde el punto de vista de la documentación, se ha diagramado la arquitectura usando Draw.io, se han documentado las decisiones técnicas tomadas (y sus motivos), y se ha comenzado con la escritura de la memoria final, habiendo concluido los dos primeros capítulos (pendiente revisión y aprobación).

Queda pendiente comenzar con la documentación de arquitectura técnica, documento de manual de usuario y realizar los videos demostrativos de la plataforma, además del material necesario para la entrega final.

Desde el punto de vista de la gestión de proyecto, además del trabajo inicial con respecto a la planificación y análisis de riesgos, se ha realizado el seguimiento del avance de las tareas, documentando las decisiones tomadas y los control y gestión de riesgos del proyecto.

2. Avance en las tareas

a) Indicar a continuación para cada una de las tareas su estado de situación según su criterio, utilizando verde si considera que es satisfactorio, amarillo si considera que es insatisfactorio por sobrecostos y/o demoras, y rojo si lo considera muy insatisfactorio por sobrecostos y/o demoras.

Si a la fecha de completar este informe no está previsto que la tarea haya comenzado entonces deje la celda correspondiente en blanco, sin pintarla con ningún color.

En subcelda inferior izquierda colocar:

- ** si los recursos u horas utilizadas fueron o están siendo muy inferior a lo planificado.
- * si los recursos u horas utilizadas fueron o están siendo inferior a lo planificado.
- \$ si los recursos u horas utilizadas fueron o están siendo de acuerdo a lo planificado.
- \$\$ si los recursos u horas utilizadas fueron o están siendo superior a lo planificado.
- \$\$\$ si los recursos u horas utilizadas fueron o están siendo muy superior a lo planificado.

En subcelda inferior derecha colocar:

- -- si la tarea se ejecutó o se está ejecutando mucho más rápido de lo previsto
- - si la tarea se ejecutó o se está ejecutando más rápido de lo previsto
- = si la tarea se ejecutó o se está ejecutando en el tiempo previsto.
- + si la tarea se ejecutó o se está ejecutando con demoras.
- ++ si la tarea se ejecutó o se está ejecutando con demoras muy significativas.

IMPORTANTE: Indicar con **borde grueso** las tareas que forman parte del camino crítico

Cloud - Análisis y selección de arquitectura		Cloud - Set-up arquitectura		Cloud - POC backend		Cloud - POC frontend negocio	
\$	-	\$	-	\$	-	\$	=
Cloud - POC frontend admin		Cloud - Desarrollo funcionalidad		Cloud - Testing backend and frontend		Cloud - POC data pipeline analítico	
\$	=	\$	=	\$	=	\$	=
Cloud - Desarrollo modelo datos data pipeline analítico		Cloud - Desarrollo de procesamiento de datos data pipeline analítico		Cloud - Testing data pipeline analítico		Cloud - POC integración MQTT	
\$	=	\$	=	\$	=	\$	-
Cloud - Desarrollo integración MQTT robot							
\$	-						

Blockchain - Análisis y selección de arquitectura		Blockchain - Set-up arquitectura		Blockchain - POC Smart Contract		Blockchain - Desarrollo Smart Contract - Persistencia	
\$	-	\$	-	\$	-	\$	=
Blockchain - Desarrollo Smart Contract - Comportamiento		Blockchain - POC dApp		Blockchain - Desarrollo dApp		Blockchain - Testing	
\$	-	\$	-	\$	-	\$	=
CI/CD - Set-up ambiente		CI/CD - Set-up imagen Docker		CI/CD - Config. Github			
\$	+	\$	+	\$	=		
Documentación - Memoria técnica		Doc. arquitectura técnica		Documentación - Videos demostrativos		Documentación - Manual de usuario	
\$	-	\$	=	\$	+	\$	+
Gestión de proyecto - Definición de alcance		Gestión de proyecto - Armado plan		Gestión de proyecto - Reconocimiento riesgos		Gestión de proyecto - Def. proceso de calidad	
\$	=	\$	=	\$	=	\$	=
Gestión de proyecto - Doc. de planificación		Gestión de proyecto - Seguimiento y control					
	=	\$	=				

3. Cumplimiento de los requerimientos

a) Indicar a continuación para cada uno de los requerimientos el estado de situación según su criterio, utilizando verde si considera que ya se ha cumplido, amarillo si considera que aún no se ha cumplido pero se podrá cumplir, y rojo si considera que aún no se ha cumplido y tiene dudas si se podrá cumplir.

Si considera que es necesario modificar los requerimientos respecto a los indicados en la planificación inicial entonces incluya acá los requerimientos actualizados, **marcando en negrita** aquellos que son nuevos o se han modificado.

Req Funcional #1: Envío de datos vía MQTT
Req Funcional #2: Categorización de valores censados en el robot
Req Funcional #3: Almacenamiento de las mediciones ambientales
Req Funcional #4: Almacenamiento de datos críticos en medio inmutable
Req Funcional #5: Procesamiento de los datos de mediciones para generar métricas de valor.
Req Funcional #6: Interfaz frontend para administrador y negocio
Req No Funcional #1: Backend operacional
Req No Funcional #2: Backend analítico

Req No Funcional #3: Envío de datos vía MQTT
Req No Funcional #4: Uso de blockchain para lecturas críticas - Este requerimiento se repite con el requerimiento Funcional #2.
Req No Funcional #5: Implementación de Smart Contracts para la gestión de datos en blockchain.
Req No Funcional #6: Implementación de una dApp para comunicarse con los Smart Contracts.
Req No Funcional #7: Implementación de seguridad en el sistema.
Req Documentación #1: Video demostrativo
Req Documentación #2: Documentación de arquitectura técnica
Req Documentación #3: Manual de usuario
Req Documentación #4: Memoria final
Req Testing #1: Tests unitarios para cada componente
Req Testing #2: Tests funcional del sistema

Req Opcional #1-a: Despliegue dApp IPFS o cloud

Req Opcional #1-b: Agregado de hardware adicional en el robot

Req Opcional #1-c: Automatización de infraestructura como código

Req Opcional #2-a: Almacenamiento de datos adicionales

Req Opcional #2-b: Gobierno de datos

Req Opcional #2-b: Métricas adicionales

4. Gestión de riesgos

a) Indicar a continuación para cada uno de los riesgos el estado de situación según su criterio, utilizando verde si considera que el riesgo ya no se manifestará o es muy improbable que se manifieste, amarillo si considera que es posible que es improbable que el riesgo se manifieste o si se manifiesta estima que será fácilmente controlado, y rojo si considera que es muy probable que el riesgo se manifieste y que no pueda ser controlado fácilmente.

Si considera que es necesario modificar los riesgos respecto a los presentados en la planificación inicial entonces incluya acá los riesgos actualizados, **marcando en negrita** aquellos que son nuevos o se han modificado, e indicando para ellos los valores de S, O y RPN, junto con su respectiva justificación.

Riesgo #1: Demora
Riesgo #2: No contar con toda la funcionalidad
Riesgo #3: Calidad insuficiente
Riesgo #4: Desvío de costos
Riesgo #4: Indisponibilidad de recursos