

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

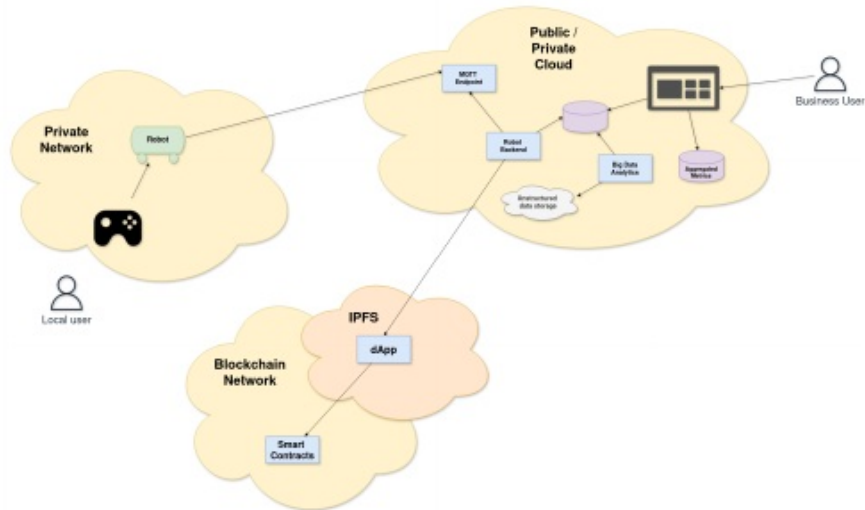


Figura 1. Arquitectura del sistema

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.

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

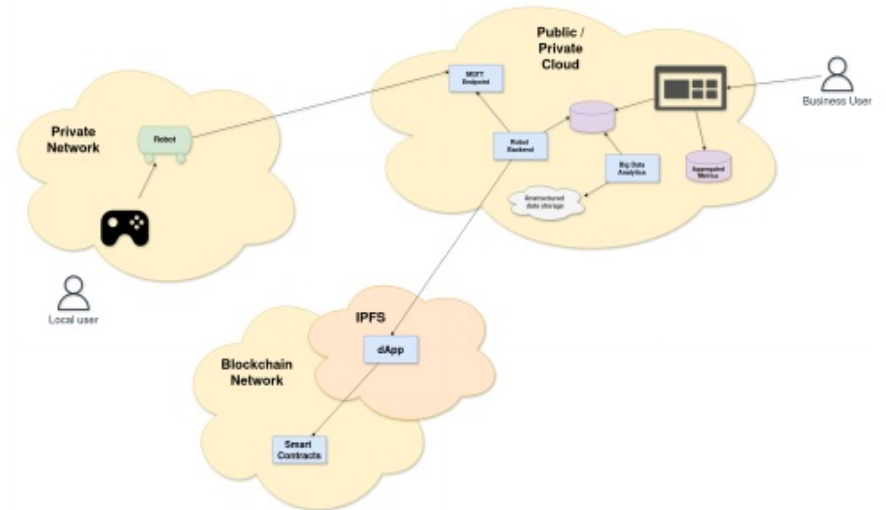


Figura 1. Arquitectura del sistema.

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.

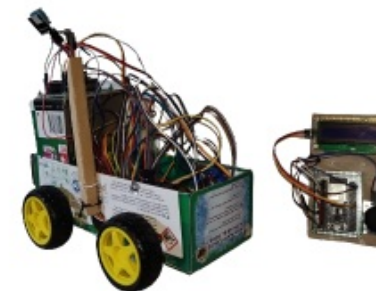


Figura 2. Robot de exploración ambiental.

- 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 máximo o mínimo 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 debe brindar 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.

- 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 máximo o mínimo 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. ~~La~~ plataforma debe brindar 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.

14. Gestión de la calidad

1. Funcionalidad del *front-end* de negocio: métricas y gráficos de explotación.
 - 1.1. Verificación previa a la entrega: se verificará esta funcionalidad mediante la ejecución de tests funcionales y unitarios.
 - 1.2. Validación: el cliente validará de forma manual la ejecución de la funcionalidad de los reportes y gráficos de explotación en el producto final.
2. Funcionalidad del *front-end* de administración: control de la plataforma.
 - 2.1. Verificación previa a la entrega: se verificará esta funcionalidad mediante la ejecución de tests funcionales y unitarios.
 - 2.2. Validación: el cliente validará de forma manual la ejecución de la funcionalidad de administración de la plataforma en el producto final.
3. Funcionalidad de la DApp, almacenamiento en blockchain y ejecución de *smart contracts*.
 - 3.1. Verificación previa a la entrega: se verificará esta funcionalidad mediante la ejecución de tests funcionales y unitarios.
 - 3.2. Validación: el cliente validará de forma manual la ejecución de la funcionalidad en la dApp y de los *smart contracts* accediendo a la red blockchain mediante el navegador web.
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 la completitud y claridad de los documentos entregados.
5. Testing.
 - 5.1. Verificación previo a la entrega: Se verificará el nivel de cumplimiento, y de ser posible la cobertura, de los casos de test por funcionalidad 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.

14. Gestión de la calidad

1. Funcionalidad del *front-end* de negocio: métricas y gráficos de explotación.
 - 1.1. Verificación previa a la entrega: se verificará esta funcionalidad mediante la ejecución de tests funcionales y unitarios.
 - 1.2. Validación: el cliente validará de forma manual la ejecución de la funcionalidad de los reportes y gráficos de explotación en el producto final.
2. Funcionalidad del *front-end* de administración: control de la plataforma.
 - 2.1. Verificación previa a la entrega: se verificará esta funcionalidad mediante la ejecución de tests funcionales y unitarios.
 - 2.2. Validación: el cliente validará de forma manual la ejecución de la funcionalidad de administración de la plataforma en el producto final.
3. Funcionalidad de la DApp, almacenamiento en blockchain y ejecución de *smart contracts*.
 - 3.1. Verificación previa a la entrega: se verificará esta funcionalidad mediante la ejecución de tests funcionales y unitarios.
 - 3.2. Validación: el cliente validará de forma manual la ejecución de la funcionalidad en la dApp y de los *smart contracts* accediendo a la red blockchain mediante el navegador web.
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 la completitud y claridad de los documentos entregados.
5. Testing.
 - 5.1. Verificación previo a la entrega: Se verificará el nivel de cumplimiento, y de ser posible la cobertura, de los casos de test por funcionalidad 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.

- 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.