



Facultad 4

Monitoreo y Control de datos en Estación de Carga de bajo Costo

Informe Técnico de la asignatura de Proyecto de Investigación y Desarrollo II

Autor(es): Yonier Lora Ortega.

Tutor(es): Yanisleydis Rodríguez Tamayo.

Co-tutor: Julio Alberto Leyva Durán.

La Habana, julio de 2023.

Año 64 del Triunfo de la Revolución.

DATOS DE CONTACTO

Julio Alberto Leyva Durán: Profesor Asistente en la Facultad 4, de la UCI. Especialista Superior en la Unidad Básica de Integración de la XETID.

Yanisleydis Rodríguez Tamayo.: Especialista B del Centro de Tecnologías Interactivas de la Facultad 4 en la UCI.

RESUMEN

El objetivo principal de la investigación se refiere al monitoreo y control de las variables de estados en una Estación de Carga con elementos de hardware libre, específicamente en la plataforma Arduino, vinculado a componentes con sensores y actuadores que se encargan de gestionar la carga de un vehículo de tipo ligero. La comunicación es a través del canal de comunicación RS232, en un formato de datos json.

PALABRAS CLAVE

Estación de Carga, Arduino, Comunicación RS-232.

ABSTRACT

The main objective of the research refers to the monitoring and control of state variables in a Charging Station with free hardware elements, specifically on the Arduino platform, linked to components with sensors and actuators that are responsible for managing the charging of a light type vehicle. Communication is through the RS232 communication channel, in a json data format.

KEYWORDS

Charging Station, Arduino, RS-232 Communication.

TABLA DE CONTENIDOS

INTRODUCCIÓN	5
DESARROLLO	7
I. Nombre del Epígrafe I	7
II. Nombre del Epígrafe II	8
III. Nombre del Epígrafe III	Error! Bookmark not defined.
IV. Nombre del Epígrafe IV	9
V. Nombre del Epígrafe V	14
VI. Nombre del Epígrafe VI	14
VII. Nombre del Epígrafe VII	Error! Bookmark not defined.
VIII. Nombre del Epígrafe VIII	Error! Bookmark not defined.
CONCLUSIONES	15
RECOMENDACIONES	20
REFERENCIAS BIBLIOGRÁFICAS	22
ANEXOS	23

INTRODUCCIÓN

La revolución industrial actualmente se encuentra inmersa en las investigaciones de la energía verdes, especialmente en el diseño de estaciones de carga para vehículos eléctricos, como automóviles, motocicletas o bicicletas eléctricas. Suele estar ubicada en lugares públicos como estacionamientos, centros comerciales o en la calle, y puede ser operada por diferentes proveedores de servicios de carga.

La estación de carga en sí misma puede ser una estructura independiente o estar integrada en un poste o pared. En general, las estaciones de carga están equipadas con un enchufe o conector que se conecta al puerto de carga del vehículo. También puede haber una pantalla digital que muestra información sobre el proceso de carga, como el tiempo de carga restante o el nivel de batería.

Algunas estaciones de carga pueden proporcionar diferentes niveles de carga, que se refieren a la velocidad a la que se carga la batería del vehículo. Por ejemplo, una estación de carga rápida puede cargar un vehículo en menos de una hora, mientras que una estación de carga lenta puede tardar varias horas.

En general, las estaciones de carga son una parte crucial de la infraestructura necesaria para impulsar la adopción de vehículos eléctricos y reducir las emisiones de gases de efecto invernadero asociadas con los vehículos que funcionan con combustibles fósiles (Márquez, 2019).

Actualmente en el Centro de tecnologías interactivas de la Universidad de Ciencias Informáticas se lleva a cabo el desarrollo de Aplicaciones para el sector de la industria de la energía renovables, específicamente en la implementación de plataformas que gestionen diferentes Puntos de Cargas mediante el protocolo OCPP, para la implementación y comprensión de los conceptos de los componentes que intervienen en el proceso de un Punto de Carga se desarrolla prototipos de hardware que poseen la misma analogía de dichos Puntos de Carga, los cuales poseen un conjunto de funcionalidades bien definidas

que ayudan a los analistas, arquitectos, desarrolladores y probadores a corregir y limitar la gama de funcionalidades y requerimientos necesarios para el desarrollo de este tipo de producto.

Actualmente se ha desarrollado un prototipo de bajo costo económico que presenta las funcionalidades conceptuales básicas de una Estación de Carga, el mismo provee la medición y el control del consumo eléctrico de automóviles, motocicletas o bicicletas eléctricas. Todos los componentes están integrados en un mismo gabinete que mediante un display el usuario puede visualizar el consumo correspondiente a una determinada medición al día, pero no posee una herramienta que muestre la acumulación del consumo eléctrico o un historial, así como la autorización debida de un conector de alimentación eléctrica a realizar una determinada carga por un determinado usuario, y la rapidez con la que se necesite una determinada carga, así como la localización de la misma.

Partiendo de la situación anterior se plantea el siguiente problema de la investigación: ¿Cómo adquirir los datos provenientes de la Estación de Carga mediante la interface de comunicación RS-232?

Se define como **objeto de estudio** los procesos para la Medición y Control de datos una Estación de Carga en plataforma de Hardware Libre, y como **campo de acción** los procesos para la medición y control de datos en Estaciones de Carga mediante comunicación RS232 en plataforma de Hardware Libre.

Se define como **objetivo general**: Implementar una aplicación informática para el Monitoreo y Control de datos para una Estación de Carga mediante comunicación RS232 en plataforma de Hardware Libre.

DESARROLLO

En base a este proyecto de creación de una aplicación informática para el monitoreo y control de datos para una Estación de Carga mediante comunicación RS232 en plataforma de Hardware Libre, se realizó una búsqueda sobre temas y estudios antes desarrollados semejantes al proyecto, lo cual permitió obtener un análisis e instrucciones para tener una base y obtener conceptos más específicos.

1. Estación de Carga

Una estación de carga de vehículo eléctrico es, definida de manera formal, un elemento dentro de una infraestructura que proporciona suministro eléctrico a vehículos que funcionan a base de baterías (total o parcialmente). Las estaciones de carga suelen tener a disposición de los usuarios varios conectores que cumplen diferentes estándares.

1.1 Estación de Carga sobre plataforma de Hardware libre

Las plataformas de hardware libre son sistemas electrónicos diseñados con el objetivo de ser compartidos públicamente, permitiendo que cualquier persona pueda acceder a la información necesaria para replicar el diseño y construir su propia versión del dispositivo. Estas plataformas se caracterizan por su enfoque en la colaboración y la comunidad, y a menudo son impulsadas por una comunidad de desarrolladores y entusiastas de la tecnología.

Actualmente Arduino y Raspberry Pi son plataformas electrónica programable que se utilizan para crear proyectos interactivos y dispositivos electrónicos. El diseño del prototipo de Estación de Carga es basado en una placa Arduino, con diferentes periféricos y componentes como se ilustra en la figura 1.

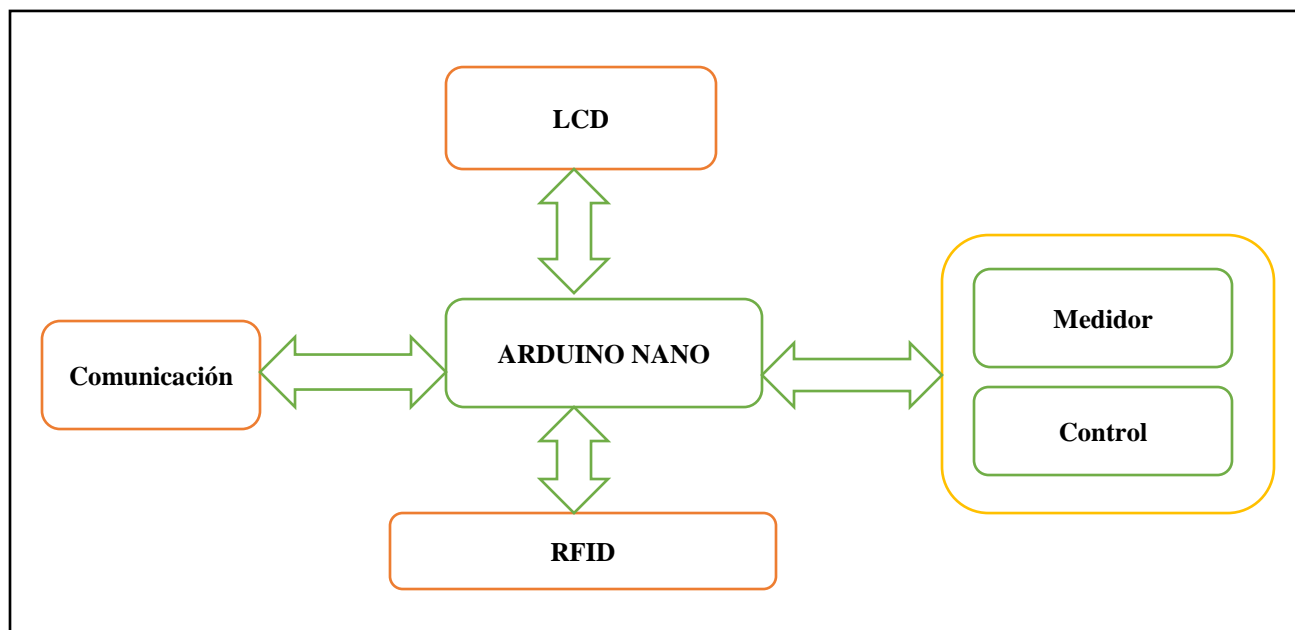


Figura 1: Diagrama de bloques del prototipo de Estación de Carga.

A modo de resumen se detallan los elementos visualizados en la figura 1:

Arduino nano: Núcleo de procesamiento donde se ejecuta el sistema embebido para gestionar los procesos de medición y control de la Estación de Carga.

Bloque Medidor y Control: Está compuesto por el sensor (SCT-013 o ACS712) y el relé SRD-05VDC-SL-C. Este bloque tiene la responsabilidad de realizar las mediciones y conmutación de la corriente eléctrica consumido por el vehículo eléctrico.

LCD: Este dispositivo se encarga de visualizar los valores de voltaje, corriente y potencia consumida por el vehículo eléctrico.

RFID: El sensor RFID es el encargado de leer el código de una tarjeta de este tipo, la cuál portará el usuario para el acceso a cargar el vehículo eléctrico. Para la versión de prueba de este desarrollo no se tiene en cuenta la gestión de permisos de los usuarios.

Comunicación: Por este canal se establece la comunicación mediante la interface RS232 y el conector DB9 (COM7), con los parámetros estándares (Velocidad de Transferencia: 9600, Bits de Datos: 8, Bits de Paro: 1, Paridad: Ninguna).

1.1.2 Formato de los Datos encapsulados en la Estación de Carga

El formato de datos se refiere a la forma en que se estructuran y organizan los datos en un sistema de almacenamiento o transmisión. Esto incluye cómo se representan los datos en términos de bits, bytes, palabras, números enteros, números de punto flotante, caracteres y otros tipos de datos (S. Mukherjee,2019).

El núcleo de procesamiento Arduino nano mediante la interface de comunicación RS232, maneja las solicitudes de los mensajes mediante la implementación del formato de datos JSON(*JavaScript Object Notation, por su siglas en inglés*), utiliza una estructura de pares clave-valor para representar los datos, donde los datos se agrupan en objetos y los objetos se pueden anidar en otros objetos. Cada par clave-valor consta de una llave (una cadena de texto) y un valor asociado (que puede ser un número, una cadena de texto, un objeto o una matriz). La sintaxis de JSON es simple y su estructura para el intercambio de datos es la siguiente figura 2.

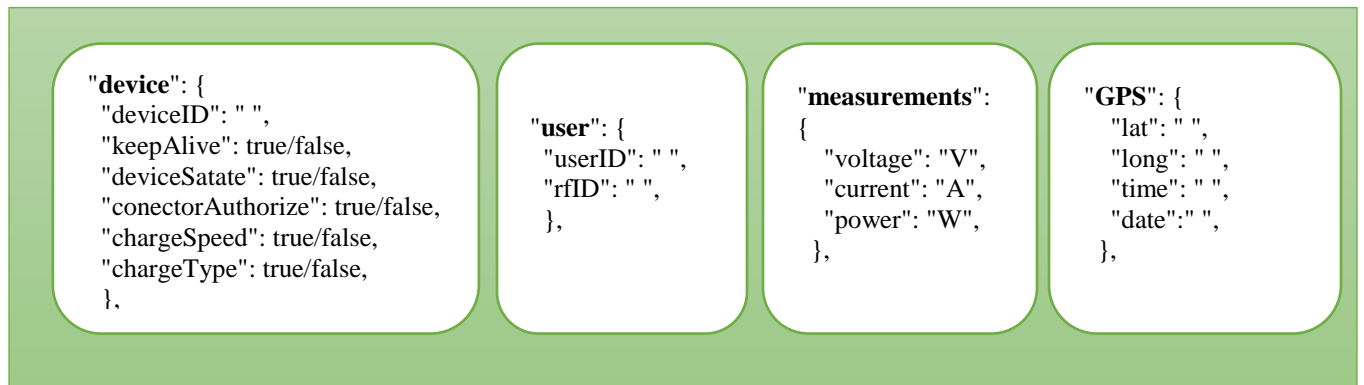


Figura 2: Sintaxis de JSON para el intercambio de datos.

1.2. Herramientas, Tecnologías y Metodología aplicadas en el desarrollo de la solución

1.2.1 Entorno de Desarrollo Integrado

Java NetBeans IDE es un entorno de desarrollo integrado (IDE) gratuito y de código abierto para el desarrollo de aplicaciones Java. Soporte multiplataforma: NetBeans IDE es compatible con Windows, macOS y Linux, lo que permite a los desarrolladores trabajar en diferentes plataformas. Soporte para múltiples lenguajes de programación: Además de Java, NetBeans IDE también admite otros lenguajes de programación como C, C++, PHP, HTML, CSS y JavaScript.

Conjunto de bibliotecas empleadas en la solución soportadas por Java NetBeans:

Biblioteca para el acceso al puerto serie RS232

- `import com.fazecast.jSerialComm.SerialPort;`

Biblioteca para el acceso a Fichero

- `import java.io.File;`
- `import java.io.IOException;`
- `import java.io.FileWriter;`
- `import java.util.Scanner;`
-

Biblioteca para el manejo de JSON

- `import org.json.JSONObject;`

Figura 3: Conjunto de Bibliotecas empleadas en el Proyecto.

1.2.2 Visual Paradigm 16.2

Visual Paradigm es una herramienta Computer Aided Software Engineering (CASE, por sus siglas en inglés) que emplea UML como lenguaje de modelado. Las herramientas de Ingeniería de Software Asistida por Computadora se pueden definir como un conjunto de programas y ayudas que dan asistencia a los analistas, desarrolladores e ingenieros de software, durante el ciclo de vida de un software, proporcionándole un aumento en su productividad y logrando un mayor ahorro de tiempo (Ambler 2017).

Visual Paradigm soporta el ciclo de vida de desarrollo de software completo, desde el análisis y diseño hasta la construcción, pruebas e incluso despliegue. Permite representar gráficamente varios diagramas y facilita la generación de código. Es una herramienta multiplataforma, fácil de instalar y utilizar. A continuación, se listan algunas de sus principales ventajas según el sitio oficial (Visual Paradigm 2022):

- Soporte nativo de UML.
- Diagramas de procesos de negocio.
- Diagramas entidad relación y transformación en tablas de bases de datos.
- Soporte a técnicas de ingeniería inversa.
- Capacidad de convertir un modelo en código.

1.2.3 Lenguaje de Modelado (UML) 8.0

Como lenguaje de modelado se seleccionó UML en su versión 8.0, dado que es el lenguaje de modelado de Sistemas de software más conocido y utilizado en la actualidad. Es una consolidación de muchas de las notaciones y conceptos más usados en la metodología orientada a objetos. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios, funciones del sistema, aspectos concretos como expresiones de lenguajes de programación, componentes de software reutilizables y esquemas de bases de datos.

Además, permite especificar y visualizar un sistema. Se puede utilizar para definir un sistema de software, detallar los artefactos, especificar su documentación y construcción. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, contando con varios tipos de diagramas para llegar a representar los diferentes puntos de vista de un sistema (Jacobson, Booch, Rumbaugh 2006).

1.2.4 Metodología de desarrollo

A decir de (Pressman 2021), “Una metodología es el conjunto ordenado de pasos a seguir para cumplir un objetivo. Dicho objetivo, en la ingeniería de software, es el desarrollo de software de alta calidad que cumple con las necesidades del cliente dentro de un plan y un presupuesto predecible. Para esto es necesario proveer un enfoque disciplinario para asignar tareas y responsabilidades dentro del desarrollo del sistema, determinar un camino metódico y sistemático para desarrollar, diseñar y validar una arquitectura y reducir en gran medida los riesgos que representa la construcción de sistemas de software”.

Metodología AUP variante UCI

Para lograr la mayor eficiencia en el ciclo de vida de la actividad productiva en la UCI se decidió una variación en la metodología AUP (Rodriguez, Vazquez, Aliaga 2012). La metodología consta de 3 fases:

Inicio: durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance de este, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no.

Ejecución: en esta fase se ejecutan las actividades requeridas para desarrollar el software incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se realizan pruebas al producto.

Cierre: en esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Para el desarrollo de la propuesta de solución se utiliza el escenario 4 de historia de usuario, debido a que el proyecto fue evaluado y como resultado se obtuvo un negocio bien definido. El cliente está siempre acompañado al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. La duración del proyecto no es muy extensa, y no se debe generar mucha información y el equipo de desarrollo está conformado por una sola persona.

1.3. Diseño de la Solución

A continuación se describe la propuesta de solución, las tareas de ingeniería realizadas y los principales artefactos obtenidos en esta fase de desarrollo, los requerimientos funcionales y no funcionales que debe cumplir, así como las Historias de usuarios correspondientes. Por último, se describe la arquitectura del sistema a desarrollar.

1.3.1 Levantamiento de Requisitos

Los requisitos de un sistema son la descripción de lo que es sistema debe hacer, los servicios que proporciona y las restricciones de sus operaciones. Dichos requerimientos reflejan las necesidades de los clientes del sistema. El proceso de obtención, análisis, documentación y validación de esos servicios y restricciones se llama ingeniería de requisitos. Los requisitos se clasifican en funcionales (servicios que el sistema debe proporcionar) y no funcionales (restricciones de los servicios o funciones ofrecidos por el sistema) (Sommerville 2020). A continuación, se describe cómo se obtuvieron los requisitos de la propuesta de solución, su especificación, descripción y validación.

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera que este debe reaccionar a entradas particulares y cómo se debe comportar en situaciones particulares (Sommerville 2011).

RF-1 Seleccionar puerto de conexión: la aplicación debe permitir escoger el puerto para la conexión con el dispositivo Estación de Carga.

RF-2 Conectar: la aplicación debe permitir conectarse al dispositivo Estación de Carga.

RF-3 Desconectar: la aplicación debe permitir al dispositivo Estación de Carga.

RF-4 Iniciar Medición de datos: la aplicación debe permitir iniciar la medición de los datos (*valores de voltaje, corriente y potencia consumida por el vehículo eléctrico*) del dispositivo Estación de Carga.

RF-5 Finalizar Medición de datos: la aplicación debe permitir terminar el proceso de Medición de datos de la Estación de Carga.

RF-6 Salvar datos Medidos: la aplicación debe permitir almacenar los datos en un fichero (formato json).

RF-7 Visualizar datos en tiempo real: la aplicación debe mostrar los datos Medidos.

RF-8 Establecer parámetro de control de carga lenta o rápida: La aplicación debe de permitir activar/desactivar la conmutación de carga lenta o rápida.

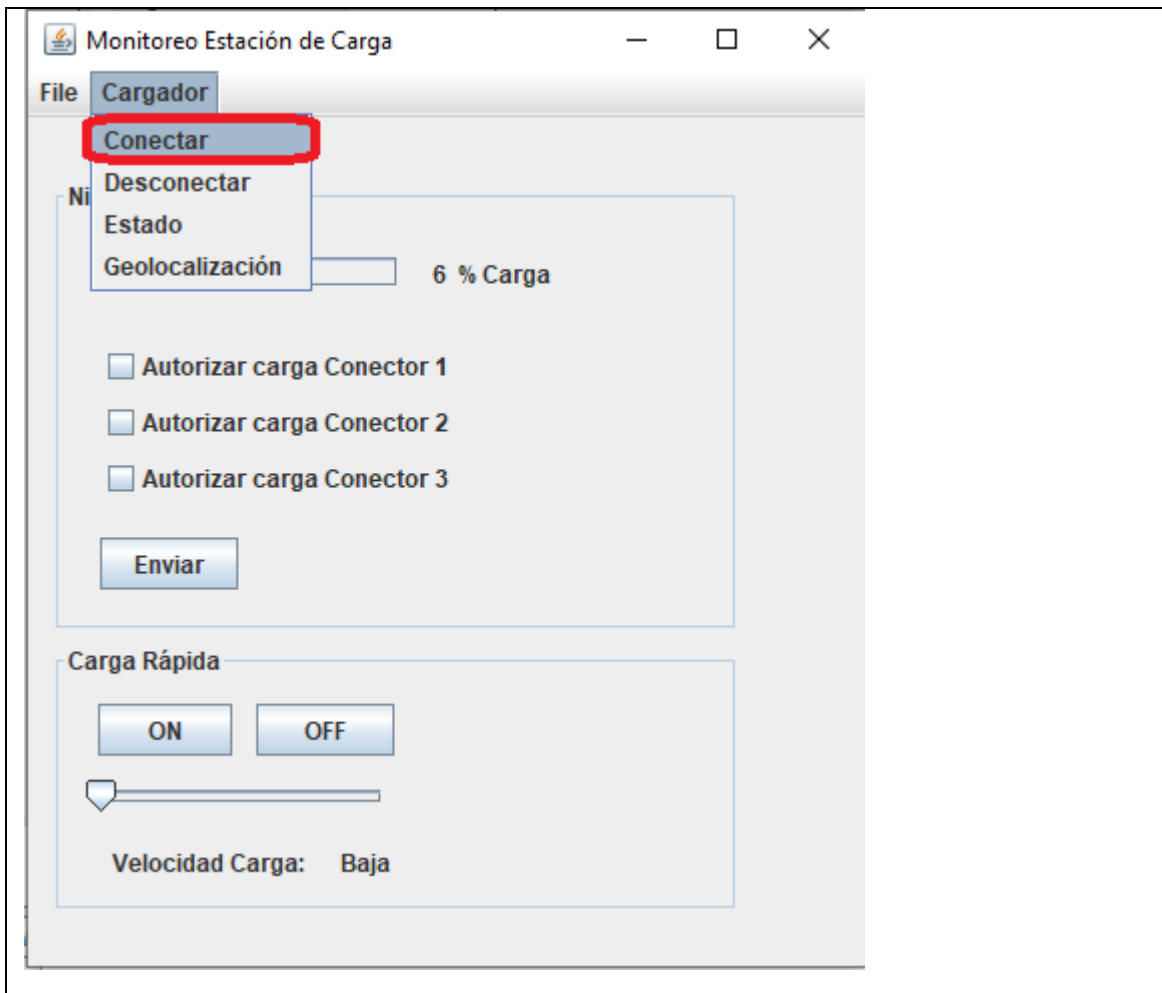
RF-9 Establecer la autorización de carga a un determinado conector: La aplicación debe de permitir activar/desactivar la conmutación de carga de un determinado conector .

1.3.2 Descripción de requisitos de software

En el escenario 4 para la obtención de requisitos de la metodología AUP-UCI se emplean las historias de usuarios (HU) como mecanismos de descripción de los requisitos funcionales. Las historias de usuarios son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan estimar el tiempo de desarrollo e implementarlas sin dificultad (Joskowicz 2008). A continuación, se muestran algunas de las historias de usuario de diferentes requisitos de la propuesta de solución con diferentes niveles de complejidad.

Historia de usuario	
Número: 1	Número del requisito: Gestionar Conexión

Programador: Yonier Lora Ortega	Iteración asignada: 1
Prioridad: Alta	Tiempo estimado: 96 h
Riesgo de desarrollo: falta de experiencia con el uso de la tecnología	Tiempo real: 96 h
Descripción: <ul style="list-style-type: none"> • Objetivo: Gestionar la conexión al dispositivo Estación de Carga, para poder medir los datos. • Precondiciones: Se debe tener conectado el dispositivo al computador. • Acciones a realizar: <p>Seleccionar puerto de conexión: COM7 (puerto seleccionado por el Arduino).</p> <p>Conectar:</p> <p>Al dar clic en el botón conectar se iniciará la conexión con el dispositivo mediante el puerto COM7. En caso de no conectar se mostrará el icono sin conexión y se notificará.</p> <p>Desconectar:</p> <p>Al dar clic en el botón desconectar se detendrá la conexión con el dispositivo.</p>	
Observación:	
Prototipo de interfaz:	



1.3.3 Diagrama de despliegue

El modelo de despliegue se realiza como parte de la implementación para describir la distribución física del sistema. Establece la correspondencia entre la arquitectura lógica, los procesos y los nodos. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. Los nodos poseen relaciones que representan medios de comunicación entre ellos (Pressman 2021). Se reflejan en este artefacto los protocolos de comunicación mediante los cuales se comunican los nodos respectivos.

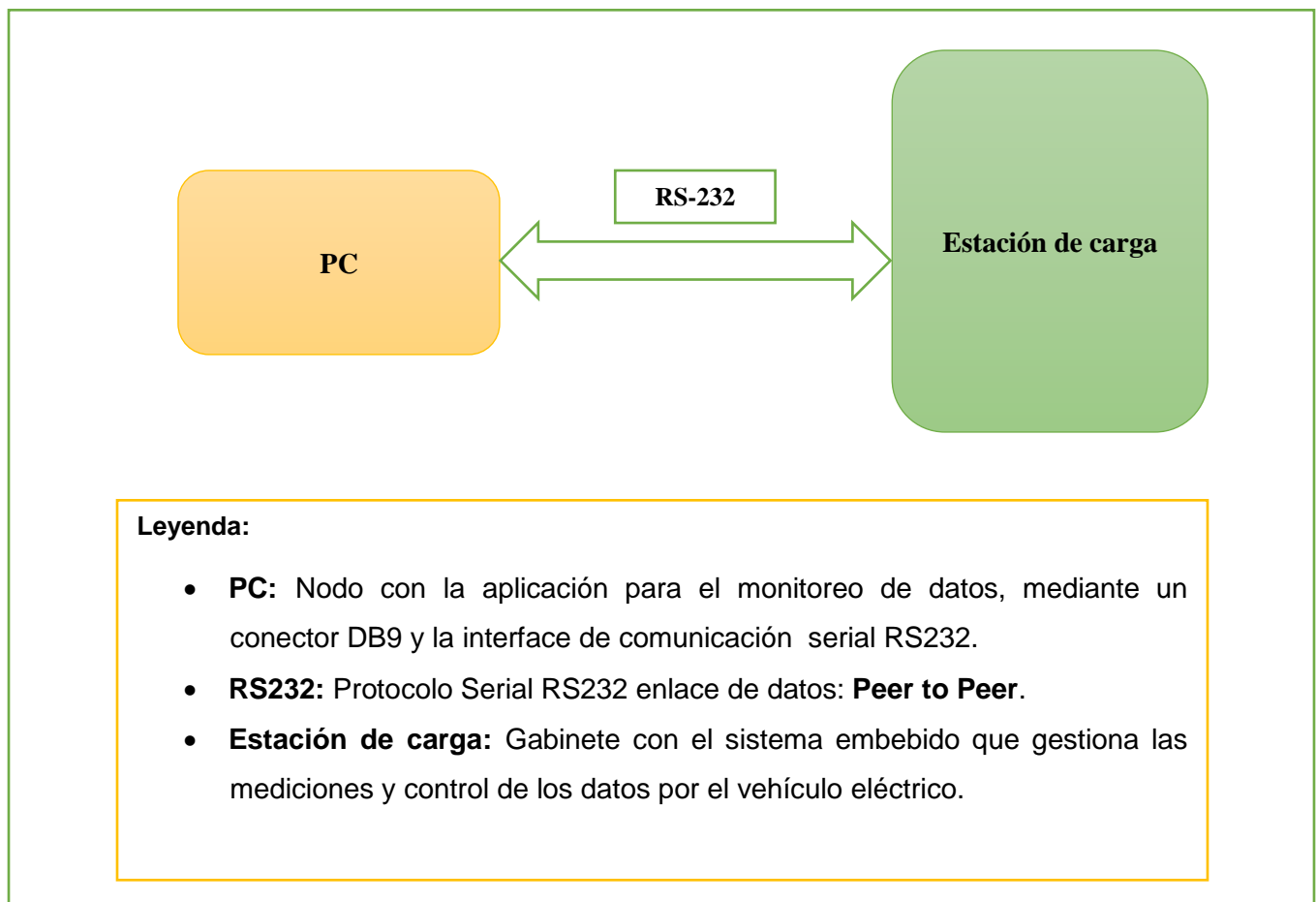


Figura 4 Diagrama de despliegue (Fuente: Elaboración propia).

CONCLUSIONES

Considerando los resultados descritos en este informe, la necesidad y el objetivo planteado por la investigación se arriban a las siguientes conclusiones:

El empleo de las herramientas y tecnologías seleccionadas para la implementación de la solución propició la correspondencia entre los resultados obtenidos y los esperados, lo cual pudo asegurar el nivel de precisión en el análisis y diseño de la aplicación.

El empleo de la metodología AUP-UCI en su escenario 4 permitió que se trazara una buena planificación de todo el proyecto, ahorrando tiempo y recursos.

Se implementó una aplicación para la Medición y Control de datos para el prototipo de Estación de Carga, que mediante el uso de la plataforma Arduino como núcleo de procesamiento de la Estación experimental, demuestran la factibilidad del usos del hardware libre y de bajo coste, siendo viable para un escalado industrial.

RECOMENDACIONES

Con el objetivo de mejorar el funcionamiento y utilidad de la Aplicación para la adquisición de datos:

- Adaptarlo a otras plataformas (Android).
- Integrar las funcionalidades para la gestión de los permisos de los usuarios mediante la identificación de tarjetas RFID.
- Incluir otras interfaces y protocolos de comunicación, tanto en el Sistema Embebido de la Estación de Carga y en la Aplicación.

REFERENCIAS BIBLIOGRÁFICAS

Márquez, M., & Arévalo, J. (2019). Diseño de una estación de carga para vehículos eléctricos en un centro comercial. Revista de Investigación Académica, 19, 1-12. Recuperado de <https://revistainvestigacionacademicauatx.com/index.php/RIAUATX/article/view/268>

S. Mukherjee, "Data Formats and Data Interchange," en Handbook of Data Science Approaches, vol. 1, eds. I. Rudomilov and V. Larin, Springer International Publishing, 2019, pp. 197-229.

Visual Paradigm Product Overview, sin fecha. en línea. [Accedido 8 julio 2023]. Recuperado a partir de: https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html

¿Qué es un requerimiento funcional? - La Oficina de Proyectos de Informática, sin fecha. en línea. [Accedido 8 julio 2023]. Recuperado a partir de: <http://www.pmoinformatica.com/2018/05/que-es-requerimiento-funcional.html>.

PRESSMAN, Roger S, 2021. INGENIERÍA DE SOFTWARE. 9a EDICION. casadellibro. en línea. 27 abril 2021. [Accedido 8 julio 2023]. Recuperado a partir de: <https://www.casadellibro.com/libro-ingenieria-de-software-9-edicion/9781456287726/12328822>.

PÉREZ GUERRA, Mayeleiny y RONDÓN MILANÉS, Raúl Alejandro, 2012. Extensión de la herramienta Visual Paradigm for UML para la administración de requisitos. en línea. 2012. [Accedido 8 julio 2023]. Recuperado a partir de: https://repositorio.uci.cu/jspui/handle/ident/TD_05808_12Accepted: 2016-09-14T19:17:21Z.

JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James, 2006. El proceso unificado de desarrollo de software. Madrid: Addison Wesley. ISBN 978-84-7829-036-9.

CRESPO, Enrique, 2016. DAQ | Aprendiendo Arduino. en línea. 2016. [Accedido 8 julio 3. Recuperado a partir de: <https://aprendiendoarduino.wordpress.com/tag/daq/>

ANEXOS

