



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
XETID, FACULTAD DE TECNOLOGÍAS INTERACTIVAS

MÓDULO PARA MONITOREAR DE FORMA REMOTA ESTACIONES DE CARGA PARA VEHÍCULOS ELÉCTRICOS

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Ailén Beatríz Ponce Rosabal

Tutores: P.A., Ing. Saily Salas Hechavarria, Ms. C

Ing. Osberto Prieto Pérez

La Habana, 2024

Solo llegarán a la cima los que en su empeño tengan Fe

A mi madre, a Dios y a mí.

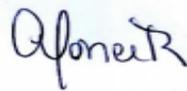
Agradecimientos

Me imaginé tantas veces escribiendo esto, y ahora no se ni como empezar. Primero, por supuesto, agradecer a mi madre, que me ha dejado seguir el camino que quiero apoyándome y dejándome saber siempre que me seguiría al fin del mundo. Luego, a mi novio, que ha sabido aguantar como un campeón mi estrés durante todo este tiempo. También mis suegros y sus hermanos, que se han convertido en mi segunda familia. No puedo dejar de mencionar a mis amigos, que me daban esperanzas, sabiendo en el fondo que la teníamos difícil. Quiero agradecer a todos los que en algún momento me dijeron que yo si podía Lili, Kevin, Rachel, la verdad, escuchar eso, aunque sea un poco, alivia. Quedan muchas historias, conversaciones y vivencias que nos unieron y nos hicieron crecer. No puedo dejar fuera a mis tutores, Osberto y Sailyn, ustedes fueron un gran apoyo, gracias por creer en mi y ayudarme, sin ustedes no habría sido posible. A los profesores que me hicieron amar la carrera, no lo creía, pero de esos todavía existen. Tal vez esté olvidando a alguien, pero de forma general simplemente quiero agradecer a Dios, por llevarme hasta aquí y arreglar siempre mi camino al final del día. A las personas que estuvieron al pendiente de mí, muchas gracias. Para finalizar, siento que hoy merezco aplaudirme, por levantarme siempre, muchas veces sola, pero seguir luchando por mis metas.

Declaración de autoría

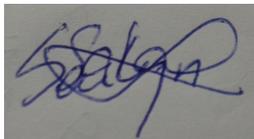
Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los 2 días del mes de diciembre del año 2024 .



Ailén Beatriz Ponce Rosabal

Autor



P.A., Ing. Sailyn Salas Hechavarria, Ms. C

Tutora



Ing. Osberto Prieto Pérez

Tutor

El agotamiento de las reservas de petróleo y la creciente demanda energética plantean una urgente necesidad de alternativas sostenibles en el transporte. La adopción de vehículos eléctricos (VE) se presenta como una solución viable, pero enfrenta desafíos como la limitada autonomía de las baterías y la falta de estaciones de carga adecuadas. En respuesta a este problema, la Empresa de Tecnologías de la Información para la Defensa (XETID) propone la implementación de estaciones de carga. Sin embargo, estas carecen de un sistema de monitoreo remoto que se adapte al modelo cubano y permita a los usuarios conocer el estado de carga sin necesidad de estar presentes físicamente. Con el propósito de facilitar este proceso el presente trabajo propone un módulo que proporciona información en tiempo real sobre estaciones de carga mejorando la gestión y la eficiencia del servicio. Para el desarrollo del mismo se utilizó la metodología XP con el fin de guiar el proceso. Este fue diseñado e implementado utilizando Python y el framework Flask. Además, se llevaron a cabo diversas pruebas, incluyendo integración, usabilidad y aceptación, para verificar su funcionalidad. Los resultados demostraron que el sistema cumple con los requisitos establecidos, satisfaciendo así las necesidades del cliente y contribuyendo a la promoción de tecnologías limpias en el sector del transporte. Esta solución no solo mejora la experiencia del usuario, sino que también apoya la transición hacia un modelo de movilidad más sostenible.

Palabras clave: Estaciones de carga, Monitoreo remoto, Sistemas de gestión, Tecxol, Vehículos eléctricos.

Introducción	1
1 Fundamentación teórica	3
1.1 Estaciones de carga eléctrica	3
1.1.1 Electromovilidad en América Latina	4
1.1.2 Clasificaciones	5
1.2 Recolección de datos	7
1.3 Sistemas homólogos	9
1.4 Herramientas y tecnologías	11
1.5 Metodología de desarrollo	14
Conclusiones parciales	16
2 Análisis y diseño del software	17
2.1 Propuesta de solución	17
2.2 Fase I: Planificación	19
2.2.1 Historias de usuario	19
2.2.2 Requisitos no funcionales	21
2.2.3 Estimación de esfuerzos por cada HU	22
2.2.4 Plan de iteraciones	23
2.3 Fase II: Diseño del sistema	24
2.3.1 Arquitectura del software	25
2.3.2 Patrones de diseño	26
2.3.3 Tarjetas Clase – Responsabilidad – Colaborador (CRC)	27
Conclusiones parciales	28
3 Implementación y pruebas del sistema	29
3.1 Fase III: Desarrollo	29
3.1.1 Tareas de ingeniería	29
3.2 Fase IV: Pruebas	30
3.2.1 Pruebas unitarias	31

3.2.2 Pruebas de Integración	34
3.2.3 Pruebas de Sistema	34
3.2.4 Pruebas de Aceptación	36
Conclusiones parciales	36
Conclusiones	38
Recomendaciones	39
Referencias bibliográficas	40
Apéndices	43
A Historias de Usuario	44
B Tarjetas CRC	58
C Tareas de Ingeniería	59
D Casos de Prueba	62

Índice de figuras

2.1	Propuesta de solución	18
2.2	Diseño de arquitectura	25
3.1	Resultados de las pruebas unitarias	33
3.2	Respuesta a las pruebas unitarias	34
3.3	Resultados de la lista de chequeo	35

Índice de tablas

1.1	Tipo de carga según la potencia	7
1.2	Sistemas homólogos	10
1.3	COMPARACIÓN DE METODOLOGÍAS	15
2.1	Historia de usuario # 1	20
2.2	Estimación de esfuerzo por historia de usuario	23
2.3	Plan de duración de las iteraciones	24
2.4	Tarjeta CRC # 1	28
3.1	Tarea de ingeniería # 1	30
3.2	Prueba de aceptación # 1	36
A.1	Historia de usuario # 2	44
A.2	Historia de usuario # 3	45
A.3	Historia de usuario # 4	45
A.4	Historia de usuario # 5	46
A.5	Historia de usuario # 6	47
A.6	Historia de usuario # 7	47
A.7	Historia de usuario # 8	48
A.8	Historia de usuario # 9	49
A.9	Historia de usuario # 10	49
A.10	Historia de usuario # 11	50
A.11	Historia de usuario # 12	51
A.12	Historia de usuario # 13	52
A.13	Historia de usuario # 14	53
A.14	Historia de usuario # 15	54
A.15	Historia de usuario # 16	55
A.16	Historia de usuario # 17	56
B.1	Tarjeta CRC # 2	58
B.2	Tarjeta CRC # 3	58

C.1	Tarea de ingeniería # 2	59
C.2	Tarea de ingeniería # 3	59
C.3	Tarea de ingeniería # 4	60
C.4	Tarea de ingeniería # 5	60
C.5	Tarea de ingeniería # 6	60
C.6	Tarea de ingeniería # 7	61
D.1	Prueba de aceptación # 2	62
D.2	Prueba de aceptación # 3	62
D.3	Prueba de aceptación # 4	63
D.4	Prueba de aceptación # 5	63
D.5	Prueba de aceptación # 6	64
D.6	Prueba de aceptación # 7	64
D.7	Prueba de aceptación # 8	65
D.8	Prueba de aceptación # 9	65
D.9	Prueba de aceptación # 10	66
D.10	Prueba de aceptación # 11	66
D.11	Prueba de aceptación # 12	67
D.12	Prueba de aceptación # 13	68
D.13	Prueba de aceptación # 14	68
D.14	Prueba de aceptación # 15	69
D.15	Prueba de aceptación # 16	69
D.16	Prueba de aceptación # 17	70

En la actualidad, el agotamiento de las reservas de petróleo a nivel mundial se ha convertido en un tema crítico. Sumado a la creciente demanda energética en países en desarrollo, impulsa la urgente necesidad de buscar alternativas sostenibles que sustituyan los bienes y servicios basados en derivados del petróleo. Uno de los sectores que se enfrenta a este desafío es el transporte, reconocido como uno de los principales motores del desarrollo económico y social. En respuesta a los problemas asociados con el uso de combustibles fósiles, se ha iniciado un cambio hacia la adopción de vehículos eléctricos (VE). Esta transición no solo representa un esfuerzo por reducir la dependencia de recursos no renovables, sino que también abre la puerta a un futuro más sostenible y menos contaminante. Esta alternativa trae consigo también problemas a pesar de su capacidad para disminuir el uso de combustible. Uno de estos problemas es la recarga de estos vehículos y el hecho de que puede recorrer menores distancias al depender de su batería. Frente a estos problemas la Empresa de Tecnologías de la Información para la Defensa (XETID) se ha propuesto llevar a cabo la implementación de estaciones de carga para vehículos eléctricos. Las cuales se pueden considerar la solución definitiva para lograr expandir el uso de vehículos eléctricos y por tanto disminuir la demanda de combustible fósil. Como parte de esta evolución, el desarrollo de infraestructuras de recarga para vehículos eléctricos ha cobrado una importancia significativa. Las estaciones de carga no solo facilitan la movilidad de los usuarios de vehículos eléctricos, sino que también son fundamentales para la integración de estos vehículos en el sistema energético. De esta manera, se busca no solo fomentar el uso de tecnologías limpias, sino también contribuir a un mejor aprovechamiento de los recursos energéticos disponibles. Las estaciones de carga actuales presentan un inconveniente para los usuarios: la falta de un sistema de monitoreo remoto. Si bien las estaciones cuentan con una pantalla integrada, el usuario debe permanecer presente y atento a su funcionamiento, limitando la comodidad y la eficiencia del proceso de carga. Este problema se traduce en:

- Falta de información en tiempo real: El usuario no puede conocer el avance de la carga, el voltaje de salida, la potencia utilizada y otros datos relevantes.
- Pérdida de tiempo: El usuario debe estar físicamente presente junto a la estación para obtener información sobre el proceso de carga, lo que implica un tiempo de espera innecesario.
- Dificultad para la gestión: Sin monitoreo remoto, es complejo identificar posibles errores o anomalías en el funcionamiento de la estación.

En consecuencia, el desarrollo de esta herramienta no solo solucionaría un problema para los usuarios, sino que también aumentaría el atractivo del producto, brindándoles una experiencia más cómoda y eficiente.

A partir de la situación planteada se define como **problema científico**: ¿Cómo desarrollar un módulo que proporcione de forma remota información actualizada sobre una estación de carga para vehículos eléctricos al usuario que la está utilizando? En esta investigación se ha utilizado como **objeto de estudio** las estaciones de carga eléctrica enmarcado en el **campo de acción** de los sistemas de gestión para estaciones de carga para vehículos eléctricos. Para darle solución al problema planteado se propone como objetivo general: desarrollar un módulo para monitorear de forma remota estaciones de carga para vehículos eléctricos permita proporcionar información en tiempo real a los usuarios sobre las variables que mide y controla la estación de carga. Para dar cumplimiento al objetivo general se trazaron los siguientes **objetivos específicos**:

- Investigar las tendencias tecnológicas actuales y los estándares más adoptados a nivel mundial en las estaciones de carga para vehículos eléctricos.
- Identificar los requisitos funcionales y no funcionales que garanticen el cumplimiento de las necesidades de software.
- Analizar la metodología de desarrollo seleccionada para garantizar un ambiente de desarrollo de software adecuado.
- Diseñar el módulo para monitorear de forma remota estaciones de carga para vehículos eléctricos.
- Implementar las funcionalidades del módulo para monitorear de forma remota estaciones de carga para vehículos eléctricos.
- Realizar pruebas del software al módulo para monitorear de forma remota estaciones de carga para vehículos eléctricos.

Fundamentación teórica

En el capítulo se abordará una base teórica sólida que respalda y justifica la investigación. Para ello se realizará un estudio del estado del arte actual de las estaciones de carga eléctrica y sus sistemas de gestión. Se analizará las diferentes herramientas, metodologías y tecnologías para la elección de las más adecuadas a este tipo de proyectos.

1.1. Estaciones de carga eléctrica

Las estaciones de carga para vehículos eléctricos (VE) son instalaciones diseñadas específicamente para suministrar energía eléctrica a vehículos que funcionan con baterías. Conceptualmente, las estaciones de carga actúan como los equivalentes de las gasolineras para los automóviles convencionales, proporcionando un punto de acceso para la carga de energía. Estas están equipadas con distintos tipos de conectores y sistemas de carga que se adaptan a las especificaciones de los vehículos eléctricos, permitiendo diferentes velocidades de carga y tipos de corriente. Su función principal es permitir la recarga de las baterías de estos vehículos, garantizando que los usuarios puedan mantener su autonomía y utilizar sus vehículos de manera eficiente (Panduro Del Aguila y Perez Espinoza, 2023).

Las estaciones son fundamentales para facilitar la adopción masiva de vehículos eléctricos. Se consideran un componente clave en la transición hacia un sistema de transporte más sostenible. Contribuyen a la estabilidad de la red eléctrica y promueven el uso de energías renovables. Estas infraestructuras mejoran la movilidad de los usuarios. También, fomentan la inversión en tecnologías limpias y ayudan a reducir las emisiones de gases de efecto invernadero siendo respetuosos con el medio ambiente. Son un elemento crucial en el ecosistema del transporte eléctrico, proporcionando la infraestructura necesaria para soportar el crecimiento del mercado de vehículos eléctricos.

1.1.1. Electromovilidad en América Latina

A raíz de una investigación exhaustiva, se encontraron varios estudios de investigación sobre este tema, de los cuales se profundizó en América Latina. Ejemplo de ello son los que se resumen a continuación:

1. Un trabajo de investigación sobre la implementación de estaciones de recarga para vehículos eléctricos en Perú, destacó su importancia en la reducción de la contaminación y la mejora en la calidad del aire (Pachas Pachas y Pacahuala Virgilio, 2023). Los beneficios del transporte eléctrico que incluyen su ecoeficiencia y el ahorro de tiempo en comparación con la recarga en casa. Destacó como puntos clave que:

- Las estaciones de recarga buscan abordar la adopción masiva de los vehículos eléctricos mediante estas infraestructuras.
- Se pretende reducir el tiempo de recarga a 20 minutos en comparación con las cuatro a ocho horas que tomaría en un hogar.
- Se estima que el mercado objetivo incluye tanto a propietarios de vehículos eléctricos como a empresas que ofrecen ese servicio.
- El proyecto propone ubicaciones estratégicas para las estaciones, como restaurantes y centros comerciales, para maximizar la comodidad del usuario.
- La necesidad de transporte sostenible y la creciente preocupación por el medio ambiente impulsan la demanda de vehículos eléctricos.
- Se prevé que contar con estaciones de recarga facilitará el uso cotidiano de vehículos eléctricos, mitigando la limitación de autonomía actual.
- La implementación de un sistema de pago mediante una aplicación móvil para facilitar la recarga y el acceso a los usuarios.

2. Otro estudio realizado sobre la infraestructura de Carga para Vehículos Eléctricos en América Latina y el Caribe muestra como resultado los desafíos y oportunidades en la infraestructura para la carga de vehículos eléctricos en América Latina y el Caribe (De La; Cedeño; G. E. M. Ibarra; J. J. V. Ibarra et al., 2024). Destaca cómo la transición hacia la movilidad eléctrica es fundamental para reducir las emisiones de carbono, mejorar la calidad de vida urbana y promover el desarrollo económico. Sin embargo, la región enfrenta obstáculos significativos, como la falta de infraestructura adecuada, altos costos de implementación, marcos regulatorios inadecuados, y una baja conciencia pública sobre los beneficios de este tipo de movilidad. Se presentan casos de éxito en países como Chile y Colombia que ofrecen importantes lecciones, y se concluye con recomendaciones estratégicas para facilitar la implementación de infraestructuras que impulsen un futuro sostenible. Se destacan como puntos clave que:

- La movilidad eléctrica es esencial para disminuir las emisiones de CO₂ y mejorar la calidad del aire en las ciudades.
- La falta de infraestructura adecuada limita la adopción de vehículos eléctricos en América Latina y el Caribe.

- Costos elevados de instalación y mantenimiento son obstáculos importantes en la expansión de la infraestructura de carga.
- Las políticas públicas deben ser coherentes y colaborativas entre el sector público y privado para promover la movilidad eléctrica.
- Países como Chile y Colombia están liderando el desarrollo de infraestructuras de carga, sirviendo como modelos para otros.
- La integración de la infraestructura de carga con energías renovables maximiza los beneficios ambientales y económicos.
- La educación y sensibilización del público son cruciales para fomentar la aceptación y el uso de vehículos eléctricos.

1.1.2. Clasificaciones

Otra información que es importante conocer sobre las estaciones de carga es las distintas clasificaciones que pueden tener de acuerdo a distintos aspectos. De acuerdo al tipo de corriente que utilizan las estaciones de carga, se clasifican de dos formas fundamentales: corriente alterna (AC) y corriente continua (DC). A continuación, se detalla el tipo de corriente y las razones detrás de su uso:

Estaciones de Carga de Corriente Alterna (AC)

Tipo de Carga: Normalmente se utilizan para cargas de baja a media potencia.

Uso: Estas estaciones son comunes en entornos residenciales y comerciales, donde los vehículos pueden estar estacionados durante períodos prolongados.

Razón: La corriente alterna es más fácil de manejar y distribuir en la red eléctrica existente. Además, muchos vehículos eléctricos están equipados con cargadores que pueden aceptar corriente alterna, lo que permite una carga más lenta pero eficiente.

Estaciones de Carga de Corriente Continua (DC)

Tipo de Carga: Se utilizan para cargas rápidas y de alta potencia.

Uso: Estas estaciones son ideales para ubicaciones estratégicas, como estaciones de servicio o áreas de descanso en carreteras, donde los conductores necesitan cargar rápidamente sus vehículos.

Razón: La corriente continua permite una carga más rápida porque se puede suministrar directamente a la batería del vehículo, evitando la conversión de AC a DC que ocurre en los cargadores de corriente alterna. Esto es especialmente útil para vehículos eléctricos que requieren tiempos de carga reducidos.

La elección entre corriente alterna y corriente continua en las estaciones de carga depende de factores como la velocidad de carga requerida, la infraestructura eléctrica disponible y las especificaciones del vehículo eléctrico. Las estaciones de carga AC son más comunes para uso diario y carga prolongada, mientras que las estaciones de carga DC son preferidas para cargas rápidas y en situaciones donde el tiempo es un factor crítico. Por otro lado, las estaciones pueden estar alimentadas por diferentes fuentes de energía, lo que las convierte en una parte integral de un sistema energético más amplio y sostenible (Acharige; Haque; Arif; Hosseinzadeh; Hasan y Oo, 2023). Algunos de los diferentes tipos de estaciones de carga eléctrica en función

de su fuente de energía y tecnología son:

- Estaciones de carga conectadas a la red eléctrica o carga convencional: estas estaciones están conectadas a la red eléctrica existente y utilizan energía de fuentes convencionales (como generación de electricidad a partir de combustibles fósiles).
- Estaciones de carga solar: estas estaciones están equipadas con paneles solares que convierten la energía solar en electricidad. La energía generada se utiliza para cargar vehículos eléctricos, lo que hace que la operación sea más sostenible y reduce la dependencia de la red eléctrica. Pueden ser instaladas en áreas públicas o privadas y contribuyen a la generación de energía limpia.
- Estaciones de carga eólica: estas estaciones aprovechan la energía del viento mediante aerogeneradores para generar electricidad. Estas pueden funcionar independientemente de la red.
- Estaciones de carga basadas en baterías: estas estaciones almacenan energía en baterías previamente cargadas, que pueden ser alimentadas por diversas fuentes (incluyendo renovables) y luego utilizadas para cargar vehículos eléctricos en momentos de alta demanda o cuando la producción de energía renovable es baja.
- Estaciones de carga inteligentes: integran tecnología de gestión de energía y pueden conectarse a varias fuentes, optimizando la carga según la disponibilidad de energía y las necesidades de los usuarios. Estas estaciones pueden, por ejemplo, priorizar el uso de energía renovable cuando está disponible.
- Estaciones de carga híbridas: combinan diferentes fuentes de energía, como solar y eólica, para generar electricidad para la carga de vehículos. Estas estaciones aprovechan las ventajas de múltiples tecnologías, lo que les permite ofrecer un suministro más confiable y reducir el impacto ambiental.

Para el desarrollo de la propuesta se utilizará una estación de carga híbrida pues va a contar con paneles solares y baterías que se recargarán con esta energía para brindarla cuando no esté disponible el recurso solar.

Otras clasificaciones

Otros criterios de clasificación sería su ubicación (públicas, privadas, en carretera o en áreas urbanas) y su capacidad de carga (carga lenta, carga rápida, y carga ultrarrápida). Cada tipo de carga tiene sus propias características y es adecuada para diferentes necesidades. Por ejemplo, las estaciones de carga lenta son ideales para cargas nocturnas en hogares, mientras que las estaciones de carga rápida son más adecuadas para recargas durante viajes largos (Brenna; Foadelli; Leone y Longo, 2020). En la tabla 1.1 se muestran los distintos niveles de potencia de carga y de acuerdo a estos el tipo de carga y su uso.

Tabla 1.1. Tipo de carga según la potencia

Nivel	Potencia	Tiempo estimado	Tipo de carga	Uso típico
Nivel 1	2 Kw	4-11 horas	Lenta	Apropiadas para la recarga en entornos residenciales o en lugares donde los vehículos permanecerán estacionados por periodos prolongados, como garajes o estacionamientos. Suelen utilizarse en hogares y permiten una carga gradual, lo que es ideal para el uso diario.
Nivel 2	20 Kw	1-4 horas	Semi-rápida	Generalmente se encuentran en espacios públicos, como centros comerciales o plazas de aparcamiento, y permiten una recarga más rápida en comparación con la carga lenta, siendo adecuadas para usuarios que necesiten recargar su vehículo durante cortos períodos de tiempo.
Nivel 3	100 Kw	Menos 30 minutos	Rápida y ultrarrápida	Estas estaciones están diseñadas para un reabastecimiento rápido de energía y son ideales para viajes largos, donde los usuarios pueden necesitar cargar su vehículo en menos de una hora. Utilizan corriente directa para acelerar el proceso de carga.

La estación a utilizar para el funcionamiento del módulo sería de Nivel 2, de carga semi-rápida, con el uso de corriente alterna para suministrar los vehículos.

1.2. Recolección de datos

Para diseñar un sistema de monitoreo efectivo para estaciones de carga, es crucial comprender las variables clave que determinan su funcionamiento. Entre las diversas variables que se miden, las de mayor interés para este proyecto son:

- La **potencia** en una estación de carga eléctrica se refiere a la cantidad de energía eléctrica que se suministra a un vehículo eléctrico (VE) en un intervalo de tiempo determinado. Se mide en vatios (W) o kilovatios (kW), siendo 1 kW equivalente a 1,000 W (Personal, 2024). La potencia de carga es un factor crucial porque determina el tiempo que toma cargar las baterías del vehículo. Por ejemplo:
Carga de baja potencia (como un enchufe doméstico) suele ofrecer entre 1,5 kW y 3,7 kW, lo que puede tardar varias horas en cargar un vehículo.
Carga de potencia media (puntos de carga públicos) puede ofrecer entre 7 kW y 22 kW, lo que reduce significativamente el tiempo de carga.
Carga rápida (estaciones de carga rápida) puede ofrecer 50 kW o más, permitiendo recargar una gran parte de la batería en menos de una hora.
- El **voltaje de salida** en una estación de carga eléctrica se refiere a la diferencia de potencial eléctrico que se suministra al vehículo eléctrico (VE) durante el proceso de carga. Se mide en voltios (V) y es un factor crucial para el funcionamiento adecuado de la batería del vehículo. El voltaje de salida puede variar dependiendo del tipo de estación de carga y del vehículo (Electrical and Engineers, 2023). Aquí hay algunos puntos clave sobre el voltaje:
Carga de bajo voltaje: Las estaciones de carga domésticas suelen proporcionar un voltaje de entre 110 y 240 V, dependiendo de la instalación eléctrica.
Carga de media y alta voltaje: Las estaciones de carga pública y rápida pueden ofrecer voltajes que oscilan entre 400 V y 800 V (o más), lo que permite una carga más rápida.
Compatibilidad: Es importante que el voltaje de la estación de carga sea compatible con el voltaje de la batería del vehículo. Cada modelo de vehículo eléctrico tiene un rango específico de voltaje para su sistema de carga.
Efecto en la Carga: Un voltaje de salida más alto puede permitir una carga más rápida, siempre y cuando la infraestructura del vehículo y la estación sean capaces de manejar ese voltaje.
- El **consumo de corriente** en una estación de carga eléctrica se refiere a la cantidad de corriente que la estación extrae para suministrarla al vehículo eléctrico (VE) durante el proceso de carga. Este consumo varía según varios factores, incluidos el tipo de cargador, el voltaje de salida, la capacidad de la batería del vehículo y la tarifa de carga configurada. Las estaciones modernas a menudo tienen sistemas de medición que permiten monitorear el consumo de corriente en tiempo real, lo que es útil tanto para la gestión de la energía como para el usuario. La eficiencia de la carga también puede afectar el consumo de corriente. Cargadores de alta calidad generalmente tienen una eficiencia mayor, lo que significa que menos energía se pierde en forma de calor.
- A partir de las variables ya explicadas se obtienen otras métricas que son de interés al usuario. Estas variables son: el **recurso solicitado** que se refiere a la cantidad de energía (kWh) requerida por el cliente, siendo el **recurso consumido** la cantidad de energía (kWh) que ha consumido el cliente a medida que ha ido pasando el tiempo de carga, quedando finalmente el **tiempo restante**, que sería el tiempo estimado que queda para completar el consumo del recurso solicitado.

Integración Tecnológica para obtener las variables

Para obtener todas estas variables la estación de carga está equipada con un dispositivo conocido como Analizador de Red Trifásico SENECA modelo S203TA-D, el cual se conecta a una Raspberry Pi 4. Este dispositivo se utiliza para censar datos, como voltaje, corriente y potencia. Las variables recolectadas se registran utilizando un protocolo de comunicación denominado Modbus. Este protocolo es un sistema de comunicación abierto que permite la transmisión de información a través de redes en serie entre dispositivos electrónicos. Su funcionamiento se basa en una arquitectura maestro-esclavo, donde un dispositivo maestro envía solicitudes a dispositivos esclavos y espera respuestas. En este contexto, el dispositivo que solicita la información se designa como maestro Modbus, mientras que los dispositivos que proporcionan dicha información son considerados esclavos Modbus.

De forma general, el sistema para recolectar los datos sería conectando la Raspberry Pi a través del protocolo Modbus con el dispositivo SENECA. Estos dispositivos se encontrarán dentro de la propia estación de carga, así como un proveedor de wifi que más adelante se utilizará. Primero, se utiliza un componente previamente implementado que establece la comunicación entre la Raspberry Pi y el dispositivo SENECA, enviando los datos recolectados al archivo principal del programa. En este archivo se encuentran todas las variables de interés al usuario, las cuales son declaradas variables globales para facilitar su utilización en cualquier parte del sistema, por lo tanto al llevar a cabo las funciones que requieren el uso de estas variables, se indica su alcance global, lo que permite que adopten el mismo valor. De esta manera, se obtendría el valor de las variables que está tomando en ese momento la estación de carga. Una vez se tienen disponibles estos valores y el usuario ya ha realizado su pago, al encontrarse haciendo uso de la estación podrá conectarse al sistema de gestión. Esta conexión se realizará al obtener acceso a la wifi proveída por la propia estación de carga y entrar al sitio web. Una vez dentro tendrá acceso a todos los datos recolectados que son de su interés.

1.3. Sistemas homólogos

De acuerdo con la Real Academia Española (RAE) un **sistema** es un conjunto de cosas que relacionadas entre sí ordenadamente contribuyen a determinado objeto (Española, 2024), siendo la **gestión** la acción y efecto de administrar (ibíd.). Por lo tanto, se puede entender como **sistema de gestión** aquel sistema que relacionado a un tema o lugar específico contribuye en su administración. Al introducirlo en el ámbito de las estaciones de carga se entiende como aquellos sistemas que asisten a los usuarios o administradores en la gestión de las mismas. Un sistema de gestión para estaciones de carga para vehículos eléctricos es un sistema centralizado por cada una de las estaciones de carga que contiene la información sobre las estaciones de carga a la que pertenezcan. Como es de esperarse en un tema tan innovador como este existen muchas plataformas orientadas a la gestión de sus estaciones de recarga o incluso la localización de las mismas. En el ámbito internacional existen muchos ejemplos de este tipo de servicios, algunos de ellos son los que se muestran en la tabla 1.2 :

Tabla 1.2. Sistemas homólogos

Nombre	Plataforma	Funcionalidades
miSolum	Aplicación móvil	Permite configurar, monitorizar y gestionar a distancia las estaciones de carga. Consulta el estado de los puntos de recarga y define las zonas de recarga, el número de vehículos por estación, el tipo de conectores o el grupo de usuarios que pueden acceder a cada estación. Visualiza las sesiones de recarga, estados, energía consumida, ahorro de CO2, entre otros, en tiempo real.
Ambika	Aplicación para Android	Digitaliza y automatiza el despliegue y la gestión de estaciones de recarga. Se centra en optimizar el uso de flotas de vehículos eléctricos, facilitando la monitorización del estado de los vehículos, la planificación de rutas, la gestión de la carga y la programación de horarios.
EVcharge	Aplicación móvil multi-plataforma	Gestiona y opera puntos de recarga inteligentes para escalar un negocio. Gestiona la operativa de todos los cargadores: las tarifas, usuarios, organiza las tareas de mantenimiento, analiza consumos y mucho más. Dependiendo de la ubicación y el proveedor de carga, algunos puntos de carga pueden ser gratuitos, mientras que otros requieren el pago de una tarifa.
EVectrum	Plataforma de software como servicio	Es una solución B2B para que las empresas puedan gestionar su electromovilidad. Incluye gestionar sus cargadores propios y los de sus empleados. Facilita que su flota de vehículos eléctricos se pueda mover, tanto cargando en casa, como en la vía pública.
EVBox everon	Plataforma web con apk disponible	Permite gestionar múltiples estaciones de carga desde una sola interfaz, facilitando la supervisión y operación eficiente de toda la red. Ofrece visibilidad en tiempo real del estado de las estaciones de carga.

En el ámbito nacional, no se encontró plataforma o aplicación dedicada a la gestión de estaciones de carga para vehículos eléctricos (VE). Esta situación es comprensible, considerando que la tecnología de VE aún está en desarrollo y adopción temprana en Cuba. Sin embargo, la ausencia de una plataforma de gestión especializada representa una oportunidad única para desarrollar una solución propia que atienda las necesidades específicas del mercado local. Se requiere un sistema de gestión de estaciones de recarga que sea de bajo costo o incluso gratuito y que se adapte a las particularidades del mercado cubano. La creación de esta plataforma no solo cubriría una necesidad latente, sino que también impulsaría la adopción de la tecnología de VE en el país.

Tras la investigación de las soluciones homólogas existentes, se decidió tomar como modelo la funcionalidad de visualización en tiempo real de información de las estaciones de carga. Esta característica, de fácil implementación y utilidad general, será la base de la primera versión del proyecto. Sin embargo, la mayoría de las funcionalidades de las soluciones estudiadas no son aplicables al proyecto actual. Estas plataformas suelen estar diseñadas para gestionar flotas de vehículos y son utilizadas por el mismo proveedor, lo cual no se ajusta a la necesidad actual de la plataforma.

1.4. Herramientas y tecnologías

Con el creciente desarrollo de las tecnologías y herramientas para la creación de proyectos informáticos, existe todo un mundo para escoger la mejor opción o la que más se adapte a lo que se quiere desarrollar. El desarrollo de un módulo informático efectivo requiere la selección de herramientas y tecnologías que se complementen para lograr una experiencia de usuario fluida, mantenimiento sencillo y escalabilidad. Este estudio analizará diversas herramientas disponibles, considerando sus características y ventajas, para finalmente justificar la elección de un conjunto específico de tecnologías para el desarrollo de un módulo informático. A continuación, se muestra un conjunto de algunas de las herramientas y tecnologías que se tuvieron en cuenta a utilizar en la investigación, antes de dar paso a la elección:

- **Entornos de Desarrollo Integrado (IDE)**

Eclipse: Popular entre desarrolladores de Java, ofrece potentes herramientas, pero puede resultar pesado pues requiere muchos recursos del ordenador para funcionar.

PyCharm: Excelente para desarrollo en Python, con características avanzadas, aunque puede ser caro en su versión profesional.

Visual Studio: Ideal para aplicaciones de Windows y .NET, pero su uso se limita generalmente a este ecosistema.

Visual Studio Code: Ligero y versátil, soporta múltiples lenguajes y tiene una gran cantidad de extensiones.

- **Lenguajes de Marcado y Estilo**

HTML: Lenguaje estándar para crear páginas web.

CSS: Utilizado para el estilo y diseño de páginas web.

SASS/LESS: Preprocesadores de CSS que añaden características avanzadas, pero introducen complejidad adicional.

- **Lenguajes de Programación**

JavaScript: Fundamental para la interactividad del lado del cliente, con amplias librerías y frameworks.

TypeScript: Superset de JavaScript que añade tipado estático, ideal para aplicaciones a gran escala.

Python: Conocido por su simplicidad y versatilidad, ampliamente usado en diversos campos, incluido el desarrollo web.

Java: Popular en la creación de aplicaciones empresariales, aunque su configuración puede ser más compleja.

- **Marcos de trabajo (framework) para Desarrollo Web**

Django: Framework robusto para Python, ideal para aplicaciones grandes, pero puede ser más complejo de aprender.

Flask: Un micro-framework ligero para Python, perfecto para aplicaciones simples y rápidas.

- **Marcos de trabajo (framework) de Diseño Frontend**

Bootstrap: Framework CSS popular para un diseño responsivo y componente móvil.

Foundation: Alternativa a Bootstrap, menos conocida pero muy potente.

- **Sistemas de Gestión de Bases de Datos (SGBD)**

MySQL: Muy utilizado, ideal para aplicaciones web y con un rendimiento sólido.

PostgreSQL: Avanzado y ofrece características complejas, pero puede ser más complicado de administrar.

SQLite: Ligero y fácil de usar, ideal para aplicaciones pequeñas o pruebas.

MongoDB: Base de datos NoSQL que es ideal para datos no estructurados, pero puede no ser necesaria en todos los casos.

Después de evaluar las diversas herramientas y tecnologías disponibles, se ha decidido optar por las siguientes:

- **Visual Studio Code (VS Code)**

Tipo: Entorno de Desarrollo Integrado (IDE)

Versión: 1.79.2

Justificación: VS Code es ampliamente utilizado por su versatilidad, facilidad de uso y capacidad para adaptarse a diferentes flujos de trabajo. Su amplia gama de extensiones permite optimizar la productividad, lo que es fundamental en el desarrollo de un módulo informático. Por lo que sería ideal para escribir y editar el código fuente del módulo.

- **HTML (HyperText Markup Language)**

Tipo: Lenguaje de marcado

Versión: HTML5

Justificación: HTML es esencial para definir la estructura de las páginas web. Su simplicidad y ver-

satilidad lo hacen indispensable para cualquier tipo de desarrollo web.

- **CSS (Cascading Style Sheets)**

Tipo: Hojas de estilo

Versión: CSS3

Justificación: CSS es fundamental para crear interfaces atractivas y responsivas, mejorando la experiencia del usuario en el módulo informático. Será de gran utilidad para darle estilo y presentación a las páginas.

- **JavaScript**

Tipo: Lenguaje de programación

Versión: ES2022

Justificación: JavaScript es indispensable para agregar funcionalidades dinámicas y mejorar la interacción del usuario con la interfaz, lo que es clave en un entorno web moderno. En este proyecto servirá para habilitar funciones dinámicas, como la actualización en tiempo real de datos.

- **Bootstrap**

Tipo: Framework CSS

Versión: 4.0.0-beta

Justificación: Bootstrap ayuda a mantener un diseño consistente y atractivo, ya que proporciona un conjunto de estilos y componentes que se adaptan a diferentes dispositivos. Proporciona componentes predefinidos como botones y formularios, así como estilos CSS para una apariencia consistente.

- **Python**

Tipo: Lenguaje de programación

Versión: 3.12.2

Justificación: Python es un lenguaje poderoso y versátil, ideal para desarrollo web y otras aplicaciones. Su simplicidad y eficiencia lo convierten en una excelente opción para el desarrollo del backend del módulo. Se ocupará de manejar el procesamiento de datos y la comunicación con la base de datos.

- **Flask**

Tipo: Framework web para Python

Versión: 2.0.2

Justificación: Flask es ideal para aplicaciones web pequeñas y medianas por su simplicidad y flexibilidad. Será de utilidad para la creación de rutas, el manejo de autenticación y la visualización de datos..

- **MySQL**

Tipo: Sistema de gestión de bases de datos (SGBD)

Versión: 0.0.3

Justificación: MySQL proporciona un sistema robusto para el almacenamiento y manejo de datos, siendo una opción confiable que se integra bien con aplicaciones desarrolladas en Flask.

La combinación de estas herramientas y tecnologías permitirá el desarrollo de un módulo informático efi-

caz y moderno. Visual Studio Code como IDE proporcionará un entorno de trabajo óptimo, mientras que HTML, CSS, JavaScript y Bootstrap garantizarán una interfaz de usuario atractiva y funcional. Python y Flask ofrecerán la flexibilidad necesaria para el backend, y MySQL asegurará un manejo de datos eficiente. Esta elección estratégica de tecnologías crea una base sólida para el desarrollo y mantenimiento del proyecto, alineándose con las mejores prácticas del desarrollo web contemporáneo.

1.5. Metodología de desarrollo

Para desarrollar un software de manera efectiva es necesario definir una metodología de desarrollo. Las metodologías de software son marcos de trabajo que tienen como función guiar todo el proceso de desarrollo mediante la planificación y control de todas las actividades a llevar a cabo.

Según (Singh; Bagga y Kaur, 2020) afirman que existen diversos modelos para desarrollar un software que implementa un modelo predictivo, y entre ellos proponen las metodologías tradicionales y ágiles, de los cuales destacan los modelos: Cascada, Espiral, Scrum y XP. Varias bibliografías explican que el uso de ellas aumenta la probabilidad de éxito en el desarrollo, mediante la utilización de buenas prácticas (Morales–Carrillo; Cedeño–Valarezo; Bravo y Calderón, 2022) . En el estudio realizado de las distintas metodologías en base a algunos aspectos específicos se obtienen los resultados reflejados en la siguiente tabla 1.3:

Tabla 1.3. COMPARACIÓN DE METODOLOGÍAS

Aspecto	Metodologías ágiles	Metodologías tradicionales
Enfoque	Iterativo e incremental; se desarrolla en ciclos cortos, denominados "sprints"	Secuencial; el desarrollo se realiza en fases discretas y lineales.
Planificación	La planificación es flexible y se ajusta a los cambios a lo largo del desarrollo.	La planificación es exhaustiva y se establece al inicio del proyecto.
Requisitos	Los requisitos pueden cambiar y evolucionar durante el proceso, permitiendo adaptaciones.	Los requisitos se definen detalladamente al comienzo del proyecto y se espera que permanezcan constantes.
Colaboración	Se promueve la colaboración constante entre equipos multidisciplinarios y el cliente.	La comunicación tiende a ser formal y se realiza principalmente en las reuniones programadas.
Entrega de producto	Las entregas se realizan de forma continua y frecuente (al final de cada sprint).	La entrega se realiza una vez que se completa todo el ciclo de desarrollo, al final del proyecto.
Control de calidad	Se realiza de forma continua a lo largo del desarrollo, con pruebas integradas en cada iteración.	Se lleva a cabo una fase final de pruebas al completar el desarrollo, lo que puede resultar en un mayor riesgo de errores.
Documentación	La documentación es ligera y se centra en lo esencial; se prioriza la comunicación cara a cara.	La documentación es extensa y detallada, documentando cada fase del proyecto.
Adaptación al cambio	Se acepta y promueve el cambio, adaptándose a nuevas necesidades o feedback del cliente.	Cambios en requisitos pueden ser difíciles y costosos de implementar una vez que se ha avanzado en el proyecto.
Riesgo	Los riesgos se gestionan de forma continua gracias a la retroalimentación frecuente y al enfoque iterativo.	Los riesgos se evalúan al inicio del proyecto, pero pueden no ser detectados hasta más tarde en el proceso.
Foco en el cliente	Se pone énfasis en la satisfacción del cliente y sus necesidades a través de revisiones regulares y retroalimentación continua.	El cliente está involucrado principalmente al inicio para definir requisitos y al final para realizar pruebas.

Luego de este análisis se decide utilizar una metodología de desarrollo ágil, ya que se ajusta mejor al tipo de proceso que se quiere realizar. Una de las metodologías más utilizadas en esta categoría y que se ajusta perfectamente a las necesidades de este proyecto es Programación Extrema (XP). Elegir la metodología XP es beneficioso para equipos que buscan una alta calidad del software, flexibilidad ante cambios y una

fuerte colaboración con los clientes. Es muy útil para este proyecto ya que los requisitos pueden evolucionar rápidamente durante el desarrollo y se tiene el fin de entregar un producto de calidad en el menor tiempo y con el mínimo de errores posibles.

XP pone el énfasis en: la satisfacción del cliente y en la entrega rápida de software incremental, los equipos pequeños y muy motivados para efectuar el proyecto, los métodos informales, los productos del trabajo con mínima ingeniería de software y la sencillez general en el desarrollo (Pressman, 2010).

Esta metodología tiene como principios (Sommerville, 2011):

- Planeación Incremental (The Planning Game)
- Pequeñas entregas (small releases)
- Metáfora (Metaphor)
- Diseño Simple (Simple Design)
- Pruebas (Testing)
- Refactorización (Refactoring)
- Programación en Parejas (Pair Programming)
- Propiedad Colectiva (Collective Ownership)
- Integración Colectiva (Continuous Integration)
- 40 horas semanales
- Cliente en Casa
- Estándares de Codificación

Además de sus principios cuenta con 4 etapas principales: planificación, diseño, desarrollo y pruebas (IEEE, 1998). Estas se estarán desarrollando a lo largo de los próximos capítulos, llevando a cabo mediante ellas un proceso estructurado enfocado en la entrega temprana y frecuente de software funcional y valorado por el cliente.

Conclusiones parciales

Se concluye sobre la fundamentación teórica:

- El estudio del arte de la situación actual de las estaciones de carga eléctrica demostró ser un factor crucial para la expansión de la movilidad eléctrica.
- El análisis de sistemas homólogos proporcionó una visión más clara de las principales funcionalidades que se deben implementar en estos sistemas.
- Para el posterior desarrollo, se eligió Python como lenguaje de programación y Flask como framework.
- La elección de la metodología de desarrollo XP fue importante para guiar el proceso de desarrollo.

Análisis y diseño del software

Este capítulo se centra en la planeación y análisis del diseño del software, que serían las primeras fases de la metodología Extreme Programming (XP). La planeación en el desarrollo ágil implica escuchar al cliente, crear historias de usuario y asignarles valores y costos. El diseño sigue el principio de "mantenerlo sencillo" se guía por la implementación de las historias de usuario (Pressman, 2010). También, se establecerá la relevancia de estas fases en el marco de XP para garantizar un producto final de alta calidad y alineado con las necesidades del cliente. Este capítulo establece una base sólida preparando el camino para las fases de desarrollo y pruebas que se abordarán en el próximo capítulo.

2.1. Propuesta de solución

Luego de analizar los sistemas homólogos para definir las principales funcionalidades y las diferencias que deben resaltar en el proyecto; así como las herramientas y tecnologías a utilizar se puede dar paso a la descripción de la propuesta de solución. La propuesta de solución se centra en el desarrollo de una página web que permita a los usuarios visualizar en tiempo real las variables de operación de una estación de carga eléctrica mientras están cargando su vehículo. La solución se fundamenta en la usabilidad y en la capacidad de respuesta ante las necesidades del cliente. Esto se representa en la figura 2.1.

El proyecto se enfoca en ofrecer información útil al usuario final. Se busca brindar información sobre el proceso de carga, el estado de la estación y otras características relevantes, sin enfocarse en la gestión de flotas. Se prioriza una solución centrada en la experiencia del usuario y en facilitar la información para el cliente, una vez realizado el pago del servicio. En esencia, se desea desarrollar una aplicación web interactiva que permita a los usuarios de estaciones de carga eléctrica monitorizar en tiempo real los parámetros relevantes de la carga, como voltaje, corriente y tiempo restante hasta la finalización de la carga.

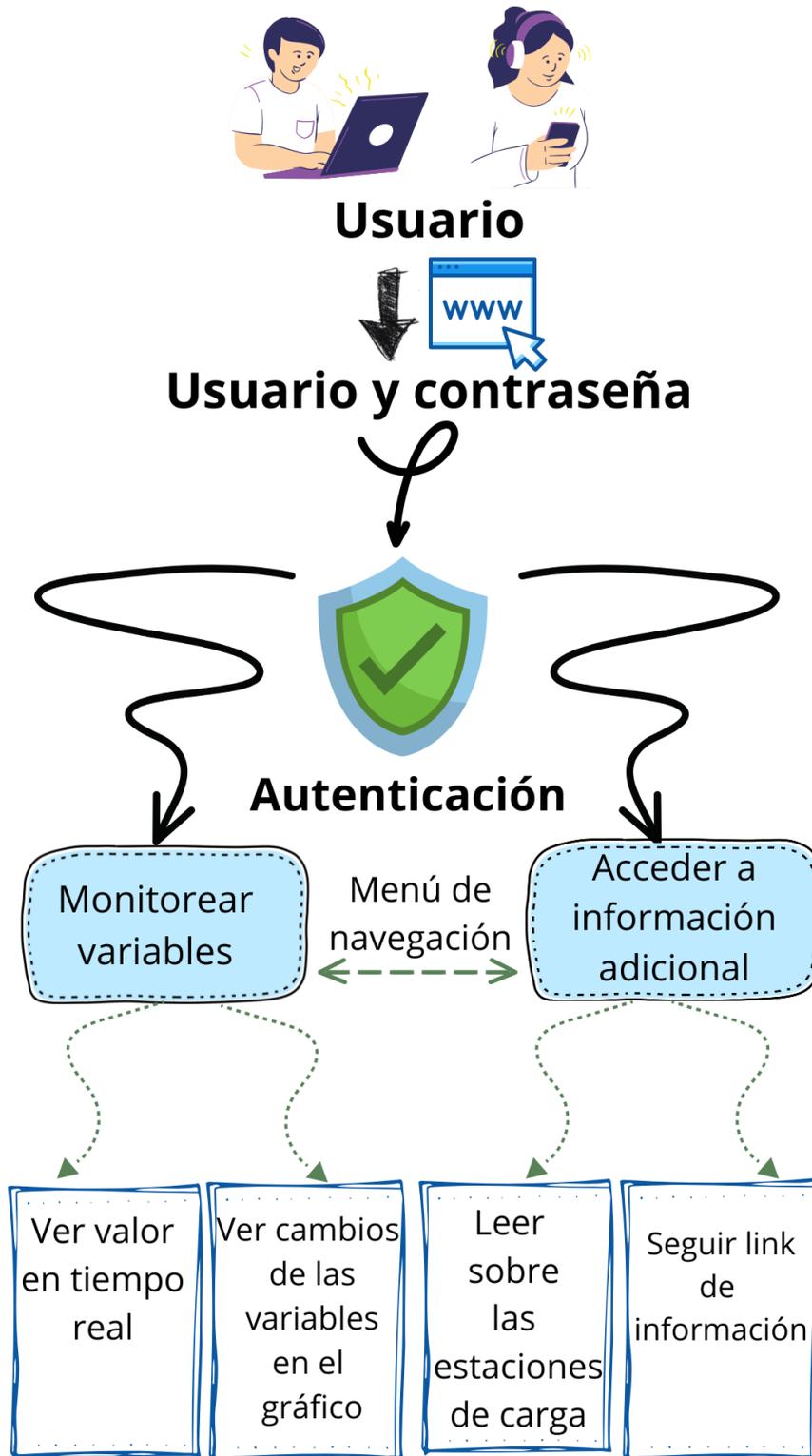


Figura 2.1. Propuesta de solución

2.2. Fase I: Planificación

En la fase de planificación se establece un enfoque centrado en la colaboración y la flexibilidad. Este proceso comienza con la identificación y creación de historias de usuario, las cuales representan funcionalidades que los clientes desean en el software. Estas historias se priorizan según su valor y relevancia para el usuario final. A continuación, el equipo se compromete a la planificación de iteraciones cortas, que típicamente duran entre una y tres semanas. Durante estas iteraciones, se define qué historias se implementarán, permitiendo que el equipo se adapte rápidamente a los cambios en los requisitos o en las prioridades del cliente.

El proceso de planificación implica la participación activa de todos los miembros del equipo, lo que fomenta un sentido de responsabilidad compartida. Además, se llevan a cabo revisiones periódicas para evaluar el progreso y realizar ajustes necesarios en las próximas iteraciones. En esencia, la fase de planificación en XP se caracteriza por su enfoque adaptable y colaborativo, buscando proporcionar un desarrollo ágil que se alinee con las expectativas del cliente a lo largo de todo el proceso (Molina y Ferreiro Ávila, 2023).

2.2.1. Historias de usuario

Las historias de usuario son una herramienta clave en XP que permite al equipo de desarrollo entender mejor las necesidades del usuario, priorizar tareas y asegurar que el software desarrollado cumpla con las expectativas del cliente. Son una herramienta fundamental en las metodologías de desarrollo ágil, especialmente en Extreme Programming (XP). Se utilizan para capturar los requisitos del software desde la perspectiva del usuario final, facilitando la comunicación entre los desarrolladores y los interesados en el proyecto. Cada historia de usuario describe una funcionalidad específica que el software debe ofrecer, formulada de manera sencilla y comprensible.

Además, aunque no es necesario suelen llevar información adicional, como criterios de aceptación, que especifican las condiciones que deben cumplirse para que la historia se considere completa. Esto permite a los equipos de desarrollo tener una guía clara sobre lo que se espera y facilita la realización de pruebas de aceptación al final de cada iteración.

La metodología XP sugiere que el tiempo ideal para desarrollar una historia de usuario es de aproximadamente tres semanas, lo que permite mantener un enfoque ágil y adaptable a los cambios en los requisitos (Reyes Cuba y Marín Apaza, 2019). Las historias de usuario también ayudan a identificar riesgos y a establecer prioridades en el desarrollo, lo que contribuye a una gestión más eficiente del proyecto (Peralta Marini e Hilaraca Apaza, 2020). A continuación se definen los aspectos fundamentales que deben incluir las HU para el proyecto de desarrollo según (Beck, 1999):

Número: Es el número asignado a cada HU.

Nombre: Atributo que posee el nombre de la HU.

Usuario: Es el usuario del sistema que utiliza o protagoniza el requisito que se describe.

Prioridad en negocio: Evidencia el nivel de prioridad de la HU en el negocio.

Riesgo en desarrollo: Evidencia el nivel de riesgo en caso de no realizarse.

Puntos estimados: Atributo que contiene la estimación hecha por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es uno equivale a una semana ideal de trabajo de lunes a viernes. En la metodología XP está definida una semana ideal como cinco días hábiles, trabajando cuarenta horas semanales.

Iteración asignada: Precisa la iteración en la que será desarrollada la HU.

Descripción: Posee una breve descripción de lo que realizará la HU.

Observaciones: Brinda información extra que se estime agregar para hacer más comprensible la HU.

Después de haber establecido los elementos fundamentales y basándose en los parámetros previamente definidos, a continuación 2.1, se presenta una de las HU, donde se hará la variación de incluir el prototipo de interfaz correspondiente y el programador responsable a cada requisito dentro de cada HU. El resto se encuentra en el Anexo A.

Tabla 2.1. Historia de usuario # 1

Historia de usuario	
Número: 1	Nombre: Acceder al inicio de sesión
Usuario: Usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.4	Iteración asignada: 2
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: La página de inicio al ser la primera impresión del cliente va a ser la puerta de entrada a la marca y lo que ofrece. Contará con el botón Iniciar que dará acceso al inicio de sesión. También tiene el slogan del proyecto con su nombre para hacer alusión al mismo.	
Observaciones:	
Interfaz:	
	

Una vez descritas las historias de usuario, se describen a continuación los requisitos no funcionales.

2.2.2. Requisitos no funcionales

Los requisitos no funcionales son características que describen cómo debe comportarse un sistema, en lugar de qué funcionalidades debe ofrecer. Se centran en aspectos como el rendimiento, la seguridad, la usabilidad, la escalabilidad y la compatibilidad, entre otros (Sommerville, 2011). Los requisitos no funcionales son cruciales para el desarrollo y éxito de sistemas de software así como su correcta identificación y gestión son fundamentales en el ciclo de vida del desarrollo de software. A menudo se consideran tan importantes como los requisitos funcionales, ya que impactan en la experiencia del usuario y en el éxito general del software. A continuación se especifican los requisitos no funcionales para el proyecto:

RNF de Usabilidad:

- El sistema debe poder usarse por cualquier persona con conocimientos básicos de navegación web.
- La aplicación web debe tener los iconos de cada variable acorde a la temática de la misma.
- El sistema debe ser accesible y adaptable a todos los dispositivos con acceso a la web.
- La aplicación web debe tener un diseño minimalista y fácil de entender.
- La aplicación web estará dirigida únicamente a mostrar información al usuario.

RNF de Interfaz o Apariencia externa:

- La aplicación web debe contar con interfaces donde predominen los colores claros y se muestre el logo de Tecxol.
- La aplicación web debe contar con una interfaz de usuario adaptable a diferentes dispositivos y tamaños.
- La aplicación web debe contar con interfaces con letra y tamaño legibles, así como vocablos de fácil comprensión.
- El diseño debe ser simple y claro con elementos reconocibles que identifiquen cada acción.
- La aplicación web debe tener una navegabilidad sencilla, siguiendo con las buenas prácticas de diseño web.

RNF de Software:

- La aplicación deberá ser compatible con los navegadores web más comunes (Chrome, Firefox, Safari, Edge) y estar optimizada para dispositivos móviles.
- El módulo debe conectarse a un sistema Wifi proporcionada por la estación de carga.

RNF de Hardware:

- El sistema del usuario debe tener al menos 2GB de RAM para soportar la carga del sistema.

RNF de Rendimiento:

- Los tiempos de carga de la página no deberán exceder los 2 segundos bajo condiciones normales.
- El sistema debe ser capaz de gestionar toda la información y asumir satisfactoriamente una carga de al menos 200 solicitudes simultaneas.
- El sistema debe estar disponible en todo momento.
- El sistema debe soportar un alto volumen de datos y mantener un rendimiento fluido.

RNF de Seguridad:

- La aplicación web deberá utilizar autenticación y autorización.

RNF de Restricciones del diseño y la implementación:

- El sistema debe estar desarrollado utilizando tecnologías de código abierto.
- El sistema debe estar escrito usando HTML superior a la versión 3.2.
- El sistema debe estar escrito usando CSS superior a la versión 2.1.
- El sistema debe estar escrito usando Javascript superior a ECMAScript 1999 (ES3).
- El sistema debe utilizar una base de datos relacional.

Una vez descritos los requisitos no funcionales se deben estimar los esfuerzos a realizar por cada HU definida.

2.2.3. Estimación de esfuerzos por cada HU

La estimación de esfuerzos por cada historia de usuario es un proceso utilizado en metodologías ágiles para evaluar cuánto trabajo se requiere para completar una historia de usuario específica. Esto se realiza generalmente durante la planificación de iteraciones o sprints, donde los equipos discuten y analizan cada historia de usuario en términos de tiempos, recursos y complejidad (Beck, 2000). La importancia de la estimación consiste en (Patton y Economy, 2014):

- Planificación: Ayuda a los equipos a planificar mejor el trabajo que se puede completar en una iteración.
- Priorización: Facilita la identificación de historias que ofrecen más valor en relación al esfuerzo necesario.
- Gestión de expectativas: Permite a los interesados entender el trabajo pendiente y establecer expectativas realistas.

En la planificación de las historias de usuario se estima el esfuerzo o tiempo empleado teniendo en cuenta que una semana equivale a cinco días laborables según se explicó en los puntos estimados anteriormente. En la siguiente tabla se muestra la estimación de esfuerzos por HU correspondientes a esta investigación:

Tabla 2.2. Estimación de esfuerzo por historia de usuario

Iteración	Historias de usuario		Puntos estimados (semanas)
1	1	Autenticar usuario	1
	2	Visualizar tiempo restante	0.6
	3	Visualizar potencia	0.6
	4	Visualizar voltaje de salida	0.6
	5	Visualizar recurso solicitado	0.6
	6	Visualizar recurso consumido	0.6
	7	Visualizar consumo de corriente	0.6
	8	Mostrar variación de la potencia en gráfico	0.6
	9	Mostrar variación del voltaje de salida en gráfico	0.6
	10	Mostrar variación del recurso consumido en gráfico	0.6
	11	Mostrar variación del consumo de corriente en gráfico	0.6
2	12	Acceder al inicio de sesión	0.4
	13	Cerrar sesión	0.3
	14	Mostrar información del proyecto	0.3
	15	Mostrar menú de navegación	1
	16	Mostrar información adicional	1
	17	Mostrar aviso de pérdida de conexión	0.5
Total			10.5

Como se evidenció en la tabla anterior el total de estimación de semanas de esfuerzo es 10,5. Una vez conocido esto se debe analizar el plan de iteraciones.

2.2.4. Plan de iteraciones

Un plan de iteraciones es un componente fundamental en metodologías ágiles, que detalla cómo se llevará a cabo el trabajo en un ciclo de desarrollo definido. Este documento proporciona una guía para el equipo sobre qué se va a abordar, los objetivos que se persiguen y cómo se medirá el progreso durante la iteración (Cohn, 2005). Su importancia en el proceso de planificación es :

- **Enfoque y Dirección:** Proporciona una estructura que guía al equipo sobre qué trabajar, ayudando a mantener el enfoque en las prioridades.
- **Transparencia:** Facilita la comunicación entre los miembros del equipo y los interesados, garantizando que todos estén alineados con los objetivos.
- **Mejora Continua:** Permite al equipo reflexionar al final de la iteración sobre lo que funcionó y lo que puede mejorarse en el futuro.

Un plan de iteraciones efectivo es clave para el éxito de los proyectos ágiles, ya que ayuda a los equipos a mantener la organización y efectividad en su trabajo. A continuación se presenta la estimación por semanas

para la entrega de cada iteración.

Tabla 2.3. Plan de duración de las iteraciones

Iteración	Historias de usuario		Duración (semanas)
1	1	Autenticar usuario	7.0
	2	Visualizar tiempo restante	
	3	Visualizar potencia	
	4	Visualizar voltaje de salida	
	5	Visualizar recurso solicitado	
	6	Visualizar recurso consumido	
	7	Visualizar consumo de corriente	
	8	Mostrar variación de la potencia en gráfico	
	9	Mostrar variación del voltaje de salida en gráfico	
	10	Mostrar variación del recurso consumido en gráfico	
	11	Mostrar variación del consumo de corriente en gráfico	
2	12	Acceder al inicio de sesión	3.5
	13	Cerrar sesión	
	14	Mostrar información del proyecto	
	15	Mostrar menú de navegación	
	16	Mostrar información adicional	
	17	Mostrar aviso de pérdida de conexión	
Total			10.5

Como se puede apreciar en la tabla anterior el proceso de desarrollo constará de 2 iteraciones. En la primera se estarán implementando las funcionalidades principales con las que debe cumplir el sistema, así como la seguridad mediante la autenticación. Para la segunda iteración se estaría llevando a cabo la implementación de aquellas funcionalidades ligadas a ofrecer información sobre el proyecto así como el cierre de sesión. Estas últimas funcionalidades aunque no sean de alta prioridad y riesgo no significa que sean innecesarias al cliente. Una vez realizada la planificación de las historias de usuario así como su plan de iteraciones se puede dar inicio a la próxima fase en la metodología XP consistente en el diseño del sistema.

2.3. Fase II: Diseño del sistema

La fase de diseño en la metodología Extreme Programming (XP) se centra en la creación de una arquitectura flexible y adaptable que facilite la evolución del software a lo largo del tiempo. En esta fase, se trabaja en conjunto para definir la estructura del código y los componentes del sistema, utilizando prácticas como el diseño simple, donde se prioriza la implementación de soluciones directas y claras. El diseño en XP es iterativo y mejora constantemente a medida que se realizan nuevas historias de usuario, permitiendo a los equipos refinar el sistema en respuesta a la retroalimentación continua y a los cambios en los requisitos del

cliente. Como ya se ha explicado el primer paso en esta fase es definir una arquitectura de software.

2.3.1. Arquitectura del software

La arquitectura de software se refiere a la estructura fundamental de un sistema de software, que abarca los componentes del sistema, sus interacciones, y los principios y directrices que rigen su diseño y evolución (Ken, 2024). Este concepto es esencial para garantizar que el software sea escalable, mantenible y capaz de satisfacer los requisitos funcionales y no funcionales.

Para el módulo a desarrollar se propone el uso de la arquitectura en capas. La arquitectura en capas organiza el código en diferentes capas, cada una con una responsabilidad específica. Las capas se conectan de forma jerárquica, donde las capas superiores dependen de las capas inferiores (Castro, 2015). Esta arquitectura es beneficiosa para este proyecto por su:

- Modularidad: Cada capa es independiente y se puede desarrollar y probar de forma aislada.
- Reutilización de código: Las capas inferiores se pueden reutilizar en otras aplicaciones.
- Mantenimiento: Es más fácil actualizar o modificar una capa sin afectar a otras capas.
- Escalabilidad: Se puede escalar cada capa de forma independiente según las necesidades del proyecto.
- Claridad: La estructura en capas facilita la comprensión y el mantenimiento del código.

El alcance de este módulo hasta el presente pretende únicamente mostrar información de la estación de carga concerniente al cliente por lo que el diseño arquitectónico quedaría de la siguiente forma:

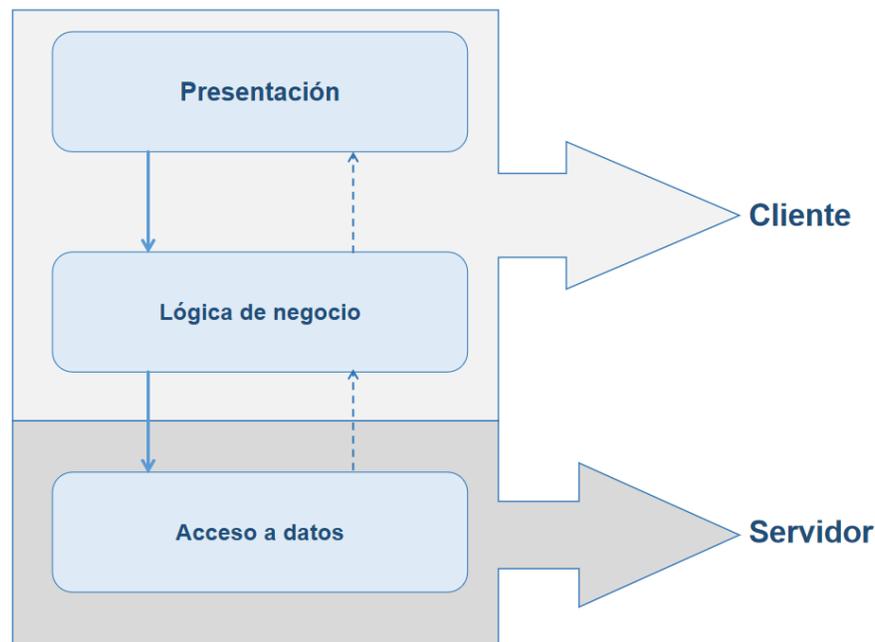


Figura 2.2. Diseño de arquitectura

Como indica la imagen 2.2 la arquitectura tendrá 3 capas fundamentales: presentación, lógica de negocio y acceso a datos. Las primeras 2 capas estarán dadas al funcionamiento del lado del cliente y la última del lado del servidor. Antes de esclarecer la función de cada capa es necesario especificar que las flechas que indican hacia abajo se refieren a cómo sería la interacción entre las capas en la llamada de métodos. En el caso de las flechas de líneas discontinuas que indican hacia arriba se refieren a la interacción al retornar la respuesta de las llamadas. A continuación se explica la función de cada capa teniendo en cuenta la estructura del código:

- **Presentación:** esta primera capa se encarga de la interfaz de usuario y la interacción con el mismo. Para ello dentro de la carpeta templates se encuentran los archivos HTML para la interfaz de usuario y en la carpeta static están contenidos los archivos estáticos (imágenes, JavaScript, etc.) y los archivos css para la interfaz de usuario.
- **Lógica de negocio:** contiene la lógica del negocio de la aplicación, las reglas de negocio y los procesos. Todos estos procesos se llevan a cabo en el archivo main_.py. Dentro de este archivo se maneja Flask que sería el framework de apoyo para obtener los datos.
- **Acceso a datos:** se encarga de la interacción con la base de datos y el acceso a la información. Para esto se utilizará MySQL para gestionar la base de datos y el archivo Database Script.sql para crear y editar las tablas.

Una vez planteado el diseño de la arquitectura es posible determinar los patrones de diseño a seguir.

2.3.2. Patrones de diseño

Los patrones de diseño brindan soluciones a problemas que se presentan durante el desarrollo de software, evitan duplicaciones de código y facilitan su reutilización. Los patrones de diseño son estructuras bien definidas que permiten mantener una lógica de organización en el código de un sistema, gracias a esto se puede crear software de calidad, con más facilidad de mantenimiento y con una mejor comprensión del código al buscar modularidad en el sistema (Alvarez; Larrea y Valencia, 2022). Ellos están clasificados en grandes grupos como se explicará a continuación. Uno de los grupos de patrones son los Patrones Generales de Asignación de Responsabilidades en Software (GRASP), estos son un conjunto de patrones de diseño de software que se utilizan para asignar responsabilidades en un diseño orientado a objetos (Ortega, 2021). De ellos se han utilizado:

- **Experto en información:** La clase Flask actúa como un .experto en información.^{al} encargarse de manejar las solicitudes HTTP y las respuestas de la aplicación. Esto significa que centraliza la lógica de control de flujo de la aplicación, gestionando las rutas y la interacción con las vistas y los modelos. Este patrón asegura que la lógica relacionada con la gestión de la aplicación se mantenga organizada y encapsulada dentro de la instancia de Flask.
- **Bajo acoplamiento:** La clase Flask actúa como un .experto en información.^{al} encargarse de manejar las solicitudes HTTP y las respuestas de la aplicación. Esto significa que centraliza la lógica de control

de flujo de la aplicación, gestionando las rutas y la interacción con las vistas y los modelos. Este patrón asegura que la lógica relacionada con la gestión de la aplicación se mantenga organizada y encapsulada dentro de la instancia de Flask.

- Alta cohesión: La función `addData()` en `chart_data.js` muestra alta cohesión porque está diseñada para realizar una tarea específica: agregar datos a una gráfica. La alta cohesión se refiere a que los elementos dentro de un módulo o clase están relacionados y trabajan juntos para cumplir con un propósito específico. Esto mejora la claridad del código y facilita su comprensión y mantenimiento, ya que cada función o módulo tiene una responsabilidad bien definida.

Por otra parte se encuentran los patrones GOF, también conocidos como Gang of Four (GoF), son un conjunto de patrones de diseño de software que son ampliamente reconocidos y utilizados en la comunidad de desarrollo de software para abordar problemas comunes de diseño y arquitectura de software (Gamma, 1995). De estos se han utilizado:

- Instancia única(Singleton): La clase Flask actúa como un Singleton en el sentido de que se crea una única instancia de la aplicación Flask. Esto es típico en aplicaciones web, donde se desea que solo haya una instancia de la aplicación que maneje todas las solicitudes y respuestas. Esta única instancia gestiona el estado y la configuración de la aplicación, asegurando que no haya múltiples instancias que puedan provocar inconsistencias.
- Método plantilla: La clase Flask actúa como un Singleton en el sentido de que se crea una única instancia de la aplicación Flask. Esto es típico en aplicaciones web, donde se desea que solo haya una instancia de la aplicación que maneje todas las solicitudes y respuestas. Esta única instancia gestiona el estado y la configuración de la aplicación, asegurando que no haya múltiples instancias que puedan provocar inconsistencias.
- Controlador frontal: `main_.py` actúa como un controlador frontal al manejar todas las solicitudes entrantes a la aplicación. En este patrón, una única clase o archivo (en este caso, `main_.py`) se encarga de recibir todas las solicitudes HTTP y dirigirlas a las funciones adecuadas (controladores) basadas en las rutas. Esto centraliza la lógica de enrutamiento y simplifica la gestión de las solicitudes.
- Método de fábrica: Aunque la clase Flask no es un método de fábrica en el sentido tradicional, se puede argumentar que actúa como tal al crear instancias de objetos y manejar las solicitudes. En este caso, Flask proporciona un marco para crear la aplicación y gestionar sus componentes, lo que puede verse como una forma de fábrica para crear y configurar objetos relacionados con la aplicación web.

Otros patrones utilizados han sido (Alvarez; Larrea y Valencia, 2022):

- Repositorio: El archivo `Database Script.sql` actúa como un repositorio, ya que se encarga de almacenar y recuperar los datos de la aplicación.
- El Patrón Modelo Vista Vista Modelo (MVVM en sus siglas del Inglés) permite aislar limpiamente la lógica de negocios y presentación de una aplicación de su interfaz de usuario, algo demostrado en el diseño arquitectónico.

Una vez definidos los patrones que serán de ayuda en este proyecto se debe realizar otra tarea ingenieril de la metodología XP que serían las tarjetas CRC.

2.3.3. Tarjetas Clase – Responsabilidad – Colaborador (CRC)

En el caso de la metodología XP el sustituto del diagrama de clases UML son las tarjetas CRC. Las tarjetas CRC son útiles para simular el proceso y detectar problemas potenciales y son un mecanismo eficaz para pensar en el software en un contexto orientado a objetos (Martínez del Cerro Parada, 2021). Las tarjetas CRC están compuestas por los siguientes elementos:

Clase: representa una colección de objetos similares.

Responsabilidad: es aquello que la clase conoce o realiza. Una clase puede cambiar el valor de lo que sabe pero no puede cambiar el valor de lo que saben otras clases.

Colaboraciones: adoptan una de dos formas, pueden ser un pedido de información o una petición de que se realice una operación.

A continuación se presentan dos tarjetas CRC correspondientes a esta investigación, el resto se encuentran en el Anexo B.

Tabla 2.4. Tarjeta CRC # 1

Tarjeta CRC	
Clase: Flask	
Responsabilidad	Colaboración
Crear y configurar la aplicación Flask Manejar solicitudes y respuestas Proporcionar lógica de negocio para la aplicación	Con la clase person para autenticar usuarios Con la clase database para acceder a la base de datos

Conclusiones parciales

Al finalizar las primeras dos fases de la metodología XP se concluye que:

- Fue crucial definir claramente los requisitos funcionales y no funcionales ya que precisó lo que se busca lograr con la propuesta de solución.
- La estimación de 10.5 semanas para dos iteraciones fue valiosa para planificar el tiempo de implementación, priorizando según la importancia de cada historia de usuario (HU).
- La arquitectura en capas fue la guía principal para el funcionamiento general del software.
- Los patrones de diseño seleccionados complementaron el proceso de diseño para la implementación de manera efectiva.

Implementación y pruebas del sistema

En este capítulo, se tiene como objetivo detallar la implementación y las pruebas efectuadas en el módulo de monitoreo, aplicando la metodología de Programación Extrema (XP). Se abordarán las dos últimas fases de la metodología pues son fundamentales para el éxito del proyecto. En primer lugar, la fase de implementación o desarrollo, donde se realizarán las tareas de ingeniería necesarias para la construcción del módulo. En segundo lugar, la fase de pruebas, donde se llevará a cabo una evaluación exhaustiva de todas las funcionalidades implementadas en el sistema con el fin de asegurar su calidad y correcto funcionamiento.

3.1. Fase III: Desarrollo

Esta fase de desarrollo en la Metodología de Programación Extrema (XP) representa un punto crucial en el ciclo de vida del proyecto de desarrollo de software (Beck, 2000). En esta etapa, las especificaciones y diseños iniciales pasan a ser un producto funcional, mediante distintas prácticas y principios. Para ello, se deben definir las tareas de ingeniería necesarias para cumplir con las historias de usuario descritas con anterioridad.

3.1.1. Tareas de ingeniería

Las tareas de ingeniería relacionadas con las historias de usuario incluyen el análisis de requerimientos para comprender las necesidades del usuario, la definición de criterios de aceptación que establecen condiciones de finalización, y la estimación del esfuerzo necesario para su implementación (Beck, 1999). Posteriormente, se realizan diseños técnicos, desarrollo del código, y pruebas para asegurar que la funcionalidad cumple con los requisitos. Finalmente, se lleva a cabo una revisión y demostración de la historia completada, junto con la documentación correspondiente. Estas actividades garantizan que las historias de usuario se transformen en soluciones efectivas para los usuarios. En este caso, se agrupan varias historias de usuario en una misma tarea de ingeniería. Esta práctica se puede utilizar en situaciones específicas donde múltiples historias de usuario están relacionadas o dependen entre sí, y su implementación puede beneficiarse de ser tratadas

como una única tarea. A continuación se expone un ejemplo de las tareas de ingeniería a desarrollar, el resto se encuentran en el Anexo C:

Tabla 3.1. Tarea de ingeniería # 1

Tarea	
Número de tarea: 1	Número de Historia de usuario: 2
Nombre de la tarea: Desarrollar la autenticación de usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 12 de agosto de 2024	Fecha de fin: 16 de agosto de 2024
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Consiste en implementar un sistema seguro que permita verificar la identidad de los usuarios al acceder a la aplicación. Este proceso incluye la creación de métodos de inicio de sesión, utilizando técnicas de encriptación para proteger las credenciales. Se deben establecer mecanismos de validación multifactor para mejorar la seguridad. El objetivo es garantizar que solo los usuarios autorizados puedan acceder a las funcionalidades y datos sensibles de la aplicación, asegurando así la integridad y confidencialidad de la información.	

Una vez que se han definido las tareas de ingeniería, es necesario llevar a cabo distintas pruebas del sistema para verificar la calidad de la solución implementada.

3.2. Fase IV: Pruebas

Las pruebas de software son un proceso sistemático y controlado que tiene como objetivo evaluar y verificar un sistema o programa de software para identificar defectos, errores o problemas en su funcionamiento (Pressman, 2010). En XP, se enfatiza la automatización de pruebas, la colaboración con el cliente y la retroalimentación continua para entregar software de alta calidad de manera incremental.

La fase de pruebas en un sistema informático es un proceso crítico que busca identificar y corregir errores, verificar el cumplimiento de los requisitos y asegurar el correcto funcionamiento del software en diversas condiciones. Durante esta fase, se aplican diferentes tipos de pruebas que abarcan varios aspectos del software. Las mismas pueden ser manuales o automatizadas. Existen varios niveles de prueba, entre ellos:

- **Pruebas Unitarias**

Las pruebas unitarias son el nivel más bajo de pruebas y se enfocan en la verificación de componentes individuales del código, como funciones, métodos o clases. Estas pruebas son generalmente automatizadas y permiten identificar errores en etapas tempranas del proceso de desarrollo. Su objetivo es confirmar que cada unidad de código se comporta como se espera de forma aislada.

- **Pruebas de Integración**

Las pruebas de integración se realizan una vez que se han completado las pruebas unitarias. Este nivel de prueba verifica la interacción entre diferentes módulos o componentes del sistema para asegurarse de que

funcionen correctamente juntos. Se evalúan tanto las interfaces entre módulos como el flujo de información y el comportamiento del sistema en conjunto. Las pruebas de integración pueden ser incrementales (probar los módulos a medida que se integran) o realizarse como un todo.

- **Pruebas de Sistema**

Las pruebas de sistema evalúan el sistema completo en su entorno operativo para asegurarse de que cumple con los requisitos establecidos. Este nivel de prueba abarca tanto pruebas funcionales como no funcionales, incluyendo rendimiento, seguridad y usabilidad. Los testers realizan pruebas que simulan el uso real del sistema por parte de los usuarios finales.

- **Pruebas de Aceptación**

Las pruebas de aceptación son el nivel en el que se determina si el sistema es aceptable para el cliente o usuario final. Suele incluir pruebas de aceptación del usuario, donde los usuarios finales validan que el software cumple con sus necesidades y expectativas. Este nivel de prueba asegura que el producto final satisface los requisitos del cliente y está listo para su implementación en producción.

Por último, aunque no son un nivel separado, las **pruebas de regresión** se realizan en los distintos niveles para asegurar que los cambios en el software (como nuevas funcionalidades o correcciones de errores) no afecten negativamente a las características existentes. Estas pruebas son esenciales para mantener la estabilidad del sistema a medida que evoluciona.

En conjunto, estos niveles de prueba forman un enfoque estructurado para garantizar que el software sea de alta calidad y esté listo para su despliegue. Cada nivel tiene su propósito específico y contribuye a detectar diferentes tipos de errores, lo que maximiza la fiabilidad y satisfacción del producto final.

En este proyecto se llevará a cabo un tipo de prueba en cada nivel. Toda esta fase tiene como objetivo asegurar que el producto final sea fiable, eficiente y seguro antes de su implementación en el entorno de producción. Esto no solo minimiza el riesgo de fallos y errores que puedan afectar a los usuarios, sino que también proporciona confianza en la calidad del software entregado, mejorando así la satisfacción del cliente y facilitando una adopción más fluida del sistema. Una fase de pruebas bien ejecutada contribuye al éxito del proyecto y a una buena reputación para el equipo de desarrollo.

3.2.1. Pruebas unitarias

Las pruebas unitarias como se explicó anteriormente se enfocan en comprobar el funcionamiento de las unidades más pequeñas de código, utilizando frameworks específicos que permiten ejecutar pruebas de forma rápida y repetida de forma automatizada. Un conjunto sólido de pruebas unitarias proporciona una red de seguridad valiosa para escribir software de alta calidad (Mamani, 2023). A continuación se muestran las pruebas realizadas en el archivo `main_.py`, para la clase `FlaskServer`:

Código fuente 3.1. Pruebas unitarias

```
1 import unittest
```

```
2 from flask import Flask
3 from main_ import FlaskServer
4
5 class FlaskServerTestCase(unittest.TestCase):
6     def setUp(self):
7         self.app = FlaskServer(__name__).appf.test_client()
8         self.app.testing = True
9
10    def test_home_page_get(self):
11        response = self.app.get('/')
12        self.assertEqual(response.status_code, 200)
13        self.assertIn(b'Tecxol', response.data)
14
15    def test_login_page_get(self):
16        response = self.app.get('/login')
17        self.assertEqual(response.status_code, 200)
18        self.assertIn(b'login', response.data)
19
20    def test_login_page_post_valid(self):
21        response = self.app.post('/login', data={
22            'username': 'amansingh',
23            'password': 'tesis'
24        })
25        self.assertEqual(response.status_code, 302)
26
27    def test_login_page_post_invalid(self):
28        response = self.app.post('/login', data={
29            'username': 'usuario',
30            'password': 'user'
31        })
32        self.assertEqual(response.status_code, 200)
33        self.assertIn(b'invalid Username or Password', response.data)
34
35    def test_protected_route_without_login(self):
36        response = self.app.get('/protected')
37        self.assertEqual(response.status_code, 404)
38
39    def test_logout(self):
40        self.app.post('/login', data={
41            'username': 'amansingh',
42            'password': 'tesis'
43        })
44        response = self.app.get('/logout')
45        self.assertEqual(response.status_code, 404)
46
47    def test_login_success(self):
```

```
48     response = self.app.post('/login', data={
49         'username': 'amansingh',
50         'password': 'tesis'
51     })
52     self.assertEqual(response.status_code, 302)
53     self.assertIn('/overview/', response.location)
54
55     def test_overview_with_invalid_session(self):
56         response = self.app.get('/overview/invalid_user/invalid_session')
57         self.assertEqual(response.status_code, 302)
58         self.assertIn('/login', response.location)
59
60     def test_profile_with_invalid_session(self):
61         response = self.app.get('/profile/invalid_user/invalid_session')
62         self.assertEqual(response.status_code, 302)
63         self.assertIn('/login', response.location)
64
65
66 if __name__ == '__main__':
67     unittest.main()
```

En total se realizaron 2 iteraciones en estas pruebas. En la 1ra iteración el 30% de las pruebas presentó algunas dificultades como redirección incorrecta o responder al error de forma contraria a lo que debería ser su funcionamiento normal. Al realizar acciones correctivas fueron subsanados estos errores. Luego, en las 2da iteración se obtuvo un 100% de satisfacción en la respuesta de las unidades a las pruebas, como se representa en la siguiente imagen:

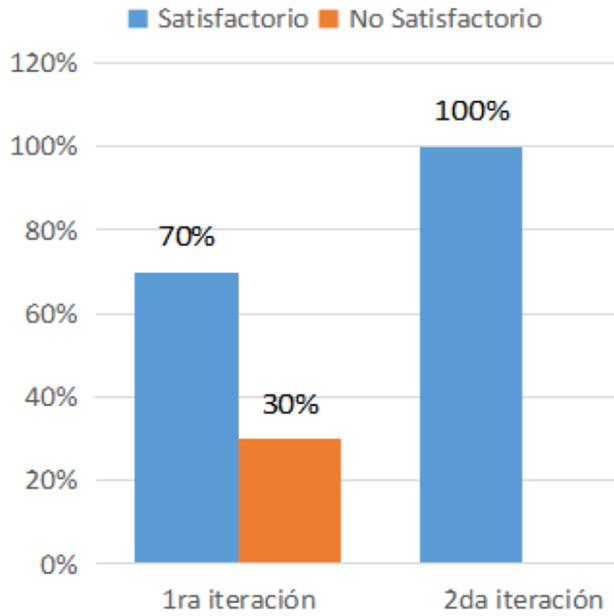


Figura 3.1. Resultados de las pruebas unitarias

Finalmente se obtuvo como resultado 9 pruebas unitarias satisfactorias para las unidades de código testeadas como se muestra a continuación:

```

..[result] Database connected!
[ERROR!]
list index out of range
User not logged in!
.[result] Database connected!
.[result] Database connected!
.[result] Database connected!
....
-----
Ran 9 tests in 11.707s

OK
    
```

Figura 3.2. Respuesta a las pruebas unitarias

Las pruebas realizadas arrojaron resultados satisfactorios y permitieron comprobar que las funciones implementadas realizan la actividad esperada. Además, estas permitieron prevenir posibles fallos o errores que podrían tomar mayor magnitud al continuar. Una vez terminadas las pruebas planificadas para este nivel, se

puede continuar al siguiente que sería las pruebas de integración.

3.2.2. Pruebas de Integración

Las pruebas de integración generalmente se usan para identificar los flujos a través del sistema de software (Sánchez y Barrezueta, 2022). El objetivo que tiene esta es verificar que el módulo de la página web puede recopilar y mostrar correctamente los datos del dispositivo SENECA trifásico guardados en una Raspberry Pi se explicó anteriormente, asegurando que la comunicación entre ambos dispositivos funcione adecuadamente. En este tipo de pruebas, no existe un consenso definido sobre el procedimiento a seguir, lo que deja la metodología a consideración del equipo de desarrollo. Para esta prueba de integración, se seguirán los siguientes **pasos a realizar**:

1. Configuración de la Raspberry Pi.
2. Asegurar el acceso a la red local.
3. Verificar que el dispositivo SENECA trifásico esté correctamente instalado y configurado.
4. Confirmar que el módulo de la página web a probar esté instalado en la Raspberry Pi.

Una vez completados estos pasos, los **resultados esperados** son los siguientes:

- El módulo debe ser capaz de recopilar datos del dispositivo SENECA y mostrarlos en la interfaz web sin errores.
- Los valores de potencia, voltaje y corriente deben reflejar con precisión los datos medidos por el dispositivo SENECA y ser visibles en la estación de carga.

Después de llevar a cabo los pasos mencionados, los **resultados obtenidos** fueron:

- Los datos se mostraron correctamente y la comunicación se mantuvo estable.
- Los valores de potencia, voltaje y corriente presentados en la página web coincidían en tiempo real con los mostrados en la estación de carga.

Para concluir la prueba, se realizó una comparación entre los resultados esperados y los resultados obtenidos. Dado que se lograron los resultados esperados en esta integración, se califica como satisfactoria.

3.2.3. Pruebas de Sistema

Estas pruebas evalúan la funcionalidad de todo un componente o sistema en conjunto, sin tener en cuenta los estados intermedios de la aplicación. Se diferencian principalmente de las pruebas de integración en que se enfocan en evaluar funcionalidades específicas del sistema y como se comporta en algunas situaciones, mientras que las de integración sólo asegura que interactúen de manera correcta (Guardo Vásquez, 2023). Entre las pruebas de sistema como se explicó anteriormente se encuentran las pruebas de usabilidad, estas se centran en evaluar de forma manual distintos aspectos para definir la facilidad y satisfacción con la que los usuarios interactúan con el sistema. Utiliza una lista de chequeo que considera aspectos como la claridad de la interfaz, la facilidad de navegación, la consistencia del diseño, la retroalimentación del sistema, la

facilidad de aprendizaje y la satisfacción general del usuario.

Para entender los resultados de la misma se debe definir que la forma de evaluar los indicadores en cuestión puntuando 1 en caso de mal (cuando la respuesta al indicador sea "No") y 0 en caso que el elemento revisado no presente errores (cuando la respuesta al indicador sea "Sí"), además, NP (No Procede) se usa para especificar que el indicador a evaluar no se puede aplicar a este sistema. A continuación se exponen los resultados de esta prueba:

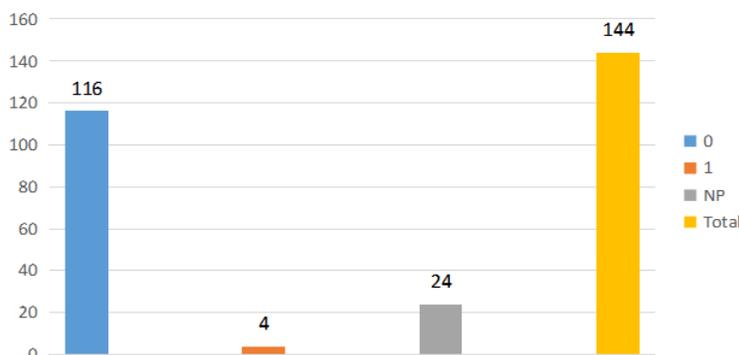


Figura 3.3. Resultados de la lista de chequeo

La prueba de usabilidad realizada reveló que la aplicación web cumple con los requisitos esenciales en una aplicación web, con un alto grado de satisfacción en las funcionalidades implementadas. De los 144 indicadores evaluados, un total de 116 respondieron de forma positiva, lo que representa un 80,6% de aprobación general. No procedieron al sistema implementado 24 indicadores representando el 16,7% del total, por lo que solo 4 de todos respondieron negativamente a la prueba conformando el 2,7% restante. Los indicadores que arrojaron respuestas negativas son:

- El sitio web no utiliza más de siete opciones principales en el menú. Esta característica, aunque no se cumple, no interfiere con el funcionamiento de la página ya que en las opciones que tiene el menú abarca todas las funcionalidades que debe tener.
- No es posible aumentar o disminuir el tamaño de la letra. La ausencia de esta característica tampoco interfiere de forma negativa al sitio ya que el tamaño utilizado es apropiado para su uso.
- Aún cuando la aplicación muestra información obtenida de una base de datos el usuario no puede elegir cuántos elementos desea ver en la página. Esta funcionalidad tampoco interfiere en gran medida porque la cantidad de datos a mostrar no es exagerada.
- No se proporciona texto para elementos no textuales para explicar el contenido a discapacitados visuales. Esta característica se debe tener en cuenta para futuras actualizaciones del sitio web.

Estos resultados sugieren que la aplicación web es intuitiva, fácil de usar y cumple con las expectativas. Sin embargo, es necesario analizar en profundidad los 4 indicadores negativos para identificar las áreas de mejora y optimizar la experiencia del usuario. La aplicación ofrece un alto nivel de satisfacción y funcionalidad, pero el análisis de los puntos débiles permitirá realizar las mejoras necesarias para alcanzar un nivel de

usabilidad aún más alto. De esta prueba no se dejan plasmadas acciones correctivas específicas porque las respuestas negativas no son de alto riesgo o importancia. Al finalizar las pruebas del sistema de forma general es momento de obtener la aprobación mediante las pruebas de aceptación.

3.2.4. Pruebas de Aceptación

Las pruebas de aceptación se centran en el comportamiento y capacidades de todo el sistema o producto en el entorno real (Marin Diaz; Trujillo Casañola y Buedo Hidalgo, 2020). Se llevan a cabo mediante casos de prueba por cada HU que reflejan escenarios del mundo real, asegurando que el software cumple con los requisitos del cliente; estos casos incluyen pruebas funcionales y criterios de aceptación que deben cumplirse para considerar el software como aceptable. A continuación se describe una de las pruebas realizadas, el resto se encuentran en el Anexo D:

Tabla 3.2. Prueba de aceptación # 1

Caso de prueba de aceptación	
Código: CPA_1	Historia de usuario: 1
Nombre: Acceder al inicio de sesión	
Descripción: Prueba de la página principal que da acceso al inicio de sesión.	
Condiciones de ejecución:	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Conectarse a la wifi de la estación de carga. 2. Entrar al sitio web mediante el código QR de la estación. 3. Presionar el botón Iniciar 4. Entrar en la página de inicio de sesión. 	
Resultados esperados: Se debe entrar al sitio web y debe permitir el acceso a la página de inicio de sesión.	

Se realizaron 17 casos de prueba de aceptación en dos iteraciones. En la primera iteración se encontraron fallas con el requisito No.17(Mostrar aviso de pérdida de conexión) por lo que se necesitaron acciones correctivas. Luego de las acciones correctivas se realizaron las pruebas de regresión junto a la segunda iteración dando como resultado en esta un 100 % de satisfacción con los casos de prueba. Terminando así el proceso de pruebas con un producto completamente funcional de acuerdo a los requisitos y las peticiones del cliente.

Conclusiones parciales

Al finalizar las últimas dos fases de la metodología XP se concluye que:

- Las tareas de ingeniería definidas fueron fundamentales para guiar el proceso de implementación.

- El proceso de pruebas realizado en los diferentes niveles verificó la calidad de la solución desarrollada, logrando un 100 % de aceptación.

Conclusiones

Al finalizar la presente investigación de acuerdo a los objetivos planteados se arriba a las siguientes conclusiones:

- El estudio del arte y de sistemas homólogos permitió identificar los estándares más adoptados a nivel mundial.
- La selección de XP como metodología de desarrollo garantizó un entorno adecuado para el desarrollo de software.
- La identificación de requisitos aseguró el cumplimiento de las necesidades del software.
- Visual Studio Code, Python con Flask, y el resto de herramientas y tecnologías seleccionadas fueron fundamentales para el proceso de implementación.
- Las fases de diseño, implementación y pruebas fueron claves para el éxito de la solución desarrollada.

Recomendaciones

A partir de los resultados obtenidos se recomienda:

- Según sea la aceptación general del público, desarrollar una versión móvil para brindar una mejor experiencia al usuario.

Referencias bibliográficas

- ACHARIGE, Sithara; HAQUE, M.E.; ARIF, Mohammad; HOSSEINZADEH, Nasser; HASAN, Kazi y OO, Aman, 2023. Review of Electric Vehicle Charging Technologies, Standards, Architectures, and Converter Configurations. *IEEE Access*. Vol. PP, págs. 1-1. Disponible desde DOI: 10.1109/ACCESS.2023.3267164 (vid. pág. 5).
- ALVAREZ, Oscar Danilo Gavilánez; LARREA, Natalia Patricia Layedra y VALENCIA, Marco Vinicio Ramos, 2022. Análisis comparativo de Patrones de Diseño de Software. *Polo del Conocimiento: Revista científico-profesional*. Vol. 7, n.º 7, págs. 2146-2165 (vid. págs. 26, 27).
- BECK, Kent, 1999. Embracing change with extreme programming. *Computer*. Vol. 32, n.º 10, págs. 70-77 (vid. págs. 19, 29).
- BECK, Kent, 2000. *Extreme programming explained: embrace change*. Addison–Wesley (vid. págs. 22, 29).
- BRENNAN, Morris; FOIADELLI, Federica; LEONE, Carola y LONGO, Michela, 2020. Electric vehicles charging technology review and optimal size estimation. *Journal of Electrical Engineering & Technology*. Vol. 15, págs. 2539-2552 (vid. pág. 6).
- CASTRO, Luis, 2015. *Arquitectura del Software*. Cengage Learning Editores (vid. pág. 25).
- COHN, Mike, 2005. *Agile estimating and planning*. Pearson Education (vid. pág. 23).
- DE LA, Leonela Del Rocio; CEDEÑO, Janeth Andreina García; IBARRA, Galo Eduardo Maldonado; IBARRA, Jerson Joseph Valdez et al., 2024. Desafíos y Oportunidades en la Infraestructura de Carga para Vehículos Eléctricos en América Latina y el Caribe. *Reincisol*. Vol. 3, n.º 6, págs. 984-1007 (vid. pág. 4).
- ELECTRICAL, Institute of y ENGINEERS, Electronics, 2023. *IEEE Standard for Electric Vehicle Charging*. Url: <https://IEEE.org>. Este estándar explica las especificaciones de carga para vehículos eléctricos, incluyendo el voltaje de salida. (vid. pág. 8).
- ESPAÑOLA, Real Academia, 2024. *Diccionario de la lengua española* [Consulta realizada el 16 de noviembre de 2024]. 23.ª. Url: <https://dle.rae.es>. [versión 23.7 en línea] (vid. pág. 9).
- GAMMA, Erich, 1995. Design patterns: elements of reusable object-oriented software. *Person Education Inc* (vid. pág. 27).
- GUARDO VÁSQUEZ, Santiago José, 2023. Testing para un sistema de publicación, entrega y corrección de prácticas de programación (vid. pág. 34).

- IEEE, 1998. *IEEE Standard for Software Life Cycle Processes*. Institute of Electrical y Electronics Engineers. N.º IEEE Std 12207-1998 (vid. pág. 16).
- KEN, Arizbé, 2024. *Arquitectura de software: ¿Qué es y qué tipos hay?* Url: <https://www.gluo.mx/blog/arquitectura-de-software-que-es-y-que-tipos-hay>. Accedido: 17 de noviembre de 2024 (vid. pág. 25).
- MAMANI, Cesar Adolfo Laura, 2023. Pruebas de Software para Microservicios. *Innovación y Software*. Vol. 4, n.º 1, págs. 151-160 (vid. pág. 31).
- MARIN DIAZ, Aymara; TRUJILLO CASAÑOLA, Yaimí y BUEDO HIDALGO, Denys, 2020. Estrategia de pruebas para organizaciones desarrolladoras de software. *Revista Cubana de Ciencias Informáticas*. Vol. 14, n.º 3, págs. 83-104 (vid. pág. 36).
- MARTÍNEZ DEL CERRO PARADA, Rachel, 2021. *Aplicación Android para la capacitación sobre el Sistema Integral de Gestión Empresarial DISTRA*. B.S. thesis. Universidad de las Ciencias Informáticas. Facultad 1 (vid. pág. 27).
- MOLINA, Alejandro Javier y FERREIRO ÁVILA, Enrique, 2023. *Módulo de Configuración para Sistema de Gestión de Alimentación XABAL SIGA versión 2.0*. B.S. thesis. Universidad de las Ciencias Informáticas. Facultad 4. (vid. pág. 19).
- MORALES-CARRILLO, Jessica; CEDEÑO-VALAREZO, Luis; BRAVO, Jesús Stefano Cajape y CALDERÓN, Jonathan Geovanny Ormaza, 2022. Metodologías de desarrollo de software y su ámbito de aplicación: Una revisión sistemática. *Revista Ibérica de Sistemas e Tecnologías de Información*. N.º E47, págs. 29-45 (vid. pág. 14).
- ORTEGA, Gilberto Andrés Vargas, 2021. Lineamientos para el diseño de aplicaciones web soportados en patrones GRASP. *Ciencia e Ingeniería: Revista de investigación interdisciplinaria en biodiversidad y desarrollo sostenible, ciencia, tecnología e innovación y procesos productivos industriales*. Vol. 8, n.º 2, págs. 4 (vid. pág. 26).
- PACHAS PACHAS, Diego Antonio y PACAHUALA VIRGILIO, Juan Adrian, 2023. Estudio del beneficio técnico y económico de adquirir un vehículo eléctrico frente a un vehículo a combustión interna en el parque automotor Lima-Perú (vid. pág. 4).
- PANDURO DEL AGUILA, Alvino y PEREZ ESPINOZA, Julio Cesar, 2023. Diseño de una estación de carga de baterías para el abastecimiento de energía de vehículos eléctricos (vid. pág. 3).
- PATTON, Jeff y ECONOMY, Peter, 2014. *User story mapping: discover the whole story, build the right product*. O'Reilly Media, Inc. (vid. pág. 22).
- PERALTA MARINI, Jhonny Yerson e HILASACA APAZA, Ever Saul, 2020. Desarrollo de un sistema de control de inventarios para pymes comercializadoras aplicando la metodología personalizada de XP (vid. pág. 19).

- PERSONAL, Autarquía, 2024. *¿Cuál es la potencia instalada a efectos legales de una estación de recarga de vehículos eléctricos?* Url: <https://autarquiapersonal.com/2024/02/03/cual-es-la-potencia-instalada-a-efectos-legales-de-una-estacion-de-recarga-de-vehiculos-electricos/>. Accedido: 2024-11-17 (vid. pág. 8).
- PRESSMAN, R. S., 2010. *Ingeniería del Software: Un Enfoque Práctico*. McGraw-Hill (vid. págs. 16, 17, 30).
- REYES CUBA, Percy Pavel y MARÍN APAZA, Roger Prudencio, 2019. *Aplicación web empleando la metodología XP para la gestión académica del Instituto de Informática de la Universidad Nacional del Altiplano Puno*. Url: <http://repositorio.unap.edu.pe/handle/20.500.14082/16057>. Tesis de maestría. Universidad Nacional del Altiplano Puno. Accedido: 17 de noviembre de 2024 (vid. pág. 19).
- SÁNCHEZ, Patricia María Marcillo y BARREZUETA, Lugio David Román, 2022. Análisis de la información generada para mantener la escalabilidad y persistencia del proceso de desarrollo de software. *Serie Científica de la Universidad de las Ciencias Informáticas*. Vol. 15, n.º 8, págs. 193-227 (vid. pág. 34).
- SINGH, J.; BAGGA, S. y KAUR, R., 2020. Software-based Prediction of Liver Disease with Feature Selection and Classification Techniques. *Procedia Computer Science*. Vol. 167, págs. 1970-1980. Disponible desde DOI: [10.1016/j.procs.2020.03.226](https://doi.org/10.1016/j.procs.2020.03.226) (vid. pág. 14).
- SOMMERVILLE, Ian, 2011. *Software Engineering*. Addison–Wesley (vid. págs. 16, 21).

Apéndices

Historias de Usuario

Tabla A.1. Historia de usuario # 2

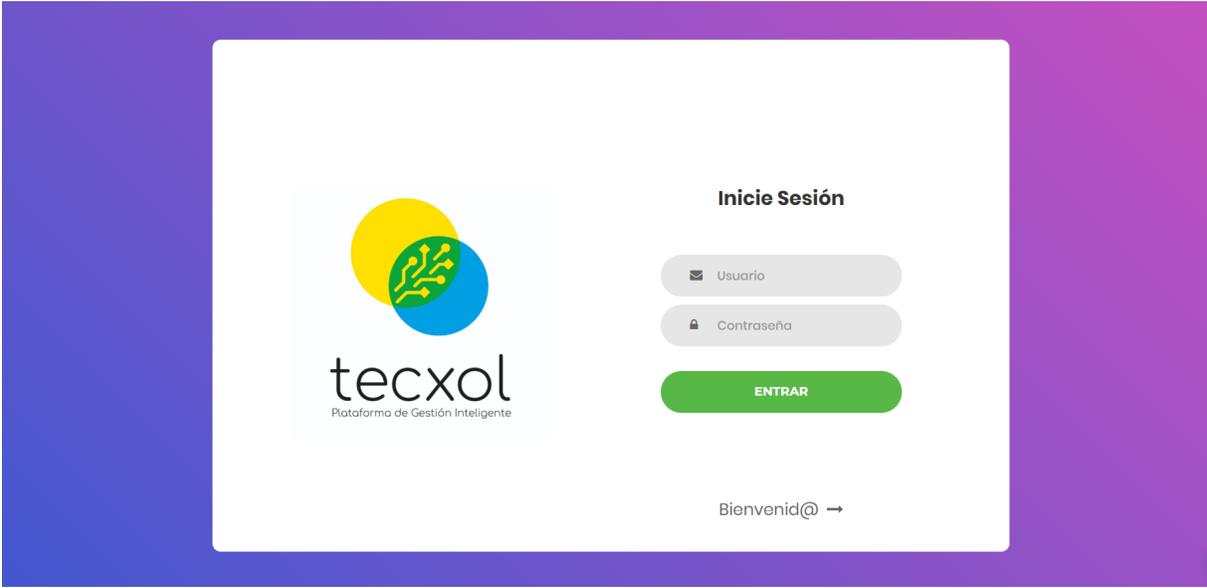
Historia de usuario	
Número: 2	Nombre: Autenticar usuario
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Muestra una pantalla con los campos Usuario y Contraseña y el botón Entrar. Al completar correctamente los campos y pinchar el botón se realiza el inicio de sesión. También tiene el logo del proyecto Tecxol.	
Observaciones:	
Interfaz:	
	

Tabla A.2. Historia de usuario # 3

Historia de usuario	
Número: 3	Nombre: Cerrar sesión
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.3	Iteración asignada: 2
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Se muestra en el menú siempre el botón Salir, al presionarlo se redirecciona directamente a la página de inicio, cerrando así la sesión.	
Observaciones: El usuario debe estar autenticado para poder realizar la acción.	
Interfaz:	
 <p>The screenshot shows a mobile application interface. At the top, there is a dark header with a hamburger menu icon and the text 'Menú' and 'Datos de la estación de carga'. Below the header, there is a white sidebar menu on the left with the text 'Hola. tesis', 'Menú', 'Cerrar Menú', 'Centro de datos', and 'Información'. A red 'Salir' button is visible at the bottom of the sidebar. The main content area on the right is dark grey and displays several data cards: '1000 W', '124 V', '0.05 kW/h', and '2 A'. There are also icons for a multimeter and a hand pointing to a button.</p>	

Tabla A.3. Historia de usuario # 4

Historia de usuario	
Número: 4	Nombre: Visualizar tiempo restante
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.6	Iteración asignada: 1
Programador responsable: Ailén Beatríz Ponce Rosabal	

Continúa en la próxima página

Tabla A.3. Continuación de la página anterior

Descripción: Muestra en tiempo real el tiempo estimado restante para completar la carga, con un icono en correspondencia a la variable y el color rojo asignado a la misma.
Observaciones: El usuario debe estar autenticado para poder visualizar el tiempo restante.
Interfaz:

Tabla A.4. Historia de usuario # 5

Historia de usuario	
Número: 5	Nombre: Visualizar potencia
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.6	Iteración asignada: 1
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Muestra en tiempo real la potencia de la corriente que está recibiendo el vehículo, con un icono en correspondencia a la variable y el color azul asignado a la misma.	
Observaciones: El usuario debe estar autenticado para poder visualizar la potencia.	

Continúa en la próxima página

Tabla A.4. Continuación de la página anterior

Interfaz:

Datos de la estación de carga

00:57:30 Tiempo Restante	1000 W Potencia	123 V Voltaje de salida	1 kW/h Recurso solicitado
0.0416667 kW/h Recurso consumido	2 A Consumo de corriente		

Tabla A.5. Historia de usuario # 6

Historia de usuario									
Número: 6	Nombre: Visualizar voltaje de salida								
Usuario: Usuario									
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto								
Puntos estimados: 0.6	Iteración asignada: 1								
Programador responsable: Ailén Beatríz Ponce Rosabal									
Descripción: Muestra en tiempo real el voltaje de salida de la estación de carga, con un icono en correspondencia a la variable y el color morado asignado a la misma.									
Observaciones: El usuario debe estar autenticado para poder visualizar el voltaje de salida.									
Interfaz:									
<p>Datos de la estación de carga</p> <table border="1"> <tr> <td> 00:57:30 Tiempo Restante</td> <td> 1000 W Potencia</td> <td> 123 V Voltaje de salida</td> <td> 1 kW/h Recurso solicitado</td> </tr> <tr> <td> 0.0416667 kW/h Recurso consumido</td> <td> 2 A Consumo de corriente</td> <td></td> <td></td> </tr> </table>		00:57:30 Tiempo Restante	1000 W Potencia	123 V Voltaje de salida	1 kW/h Recurso solicitado	0.0416667 kW/h Recurso consumido	2 A Consumo de corriente		
00:57:30 Tiempo Restante	1000 W Potencia	123 V Voltaje de salida	1 kW/h Recurso solicitado						
0.0416667 kW/h Recurso consumido	2 A Consumo de corriente								

Tabla A.6. Historia de usuario # 7

Historia de usuario	
Número: 7	Nombre: Visualizar recurso solicitado
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.6	Iteración asignada: 1
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Muestra en tiempo real la cantidad de recurso solicitado por el cliente, con un icono en correspondencia a la variable y el color naranja asignado a la misma.	
Observaciones: El usuario debe estar autenticado para poder visualizar el recurso que solicitó.	
Interfaz:	

Tabla A.7. Historia de usuario # 8

Historia de usuario	
Número: 8	Nombre: Visualizar recurso consumido
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.6	Iteración asignada: 1
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Muestra en tiempo real la cantidad de recurso que ha sido consumido por el cliente, con un icono en correspondencia a la variable y el color verde asignado a la misma.	
Observaciones: El usuario debe estar autenticado para poder visualizar el recurso consumido.	

Continúa en la próxima página

Tabla A.7. Continuación de la página anterior

Interfaz:

Datos de la estación de carga

Datos de la estación de carga

00:57:30 Tiempo Restante	1000 W Potencia	123 V Voltaje de salida	1 kW/h Recurso solicitado
0.0416667 kW/h Recurso consumido	2 A Consumo de corriente		

Tabla A.8. Historia de usuario # 9

Historia de usuario									
Número: 9	Nombre: Visualizar consumo de corriente								
Usuario: Usuario									
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto								
Puntos estimados: 0.6	Iteración asignada: 1								
Programador responsable: Ailén Beatríz Ponce Rosabal									
Descripción: Muestra en tiempo real el consumo de corriente del cliente, con un icono en correspondencia a la variable y el color amarillo asignado a la misma.									
Observaciones: El usuario debe estar autenticado para poder visualizar el consumo de corriente.									
Interfaz:									
<p>Datos de la estación de carga</p> <p>Datos de la estación de carga</p> <table border="1"> <tr> <td> 00:57:30 Tiempo Restante</td> <td> 1000 W Potencia</td> <td> 123 V Voltaje de salida</td> <td> 1 kW/h Recurso solicitado</td> </tr> <tr> <td> 0.0416667 kW/h Recurso consumido</td> <td> 2 A Consumo de corriente</td> <td></td> <td></td> </tr> </table>		00:57:30 Tiempo Restante	1000 W Potencia	123 V Voltaje de salida	1 kW/h Recurso solicitado	0.0416667 kW/h Recurso consumido	2 A Consumo de corriente		
00:57:30 Tiempo Restante	1000 W Potencia	123 V Voltaje de salida	1 kW/h Recurso solicitado						
0.0416667 kW/h Recurso consumido	2 A Consumo de corriente								

Tabla A.9. Historia de usuario # 10

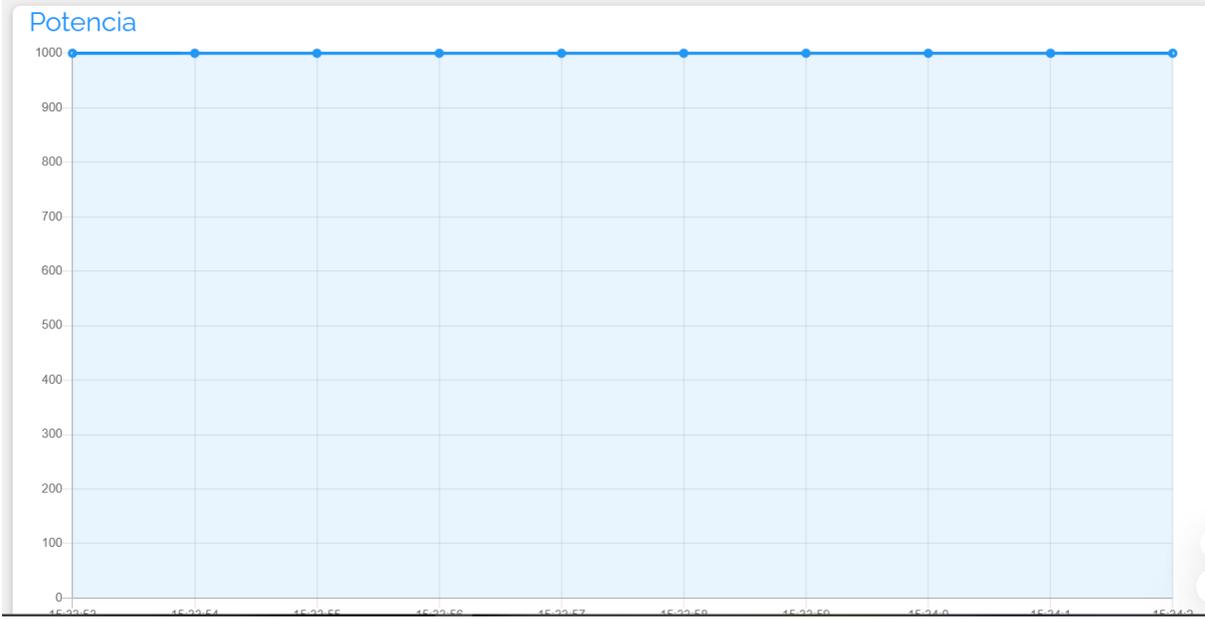
Historia de usuario	
Número: 10	Nombre: Mostrar variación de potencia en gráfico
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.6	Iteración asignada: 1
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Muestra un gráfico donde el usuario puede observar la fluctuación de la potencia en el transcurso de la carga de su vehículo. El gráfico se encuentra en correspondencia al color asignado a la variable, en este caso azul, así como el nombre de la variable para evitar confusiones.	
Observaciones: El usuario debe estar autenticado para acceder a la información del gráfico.	
Interfaz:	
	

Tabla A.10. Historia de usuario # 11

Historia de usuario	
Número: 11	Nombre: Mostrar variación del voltaje de salida en gráfico
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.6	Iteración asignada: 1
Programador responsable: Ailén Beatríz Ponce Rosabal	

Continúa en la próxima página

Tabla A.10. Continuación de la página anterior

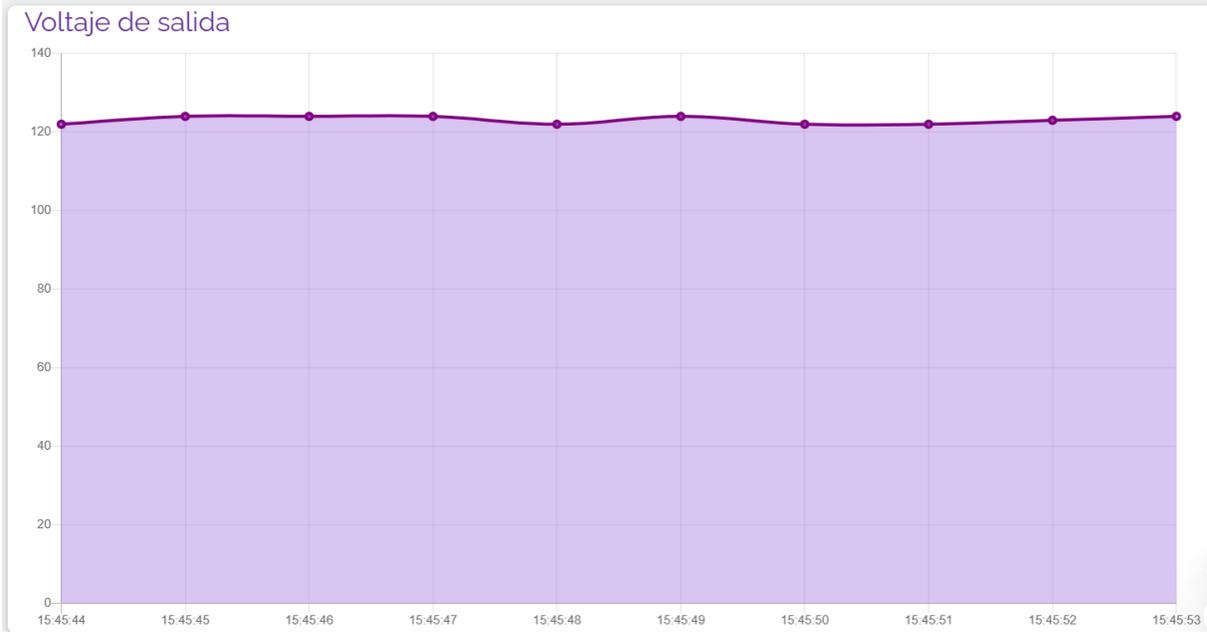
<p>Descripción: Muestra un gráfico donde el usuario puede observar la fluctuación del voltaje de salida en el transcurso de la carga de su vehículo. El gráfico se encuentra en correspondencia al color asignado a la variable, en este caso morado, así como el nombre de la variable para evitar confusiones.</p>																						
<p>Observaciones: El usuario debe estar autenticado para acceder a la información del gráfico.</p>																						
<p>Interfaz:</p>  <p>El gráfico muestra el voltaje de salida en un eje vertical que va de 0 a 140 y el tiempo en el eje horizontal que va de 15:45:44 a 15:45:53. La línea morada fluctúa entre aproximadamente 120V y 125V.</p> <table border="1"> <caption>Datos del gráfico de Voltaje de salida</caption> <thead> <tr> <th>Tiempo</th> <th>Voltaje (V)</th> </tr> </thead> <tbody> <tr><td>15:45:44</td><td>122</td></tr> <tr><td>15:45:45</td><td>124</td></tr> <tr><td>15:45:46</td><td>123</td></tr> <tr><td>15:45:47</td><td>124</td></tr> <tr><td>15:45:48</td><td>121</td></tr> <tr><td>15:45:49</td><td>124</td></tr> <tr><td>15:45:50</td><td>121</td></tr> <tr><td>15:45:51</td><td>121</td></tr> <tr><td>15:45:52</td><td>123</td></tr> <tr><td>15:45:53</td><td>124</td></tr> </tbody> </table>	Tiempo	Voltaje (V)	15:45:44	122	15:45:45	124	15:45:46	123	15:45:47	124	15:45:48	121	15:45:49	124	15:45:50	121	15:45:51	121	15:45:52	123	15:45:53	124
Tiempo	Voltaje (V)																					
15:45:44	122																					
15:45:45	124																					
15:45:46	123																					
15:45:47	124																					
15:45:48	121																					
15:45:49	124																					
15:45:50	121																					
15:45:51	121																					
15:45:52	123																					
15:45:53	124																					

Tabla A.11. Historia de usuario # 12

Historia de usuario	
Número: 12	Nombre: Mostrar variación del recurso consumido en gráfico
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.6	Iteración asignada: 1
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Muestra un gráfico donde el usuario puede observar la fluctuación del recurso consumido en el transcurso de la carga de su vehículo. El gráfico se encuentra en correspondencia al color asignado a la variable, en este caso verde, así como el nombre de la variable para evitar confusiones.	
Observaciones: El usuario debe estar autenticado para acceder a la información del gráfico.	

Continúa en la próxima página

Tabla A.11. Continuación de la página anterior

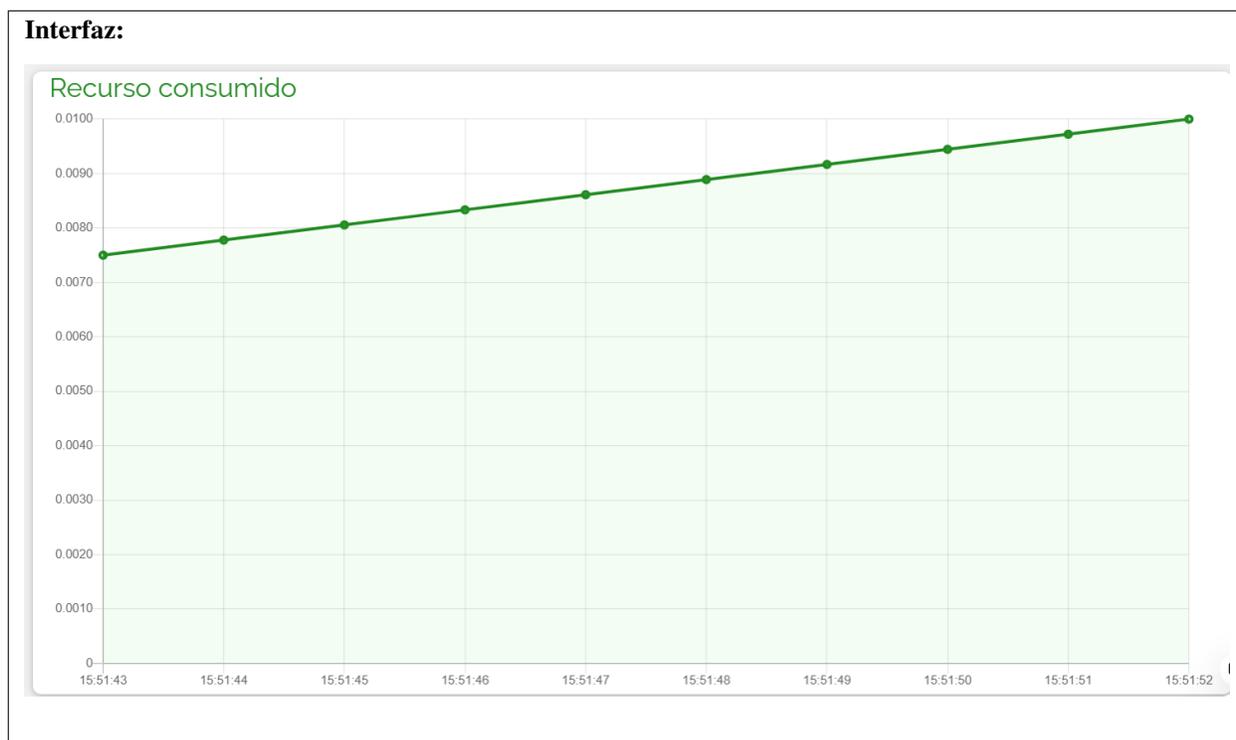


Tabla A.12. Historia de usuario # 13

Historia de usuario	
Número: 13	Nombre: Mostrar variación del consumo de corriente en gráfico
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 0.6	Iteración asignada: 1
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Muestra un gráfico donde el usuario puede observar la fluctuación del consumo de corriente en el transcurso de la carga de su vehículo. El gráfico se encuentra en correspondencia al color asignado a la variable, en este caso amarillo, así como el nombre de la variable para evitar confusiones.	
Observaciones: El usuario debe estar autenticado para acceder a la información del gráfico.	

Continúa en la próxima página

Tabla A.12. Continuación de la página anterior

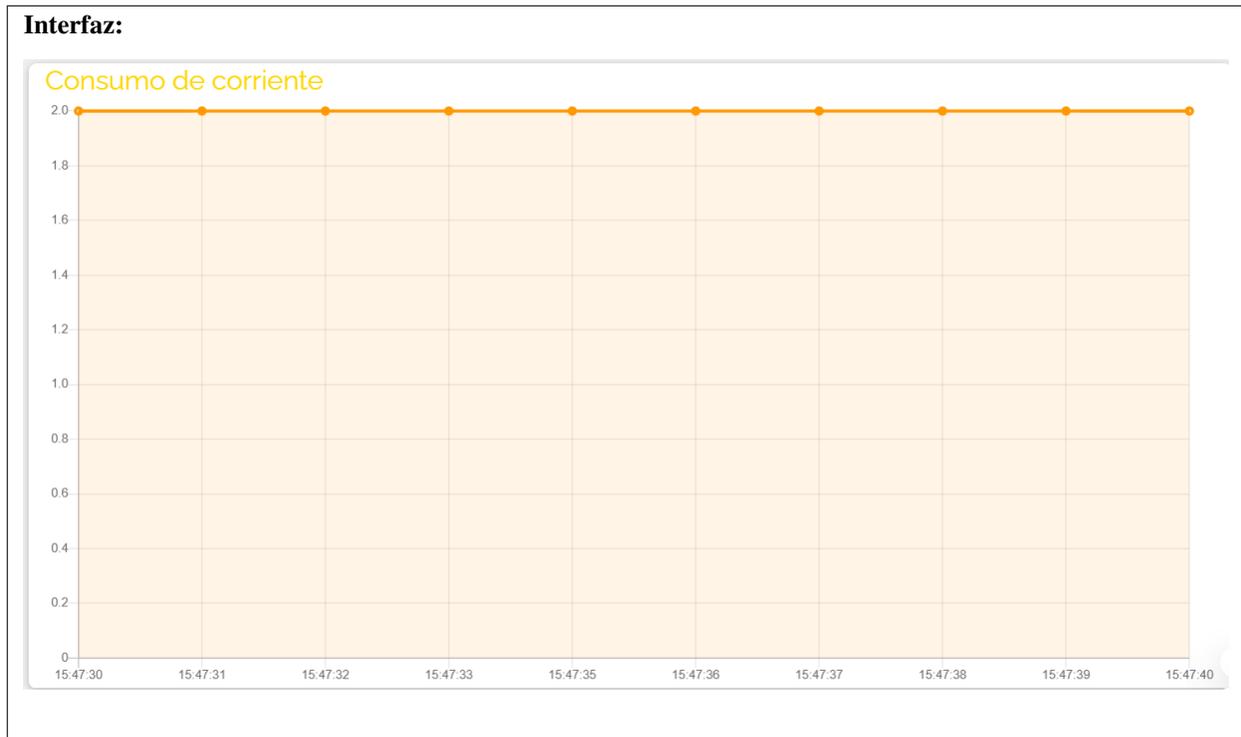


Tabla A.13. Historia de usuario # 14

Historia de usuario	
Número: 14	Nombre: Mostrar información del proyecto
Usuario: Usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.3	Iteración asignada: 2
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Informa en la página de inicio sobre la importancia de este proyecto. También proporciona un enlace a la página oficial del proyecto de la estación para informar sobre el mismo a los usuarios.	
Observaciones: Para acceder al enlace de información sobre la estación el usuario debe estar autenticado.	

Continúa en la próxima página

Tabla A.13. Continuación de la página anterior

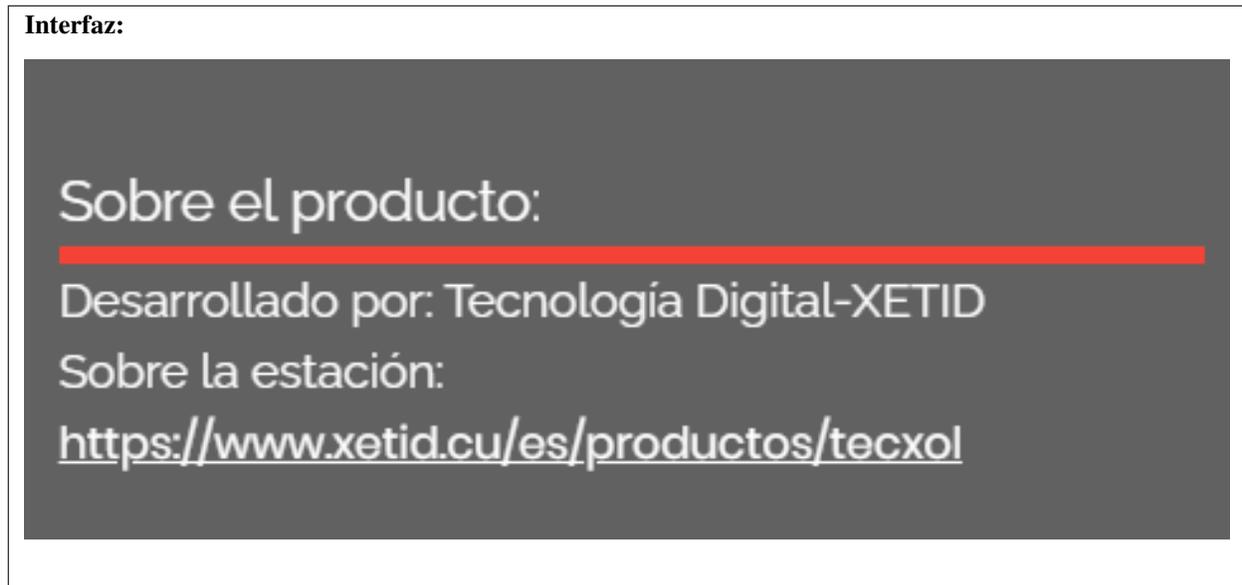


Tabla A.14. Historia de usuario # 15

Historia de usuario	
Número: 15	Nombre: Mostrar menú de navegación
Usuario: Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Muestra el menú para navegar en las 2 páginas del software y permite cambiar entre ellas. Contiene el botón Salir para cerrar sesión. Cuenta con 2 formatos, al ser responsive cuando el tamaño de pantalla es pequeño el menú se vuelve desplegable y tiene un botón adicional Cerrar Menú para volver a plegarlo.	
Observaciones: El usuario debe estar autenticado para acceder al menú.	

Continúa en la próxima página

Tabla A.14. Continuación de la página anterior

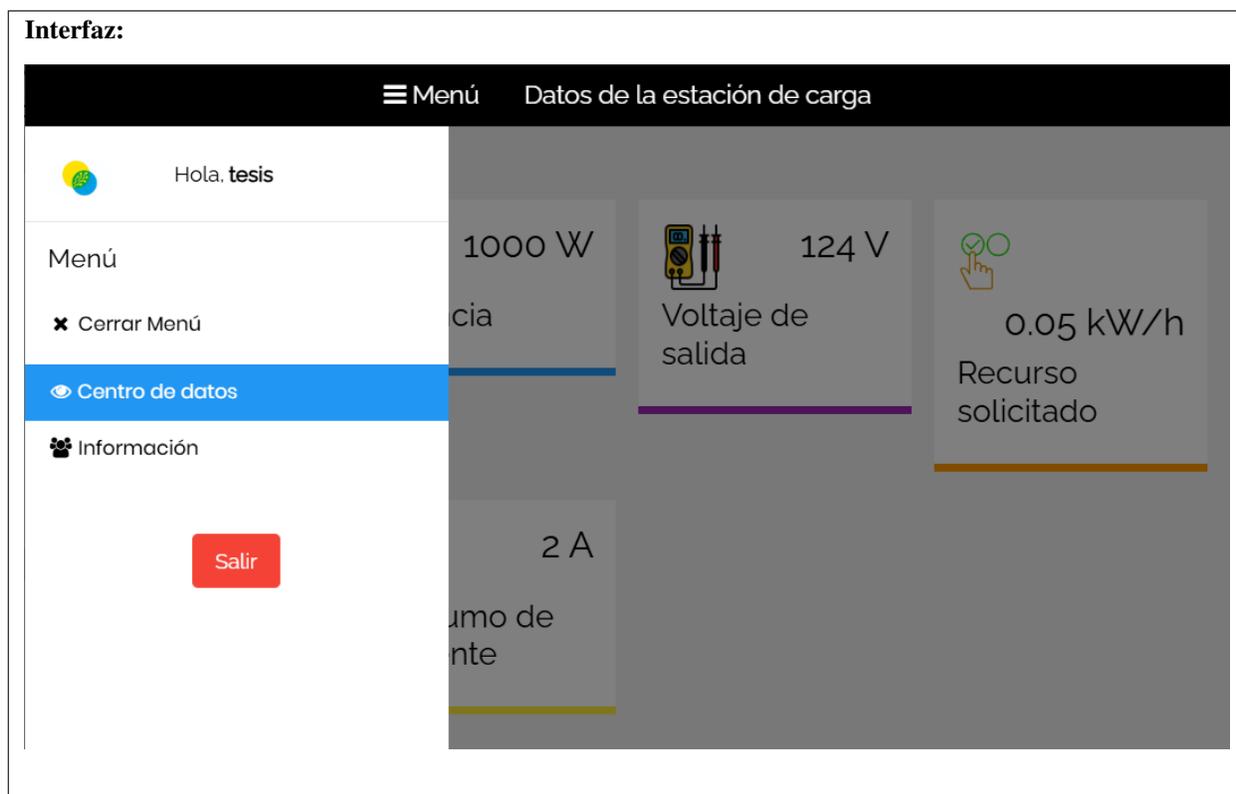


Tabla A.15. Historia de usuario # 16

Historia de usuario	
Número: 16	Nombre: Mostrar información adicional
Usuario: Usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Muestra varias imágenes con información educativa principalmente sobre vehículos eléctricos y estaciones de carga para los usuarios en caso de que sean de interés y a modo de entretenimiento.	
Observaciones: El usuario debe estar autenticado para acceder a la información..	

Continúa en la próxima página

Tabla A.15. Continuación de la página anterior

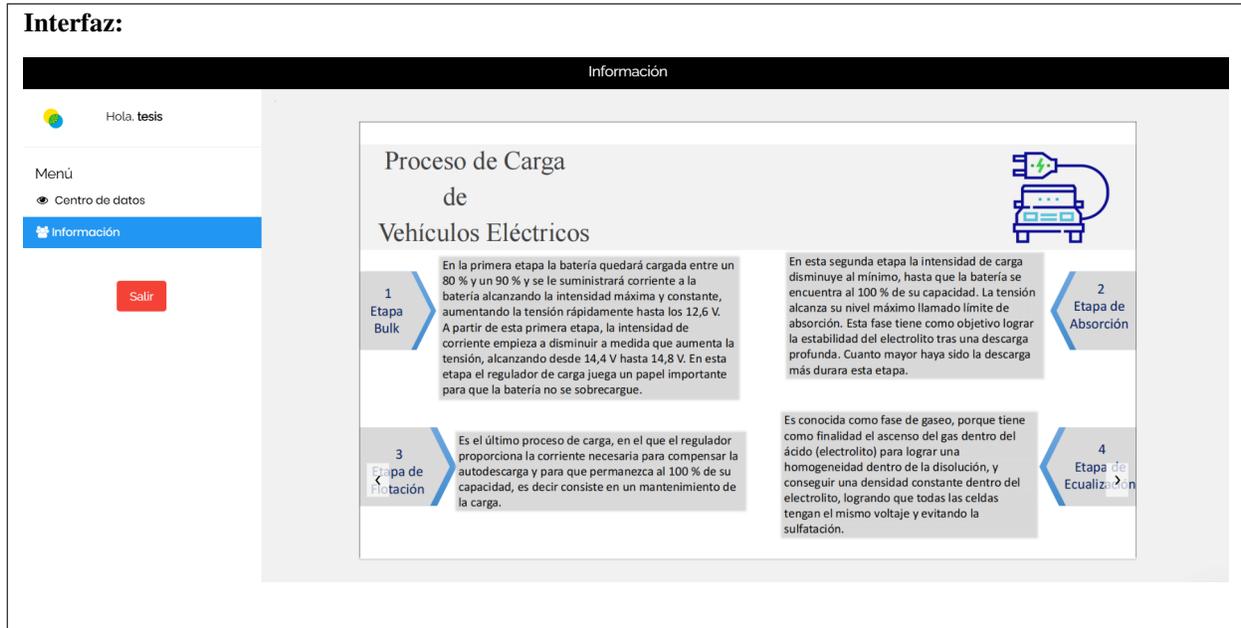
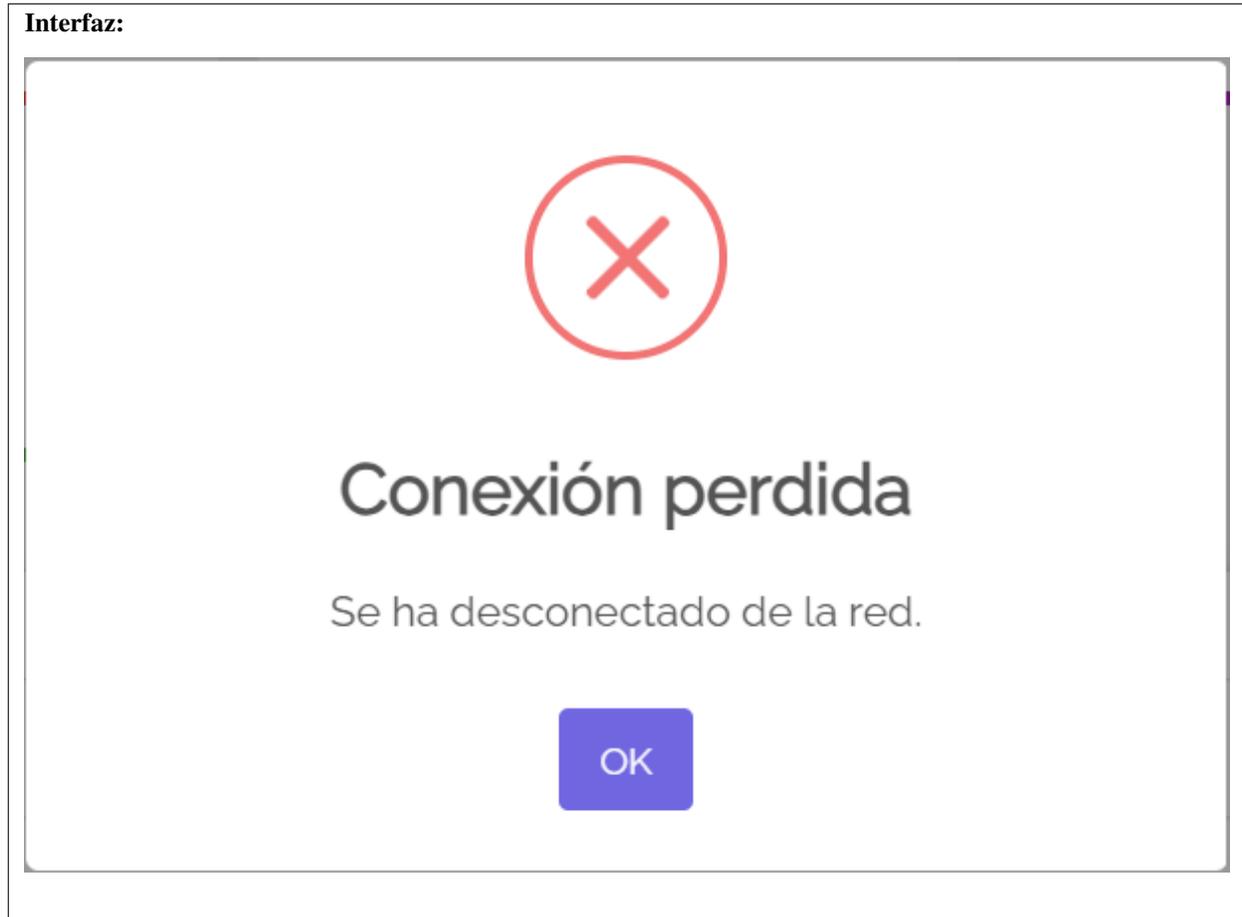


Tabla A.16. Historia de usuario # 17

Historia de usuario	
Número: 17	Nombre: Mostrar aviso de pérdida de conexión
Usuario: Usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Muestra una página notificando que se ha perdido la conexión a la estación de carga, por lo tanto la página no podrá funcionar.	
Observaciones:	

Continúa en la próxima página

Tabla A.16. Continuación de la página anterior



Tarjetas CRC

Tabla B.1. Tarjeta CRC # 2

Tarjeta CRC	
Clase: Person	
Responsabilidad	Colaboración
Representar a un usuario Autenticar usuarios Proporcionar información del usuario	Con la clase FlaskServer para autenticar usuarios Con la clase database para acceder a la base de datos

Tabla B.2. Tarjeta CRC # 3

Tarjeta CRC	
Clase: Database	
Responsabilidad	Colaboración
Proporcionar acceso a la base de datos Realizar operaciones de lectura y escritura	Con la clase FlaskServer para proporcionar datos para la aplicación Con la clase person para obtener información del usuario

Tareas de Ingeniería

Tabla C.1. Tarea de ingeniería # 2

Tarea	
Número de tarea: 2	Número de Historia de usuario: 4,5,6,7,8,9
Nombre de la tarea: Desarrollar la visualización de datos de la estación de carga	
Tipo de tarea: Desarrollo	Puntos estimados: 3,6
Fecha de inicio: 19 de agosto de 2024	Fecha de fin: 11 de septiembre de 2024
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Se centra en desarrollar una interfaz intuitiva que muestre de manera clara y efectiva información crucial, como el voltaje de salida, la corriente consumida y la potencia. Se proporcionará en tiempo real el valor de estas variables, facilitando así el monitoreo. El objetivo es ofrecer una herramienta poderosa que mejore la gestión de la estación de carga.	

Tabla C.2. Tarea de ingeniería # 3

Tarea	
Número de tarea: 3	Número de Historia de usuario: 10,11,12,13
Nombre de la tarea: Desarrollar la visualización de datos de la estación de carga en gráficos	
Tipo de tarea: Desarrollo	Puntos estimados: 2,4
Fecha de inicio: 12 de septiembre de 2024	Fecha de fin: 27 de septiembre de 2024
Programador responsable: Ailén Beatríz Ponce Rosabal	
Descripción: Se enfoca en el desarrollo de la representación visual de las variables claves de la estación de carga, como el voltaje de salida, la potencia y el consumo de corriente. Se implementarán gráficos de puntos que se irán uniendo para facilitar la interpretación de los datos a lo largo del tiempo. Esta visualización permitirá a los usuarios identificar patrones, tendencias y posibles anomalías en el funcionamiento de la estación, mejorando así la capacidad de análisis y monitoreo. El objetivo es proporcionar una herramienta gráfica que optimice la comprensión de la información, apoyando la eficiente gestión de la estación de carga.	

Tabla C.3. Tarea de ingeniería # 4

Tarea	
Número de tarea: 4	Número de Historia de usuario: 1,3
Nombre de la tarea: Desarrollar la página inicial y el cierre de sesión	
Tipo de tarea: Desarrollo	Puntos estimados: 0,7
Fecha de inicio: 30 de septiembre de 2024	Fecha de fin: 3 de octubre de 2024
Programador responsable: Ailén Beatríz Ponce Rosabal	
<p>Descripción: Se centra en desarrollar una experiencia de usuario fluida y segura dentro de la aplicación. La página de inicio servirá como el punto de acceso principal, proporcionando una navegación intuitiva y mostrando información relevante sobre la estación de carga, así como el acceso a la autenticación. Al mismo tiempo, se desarrollará el mecanismo de cierre de sesión que garantice la desconexión segura del usuario y su redirección a esta página de inicio, protegiendo así los datos. Esta actividad no solo optimiza la usabilidad de la aplicación, sino que también refuerza la seguridad al permitir a los usuarios finalizar su sesión de manera eficiente. El objetivo es crear un entorno accesible y seguro, mejorando la interacción del usuario con la plataforma.</p>	

Tabla C.4. Tarea de ingeniería # 5

Tarea	
Número de tarea: 5	Número de Historia de usuario: 14,16
Nombre de la tarea: Desarrollar la visualización de información adicional	
Tipo de tarea: Desarrollo	Puntos estimados: 1,3
Fecha de inicio: 3 de octubre de 2024	Fecha de fin: 11 de octubre de 2024
Programador responsable: Ailén Beatríz Ponce Rosabal	
<p>Descripción: Se enfoca en desarrollar una sección que ofrezca detalles relevantes y contextuales acerca de las características técnicas y objetivos del sistema. Esta sección incluirá descripciones de las variables utilizadas, beneficios de la estación de carga y guías sobre el uso eficiente de la estación. El objetivo es proporcionar a los usuarios un recurso informativo completo que enriquezca su experiencia, fomente un uso adecuado y promueva el interés en el sistema, contribuyendo así a la adopción exitosa de las estaciones de carga.</p>	

Tabla C.5. Tarea de ingeniería # 6

Tarea	
Número de tarea: 6	Número de Historia de usuario: 15
Nombre de la tarea: Desarrollar el menú de navegación	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 14 de octubre de 2024	Fecha de fin: 18 de octubre de 2024
Programador responsable: Ailén Beatríz Ponce Rosabal	

Continúa en la próxima página

Tabla C.5. Continuación de la página anterior

<p>Descripción: Se centra en diseñar una interfaz intuitiva y accesible que facilite el acceso a las diversas funcionalidades de la aplicación. El menú incluirá enlaces claros y categorizados a las secciones. Se priorizará la usabilidad, asegurando que los usuarios puedan navegar de manera eficiente y encontrar la información que necesitan sin complicaciones. Además, el diseño será responsivo, adaptándose a diferentes dispositivos y tamaños de pantalla. El objetivo de esta actividad es proporcionar una estructura de navegación clara y eficiente que mejore la interacción del usuario con la aplicación, promoviendo una experiencia positiva y fluida.</p>

Tabla C.6. Tarea de ingeniería # 7

Tarea	
Número de tarea: 7	Número de Historia de usuario: 17
Nombre de la tarea: Desarrollar alerta cuando se pierda la conexión	
Tipo de tarea: Desarrollo	Puntos estimados: 0,5
Fecha de inicio: 21 de octubre de 2024	Fecha de fin: 23 de octubre de 2024
Programador responsable: Ailén Beatríz Ponce Rosabal	
<p>Descripción: Se centra en desarrollar un mecanismo proactivo que asegure la continuidad del servicio y la experiencia del usuario. Este sistema de alertas se activará automáticamente al detectar la desconexión de la red, informando al usuario mediante una notificación visual. El objetivo es garantizar que los usuarios estén informados de manera oportuna, mejorando así la confiabilidad y la percepción del sistema.</p>	

Tabla D.1. Prueba de aceptación # 2

Caso de prueba de aceptación	
Código: CPA_2	Historia de usuario: 2
Nombre: Autenticar usuario	
Descripción: Prueba de la funcionalidad de autenticar usuarios.	
Condiciones de ejecución: El usuario debe haber entrado en la página principal del sitio.	
Pasos de ejecución: 1.Muestra una interfaz con los campos Usuario y Contraseña y el botón Entrar 2.Llena los campos 3.Presiona el botón Entrar 4.Se muestra la interfaz de Datos de la estación de carga.	
Resultados esperados: El usuario debe poder autenticarse de forma satisfactoria y alertar en caso de llenar mal o no llenar los campos.	

Tabla D.2. Prueba de aceptación # 3

Caso de prueba de aceptación	
Código: CPA_3	Historia de usuario: 3
Nombre: Cerrar sesión	
Descripción: Prueba de la funcionalidad para cerrar sesión.	
Condiciones de ejecución: El usuario debe estar autenticado.	

Continúa en la próxima página

Tabla D.2. Continuación de la página anterior

<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Se muestra el botón Salir en el menú de navegación 2. Presiona el botón 3. Cierra la sesión actual del usuario redireccionándolo a la página principal.
<p>Resultados esperados: El usuario debe poder cerrar sesión de forma satisfactoria y ser enviado a la página inicial.</p>

Tabla D.3. Prueba de aceptación # 4

Caso de prueba de aceptación	
Código: CPA_4	Historia de usuario: 4
Nombre: Visualizar tiempo restante	
Descripción: Prueba de la funcionalidad para visualizar en tiempo real el tiempo restante de carga.	
Condiciones de ejecución: El usuario debe estar autenticado.	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Entra en la página del centro de datos 2. Localiza el campo que indica el "Tiempo Restante" en la interfaz. 3. Mantén la página abierta y observa el tiempo restante durante unos minutos (recomendable 5 minutos). 4. Asegúrate de que el tiempo restante se actualice en tiempo real y que el valor disminuya automáticamente. 5. Navega a otra sección de la aplicación y luego vuelve a la vista de la estación de carga sin refrescar la página. 	
Resultados esperados: El usuario debe poder visualizar en tiempo real el Tiempo Restante de carga.	

Tabla D.4. Prueba de aceptación # 5

Caso de prueba de aceptación	
Código: CPA_5	Historia de usuario: 5
Nombre: Visualizar potencia	
Descripción: Prueba de la funcionalidad para visualizar en tiempo real la potencia con que funciona la estación de carga.	
Condiciones de ejecución: El usuario debe estar autenticado.	

Continúa en la próxima página

Tabla D.4. Continuación de la página anterior

<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Entra en la página del centro de datos 2. Localiza el campo que indica la "Potencia.^{en} la interfaz. 3. Mantén la página abierta y observa durante unos minutos (recomendable 5 minutos). 4. Asegúrate de que la potencia se actualice en tiempo real de acuerdo a los datos de la estación. 5. Navega a otra sección de la aplicación y luego vuelve a la vista de la estación de carga sin refrescar la página.
<p>Resultados esperados: El usuario debe poder visualizar en tiempo real la potencia con que está trabajando la estación de carga.</p>

Tabla D.5. Prueba de aceptación # 6

Caso de prueba de aceptación	
Código: CPA_6	Historia de usuario: 6
Nombre: Visualizar voltaje de salida	
Descripción: Prueba de la funcionalidad para visualizar en tiempo real el voltaje de salida que está teniendo la estación de carga.	
Condiciones de ejecución: El usuario debe estar autenticado.	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Entra en la página del centro de datos 2. Localiza el campo que indica el "Voltaje de salida.^{en} la interfaz. 3. Mantén la página abierta y observa durante unos minutos (recomendable 5 minutos). 4. Asegúrate de que el voltaje se actualice en tiempo real de acuerdo a los datos de la estación. 5. Navega a otra sección de la aplicación y luego vuelve a la vista de la estación de carga sin refrescar la página. 	
Resultados esperados: El usuario debe poder visualizar en tiempo real el voltaje de salida con que está trabajando la estación de carga.	

Tabla D.6. Prueba de aceptación # 7

Caso de prueba de aceptación	
Código: CPA_7	Historia de usuario: 7
Nombre: Visualizar recurso solicitado	
Descripción: Prueba de la funcionalidad para visualizar el recurso solicitado en la estación de carga.	

Continúa en la próxima página

Tabla D.6. Continuación de la página anterior

Condiciones de ejecución: El usuario debe estar autenticado.
Pasos de ejecución: 1.Entra en la página del centro de datos 2.Localiza el campo que indica el Recurso solicitado. ^{en} la interfaz. 3.3.Mantén la página abierta y observa. 4.Navega a otra sección de la aplicación y luego vuelve a la vista de la estación de carga sin refrescar la página.
Resultados esperados: El usuario debe poder visualizar en tiempo real el recurso que solicitó en la estación de carga.

Tabla D.7. Prueba de aceptación # 8

Caso de prueba de aceptación	
Código: CPA_8	Historia de usuario: 8
Nombre: Visualizar recurso consumido	
Descripción: Prueba de la funcionalidad para visualizar el recurso consumido de la estación de carga.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: 1.Entra en la página del centro de datos 2.Localiza el campo que indica el Recurso consumido. ^{en} la interfaz. 3.Mantén la página abierta y observa durante unos minutos (recomendable 5 minutos). 4.Asegúrate de que el recurso consumido se actualice en tiempo real. 5.Navega a otra sección de la aplicación y luego vuelve a la vista de la estación de carga sin refrescar la página.	
Resultados esperados: El usuario debe poder visualizar en tiempo real el recurso que ha consumido de la estación.	

Tabla D.8. Prueba de aceptación # 9

Caso de prueba de aceptación	
Código: CPA_9	Historia de usuario: 9
Nombre: Visualizar consumo de corriente	
Descripción: Prueba de la funcionalidad para visualizar el consumo de corriente en la estación de carga.	

Continúa en la próxima página

Tabla D.8. Continuación de la página anterior

Condiciones de ejecución: El usuario debe estar autenticado.
Pasos de ejecución: 1.Entra en la página del centro de datos 2.Localiza el campo que indica el Consumo de corriente. ^{en} la interfaz. 3.Mantén la página abierta y observa durante unos minutos (recomendable 5 minutos). 4.4.Asegúrate de que el consumo de corriente se actualice en tiempo real. 5.Navega a otra sección de la aplicación y luego vuelve a la vista de la estación de carga sin refrescar la página.
Resultados esperados: El usuario debe poder visualizar en tiempo real el consumo de corriente de la estación.

Tabla D.9. Prueba de aceptación # 10

Caso de prueba de aceptación	
Código: CPA_10	Historia de usuario: 10
Nombre: Mostrar variación de la potencia en gráfico	
Descripción: Prueba de la funcionalidad para visualizar la fluctuación de la potencia en la estación de carga mediante un gráfico.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: 1.Entra en la página del centro de datos 2.Examina la interfaz y localiza el gráfico que representa la "Potencia". 3.Verifica que el gráfico se muestre correctamente, con las etiquetas adecuadas en los ejes. 4.Mantén la página abierta y observa durante unos minutos (recomendable 5 minutos). 5.Asegúrate de que el gráfico no se congele y que la interfaz permanezca activa. 6.Asegúrate de que el gráfico se actualice en tiempo real y representa correctamente los cambios en la potencia. 7.Navega a otra sección de la aplicación y luego vuelve a la vista de la estación de carga sin refrescar la página.	
Resultados esperados: El usuario debe poder visualizar en tiempo real mediante un gráfico la variación de la potencia.	

Tabla D.10. Prueba de aceptación # 11

Caso de prueba de aceptación	
Código: CPA_11	Historia de usuario: 11

Continúa en la próxima página

Tabla D.10. Continuación de la página anterior

Nombre: Mostrar variación del voltaje de salida en gráfico
Descripción: Prueba de la funcionalidad para visualizar la fluctuación del voltaje de salida en la estación de carga mediante un gráfico.
Condiciones de ejecución: El usuario debe estar autenticado.
Pasos de ejecución: <ol style="list-style-type: none"> 1. Entra en la página del centro de datos 2. Examina la interfaz y localiza el gráfico que representa el "Voltaje de salida". 3. Verifica que el gráfico se muestre correctamente, con las etiquetas adecuadas en los ejes. 4. Mantén la página abierta y observa durante unos minutos (recomendable 5 minutos). 5. Asegúrate de que el gráfico no se congele y que la interfaz permanezca activa. 6. Asegúrate de que el gráfico se actualice en tiempo real y representa correctamente los cambios en el voltaje de salida. 7. Navega a otra sección de la aplicación y luego vuelve a la vista de la estación de carga sin refrescar la página.
Resultados esperados: El usuario debe poder visualizar en tiempo real mediante un gráfico la variación del voltaje de salida.

Tabla D.11. Prueba de aceptación # 12

Caso de prueba de aceptación	
Código: CPA_12	Historia de usuario: 12
Nombre: Mostrar variación del recurso consumido en gráfico	
Descripción: Prueba de la funcionalidad para visualizar la fluctuación del recurso consumido en la estación de carga mediante un gráfico.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Entra en la página del centro de datos 2. Examina la interfaz y localiza el gráfico que representa el Recurso consumido". 3. Verifica que el gráfico se muestre correctamente, con las etiquetas adecuadas en los ejes. 4. Mantén la página abierta y observa durante unos minutos (recomendable 5 minutos). 5. Asegúrate de que el gráfico no se congele y que la interfaz permanezca activa. 6. Asegúrate de que el gráfico se actualice en tiempo real y representa correctamente los cambios en el recurso consumido. 7. Navega a otra sección de la aplicación y luego vuelve a la vista de la estación de carga sin refrescar la página. 	
Resultados esperados: El usuario debe poder visualizar en tiempo real mediante un gráfico la variación del recurso consumido.	

Tabla D.12. Prueba de aceptación # 13

Caso de prueba de aceptación	
Código: CPA_13	Historia de usuario: 13
Nombre: Mostrar variación del consumo de corriente en gráfico	
Descripción: Prueba de la funcionalidad para visualizar la fluctuación del consumo de corriente en la estación de carga mediante un gráfico.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Entra en la página del centro de datos 2. Examina la interfaz y localiza el gráfico que representa el Consumo de corriente". 3. Verifica que el gráfico se muestre correctamente, con las etiquetas adecuadas en los ejes. 4. Mantén la página abierta y observa durante unos minutos (recomendable 5 minutos). 5. Asegúrate de que el gráfico no se congele y que la interfaz permanezca activa. 6. Asegúrate de que el gráfico se actualice en tiempo real y representa correctamente los cambios en el recurso consumido. 7. Navega a otra sección de la aplicación y luego vuelve a la vista de la estación de carga sin refrescar la página. 	
Resultados esperados: El usuario debe poder visualizar en tiempo real mediante un gráfico la variación del consumo de corriente.	

Tabla D.13. Prueba de aceptación # 14

Caso de prueba de aceptación	
Código: CPA_14	Historia de usuario: 14
Nombre: Mostrar información del proyecto	
Descripción: Prueba de la funcionalidad para visualizar información sobre el proyecto Tecxol.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Elige en el menú de navegación la página de Información 2. Selecciona en el pie de página el enlace luego de Sobre la estación: 3. Debe dirigir al sitio oficial de la XETID donde habla sobre el proyecto Tecxol 4. Cierra sesión 5. Baja en la pantalla principal Debe aparecer información sobre la importancia del proyecto 	
Resultados esperados: El usuario debe poder visualizar información sobre el proyecto Tecxol, proveniente desde el sitio web.	

Tabla D.14. Prueba de aceptación # 15

Caso de prueba de aceptación	
Código: CPA_15	Historia de usuario: 15
Nombre: Mostrar menú de navegación	
Descripción: Prueba de la funcionalidad del menú de navegación.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: <ol style="list-style-type: none"> 1.Observa la interfaz y ubica el menú de navegación en el lateral izquierdo o en el botón izquierdo del encabezado. 2.Asegúrate de que el menú sea accesible y que contenga las opciones de navegación Centro de Datos e Información. 3.Verifica que la aplicación redirija correctamente a la sección correspondiente. 4.Asegúrate de que la información de esa sección se carga sin errores. 5.Asegúrate de que regreses a la página anterior sin problemas y que la información se mantiene correctamente. 6.Verifica que el menú se ajuste correctamente a diferentes tamaños de pantalla y que todas las opciones de navegación sean accesibles. 7.Asegúrate de que la aplicación muestre la pantalla de inicio de sesión y que no se acceda al menú sin volver a autenticarse. 	
Resultados esperados: El usuario debe poder navegar entre las diferentes secciones en el menú del sitio web	

Tabla D.15. Prueba de aceptación # 16

Caso de prueba de aceptación	
Código: CPA_16	Historia de usuario: 16
Nombre: Mostrar información adicional	
Descripción: Prueba de la funcionalidad de mostrar información de entretenimiento al usuario.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: <ol style="list-style-type: none"> 1.Elige en el menú de navegación la página de Información. 2.Verifica que en el área central se muestre una imagen informativa y los botones de adelante y atrás. 3.Prueba los botones y verifica que realicen la función de adelante y atrás entre las imágenes expuestas 	
Resultados esperados: El usuario debe poder navegar entre las diferentes secciones en el menú del sitio web	

Tabla D.16. Prueba de aceptación # 17

Caso de prueba de aceptación	
Código: CPA_17	Historia de usuario: 17
Nombre: Mostrar aviso de pérdida de conexión	
Descripción: Prueba de la funcionalidad de alertar al usuario en caso de pérdida de la conexión a la wifi de la estación de carga.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Pasos de ejecución: 1.Navega entre las secciones del menú y verifica la conexión a la wifi de la estación. 2.Desconecta el dispositivo desde el que se encuentra conectado. 3.Debe aparecer una alerta que informe que ha perdido la conexión a la wifi de la estación de carga y el botón OK. 4.Presiona el botón OK	
Resultados esperados: El usuario debe poder navegar entre las diferentes secciones en el menú del sitio web	

Lista de Chequeo Usabilidad de Sitios Web

Módulo para monitorear de forma remota estaciones de carga para vehículos eléctricos

Sitio web de monitoreo para Tecxol

Versión 1.0

Introducción

El uso de las listas de chequeos en el Laboratorio de Pruebas de Software ha agilizado las pruebas, teniendo en cuenta que constituyen una guía básica y única para el probador en la revisión de los artefactos. Constituyen un apoyo en la ejecución de evaluaciones estáticas garantizando una mayor calidad en los artefactos de apoyo a los sistemas desarrollados por la Universidad de Ciencias Informáticas (UCI), estableciendo un orden de revisión por subtítulos y acápites que coinciden con la organización del artefacto a evaluar.

Esta lista de chequeo cuenta con varios puntos, los cuales serán clasificados antes de ser aplicados, teniendo en cuenta su pertinencia.

Propósito y objetivos

Esta plantilla ha sido confeccionada para guiar a desarrolladores, especialistas y expertos técnicos en la verificación de los productos desarrollados por la UCI en cuanto de los elementos establecidos en las pruebas de Usabilidad.

Tiene un carácter flexible para el especialista de calidad o el desarrollador teniendo en cuenta que pueden surgir modificaciones e inclusiones.

Alcance

Esta plantilla es aplicable en cada una de las verificaciones, referentes al uso de las pruebas de usabilidad, a los productos de la UCI.

Forma de uso

Evaluación: Es la forma de evaluar el indicador en cuestión. El mismo se evalúa de 1 en caso de mal (cuando la respuesta al indicador sea “No”) y 0 en caso que elemento revisado no presente errores (cuando la respuesta al indicador sea “Sí”).

NP (No Procede): Se usa para especificar que el indicador a evaluar no se puede aplicar en ese caso.

Observación: Especifica los señalamientos o sugerencias que quiera incluir la persona que aplica la lista de chequeo.

Estructura de la lista de chequeo

Elementos definidos por la metodología				
No	Indicador a evaluar	Evaluación	NP	Observación
Visibilidad del sistema				
1	¿La página refleja la identidad de la empresa (logos, compañía...)?	0		
2	¿Cada pantalla empieza con un título que describe su contenido?	0		
3	¿Cuándo se selecciona un icono se diferencia de los no seleccionados?	0		
4	¿Los enlaces del menú se resaltan cuando se seleccionan?	0		
5	¿Los iconos que aparecen se identifican claramente con lo que representan?	0		
6	¿El menú de navegación aparece en un lugar destacado?	0		
7	¿No utiliza más de siete opciones principales en el menú de navegación?	1		Aunque son solo 2 las opciones que presenta el menú, cumple con todos los requisitos funcionales de la aplicación
8	¿Si la respuesta a una acción se retrasa, aparece un mensaje o indicio como que el sistema está procesando	0		

	la acción?			
9	¿El sitio le indica al usuario en que parte de la estructura del sitio web se encuentra, es decir si muestra 'migas de pan'?	0		
10	¿El nombre de los enlaces es el mismo que el título de la página a la que dirige?	0		Únicamente no aplica a la página donde se muestra todos los datos
11	¿El logo de la organización está ubicado en el mismo lugar en todas las páginas, y hacer click en el logo retorna al usuario a la página más lógica (Ejemplo: la página de inicio)?	0		Aunque el logo de la organización si se encuentra ubicado en el mismo lugar en todas las páginas, no realiza la acción de retornar a la página más lógica
12	¿Los títulos de las páginas, tablas e imágenes son descriptivos y distintivos?	0		
13	¿Las etiquetas de las categorías describen con precisión la información de las mismas?	0		
14	¿Cuándo una tarea involucra documentos fuente, la interfaz es compatible con las características del documento fuente?		X	En la aplicación solo se muestra información
15	¿Las imágenes se muestren con buena	0		

	resolución?			
16	¿No se muestran errores ortográficos?	0		
17	¿No hay ninguna imagen con información relevante?	0		Si existen imágenes con información relevante
Lenguaje común entre sistema y usuario				
18	¿El lenguaje es simple, con un tono adecuado?	0		
19	¿La información que se presenta en la aplicación es fácil de entender y memorizar?	0		
20	¿Utiliza los conceptos establecidos para las funciones estándar? ("buscar" para las búsquedas, etc.)	0		
21	¿Evita el lenguaje técnico: términos informáticos o propios de Internet?	0		
22	¿Se utiliza siempre la misma nomenclatura para las mismas funciones?	0		
23	¿Los acrónimos y abreviaturas son definidos al ser usados por primera vez?		X	No se utilizan acrónimos ni abreviaturas
24	¿No hace uso de términos extranjeros?	0		
25	¿Utiliza un texto específico y descriptivo en los vínculos?	0		
26	¿La información es de rápida lectura, y con una disposición asequible?	0		
27	¿Los vínculos basados en nombres de la gente,		X	

	conducen a las biografías cortas o a sus propios blogs, no a un correo electrónico?			
28	¿Si se desea incluir un enlace de correo electrónico, se muestra el correo y no el nombre de la persona?		X	
Libertad y control por parte del usuario				
29	¿Existe una manera lógica de acceder a páginas relacionadas o a otras secciones?	0		
30	¿Tras una acción relevante hay una opción de vuelta atrás?		X	El usuario solo observa información, no realiza acciones o peticiones en la página
31	¿Si una acción tiene consecuencias, el sistema proporciona información y pide confirmación antes de continuar?		X	
32	¿En las páginas internas hay un acceso a la página de inicio en una zona visible y reconocible?		X	
33	¿El Sitio cuenta con un mapa o buscador que facilite el acceso directo a los contenidos?	0		
34	¿Al dar click en el botón "Atrás" siempre lleva al usuario de vuelta a la página de dónde vino?	0		
35	¿El sitio no deshabilita el botón "Atrás" y dicho botón aparece activo en la barra de herramientas	0		

	del navegador en todas las páginas?			
36	¿El sitio evita que los usuarios se registren de manera innecesaria?	0		
37	¿La página se ve con cualquier resolución de pantalla?	0		
38	¿Es posible aumentar y disminuir el tamaño de letra?	1		
39	¿Las imágenes de los productos se pueden ampliar?		X	No se ofrecen productos en la página
40	¿Ofrece el contenido en otros formatos, como dispositivos móviles?	0		
41	¿Es posible imprimir la web sin perder información?	0		
42	¿Se pueden guardar las páginas web?	0		
43	¿El sitio provee una clara retroalimentación cuando una tarea ha sido completada exitosamente?	0		
44	¿Una imagen que sirve como enlace es fácilmente distinguible?		X	
45	¿En caso que se muestre información relacionada con registros obtenidos de la base de datos existe un sistema de navegación donde el usuario pueda especificar cuantos elementos desea ver en la página?	1		

46	¿Los usuarios son informados si es necesario un plug-in del navegador o resolución específico?	0		
47	¿Las páginas que utilizan nuevas tecnologías siguen funcionando cuando dicha tecnología no está presente (por ejemplo, los plug-ins de Flash)?	0		
48	¿Cuándo es necesaria la descarga de un plug-in, hay un enlace a la página donde obtenerlo?		X	
49	¿El motor de búsqueda maneja correctamente (No arroja ningún resultado) las búsquedas vacías (cuando no se introduce nada)?		X	
50	¿Las etiquetas de navegación y links contienen las “palabras clave” que los usuarios necesitan para alcanzar su objetivo?	0		
51	¿Los usuarios pueden ordenar y filtrar los resultados?		X	No es necesario hacer búsquedas
52	¿Los enlaces que invocan acciones (ej. descargas, nuevas ventanas) están claramente distinguidos de los links que cargan otras páginas?		X	No es necesario hacer búsquedas
53	¿La página de resultados de una búsqueda indica claramente cuántos resultados tuvo la		X	No es necesario hacer búsquedas

	búsqueda?			
54	¿La página de resultados de una búsqueda no muestra resultados duplicados (ni duplicados reales ni duplicados muy parecidos)?		X	No es necesario hacer búsquedas
55	¿La caja de búsqueda es suficientemente grande para manejar la longitud de las consultas más comunes?		X	No es necesario hacer búsquedas
56	¿Las búsquedas cubren todo el sitio, no una porción de él?		X	No es necesario hacer búsquedas
57	¿La interfaz de búsqueda está ubicada donde los usuarios esperan encontrarla (en la parte superior derecha de la página)?		X	No es necesario hacer búsquedas
Consistencia y estándares				
58	¿Las imágenes tienen tamaños adecuados que no dificultan el acceso a las páginas?	0		
59	¿Existe un cambio visible cuando el ratón apunta a algo clickeable (excluyendo los cambios de cursor)?	0		
60	¿Todos los botones "clickeables" son efectivamente presionables?	0		
61	¿Los íconos son visualmente y	0		

	conceptualmente distintos pero mantienen una armonía?			
62	¿Para tareas similares, los diálogos, formularios son similares?	0		
63	¿Existe una clara distinción entre campos "requeridos" y "opcionales" en los formularios?	0		Solo se utiliza el formulario para iniciar sesión
64	¿Las preguntas en los formularios están agrupadas de manera lógica y cada grupo tiene un título descriptivo?		X	
65	¿Los campos en los formularios contienen ayudas, ejemplos o modelos de respuestas para demostrar el dato que se debe introducir?	0		
66	¿El nombre de los botones de un formulario es adecuado, aplicado a la acción, no general (Ej.: utilizar "Enviar" en vez de "OK")?	0		
67	¿Las listas de opciones, botones de radio y casillas son preferibles a las cajas de texto en los formularios?	0		
68	¿Se mantiene una navegación consistente y coherente en todas las pantallas?	0		
69	¿La distribución y ubicación de los elementos estructurales que contienen las páginas se mantiene constante a lo largo de la aplicación?	0		

70	¿Se usa la misma fuente para todos los navegadores?	0		
71	¿Puede utilizarse en cualquier navegador?	0		
72	¿Se mantiene una tipografía coherente en todo el sitio web?	0		
73	¿Existen enlaces redundantes? Enlaces redundantes: Son enlaces con rótulos diferentes que llevan a una misma página.	0		
74	¿El link al mapa del sitio aparece en todas las páginas del sitio?		X	
75	¿Añade una descripción en las imágenes?	0		
76	¿Señala claramente los vínculos a archivos PDF como tal?		X	
77	¿Especifica el tamaño de los archivos PDF?		X	
78	¿El sitio tiene una URL correcta, clara y fácil de recordar?	0		
79	¿Tiene un tiempo de respuesta rápida?	0		
80	¿Los vínculos llevan a donde prometen ir?	0		
81	¿El sistema de navegación es amplio y sencillo (muchos ítems en un menú) en vez de un menú profundo?	0		
82	¿Se usan nombres estandarizados ("mapa web", "acerca de...")?	0		
83	¿Se proporciona un texto equivalente para todo	1		

	elemento no textual, tales como imágenes, para explicar su contenido a discapacitados visuales?			
84	¿El documento está estructurado para que pueda ser leído con o sin una hoja de estilo, utilizando adecuadamente los tags de HTML?	0		
85	¿Las presentaciones multimedia, en caso de existir están sincronizadas con sus subtítulos?		X	
86	¿El cambio de idioma en los textos está identificado? El texto que se encuentre en otro idioma debe estar en cursiva o en un formato diferente al del resto del texto.	0		
87	¿El sitio funciona correctamente en un lector de pantalla y/o navegador de voz?	0		
88	¿Es posible la navegación sin ratón?	0		
89	¿Se han creado atajos de teclado?	0		
90	¿Todos los estilos se han creado en hojas CSS?	0		
Estética y diseño minimalista				
91	¿Cumple el sitio con el principio de usabilidad de realizar las operaciones con un máximo de tres click?	0		
92	¿Existe suficiente contraste entre el color del fondo y el del texto?	0		
93	¿Los tipos y tamaños de letra son legibles y distinguibles?	0		

94	¿Añade color de fondo a los div que llevan imagen de fondo? (Para los usuarios que desactivan las imágenes, desaparece el contraste entre texto y fondo, convirtiéndose en texto ilegible.)	0		
95	¿Poseen las páginas animaciones innecesarias?	0		
96	¿El sitio puede ser usado sin desplazamiento horizontal?	0		
97	¿El uso de los colores es moderado?	0		
98	¿Se usan los estilos (negritas, cursivas...) con moderación? Si todo está resaltado con negrita o cursiva, el cerebro se acostumbra y deja de parecerle destacado.	0		
99	¿Utiliza la misma tipografía los mismos aspectos, ejemplo: mantener una tipografía estándar por elementos?	0		
100	¿Utiliza un interlineado adecuado para una buena lectura?	0		
101	¿Se usan frases breves y concisas: que resuman los puntos clave y vayan al grano?	0		
102	¿Los párrafos son cortos?	0		
103	¿Los textos están corregidos?	0		

104	¿Resalta en negrita los conceptos principales de los textos densos?	0		
105	¿Utiliza listas de boliche y numeradas?	0		
106	¿El sitio evita el uso excesivo del texto en mayúsculas?	0		
107	¿El texto dentro del sitio no debe estar justificado?	0		
108	¿No se utiliza el texto subrayado en el cuerpo del texto para resaltar a menos que sea un hipervínculo?	0		
Prevención de errores				
109	¿Existe suficiente espacio entre los elementos de acción (links, botones, etc.) para prevenir que el usuario haga click en el elemento incorrecto?	0		
110	¿Hay ausencia de enlaces rotos o que no lleven a ninguna página?	0		
111	¿Se dan indicaciones para completar campos problemáticos?	0		
112	¿Los botones de acción, (tales como "Enviar") siempre son invocados por el usuario y no automáticamente invocados por el sistema cuando el último campo de un formulario ha sido lleno?	0		
113	¿El espacio entre los campos del formulario es suficiente como para distinguirlos unos de otros?	0		

114	¿Se evita el contenido importante del sitio en ventanas emergentes?	0		
115	¿La zona de acción del vínculo esta ampliada para acertar a la primera?	0		
116	¿El buscador (si existe) permite errores tipográficos y ortográficos (tildes)?		X	
Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores				
117	¿En caso de errores de consistencia dentro del sitio, ¿se ofrece un mensaje de personalizado mediante una página explicativa?, (Por ejemplo: Error 404 para página inexistente)	0		
118	¿El mensaje de error permite volver a la situación anterior?	0		
119	¿Entrega información de contacto con múltiples opciones dentro y fuera de Internet? (Por ejemplo: Formulario, Teléfono, Fax, Correo electrónico, Chat, y hasta Wave, Buzz, Skype..., teléfono institucional, mesa de ayuda)	0		
120	¿Ofrece área de Preguntas Frecuentes con datos de ayuda a usuarios?	0		
121	¿Ofrece páginas de	0		

	ayuda que explican cómo usar el Sitio?			
122	¿Si la ayuda obliga a salir de la zona principal, se proporciona un medio para moverse entre esa ventana y la ayuda?	0		
123	¿La ayuda no interrumpe la tarea del usuario?	0		
124	¿El sitio está diseñado para necesitar el mínimo de ayuda y de instrucciones?	0		
125	¿La ayuda está organizada en pasos?	0		
126	¿Se dan ejemplos para facilitar la tarea?	0		
127	¿Se utilizan explicaciones cortas en la ayuda?	0		
Ayuda y documentación				
128	¿Existe un vínculo a los datos de contacto en un lugar bien visible en todas las páginas web del sitio?	0		
129	¿Información necesaria en página de contacto? (Atención al cliente, Consultas, Soporte técnico, Solicitudes de empleo.)	0		
130	¿Existe una página con las indicaciones sobre la protección de datos en el sitio web?	0		
131	¿Incluye un vínculo a los datos legales en todas las páginas?	0		

132	¿Puede el usuario ponerse en contacto con el encargado del Sitio Web para hacer sugerencias o comentarios?	0		
133	¿Funcionan correctamente los formularios de contacto?	0		
134	¿Hay alguien encargado de recibir y contestar estos mensajes?	0		
135	¿El sitio soporta a los usuarios novatos y expertos brindando diferentes niveles de explicación? (ej. en páginas de ayuda y mensajes de error)	0		
136	¿La política de privacidad del sitio es fácil de encontrar, especialmente esas páginas que piden información personal?	0		
137	¿Cuándo existen múltiples pasos en una tarea, el sitio muestra todos los pasos que deben ser completados y provee una retroalimentación al usuario indicándole la posición actual en toda la ruta de la tarea?	0		
138	¿La funcionalidad de los controles para nuevos dispositivos es exactamente la misma que para los otros dispositivos?	0		
Flexibilidad y eficiencia de uso				

139	Se defina de manera correcta gráficos y tablas utilizando atributos (leyendas, unidades de medida, etc.)	0		
140	¿Las partes o secciones más importantes de los sitios son accesibles desde la página de inicio?	0		
141	¿Las páginas no requieren volver a escribir la información solicitada en páginas anteriores?	0		
142	¿Existen aceleradores, accesos rápidos a operaciones frecuentes?	0		
143	¿El cursor se desplaza adecuadamente en un formulario al presionar "tabulador"?	0		
144	¿Se implementen validaciones antes de que el usuario envíe información?	0		