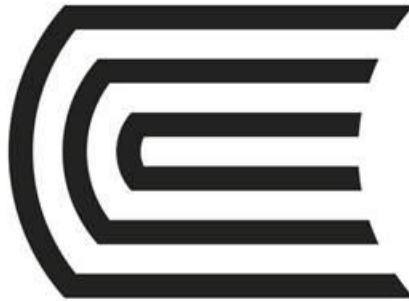


“Año de la recuperación y consolidación de la economía peruana”



**Universidad
Continental**

INFORME DE AVANCE DE PROYECTO

"PLATAFORMA DE CONSUMO ELÉCTRICO"

Docente: Rosario Delia Osorio Contreras

NRC: 62152

Grupo 9

Integrantes:

RICK BERNIE CRISPIN BENDEZU	100%
PAMELA MILAGROS MONTERO PEREZ	100%
JOSE CARLOS CCENTE MEJIA	100%
JOSE ENRIQUE JAMPIER OSORES GONZALES	100%

Índice

Capítulo 1. Presentación del Proyecto.....	5
ODS vinculado.....	5
Organización o institución beneficiaria.....	5
Problema identificado.....	5
Solución propuesta.....	5
Capítulo 2. Análisis de Necesidades y Requerimientos.....	7
Requerimientos Funcionales (RF).....	7
Requerimientos no funcionales (RNF):.....	11
Requerimientos de dominio.....	15
Capítulo 3. Modelos Iniciales del Sistema.....	18
Modelo funcional (diagrama de contexto, casos de uso generales):.....	18
Modelo de procesos.....	19
Diagrama de actividad UML.....	20
Modelo de datos (Modelo E-R).....	21
Capítulo 4. Modelos de Diseño.....	22
Modelo estructural (diagrama de clases inicial):.....	22
Modelo de interacción (diagrama de secuencia):.....	23
Capítulo 5. Metodología de Trabajo (SCRUM).....	24
Definición de la metodología ágil usada.....	24
Backlog del producto (para Jira).....	24
Planificación de Sprints.....	27
Herramientas utilizadas.....	28
Capítulo 6. Diseño de arquitectura y patrones.....	29
Estrategias de diseño del software.....	29
Tipo de arquitectura del sistema.....	30
Patrones de diseño aplicados.....	31
Diseño estructural.....	33
Capítulo 7. Diseño detallado de la base de Datos.....	37

Modelo lógico y físico.....	37
Script en PostgreSQL.....	38
Procedimientos almacenados, vistas y triggers.....	44
Seguridad y respaldo.....	51
Capítulo 8. Diseño detallado de sistemas en red y móviles.....	54
8.1. Modelo de Comunicación.....	54
8.1.1. Comunicación IoT (Dispositivo → Gateway).....	54
8.1.2. Comunicación Gateway → Servidor en la Nube.....	54
8.1.3. Comunicación Cliente–Servidor.....	54
8.2. Diseño del Sistema Web y Móvil.....	55
Aplicación Web.....	55
Aplicación Móvil.....	55
8.3. Gestión de Datos en Red.....	55
8.3.1. Procesamiento en Tiempo Real.....	55
8.3.2. Almacenamiento.....	56
8.3.3. Análisis y Alertas.....	56
8.4. Seguridad en Red y Móviles.....	56
8.4.1. Seguridad de comunicación.....	56
8.4.2. Seguridad en backend.....	56
8.4.3. Seguridad en aplicaciones móviles.....	56
8.4.4. Respaldo y Recuperación.....	56
8.5. Justificación Técnica.....	57
Capítulo 9: Diseño de Interfaz y Experiencia de Usuario (UX/UI).....	58
9.1. Perfil del usuario / usuario meta.....	58
Definición del usuario final según ODS y necesidades reales.....	58
Contexto del uso del sistema.....	58
9.2. Principios de diseño aplicados (HCI) explicar cómo se aplican en el prototipo..	58
Consistencia.....	58
Visibilidad.....	58

Accesibilidad.....	59
Control del usuario.....	59
Retroalimentación.....	59
Simplicidad.....	59
9.3 Diseño del prototipo (baja y alta fidelidad).....	60
Prototipos realizados en Figma.....	60
Capturas de todas las pantallas principales.....	61
Justificaciones del diseño.....	64
9.4. Flujo de navegación del sistema.....	65
Mapa de navegación que representa el recorrido del usuario.....	65
Diagrama de interacción o módulo-navegación.....	66
Capítulo 10. Evaluación del Diseño y Matriz de Trazabilidad.....	68
10.1. Matriz de trazabilidad.....	68
A. Requerimientos Funcionales (RF).....	68
B. Requerimientos No Funcionales (RNF).....	70
10.2. Evaluación del diseño del sistema.....	71
10.2.1. Coherencia entre arquitectura, prototipo y base de datos.....	71
10.2.2. Identificación de mejoras y ajustes.....	71
10.2.3. Evaluación del cumplimiento de principios de usabilidad (HCI).....	72
10.3. Retroalimentación obtenida.....	73
10.3.1. Errores detectados.....	73
10.3.2. Cambios realizados.....	73
10.3.3. Mejoras propuestas (Trabajo Futuro).....	74
10.4. Reflexión sobre el aporte al ODS.....	74
10.4.1. El diseño de la interfaz como herramienta de conciencia.....	74
10.4.2. La navegación orientada a la acción.....	75
10.4.3. Las funcionalidades y la eficiencia energética.....	75
Conclusiones y Recomendaciones.....	77
Conclusión del equipo:.....	77

Lecciones aprendidas:.....	77
Recomendaciones para futuras mejoras del sistema:.....	77
Referencias bibliográficas.....	79
Anexos.....	80

Capítulo 1. Presentación del Proyecto

ODS vinculado

El presente proyecto se articula con el Objetivo de Desarrollo Sostenible N.º 7: Energía asequible y no contaminante, el cual busca garantizar el acceso universal a una energía moderna, segura y sostenible, promoviendo al mismo tiempo la eficiencia energética [1]. La iniciativa se enmarca dentro de este objetivo, al ofrecer una solución que fomente el consumo responsable y consciente de electricidad en los hogares.

Organización o institución beneficiaria

La plataforma está orientada principalmente a familias de entornos urbanos y periurbanos, que suelen enfrentar dificultades para gestionar su consumo eléctrico de manera eficiente. De forma complementaria, pueden beneficiarse instituciones educativas y programas municipales que promuevan la sostenibilidad energética, así como empresas distribuidoras de electricidad interesadas en optimizar la relación con sus usuarios mediante el uso de herramientas tecnológicas [2].

Problema identificado

Actualmente, la mayoría de los hogares cuentan únicamente con la información proporcionada por el recibo mensual de electricidad. Dicho documento se limita a mostrar el monto global del consumo, sin detallar los horarios, dispositivos o patrones que generan mayor gasto [3]. Esta limitación dificulta la identificación de consumos anómalos, eleva los costos económicos para las familias y reduce la conciencia sobre el impacto ambiental derivado del uso excesivo de energía eléctrica.

De acuerdo con la Agencia Internacional de Energía, mejorar la eficiencia energética en el sector residencial puede reducir hasta en un 25 % el gasto eléctrico promedio en países en desarrollo [4]. Sin embargo, la falta de herramientas accesibles y adaptadas al usuario doméstico limita la capacidad de implementar medidas correctivas efectivas.

Solución propuesta

Para responder a este problema, se plantea el desarrollo de una plataforma de seguimiento de consumo eléctrico doméstico. La propuesta combina un sistema de

medición instalado en el hogar con una aplicación digital que organiza y presenta la información de manera clara y útil para el usuario.

La plataforma no se limita a mostrar números, sino que traduce los datos en conocimiento práctico. A través de gráficas, alertas personalizadas y reportes periódicos, el usuario podrá identificar picos de consumo, comparar su gasto entre distintos períodos y recibir sugerencias sobre cómo reducirlo.

El proyecto también contempla un módulo de recomendaciones inteligentes, basado en patrones de uso. Este componente ofrecerá consejos específicos, como reorganizar horarios de uso de electrodomésticos o desconectar equipos en espera, con el fin de reducir el gasto mensual y mejorar la eficiencia energética del hogar.

En conjunto, la solución busca **tres impactos** principales:

Económico, al ayudar a las familias a ahorrar en sus recibos eléctricos, en el nivel social y ambiental, el sistema busca no solo reducir el costo del consumo eléctrico, sino también generar conciencia sobre el uso racional de los recursos energéticos, en coherencia con el ODS 7 y con las políticas de sostenibilidad global [1].

Así, la plataforma no solo se concibe como una herramienta tecnológica, sino también como un medio de sensibilización y educación energética, capaz de impulsar cambios en los hábitos de consumo doméstico de manera sostenida.

Capítulo 2. Análisis de Necesidades y Requerimientos

Requerimientos Funcionales (RF)

Código	Requisito
RF1	Registrar usuarios y hogares.
RF2	Mostrar consumo en tiempo real.
RF3	Generar reportes mensuales.
RF4	Emitir alertas de consumo alto.
RF5	Ofrecer consejos de eficiencia.

Código	Requerimiento	Dependencia	Descripción de la dependencia
RF1.1	Registro de nuevos usuarios.	-	Requerimiento independiente, base para que existan usuarios en el sistema.
RF1.2	Autenticación de usuarios.	RF1.1	Depende de que el usuario esté previamente registrado.

RF1.3	Recuperación de contraseña.	RF1.2	Depende de la autenticación, ya que la recuperación se realiza sobre cuentas activas.
RF1.4	Gestión de cuentas por administrador.	RF1.1	Requiere que existan cuentas registradas para poder administrarlas.
RF2.1	Registro de hogares asociados al usuario.	RF1.1	Depende de que el usuario exista para asociar un hogar.
RF2.2	Asociación de medidor inteligente a cada hogar.	RF2.1	Depende del registro de hogares para poder vincular el medidor.
RF2.3	Recepción de datos en tiempo real desde medidores.	RF2.2	Solo puede ejecutarse si existe un medidor asociado a un hogar.
RF3.1	Visualización del consumo en tiempo real.	RF2.3	Depende de la recepción de datos desde los medidores.
RF3.2	Consulta de consumos diarios, semanales y mensuales.	RF2.3	Requiere almacenamiento de los datos recibidos del medidor.

RF3.3	Visualización de gráficos de evolución del consumo eléctrico.	RF3.2	Depende de la existencia de datos históricos para graficarlos.
RF3.4	Filtros de consumo por hogar o periodo de tiempo.	RF3.2	Depende de que existan datos históricos en la base de datos.
RF4.1	Generación de reportes mensuales en PDF/Excel.	RF3.2	Requiere datos históricos de consumo para elaborar el reporte.
RF4.2	Comparativas con meses anteriores en reportes.	RF4.1	Depende de que ya exista la generación de reportes básicos.
RF4.3	Exportación y descarga de reportes.	RF4.1	Depende de la existencia del reporte en la plataforma.
RF4.4	Reportes globales para administrador/empresa.	RF4.1	Requiere consolidar reportes individuales en reportes globales.

RF5.1	Envío de alertas al superar un umbral de consumo.	RF2.3, RF3.2	Depende de la recepción de datos y su almacenamiento para verificar el umbral.
RF5.2	Configuración de límites de consumo por el usuario.	RF1.1	Depende de la existencia de usuarios registrados para personalizar sus límites.
RF5.3	Notificaciones vía correo y plataforma.	RF5.1	Depende de la existencia de alertas generadas.
RF6.1	Sugerencias de ahorro en base al historial de consumo.	RF3.2	Requiere la existencia de datos históricos.
RF6.2	Comparativas de consumo con promedios de otros hogares.	RF3.2, RF4.4	Depende de datos históricos y de reportes globales.
RF6.3	Consejos prácticos predefinidos.	-	Requerimiento independiente (puede implementarse como módulo estático).
RF7.1	Gestión de catálogos de	RF2.2	Relacionado a la asociación de medidores en los hogares.

	medidores por administrador.		
RF7.2	Consultar estadísticas globales del sistema.	RF4.4	Depende de que existan reportes globales.
RF7.3	Asignación de roles (usuario, admin, entidad reguladora).	RF1.1	Depende de la creación de usuarios para asignarles roles específicos.

Requerimientos no funcionales (RNF):

Código	Requerimiento
RNF1	Seguridad de datos
RNF2	Escalabilidad de la plataforma
RNF3	Interfaz amigable
RNF4	Acceso multiplataforma

Código	Requerimiento	Dependencia	Descripción de la dependencia
RNF1.1	Los datos de usuarios y consumos deben estar encriptados (AES-256, HTTPS).	RF1.1, RF2.3	Depende de la existencia de usuarios y recepción de datos para aplicar encriptación.
RNF1.2	Autenticación segura con hash de contraseñas (bcrypt o similar).	RF1.2	Se aplica al proceso de login de usuarios.
RNF1.3	Inclusión de roles y permisos diferenciados.	RF7.3	Depende de la gestión de roles para aplicarlos con seguridad.
RNF2.1	Procesamiento en tiempo real con retardo máximo de 5 segundos.	RF2.3, RF3.1	Depende de la recepción de datos y su visualización.
RNF2.2	Soportar al menos 10,000 usuarios concurrentes en la primera fase.	RF1.1, RF3.1	Relacionado a la creación de usuarios y acceso al sistema en tiempo real.

RNF2.3	Arquitectura escalable horizontalmente (microservicios/cloud).	RNF2.2	Depende de la necesidad de soportar grandes volúmenes de usuarios.
RNF3.1	Interfaz intuitiva y accesible para usuarios sin conocimientos técnicos.	RF3.1, RF3.3	Depende de la visualización de datos y gráficos de consumo.
RNF3.2	Disponibilidad en versión web y móvil (responsive o app).	RF3.1, RF3.2	Relacionado con la presentación de datos de consumo en diferentes plataformas.
RNF3.3	Tiempo de respuesta promedio < 2 segundos por consulta.	RF3.2, RF3.3	Depende de consultas de consumo y visualización de gráficos.
RNF4.1	Disponibilidad mínima del 99,5% anual.	Todos los RF	Aplica de forma general al sistema completo.
RNF4.2	Copias de seguridad automáticas diarias.	RF2.3, RF3.2, RF4.1	Depende de los datos de consumo y reportes almacenados.

RNF4.3	Recuperación ante fallos en menos de 1 hora.	RNF4.2	Relacionado con el respaldo de datos y continuidad del servicio.
RNF5.1	Integración con distintos modelos de medidores inteligentes estándar.	RF2.2, RF2.3	Depende de la asociación y recepción de datos de los medidores.
RNF5.2	Funcionamiento en navegadores modernos (Chrome, Edge, Firefox, Safari).	RF3.1, RF3.2, RF3.3	Relacionado con la visualización de consumo y reportes desde el front-end.
RNF5.3	Compatibilidad de app móvil con Android e iOS.	RF3.1, RF3.2	Depende de la visualización de datos y reportes en móviles.
RNF6.1	Código documentado y con estándares (Clean Code, SOLID).	Todos los RF	Aplica a toda la lógica del sistema y su desarrollo.
RNF6.2	Pruebas unitarias y de integración.	Todos los RF	Requiere que existan funcionalidades implementadas para validarlas.

RNF6.3	Despliegues continuos (CI/CD).	RNF6.1, RNF6.2	Depende de la calidad del código y de la implementación de pruebas.
--------	--------------------------------	----------------	---------------------------------------------------------------------

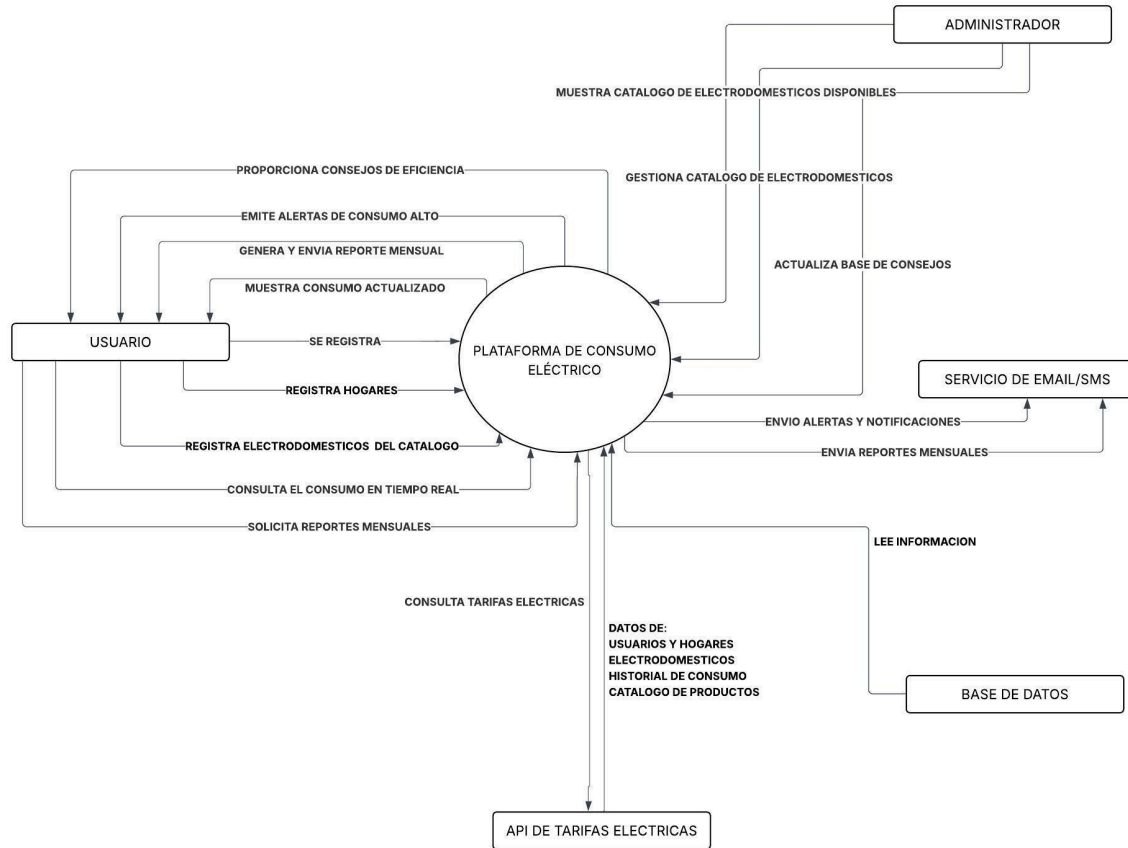
Requerimientos de dominio

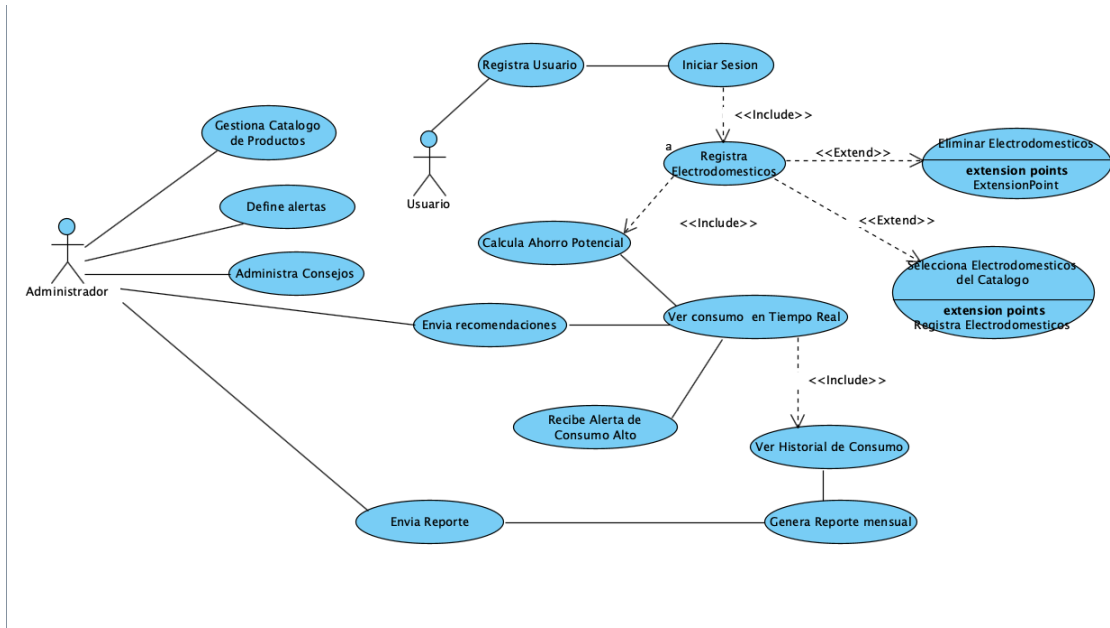
Código	Requerimiento de Dominio	Dependencia (código)	Descripción de la dependencia
RD-01	Cumplimiento de la normativa nacional de eficiencia energética	RNF-05	Se requiere el cumplimiento de estándares de seguridad y normativas oficiales del sector energético.
RD-02	Integración con proveedores de datos energéticos oficiales (empresas eléctricas)	RF-01, RF-02	Depende de la recolección y monitoreo de datos de consumo eléctrico para garantizar información verídica.
RD-03	Considerar tarifas diferenciadas por horarios y regiones	RF-05	El cálculo de tarifas debe ajustarse a las regulaciones vigentes

			y tarifas locales establecidas por las empresas eléctricas.
RD-04	Cumplimiento con políticas de privacidad de datos personales (Ley de Protección de Datos)	RNF-03	Depende del manejo seguro de datos de los usuarios, asegurando confidencialidad y consentimiento informado.
RD-05	Interoperabilidad con sistemas de gestión energética (SGE) y plataformas de smart grid	RF-07	Requiere compatibilidad para integrarse en contextos más amplios de gestión y eficiencia energética.
RD-06	Escalabilidad para adaptarse a cambios en políticas de energías renovables	RNF-01, RNF-04	Necesita que el sistema soporte la incorporación de nuevas métricas o regulaciones sin rediseño completo.
RD-07	Inclusión de métricas alineadas con ODS 7 y reportes ambientales	RF-08	Los reportes deben estar alineados con indicadores internacionales y estándares de sostenibilidad.

Capítulo 3. Modelos Iniciales del Sistema

Modelo funcional (diagrama de contexto, casos de uso generales):





Modelo de procesos

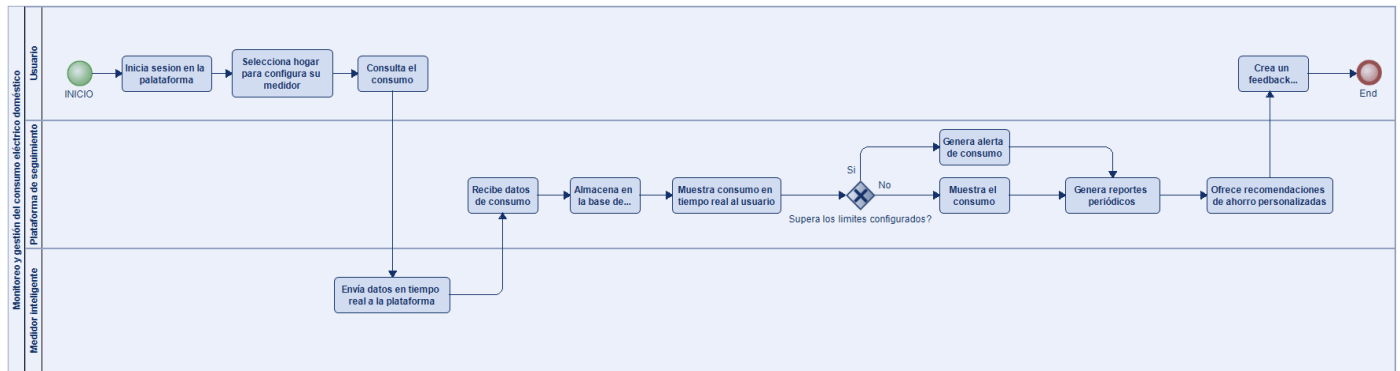
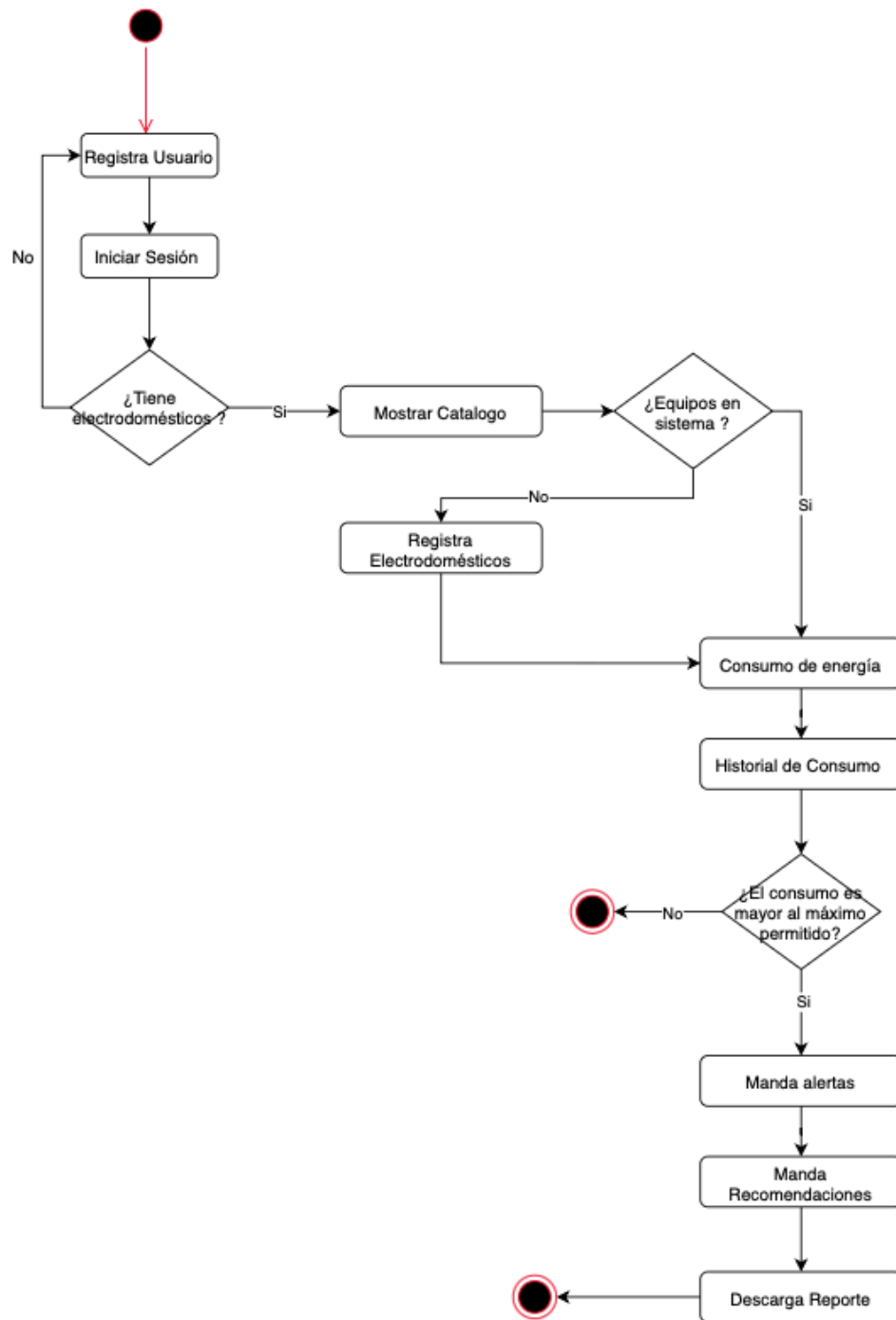
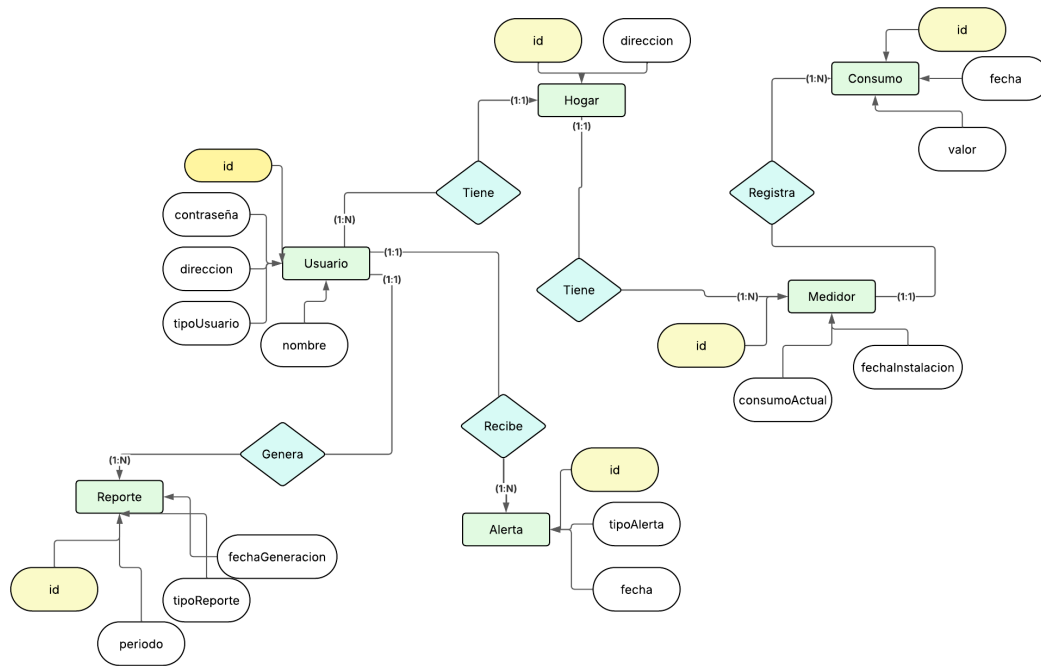


Diagrama de actividad UML

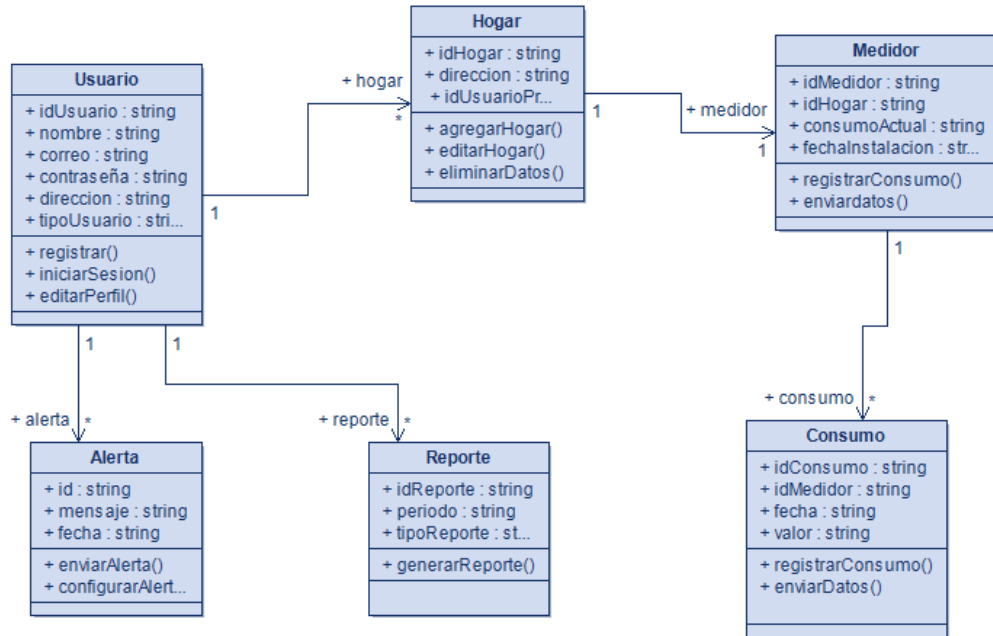


Modelo de datos (Modelo E-R)

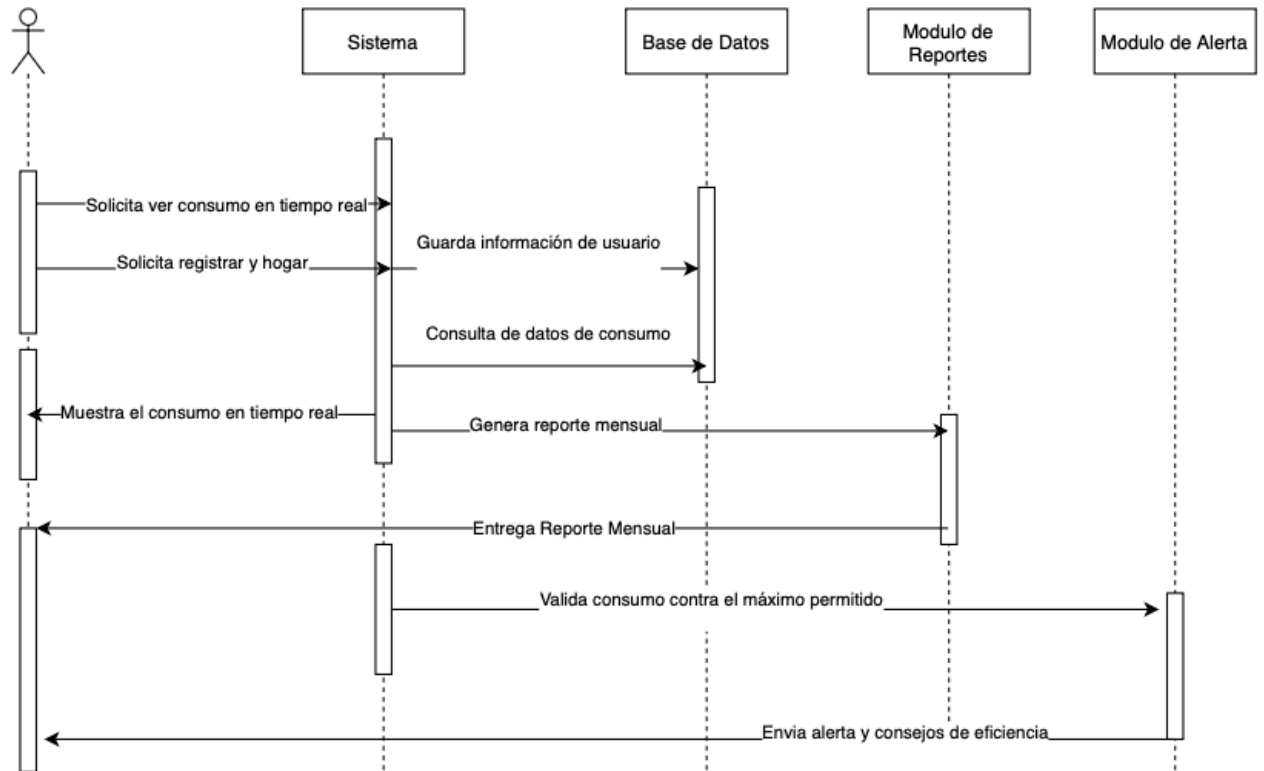


Capítulo 4. Modelos de Diseño

Modelo estructural (diagrama de clases inicial):



Modelo de interacción (diagrama de secuencia):



Capítulo 5. Metodología de Trabajo (SCRUM)

Definición de la metodología ágil usada

Se emplea SCRUM como metodología ágil para el desarrollo de la plataforma de seguimiento de consumo eléctrico doméstico. Esta metodología permite:

- Desarrollar el sistema de manera iterativa e incremental mediante sprints.
- Ajustar los requisitos a necesidades cambiantes de los usuarios y normativas energéticas.
- Garantizar entregas tempranas de funcionalidades completas.

SCRUM se caracteriza por roles definidos (Product Owner, Scrum Master, Development Team) y ceremonias estructuradas (Sprint Planning, Daily Standup, Sprint Review, Sprint Retrospective) que facilitan la entrega continua de valor al usuario [5].

Backlog del producto (para Jira)

EPIC	Código HU	Historia de Usuario / Resumen	Descripción	Prioridad	Estimación
Gestión de Usuarios	HU01	Registro de usuarios	Registro de usuarios proporcionando nombre, correo, contraseña y dirección para acceder al sistema.	Alta	3
Gestión de Usuarios	HU02	Autenticación de usuarios	Iniciar sesión de forma segura mediante correo y contraseña,	Alta	3

			protegiendo datos del usuario.		
Gestión de Hogares y Medidores	HU03	Registro de hogares	Registrar hogares y asociarlos a medidores inteligentes para monitoreo de consumo.	Alta	3
Gestión de Hogares y Medidores	HU04	Asociación de medidores inteligentes	Asociar medidores a hogares y recibir datos de consumo en tiempo real.	Alta	5
Monitoreo de Consumo	HU05	Visualización en tiempo real	Panel con consumo instantáneo y comparación con límites configurables.	Alta	5
Monitoreo de Consumo	HU06	Filtros de consumo	Filtrar consumo por hogar y periodo de tiempo seleccionado.	Media	3

Monitoreo de Consumo	HU07	Consulta de historial completo	Consultar consumos diarios, semanales y mensuales mediante gráficos históricos.	Media	5
Reportes	HU08	Generación de reportes mensuales	Generar reportes en PDF/Excel incluyendo comparativas de meses anteriores.	Media	3
Reportes	HU09	Reportes consolidados	Generar reportes globales por hogar, zona o periodo para análisis administrativo.	Media	5
Alertas	HU10	Alertas de consumo	Notificación al usuario cuando el consumo supera un límite predefinido.	Alta	3
Alertas	HU11	Configuración de límites	Definir límites de consumo por hogar para personalizar alertas.	Media	3

Recomendaciones	HU12	Consejos de eficiencia energética	Sugerencias basadas en historial de consumo y comparativas con promedios de otros usuarios.	Media	5
Administración	HU13	Consultar estadísticas globales	Visualización de reportes consolidados y exportación de datos para análisis de eficiencia.	Media	5
Administración	HU14	Gestión de roles y permisos	Asignar roles (usuario, admin, regulador) para controlar accesos y funcionalidades.	Alta	3

Planificación de Sprints

Sprint 1 (2 semanas):

- Implementar funcionalidades básicas: registro y autenticación de usuarios, registro de hogares, asociación de medidores, visualización en tiempo real y filtros básicos de consumo.
- HU asignadas: HU01, HU02, HU03, HU04, HU05, HU06

Sprint 2 (2 semanas):

- Implementar funcionalidades avanzadas: historial de consumo, reportes, alertas, recomendaciones, administración y gestión de roles.

- HU asignadas: HU07, HU08, HU09, HU10, HU11, HU12, HU13, HU14

Herramientas utilizadas

Herramienta	Uso principal
Jira	Gestión de backlog, planificación de sprints, seguimiento de historias y tareas.
Draw.io / Dragrams	Creación de diagramas UML (casos de uso, actividad, secuencia, procesos).
Visual Paradigm	Diseño de modelo de datos (E-R) y diagramas de clases.
GitHub	Control de versiones y colaboración en desarrollo.

Capítulo 6. Diseño de arquitectura y patrones

Estrategias de diseño del software

Modularidad y separación de responsabilidades

- Dividir el sistema en módulos/servicios con responsabilidad única (ingestión de datos, almacenamiento, procesamiento/analítica, recomendaciones, notificaciones, UI, autenticación).
- Favorecer interfaces bien definidas (APIs REST/gRPC / mensajería) para desacoplar.

Diseñar pensando en la escalabilidad

- Partir con una arquitectura que permita escalar horizontalmente (stateless services, colas/brokers para picos de mensajes).
- Uso de base de datos especializada: *time-series DB* para lecturas de consumo (alta velocidad de inserción), RDBMS para usuarios/configuración, almacén de datos para reports.

Resiliencia y tolerancia a fallas

- Retries exponenciales, circuit breakers, timeouts.
- Que el sistema siga aceptando/guardando datos en el gateway local si la conexión a nube cae (buffer en gateway/edge).

Diseño orientado a eventos donde tenga sentido

- Ingesta → events → stream processing → recomendaciones/alerts.
- Forma natural para picos, analítica en tiempo real y mayor desacople.

Seguridad y privacidad por diseño

- Autenticación y autorización (OAuth2 + JWT), RBAC.
- Encriptación in transit (TLS) y at rest para datos sensibles.
- Minimizar datos personales enviados; anonimizar para análisis agregados.

Observabilidad

- Logs estructurados, métricas (Prometheus), tracing distribuido (OpenTelemetry), alertas SLO/SLIs.

UX centrado en el usuario

- Interfaz clara, prioridades de información: consumo por hora, picos, sugerencias accionables.
- Notificaciones configurables (umbral, horario, tipo).

Testing & Calidad

- Tests unitarios, integración, E2E (automatizar pipelines CI/CD).
- Pruebas de carga en componentes de ingestión/streaming.

Extensibilidad

- Plugins para nuevos tipos de sensores o algoritmos de recomendación.
- Versionado de API.

Tipo de arquitectura del sistema

Arquitectura propuesta: Microservicios orientados a eventos (Event-driven microservices) con API Gateway.

Motivos:

- Necesitas escalabilidad independiente (ingesta masiva de datos vs UI vs ML).
- Procesamiento en tiempo real (detección de picos, alertas) se beneficia de streaming/event-driven.
- Permite desplegar y evolucionar módulos (ML, recomendaciones) sin afectar otros.
- Facilita integración con dispositivos IoT (MQTT/bridge → broker → ingest service).

Elementos principales de la arquitectura

- Edge / Gateway local: dispositivo o hub que lee sensores (CT clamps, smart plugs) y publica por MQTT/HTTPS. Buffer local si la conexión falla.
- Broker de Mensajería: MQTT para telemetría IoT; Kafka o RabbitMQ para stream y procesamiento en backend.
- Ingest Service / Collector: consume mensajes del broker y valida/normaliza, persiste en Time-Series DB y publica eventos para downstream.
- Time-Series DB: InfluxDB, TimescaleDB (Postgres extension) o Prometheus (si fuera apropiado) para series temporales.
- Service de Procesamiento / Stream Processor: Apache Kafka Streams, Flink o un worker que calcule agregados, detección de picos, anomalías.
- Motor de Recomendaciones: servicio que recibe features/series y genera recomendaciones (reglas + ML). Entrenamiento off-line + predicción en línea.
- API Gateway: punto único para clientes, enrutamiento, rate-limiting, caching.
- Auth Service: OAuth2 / OpenID Connect, JWT.
- User / Config Service: RDBMS (Postgres) para usuarios, casas, dispositivos, preferencias.
- Notification Service: push (FCM/APNs), email, SMS.
- Dashboard / Mobile WebApp: SPA (React) + App (React Native).
- Admin & Reporting Service: generación de reportes periódicos, exportes (PDF/CSV).
- Observability stack: Prometheus, Grafana, ELK/Opensearch, Jaeger.

Patrones de diseño aplicados

Arquitectónicos

- Event-Driven Architecture: para ingestión y procesamiento.

- Microservices: separación según bounded contexts (telemetry, auth, user, recommendations).
- API Gateway: patrón de fachada para clientes.

Integración

- Adapter / Anti-Corruption Layer: adaptar formatos heterogéneos de sensores al modelo interno.
- Message Broker (Publish-Subscribe): desacoplamiento entre productores y consumidores.

Diseño backend

- Repository Pattern: encapsular acceso a bases de datos (TS DB, RDBMS).
- Factory Method / Abstract Factory: para crear handlers de dispositivos diferentes (CT clamp, smart plug).
- Strategy Pattern: seleccionar algoritmo de detección de anomalías o recomendación según perfil.
- Observer / Publish-Subscribe (In-app): para notificaciones internas cuando se detecta un pico.
- Circuit Breaker: proteger servicios dependientes en caso de fallo.
- CQRS (Command Query Responsibility Segregation): separar lectura intensiva (dashboards, queries) de escritura (ingest).
- Event Sourcing (opcional): para historial completo de eventos (útil si quieres reconstruir estados o auditoría). Úsalo si la trazabilidad es crítica.
- Decorator: para añadir capas cross-cutting (caching, logging) sobre handlers.

Frontend

- Flux/Redux Architecture: para manejo de estado en SPA/mobile.
- Facade Pattern: para simplificar distintos endpoints a la UI.

Diseño estructural

A continuación incluyo un diagrama de componentes



Descripción de los componentes:

Componente	Descripción
------------	-------------

Sensor (CT / Smart Plug)	Dispositivo físico que mide el consumo eléctrico en tiempo real.
Gateway del Hogar	Equipo intermedio (por ejemplo, Raspberry Pi o ESP32) que recibe los datos de los sensores, los almacena temporalmente y los envía al servidor mediante MQTT o HTTP.
Broker MQTT	Servidor que recibe los mensajes de telemetría enviados por los dispositivos IoT.
Message Broker (Kafka/Rabbit MQ)	Sistema de mensajería que distribuye los eventos entre microservicios del backend.
Servicio de Ingesta	Microservicio que recibe, valida y normaliza los datos provenientes de los dispositivos. Los guarda y genera eventos para otros servicios.
Procesador de Flujo	Analiza los datos en tiempo real, calcula consumos promedio, identifica picos y detecta anomalías.
Base de Datos de Series Temporales	Almacena las lecturas de consumo con marca temporal para consultas históricas y estadísticas.

Almacén de Eventos (opcional)	Guarda cada evento para auditoría o reconstrucción del historial.
Motor de Recomendaciones	Genera sugerencias inteligentes basadas en patrones de consumo y modelos de Machine Learning.
Servicio de Autenticación	Gestiona el inicio de sesión, tokens y roles de usuario mediante OAuth2/JWT.
API Gateway	Punto de entrada único para las solicitudes de las aplicaciones web/móviles; maneja autenticación, rate limiting y enrutamiento.
Servicio de Usuarios y Configuración	Administra la información de usuarios, viviendas, dispositivos y preferencias.
Servicio de Notificaciones	Envía alertas al usuario (push, correo o SMS) sobre consumos altos o recomendaciones.
Servicio de Reportes	Genera reportes mensuales o personalizados en formatos como PDF o CSV.
Panel Administrativo	Interfaz para la gestión del sistema, usuarios, métricas y monitoreo.

Aplicación Web/Móvil	Interfaz para el usuario final, donde visualiza el consumo, recibe alertas y accede a las recomendaciones.
Analítica y Entrenamiento de ML	Procesos por lotes que entrenan modelos predictivos a partir de datos históricos.
Monitoreo	Conjunto de herramientas para las métricas, logs y trazas de los servicios.

Capítulo 7. Diseño detallado de la base de Datos

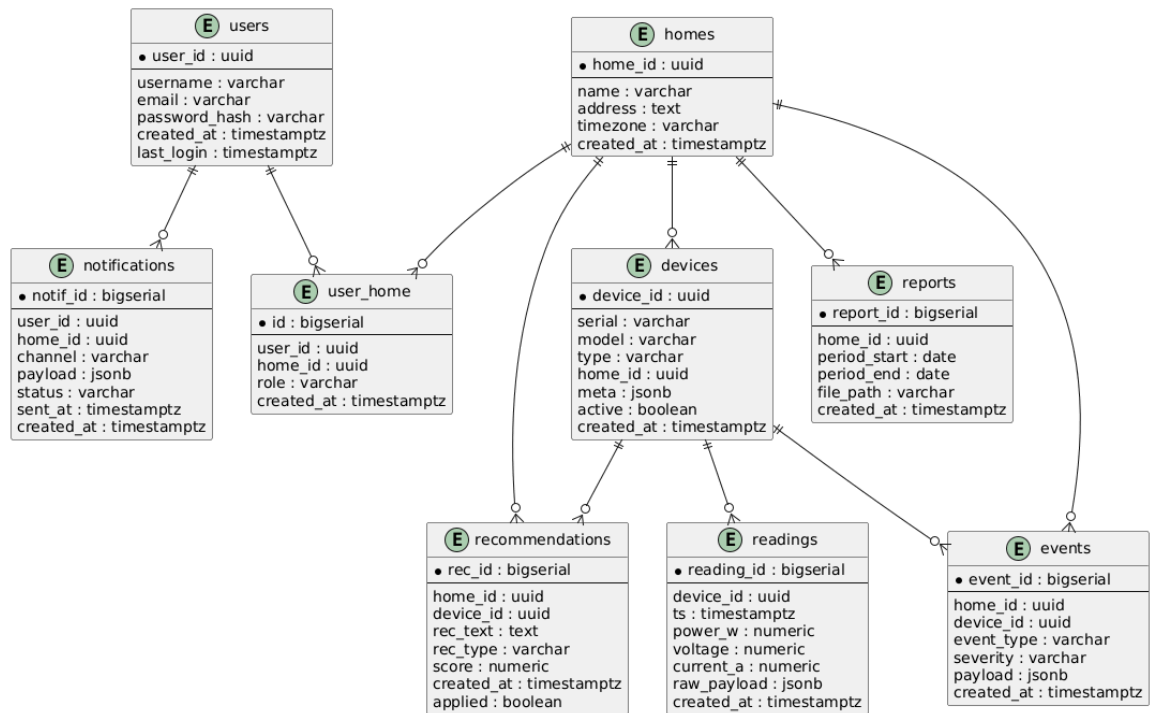
Modelo lógico y físico

Entidades clave:

- users: usuarios del sistema.
- homes: viviendas / hogares (una casa puede tener varios dispositivos y varios usuarios asociados).
- devices: sensores/medidores (CT clamps, smart plugs).
- readings: lecturas temporales (time-series).
- events: eventos publicados (spikes, anomalías).
- recommendations: recomendaciones generadas.
- notifications: alertas / envíos a usuarios.
- reports: metadata de reportes generados.
- user_home: relación usuarios <-> hogares (roles).
- device_mappings: mapeo dispositivo → hogar (opcional).

Diagrama ER:

ER Diagram - Plataforma Consumo Eléctrico



Script en PostgreSQL

-- 0. Habilitar extensiones (ejecutar como superuser)

```
CREATE EXTENSION IF NOT EXISTS "uuid-oss";
```

```
CREATE EXTENSION IF NOT EXISTS timescaledb;
```

-- 1. Esquema base

```
CREATE SCHEMA IF NOT EXISTS energy;
```

```
SET search_path = energy, public;
```

-- 2. Tabla users

```
CREATE TABLE IF NOT EXISTS users (
```

```
user_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),

username VARCHAR(100) UNIQUE NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

password_hash VARCHAR(255) NOT NULL,

full_name VARCHAR(255),

created_at TIMESTAMPTZ NOT NULL DEFAULT now(),

last_login TIMESTAMPTZ

);
```

-- 3. Tabla homes

```
CREATE TABLE IF NOT EXISTS homes (

home_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),

name VARCHAR(150) NOT NULL,

address TEXT,

timezone VARCHAR(64) DEFAULT 'America/Lima',

created_at TIMESTAMPTZ NOT NULL DEFAULT now()

);
```

-- 4. Relación users <-> homes

```
CREATE TABLE IF NOT EXISTS user_home (

id BIGSERIAL PRIMARY KEY,

user_id UUID NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
```



```

home_id UUID NOT NULL REFERENCES homes(home_id) ON DELETE CASCADE,

role VARCHAR(32) NOT NULL DEFAULT 'owner', -- owner, member, viewer

created_at TIMESTAMPTZ NOT NULL DEFAULT now(),

UNIQUE(user_id, home_id)

);

```

-- 5. devices

```

CREATE TABLE IF NOT EXISTS devices (

device_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),

serial VARCHAR(150) UNIQUE,

model VARCHAR(100),

type VARCHAR(50), -- ct_clamp, smart_plug, etc

home_id UUID REFERENCES homes(home_id) ON DELETE SET NULL,

meta JSONB,

active BOOLEAN DEFAULT true,

created_at TIMESTAMPTZ NOT NULL DEFAULT now()

);

```

-- 6. readings (time-series) -> hypertable

```

CREATE TABLE IF NOT EXISTS readings (

reading_id BIGSERIAL NOT NULL,

device_id UUID NOT NULL REFERENCES devices(device_id) ON DELETE
CASCADE,

```

```

ts TIMESTAMPTZ NOT NULL,

power_w NUMERIC(12,3) NULL,

voltage NUMERIC(10,3) NULL,

current_a NUMERIC(10,3) NULL,

raw_payload JSONB,

created_at TIMESTAMPTZ NOT NULL DEFAULT now(),

PRIMARY KEY (reading_id)
);

SELECT create_hypertable('energy.readings', 'ts', chunk_time_interval => INTERVAL '1
day', if_not_exists => TRUE);

-- Indexes para lectura

CREATE INDEX IF NOT EXISTS idx_readings_device_ts ON readings(device_id, ts
DESC);

CREATE INDEX IF NOT EXISTS idx_readings_ts ON readings(ts DESC);

-- 7. events

CREATE TABLE IF NOT EXISTS events (

    event_id BIGSERIAL PRIMARY KEY,

    home_id UUID REFERENCES homes(home_id) ON DELETE CASCADE,

    device_id UUID REFERENCES devices(device_id) ON DELETE SET NULL,

    event_type VARCHAR(100) NOT NULL,

```

```
severity VARCHAR(20) DEFAULT 'info',

payload JSONB,

created_at TIMESTAMPTZ NOT NULL DEFAULT now()

);


-- 8. recommendations

CREATE TABLE IF NOT EXISTS recommendations (

    rec_id BIGSERIAL PRIMARY KEY,

    home_id UUID REFERENCES homes(home_id) ON DELETE CASCADE,

    device_id UUID REFERENCES devices(device_id) ON DELETE SET NULL,

    rec_text TEXT NOT NULL,

    rec_type VARCHAR(64),

    score NUMERIC(5,3),

    created_at TIMESTAMPTZ NOT NULL DEFAULT now(),

    applied BOOLEAN DEFAULT false

);


-- 9. notifications

CREATE TABLE IF NOT EXISTS notifications (

    notif_id BIGSERIAL PRIMARY KEY,

    user_id UUID REFERENCES users(user_id) ON DELETE CASCADE,

    home_id UUID REFERENCES homes(home_id) ON DELETE CASCADE,
```

```

channel VARCHAR(32) NOT NULL, -- push, email, sms

payload JSONB,

status VARCHAR(32) DEFAULT 'pending', -- pending, sent, failed

sent_at TIMESTAMPTZ,

created_at TIMESTAMPTZ NOT NULL DEFAULT now()

);

-- 10. reports metadata

CREATE TABLE IF NOT EXISTS reports (

    report_id BIGSERIAL PRIMARY KEY,

    home_id UUID REFERENCES homes(home_id) ON DELETE CASCADE,

    period_start DATE NOT NULL,

    period_end DATE NOT NULL,

    file_path VARCHAR(1024),

    created_at TIMESTAMPTZ NOT NULL DEFAULT now()

);

-- 11. Materialized table for daily aggregates (optional)

CREATE TABLE IF NOT EXISTS daily_consumption (

    id BIGSERIAL PRIMARY KEY,

    home_id UUID NOT NULL,

    date DATE NOT NULL,

```

```

total_energy_wh NUMERIC(18,3) NOT NULL DEFAULT 0,

created_at TIMESTAMPTZ NOT NULL DEFAULT now(),

UNIQUE(home_id, date)

);

-- 12. Roles de ejemplo (crear si no existen)

-- Nota: Recomendado ejecutar como superuser / admin DB

DO $$

BEGIN

    IF NOT EXISTS (SELECT 1 FROM pg_roles WHERE rolname = 'app_write') THEN

        CREATE ROLE app_write LOGIN PASSWORD 'change_me' NOINHERIT;

    END IF;

    IF NOT EXISTS (SELECT 1 FROM pg_roles WHERE rolname = 'app_readonly') THEN

        CREATE ROLE app_readonly LOGIN PASSWORD 'change_me' NOINHERIT;

    END IF;

END$$;

```

Procedimientos almacenados, vistas y triggers

Procesos almacenados

Procedimiento para ingesta atómica:

```

CREATE OR REPLACE FUNCTION energy.sp_insert_reading(

    p_device_id UUID,

```

```

p_ts TIMESTAMPTZ,

p_power_w NUMERIC,

p_voltage NUMERIC,

p_current_a NUMERIC,

p_raw JSONB

) RETURNS BIGINT

LANGUAGE plpgsql

AS $$

DECLARE

    v_reading_id BIGINT;

BEGIN

    -- Validaciones básicas

    IF p_device_id IS NULL OR p_ts IS NULL THEN

        RAISE EXCEPTION 'device_id and ts are required';

    END IF;

    INSERT INTO energy.readings(device_id, ts, power_w, voltage, current_a,
raw_payload)

    VALUES (p_device_id, p_ts, p_power_w, p_voltage, p_current_a, p_raw)

    RETURNING reading_id INTO v_reading_id;

    -- Opcional: crear evento si power supera umbral (ejemplo simple)

    IF p_power_w IS NOT NULL AND p_power_w > 5000 THEN -- umbral demo en W

```

```

INSERT INTO energy.events(home_id, device_id, event_type, severity, payload)

VALUES (

    (SELECT home_id FROM energy.devices WHERE device_id = p_device_id),

    p_device_id,

    'high_power',

    'warning',

    jsonb_build_object('power_w', p_power_w, 'ts', p_ts)

);

END IF;

RETURN v_reading_id;

END;

$$;

```

Proceso para crear metadatos y devuelve path:

```

CREATE OR REPLACE FUNCTION energy.sp_generate_monthly_report(

    p_home_id UUID,

    p_period_start DATE,

    p_period_end DATE,

    OUT p_report_id BIGINT,

    OUT p_file_path VARCHAR

) RETURNS RECORD

```

```
LANGUAGE plpgsql

AS $$

DECLARE

    v_report_id BIGINT;

    v_path TEXT;

BEGIN

    -- Inserta metadata, la generación del archivo se delega a la app (worker) que leerá
    datos

    INSERT INTO energy.reports(home_id, period_start, period_end, created_at)

    VALUES (p_home_id, p_period_start, p_period_end, now())

    RETURNING report_id INTO v_report_id;

    -- Sugerir path (convención)

    v_path := format('/reports/%s/report_%s_%s.pdf', p_home_id::text, p_period_start::text,
    p_period_end::text);

    UPDATE energy.reports SET file_path = v_path WHERE report_id = v_report_id;

    p_report_id := v_report_id;

    p_file_path := v_path;

END;

$$;
```


Vistas

Consumo agregado por día y hogar

```
CREATE OR REPLACE VIEW energy.view_daily_consumption AS

SELECT

    d.home_id,

    date_trunc('day', r.ts)::date AS day,

    SUM( COALESCE(r.power_w,0) * (

        COALESCE(

            extract(epoch from lead(r.ts) OVER (PARTITION BY r.device_id ORDER BY r.ts) -

r.ts),

            0

        ) / 3600.0

    )) AS total_wh

FROM energy.readings r

JOIN energy.devices d ON r.device_id = d.device_id

GROUP BY d.home_id, day

ORDER BY d.home_id, day;
```

Top 10 dispositivos por consumo en un rango

```
CREATE OR REPLACE VIEW energy.view_top_consumers AS

SELECT

    d.device_id,

    d.home_id,
```

```

d.model,

SUM(COALESCE(r.power_w,0) * COALESCE(extract(epoch from lead(r.ts) OVER
(PARTITION BY r.device_id ORDER BY r.ts) - r.ts),0) / 3600.0) AS total_wh

FROM energy.readings r

JOIN energy.devices d ON r.device_id = d.device_id

GROUP BY d.device_id, d.home_id, d.model

ORDER BY total_wh DESC

LIMIT 10;

```

Triggers para detectar picos y crear notificación / eventos:

```

-- Función que inserta evento y notificación en caso de pico

CREATE OR REPLACE FUNCTION energy.trg_after_insert_readings()

RETURNS trigger

LANGUAGE plpgsql

AS $$

DECLARE

    v_home_id UUID;

    v_user UUID;

BEGIN

    -- Obtener home_id del device

    SELECT home_id INTO v_home_id FROM energy.devices WHERE device_id =
NEW.device_id;

```

```

-- Ejemplo de regla simple: si power_w > umbral (se puede leer de user prefs)

IF NEW.power_w IS NOT NULL AND NEW.power_w > 5000 THEN

    INSERT INTO energy.events(home_id, device_id, event_type, severity, payload,
created_at)

        VALUES      (v_home_id,    NEW.device_id,    'spike_power',    'high',
jsonb_build_object('power_w', NEW.power_w, 'ts', NEW.ts), now());

-- Crear notificación para todos los usuarios del hogar (simplificado)

    FOR v_user IN SELECT user_id FROM energy.user_home WHERE home_id =
v_home_id

    LOOP

        INSERT INTO energy.notifications(user_id, home_id, channel, payload, status,
created_at)

            VALUES (v_user, v_home_id, 'push', jsonb_build_object('title','Pico de
consumo','body', format('Se detectó %s W en el dispositivo %s', NEW.power_w,
NEW.device_id)), 'pending', now());

    END LOOP;

END IF;

RETURN NEW;

END;

$$;

-- Asociar trigger AFTER INSERT

```

```
CREATE TRIGGER trg_readings_after_insert
AFTER INSERT ON energy.readings
FOR EACH ROW
EXECUTE FUNCTION energy.trg_after_insert_readings();
```

Seguridad y respaldo

Seguridad a nivel de base de datos

Principio de menor privilegio

- Crear roles separados:
 - app_write — permisos INSERT/UPDATE/SELECT en tablas necesarias (usado por ingest workers).
 - app_readonly — permisos SELECT para dashboards.
 - db_admin — para migraciones y administración.
- No usar el superuser desde la app.

Conexiones y credenciales

- Usar conexiones TLS (sslmode=require).
- Guardar credenciales en vault (HashiCorp Vault, AWS Secrets Manager) o en secretos Kubernetes.
- Rotación periódica de claves/secretos.

Cifrado

- cifrado at-rest (discos cifrados LUKS / cifrado en el proveedor cloud).
- cifrado in-transit (TLS).
- cifrado de columnas sensibles (ej. datos personales) con pgcrypto si necesario.

Auditoría

- Habilitar audit logs (pgaudit) para acciones de DBA y accesos sospechosos.
- Registrar accesos y cambios en tablas críticas.

Validación y límites

- Validar datos en la capa de DB (CHECK constraints) y en la API.
- Límites por usuario/home en la app para evitar ingest maliciosa.

Backup / Recuperación

Backups completos y WAL archiving

- Ejemplo: pg_basebackup para backups base diarios + archive_mode = on y archive_command para guardar WALs en S3/objstore.
- Política: backups completos diarios / incrementales (o base semanal + WALs continuos) dependiendo RPO/RTO.

Almacenamiento

- Mantener backups en replicación geo (S3 buckets en distinta región).
- Versionado y encriptación de backups (server-side encryption).

Retención

- Retención mínima 30 días (ajustar según normativas).
- Retención a largo plazo (1 año) para auditoría si procede.

Pruebas de restauración

- Realizar restauraciones periódicas (mensual) en entorno staging para validar backups.

Alta disponibilidad

- Replica en caliente (streaming replication) + failover automático (Patroni, repmgr).

- Para Timescale, considerar replicas y configuración para cargas de lectura.

Automatización

- Jobs automáticos (cron/Kubernetes CronJob) para backup + upload a S3 + verificación checksum + notificación.

Capítulo 8. Diseño detallado de sistemas en red y móviles

8.1. Modelo de Comunicación

El sistema de monitoreo de consumo eléctrico implementa un modelo de comunicación distribuido basado en tres capas principales: dispositivos IoT, gateway doméstico y servicios en la nube. Esta arquitectura permite el envío de datos en tiempo real, actualizaciones constantes y la generación inmediata de alertas.

8.1.1. Comunicación IoT (Dispositivo → Gateway)

Los sensores o medidores inteligentes instalados en cada hogar recopilan información de consumo eléctrico y la transmiten al gateway mediante:

- MQTT sobre TLS: protocolo ligero ideal para telemetría en tiempo real.
- Conexiones persistentes con reintentos automáticos en caso de pérdida de señal.
- Mensajes tipo publish/subscribe optimizados para consumo energético bajo.

8.1.2. Comunicación Gateway → Servidor en la Nube

El gateway agrupa, valida y transmite los datos hacia la nube mediante:

- HTTPS (TLS 1.3) para garantizar confidencialidad.
- Validación de paquetes y almacenamiento temporal (buffer) si no hay conexión.
- Envío periódico o por eventos (picos de consumo).

8.1.3. Comunicación Cliente–Servidor

Los usuarios acceden al sistema mediante la aplicación web o móvil, la cual se comunica con el backend a través de:

- API REST para operaciones generales (consultas, configuración, reportes).
- WebSockets (WSS) para consumo en tiempo real.
- Autenticación mediante JWT, distribuida por el API Gateway.

8.2. Diseño del Sistema Web y Móvil

El sistema cuenta con dos interfaces principales: la aplicación web y la aplicación móvil, ambas orientadas a facilitar la interpretación del consumo eléctrico del hogar.

Aplicación Web

- Desarrollada bajo el enfoque SPA (Single Page Application).
- Vista principal: panel con consumo en tiempo real, gráficos diarios y comparativas.
- Funciones destacadas:
 - Gestión de hogares y dispositivos.
 - Configuración de límites de consumo.
 - Visualización de reportes en PDF/Excel.
 - Acceso a recomendaciones inteligentes.

Aplicación Móvil

- Construida en React Native / Flutter para su compatibilidad con Android e iOS.
- Incluye funciones de:
 - Notificaciones push de alertas de consumo.
 - Panel simplificado de consumo.
 - Configuración rápida de límites y alarmas.
- Uso optimizado de recursos de red y batería.

8.3. Gestión de Datos en Red

Debido al gran volumen de lecturas generadas por los medidores, el sistema utiliza estrategias eficientes de gestión de datos:

8.3.1. Procesamiento en Tiempo Real

- Las lecturas entrantes se procesan en un Servicio de Ingesta conectado a un message broker.
- Los datos se envían a una base de datos de series temporales (TimescaleDB) optimizada para lecturas continuas.

8.3.2. Almacenamiento

- Datos de usuario, configuraciones y hogares → BD relacional (PostgreSQL).
- Lecturas eléctricas → Base de series temporales.
- Reportes generados → almacenamiento en archivos externos (PDF/Excel).

8.3.3. Análisis y Alertas

- El motor de análisis detecta picos según umbrales definidos.
- Eventos generados se envían al servicio de notificaciones para alertar al usuario.

8.4. Seguridad en Red y Móviles

Medidas aplicadas:

8.4.1. Seguridad de comunicación

- Cifrado extremo a extremo mediante TLS 1.3.
- Certificados digitales para dispositivos IoT.
- Validación de integridad de paquetes.

8.4.2. Seguridad en backend

- Autenticación con OAuth2 + JWT.
- Control de accesos por roles (RBAC).
- Políticas de CORS correctamente definidas.
- Auditoría de eventos e intentos de acceso.

8.4.3. Seguridad en aplicaciones móviles

- Almacenamiento seguro de tokens en Keychain/SecureStorage.
- Uso de HTTPS obligatorio (WSS para WebSockets).
- Verificación del dominio (SSL pinning opcional).

8.4.4. Respaldo y Recuperación

- Copias automáticas de bases de datos.

- Políticas de recuperación ante fallos.
- Replicación en múltiples zonas de disponibilidad.

8.5. Justificación Técnica

- **MQTT** permite comunicación ligera ideal para dispositivos IoT.
- **Event-driven architecture** permite manejar miles de dispositivos simultáneamente.
- **SPA + Aplicación móvil** mejora la experiencia del usuario final con interfaces rápidas.
- **REST + WebSockets** equilibra compatibilidad y tiempo real.
- **TimescaleDB** es óptima para series temporales, reduciendo costos de consulta.
- Las medidas de **seguridad y respaldo** garantizan cumplimiento normativo y protección de datos.

Capítulo 9: Diseño de Interfaz y Experiencia de Usuario (UX/UI)

9.1. Perfil del usuario / usuario meta

Definición del usuario final según ODS y necesidades reales.

El usuario final pertenece a entornos urbanos o periurbanos y presenta las siguientes características:

- Persona responsable del uso y pago del servicio eléctrico en el hogar.
- Interés en reducir costos y consumo.
- Uso frecuente de smartphone y aplicaciones básicas.
- Necesita información clara, accesible y útil para tomar decisiones.

Contexto del uso del sistema

El usuario utiliza la aplicación en situaciones como:

- Revisar el consumo en tiempo real.
- Detectar picos o consumos anómalos.
- Configurar límites de alerta.
- Consultar reportes mensuales.
- Recibir sugerencias de ahorro energético.

9.2. Principios de diseño aplicados (HCI) explicar cómo se aplican en el prototipo

Consistencia

- Botones y colores mantienen el mismo estilo en toda la aplicación.
- El menú siempre ocupa la misma posición lateral.
- Íconos y gráficos siguen un patrón visual uniforme.

Visibilidad

- El panel principal muestra los indicadores más relevantes: consumo actual, histórico y alertas.
- Los estados del sistema se muestran mediante colores
- Notificaciones emergentes informan al usuario sobre eventos importantes.

Accesibilidad

- Contrastes suficientes en colores (púrpura claro y fondo blanco).
- Tipografías grandes y legibles.
- Permite uso en pantallas pequeñas y dispositivos de bajo rendimiento.

Control del usuario

- El usuario puede configurar sus propios límites de consumo.
- Tiene control sobre los hogares y dispositivos registrados.
- Puede elegir qué notificaciones recibir.

Retroalimentación

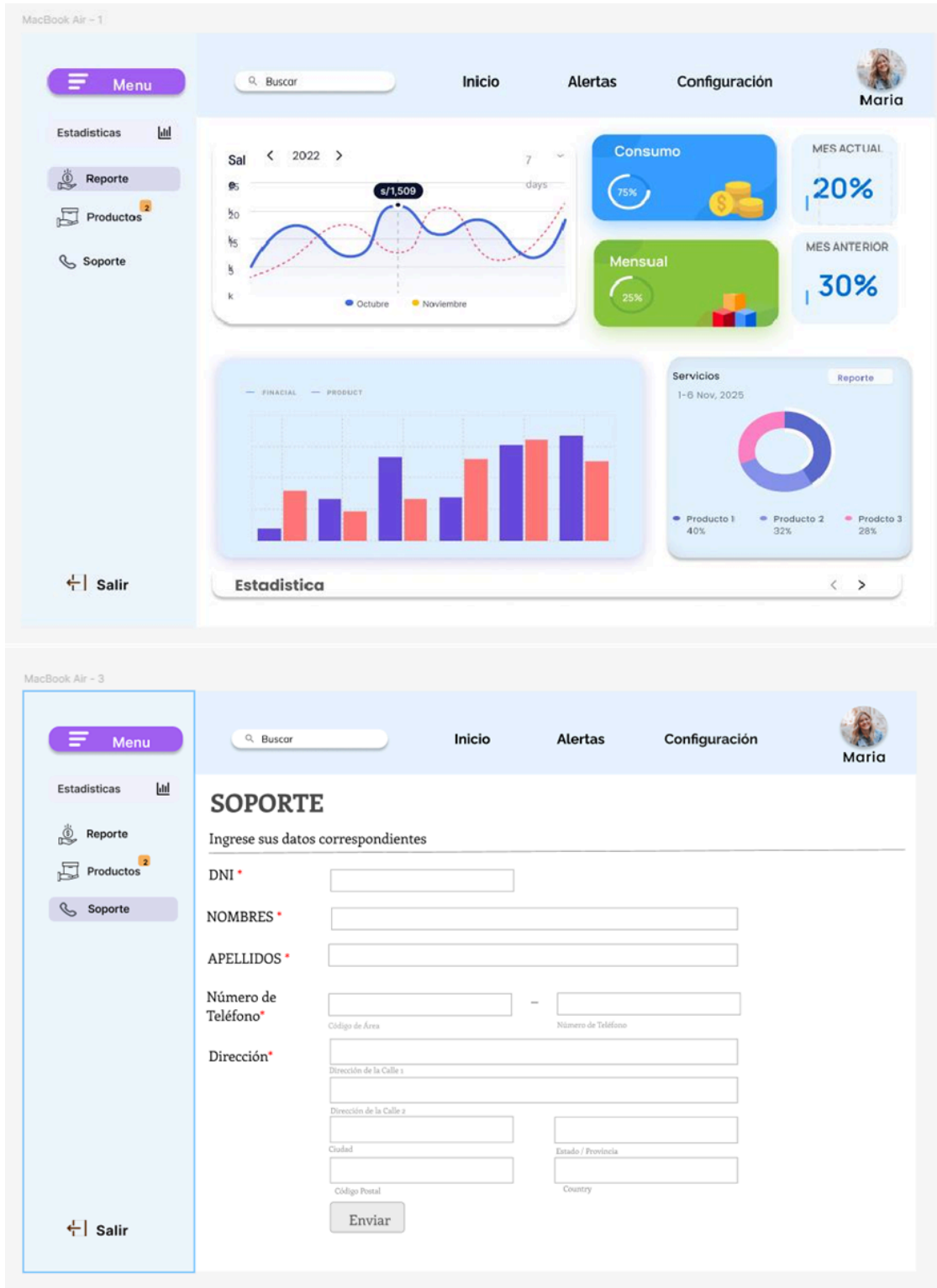
- Los formularios indican errores con mensajes claros.
- Los cambios se confirman mediante alertas visuales.
- Gráficos responden en tiempo real a datos de consumo.

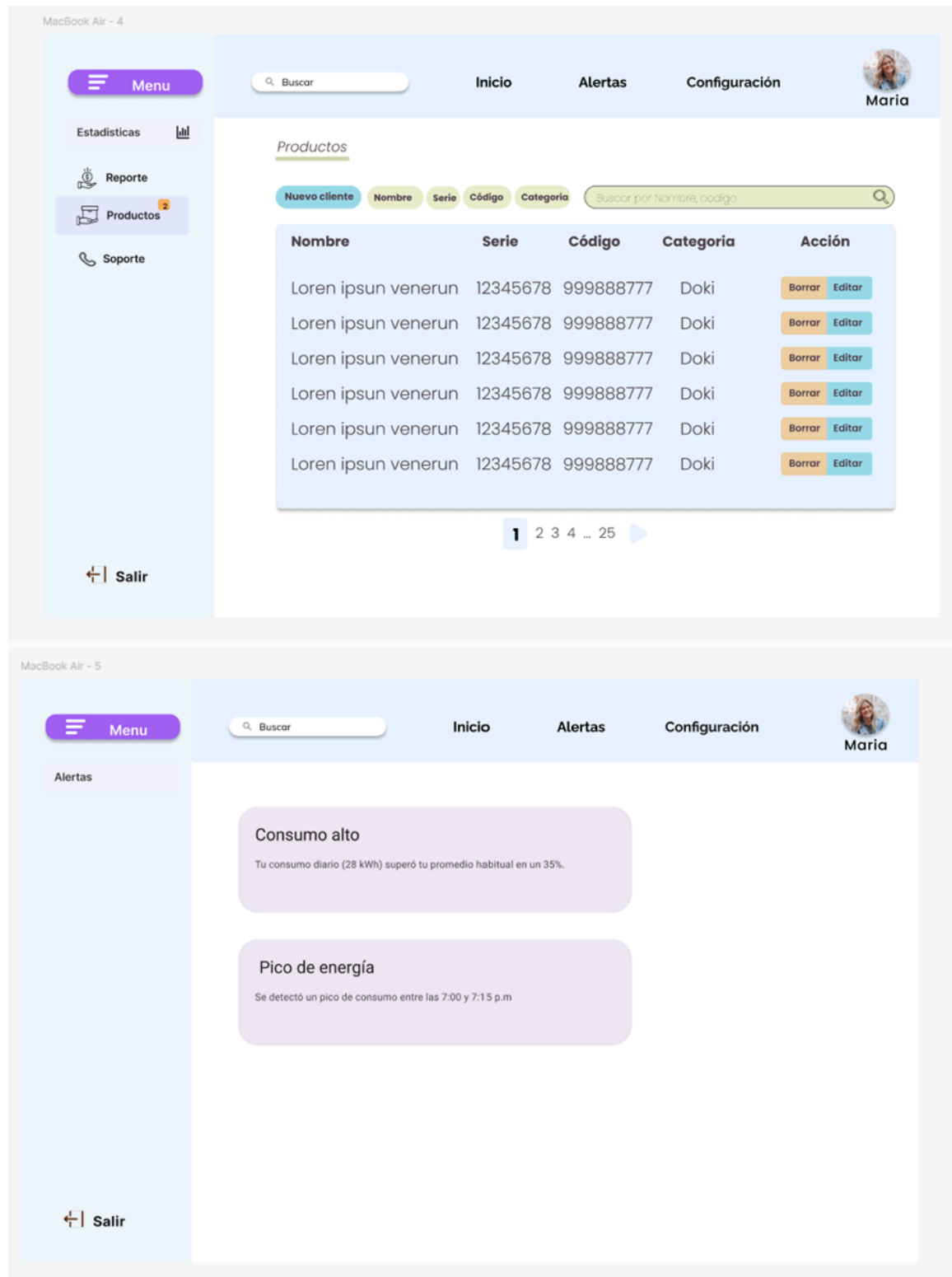
Simplicidad

- Interfaz limpia, sin saturación de elementos.
- Menú lateral con solo las opciones necesarias.
- Formularios cortos con etiquetas claras.

9.3 Diseño del prototipo (baja y alta fidelidad)

Prototipos realizados en Figma





Capturas de todas las pantallas principales



REGISTRO DE USUARIO

Ingrese sus datos correspondientes

DNI *

Usuario *

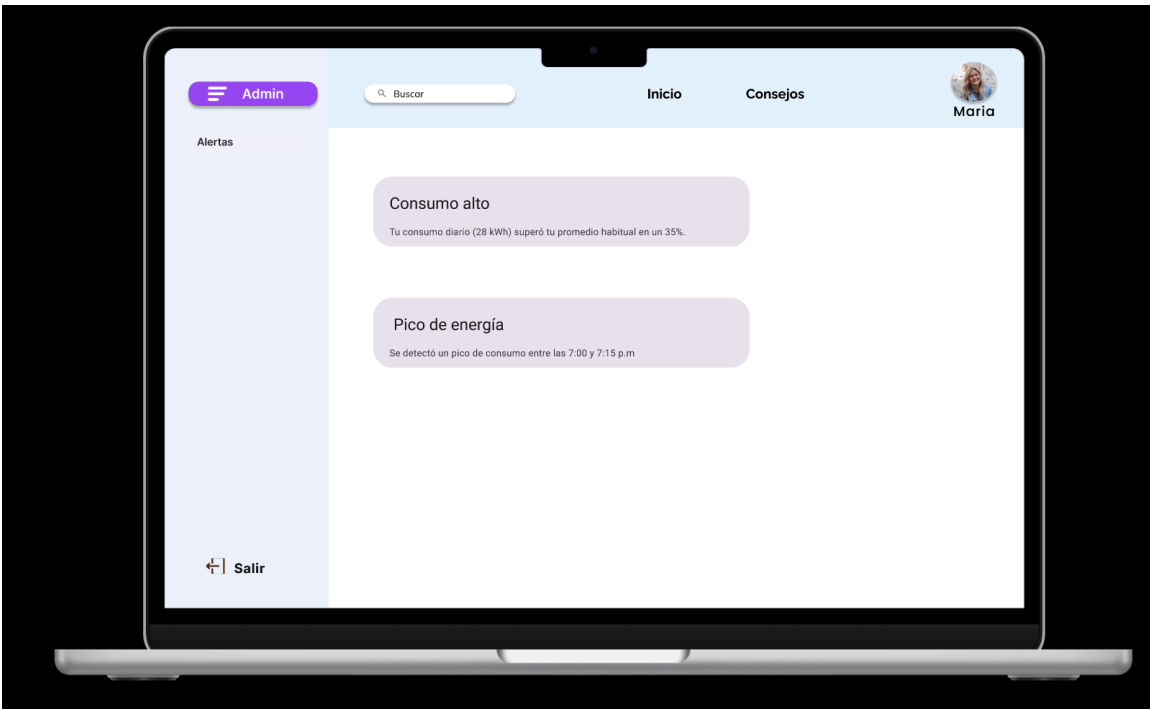
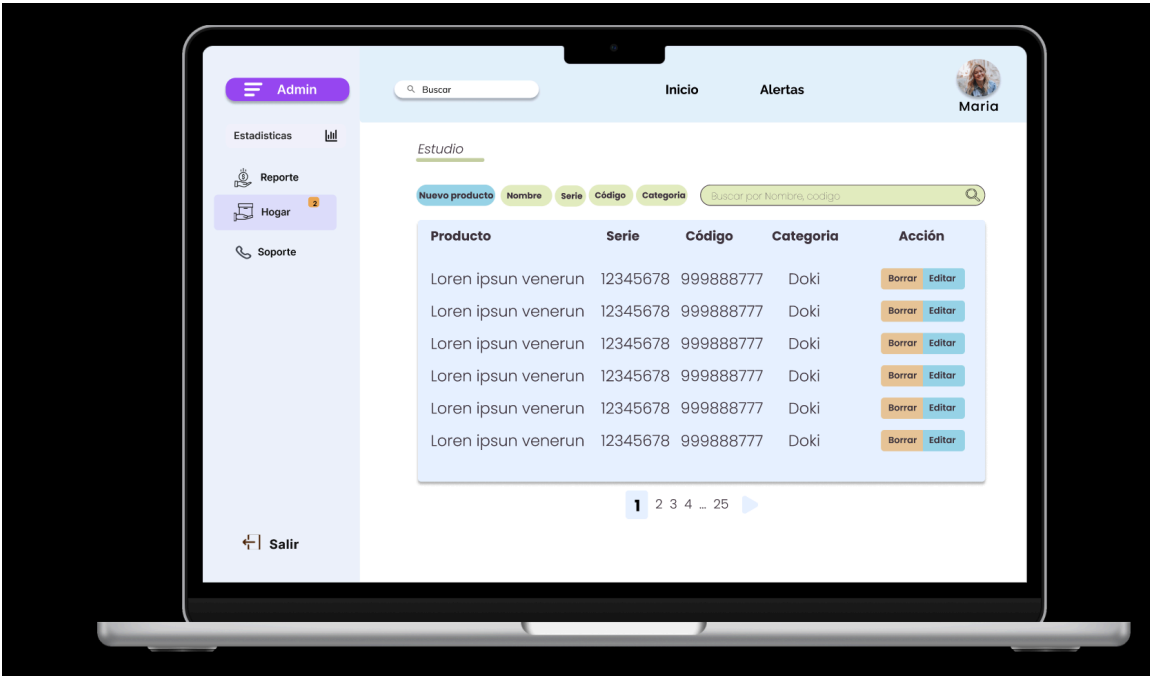
Contraseña *

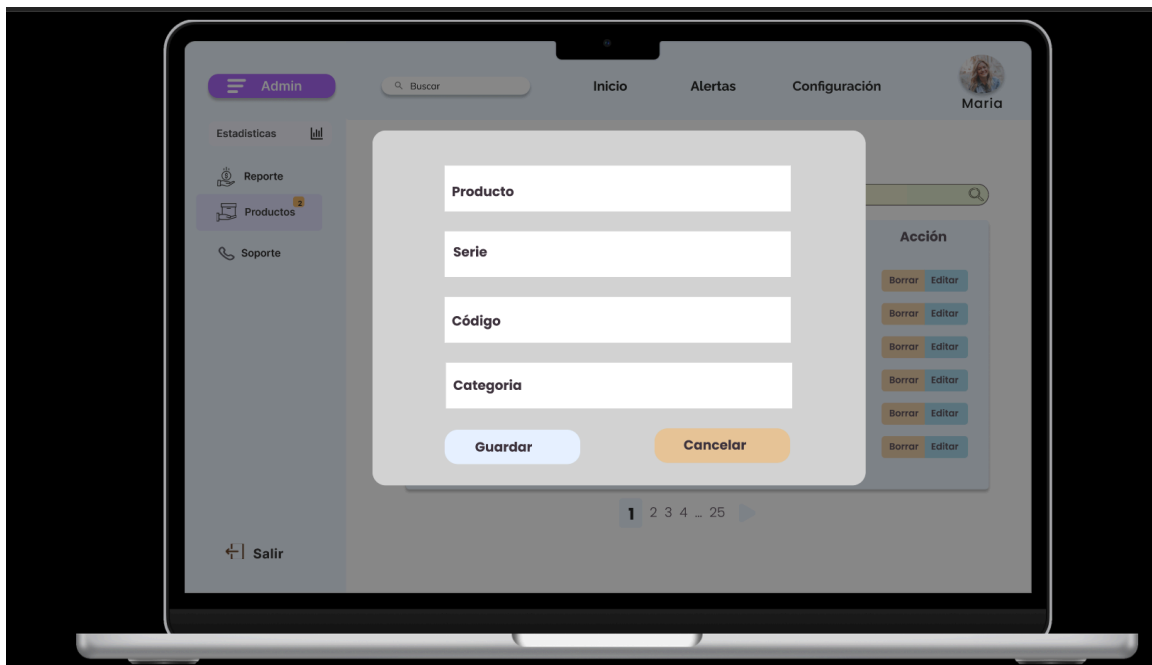
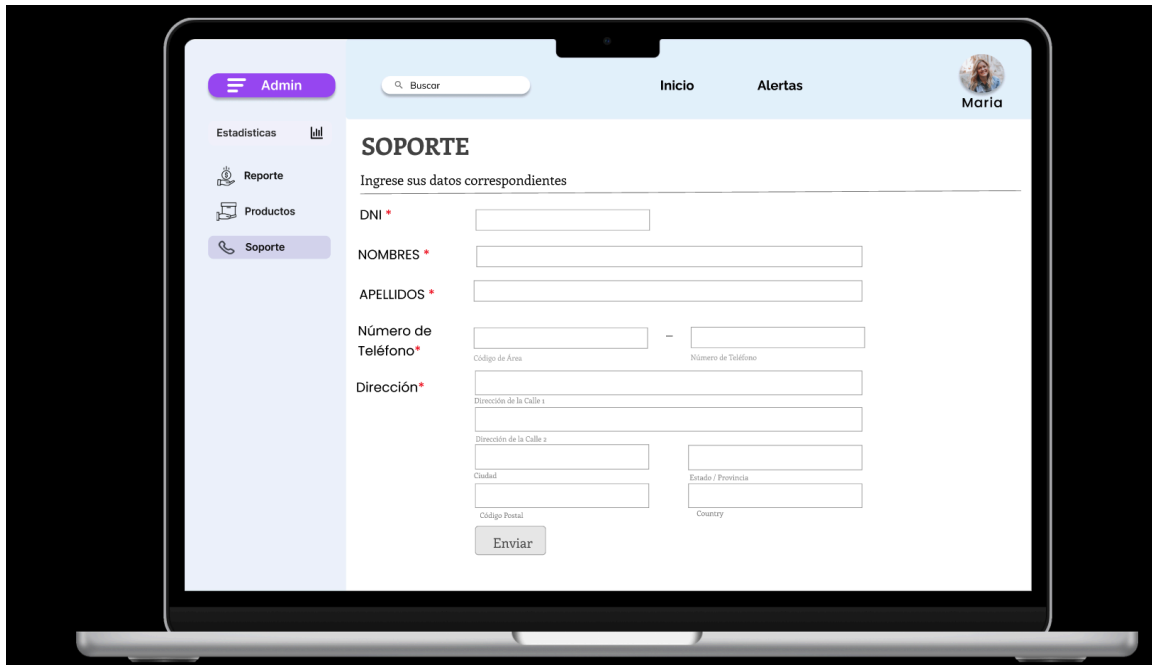
HOGAR *

DORMITORIO

SALON

ESTUDIO





Justificaciones del diseño

1. Colores

La interfaz utiliza una paleta en tonos suaves, lo cual transmite tranquilidad y profesionalismo.

El menú lateral mantiene un fondo violeta claro para diferenciarse del área de contenido, generando una separación visual clara entre navegación y formulario. Los acentos en color púrpura (botones y elementos activos) permiten identificar rápidamente la sección seleccionada sin saturar la pantalla.

2. Tipografías

Se emplea una tipografía moderna y redondeada que mejora la legibilidad y aporta una estética amigable.

Los títulos usan un mayor peso tipográfico, lo que facilita la jerarquización de la información, permitiendo al usuario identificar rápidamente las secciones principales.

3. Organización de la Información

La estructura del formulario sigue un diseño alineado y ordenado, con campos claramente etiquetados y validados mediante asteriscos rojos para indicar

La barra superior agrupa funciones complementarias como el buscador y el perfil del usuario, separándolas del contenido principal sin interferir con la captura de datos.

4. Usabilidad

El diseño mantiene espacios amplios entre campos, evitando errores de digitación y mejorando la experiencia en pantallas medianas o grandes.

9.4. Flujo de navegación del sistema

Mapa de navegación que representa el recorrido del usuario

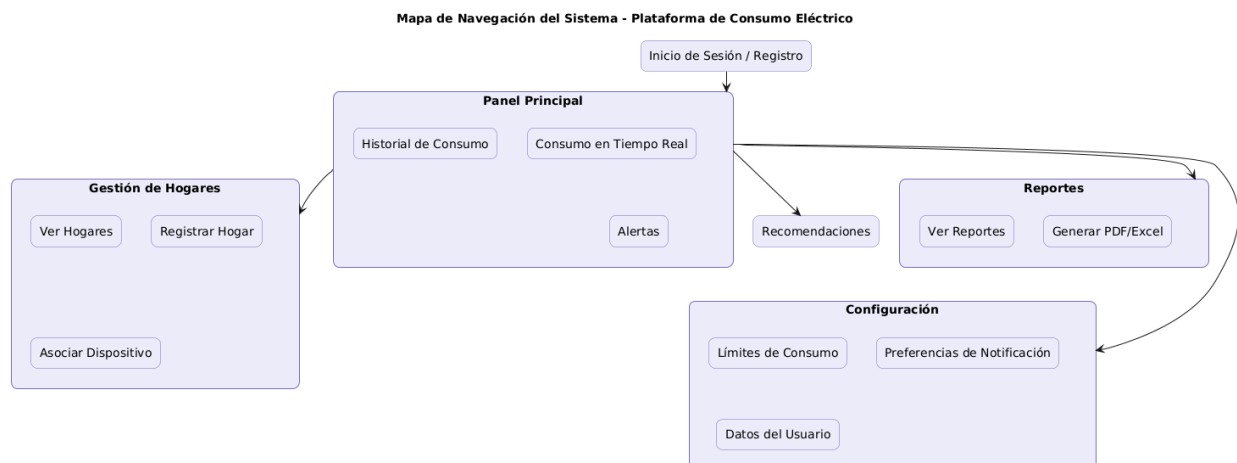
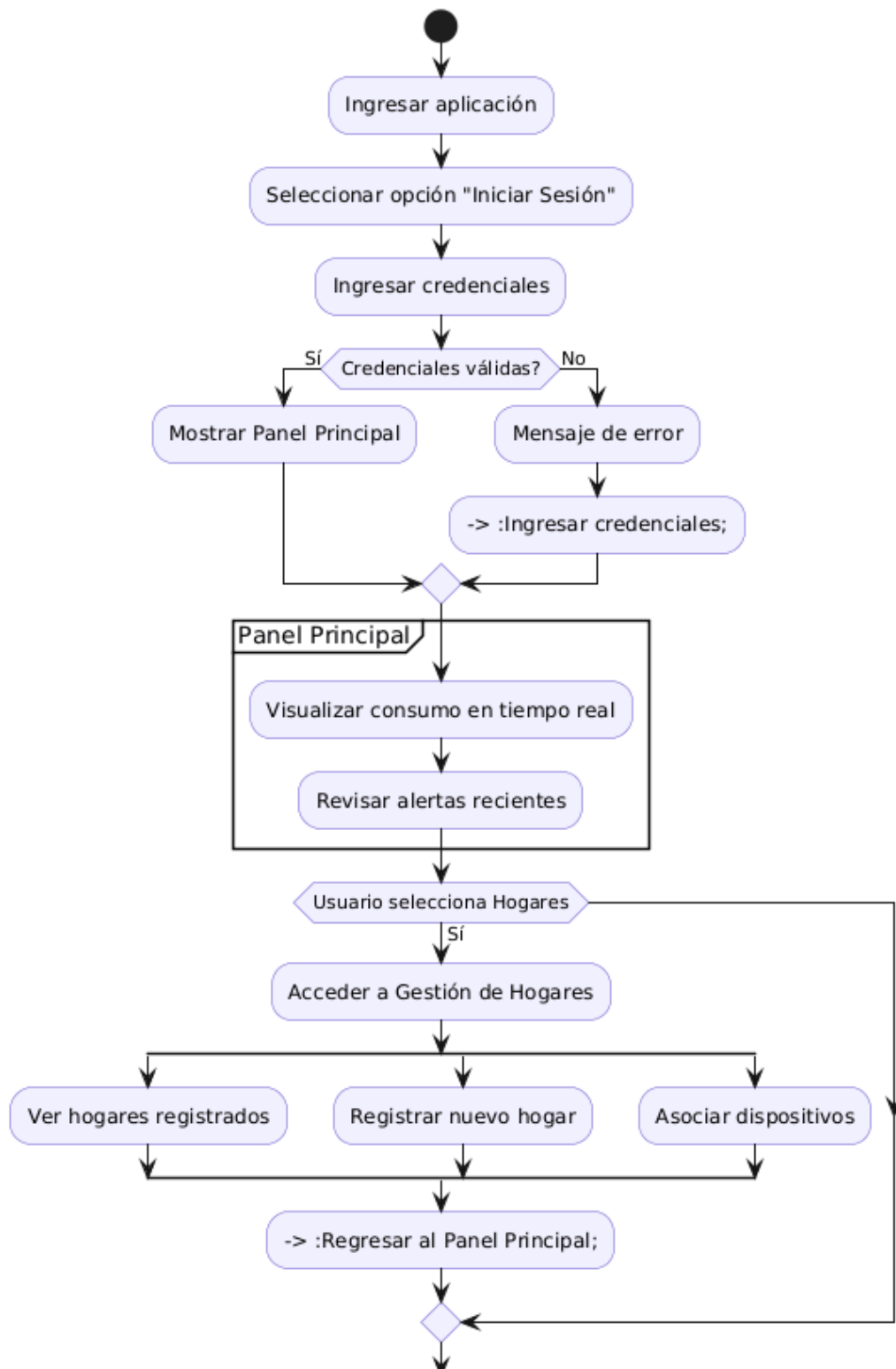
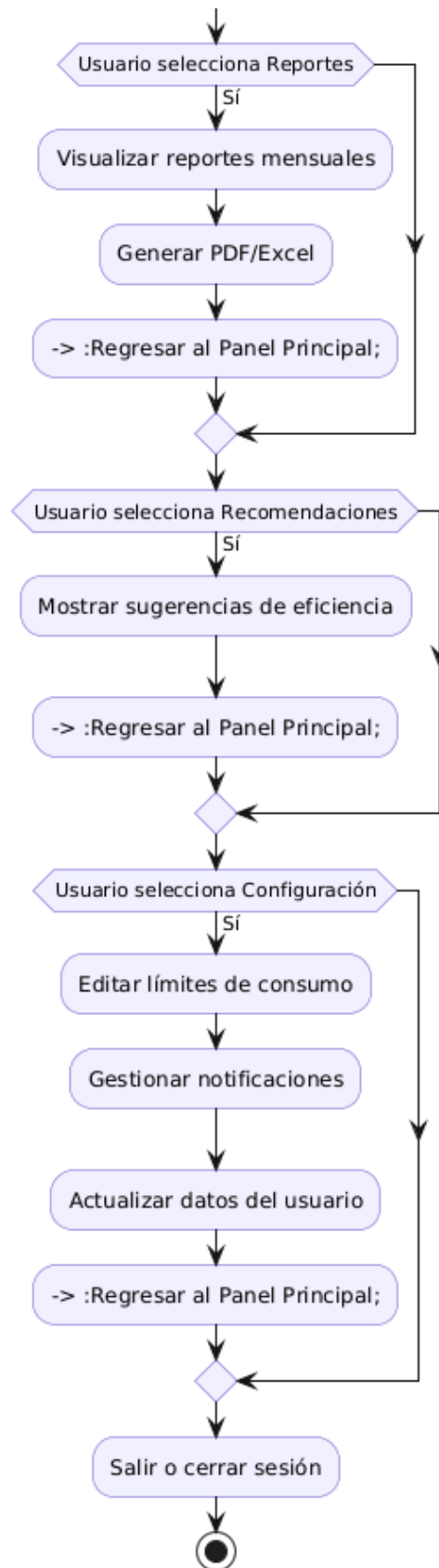


Diagrama de interacción o módulo-navegación.





Capítulo 10. Evaluación del Diseño y Matriz de Trazabilidad

10.1. Matriz de trazabilidad

La siguiente matriz relaciona los Requerimientos Funcionales (RF) y No Funcionales (RNF) con los casos de uso, los componentes del backend (Modelo y Controlador) y las vistas del frontend (Prototipos).

A. Requerimientos Funcionales (RF)

Req.	Descripción	Caso de Uso	Clase o Módulo (Backend)	Pantalla del Prototipo	Estado
RF-01	Registro de Hogar: Permitir al usuario registrar su vivienda y configurar los datos básicos del servicio eléctrico.	CU-01: Gestionar Cuenta	Usuario.java, Vivienda.java	frmRegistroUsuario.java	Cumplido
RF-02	Monitoreo de Consumo: Visualizar el consumo eléctrico actual e histórico (diario/mensual) en kWh y costo estimado.	CU-02: Visualizar Dashboard	ConsumoService, Medicion.java	DashboardPrincipal (Gráficos)	Cumplido

RF-03	Gestión de Dispositivos: Registrar electrodomésticos para identificar cuáles generan mayor gasto.	CU-03: Gestionar Dispositivos	Dispositivo.java, CalculoConsumo.java	frmMisElectrodomesticos	Cumple
RF-04	Alertas de Consumo: Notificar al usuario cuando se detecten picos anómalos o se supere un límite de gasto configurado.	CU-04: Gestionar Alertas	AlertaService.java, Notificacion.java	PanelNotificaciones	Cumple
RF-05	Recomendaciones Inteligentes: Mostrar consejos para reducir el consumo basados en los patrones detectados.	CU-05: Consultar Tips	RecomendacionDAO, TipEnergetico.java	frmConsejosAhorro	Parcial
RF-06	Simulación de Costos: Calcular el ahorro potencial si se aplican las	CU-06: Simular Ahorro	Simulador.java, Tarifa.java	frmSimuladorCostos	Cumple

	sugerencias de eficiencia.				
--	----------------------------	--	--	--	--

B. Requerimientos No Funcionales (RNF)

Req.	Descripción	Atributo Calidad	Elemento de Diseño / Arquitectura	Estado
RNF-01	Claridad de Datos: Las gráficas deben ser comprensibles para usuarios sin conocimientos técnicos de electricidad.	Usabilidad	Librería de Gráficos (Charts) / Diseño UI limpio	Cumple
RNF-02	Precisión: El cálculo del costo debe reflejar las tarifas vigentes de la empresa distribuidora.	Fiabilidad	Módulo Tarifario actualizable	Cumple
RNF-03	Disponibilidad: La plataforma debe estar accesible para consulta de datos históricos 24/7.	Disponibilidad	Base de Datos en la Nube (Cloud SQL/Firebase)	Cumple

RNF-04	Escalabilidad: Debe permitir agregar nuevos sensores o medidores IoT en el futuro.	Mantenibilidad	Patrón de Diseño Factory / Interfaces	Cumple
---------------	-------------------------------------------------------------------------------------------	----------------	---------------------------------------	---------------

10.2. Evaluación del diseño del sistema

10.2.1. Coherencia entre arquitectura, prototipo y base de datos

El sistema presenta una arquitectura sólida orientada a servicios, lo cual es vital para procesar los datos de consumo:

1. Mapeo de Datos Reales:

Existe una correspondencia directa entre los recibos de luz físicos (mencionados en el problema) y el Modelo de Base de Datos. Las tablas Lecturas, Tarifas y Consumo_Historico permiten reconstruir digitalmente la facturación para que el usuario entienda "por qué paga lo que paga".

2. Flujo de Información (Backend-Frontend):

La arquitectura separa la lógica de cálculo (conversión de Watts a Soles/Dólares) de la interfaz visual. Esto asegura que si las tarifas eléctricas cambian, solo se actualiza el backend (TarifaService) sin necesidad de rediseñar las pantallas del usuario.

3. Integración con ODS 7:

El diseño de la base de datos incluye campos específicos para medir el "Impacto Ambiental" (ej. cálculo de huella de carbono basada en kWh), lo cual conecta directamente el software con el objetivo de sostenibilidad del proyecto.

10.2.2. Identificación de mejoras y ajustes

Durante el desarrollo del prototipo de la plataforma de energía, se realizaron los siguientes ajustes:

1. Simplificación de Unidades de Medida:

Problema: Inicialmente, el prototipo mostraba demasiados datos técnicos (Voltaje, Amperaje, Factor de Potencia).

Ajuste: Se simplificó la vista principal para mostrar solo Dinero (Costo) y Energía (kWh), moviendo los datos técnicos a una vista avanzada, mejorando la comprensión para las familias (público objetivo).

2. Alertas Proactivas:

Mejora: Se añadió una funcionalidad no prevista inicialmente: alertas predictivas ("A este ritmo, tu recibo superará los X soles"), en lugar de sólo alertas reactivas.

3. Optimización de Gráficos:

Ajuste Técnico: Se optimizó la consulta SQL para los reportes históricos, ya que la carga de datos de consumo anual era lenta. Se implementaron índices en la tabla de Lecturas por fecha.

10.2.3. Evaluación del cumplimiento de principios de usabilidad (HCI)

Dado que la plataforma está dirigida a familias (entornos domésticos), la usabilidad es crítica:

- **Visibilidad del Estado (Cumple):**

El usuario sabe en todo momento si su consumo está en un nivel "Verde" (Eficiente), "Amarillo" (Precaución) o "Rojo" (Excesivo) mediante indicadores visuales tipo semáforo en el Dashboard.

- **Relación entre el sistema y el mundo real (Cumple):**

La aplicación traduce el consumo abstracto de energía en metáforas comprensibles (ej. "Has gastado el equivalente a dejar un foco prendido por 5 días"), cumpliendo con el objetivo educativo del proyecto.

- **Estética y Diseño Minimalista (Cumple):**

Se prioriza la limpieza visual. Los consejos de ahorro (ODS 7) se presentan como tarjetas simples y accionables (ej. "Desconecta el microondas"), evitando saturar al usuario con texto técnico.

- **Prevención de Errores (Cumple):**

Al ingresar lecturas manuales o registrar electrodomésticos, el sistema valida que los valores sean lógicos (no permite consumos negativos ni potencias irreales), guiando al usuario para mantener la integridad de los datos.

10.3. Retroalimentación obtenida

Para evaluar la efectividad del prototipo y detectar fallos antes de la implementación final, se llevó a cabo una Simulación de Pruebas de Usuario y una Revisión Heurística. A continuación, se detallan los hallazgos y correcciones:

10.3.1. Errores detectados

Durante las pruebas con usuarios representativos (público general), se identificaron los siguientes problemas:

1. **Confusión en la terminología:** Los usuarios no entendían la diferencia entre kW (potencia) y kWh (energía), lo que dificulta interpretar los gráficos de consumo diario.
2. **Visibilidad de alertas:** Las notificaciones de "Exceso de consumo" aparecían como pequeños iconos en la esquina superior, pasando desapercibidas para el 40% de los usuarios probados.
3. **Complejidad en el registro de dispositivos:** El formulario para agregar un nuevo electrodoméstico pedía demasiados datos técnicos (voltaje, amperaje) que el usuario común desconoce.

10.3.2. Cambios realizados

En base a los errores detectados, se aplicaron las siguientes correcciones en el diseño y código:

1. **Simplificación de Unidades:** Se agregó un botón de alternancia ("Switch") en el Dashboard principal que permite cambiar la vista de kWh a Moneda Local (Soles/Dólares). Ahora el usuario ve prioritariamente cuánto dinero está gastando, lo cual es más comprensible.
2. **Rediseño de Alertas:** Se implementaron ventanas emergentes (Pop-ups) y cambios de color en el fondo de la pantalla (a rojo suave) cuando el consumo supera el límite establecido, asegurando que la alerta sea vista.
3. **Base de Datos de Electrodomésticos:** Se precargó una base de datos con valores promedio. Ahora el usuario solo selecciona "Refrigeradora Mediana" de una lista desplegable, y el sistema autocompleta la potencia estimada, permitiendo editarla solo si es necesario.

10.3.3. Mejoras propuestas (Trabajo Futuro)

Para versiones posteriores del sistema, se proponen las siguientes funcionalidades basadas en el feedback recibido:

- **Gamificación:** Implementar un sistema de medallas (ej. "Hogar Ecoeficiente del Mes") para incentivar la competencia sana entre vecinos o amigos.
- **Modo Oscuro (Dark Mode):** Dado que la aplicación promueve el ahorro de energía, incluir una interfaz oscura nativa sería coherente con el mensaje, ahorrando batería en dispositivos móviles con pantallas OLED.

10.4. Reflexión sobre el aporte al ODS

El proyecto "Plataforma de Consumo Eléctrico" fue concebido bajo los lineamientos del Objetivo de Desarrollo Sostenible N.º 7 (Energía asequible y no contaminante). A continuación, se demuestra cómo cada capa del software contribuye a la meta 7.3: duplicar la tasa mundial de mejora de la eficiencia energética.

10.4.1. El diseño de la interfaz como herramienta de conciencia

La interfaz no es meramente decorativa; actúa como un educador visual.

- **Psicología del Color:** Se utiliza un "Semáforo Energético". Los gráficos se tornan verdes cuando el consumo es eficiente y rojos cuando es excesivo. Esto

genera un condicionamiento visual inmediato en el usuario, incentivando a mantener su interfaz en "verde", lo que se traduce en un menor consumo real de electricidad.

- **Visualización de Impacto:** Al mostrar no solo el costo, sino también la "Huella de Carbono" equivalente (ej. Tu consumo de hoy equivale a talar 2 árboles), la interfaz tangibiliza el impacto ambiental, fomentando la responsabilidad social.

10.4.2. La navegación orientada a la acción

La estructura de navegación fue diseñada para reducir la fricción entre la intención de ahorrar y la acción de ahorrar.

- **Accesibilidad a Consejos:** El módulo de "Tips de Ahorro" no está escondido en un submenú; se presenta de forma contextual. Si el usuario navega a la sección de "Lavadora", el sistema muestra inmediatamente consejos específicos (ej. "Usa agua fría para ahorrar 80% de energía").
- **Reportes Intuitivos:** La navegación permite comparar meses anteriores con un solo clic, permitiendo a las familias evaluar si sus cambios de hábitos están funcionando realmente, reforzando comportamientos positivos.

10.4.3. Las funcionalidades y la eficiencia energética

Las funciones del backend están programadas para maximizar la eficiencia:

- **Detección de "Consumo Fantasma":** El algoritmo del sistema es capaz de identificar consumos constantes y bajos en horas de la madrugada, alertando al usuario sobre aparatos en modo stand-by que deberían desconectarse, atacando uno de los mayores desperdicios energéticos en los hogares.
- **Simulador de Costos:** Esta funcionalidad permite al usuario proyectar escenarios (ej. "¿Cuánto ahorraría si cambio mis focos a LED?"). Al cuantificar el ahorro económico futuro, la herramienta motiva la inversión en tecnología más limpia y eficiente.

Conclusiones y Recomendaciones

Conclusión del equipo:

El equipo concluye que el desarrollo del proyecto completo permitió comprender y aplicar de manera práctica conceptos fundamentales de análisis, diseño e implementación de sistemas. La definición clara de los requerimientos funcionales, no funcionales y de dominio fue clave para orientar correctamente todas las etapas del proyecto, desde la planificación hasta la implementación. La elaboración de modelos funcionales y de datos, así como diagramas UML de clase, actividad y secuencia, facilitó la visualización de la arquitectura del sistema y la comprensión de los flujos de información. Además, la metodología ágil SCRUM demostró ser efectiva para organizar el trabajo en sprints, priorizar tareas y asegurar entregas parciales que pudieron ser evaluadas y ajustadas de manera iterativa.

Lecciones aprendidas:

Durante la ejecución del proyecto, el equipo aprendió la importancia de mantener una comunicación constante y efectiva entre todos los miembros, lo que permitió resolver dudas rápidamente y asegurar una visión compartida del sistema. Se valoró la utilidad de las herramientas de gestión y modelado, como Jira, Draw.io, Visual Paradigm y GitHub, que facilitaron la planificación, seguimiento y control del desarrollo. Asimismo, se identificó que documentar correctamente los requerimientos y mantenerlos vinculados a historias de usuario reduce la posibilidad de errores o malentendidos durante la implementación. Por último, se aprendió que los diagramas y modelos no solo sirven para diseñar, sino también para mejorar la comunicación con futuros usuarios o clientes, y para facilitar el mantenimiento del sistema.

Recomendaciones para futuras mejoras del sistema:

Para versiones futuras del sistema se recomienda incorporar módulos de análisis predictivo y recomendaciones personalizadas, que permitan optimizar el consumo eléctrico de manera más proactiva. También se sugiere mejorar la interfaz de usuario, incluyendo dashboards interactivos y reportes visuales más intuitivos. La integración de notificaciones móviles y alertas configurables puede aumentar la efectividad del sistema y la satisfacción del usuario. Asimismo, fortalecer el sistema de roles y permisos

permitiría un control más preciso de accesos para distintos tipos de usuarios, incluyendo administradores y entidades externas. Finalmente, se recomienda mantener la documentación actualizada y continuar con la práctica de metodologías ágiles, para facilitar futuras ampliaciones, mantenimiento y mejoras del sistema.

Referencias bibliográficas

- [1] ORGANIZACIÓN DE NACIONES UNIDAS. Objetivos de Desarrollo Sostenible. Nueva York: ONU, 2015. Disponible en: <https://www.un.org/sustainabledevelopment/es/energy/>
- [2] CEPAL. Eficiencia energética en América Latina y el Caribe: situación y perspectivas. Santiago de Chile: CEPAL, 2020.
- [3] MINISTERIO DE ENERGÍA Y MINAS DEL PERÚ. Informe anual del consumo eléctrico residencial 2022. Lima: MINEM, 2023.
- [4] INTERNATIONAL ENERGY AGENCY. Energy Efficiency 2022. Paris: IEA, 2022.
- [5] Schwaber, K.; Beedle, M. *Agile Software Development with Scrum*. Upper Saddle River: Prentice Hall, 2002. ISBN 978-0130676340.

Anexos

Evidencias gráficas (capturas de Jira, capturas de GITHUB y commits, evidencias de trabajo en equipo).

link de github:

<https://github.com/RickCrispin/AnalisisDeSoftware-ConsumoElectrico.git>

Link de Canva:

<https://www.canva.com/design/DAG0fE7Sx5w/Q1qpUBdMRmirmjVkuUDMaw/edit>

Link de Jira:

<https://ccmjc.atlassian.net/jira/software/projects/PCE/boards/100/backlog?atlOrigin=eyJpIjoiMzQ0YzlmNDIjNWlyNGY2MDhjZjc2OWJkODhINjc3ZTkjLCJwIjoiaj9>

Proyectos

Plataforma_Consumo_Electrico

Resumen

Cronograma

Backlog

Tablero

Calendario

Lista

Formularios

Metas

Todas las actividades

Código

Actividades archivadas

Páginas

Accesos rápidos

Buscar en la lista

Filtro

Grupo

Buscar

+ Crear

Versión de prueba de Premium

Chat de Rovo

<input type="checkbox"/>	Tipo	Clave	Resumen	Estado	Comentarios	Sprint	Persona asignada	Fecha de vencimiento	Etiquetas	Creada	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-1	Gestión de Usuarios	TAREAS POR HACER	<div>Añadir comentario</div>		<div>PAMELA MILAGROS MONTERO PEREZ</div>			30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-8	HU01: Registro de usuarios	TAREAS POR HACER	<div>Añadir comentario</div>	Tablero Sprint 1	<div>PAMELA MILAGROS MONTERO PEREZ</div>		Alta-prioridad	30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-9	HU02: Autenticación de usuarios	TAREAS POR HACER	<div>Añadir comentario</div>	Tablero Sprint 1	<div>PAMELA MILAGROS MONTERO PEREZ</div>		Alta-prioridad	30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-2	Gestión de Hogares y Medidores	TAREAS POR HACER	<div>Añadir comentario</div>		<div>JOSE CARLOS CCENTE MEJIA</div>			30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-12	HU03: Registro de hogares	TAREAS POR HACER	<div>Añadir comentario</div>	Tablero Sprint 1	<div>JOSE CARLOS CCENTE MEJIA</div>		Alta-prioridad	30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-13	HU04: Asociación de medidores inteligentes	TAREAS POR HACER	<div>Añadir comentario</div>	Tablero Sprint 1	<div>JOSE CARLOS CCENTE MEJIA</div>		Alta-prioridad	30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-3	75328236@continental.edu.pe	TAREAS POR HACER	<div>Añadir comentario</div>		<div>jose enrique</div>			30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-14	HU05: Visualización en tiempo real	TAREAS POR HACER	<div>Añadir comentario</div>	Tablero Sprint 1	<div>jose enrique</div>		Alta-prioridad	30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-15	HU06: Filtros básicos de consumo	TAREAS POR HACER	<div>Añadir comentario</div>	Tablero Sprint 1	<div>jose enrique</div>		Media-Prioridad	30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-16	HU07: Consulta de historial completo	TAREAS POR HACER	<div>Añadir comentario</div>	Tablero Sprint 2	<div>jose enrique</div>		Media-Prioridad	30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-4	Reportes	TAREAS POR HACER	<div>Añadir comentario</div>		<div>CRISPIN BENDEZU RICK BERNIE</div>			30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-17	HU08: Generación de reportes mensuales	TAREAS POR HACER	<div>Añadir comentario</div>	Tablero Sprint 2	<div>CRISPIN BENDEZU RICK BERNIE</div>		Media-Prioridad	30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-18	HU09: Reportes consolidados para administrador	TAREAS POR HACER	<div>Añadir comentario</div>	Tablero Sprint 2	<div>CRISPIN BENDEZU RICK BERNIE</div>		Media-Prioridad	30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-5	Alertas	TAREAS POR HACER	<div>Añadir comentario</div>		<div>PAMELA MILAGROS MONTERO PEREZ</div>			30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-19	HU10: Alertas de consumo	TAREAS POR HACER	<div>Añadir comentario</div>	Tablero Sprint 2	<div>PAMELA MILAGROS MONTERO PEREZ</div>		Alta-prioridad	30 sept 2	
<input type="checkbox"/>	<div><div></div><div></div></div>	PCE-20	HU11: Configuración de límites de consumo	TAREAS POR HACER	<div>Añadir comentario</div>	Tablero Sprint 2	<div>PAMELA MILAGROS MONTERO PEREZ</div>		Media-Prioridad	30 sept 2	

+ Crear

Proyectos

Plataforma_Consumo_Electrico

...

Resumen

Cronograma

Backlog

Tablero

Calendario

Lista

Formularios

Metas

Todas las actividades

Código

Actividades archivadas

Páginas

Accesos rápidos

+

Buscar en la lista

Filtro

Grupo

<input type="checkbox"/>	Tipo	Clave	Resumen	Estado	Comentarios	Sprint	Persona asignada	Fecha de vencimiento	Etiquetas	Creada	+
<input type="checkbox"/>		PCE-13	HU04: Asociación de medidores inteligentes	TAREAS POR HACER	Añadir comentario	Tablero Sprint 1	JOSE CARLOS CCENTE MEIA		Alta-prioridad	30 sept 2	
<input type="checkbox"/>		PCE-3	75328236@continental.edu.pe	TAREAS POR HACER	Añadir comentario		jose enrique			30 sept 2	
<input type="checkbox"/>		PCE-14	HU05: Visualización en tiempo real	TAREAS POR HACER	Añadir comentario	Tablero Sprint 1	jose enrique		Alta-prioridad	30 sept 2	
<input type="checkbox"/>		PCE-15	HU06: Filtros básicos de consumo	TAREAS POR HACER	Añadir comentario	Tablero Sprint 1	jose enrique		Media-Prioridad	30 sept 2	
<input type="checkbox"/>		PCE-16	HU07: Consulta de historial completo	TAREAS POR HACER	Añadir comentario	Tablero Sprint 2	jose enrique		Media-Prioridad	30 sept 2	
<input type="checkbox"/>		PCE-4	Reportes	TAREAS POR HACER	Añadir comentario		CRISPIN BENDEZU RICK BERNIE			30 sept 2	
<input type="checkbox"/>		PCE-17	HU08: Generación de reportes mensuales	TAREAS POR HACER	Añadir comentario	Tablero Sprint 2	CRISPIN BENDEZU RICK BERNIE		Media-Prioridad	30 sept 2	
<input type="checkbox"/>		PCE-18	HU09: Reportes consolidados para administrador	TAREAS POR HACER	Añadir comentario	Tablero Sprint 2	CRISPIN BENDEZU RICK BERNIE		Media-Prioridad	30 sept 2	
<input type="checkbox"/>		PCE-5	Alertas	TAREAS POR HACER	Añadir comentario		PAMELA MILAGROS MONTERO PEREZ			30 sept 2	
<input type="checkbox"/>		PCE-19	HU10: Alertas de consumo	TAREAS POR HACER	Añadir comentario	Tablero Sprint 2	PAMELA MILAGROS MONTERO PEREZ		Alta-prioridad	30 sept 2	
<input type="checkbox"/>		PCE-20	HU11: Configuración de límites de consumo	TAREAS POR HACER	Añadir comentario	Tablero Sprint 2	PAMELA MILAGROS MONTERO PEREZ		Media-Prioridad	30 sept 2	
<input type="checkbox"/>		PCE-6	Recomendaciones	TAREAS POR HACER	Añadir comentario		jose enrique			30 sept 2	
<input type="checkbox"/>		PCE-21	HU12: Consejos de eficiencia energética	TAREAS POR HACER	Añadir comentario	Tablero Sprint 2	jose enrique		Media-Prioridad	30 sept 2	
<input type="checkbox"/>		PCE-7	Administración	TAREAS POR HACER	Añadir comentario		CRISPIN BENDEZU RICK BERNIE			30 sept 2	
<input type="checkbox"/>		PCE-22	HU13: Consultar estadísticas globales	TAREAS POR HACER	Añadir comentario	Tablero Sprint 2	CRISPIN BENDEZU RICK BERNIE		Media-Prioridad	30 sept 2	
<input type="checkbox"/>		PCE-23	HU14: Gestión de roles y permisos	TAREAS POR HACER	Añadir comentario	Tablero Sprint 2	JOSE CARLOS CCENTE MEIA		Alta-prioridad	30 sept 2	

Proyectos

Plataforma_Consumo_Electrico

...

Resumen

Cronograma

Backlog

Tablero

Calendario

Lista

Formularios

Metas

Todas las actividades

Código

Actividades archivadas

Páginas

Accesos rápidos

+

Buscar en el backl...

Filter

Tablero Sprint 1

Añadir fechas (5 actividades)

Implementar la base de la plataforma y funciones esenciales para registro, autenticación, registro de hogares, asociación de medidores y monitoreo básico.

0

5

0

Iniciar sprint

...

PCE-8

HU01: Registro de usuarios

GESTIÓN DE USUARIOS

TAREAS POR HACER

PCE-9

HU02: Autenticación de usuarios

GESTIÓN DE USUARIOS

TAREAS POR HACER

PCE-12

HU03: Registro de hogares

GESTIÓN DE HOGARES...

TAREAS POR HACER

PCE-13

HU04: Asociación de medidores inteligentes

GESTIÓN DE HOGARES...

TAREAS POR HACER

PCE-14

HU05: Visualización en tiempo real

75328236@CONTINEN...

TAREAS POR HACER

PCE-15

HU06: Filtros básicos de consumo

75328236@CONTINEN...

TAREAS POR HACER

6 de 6 actividades visibles

Estimación: 0 de 0

Tablero Sprint 2

Añadir fechas (8 actividades)

Añadir funcionalidades avanzadas como reportes, alertas, recomendaciones, administración y estadísticas globales.

0

8

0

Iniciar sprint

...

PCE-16

HU07: Consulta de historial completo

75328236@CONTINEN...

TAREAS POR HACER

PCE-17

HU08: Generación de reportes mensuales

REPORTES

TAREAS POR HACER

PCE-18

HU09: Reportes consolidados para administrador

REPORTES

TAREAS POR HACER

PCE-19

HU10: Alertas de consumo

ALERTAS

TAREAS POR HACER

PCE-20

HU11: Configuración de límites de consumo

ALERTAS

TAREAS POR HACER

PCE-21

HU12: Consejos de eficiencia energética

RECOMENDACIONES

TAREAS POR HACER

PCE-22

HU13: Consultar estadísticas globales

ADMINISTRACIÓN

TAREAS POR HACER

PCE-23

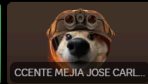
HU14: Gestión de roles y permisos

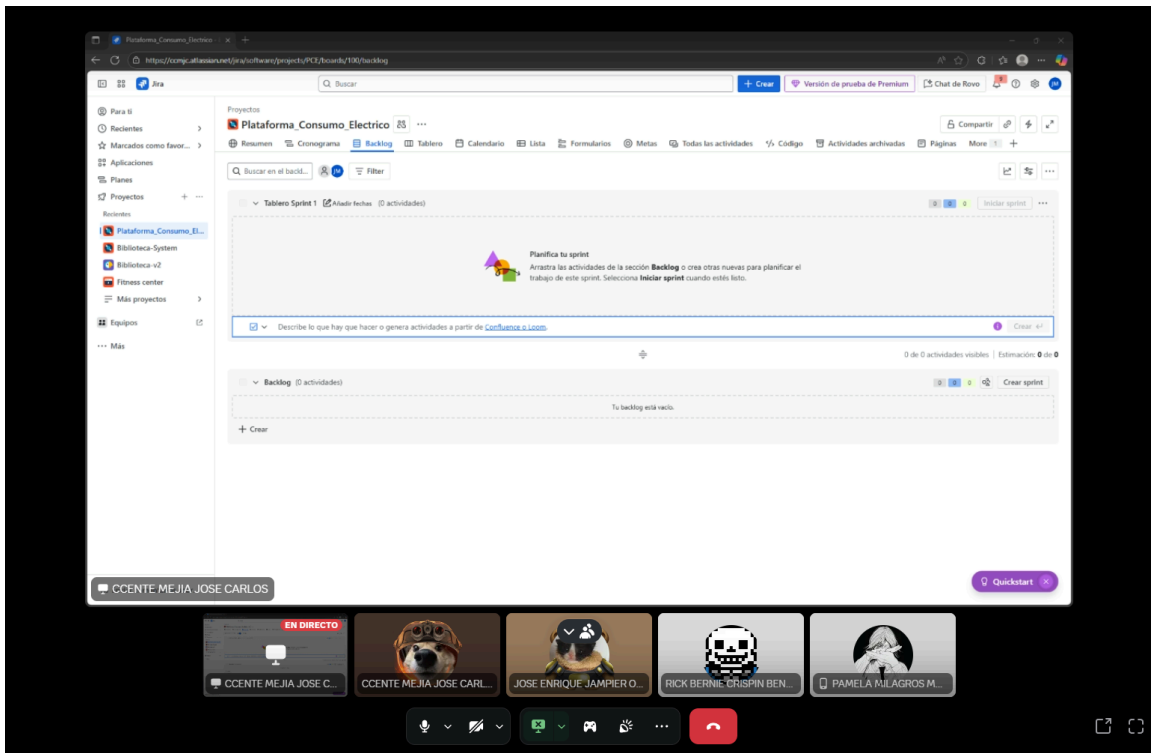
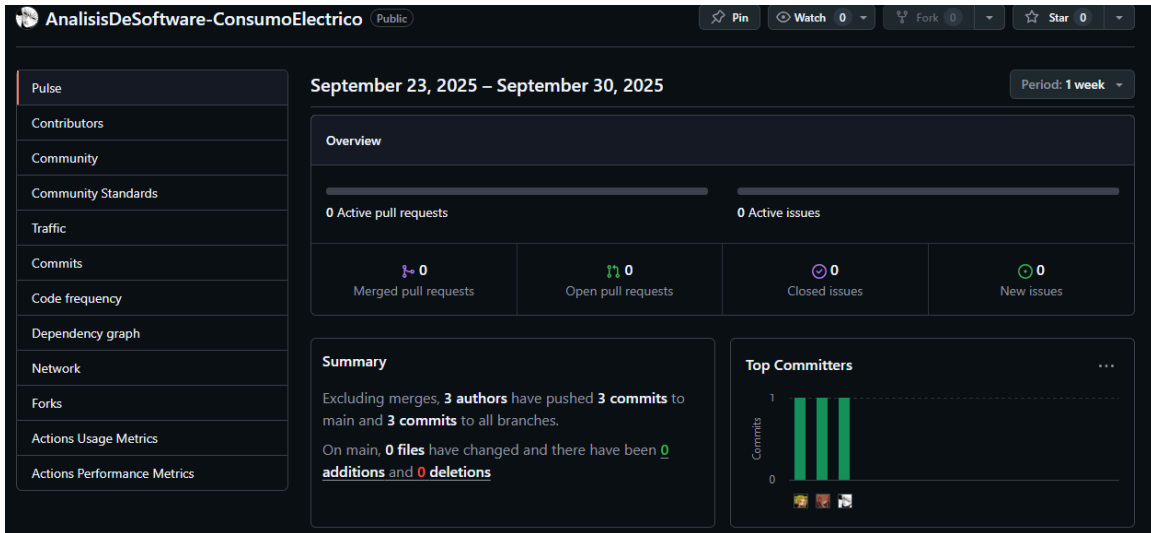
ADMINISTRACIÓN

TAREAS POR HACER

Plantilla_Informe_Proyecto_ADS

RNF2.1	Procesamiento en tiempo real con retardo máximo de 5 segundos.	RF2.3, RF3.1	Depende de la recepción de datos y su visualización.
RNF2.2	Soportar al menos 10,000 usuarios concurrentes en la primera fase.	RF1.1, RF3.1	Relacionado a la creación de usuarios y acceso al sistema en tiempo real.
RNF2.3	Arquitectura escalable horizontalmente (microservicios/cloud).	RNF2.2	Depende de la necesidad de soportar grandes volúmenes de usuarios.
RNF3.1	Interfaz intuitiva y accesible para usuarios sin conocimientos técnicos.	RF3.1, RF3.3	Depende de la visualización de datos y gráficos de consumo.
RNF3.2	Disponibilidad en versión web y móvil (responsive o app).	RF3.1, RF3.2	Relacionado con la presentación de datos de consumo en diferentes plataformas.
RNF3.3	Tiempo de respuesta promedio < 2 segundos por consulta.	RF3.2, RF3.3	Depende de consultas de consumo y visualización de gráficos.
RNF4.1	Disponibilidad mínima del 99,5% anual.	Todos los RF	Aplica de forma general al sistema completo.
RNF4.2	Copias de seguridad	RF2.3, RF3.2, RF4.1	Depende de los datos de consumo y visualización.





Commits

main		All users	All time
Commits on Sep 30, 2025			
Avance informe	RickCrispin committed 2 minutes ago	15ec370	
Avance informe finalizado	RickCrispin committed 3 minutes ago	0a7d567	
Caso de uso	Milagrosz authored 15 minutes ago	Verified c#2abc7	
Diagrama de Contexto	Milagrosz authored 16 minutes ago	Verified dceda35	
Avance diagrama de procesos	RickCrispin committed 18 minutes ago	d9fd911	
Diagramas de clases y ER	ki3mi committed 23 minutes ago	9a03229	

```
MINGW64:/c/Users/ki3mi/Documents/analisis-software/evidencias
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Clases Class diagram.png
create mode 100644 ER.png

ki3mi@DESKTOP-HN6V636 MINGW64 ~/Documents/analisis-software/evidencias (master)
$ git branch -M main

ki3mi@DESKTOP-HN6V636 MINGW64 ~/Documents/analisis-software/evidencias (main)
$ git remote add origin https://github.com/RickCrispin/AnalisisDeSoftware-ConsumoElectrico.git

ki3mi@DESKTOP-HN6V636 MINGW64 ~/Documents/analisis-software/evidencias (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 66.43 KiB | 33.22 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/RickCrispin/AnalisisDeSoftware-ConsumoElectrico.git
* [new branch]      main -> main

ki3mi@DESKTOP-HN6V636 MINGW64 ~/Documents/analisis-software/evidencias (main)
$ |
```

RickCrispin / AnalisisDeSoftware-ConsumoElectrico

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Commits

main All users All time

Commits on Sep 30, 2025

Caso de uso	Verified	c72abc7	><
Diagrama de Contexto	Verified	dceda35	><
Avance diagrama de procesos		d9fd911	><
Diagramas de clases y ER		9a83229	><

- Pulse
- Contributors
- Community
- Community Standards
- Traffic
- Commits
- Code frequency
- Dependency graph
- Network
- Forks
- Actions Usage Metrics
- Actions Performance Metrics

Contributors

Contributions per week to main, excluding merge commits

Period: All Contributions: Commits

