



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE SOFTWARE

Periodo académico: 15 de septiembre 2025 – 13 de febrero 2026

Asignatura: Aplicaciones Informáticas II

Paralelo: Octavo “1”

Profesor: Ing. Julio Santillán

Tema: Historias técnicas

Fecha de Entrega: 14 de octubre de 2025

Grupo: Trabajo individual

Estudiante(s): Jeferson Charco (7321)

Historias Técnicas

1. Historia técnica H.T.01 – Monitoreo, Alta Disponibilidad y Escalabilidad del Sistema

Campo	Detalle
Título	Implementar monitoreo de uso de CPU y memoria.
Descripción	Se implementará una infraestructura que garantice rendimiento, disponibilidad y escalabilidad del sistema. Esto incluye la instalación de herramientas de monitoreo continuo (Prometheus y Grafana) , configuración de alta disponibilidad con balanceadores de carga, respaldos automáticos y réplicas de base de datos, y un diseño escalable que permita aumentar la capacidad horizontal o vertical sin modificar la arquitectura base. El objetivo es mantener un rendimiento óptimo, con uso de CPU $\leq 70\%$, RAM $\leq 80\%$, disponibilidad $\geq 99\%$, y respuesta estable ante el incremento de usuarios o transacciones.

Criterios de Aceptación	<ul style="list-style-type: none"> - El sistema mantiene una disponibilidad mensual $\geq 99\%$ - Los respaldos automáticos se ejecutan al menos 1 vez por día y pueden restaurarse correctamente. - Nuevos módulos pueden integrarse al sistema sin modificar el código base existente.
Tareas	<ul style="list-style-type: none"> - El sistema registra métricas de CPU y RAM en tiempo real mediante Prometheus y Grafana. - Bajo pruebas de carga de desarrollo (hasta 30 usuarios simulados), el uso de CPU no supera el 70% y la RAM no supera el 80%. - Los respaldos automáticos locales se generan y restauran exitosamente en pruebas de recuperación. - Nuevos módulos pueden integrarse sin fallos ni cambios en la estructura existente. - Se documenta la configuración de monitoreo, respaldos y simulación.
Pruebas	Pruebas de sistema bajo carga y validación de alertas de monitoreo.
Dependencias	RNF-1 – Capacidad de usuarios simultáneos RNF-2 – Rendimiento en procesamiento de transacciones RNF-3 – Disponibilidad del sistema RNF-4 – Escalabilidad
Estimación de Esfuerzo	5 horas.
Prioridad	Media.

2. Historia técnica H.T.02 – Implementación de autenticación segura y control de acceso por roles

Campo	Detalle
Título	Implementar autenticación segura y control de acceso basado en roles.
Descripción	Se debe implementar un sistema de autenticación con correo y contraseña, utilizando hashing criptográfico (bcrypt) para el almacenamiento seguro. El sistema restringirá el acceso a módulos

	según el rol del usuario (administrador, lector, cajero, usuario común) y controlará la gestión de sesiones, cerrándolas automáticamente tras 15 minutos de inactividad. También se debe impedir el acceso simultáneo desde varios dispositivos con la misma cuenta.
Criterios de Aceptación	<ul style="list-style-type: none"> - Solo usuarios autenticados pueden acceder al sistema. - Las contraseñas se almacenan cifradas (hash + salt). - Accesos restringidos correctamente según rol. - Sesión se cierra automáticamente tras 1 hora de uso del sistema.
Tareas	<ul style="list-style-type: none"> - Implementar autenticación segura con hashing. - Configurar middleware de control de roles. - Implementar cierre automático de sesión. - Probar diferentes roles y accesos.
Pruebas	Pruebas unitarias de login/logout, pruebas de integración de roles y pruebas de seguridad de sesión.
Dependencias	RNF 5 – Autenticación y control de acceso, RNF 8 – Gestión de sesiones.
Estimación de Esfuerzo	5 horas.
Prioridad	Alta.

3. Historia técnica H.T.03 – Cifrado de comunicaciones y protección de integridad de datos

Campo	Detalle
Título	Implementar cifrado de comunicaciones HTTPS y validaciones de integridad de datos.
Descripción	Todas las comunicaciones entre cliente y servidor deben realizarse bajo protocolo HTTPS (SSL/TLS). Además, se implementarán validaciones automáticas y restricciones de integridad (claves foráneas, índices únicos y triggers) para proteger la información

	crítica de usuarios, lecturas, facturas y pagos frente a alteraciones accidentales o maliciosas.
Criterios de Aceptación	<ul style="list-style-type: none"> -Todas las conexiones usan HTTPS - Se genera y configura un certificado SSL/TLS autofirmado con OpenSSL para entorno de desarrollo. - El backend (FastAPI) y frontend (React) se ejecuta correctamente con HTTPS. - Se documenta el procedimiento para migrar de certificado OpenSSL (desarrollo) a CA (producción).
Tareas	<ul style="list-style-type: none"> - Configurar HTTPS en el servidor. - Implementar redirección automática HTTP → HTTPS. - Aplicar restricciones de integridad en base de datos. - Implementar validaciones de entrada y consistencia. - Validar funcionalidad mediante pruebas controladas.
Pruebas	Pruebas de integración HTTPS, pruebas de base de datos y validación de integridad de registros.
Dependencias	RNF 6 – Cifrado de comunicaciones, RNF 7 – Integridad de la información.
Estimación de Esfuerzo	6 horas.
Prioridad	Alta.

4. Historia técnica H.T.04 – Protección ante errores y ataques comunes (seguridad del backend)

Campo	Detalle
Título	Implementar medidas de seguridad ante ataques comunes y manejo seguro de errores.
Descripción	Se deben implementar medidas de seguridad contra ataques SQL Injection, XSS, CSRF y otros. Esto incluye validación y escape de entradas en el servidor, uso de tokens de seguridad, y manejo

	controlado de errores para evitar revelar información sensible. Los mensajes de error deben ser genéricos y seguros.
Criterios de Aceptación	<ul style="list-style-type: none"> - El sistema bloquea intentos de inyección SQL y XSS. - Los formularios cuentan con tokens CSRF válidos. - No se revelan datos del sistema en mensajes de error. - Se valida toda entrada del usuario en servidor.
Tareas	<ul style="list-style-type: none"> - Implementar validaciones y sanitización de entradas. - Activar tokens CSRF en formularios. - Configurar manejo controlado de excepciones. - Ejecutar pruebas de seguridad (penetration testing).
Pruebas	Pruebas de penetración controladas (OWASP) y pruebas unitarias de validaciones.
Dependencias	RNF 9 – Seguridad ante errores y ataques.
Estimación de Esfuerzo	6 horas.
Prioridad	Alta.

5. Historia Técnica H.T.05 – Implementación de tolerancia y recuperación ante fallos

Campo	Detalle
Título:	Implementación de tolerancia y recuperación ante fallos
Descripción:	Se implementarán mecanismos automáticos de recuperación del sistema ante fallos leves del servidor o interrupciones temporales de red. Esto incluirá balanceo de carga, reinicio automático de servicios y retención de sesiones activas para evitar pérdida de información.
Criterios de Aceptación:	<ul style="list-style-type: none"> - Los servicios (backend, base de datos, caché) se reinician automáticamente cuando se detienen manualmente. - Las sesiones activas simuladas se restauran después del reinicio de servicios.

	- La integridad de datos se mantiene tras la simulación de fallo (no se pierden registros de prueba).
Tareas:	<ul style="list-style-type: none"> - Implementar scripts de reinicio automático de servicios (backend, base de datos, caché). - Programar reconexión automática a base de datos y cache. - Configurar retención de sesiones en memoria o en almacenamiento temporal. - Documentar pasos para simular y verificar fallos en desarrollo.
Pruebas:	<ul style="list-style-type: none"> - Simulación de caída de servidor o servicios locales. - Reinicio automático de servicios y verificación de reconexión. - Verificación de restauración de sesiones activas. - Comprobación de integridad de datos de prueba. - Medición y registro del tiempo de recuperación.
Dependencias:	RNF 10 – Tolerancia a fallos
Estimación de Esfuerzo:	4 horas
Prioridad:	Alta

6. Historia Técnica H.T.06 – Automatización de respaldos y restauración

Campo	Detalle
Título:	Automatización de respaldos y restauración de datos
Descripción:	Se desarrollará un sistema automático de copias de seguridad de la base de datos que se ejecute diariamente, con almacenamiento seguro (cifrado) y capacidad de restauración controlada en caso de pérdida o corrupción de datos.
Criterios de Aceptación:	<ul style="list-style-type: none"> - Se realizan respaldos automáticos al menos una vez al día. - Los archivos de respaldo están cifrados y almacenados de forma segura. - La restauración se realiza correctamente y sin pérdida de datos.

Tareas:	<ul style="list-style-type: none"> - Configurar tareas automáticas (cron job) para generar copias de seguridad. - Implementar cifrado de respaldo y almacenamiento en servidor seguro o nube. - Crear script de restauración con validación de integridad. - Documentar el proceso.
Pruebas:	<ul style="list-style-type: none"> - Prueba de ejecución automática diaria. - Prueba de restauración completa y validación de integridad. - Verificación de cifrado en archivos de respaldo.
Dependencias:	RNF 11 – Respaldo y restauración de datos
Estimación de Esfuerzo:	4 horas
Prioridad:	Alta

7. Historia Técnica H.T.07 – Implementación de accesibilidad remota 24/7

Campo	Detalle
Título:	Implementación de accesibilidad remota 24/7
Descripción:	Se configurará el sistema para garantizar su disponibilidad continua las 24 horas del día y los 7 días de la semana, accesible desde cualquier dispositivo con conexión a Internet mediante navegadores web modernos.
Criterios de Aceptación:	<ul style="list-style-type: none"> - El sistema puede ser accedido correctamente desde PC, laptop, tablet y smartphone. - No existen restricciones de acceso fuera del horario laboral. - Los usuarios pueden realizar operaciones normales en cualquier momento del día.
Tareas:	<ul style="list-style-type: none"> - Verificar compatibilidad multiplataforma y responsividad. - Optimizar configuración del servidor web. - Configurar monitoreo de uptime continuo. - Validar acceso remoto desde diferentes redes.

Pruebas:	<ul style="list-style-type: none"> - Pruebas de acceso remoto desde distintos dispositivos. - Pruebas de conexión ininterrumpida por 24 horas. - Validación de rendimiento fuera del horario laboral.
Dependencias:	RNF-13 – Accesibilidad remota
Estimación de Esfuerzo:	4 horas
Prioridad:	Alta

8. Historia Técnica H.T.08 – Sistema de notificación de mantenimiento programado

Campo	Detalle
Título:	Sistema de notificación de mantenimiento programado
Descripción:	Se notificará con al menos 24 horas de anticipación cualquier mantenimiento planificado que afecte la disponibilidad del sistema. Las notificaciones serán visibles en el panel del usuario y por correo electrónico.
Criterios de Aceptación:	<ul style="list-style-type: none"> - Las notificaciones de mantenimiento se envían con un mínimo de 24 horas de anticipación. - Los usuarios reciben la notificación dentro del sistema y por correo electrónico. - Los registros de mantenimiento quedan almacenados en logs.
Tareas:	<ul style="list-style-type: none"> - Implementar módulo de notificaciones en interfaz de usuario. - Configurar envío automático de correos de alerta. - Registrar y auditar mantenimientos programados.
Pruebas:	<ul style="list-style-type: none"> - Pruebas de envío de notificaciones automáticas. - Verificación de recepción de correo electrónico. - Revisión de registro de logs de mantenimiento.
Dependencias:	RNF-14 – Notificación de mantenimiento programado, RNF-12 – Porcentaje de disponibilidad del sistema

Estimación de Esfuerzo:	5 horas
Prioridad:	Media

9. Historia Técnica H.T.09 – Diseño modular y mantenimiento del sistema

Campo	Detalle
Título:	Implementación de arquitectura modular y mantenimiento del sistema.
Descripción:	El sistema será diseñado con una arquitectura modular , de modo que cada componente (frontend, backend y base de datos) pueda actualizarse, corregirse o mejorarse sin afectar el funcionamiento general. Esta historia incluye la configuración de buenas prácticas de versionado, la aplicación de mantenimiento correctivo y preventivo, y la validación de integridad de datos después de cada actualización.
Criterios de Aceptación:	<ul style="list-style-type: none"> - El sistema cuenta con módulos independientes (frontend, backend, base de datos) ejecutables por separado. - Existe un repositorio Git donde se puede ver una rama principal (main) y una confirmación (commit) con la estructura modular inicial. - Se ejecuta un procedimiento de mantenimiento preventivo (limpieza de logs, comprobación de endpoints y revisión de base de datos). - Se documenta un plan de mantenimiento preventivo en un documento técnico.
Tareas:	<p>Diseñar estructura modular de carpetas (frontend, backend, db).</p> <p>Implementar control de versiones con Git y repositorio en git hub.</p> <p>Definir y documentar procedimientos de actualización de dependencias.</p> <p>Crear script de mantenimiento preventivo (limpieza de logs, verificación de DB).</p> <p>Documentar evidencias y plan de mantenimiento preventivo.</p>

Pruebas:	<ul style="list-style-type: none"> - Pruebas unitarias de corrección de errores. - Validación de integridad de base de datos post-corrección. - Verificación de registros en historial de mantenimiento.
Dependencias:	RNF-15 – Facilidad de actualización del sistema RNF-16 – Mantenimiento correctivo RNF-17 – Mantenimiento preventivo
Estimación de Esfuerzo:	8 horas
Prioridad:	Alta

10. Historia Técnica H.T.10 – configuración del entorno de desarrollo con el uso de tecnologías portable y multiplataforma.

Campo	Detalle
Título:	Implementación con tecnologías portables y multiplataforma
Descripción:	<p>El sistema será desarrollado utilizando un stack tecnológico multiplataforma y de código abierto, que incluye React para el frontend, FastAPI para el backend y PostgreSQL como base de datos.</p> <p>En este punto se preparará el entorno de desarrollo, la instalación de dependencias, la configuración de conexión entre los servicios y la verificación del entorno operativo para garantizar la portabilidad, mantenimiento y escalabilidad del software.</p>
Criterios de Aceptación:	<ul style="list-style-type: none"> -El proyecto React se ejecuta correctamente en el entorno local. - El servidor FastAPI se ejecuta correctamente con sin errores. - El frontend puede comunicarse con el backend a través de una API. - La base de datos se conecta correctamente desde el backend. -Se cuenta con documentación en un archivo README.md.
Tareas:	<ul style="list-style-type: none"> - Configuración del entorno de desarrollo Instalar Node.js, Python 3.10+ y PostgreSQL - Crear proyecto con FastAPI y estructura base e instalar dependencias. - Crear proyecto React enstalar dependencias iniciales.

	- Crear README.md con pasos de instalación, comandos y dependencias.
Pruebas:	<ul style="list-style-type: none"> - Pruebas de ejecución del frontend (npm start o npm run dev). - Pruebas de ejecución del backend (uvicorn main:app --reload). - Pruebas de conexión a la base de datos y creación de tablas. - Pruebas de integración entre frontend y backend (API de prueba).
Dependencias:	RNF-19 – Uso de tecnologías portables, RNF-20 compatibilidad con navegador y RNF-18 – Compatibilidad multiplataforma
Estimación de Esfuerzo:	6 horas
Prioridad:	Alta

11. Historia Técnica HT-11 – Diseño del modelo de base de datos (PostgreSQL)

Campo	Detalle
Título	Diseño del modelo de base de datos PostgreSQL
Descripción	Se diseñará un modelo de base de datos relacional que permita almacenar de manera eficiente la información de usuarios, medidores, facturación, multas, pagos y registros históricos, asegurando integridad y facilidad de consulta.
Criterios de Aceptación	<ul style="list-style-type: none"> - Modelo de base de datos documentado - Tablas y relaciones reflejan los requisitos funcionales. - El diagrama entidad-relación (ER) debe incluir todas las entidades y relaciones identificadas en los requerimientos funcionales (usuarios, medidores, facturación, multas, pagos).
Tareas	<ul style="list-style-type: none"> - Definir entidades y atributos principales. - Diseñar diagramas ER (Entidad-Relación). - Revisar consistencia con requerimientos. - Documentar el modelo final.

	<ul style="list-style-type: none"> - Normalizar tablas y relaciones. - Crear diagramas de relaciones finalizados. - Revisar restricciones y reglas de integridad.
Pruebas	<ul style="list-style-type: none"> - Validación del modelo con consultas de prueba. - Verificación de consistencia de relaciones.
Dependencias	RNF-19 Uso de tecnologías portables
Estimación de Esfuerzo	5 horas
Prioridad	Alta

12. Historia Técnica HT-12 – Integración del backend con PostgreSQL

Campo	Detalle
Título	Conexión del backend con la base de datos PostgreSQL
Descripción	El backend en Python se integrará con PostgreSQL usando un ORM para permitir consultas, inserciones y actualizaciones seguras.
Criterios de Aceptación	<ul style="list-style-type: none"> - El backend puede conectarse correctamente a PostgreSQL usando las credenciales configuradas en el entorno (.env). - La configuración de la base de datos debe estar documentada en el repositorio (README). - Errores de conexión y transacciones manejados correctamente.
Tareas	<ul style="list-style-type: none"> - Configurar conexión al servidor PostgreSQL. - Implementar modelos ORM para tablas existentes. - Crear pruebas unitarias de operaciones CRUD. - Documentar configuración y acceso a la base de datos.
Pruebas	<ul style="list-style-type: none"> - Pruebas de conexión y transacciones. - Validación de consultas y registros. - Verificación de manejo de errores.
Dependencias	RNF-21 – Diseño y compatibilidad de la base de datos

Estimación de Esfuerzo	4 horas
Prioridad	Alta