



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 – Optimización para usuarios simultáneos

| Campo | Detalle |
|--------------------------------|---|
| Título | Optimizar el sistema para soportar al menos 50 usuarios concurrentes sin degradación perceptible. |
| Descripción | Configurar y optimizar el servidor de aplicaciones, base de datos y caché para soportar un mínimo de 50 usuarios simultáneos. Realizar pruebas de carga y ajuste de parámetros (threads, conexiones, caché y tiempo de respuesta) para garantizar estabilidad y buen rendimiento. |
| Criterios de Aceptación | <ul style="list-style-type: none"> - Soporta 50 usuarios simultáneos sin errores ni caídas. - Tiempo de respuesta promedio menor o igual a 4 segundos. - El sistema mantiene estabilidad durante las pruebas de carga. |
| Tareas | <ul style="list-style-type: none"> - Configurar servidor y base de datos. - Implementar sistema de caché. |

| | |
|-------------------------------|---|
| | <ul style="list-style-type: none"> - Realizar pruebas de carga (JMeter o similar). - Ajustar parámetros hasta alcanzar indicadores. |
| Pruebas | Pruebas de rendimiento y estrés del sistema en entorno de prueba controlado. |
| Dependencias | Relacionada con RNF 2 (procesamiento de transacciones) y RNF 3 (rendimiento del servidor). |
| Estimación de Esfuerzo | 18 horas. |
| Prioridad | Alta. |

2. Historia técnica H.T.02 – Monitoreo y control de recursos del servidor

| Campo | Detalle |
|--------------------------------|--|
| Título | Implementar monitoreo de uso de CPU y memoria. |
| Descripción | Implementar un sistema de monitoreo continuo con Grafana, Prometheus para asegurar que el uso de CPU sea <70% y la RAM <80% bajo operación normal. Configurar alertas ante sobreuso. |
| Criterios de Aceptación | <ul style="list-style-type: none"> - CPU \leq70% y RAM \leq80% durante operaciones normales. - Se generan alertas al superar límites. - No hay degradación visible del rendimiento. |
| Tareas | <ul style="list-style-type: none"> - Configurar herramientas de monitoreo. - Analizar procesos de alto consumo. - Optimizar código o consultas pesadas. - Validar resultados mediante métricas del servidor. |
| Pruebas | Pruebas de sistema bajo carga y validación de alertas de monitoreo. |
| Dependencias | RNF 1 (usuarios simultáneos) |
| Estimación de Esfuerzo | 12 horas. |
| Prioridad | Media. |

3. Historia técnica H.T.03 – Alta disponibilidad del sistema

| Campo | Detalle |
|--------------------------------|---|
| Título | Configurar alta disponibilidad con 99% de tiempo operativo. |
| Descripción | Implementar balanceadores de carga, réplicas de base de datos y respaldos automáticos que aseguren 99% de disponibilidad mensual. Gestionar mantenimientos programados sin afectar la disponibilidad. |
| Criterios de Aceptación | <ul style="list-style-type: none"> - 99% de disponibilidad mensual. - Mantenimientos sin interrupciones. - Reportes automáticos de uptime. |
| Tareas | <ul style="list-style-type: none"> - Configurar respaldos automáticos. - Validar uptime con herramientas. - Documentar plan de mantenimiento. |
| Pruebas | Pruebas de recuperación ante fallos (failover testing) y simulaciones de interrupciones. |
| Dependencias | RNF 3 (rendimiento del servidor) y RNF 5 (escalabilidad). RNF-16 – Porcentaje de disponibilidad del sistema y RNF-17 – Tolerancia a fallos del servidor |
| Estimación de Esfuerzo | 10 horas. |
| Prioridad | Alta. |

4. Historia técnica H.T.04 – Escalabilidad del sistema

| Campo | Detalle |
|--------------------------------|---|
| Título | Diseñar arquitectura escalable sin necesidad de rediseño estructural. |
| Descripción | Garantizar que el sistema escale horizontal o verticalmente sin modificar la arquitectura. Implementar diseño modular, microservicios o contenedores y configurar autoescalamiento en la nube. |
| Criterios de Aceptación | <ul style="list-style-type: none"> - La infraestructura soporta aumento de usuarios y transacciones. - Nuevos módulos pueden integrarse sin afectar funcionamiento actual. - Pruebas de escalabilidad superadas sin errores. |

| | |
|-------------------------------|--|
| Tareas | <ul style="list-style-type: none"> - Analizar arquitectura actual. - Implementar diseño modular. - Configurar escalado automático. - Realizar pruebas de carga progresiva. |
| Pruebas | Pruebas de escalabilidad y rendimiento progresivo con escenarios simulados. |
| Dependencias | RNF 4 (disponibilidad) y RNF 1 (usuarios simultáneos). |
| Estimación de Esfuerzo | 4 horas. |
| Prioridad | Alta. |

5. Historia técnica H.T.05 – 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 15 minutos de inactividad. - No se permite inicio simultáneo en más de dos dispositivos. |
| Tareas | <ul style="list-style-type: none"> - Implementar autenticación segura con hashing. - Configurar middleware de control de roles. - Implementar cierre automático de sesión. |

| | |
|-------------------------------|--|
| | <ul style="list-style-type: none"> - Validar no duplicidad de sesiones activas. - Probar diferentes roles y accesos. |
| Pruebas | Pruebas unitarias de login/logout, pruebas de integración de roles y pruebas de seguridad de sesión. |
| Dependencias | RNF 6 – Autenticación y control de acceso, RNF 10 – Gestión de sesiones. |
| Estimación de Esfuerzo | 5 horas. |
| Prioridad | Alta. |

6. Historia técnica H.T.06 – 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 (sin contenido mixto). - El certificado SSL/TLS es válido y actualizado. - La base de datos implementa restricciones y triggers activos. - La información se valida antes de ser almacenada o modificada. |
| 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 7 – Cifrado de comunicaciones, RNF 9 – Integridad de la información. |
| Estimación de Esfuerzo | 6 horas. |
| Prioridad | Alta. |

7. Historia técnica H.T.07 – Registro de auditoría y trazabilidad del sistema

| Campo | Detalle |
|--------------------------------|--|
| Título | Implementar módulo de auditoría y trazabilidad de actividades críticas. |
| Descripción | Se debe crear un sistema de logs que registre automáticamente las actividades críticas: inicios y cierres de sesión, creación/modificación/eliminación de usuarios, y todas las transacciones de facturación, pagos y cambios de medidor. Los registros deben incluir usuario, acción, fecha/hora e IP, y almacenarse de forma segura. |
| Criterios de Aceptación | <ul style="list-style-type: none"> - Todas las acciones críticas generan un log automático. - Los logs incluyen usuario, acción, hora e IP. - Los registros se almacenan de forma inalterable. - Se puede consultar la trazabilidad completa de un evento. |
| Tareas | <ul style="list-style-type: none"> - Diseñar estructura de la tabla de auditoría. - Implementar middleware de registro automático. - Configurar rotación y almacenamiento seguro de logs. - Crear módulo de consulta de auditoría. |
| Pruebas | Pruebas unitarias y de integración para verificar generación de logs en todos los casos definidos. |
| Dependencias | RNF 8 – Registro de auditoría y trazabilidad. |
| Estimación de Esfuerzo | 5 horas. |

| | |
|------------------|--------|
| Prioridad | Media. |
|------------------|--------|

8. Historia técnica H.T.08 – 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 11 – Seguridad ante errores y ataques. |
| Estimación de Esfuerzo | 6 horas. |
| Prioridad | Alta. |

9. Historia Técnica H.T.09 – 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"> - El sistema se recupera automáticamente sin pérdida de datos ante fallos leves. - El tiempo máximo de recuperación no supera los 5 minutos. - Las sesiones activas se restauran correctamente después de un fallo. |
| Tareas: | <ul style="list-style-type: none"> - Implementar scripts de reinicio automático. - Programar reconexión automática de base de datos y caché. - Verificar retención de sesión en memoria. |
| Pruebas: | <ul style="list-style-type: none"> - Pruebas de caída del servidor (simulación de fallo). - Pruebas de reconexión automática y recuperación de servicios. - Medición del tiempo de recuperación. - Verificación de integridad de datos post-fallo. |
| Dependencias: | RNF 12 – Tolerancia a fallos y RNF 13 – Recuperación ante fallos. |
| Estimación de Esfuerzo: | 3 horas |
| Prioridad: | Alta |

10. Historia Técnica H.T.10 – 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 |

| | |
|--------------------------------|---|
| | <p>segura.</p> <ul style="list-style-type: none"> - 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 14 – Respaldo y restauración de datos |
| Estimación de Esfuerzo: | 4 horas |
| Prioridad: | Alta |

11. Historia Técnica H.T.11 – 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. |

| | |
|--------------------------------|--|
| | <ul style="list-style-type: none"> - 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-18 – Accesibilidad remota |
| Estimación de Esfuerzo: | 4 horas |
| Prioridad: | Alta |

12. Historia Técnica H.T.12 – 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-19 – Notificación de mantenimiento programado |

| | |
|--------------------------------|---------|
| Estimación de Esfuerzo: | 5 horas |
| Prioridad: | Media |

13. Historia Técnica H.T.13 – Diseño modular para facilidad de actualización

| Campo | Detalle |
|---------------------------------|--|
| Título: | Diseño modular para facilidad de actualización del sistema |
| Descripción: | El sistema será desarrollado con una arquitectura modular, permitiendo que nuevas funcionalidades o actualizaciones de software se integren sin afectar el funcionamiento general del sistema. Esto facilita la escalabilidad y reduce riesgos de errores durante actualizaciones. |
| Criterios de Aceptación: | <ul style="list-style-type: none"> - Las actualizaciones pueden implementarse sin interrupciones significativas del sistema. - Los módulos se pueden actualizar de forma independiente. - No se producen errores en módulos no actualizados. |
| Tareas: | <ul style="list-style-type: none"> - Rediseñar o validar la arquitectura modular. - Identificar dependencias entre módulos. - Documentar procedimientos de actualización modular. - Probar actualizaciones en entorno controlado. |
| Pruebas: | <ul style="list-style-type: none"> - Pruebas de actualización en entorno de prueba. - Validación de que módulos no afectados funcionan correctamente. - Verificación de integración post-actualización. |
| Dependencias: | RNF-21 – Facilidad de actualización del sistema |
| Estimación de Esfuerzo: | 4 horas |
| Prioridad: | Alta |

14. Historia Técnica H.T.14 – Mantenimiento correctivo

| Campo | Detalle |
|---------------------------------|---|
| Título: | Implementación de mantenimiento correctivo seguro y plan de mantenimiento preventivo del sistema. |
| Descripción: | Se habilitarán procesos y herramientas para corregir errores detectados durante el uso del sistema sin afectar los datos almacenados. Se establecerán procedimientos de rollback y validación de integridad tras correcciones. |
| Criterios de Aceptación: | <ul style="list-style-type: none"> - Los errores se corrigen sin pérdida de datos. - El sistema permanece estable tras aplicar correcciones. - Las correcciones quedan registradas en el historial de mantenimiento. |
| Tareas: | <ul style="list-style-type: none"> - Configurar entorno de pruebas para correcciones. - Implementar procesos de rollback y backup antes de cambios. - Documentar cada corrección aplicada. - Definir cronograma de mantenimiento preventivo. - Crear scripts de limpieza y optimización. - Configurar alertas de rendimiento. |
| 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-22 – Mantenimiento correctivo, RNF-23 – Mantenimiento preventivo. |
| Estimación de Esfuerzo: | 8 horas |
| Prioridad: | Alta |

15. Historia Técnica H.T.15 – Compatibilidad multiplataforma

| Campo | Detalle |
|----------------|---|
| Título: | Garantizar compatibilidad multiplataforma |

| | |
|---------------------------------|---|
| Descripción: | El sistema será desarrollado para ejecutarse en diferentes plataformas (Windows, Linux, macOS) sin requerir cambios significativos en el código fuente. Se utilizarán tecnologías web estandarizadas y frameworks portables para asegurar que la funcionalidad se mantenga en todos los entornos. |
| Criterios de Aceptación: | <ul style="list-style-type: none"> - El sistema se ejecuta correctamente en Windows, Linux y macOS. - No se requieren modificaciones en el código fuente para cada plataforma. - Todas las funciones principales funcionan en cada sistema operativo. |
| Tareas: | <ul style="list-style-type: none"> - Configurar entornos de prueba para cada sistema operativo. - Verificar compatibilidad del código y librerías utilizadas. - Ajustar configuraciones de deployment multiplataforma. - Documentar procedimientos de ejecución por plataforma. |
| Pruebas: | <ul style="list-style-type: none"> - Pruebas funcionales completas en cada sistema operativo. - Validación de instalación y ejecución sin errores. - Verificación de consistencia de la interfaz y funcionalidades. |
| Dependencias: | RNF-26 – Compatibilidad multiplataforma, RNF-28 – Compatibilidad con navegadores |
| Estimación de Esfuerzo: | 4 horas |
| Prioridad: | Alta |

16. Historia Técnica H.T.16 – Uso de tecnologías portables y multiplataforma

| Campo | Detalle |
|---------------------|---|
| Título: | Implementación con tecnologías portables y multiplataforma |
| Descripción: | El sistema será desarrollado utilizando tecnologías multiplataforma y de código abierto, incluyendo React para frontend, FastAPI para backend y PostgreSQL como base de datos, garantizando portabilidad, mantenimiento y escalabilidad del software. |

| | |
|---------------------------------|--|
| Criterios de Aceptación: | <ul style="list-style-type: none"> - El sistema utiliza React, FastAPI y PostgreSQL en su arquitectura. - La aplicación puede desplegarse en cualquier entorno compatible con estas tecnologías. - Se cumplen estándares de código abierto y multiplataforma. |
| Tareas: | <ul style="list-style-type: none"> - Configurar proyectos con React y FastAPI. - Configurar base de datos PostgreSQL multiplataforma. - Validar que los módulos y dependencias sean compatibles. - Documentar stack tecnológico y guías de despliegue. |
| Pruebas: | <ul style="list-style-type: none"> - Pruebas de despliegue en entornos Windows, Linux y macOS. - Pruebas de integración frontend-backend. - Validación de acceso y operaciones de base de datos. |
| Dependencias: | RNF-27 – Uso de tecnologías portables |
| Estimación de Esfuerzo: | 4 horas |
| Prioridad: | Alta |

17. Historia Técnica HT-17 – 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 y aprobado. - Tablas y relaciones reflejan los requerimientos funcionales. - Permite escalabilidad y mantenimiento. |
| 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 |
| Estimación de Esfuerzo | 5 horas |
| Prioridad | Alta |

18. Historia Técnica HT-18 – 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. - CRUD funcional para entidades principales. - 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 | HT-23 |
| Estimación de Esfuerzo | 3 horas |
| Prioridad | Alta |