

Rúbrica para Avance: Sistema Contable (Libro Diario, Mayor y Login)

Descripción: Esta rúbrica evalúa el avance de un sistema contable desarrollado en Java (Java SE, Java with Spring Boot o equivalente), con base de datos PostgreSQL. El avance debe incluir: implementación funcional del login, registro de operaciones en libro diario, movimientos en libro mayor, esquema de base de datos, documentación técnica y evidencia (capturas, mensajes de prueba, scripts). Puntaje total: 100 puntos.

Criterio	Puntaje Máx.	Excelente	Bueno	Aceptable	Insuficiente
Funcionalidad: Login (autenticación y manejo de sesiones)	20	Login seguro con validación, hashing de contraseñas, manejo correcto de sesiones y roles.	Login funcional con hashing, manejo básico de sesiones.	Login parcialmente funcional; autenticación no robusta.	Login ausente o no funcional.
Funcionalidad: Libro Diario (registro y validación de asientos)	25	Registro completo de asientos contables, validaciones (balance débitos/créditos), búsqueda y filtrado.	Registro e inserción de asientos con validaciones básicas.	Registro parcial con falta de validaciones.	Funcionalidad ausente o no registra asientos de manera confiable.
Funcionalidad: Libro Mayor (mayorización de saldos por cuenta)	25	Generación correcta del mayor con saldos iniciales, movimientos, saldos finales por periodo.	Mayor funcional que muestra movimientos y saldos con limitaciones.	Mayor parcial, cálculos incorrectos en algunos casos.	No hay mayor implementado o es incorrecto.
Diseño de la Base de Datos (esquema, integridad, normalización)	10	Modelo ER claro, tablas normalizadas, claves foráneas, índices para consultas críticas.	Modelo consistente con claves y normalización básica.	Modelo incompleto o con problemas de integridad.	Esquema pobre, sin integridad referencial.
Calidad del Código (estructura, patrones, legibilidad, comentarios)	10	Código organizado, uso apropiado de patrones, capas (DAO/Service/Controller), comentarios útiles.	Código razonablemente organizado; algunas buenas prácticas aplicadas.	Código desordenado; mezcla de responsabilidades.	Código no legible, sin estructura ni comentarios.
Seguridad y Buenas Prácticas (gestión de contraseñas, inyección SQL, validaciones)	5	Medidas de seguridad aplicadas: hashing, prepared statements/ORM, validación de entradas.	Algunas medidas de seguridad aplicadas; puntos críticos cubiertos.	Medidas de seguridad insuficientes; riesgos conocidos sin mitigar.	Ausencia de consideraciones de seguridad.
Documentación y Entrega (README, guía de instalación, diagramas ER y de flujo)	3	README completo, pasos para ejecutar, diagramas incluidos.	README presente con instrucciones básicas.	Documentación parcial o poco clara.	No hay documentación relevante.
Interfaz y Usabilidad (aplicación: web o escritorio)	2	Interfaz clara, flujo de usuario coherente y mensajes de error útiles.	Interfaz funcional con algunos problemas de usabilidad.	Interfaz cruda o difícil de usar.	Sin interfaz o inusable.

Escala de Puntuación:
Excelente: 90-100% del puntaje asignado para el criterio.
Bueno: 75-89%.
Aceptable: 60-74%.
Insuficiente: <60%.

Lista de verificación para entrega del avance:

1. Código fuente subido a repositorio (enlace y commit ID).
2. Script de creación de base de datos (PostgreSQL) y datos de ejemplo.
3. Capturas de pantalla o demo corto que muestre login, registro de asiento y vista del mayor.
4. README con pasos para ejecutar (requisitos, comandos, puerto, variables de entorno).
5. Diagramas: ER y diagrama de clases (puede ser imagen o PDF adicional).

Notas para el evaluador:

- Si alguna funcionalidad no aplica por decisión del equipo, debe justificarse en el README y en la presentación del avance.
- Para sistemas con Spring Boot, se espera la configuración de application.properties o application.yml con perfiles (dev/test) y ejemplo de variables de entorno sin credenciales reales.
- La seguridad básica (hash de contraseñas y uso de prepared statements o JPA/Hibernate) es obligatoria salvo que se documente una alternativa segura.