

PAUTAS DE CORRECCION DE TPI

Reglas excluyentes (antes de puntuar)

- Sin video calificación final = 0.
- Repositorio no público o sin SQL de creación / datos de prueba: 0.
- No hay relación 1→1 real (en código y BD): 0.

Rúbrica con puntajes por nivel

Rúbrica de evaluación (detalle por nivel y puntajes)

#	Criterio (máx.)	Excelente	Adecuado	Básico	Insuficiente
1	Video: presentación + demo + explicación (30)	10–15 min; 4 a cámara; demo CRUD completo (A y B); explica capas (entities/DAO/Service/AppMenu); muestra rollback con fallo simulado. → 30 pts	9–16 min; 3–4 a cámara; demo casi completa; explicación parcial de capas; rollback mencionado o poco claro. → 21 pts	<9 o >16; 1–2 a cámara; demo incompleta ; no se evidencia rollback. → 12 pts	Video ausente/inusable. → 0 pts
2	Correctitud funcional (CRUD + 1→1 + integridad) (20)	CRUD completo en A y B (incluye eliminar lógico); 1→1 unidireccional efectiva (A→B); unicidad (no >1 B por A); búsquedas por campo clave. → 20 pts	CRUD casi completo; 1→1 correcto con bordes ; pocos errores. → 14 pts	CRUD incompleto; 1→1 frágil ; errores frecuentes. → 8 pts	CRUD falla; 1→1 no se cumple . → 0 pts
3	Transacciones (commit/rollback) (10)	<code>setAutoCommit(false)</code> ; misma Connection entre DAO; commit al final; rollback ante	Manejo transaccional correcto con mínimos descuidos. → 7 pts	Uso parcial/inconsistente ; operaciones sin cobertura transaccional. → 4 pts	Sin transacciones / mal uso. → 0 pts

TECNICATURA UNIVERSITARIA
EN PROGRAMACIÓN
A DISTANCIA



#	Criterio (máx.)	Excelente	Adecuado	Básico	Insuficiente
		excepción; orden consistente (crear B → asociar → crear A). → 10 pts			
4	Arquitectura (DAO/Service/capas) (10)	Capas claras : DAO = persistencia (PreparedStatement), Service = validaciones + transacciones , AppMenu = interacción; excepciones bien ubicadas. → 10 pts	Pequeñas fugas entre capas. → 7 pts	Acoplamientos visibles; lógica mezclada. → 4 pts	Desorden; sin patrón DAO real. → 0 pts
5	Reproducibilidad: Repo + README + SQL (10)	Repo público ; README con requisitos, pasos de creación de base (SQL), ejecución, datos de prueba y enlace al video ; corre desde cero . → 10 pts	README usable ; falta algún detalle menor. → 7 pts	README escaso ; scripts incompletos . → 4 pts	Repo privado/inaccesible o sin SQL/README. → 0 pts
6	AppMenu & manejo de errores (5)	Menú claro ; validaciones de entrada (números/formatos); mensajes de éxito/error ; casos borde contemplados. → 5 pts	Menú usable con pequeñas fallas. → 3 pts	Menú confuso ; validaciones débiles . → 2 pts	No implementado / inusable. → 0 pts
7	Calidad de código (5)	Nombres significativos ; formato consistente ; <code>toString()</code> útiles (sin recursión); poca duplicación; comentarios justos . → 5 pts	Legible con detalles a mejorar. → 3 pts	Estilo irregular ; varias malas practicas. → 2 pts	Desprolijo; difícil de mantener. → 0 pts
8	Trabajo en equipo & Git (5)	Commits repartidos entre los 4; mensajes claros; ramas/PRs o tag de entrega. → 5 pts	Participación mayoritaria de 3; historial razonable. → 3 pts	1–2 concentran la mayoría; pocos commits. → 2 pts	Evidencia de trabajo individual / push final único. → 0 pts

#	Criterio (máx.)	Excelente	Adecuado	Básico	Insuficiente
9	UML coherente con código (3)	Diagrama actualizado y consistente con clases/relaciones finales (1→1 unidireccional). → 3 pts	Diferencias menores . → 2 pts	Varias inconsistencias . → 1 pt	Inexistente / irreconocible. → 0 pts
10	Informe (PDF) (2)	Breve (6–8 págs.), claro; decisiones de diseño; transacciones y pruebas; cita fuentes/herramientas (incl. IA). → 2 pts	Correcto pero superficial o extenso. → 1 pt	Incompleto / no aporta. → 0 pts	Falta. → 0 pts

Guías para cada nivel (síntesis)

- **Video** : 10–15 min, 4 a cámara, CRUD completo, capas explicadas y **rollback mostrado** (Excelente). Menos cobertura o sin demo de rollback baja a Adecuado/Básico.
- **Correctitud**: CRUD completo A y B, 1→1 efectivo, baja lógica y búsquedas → Excelente; faltantes degradan.
- **Transacciones**: `autocommit=false`, operaciones atómicas `A↔B`, `commit/rollback` correctos, misma `Connection` compartida.
- **Arquitectura**: responsabilidades claras (validaciones en Service; DAO sólo persistencia).
- **Repo/README/SQL**: corre “desde cero” con scripts de **creación** y **datos de prueba**, y pasos claros.
- **AppMenu**: validaciones de entrada, mensajes claros, casos borde.
- **Código**: nombres, formato, `toString`, sin duplicidad evidente.
- **Equipo & Git**: commits repartidos, mensajes claros.
- **UML**: consistente con lo entregado.
- **Informe**: breve, claro, incluye decisiones y transacciones.

Penalizaciones (acumulan)

- No demostrar **rollback** en el video: **-8 pts.**
- No ejecutar el proyecto con los **scripts SQL** provistos (creación y datos): **-10 pts.**
- Sin **baja lógica** implementada: **-3 pts.**
- **Plagio / copia: 0 final.**

Bonus (hasta +5 pts, sin exceder 100)

- Script único para recrear todo (crear BD + datos + ejecutar): **+3**
- Logging útil y filtros de entrada robustos: **+2**