

Proyecto 4

Objetivo General

Diseñar e implementar un sistema completo que integre:

- Modelado de datos y normalización.
- Acceso a datos mediante ORM.
- Validaciones mediante restricciones, triggers y funciones SQL.
- Tipos de datos personalizados.
- Vistas SQL y reportería con múltiples filtros.
- Exportación de datos.
- Evaluación grupal con reflexión crítica y coevaluación.

Requerimientos Generales

- Trabajo en grupos de 5 personas.
- La nota será individual, determinada por una coevaluación y autoevaluación.
- Deben usar exclusivamente un ORM (Eloquent, SQLAlchemy, Sequelize, Prisma, etc.).
- No se permite el uso de SQL crudo en la aplicación.

Especificaciones Técnicas

Modelo de Datos

- Al menos 20 tablas.
- Mínimo 3 tablas de cruce (N:M).
- Relaciones 1:N y N:M, atributos multivaluados o derivados.
- Normalización hasta 3FN.
- 5 tipos de datos personalizados.
- Entregar diagrama E-R con cardinalidades y claves.

Implementación CRUD

- Al menos 3 CRUDs completos.
- Cada uno debe permitir crear, ver, actualizar y eliminar registros.
- CRUDs deben tener vistas de índice basadas en una VIEW SQL.

Validaciones y Reglas

- CHECK, NOT NULL, UNIQUE, DEFAULT.
- Mínimo 3 triggers funcionales.
- Mínimo 3 funciones SQL definidas por el usuario.
- Validaciones tanto en la aplicación como en la base de datos.

Vistas SQL

- Al menos 2 vistas (VIEW).
- Cada vista debe facilitar la visualización de datos complejos.

Datos de Prueba

- Al menos 1000 registros distribuidos entre las tablas.
- Datos coherentes y variados.

Reportes y Filtros

- Desarrollar 3 reportes distintos.
- Cada uno debe tener al menos 5 filtros significativos.
- Exportación obligatoria a CSV.
- Visualización clara y organizada de los resultados.

Documentación

- README.md con descripción del sistema e instrucciones.
- Scripts SQL: schema.sql y data.sql.
- Diagrama E-R en imagen o PDF.

Reflexión

El grupo debe entregar una reflexión escrita que responda a las siguientes preguntas:

1. ¿Cuál fue el aporte técnico de cada miembro del equipo?
2. ¿Qué decisiones estructurales se tomaron en el modelo de datos y por qué?
3. ¿Qué criterios siguieron para aplicar la normalización?
4. ¿Cómo estructuraron los tipos personalizados y para qué los usaron?
5. ¿Qué beneficios encontraron al usar vistas para el índice?
6. ¿Cómo se aseguraron de evitar duplicidad de datos?
7. ¿Qué reglas de negocio implementaron como restricciones y por qué?
8. ¿Qué trigger resultó más útil en el sistema? Justifica.
9. ¿Cuáles fueron las validaciones más complejas y cómo las resolvieron?
10. ¿Qué compromisos hicieron entre diseño ideal y rendimiento?
11. ¿Qué estrategia usaron para distribuir los datos de prueba?
12. ¿Qué tablas fueron más difíciles de poblar y por qué?
13. ¿Qué harían diferente si pudieran rediseñar la base?
14. ¿Cómo estructuraron los 3 CRUDs para mantener la consistencia?
15. ¿Qué aprendizaje obtuvieron del uso del ORM?
16. ¿Cómo reutilizarían su diseño en otros proyectos?
17. ¿Qué tan escalable consideran que es el sistema?
18. ¿Qué limitaciones encontraron al usar vistas?
19. ¿Qué estrategias implementaron para controlar integridad referencial?
20. ¿Qué impacto tuvo la coevaluación en su percepción del trabajo en equipo?

Rúbrica de Evaluación (100 puntos)

Criterio	Puntos
Modelo E-R completo, normalizado, con 20+ tablas	15
Implementación con ORM y 3 CRUDs funcionales	15
Vistas (al menos 2), funciones SQL (3+) y tipos personalizados (5+)	10
Validaciones (restricciones, checks, triggers)	10
Datos de prueba coherentes (mínimo 1000 registros)	5
3 reportes con 5 filtros cada uno y exportación a CSV	15
Uso correcto de vistas en los índices	5
Documentación (README, scripts, diagrama, versión)	10
Reflexión grupal profunda y bien argumentada	15
Total	100