

INFORME DE VERIFICACIÓN DE DISEÑO

BASE DE DATOS – SEGUNDA ITERACIÓN – JUAN GONZALEZ – 2021114063

Resumen.

En el presente informe, se realizó verificación de los informes y productos entregables de los diferentes diseñadores, la verificación abarca el análisis de 22 tablas implementadas, sus relaciones, reglas de integridad, normalización y correspondencia con las 36 historias de usuario documentadas, entendiéndolo como un proceso orgánico al ritmo de desarrollo del microdiseño del curso de Base de Datos, para el semestre 2025 – II.

Verificación de entregables.

Dentro del rol adquirido, en concordancia con los requerimientos para el presente informe se solicitaron a los compañeros del grupo los trabajos adelantados de acuerdo con sus roles, siendo:

- Camilo Andrés Mancilla Fontalvo, como diseñador lógico.
- David Brunal Triana, como diseñador físico.
- Jaime Andrés Rincón Galezzo, como especificador de requisitos.
- Luis Felipe de Jesús Muñoz Camacho, como diseñador conceptual.

Verificación de normalización.

Tablas que actualmente no se encuentran en 3FN:

Tabla	Campo Problemático	Tipo de Violación	HU Afectada
Ciudad	Departamento (FK directo)	Debe ser entidad independiente	2.9
planes_de_alquiler	Cadena 1:1 con duracion y tarifas	Estructura innecesariamente compleja	3.1, 3.2
estado_Bicicleta	Clave primaria no estable	PK debería ser ID numérico	1.2
Beneficios_y_Condiciones	nombre como PK	PK no estable (texto)	3.4
Metodo_Pago	Nombre como PK	PK no estable (texto)	3.3
Horario	PK tipo TIME incorrecto	Tipo de dato inadecuado para PK	2.4
Punto_De_Alquiler	Fotos como TEXT	Debería ser tabla relacionada	2.7
Relación N:N faltante	plan_metodo_pago	Tabla intermedia ausente	3.3

Validación de identificadores.

Problemas de estabilidad identificados en las claves:

Tabla	PK Actual	Problema	Corrección
estado_Bicicleta	Nombre (VARCHAR)	Texto mutable como PK	Agregar id_estado INT
Beneficios_y_Condiciones	nombre (VARCHAR)	Texto mutable como PK	Agregar id_beneficio INT
Metodo_Pago	Nombre (VARCHAR)	Texto mutable como PK	Agregar id_metodo INT
Horario	id_horario (TIME)	Tipo de dato inadecuado	Cambiar a INT IDENTITY

Restricciones UNIQUE Faltantes:

Campo	Tabla	Justificación	HU
Numero_Poliza	Seguro	Pólizas son únicas por aseguradora	1.5
email	Usuario	Correo es identificador de login	4.1
codigo_dane	Departamento	Código DANE es único por departamento	2.9

Reglas de integralidad.

Campos obligatorios (Que no sean NULL) que faltan:

Campo	Tabla	HU que lo requiere	Criterio
tipo_uso	Bicicleta	1.6	'Esta clasificación debe ser obligatoria'
tipo_asistencia	Bicicleta	1.7	'Esta clasificación debe ser obligatoria'
id_historial	Reseña	4.4	'Solo usuarios que completaron recorrido pueden reseñar'
email	Usuario	4.1	'Correo electrónico obligatorio para registro'
password_hash	Usuario	4.1	'Contraseña obligatoria para autenticación'

Restricciones faltantes:

Tabla	Campo	Restricción	HU
Reseña	calificacion	CHECK (calificacion >= 1 AND calificacion <= 5)	4.4
Duracion	fecha_fin	CHECK (fecha_fin IS NULL OR fecha_fin >= fecha_inicio)	3.2
Punto_De_Alquiler	Latitud	CHECK (Latitud >= -90 AND Latitud <= 90)	2.8
Punto_De_Alquiler	Longitud	CHECK (Longitud >= -180 AND Longitud <= 180)	2.8
Pago	monto	CHECK (monto > 0)	3.2
Tarifa	precio	CHECK (precio > 0)	3.2
Bicicleta	tipo_uso	CHECK (tipo_uso IN ('montaña', 'urbana', 'ruta'))	1.6
Bicicleta	tipo_asistencia	CHECK (tipo_asistencia IN ('convencional', 'eléctrica'))	1.7
Multimedia	tipo	CHECK (tipo IN ('jpg', 'png', 'mp4'))	5.2
Punto_De_Alquiler	Capacidad_Bicicletas	CHECK (Capacidad_Bicicletas > 0)	2.3

Es importante resaltar que aún no hay integralidad referencial, se sugiere lo siguiente:

Relación	ON DELETE	ON UPDATE	Justificación
Bicicleta → Punto_De_Alquiler	RESTRICT	CASCADE	No eliminar puntos con bicicletas
Bicicleta → estado_Bicicleta	RESTRICT	CASCADE	Validar estados antes de eliminar
Historial_de_alquiler → Usuario	RESTRICT	CASCADE	Preservar historial de alquileres
Reseña → Usuario	CASCADE	CASCADE	Eliminar reseñas con usuario
Multimedia → Reseña	CASCADE	CASCADE	Eliminar archivos con reseña
Usuario → Turista/Moderador/Admin	CASCADE	CASCADE	Mantener integridad de herencia

Trazabilidad de las historias con el desarrollo del diseño conceptual, lógico y físico.

Categoría HU	Total HU	Implementadas
1. Bicicletas (Jaime)	7	5
2. Puntos Alquiler (Luis)	9	8
3. Planes (Camilo)	6	4
4. Usuarios (David)	11	9
5. Multimedia (Juan)	3	3
TOTAL	36	29

Aquí se denota por nombre, porque cada uno de los integrantes realizó aportes en las historias de usuario, y hubo un ambiente colaborativo para el desarrollo de la base de datos, en cada etapa de su desarrollo.

Verificación de la cardinalidad implementada.

Relación	Cardinalidad	Justificación de Negocio
Bicicleta - Punto_De_Alquiler	N:1 (obligatoria)	Cada bici debe estar en un punto
Reseña - Usuario	N:1 (obligatoria)	Usuario es autor obligatorio
Multimedia - Reseña	N:1 (obligatoria)	Archivos pertenecen a reseña
Punto_De_Alquiler - Ciudad	N:1 (obligatoria)	Punto siempre en una ciudad
Bicicleta - Seguro	N:1 (opcional)	No todas tienen seguro inmediato
Usuario – (Administrador, Moderador y turista)	1:1 (especialización)	Jerarquía exclusiva y total

Correcciones necesarias para siguiente iteración.

#	Corrección	Acción Requerida	HU
1	Agregar campos email y password_hash en Usuario	ALTER TABLE Usuario ADD email VARCHAR(255) NOT NULL UNIQUE, password_hash VARCHAR(255) NOT NULL	4.1
2	Agregar campos tipo_uso y tipo_asistencia en Bicicleta	ALTER TABLE Bicicleta ADD tipo_uso VARCHAR(20) NOT NULL CHECK(...), tipo_asistencia VARCHAR(20) NOT NULL CHECK(...)	1.6, 1.7
3	Crear tabla Departamento independiente	CREATE TABLE Departamento (codigo_dane INT PRIMARY KEY, nombre VARCHAR(100)); Refactorizar Ciudad.Departamento	2.9
4	Simplificar estructura planes-duracion-tarifas	Consolidar en planes_de_alquiler con campos directos (precio, fecha_inicio, fecha_fin)	3.1, 3.2
5	Crear tabla plan_metodo_pago (N:N)	CREATE TABLE plan_metodo_pago (id_plan INT, id_metodo INT, PRIMARY KEY (id_plan, id_metodo))	3.3
6	Agregar FK id_historial en Reseña	ALTER TABLE Reseña ADD id_historial INT FOREIGN KEY REFERENCES Historial_de_alquiler(id_alquiler)	4.4
7	Agregar CHECK para calificación	ALTER TABLE Reseña ADD CONSTRAINT chk_calificacion CHECK (calificacion >= 1 AND calificacion <= 5)	4.4
8	Validar capacidad de bicicletas	Crear trigger que valide COUNT(bicicletas) <= Capacidad_Bicicletas del punto	2.3
9	Validar rol único por usuario	Crear trigger que garantice un usuario en exactamente una tabla de rol	4.11
10	Especificar políticas ON DELETE/UPDATE	Agregar cláusulas ON DELETE/UPDATE a todas las FK según tabla en sección 3.3	N/A

Conclusiones.

La segunda iteración del diseño de BICI-GO presenta una estructura base sólida con 22 tablas que cubren parte de las historias de usuario y cumplen totalmente con 1FN y 2FN. Sin embargo, apenas y poco más de las tablas cumplen 3FN (13 de 22), falta la normalización completa del módulo de departamentos, la estructura planes-duración-tarifas es innecesariamente compleja, no existen restricciones CHECK para validaciones de negocio, faltan campos críticos como email y password en Usuario, y las políticas de integridad referencial (ON DELETE/UPDATE) no están especificadas.