

Colnexus Reporte Final

Daniel Alejandro Duarte Duarte

Mónica Castro Benítez

Juan Felipe Vela Jiménez

José Alejandro Contreras Obregón

Prof. Andrés Oswaldo Calderón Romero

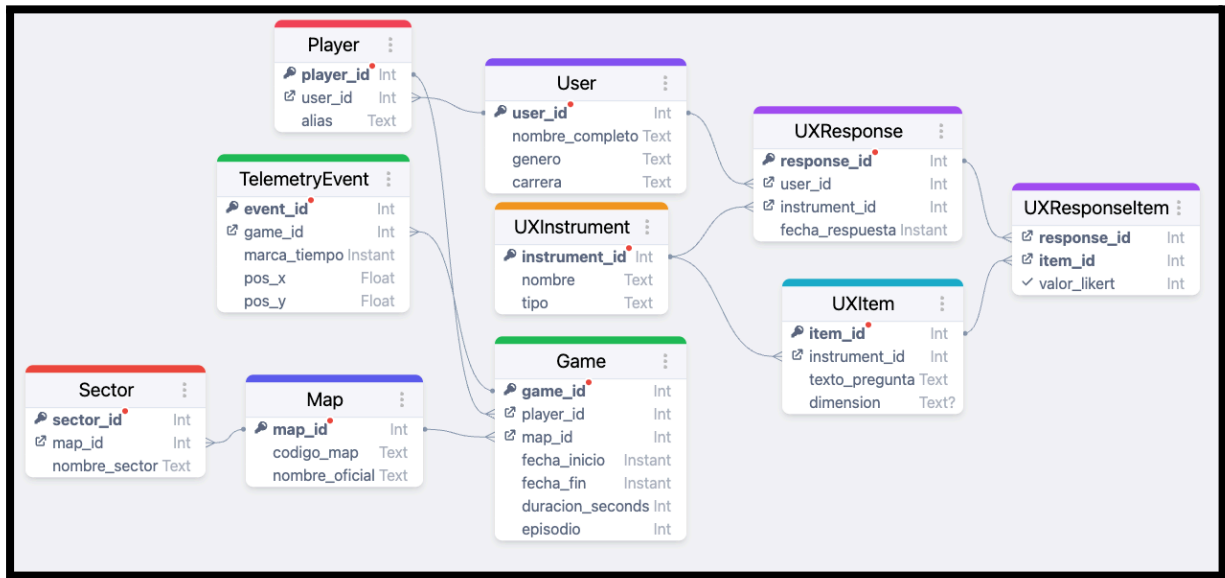
25 de Noviembre de 2025

Introducción

El presente informe detalla el diseño conceptual y lógico para un sistema de base de datos (BD) destinado a recopilar datos de telemetría de juego y experiencia de usuario (UX) generados por el puerto fuente *Chocolate-Doom*. Este proyecto se enmarca dentro de un esfuerzo de investigación para analizar las dinámicas de juego y el rendimiento en un entorno deliberadamente diseñado para emular la tecnología de los años 90 y generar una propuesta que atienda la consultoría la cual nos fue presentada.

1. Diseño de DB

1.1 Diagrama E-R



1.2 Esquema Relacional

Tabla	Atributos	Llaves/Relaciones
GameSession		
GameLevel		
TelemetryTick		
UXItem		
UXResponse		

1.3 Raciocinio

Separación entre telemetría y sesiones:

- GameSession actúa como tabla padre que agrupa todos los *ticks* de telemetría de una partida, lo que facilita consultas por sesión (duración, nivel jugado, jugador).
- TelemetryTick almacena datos por “tic” del juego (posición, acción, número de tick), lo que permite un análisis muy granular de comportamiento durante la partida.

Modelado de niveles (GameLevel)

- Tener una tabla GameLevel (o similar) para los niveles del juego permite normalizar la información de los niveles y evitar duplicación de datos de nivel en cada sesión.
- Además, facilita extender la información del nivel (por ejemplo, nombre, dificultad, diseño) sin repetirlo en cada sesión.

Instrumento UX (cuestionario PENS)

- UXItem representa preguntas o ítems del cuestionario, categorizados por dimensión (autonomía, competencia, etc.), lo que permite mantener la estructura del instrumento en la base de datos.
- UXResponse almacena las respuestas de los jugadores a cada ítem para cada sesión, vinculando la experiencia subjetiva (cuestionario) con la sesión de juego real. El score se guarda y está posiblemente validado con un CHECK para asegurar que esté en un rango válido (por ejemplo, 1–7).

Integridad referencial

- Las claves foráneas (session_id, item_id, level_id) aseguran que los datos de telemetría y respuestas UX siempre estén relacionados con una sesión válida y con ítems válidos del cuestionario.
- Esto previene entradas huérfanas (“orphan rows”) y garantiza consistencia entre tablas.

Flexibilidad y escalabilidad

- Al separar los ticks en su propia tabla, el esquema puede escalar para manejar grandes volúmenes de datos sin sobrecargar la tabla de sesiones.
- También permite construir consultas complejas: por ejemplo, cruzar la posición o tipo de acción en ciertos “ticks” con las puntuaciones del cuestionario para explorar correlaciones entre comportamiento de juego y experiencia del usuario.

Análisis relacional enriquecido

- Tener tanto datos objetivos (telemetría) como subjetivos (respuestas PENS) en la base de datos permite análisis de tipo “¿cómo influye la experiencia de juego en el comportamiento?” o “¿existen patrones de acción para jugadores que reportan más autonomía o competencia?”.

- Permite usar SQL para agregaciones, filtrados, métricas por sesión o por jugador, posibilitando también la generación de dashboards o reportes.

2. DDL Constraints Appendix

A continuación, una serie de fragmentos del archivo db_struct.sql que define tablas, claves primarias y foráneas a modo de ejemplificación para los DDL Constraints:

Figura 1: *Creación de tablas Inicial*

```
12 CREATE TABLE "User" (  
13     user_id SERIAL PRIMARY KEY,  
14     nombre_completo TEXT NOT NULL,  
15     genero TEXT NOT NULL,  
16     carrera TEXT NOT NULL  
17 );  
18  
19 CREATE TABLE Player (  
20     player_id SERIAL PRIMARY KEY,  
21     user_id INTEGER NOT NULL REFERENCES "User"(user_id),  
22     alias TEXT NOT NULL  
23 );  
24  
25 CREATE TABLE Map (  
26     map_id SERIAL PRIMARY KEY,  
27     codigo_map VARCHAR(50) NOT NULL,  
28     nombre_oficial VARCHAR(200) NOT NULL  
29 );  
30  
31 CREATE TABLE Game (  
32     game_id SERIAL PRIMARY KEY,  
33     player_id INTEGER NOT NULL REFERENCES Player(player_id),  
34     map_id INTEGER NOT NULL REFERENCES Map(map_id),  
35     fecha_inicio TIMESTAMP NOT NULL,  
36     fecha_fin TIMESTAMP NOT NULL,  
37     duracion_seconds INTEGER NOT NULL,  
38     episodio INTEGER NOT NULL  
39 );
```

Figura: *Creación de tablas Inicial (continuación)*

```

41 CREATE TABLE Sector (
42     sector_id SERIAL PRIMARY KEY,
43     map_id INTEGER NOT NULL REFERENCES Map(map_id),
44     nombre_sector VARCHAR(150) NOT NULL
45 );
46
47 CREATE TABLE TelemetryEvent (
48     event_id SERIAL PRIMARY KEY,
49     game_id INTEGER NOT NULL REFERENCES Game(game_id),
50     marca_tiempo TIMESTAMP NOT NULL,
51     pos_x NUMERIC NOT NULL,
52     pos_y NUMERIC NOT NULL
53 );
54
55 CREATE TABLE UXInstrument (
56     instrument_id SERIAL PRIMARY KEY,
57     nombre VARCHAR(100) NOT NULL,
58     tipo VARCHAR(50) NOT NULL
59 );
60
61 CREATE TABLE UXItem (
62     item_id SERIAL PRIMARY KEY,
63     instrument_id INTEGER NOT NULL REFERENCES UXInstrument(instrument_id),
64     texto_pregunta TEXT NOT NULL,
65     dimension VARCHAR(100)
66 );
67
68 CREATE TABLE UXResponse (
69     response_id SERIAL PRIMARY KEY,
70     user_id INTEGER NOT NULL REFERENCES "User"(user_id),
71     instrument_id INTEGER NOT NULL REFERENCES UXInstrument(instrument_id),
72     fecha_respuesta TIMESTAMP NOT NULL,
73     respuestas_json TEXT DEFAULT '{} '
74 );
75
76 CREATE TABLE UXResponseItem (
77     response_id INTEGER NOT NULL REFERENCES UXResponse(response_id),
78     item_id INTEGER NOT NULL REFERENCES UXItem(item_id),
79     valor_likert INTEGER NOT NULL CHECK (valor_likert BETWEEN 1 AND 7),
80     PRIMARY KEY (response_id, item_id)
81 );

```

3. Descripción ETL

El repositorio incluye el archivo loader_script.cxx, que transforma los logs TSV de telemetría de juego y genera un script data_loader.sql con sentencias INSERT ya válidas.

Los pasos básicos son:

1. Limpiar y normalizar los datos de los logs, por ejemplo sustituyendo nombres de jugador por códigos anónimos.
2. Validar campos, por ejemplo que tick_number sea no nulo y que session_id exista.
3. Generar código SQL para cargar los datos en las tablas definidas en el DDL.
4. Cargar los datos de UX mediante ux_instrument_pens.sql.

El log de telemetría se ubica en la carpeta Data, con la convención de nombre:

NombreJugador_IDPartida.txt.

Muestra de telemetría cruda: Aquí un ejemplo del tipo de archivo que se encuentra en la carpeta Data, siguiendo el formato TSV esperado:

When: 2025-11-04 19:49:25 Episode: 1 Map: 5									
timestamp	tic	x	y	z	angle	momx	momx	momx	
2025-11-04 19:49:25		171	-224	-624	0	85.78	-1	-2	0
2025-11-04 19:49:27		206	-192	-594	0	187.03	-1	0	0
2025-11-04 19:49:28		241	-281	-627	0	2.81	2	2	0
2025-11-04 19:49:29		276	-26	-590	0	104.06	1	1	0
2025-11-04 19:49:30		311	-103	-576	0	97.03	-12	-2	0
2025-11-04 19:49:31		346	-376	-230	0	91.41	-3	15	0
2025-11-04 19:49:32		381	-212	-270	-9	82.97	12	-12	0
When: 2025-11-04 19:49:33 Episode: 1 Map: 1									
timestamp	tic	x	y	z	angle	momx	momx	momx	
2025-11-04 19:49:35		417	1055	-3538	0	91.76	-1	6	0
2025-11-04 19:49:36		452	963	-3282	-16	167.34	-6	6	0
2025-11-04 19:49:37		487	697	-3226	0	179.65	-9	-1	0
2025-11-04 19:49:38		522	399	-3225	-8	184.92	-9	-1	0
2025-11-04 19:49:39		557	107	-3234	104	186.68	-9	-1	0
2025-11-04 19:49:40		592	-184	-3240	128	177.89	-9	0	0
2025-11-04 19:49:41		648	-288	-3247	104	348.40	4	-2	0
2025-11-04 19:49:42		683	-42	-3265	104	25.31	7	2	0
2025-11-04 19:49:43		718	188	-3108	50	51.68	5	5	0
2025-11-04 19:49:44		753	171	-3151	-8	32.34	-3	-4	0
2025-11-04 19:49:45		788	239	-3100	-8	111.45	-1	2	0
2025-11-04 19:49:46		823	103	-3066	-8	234.49	-4	-5	0
2025-11-04 19:49:47		858	144	-3153	-8	357.54	2	-2	0
2025-11-04 19:49:48		893	239	-3088	-8	76.64	0	4	0
2025-11-04 19:49:49		928	205	-3025	-8	87.19	-1	0	0

4. Queries y Resultados

1. Average duration of game sessions per map.

SQL Code	
Printed Result	

2. Players with the highest average proximity.

SQL Code	
Printed Result	

3. Shortest and longest trajectory distances per player.

SQL Code	
Printed Result	

4. List UX survey responses for players with above-average trajectories duration.

SQL Code	
Printed Result	

5. Most Visited Sector (Hotspot) per Episode and Map.

SQL Code	
Printed Result	

6. Number of Tics Where Players Were Together in a Sector.

SQL Code	
Printed Result	

7. Average UX Score for Players with the Shortest Trajectory per Episode.

SQL Code	
Printed Result	

8. Total Distance Traveled and Average Speed per Player, Analyzing All Games for a Player.

SQL Code	
Printed Result	

5. Nota Ética

Este proyecto analiza telemetría de juego proveniente del motor modificado

Chocolate-Doom con el objetivo de diseñar e implementar una base de datos relacional que permita la ingestión, validación y análisis de datos de juego por tick, junto con respuestas del cuestionario de experiencia del jugador (PENS: *Player Experience of Need Satisfaction*).

De acuerdo con la descripción del repositorio, los datos incluyen archivos de telemetría de sesiones de juego (TSV) y resultados del instrumento PENS suministrados por participantes voluntarios. A continuación se describen las consideraciones éticas aplicables:

1. Naturaleza de los datos: El proyecto utiliza telemetría de juego de DOOM y respuestas del cuestionario PENS. Los datos consisten únicamente en métricas de juego (movimiento, acciones, rendimiento) y valoraciones de experiencia del jugador.

2. Privacidad y anonimato: No se almacena información personal identificable. Cualquier nombre presente en los archivos originales se reemplaza por códigos anónimos. La base de datos no contiene correos, IPs ni datos sensibles.

3. Minimización y propósito: Solo se recopila la información necesaria para el análisis académico del comportamiento de juego. Los datos se utilizan exclusivamente con fines educativos y no para perfilamiento o usos externos.

4. Riesgos y mitigación: El riesgo ético es bajo debido a la naturaleza no sensible de los datos y al uso de pseudonimización. Los resultados del análisis se presentan de forma agregada, sin exponer sesiones individuales de manera identificable.

5. Responsabilidad: El equipo se compromete a manejar todos los datos de forma responsable, garantizando privacidad, transparencia y respeto por los participantes.

6. Diccionario de Datos

Tabla	Atributo	Tipo de dato	PK / FK	Restricciones / Comentarios
User	user_id	SERIAL	PK	Identificador único del usuario.
	nombre_completo	TEXT		No nulo.
	genero	TEXT		Control semántico (masculino, femenino, otro).
	carrera	TEXT		Información académica.
Player	player_id	SERIAL	PK	Identificador del alias del jugador.
	user_id	INTEGER	FK → User(user_id)	Relaciona el alias con el usuario dueño.
	alias	TEXT		Alias único dentro del estudio.

Map	map_id	SERIAL	PK	Identificador del mapa.
	codigo_map	VARCHAR(50)		Código interno del motor del juego.
	nombre_oficial	VARCHAR(200)		Nombre descriptivo.
Sector	sector_id	SERIAL	PK	Identificador del sector.
	map_id	INTEGER	FK → Map(map_id)	Un sector pertenece a un mapa.
	nombre_sector	VARCHAR(150)		Nombre o referencia espacial.
Game	game_id	SERIAL	PK	Identificador de la sesión de juego.
	player_id	INTEGER	FK → Player(player_id)	Jugador que participó.
	map_id	INTEGER	FK → Map(map_id)	Mapa donde se jugó.
	fecha_inicio	TIMESTAMP		No nulo.
	fecha_fin	TIMESTAMP		No nulo.
	duracion_seconds	INTEGER		Validación de duración.
	episodio	INTEGER		Episodio jugado.
TelemetryEvent	event_id	SERIAL	PK	Identificador del evento.
	game_id	INTEGER	FK → Game(game_id)	Evento asociado a una sesión.
	marca_tiempo	TIMESTAMP		Orden temporal.
	pos_x	NUMERIC		Coordenada X.
	pos_y	NUMERIC		Coordenada Y.
UXInstr	instrument_id	SERIAL	PK	Identificador del instrumento UX.

ument				
	nombre	VARCHAR(100)		Nombre del cuestionario.
	tipo	VARCHAR(50)		Tipo o categoría del instrumento.
UXRes ponse	response_id	SERIAL	PK	Identificador de la respuesta.
	user_id	INTEGER	FK → User(user_id)	Usuario que respondió.
	instrument_id	INTEGER	FK → UXInstrument(i nstrument_id)	Instrumento aplicado.
	fecha_respuesta	TIMESTAMP		Registro temporal.
UXItem	item_id	SERIAL	PK	Identificador del ítem.
	instrument_id	INTEGER	FK → UXInstrument(i nstrument_id)	Pertenece a un instrumento.
	texto_pregunta	TEXT		Texto de la pregunta.
	dimension	VARCHAR(100)		Dimensión psicológica.
UXRes ponselt em	response_id	INTEGER	FK → UXResponse(r esponse_id)	Relaciona respuesta con ítem.
	item_id	INTEGER	FK → UXItem(item_i d)	Ítem respondido.
	valor_likert	INTEGER		CHECK (valor_likert BETWEEN 1 AND 7).

Conclusión

ColNexus logró integrar datos de telemetría de DOOM y respuestas del cuestionario PENS dentro de una base de datos relacional diseñada para garantizar consistencia, integridad y capacidad analítica. A través de un proceso ETL propio, los datos crudos fueron limpiados,

normalizados y transformados en un conjunto estructurado apto para consultas tanto descriptivas como exploratorias.

El sistema resultante permite vincular métricas objetivas de comportamiento en el juego con evaluaciones subjetivas de la experiencia del jugador, habilitando análisis más completos sobre patrones de interacción y rendimiento. El proyecto demuestra la eficacia de combinar técnicas de ingeniería de datos y modelado relacional en un entorno académico, y deja una base sólida para futuras ampliaciones orientadas al análisis avanzado o la visualización de resultados.