

Pontificia Universidad Javeriana

Proyecto

Diego Alejandro Negro

Jhoan Alberto Galeano Laya

Departamento de sistemas, Bases de datos

Oswaldo Calderón

15 de octubre 2025

I. SUPUESTOS GENERALES

1. Los datos de telemetría son generados de manera automática por el motor del videojuego y almacenados en archivos de texto.
2. Cada línea de telemetría está asociada a un identificador único de jugador y un identificador de partida que permite reconstruir trayectorias y analizar comportamientos dentro del entorno del juego.
3. Cada registro de telemetría corresponde a un tick o instante de tiempo, este incluye información sobre la posición del jugador, ya sea x,y o z, la dirección y las estadísticas del estado del jugador como la salud, la munición y la armadura.
4. Los usuarios pueden participar en múltiples partidas y adoptar diferentes alias o personajes.
5. Se aplican instrumentos de medición de experiencia de usuario (UX), como PENS, GUESS o BANGS, cuyas respuestas se asocian a cada usuario y partida para el análisis correlacional.
6. Las partidas están organizadas en episodios, mapas y sectores, lo que permite vincular los datos que son espaciales a una estructura jerárquica del entorno del juego.
7. Todos los datos que se recolectan se centralizan en una base de datos relacional implementada en PostgreSQL.

II. REQUISITOS FUNCIONALES

1. El sistema debe registrar información de los usuarios voluntarios.
2. El sistema debe registrar la formación de los datos demográficos básicos (edad, género, experiencia y consentimiento).
3. El sistema debe almacenar la telemetría por tic con los valores de posición, orientación y estadísticas del jugador.
4. El sistema debe registrar los instrumentos de experiencia de usuario, sus ítems y las respuestas asociadas a cada usuario y partida.
5. El sistema debe gestionar los episodios, sectores y mapas donde se desarrolla cada partida.
6. El sistema debe permitir realizar la consulta analítica de la trayectoria completa de un jugador en una partida.
7. El sistema debe permitir realizar la consulta analítica de mapas de calor o densidad por sector.
8. El sistema debe permitir realizar la consulta analítica del análisis de proximidad y cooperación entre jugadores.
9. El sistema debe permitir realizar la consulta analítica del resumen general de desempeño y experiencia por grupo o usuario.
10. El sistema debe permitir la carga y la validación automática de archivos TSV de telemetría.

III. REQUISITOS NO FUNCIONALES

1. El sistema debe implementarse en PostgreSQL.
2. El sistema debe implementar en una base de datos los tipos avanzados como uuis, jsonb, geometry para el manejo eficiente de datos.
3. El sistema debe ser capaz de almacenar y consultar grandes volúmenes de datos de telemetría.
4. El sistema del diseño debe permitir la incorporación de nuevos instrumentos UX o tipos de eventos de telemetría sin modificar la estructura base.
5. El sistema debe mantener la integridad de los datos en caso de fallos y permitir respaldos periódicos automáticos.
6. El sistema debe permitir a los datos protegerse mediante autenticación y restricciones de acceso.
7. El sistema debe permitir que los scripts de carga y consulta deben funcionar en entornos de diferentes sistemas operativos.
8. El sistema debe documentar la estructura del modelo para facilitar modificaciones a futuro.

IV. REQUISITOS ETICOS

1. Todos los usuarios deben aceptar explícitamente el consentimiento informado antes de participar en la recolección de datos.
2. Los datos recolectados solo se utilizarán con fines académicos y no comerciales.
3. Los identificadores personales se deben anonimizar o reemplazar por UUIDs para proteger la identidad de los participantes.
4. Los investigadores deben cumplir las políticas institucionales sobre el manejo de datos personales y los principios de la ley de protección de datos personales.
5. Se debe garantizar la confidencialidad y privacidad de la información sensible.
6. Las métricas o análisis representados deben evitar interpretaciones discriminatorias o sesgos.
7. Los participantes pueden solicitar en cualquier momento la eliminación de sus datos del sistema.

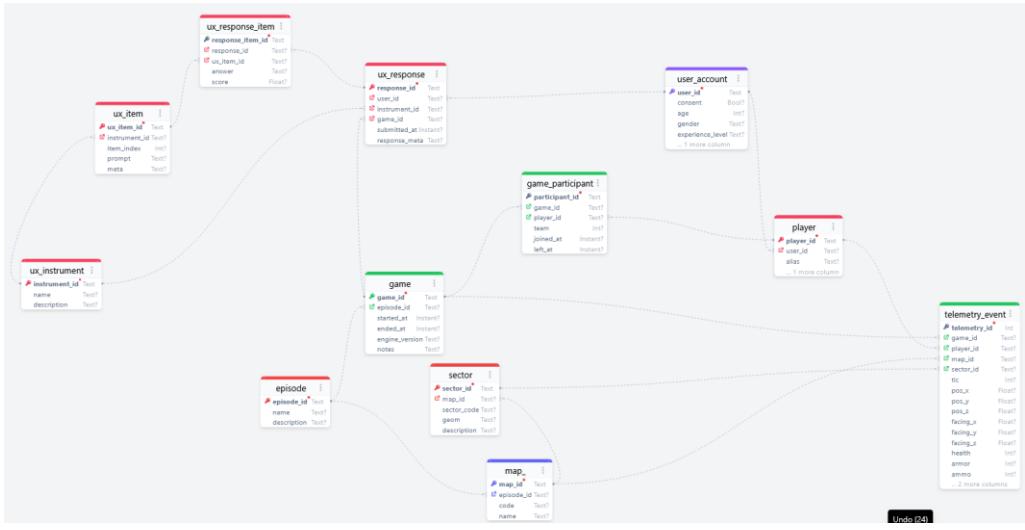


Fig 1. Diagrama E.R.

V. DICCIONARIO DE DATOS

El siguiente diccionario de datos muestra las entidades y relaciones que se presentan en la figura 1 que es el diagrama E.R, cada tabla del modelo lógico corresponde a una entidad del modelo generado en Azimutt.

- User_account.

Columna	Tipo (Postgres)	PK / FK	Null?	Descripción
user_id	uuid	PK	NOT NULL	Identificador del voluntario (uuid).
created_at	timestamptz		NOT NULL DEFAULT now()	Fecha/hora de creación del registro.
consent	boolean		NOT NULL	Indica si dio consentimiento informado.
age	smallint		NULL	Edad del participante.
gender	text		NULL	Género (opcional).
experience_level	text		NULL	Nivel de experiencia en videojuegos.
notes	text		NULL	Observaciones adicionales.

- player.

Columna	Tipo	PK / FK	Null?	Descripción
player_id	uuid	PK	NOT NULL	Identificador del alias/jugador.
user_id	uuid	FK → user_account(user_id)	NULL (ON DELETE SET NULL)	Si está vinculado a un usuario voluntario.
alias	text		NOT NULL	Nombre del alias en el juego.
created_at	timestamptz		NOT NULL DEFAULT now()	Fecha de creación del alias.

- episode.

Columna	Tipo	PK / FK	Null?	Descripción
episode_id	uuid	PK	NOT NULL	Identificador del episodio/nivel.
name	text		NOT NULL	Nombre del episodio.
description	text		NULL	Descripción del episodio.

- Map_

Columna	Tipo	PK / FK	Null?	Descripción
map_id	uuid	PK	NOT NULL	Identificador del mapa.
episode_id	uuid	FK → episode(episode_id)	NOT NULL	Episodio al que pertenece el mapa.
code	text		NOT NULL	Código/identificador del mapa (map_code).
name	text		NULL	Nombre legible del mapa.

- Sector.

Columna	Tipo	PK / FK	Null?	Descripción
sector_id	uuid	PK	NOT NULL	Identificador del sector.

map_id	uuid	FK → map_(map_id)	NOT NULL	Mapa al que pertenece el sector.
sector_code	text		NOT NULL	Código/etiqueta del sector dentro del mapa.
geom	geometry (PostGIS) o text		NULL	Polígono/point que delimita el sector (si usas PostGIS).
description	text		NULL	Texto descriptivo del sector.

- Game.

Columna	Tipo	PK / FK	Null?	Descripción
game_id	uuid	PK	NOT NULL	Identificador de la partida/sesión.
episode_id	uuid	FK → episode(episode_id)	NULL	Episodio jugado (opcional).
started_at	timestamptz		NULL	Fecha/hora de inicio.
ended_at	timestamptz		NULL	Fecha/hora de fin.
engine_version	text		NULL	Versión del motor/engine usado.
notes	text		NULL	Observaciones sobre la partida.

- Game_participant.

Columna	Tipo	PK / FK	Null?	Descripción
participant_id	uuid	PK	NOT NULL	PK de la relación participante.
game_id	uuid	FK → game(game_id)	NOT NULL	Partida en la que participa el jugador.
player_id	uuid	FK → player(player_id)	NOT NULL	Player (alias) participante.
team	smallint		NULL	Número de equipo (si aplica).

joined_at	timestamptz		NULL	Hora de ingreso a la partida.
left_at	timestamptz		NULL	Hora de salida.
(constraint)		UNIQUE(game_id, player_id)		Evita duplicar la misma participación.

- Telemetry_event

Columna	Tipo	PK / FK	Null?	Descripción
telemetry_id	bigserial	PK	NOT NULL	PK autonumérica (bigserial).
game_id	uuid	FK → game(game_id)	NOT NULL	Partida a la que corresponde el evento.
player_id	uuid	FK → player(player_id)	NOT NULL	Jugador que generó el evento.
tic	bigint		NOT NULL	Número de tic/frame emitido por el motor.
pos_x	double precision		NULL	Coordenada X.
pos_y	double precision		NULL	Coordenada Y.
pos_z	double precision		NULL	Coordenada Z (altura).
facing_x	double precision		NULL	Vector/dirección X (si aplica).
facing_y	double precision		NULL	Vector/dirección Y.
facing_z	double precision		NULL	Vector/dirección Z.
health	smallint		NULL	Salud del jugador en ese tic.
armor	smallint		NULL	Armadura en ese tic.
ammo	integer		NULL	Munición.
map_id	uuid	FK → map_(map_id)	NULL	Mapa donde ocurre el evento (opcional).
sector_id	uuid	FK → sector(sector_id)	NULL	Sector donde ocurre (opcional).

raw_line	text		NULL	Línea cruda del log para trazabilidad.
created_at	timestamptz		NOT NULL DEFAULT now()	Timestamp de ingestión.
(constraint)		UNIQUE(game_id, player_id, tic)		Evita duplicados por tic.

- Ux_instrument.

Columna	Tipo	PK / FK	Null?	Descripción
instrument_id	uuid	PK	NOT NULL	Identificador del instrumento (PENS/GUESS/BANGS).
name	text		NOT NULL	Nombre del instrumento.
description	text		NULL	Descripción / notas.

- Ux_item.

Columna	Tipo	PK / FK	Null?	Descripción
ux_item_id	uuid	PK	NOT NULL	Identificador del ítem/pregunta.
instrument_id	uuid	FK → ux_instrument(instrument_id)	NOT NULL	Instrumento al que pertenece.
item_index	integer		NOT NULL	Orden dentro del instrumento.
prompt	text		NULL	Texto de la pregunta/ítem.
meta	jsonb		NULL	Metadatos (opciones, escala, scoring).

- Ux_response.

Columna	Tipo	PK / FK	Null?	Descripción
response_id	uuid	PK	NOT NULL	Identificador de la respuesta (encuesta completa).
user_id	uuid	FK → user_account(user_id)	NULL	Usuario que completó la encuesta.
instrument_id	uuid	FK → ux_instrument(instrument_id)	NULL	Instrumento respondido.
game_id	uuid	FK → game(game_id)	NULL	Partida relacionada (si aplica).
submitted_at	timestamptz		NOT NULL DEFAULT now()	Fecha/hora envío.
response_meta	jsonb		NULL	Metadatos (duración, device, ip hashed, etc.).

- Ux_response_item.

Columna	Tipo	PK / FK	Null?	Descripción
response_item_id	uuid	PK	NOT NULL	PK del ítem-respuesta.
response_id	uuid	FK → ux_response(response_id)	NOT NULL	FK a la respuesta global.
ux_item_id	uuid	FK → ux_item(ux_item_id)	NULL	Ítem al que responde.
answer	text		NULL	Texto/selección de la respuesta.
score	numeric		NULL	Valor numérico si aplica (p. ej. escala Likert).

VI. JUSTIFICACION DE NORMALIZACION (3NF)

- **Atomicidad:** Cada atributo de cada una de las tablas contiene valores atómicos, dado que las posiciones están separadas en pos_x, pos_y, pos_z, no hay listas ni conjuntos en una columna, a excepción jsonb para los metadatos explícitos.
- **Dependencia completa sobre la PK:** Las tablas donde PK es simple como player_id tienen atributos que dependen únicamente de esa PK, en tablas con PK compuesta que no es el caso en tablas normales, en telemetry_id es una restricción única aplicada a evitar duplicados, pero la PK real es telemetry_id para este caso, no hay atributos que dependan solo de una parte de una PK compuesta.
- **Eliminación de dependencias transitivas:** No existen dependencias transitivas en las tablas principales, ya sea game_participan, en este caso no se guarda user o alias, ya que esto está normalizado en player y user_account, entonces, cualquier dato derivado se puede denormalizar por rendimiento, pero debe ser documentado.

VII. CONCLUSIONES

- El desarrollo del diseño conceptual permitió estructurar la información proveniente de la telemetría de DOOM y de cada uno de los instrumentos de experiencia de usuario (UX), y a través del modelo E-R se identificaron las entidades principales del dominio.
- La aplicación de las tres primeras formas normales permite validar que cada atributo dependa únicamente de su clave primaria.
- El modelo general ofrece una base sólida para la fase de implementación.