

Proyecto de Diseño de una Base de Datos de Telemetría y UX para la Investigación de Chocolate-Doom: Diseño Conceptual y Lógico (DBS 2025-30)

Colnexus

Integrantes:

- Juan Felipe Vela Jimenez
- Jose Alejandro Contreras Obregón
- Daniel Alejandro Duarte Duarte
- Monica Maria Castro Benitez

1. Introducción y contexto del proyecto

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.

1.1. Contextualización de la consultoría Chocolate-Doom

El puerto fuente *Chocolate-Doom* es fundamental para esta investigación debido a su filosofía de diseño conservadora. Este software no sólo reproduce la apariencia y la sensación de los juegos originales (Doom, Heretic, Hexen, Strife), sino que, de manera crucial para el análisis, preserva las limitaciones y errores (*bugs*) de las versiones vanilla de DOS. Este enfoque establece un objetivo único para el repositorio de datos: debe funcionar no solo como un sistema de *big data* para métricas de rendimiento y juego (telemetría), sino también como una herramienta de clasificación y análisis para eventos que son resultado directo de la emulación del código original.

La implicación directa de este enfoque genera unos requerimientos específicos para el modelado de datos, esta base de datos debe estar estructurada para permitir a los investigadores correlacionar eventos de juego (Como un *crash*, un comportamiento inesperado de un enemigo, o la muerte del jugador) con variables de la sesión, la versión específica del software y las características inherentes del juego fuente (Doom o Strife). Esto requiere el uso de dimensiones de referencia detalladas para *Mapas*, *Jugadores* y *Eventos de Telemetría*, que contengan metadatos explícitos sobre las condiciones del entorno de juego y las características del escenario.

1.2. Estructura del informe (Modelo de datos)

Con base en las prácticas estándares en cuanto a la arquitectura de bases de datos, este documento se centra en la primera etapa del diseño, buscando diferenciar claramente el modelo conceptual (orientado a la representación de entidades del dominio y sus relaciones, crucial para la comunicación con expertos del dominio y partes interesadas) del modelo lógico (orientado a la implementación relacional, tipificación de datos, integridad y normalización). Este análisis proporciona los cimientos técnicos necesarios para que los administradores de bases de datos (DBA) y los desarrolladores puedan proceder con la fase de modelo físico.

2. Supuestos clave y marco de requisitos

Para garantizar la viabilidad del diseño lógico, se establecen supuestos operacionales y se definen los requisitos esenciales, categorizados como funcionales, no funcionales y éticos.

2.1. Supuestos operacionales y tecnológicos

- **Fuente de datos:** Se asume que los datos son generados por un módulo modificado de *Chocolate-Doom* que asegura que cada evento se registre con atomicidad y con una marca de tiempo precisa, garantizando el orden cronológico dentro de cada sesión de juego.
- **Volumen y rendimiento:** El sistema manejará un volumen de datos de alta velocidad, característico de los sistemas de telemetría, donde la tasa de inserción es prioritaria. Este volumen justifica la adopción de claves auto-incrementales para las tablas de hechos, mejorando la eficiencia de los *joins* y la integridad referencial.
- **Calidad de UX:** Los datos de UX, aunque representan un volumen menor, son determinantes. Se requiere una vinculación estricta con las métricas de rendimiento para análisis correlacionales; por ejemplo, determinar si una configuración gráfica particular o un mapeo de control se correlaciona con un rendimiento de juego superior o inferior.

2.2. Requisitos funcionales (RF)

Los requisitos funcionales describen las acciones, comportamientos e interacciones específicas que el sistema debe ejecutar.

- **Registro de sesión completo:** El sistema tiene la obligación de registrar de forma exhaustiva el ciclo de vida de cada *SesionDeJuego*, incluyendo el momento preciso de inicio y fin, la duración total, la versión del software utilizada y el estado final de la sesión

(e.g., victoria, derrota, desconexión por error, salida normal).

- **Eventos granulares de telemetría:** El sistema debe capturar eventos de juego atómicos y detallados. Esto incluye, pero no se limita a, Muertes, Disparos, Recolección de Ítems, Cambios de Nivel y el uso de *power-ups*. La naturaleza de la investigación de *Chocolate-Doom*, centrada en los límites del sistema vanilla , exige que estos eventos incluyan metadatos detallados, estos metadatos pueden abarcar coordenadas espaciales del jugador (X, Y, Z), dirección de la visión y, si es posible, el estado interno del *tick* del juego al momento exacto del evento. Para manejar esta heterogeneidad de información contextual, se empleará un atributo tipo JSON o CLOB para almacenar datos cualitativos específicos.
- **Registro de configuración estática:** Es indispensable almacenar la ConfiguraciónUX utilizada por el jugador al comienzo de cada sesión. Esto asegura la trazabilidad de los parámetros ambientales, tales como la resolución de pantalla, los ajustes de sonido y el mapeo de controles, permitiendo correlacionar el diseño de la interfaz con los resultados de la telemetría.

2.3. Requisitos no funcionales (RNF)

Los requisitos no funcionales definen los atributos de calidad, como el rendimiento y la escalabilidad, del sistema.

- **Tasa de inserción:** El sistema debe estar optimizado para la inserción masiva. Se requiere una latencia de inserción mínima que permita soportar una tasa agregada de al menos 20,000 registros de eventos por segundo durante picos de actividad de la comunidad de jugadores/investigadores.
- **Escalabilidad de almacenamiento:** Dado el potencial crecimiento exponencial de los datos de telemetría, el diseño lógico debe facilitar la escalabilidad horizontal. La elección de la Tercera Forma Normal (3NF) y la adecuada separación de las dimensiones (tablas de referencia) de las tablas de hechos de alto volumen (eventos) es esencial para minimizar la redundancia y optimizar el espacio de almacenamiento, apoyando la estrategia de escalabilidad.
- **Seguridad:** Se requiere la implementación de controles de acceso robustos para restringir la manipulación y la consulta de datos, especialmente aquellos que, incluso tras la anonimización, pudieran considerarse sensibles o de acceso restringido para fines de investigación.

2.4. Requisitos éticos y de gobernanza de datos

La gestión ética de los datos es un pilar fundamental en cualquier proyecto de investigación. Estos requisitos se traducen directamente en restricciones obligatorias de diseño.

- **Anonimización obligatoria:** Cualquier información que permita la identificación directa de un jugador (PII), incluyendo direcciones IP o identificadores persistentes de dispositivos, debe ser sometida a un proceso riguroso de anonimización o pseudoanonimización inmediatamente después de la ingesta.
- **Segregación de PII:** Para proteger la privacidad y simplificar la gobernanza, la información que, aunque anonimizada o indirecta (como el Alias único), aún permita una vinculación personal, debe almacenarse en una tabla segregada del núcleo de la base de datos de telemetría. Esta segregación de la PII implica que la tabla principal JUGADOR utilizará una clave primaria opaca, el Jugador_ID (una clave sucedánea), que es independiente de cualquier identificador externo sensible. Solo una tabla auxiliar y de acceso restringido, referenciada por este Jugador_ID, contendrá los metadatos potencialmente sensibles.

3. Diseño conceptual: el modelo entidad-relación (DER)

El Modelo Entidad-Relación (DER) proporciona una representación visual y conceptual de las estructuras de datos y sus interconexiones, utilizando entidades y relaciones definidas por cardinalidades.

3.1. Entidades fundamentales

El modelo conceptual identifica las siguientes siete entidades primarias que capturan los elementos centrales del dominio *Chocolate-Doom* y la actividad de recopilación de telemetría:

Entidad	Descripción
User	Representa al estudiante voluntario que participa en el estudio. Contiene información demográfica y de consentimiento para el manejo ético de los datos.
Player	Identidad utilizada dentro del juego. Cada Player está asociado a un User, pero el mismo User puede tener múltiples alias o identidades de juego.
Game (Sesión de Juego)	Instancia única de una sesión de juego, con inicio y fin definidos. Asociada a un Player y realizada en un único Mapa. Representa una ejecución completa del juego por parte del jugador.

Map	Escenario jugable general donde ocurre una sesión de juego. Los mapas representan niveles del juego y pueden ser reutilizados en múltiples sesiones..
Sector	Subdivisión espacial de un mapa. Cada mapa se compone de múltiples sectores que identifican regiones explorables o transitadas del entorno.
TelemetryEvent	Registro atómico capturado durante el juego que representa el estado del jugador en un instante dado (ej. posición X,Y, tiempo o marca temporal). Permite reconstruir trayectorias y analizar patrones de desplazamiento.
UXInstrument	Instrumento estandarizado para medir la experiencia de usuario (como PENS, GUESS o BANGS), incluyendo su definición y estructura.
UXResponse	Respuestas proporcionadas por un participante a un UXInstrument, permitiendo el análisis subjetivo de experiencia y su relación con el comportamiento observable en el juego.
UXItem	Ítems o preguntas que componen un instrumento UX. Cada uno pertenece a un instrumento y está asociado a una dimensión específica (por ejemplo, Competencia o Autonomía).
UXResponseItem	Asociación entre una respuesta y los ítems individuales respondidos por el usuario. Permite almacenar los valores Likert para cada pregunta del instrumento aplicado.

3.2. Relaciones y cardinalidades clave

Las cardinalidades describen la interacción entre dos elementos de la base de datos, vitales para el diseño conceptual.

Relación	Cardinalidad	Descripción
User — Player	1 : N	Un jugador puede iniciar múltiples sesiones, pero cada Player pertenece estrictamente a un único User.
Player — Game (Sesión de Juego)	1 : N	Un jugador puede participar en múltiples sesiones de juego. Cada sesión es realizada por un solo jugador (Player).
Game — Map	1 : N	Un mapa puede ser usado en muchas sesiones, pero cada sesión se desarrolla en un único mapa
Map — Sector	1 : N	Cada mapa está compuesto de múltiples sectores. Un sector pertenece únicamente a un mapa.
Player — TelemetryEvent	1 : N (implícita)	Aunque Player no aparece directamente en TelemetryEvent, todos los eventos de una sesión corresponden al Player que ejecuta ese Game. La relación se hereda por la FK de Game.

Game — TelemetryEvent	1 : N	Una sesión de juego produce múltiples eventos de telemetría registrados a lo largo del tiempo. Cada evento pertenece a un único Game.
TelemetryEvent — Sector	N : 1	Cada evento ocurre dentro de un sector particular del mapa, lo que permite análisis espaciales.
User — UXResponse	1 : N	Un usuario puede responder uno o varios instrumentos UX a lo largo del estudio.
UXInstrument — UXResponse	1 : N	Cada respuesta corresponde a un instrumento UX específico aplicado al usuario (como PENS, GUESS o BANGS).
UXInstrument — UXItem	1 : N	Un instrumento puede contener múltiples ítems o preguntas.
UXResponse — UXItem (mediante UXResponseItem)	N : M	Representa las respuestas específicas de un usuario a cada ítem de un instrumento.

El siguiente extracto representa las entidades clave y sus atributos primarios en el modelo conceptual:

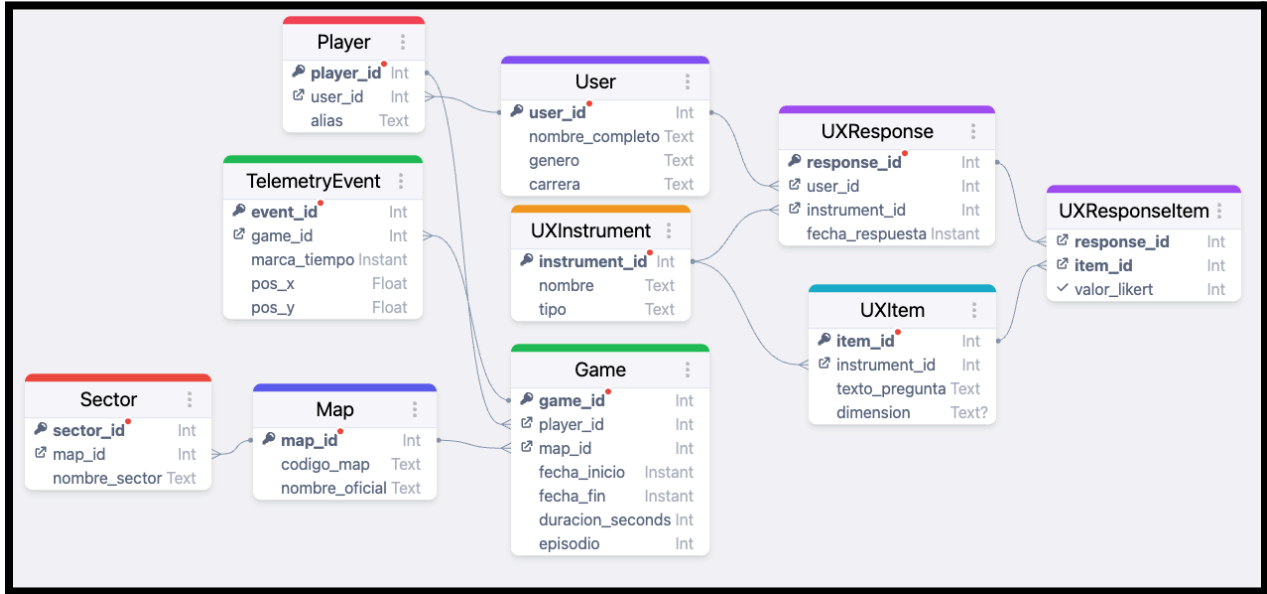
Tabla 3.3: Entidades Clave y Atributos Primarios:

Entidad	Atributos Principales	PK	Relación clave
Player	player_id, user_id(FK), alias	player_id	N : 1 con User 1 : N con Game.
Game	game_id, player_id(FK), map_id(FK), fecha_inicio, fecha_fin, duracion_seconds, episodio	game_id	N : 1 con Player N : 1 con Map
Map	map_id, codigo_map, nombre_oficial	map_id	1 : N con Sector
Sector	sector_id, map_id(FK), nombre_sector	sector_id	N : 1 con Map
TelemetryEvent	event_id, game_id(FK), marca_tiempo, pos_x, pos_y	event_id	N : 1 con Game
UXInstrument	instrument_id, nombre, tipo	instrument_id	1 : N con UXResponse
UXResponse	response_id, user_id(FK), instrument_id(FK), fecha_respuesta	response_id	N : 1 con User N : 1 con UXInstrument

UXItem	item_id, instrument_id (FK), texto_pregunta, dimension	item_id	N : 1 con UXInstrument
UXResponseItem	response_id (FK), item_id (FK), valor_likert	(response_id, item_id)	N : M entre UXResponse y UXItem

3.3. Esquema E-R

Se genera el esquema correspondiente usando Azimutt



4. Diseño lógico: esquema relacional, normalización y diccionario de datos

El diseño lógico transforma el modelo conceptual en un esquema relacional específico, asegurando la integridad de los datos mediante la normalización y la definición rigurosa de claves y restricciones.

4.1. Derivación y justificación de normalización (3NF)

El objetivo primario de la normalización es eliminar la redundancia y prevenir las anomalías de actualización, inserción y borrado. Para un sistema de investigación, donde la coherencia y la

precisión de los datos son imperativas, se justifica la adhesión estricta a la Tercera Forma Normal (3NF). La 3NF asegura que los atributos no clave dependan directamente de la clave primaria completa y no de otros atributos no clave (eliminando dependencias transitivas).

Se ha optado por el uso sistemático de claves autoincrementales (BIGINT auto-incrementales) como claves primarias para las tablas de hechos (*Game*, *TelemetryEvent*). Este enfoque es crítico para satisfacer el RNF de alto rendimiento (RNF-PER01), ya que las claves autoincrementales son pequeñas, deterministas y optimizan la velocidad de la ingesta de datos y las operaciones de *join*.

4.2. Esquema relacional normalizado (3NF)

A continuación, se presenta el esquema relacional en notación formal, donde los atributos subrayados Atributo representan la Clave Primaria (PK) y los atributos seguidos de FK son Claves Foráneas.

1. **USER** (user_id, nombre_completo, genero, carrera)
2. **PLAYER** (player_id, user_id (FK), alias)
3. **MAP** (map_id, codigo_map, nombre_oficial)
4. **SECTOR** (sector_id, map_id (FK), nombre_sector)
5. **GAME** (game_id, player_id (FK), map_id (FK), fecha_inicio, fecha_fin, duracion_seconds, episodio)
6. **TELEMETRYEVENT** (event_id, game_id (FK), marca_tiempo, pos_x, pos_y)
7. **UXINSTRUMENT** (instrument_id, nombre, tipo)
8. **UXRESPONSE** (response_id, user_id (FK), instrument_id (FK), fecha_respuesta)
9. **UXITEM** (item_id, instrument_id (FK), texto_pregunta, dimension)
10. **UXRESPONSEITEM** ((response_id (FK), item_id (FK)), valor_likert)

4.3. Análisis de dependencias funcionales y justificación de 3NF

La estructura propuesta garantiza el cumplimiento de la 3NF, esencial para la consistencia en el análisis de datos históricos.

- **Cumplimiento de 2NF:**

Todas las tablas que poseen claves compuestas, como UXResponseItem, aseguran que todos los atributos no clave (por ejemplo, *valor_likert*) dependen completamente de la clave primaria compuesta (response_id, item_id). Esto evita dependencias parciales y

asegura que los datos solo dependan de la combinación completa de la clave.

- **Cumplimiento de 3NF (Eliminación de transitividad):**

Cada atributo no clave depende únicamente de la clave primaria de su tabla y no de otros atributos no clave. Por ejemplo, en la tabla Game, los campos *fecha_inicio*, *fecha_fin* y *duracion_seconds* dependen directamente de *game_id*, y no de atributos derivados como *player_id* o *map_id*. Del mismo modo, en UXResponse, la columna *fecha_respuesta* depende únicamente de *response_id*, evitando dependencias transitivas con los datos de usuario o instrumento.

Esta estructura asegura que la información de telemetría y experiencia de usuario esté completamente normalizada, evitando redundancias, mejorando la integridad referencial y permitiendo un rendimiento óptimo en la inserción y análisis de los datos.

4.4. Diccionario de datos y restricciones de integridad

El Diccionario de Datos establece los detalles técnicos de cada atributo y sus restricciones, asegurando la calidad de los datos recopilados.

Tabla 4.4: Diccionario de datos del esquema lógico

Tabla	Atributo	Tipo de dato	PK / FK	Restricciones / Comentarios	Tabla
User	user_id	SERIAL	PK	Identificador único del usuario.	User
	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.	Player
	user_id	INTEGER	FK →	Relaciona el alias con el	

			User(user_id)	usuario dueño.	
	alias	TEXT		Alias único dentro del estudio.	
Map	map_id	SERIAL	PK	Identificador del mapa.	Map
	codigo_map	VARCHAR(50)		Código interno del motor del juego.	
	nombre_oficial	VARCHAR(200)		Nombre descriptivo.	
Sector	sector_id	SERIAL	PK	Identificador del sector.	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.	Game
	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_segundos	INTEGER		Validación de duración.	
	episodio	INTEGER		Episodio jugado.	
TelemetryEvent	event_id	SERIAL	PK	Identificador del evento.	TelemetryEvent
	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.	
UXInstrument	instrument_id	SERIAL	PK	Identificador del instrumento UX.	UXInstrument
	nombre	VARCHAR(100)		Nombre del cuestionario.	
	tipo	VARCHAR(50)		Tipo o categoría del instrumento.	
UXResponse	response_id	SERIAL	PK	Identificador de la respuesta.	UXResponse
	user_id	INTEGER	FK → User (user_id)	Usuario que respondió.	
	instrument_id	INTEGER	FK → UXInstrument (instrument_id)	Instrumento aplicado.	

			t_id)		
	fecha_respuesta	TIMESTAMP		Registro temporal.	
UXItem	item_id	SERIAL	PK	Identificador del ítem.	UXItem
	instrument_id	INTEGER	FK → UXInstrumente(instrument_id)	Pertenece a un instrumento.	
	texto_pregunta	TEXT		Texto de la pregunta.	
	dimension	VARCHAR(100)		Dimensión psicológica.	
UXResponseItem	response_id	INTEGER	FK → UXResponse(response_id)	Relaciona respuesta con ítem.	UXResponseItem
	item_id	INTEGER	FK → UXItem(item_id)	Ítem respondido.	
	valor_likert	INTEGER		CHECK (valor_likert BETWEEN 1 AND 7).	

4.5. Integridad referencial y restricciones operacionales

La implementación de la integridad referencial a través de Claves Foráneas (FK) es vital para asegurar que las referencias a las dimensiones sean válidas.

- **Restricciones de integridad referencial:**

- Para la relación entre Game y TelemetryEvent, se aplica ON DELETE CASCADE. Esto significa que si se elimina una sesión (por ejemplo, por corrupción de datos), todos sus eventos asociados deben eliminarse automáticamente, ya que no tienen sentido fuera de ese contexto.
- Sin embargo, para las referencias a las dimensiones principales (Player, Map, User), se aplica ON DELETE RESTRICT (o NO ACTION). Esto evita la eliminación accidental de datos de dimensión que son referenciados por millones de filas de hechos, preservando la coherencia analítica.
- **Restricciones de comprobación:** Se deben utilizar restricciones de comprobación para validar la calidad de los datos de investigación. Por ejemplo, en la tabla de referencia se puede imponer un CHECK para asegurar que el Tipo_Causa solo pueda ser 'Bug_Vanilla', 'Limite_Tecnico', o 'Error_Jugador', lo que facilita la clasificación necesaria para el análisis de la autenticidad retro de *Chocolate-Doom*.

5. Consideraciones avanzadas y conclusiones

5.1. Implementación de requisitos éticos en el diseño lógico

El requisito ético de segregación de la PII se resuelve mediante un diseño en capas que aísla la información sensible. La clave principal de la tabla JUGADOR es el Jugador_ID, una clave sucedánea opaca que no contiene información directamente identificable.

Para cumplir con la gobernanza de datos , se debe introducir una tabla auxiliar de alto control de acceso:

- **Tabla lógica adicional (PII):** El sistema asegura el cumplimiento ético mediante la anonimización de datos personales, limitando la identificación directa a los campos estrictamente necesarios en la tabla *User*.

Dado que la inmensa mayoría de la base de datos de telemetría solo referencia el Jugador_ID (la clave sucedánea sin significado externo), el diseño garantiza que el núcleo de la investigación es anónimo, reduciendo la exposición a riesgos legales y de privacidad.

5.2. Preparación para la fase de diseño físico

Aunque el diseño lógico es conceptualmente independiente del Sistema de Gestión de Bases de Datos (DBMS), la necesidad de cumplir con la alta tasa de inserción (RNF-PER01) requiere consideraciones para la optimización física.

- **Indexación:** Se recomienda la indexación intensiva de todas las claves foráneas en las tablas de hechos de alta cardinalidad, especialmente *game_id* y *map_id* en *TelemetryEvent*. La velocidad de las consultas analíticas depende directamente de la eficiencia de los *joins* a través de estas claves.
- **Particionamiento:** Dada la naturaleza de serie temporal de la telemetría, la tabla *EVENTO_DE_TELEMETRIA* debe ser particionada. El particionamiento basado en la fecha o el rango de la *Marca_Tiempo_Absoluta* o la *Hora_Inicio* de la sesión mejorará significativamente el rendimiento para consultas de análisis de tendencias.

5.3. Conclusiones del diseño lógico

El diseño lógico para la Base de Datos de Telemetría y UX de *Chocolate-Doom* se ha desarrollado con un enfoque en la rigurosidad científica y la escalabilidad operativa. La adhesión a la Tercera Forma Normal (3NF) garantiza la integridad y la coherencia de los datos históricos, esenciales para la investigación. La utilización estratégica de claves auto-incrementales aborda los requisitos no funcionales de rendimiento en entornos de alta ingesta de datos.

Finalmente, el diseño propuesto cumple con los requisitos de telemetría y análisis de experiencia de usuario, asegurando integridad, consistencia y escalabilidad del sistema mediante la 3NF. Al segregar la información personal identificable, el esquema cumple con los requisitos éticos de gobernanza de datos, lo que resulta en un modelo de datos robusto, extensible y apto para la publicación académica (DBS 2025-30)

6. Instrumento User Experience

Este proyecto incorpora el instrumento PENS (Player Experience of Need Satisfaction) como instrumento de evaluación de la experiencia del usuario de los jugadores en *Chocolate Doom*. El cuestionario fue descargado desde su fuente original y posteriormente traducido y adaptado al español para el contexto de este proyecto. El cuestionario se encuentra dividido en cinco dimensiones, cada una medida en ítems en escala de Likert de 1 a 7 (donde 1 = Totalmente en desacuerdo y 7 = Totalmente de acuerdo).

Dimensión Competencia:

1. Me siento competente en el juego.
2. Me siento muy capaz y eficaz cuando juego.
3. Mi habilidad para jugar está bien equilibrada con los desafíos del juego.

Dimensión Autonomía:

1. El juego me ofrece opciones y decisiones interesantes.
2. El juego me permite hacer cosas interesantes.
3. Sentí mucha libertad dentro del juego.

Dimensión Relación

1. Las relaciones que formo en el juego me resultan satisfactorias.
2. Considero importantes las relaciones que establezco en el juego.
3. No me siento cercano/a a otros jugadores. (*Pregunta invertida*)

Dimensión Presencia/Inmersión

1. Al jugar, siento que me transporto a otro tiempo y lugar.
2. Explorar el mundo del juego se siente como realizar un viaje real.
3. Al moverme dentro del mundo del juego, siento como si realmente estuviera allí.
4. No me afectan emocionalmente los eventos del juego. (*Pregunta invertida*)
5. El juego fue emocionalmente envolvente.
6. Experimenté emociones tan intensas en el juego como en la vida real.
7. Al jugar, sentí que era parte de la historia.
8. Cuando logré algo en el juego, sentí orgullo genuino.
9. Reaccioné ante los eventos y personajes del juego como si fueran reales.

Dimensión Controles intuitivos

1. Aprender los controles del juego fue fácil.
2. Los controles del juego son intuitivos.
3. Cuando quise hacer algo en el juego, fue fácil recordar qué control debía usar.

Fuentes citadas

Chocolate Doom - The Doom Wiki at DoomWiki.org, https://doomwiki.org/wiki/Chocolate_Doom
2. Chocolate Doom, https://www.chocolate-doom.org/wiki/index.php/Chocolate_Doom 3. Explicación del modelado de datos: Conceptual, físico y lógico - Couchbase, <https://www.couchbase.com/blog/es/conceptual-physical-logical-data-models/> 4. Modelado dimensional: Claves primarias y foráneas - IBM, <https://www.ibm.com/docs/es/ida/9.1.2?topic=entities-primary-foreign-keys> 5. Modelo de datos — documentación de Bases de Datos I | Proyecto II - 0.0.1, <https://manual-tecnico-bd-oracle.readthedocs.io/es/latest/Modelo%20de%20datos.html> 6. La gestión ética de los datos - IADB Publications, <https://publications.iadb.org/es/la-gestion-etica-de-los-datos> 7. Create an Entity Relationship Diagram in Lucidchart - Lucid Help Center,

<https://help.lucid.co/hc/en-us/articles/16471565238292-Create-an-Entity-Relationship-Diagram-in-Lucidchart> 8. Clave primaria, integridad referencial, comprobación y restricciones de unicidad - IBM,
<https://www.ibm.com/docs/es/db2/11.1.0?topic=concepts-primary-key-referential-integrity-check-unique-constraints>