

Proyecto Semestral: Modelo de Datos

Bases de Datos

“Prestigio-2A”

Diego Altamirano Javier Mendoza
diego.altamiranot@sansano.usm.cl javier.mendoza@sansano.usm.cl

Felipe Quintanilla Kevin Reyes
felipe.quintanilla@sansano.usm.cl kevin.reyesv@sansano.usm.cl

Carlos Buil A.
cbuil@inf.utfsm.cl

28 de Septiembre

Introducción

Debido al contexto de pandemia, la demanda de plataformas de juegos digitales ha incrementado en una considerable medida; cada vez se invierte más en publicidad atrayendo a más usuarios (tal como se ha visto con Epic, por ejemplo). Por esto, la Universidad ha decidido que este es un buen rubro para invertir y convertir en una fuente de ingresos (y prestigio).

A ustedes, alumnos de Bases de Datos, se les ha destinado la tarea de diseñar e implementar el back-end de una plataforma que logre competir con las grandes empresas insertas en el mercado actual. Por supuesto, la institución confía en su capacidad de diseño y análisis del negocio para proponer una solución tecnológica que logre, en menor medida, replicar a la competencia.

TLDR; Make Steam 2

Contexto

Si ustedes no están familiarizados con el contexto de plataformas de juegos, el diseño base de la plataforma es similar a una tienda de retail: El sistema se encarga de almacenar y vender **Juegos** (como **este**) y otros Contenidos Digitales (Expansiones, Banda Sonora, Algunos Programas, etc). Al registrarse en la página se obtiene acceso a la **Tienda**, donde se le muestran todos los juegos disponibles. Tras la compra de uno, este se añade a su **Biblioteca**, la cual contiene todos los juegos comprados por el usuario. Un usuario puede jugar a cualquier juego que esté en su biblioteca.

Se recomienda que, para investigar más a fondo, visite páginas de plataformas conocidas, tales como **Steam**, **Epic Games** o **GoG**. El registro es gratuito y su exploración también.

Modelo de Datos

En esta entrega, cada grupo tiene la misión de diseñar un modelo que cumpla la función de una plataforma de venta de juegos. Algunas funcionalidades que su modelo debe cumplir son:

- Debe existir una entidad Juego y una entidad Usuario.
- Un Usuario puede comprar un Juego.
 - Es su decisión si el usuario es permanente propietario, o la compra sólo representa un préstamo o arriendo del producto. Para estudiar esta diferencia, se recomienda leer la sección **Información Extra: DRM**.
- Cada compra se realiza en una fecha/hora particular, es relevante almacenarla.
- Los Juegos pueden estar en oferta, con un porcentaje de descuento. Además, existen los "bundles" (paquetes), que ofrecen un conjunto de juegos con un porcentaje de descuento.

Respecto a la sintaxis y semántica de su modelo, este debe cumplir:

- Debe tener un mínimo de 10 tablas.
- Debe poseer al menos una relación M:N con su entidad intersección (esta última sí contribuye al mínimo de tablas).
- Las claves primarias deben ser indicadas explícitamente (no pueden quedar tablas sin su clave primaria).
- Las claves foráneas deben ser indicadas explícitamente. Además, deben ser llamadas de la misma forma que su referencia (es decir, el mismo nombre que el atributo de la tabla original).
- Declare información relevante para el sistema.
- Cada atributo debe tener su tipo de dato asociado (que tenga sentido). Además, debe indicar si puede ser NULLABLE o no.
- Su diagrama debe ser realizado utilizando MySQL WorkBench.

API REST en FLASK

Además, deberá desarrollar una API (Application Programming interface) en el framework FLASK de Python, la cual utilizará para crear, leer, actualizar y eliminar datos de su base de datos (las operaciones mencionadas son conocidas como **CRUD**).

- El código de Python deberá ser subido al repositorio de Gitlab de su grupo.
- Debe entregar 1 ruta para cada operación del CRUD, es decir, **4 rutas** en total (Create, Read, Update, Delete).
- La API a desarrollar será sólo para **una entidad** en esta primera entrega. Por ejemplo, si selecciona la entidad **Juego**, entonces deberá desarrollar una API que le permita:
 - Insertar un juego.
 - Actualizar datos de un juego.
 - Leer datos de un juego.
 - Eliminar un juego.

Recomendaciones

- No se esmeren en llenar las tablas con atributos, considere sólo aquellos que le den consistencia al sistema. A su vez, si su entidad no es una intersección, busque añadir al menos un atributo no primario/foráneo.
- Si sienten que su conocimiento sobre estas plataformas es muy limitado (y la exploración no rinde frutos), no duden en consultar a sus compañeros o a nosotros ayudantes por los canales de consultas.
- Desde lo anterior, si desea crear supuestos o consideraciones para su modelo, por favor documéntelas y adjúntelas en sus entregas. Esto nos ayudará mucho en los tiempos y eficiencia de corrección.
- Esta entrega sirve como pilar para el resto de su proyecto semestral, por lo que un buen desarrollo les puede simplificar el trabajo futuro. Aún así, es muy probable que su modelo sufra modificaciones en el transcurso del semestre, por lo que no se preocupen, se podrá mejorar.

Condiciones Generales

- El formato de entrega es un archivo .mwb de su modelo. Además, un pdf en el que deberán adjuntar una imagen del modelo, junto con los supuestos y consideraciones que tuvieron al desarrollarlo. Además, en este mismo deben estar presentes las rutas de acceso a su API realizada en FLASK.
- Para la entrega e implementación de la API, ustedes deberán subir el código a su repositorio en GitLab. Para mantener el orden del mismo (y de las futuras entregas), le pediremos que arme un repositorio con el nombre de su equipo, y arme una carpeta de nombre "*Entrega_1*". Dentro de esta deben incluir todos los archivos necesarios.
- La fecha de entrega es el **Jueves 15 de Octubre** a las **23:55 PM**, tareas atrasadas recibirán un **descuento** lineal hasta las 14:00 PM del día Viernes 16 de octubre.

Información Extra: DRM

A pesar que las plataformas más utilizadas hoy en día son Steam y Epic (entre otros), muchas de estas están sujetas a las prácticas y tecnologías de la **Gestión de derechos digitales**¹ (DRM por sus siglas en inglés), la cual limita las atribuciones que obtiene un usuario o cliente al comprar un producto digital.

En síntesis, una plataforma sujeta a DRM cambia las reglas del juego, tal que un comprador no se convierte en el "propietario" de un producto, si no más bien en un "arrendatario" del mismo. Esto implica que, en cualquier momento, la compañía puede requisar el acceso al ítem digital y cerrarlo permanentemente si así lo desea.

Aún así, existen plataformas que no siguen estas prácticas y buscan fomentar el software libre, y la legítima propiedad de un producto para el usuario. Un ejemplo de esto es GoG, la cual se proclama directamente en contra de la iniciativa DRM², haciendo un llamado a educar a las personas y fomentar la discusión sobre este tema.

¹Gestión de derechos digitales - Wikipedia: https://es.wikipedia.org/wiki/Gesti%C3%B3n_de_derechos_digitales

²Anuncio oficial de GoG: https://www.gog.com/news/the_fck_drm_initiative