

Prueba de evaluación del resultado de aprendizaje:

IB04 - Diseñar una base de datos relacional a partir de la especificación de un problema

Criterios de evaluación y de calificación

- La **calificación** de este resultado de aprendizaje es de **2 puntos** de la nota final.
- Los niveles **aprobado** y **avanzado** se detallan en los [criterios publicados en el AV](#).
- Se debe demostrar el nivel **aprobado** en la realización del **diseño conceptual**. Este nivel otorga 0,5 puntos. Se pueden alcanzar hasta 0,5 más si no se cometen errores (nivel **avanzado**).
- Se debe demostrar el nivel **aprobado** en la realización del **diseño lógico**. Este nivel otorga 0,5 puntos. Se pueden alcanzar hasta 0,5 más si no se cometen errores (nivel **avanzado**).
- Para **aprobar** este resultado de aprendizaje se debe demostrar el nivel aprobado en el **diseño conceptual** y en el **diseño lógico**.
- En caso de no superar esta prueba, tendrás una **nueva oportunidad** con fecha de entrega el día del examen de la primera convocatoria: el 2 de junio de 2020.

Al hacer esta prueba te comprometes a cumplir el siguiente código de comportamiento:

1. Hacer la prueba de forma **individual**, sin recibir ayuda por parte de otras personas.
2. **No intercambiar** ningún tipo de información sobre el contenido de esta prueba ni sobre tus respuestas hasta finalizar el plazo de entrega: a las **15:00 del martes 19 de mayo de 2020**.

El incumplimiento de alguna de las normas de comportamiento puede comportar la calificación de suspenso en esta prueba y en la asignatura.

Si tienes dudas sobre el **enunciado** puedes preguntarlas a la profesora por la [tutoría virtual](#), pero antes de hacerlo te aconsejamos leer con calma el enunciado y reflexionar: pensar en el uso que se va a dar a estos datos es fundamental, porque comprenderlos es el primer paso para hacer un diseño correcto.

En tus soluciones, añade **comentarios** para explicar las decisiones de diseño que consideres conveniente.

Cuando acabes, sube tus **soluciones** en formato PDF para que la profesora pueda hacer anotaciones desde el aula virtual. Los formatos de imagen no siempre se pueden comentar, por eso solo se aceptan ficheros en formato PDF (puedes subir varios).

Descripción de requisitos

Tenemos una empresa de programación de juegos para móviles y queremos recoger datos sobre el comportamiento de los jugadores para estudiar cómo están funcionando las distintas estrategias de monetización que utilizamos en nuestros juegos. Es lo que se denomina *game metrics*¹.

La base de datos que debes diseñar deberá permitirnos guardar los datos de las instalaciones que se han hecho de nuestros juegos y todas las compras que los jugadores han realizado. Mediante la recogida de estos datos será posible obtener diversas estadísticas como por ejemplo: el porcentaje de usuarios que han hecho alguna compra en un día determinado, el número de usuarios que han hecho su primera compra en un período de tiempo concreto, el número de usuarios que vuelven a jugar después de un número determinado de días desde que instalaron el juego, el tiempo medio que transcurre entre que un usuario pierde una vida y realiza una compra, etc. **En los párrafos que siguen se describen con detalle los datos que deben almacenarse en la base de datos. Presta mucha atención porque esta base de datos es fundamental para nuestro negocio.**

¹ Para elaborar este enunciado hemos usado información de la web gameanalytics.com.

Cada vez que se instala un juego en un móvil se debe guardar en la base de datos la información que interesa para poder hacer el rastreo de datos y las posteriores analíticas: nombre del juego instalado, *store* de la que ha descargado el juego (App Store, Google Play, etc.) e identificador del usuario usado para la descarga, modelo del dispositivo móvil y sistema operativo (Android, iOS, etc.), fecha y hora de instalación (*timestamp*) y país. A cada instalación de la que guardemos datos se le asignará un identificador numérico que será único y que se guardará tanto en la base de datos como en el móvil en el que se ha instalado el juego. Así, cuando el móvil nos mande los datos rastreados, adjuntará el identificador de la instalación a la que corresponden.

Además de los datos que se han descrito en el párrafo anterior, hay que guardar la versión que se ha instalado del juego y los recursos iniciales de los que se dispone en dicha instalación, que son dos: el número de vidas o de partidas (a este recurso le llamaremos vidas/partidas) y la cantidad de tiempo o de oro (le llamaremos tiempo/oro). Estos recursos iniciales pueden ser distintos en cada instalación del mismo juego si corresponde a versiones distintas del juego o corresponde a sistemas operativos diferentes. Por ejemplo, actualmente el juego *Cocktail vs. Mocktail* tiene disponible para iOS la versión 2.3.10 con unos recursos iniciales de 10 vidas y 50 unidades de tiempo, mientras que para Android tiene disponible la versión 2.2.00 con unos recursos iniciales de 5 vidas y 100 unidades de tiempo. Es necesario guardar los recursos iniciales para cada instalación ya que se pretende hacer estudios detallados sobre la influencia de los valores iniciales de los recursos sobre el comportamiento de los usuarios. Por ejemplo, se puede estudiar cómo el número de vidas/partidas iniciales influye en el tiempo que se tarda en hacer la primera compra.

A continuación se describe la información de rastreo que se recogerá en el móvil y que se enviará a nuestra empresa para guardarla en la base de datos y así poder obtener las estadísticas que nos interesan. Cada vez que un usuario abre una sesión del juego accediendo a la instalación en su móvil, se registrará el instante de inicio de sesión y se irán registrando también todos los eventos que sucedan asociados a dicha sesión. Cada evento tiene un tipo de flujo, que puede ser: ganar un recurso (*source*) o perder un recurso (*sink*). Un ejemplo de flujo *sink* es cuando se pierde una vida porque se empieza a jugar; un ejemplo de flujo *source* es cuando se gana tiempo viendo un anuncio.

De cada evento que suceda se debe guardar: el instante de tiempo, de qué tipo de flujo es (*sink* o *source*), a qué recurso afecta (podrá ser un valor entre: vida, partidas, tiempo, oro) y la cantidad en que el recurso se ve afectado (en valor absoluto). Por ejemplo: se guarda un evento de tipo *sink* cuando un jugador compra un botijo pagando 50 unidades de oro y se guarda un evento de tipo *source* cuando un jugador compra 100 unidades de oro pagando 0,99€. Los eventos que conllevan una transacción económica deben guardar también el importe de ésta.

- 1) **Diseño conceptual.** A partir de los requisitos de datos que se acaban de describir dibuja un esquema conceptual utilizando el modelo entidad/relación que se ha utilizado en la asignatura (el que se usa en el libro o el de la aplicación Dia). Asegúrate de que en el esquema no falta ninguno de los datos especificados y que todos están colocados en la entidad o la relación que les corresponde. No olvides que debes poner la cardinalidad en todas las relaciones y que la cardinalidad por defecto de los atributos es (1,1), así que si es distinta de ésta, debes especificarla.
- 2) **Diseño lógico.** A partir del esquema conceptual dibujado en el apartado anterior obtén las tablas de la base de datos relacional correspondiente (esquema lógico). Haz un diagrama como los que aparecen en el libro o como los que genera la aplicación Vertabelo. Recuerda que para cada tabla debes indicar:
 - los nombres de todas las columnas y cuáles aceptan nulos (no es necesario especificar el tipo de datos),
 - la/s columna/s que forman la clave primaria (PK),

