Instrucciones

- <u>Descargue el proyecto del examen del siguiente repositorio: https://github.eii.us.es/IISSI-2223/examen-lab-iissi.git .</u>
- Prepare el proyecto creando la BD ModeloExamen en HeidiSQL, revisando settings.py, y ejecutando "silence createdb", "silence createapi", "silence createtests" y finalmente "silence run".
- Implemente la solución a los ejercicios en los archivos **sql/solucion.sql** (para código sql) y **tests/solucion.http** (para peticiones http). Estos ejercicios están basados en unos requisitos adicionales que se suministran a continuación.
- <u>Incluya capturas de pantalla HeidiSQL/vsCode que muestre el resultado de ejecutar el ejercicio al</u> final de este documento.
- Debe subir a la actividad de EV este documento con las capturas de pantallas, y los archivos sql/solucion.sql y tests/solucion.http. Antes de subir la actividad, avise a un profesor para que valide la misma.
- Es necesario completar un nivel para que se valoren los siguientes niveles.

Catálogo de requisitos

RI-1-01	Requisito de información
Como:	Profesor de la asignatura
Quiero:	Tener disponible información sobre Concursos : Un concurso es una competición en la que se intenta
	derrotar a los oponentes en un videojuego. Sus atributos son: el videojuego del concurso, el nombre
	del concurso, la cuantía del premio, y la fecha en la que tiene lugar. Todos los atributos son
	obligatorios.
Para:	Evaluar al estudiante

RN-1-01	Primera regla de negocio
Como:	Profesor de la asignatura
Quiero:	No puede haber varios concursos de un mismo videojuego el mismo día.
Para:	Evaluar al estudiante

RN-1-02	Segunda regla de negocio
Como:	Profesor de la asignatura
Quiero:	La cuantía del premio debe ser mayor o igual a 10€, y menor o igual que 500€.
Para:	Evaluar al estudiante

RN-1-03	Tercera regla de negocio
Como:	Profesor de la asignatura
Quiero:	La fecha del concurso debe ser anterior al año 2050.
Para:	Evaluar al estudiante

PA- 1-01	Inserción de nuevas entidades
Como:	Profesor de la asignatura
Quiero:	• Insertar Concurso titulado "Concurso 1" del videojuego con ID=2, con premio de 20€, en el
	01/01/2020.

- Insertar Concurso titulado "Concurso 2" del videojuego con ID=2, con premio de 100€, en el 15/04/2019.
- Insertar Concurso titulado "Concurso 3" del videojuego con ID=1, con premio de 70€, en el 10/10/2021.
- Insertar Concurso titulado "Concurso 4" del videojuego con ID=1, con premio de 70€, en el 01/01/2100. (RN-1-03)
- Insertar Concurso titulado "Concurso 5" del videojuego con ID=55, con premio de 70€, del año 2021. (referencia no existente)
- Insertar Concurso titulado "Concurso 6" del videojuego con ID=2, con premio de 70€, en el 01/01/2020. (RN-1-01)

Para: Evaluar al estudiante.

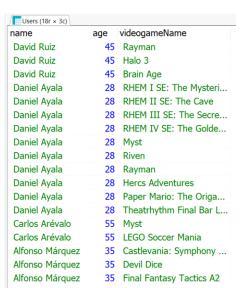
Nivel C. Máximo 7 puntos

- 1.(2)- Implemente los requisitos proporcionados (RI/RN) en MariaDB sin usar triggers.
- 2.(1)- Use Insert para implementar las pruebas de aceptación. Incluya captura de pantalla con los datos de la tabla después de las pruebas.

Use Delete para eliminar el concurso con ID=1. Incluya captura de pantalla con los datos de la tabla después de la llamada.

Use Update para actualizar los concursos con IDs 2 y 3 para que tengan 300€ de premio. Incluya captura de pantalla con los datos de la tabla después de la llamada.

- 3.(0.5)- Escriba y realice peticiones HTTP para obtener el listado de videojuegos de la base de datos.
- 4.(1)- Cree una consulta que devuelva el nombre y edad de todos los usuarios, y los nombres de sus videojuegos:



- 5.(1.5)- Cree un disparador que antes de insertar en la nueva tabla comprueba si el premio está por debajo del mínimo existente en la tabla. Si lo está, muestra un mensaje de error incluyendo el valor que se intentó dar y el mínimo. Pruebe el disparador. **Incluya captura de pantalla con los datos de la tabla después de que se muestre el mensaje de error.**
- 6.(0.5)- Cree una función que, a partir del identificador de un Concurso, devuelva la cantidad de concursos con un premio mayor que el de dicho Concurso. Use la función para obtener el resultado a partir del elemento con ID=2. Incluya capturas de pantallas.

(0.5)- Cree un procedimiento que, a partir del identificador de un Concurso, lo borre solo si el resultado de la función anterior es mayor a 0.

Nivel B. Máximo 9 puntos

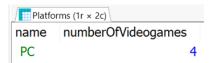
7.(1)- Cree una consulta que devuelva el nombre de cada usuario y el número de videojuegos que tiene cada uno:



8.(1)- Implemente las pruebas de aceptación como peticiones POST HTTP. **Recuerde limpiar la tabla previamente para evitar conflictos**. Incluya capturas de pantallas. Las peticiones POST se pueden realizar con o sin autenticación. Incluya capturas de pantallas.

Nivel A. Máximo 10 puntos

9.(0.5)- Cree una consulta que devuelva el nombre de la plataforma con más videojuegos a partir de 2005, y ese número de videojuegos:



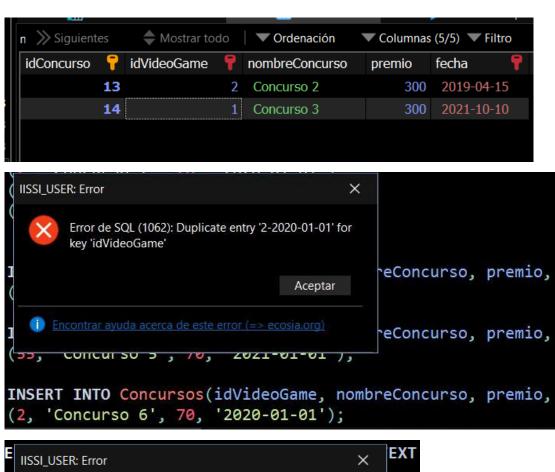
10.(0.5)- Cree un procedimiento **pAddTwoContests(...)** que, dentro de una transacción, inserte dos Concursos a partir de los datos suministrados de los dos.

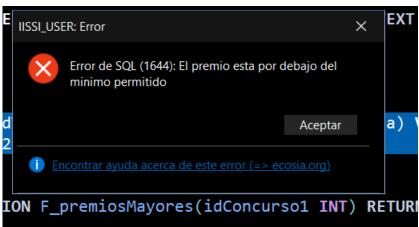
Realice dos llamadas: una que inserte dos correctamente, y una en la que el segundo rompa alguna restricción y aborte la transacción. Incluya capturas de pantallas

Capturas de pantalla

Inserte aquí las capturas de pantalla COMPLETA que muestre la ejecución de los ejercicios. Para realizar una captura de pantalla puede usar la tecla "impr pant" del teclado y pulsar "ctrl+v" para pegar la captura de pantalla.

```
CREATE OR REPLACE TABLE Concursos(
    idConcurso INT NOT NULL AUTO_INCREMENT,
    idVideoGame INT NOT NULL,
    nombreConcurso VARCHAR(60) NOT NULL,
    premio INT NOT NULL,
    fecha DATE NOT NULL,
    PRIMARY KEY (idConcurso),
    FOREIGN KEY (idVideoGame) REFERENCES Videogames(idVideoGame),
    CONSTRAINT RN2 CHECK (premio>=10 AND premio<=500),
    CONSTRAINT RN3 CHECK (YEAR(fecha) < 2050),
    UNIQUE(idVideoGame, fecha)
 );
modeloexamen.concursos: 3 filas en total (aproxin 🔊 Siguientes
                                                       ♠ Mostrar todo
▼ Ordenad
            idVideoGame
                             nombreConcurso
 idConcurso
                                                            fecha
                                                  premio
            11
                              2 Concurso 1
                                                        20 2020-01-01
            12
                              2 Concurso 2
                                                       100 2019-04-15
            13
                              1 Concurso 3
                                                        70 2021-10-10
                                             mio<=500),
    IISSI USER: Error
);
          Error de SQL (4025): CONSTRAINT `RN3` failed for
           'modeloexamen'.'concursos'
INSE
                                              ncurso, premio, fecha) VALUES
(2,
                                    Aceptar
(2,
(1,
      Encontrar ayuda acerca de este error (=> ecosia.org)
INSERT INTO Concursos(idVideoGame, nombreConcurso, premio, fecha) VALUES
(1, 'Concurso 4', 70, '2100-01-01');
     CONSTRAINT KNZ CHECK (Premio>=10 AND
                                                      2050)
   IISSI USER: Error
          Error de SQL (1452): Cannot add or update a child
          row: a foreign key constraint fails
                                                      reConcurso, premio,
          ('modeloexamen'.'concursos', CONSTRAINT
          `concursos_ibfk_1` FOREIGN KEY (`idVideoGame`)
          REFERENCES 'videogames' ('idVideogame'))
                                          Aceptar
                                                      reConcurso, premio,
    Encontrar ayuda acerca de este error (=> ecosia.org)
 INSERT INTO Concursos(idVideoGame, nombreConcurso, premio,
  (55, 'Concurso 5', 70, '2021-01-01');
```





```
237 DELIMITER;
 238
      SELECT F_premiosMayores(13);
 239
 240
 241 DELIMITER //
 242 CREATE OR REPLACE PROCEDURE F_borra
 243 BEGIN
 244 DECLARE f1 INT;
 245 SET f1 = (SELECT F_premiosMayores(id
          IF (f1=0) THEN DELETE FROM concu
 246
 Resultado #1 (1r × 1c)
F_premiosMayores(13)
                       250
                       251 CALL F_borrarConcurso(13);
                       252
                         Filtro: Expresión regular
ALL F_borrarConcurso(13);
```

Filas afectadas: 0 Filas encontradas: 0 Advertencias: 0 Durac

```
212
      UPDATE concursos SET premio=300 WHERE idConcurso=13;
 213
      UPDATE concursos SET premio=300 WHERE idConcurso=14;
214
215
216
      SELECT users.name, users.age, videogames.name
217
      FROM users, videogames;
Resultado #1 (96r × 3c)
name
David Ruiz
                   45 RHEM I SE: The Mysterious Land
Daniel Ayala
                   28 RHEM I SE: The Mysterious Land
Carlos Arévalo
                   55 RHEM I SE: The Mysterious Land
Alfonso Márquez
                   35 RHEM I SE: The Mysterious Land
David Ruiz
                   45 RHEM II SE: The Cave
Daniel Ayala
                   28 RHEM II SE: The Cave
Carlos Arévalo
                   55 RHEM II SE: The Cave
                   35 RHEM II SE: The Cave
Alfonso Márquez
```