Instrucciones

- Descargue el proyecto del examen del siguiente repositorio: https://github.eii.us.es/IISSI-2223/examen-lab-iissi.git .
- 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.
- Incluya capturas de pantalla HeidiSQL/vsCode que muestre el resultado de ejecutar el ejercicio al 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 **Noticias**: Una noticia es una publicación sobre algún videojuego. Sus atributos son: el videojuego, el titular de la noticia, el número de palabras de su texto, y la fecha de publicación. Todos los atributos son obligatorios.
Para:	Evaluar al estudiante

RN-1-01	**Primera regla de negocio**
Como:	Profesor de la asignatura
Quiero:	No se puede publicar más de una noticia por videojuego al día.
Para:	Evaluar al estudiante

RN-1-02	**Segunda regla de negocio**
Como:	Profesor de la asignatura
Quiero:	El número de palabras debe ser superior o igual a 100 e inferior o igual a 2000.
Para:	Evaluar al estudiante

RN-1-03	**Tercera regla de negocio**
Como:	Profesor de la asignatura
Quiero:	El año de la fecha debe ser anterior a 2099.
Para:	Evaluar al estudiante

PA- 1-01	Inserción de nuevas entidades
Como:	Profesor de la asignatura
Quiero:	✔ Insertar Noticia titulada “Noticia 1” sobre el videojuego con ID=2, con 105 palabras, publicada el 02/08/2022. ✔ Insertar Noticia titulada “Noticia 2” sobre el videojuego con ID=2, con 200 palabras, publicada el 06/05/1993. ✔ Insertar Noticia titulada “Noticia 3” sobre el videojuego con ID=1, con 1050 palabras, publicada el 01/01/2026. ✘ Insertar Noticia titulada “Noticia 4” sobre el videojuego con ID=1, con 7000 palabras, publicada el 05/05/2990. (**RN-1-03**) ✘ Insertar Noticia titulada “Noticia 5” sobre el videojuego con ID=16, con 200 palabras, publicada el 04/01/2020. (**referencia no existente**) ✘ Insertar Noticia titulada “Noticia 6” sobre el videojuego con ID=2, con 200 palabras, publicada el 02/08/2022. (**RN-1-01**)
Para:	Evaluar al estudiante.

Nivel C. Máximo 7 puntos

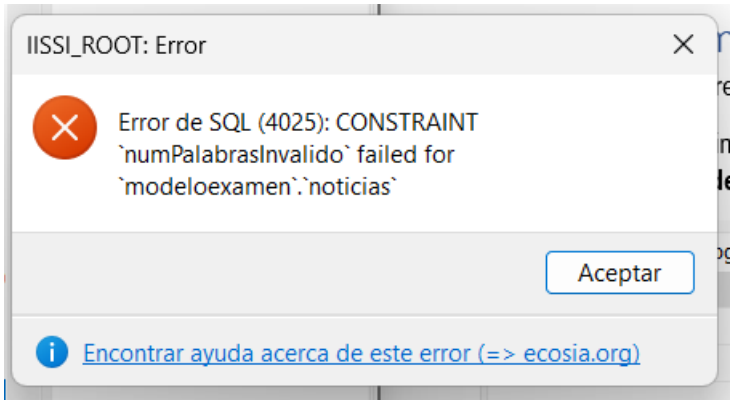
1.(2)- Implemente los requisitos proporcionados (RI/RN) en MariaDB sin usar triggers.

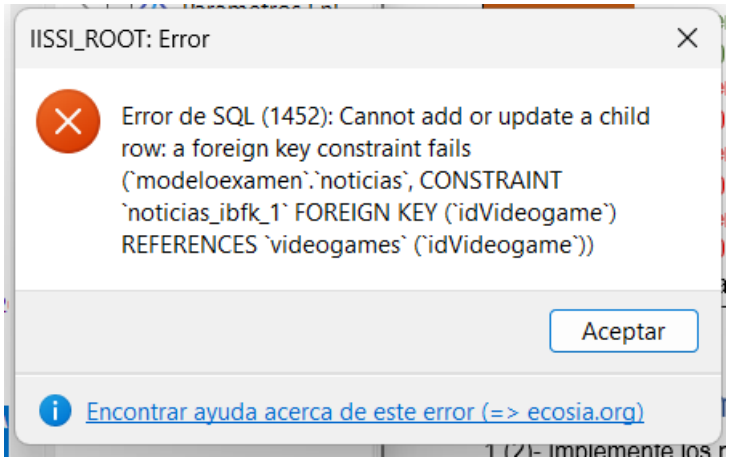
```
CREATE OR REPLACE TABLE Noticias(
   idNoticia INT NOT NULL AUTO_INCREMENT,
   idVideogame INT NOT NULL,
   titular VARCHAR(64) NOT NULL,
   numPalabras INT NOT NULL,
   fecha DATE NOT NULL,
   UNIQUE (idVideogame, fecha),
   PRIMARY KEY(idNoticia),
   FOREIGN KEY (idVideogame) REFERENCES videogames(idVideogame),
   CONSTRAINT numPalabrasInvalido CHECK (numPalabras>=100 AND numPalabras<=2000),
   CONSTRAINT fechaInvalida CHECK (YEAR(fecha)<2099)
);
```

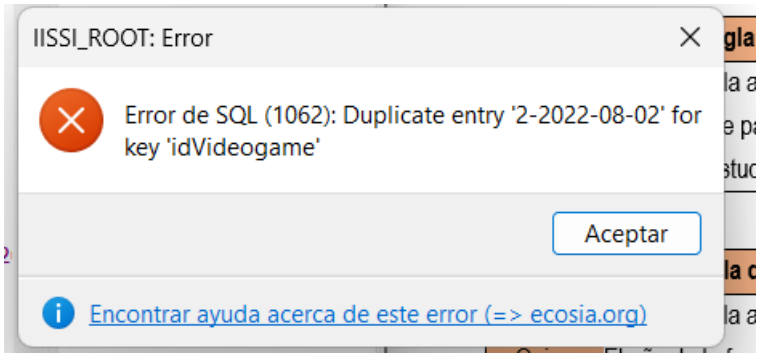
2.(1)- Use Insert para implementar las pruebas de aceptación. **Incluya captura de pantalla con los datos de la tabla después de las pruebas**.

```
INSERT INTO noticias(idNoticia, idVideogame, titular, numPalabras, fecha) VALUES
   (1, 2, 'noticia1', 105, '2022-8-2'),
   (2, 2, 'noticia2', 200, '1993-05-06'),
   (3, 1, 'noticia3', 1050, '2026-01-01'),
   (4, 1, 'noticia4', 7000, '2990-05-05'),
   (5, 16, 'noticia5', 200, '2020-01-04'),
   (6, 2, 'noticia6', 200, '2022-08-02')
```

idNoticia	idVideogame	titular	numPalabras	fecha
1	2	noticia1	105	2022-08-02
2	2	noticia2	200	1993-05-06
3	1	noticia3	1.050	2026-01-01







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

```
DELETE FROM  noticias
   WHERE idNoticia=1;
```

idNoticia	idVideogame	titular	numPalabras	fecha
2	2	noticia2	200	1993-05-06
3	1	noticia3	1.050	2026-01-01

Use Update para actualizar los elementos con IDs 2 y 3 para que tengan 300 palabras. **Incluya captura de pantalla con los datos de la tabla después de las llamadas**.

```
UPDATE noticias
   SET numPalabras=300
   WHERE idNoticia = 2 OR idNoticia = 3;
```

idNoticia	idVideogame	titular	numPalabras	fecha
2	2	noticia2	300	1993-05-06
3	1	noticia3	300	2026-01-01

3.(0.5)- Escriba y realice peticiones HTTP para obtener el listado de plataformas de la base de datos.

```
Send Request
get {{BASE}}/platforms
```


```
[
  {
    "idPlatform": 1,
    "isHandheld": null,
    "name": "PC"
  },
  {
    "idPlatform": 2,
    "isHandheld": 0,
    "name": "Playstation 1"
  },
  {
    "idPlatform": 3,
    "isHandheld": 1,
    "name": "Game Boy Advance"
  },
  {
    "idPlatform": 4,
    "isHandheld": null,
    "name": "Switch"
  },
  {
    "idPlatform": 5,
    "isHandheld": 0,
    "name": "Playstation 5"
  },
  {
    "idPlatform": 6,
    "isHandheld": 1,
    "name": "Nintendo DS"
  },
```

```
{
  "idPlatform": 7,
  "isHandheld": 0,
  "name": "XBox 360"
},
{
  "idPlatform": 8,
  "isHandheld": 1,
  "name": "Nintendo 3DS"
}
```


4.(1)- Cree una consulta que devuelva el precio máximo de los videojuegos con valoración de 9:

```
214  SELECT MAX(price) AS maxPrice9ScoredVideogame
215     FROM videogames
216     WHERE score = 9;
217
```

Resultado #1 (1r × 1c)

maxPrice9ScoredVideogame
59,99

Videogames (1r × 1c)

maxPrice9ScoredVideoga...
59,99

5.(1.5)- Cree un disparador que, antes de actualizarse una noticia, si la cantidad de palabras fuera menor que la mitad de la media existente en la tabla, la establezca como 1500 en su lugar. Pruebe el disparador. **Incluya captura de pantalla con los datos de la tabla después de ejecutar la llamada.**

```
delimiter //
CREATE OR REPLACE TRIGGER tActualizarNumPalabras
BEFORE UPDATE ON Noticias
FOR EACH ROW
BEGIN
   DECLARE mediaExistente INT;
   SELECT AVG(numPalabras) INTO mediaExistente FROM Noticias;
   IF NEW.numPalabras < mediaExistente / 2 THEN
     SET NEW.numPalabras = 1500;
   END IF;
END //
delimiter ;

UPDATE Noticias SET numPalabras = 100 WHERE idNoticia = 2;
```

idNoticia	idVideogame	titular	numPalabras	fecha
2	2	noticia2	1.500	1993-05-06
3	1	noticia3	300	2026-01-01

6.(0.5)- Cree una función que, a partir del identificador de una Noticia, devuelva la diferencia entre su número de palabras y el mínimo existente en la tabla. Use la función para obtener el resultado a partir del elemento con ID=2. Incluya capturas de pantallas.

```
234  delimiter //
235  CREATE OR REPLACE FUNCTION fDiferenciaPalabras(noticiaId INT) RETURNS INT
236  BEGIN
237    DECLARE minPalabras INT
238    SELECT MIN(numPalabras) INTO minPalabras FROM Noticias;
239    RETURN (SELECT numPalabras FROM Noticias WHERE idNoticia = noticiaId) - minPalabras;
240  END //
241  delimiter ;
242
243  SELECT fDiferenciaPalabras(4);
244
```

Resultado #1 (1r × 1c)

fDiferenciaPalabras(4)
49

(0.5) Cree un procedimiento que, a partir del identificador de una Noticia, la borre solo si el resultado de la función anterior es menor a 50.

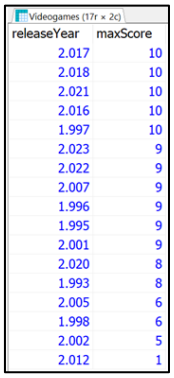
```
INSERT INTO noticias(idNoticia, idVideogame, titular, numPalabras, fecha) VALUES
   (4, 2, 'noticia1', 349, '2022-8-2')

DELIMITER //
CREATE OR REPLACE PROCEDURE pBorrarNoticia(noticiaId INT)
BEGIN
  DECLARE diferencia INT;
  SET diferencia = fDiferenciaPalabras(noticiaId);
  IF diferencia < 50 THEN
    DELETE FROM Noticias WHERE idNoticia = noticiaId
  END IF;
END //
DELIMITER ;

CALL pBorrarNoticia(4);
```

idNoticia	idVideogame	titular	numPalabras	fecha
2	2	noticia2	300	1993-05-06
3	1	noticia3	300	2026-01-01
4	2	noticia1	349	2022-08-02

idNoticia	idVideogame	titular	numPalabras	fecha
2	2	noticia2	1.500	1993-05-06
3	1	noticia3	300	2026-01-01

Nivel B. Máximo 9 puntos

7.(1)- Cree una consulta que devuelva la valoración máxima de los videojuegos de cada año, ordenados por valoración máxima de mayor a menor:



Videogames (17r × 2c)

releaseYear	maxScore
2.017	10
2.018	10
2.021	10
2.016	10
1.997	10
2.023	9
2.022	9
2.007	9
1.996	9
1.995	9
2.001	9
2.020	8
1.993	8
2.005	6
1.998	6
2.002	5
2.012	1

```
SELECT YEAR(releaseDate) AS releaseYear, MAX(score) AS maxScore
FROM Videogames
GROUP BY YEAR(releaseDate)
ORDER BY MaxScore DESC;
```

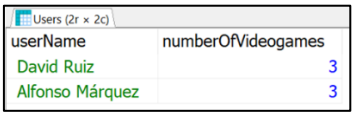
Videogames (17r × 2c)

releaseYear	maxScore
2.017	10
2.018	10
2.021	10
2.016	10
1.997	10
2.007	9
1.996	9
1.995	9
2.001	9
2.023	9
2.022	9
2.020	8
1.993	8
2.005	6
1.998	6
2.002	5
2.012	1

8.(1)- Implemente las pruebas de aceptación como peticiones POST HTTP. **Recuerde limpiar los datos de la tabla insertados en los apartados anteriores**. 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 los usuarios mayores de 30 años y que tengan más de 2 videojuegos, junto con el número de tales videojuegos:



Users (2r × 2c)

userName	numberOfVideogames
David Ruiz	3
Alfonso Márquez	3

10.(0.5)- Cree un procedimiento **pAddTwoNews(…)** que, dentro de una transacción, inserte dos Noticias a partir de los datos suministrados de las 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.