# Instrucciones

* Descargue el proyecto del examen del siguiente repositorio: <https://github.eii.us.es/IISSI-2324/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.
* Debe subir a la actividad de EV 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.  Un concurso puede ser online o presencial (clasificación **completa disjunta**). Los concursos online tienen el nombre de la web en la que se desarrolla, y los presenciales la ubicación donde tienen lugar. |
| 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. Online en la web “Twitch”. * Insertar Concurso titulado “Concurso 2” del videojuego con ID=2, con premio de 100€, en el 15/04/2019. Presencial en “Calle Avicena”. * Insertar Concurso titulado “Concurso 3” del videojuego con ID=1, con premio de 70€, en el 10/10/2021. Online en la web “Youtube”. * Insertar Concurso titulado “Concurso 4” del videojuego con ID=1, con premio de 70€, en el 01/01/2100. Presencial en “Sevilla” (**RN-1-03**) * Insertar Concurso titulado “Concurso 5” del videojuego con ID=55, con premio de 70€, del año 2021. Presencial en “Madrid” (**referencia no existente**) * Insertar Concurso titulado “Concurso 6” del videojuego con ID=2, con premio de 70€, en el 01/01/2020. Online en la web “Twitch” (**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. Implemente la herencia mediante una sola tabla con discriminante.

2.(1)- Use Insert para implementar las pruebas de aceptación.

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:

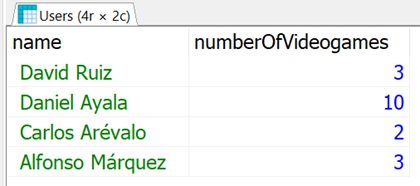


5.(1.5)- Cree un disparador que después de insertar una reseña, calcula el promedio de rating de todas las reseñas del videojuego. Luego, actualiza el campo score en Videogames para reflejar este promedio. Pruebe el disparador.

6.(1)- Cree una función que, a partir del uniqueCode de un videojuego, devuelva el número de logros (registrados en la tabla Achievements) asociados a ese videojuego. Pruebe la función.

# 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.

# Nivel A. Máximo 10 puntos

9.(1)- Cree un procedimiento **pAddTwoGenres(…)** que, dentro de una transacción, inserte dos Géneros 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.