Egibide

APRENDIZAJE COLABORATIVO BASADO EN RETOS

2023/2024

RETO: “CONSULTORÍA E-SPORT” (Formato alumno)

DAW

DESARROLLO DE APLICACIONES WEB

1

MÓDULOS IMPLICADOS
0484. Bases de datos (6 horas semanales) 0485. Programación (8 horas semanales) 0487. Entornos de desarrollo (3 horas semanales) 0373. Lenguajes de marcas y sistemas de gestión de información (4 horas semanales)

DURACIÓN	ORGANIZACIÓN
4 semanas y 2 días	Grupos de 4 personas

EL RETO

La empresa de E-Sport os ha contratado para que realicéis mejoras en su sistema de información. Desean una aplicación que gestione **competiciones** e-sport.

La aplicación contemplará dos etapas:

1. Etapa 1: inscripción de equipos y jugadores, así como la generación del calendario de los distintos enfrentamientos que se darán a lo largo de todas las jornadas.
2. Etapa del campeonato donde se guardarán los resultados de los enfrentamientos.

Para poder empezar con la etapa del campeonato, la etapa de inscripción tiene que estar cerrada.

De cada competición e-sport interesa conocer el nombre de la competición, la fecha de inicio y finalización de la misma, el juego, los equipos de e-sports que participan, así como el equipo ganador, si la competición ya se ha determinado. De cada juego se conocerá el nombre del juego, la empresa desarrolladora y la fecha de lanzamiento. Un mismo equipo puede participar en varias competiciones, pero no es obligatorio que participe en varias.

La aplicación almacenará la información de cada jugador (nombre completo, nacionalidad, fecha nacimiento, nickname, rol, sueldo…), equipos (nombre, fecha de fundación, patrocinador o patrocinadores, jugadores, staff,...), jornadas del calendario de cada competición, los enfrentamientos/las partidas de la competición (fecha del enfrentamiento, número de jornada, equipos que intervienen en cada enfrentamiento, hora del enfrentamiento, resultado …). De las personas del staff se quiere registrar el nombre completo, el puesto y sueldo. Hay que tener en cuenta que mínimo habrá un entrenador y puede que haya, además, un asistente de entrenador.

E-Sport ha dejado las siguientes restricciones:

- La empresa exige que el salario mínimo de los jugadores sea mayor que el salario minino interprofesional.
- El salario total de los jugadores de un equipo no podrá ser superior a 200.000 euros anuales.
- Se diseñará un sistema para generar un calendario de los distintos enfrentamientos que se darán a lo largo de todas las jornadas de la competición. La competición sera de todos contra todos.

- La competición tendrá una jornada por semana. Los enfrentamientos de cada jornada se jugarán íntegramente en un día.
- Las jornadas que conforman una competición no se pueden generar, tanto si hay equipos sin jugadores, como si hay equipos con menos de dos jugadores.
- Una vez generado el calendario de una competición se cerrará la etapa de inscripciones y por tanto, no se podrá modificar la estructura de equipos y jugadores por equipo.
- Los equipos deberán estar formados por seis jugadores como máximo y mínimo dos. Para simplificar el proyecto el número de equipos que participan en una competición será par.

Se desea que haya dos perfiles para acceder a la aplicación: administrador y usuario.

- Los administradores son los encargados de realizar las siguientes tareas:
  - CRUD de las tablas que formen el sistema diseñado.
  - Cerrar la etapa de inscripción de una competición
  - Generar el calendario de una competición.
  - Introducir los resultados de los enfrentamientos.
  - Ver los resultados de todas las jornadas y la clasificación general.

- Los usuarios son los encargados de realizar la siguiente tarea:
  - Visualizar los resultados de la última jornada y clasificación.

En cuanto al interfaz de usuario, la empresa quiere que sea lo más intuitivo posible.

La empresa detectó que su sistema de información anterior tenía un número elevado de consultas a la base de datos, por lo que desea que informaciones como la clasificación y los resultados de las jornadas de la competición, incluyendo el resultado de la última jornada, se almacenen en ficheros XML de forma local. Estos ficheros deben almacenar la fecha en la que expira la información recuperada para que el sistema de información sepa cuando volver a realizar la consulta oportuna a la base de datos para sobrescribir el fichero. En caso de que el fichero se elimine, el SI deberá hacer la petición a la base de datos y crear el fichero.

OBJETIVOS / RESULTADOS DE APRENDIZAJE

Bases de datos

RA2	**Crea bases de datos, definiendo su estructura y las características de sus elementos según el modelo relacional**
RA3	**Consulta la información almacenada en una base de datos, empleando asistentes, herramientas gráficas y el lenguaje de manipulación de datos.**
RA4	**Modifica la información almacenada en la base de datos utilizando asistentes, herramientas gráficas y el lenguaje de manipulación de datos**
RA5	**Desarrolla procedimientos almacenados, evaluando y utilizando las sentencias del lenguaje incorporado en el sistema gestor de bases de datos.**
RA6	**Diseña modelos relacionales normalizados, interpretando diagramas entidad/relación**

Programación

RA1	**Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.**
RA2	**Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.**
RA3	**Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.**
RA4	**Desarrolla programas organizados en clases, analizando y aplicando los principios de la programación orientada a objetos.**
RA5	**Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.**
RA6	**Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos**
RA7	**Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.**
RA9	**Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.**

Lenguajes de marcas y sistemas de gestión de información

RA1	**Reconoce las características de lenguajes de marcas, analizando e interpretando fragmentos de código.**
RA2	**Establece mecanismos de validación para documentos XML, utilizando métodos para definir su sintaxis y estructura**
RA4	**Gestiona información en formato XML, analizando y utilizando tecnologías de almacenamiento y lenguajes de consulta.**

Entornos de desarrollo

RA1	**Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento.**

RA3	**Verifica el funcionamiento de programas, diseñando y realizando pruebas.**
RA4	**Optimiza código, empleando las herramientas disponibles en el entorno de desarrollo**
RA5	**Genera diagramas de clases, valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.**
RA6	**Genera diagramas de comportamiento, valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.**

TAREAS A REALIZAR

Esquema orientativo de pasos a seguir

1. Análisis del proyecto.
2. Puesta en marcha de los repositorios de GitHub.
3. Diseños
    1. Modelo entidad-relación y modelo relacional.
    2. Diagrama de clases ,diagrama de casos de uso y diagramas de secuencia.
    3. Diseño de los ficheros XML y sus plantillas de validación (DTD y XSD).
4. Desarrollo
    1. Un sólo script SQL que incluya el borrado y la creación de las tablas, vistas y otros objetos necesarios para gestión del modelo.
    2. Un sólo script SQL que incluya el borrado y la creación de los disparadores necesarios para gestión del modelo.
    3. Procedimientos PL/SQL almacenados, empaquetados y funciones.
    4. Un único script que contenga todos los procedimientos PL/SQL anónimos destinados a probar la funcionalidad de los procedimientos almacenados, empaquetados y funciones. Cada procedimiento anónimo debe tener un comentario explicando el procedimiento que está probando y dónde se utiliza en la aplicación Java.
    5. Script SQL para la carga de datos inicial de base de datos si fuera necesario.
    6. Conexión a la base de datos, carga de datos y CRUD
    7. Interfaces de usuario (Vistas)
    8. Controladores.
    9. Programación de la lógica para el uso y generación de los ficheros XML.
5. Pruebas de funcionamiento y optimización
6. Documentación
7. Creación ejecutable
8. Presentación.

Obtener información

Aunque cada miembro del equipo se dedique más a una tarea determinada, todos se responsabilizarán del trabajo de los demás y deberán conocer la evolución del desarrollo global del reto.

Explorar estrategias

Además del software o las pautas propuestas por el profesorado se pueden probar alternativas adicionales.

Actuar

El reto está dividido en fases.

El desarrollo se hará siguiendo las normas para el trabajo en equipo que se explican en el curso sobre Git cuyo enlace está en moodle.

Al final de la fases es obligatorio entregar los elementos especificados. Mirar las fechas de entrega en la temporización.

FASE UNO

Esta fase incluye los puntos uno y dos de los pasos a seguir. Se debe realizar un análisis del enunciado propuesto. En dicho análisis, se deben identificar los puntos más relevantes a tener en cuenta.

Los elementos a entregar en esta fase son los siguientes:

- Modelo entidad-relación. Este modelo debe ser coherente y respetar el enunciado. Todo aquello que no pueda ser representado en el modelo debe estar comentado. Por ejemplo: la restricción de que un equipo tenga como máximo 6 miembros no pueden ser modelada, por lo que estará anotada y más adelante se tratara de forma adecuada.
- Modelo relacional. Este modelo debe de ser coherente con el modelo entidad-relación.
- Diagrama de clases. Este diagrama debe ser coherente y respetar el enunciado. Es conveniente revisar los diferentes patrones de diseño que puedan resultar de ayuda e introducirlos en el diseño. Para elección de tipos es conveniente revisar los tipos de datos del modelo entidad relación y las conversiones entre el SGBD y el lenguaje de programación.

FASE DOS

Esta fase incluye una parte de revisión del diseño en caso de que se hayan encontrado errores graves, que puedan hacer que la implementación no se vaya a poder llevar a cabo. También incluye el desarrollo de los siguientes elementos:

- Diagrama de casos de uso
- Diagramas de secuencia. Se debe desarrollar como mínimo 2 casos de uso.
- Relacionado con la base de datos.
    1. Script SQL de borrado y creación de las tablas, vistas y sinónimos.
    2. Script SQL de borrado y creación de los disparadores. Como minino debemos tener disparadores para:
        - Controlar que no haya más de 6 jugadores en un equipo.
        - Controlar que para poder generar el calendario de una competición todos los equipos tienen que tener un mínimo de dos jugadores.
        - Controlar que una vez generado el calendario de la competición, no se pueden modificar, ni los equipos, ni los jugadores de cada equipo.
    3. Procedimientos almacenados. Estos procedimientos estarán diseñados para ser utilizados desde la aplicación Java. Hay que programar mínimo dos procedimientos, de los cuales uno tiene que ser la obtención de algún informe que **no** se obtenga en XML. Ejemplo de posible informe: relación de los equipos que conforman la competición en un momento determinado, con todos los datos que se tengan de la misma: nombre del juego, fechas, relación de equipos incluyendo para cada equipo el nombre del mismo, los miembros del staff y la cantidad de jugadores que hay en ese equipo.
    4. El paquete podrá contener procedimientos, funciones y tipos de datos. Como mínimo debe de existir un paquete que no tenga relación con XML y que no duplique la funcionalidad de los procedimientos almacenados.
    5. Un único script SQL que contenga todos los procedimientos anónimos destinados a probar la funcionalidad de los procedimientos almacenados, empaquetados y funciones. Cada procedimiento anónimo debe tener un comentario explicando el procedimiento que está probando y dónde se utiliza en la aplicación Java.
    6. Un script SQL con las pruebas funcionales de los disparadores.

7. Un script SQL SQL para la carga de datos inicial de base de datos si fuera necesario.

FASE TRES

En esta fase hay que realizar el desarrollo de los siguientes elementos:

- Codificación de las clases referentes al modelo.
- Estructura de los ficheros XML y de sus plantillas de validación (DTD y XSD).

FASE CUATRO

En esta fase hay que realizar el desarrollo de los siguientes elementos:

- Interfaces de la aplicación. Esto interfaces estarán en una carpeta llamada Views .
- Desarrollo de los controladores.
- Desarrollo de los procedimientos que generan los ficheros XML a partir de las consultas realizadas a la BD.
- Validación mediante DTD y XSD de los ficheros XML obtenidos.

FASE CINCO

Esta fase incluye las pruebas de funcionamiento, optimización, documentación del proyecto, creación del ejecutable y la preparación de la presentación. En esta fase se deberían desarrollar los siguientes elementos:

- Fichero que incluya las pruebas unitarias realizadas y capturas de las pruebas realizadas con sus respectivos resultados.
- Posibles optimizaciones (opcional).
- Documentación del proyecto, siguiendo las normas de formato de entrega de tareas:
    - Documentación obtenida mediante Javadoc.
    - De manera opcional se puede elaborar un manual del administrador que debe explicar de forma clara como poner todos los elementos del sistema en marcha desde cero.
- Ejecutable obtenido desde el IDE utilizado para el desarrollo.

- Presentación que se usara para la defensa del proyecto.

Logros

- Sistema de información en funcionamiento:
  - Base de datos
  - Proyecto y ejecutable Java.
- Repositorio de GitHub que contenga todos los productos desarrollados en las distintas fases. Si un producto se desarrolla con una herramienta, en el repositorio está el fuente obtenido con la herramienta y el producto resultante en un formato legible como puede ser pdf, jpeg, … . El primer nivel del repositorio consistirá en una carpeta correspondiente a cada módulo.
- Presentación/exposición que muestre el funcionamiento del proyecto durante la que se atenderán preguntas sobre la estructura y el proceso de desarrollo.

CRITERIOS DE EVALUACIÓN

Criterios para la elaboración de las calificaciones parciales

- La nota del reto se obtendrá de la siguiente manera:
  - 20% competencias transversales:
    - 10% auto-evaluación.
    - 60% valoración del equipo.
    - 30% evaluación del profesorado.
  - 50% valoración del reto.
  - 30% examen escrito y/o actividades propuestas.

  Rúbricas técnicas y transversales disponibles en SET
  https://tknika.setskills.org/

- Para que los módulos puedan ser evaluados, no se podrá tener grado cero en NINGUNA de las rubricas asociadas a dicho módulo.
- Para dar un reto por superado, habrá que obtener un mínimo de cuatro puntos en cada una de las partes.
- Si un reto no ha sido superado por no haber alcanzado el mínimo correspondiente en la parte de evaluación, se realizará una prueba práctica individual.
- Si un reto no ha sido superado por no haber alcanzado el mínimo en el examen, se realizará un nuevo examen, siendo la nota máxima final posible, 5.
- Si la diferencia de nota entre las diferentes partes (prueba y reto) es superior a dos puntos y la nota más baja es inferior a 6.5, la calificación final será la más baja de las dos partes mencionadas.
- La nota de la evaluación se obtendrá de la media ponderada de cada uno los retos concluidos durante la evaluación; siempre y cuando estén aprobados. Si algún reto no ha sido superado, no se podrá aprobar la evaluación.
- Si la nota del reto es superior al cinco pero alguna de las partes está suspendida, se pondrá un cinco.
- Si un modulo no se ha visto implicado en ninguno de los retos de la evaluación, no se evaluará.
- En el caso de que para un módulo que intervenga en el reto no se haya realizado prueba de evaluación, se utilizará para el cálculo el resultado de las rúbricas del grupo de competencias correspondiente al módulo.
- Si en una evaluación no se finalizara ningún reto, no se evaluarán los módulos implicados. En el boletín de notas, se informará del motivo.
- Para dar por superada la recuperación la nota a alcanzar será de 5 y en ese caso, la nota máxima de la evaluación personal será 5.

La nota del reto se multiplicará por el número de miembros del grupo que hayan participado activamente en el mismo y entre ellos se repartirán los puntos.

Procedimiento para la calificación en la primera evaluación final

- El alumno/a que tenga pendiente algún reto, deberá recuperar en la primera evaluación final la parte o partes pendientes a lo largo del curso. La no superación de dicha prueba conllevará acudir a la segunda evaluación final con todos los contenidos del módulo.

- La nota de la primera evaluación final será obtenida como media ponderada de las notas de los retos, siendo condición necesaria, tener aprobado todos.

- En la prueba de la primera evaluación final será obligatoria la entrega de las actividades y trabajos realizados durante el curso. La falta de dicho requisito supondrá la no superación del modulo. Si las actividades y/o trabajos no han sido superados, se realizará una prueba específica para él con dichos contenidos.

Estos criterios podrán ser modificados en hipotéticos escenarios de semi-presencialidad o no presencialidad condicionados por la evolución de la situación sanitaria, y que serán debidamente informados al alumnado con la mayor antelación posible.

RECURSOS

Documentación y materiales

Todo el material necesario para el desarrollo del reto está disponible en el curso de moodle del reto: https://ikas.egibide.org/moodle/course/view.php?id=1563

Profesores de referencia

Módulo	Profesor	Email
Bases de datos	Blanca Isasi-Isasmendi	bisasi@egibide.org
Programación	Nieves Ruiz	mnruiz@egibide.org
Entornos de desarrollo	Eider Arbaiza	earbaiza@egibide.org
Lenguajes de marcas	Eva Ugarte	emugarte@egibide.org

TEMPORIZACIÓN

El reto comienza el día 22 de abril y finalizará el viernes 31 de mayo.

Calendario semanal previsto

	L	M	X	J	V
Semana					
1 22/4	Inicio	Entrega fase uno			
2 29/4					
3 6/5			Entrega fase dos		Entrega fase tres
4 13/5					
5 20/5			Entrega fase cuatro		Entrega fase cinco
6 27/5	Presentación	Examen ED	Examen PROG	Examen BBDD	Examen LM