



APRENDIZAJE COLABORATIVO BASADO EN RETOS

2020/2021

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

3 semanas

ORGANIZACIÓN

Grupos de 3 - 4 personas

RETO UNO

“CONSULTORÍA E-SPORT”

EL RETO

La empresa de E-Sport os ha contratado para que realicéis mejoras en su sistema de información. Desean una aplicación que almacene la información de cada jugador (nombre, nickname, rol, sueldo...), equipos (nombre, jugadores, entrenador, si lo tiene asistente de entrenador, dueño de cada equipo, ...), el calendario de las jornadas de la liga (jornada, fecha, equipos que intervienen, hora de la partida,...) y el resultado de cada uno de los partidos.

E-Sport ha dejado las siguientes restricciones:

- La empresa exige que el salario mínimo de los jugadores sea mayor que el salario mínimo interprofesional.
- Los equipos deberán estar formados por seis miembros como máximo.
- El salario total de los equipos no podrá ser superior a 200.000 euros anuales.
- La liga no puede comenzar si hay equipos sin jugadores.
- Se diseñará un sistema para emparejar a los equipos en un torneo todos contra todos.
- La liga contará una jornada por semana. Las partidas se jugarán íntegramente en un día.
- Para simplificar el proyecto el número de equipos será par.
- Una vez generado el calendario de la liga no se podrá modificar la estructura de equipos y jugadores por equipo.

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 jugadores, equipos, dueños y usuarios.
 - Lanzar los procedimientos diseñados para generar el calendario y realizar los emparejamientos.
 - Introducir los resultados de cada jornada.
 - 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 temporada, 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. Diseño
 1. Puesta en marcha de los repositorios de GitHub.
 2. Modelo entidad-relación y modelo relacional.
 3. Diagrama de clases y diagramas de secuencia.
 4. Diseño de los ficheros XML y sus plantillas de validación (DTD y XSD).
3. Desarrollo
 1. Script para creación de base de datos incluyendo disparadores para gestión del modelo.
 2. Script/s para los procedimientos almacenados, empaquetados y funciones.
 3. Conexión a la base de datos, carga de datos y CRUD
 4. Interfaces de usuario (Vistas)
 5. Núcleo (controlador)
 6. Programación de la lógica para el uso y generación de los ficheros XML.
4. Pruebas de funcionamiento y optimización
5. Documentación
6. Creación ejecutable
7. 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 fase uno es obligatorio entregar los elementos especificados. Mirar las fechas de entrega en la temporización. El resto de las fases son orientativas y se incluyen para facilitar la planificación.

FASE UNO

Esta fase incluye los puntos uno y dos de los pasos a seguir. Se debe realizar un análisis del enunciado para poder llevar a cabo las siguientes tareas. En dicho análisis, se debe identificar los puntos más relevantes a tener cuenta.

Los elementos a entregar 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, y el desarrollo de los siguientes elementos:

- Diagramas de secuencia. Se debe desarrollar como mínimo 2 casos de uso.
- Estructura de los ficheros XML y de sus plantillas de validación (DTD y XSD).
- Script para la creación de la base de datos. Este fichero debe respetar el enunciado, por lo que debe incluir todas las restricciones necesarias para mantener la integridad de la base de datos y las restricciones impuestas por E-Sport. Como mínimo incluirá los disparadores:
 1. Controlar que no haya más de 6 jugadores en un equipo.
 2. Controlar que para poder generar el calendario todos los equipos tienen que tener mínimo dos jugadores.
 3. Controlar que una vez generado el calendario, no se pueden modificar, ni los equipos, ni los jugadores de cada equipo.
- Script para la carga de datos inicial de base de datos si fuera necesario.

- Procedimientos almacenados. Mínimo dos de ellos orientados a ser utilizados desde java para obtener informes(por ejemplo: partidos ganados por un equipo;datos de los participantes en la liga relación de los equipos que conforman la liga en un momento determinado, con todos los datos del equipo, su dueño y la cantidad de jugadores que hay en cada uno)
- Paquetes: que puedan contener procedimientos, funciones y tipos de datos. Como mínimo un paquete para incluir elementos que mejoren la eficiencia del sistema.

FASE TRES

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

- Codificación de las clases referentes al modelo.
- Interfaces de la aplicación. Esto interfaces estarán en una carpeta llamada Views .
- Desarrollo del o 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 CUATRO

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 con los productos desarrollados, incluidos los documentos y fichas elaboradas en las distintas fases.
- 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 módulo. 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=1390>

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 Maider Diaz	emugarte@egibide.org maiderdiaz@egibide.org

TEMPORIZACIÓN

El reto comienza el día 4/05/2021 y termina el 2/06/2021.

Calendario semanal previsto

		M	X	J	V
Semana		4/5	5/5	6/5	7/5
1		Inicio 8:15 Entrega de la fase 1 a las 15:00			

		M	X	J	V
Semana		11/05	12/05	13/5	14/5
2		Entrega de la fase 2 a las 15:00			

		M	X	J	V
Semana		18/05	19/05	20/05	21/05
3					

		M	X	J	V
Semana		25/05	26/05	27/05	28/05
4		Entrega a las 14:15 Evaluación de transversales	Examen LM	Examen BBDD	Examen programación

	L	M	X
Semana	31/05	1/6	2/6
5		Examen ED	8:15 Presentaciones (15 – 20 minutos por equipo) Reparto de puntos