Egibide

APRENDIZAJE COLABORATIVO BASADO EN RETOS

2024/2025

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)	
DURACIÓN	**ORGANIZACIÓN**
Todo el curso	Grupos de 5 - 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 gestione una competición e-sport asociada a un juego concreto.

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 de la competición.
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.

La aplicación almacenará la información de cada jugador (nombre, apellido, nacionalidad, fecha nacimiento, nickname, rol, sueldo), equipos (nombre, fecha de fundación, jugadores), jornadas del calendario de la competición, los enfrentamientos/las partidas de la competición (fecha del enfrentamiento, número de jornada, equipos que intervienen en cada enfrentamiento, hora del enfrentamiento, resultado).

E-Sport ha dejado las siguientes restricciones:

- Para simplificar el proyecto el número de equipos que participan en la competición será par.
- Los equipos deberán estar formados por seis jugadores como máximo.
- La empresa exige que el salario mínimo de los jugadores sea mayor que el salario minino inter-profesional.
- 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 la competición no se pueden generar si hay equipos con menos de 2 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.

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 todos los informes.
- Los usuarios son los encargados de realizar la siguiente tarea:
  - Visualizar el informe de la relación de equipos que conforman la competición.
  - Visualizar el informe correspondiente a los resultados de la última jornada.

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

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.**

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.
4. Desarrollo
    1. Conexión a la base de datos, carga de datos y CRUD
    2. Interfaces de usuario (Vistas)
    3. Controladores.
    4. Programación de la lógica para el uso.
    5. Un sólo script SQL que incluya el borrado y la creación de las tablas, vistas y otros objetos necesarios para gestión del modelo.
    6. Script SQL para la carga de datos inicial de base de datos si fuera necesario.
    7. Un único script SQL que incluya el borrado y la creación de los disparadores (triggers) necesarios para gestión del modelo.
    8. Un sólo script con los procedimientos PL/SQL almacenados y funciones para la obtención de informes en Java.
    9. Un único script que contenga todos los procedimientos PL/SQL anónimos destinados a probar la funcionalidad de los procedimientos almacenados y funciones. Cada procedimiento anónimo debe tener un comentario explicando el procedimiento que está probando y dónde se utiliza en la aplicación Java.
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***. En las dos primeras fases, trabajareis con estructuras de datos en memoria para manejar información temporalmente mientras el programa esta en ejecución. Sin embargo, si el sistema se apaga o si es necesario acceder a esos datos en otro momento o desde otra aplicación, los datos en memoria desaparecen. Para garantizar la permanencia de los datos y la capacidad de recuperarlos en el futuro, necesitamos almacenarlos en un lugar que no dependa de la ejecución continua del programa. Aquí es donde entran en juego las bases de datos. En la fase dos, para hacer los datos persistentes durante la fase 3, diseñaremos un Modelo Entidad-Relación (MER) que nos proporcione una representación gráfica y lógica de cómo se organizan y relacionan los datos. Una vez que tenemos el MER, lo transformaremos en un modelo relacional.

A partir de la fase 2, 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

- Desarrollar un programa para guardar en memoria los datos de un equipo y de un jugador validando todos los campos de entrada y llevando a cabo la entrada y salida de datos desde consola.
- Generar un documento donde se reflejen los requisitos funcionales y no funcionales a tener en cuenta en esta aplicación.

Los elementos a **entregar** en esta fase son los siguientes:

- Código de cada uno de los integrantes del equipo.
- Código del equipo.
- Documento de Requisitos.

FASE DOS

- Crear la estructura necesaria en Git para trabajar en el desarrollo del proyecto.
- Crear el diagrama de clases.
- Crear el diagrama de casos de uso.
- Crear un diagrama de secuencia que recoja uno de los casos de uso planteados en el diagrama anterior
- Codificación del diagrama de clases.
- Codificación del alta, baja y modificación de equipos y jugadores de la competición. Los datos serán almacenados como objetos en colecciones de datos.
- Generación de jornadas y de enfrentamientos en java.
- Diagrama MER. 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 y reflejar todas las restricciones.

Los elementos a **entregar** en esta fase son los siguientes:

- Diagrama de clases
- Diagrama de casos de uso
- Diagrama de secuencia
- Código del proyecto (codificación del diagrama de clases; alta, baja y modificación de equipos y jugadores; generación de jornadas y de enfrentamientos).
- Fichero pdf o jpeg donde aparezca el diagrama MER.
- Fichero pdf o jpeg donde aparezca el diseño relacional.

FASE TRES

- Utilizando el JUnit generar pruebas para comprobar el código creado
- Desarrollo del entorno gráfico de la aplicación siguiendo la estructura MVC.
- Codificación del alta, baja y modificación de equipos y jugadores almacenando los datos en una base de datos siguiendo la estructura MVC.
- Codificación del resto del proyecto siguiendo la estructura MVC (guardar los resultados de los enfrentamientos, ejecución de los distintos procedimientos, visualización de informes, etc...).
- Diseño físico de la base de datos (tablas, vistas, otros objetos y disparadores (=triggers)).
- Procedimiento almacenado en la base de datos, que permita después en Java, ver el informe de la relación de los equipos que conforman la competición incluyendo para cada equipo el nombre del mismo, la fecha de creación, la cantidad de jugadores que hay en ese equipo, el salario máximo, el salario mínimo y la media de los salarios de

los jugadores de ese equipo. Las excepciones serán visualizadas en el programa Java.

- Procedimiento almacenado en la base de datos, que permita después en Java, ver el informe con la relación de los jugadores de un equipo concreto. De cada jugador se verá el nombre, apellido, rol y salario. El nombre del equipo le llegará como parámetro. Las excepciones serán visualizadas en el programa Java.

  Los elementos a **entregar** en esta fase son los siguientes:

  - Pruebas documentadas
  - Documentación del código
  - Código del proyecto
  - Un sólo script SQL que incluya el borrado y la creación de las tablas, vistas y otros objetos necesarios para gestión del modelo.
  - Si se considera necesario, un script SQL para la carga de datos inicial de base de datos.
  - Un sólo script SQL que incluya el borrado y la creación de los disparadores (triggers) necesarios para gestión del modelo. Como minino debemos tener disparadores (triggers) para:
    - Asegurar que el salario de los jugadores cumple el enunciado.
    - Controlar que no haya más de 6 jugadores en un equipo.
    - Antes de generar el calendario de una competición, garantizar que todos los equipos tienen 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.
  - Un script con las sentencias para probar los disparadores programados.
  - Un solo script que contenga todos los procedimientos PL/SQL almacenados y funciones.
  - Un único script que contenga todos los procedimientos PL/SQL anónimos destinados a probar la funcionalidad de los procedimientos almacenados y funciones. Cada procedimiento anónimo debe tener un comentario explicando el procedimiento que está probando y dónde se utiliza en la aplicación Java.

CRITERIOS DE EVALUACIÓN

Procedimiento para la calificación en la primera evaluación final

- La nota del reto se obtendrá de la siguiente manera:
  - 20% competencias transversales:
    - 10% auto-evaluación.
    - 60% valoración del equipo.
    - 30% evaluación del profesorado.
  - 80% valoración del reto.

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

- La nota del reto supondrá un 20% de la nota de los módulos implicados en la 1OF siempre y cuando se tengan aprobadas las tres evaluaciones del mismo.

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=1624

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

TEMPORIZACIÓN

El reto comienza el día 9 de septiembre y finalizará el día 9 de mayo

Calendario previsto

Mensual					
Septiembre		9 – 13 Actividades	16 – 20 Actividades	23 – 27 Actividades	
Octubre	30 – 4 Actividades	7 – 11 Actividades	14 – 18 Actividades	21 – 25 Actividades	28 – 1 Actividades
Noviembre	4 – 8 Fase uno	11 – 15 Actividades	18 – 22 Actividades	25 – 29 Actividades	
Diciembre	2 – 5 Actividades	9-13 Actividades	16 – 20 Actividades		
Enero		7 – 10 Actividades	13 – 17 Actividades	20 – 24 Actividades	27 – 31 Actividades
Febrero	3 – 7 Actividades Fase dos (desde el 7)	10 – 14 Actividades Fase dos	17 – 21 Actividades	24 – 28 Actividades	
Marzo	5 – 7 Actividades	10 – 14 Actividades	17 – 21 Actividades	24 – 28 Actividades	
Abril	31 – 4 Actividades	7 – 11 Actividades	14 – 16 Actividades		
Mayo	5 – 9				

El curso 2024-2025 es el primer curso en que por la aplicación de la Ley Orgánica 2/2022 de ordenación e integración de la Formación Profesional, todos los ciclos van a tener formato Dual. Esto supone una estancia formativa en la empresa del alumnado de FP. Las fechas de

realización de las mismas, condiciona la fase tres por lo que no se han puesto en la temporización. Se comunicarán lo antes posible.