Egibide

APRENDIZAJE COLABORATIVO BASADO EN RETOS

2022/2023

RETO: CONSULTORÍA E-SPORT

DAM

DESARROLLO DE APLICACIONES MULTIPLATAFORMA

1

MÓDULOS IMPLICADOS	
0484. Bases de datos 0485. Programación 0487. Entornos de desarrollo	
DURACIÓN	ORGANIZACIÓN
4-5 semanas	Grupos de 3-4 personas

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

ÍNDICE

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

EL RETO

La empresa E-Sport os ha contratado para que realicéis mejoras en su sistema de información.

Desean que se almacene la información de cada jugador (nombre, nickname, sueldo…), dueño de cada equipo (nombre, equipo…), equipo (nombre, jugadores…), el calendario de las jornadas de la liga (equipos que intervienen, fecha…) 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 diseñara 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.
- Las partidas se jugarán a un solo enfrentamiento. Se usará el sistema de todos contra todos (https://es.wikipedia.org/wiki/Sistema_de_todos_contra_todos). Para simplificar el proyecto el número de equipos será par.

Se desea que haya tres perfiles para acceder a la aplicación: administrador, dueño y usuario.

- Los administradores son los encargados de realizar las siguientes tareas:
  - CRUD de jugadores, equipos, dueños y usuarios.
  - Generar el calendario.
  - Introducir los resultados de cada jornada.
- Los dueños son los encargados de realizar las siguientes tareas:
  - Confeccionar su equipo para la temporada, cumpliendo las restricciones marcadas en el enunciado.
  - Visualizar los resultados de la última jornada y la clasificación.
- Los usuarios pueden ver los resultados de todas las jornadas y la clasificación general.

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

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

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

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**

a) Se ha analizado el formato de almacenamiento de la información.
b) Se han creado las tablas y las relaciones entre ellas.
c) Se han seleccionado los tipos de datos adecuados.
d) Se han definido los campos clave en las tablas.
e) Se han implantado las restricciones reflejadas en el diseño lógico.
h) Se han utilizado asistentes, herramientas gráficas y los lenguajes de definición y control de datos.

RA3 **Consulta la información almacenada en una base de datos, empleando asistentes, herramientas gráficas y el lenguaje de manipulación de datos.**

a) Se han identificado las herramientas y sentencias para realizar consultas.
b) Se han realizado consultas simples sobre una tabla.
c) Se han realizado consultas sobre el contenido de varias tablas mediante composiciones internas.
d) Se han realizado consultas sobre el contenido de varias tablas mediante composiciones externas.
e) Se han realizado consultas resumen.
f) Se han realizado consultas con subconsultas.

RA4 **Modifica la información almacenada en la base de datos utilizando asistentes, herramientas gráficas y el lenguaje de manipulación de datos**

a) Se han identificado las herramientas y sentencias para modificar el contenido de la base de datos.
b) Se han insertado, borrado y actualizado datos en las tablas.
d) Se han diseñado guiones de sentencias para llevar a cabo tareas complejas.
e) Se ha reconocido el funcionamiento de las transacciones.
f) Se han anulado, parcial o totalmente, los cambios producidos por una transacción.
g) Se han identificado los efectos de las distintas políticas de bloqueo de registros.
h) Se han adoptado medidas para mantener la integridad y consistencia de la información.

RA5 **Desarrolla procedimientos almacenados, evaluando y utilizando las sentencias del lenguaje incorporado en el sistema gestor de bases de datos.**

a) Se han identificado las diversas formas de automatizar tareas.
b) Se han reconocido los métodos de ejecución de guiones.

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

c) Se han identificado las herramientas disponibles para editar guiones.
d) Se han definido y utilizado guiones para automatizar tareas.
e) Se ha hecho uso de las funciones proporcionadas por el sistema gestor.
f) Se han definido funciones de usuario.
g) Se han utilizado estructuras de control de flujo.
h) Se han definido disparadores.
i) Se han utilizado cursores.

RA6 Diseña modelos relacionales normalizados, interpretando diagramas entidad/relación

a) Se han utilizado herramientas gráficas para representar el diseño lógico.
b) Se han identificado las tablas del diseño lógico.
c) Se han identificado los campos que forman parte de las tablas del diseño lógico.
d) Se han analizado las relaciones entre las tablas del diseño lógico.
e) Se han identificado los campos clave.
f) Se han aplicado reglas de integridad.
g) Se han aplicado reglas de normalización.
h) Se han analizado y documentado las restricciones que no pueden plasmarse en el diseño lógico.

Programación

RA1 Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.

a) Se han identificado los bloques que componen la estructura de un programa informático.
b) Se han creado proyectos de desarrollo de aplicaciones.
c) Se han utilizado entornos integrados de desarrollo.
d) Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.
e) Se ha modificado el código de un programa, para crear y utilizar variables.
f) Se han creado y utilizado constantes y literales.
g) Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.
h) Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas.
i) Se han introducido comentarios en el código.

RA2 Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.

a) Se han identificado los fundamentos de la programación orientada a objetos.
b) Se han escrito programas simples.
c) Se han instanciado objetos a partir de clases predefinidas.
d) Se han utilizado métodos y propiedades de los objetos.
e) Se han escrito llamadas a métodos estáticos.
f) Se han utilizado parámetros en la llamada a métodos.
g) Se han incorporado y utilizado librerías de objetos.
h) Se han utilizado constructores.
i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

de programas simples.

RA3 **Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.**

a) Se ha escrito y probado código que haga uso de estructuras de selección.
b) Se han utilizado estructuras de repetición.
c) Se han reconocido las posibilidades de las sentencias de salto.
d) Se ha escrito código utilizando control de excepciones.
e) Se han creado programas ejecutables utilizando diferentes estructuras de control.
f) Se han probado y depurado los programas.
g) Se ha comentado y documentado el código.

RA4 **Desarrolla programas organizados en clases, analizando y aplicando los principios de la programación orientada a objetos.**

a) Se han reconocido la sintaxis, estructura y componentes típicos de una clase.
b) Se han definido clases.
c) Se han definido propiedades y métodos.
d) Se han creado constructores.
e) Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.
f) Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.
g) Se han definido y utilizado clases heredadas.
h) Se han creado y utilizado métodos estáticos.
i) Se han definido y utilizado interfaces.
j) Se han creado y utilizado conjuntos y librerías de clases.

RA5 **Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.**

a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
b) Se han aplicado formatos en la visualización de la información.
c) Se han reconocido las posibilidades de entrada/salida del lenguaje y las librerías asociadas.
d) Se han utilizado ficheros para almacenar y recuperar información.
e) Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.
f) Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples.
g) Se han programado controladores de eventos.
h) Se han escrito programas que utilicen interfaces gráficos para la entrada y salida de información.

RA6 **Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos**

a) Se han escrito programas que utilicen arrays.
b) Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados.
c) Se han utilizado listas para almacenar y procesar información.
d) Se han utilizado iteradores para recorrer los elementos de las listas.

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

e) Se han reconocido las características y ventajas de cada una de las colecciones de datos disponibles.
f) Se han creado clases y métodos genéricos.
g) Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.
h) Se han identificado las clases relacionadas con el tratamiento de documentos XML.
i) Se han realizado programas que realicen manipulaciones sobre documentos XML.

RA7 **Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.**

a) Se han identificado los conceptos de herencia, superclase y subclase.
b) Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.
c) Se ha reconocido la incidencia de los constructores en la herencia.
d) Se han creado clases heredadas que sobrescriban la implementación de métodos de la superclase.
e) Se han diseñado y aplicado jerarquías de clases.
f) Se han probado y depurado las jerarquías de clases.
g) Se han realizado programas que implementen y utilicen jerarquías de clases.
h) Se ha comentado y documentado el código.

RA9 **Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y consistencia de los datos.**

a) Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos relacionales.
b) Se han programado conexiones con bases de datos.
c) Se ha escrito código para almacenar información en bases de datos.
d) Se han creado programas para recuperar y mostrar información almacenada en bases de datos.
e) Se han efectuado borrados y modificaciones sobre la información almacenada.
f) Se han creado aplicaciones que ejecuten consultas sobre bases de datos.
g) Se han creado aplicaciones para posibilitar la gestión de información presente en bases de datos relacionales.

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.**

a) Se ha reconocido la relación de los programas con los componentes del sistema informático: memoria, procesador, periféricos, entre otros.
b) Se han identificado las fases de desarrollo de una aplicación informática.
c) Se han diferenciado los conceptos de código fuente, objeto y ejecutable.
d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales.
e) Se han clasificado los lenguajes de programación.
f) Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en programación

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

RA3 **Verifica el funcionamiento de programas, diseñando y realizando pruebas.**

a) Se han identificado los diferentes tipos de pruebas.
b) Se han definido casos de prueba.
c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.
d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento.
e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.
f) Se han efectuado pruebas unitarias de clases y funciones.
g) Se han implementado pruebas automáticas.
h) Se han documentado las incidencias detectadas.

RA4 **Optimiza código, empleando las herramientas disponibles en el entorno de desarrollo**

a) Se han identificado los patrones de refactorización más usuales.
b) Se han elaborado las pruebas asociadas a la refactorización.
c) Se ha revisado el código fuente usando un analizador de código.
d) Se han identificado las posibilidades de configuración de un analizador de código.
e) Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.
f) Se ha realizado el control de versiones integrado en el entorno de desarrollo.
g) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.

RA5 **Genera diagramas de clases, valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.**

a) Se han identificado los conceptos básicos de la programación orientada a objetos.
b) Se ha instalado el módulo del entorno integrado de desarrollo que permite la utilización de diagramas de clases.
c) Se han identificado las herramientas para la elaboración de diagramas de clases.
d) Se ha interpretado el significado de diagramas de clases.
e) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.
f) Se ha generado código a partir de un diagrama de clases.
g) Se ha generado un diagrama de clases mediante ingeniería inversa.

RA6 **Genera diagramas de comportamiento, valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.**

a) Se han identificado los distintos tipos de diagramas de comportamiento.
b) Se ha reconocido el significado de los diagramas de casos de uso.
c) Se han interpretado diagramas de interacción.
d) Se han elaborado diagramas de interacción sencillos.
e) Se ha interpretado el significado de diagramas de actividades.

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

f) Se han elaborado diagramas de actividades sencillos.
g) Se han interpretado diagramas de estados.
h) Se han planteado diagramas de estados sencillos.

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

TAREAS A REALIZAR

Esquema orientativo de pasos a seguir

1. Análisis del proyecto.
2. Diseño:
   a. Puesta en marcha del repositorio Git.
   b. Planificación del proyecto.
   c. Modelo entidad-relación y modelo relacional.
   d. Diagrama de clases, casos de uso, diagramas de secuencia.
3. Desarrollo:
   a. Script para creación de base de datos incluye disparadores para gestión del modelo.
   b. Script/s para los procedimientos almacenados, empaquetados y funciones.
   c. Conexión a la base de datos, carga de datos y CRUD.
   d. Interfaz de usuario (vistas).
   e. Núcleo (controlador)
4. Pruebas de funcionamiento y optimización.
5. Documentación.
6. Creación del 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 en el repositorio Git proporcionado, siendo éste el único lugar donde deberá alojarse toda la documentación, planificación, entregables y código fuente producido por el equipo.

Al final de la fase uno es obligatorio entregar los elementos especificados. El resto de las fases son orientativas y se incluyen para facilitar la planificación.

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

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:

- Plan de trabajo. El equipo creará una hoja de cálculo en el repositorio del proyecto donde se irá detallando la planificación prevista para cada día.
- 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.
- Casos de uso. Se debe identificar los diferentes casos de uso y sus actores correspondientes.

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.
- Diferentes scripts para la creación de todos los objetos de la base de datos.
    1. Script de borrado y creación de las tablas, vistas y sinónimos.
    2. Script 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 todos los equipos tienen que tener mínimo dos jugadores.
    3. Script de creación de los procedimientos almacenados. Mínimo dos procedimientos. Estos procedimientos estarán diseñados para ser utilizados

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

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)

4. Scripts de creación de paquetes. El paquete podrá contener procedimientos, funciones y tipos de datos. Como mínimo debe de existir un paquete.
5. Script con los procedimientos anónimos destinados a probar los procedimientos almacenados y los trigger.

- Script para la carga de datos inicial de base de datos si fuera necesario.
- Codificación de las clases referentes al modelo.
- Interfaces de la aplicación. Estos interfaces estarán en una carpeta llamada views.
- Desarrollo del núcleo en el que se tiene que incluir un sistema para emparejar a los equipos en un torneo todos contra todos.
- Desarrollo del o de los controladores.

FASE TRES

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.
  - Manual del administrador que debe explicar de forma clara como poner todos los elementos del sistema en marcha desde cero.
  - Manual de usuario, que opcionalmente puede ser un screencast.
- 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 Java.

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

- Repositorio Git 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.

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

RECURSOS

Profesores de referencia

Módulo	Profesor	Email
Bases de datos Entornos de desarrollo	Eider Arbaiza	earbaiza@egibide.org
Programación	Ion Jaureguialzo	ijaureguialzo@egibide.org

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.

TEMPORIZACIÓN

El reto comienza el día 02/05/2023 y termina el 31/05/2023.

Calendario semanal previsto

	L	M	X	J	V
Semana	**01/05**	**02/05**	**03/05**	**04/05**	**05/05**
1		Enunciado	Reto	Entrega Fase 1	Reto

	L	M	X	J	V
Semana	**08/05**	**09/05**	**10/05**	**11/05**	**12/05**
2	Reto	Reto	Reto	Reto	Reto

	L	M	X	J	V
Semana	**15/05**	**16/05**	**17/05**	**18/05**	**19/05**
3	Reto	Reto	Reto	Examen ED	Reto

	L	M	X	J	V
Semana	**22/05**	**23/05**	**24/05**	**25/05**	**26/05**
4	Examen PROG	Reto	Reto	Reto	Examen BD

	L	M	X	J	V
Semana	**29/05**	**30/05**	**31/05**	**01/06**	**02/06**
5	Reto	Entrega	Defensa		

La fecha del examen de FOL está por determinar.

Excepto si se especifica lo contrario, esta obra está bajo licencia CC BY 4.0.
© 2023 Departamento de Informática, Egibide. Algunos derechos reservados.