



APRENDIZAJE COLABORATIVO BASADO EN RETOS

2017/2018

RETO: “CONSULTORÍA E-SPORT”

DAW

DESARROLLO DE APLICACIONES WEB

1

MÓDULOS IMPLICADOS

0483. Sistema informáticos
0484. Bases de datos
0485. Programación
0487. Entornos de desarrollo
0373. Lenguajes de marcas y sistemas de gestión de información
E-200. Inglés Técnico

DURACIÓN

7 semanas

ORGANIZACIÓN

Grupos de 3 - 4 personas

RETO

“CONSULTORÍA E-SPORT”

ÍNDICE

El reto.....	3
Aplicación.....	3
Gestor de contenido.....	4
Registro del estado de red.....	4
Objetivos / Resultados de Aprendizaje.....	5
Bases de datos.....	5
Programación.....	6
Lenguajes de marcas y sistemas de gestión de información.....	8
Entornos de desarrollo.....	9
Sistemas informáticos.....	11
Inglés técnico.....	11
Tareas a realizar.....	13
Esquema orientativo de pasos a seguir.....	13
Obtener información.....	13
Explorar estrategias.....	13
Actuar.....	13
FASE UNO.....	14
FASE DOS.....	15
FASE TRES.....	16
FASE CUATRO.....	16
Logros.....	17
Criterios de evaluación.....	18
Criterios para la elaboración de las calificaciones parciales.....	18
Procedimiento para la calificación en la primera evaluación final.....	18
Competencias técnicas.....	20
Competencias transversales.....	20
Recursos.....	21
Documentación y materiales.....	21
Profesores de referencia.....	21
Temporización.....	22
Calendario semanal previsto.....	22
Formato para entrega de tareas.....	24
Tipo de letra.....	24
Párrafo.....	24
Encabezado y pie de página.....	24
Capturas de pantalla e imágenes.....	24
Citas y bibliografía.....	24
Portada.....	24
Tablas.....	24

EL RETO

La empresa de E-Sport os ha contratado para que realicéis mejoras en su sistema de información en varios sentidos: desean un gestor de contenidos, una aplicación y un registro del estado de la red local.

Aplicación

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 constará de $(n-1)*2$ jornadas (8 equipos → 14 jornadas).

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 en base a los jugadores disponibles.
 - Fijar los miembros de los equipos para el resto de la temporada, cumpliendo las restricciones marcadas en el enunciado.
 - Visualizar los resultados de la última jornada y clasificación.
- Los usuarios pueden ver los resultados de todas las jornadas y la clasificación general.

En cuanto al interfaz de usuario, la empresa quiere que sea lo más intuitivo posible. Le gustaría que desde el perfil de usuario se pueda ver los resultados de la última jornada, la clasificación actual y si es posible un gráfico que muestre la evolución de los equipos en la competición.

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.

Gestor de contenido

E-Sport ha oído hablar de “WordPress” como gestor de contenidos y está valorando usarlo para crear un blog. Desean ver un prototipo. Os proponen crear un blog con Wordpress en el que E-Sport pueda seguir el desarrollo de vuestro trabajo.

Registro del estado de red

Debido a diferentes problemas que han tenido con la red de área local desean tener un fichero de log que en el que se registre el estado de la red. En principio, sería suficiente con que en el fichero apareciera si todos los nodos del sistema de información son alcanzables entre ellos.

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.

- 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

de programas simples.

RA3 Escribir y depurar 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 Desarrollar 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 Realizar 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 Escribir 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.

- 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.

Lenguajes de marcas y sistemas de gestión de información

RA1 Reconoce las características de lenguajes de marcas, analizando e interpretando fragmentos de código.

- a) Se han identificado las características generales de los lenguajes de marcas.
- b) Se han reconocido las ventajas que proporcionan en el tratamiento de la información.
- c) Se han clasificado los lenguajes de marcas e identificado los más relevantes.
- d) Se han diferenciado sus ámbitos de aplicación.
- e) Se han reconocido la necesidad y los ámbitos específicos de aplicación de un lenguaje de marcas de propósito general.
- f) Se han analizado las características propias del lenguaje XML.
- g) Se ha identificado la estructura de un documento XML y sus reglas sintácticas.

- h) Se ha contrastado la necesidad de crear documentos XML bien formados y la influencia en su procesamiento.
- i) Se han identificado las ventajas que aportan los espacios de nombres.

RA2 Establece mecanismos de validación para documentos XML, utilizando métodos para definir su sintaxis y estructura

- a) Se ha establecido la necesidad de describir la información transmitida en los documentos XML y sus reglas.
- b) Se han identificado las tecnologías relacionadas con la definición de documentos XML.
- c) Se ha analizado la estructura y sintaxis específica utilizada en la descripción.
- d) Se han creado descripciones de documentos XML.
- e) Se han utilizado descripciones en la elaboración y validación de documentos XML.
- f) Se han asociado las descripciones con los documentos.
- g) Se han utilizado herramientas específicas.
- h) Se han documentado las descripciones

RA4 Gestiona información en formato XML, analizando y utilizando tecnologías de almacenamiento y lenguajes de consulta.

- a) Se han identificado los principales métodos de almacenamiento de la información usada en documentos XML.
- b) Se han identificado los inconvenientes de almacenar información en formato XML.
- c) Se han establecido tecnologías eficientes de almacenamiento de información en función de sus características.
- d) Se han utilizado sistemas gestores de bases de datos relacionales en el almacenamiento de información en formato XML.
- e) Se han utilizado técnicas específicas para crear documentos XML a partir de información almacenada 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

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

Sistemas informáticos

RA1 Gestiona la información del sistema, identificando las estructuras de almacenamiento y aplicando medidas para asegurar la integridad de los datos.

- a) Se han comparado sistemas de archivos.
- b) Se ha identificado la estructura y función de los directorios del sistema operativo.
- c) Se han utilizado herramientas en entorno gráfico y comandos para localizar información en el sistema de archivos.
- d) Se han creado diferentes tipos de particiones y unidades lógicas.
- e) Se han realizado copias de seguridad.
- f) Se han automatizado tareas.
- g) Se han instalado y evaluado utilidades relacionadas con la gestión de información.

RA2 Interconecta sistemas en red, configurando dispositivos y protocolos.

- a) Se ha configurado el protocolo TCP/IP.
- b) Se han configurado redes de área local cableadas.
- c) Se han configurado redes de área local inalámbricas.
- d) Se han utilizado dispositivos de interconexión de redes.
- e) Se ha configurado el acceso a redes de área extensa.
- f) Se han gestionado puertos de comunicaciones.
- g) Se ha verificado el funcionamiento de la red mediante el uso de comandos y herramientas básicas.
- h) Se han aplicado protocolos seguros de comunicaciones.

RA1 Interpreta y utiliza información oral relacionada con el ámbito profesional del título, su formación personal, así como del producto/servicio que se ofrece, identificando y describiendo características y propiedades de los mismos, tipos de empresas y ubicación de las mismas.

- a) Se ha reconocido la finalidad del mensaje directo, telefónico o por otro medio auditivo.
- b) Se han emitido mensajes orales precisos y concretos para resolver situaciones puntuales: una cita, fechas y condiciones de envío/recepción de un producto, funcionamiento básico de una máquina/aparato.
- c) Se han reconocido las instrucciones orales y se han seguido las indicaciones emitidas en el contexto de la empresa.
- d) Se han utilizado los términos técnicos precisos para describir los productos o servicios propios del sector.
- e) Se ha tomado conciencia de la importancia de comprender globalmente un mensaje, sin necesidad de entender todos y cada uno de los elementos del mismo.
- f) Se han resumido las ideas principales de informaciones dadas, utilizando sus propios recursos lingüísticos.

RA2 Interpreta y cumplimenta documentos escritos propios del sector y de las transacciones comerciales internacionales: manual de características y de funcionamiento, hoja de pedido, hoja de recepción o entrega, facturas, reclamaciones.

- a) Se ha extraído información específica en mensajes relacionados con el

producto o servicio ofertado (folletos publicitarios, manual de funcionamiento) así como de aspectos cotidianos de la vida profesional.

c) Se ha interpretado el mensaje recibido a través de soportes telemáticos: e-mail, fax, entre otros.

f) Se ha utilizado correctamente la terminología y vocabulario específico de la profesión.

g) Se han utilizado las fórmulas de cortesía en presentaciones y despedidas propias del documento a elaborar.

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, casos de uso, diagramas de secuencia.
 4. Diseño y documentación del blog utilizando Wordpress.
 1. Arquitectura.
 2. Análisis de los usuarios y permisos.
 3. Análisis de las copias de seguridad.
 5. Diseñar la estructura del fichero de log sobre el estado de la red LAN.
 6. Diseño de los ficheros XML y sus plantillas de validación (DTD y XSD).
3. Desarrollo
 1. Script para creación de base de datos incluye disparadores para gestión del modelo
 2. Conexión a la base de datos, carga de datos y CRUD
 3. Interfaces de usuario (Vistas)
 4. Núcleo
 5. Procedimientos almacenados
 6. Programación de la lógica para el uso de los ficheros XML
 7. Instalación y configuración de Wordpress
 8. Script estado de red
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 este curso sobre Git: <https://goo.gl/QCU8he>.

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.

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 unidad de equipo en Google Drive en la que creará una hoja de cálculo compartida con el profesorado.
- 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 tratará 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.
- Diagramas de secuencia. Se debe desarrollar como mínimo 2 casos de uso.
- Implementación del blog y su documentación. Este documento contendrá:
 - Explicación de los conceptos front-end y back-end.
 - Explicación de la estructura de directorios utilizados y su uso.
 - Resultado del análisis sobre usuarios necesarios, tanto para el sistema operativo, como para la base de datos de WordPress.
 - Ficha del servidor que recoja las características más relevantes: nombre, procesador, RAM, sistema operativo, usuarios S.O., datos protocolo TCP/IP (dirección, puertos, ...), datos sistemas instalados (nombre del producto, versión, usuarios, directorios de instalación, ...)
 - Pasos de la instalación de WordPress, razonando lo que se hace en cada paso.

- Pasos de configuración de WordPress, explicando el porque de cada paso.
- Resultado del análisis de como realizar las copias de seguridad del blog.
- Licencia utilizada para los contenidos del blog.
- Relación de etiquetas para el contenido del blog.
- Entradas en el blog:
 - Entrada de presentación de los miembros del equipo de trabajo y del trabajo a desarrollar. La entrada será en castellano y en inglés dado que uno de los miembros directivos de E-Sport no habla castellano.
 - Entrada que muestre el plan de trabajo.
- Estructura del fichero de log para registrar el estado de la red.
- Estructura de los ficheros XML y de sus plantillas de validación (DTD y XSD).

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:

- Fichero DDL 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. Como mínimo incluirá dos disparadores
 1. Controlar que no haya más de 6 miembros en un equipo
 2. Controlar que el salario total de un equipo sea inferior a 200.000
- Fichero para la carga de datos inicial de base de datos si fuera necesario
- Codificación de las clases referentes al modelo.
- Interfaces de la aplicación. Estas interfaces estarán en una carpeta llamada Views .
- Procedimientos almacenados. Como mínimo dos paquetes de procedimientos almacenados y cada uno de ellos debe de incluir procedimientos que mejoren la eficiencia del sistema.
- 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.
- Entrada semanal en el blog (en castellano y en inglés).

FASE TRES

En esta fase se continua con el desarrollo del SI. En esta fase se deberían desarrollar los siguientes elementos:

- Clases necesarias para el manejo de los ficheros XML. Estas clases deben estar una carpeta que no sea la de las vistas, controladores ni núcleo.
- Modificación del núcleo para utilizar los ficheros XML.
- Codificación del script para monitorizar el estado de la red.
- Entrada semanal en el blog (en castellano y en inglés).

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:

- Documento que explique la forma en la que se realizan las copias de seguridad del blog implementado en WordPress.
- 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:
 - Blog.
 - Base de datos.
 - Proyecto Java.
 - Script para la gestión del estado de la red.

- 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. La presentación/exposición comenzará con una breve introducción en inglés para presentar el equipo y el proyecto.

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.
- 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.

COMPETENCIAS TÉCNICAS

Las rúbricas para la evaluación de las competencias técnicas se publicarán en [SET](#).

COMPETENCIAS TRANSVERSALES

Las rúbricas para la evaluación de las competencias transversales de FP de Euskadi se publicarán en [SET](#).

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=1080>

Profesores de referencia

Módulo	Profesor	Email
Bases de datos Sistemas informáticos	Blanca Isasi-Isasmendi	bisasi@egibide.org
Programación	Nieves Ruiz	mnruiz@egibide.org
Entornos de desarrollo	Ion Marín	imarin@egibide.org
Lenguajes de marcas	Marcos García	mgarciadegodos@egibide.org
Inglés Técnico	Izaskun Arratibel	iarratibel@egibide.org

TEMPORIZACIÓN

El reto comienza el día 16/04/2018 y termina el 31/05/2018, 7 semanas en total.

Calendario semanal previsto

	L	M	X	J	V
Semana	16/04	17/04	18/04	19/04	20/04
1	Reto	Reto	Reto	Reto	Primer entregable

	L	M	X	J	V
Semana	23/04	24/04	25/04	26/04	27/04
2	Reto	Reto	Reto	Reto	Reto

	L	M	X	J	V
Semana	30/04	01/05	02/05	03/05	04/05
3	Puente	Puente	Reto	Reto	Reto

	L	M	X	J	V
Semana	07/05	08/05	09/05	10/05	11/05
4	Reto	Reto	Reto	Reto	Reto

	L	M	X	J	V
Semana	14/05	15/05	16/05	17/05	18/05
5	Reto	Reto	Reto	Reto	Reto

	L	M	X	J	V
Semana	21/05	22/05	23/05	24/05	25/05
6	Entrega	Examen IT Preparar presentación	Presentación Transversales	Examen PROG	Examen LM

	L	M	X	J	V
Semana	28/05	29/05	30/05	31/05	01/06
7	Examen GBD	Examen ED	Examen SI	Entrevistas	

FORMATO PARA ENTREGA DE TAREAS

Tipo de letra

Se utilizará un único tipo de letra en todo el documento en los siguientes tamaños:

- Título o enunciado de cada apartado: tamaño 12 negrita.
- Texto: tamaño 10-12 normal.
- Nota al pie, leyendas y figuras: tamaño 9 normal.

Párrafo

Párrafos justificados en los márgenes derecho e izquierdo o alineados a la izquierda.
Espacio sencillo en el cuerpo del párrafo y espacio de 1,5 – 2 entre los párrafos.

Encabezado y pie de página

En el encabezado debe aparecer el nombre del fichero y la fecha.
En el pie de página debe aparecer el número de página.

Capturas de pantalla e imágenes

Las capturas e imágenes tienen que ser legibles al 100% del tamaño del documento.
Las capturas tienen que ser de la instalación del alumno y será perfectamente visible la fecha y hora.

Citas y bibliografía

Siempre que se vaya a citar, copiar o utilizar cualquier definición o explicación de una página hay que indicarlo.

Portada

En la portada deberá aparecer:

- Módulo, unidad o reto.
- Grupo o nombre y apellidos del alumno.
- Índice. (ejemplo: <https://www.youtube.com/watch?v=GN5eAEqP8o8>)

Tablas

Competencia	Insuficiente	Suficiente	Destacado
Texto	Texto	Texto	Texto