



APRENDIZAJE COLABORATIVO BASADO EN RETOS

2018/2019

RETO: “ARABA MENDI TALDEA”

DAM

DESARROLLO DE APLICACIONES MULTIPLATAFORMA

1

MÓDULOS IMPLICADOS	
0484. Bases de datos 0485. Programación 0487. Entornos de desarrollo	
DURACIÓN	ORGANIZACIÓN
6 semanas	Grupos de 2-4 personas



ÍNDICE

El reto.....	3
Objetivos / Resultados de Aprendizaje.....	6
Bases de datos.....	6
Programación.....	7
Entornos de desarrollo.....	10
Tareas a realizar.....	12
Esquema orientativo de pasos a seguir.....	12
Obtener información.....	12
Explorar estrategias.....	12
Actuar.....	12
FASE UNO.....	12
FASE DOS.....	13
FASE TRES.....	14
Logros.....	14
Criterios de evaluación.....	15
Criterios para la elaboración de las calificaciones parciales.....	15
Recursos.....	16
Profesores de referencia.....	16
Temporización.....	17
Calendario semanal previsto.....	17

EL RETO

El grupo de montaña Araba os ha contratado para que realicéis un sistema para gestionar la información que necesitan digitalmente.

En el grupo de montaña hay socios adultos y menores de edad que tendrán asignado un responsable. También existe una junta formada por socios del club que periódicamente realizan reuniones que son un tipo de actividad.

Existen diferentes tipos de cuotas dependiendo si es infantil o de adulto cuyo importe va cambiando anualmente. Hay que almacenar que cada socio abone la cuota que le corresponde anualmente.

El grupo de montaña ofrece diferentes tipos de actividades que organizan los socios.

Desean que se almacene la siguiente información de:

- Perfiles de acceso. Los perfiles disponibles serán Administrador/a y usuario/a.
- Socios: Cada socio vendrá descrito por un código de socio, su nombre y apellidos, la fecha de nacimiento. También se recogerá el DNI de aquellos socios que lo tengan(puede que los menores de edad no lo tengan), el teléfono, el email, y el perfil con el que puede acceder a la aplicación. Es interesante saber la fecha en la que se dio de alta en el club, así como si ha causado baja también saber cuando ha tenido lugar. Es necesario para todos los socios menores de edad tener un responsable, que obviamente será un socio mayor de edad.
- Cargos de la junta. Los cargos serán Presidente/a, Vicepresidente/a, Secretario/a, Tesorero/a y vocal.
- Junta: Se debe recoger la información de la junta, con sus socios y el cargo que ostentan, además de las fechas de inicio de esa junta y de fin.
- Tipos de cuotas: Es necesario almacenar una relación de los distintos tipos de cuotas a pagar, por ejemplo dependiendo de la edad, el importe de la misma y el año de validez de dicha cuota. Hay que conocer si los socios han pagado de forma correcta su respectiva cuota.
- Tipos de actividades. Los tipos de actividades serán 'SALIDA AL MONTE', 'ALBERGUE FIN DE SEMANA', 'REUNIÓN', 'COMIDA' y 'OTROS'.
- Actividades: Las actividades programadas se distinguen a través de un código, se debe saber con que tipo de actividad están vinculadas, el socio que la organiza así como la fecha en la que se celebra, junto con una breve descripción. También interesa saber la dificultad de la misma(“ALTA” o “MEDIA” o “BAJA”). Cada actividad tiene un precio. Se puede dar la situación de que se suspenda, es conveniente

conocer los motivos de dicha suspensión. En el caso en que no se haya suspendido interesa conocer los socios que han asistido a ella.

Araba Mendi Taldea desea establecer las siguientes restricciones:

- Los socios menores de edad deberán tener un responsable adulto que también sea socio.
- No se admiten socios menores de 4 años.
- Los cargos de la junta serán: presidente/a, vicepresidente/a, secretario/a, tesorero/a y 5 vocales.
- En la junta un socio podrá ostentar varios cargos diferentes en diferentes fechas, pero no se pueden solapar los periodos de cada uno de los cargos.
- Los socios deberán abonar su cuota antes del 31/12 del año anterior al que corresponde la cuota, si no, se le dará de baja a él y los socios de los que sea responsable.
- Todos los socios tendrán acceso a la aplicación. Por defecto todos tendrán perfil “Usuario” y los miembros de la junta tendrán perfil “Administrador”

En lo que se refiere al acceso a la aplicación:

- Únicamente los usuarios con perfil administrador serán los encargados de realizar las siguientes tareas:
 - CRUD de tipos de actividad, junta, cargos, tipos de cuotas, cuotas y socios.
 - Generar el calendario anual de actividades, decidirán las fechas en las que se realizarán las salidas, albergues, comidas y otras actividades. (Las reuniones no aparecerán en el calendario)
- Todos los socios serán los encargados de realizar las siguientes tareas:
 - Organizar una o varias actividades en base a las fechas propuestas por la junta, rellenando los datos solicitados para la organización de la actividad y una vez realizada la actividad rellenar el resto de datos.
 - Proponer y organizar las actividades a realizar para el año, cumpliendo las restricciones marcadas en el enunciado.
 - Visualizar el calendario de actividades; de las que ya se ha realizado y de las próximas.

- Sólo los usuarios administradores podrán ver todos los datos de los socios debido a la ley de protección de datos, el resto de usuarios sólo tendrán acceso al nombre y apellido de los socios.
- Cada socio podrá acceder a su ficha de socio para actualizar sus datos de teléfono o email.
- Todos los usuarios pueden consultar el calendario, las actividades y quienes son los miembros de la junta.

En cuanto al interfaz de usuario, se desea quiere que sea lo más intuitivo posible.

Le gustaría que al entrar se pueda ver el calendario de las actividades planificadas para lo que se utilizará la vista Calendario.

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.
- g) Se han creado los usuarios y se les han asignado privilegios.
- 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 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.

- 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.

RA8 Utiliza Bases de Datos Orientadas a Objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

- a) Se han identificado las características de las Bases de Datos Orientadas a Objetos.
- b) Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.
- c) Se han instalado sistemas gestores de bases de datos orientados a objetos.
- d) Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.
- e) Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.
- f) Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.
- g) Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.
- h) Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.

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

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.

TAREAS A REALIZAR

Esquema orientativo de pasos a seguir

1. Análisis del proyecto.
2. Diseño
 1. Modelo entidad relación
 2. Diagrama de clases, de casos de uso y de secuencia
3. Desarrollo
 1. Script para creación de base de datos incluye disparadores para gestión del modelo
 2. Interfaz de usuario (vistas)
 3. Desarrollo de la lógica de la aplicación (controladores).
 4. Procedimientos almacenados (modelo)
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: <http://bit.ly/2X1WGq1>.

Al final de la fase uno es obligatorio entregar los elementos especificados.

FASE UNO

Esta fase incluye los puntos uno y dos de los pasos a seguir. Lo primero que se debe hacer 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.
- 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.

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 un disparador que controle que que los socios menores de edad tengan un responsable que será otro socio.
- En el fichero DDL se incluirá una vista “Calendario” en la que se tendrá información para cada año de las actividades planificadas y realizadas y del socio que la organiza. La vista mostrará la siguiente información: fecha de la actividad, descripción del tipo de actividad, descripción de la actividad, precio si es diferente de 0.0 y nombre y apellido del socio que la organiza. Y si la actividad ya se ha realizado se mostrará también la información de asistentes o motivo de la suspensión.
- Fichero para la carga de datos inicial de base de datos si fuera necesario
- Codificación de las clases referentes al modelo (Socio, Actividad, Cuota...), conexión (Clase que gestione la conexión y clase que incluya los parámetros de la conexión) y acceso a datos. Cada una de las tablas de la base de datos a la que se vaya acceder debe tener una única clase de acceso a datos.
- Interfaces de la aplicación. Estarán en una carpeta llamada Views y solo deben tener llamadas a los controladores que se desarrollaran más adelante.
- Procedimientos almacenados. Como mínimo un paquete de procedimientos almacenados que incluya procedimientos que mejoren la eficiencia del sistema.
- Desarrollo de los controladores de las vistas que deben estar en una carpeta llamada Controllers.

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 deben entregar los siguientes elementos:

- Fichero que incluya las pruebas unitarias realizadas y capturas de las pruebas realizadas con sus respectivos resultados.
- Documentación del proyecto. Esta documentación debe incluir la documentación obtenida mediante javadoc y un manual de usuario. Este manual 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 usará para la defensa del proyecto.

Logros

- Sistema de información en funcionamiento:
 - Base de datos.
 - Proyecto Java.
- Repositorio de GitHub/GitLab con los productos desarrollados.
- 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 de cada módulo se obtendrá de la siguiente manera:
 - 60% valoración del reto.
 - 40% examen escrito.
- Para dar el módulo por superado, habrá que obtener un mínimo de cuatro puntos en cada una de las partes.

RECURSOS

Profesores de referencia

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

TEMPORIZACIÓN

El reto comienza el día 02/05/2019 y termina el 24/05/2019, 4 semanas en total.

Calendario semanal previsto

	L	M	X	J	V
Semana				02/05	03/05
1				Reto	ER y clases

	L	M	X	J	V
Semana	06/05	07/05	08/05	09/05	10/05
2	Reto	Reto	Reto	Reto	Reto

	L	M	X	J	V
Semana	13/05	14/05	15/05	16/05	17/05
3	Reto	Reto	Reto	Reto	Reto

	L	M	X	J	V
Semana	20/05	21/05	22/05	23/05	24/05
4	Reto	Reto	Reto	Reto	Entrega

	L	M	X	J	V
Semana	27/05	28/05	29/05	30/05	31/05
5	Examen BD	Examen Prog	Presentación	Examen ED	