

## Proyecto transversal Primero de DUAL 24-25

|   |          |
|---|----------|
| <b>Proyecto transversal Primero de DUAL 24-25</b> | <b>1</b> |
| Consideraciones:                                  | 2        |
| Entregables:                                      | 2        |
| Definición de casos de usos:                      | 2        |
| Bocetos:  | 3        |
| Modelo de datos:                                  | 3        |
| Frontend:   | 3        |
| Base de datos:                                    | 3        |
| Backend:  | 3        |
| Ayuda Proyecto web en Eclipse:                    | 4        |
| Pasos de instalación proyecto ejemplo:            | 4        |
| Estructura del proyecto:                          | 4        |
| Proyectos:  | 6        |
| Sistema: Torneo de Debate                         | 6        |
| Sistema: Competición deportiva                    | 7        |
| Sistema: Gestión de biblioteca (NTT Data)         | 7        |

## Consideraciones:

- El objetivo de este proyecto es que pongas en prácticas diferentes conceptos vistos en los módulos de este primer curso de manera transversal.
- Además, se persigue un segundo objetivo: que seas capaz de investigar y aplicar conceptos nuevos.
- Como ayuda te proporcionaremos diferentes materiales: un documento con los estilos de diseño para css, un proyecto ejemplo para que te sirva de guía en la conexión de cada parte.
- Este proyecto es evaluable para los diferentes módulos implicados.

**TODO: Definición de los criterios de evaluación de cada módulo así como rúbrica de los diferentes apartados.**

- **LENGUAJE DE MARCAS:** Tanto el documento de estilos como la rúbrica de evaluación están disponibles en la moodle de lenguaje de marcas.

## Entregables:

El proyecto se entregará en un repositorio de **GITHUB** que deberá contener :

1. Un **documento** con los siguientes apartados: **TODO Rúbrica**
  - a. Definición de caso de uso (Entornos)
  - b. Bocetos (LM)
  - c. Modelo de datos Definición de entidades y relaciones (E/R) (BD)
  - d. UML: Diagrama de clases (Entornos)
2. **Vídeo** dónde se explica muestra la funcionalidad (Entornos) **TODO Rúbrica**
3. Carpeta **frontend** con subcarpetas: html y css (LM)
4. Carpeta **scriptBD**: baseDato.sql (tabla e inserts) (BD) **TODO Rúbrica**
5. Carpeta **src**: con subcarpetas: modelo, repositorio, jdbc, servlet El servlet será proporcionado (Programación) **TODO Rúbrica**

## Definición de casos de usos:

Especificación de cada caso de uso.

El documento deberá incluir una lista de casos de usos. Para cada caso de uso debéis definir los siguientes puntos:

- Identificador Caso de uso
- Título del Caso de Uso
- Descripción
- Precondición. Situación o estado previo. Por ejemplo, que el usuario esté logueado o que se haya realizado una búsqueda y selección de un producto.
- Requisitos relacionados
- Secuencia de pasos que ocurren.
- Postcondición. Situación final que se alcanza tras la secuencia de pasos.
- Excepciones. Situaciones que pueden hacer que no se alcance la postcondición, definiendo la secuencia de pasos a seguir en ese caso. Por

ejemplo, que no haya existencias de un artículo o que no se haya podido realizar el pago.

### **Bocetos:**

Para cada caso de uso: definición de las pantallas implicadas con boceto y ruta de navegación (en figma)

### **Modelo de datos:**

- Definición de entidades: atributos, claves primarias, claves ajenas y relaciones con cardinalidades
- E/R
- Definición de tablas
- Diagrama UML de clases

### **Frontend:**

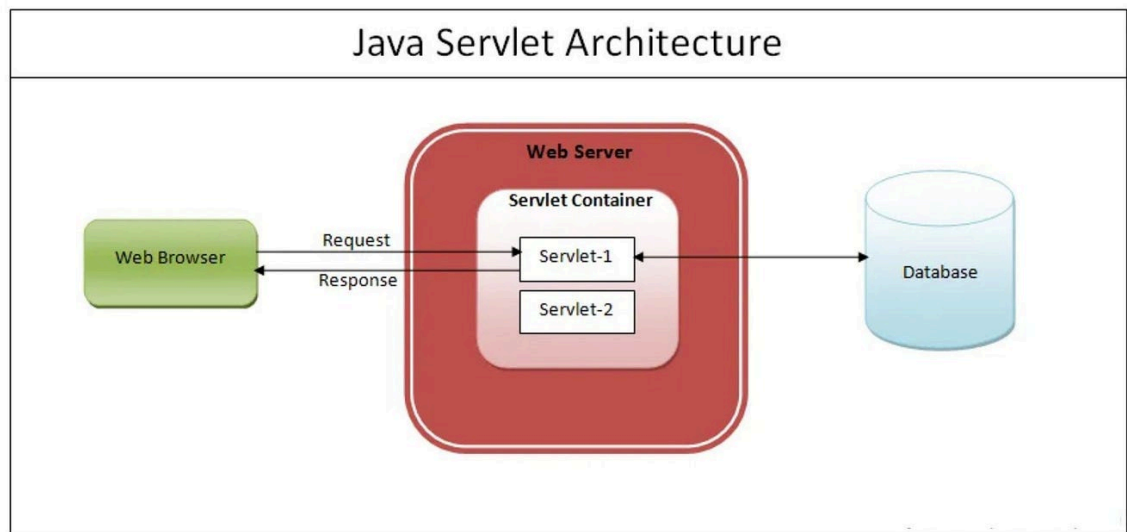
Estas carpetas deberán contener los ficheros htmls aplicando el estilo proporcionado (envío de datos mediante post a servlet proporcionado)

### **Base de datos:**

Deberá incluir la creación de cada tabla del modelo, así como los usuarios, los inserts para que las tablas tengan datos en una base de datos relacional mysql.

### **Backend:**

- Deberás implementar el diagrama de clases de acuerdo a tu diagrama de clases
- Considera que los datos del html viajarán por petición post y será una clase Servlet (proporcionada) la que recoja la petición. En esta clase tendrás que añadir la ruta de la página que sea y el método que recoja los datos recibidos.
- Los datos se persistirán mediante jdbc. Para esta parte puedes apoyarte en el material proporcionado en la [unidad persistencia](#)  
Para que te sirva de guía te proporcionaremos un proyecto ejemplo con un formulario html, el servlet que recoge la petición y la persistencia de esos datos en una base de datos mediante jdbc

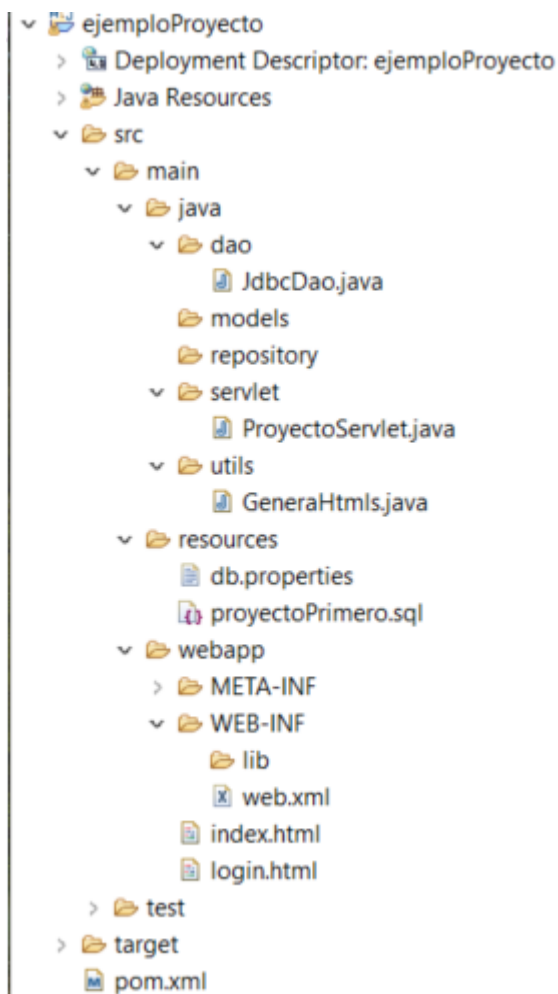


## Ayuda Proyecto web en Eclipse:

### Pasos de instalación proyecto ejemplo:

1. Descarga y descomprime en tu workspace el proyecto ejemplo.New Maven Project.
2. Ejecuta Run as Maven Clean
3. Ejecuta Run as Maven install
4. Instala en Eclipse un servidor Apache 9
5. Agrega tu proyecto para que se despliegue en el servidor
6. Comprueba que se ha desplegado correctamente abriendo esta url en el navegador:  
<http://localhost:8080/ejemploProyecto/login.html>

## Estructura del proyecto:



### **Carpeta src/main/java** → Clases Java

dao → JdbcDao.java Contendrá todas las consultas a la BD

servlet → Esta carpeta contendrá las clases servlets que recogerán las peticiones http request y conectarán con nuestro modelo y dao. Si agregas otro servlet, deberás manejar el modificar el fichero web.xml para mapear qué peticiones (url) mapea ese servlet  
utils → GeneraHtmls Aquí tienes que añadir los métodos que te generan html cuando necesitamos que sea dinámico.

### **Carpeta src/main/resources** Contiene:

db.properties → Con los datos de conexión a la BD, url, usuario y clave

proyectoPrimero.sql → Contiene el script sql de creación de la base de datos de ejemplo, de la tabla que usa así como inserts. Tendrás que crear uno parecido para tu BASE DE DATOS

### **Carpeta webapp/WEB-INF:** Contiene:

Los ficheros html, los ficheros css

Un fichero llamado web.xml que es el que contiene el mapeo de los servlets y la ruta base por defecto

## Proyectos:

### Sistema: Torneo de Debate

- Demo: 5/05/25
- Caso de Uso:

#### Grupo 1: Equipos e index Juanma

1. Index y menú
2. Alta de participantes en equipos
3. Listado y búsqueda de equipos por fase
4. Consulta y modificación del detalle de un equipo

#### Grupo 2: Puntuaciones Eva

5. Registro de puntuaciones de una prueba de un equipo: fase, desplegable equipos de esa fase para equipo A y equipo B, puntuación por equipo, observaciones y penalizaciones por equipo, equipo ganador (selecciona por defecto mayor puntuación), si mejor orador: lista participantes de ambos equipos.
6. Listado y búsqueda del resultado de los debates por fase
7. Consulta y modificación del detalle del resultado de un cruce: puntuación equipo A, puntuación equipo B, observaciones, equipo ganador y mejor orador (opcional)

#### Grupo 3: Pruebas Cristina & Daniel

8. Alta de pruebas: idFase, nombre de prueba, número de equipos (par), número de jueces, mejorOrador(V/F), checkbox salas
9. Sorteo de cruces por fase (a partir del listado de equipos por una fase, botón sorteo y generar los cruces y la sala)
10. Listado de cruces por fases (si celebrado: enlace al resultado del cruce)

#### Grupo 4: Listados Masté & Rodrigo

11.

12. Listado de mejores oradores ordenados por puntuación.
13. Consulta del detalle de la puntuación del mejor orador.
14. Listados equipos clasificados por fases
15. Listado equipos participantes por fases

## PUNTO 1 : **Definición de casos de usos:**

Especificación de cada caso de uso.

El documento deberá incluir una lista de casos de usos. Para cada caso de uso debéis definir los siguientes puntos:

- Identificador Caso de uso
- Título del Caso de Uso
- Descripción
- Precondición. Situación o estado previo. Por ejemplo, que el usuario esté logueado o que se haya realizado una búsqueda y selección de un producto.
- Requisitos relacionados
- Secuencia de pasos que ocurren.
- Postcondición. Situación final que se alcanza tras la secuencia de pasos.
- Excepciones. Situaciones que pueden hacer que no se alcance la postcondición, definiendo la secuencia de pasos a seguir en ese caso. Por ejemplo, que no haya existencias de un artículo o que no se haya podido realizar el pago.

### **Identificador de caso de uso .**

Nuestro programa está muy resumido como una administración donde tendremos a los participantes del Torneo y sus respectivos Equipos. Donde esté

### **Título del Caso de Uso.**

El título es Torneo Debate.

### **Descripción.**

Lo que haremos es donde lo primero que tendremos un menú (en java) donde en este se podrá :

1. Añadir participante.
2. Formar Equipos (Equipos con el mismo número personas)
3. Equipos Por Fase
4. Modificación de Equipos

También tendrá 2 listas donde en una tendremos a las personas que se han inscrito en el Torneo y otra de los Equipos que se han formado.

Después tendremos otra lista donde en esta tendremos a los equipos donde nos dirán en qué fase se encuentran cada uno de ellos.

Y por último tendremos que mediante una base de datos hacer consultas de los detalles de cada equipo.

### **Precondición.**

La precondición sería que una vez logueado este tendría que inscribirse al torneo de debate donde después se unirá a un equipo.

### **Requisitos Relacionados.**

Para la parte de Alta de participantes tendremos una página donde tenemos los nombres y datos de los participantes.

Para el listado y búsqueda de equipos tenemos una página donde tendremos el nombre y participantes que están en un equipo.

Y por último las consultas y modificación del detalle de un equipo tendremos una página donde se encuentran los los equipos y podremos ver los detalles del equipo que seleccionemos.

### **Secuencia de pasos que ocurren.**

1. Primero tenemos una página con un menú donde se podrá interactuar con las siguientes opciones : Alta de participantes en equipos Listado y búsqueda de equipos por fase y Consulta y modificación del detalle de un equipo.
2. En segundo lugar una vez que el usuario se haya dado de alta (registrado en el torneo) este se unirá a un equipo donde dicho equipo se enfrentará a otro por pasar de fase. Donde se necesitara saber el nombre del equipo y los participantes que están en el.
3. Después tendremos una página donde nos aparecerá que equipos habran pasado de fase.



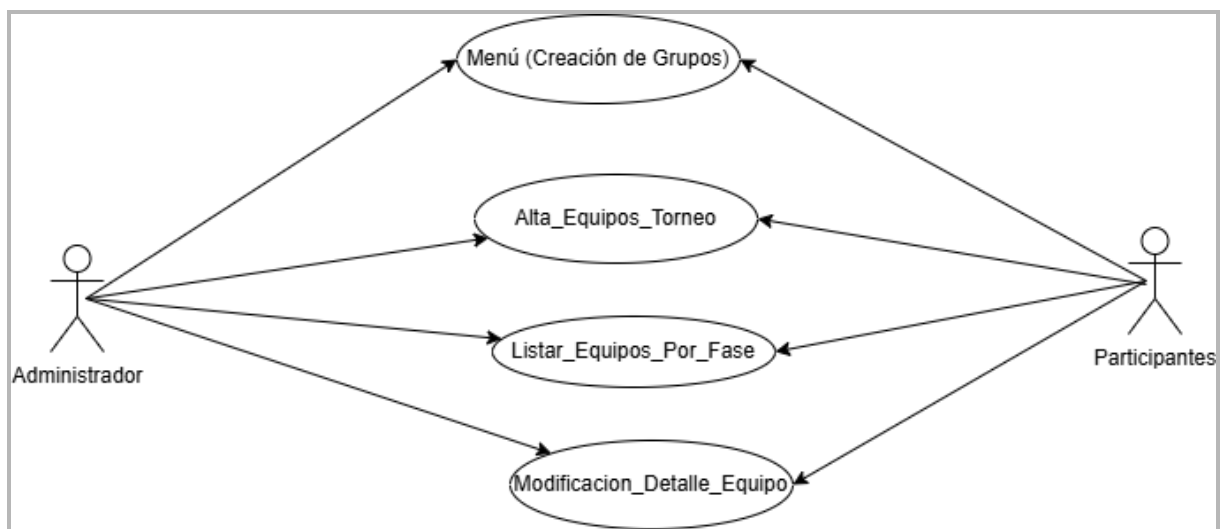
4. Y por último tenemos una página donde nos aparecerán los diferentes equipos como en la anterior pero la diferencia será que si seleccionas un equipo te aparecerán los datos y participantes de ese equipo.

### Post-Condición.

La Post-Condición será saber el nombre de cada equipo y participante del Torneo de Debate.

### Excepciones.

### Diagrama casos de uso Imagen



### Bocetos:

Para cada caso de uso: definición de las pantallas implicadas con boceto y ruta de navegación (en figma)

## **Modelo de datos:**

- Definición de entidades: atributos, claves primarias, claves ajenas y relaciones con cardinalidades

Tenemos una Entidad llamada Administrador donde esta contendrá los siguientes atributos :

id\_administrador (clave primaria), nombre, dirección, teléfono y una dirección .

Después tendremos otra entidad llamada Participantes donde sus atributos son :

id\_participantes (clave primaria), nombre, apellidos, dirección, teléfono, correo y una dirección .

Una vez tengamos los participantes apuntados al torneo, tendremos una entidad llamada equipo donde tendrá los siguientes atributos: id\_equipo (clave primaria)

- E/R

- Definición de tablas

Tabla Equipos : tenemos los atributos id\_equipo, nombreEquipo y anioCreacion.

Tabla Participantes : tenemos los atributos id\_participante, nombre, apellidos telefono y correo.

Tabla Fase : tenemos los atributos id\_fase, nombreFase.

- Diagrama UML de clases

