

	<p>¿Aceptas el Reto?</p> <p>PROYECTO INTERMODULAR COMERCIO INTERNACIONAL - INFORMÁTICA Y COMUNICACIONES</p>	
	<p>MINIHITO 2 DI DAM2 (18/11/2025)</p>	<p>Rosa María Zapata Calle/Alfonso Sánchez Caro</p>

Partimos del análisis desarrollado en clase, en el que después de desarrollar el diagrama de casos de uso, tendremos el siguiente prototipo, y recursos para poder desarrollar apartados de este proyecto mediante minihitos.

SIMPLIFICACIÓN POR LIMITACIÓN TEMPORAL:

- Se proporcionará el modelo de la base de datos a modo de simplificación.
- Se reduce a la carga del alumnado en la parte 2, siendo opcional la gestión (creación, modificación y eliminación de grupos). SÓLO SE SOLICITA LA PARTE DE LA VISTA.

En este minihito nos encargaremos de desarrollar la funcionalidad “participantes”, según vemos en el siguiente propotipado:

Para poder ser evaluado, el proyecto deberá estar separado en capas, (vista, dominio y persistencia) obligatoriamente. Así mismo, el código debe estar modularizado.

PARTE 1.

A. Pantalla 1. funcionalidad participantes-alumnado



Lo que se pretende es que **se inserten en la base de datos** 10 elementos en la tabla “alumnado”.

Tendremos en cuenta las siguientes consideraciones:

El id no se mostrará en el datagrid. Sólo se mostrarán los datos identificativos del alumnado(nombre, apellidos y curso) . El campo curso, será o SCI1, SCI2 correspondiente a primer curso de comercio internacional y segundo curso de comercio internacional.

Se podrá **modificar y eliminar también en base de datos**, de forma sincronizada con la vista del proyecto.

	<p>¿Aceptas el Reto?</p> <p>PROYECTO INTERMODULAR COMERCIO INTERNACIONAL - INFORMÁTICA Y COMUNICACIONES</p>	
	<p>MINIHITO 2 DI DAM2 (18/11/2025)</p>	<p>Rosa María Zapata Calle/Alfonso Sánchez Caro</p>

TAREAS A REALIZAR: (Máx. 4 puntos)

- La interfaz deberá :
- Cargar en la vista el alumnado que exista en la base de datos.
- Añadir nuevo alumnado tanto en la vista, de forma sincronizada en la base de datos.
- Modificar alumnado tanto en la vista como en la base de datos.
- Eliminar alumnado tanto en vista como en la base de datos, con confirmación de eliminación.

EVALUACIÓN:

<p>-Crear la interfaz gráfica con los contenedores pertinentes y controles pertinentes.</p> <p>-Conexión de base de datos.(0,5 cargar;0,5 crear;0,5 modificar, 0,5 eliminar)</p>	<p>(máximo 1,5 puntos)</p> <p>(máximo 2 puntos)</p>
---	---

PARTE 2.

B.Pantalla 1. funcionalidad participantes-grupos



En este apartado, como vemos tenemos varios componentes en los que se muestran un listado de alumnado no asignado a ningún grupo en el lado izquierdo y un listado de aquellos seleccionados en la derecha. Se deberán pasar con los diferentes botones que están diseñados para ello.

En el lado derecho, tenemos un componente donde incluiremos el nombre del grupo y un botón para crear el mismo grupo.

Un vez creado el grupo, mediante el botón a tal fin definido, aparecerá en el listado inferior que a priori tendrá el listado de grupos existentes.

Aquellos grupos creados que podemos ver en el listado inferior, se podrán seleccionar, modificar tanto el nombre como los componentes del mismo.

También se podrá eliminar grupos, liberando en este caso el alumnado que estaba en este grupo, que lo que hará en volver a pertenecer al grupo de alumnos sin asignar.

	¿Aceptas el Reto? PROYECTO INTERMODULAR COMERCIO INTERNACIONAL - INFORMÁTICA Y COMUNICACIONES	
	MINIHITO 2 DI DAM2 (18/11/2025)	<i>Rosa María Zapata Calle/Alfonso Sánchez Caro</i>

*tips ayuda: En cuanto a la modificación, podremos utilizar el mismo botón o tener los botones crear y modificar, por separado, de cara a utilizar la respuesta más personal y propia que tengais.

Tareas a realizar:

- Cargar alumnado no asignado (cuyo campo sea nulo) en el listado de alumnado no asignado.
- Crear grupo con alumnado no asignado (NO OBLIGATORIO CARGAR EN BASE DE DATOS, SÓLO EN VISTA CON DATABIDING AUNQUE SÍ VALORABLE).
- Modificar grupo (NO OBLIGATORIO CARGAR EN BASE DE DATOS, SÓLO EN VISTA CON DATABIDING AUNQUE SÍ VALORABLE).
- Eliminar grupo (NO OBLIGATORIO CARGAR EN BASE DE DATOS, SÓLO EN VISTA CON DATABIDING AUNQUE SÍ VALORABLE).

En eliminar, se deberá mostrar una ventana donde se confirme que se quiere eliminar.

-Crear la interfaz gráfica con los contenedores pertinentes y controles pertinentes. -Cargar alumnado no asignado a grupo -Funcionalidad completa a nivel de vista	(Máximo 2 puntos) (Máximo 1 punto) (Máximo 2,5 puntos)
--	--

Valoración opcional:

CRUD GRUPOS EN BASE DE DATOS	<i>(Máximo 1 punto)</i>
-------------------------------------	-------------------------

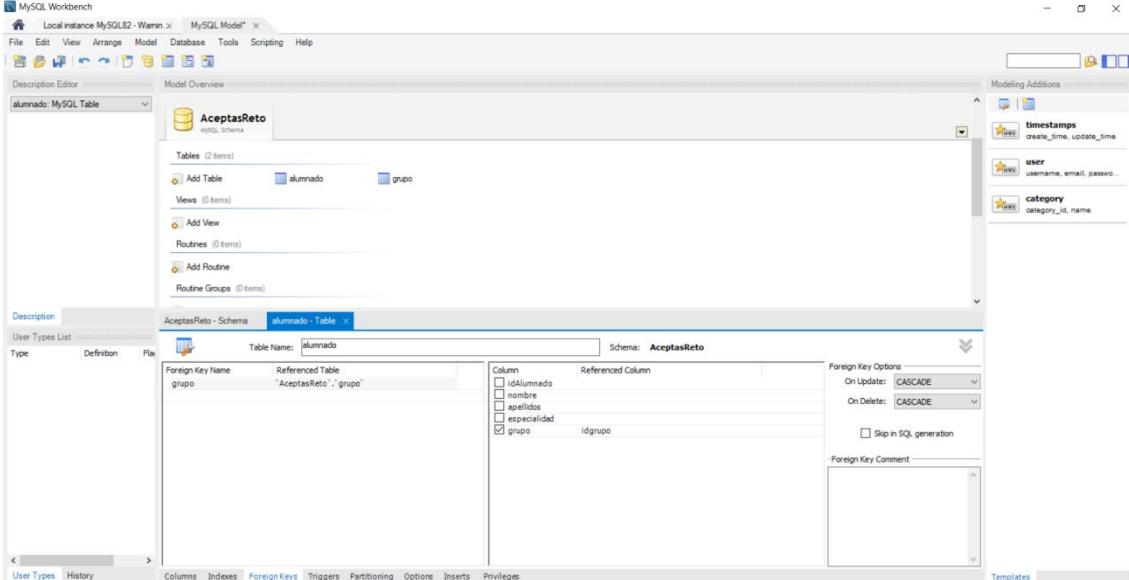
1. Mediante WPF en Visual Studio.net, DISEÑAR el interfaz e IMPLEMENTAR la funcionalidad teniendo en cuenta los patrones de diseño Maestro/Esclavo y Modelo-Vista-Controlador.
2. Se debe subir la tarea realizada a la plataforma Educamos CLM comprimida en un documento ZIP. Así como el modelo, cuyo espacio de nombres se deberá llamar “aceptasreto”.

	<p>¿Aceptas el Reto?</p> <p>PROYECTO INTERMODULAR COMERCIO INTERNACIONAL - INFORMÁTICA Y COMUNICACIONES</p>	
	<p>MINIHITO 2 DI DAM2 (18/11/2025)</p>	<p>Rosa María Zapata Calle/Alfonso Sánchez Caro</p>

RECURSOS DE AYUDA:

MODELO DE BASE DE DATOS:

Crear una relación foránea entre tablas:



The screenshot shows the MySQL Workbench interface with the following details:

- File Edit View Arrange Model Database Tools Scripting Help**
- Description Editor**: Shows the schema **AceptasReto** and table **alumnado**.
- Model Overview**: Shows the **Tables** section with **alumnado** and **grupo** listed.
- User Types List**: Shows the **alumnado - Table** definition.
- Foreign Keys**: Shows a foreign key relationship being defined from **alumnado** to **grupo**. The **Foreign Key Name** is **grupo**, and the **Referenced Table** is **'AceptasReto'.'grupo'**. The **Column** is **idgrupo** and the **Referenced Column** is **idgrupo**. The **On Update** and **On Delete** options are set to **CASCADE**.
- Modeling Additions**: Shows **timestamps**, **user**, and **category** models.

-SELECT distinct g.nombre FROM alumnado a inner join grupo g on a.grupo=g.idgrupo;

-SELECT distinct g.nombre FROM alumnado a, grupo g where a.grupo=g.idgrupo;