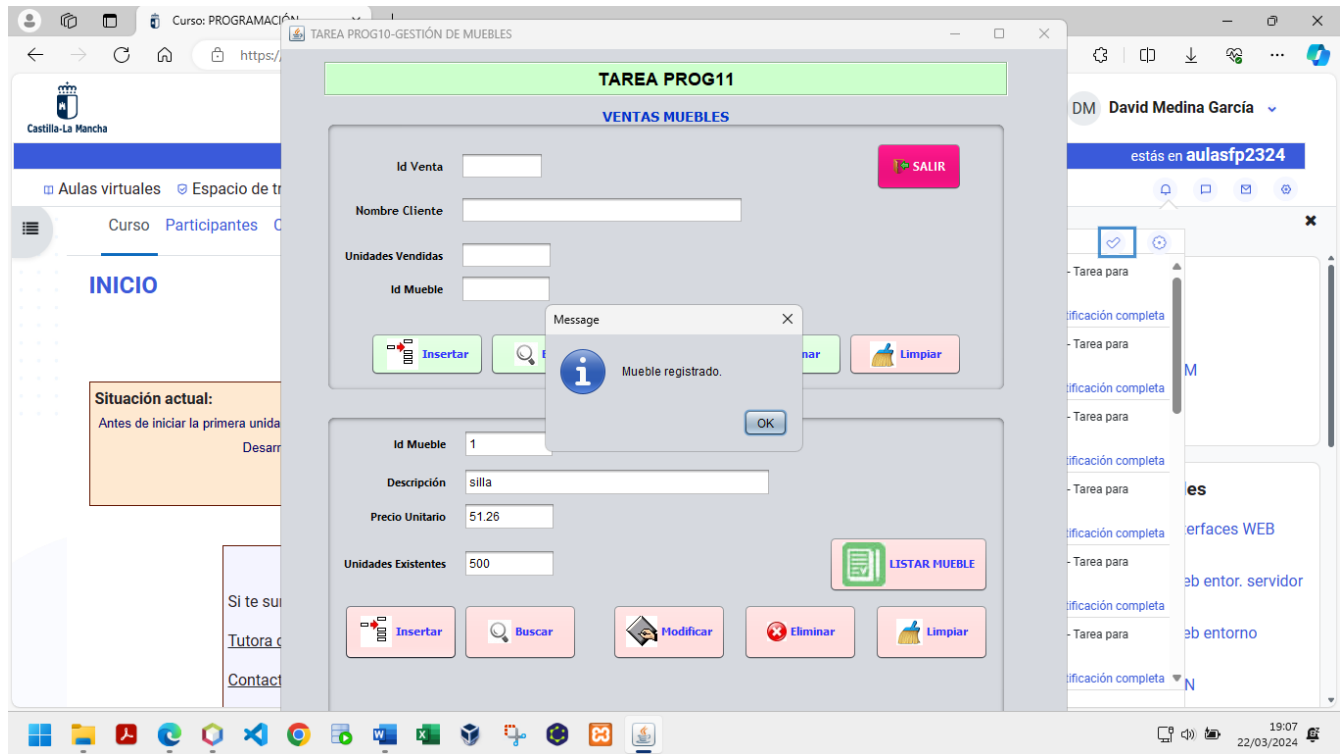


**CICLO : DESARROLLO APLICACIONES WEB-DISTANCIA**  
**MÓDULO : PROGRAMACION**  
**ALUMNO : DAVID MEDINA GARCIA**  
**TAREA : PROG11**

- **Insertar muebles**



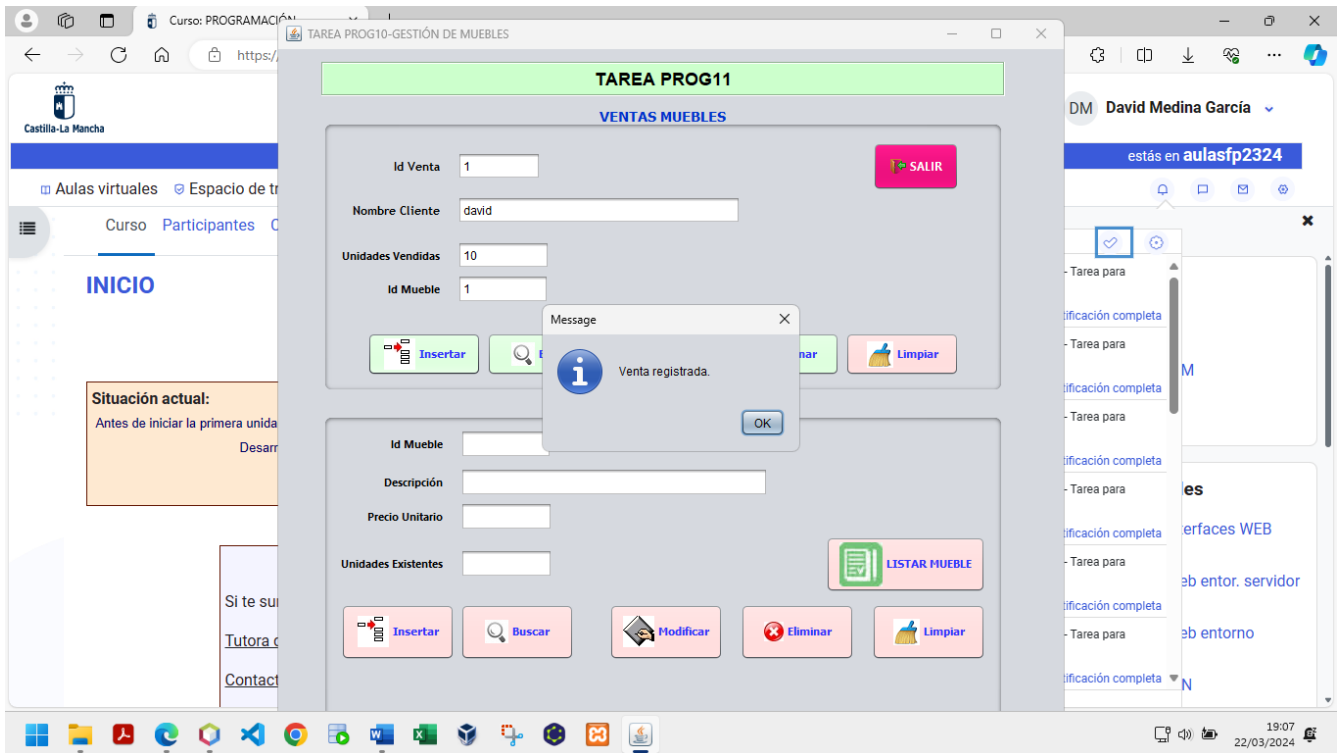
En este apartado se usarán los siguientes métodos:

- **verificarCamposVacios:** Se le pasan una serie de parámetros para comprobar que no estén vacíos.
- **validarCampos:** Se le pasa un campo para verificar que cumple con el formato requerido. Valida los campos **idMueble**, **unidades existentes** y **precio unitario**.
- **validarCampoDescripcion:** Se le pasa el campo **descripción** para comprobar que cumple con el formato requerido.
- **conexion.leerMueble:** este método recupera un objeto con el **idMueble** del formulario muebles.
- **llenarMueble:** Da valor a los atributos del objeto global **Mueble m**.
- **conexion.insertarMueble:** método de la clase **conexionBD** al que se le pasa un objeto **mueble** para insertarlo en la base de datos.
- **limpiarFormularioMueble:** limpia los campos del formulario muebles.

También se comprueba (sin métodos) que el objeto **mueble** no esté ya registrado.

Si todo está bien se mostrará un mensaje de informando del registro.

- Insertar ventas



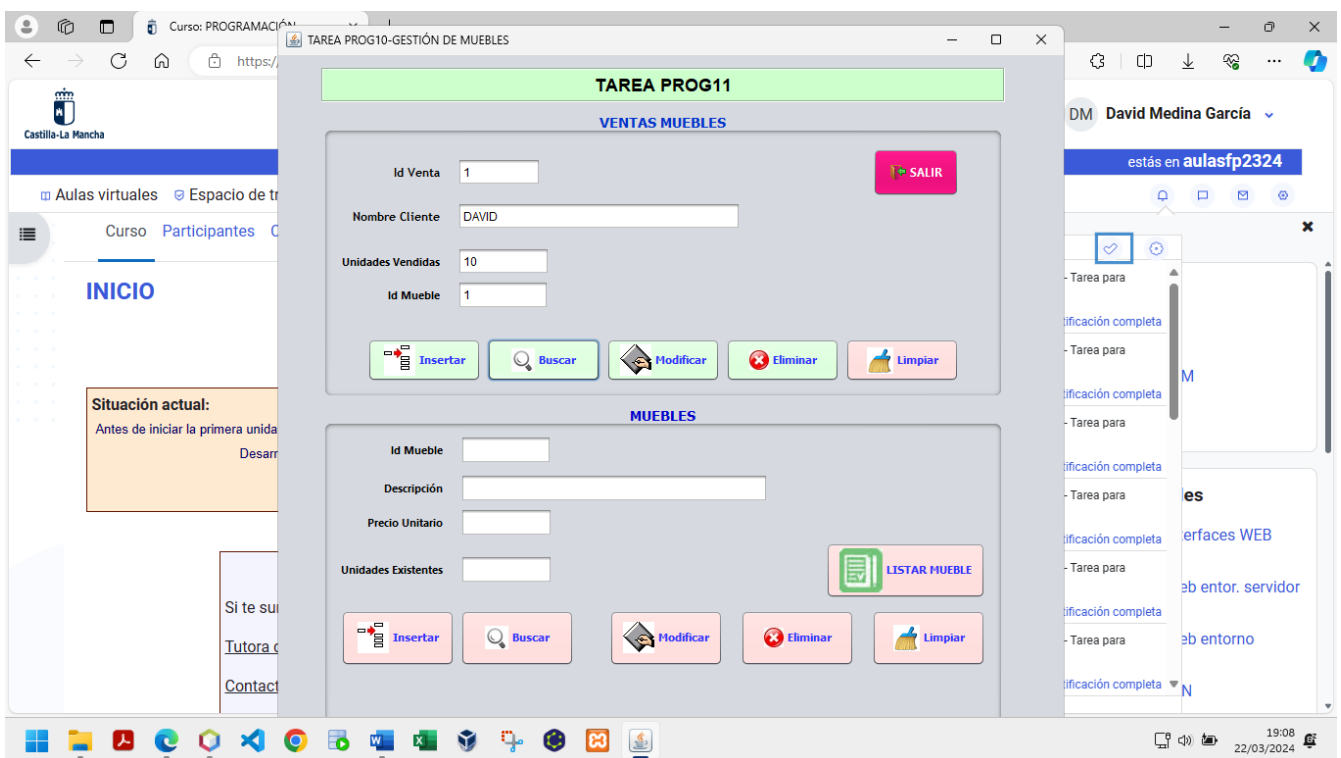
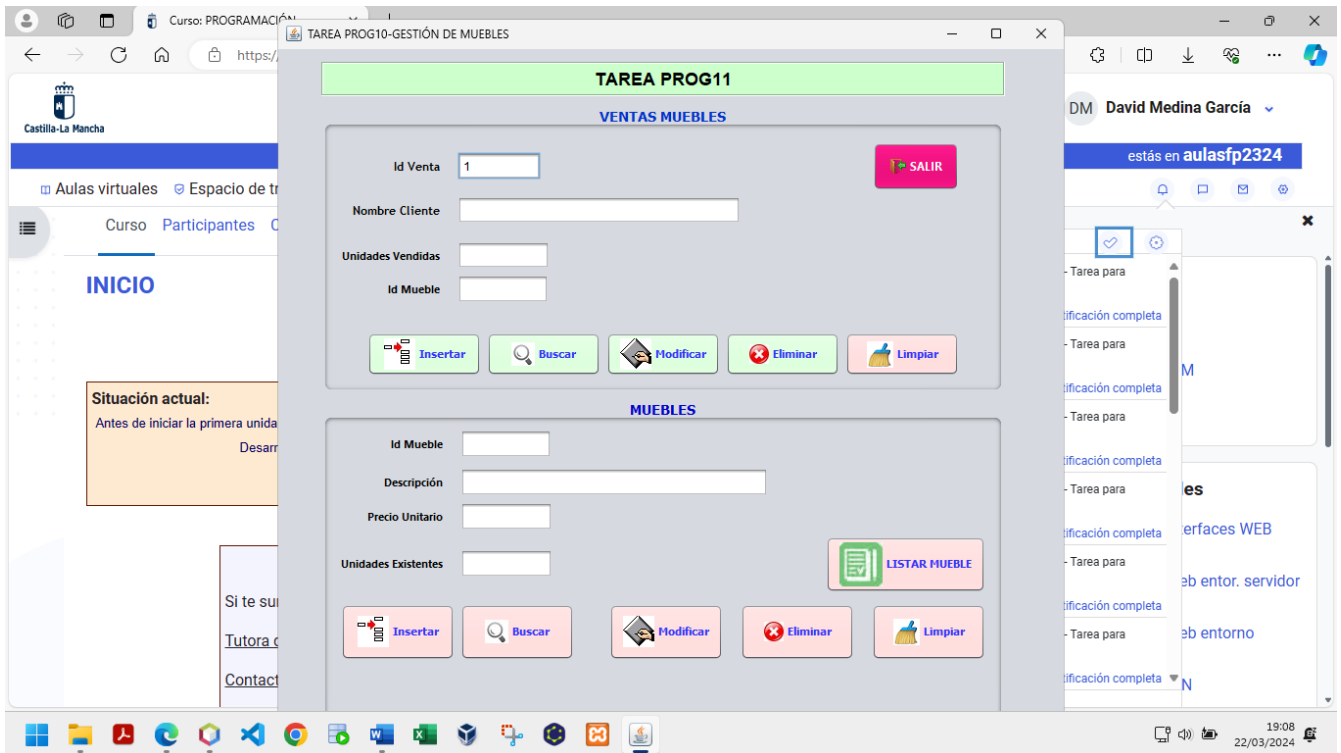
En este apartado se usarán los siguientes métodos:

- **verificarCamposVacios:** Se le pasan una serie de parámetros para comprobar que no estén vacíos.
- **validarCampos:** Se le pasa un campo para verificar que cumple con el formato requerido. Valida los campos **idMueble**, **unidades vendidas**.
- **validarCampoNombreCli:** Se le pasa el campo **nombre cliente** para comprobar que cumple con el formato requerido.
- **conexion.leerVenta:** este método recupera un objeto con el **idVenta** del formulario ventas.
- **llenarVenta:** Da valor a los atributos del objeto global **Venta v**.
- **conexion.insertarVenta:** método de la clase **conexionBD** al que se le pasa un objeto **venta** para insertarlo en la base de datos.
- **conexion.actualizarUnidadesExistentesDeVentaNueva:** A este método se le pasan las **unidades vendidas** y el **idMuebleVenta** para actualizar las **unidades existentes** del **mueble** en la base de datos.
- **limpiarFormularioVenta:** limpia los campos del formulario ventas.

También se comprueba (sin métodos) que el objeto **venta** no esté ya registrado, además de que el objeto **mueble** exista y que las **unidades vendidas** no sean mayores a las **unidades existentes**.

Si todo está bien se mostrará un mensaje de informando del registro.

- **Buscar venta**



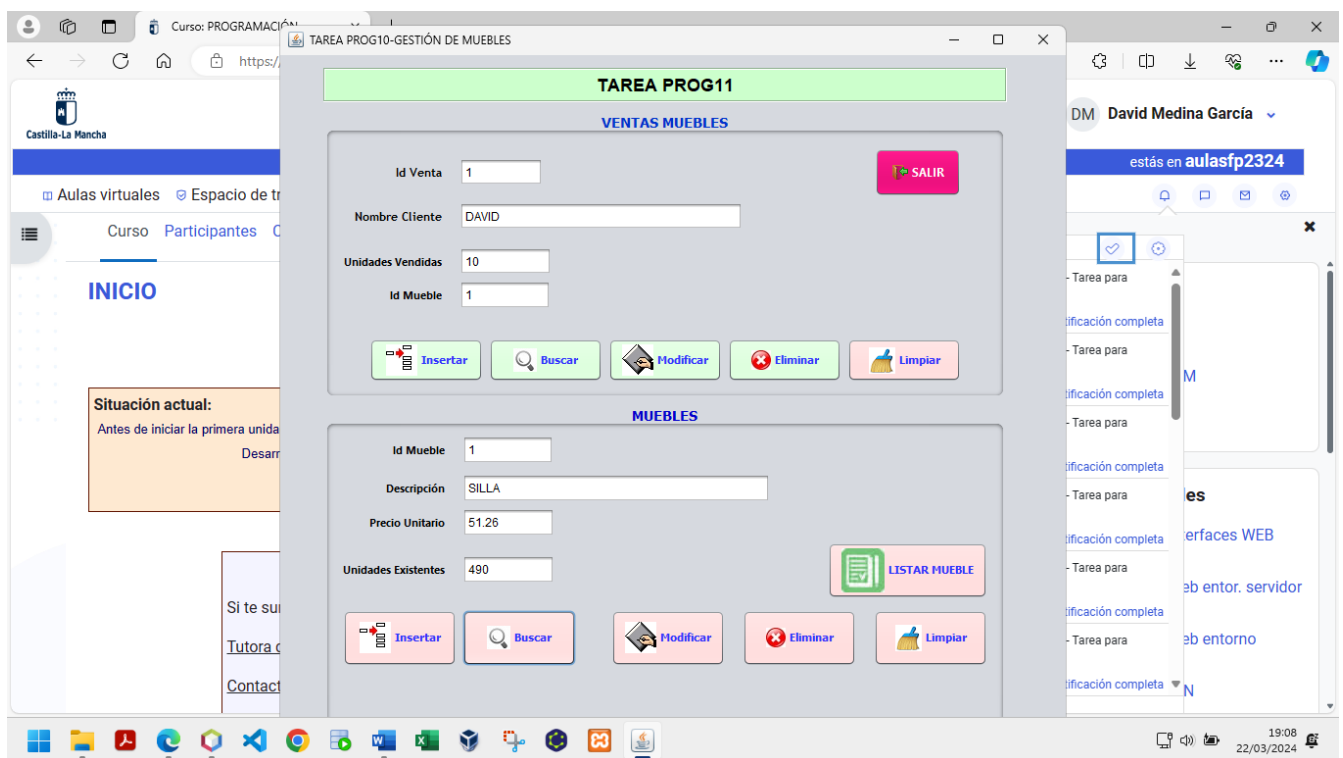
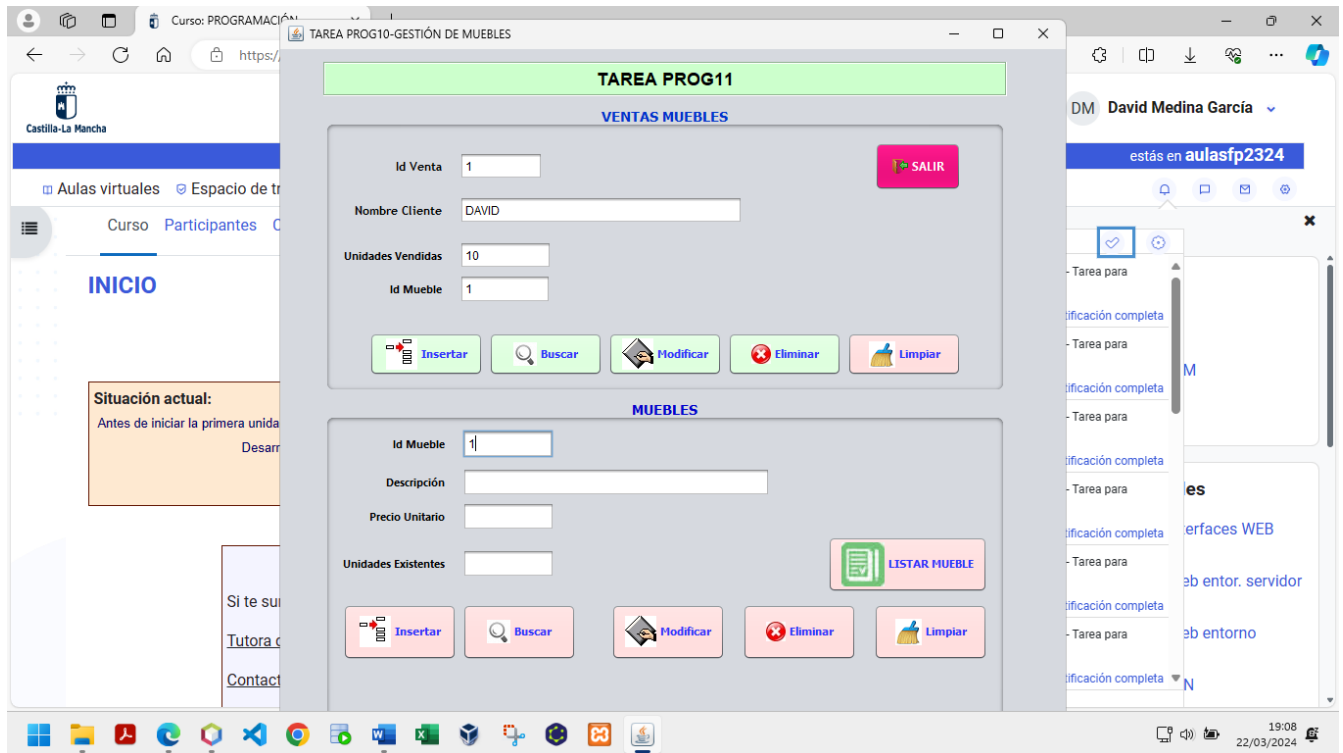
En este apartado se usarán los siguientes métodos:

- **verificarCamposVacios:** Se le pasan una serie de parámetros para comprobar que no estén vacíos.
- **conexion.leerVenta:** este método recupera un objeto con el **idVenta** del formulario ventas.
- **visualizarVenta:** Este método rellena los campos del formulario ventas con los datos del objeto **venta**.

También se comprueba (sin métodos) que el objeto **venta** este registrado.

\* Se guarda el objeto **mueble** para usar su **idMueble** después en el botón **modificar**.

- **Buscar mueble**

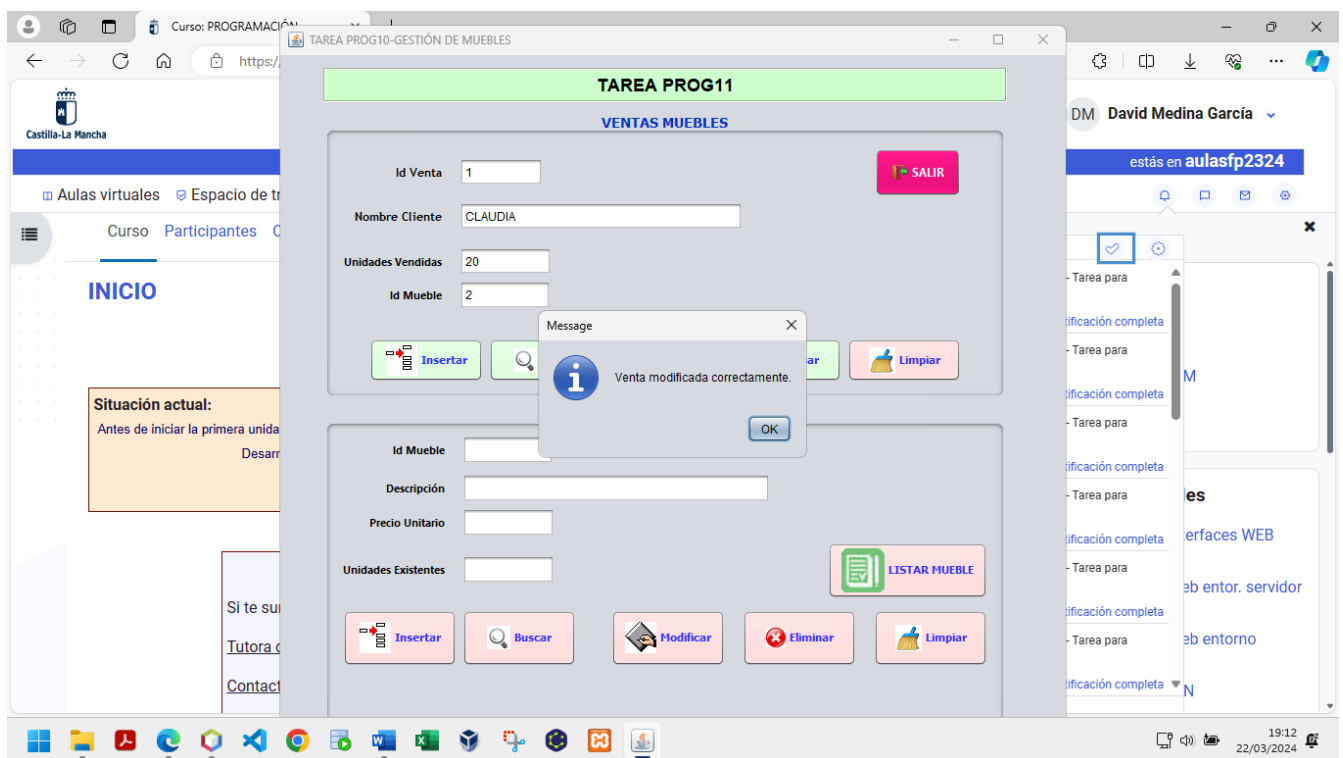
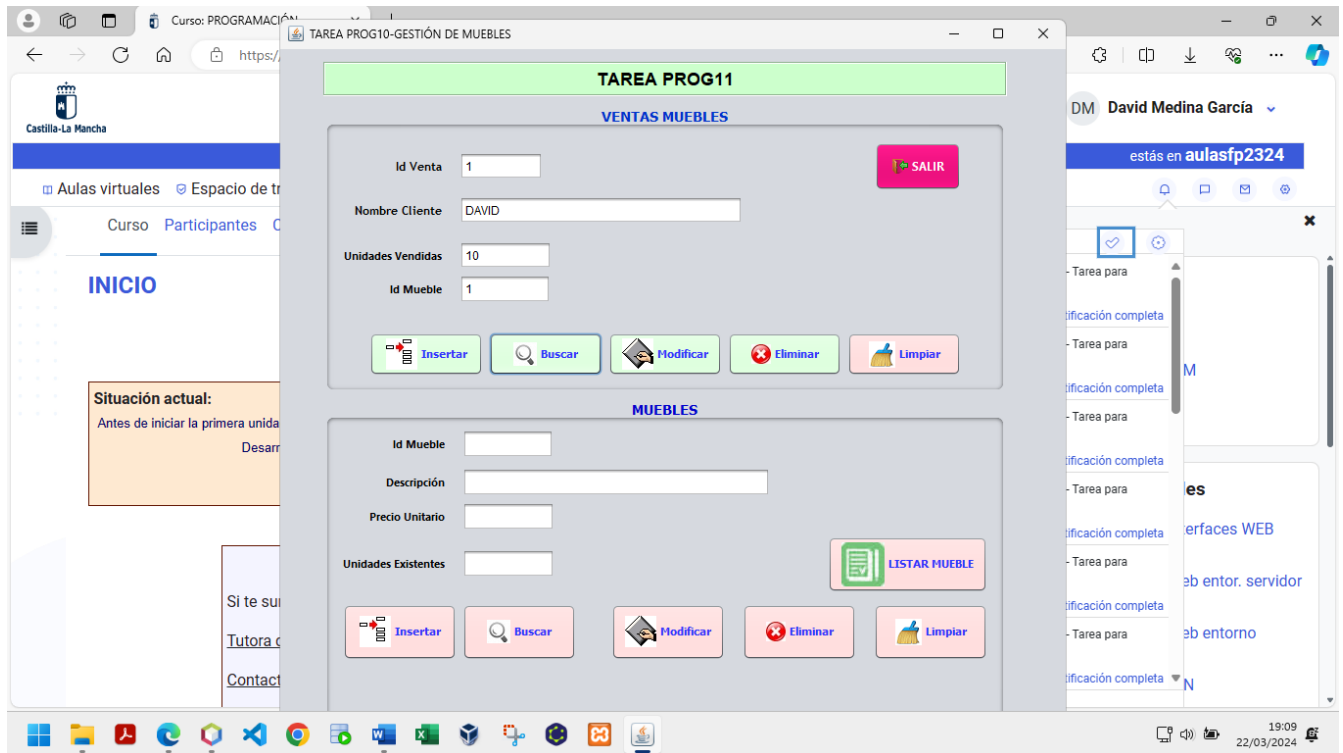


En este apartado se usarán los siguientes métodos:

- **verificarCamposVacios:** Se le pasan una serie de parámetros para comprobar que no estén vacíos.
- **conexion.leerMueble:** este método recupera un objeto con el **idMueble** del formulario muebles.
- **visualizarMueble:** Este método rellena los campos del formulario muebles con los datos del objeto **mueble**.

También se comprueba (sin métodos) que el objeto **mueble** este registrado.

- **Modificar venta**



\*Lo primero que se hace es recoger el **idMueble (idMuebleAnterior)** del objeto **mueble** que creamos al dar en **buscar**.

\*También recogemos las **unidadesVendidas (unidadesVendidasAnterior)** del objeto **venta** para usarlo si cambiamos el mueble de la venta.

Este apartado es el más complejo. Se usarán los siguientes métodos:

- **verificarCamposVacios:** Se le pasan una serie de parámetros para comprobar que no estén vacíos.

- **validarCampos:** Se le pasa un campo para verificar que cumple con el formato requerido. Valida los campos **idVenta**, **unidades vendidas** y **idMueble**.
- **validarCampoNombreCli:** Se le pasa el campo **nombre cliente** para comprobar que cumple con el formato requerido.
- **Conexión.leerUnidadesExistentes:** le pasamos el **idMueble** para almacenar sus **unidades existentes** en la base de datos.
- **conexion.leerVenta:** este método recupera un objeto con el **idVenta** del formulario ventas.
- **conexion.leerMueble:** este método recupera un objeto con el **idMueble** del formulario ventas.
- **conexion. actualizarUnidadesExistentesDeVentaRegistradaMismoMueble:** Actualiza las unidades existentes en la base de datos, si es el mismo **mueble** (mismo **idMueble**).
- **conexion. actualizarUnidadesExistentesDeVentaRegistradaDistintoMueble:** Actualiza las unidades existentes en la base de datos, si no es el mismo **mueble** (distinto **idMueble**).
- **conexion.modificarVenta:** Se le pasan los nuevos datos para modificar la venta en la base de datos.
- **limpiarFormularioVenta:** limpia los campos del formulario ventas.

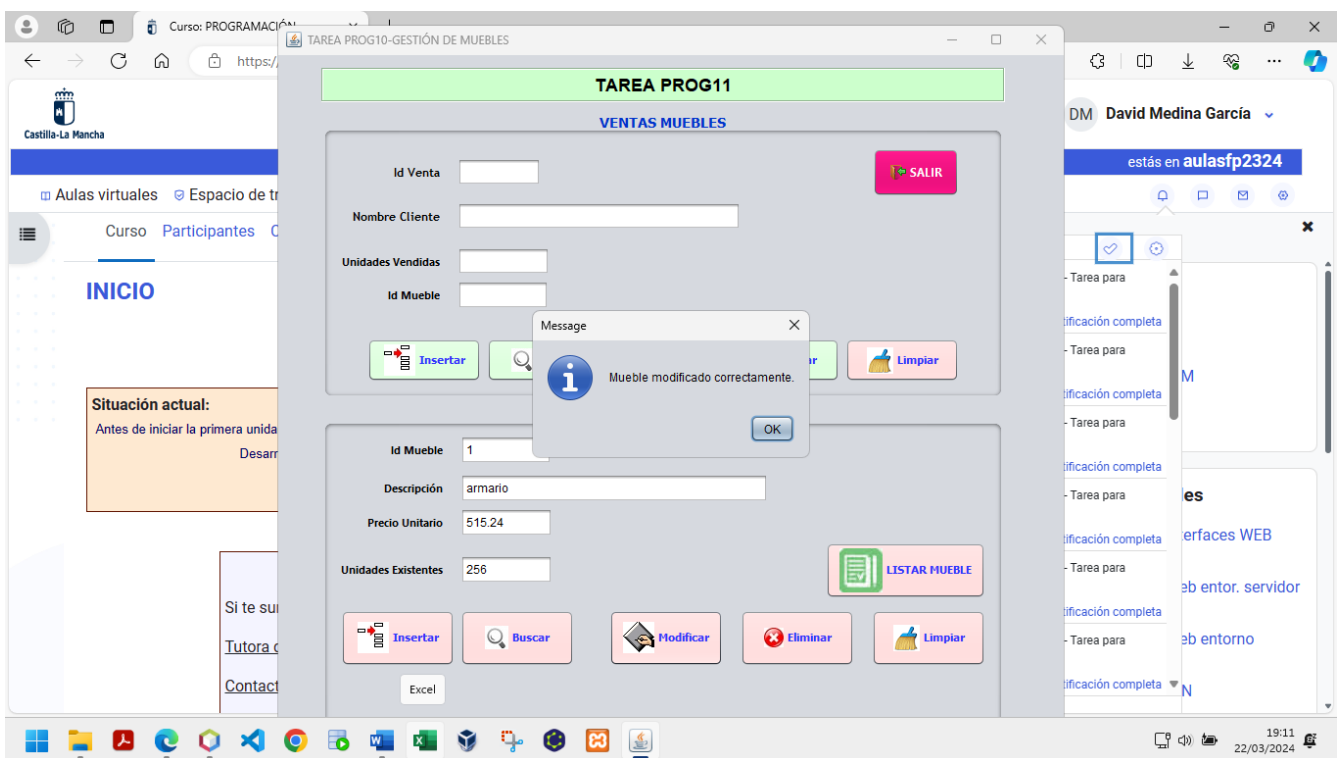
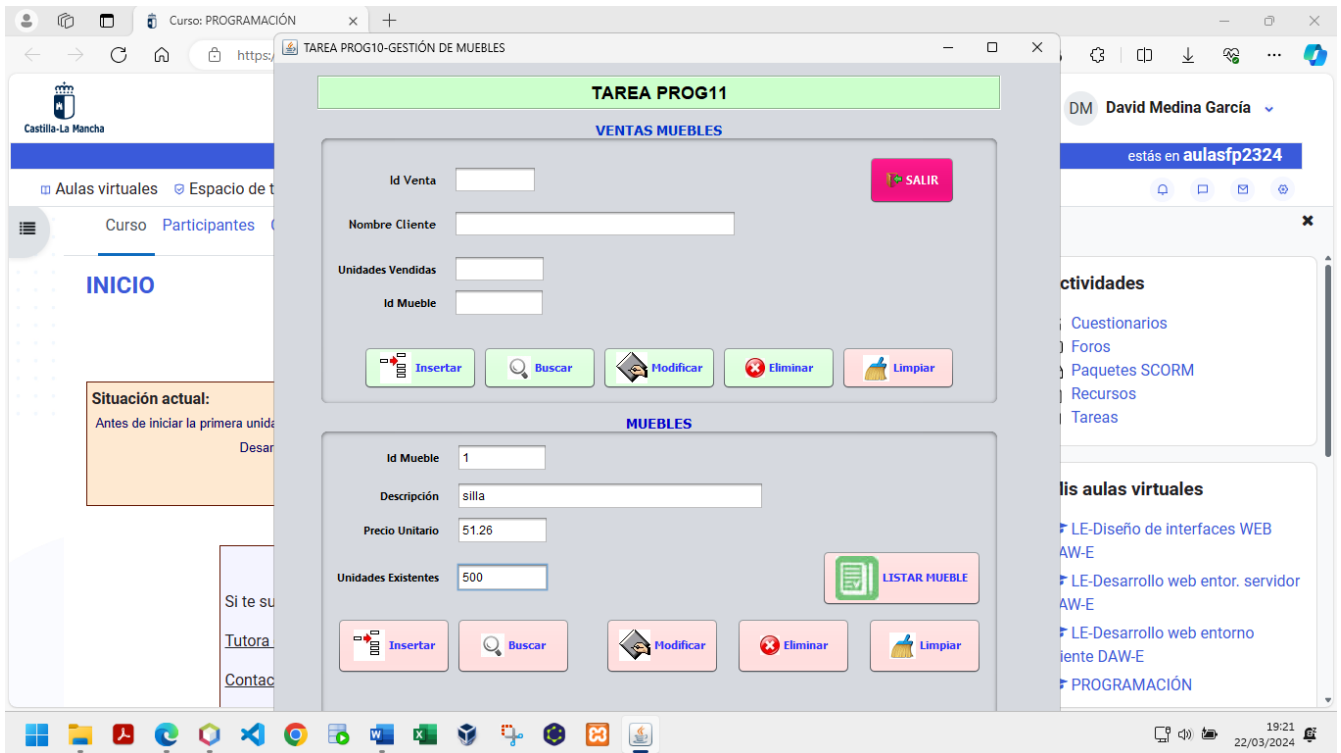
El método **actualizarUnidadesExistentesDeVentaRegistradaDistintoMueble** se usa cuando cambiamos el **idMueble** en la modificación de la venta (el **idMueble** que recogemos del objeto al empezar el método (**idMuebleAnterior**), es distinto del puesto en el formulario). Entonces significa que ha habido un cambio en el pedido, por tanto hay que recuperar las unidades vendidas del anterior mueble (**unidadesVendidasAnterior**).

Con este método se devuelven las unidades que no se venden(**unidadesVendidasAnterior**) al mueble anterior (**idMuebleAnterior**) y se restan las unidades del formulario al mueble que si se vende.

También se comprueba (sin métodos) que el objeto **venta** este registrado, además de que el objeto **mueble** exista y que las **unidades vendidas** no sean mayores a las **unidades existentes**.

Si todo está bien se mostrará un mensaje de informando de la modificación.

- **Modificar mueble**



En este apartado se usarán los siguientes métodos:

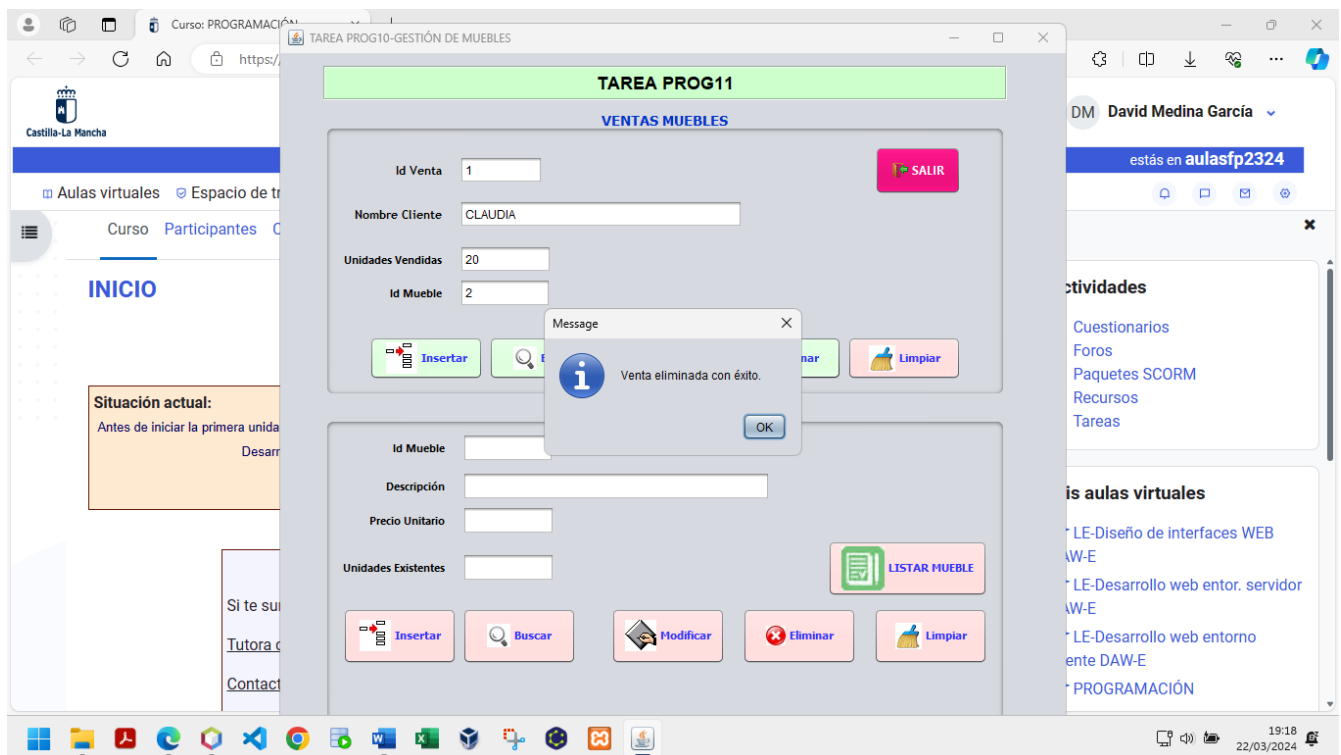
- **verificarCamposVacios:** Se le pasan una serie de parámetros para comprobar que no estén vacíos.
- **validarCampos:** Se le pasa un campo para verificar que cumple con el formato requerido. Valida los campos **idMueble**, **unidades existentes** y **precio unitario**.
- **validarCampoDescripcion:** Se le pasa el campo **descripción** para comprobar que cumple con el formato requerido.

- **conexion.leerMueble:** este método recupera un objeto con el **idMueble** del formulario muebles.
- **conexion.modificarMueble:** se le pasan los datos del formulario para modificar el mueble en la base de datos.
- **limpiarFormularioMueble:** limpia los campos del formulario muebles.

También se comprueba (sin métodos) que el objeto **mueble** este registrado.

Si todo está bien se mostrará un mensaje de informando de la modificación.

## • Eliminar venta



En este apartado se usarán los siguientes métodos:

- **verificarCamposVacios:** Se le pasan una serie de parámetros para comprobar que no estén vacíos.
- **conexion.leerVenta:** este método recupera un objeto con el **idVenta** del formulario ventas.
- **conexion.eliminarVenta:** Se le pasa el **idVenta** que se va a eliminar de la base de datos. (En este método también se actualizan las unidades existentes de los muebles en la base de datos).
- **limpiarFormularioVenta:** limpia los campos del formulario ventas.

También se comprueba (sin métodos) que el objeto **venta** este registrado.

Se pide una confirmación para eliminar la venta y, si todo está bien, se mostrará un mensaje de informando la eliminación.



Curso: PROGRAMACIÓN DE MUEBLES

TAREA PROG10-GESTIÓN DE MUEBLES

**TAREA PROG11**

VENTAS MUEBLES

David Medina García

estás en aulasfp2324

**Listado de Ventas**

ID VENTA	NOMBRE CLIENTE	DESCRIP. MUEBLE	PRECIO(\$)	UNIDADES VEND.	TOTAL VENTA(\$)
2	DAVID	ARMARIO	515.24	15	7728.6
3	MARIO	SILLA	45.65	30	1369.5

**TOTAL VENDIDO:** 9098,10\$

Insertar Buscar Modificar Eliminar Limpiar

Situación actual:  
Antes de iniciar la primera

Castilla-La Mancha

Aulas virtuales Espacio de trabajo

Curso Participantes

INICIO

Actividades

Cuestionarios  
Foros  
Paquetes SCORM  
Recursos  
Tareas

Las aulas virtuales

Diseño de interfaces WEB  
Desarrollo web entor. servidor  
LE-Desarrollo web entorno  
ente DAW-E  
PROGRAMACIÓN

19:18  
22/03/2024

- Eliminar mueble

Curso: PROGRAMACIÓN DE MUEBLES

TAREA PROG10-GESTIÓN DE MUEBLES

**TAREA PROG11**

VENTAS MUEBLES

David Medina García

estás en aulasfp2324

Id Venta

Nombre Cliente

Unidades Vendidas

Id Mueble

Message

El mueble con ID: 1, no puede ser borrado porque tiene ventas asociadas.

OK

Id Mueble 1

Descripción ARMARIO

Precio Unitario 515.24

Unidades Existentes 241

LISTAR MUEBLE

Insertar Buscar Modificar Eliminar Limpiar

Situación actual:  
Antes de iniciar la primera unidad

Desarrollo

Si te sur

Tutora

Contacto

Castilla-La Mancha

Aulas virtuales Espacio de trabajo

Curso Participantes

INICIO

Actividades

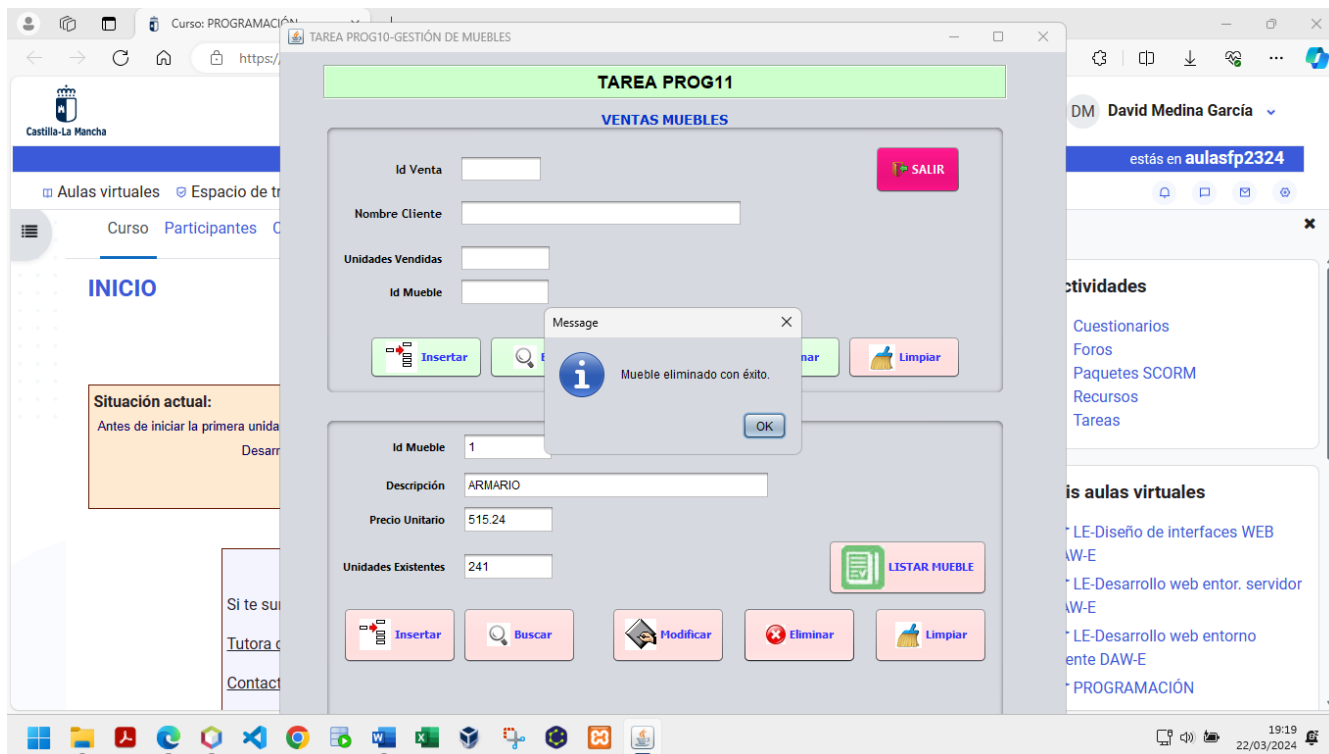
Cuestionarios  
Foros  
Paquetes SCORM  
Recursos  
Tareas

Las aulas virtuales

Diseño de interfaces WEB  
Desarrollo web entor. servidor  
LE-Desarrollo web entorno  
ente DAW-E  
PROGRAMACIÓN

19:19  
22/03/2024

\*Si hay algún registro de venta que tenga el mueble a eliminar, no se permite hacer la operación.



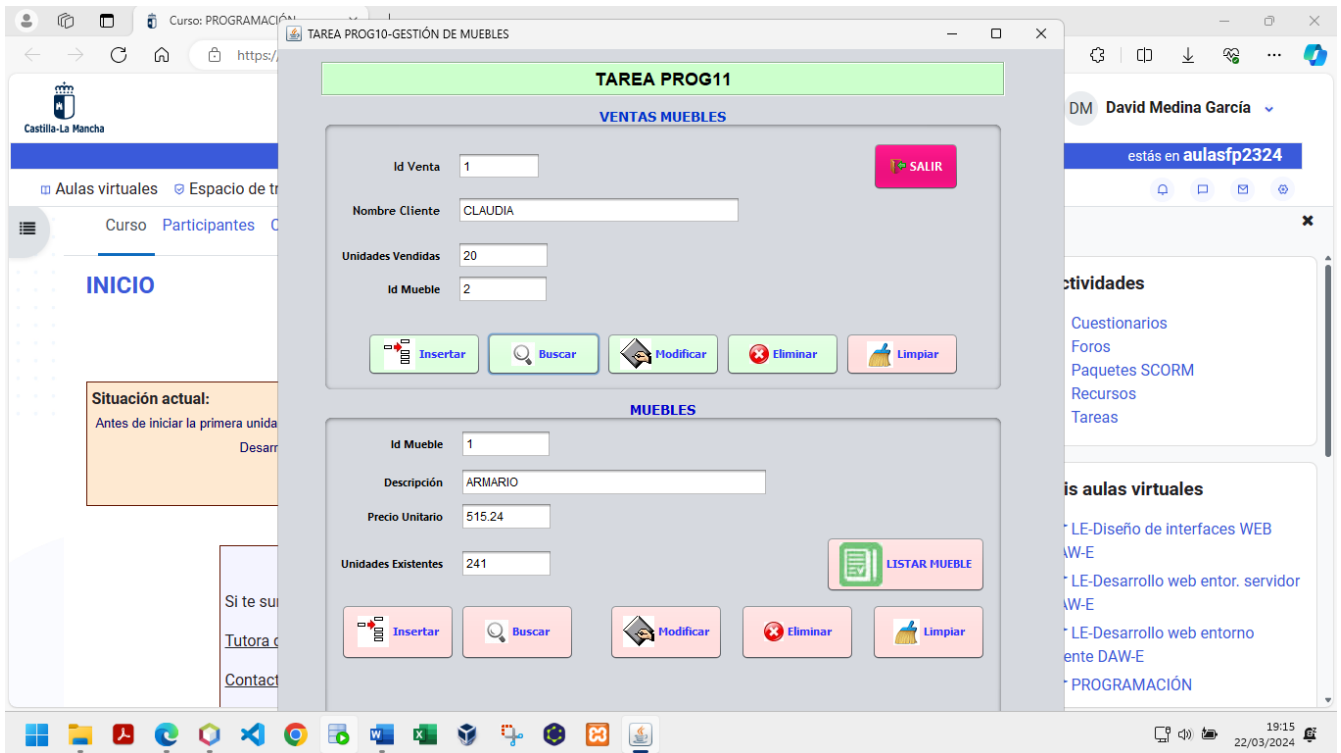
En este apartado se usarán los siguientes métodos:

- **verificarCamposVacios:** Se le pasan una serie de parámetros para comprobar que no estén vacíos.
- **conexion.leerMueble:** este método recupera un objeto con el **idMueble** del formulario muebles.
- **conexion.muebleEnRegistroVentas:** se le pasa el **idMueble** y comprueba que el **mueble** no tenga **ventas** asociadas.
- **conexion.eliminarMueble:** Se le pasa el **idMueble** que se va a eliminar de la base de datos.
- **limpiarFormularioMueble:** limpia los campos del formulario muebles.

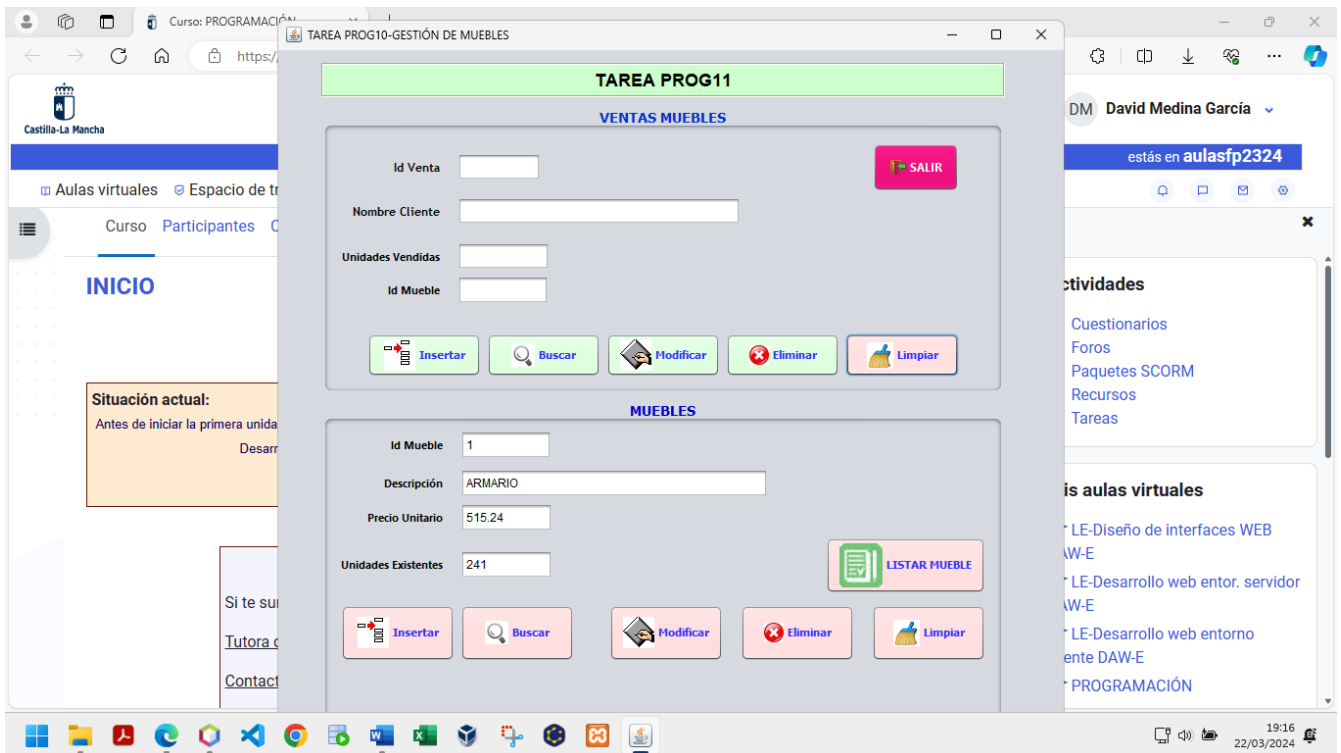
También se comprueba (sin métodos) que el objeto **mueble** este registrado.

Se pide una confirmación para eliminar el mueble y, si todo está bien, se mostrará un mensaje de informando la eliminación.

- Limpiar formularios

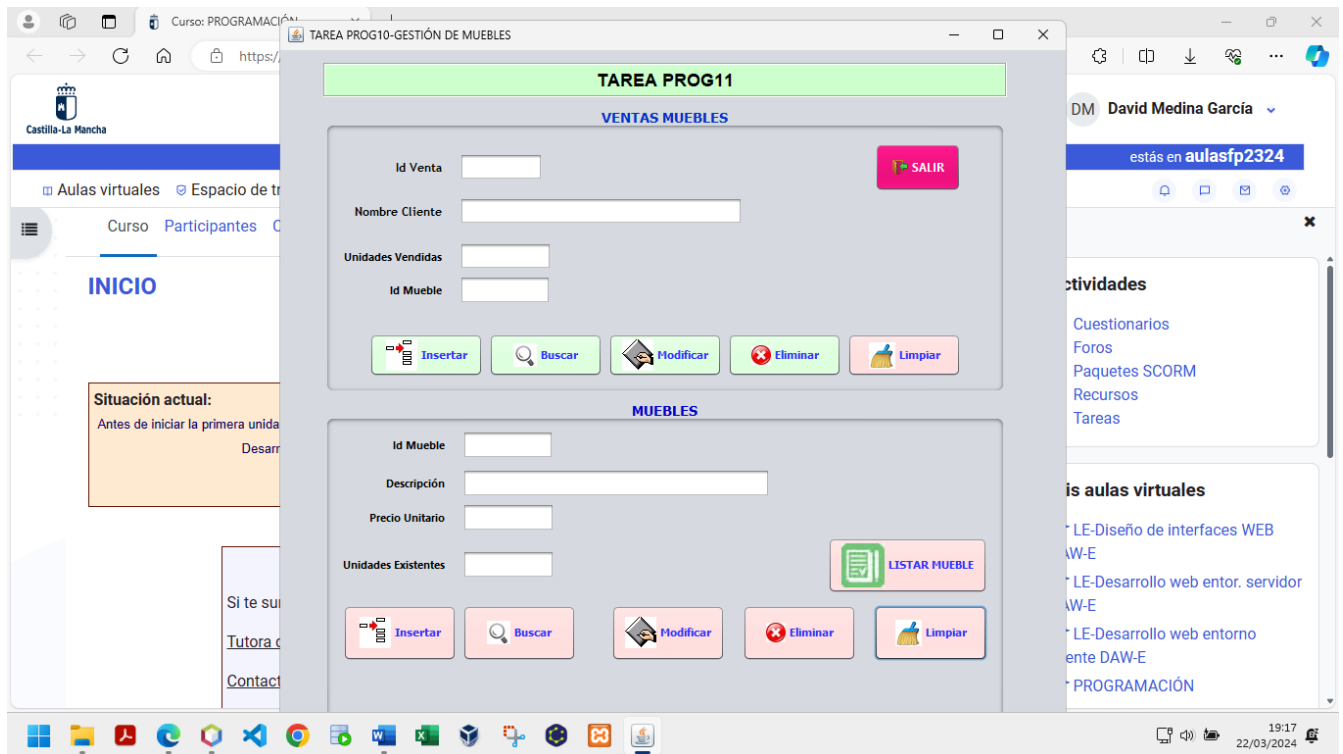


## VENTAS



Se usa el método `limpiarFormularioVenta` para dejar todos los campos de dicho formulario vacíos.

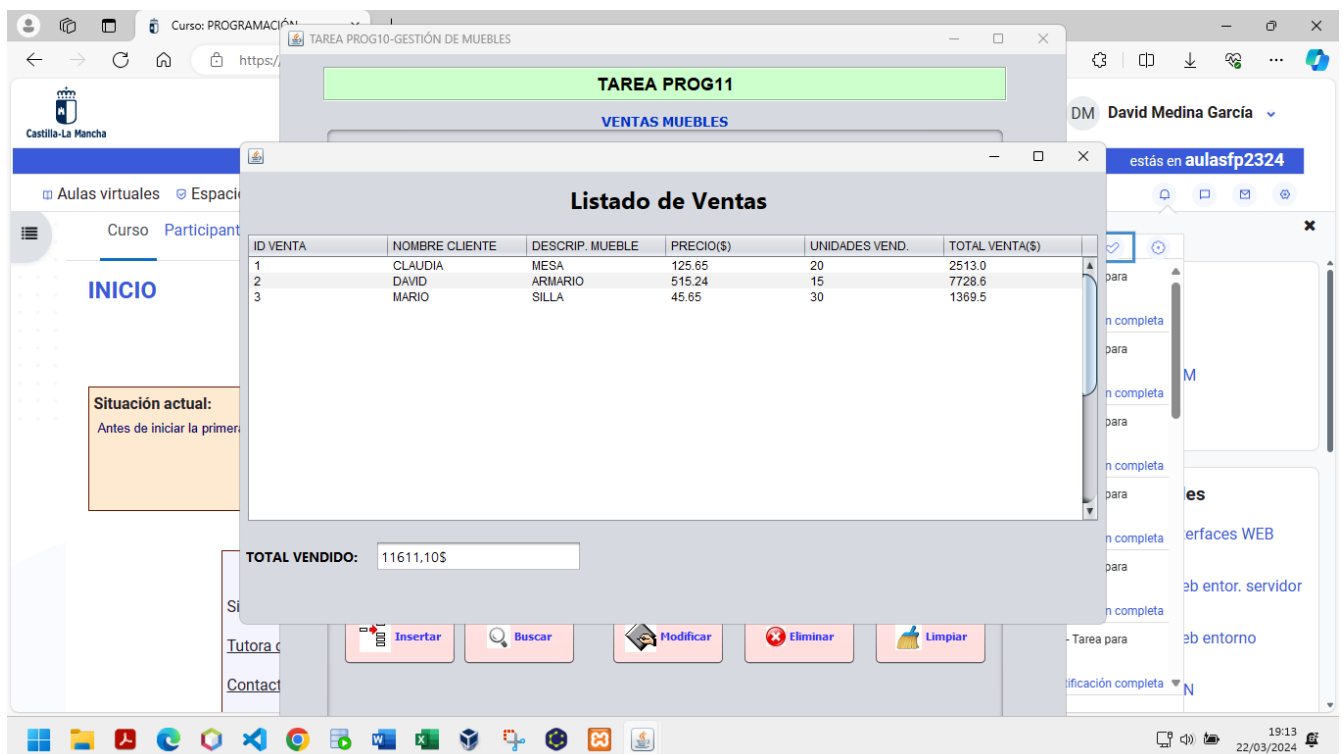
## MUEBLES



Se usa el método `limpiarFormularioMueble` para dejar todos los campos de dicho formulario vacíos.

- **Lista ventas (botón listar muebles)**

\*Antes de nada insertamos una serie de muebles y ventas pues los habíamos borrado antes.



The screenshot shows a web browser window at the top with the URL `https://aulasfp2324.castillalamancha.es/course/view.php?id=554#section-0`. The page header includes the 'educamosCLM' logo and the user name 'David Medina García'. Below the header, there are navigation links for 'Aulas virtuales', 'Espacio de trabajo', and 'Mis calificaciones'. The main content area shows a course titled 'Curso Participantes Calificaciones'.

Below the browser window, an IDE (IntelliJ IDEA) is open, showing a project named 'TAREA PROG11'. The IDE has several tabs open: 'AppGestionMueblesProg11David.java', 'ConexionBD.java', and 'Output - Tarea\_Prog11\_23\_24\_Apellido1Apellido2Nombre (run)'. The 'Output' tab is active, displaying the following text:

```

run:
Conexion Realizada
ID. VENTA  NOMBRE CLIENTE  DESCRIP. MUEBLE  PRECIO($)  UNIDADES VEND.  TOTAL VENTA($)
=====  =====
1          CLAUDIA           MESA             125,65      20              2513,00
2          DAVID             ARMARIO          515,24      15              7728,60
3          MARIO            SILLA            45,65       30              1369,50

TOTAL VENDIDO: 11611,10$

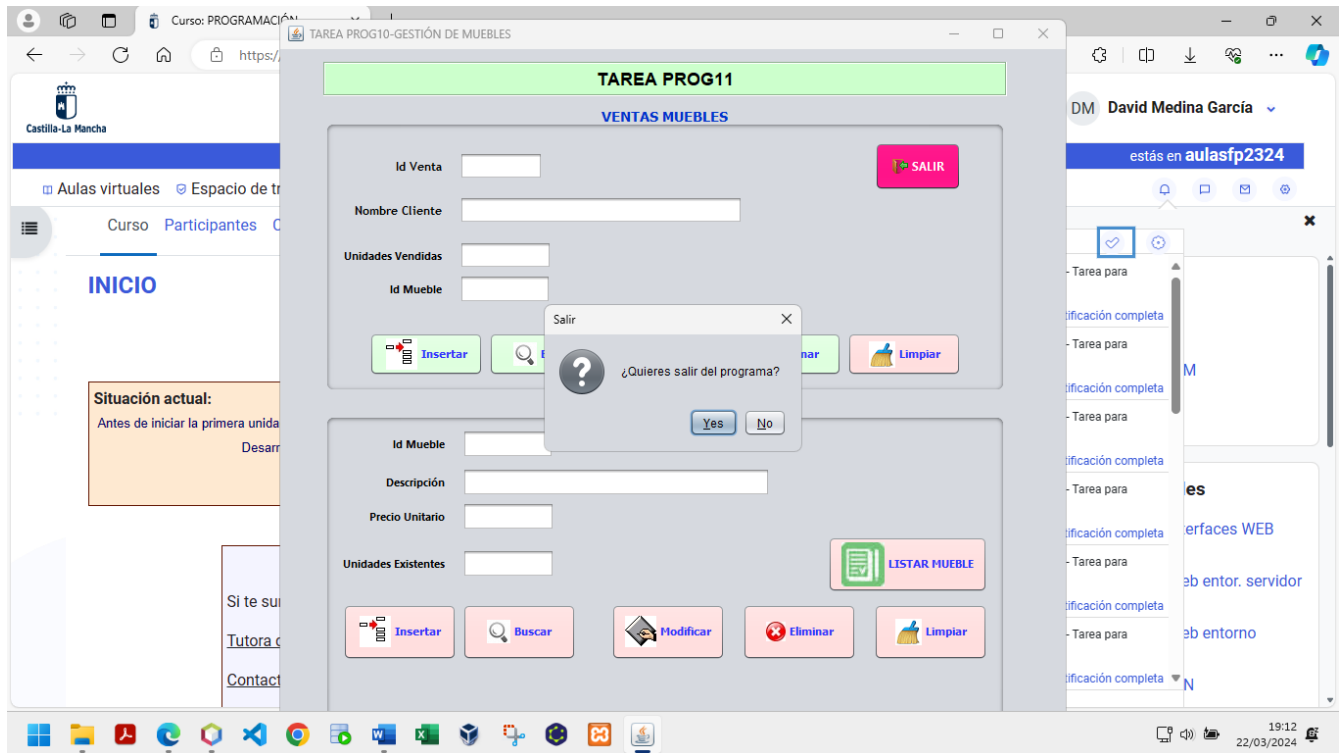
```

En este apartado se usarán los siguientes métodos:

- **limpiarTabla:** se le pasa una tabla para limpiarla.
- **Conexion.llenarTabla:** se le pasa una que llenara con los datos de las ventas de la base de datos.
- **Conexio.mostrarVentasConsola:** muestra los datos de las ventas de la base de datos por consola.
- **Conexio.totalVendido:** se calcula el total vendido de la base de datos.

Finalmente se muestra toda la información tanto en una tabla como por consola.

- **Salir**



Se solicita una confirmación antes de salir del programa.

Se cierra la conexión con el método **conexión.cerrar**.