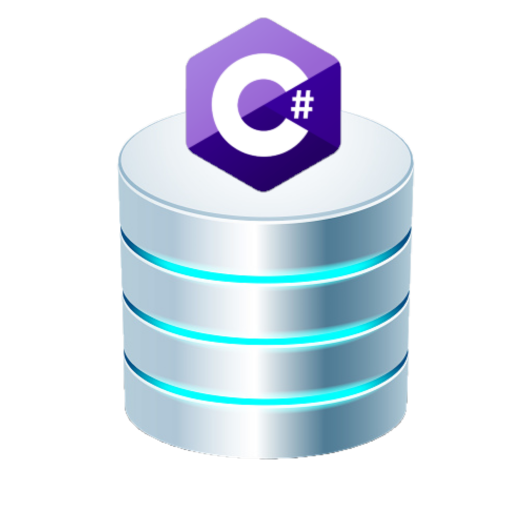
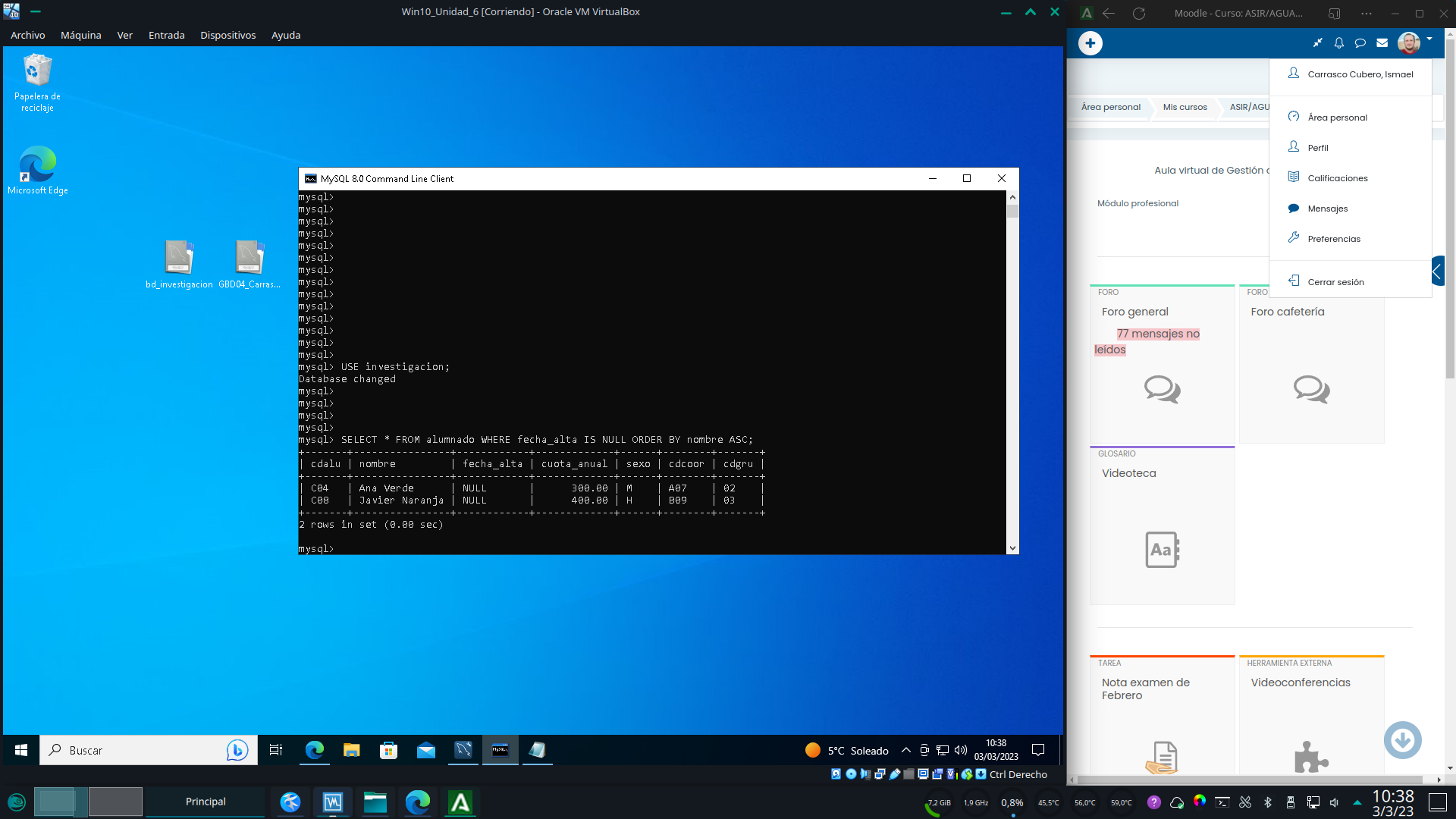
Gestión de base de datos



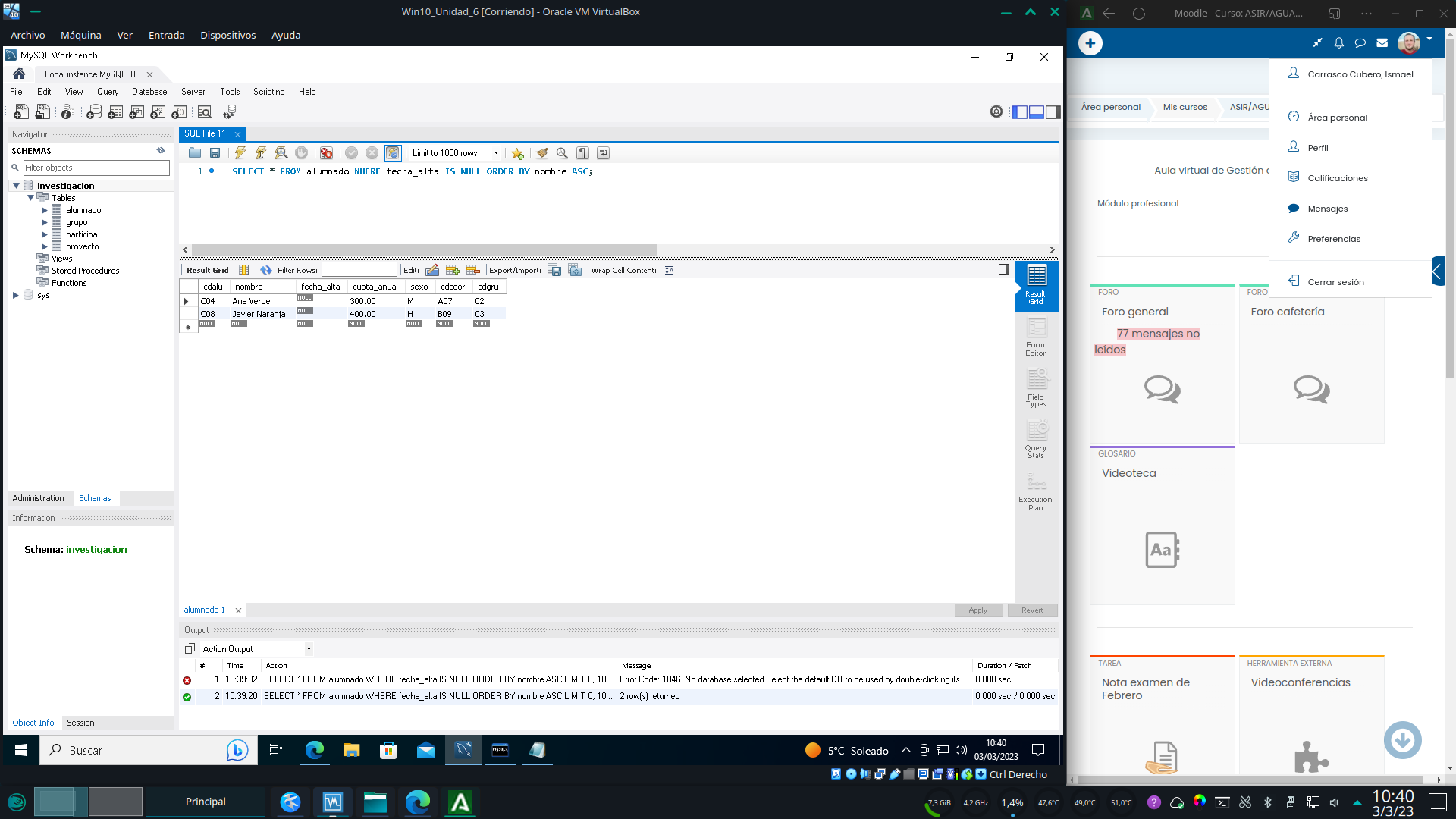
Tarea 4 Ismael Carrasco Cubero

1) Listado de todos los datos del alumnado cuya fecha de alta es desconocida (nula). Ordenar por nombre de alumnado.

SELECT \* FROM alumnado WHERE fecha\_alta IS NULL ORDER BY nombre ASC;



**2)** Listado con el código, nombre, año de alta (no la fecha) y cuota\_anual de las alumnas con cuota anual superior a 200 euros y año de alta entre el 2000 y 2015 (incluidos límites). Utiliza la función year() y ordenar de mayor a menor cuota. El encabezado debe ser: 'Código', 'Alumna', 'Año' y 'Cuota'. Debes hacer dos versiones:



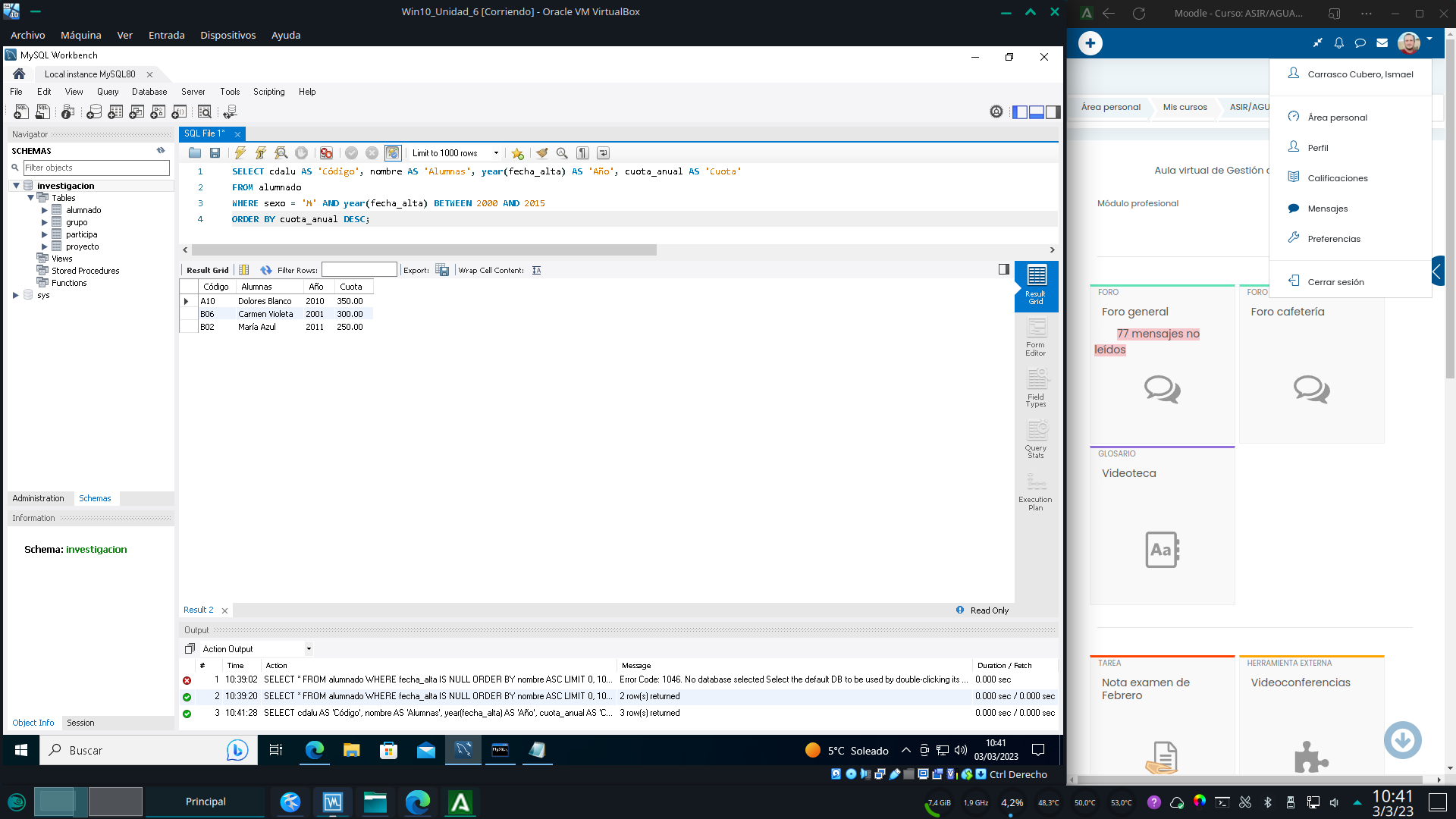
**a)** Usando BETWEEN donde sea posible.

SELECT cdalu AS 'Código', nombre AS 'Alumnas', year(fecha\_alta) AS 'Año', cuota\_anual AS 'Cuota'

FROM alumnado

WHERE sexo = 'M' AND year(fecha\_alta) BETWEEN 2000 AND 2015

ORDER BY cuota\_anual DESC;



**b)** Sin usar BETWEEN.

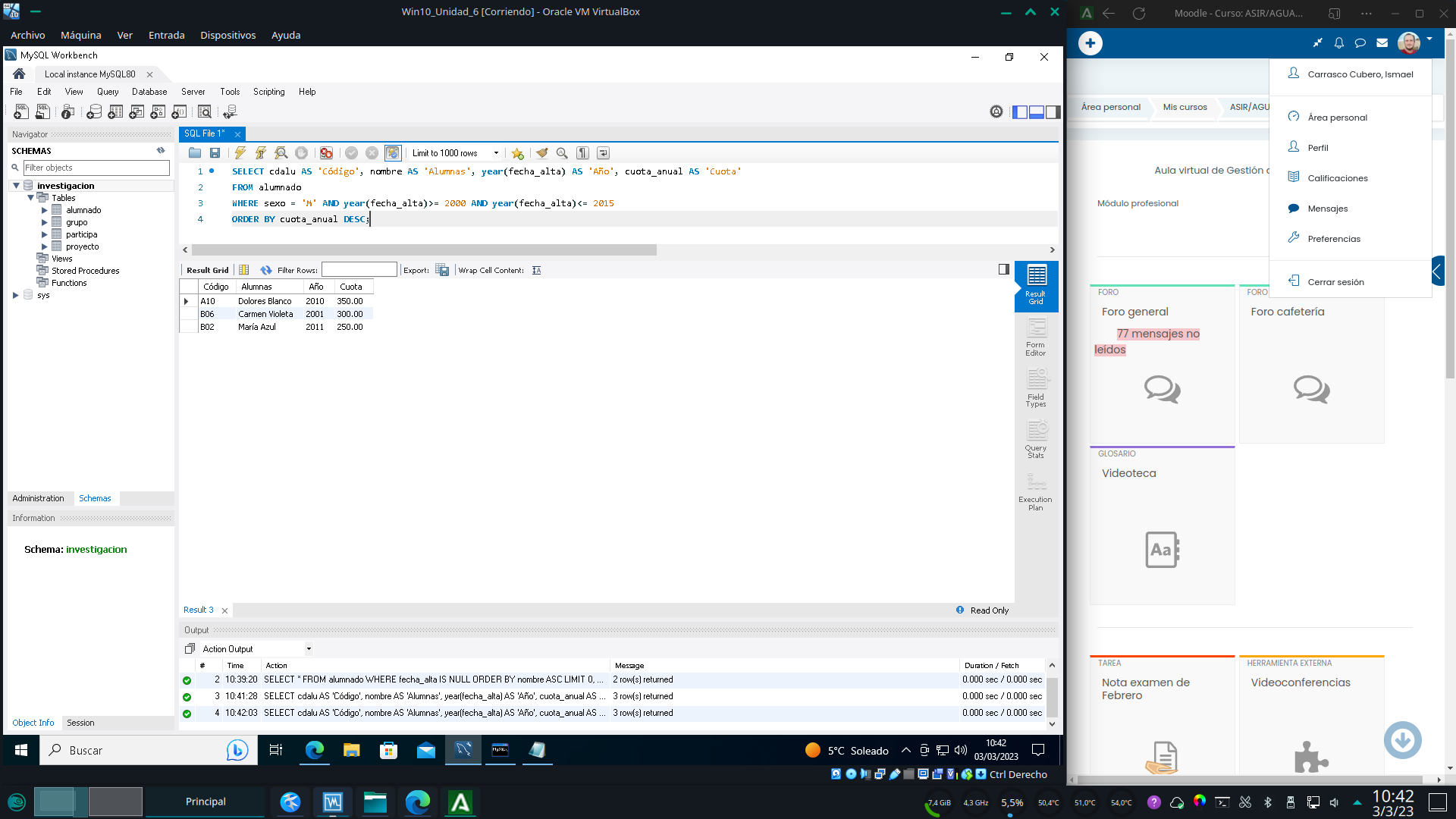
SELECT cdalu AS 'Código', nombre AS 'Alumnas', year(fecha\_alta) AS 'Año', cuota\_anual AS 'Cuota'

FROM alumnado

WHERE sexo = 'M' AND year(fecha\_alta)>= 2000 AND year(fecha\_alta)<= 2015

ORDER BY cuota\_anual DESC;

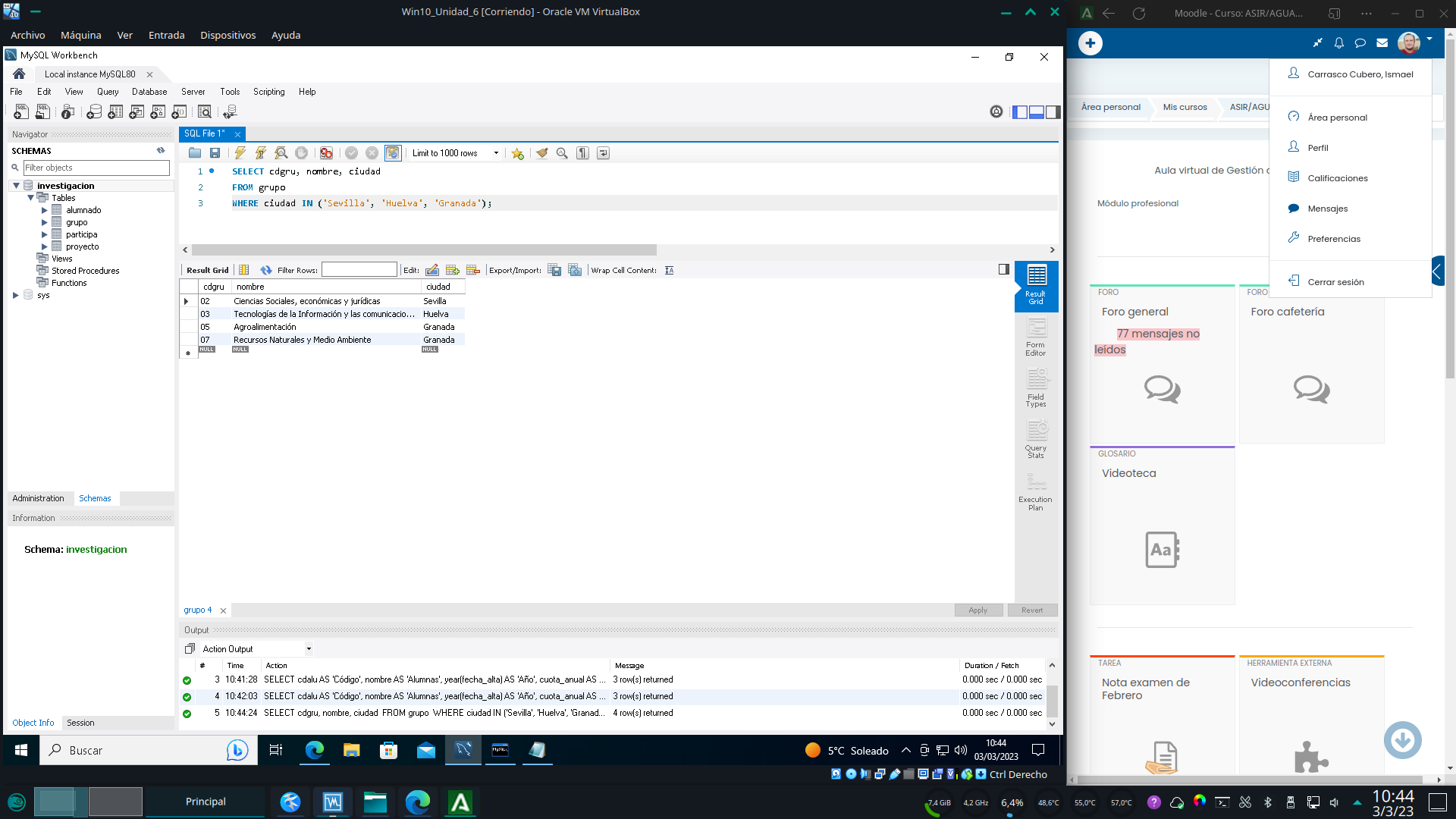
**3)** Listado con el código, nombre y ciudad de los grupos de investigación que están en Sevilla, Huelva o Granada. Debes redactarla usando dos versiones:



**a)** Con el operador IN().

SELECT cdgru, nombre, ciudad

FROM grupo WHERE ciudad IN ('Sevilla', 'Huelva', 'Granada');

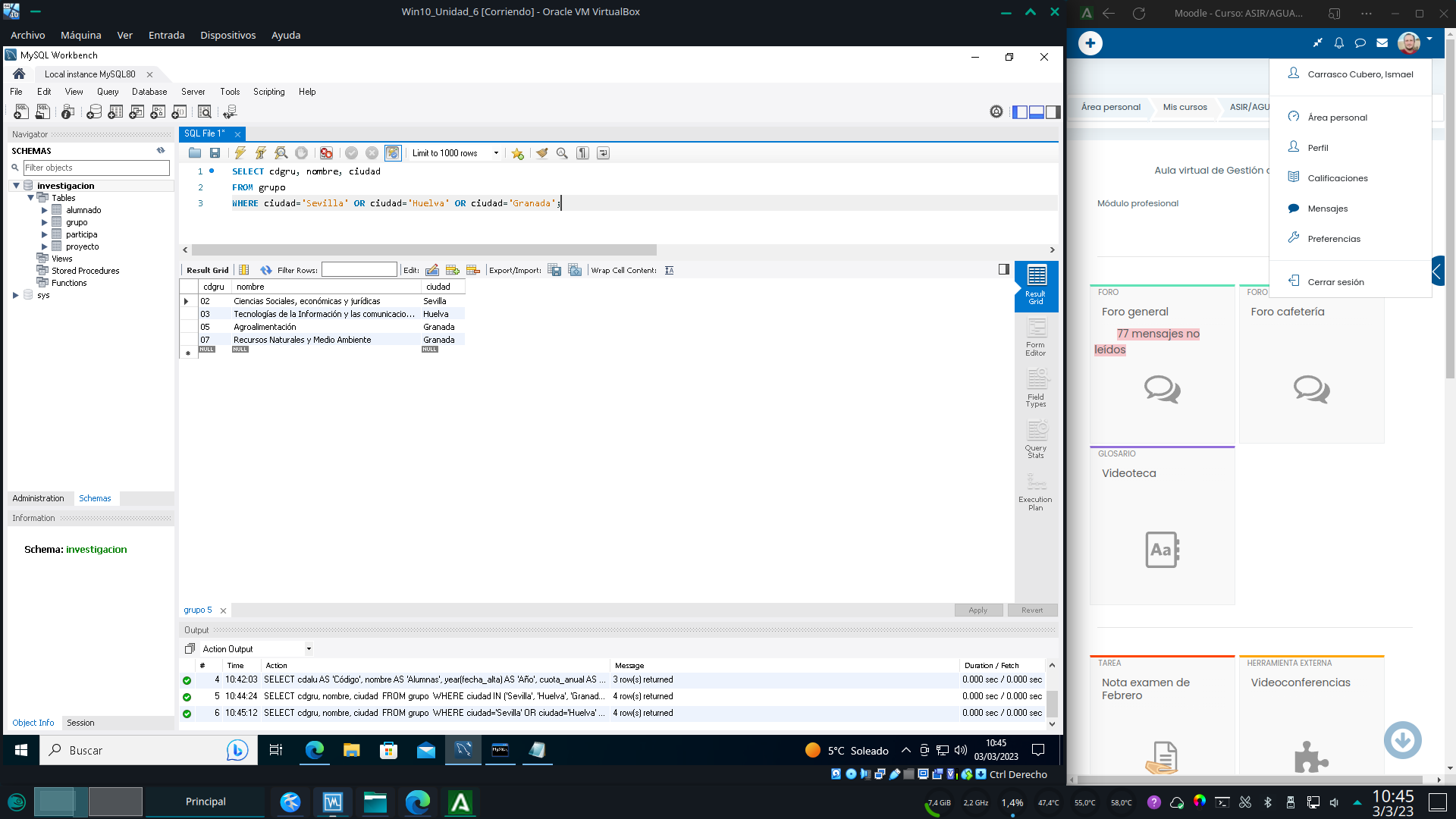


**b)** Sin IN(), con operadores lógicos y relacionales.

SELECT cdgru, nombre, ciudad

FROM grupo

WHERE ciudad='Sevilla' OR ciudad='Huelva' OR ciudad='Granada';



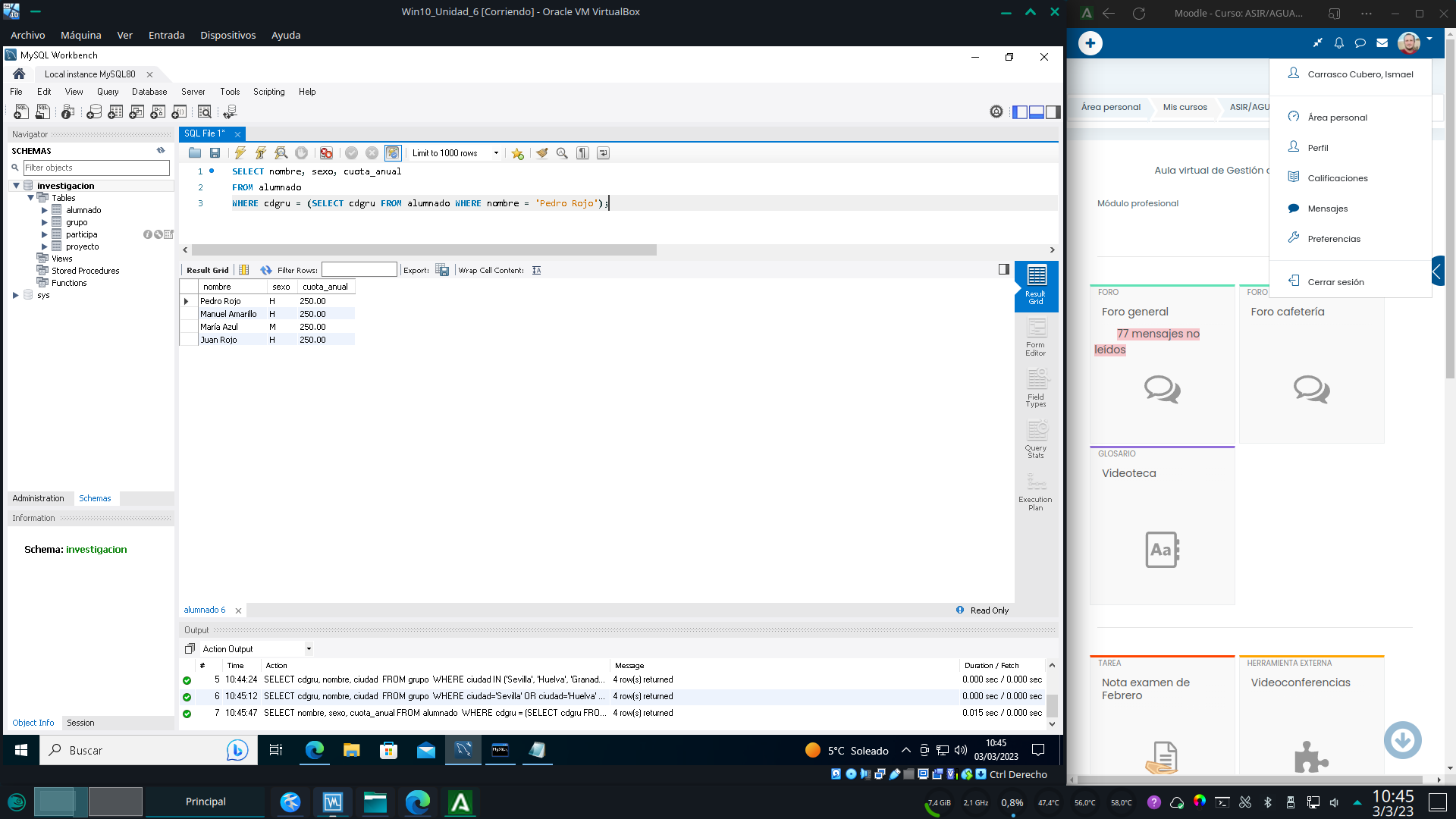
**4)** Listado con el nombre, sexo y cuota del alumnado del mismo grupo que el alumno 'Pedro Rojo'. Con subconsulta. Se debe codificar usando dos versiones:

**a)** Que aparezca el elemento (tupla o registro) utilizado para realizar la comparación.

SELECT nombre, sexo, cuota\_anual

FROM alumnado

WHERE cdgru = (SELECT cdgru FROM alumnado WHERE nombre = 'Pedro Rojo');



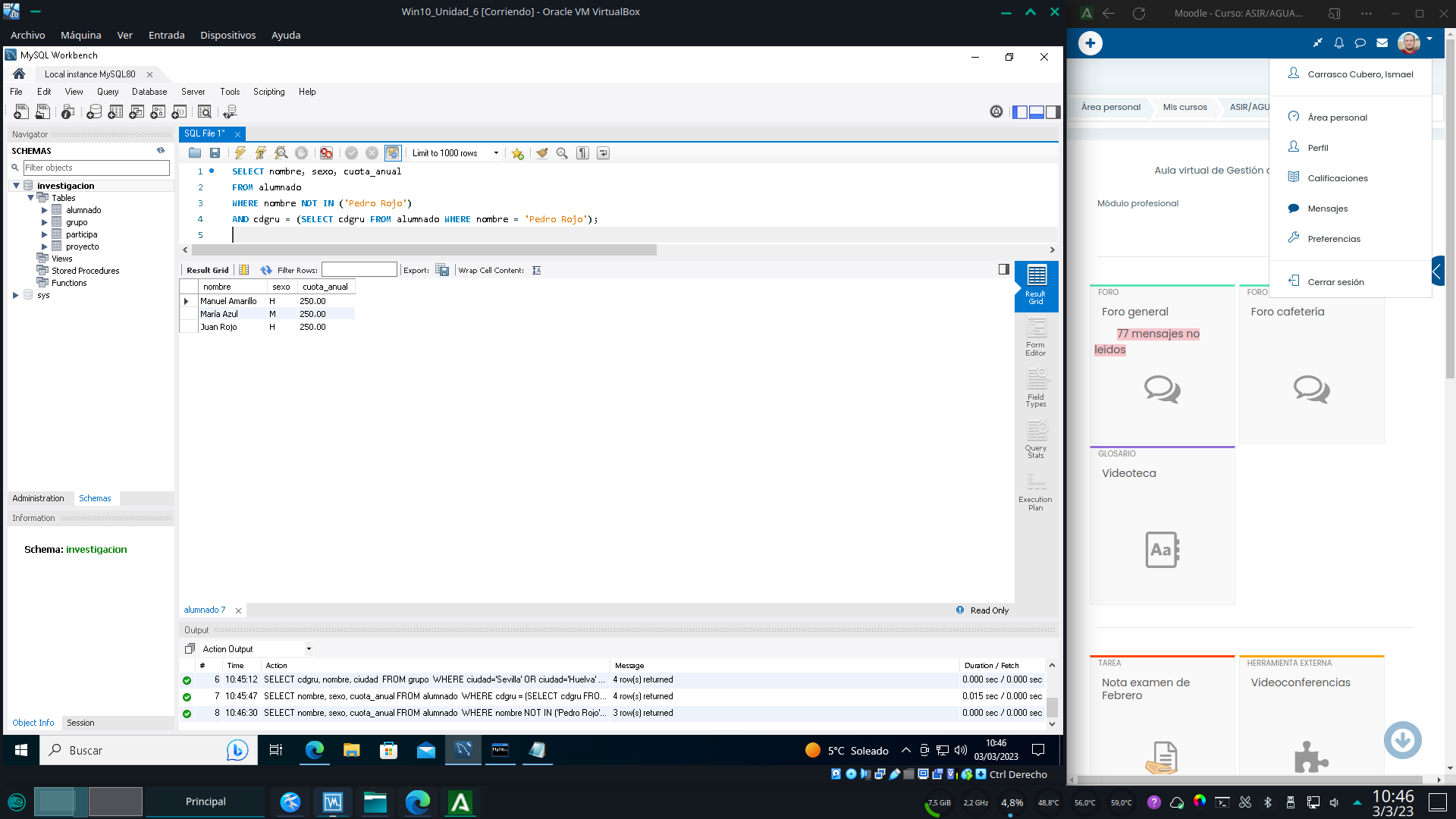
**b)** Que NO aparezca el elemento (tupla o registro) utilizado para realizar la comparación.

SELECT nombre, sexo, cuota\_anual

FROM alumnado

WHERE nombre NOT IN ('Pedro Rojo')

AND cdgru = (SELECT cdgru FROM alumnado WHERE nombre = 'Pedro Rojo');



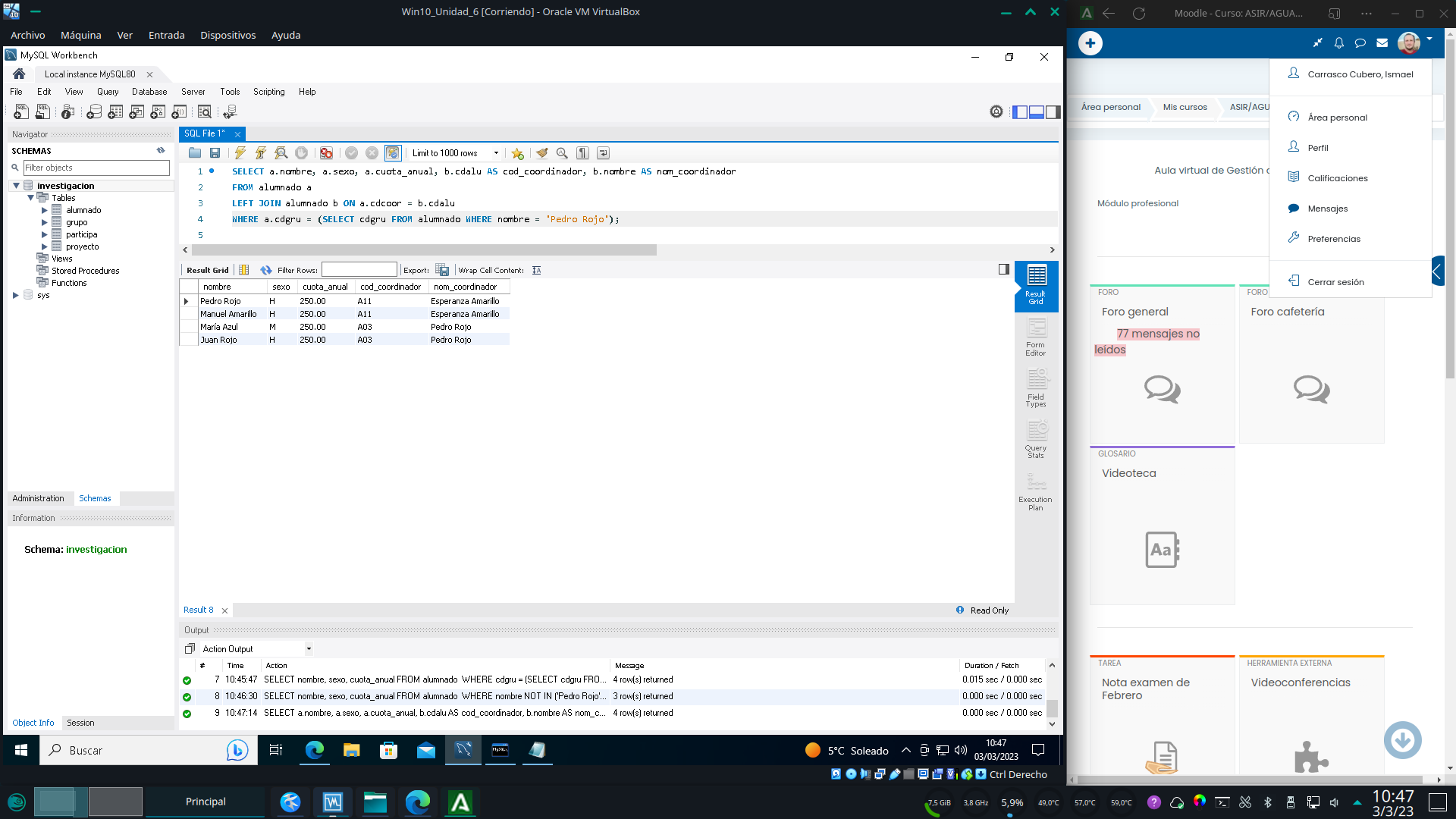
**5)** Modifica la consulta 4 para que también se muestre el código y nombre de quien coordina a esos alumnos y alumnas.

SELECT a.nombre, a.sexo, a.cuota\_anual, b.cdalu AS cod\_coordinador, b.nombre AS nom\_coordinador

FROM alumnado a

LEFT JOIN alumnado b ON a.cdcoor = b.cdalu

WHERE a.cdgru = (SELECT cdgru FROM alumnado WHERE nombre = 'Pedro Rojo');



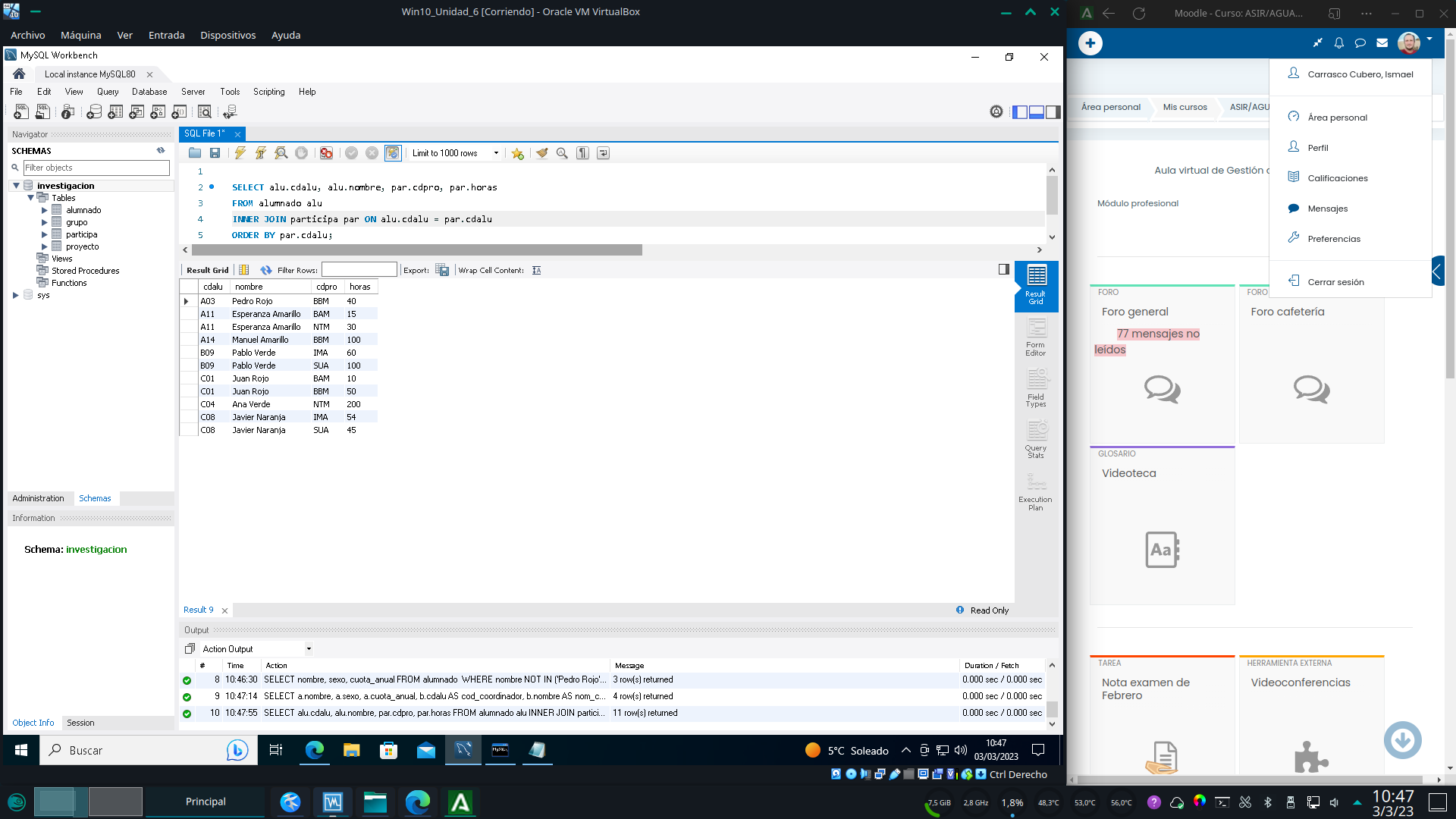
**6)** Código y nombre de los alumnos y alumnas y código del proyecto en el que está participando indicando las horas realizadas. Solo los alumnos y alumnas que participan en proyectos. El listado debe aparecer ordenado alfabéticamente por código de alumnado. (Con JOIN).

SELECT alu.cdalu, alu.nombre, par.cdpro, par.horas

FROM alumnado alu

INNER JOIN participa par ON alu.cdalu = par.cdalu

ORDER BY par.cdalu;



**7)** Lista donde aparezcan los nombres de todos los alumnos y alumnas, nombres de sus grupos y nombres de proyectos en los que participan, así como las horas invertidas en cada proyecto. Los alumnos y alumnas que no participan en proyectos aparecerán también en la lista y en lugar del proyecto aparecerá 'sin proyecto' y en horas un cero. El encabezado debe ser: 'Nombre alumno/alumna', 'Nombre grupo', 'Nombre proyecto' y 'Horas' (Con JOIN).

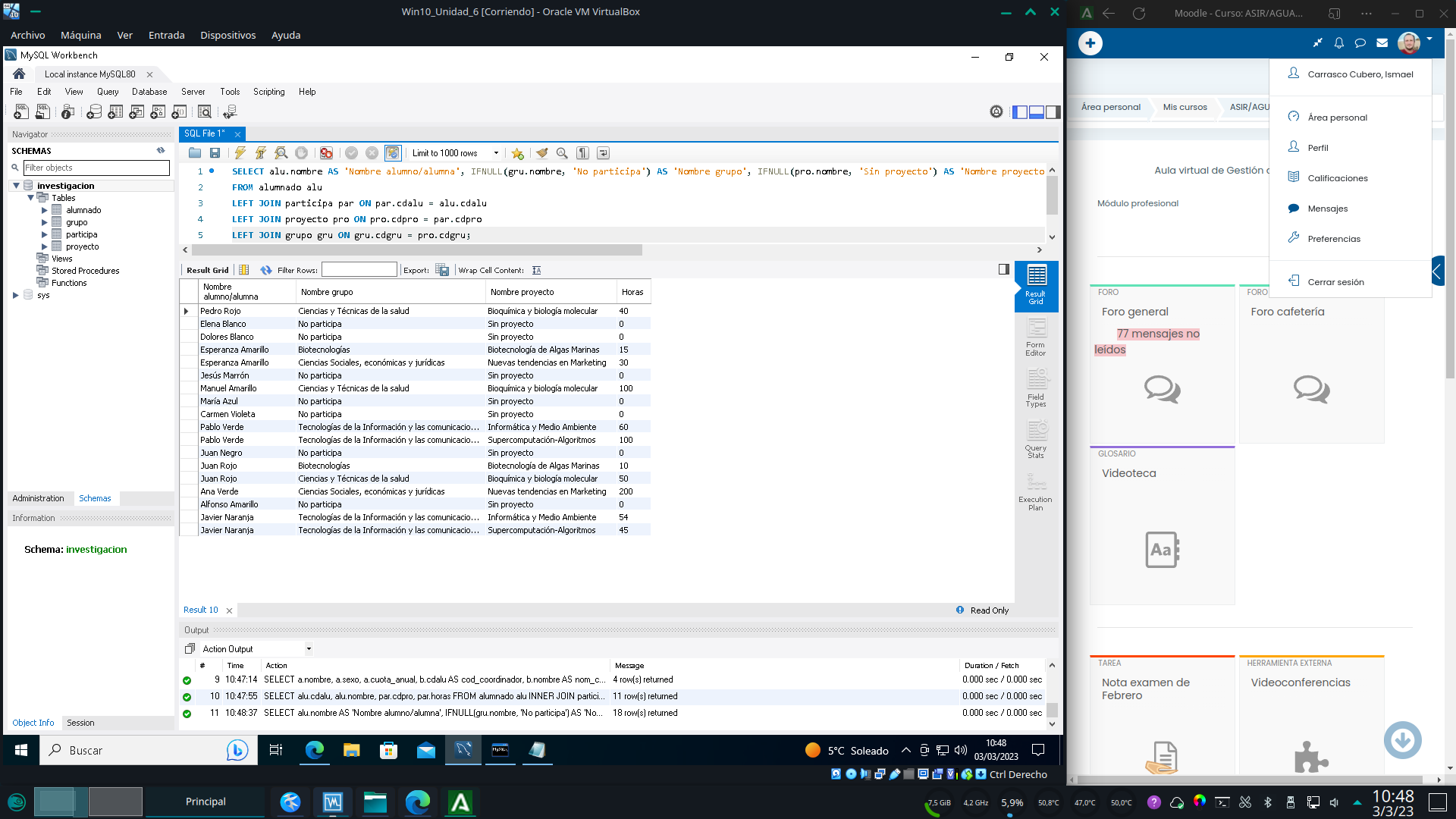
SELECT alu.nombre AS 'Nombre alumno/alumna', IFNULL(gru.nombre, 'No participa') AS 'Nombre grupo', IFNULL(pro.nombre, 'Sin proyecto') AS 'Nombre proyecto', IFNULL(par.horas, '0') AS 'Horas'

FROM alumnado alu

LEFT JOIN participa par ON par.cdalu = alu.cdalu

LEFT JOIN proyecto pro ON pro.cdpro = par.cdpro

LEFT JOIN grupo gru ON gru.cdgru = pro.cdgru;

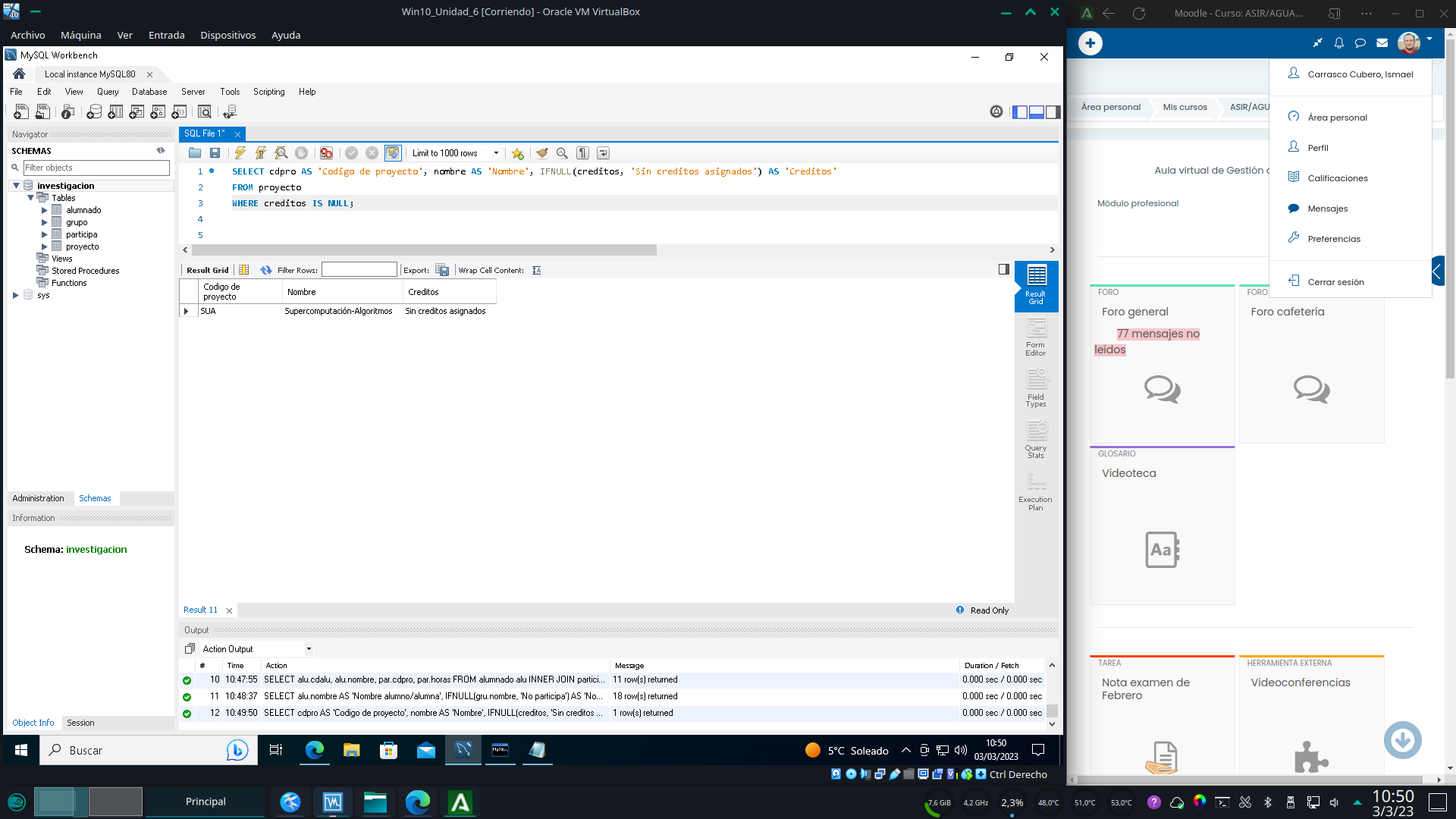


**8)** Listado con el código, nombre y créditos de los proyectos sin créditos asignados (nulos).

SELECT cdpro AS 'Codigo de proyecto', nombre AS 'Nombre', IFNULL(creditos, 'Sin creditos asignados') AS 'Creditos'

FROM proyecto

WHERE creditos IS NULL;



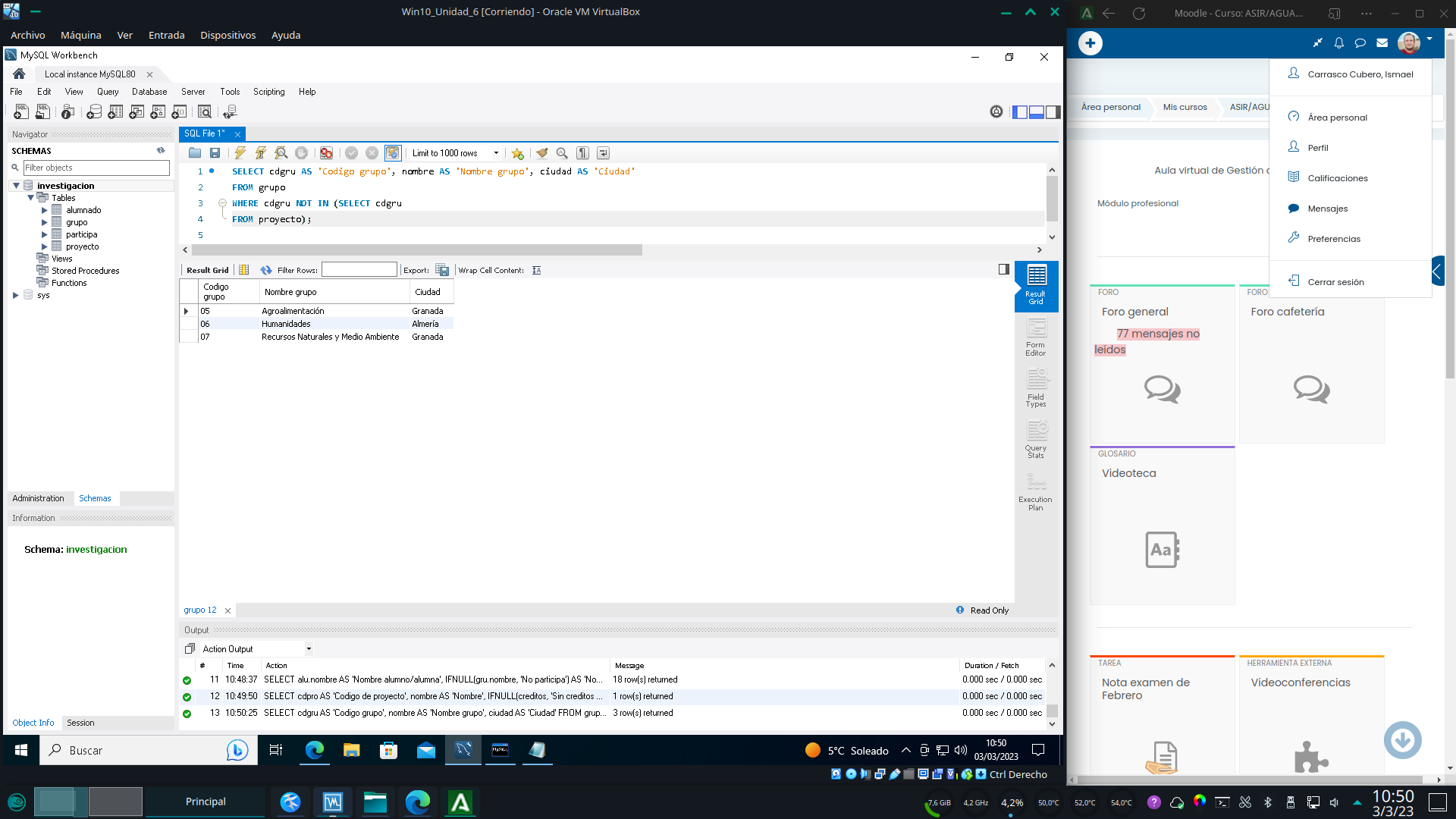
**9)** Listado con todos los datos de los grupos de investigación sin proyectos. (Con subconsulta).

SELECT cdgru AS 'Codigo grupo', nombre AS 'Nombre grupo', ciudad AS 'Ciudad'

FROM grupo

WHERE cdgru NOT IN (SELECT cdgru

FROM proyecto);



**10)** Para cada grupo de investigación su código, su nombre y el año de alta más reciente y más antiguo de su alumnado. Solo el año, no la fecha completa. El resultado debe estar ordenando por nombre de grupo.

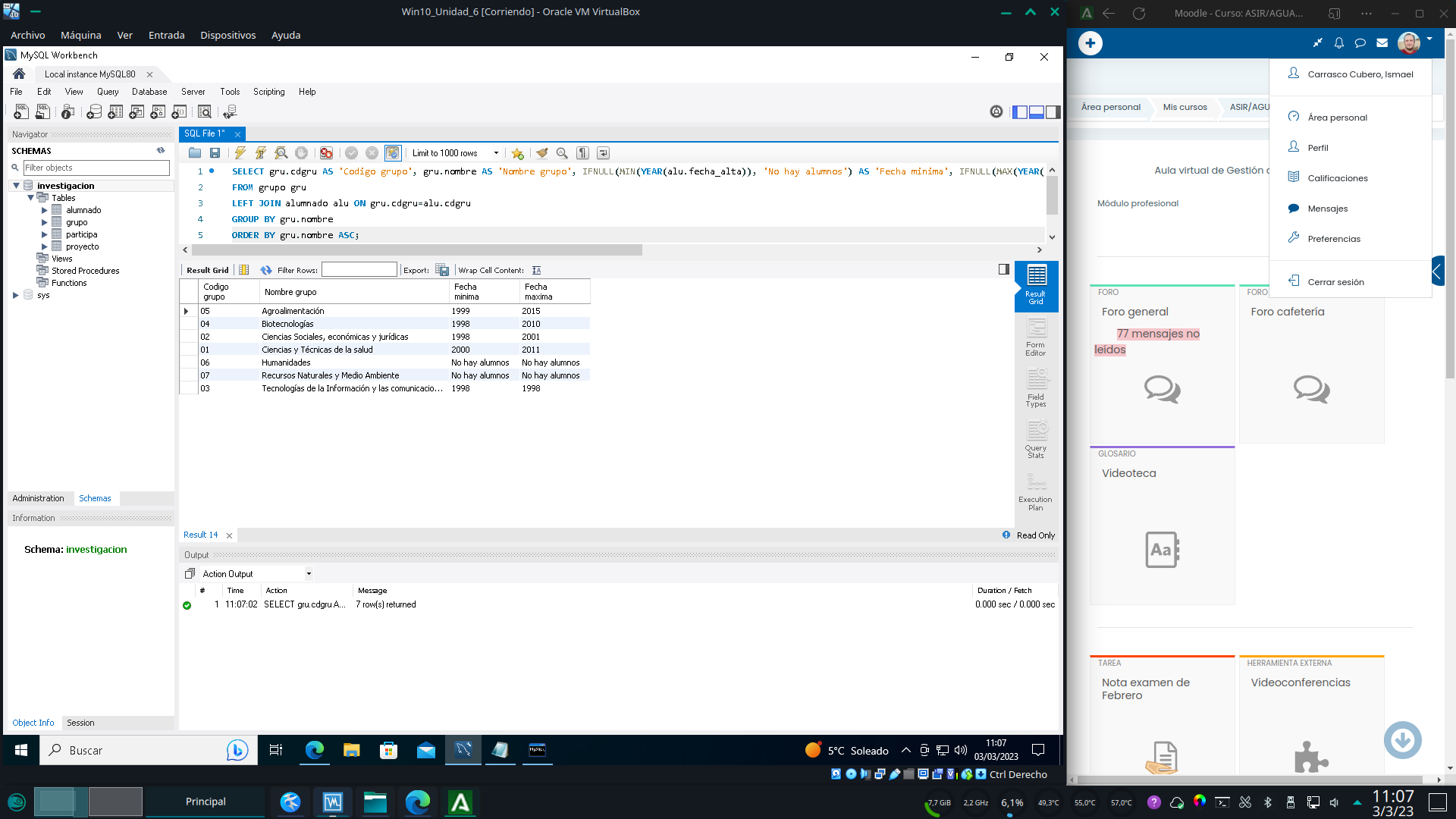
SELECT gru.cdgru AS 'Codigo grupo', gru.nombre AS 'Nombre grupo', IFNULL(MIN(YEAR(alu.fecha\_alta)), 'No hay alumnos') AS 'Fecha minima', IFNULL(MAX(YEAR(alu.fecha\_alta)), 'No hay alumnos') AS 'Fecha maxima'

FROM grupo gru

LEFT JOIN alumnado alu ON gru.cdgru=alu.cdgru

GROUP BY gru.nombre

ORDER BY gru.nombre ASC;



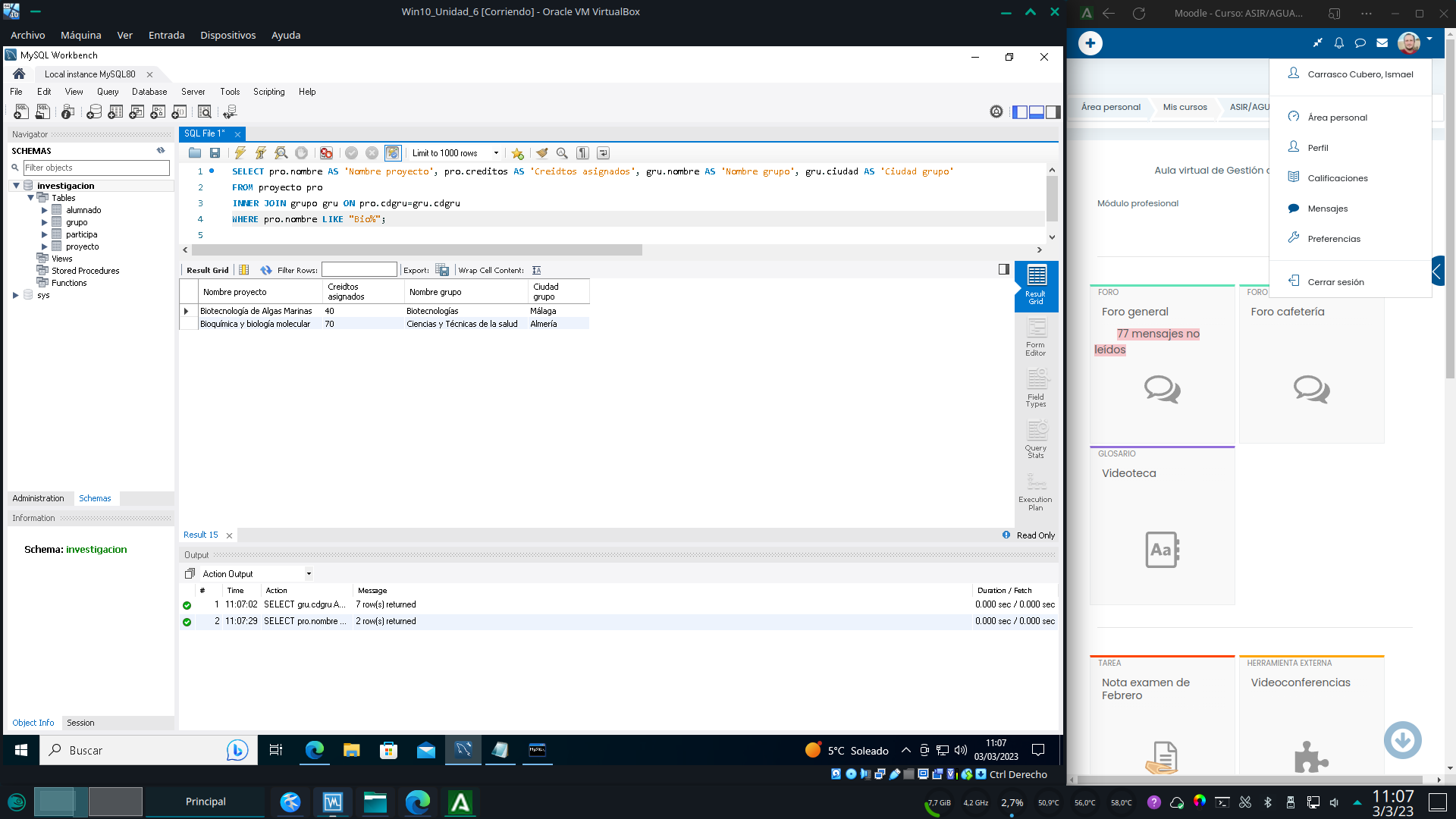
**11)** Lista de los nombres de proyectos en los que aparezca la palabra 'bio', indicando sus créditos, el nombre del grupo que lo dirige y su ciudad.

SELECT pro.nombre AS 'Nombre proyecto', pro.creditos AS 'Creidtos asignados', gru.nombre AS 'Nombre grupo', gru.ciudad AS 'Ciudad grupo'

FROM proyecto pro

INNER JOIN grupo gru ON pro.cdgru=gru.cdgru

WHERE pro.nombre LIKE "Bio%";



**12)** Total de alumnado por grupo de investigación con encabezado 'Total alumnado' y total de sus cuotas con encabezado 'Total cuotas grupo', indicando además el código, nombre del grupo y ciudad, y ordenando de mayor a menor importe en cuotas. Los grupos sin alumnado deben aparecer en el listado.

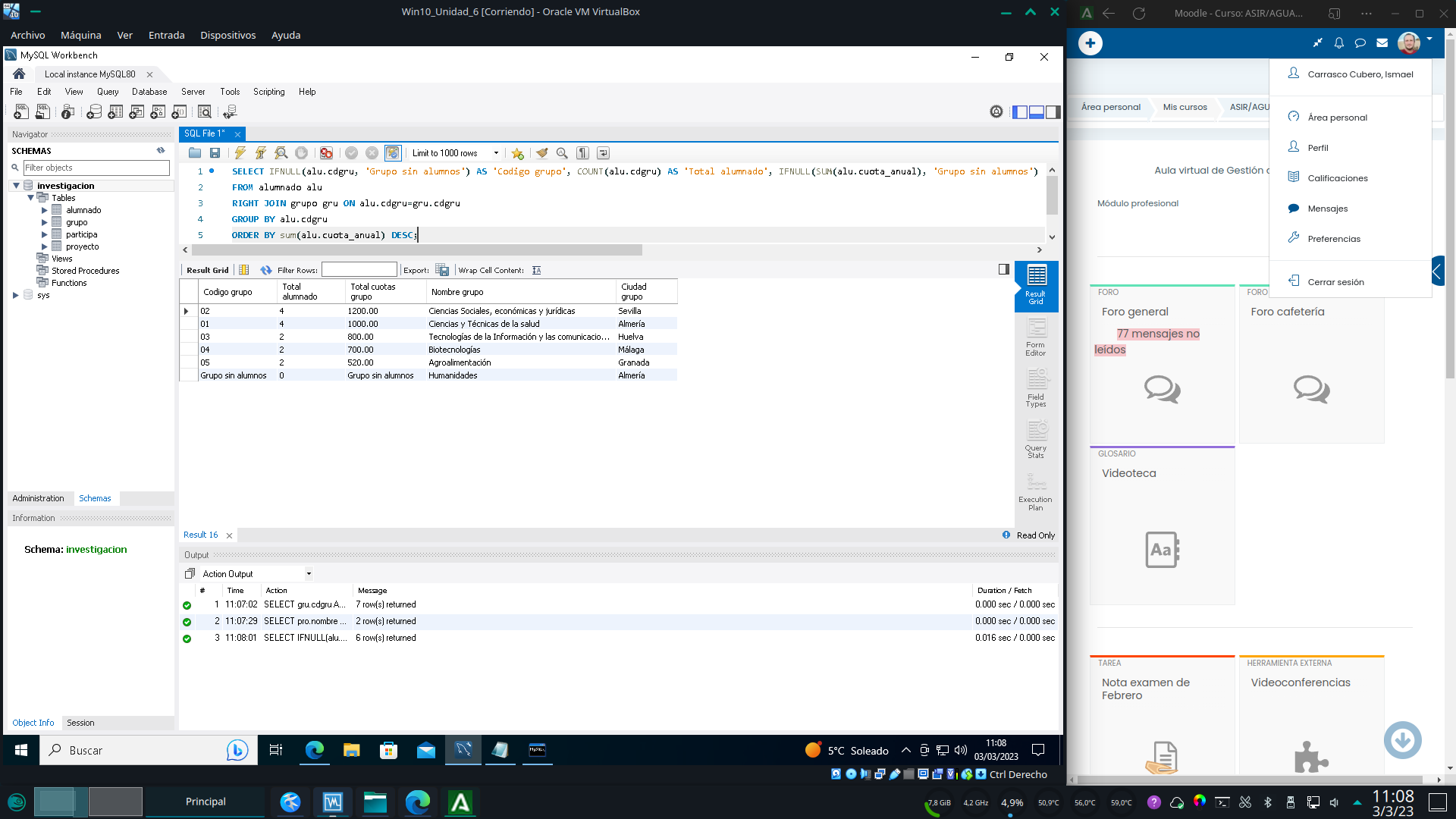
SELECT IFNULL(alu.cdgru, 'Grupo sin alumnos') AS 'Codigo grupo', COUNT(alu.cdgru) AS 'Total alumnado', IFNULL(SUM(alu.cuota\_anual), 'Grupo sin alumnos') AS 'Total cuotas grupo', gru.nombre AS 'Nombre grupo', gru.ciudad AS 'Ciudad grupo'

FROM alumnado alu

RIGHT JOIN grupo gru ON alu.cdgru=gru.cdgru

GROUP BY alu.cdgru

ORDER BY sum(alu.cuota\_anual) DESC;



**13)** Para cada proyecto, su código de proyecto y total de alumnos y alumnas que participan en él. En el listado se deben mostrar solo los proyectos en los que participan 3 alumnos o alumnas o menos. Ordenar de mayor a menor número de participantes.

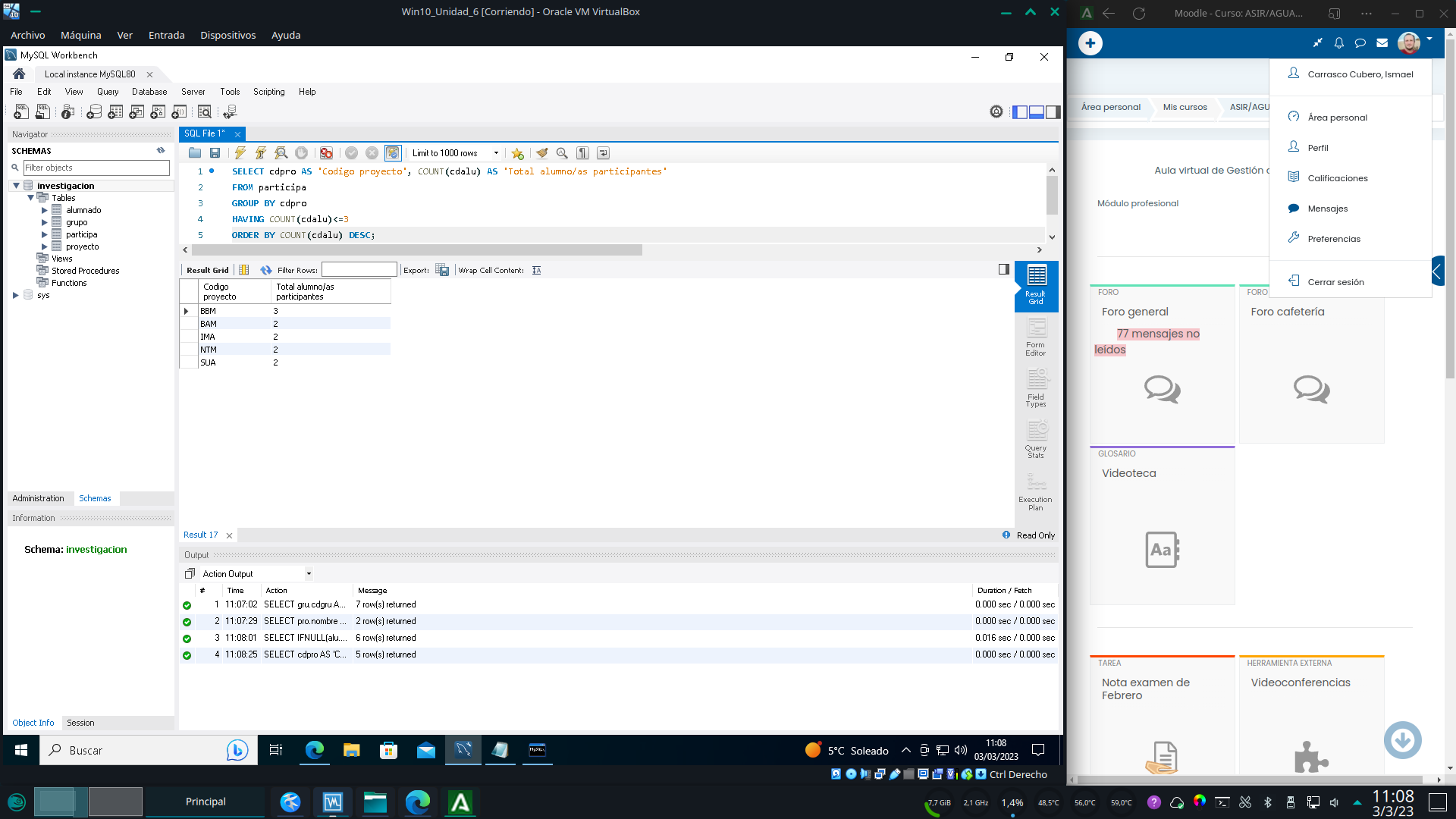
SELECT cdpro AS 'Codigo proyecto', COUNT(cdalu) AS 'Total alumno/as participantes'

FROM participa

GROUP BY cdpro

HAVING COUNT(cdalu)<=3

ORDER BY COUNT(cdalu) DESC;



**14)** Modificar la consulta anterior de manera que se limite el listado a dos de los proyectos con más alumnos y alumnas, incluyendo en el listado el nombre y los créditos del proyecto.

SELECT par.cdpro AS 'Codigo proyecto', COUNT(par.cdalu) AS 'Total alumno/as participantes', pro.nombre, pro.creditos

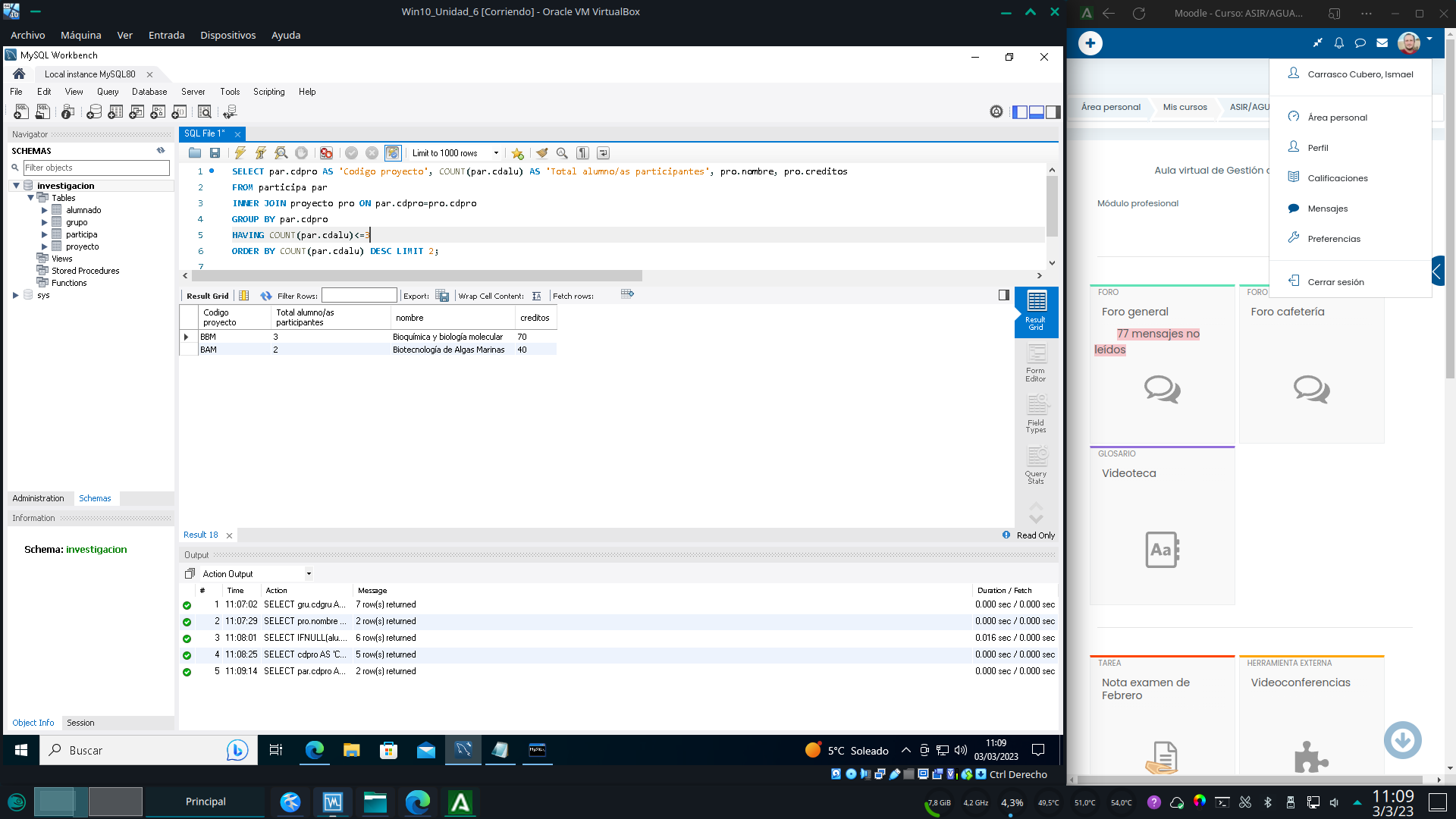
FROM participa par

INNER JOIN proyecto pro ON par.cdpro=pro.cdpro

GROUP BY par.cdpro

HAVING COUNT(par.cdalu)<=3

ORDER BY COUNT(par.cdalu) DESC LIMIT 2;



**15)** Listado que muestre el total de alumnos y alumnas dados de alta en grupos de investigación en cada año. Limitar el listado a dos de los años con más altas.

SELECT year(fecha\_alta) AS 'Año de alta', COUNT(cdalu) AS 'Numero de altas'

FROM alumnado

GROUP BY year(fecha\_alta)

ORDER BY COUNT(cdalu) DESC LIMIT 2;

