

PRE- LABORATORIO-
PRACTICA SUPERVISADA

Pablo Giovani Ochoa Quan

Clave: 23

V Bachillerato B

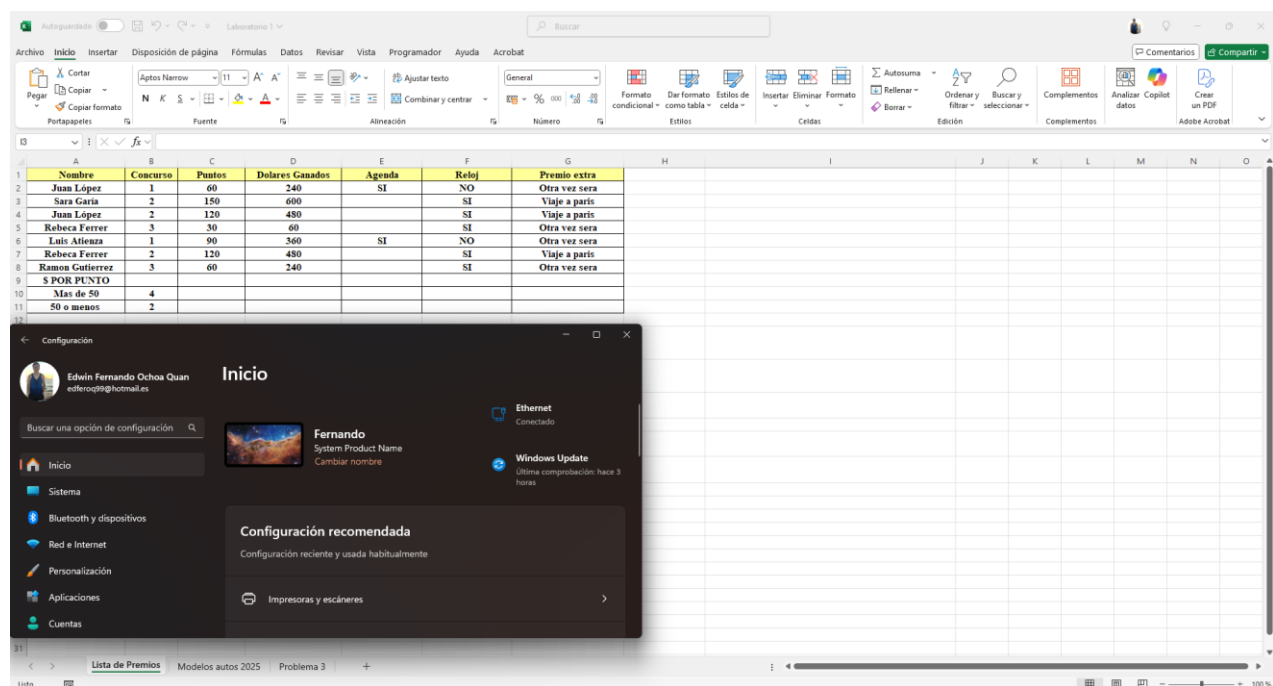
Practica Supervisada

Prof. Luis Orozco

Nueva Guatemala de la Asunción, 28 de abril de 2025

MS Excel

Problema 1:



Nombre	Concurso	Puntos	Dolares Ganados	Agenda	Reloj	Premio extra
Juan López	1	60	240	SI	NO	Otra vez sera
Sara García	2	150	600		SI	Viaje a paris
Juan López	2	120	480		SI	Viaje a paris
Rebeca Ferrer	3	30	60		SI	Otra vez sera
Luis Atienza	1	90	360	SI	NO	Otra vez sera
Rebeca Ferrer	2	120	480		SI	Viaje a paris
Ramon Gutierrez	3	60	240		SI	Otra vez sera
\$ POR PUNTO						
Men de 50	4					
50 o meaos	2					

Dólares ganados: Para obtener los dólares ganados, debemos colocar la siguiente formula en la casilla de dólares ganados: $=SI(Casilla\ con\ los\ puntos > 50; C2 * 4; C2 * 2)$. La celda de puntos contiene los puntos de los participantes, en la formula se coloca el signo “>” para comparar si los puntos son mayores a 50, el “;” se usa para separar estas dos posibles acciones. Si los puntos son mayores a 50, la casilla de puntos se multiplica por 4, si los puntos son menores a 50 se multiplicará por 2.

Agenda: Para obtener los datos de agenda, debemos usar la siguiente formula:

$=SI(Concurso=1;"SI";"$). Esta formula cumple con la condición de que si una persona solo participo en 1 concurso la celda agenda mostrará la palabra “SI”; si no, la celda permanecerá vacía.

Reloj: Para obtener los datos de reloj, debemos usar la siguiente formula en la casilla de reloj:

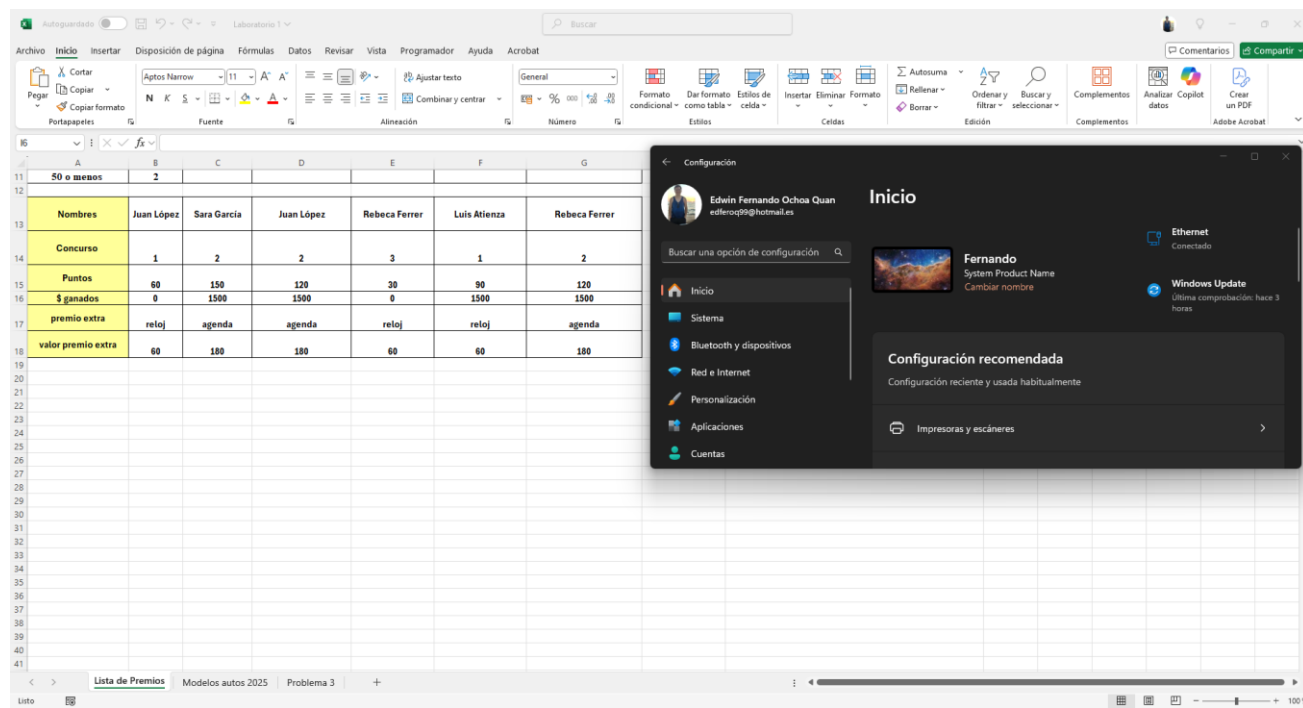
$=SI(Agenda="SI";"NO";"SI")$. Esta fórmula cumple una condición, si la celda de agenda tiene “SI”

deberá aparecer la palabra “NO” en la celda de reloj, en caso de que “agenda” este vacío aparecerá la palabra “SI”.

Premio Extra: Para obtener los datos de Premio extra, debemos usar la siguiente formula:

=SI(Puntos>100;"Viaje a Paris";" Otra vez será"). Esta formula cumple con una condición, si tiene más 100 puntos deberá aparecer “Viaje a Paris”, en caso de que tenga menos de 100 puntos aparecerá: “Otra vez será”.

Hoja 1 problema 2:



The screenshot shows a Microsoft Excel spreadsheet with a table of contest results. The table has columns for names, scores, and prizes. A Windows Start menu overlay is visible on the right side of the spreadsheet.

	A	B	C	D	E	F	G
11	50 o mas	2					
12							
13	Nombres	Juan López	Sara García	Juan López	Rebeca Ferrer	Luis Atienza	Rebeca Ferrer
14	Concurso	1	2	2	3	1	2
15	Puntos	60	150	120	30	90	120
16	\$ ganados	0	1500	1500	0	1500	1500
17	premio extra	reloj	agenda	agenda	reloj	reloj	agenda
18	valor premio extra	60	180	180	60	60	100

\$ Ganados: Para obtener los dólares ganados debemos utilizar la siguiente formula:

=SI(B15<80;"0";"1500"). Esta fórmula cumple con una condición, Si los puntos que tienen son menores a 80, los dólares ganados serán de 0, si los puntos son mayores a 80, los dólares obtenidos serán de 1500.

Premio Extra: Para obtener el premio extra debemos utilizar la siguiente formula:

=SI(B15>=120;"agenda";"reloj"). Esta formula cumple con una condición, Si los puntos son menores a 120 se colocará “reloj”, si los puntos son mayores a 120 se colocará “agenda”.

Valor premio extra: Para obtener el valor de premio extra debemos utilizar la siguiente formula: =SI(B17="agenda";"180";"60"). Esta fórmula cumple una condición, si en la celda de “Premio extra” aparece la palabra “agenda” el valor de premio extra será de 180, en caso de que no tenga la palabra “agenda” y tenga la palabra “reloj” el valor del premio extra será de 60.

Problema 2:

The screenshot displays two overlapping windows. The background window is Microsoft Excel, showing a spreadsheet with the following data:

Modelo Vendido	Precio Base	Forma Pago	Descuento	Precio Total
Mercedes 321	15060	Aplazado		15060
Ford 202	7230	Al Contado	361.5	6868.5
Peugeot 105	7230	Al Contado	361.5	6868.5
Ford 202	7230	Al Contado	361.5	6868.5
Mercedes 321	15060	Aplazado		15060
Peugeot 105	7230	Al Contado	361.5	6868.5
Mercedes 321	15060	Aplazado		15060

The foreground window is the Windows Settings application, specifically the 'Inicio' (Home) screen. It shows the user profile 'Edwin Fernando Ochoa Quan' with the email 'edferrero99@hotmail.es'. The 'Configuración recomendada' (Recommended configuration) section is visible, showing 'Impresoras y escáneres' (Printers and scanners) as the next step in the setup process. The 'Sistema' (System) settings are also visible on the left sidebar.

Precio Base: Para obtener el Precio base debemos utilizar la siguiente formula:

=SI(Modelo venido="Mercedes 321";"15060";"7230"). Esta fórmula cumple una condición, si el modelo es “Mercedes 321” el valor es de \$15060 en la celda, en caso contrario que no sea el modelo de mercedes, el valor es de \$7230.

Forma de Pago: Para obtener la forma de pago debemos utilizar la siguiente formula:

=SI(Precio base="15060";"Aplazado"; "Al Contado"). Esta fórmula cumple una condición, si el precio base es de “15060” aparecerá “Aplazado”, en caso contrario que no cumpla esta condición la forma de pago será “Al contado”.

Descuento: Para obtener el descuento debemos utilizar la siguiente formula: =SI(Forma de Pago=” Al Contado”; Precio base*0.05;""). Esta fórmula cumple una condición, si la forma de pago es “al contado” obtendrá un descuento, en caso de que no se cumpla esta condición no tendrá ningún descuento.

Precio Total: Para obtener el descuento debemos utilizar la siguiente formula:

=SI(Descuento=Precio base; 7230-361.5). Esta fórmula cumple una condición, si no tiene descuento se vuelve a copiar el precio base, en caso de que tenga descuento, se realizara la operación y se obtendrá el precio total.

Problema 3:

The screenshot shows a Microsoft Excel spreadsheet with the following data:

Cantidad	Porcentaje	Respuesta	Solución
600000	25%	150000	Muy Bien
5600	60%	3400	Dedíquese a otra cosa
740	95%	650	Dedíquese a otra cosa
50000	5%	2500	Muy Bien

Overlaid on the Excel window is the Windows Settings application, specifically the 'Inicio' (Home) page. It displays the user's name 'Edwin Fernando Ochoa Quan', system information, and a 'Configuración recomendada' (Recommended configuration) section.

Solución: Para obtener la solución debemos utilizar la siguiente formula:

=SI(Respuesta=Cantidad*Descuento;" Muy Bien"; "Dedíquese a otra cosa"). Esta fórmula cumple una condición, si la multiplicación entre la "Cantidad" y el "Porcentaje" es el mismo resultado que "Respuesta" saldrá "Muy bien", en caso contrario que no sea el mismo resultado saldrá "Dedíquese a otra cosa".

Cisco Packet Tracer

Clase: C

Máscara de Subred (por defecto): 255.255.255.0

Máscara de Subred (adaptada): 255.255.255.224

Nº total de subredes: $2^3 = 8$

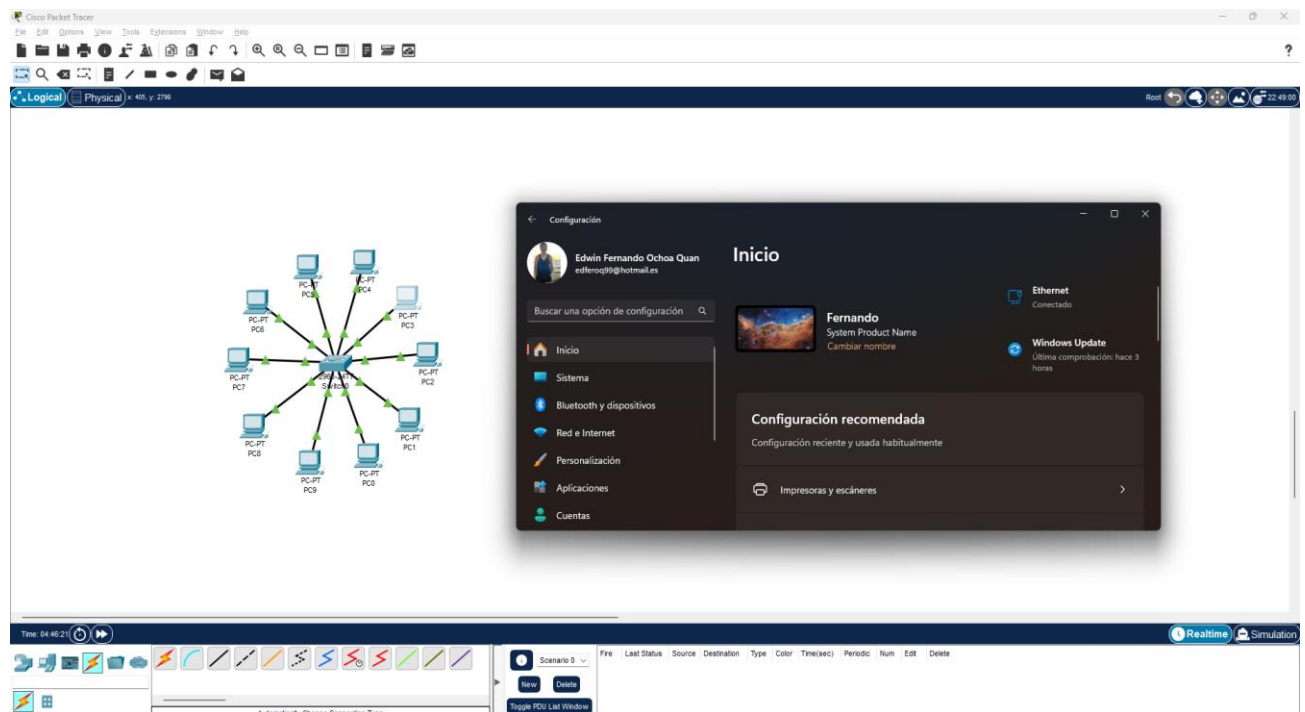
Nº de redes útiles: 8

Nº total de direcciones de host: $2^5 = 32$ por subred

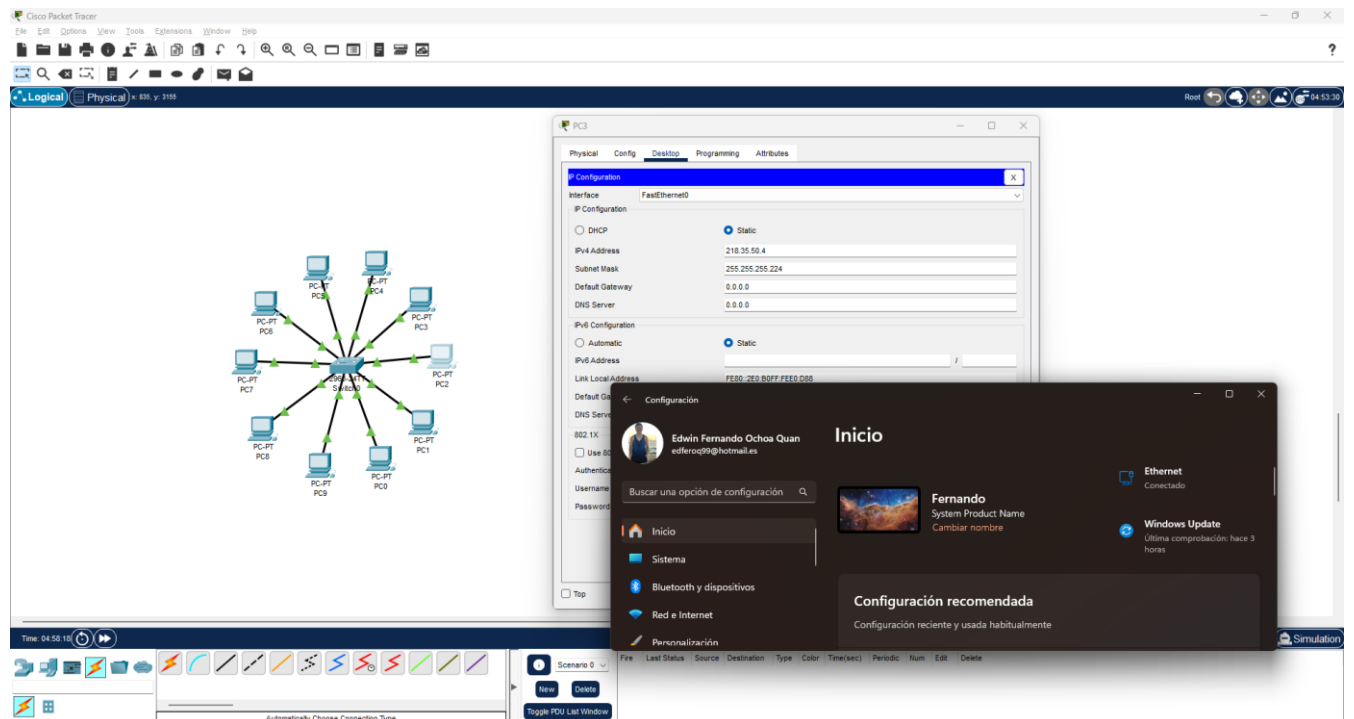
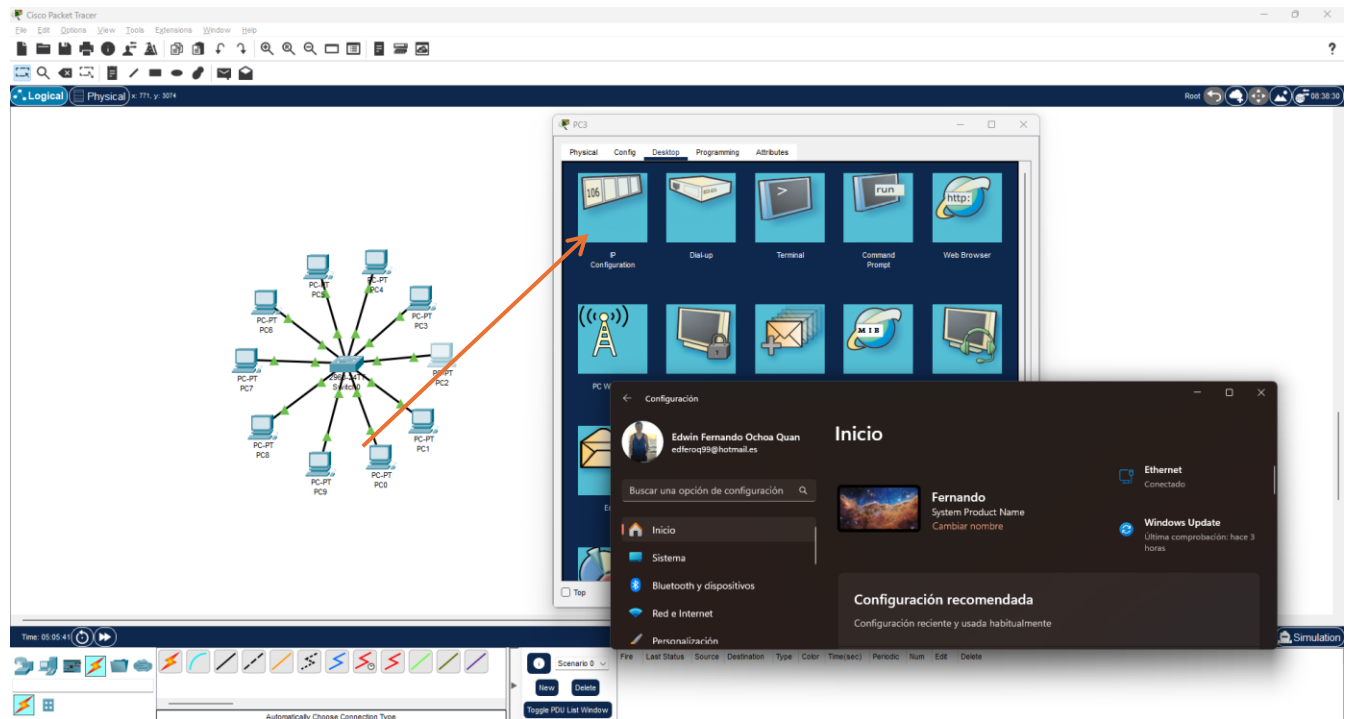
Nº de direcciones útiles: $32 - 2 = 30$ por subred

Nº de bits cogidos: 3

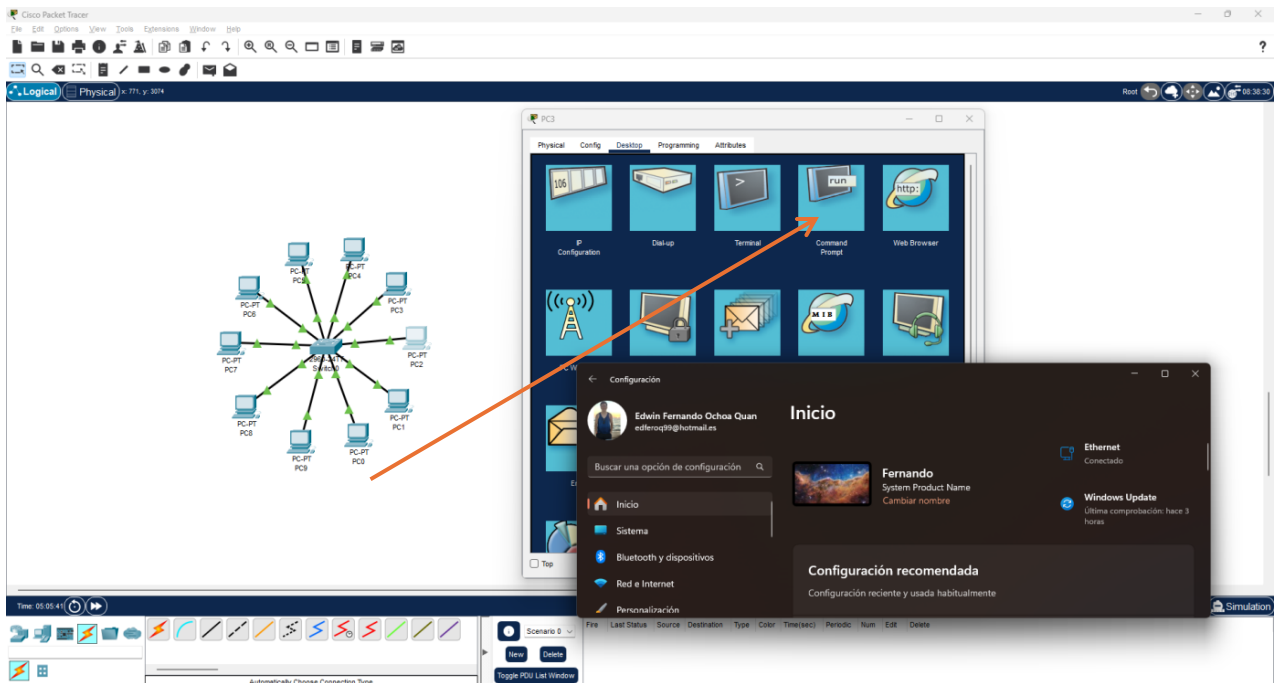
Diseño:



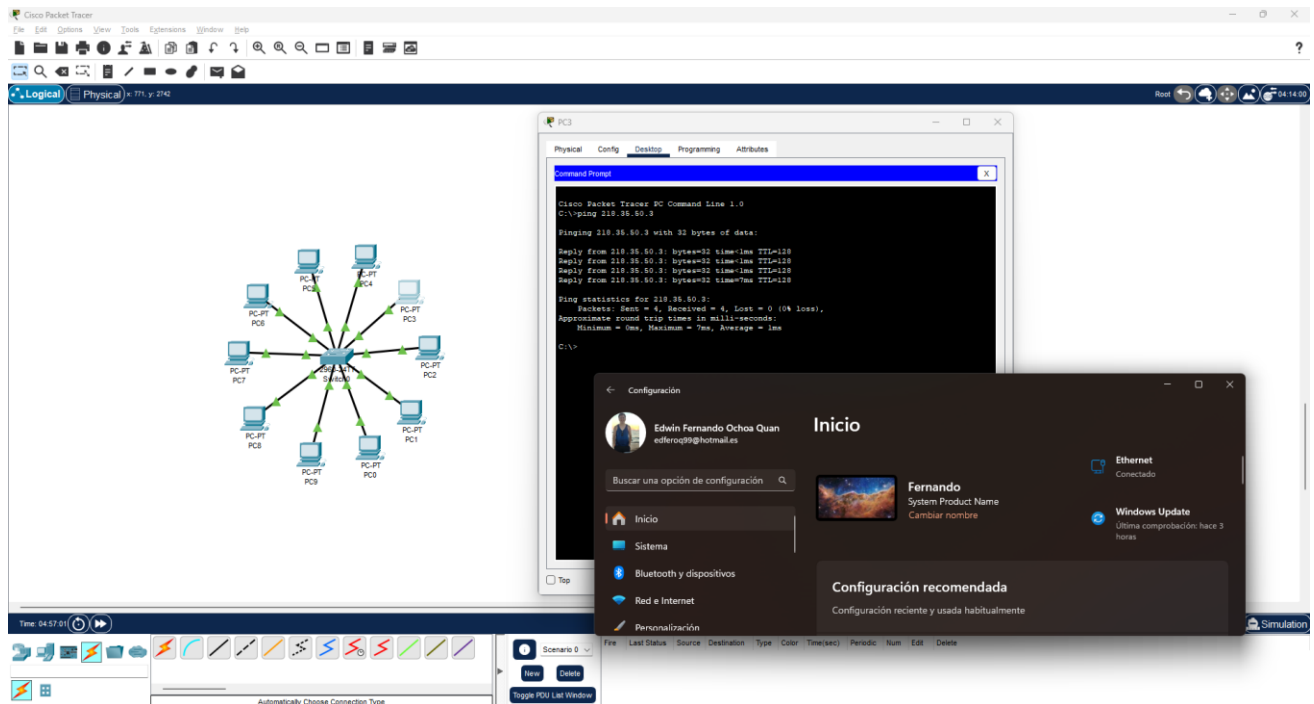
Funcionamiento de la red:



Colocamos nuestra ip: 218.35.50.1, y automáticamente se nos colocara nuestra mascar de subred, en las siguientes computadoras colocamos siempre la misma ip pero cambiamos el ultimo número, que seria .2 y así sucesivamente.



Para verificar la configuración que colocamos en ip configuration, utilizaremos el command prompt, esto nos sirve para comprobar si hay conexión entre las computadoras, utilizando el comando “ping”.



En este caso yo conecte la PC3 con la PC2, utilizando el ping y la ip de la pc que deseo conectar.

MYSQLWORKBENCH

Explicación código:

CREATE DATABASE empresafixoSA1; Creamos la base de datos empresafixoSA1.

USE empresafixoSA1; Usaremos nuestra base de datos empresa fixoSA1.

CREATE TABLE dirección (Creamos nuestra tabla direcciones que contendrá la información.

idDirección INT PRIMARY KEY, Creamos nuestro identificador de la dirección, con la llave primaria de la tabla y enteros.

Dirección VARCHAR (45) Columna para guardar la calle o avenida. Almacena hasta 45 caracteres.

Ciudad VARCHAR (45) Columna para la ciudad. Almacena hasta 45 caracteres.

País VARCHAR (45) Columna para guardar el país. Almacena hasta 45 caracteres.

);

insert into direccion(idDireccion,Direccion,Ciudad,Pais) Es el comando para insertar los datos de direccion en la tabla.

Values Se coloca los valores para cada columna.

(1,"6a Avenida 3-55 Zona 10","Guatemala","Guatemala"), registros de la tabla.

(2,"Avenida Las Américas 16-30 Zona 13","Guatemala","Guatemala"), registros de la tabla.

(3,"4a Calle 7-65 Zona 1 "," Antigua Guatemala","Guatemala"), registros de la tabla.

(4,"Boulevard Liberación 15-70 ","Guatemala","Guatemala"), registros de la tabla.

(5,"Calzada Roosevelt 22-43 Zona 7","Guatemala","Guatemala"); registros de la tabla.

CREATE TABLE cliente (Creamos nuestra tabla clientes que contendrá la información.

idCliente INT PRIMARY KEY, Creamos nuestro identificador de cliente, con la llave primaria de la tabla y enteros.

Nombre VARCHAR(45), Columna para guardar el nombre. Almacena hasta 45 caracteres.

Apellido VARCHAR(45), Columna para guardar el apellido. Almacena hasta 45 caracteres.

Edad INT, Columna para guardar la edad. La guarda en tipo entero.

Direccion_idDireccion INT, Es la relación de la dirección, con enteros.

FOREIGN KEY (Direccion_idDireccion) REFERENCES direccion(idDireccion) La tabla cliente debe coincidir con la tabla dirección, el foreign key va a hacer que cada cliente tenga una dirección válida que ya esté registrada en la tabla dirección.

);

insert into cliente(idCliente,Nombre,Apellido,Edad,Direccion_idDireccion) Es el comando para insertar los datos de clientes en la tabla.

values Se coloca los valores para cada columna.

(1,"Juliana ","Pérez",25,1), registros de la tabla.

(2,"María","González",32,2), registros de la tabla.

(3,"Juliana","Ochoa",28,3), registros de la tabla.

(4,"Sofía","Martínez",22,4), registros de la tabla.

(5,"José","Hernández",30,5); registros de la tabla.

CREATE TABLE categoría (Creamos nuestra tabla categoría que contendrá la información.

idCategoría INT PRIMARY KEY, Creamos nuestro identificador de categoría, con la llave primaria de la tabla y enteros.

Nombre VARCHAR(45) Columna para guardar el nombre de la categoría. Almacena hasta 45 caracteres.

);

insert into categoría(idCategoría,Nombre) Es el comando para insertar los datos de categoría en la tabla.

values Se coloca los valores para cada columna.

(1,"Acción"), registros de la tabla.

(2,"Drama"), registros de la tabla.

(3,"Comedia"), registros de la tabla.

(4,"Terror"), registros de la tabla.

(5,"Animación"); registros de la tabla.

CREATE TABLE películas (Creamos nuestra tabla películas que contendrá la información.

idPeliculas INT PRIMARY KEY, Creamos nuestro identificador de las películas, con la llave primaria de la tabla y enteros.

Nombre VARCHAR(45), Columna para guardar el nombre de película. Almacena hasta 45 caracteres.

Duracion INT, Columna para guardar la duración de la película. La guarda en tipo entero.

Descripcion TEXT, Columna para guardar la descripción de la película. La guarda en tipo texto.

Año INT, Columna para guardar el año de la película. La guarda en tipo entero.

Categoria_idCategoria INT, Es la relación con categoria, con enteros.

FOREIGN KEY (Categoria_idCategoria) REFERENCES categoria(idCategoria) La tabla película debe coincidir con la tabla categoría, el foreign key va a hacer que cada película tenga una categoría correcta que ya esté registrada en la tabla.

);

insert into peliculas(idPeliculas,Nombre,Duracion,Descripcion,Año,Categoria_idCategoria)
) Es el comando para insertar los datos de película y categoria en la tabla.

Values Se coloca los valores para cada columna.

(1, "pokemon1", 120, "Ash y Pikachu viven una nueva aventura para salvar el mundo",
2000, 5), registros de la tabla.

(2, "Avengers", 180, "Los superhéroes se unen para luchar contra Thanos.", 2019, 1),
registros de la tabla.

(3, "Titanic", 195, "Una historia de amor épica en el trágico hundimiento del Titanic.",
1997, 2), registros de la tabla.

(4, "pokemon1", 110, "Un nuevo villano amenaza el mundo Pokémon, y Ash debe detenerlo.", 2002, 5), registros de la tabla.

(5, "Shrek", 90, "Un ogro verde emprende una misión para rescatar a una princesa.", 2001, 3) ; registros de la tabla.

CREATE TABLE inventario (Creamos nuestra tabla inventario que contendrá la información.

idCopiasPeliculas INT PRIMARY KEY, , Creamos nuestro identificador de las copias de las películas, con la llave primaria de la tabla y enteros.

Peliculas_idPeliculas INT, Columna identificadora de las películas en formato entero.

Disponible TINYINT, validación de disponibilidad con números enteros.

FOREIGN KEY (Peliculas_idPeliculas) REFERENCES peliculas(idPeliculas) La tabla inventario debe coincidir con la tabla películas, el foreign key va a hacer que cada inventario tenga una película correcta que ya esté registrada en la tabla.

);

insert into inventario(idCopiasPeliculas,Peliculas_idPeliculas,Disponible) Es el comando para insertar los datos de inventario en la tabla.

values

(1, 1, 1), registros de la tabla si la película esta disponible es 1, si la película no está disponible es 0.

(2, 2, 1), registros de la tabla si la película está disponible es 1, si la película no está disponible es 0.

(3, 3, 0), registros de la tabla si la película está disponible es 1, si la película no está disponible es 0.

(4, 4, 1), registros de la tabla si la película está disponible es 1, si la película no está disponible es 0.

(5, 5, 1); registros de la tabla si la película está disponible es 1, si la película no está disponible es 0.

CREATE TABLE renta (Creamos nuestra tabla renta que contendrá la información.

idRenta INT PRIMARY KEY, Creamos nuestro identificador de la renta, con la llave primaria de la tabla y enteros.

Fecha_Renta DATE, columna para colocar la fecha de la renta.

Fecha_Entrega DATE, columna para colocar la fecha de entrega.

Inventario_idCopiasPeliculas INT, Columna para guardar la copia de la película en el inventario. La guarda en tipo entero.

Cliente_idCliente INT, Columna para guardar al cliente. La guarda en tipo entero.

FOREIGN KEY (Inventario_idCopiasPeliculas) REFERENCES inventario

(id_CopiasPeliculas) La tabla renta debe coincidir con la tabla inventario, el foreign key va a hacer que cada renta tenga un cliente correcto que ya esté registrada en la tabla.

FOREIGN KEY (Cliente_idCliente) REFERENCES cliente(idCliente) La tabla renta debe coincidir con la tabla cliente, el foreign key va a hacer que cada renta tenga un cliente correcto que ya esté registrada en la tabla clientes.

);

Insert into renta(idRenta, Fecha_Renta, Fecha_Entrega, Inventario_idCopiasPeliculas, Cliente_idCliente) Es el comando para insertar los datos de renta en la tabla.

Values Se coloca los valores para cada columna.

(1, "2025-04-10", "2025-04-17", 1, 1), registros de la tabla.

(2, "2025-04-12", "2025-04-19", 2, 2), registros de la tabla.

(3, "2025-04-14", "2025-04-21", 4, 3), registros de la tabla.

(4, "2025-04-16", "2025-04-23", 3, 4), registros de la tabla.

(5, "2025-04-18", "2025-04-25", 4, 5); registros de la tabla.

CREATE TABLE empleado (Creamos nuestra tabla empleados que contendrá la información.

idEmpleado INT PRIMARY KEY, Creamos nuestro identificador de la empleado, con la llave primaria de la tabla y enteros.

Nombre VARCHAR(45), Columna para guardar el nombre del empleado. Almacena hasta 45 caracteres.

Apellido VARCHAR(45), Columna para guardar el apellido del empleado. Almacena hasta 45 caracteres.

Cargo VARCHAR(45), Columna para guardar el cargo del empleado. Almacena hasta 45 caracteres.

Fecha_Contratacion DATE, columna para colocar la fecha de contratación del empleado.

Telefono VARCHAR(20) Columna para guardar el teléfono del empleado. Almacena hasta 20 caracteres.

);

insert into empleado(idEmpleado,Nombre,Apellido,Cargo,Fecha_Contratacion,Telefono)Es el comando para insertar los datos de empleados en la tabla.

Values Se coloca los valores para cada columna.

(1, "Sebastián", "López", "Supervisor de Ventas", "2018-06-12", "55667788"), registros de la tabla.

(2, "Valeria", "Morales", "Recepcionista", "2020-09-25", "44556677"), registros de la tabla.

(3, "Tomás", "Méndez", "Soporte Técnico", "2021-03-18", "33445566"), registros de la tabla.

(4, "Isabela", "Cruz", "Coordinador de Logística", "2019-12-05", "22334455"), registros de la tabla.

(5, "Diego", "Vargas", "Asistente Administrativo", "2023-01-20", "11223344"); registros de la tabla.

CREATE TABLE reserva (Creamos nuestra tabla de reserva que contendrá la información.

idReserva INT PRIMARY KEY, Creamos nuestro identificador reserva, con la llave primaria de la tabla y enteros.

Fecha_Reserva DATE, columna para colocar la fecha de reserva.

Cliente_idCliente INT, columna para identificar al cliente, en enteros.

Inventario_idCopiasPeliculas INT, columna para colocar las copias de las películas en el inventario, en enteros.

FOREIGN KEY (Cliente_idCliente) REFERENCES cliente(idCliente), La tabla reserva debe coincidir con la tabla cliente, el foreign key va a hacer que cada reserva tenga un cliente correcto que ya esté registrada en la tabla inventario.

FOREIGN KEY (Inventario_idCopiasPeliculas) REFERENCES inventario (idCopiasPeliculas) La tabla reserva debe coincidir con la tabla inventario, el foreign key va a hacer que cada reserva tenga un inventario correcto.

);

insert into reserva (idReserva ,Fecha_Reserva, Cliente_idCliente ,Inventario_idCopiasPeliculas). Es el comando para insertar los datos de reserva en la tabla.

Values Se coloca los valores para cada columna.

(1, "2025-04-15", 1, 3), registros de la tabla.

(2, "2025-04-19", 2, 1), registros de la tabla.

(3, "2025-04-24", 3, 4), registros de la tabla.

(4, "2025-05-08", 4, 2), registros de la tabla.

(5, "2025-07-22", 5, 5); registros de la tabla.

select * from direccion; Muestra todas las direcciones.

select * from cliente; Muestra todas los clientes.

select * from cliente where Nombre = "Juliana"; Muestra solo los clientes que se llamen juliana.

select * from categoria order by Nombre asc; Muestra todas las categorías ordenadas alfabéticamente de la A a la Z.

select * from peliculas; Muestra todas las películas.

select * from peliculas order by Año desc; Muestra películas ordenadas desde la más nueva hacia la más vieja.

select * from inventario; Muestra todos los inventarios.

select * from renta; Muestra todas las rentas registradas.

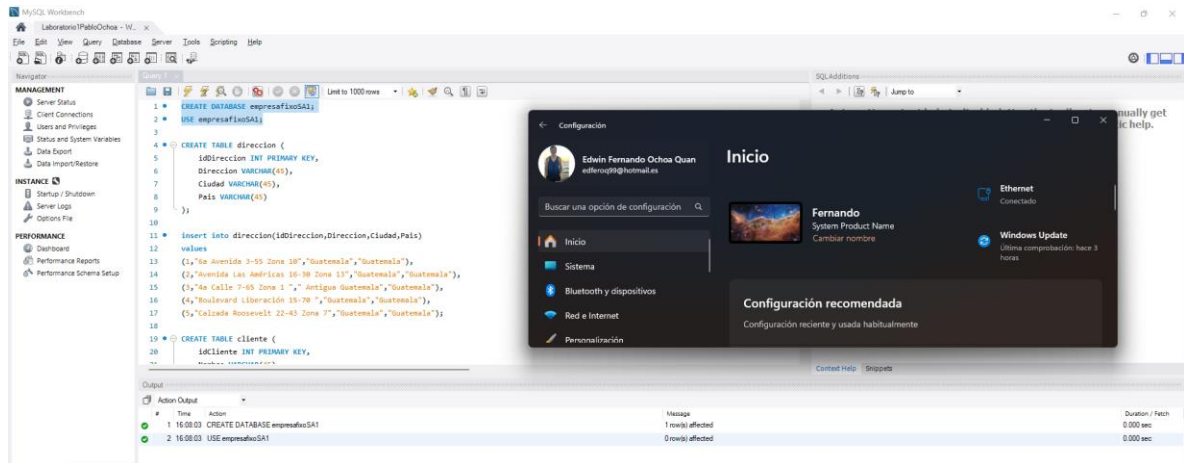
select * from empleado; Muestra todos los empleados.

select * from reserva; Muestra todas las reservas.

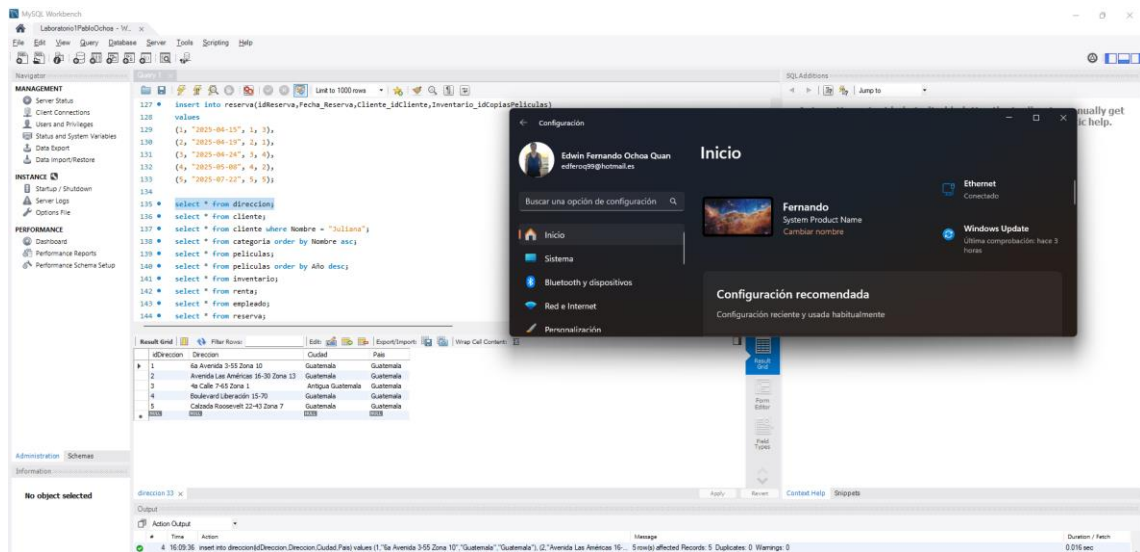
select * from peliculas where not Nombre = "pokemon1"; Muestra todas las películas excepto las que se llaman "pokemon1".

Ejecución:

Create database:



Direccion:



Cliente:

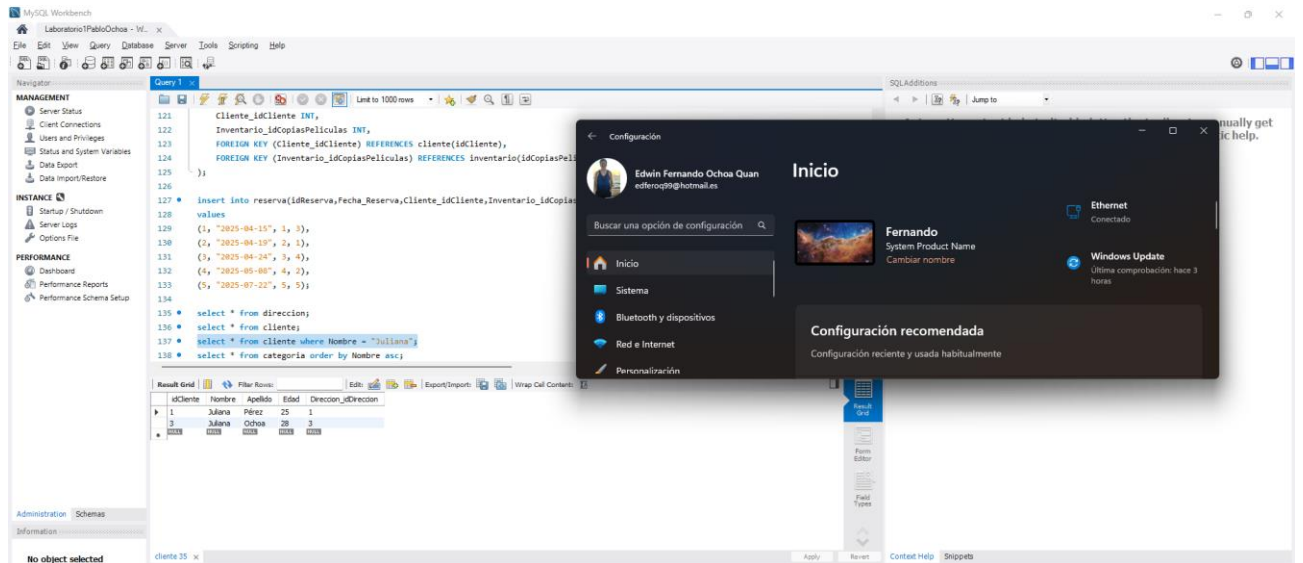
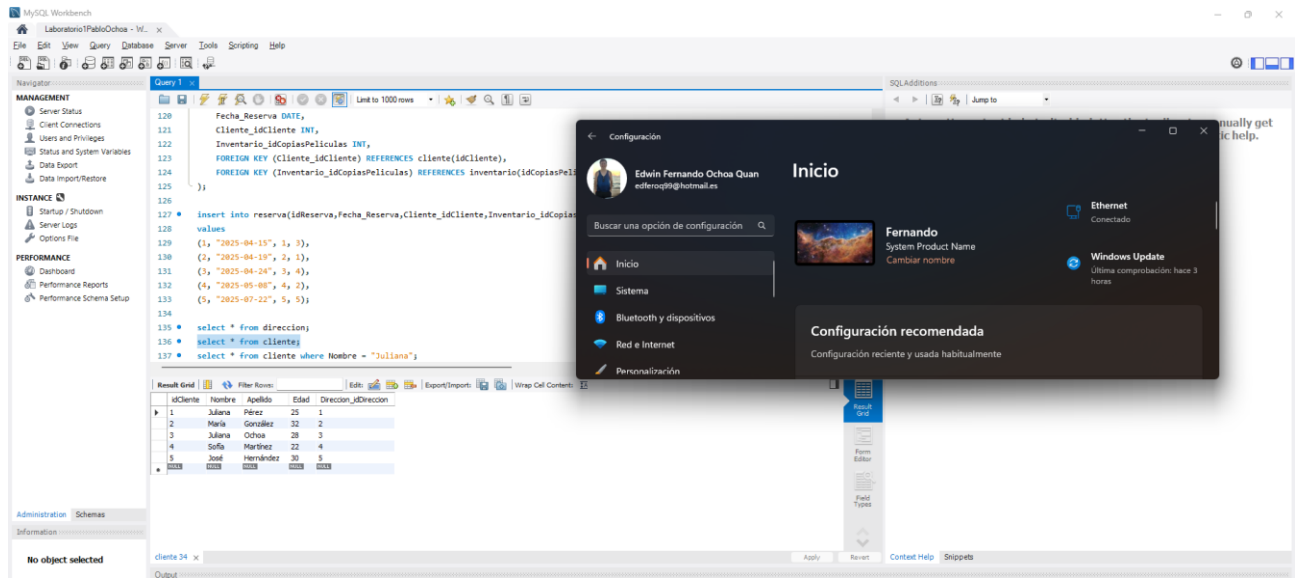


Tabla cliente solo con nombre “Juliana”.

Categoría:

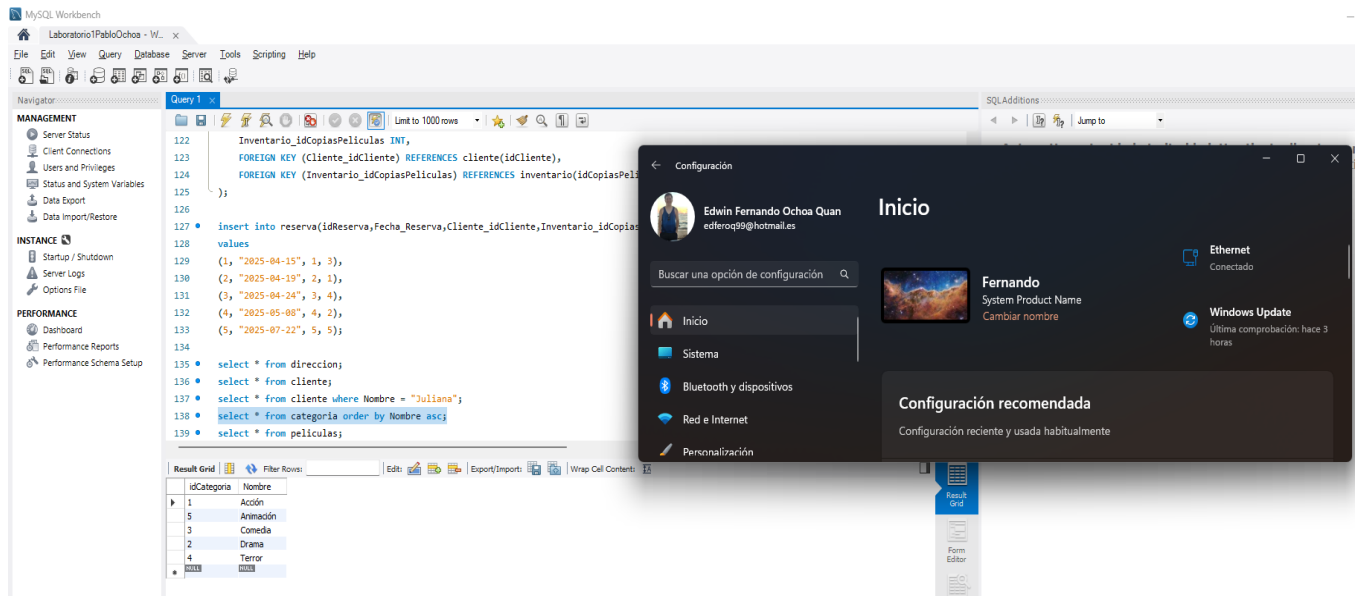
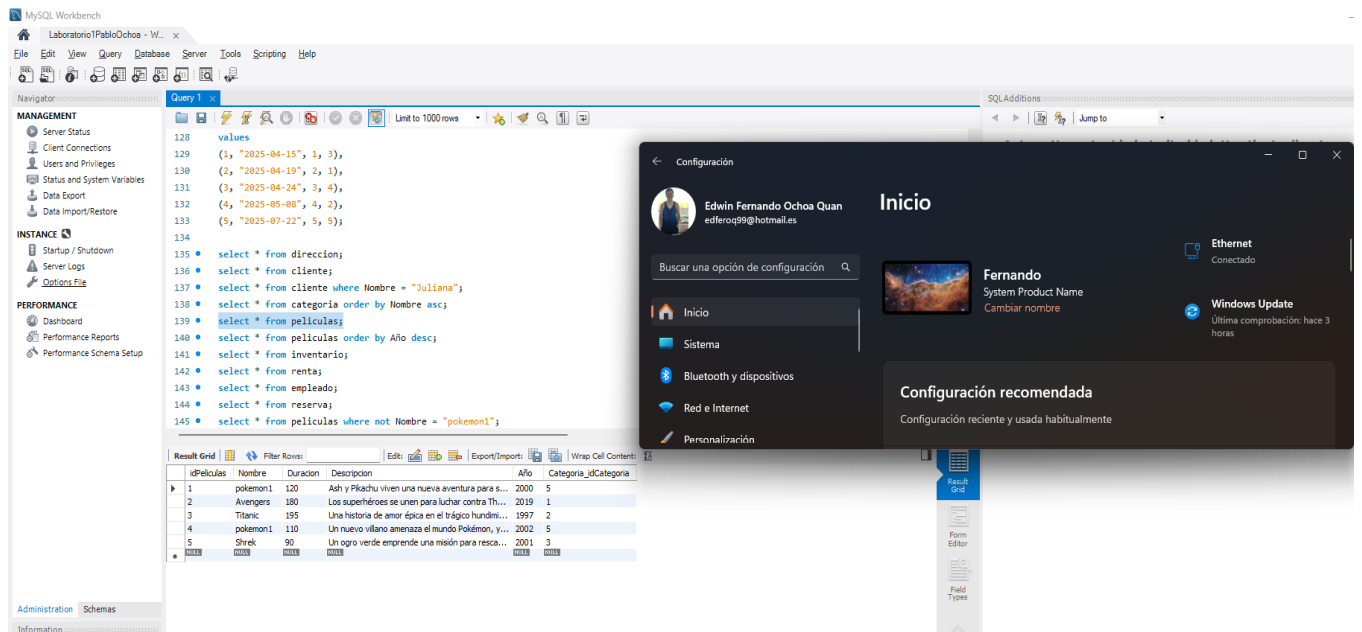


Tabla categoria escrita de forma ascendente del alfabeto (A-Z).

Películas:



MySQL Workbench

Laboratorio1PabloOchoa - W... x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Query 1

```
124 FOREIGN KEY (Inventario_idCopiasPelículas) REFERENCES Inventario(idCopiasPelículas)
125 })
126
127 * insert into reserva(idReserva,fecha_Reserva,Cliente_idCliente,Inventario_idCopias
128 values
129 (1, "2025-04-15", 1, 3),
130 (2, "2025-04-19", 2, 1),
131 (3, "2025-04-24", 3, 4),
132 (4, "2025-05-08", 4, 2),
133 (5, "2025-07-22", 5, 5))
134
135 * select * from direccion;
136 * select * from cliente;
137 * select * from cliente where Nombre = "Juliana";
138 * select * from categoria order by Nombre asc;
139 * select * from películas;
140 * select * from películas order by Año desc;
141 * select * from Inventario;
```

Result Grid

#	idPelículas	Nombre	Duración	Descripción	Año	Categoría_idCategoría
2	Avengers	180	Los superhéroes se unen para luchar contra Th...	2019	1	
4	pokemon1	110	Un nuevo villano amenaza al mundo Pokémon, y...	2002	5	
5	Shrek	90	Un ogro verde emprende una misión para resca...	2001	3	
1	pokemon1	120	Ash y Pikachu viven una nueva aventura para s...	2000	5	
3	Titanic	195	Una historia de amor épica en el trágico hundim...	1997	2	

Administration Schemas

Information

Configuración

Inicio

Edwin Fernando Ochoa Quan
edferoq99@hotmail.es

Buscar una opción de configuración

Fernando
System Product Name
Cambiar nombre

Ethernet
Conectado

Windows Update
Última comprobación: hace 3 horas

Configuración recomendada
Configuración reciente y usada habitualmente

Inicio

Sistema

Bluetooth y dispositivos

Red e Internet

Personalización

Tabla película en forma descendente por años.

MySQL Workbench

Laboratorio1PabloOchoa - W... x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Query 1

```
129 (1, "2025-04-15", 1, 3),
130 (2, "2025-04-19", 2, 1),
131 (3, "2025-04-24", 3, 4),
132 (4, "2025-05-08", 4, 2),
133 (5, "2025-07-22", 5, 5))
134
135 * select * from direccion;
136 * select * from cliente;
137 * select * from cliente where Nombre = "Juliana";
138 * select * from categoria order by Nombre asc;
139 * select * from películas;
140 * select * from películas order by Año desc;
141 * select * from Inventario;
142 * select * from renta;
143 * select * from empleado;
144 * select * from reserva;
145 * select * from películas where not Nombre = "pokemon1";
146
```

Result Grid

#	idPelículas	Nombre	Duración	Descripción	Año	Categoría_idCategoría
2	Avengers	180	Los superhéroes se unen para luchar contra Th...	2019	1	
3	Titanic	195	Una historia de amor épica en el trágico hundim...	1997	2	
5	Shrek	90	Un ogro verde emprende una misión para resca...	2001	3	

Administration Schemas

Information

No object selected

películas 39 x

Apply Revert Context Help Snippets

Configuración

Inicio

Edwin Fernando Ochoa Quan
edferoq99@hotmail.es

Buscar una opción de configuración

Fernando
System Product Name
Cambiar nombre

Ethernet
Conectado

Windows Update
Última comprobación: hace 3 horas

Configuración recomendada
Configuración reciente y usada habitualmente

Inicio

Sistema

Bluetooth y dispositivos

Red e Internet

Personalización

Tabla película sin pokemon1.

Inventario:

The screenshot shows the MySQL Workbench interface with a query editor and a Windows configuration window overlaid.

MySQL Query:

```
125 ))
126
127 insert into reserva(idReserva,fecha_Reserva,Cliente_idCliente,Inventario_idCopia,
128 values
129 (1, "2025-04-15", 1, 3),
130 (2, "2025-04-19", 2, 1),
131 (3, "2025-04-24", 3, 4),
132 (4, "2025-05-08", 4, 2),
133 (5, "2025-07-22", 5, 5);
134
135 select * from direccion;
136 select * from cliente;
137 select * from cliente where Nombre = "Juliana";
138 select * from categoria order by Nombre asc;
139 select * from peliculas;
140 select * from peliculas order by Año desc;
141 select * from inventario;
142 select * from renta;
```

Result Grid:

	idCopiaPeliculas	Peliculas_idPeliculas	Disponible
1	1	1	1
2	2	2	1
3	3	3	0
4	4	4	1
5	5	5	1

Windows Configuration Window:

Inicio

Edwin Fernando Ochoa Quan
edferoq99@hotmail.es

Buscar una opción de configuración

Inicio

Sistema

Bluetooth y dispositivos

Red e Internet

Personalización

Ethernet
Conectado

Fernando
System Product Name
Cambiar nombre

Windows Update
Última comprobación: hace 3 horas

Configuración recomendada
Configuración reciente y usada habitualmente

Empleados:

The screenshot shows the MySQL Workbench interface with a query editor and a Windows configuration window overlaid.

MySQL Query:

```
127 insert into reserva(idReserva,fecha_Reserva,Cliente_idCliente,Inventario_idCopia,Peliculas)
128 values
129 (1, "2025-04-15", 1, 3),
130 (2, "2025-04-19", 2, 1),
131 (3, "2025-04-24", 3, 4),
132 (4, "2025-05-08", 4, 2),
133 (5, "2025-07-22", 5, 5);
134
135 select * from direccion;
136 select * from cliente;
137 select * from cliente where Nombre = "Juliana";
138 select * from categoria order by Nombre asc;
139 select * from peliculas;
140 select * from peliculas order by Año desc;
141 select * from inventario;
142 select * from renta;
143 select * from empleados;
144 select * from reserva;
```

Result Grid:

	idEmpleado	Nombre	Apellido	Cargo	Fecha_Contratacion	Telefono
1	Sebastián	López	Supervisor de Ventas	2018-06-12	55667788	
2	Valeria	Morales	Recepcionista	2020-09-25	44556677	
3	Tomás	Méndez	Soporte Técnico	2021-03-18	33445566	
4	Isabella	Cruz	Coordinador de Logística	2019-12-05	22334455	
5	Diego	Vargas	Asistente Administrativo	2023-01-20	11223344	

Windows Configuration Window:

Inicio

Edwin Fernando Ochoa Quan
edferoq99@hotmail.es

Buscar una opción de configuración

Inicio

Sistema

Bluetooth y dispositivos

Red e Internet

Personalización

Ethernet
Conectado

Fernando
System Product Name
Cambiar nombre

Windows Update
Última comprobación: hace 3 horas

Configuración recomendada
Configuración reciente y usada habitualmente

Tablas adicionales relacionadas al proyecto:

Renta:

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is an INSERT statement into the 'reserva' table, followed by several SELECT statements from various tables. The result grid displays the following data:

	idRenta	Fecha_Renta	Fecha_Entrega	Inventario_idCopiaPeliculas	Cliente_idCliente
1	2025-04-10	2025-04-17	1	1	
2	2025-04-12	2025-04-19	2	2	
3	2025-04-14	2025-04-21	4	3	
4	2025-04-16	2025-04-23	3	4	
5	2025-04-18	2025-04-25	4	5	

Overlaid on the right is a Windows configuration window titled 'Inicio' for user 'Edwin Fernando Ochoa Quan'. It shows system settings like Ethernet, Windows Update, and a recommended configuration section.

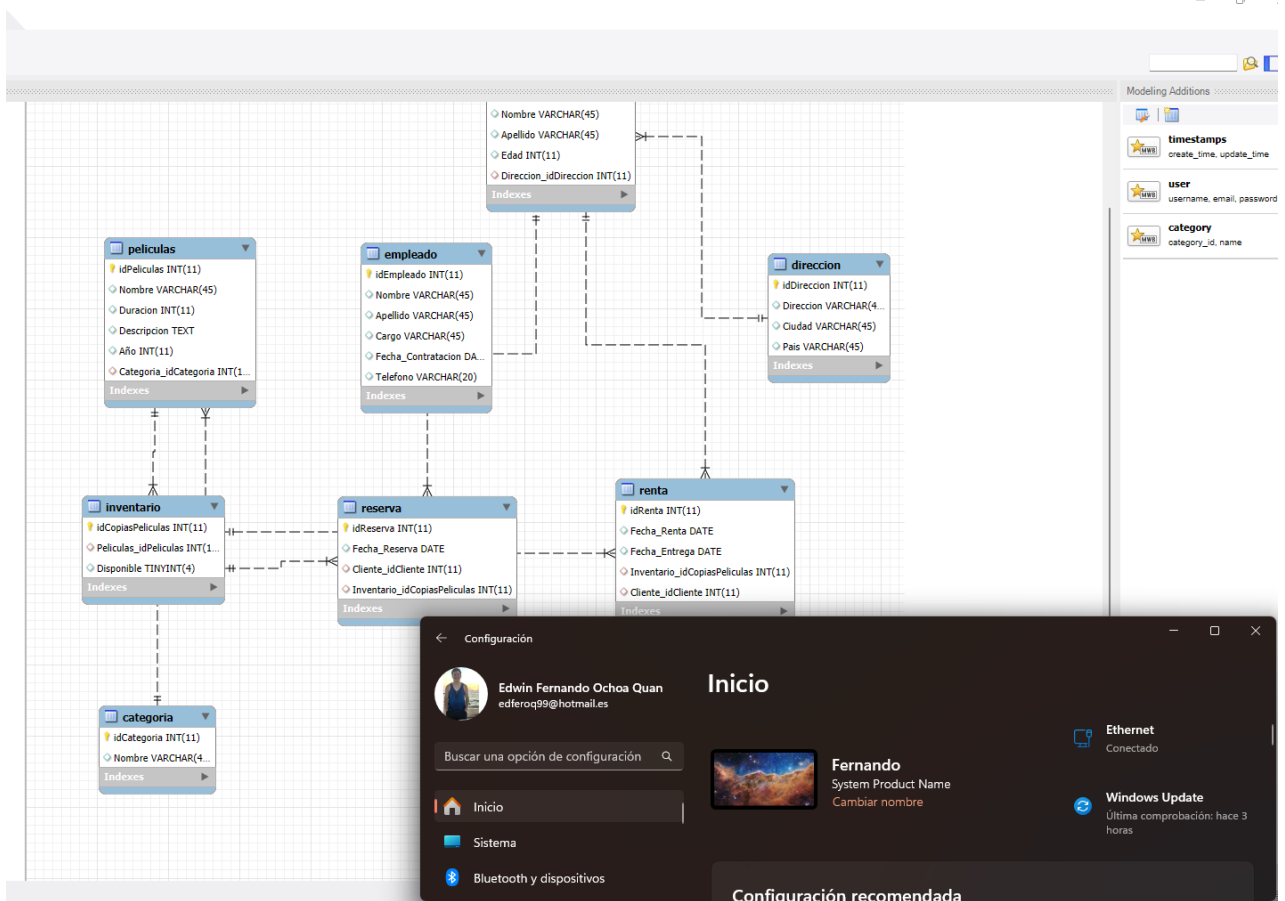
Reserva:

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is an INSERT statement into the 'reserva' table, followed by several SELECT statements from various tables. The result grid displays the following data:

	idReserva	Fecha_Reserva	Cliente_idCliente	Inventario_idCopiaPeliculas
1	2025-04-15	1	3	
2	2025-04-19	2	1	
3	2025-04-24	3	4	
4	2025-05-08	4	2	
5	2025-07-22	5	5	

Overlaid on the right is a Windows configuration window titled 'Inicio' for user 'Edwin Fernando Ochoa Quan'. It shows system settings like Ethernet, Windows Update, and a recommended configuration section.

Diagrama:



VISUAL C#

LOGIN:

```
private void button1_Click(object sender, EventArgs e) Button1 es el boton
“Ingresar”.
{
    if (textBox2.Text == "123456789") Si la caja de texto de contraseña tiene escrito
    "123456789" dejara entrar al siguiente formulario.
    {
        Form1 v1 = new Form1(); lleva a nuestra otra ventana
        v1.Show(); lleva a nuestra otra ventana
        this.Hide(); esconde la ventana actual
    }
    Else en caso que no se cumpla la condición
    {
        MessageBox.Show("La contraseña es incorrecta"); mostrara el mensaje "La
        contraseña es incorrecta".
    }
}
}
```

Crud:

```
public partial class Form1 : Form
{
    int cod = 1; creamos una variable
    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e) botón para registrar los
    datos
    {
        object[] datos = { contiene los valores de varios controles de texto.
            cod.ToString(),
            textBox1.Text, Caja de texto del nombre.
            textBox2.Text, Caja de texto de apellido.
            textBox3.Text, Caja de texto de grado.
            textBox4.Text, Caja de texto de sección.
            textBox5.Text, Caja de texto de edad.
            textBox6.Text, Caja de texto de teléfono.
            textBox7.Text, Caja de texto de Matricula.
            textBox8.Text, Caja de texto de Actual ciclo académico.
            textBox9.Text, Caja de texto del Promedio.
            textBox10.Text, Caja de texto del correo.
        };

        dataGridView1.Rows.Add(datos); Agrega la información registrada y que fue registrada
        en el object de datos.

        cod++;

        MessageBox.Show("Cita registrada"); Este mostrara un mensaje que indicara el registro
        se agrego correctamente.
```

```
}
```

```
private void button3_Click(object sender, EventArgs e) botón de eliminar.
```

```
{
```

```
    textBox1.Text = ""; Caja de texto del nombre quedara vacía.
```

```
    textBox2.Text = ""; Caja de texto de apellido quedara vacía.
```

```
    textBox3.Text = ""; Caja de texto de grado quedara vacía.
```

```
    textBox4.Text = ""; Caja de texto de sección quedara vacía.
```

```
    textBox5.Text = ""; Caja de texto de edad quedara vacía.
```

```
    textBox6.Text = ""; Caja de texto de teléfono quedara vacía.
```

```
    textBox7.Text = ""; Caja de texto de Matricula quedara vacía.
```

```
    textBox8.Text = ""; Caja de texto de Actual ciclo académico quedara vacía.
```

```
    textBox9.Text = ""; Caja de texto del Promedio quedara vacía.
```

```
    textBox10.Text = ""; Caja de texto del correo quedara vacía.
```

```
}
```

`private void button4_Click(object sender, EventArgs e)` botón de créditos.

{

`Creditos v1 = new Creditos();` se crea la ventana 3.

`v1.Show();` mostrar la ventana 3.

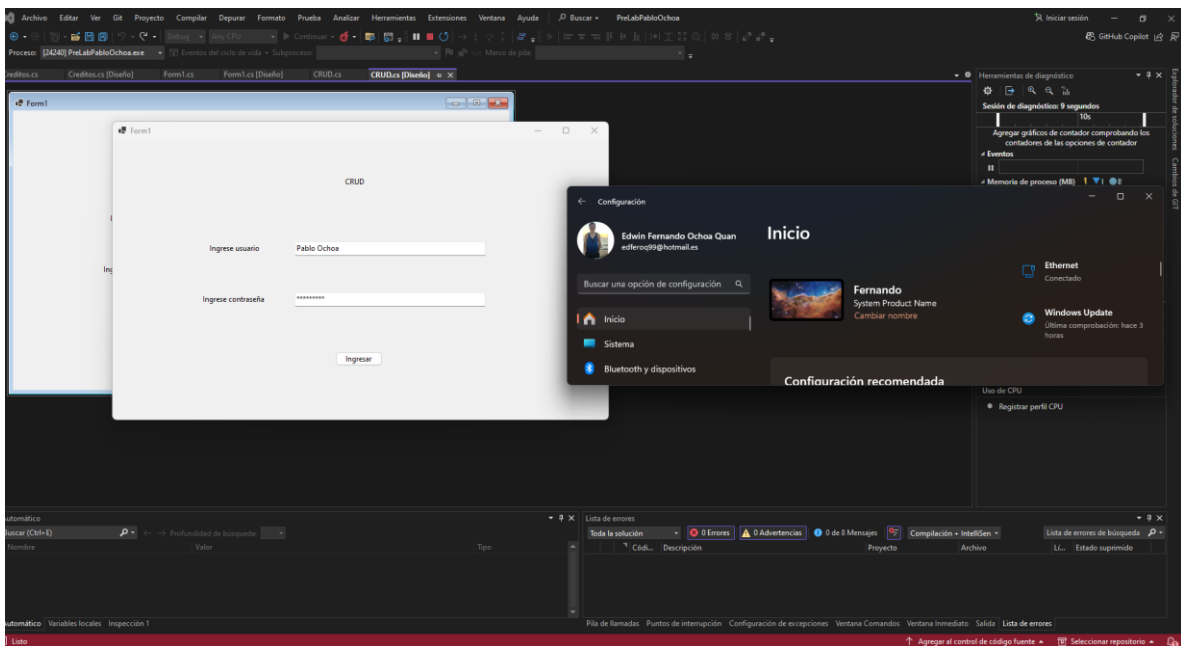
`this.Hide();` oculta la ventana.

}

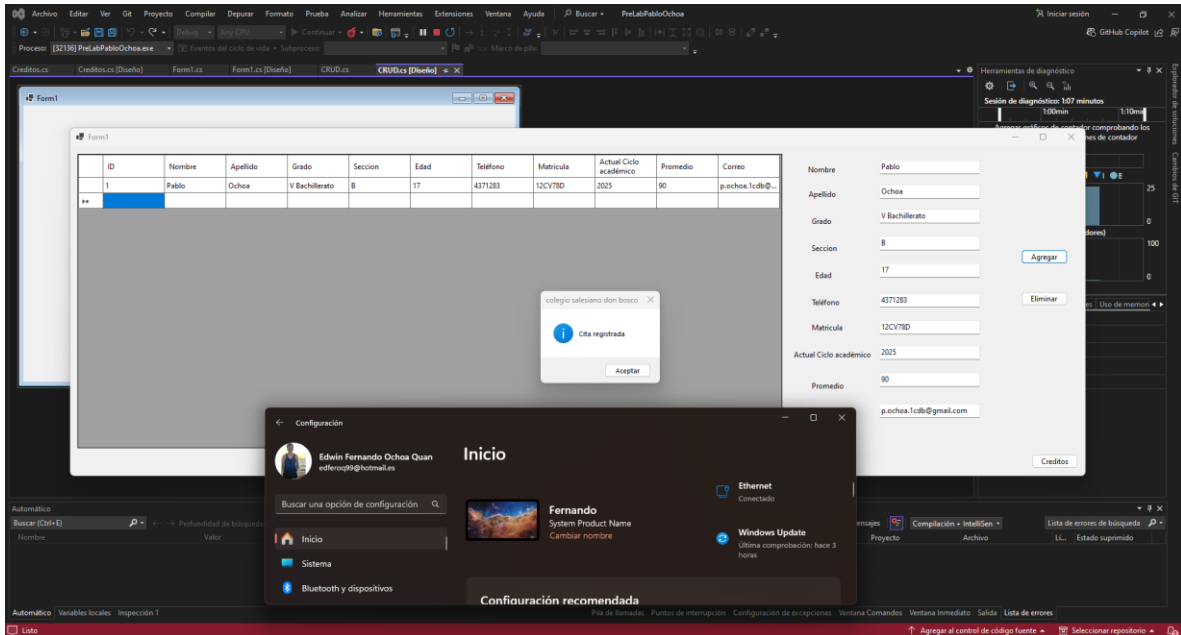
}

}

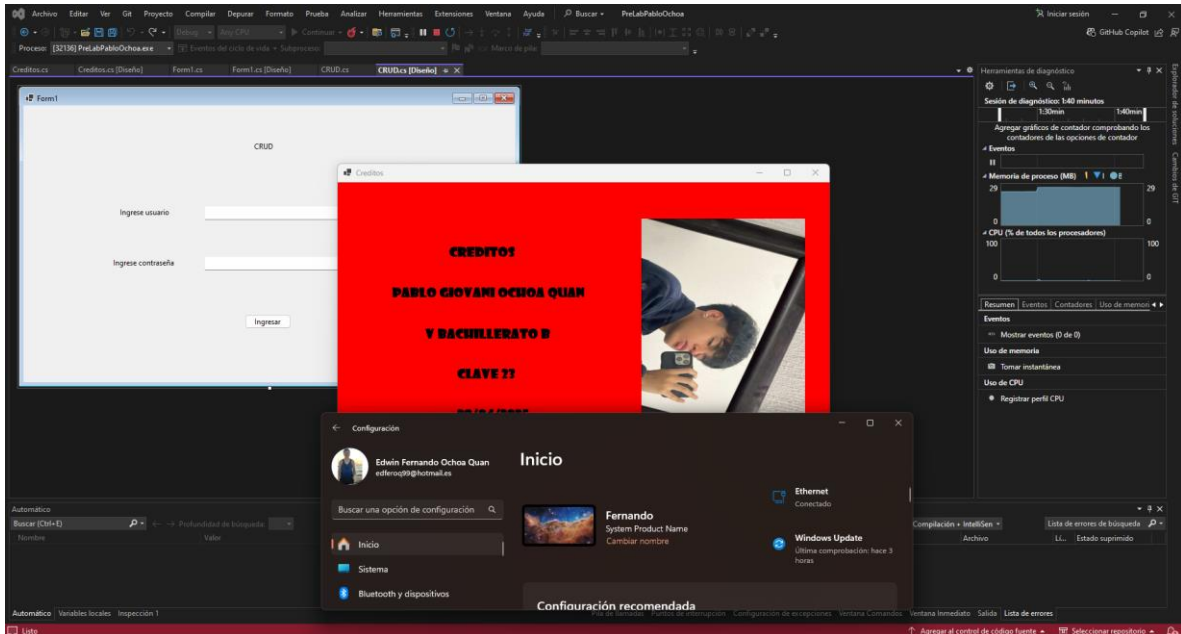
LOGIN



CRUD



CREDITOS



ENLACE GITHUB:

Todos los archivos están en subidos en este enlace de github:

<https://github.com/2012002921PabloOchoa/Pre-Laboratorio-Pablo-Ochoa-VB23.git>