

## **PRE-LABORATORIO I - PRACTICA SUPERVISADA**

Rhyo Donato Motta Pérez

Colegio Salesiano Don Bosco de Guatemala

V Bachillerato Sección “B” clave: 17

Bachillerato en CCLL con Orientación en Computación

Guatemala de la Asunción, 28 de abril de 2025

**MS EXCEL**

**Función SI**

**Table 1: Listado de premios**

	A	B	C	D	E	F	G
1	Nombre	Concurso	Puntos	Dólares ganados	Agenda	Reloj	Premio extra
2	Juan López	1	60	240	SI	NO	Otra vez será
3	Sara García	2	150	600		SI	Viaje a París
4	Juan López	2	120	480		SI	Viaje a París
5	Rebeca Ferrer	3	30	60		SI	Otra vez será
6	Luis Atienza	1	90	360	SI	NO	Otra vez será
7	Rebeca Ferrer	2	120	480		SI	Viaje a París
8	Ramón Gutiérrez	3	60	240		SI	Otra vez será
9	<b>\$ POR PUNTO</b>						
10	Más de 50	4					
11	50 o menos	2					

**Table 2: Resumen de premios**

	Nombre	Juan López	Sara García	Juan López	Rebeca Ferrer	Luis Atienza	Rebeca Ferrer
13	Concurso	1	2	2	3	1	2
14	Puntos	60	150	120	30	90	120
15	€ ganados	0	1500	1500	0	1500	1500
16	Premio extra	Reloj	Agenda	Agenda	Reloj	Reloj	Agenda
17	Valor premio extra	60€	180\$	180\$	60€	60€	180\$

## Hoja “Listado de premios”

### Columna D (Dólares ganados)

#### Celdas con Formulas:

- D2: =IF(C2>50, C2\*B10, C2\*B11)
- D3: =IF(C3>50, C3\*B10, C3\*B11)
- D4: =IF(C4>50, C4\*B10, C4\*B11)
- D5: =IF(C5>50, C5\*B10, C5\*B11)
- D6: =IF(C6>50, C6\*B10, C6\*B11)
- D7: =IF(C7>50, C7\*B10, C7\*B11)
- D8: =IF(C8>50, C8\*B10, C8\*B11)

**Fórmula ejemplo:** =IF(C2>50, C2\*B10, C2\*B11)

#### Desglose completo:

1. IF( - Inicia la función condicional que evaluará una condición lógica
2. C2>50 - Condición que compara si el valor en C2 es mayor que 50

- > es el operador de comparación "mayor que"
  - Evalúa si los puntos ganados superan el umbral de 50
3. , - Separador de argumentos en la función IF
4. C2\*B10 - Valor si verdadero (TRUE)
- \* es el operador de multiplicación
  - Multiplica los puntos (C2) por el valor en B10 (factor de conversión para  $>50$  puntos)
5. , - Separador de argumentos
6. C2\*B11 - Valor si falso (FALSE)
- Multiplica los puntos (C2) por el valor en B11 (factor para  $\leq 50$  puntos)
7. ) - Cierra la función IF

#### **Explicación detallada:**

1. Estructura IF: =IF(condición, valor\_si\_verdadero, valor\_si\_falso)
2. Condición: C2>50 - Verifica si el valor en C2 (puntos) es mayor que 50
  - Operador >: comparación "mayor que"
3. Verdadero: C2\*B10 - Si la condición se cumple:
  - Multiplica los puntos (C2) por el valor en B10 (4 en este caso)
  - Operador \*: multiplicación
4. Falso: C2\*B11 - Si no se cumple:
  - Multiplica los puntos (C2) por el valor en B11 (2 en este caso)
5. Propósito: Calcular ganancias en dólares con diferentes tasas según puntos

#### **Columna E (Agenda)**

#### **Celdas con Formulas:**

- E2: =IF(B2=1, "SI", "")
- E3: =IF(B3=1, "SI", "")
- E4: =IF(B4=1, "SI", "")

- E5: =IF(B5=1, "SI", "")
- E6: =IF(B6=1, "SI", "")
- E7: =IF(B7=1, "SI", "")
- E8: =IF(B8=1, "SI", "")
- E9: =IF(B9=1, "SI", "")
- E10: =IF(B10=1, "SI", "")
- E11: =IF(B11=1, "SI", "")

**Fórmula ejemplo:** =IF(B2=1, "SI", "")

**Desglose completo:**

1. IF( - Inicia la función condicional
2. B2=1 - Condición que verifica si el valor en B2 es igual a 1
  - o = es el operador de comparación de igualdad
3. , - Separador de argumentos
4. "SI" - Valor si verdadero (muestra el texto "SI")
  - o Las comillas "" delimitan texto literal
5. , - Separador de argumentos
6. "" - Valor si falso (muestra texto vacío)
7. ) - Cierra la función IF

**Explicación detallada:**

1. Condición: B2=1 - Verifica si el valor en B2 es igual a 1
  - o Operador =: comparación de igualdad
2. Verdadero: "SI" - Muestra el texto SI si es concurso tipo 1
  - o Comillas "" delimitan texto literal
3. Falso: "" - Muestra texto vacío si no es tipo 1
4. Propósito: Marcar concursos específicos (tipo 1) con "SI"

## Columna F (Reloj)

### Celdas con Formulas:

- F2: =IF(E2="", "SI", "NO")
- F3: =IF(E3="", "SI", "NO")
- F4: =IF(E4="", "SI", "NO")
- F5: =IF(E5="", "SI", "NO")
- F6: =IF(E6="", "SI", "NO")
- F7: =IF(E7="", "SI", "NO")
- F8: =IF(E8="", "SI", "NO")

**Fórmula ejemplo:** =IF(E2="", "SI", "NO")

### Desglose completo:

1. IF( - Inicia la función condicional
2. E2="" - Condición que verifica si E2 está vacío
  - o = comparación de igualdad
  - o "" representa texto vacío
3. , - Separador de argumentos
4. "SI" - Valor si verdadero (E2 está vacío)
5. , - Separador de argumentos
6. "NO" - Valor si falso (E2 no está vacío)
7. ) - Cierra la función IF

### Explicación detallada:

1. Condición: E2="" - Verifica si E2 está vacío
  - o "" representa texto vacío
2. Verdadero: "SI" - Si E está vacío (no es concurso tipo 1)
3. Falso: "NO" - Si E muestra "SI" (es concurso tipo 1)

4. Propósito: Crear un indicador inverso al de la columna E

### Columna G (Premio extra)

#### Celdas con Formulas:

- G2: =IF(C2>100, "Viaje a París", "Otra vez será")
- G3: =IF(C3>100, "Viaje a París", "Otra vez será")
- G4: =IF(C4>100, "Viaje a París", "Otra vez será")
- G5: =IF(C5>100, "Viaje a París", "Otra vez será")
- G6: =IF(C6>100, "Viaje a París", "Otra vez será")
- G7: =IF(C7>100, "Viaje a París", "Otra vez será")
- G8: =IF(C8>100, "Viaje a París", "Otra vez será")

Fórmula ejemplo: =IF(C2>100, "Viaje a París", "Otra vez será")

#### Desglose completo:

1. IF( - Inicia la función condicional
2. C2>100 - Condición que verifica si puntos > 100
3. , - Separador de argumentos
4. "Viaje a París" - Premio si cumple la condición
5. , - Separador de argumentos
6. "Otra vez será" - Mensaje alternativo si no cumple
7. ) - Cierra la función IF

#### Explicación detallada:

1. Condición: C2>100 - Verifica si puntos superan 100
2. Verdadero: "Viaje a París" - Premio por alto rendimiento
3. Falso: "Otra vez será" - Mensaje alternativo
4. Propósito: Asignar premios especiales según puntuación

## Fila 16 (€ ganados)

### Fórmulas para cada participante:

- C16 (Juan López): =IF(C15>100, C15\*10, 0)
- D16 (Sara García): =IF(D15>100, D15\*10, 0)
- E16 (Juan López): =IF(E15>100, E15\*10, 0)
- F16 (Rebeca Ferrer): =IF(F15>100, F15\*10, 0)
- G16 (Luis Atienza): =IF(G15>100, G15\*10, 0)
- H16 (Rebeca Ferrer): =IF(H15>100, H15\*10, 0)

**Fórmula para Juan López (C16):** =IF(C15>100, C15\*10, 0)

### Desglose completo:

1. IF( - Inicia la función condicional
2. C15>100 - Condición que verifica si los puntos en C15 superan 100
  - o > operador "mayor que"
  - o C15 referencia a los puntos del participante
3. , - Separador de argumentos
4. C15\*10 - Valor si verdadero:
  - o Multiplica puntos por 10 (ej:  $150 \times 10 = 1500\text{€}$ )
  - o \* operador de multiplicación
5. , - Separador de argumentos
6. 0 - Valor si falso (asigna 0€ cuando puntos  $\leq 100$ )
7. ) - Cierra la función IF

### Explicación detallada:

- Verifica si los puntos (fila 15) son mayores a 100
- Si es verdadero: multiplica los puntos por 10 (ej: 150 puntos  $\times 10 = 1500\text{€}$ )
- Si es falso: asigna 0€

- El umbral de 100 puntos explica por qué Juan López (60pts) y Rebeca Ferrer (30pts) tienen 0€

### Fila 17 (Premio extra)

#### Fórmulas para cada participante:

- C17 (Juan López): =IF(C15>50, IF(C15<=100, "Reloj", "Agenda"), "")
- D17 (Sara García): =IF(D15>50, IF(D15<=100, "Reloj", "Agenda"), "")
- E17 (Juan López): =IF(E15>50, IF(E15<=100, "Reloj", "Agenda"), "")
- F17 (Rebeca Ferrer): =IF(F15>50, IF(F15<=100, "Reloj", "Agenda"), "")
- G17 (Luis Atienza): =IF(G15>50, IF(G15<=100, "Reloj", "Agenda"), "")
- H17 (Rebeca Ferrer): =IF(H15>50, IF(H15<=100, "Reloj", "Agenda"), "")

**Fórmula para Luis Atienza (G17):** =IF(G15>50, IF(G15<=100, "Reloj", "Agenda"), "")

Desglose completo:

1. IF( - Primer condicional
2. G15>50 - Verifica elegibilidad mínima (puntos > 50)
3. , - Separador
4. IF(G15<=100, "Reloj", "Agenda") - Segundo condicional anidado:
  - o G15<=100 verifica rango medio
    - <= operador "menor o igual que"
  - o "Reloj" para  $50 < \text{puntos} \leq 100$
  - o "Agenda" para puntos > 100
5. , - Separador
6. "" - Valor si falso (texto vacío si puntos  $\leq 50$ )
7. ) - Cierra el primer IF

#### Explicación detallada:

- Primer IF verifica si puntos > 50 (elegibilidad para premio)

- Segundo IF anidado diferencia entre:
  - "Reloj" para  $50 < \text{puntos} \leq 100$
  - "Agenda" para  $\text{puntos} > 100$
- Si  $\text{puntos} \leq 50$ : muestra texto vacío ("")
- Esto explica por qué:
  - Juan López (60pts) y Luis Atienza (90pts) tienen "Reloj"
  - Sara García (150pts) y otros con 120pts tienen "Agenda"

## Fila 18 (Valor premio extra)

### Fórmulas para cada participante:

- C18 (Juan López): =IF(C17="Reloj", "60€", IF(C17="Agenda", "180\$", ""))
- D18 (Sara García): =IF(D17="Reloj", "60€", IF(D17="Agenda", "180\$", ""))
- E18 (Juan López): =IF(E17="Reloj", "60€", IF(E17="Agenda", "180\$", ""))
- F18 (Rebeca Ferrer): =IF(F17="Reloj", "60€", IF(F17="Agenda", "180\$", ""))
- G18 (Luis Atienza): =IF(G17="Reloj", "60€", IF(G17="Agenda", "180\$", ""))
- H18 (Rebeca Ferrer): =IF(H17="Reloj", "60€", IF(H17="Agenda", "180\$", ""))

**Fórmula para Sara García (D18):** =IF(D17="Reloj", "60€", IF(D17="Agenda", "180\$", ""))

### Desglose completo:

1. IF( - Primer condicional
2. D17="Reloj" - Verifica si premio es "Reloj"
  - = operador de igualdad
3. , - Separador
4. "60€" - Valor si verdadero (asigna valor en euros)
5. , - Separador
6. IF(D17="Agenda", "180\$", "") - Segundo condicional anidado:

- D17="Agenda" verifica segundo tipo de premio
- "180\$" valor si verdadero (en dólares)
- "" texto vacío si no es "Reloj" ni "Agenda"

7. ) - Cierra el primer IF

**Explicación detallada:**

- IF anidado que verifica el tipo de premio (fila 17)
- Asigna:
  - "60€" para premios "Reloj"
  - "180\$" para premios "Agenda"
  - Texto vacío si no hay premio
- Notar el uso de diferentes símbolos monetarios (€ y \$)

	A	B	C	D	E	F	G	H	I	J	K
1	<b>Modelo vendido</b>	<b>Precio base</b>	<b>Forma pago</b>	<b>Descuento</b>	<b>Precio total</b>						
2	Mercedes 321	15060	Aplazado		15060						
3	Ford 202	7230	Al contado	361.5	6868.5						
4	Peugeot 105	7230	Al contado	361.5	6868.5						
5	Ford 202	7230	Al contado	361.5	6868.5						
6	Mercedes 321	15060	Aplazado		15060						
7	Peugeot 105	7230	Al contado	361.5	6868.5						
8	Mercedes 321	15060	Aplazado		15060						
9											
10											
11											
12											
13											
14											
15											

Copiar

Configuración

Acerca de

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONROTEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FA880463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

## Hoja "Modelos Autos 2025"

### Columna B (Precio base)

#### Celdas con Fórmulas:

- B2: =IF(A2="Mercedes 321", 15060, 7230)
- B3: =IF(A3="Mercedes 321", 15060, 7230)
- B4: =IF(A4="Mercedes 321", 15060, 7230)
- B5: =IF(A5="Mercedes 321", 15060, 7230)
- B6: =IF(A6="Mercedes 321", 15060, 7230)
- B7: =IF(A7="Mercedes 321", 15060, 7230)
- B8: =IF(A8="Mercedes 321", 15060, 7230)

**Fórmula ejemplo:** =IF(A2="Mercedes 321", 15060, 7230)

#### Desglose completo:

1. IF( - Inicia la función condicional
2. A2="Mercedes 321" - Compara el modelo con texto exacto

3. , - Separador de argumentos
4. 15060 - Precio para Mercedes (valor hardcodeado)
5. , - Separador de argumentos
6. 7230 - Precio para otros modelos (valor hardcodeado)
7. ) - Cierra la función IF

#### **Explicación detallada:**

1. Condición: A2="Mercedes 321" - Compara el modelo exacto
2. Verdadero: 15060 - Precio para Mercedes
3. Falso: 7230 - Precio para otros modelos
4. Problema: Valores hardcodeados dificultan actualizaciones

#### **Columna C (Forma pago)**

#### **Celdas con Fórmulas:**

- C2: =IF(B2=15060, "Aplazado", "Al contado")
- C3: =IF(B3=15060, "Aplazado", "Al contado")
- C4: =IF(B4=15060, "Aplazado", "Al contado")
- C5: =IF(B5=15060, "Aplazado", "Al contado")
- C6: =IF(B6=15060, "Aplazado", "Al contado")
- C7: =IF(B7=15060, "Aplazado", "Al contado")
- C8: =IF(B8=15060, "Aplazado", "Al contado")

**Fórmula ejemplo:** =IF(B2=15060, "Aplazado", "Al contado")

#### **Desglose completo:**

1. **IF( - Inicia la función condicional**
2. **B2=15060 - Compara con valor hardcodeado**
3. , - Separador de argumentos

**4. "Aplazado" - Método de pago para Mercedes**

**5. , - Separador de argumentos**

**6. "Al contado" - Método de pago para otros**

**7. ) - Cierra la función IF**

#### **Explicación detallada:**

1. Condición: B2=15060 - Verifica precio específico
2. Verdadero: "Aplazado" - Pago diferido para Mercedes
3. Falso: "Al contado" - Pago inmediato para otros
4. Dependencia: Basada en valor hardcodeado (15060)

#### **Columna D (Descuento)**

#### **Celdas con Fórmulas:**

- D2: =IF(C2="Al contado", B2\*0.05, "")
- D3: =IF(C3="Al contado", B3\*0.05, "")
- D4: =IF(C4="Al contado", B4\*0.05, "")
- D5: =IF(C5="Al contado", B5\*0.05, "")
- D6: =IF(C6="Al contado", B6\*0.05, "")
- D7: =IF(C7="Al contado", B7\*0.05, "")
- D8: =IF(C8="Al contado", B8\*0.05, "")

**Fórmula ejemplo:** =IF(C2="Al contado", B2\*0.05, "")

#### **Desglose completo:**

1. IF( - Inicia la función condicional
2. C2="Al contado" - Verifica método de pago
3. , - Separador de argumentos
4. B2\*0.05 - Calcula 5% del precio base

- \* operador de multiplicación
  - 0.05 representa 5% (5/100)
5. , - Separador de argumentos
  6. "" - Texto vacío si no aplica
  7. ) - Cierra la función IF

#### **Explicación detallada:**

1. Condición: C2="Al contado" - Verifica método de pago
2. Verdadero: B2\*0.05 - Calcula 5% de descuento
  - $0.05 = 5\% (5/100)$
3. Falso: "" - Sin descuento (texto vacío)
4. Propósito: Aplicar descuento por pago al contado

#### **Columna E (Precio total)**

#### **Celdas con Fórmulas:**

- E2: =IF(D2<>"", B2-D2, B2)
- E3: =IF(D3<>"", B3-D3, B3)
- E4: =IF(D4<>"", B4-D4, B4)
- E5: =IF(D5<>"", B5-D5, B5)
- E6: =IF(D6<>"", B6-D6, B6)
- E7: =IF(D7<>"", B7-D7, B7)
- E8: =IF(D8<>"", B8-D8, B8)

**Fórmula ejemplo:** =IF(D2<>"", B2-D2, B2)

#### **Desglose completo:**

1. IF( - Inicia la función condicional
2. D2<>"" - Verifica si hay descuento

- ◊ operador "diferente de"
- "" texto vacío

3. , - Separador de argumentos

4. B2-D2 - Resta descuento al precio

- - operador de resta

5. , - Separador de argumentos

6. B2 - Precio base si no hay descuento

7. ) - Cierra la función IF

#### **Explicación detallada:**

1. Condición: D2◊"" - Verifica si hay descuento
  - Operador ◊: "diferente de"
2. Verdadero: B2-D2 - Resta descuento al precio
3. Falso: B2 - Mantiene precio original
4. Propósito: Calcular precio final considerando descuentos

	A	B	C	D	E	F	G	H
1	<b>Cantidad</b>	<b>Porcentaje</b>	<b>Respuesta</b>	<b>Solución</b>				
2	<b>600000</b>	<b>25%</b>	<b>150000</b>	<b>Muy bien</b>				
3	<b>5600</b>	<b>60%</b>	<b>3400</b>	<b>Dedíquese a otra cosa</b>				
4	<b>740</b>	<b>95%</b>	<b>650</b>	<b>Dedíquese a otra cosa</b>				
5	<b>50000</b>	<b>5%</b>	<b>2500</b>	<b>Muy bien</b>				
6								
7								
8								
9								
10								
11								
12								
13								

## Hoja "Problema 3"

### Columna D (Solución)

#### Celdas con Fórmulas:

- D2: =IF(C2=ROUND(A2\*B2,0), "Muy bien", "Dedíquese a otra cosa")
- D3: =IF(C3=ROUND(A3\*B3,0), "Muy bien", "Dedíquese a otra cosa")
- D4: =IF(C4=ROUND(A4\*B4,0), "Muy bien", "Dedíquese a otra cosa")
- D5: =IF(C5=ROUND(A5\*B5,0), "Muy bien", "Dedíquese a otra cosa")

**Fórmula ejemplo:** =IF(C2=ROUND(A2\*B2,0), "Muy bien", "Dedíquese a otra cosa")

#### Desglose completo:

1. IF( - Inicia la función condicional
2. C2=ROUND(A2\*B2,0) - Comparación compleja:
  - o A2\*B2 - Multiplicación de valores
  - o ROUND(...,0) - Redondea resultado a 0 decimales
  - o = compara con valor en C2

3. , - Separador de argumentos
4. "Muy bien" - Mensaje si cálculo es correcto
5. , - Separador de argumentos
6. "Dedíquese a otra cosa" - Mensaje si hay error
7. ) - Cierra la función IF

**Explicación detallada:**

1. Parte interna: ROUND(A2\*B2,0)
  - A2\*B2: Multiplica valores
  - ROUND(...,0): Redondea resultado a 0 decimales
2. Condición: C2=... - Compara con valor redondeado
3. Verdadero: "Muy bien" - Si el cálculo es correcto
4. Falso: "Dedíquese a otra cosa" - Si hay error
5. Propósito: Validar resultados de multiplicaciones

Subredes V  
Packet Tracer

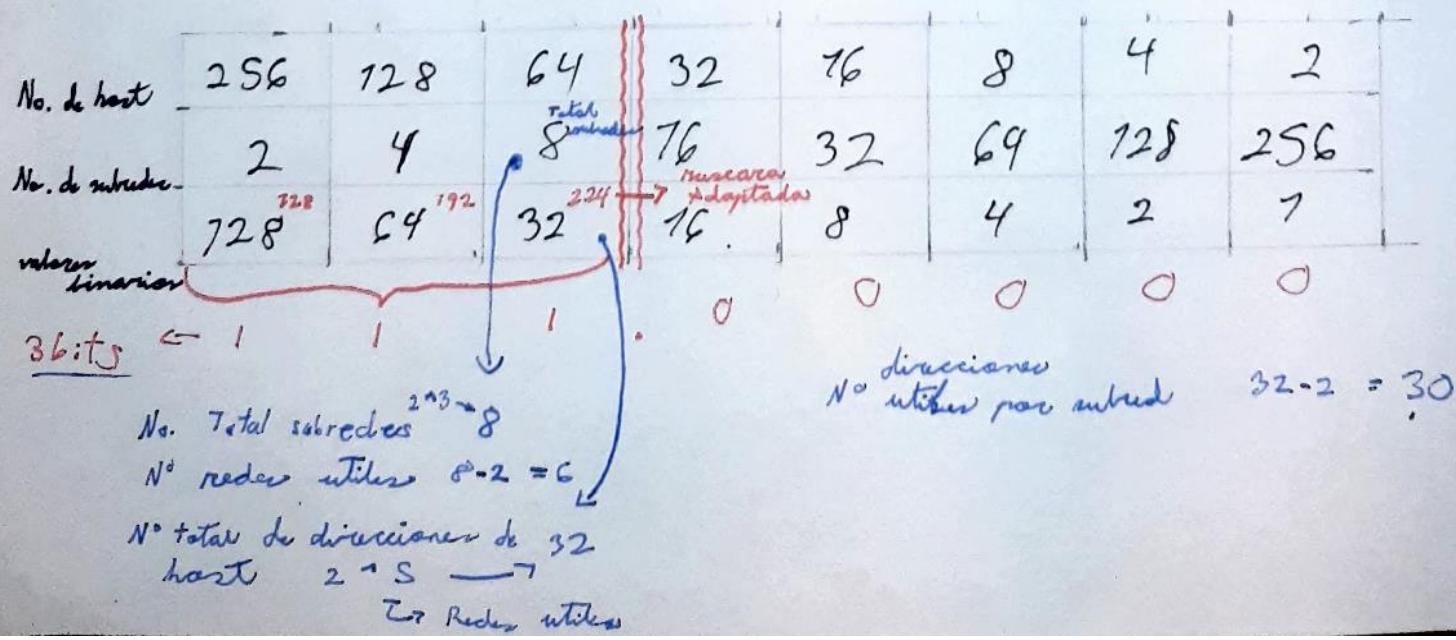
**Problema 2: Resuelva utilizando Subredes y packet tracer.**

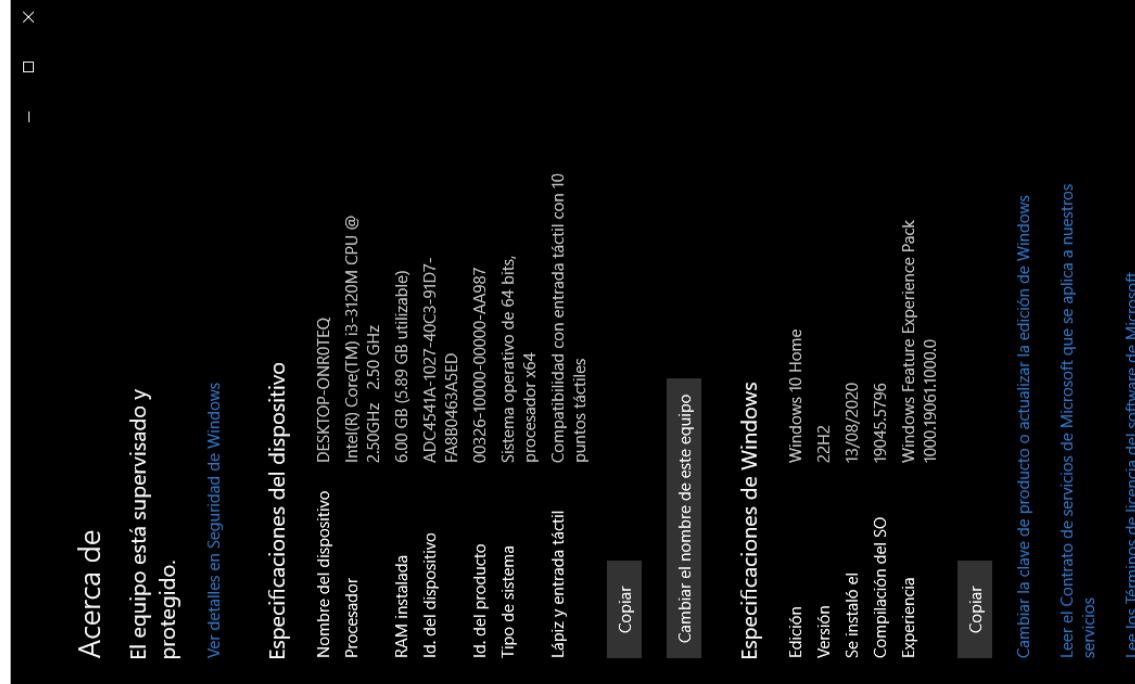
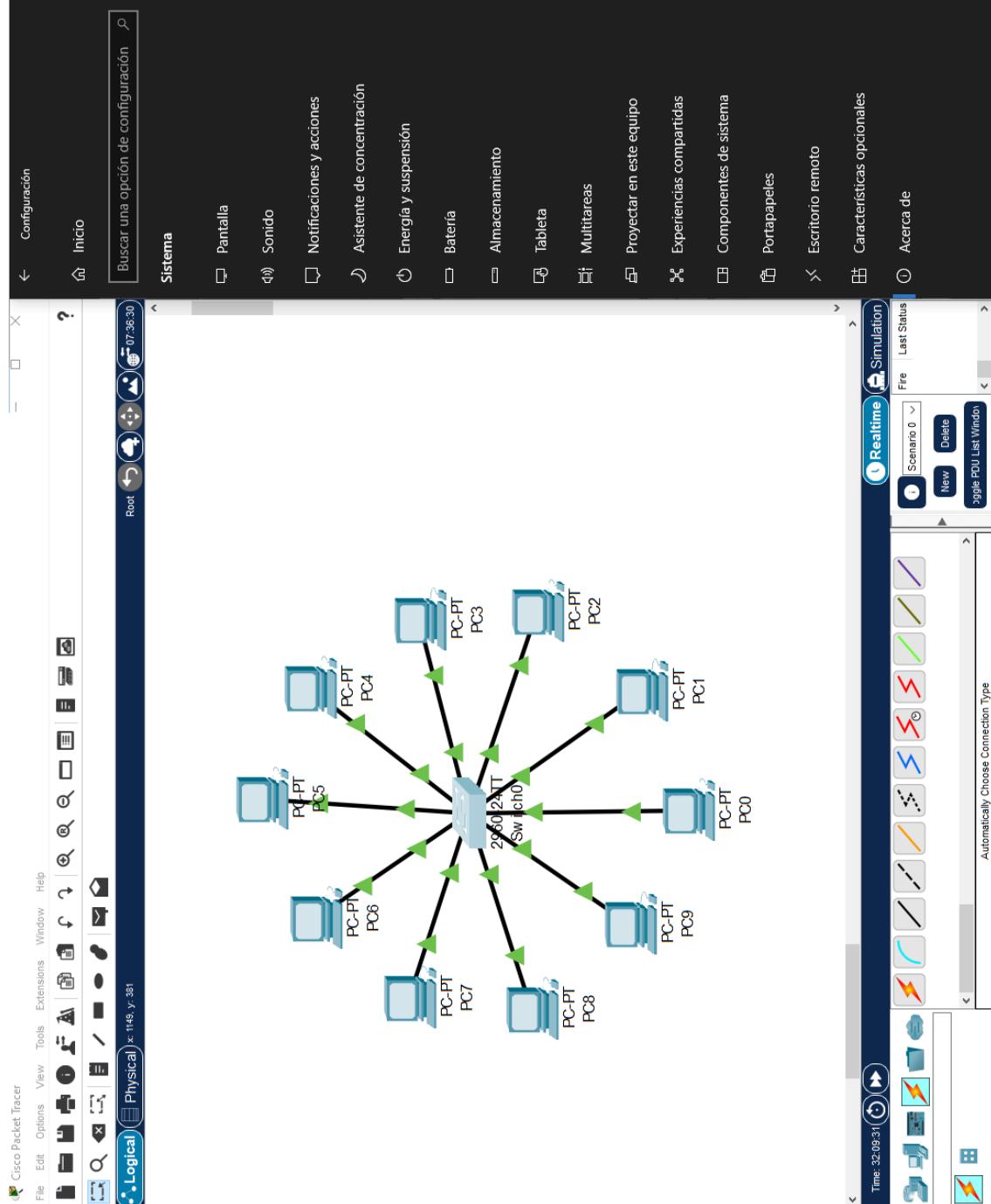
Se requiere la implementación de una red en una empresa la cual requiere el menor consumos de IP para que pueda funcionar de manera óptima.

- Determinar utilizando matemática de redes lo necesario para crear la red.

Nº de subredes útiles necesarias:	5
Dirección de Red:	218.35.50.0
Clase:	C Range C 192-223
Máscara de Subred (por defecto):	255.255.255.0
Máscara de Subred (adaptada):	255.255.255.224
Nº total de subredes:	8
Nº de redes útiles:	6
Nº total de direcciones de host:	32
Nº de direcciones útiles:	30
Nº de bits cogidos:	3

- Diseñar la red utilizando CISCO PACKET TRACER utilice al menos 10 computadoras para verificar el funcionamiento de la red.





**Cisco Packet Tracer**

File Edit Options View Tools Extensions Window Help

Logical Physical x: 792, y: 573

Root 05:09:00

Time: 32:04:47

Automatically Choose Connection Type

Realtime Simulation

Characterísticas opcionales

Acerca de

El equipo está supervisado y protegido.

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo DESKTOP-ONR0TEQ

Procesador Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz

RAM instalada 6.00 GB (5.89 GB utilizable)

Id. del dispositivo ADC4541A-1027-40C3-91D7-FAB80463A5ED

Id. del producto 00326-10000-00000-AA987

Tipo de sistema Sistema operativo de 64 bits, procesador x64

Lápiz y entrada táctil Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar Cambiar el nombre de este equipo

**Configuración**

Inicio Buscar una opción de configuración

Sistema Pantalla Sonido Notificaciones y acciones Asistente de concentración Energía y suspensión Batería Almacenamiento Tableta Multitareas Proyectar en este equipo Experiencias compartidas

Edición Windows 10 Home

Versión 22H2

Se instaló el 13/08/2020

Compilación del SO 19045.5796

Experiencia Windows Feature Experience Pack 1000.19061.1000.0

Copiar Cambiar la clave de producto o actualizar la edición de Windows

Leer el Contrato de servicios de Microsoft que se aplica a nuestros servicios

Lee los Términos de licencia del software de Microsoft

**Cisco Packet Tracer**

File Edit Options View Tools Extensions Window Help

Logical Physical x: 890, y: 538

Root 05:41:30

Time: 32:05:50

Automatically Choose Connection Type

Realtime Simulation

Characterísticas opcionales

Acerca de

El equipo está supervisado y protegido.

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo DESKTOP-ONR0TEQ

Procesador Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz

RAM instalada 6.00 GB (5.89 GB utilizable)

Id. del dispositivo ADC4541A-1027-40C3-91D7-FAB80463A5ED

Id. del producto 00326-10000-00000-AA987

Tipo de sistema Sistema operativo de 64 bits, procesador x64

Lápiz y entrada táctil Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar Cambiar el nombre de este equipo

**Configuración**

Inicio Buscar una opción de configuración

Sistema Pantalla Sonido Notificaciones y acciones Asistente de concentración Energía y suspensión Batería Almacenamiento Tableta Multitareas Proyectar en este equipo Experiencias compartidas

Edición Windows 10 Home

Versión 22H2

Se instaló el 13/08/2020

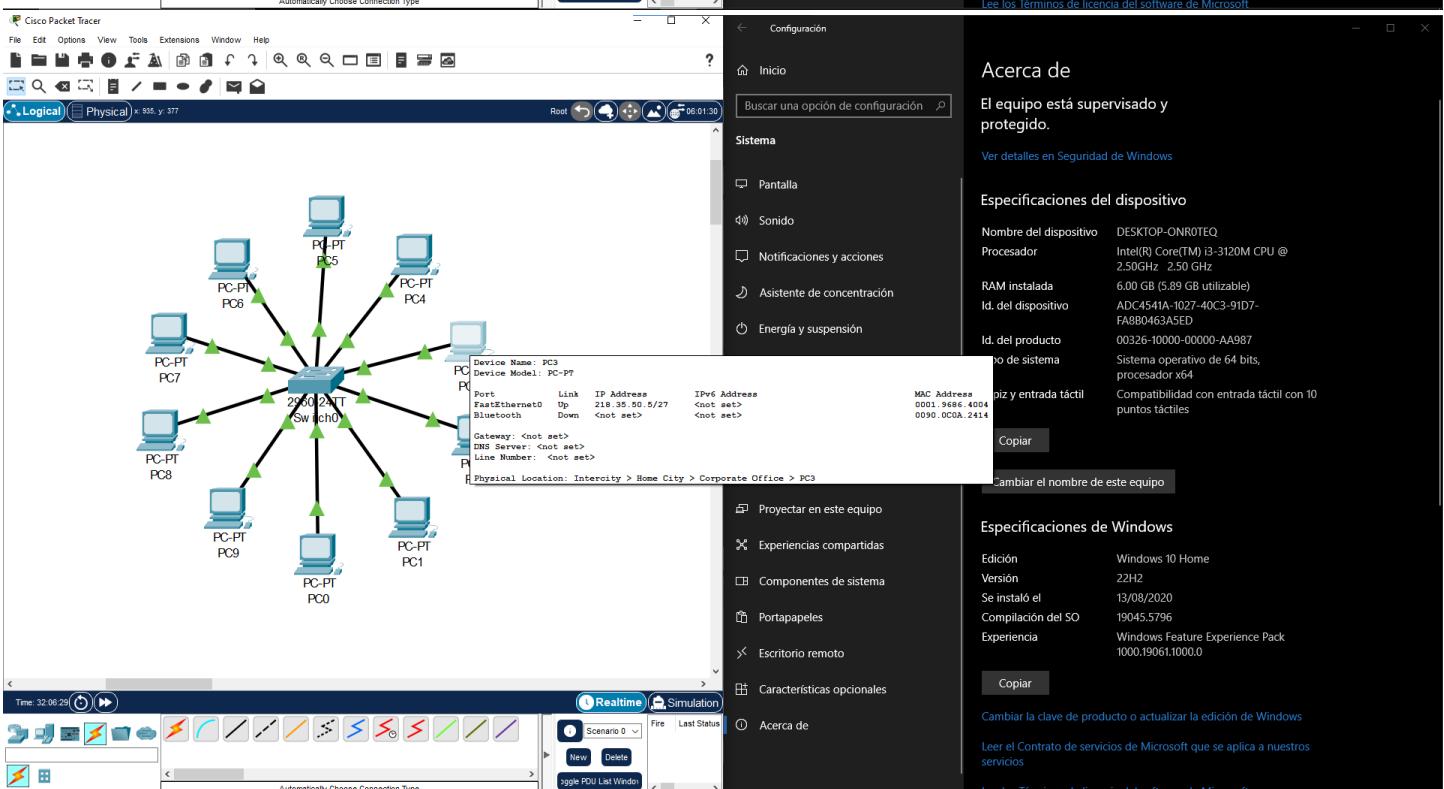
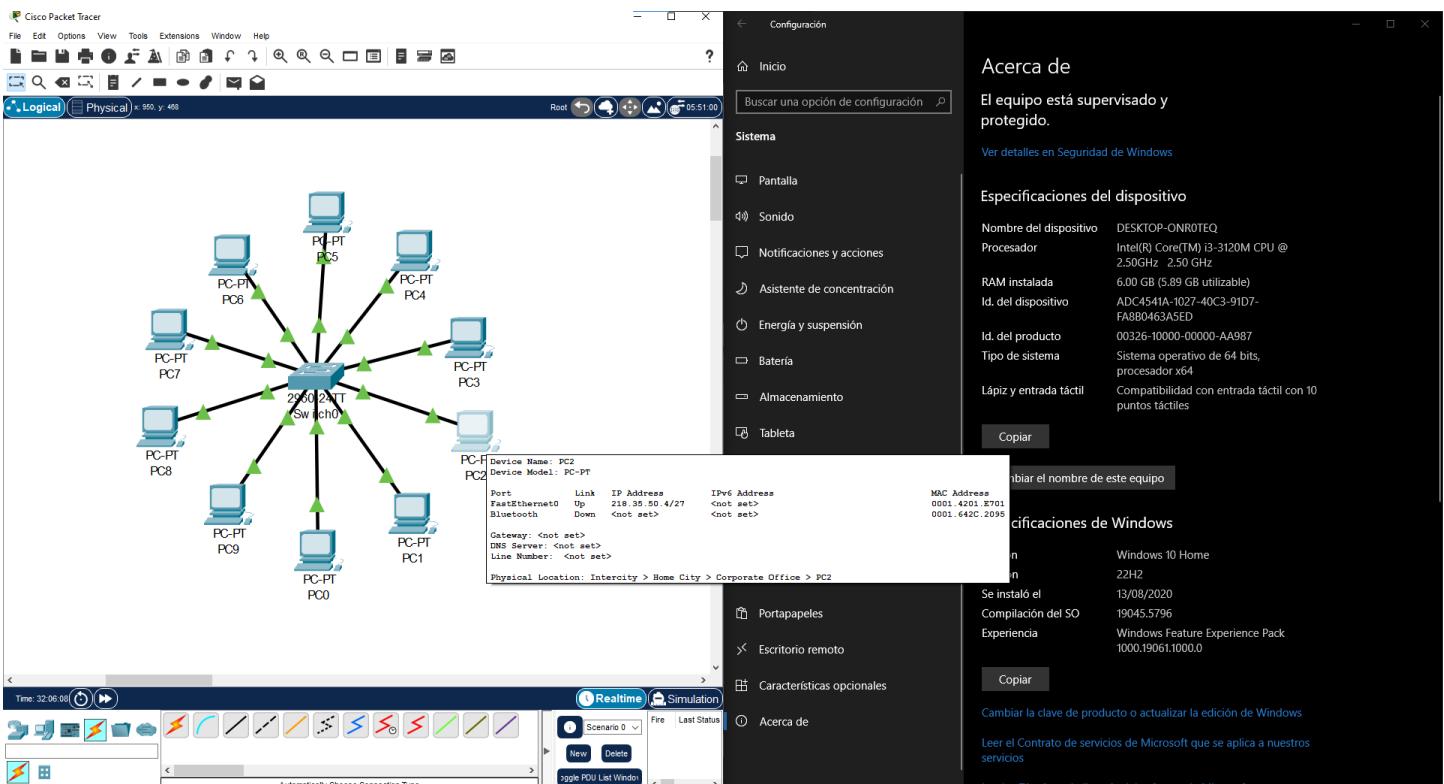
Compilación del SO 19045.5796

Experiencia Windows Feature Experience Pack 1000.19061.1000.0

Copiar Cambiar la clave de producto o actualizar la edición de Windows

Leer el Contrato de servicios de Microsoft que se aplica a nuestros servicios

Lee los Términos de licencia del software de Microsoft



**Cisco Packet Tracer**

File Edit Options View Tools Extensions Window Help

Logical Physical x: 899, y: 299

Root 06:38:00

Time: 32:07:24

Realtime Simulation

Automatically Choose Connection Type

Device Name: PC4  
Device Model: PC-PT  
Port FastEthernet0 Link IP Address 218.35.50.6/27 IPv6 Address <not set>  
MAC Address 000B.BE14.7CBA  
RAM instalada 6.00 GB (5.89 GB utilizable)  
Id. del dispositivo ADC4541A-1027-40C3-91D7-FAB80463A5ED  
Id. del producto 00326-10000-00000-AA987  
Tipo de sistema Sistema operativo de 64 bits, procesador x64  
Lápiz y entrada táctil Compatibilidad con entrada táctil con 10 puntos táctiles

Almacenamiento

Tableta

Multitareas

Proyectar en este equipo

Experiencias compartidas

Componentes de sistema

Portapapeles

Escrivtorio remoto

Características opcionales

Cambiar el nombre de este equipo

**Acerca de**

El equipo está supervisado y protegido.

Ver detalles en Seguridad de Windows

**Especificaciones del dispositivo**

Nombre del dispositivo DESKTOP-ONR0TEQ  
Procesador Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz  
RAM instalada 6.00 GB (5.89 GB utilizable)  
Id. del dispositivo ADC4541A-1027-40C3-91D7-FAB80463A5ED  
Id. del producto 00326-10000-00000-AA987  
Tipo de sistema Sistema operativo de 64 bits, procesador x64  
Lápiz y entrada táctil Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar

**Especificaciones de Windows**

Edición Windows 10 Home  
Versión 22H2  
Se instaló el 13/08/2020  
Compilación del SO 19045.5796  
Experiencia Windows Feature Experience Pack 1000.19061.1000.0

Cambiar la clave de producto o actualizar la edición de Windows

Ler el Contrato de servicios de Microsoft que se aplica a nuestros servicios

Lee los Términos de licencia del software de Microsoft

**Cisco Packet Tracer**

File Edit Options View Tools Extensions Window Help

Logical Physical x: 812, y: 265

Root 06:38:00

Time: 32:07:39

Realtime Simulation

Automatically Choose Connection Type

Device Name: PC5  
Device Model: PC-PT  
Port FastEthernet0 Link IP Address 218.35.50.7/27 IPv6 Address <not set>  
MAC Address 0001.649B.C212  
RAM instalada 6.00 GB (5.89 GB utilizable)  
Id. del dispositivo ADC4541A-1027-40C3-91D7-FAB80463A5ED  
Id. del producto 00326-10000-00000-AA987  
Tipo de sistema Sistema operativo de 64 bits, procesador x64  
Lápiz y entrada táctil Compatibilidad con entrada táctil con 10 puntos táctiles

Batería

Almacenamiento

Tableta

Multitareas

Proyectar en este equipo

Experiencias compartidas

Componentes de sistema

Portapapeles

Escrivtorio remoto

Características opcionales

Cambiar el nombre de este equipo

**Acerca de**

El equipo está supervisado y protegido.

Ver detalles en Seguridad de Windows

**Especificaciones del dispositivo**

Nombre del dispositivo DESKTOP-ONR0TEQ  
Procesador Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz  
RAM instalada 6.00 GB (5.89 GB utilizable)  
Id. del dispositivo ADC4541A-1027-40C3-91D7-FAB80463A5ED  
Id. del producto 00326-10000-00000-AA987  
Tipo de sistema Sistema operativo de 64 bits, procesador x64  
Lápiz y entrada táctil Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar

**Especificaciones de Windows**

Edición Windows 10 Home  
Versión 22H2  
Se instaló el 13/08/2020  
Compilación del SO 19045.5796  
Experiencia Windows Feature Experience Pack 1000.19061.1000.0

Cambiar la clave de producto o actualizar la edición de Windows

Ler el Contrato de servicios de Microsoft que se aplica a nuestros servicios

Lee los Términos de licencia del software de Microsoft

**Cisco Packet Tracer**

File Edit Options View Tools Extensions Window Help

Logical Physical x: 718, y: 300

Root 05:48:30

Physical Location: InterCity > Home City > Corporate Office > PC6

Device Name: PC6  
Device Model: PC-PT  
Port FastEthernet0 Link Up IP Address 218.35.50.8/27  
Bluetooth Down <not set>  
Gateway: <not set>  
DNS Server: <not set>  
Line Number: <not set>

MAC Address 0090.2197.23E9  
IPv6 Address 00D0.D3C3.4242

Switch0

PC-PT PC7  
PC-PT PC8  
PC-PT PC9  
PC-PT PC0  
PC-PT PC1  
PC-PT PC2

Time: 32:08:00

Realtime Simulation

Automatically Choose Connection Type

Scenarios: Scenario 0 New Delete Toggle PDU Last Window

**Acerca de**

El equipo está supervisado y protegido.

Ver detalles en Seguridad de Windows

**Especificaciones del dispositivo**

Nombre del dispositivo	DESKTOP-ONR0TEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FAB80463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar Cambiar el nombre de este equipo

**Especificaciones de Windows**

Edición	Windows 10 Home
Versión	22H2
Se instaló el	13/08/2020
Compilación del SO	19045.5796
Experiencia	Windows Feature Experience Pack 1000.19061.1000.0

Copiar Cambiar la clave de producto o actualizar la edición de Windows

Leyendo el Contrato de servicios de Microsoft que se aplica a nuestros servicios

Lee los Términos de licencia del software de Microsoft

**Cisco Packet Tracer**

File Edit Options View Tools Extensions Window Help

Logical Physical x: 658, y: 370

Root 05:58:30

Physical Location: InterCity > Home City > Corporate Office > PC7

Device Name: PC7  
Device Model: PC-PT  
Port FastEthernet0 Link Up IP Address 218.35.50.9/27  
Bluetooth Down <not set>  
Gateway: <not set>  
DNS Server: <not set>  
Line Number: <not set>

MAC Address 0004.9A89.BAAC  
IPv6 Address 0090.A35E.2A11

PC-PT PC6  
PC-PT PC4  
PC-PT PC9  
PC-PT PC0  
PC-PT PC1

Time: 32:08:19

Realtime Simulation

Automatically Choose Connection Type

Scenarios: Scenario 0 New Delete Toggle PDU Last Window

**Acerca de**

El equipo está supervisado y protegido.

Ver detalles en Seguridad de Windows

**Especificaciones del dispositivo**

Nombre del dispositivo	DESKTOP-ONR0TEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FAB80463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar Cambiar el nombre de este equipo

**Especificaciones de Windows**

Edición	Windows 10 Home
Versión	22H2
Se instaló el	13/08/2020
Compilación del SO	19045.5796
Experiencia	Windows Feature Experience Pack 1000.19061.1000.0

Copiar Cambiar la clave de producto o actualizar la edición de Windows

Leyendo el Contrato de servicios de Microsoft que se aplica a nuestros servicios

Lee los Términos de licencia del software de Microsoft

**Cisco Packet Tracer**

File Edit Options View Tools Extensions Window Help

Logical Physical x: 657, y: 466

Root 07:05:30

Device Name: PC8  
Device Model: PC-PT  
Port FastEthernet0 Link Up IP Address 218.35.50.10/27 IPv6 Address <not set>  
Bluetooth Down <not set><not set>  
Gateway: <not set>  
DNS Server: <not set>  
Line Number: <not set>  
Physical Location: Intercity > Home City > Corporate Office > PC8

PC-PT PC0

Realtime Simulation

Automatically Choose Connection Type Scenario 0 New Delete Toggle PDU Last Window

**Acerca de**  
El equipo está supervisado y protegido.  
Ver detalles en Seguridad de Windows

**Especificaciones del dispositivo**

Nombre del dispositivo	DESKTOP-ONR0TEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FAB80463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar Cambiar el nombre de este equipo

**Especificaciones de Windows**

Edición	Windows 10 Home
Versión	22H2
Se instaló el	13/08/2020
Compilación del SO	19045.5796
Experiencia	Windows Feature Experience Pack 1000.19061.1000.0

Copiar Cambiar la clave de producto o actualizar la edición de Windows  
Ler el Contrato de servicios de Microsoft que se aplica a nuestros servicios  
Lee los Términos de licencia del software de Microsoft

**Cisco Packet Tracer**

File Edit Options View Tools Extensions Window Help

Logical Physical x: 716, y: 523

Root 07:17:30

Device Name: PC9  
Device Model: PC-PT  
Port FastEthernet0 Link Up IP Address 218.35.50.11/27 IPv6 Address <not set>  
Bluetooth Down <not set><not set>  
Gateway: <not set>  
DNS Server: <not set>  
Line Number: <not set>  
Physical Location: Intercity > Home City > Corporate Office > PC9

PC-PT PC0

Realtime Simulation

Automatically Choose Connection Type Scenario 0 New Delete Toggle PDU Last Window

**Acerca de**  
El equipo está supervisado y protegido.  
Ver detalles en Seguridad de Windows

**Especificaciones del dispositivo**

Nombre del dispositivo	DESKTOP-ONR0TEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FAB80463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar Cambiar el nombre de este equipo

**Especificaciones de Windows**

Edición	Windows 10 Home
Versión	22H2
Se instaló el	13/08/2020
Compilación del SO	19045.5796
Experiencia	Windows Feature Experience Pack 1000.19061.1000.0

Copiar Cambiar la clave de producto o actualizar la edición de Windows  
Ler el Contrato de servicios de Microsoft que se aplica a nuestros servicios  
Lee los Términos de licencia del software de Microsoft

**MySQLWorkBench**  
**EXOS.A.**

The screenshot shows the MySQL Workbench interface. In the top-left window, titled 'Query 1', there is a SQL script. Lines 1 through 7 show the creation of a database named 'sistema\_renta\_películas'. Lines 8 through 13 show the selection of this database. Lines 14 through 24 define a table named 'direccion' with columns: 'direccion' (primary key, INT(11)), 'iddireccion' (INT(11)), 'Dirección' (VARCHAR(45)), and 'Ciudad' (VARCHAR(45)). The bottom-left window, titled 'Action Output', shows two log entries: 'CREATE DATABASE IF NOT EXISTS sistema\_renta\_películas' and 'USE sistema\_renta\_películas'. To the right, a 'Configuración' window displays system specifications: Nombre del dispositivo (DESKTOP-ONROTEQ), Procesador (Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz), RAM instalada (6.00 GB (5.89 GB utilizable)), Id. del dispositivo (ADC4541A-1027-40C3-91D7-FA880463A5ED), Id. del producto (00326-10000-00000-AA987), and Tipo de sistema (Sistema operativo de 64 bits, procesador x64). A 'Copiar' button is visible at the bottom of this window.

```

1 -- CREACIÓN DE LA BASE DE DATOS:
2 -- CREATE DATABASE: Comando para crear una nueva base de datos en MySQL
3 -- IF NOT EXISTS: Cláusula condicional que evita errores si la BD ya existe
4 -- sistema_renta_películas: Nombre asignado a nuestra base de datos
5 -- Los paréntesis son opcionales en CREATE DATABASE pero se incluyen para claridad
6 -- El punto y coma (;) marca el final de esta instrucción SQL
7 • CREATE DATABASE IF NOT EXISTS sistema_renta_películas;
8
9 -- SELECCIÓN DE LA BASE DE DATOS:
10 -- USE: Comando para especificar qué base de datos usaremos en las siguientes operaciones
11 -- sistema_renta_películas: Nombre de la base de datos que acabamos de crear
12 -- El punto y coma (;) finaliza esta instrucción
13 • USE sistema_renta_películas;
14
15 -- CREACIÓN DE TABLA dirección:
16 -- CREATE TABLE: Inicia la definición de una nueva tabla
17 -- dirección: Nombre de la tabla que almacenará información de ubicación
18 -- (): Los paréntesis encierran todas las definiciones de columnas
19 -- iddireccion: Primera columna que servirá como clave primaria
20 -- INT(11): Tipo de dato entero con display width de 11 dígitos
21 -- PRIMARY KEY: Especifica que esta columna identificará registros únicos
22 -- Dirección: Segunda columna para almacenar la dirección física
23 -- VARCHAR(45): Tipo de dato para cadenas de texto con máximo 45 caracteres
24 -- Ciudad: Tercera columna para el nombre de la ciudad
<

```

Action	Time	Message
1	21:20:00	CREATE DATABASE IF NOT EXISTS sistema_renta_películas 1 row(s) affected
2	21:20:00	USE sistema_renta_películas 0 row(s) affected

## -- CREACIÓN DE LA BASE DE DATOS:

-- CREATE DATABASE: Comando para crear una nueva base de datos en MySQL

-- IF NOT EXISTS: Cláusula condicional que evita errores si la BD ya existe

-- sistema\_renta\_películas: Nombre asignado a nuestra base de datos

-- Los paréntesis son opcionales en CREATE DATABASE pero se incluyen para claridad

-- El punto y coma (;) marca el final de esta instrucción SQL

`CREATE DATABASE IF NOT EXISTS sistema_renta_películas;`

## -- SELECCIÓN DE LA BASE DE DATOS:

-- USE: Comando para especificar qué base de datos usaremos en las siguientes operaciones

-- sistema\_renta\_películas: Nombre de la base de datos que acabamos de crear

-- El punto y coma (;) finaliza esta instrucción

`USE sistema_renta_películas;`

```

13 • USE sistema_renta_peliculas;
14
15 -- CREACIÓN DE TABLA direccion:
16 -- CREATE TABLE: Inicia la definición de una nueva tabla
17 -- direccion: Nombre de la tabla que almacenará información de ubicación
18 -- (): Los paréntesis encierran todas las definiciones de columnas
19 -- idDireccion: Primera columna que servirá como clave primaria
20 -- INT(11): Tipo de dato entero con display width de 11 dígitos
21 -- PRIMARY KEY: Especifica que esta columna identificará registros únicos
22 -- Direccion: Segunda columna para almacenar la dirección física
23 -- VARCHAR(45): Tipo de dato para cadenas de texto con máximo 45 caracteres
24 -- Ciudad: Tercera columna para el nombre de la ciudad
25 -- Pais: Cuarta columna para el nombre del país
26 -- Cada definición de columna está separada por comas (,)
27 -- El punto y coma (;) finaliza la creación de la tabla
28 • CREATE TABLE direccion (
29     idDireccion INT(11) PRIMARY KEY,
30     Direccion VARCHAR(45),
31     Ciudad VARCHAR(45),
32     Pais VARCHAR(45)
33 );
34
35 -- CREACIÓN DE TABLA categoria:
36 -- CREATE TABLE: Inicia la definición de la tabla de categorías

```

Output:

Time	Action	Message
1 21:20:00	CREATE DATABASE IF NOT EXISTS sistema_renta_peliculas	0 rows(s) affected
2 21:20:00	USE sistema_renta_peliculas	0 rows(s) affected
3 21:21:00	CREATE TABLE direccion ( idDireccion INT(11) PRIMARY KEY, Direccion VARCHAR(45), Ciudad VARCHAR(45), Pais VARCHAR(45) )	0 row(s) affected

## -- CREACIÓN DE TABLA direccion:

-- CREATE TABLE: Inicia la definición de una nueva tabla

-- direccion: Nombre de la tabla que almacenará información de ubicación

-- (): Los paréntesis encierran todas las definiciones de columnas

-- idDireccion: Primera columna que servirá como clave primaria

-- INT(11): Tipo de dato entero con display width de 11 dígitos

-- PRIMARY KEY: Especifica que esta columna identificará registros únicos

-- Direccion: Segunda columna para almacenar la dirección física

-- VARCHAR(45): Tipo de dato para cadenas de texto con máximo 45 caracteres

-- Ciudad: Tercera columna para el nombre de la ciudad

-- Pais: Cuarta columna para el nombre del país

-- Cada definición de columna está separada por comas (,) El punto y coma (;) finaliza la creación de la tabla

CREATE TABLE direccion (

idDireccion INT(11) PRIMARY KEY,

Direccion VARCHAR(45),

Ciudad VARCHAR(45),

Pais VARCHAR(45)

);

MySQL Workbench

PreLaboratorio 1 - Warning .. x

File Edit View Query Database Server Tools Scripting Help

Query 1 x

25 -- País: Cuarta columna para el nombre del país  
26 -- Cada definición de columna está separada por comas (,)  
27 -- El punto y coma (;) finaliza la creación de la tabla  
28 • CREATE TABLE dirección (  
29 idDirección INT(11) PRIMARY KEY,  
30 Dirección VARCHAR(45),  
31 Ciudad VARCHAR(45),  
32 País VARCHAR(45)  
33 );  
34  
35 -- CREACIÓN DE TABLA categoría:  
36 -- CREATE TABLE: Inicia la definición de la tabla de categorías  
37 -- categoría: Nombre de la tabla para clasificar películas  
38 -- idCategoria: Clave primaria de tipo entero  
39 -- Nombre: Campo para el nombre de la categoría (texto hasta 45 caracteres)  
40 -- El punto y coma (;) finaliza la instrucción  
41 • CREATE TABLE categoría (  
42 idCategoria INT(11) PRIMARY KEY,  
43 Nombre VARCHAR(45)  
44 );  
45  
46 -- CREACIÓN DE TABLA películas:  
47 -- CREATE TABLE: Inicia la definición de la tabla principal  
48 -- películas: Nombre de la tabla que almacenará información de películas

Output

Acción Output

#	Time	Action	Message
1	21:20:00	CREATE DATABASE IF NOT EXISTS sistema_renta_peliculas	1 row(s) affected
2	21:20:00	USE sistema_renta_peliculas	0 row(s) affected
3	21:21:00	CREATE TABLE dirección ( idDirección INT(11) PRIMARY KEY, Dirección VARCHAR(45), Ciudad VARCHAR(45), País VARCHAR(45) )	0 row(s) affected
4	21:22:36	CREATE TABLE categoría ( idCategoria INT(11) PRIMARY KEY, Nombre VARCHAR(45) )	0 row(s) affected

Configuración

Acerca de

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONRNOTEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FA8B0463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

-- CREACIÓN DE TABLA categoria:

-- CREATE TABLE: Inicia la definición de la tabla de categorías

-- categoria: Nombre de la tabla para clasificar películas

-- idCategoria: Clave primaria de tipo entero

-- Nombre: Campo para el nombre de la categoría (texto hasta 45 caracteres)

-- El punto y coma (;) finaliza la instrucción

```
CREATE TABLE categoria (
```

idCategoria INT(11) PRIMARY KEY,

Nombre VARCHAR(45)

```

46 -- CREACIÓN DE TABLA peliculas:
47 -- CREATE TABLE: Inicia la definición de la tabla principal
48 -- peliculas: Nombre de la tabla que almacenará información de películas
49 -- idPeliculas: Clave primaria de tipo entero
50 -- Nombre: Título de la película (texto hasta 45 caracteres)
51 -- Duracion: Duración en minutos (entero)
52 -- Descripcion: Campo de texto largo para la sinopsis
53 -- Año: Año de lanzamiento (entero)
54 -- Categoria_idCategoria: Clave foránea que relaciona con la tabla categoria
55 -- FOREIGN KEY: Define la relación con otra tabla
56 -- REFERENCES: Especifica la tabla (categoria) y columna (idCategoria) referenciada
57 -- El punto y coma (;) finaliza la creación de la tabla
58 CREATE TABLE peliculas (
59     idPeliculas INT(11) PRIMARY KEY,
60     Nombre VARCHAR(45),
61     Duracion INT(11),
62     Descripcion TEXT,
63     Año INT(11),
64     Categoria_idCategoria INT(11),
65     FOREIGN KEY (Categoria_idCategoria) REFERENCES categoria(idCategoria)
66 );
67
68 -- CREACIÓN DE TABLA cliente:
69 -- CREATE TABLE: Inicia la definición de la tabla de clientes

```

Output:

#	Time	Action	Message
2	21:20:00	USE sistema_renta_peliculas	0 row(s) affected
3	21:21:00	CREATE TABLE direccion ( idDireccion INT(11) PRIMARY KEY, Direccion VARCHAR(45), Ciudad VARCHAR(45), Pais VARCHAR(45) )	0 row(s) affected
4	21:22:36	CREATE TABLE categoria ( idCategoria INT(11) PRIMARY KEY, Nombre VARCHAR(45) )	0 row(s) affected
5	21:23:16	CREATE TABLE peliculas ( idPeliculas INT(11) PRIMARY KEY, Nombre VARCHAR(45), Duracion INT(11), Descripcion TEXT, Año INT(11), Categoria_idCate... )	0 row(s) affected

Configuración

Acerca de

Ver detalles en Seguridad de Windows

#### Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONROTEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FA880463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

[Copiar](#)

-- CREACIÓN DE TABLA peliculas:

-- CREATE TABLE: Inicia la definición de la tabla principal

-- peliculas: Nombre de la tabla que almacenará información de películas

-- idPeliculas: Clave primaria de tipo entero

-- Nombre: Título de la película (texto hasta 45 caracteres)

-- Duracion: Duración en minutos (entero)

-- Descripcion: Campo de texto largo para la sinopsis

-- Año: Año de lanzamiento (entero)

-- Categoria\_idCategoria: Clave foránea que relaciona con la tabla categoria

-- FOREIGN KEY: Define la relación con otra tabla

-- REFERENCES: Especifica la tabla (categoria) y columna (idCategoria) referenciada

-- El punto y coma (;) finaliza la creación de la tabla

CREATE TABLE peliculas (

idPeliculas INT(11) PRIMARY KEY,

Nombre VARCHAR(45),

Duracion INT(11),

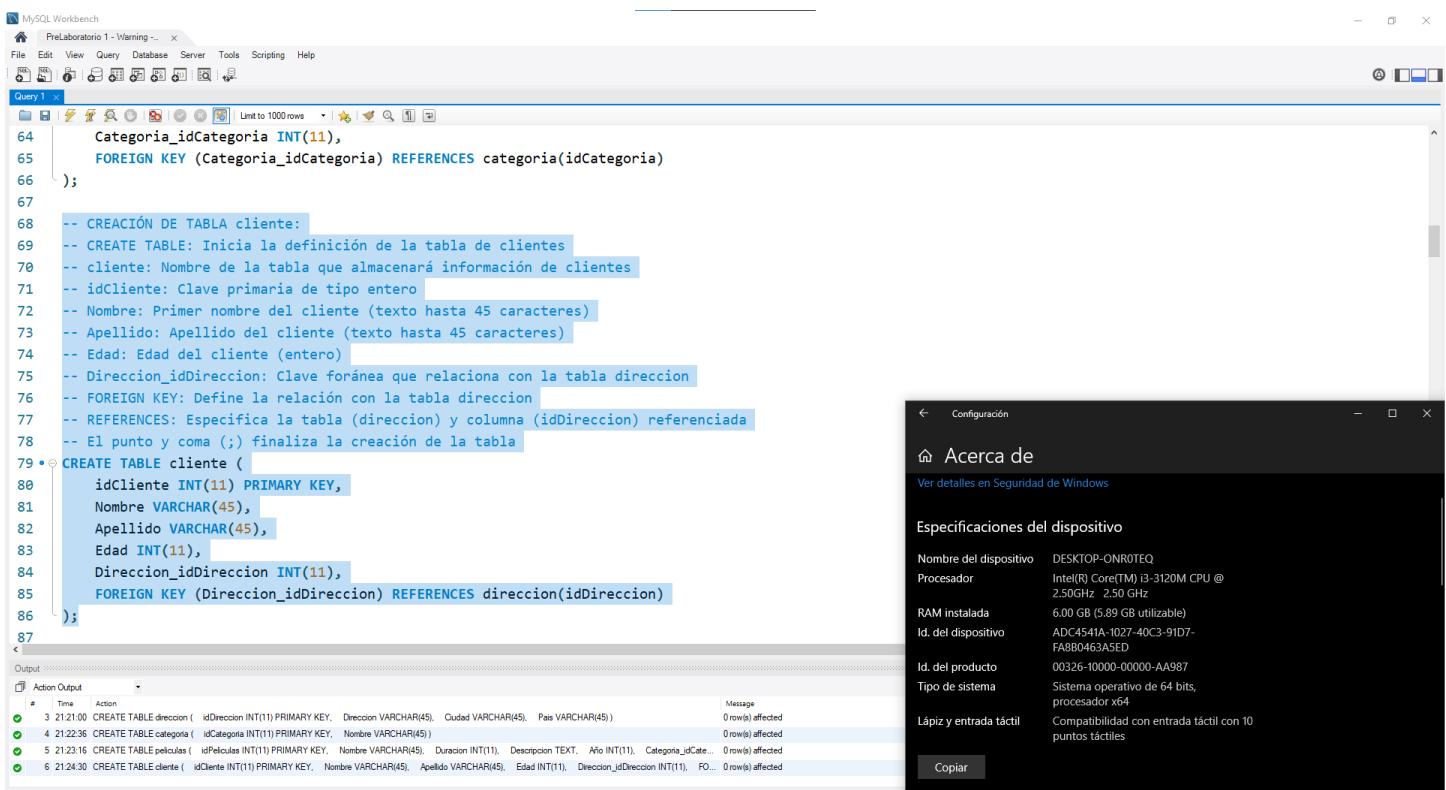
Descripcion TEXT,

Año INT(11),

```

Categoria_idCategoria INT(11),
FOREIGN KEY (Categoria_idCategoria) REFERENCES categoria(idCategoria)
);

```



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

**Query Editor (Query 1):**

```

64     Categoria_idCategoria INT(11),
65     FOREIGN KEY (Categoria_idCategoria) REFERENCES categoria(idCategoria)
66 );
67
68 -- CREACIÓN DE TABLA cliente:
69 -- CREATE TABLE: Inicia la definición de la tabla de clientes
70 -- cliente: Nombre de la tabla que almacenará información de clientes
71 -- idCliente: Clave primaria de tipo entero
72 -- Nombre: Primer nombre del cliente (texto hasta 45 caracteres)
73 -- Apellido: Apellido del cliente (texto hasta 45 caracteres)
74 -- Edad: Edad del cliente (entero)
75 -- Direccion_idDireccion: Clave foránea que relaciona con la tabla direccion
76 -- FOREIGN KEY: Define la relación con la tabla direccion
77 -- REFERENCES: Especifica la tabla (direccion) y columna (idDireccion) referenciada
78 -- El punto y coma (;) finaliza la creación de la tabla
79 CREATE TABLE cliente (
80     idCliente INT(11) PRIMARY KEY,
81     Nombre VARCHAR(45),
82     Apellido VARCHAR(45),
83     Edad INT(11),
84     Direccion_idDireccion INT(11),
85     FOREIGN KEY (Direccion_idDireccion) REFERENCES direccion(idDireccion)
86 );
87

```

**Action Output:**

#	Time	Action	Message
3	21:21:00	CREATE TABLE direccion ( idDireccion INT(11) PRIMARY KEY, Direccion VARCHAR(45), Ciudad VARCHAR(45), Pais VARCHAR(45) )	0 row(s) affected
4	21:22:36	CREATE TABLE categoria ( idCategoria INT(11) PRIMARY KEY, Nombre VARCHAR(45) )	0 row(s) affected
5	21:23:16	CREATE TABLE pelculas ( idPelculas INT(11) PRIMARY KEY, Nombre VARCHAR(45), Duracion INT(11), Descripcion TEXT, Año INT(11), Categoria_idCate... )	0 row(s) affected
6	21:24:30	CREATE TABLE cliente ( idCliente INT(11) PRIMARY KEY, Nombre VARCHAR(45), Apellido VARCHAR(45), Edad INT(11), Direccion_idDireccion INT(11), FO... )	0 row(s) affected

**System Configuration Window (Configuración):**

Acerca de

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONR0TEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FAB8B0463A5ED
Id. del producto	00326-10000-0000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

-- CREACIÓN DE TABLA cliente:

-- CREATE TABLE: Inicia la definición de la tabla de clientes

-- cliente: Nombre de la tabla que almacenará información de clientes

-- idCliente: Clave primaria de tipo entero

-- Nombre: Primer nombre del cliente (texto hasta 45 caracteres)

-- Apellido: Apellido del cliente (texto hasta 45 caracteres)

-- Edad: Edad del cliente (entero)  
-- Direccion\_idDireccion: Clave foránea que relaciona con la tabla direccion  
-- FOREIGN KEY: Define la relación con la tabla direccion  
-- REFERENCES: Especifica la tabla (direccion) y columna (idDireccion) referenciada  
-- El punto y coma (;) finaliza la creación de la tabla

```
CREATE TABLE cliente (
    idCliente INT(11) PRIMARY KEY,
    Nombre VARCHAR(45),
    Apellido VARCHAR(45),
    Edad INT(11),
    Direccion_idDireccion INT(11),
    FOREIGN KEY (Direccion_idDireccion) REFERENCES direccion(idDireccion)
);
```

```

85     FOREIGN KEY (Direccion_idDireccion) REFERENCES direccion(idDireccion)
86 );
87
88 -- CREACIÓN DE TABLA inventario:
89 -- CREATE TABLE: Inicia la definición de la tabla de inventario
90 -- inventario: Nombre de la tabla que controla las copias físicas
91 -- idCopiasPeliculas: Clave primaria de tipo entero
92 -- Peliculas_idPeliculas: Clave foránea que relaciona con la tabla peliculas
93 -- FOREIGN KEY: Define la relación con la tabla peliculas
94 -- REFERENCES: Especifica la tabla (peliculas) y columna (idPeliculas) referenciada
95 -- Disponible: Campo booleano (0 = no disponible, 1 = disponible)
96 -- TINYINT(1): Tipo de dato para valores booleanos en MySQL
97 -- El punto y coma (;) finaliza la creación de la tabla
98 • CREATE TABLE inventario (
99     idCopiasPeliculas INT(11) PRIMARY KEY,
100    Peliculas_idPeliculas INT(11),
101    Disponible TINYINT(1),
102    FOREIGN KEY (Peliculas_idPeliculas) REFERENCES peliculas(idPeliculas)
103 );
104
105 -- CREACIÓN DE TABLA renta:
106 -- CREATE TABLE: Inicia la definición de la tabla de rentas
107 -- renta: Nombre de la tabla que registra las transacciones de alquiler
108 -- idRenta: Clave primaria de tipo entero
<
Output...>
Action Output
# Time Action Message
1 21:22:36 CREATE TABLE categoria ( idCategoria INT(11) PRIMARY KEY, Nombre VARCHAR(45) ) 0 row(s) affected
2 21:23:15 CREATE TABLE peliculas ( idPeliculas INT(11) PRIMARY KEY, Nombre VARCHAR(45), Duracion INT(11), Descripcion TEXT, Año INT(11), Categoria_idCategoria INT(11) ) 0 row(s) affected
3 21:24:30 CREATE TABLE cliente ( idCliente INT(11) PRIMARY KEY, Nombre VARCHAR(45), Apellido VARCHAR(45), Edad INT(11), Direccion_idDireccion INT(11), FechaAlquiler DATE, Disponible TINYINT(1) ) 0 row(s) affected
4 21:25:13 CREATE TABLE inventario ( idCopiasPeliculas INT(11) PRIMARY KEY, Peliculas_idPeliculas INT(11), Disponible TINYINT(1), FOREIGN KEY (Peliculas_idPeliculas) REFERENCES peliculas(idPeliculas) ) 0 row(s) affected

```

Configuración

Acerca de

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONROTEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FA880463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar

-- CREACIÓN DE TABLA inventario:

-- CREATE TABLE: Inicia la definición de la tabla de inventario

-- inventario: Nombre de la tabla que controla las copias físicas

-- idCopiasPeliculas: Clave primaria de tipo entero

-- Peliculas\_idPeliculas: Clave foránea que relaciona con la tabla peliculas

-- FOREIGN KEY: Define la relación con la tabla peliculas

-- REFERENCES: Especifica la tabla (peliculas) y columna (idPeliculas) referenciada

-- Disponible: Campo booleano (0 = no disponible, 1 = disponible)

-- TINYINT(1): Tipo de dato para valores booleanos en MySQL

-- El punto y coma (;) finaliza la creación de la tabla

CREATE TABLE inventario (

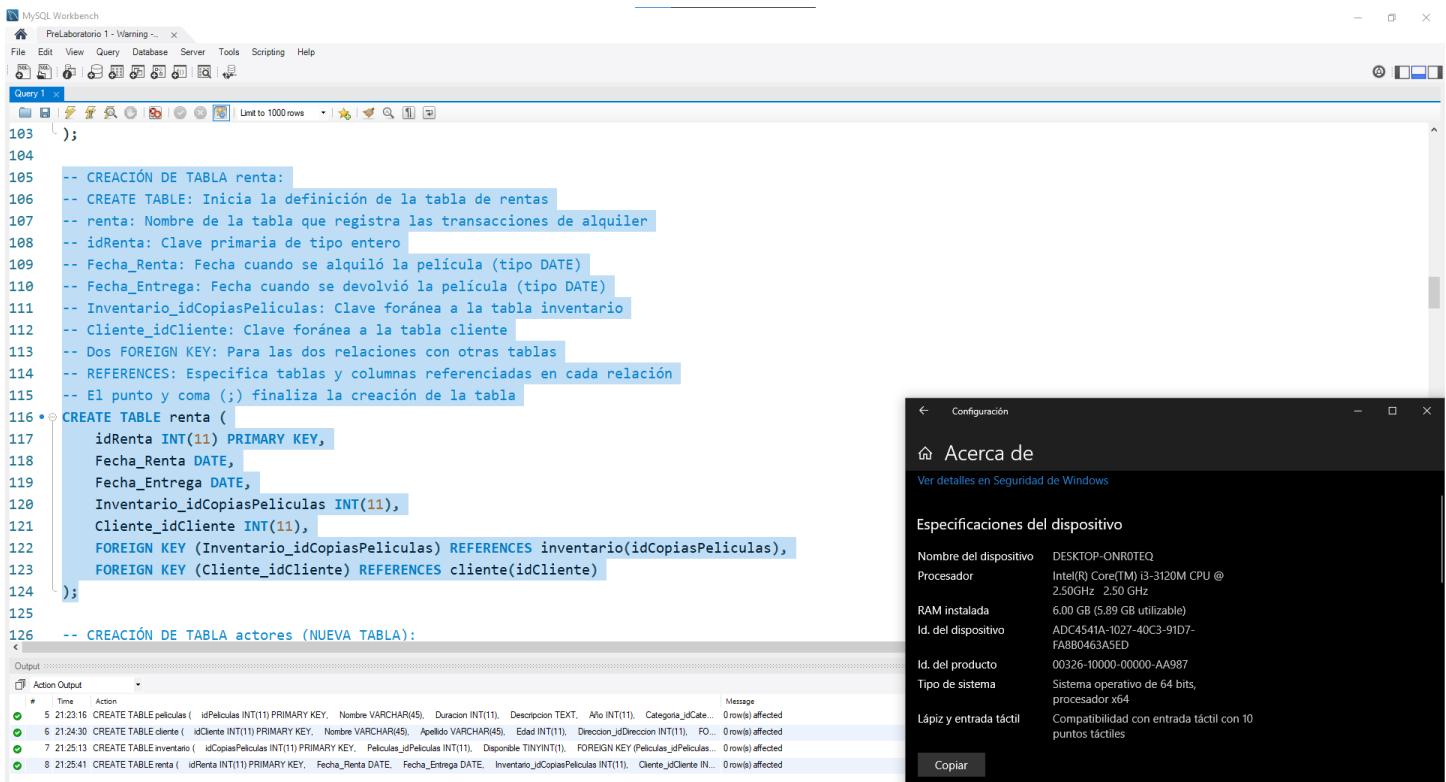
idCopiasPeliculas INT(11) PRIMARY KEY,

Peliculas\_idPeliculas INT(11),

Disponible TINYINT(1),

FOREIGN KEY (Peliculas\_idPeliculas) REFERENCES peliculas(idPeliculas)

);



```

-- CREACIÓN DE TABLA renta:
-- CREATE TABLE: Inicia la definición de la tabla de rentas
-- renta: Nombre de la tabla que registra las transacciones de alquiler
-- idRenta: Clave primaria de tipo entero
-- Fecha_Renta: Fecha cuando se alquiló la película (tipo DATE)
-- Fecha_Entrega: Fecha cuando se devolvió la película (tipo DATE)
-- Inventario_idCopiasPeliculas: Clave foránea a la tabla inventario
-- Cliente_idCliente: Clave foránea a la tabla cliente
-- Dos FOREIGN KEY: Para las dos relaciones con otras tablas
-- REFERENCES: Especifica tablas y columnas referenciadas en cada relación
-- El punto y coma (;) finaliza la creación de la tabla

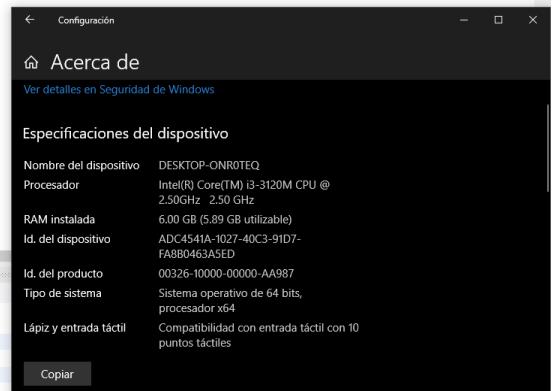
CREATE TABLE renta (
    idRenta INT(11) PRIMARY KEY,
    Fecha_Renta DATE,
    Fecha_Entrega DATE,
    Inventario_idCopiasPeliculas INT(11),
    Cliente_idCliente INT(11),
    FOREIGN KEY (Inventario_idCopiasPeliculas) REFERENCES inventario(idCopiasPeliculas),
    FOREIGN KEY (Cliente_idCliente) REFERENCES cliente(idCliente)
);

-- CREACIÓN DE TABLA actores (NUEVA TABLA):

```

Output:

#	Time	Action	Message
5	21:23:16	CREATE TABLE peliculas (	idPelículas INT(11) PRIMARY KEY, Nombre VARCHAR(45), Duracion INT(11), Descripción TEXT, Año INT(11), Categoría_idCate... 0 rows(s) affected
6	21:24:30	CREATE TABLE cliente (	idCliente INT(11) PRIMARY KEY, Nombre VARCHAR(45), Apellido VARCHAR(45), Edad INT(11), Dirección_idDirección INT(11), FO... 0 rows(s) affected
7	21:25:13	CREATE TABLE inventario (	idCopiasPeliculas INT(11) PRIMARY KEY, Películas_idPelículas INT(11), Disponible TINYINT(1), FOREIGN KEY (Películas_idPelículas... 0 rows(s) affected
8	21:25:41	CREATE TABLE renta (	idRenta INT(11) PRIMARY KEY, Fecha_Renta DATE, Fecha_Entrega DATE, Inventario_idCopiasPeliculas INT(11), Cliente_idCliente IN... 0 rows(s) affected



-- CREACIÓN DE TABLA renta:

-- CREATE TABLE: Inicia la definición de la tabla de rentas

-- renta: Nombre de la tabla que registra las transacciones de alquiler

-- idRenta: Clave primaria de tipo entero

-- Fecha\_Renta: Fecha cuando se alquiló la película (tipo DATE)

-- Fecha\_Entrega: Fecha cuando se devolvió la película (tipo DATE)

-- Inventario\_idCopiasPeliculas: Clave foránea a la tabla inventario

-- Cliente\_idCliente: Clave foránea a la tabla cliente

-- Dos FOREIGN KEY: Para las dos relaciones con otras tablas

-- REFERENCES: Especifica tablas y columnas referenciadas en cada relación

-- El punto y coma (;) finaliza la creación de la tabla

CREATE TABLE renta (

    idRenta INT(11) PRIMARY KEY,

    Fecha\_Renta DATE,

    Fecha\_Entrega DATE,

    Inventario\_idCopiasPeliculas INT(11),

    Cliente\_idCliente INT(11),

    FOREIGN KEY (Inventario\_idCopiasPeliculas) REFERENCES inventario(idCopiasPeliculas),

FOREIGN KEY (Cliente\_idCliente) REFERENCES cliente(idCliente)

);

```
124 );
125
126 -- CREACIÓN DE TABLA actores (NUEVA TABLA):
127 -- CREATE TABLE: Inicia la definición de la tabla de actores
128 -- actores: Nombre de la nueva tabla para el punto 7 del requerimiento
129 -- idActor: Clave primaria de tipo entero
130 -- Nombre: Primer nombre del actor (texto hasta 45 caracteres)
131 -- Apellido: Apellido del actor (texto hasta 45 caracteres)
132 -- FechaNacimiento: Fecha de nacimiento (tipo DATE)
133 -- Nacionalidad: País de origen del actor (texto hasta 45 caracteres)
134 -- El punto y coma (;) finaliza la creación de la tabla
135 CREATE TABLE actores (
136     idActor INT(11) PRIMARY KEY,
137     Nombre VARCHAR(45),
138     Apellido VARCHAR(45),
139     FechaNacimiento DATE,
140     Nacionalidad VARCHAR(45)
141 );
142
143 -- CREACIÓN DE TABLA peliculas_actores (NUEVA TABLA):
144 -- CREATE TABLE: Inicia la definición de la tabla de relación
145 -- peliculas_actores: Nombre de la tabla que relaciona películas con actores
146 -- idRelacion: Clave primaria de tipo entero
147 -- Peliculas idPeliculas: Clave foránea a la tabla peliculas
<
```

Output:

Time	Action	Message
6 21:24:30	CREATE TABLE cliente ( idCliente INT(11) PRIMARY KEY, Nombre VARCHAR(45), Apellido VARCHAR(45), Edad INT(11), Direccion_idDireccion INT(11), FOREIGN KEY (Direccion_idDireccion) REFERENCES Direccion(idDireccion) ON UPDATE CASCADE, FOREIGN KEY (Apellido) REFERENCES actores(Apellido) ON UPDATE CASCADE ) ENGINE=InnoDB DEFAULT CHARSET=utf8;	0 row(s) affected
7 21:25:13	CREATE TABLE inventario ( idCopiasPeliculas INT(11) PRIMARY KEY, Peliculas_idPeliculas INT(11), Disponible TINYINT(1), FOREIGN KEY (Peliculas_idPeliculas) REFERENCES peliculas(idPeliculas) ON UPDATE CASCADE ) ENGINE=InnoDB DEFAULT CHARSET=utf8;	0 row(s) affected
8 21:25:41	CREATE TABLE renta ( idRenta INT(11) PRIMARY KEY, Fecha_Renta DATE, Fecha_Entrega DATE, Inventario_idCopiasPeliculas INT(11), Cliente_idCliente INT(11), FOREIGN KEY (Inventario_idCopiasPeliculas) REFERENCES inventario(idCopiasPeliculas) ON UPDATE CASCADE, FOREIGN KEY (Cliente_idCliente) REFERENCES cliente(idCliente) ON UPDATE CASCADE ) ENGINE=InnoDB DEFAULT CHARSET=utf8;	0 row(s) affected
9 21:26:25	CREATE TABLE actores ( idActor INT(11) PRIMARY KEY, Nombre VARCHAR(45), Apellido VARCHAR(45), FechaNacimiento DATE, Nacionalidad VARCHAR(45) ) ENGINE=InnoDB DEFAULT CHARSET=utf8;	0 row(s) affected

-- CREACIÓN DE TABLA actores (NUEVA TABLA):

-- CREATE TABLE: Inicia la definición de la tabla de actores

-- actores: Nombre de la nueva tabla para el punto 7 del requerimiento

-- idActor: Clave primaria de tipo entero

-- Nombre: Primer nombre del actor (texto hasta 45 caracteres)

-- Apellido: Apellido del actor (texto hasta 45 caracteres)

-- FechaNacimiento: Fecha de nacimiento (tipo DATE)

-- Nacionalidad: País de origen del actor (texto hasta 45 caracteres)

-- El punto y coma (;) finaliza la creación de la tabla

```
CREATE TABLE actores (
```

```
    idActor INT(11) PRIMARY KEY,
```

```
    Nombre VARCHAR(45),
```

```
    Apellido VARCHAR(45),
```

```
    FechaNacimiento DATE,
```

```
    Nacionalidad VARCHAR(45)
```

```
);
```

```

-- CREACIÓN DE TABLA peliculas_actores (NUEVA TABLA):
-- CREATE TABLE: Inicia la definición de la tabla de relación
-- peliculas_actores: Nombre de la tabla que relaciona películas con actores
-- idRelacion: Clave primaria de tipo entero
-- Peliculas_idPelículas: Clave foránea a la tabla películas
-- Actores_idActor: Clave foránea a la tabla actores
-- Personaje: Nombre del personaje que interpretó el actor (texto hasta 45 caracteres)
-- Dos FOREIGN KEY: Para las relaciones con películas y actores
-- REFERENCES: Especifica tablas y columnas referenciadas en cada relación
-- El punto y coma (;) finaliza la creación de la tabla

CREATE TABLE peliculas_actores (
    idRelacion INT(11) PRIMARY KEY,
    Peliculas_idPelículas INT(11),
    Actores_idActor INT(11),
    Personaje VARCHAR(45),
    FOREIGN KEY (Peliculas_idPelículas) REFERENCES peliculas(idPelículas),
    FOREIGN KEY (Actores_idActor) REFERENCES actores(idActor)
);

-- INSERCIÓN DE DATOS EN TABLA dirección:
-- INSERT INTO: Comando para insertar nuevos registros en una tabla
-- dirección: Nombre de la tabla destino
-- (idDirección, Dirección, Ciudad, País): Lista de columnas que recibirán valores

```

Output:

#	Time	Action	Message
7	21:25:13	CREATE TABLE inventario (	idCopiasPelículas INT(11) PRIMARY KEY, Peliculas_idPelículas INT(11), Disponible TINYINT(1), FOREIGN KEY (Peliculas_idPelículas... 0 rows(s) affected
8	21:25:41	CREATE TABLE renta (	idRenta INT(11) PRIMARY KEY, Fecha_Renta DATE, Fecha_Entrega DATE, inventario_idCopiasPelículas INT(11), Cliente_idCliente IN... 0 rows(s) affected
9	21:26:25	CREATE TABLE actores (	idActor INT(11) PRIMARY KEY, Nombre VARCHAR(45), Apellido VARCHAR(45), FechaNacimiento DATE, Nacionalidad VARCHAR(4... 0 rows(s) affected
10	21:27:44	CREATE TABLE peliculas_actores (	idRelacion INT(11) PRIMARY KEY, Peliculas_idPelículas INT(11), Actores_idActor INT(11), Personaje VARCHAR(45), FORRE... 0 row(s) affected

-- CREACIÓN DE TABLA peliculas\_actores (NUEVA TABLA):

-- CREATE TABLE: Inicia la definición de la tabla de relación

-- peliculas\_actores: Nombre de la tabla que relaciona películas con actores

-- idRelacion: Clave primaria de tipo entero

-- Peliculas\_idPelículas: Clave foránea a la tabla películas

-- Actores\_idActor: Clave foránea a la tabla actores

-- Personaje: Nombre del personaje que interpretó el actor (texto hasta 45 caracteres)

-- Dos FOREIGN KEY: Para las relaciones con películas y actores

-- REFERENCES: Especifica tablas y columnas referenciadas en cada relación

-- El punto y coma (;) finaliza la creación de la tabla

CREATE TABLE peliculas\_actores (

idRelacion INT(11) PRIMARY KEY,

Peliculas\_idPelículas INT(11),

Actores\_idActor INT(11),

Personaje VARCHAR(45),

FOREIGN KEY (Peliculas\_idPelículas) REFERENCES peliculas(idPelículas),

FOREIGN KEY (Actores\_idActor) REFERENCES actores(idActor)

);

```

157 Personaje VARCHAR(45),
158 FOREIGN KEY (Peliculas_idPeliculas) REFERENCES peliculas(idPeliculas),
159 FOREIGN KEY (Actores_idActor) REFERENCES actores(idActor)
160 );
161
162 -- INserción de DATOS EN TABLA dirección:
163 -- INSERT INTO: Comando para insertar nuevos registros en una tabla
164 -- dirección: Nombre de la tabla destino
165 -- (idDireccion, Dirección, Ciudad, País): Lista de columnas que recibirán valores
166 -- VALUES: Palabra clave que precede a los valores a insertar
167 -- Cada conjunto de valores va entre paréntesis y separado por comas
168 -- 5 registros de ejemplo con información de direcciones
169 -- Cada registro tiene: id (entero), dirección (texto), ciudad (texto), país (texto)
170 -- El punto y coma (;) finaliza la instrucción de inserción
171 • INSERT INTO dirección (idDireccion, Dirección, Ciudad, País) VALUES
172 (1, 'Calle Principal 123', 'Ciudad de México', '12345'),
173 (2, 'Avenida Revolución 456', 'Guadalajara', '67890'),
174 (3, 'Boulevard Los Heroes 789', 'Monterrey', '54321'),
175 (4, 'Calle Morelos 321', 'Puebla', '98765'),
176 (5, 'Avenida Juárez 654', 'Tijuana', '13579');
177
178 -- INserción de DATOS EN TABLA categoría:
179 -- INSERT INTO: Comando para insertar en tabla categoría
180 -- (idCategoria, Nombre): Columnas que recibirán valores

```

Output:

#	Time	Action	Message
8	21:25:41	CREATE TABLE renta ( idRenta INT(11) PRIMARY KEY, Fecha_Renta DATE, Fecha_Entrega DATE, Inventario_idCopiasPelicula INT(11), Cliente_idCliente IN... ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci	0 row(s) affected
9	21:26:25	CREATE TABLE actores ( idActor INT(11) PRIMARY KEY, Nombre VARCHAR(45), Apellido VARCHAR(45), FechaNacimiento DATE, Nacionalidad VARCHAR(45), ... ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci	0 row(s) affected
10	21:27:44	CREATE TABLE peliculas_actores ( idRelacion INT(11) PRIMARY KEY, Peliculas_idPelículas INT(11), Actores_idActor INT(11), Personaje VARCHAR(45), FOREIGN KEY (Peliculas_idPelículas) REFERENCES peliculas(idPeliculas), FOREIGN KEY (Actores_idActor) REFERENCES actores(idActor)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci	0 row(s) affected
11	21:29:04	INSERT INTO dirección (idDireccion, Dirección, Ciudad, País) VALUES (1, 'Calle Principal 123', 'Ciudad de México', '12345'), (2, 'Avenida Revolución 456', 'Guadalajara', '67890'), (3, 'Boulevard Los Heroes 789', 'Monterrey', '54321'), (4, 'Calle Morelos 321', 'Puebla', '98765'), (5, 'Avenida Juárez 654', 'Tijuana', '13579');	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

## -- INserción de DATOS EN TABLA dirección:

-- INSERT INTO: Comando para insertar nuevos registros en una tabla

-- dirección: Nombre de la tabla destino

-- (idDireccion, Dirección, Ciudad, País): Lista de columnas que recibirán valores

-- VALUES: Palabra clave que precede a los valores a insertar

-- Cada conjunto de valores va entre paréntesis y separado por comas

-- 5 registros de ejemplo con información de direcciones

-- Cada registro tiene: id (entero), dirección (texto), ciudad (texto), país (texto)

-- El punto y coma (;) finaliza la instrucción de inserción

INSERT INTO dirección (idDireccion, Dirección, Ciudad, País) VALUES

(1, 'Calle Principal 123', 'Ciudad de México', '12345'),

(2, 'Avenida Revolución 456', 'Guadalajara', '67890'),

(3, 'Boulevard Los Heroes 789', 'Monterrey', '54321'),

(4, 'Calle Morelos 321', 'Puebla', '98765'),

(5, 'Avenida Juárez 654', 'Tijuana', '13579');

```

MySQL Workbench
PreLaboratorio 1 - Warning ~
File Edit View Query Database Server Tools Scripting Help
Query 1
175 (4, 'Calle Morelos 321', 'Puebla', '98765'),
176 (5, 'Avenida Juárez 654', 'Tijuana', '13579');
177
178 -- INSERCIÓN DE DATOS EN TABLA categoria:
179 -- INSERT INTO: Comando para insertar en tabla categoria
180 -- (idCategoria, Nombre): Columnas que recibirán valores
181 -- VALUES: Precede los 5 registros de categorías de películas
182 -- Cada registro tiene: id (entero) y nombre de categoría (texto)
183 -- El punto y coma (;) finaliza la instrucción
184 • INSERT INTO categoria (idCategoria, Nombre) VALUES
185 (1, 'Acción'),
186 (2, 'Comedia'),
187 (3, 'Drama'),
188 (4, 'Ciencia Ficción'),
189 (5, 'Animación');
190
191 -- INSERCIÓN DE DATOS EN TABLA peliculas:
192 -- INSERT INTO: Comando para insertar en tabla peliculas
193 -- Se especifican todas las columnas: id, nombre, duración, descripción, año y categoría
194 -- VALUES: Precede los 5 registros de películas de ejemplo
195 -- Cada registro tiene valores para todos los campos en el orden especificado
196 -- Categoria_idCategoria debe coincidir con IDs existentes en tabla categoria
197 -- El punto y coma (;) finaliza la instrucción
198 • INSERT INTO peliculas (idPelículas, Nombre, Duracion, Descripcion, Año, Categoria_idCategoria) VALUES
<
Output:
Action Output
# Time Action Message
9 21:26:25 CREATE TABLE actores ( idActor INT(11) PRIMARY KEY, Nombre VARCHAR(45), Apellido VARCHAR(45), FechaNacimiento DATE, Nacionalidad VARCHAR(45) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
10 21:27:44 CREATE TABLE peliculas_actores ( idRelacion INT(11) PRIMARY KEY, Peliculas_idPelículas INT(11), Actores_idActor INT(11), Personaje VARCHAR(45) ), FOREIGN KEY (Peliculas_idPelículas) REFERENCES peliculas(idPelículas), FOREIGN KEY (Actores_idActor) REFERENCES actores(idActor)
11 21:29:04 INSERT INTO direccion (idDireccion, Direccion, Ciudad, Pais) VALUES (1, 'Calle Principal 123', 'Ciudad de México', '12345'), (2, 'Avenida Revolución 456', 'Guadalajara', '678')
12 21:29:36 INSERT INTO categoria (idCategoria, Nombre) VALUES (1, 'Acción'), (2, 'Comedia'), (3, 'Drama'), (4, 'Ciencia Ficción'), (5, 'Animación')

```

-- INSERCIÓN DE DATOS EN TABLA categoria:

-- INSERT INTO: Comando para insertar en tabla categoria

-- (idCategoria, Nombre): Columnas que recibirán valores

-- VALUES: Precede los 5 registros de categorías de películas

-- Cada registro tiene: id (entero) y nombre de categoría (texto)

-- El punto y coma (;) finaliza la instrucción

INSERT INTO categoria (idCategoria, Nombre) VALUES

(1, 'Acción'),

(2, 'Comedia'),

(3, 'Drama'),

(4, 'Ciencia Ficción'),

(5, 'Animación');

The screenshot shows the MySQL Workbench interface. In the main window, there is a 'Query 1' tab with the following SQL code:

```

187 (3, 'Drama'),
188 (4, 'Ciencia Ficción'),
189 (5, 'Animación');

190
191 -- INserCIÓN DE DATOS EN TABLA peliculas:
192 -- INSERT INTO: Comando para insertar en tabla peliculas
193 -- Se especifican todas las columnas: id, nombre, duración, descripción, año y categoría
194 -- VALUES: Precede los 5 registros de películas de ejemplo
195 -- Cada registro tiene valores para todos los campos en el orden especificado
196 -- Categoria_idCategoria debe coincidir con IDs existentes en tabla categoria
197 -- El punto y coma (;) finaliza la instrucción
198 • INSERT INTO peliculas (idPeliculas, Nombre, Duracion, Descripcion, Año, Categoria_idCategoria) VALUES
199 (1, 'POKEMON1', 120, 'Aventuras de Pikachu y sus amigos', 1998, 5),
200 (2, 'El Viaje de Juliana', 95, 'Drama familiar', 2020, 3),
201 (3, 'Rápidos y Furiosos', 110, 'Carreras ilegales', 2001, 1),
202 (4, 'Risas Eternas', 88, 'Comedia familiar', 2019, 2),
203 (5, 'Galaxia Lejana', 135, 'Aventura espacial', 2022, 4);
204
205 -- INserCIÓN DE DATOS EN TABLA cliente:
206 -- INSERT INTO: Comando para insertar en tabla cliente
207 -- (idcliente, Nombre, Apellido, Edad, Direccion_idDireccion): Columnas destino
208 -- VALUES: Precede los 5 registros de clientes de ejemplo
209 -- Cada registro tiene: id, nombre, apellido, edad y referencia a dirección
210 -- Dirección idDireccion debe coincidir con IDs existentes en tabla direccion

```

The 'Output' pane shows the results of the execution:

#	Tiempo	Action	Message
10	21:27:44	CREATE TABLE peliculas_actores ( `idRelacion` INT(11) PRIMARY KEY, `Peliculas_idPeliculas` INT(11), `Actores_idActor` INT(11), `Personaje` VARCHAR(45), FOREIGN KEY)	0 row(s) affected
11	21:29:04	INSERT INTO direccion (idDireccion, Direccion, Ciudad, Pais) VALUES (1, 'Calle Principal 123', 'Ciudad de México', '12345'), (2, 'Avenida Revolución 456', 'Guadalajara', '45678')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
12	21:29:36	INSERT INTO categoria (idCategoria, Nombre) VALUES (1, 'Acción'), (2, 'Comedia'), (3, 'Drama'), (4, 'Ciencia Ficción'), (5, 'Animación')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
13	21:30:03	INSERT INTO peliculas (idPeliculas, Nombre, Duracion, Descripcion, Año, Categoria_idCategoria) VALUES (1, 'POKEMON1', 120, 'Aventuras de Pikachu y sus amigos', 1998, 5)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

A separate window titled 'Configuración' is also visible, showing device specifications:

- Nombre del dispositivo: DESKTOP-ONROTEQ
- Procesador: Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz, 2.50 GHz
- RAM instalada: 6.00 GB (5.89 GB utilizable)
- Id. del dispositivo: ADC4541A-1027-40C3-91D7-FA880463A5ED
- Id. del producto: 00326-10000-00000-AA987
- Tipo de sistema: Sistema operativo de 64 bits, procesador x64
- Lápiz y entrada táctil: Compatibilidad con entrada táctil con 10 puntos táctiles

## -- INserCIÓN DE DATOS EN TABLA peliculas:

-- INSERT INTO: Comando para insertar en tabla peliculas

-- Se especifican todas las columnas: id, nombre, duración, descripción, año y categoría

-- VALUES: Precede los 5 registros de películas de ejemplo

-- Cada registro tiene valores para todos los campos en el orden especificado

-- Categoria\_idCategoria debe coincidir con IDs existentes en tabla categoria

-- El punto y coma (;) finaliza la instrucción

INSERT INTO peliculas (idPeliculas, Nombre, Duracion, Descripcion, Año, Categoria\_idCategoria) VALUES

(1, 'POKEMON1', 120, 'Aventuras de Pikachu y sus amigos', 1998, 5),

(2, 'El Viaje de Juliana', 95, 'Drama familiar', 2020, 3),

(3, 'Rápidos y Furiosos', 110, 'Carreras ilegales', 2001, 1),

(4, 'Risas Eternas', 88, 'Comedia familiar', 2019, 2),

(5, 'Galaxia Lejana', 135, 'Aventura espacial', 2022, 4);

```

196 -- Categoria_idCategoria debe coincidir con IDs existentes en tabla categoria
197 -- El punto y coma (;) finaliza la instrucción
198 • INSERT INTO peliculas (idPeliculas, Nombre, Duracion, Descripcion, Año, Categoria_idCategoria) VALUES
199 (1, 'POKEMON1', 120, 'Aventuras de Pikachu y sus amigos', 1998, 5),
200 (2, 'El Viaje de Juliana', 95, 'Drama familiar', 2020, 3),
201 (3, 'Rápidos y Furiosos', 110, 'Carreras ilegales', 2001, 1),
202 (4, 'Risas Eternas', 88, 'Comedia familiar', 2019, 2),
203 (5, 'Galaxia Lejana', 135, 'Aventura espacial', 2022, 4);
204
205 -- INSERCIÓN DE DATOS EN TABLA cliente:
206 -- INSERT INTO: Comando para insertar en tabla cliente
207 -- (idCliente, Nombre, Apellido, Edad, Direccion_idDireccion): Columnas destino
208 -- VALUES: Precede los 5 registros de clientes de ejemplo
209 -- Cada registro tiene: id, nombre, apellido, edad y referencia a dirección
210 -- Direccion_idDireccion debe coincidir con IDs existentes en tabla direccion
211 -- El punto y coma (;) finaliza la instrucción
212 • INSERT INTO cliente (idCliente, Nombre, Apellido, Edad, Direccion_idDireccion) VALUES
213 (1, 'Juan', 'Pérez', 25, 1),
214 (2, 'María', 'Gómez', 30, 2),
215 (3, 'Juliana', 'Ramírez', 22, 3),
216 (4, 'Carlos', 'López', 28, 4),
217 (5, 'Ana', 'Martínez', 35, 5);
218
219 -- INSERCIÓN DE DATOS EN TABLA inventario:

```

Output:

#	Tarea	Acción	Mensaje
11	21:29:04	INSERT INTO direccion (idDireccion, Direccion, Ciudad, País) VALUES (1, 'Calle Principal 123', 'Ciudad de México', 'Méjico')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
12	21:29:36	INSERT INTO categoria (idCategoria, Nombre) VALUES (1, 'Acción'), (2, 'Comedia'), (3, 'Drama'), (4, 'Ciencia Ficción'), (5, 'Animación')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
13	21:30:03	INSERT INTO peliculas (idPeliculas, Nombre, Duracion, Descripcion, Año, Categoria_idCategoria) VALUES (1, 'POKEMON1', 120, 'Aventuras de Pikachu y sus amigos', 1998, 5)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
14	21:30:33	INSERT INTO cliente (idCliente, Nombre, Apellido, Edad, Direccion_idDireccion) VALUES (1, 'Juan', 'Pérez', 25, 1), (2, 'María', 'Gómez', 30, 2), (3, 'Juliana', 'Ramírez', 22, 3), (4, 'Carlos', 'López', 28, 4), (5, 'Ana', 'Martínez', 35, 5);	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

## -- INSERCIÓN DE DATOS EN TABLA cliente:

-- INSERT INTO: Comando para insertar en tabla cliente

-- (idCliente, Nombre, Apellido, Edad, Direccion\_idDireccion): Columnas destino

-- VALUES: Precede los 5 registros de clientes de ejemplo

-- Cada registro tiene: id, nombre, apellido, edad y referencia a dirección

-- Direccion\_idDireccion debe coincidir con IDs existentes en tabla direccion

-- El punto y coma (;) finaliza la instrucción

INSERT INTO cliente (idCliente, Nombre, Apellido, Edad, Direccion\_idDireccion) VALUES

(1, 'Juan', 'Pérez', 25, 1),

(2, 'María', 'Gómez', 30, 2),

(3, 'Juliana', 'Ramírez', 22, 3),

(4, 'Carlos', 'López', 28, 4),

(5, 'Ana', 'Martínez', 35, 5);

The screenshot shows the MySQL Workbench interface. In the main window, there is a query editor titled 'Query 1' containing the following SQL code:

```

211 -- El punto y coma (;) finaliza la instrucción
212 • INSERT INTO cliente (idCliente, Nombre, Apellido, Edad, Direccion_idDireccion) VALUES
213 (1, 'Juan', 'Pérez', 25, 1),
214 (2, 'María', 'Gómez', 30, 2),
215 (3, 'Juliana', 'Ramírez', 22, 3),
216 (4, 'Carlos', 'López', 28, 4),
217 (5, 'Ana', 'Martínez', 35, 5);
218
219 -- INSERCIÓN DE DATOS EN TABLA inventario:
220 -- INSERT INTO: Comando para insertar en tabla inventario
221 -- (idCopiasPeliculas, Peliculas_idPeliculas, Disponible): Columnas destino
222 -- VALUES: Precede los 5 registros de copias de películas
223 -- Cada registro tiene: id, referencia a película y estado de disponibilidad (0/1)
224 -- Peliculas_idPeliculas debe coincidir con IDs existentes en tabla peliculas
225 -- El punto y coma (;) finaliza la instrucción
226 • INSERT INTO inventario (idCopiasPeliculas, Peliculas_idPeliculas, Disponible) VALUES
227 (1, 1, 1),
228 (2, 2, 1),
229 (3, 3, 0),
230 (4, 4, 1),
231 (5, 5, 1);
232
233 -- INSERCIÓN DE DATOS EN TABLA renta:
234 -- INSERT INTO: Comando para insertar en tabla renta

```

The 'Output' pane shows the results of the execution:

#	Time	Action	Message
12	21:29:36	INSERT INTO categoria (idCategoria, Nombre) VALUES (1, 'Acción'), (2, 'Comedia'), (3, 'Drama'), (4, 'Ciencia Ficción'), (5, 'Animación')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
13	21:30:03	INSERT INTO peliculas (idPeliculas, Nombre, Duracion, Descripcion, Año, Categoria_idCategoria) VALUES (1, 'POKEMON1', 120, 'Aventuras de Pikachu y sus amigos', 1998, 1)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
14	21:30:33	INSERT INTO cliente (idCliente, Nombre, Apellido, Edad, Direccion_idDireccion) VALUES (1, 'Juan', 'Pérez', 25, 1), (2, 'María', 'Gómez', 30, 2), (3, 'Juliana', 'Ramírez', 22, 3), (4, 'Carlos', 'López', 28, 4), (5, 'Ana', 'Martínez', 35, 5)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
15	21:31:14	INSERT INTO inventario (idCopiasPeliculas, Peliculas_idPeliculas, Disponible) VALUES (1, 1, 1), (2, 2, 1), (3, 3, 0), (4, 4, 1), (5, 5, 1);	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

An 'About' dialog box is open in the background, displaying system specifications:

- Nombre del dispositivo: DESKTOP-ONROTEQ
- Procesador: Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
- RAM instalada: 6.00 GB (5.89 GB utilizable)
- Id. del dispositivo: ADC4541A-1027-40C3-91D7-FA880463A5ED
- Id. del producto: 00326-10000-00000-AA987
- Tipo de sistema: Sistema operativo de 64 bits, procesador x64
- Lápiz y entrada táctil: Compatibilidad con entrada táctil con 10 puntos táctiles

-- INSERCIÓN DE DATOS EN TABLA inventario:

-- INSERT INTO: Comando para insertar en tabla inventario

-- (idCopiasPeliculas, Peliculas\_idPeliculas, Disponible): Columnas destino

-- VALUES: Precede los 5 registros de copias de películas

-- Cada registro tiene: id, referencia a película y estado de disponibilidad (0/1)

-- Peliculas\_idPeliculas debe coincidir con IDs existentes en tabla peliculas

-- El punto y coma (;) finaliza la instrucción

INSERT INTO inventario (idCopiasPeliculas, Peliculas\_idPeliculas, Disponible) VALUES

(1, 1, 1),

(2, 2, 1),

(3, 3, 0),

(4, 4, 1),

(5, 5, 1);

```

233 -- INserción de datos en tabla renta:
234 -- INSERT INTO: Comando para insertar en tabla renta
235 -- (idRenta, Fecha_Renta, Fecha_Entrega, inventario_idCopiasPelículas, Cliente_idCliente): Columnas
236 -- VALUES: Precede los 5 registros de transacciones de alquiler
237 -- Cada registro tiene: id, fechas de renta y entrega, referencia a inventario y cliente
238 -- Las fechas en formato 'YYYY-MM-DD' (estándar SQL)
239 -- Las referencias deben coincidir con IDs existentes en tablas relacionadas
240 -- El punto y coma (;) finaliza la instrucción
241 • INSERT INTO renta (idRenta, Fecha_Renta, Fecha_Entrega, inventario_idCopiasPelículas, Cliente_idCliente) VALUES
242 (1, '2023-01-15', '2023-01-20', 1, 1),
243 (2, '2023-02-10', '2023-02-15', 2, 2),
244 (3, '2023-03-05', '2023-03-10', 3, 3),
245 (4, '2023-04-20', '2023-04-25', 4, 4),
246 (5, '2023-05-15', '2023-05-20', 5, 5);
247
248 -- INserción de datos en tabla actores (NUEVA TABLA):
249 -- INSERT INTO: Comando para insertar en la nueva tabla actores
250 -- (idActor, Nombre, Apellido, FechaNacimiento, Nacionalidad): Columnas destino
251 -- VALUES: Precede los 5 registros de actores de ejemplo
252 -- Cada registro tiene: id, nombre, apellido, fecha de nacimiento y nacionalidad
253 -- Las fechas en formato 'YYYY-MM-DD' (estándar SQL)
254 -- El punto y coma (;) finaliza la instrucción
255 • INSERT INTO actores (idActor, Nombre, Apellido, FechaNacimiento, Nacionalidad) VALUES

```

Output:

#	Time	Action	Message
13	21:30:03	INSERT INTO películas (idPelículas, Nombre, Duración, Descripción, Año, Categoría_idCategoría)	VALUES (1, 'POKEMON1', 120, 'Aventuras de Pikachu y sus amigos', 1998, ... 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0)
14	21:30:33	INSERT INTO cliente (idCliente, Nombre, Apellido, Edad, Dirección, idDirección)	VALUES (1, 'Juan', 'Perez', 25, 1), (2, 'Maria', 'Gómez', 30, 2), (3, 'Juliana', 'Ramírez', 22, 3), (... 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0)
15	21:31:14	INSERT INTO inventario (idCopiasPelículas, Películas_idPelículas, Disponibles)	VALUES (1, 1, 1), (2, 2, 1), (3, 3, 0), (4, 4, 1), (5, 5, 1) 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
16	21:32:04	INSERT INTO renta (idRenta, Fecha_Renta, Fecha_Entrega, inventario_idCopiasPelículas, Cliente_idCliente)	VALUES (1, '2023-01-15', '2023-01-20', 1, 1), (2, '2023-02-10', '2023-02-15', 2, 2), (3, '2023-03-05', '2023-03-10', 3, 3), (4, '2023-04-20', '2023-04-25', 4, 4), (5, '2023-05-15', '2023-05-20', 5, 5); 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

-- INserción de datos en tabla renta:

-- INSERT INTO: Comando para insertar en tabla renta

-- (idRenta, Fecha\_Renta, Fecha\_Entrega, inventario\_idCopiasPelículas, Cliente\_idCliente): Columnas

-- VALUES: Precede los 5 registros de transacciones de alquiler

-- Cada registro tiene: id, fechas de renta y entrega, referencia a inventario y cliente

-- Las fechas en formato 'YYYY-MM-DD' (estándar SQL)

-- Las referencias deben coincidir con IDs existentes en tablas relacionadas

-- El punto y coma (;) finaliza la instrucción

INSERT INTO renta (idRenta, Fecha\_Renta, Fecha\_Entrega, inventario\_idCopiasPelículas, Cliente\_idCliente)  
VALUES

(1, '2023-01-15', '2023-01-20', 1, 1),

(2, '2023-02-10', '2023-02-15', 2, 2),

(3, '2023-03-05', '2023-03-10', 3, 3),

(4, '2023-04-20', '2023-04-25', 4, 4),

(5, '2023-05-15', '2023-05-20', 5, 5);

The screenshot shows the MySQL Workbench interface. On the left, the 'Query 1' tab is active, displaying a multi-line SQL script. The script includes comments explaining the structure of the 'actores' table and its data. It uses the 'INSERT INTO' statement to insert five rows of data. On the right, a separate window titled 'Configuración' (Configuration) is open, specifically the 'Acerca de' (About) tab. This tab provides system specifications like processor, RAM, and OS information.

```

247 -- INSERCIÓN DE DATOS EN TABLA actores (NUEVA TABLA):
248 -- INSERT INTO: Comando para insertar en la nueva tabla actores
249 -- (idActor, Nombre, Apellido, FechaNacimiento, Nacionalidad): Columnas destino
250 -- VALUES: Precede los 5 registros de actores de ejemplo
251 -- Cada registro tiene: id, nombre, apellido, fecha de nacimiento y nacionalidad
252 -- Las fechas en formato 'YYYY-MM-DD' (estándar SQL)
253 -- El punto y coma (;) finaliza la instrucción
254
255 • INSERT INTO actores (idActor, Nombre, Apellido, FechaNacimiento, Nacionalidad) VALUES
256 (1, 'Robert', 'Downey Jr.', '1965-04-04', 'Estadounidense'),
257 (2, 'Scarlett', 'Johansson', '1984-11-22', 'Estadounidense'),
258 (3, 'Juliana', 'Moore', '1960-12-03', 'Estadounidense'),
259 (4, 'Dwayne', 'Johnson', '1972-05-02', 'Estadounidense'),
260 (5, 'Margot', 'Robbie', '1990-07-02', 'Australian')
261
262 -- INSERCIÓN DE DATOS EN TABLA peliculas_actores (NUEVA TABLA):
263 -- INSERT INTO: Comando para insertar en la nueva tabla de relación
264 -- (idRelacion, Peliculas_idPeliculas, Actores_idActor, Personaje): Columnas destino
265 -- VALUES: Precede los 5 registros que relacionan películas con actores
266 -- Cada registro tiene: id, referencia a película, referencia a actor y nombre de personaje
267 -- Las referencias deben coincidir con IDs existentes en tablas relacionadas
268 -- El punto y coma (;) finaliza la instrucción
269 • INSERT INTO peliculas_actores (idRelacion, Peliculas_idPeliculas, Actores_idActor, Personaje) VALUES
270 (1, 3, 4, 'Luke Hobbs'),
<

```

**Action Output**

Tipo	Acción	Mensaje
14	21:30:33 INSERT INTO cliente (idCliente, Nombre, Apellido, Edad, Direccion_idDireccion) VALUES (1, 'Juan', 'Pérez', 25, 1), (2, 'María', 'Gómez', 30, 2), (3, 'Juliana', 'Ramírez', 22, 3), ...	5 rows affected Records: 5 Duplicates: 0 Warnings: 0
15	21:31:14 INSERT INTO inventario (idCopiasPeliculas, Peliculas_idPeliculas, Disponible) VALUES (1, 1, 1), (2, 2, 1), (3, 3, 0), (4, 4, 1), (5, 5, 1)	5 rows affected Records: 5 Duplicates: 0 Warnings: 0
16	21:32:04 INSERT INTO renta (idRenta, Fecha_Renta, Fecha_Entrega, Inventario_idCopiasPeliculas, Cliente_idCliente) VALUES (1, 2023-01-15, '2023-01-20', 1, 1), (2, 2023-02-10, '2023-02-15', 2, 2)	2 rows affected Records: 2 Duplicates: 0 Warnings: 0
17	21:33:09 INSERT INTO actores (idActor, Nombre, Apellido, FechaNacimiento, Nacionalidad) VALUES (1, 'Robert', 'Downey Jr.', '1965-04-04', 'Estadounidense'), (2, 'Scarlett', 'Johansson', '1984-11-22', 'Estadounidense'), (3, 'Juliana', 'Moore', '1960-12-03', 'Estadounidense'), (4, 'Dwayne', 'Johnson', '1972-05-02', 'Estadounidense'), (5, 'Margot', 'Robbie', '1990-07-02', 'Australian')	5 rows affected Records: 5 Duplicates: 0 Warnings: 0

## -- INSERCIÓN DE DATOS EN TABLA actores (NUEVA TABLA):

-- INSERT INTO: Comando para insertar en la nueva tabla actores

-- (idActor, Nombre, Apellido, FechaNacimiento, Nacionalidad): Columnas destino

-- VALUES: Precede los 5 registros de actores de ejemplo

-- Cada registro tiene: id, nombre, apellido, fecha de nacimiento y nacionalidad

-- Las fechas en formato 'YYYY-MM-DD' (estándar SQL)

-- El punto y coma (;) finaliza la instrucción

INSERT INTO actores (idActor, Nombre, Apellido, FechaNacimiento, Nacionalidad) VALUES

(1, 'Robert', 'Downey Jr.', '1965-04-04', 'Estadounidense'),

(2, 'Scarlett', 'Johansson', '1984-11-22', 'Estadounidense'),

(3, 'Juliana', 'Moore', '1960-12-03', 'Estadounidense'),

(4, 'Dwayne', 'Johnson', '1972-05-02', 'Estadounidense'),

(5, 'Margot', 'Robbie', '1990-07-02', 'Australian');

MySQL Workbench

ReLaboratorio 1 - Warning -

File Edit View Query Database Server Tools Scripting Help

Query 1

```

259 (4, 'Dwayne', 'Johnson', '1972-05-02', 'Estadounidense'),
260 (5, 'Margot', 'Robbie', '1990-07-02', 'Australiana');

261
-- INSERCIÓN DE DATOS EN TABLA peliculas_actores (NUEVA TABLA):
-- INSERT INTO: Comando para insertar en la nueva tabla de relación
264 -- (idRelacion, Peliculas_idPeliculas, Actores_idActor, Personaje): Columnas destino
-- VALUES: Precede los 5 registros que relacionan películas con actores
266 -- Cada registro tiene: id, referencia a película, referencia a actor y nombre de personaje
267 -- Las referencias deben coincidir con IDs existentes en tablas relacionadas
268 -- El punto y coma (;) finaliza la instrucción
269 • INSERT INTO peliculas_actores (idRelacion, Peliculas_idPeliculas, Actores_idActor, Personaje) VALUES
270 (1, 3, 4, 'Luke Hobbs'),
271 (2, 2, 3, 'María González'),
272 (3, 5, 2, 'Capitana Vega'),
273 (4, 4, 5, 'Lisa Johnson'),
274 (5, 1, 1, 'Dr. Pikachu (voz)');
275
-- CONSULTA PARA MOSTRAR TODOS LOS DATOS:
277 -- SELECT * FROM: Comando para seleccionar todas las columnas de una tabla
278 -- Cada instrucción muestra el contenido completo de una tabla diferente
279 -- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,
280 -- inventario, renta, actores y peliculas_actores
281 -- Cada SELECT es una instrucción independiente terminada con punto y coma (;)
282 • SELECT * FROM direccion;

```

Output:

#	Time	Action	Message
15	21:31:14	INSERT INTO inventario (dCopiasPeliculas, Peliculas_idPeliculas, Disponible) VALUES (1, 1, 1), (2, 2, 1), (3, 3, 0), (4, 4, 1), (5, 5, 1)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
16	21:32:04	INSERT INTO renta (idRenta, Fecha_Rentra, Fecha_Entrega, inventario_idCopiasPeliculas, Cliente_idCliente) VALUES (1, 2023-01-15, 2023-01-20, 1, 1), (2, 2023-02-10, 2, 2)	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
17	21:33:09	INSERT INTO actores (idActor, Nombre, Apellido, FechaNacimiento, Nacionalidad) VALUES ('1', 'Robert', 'Downey Jr.', '1965-04-04', 'Estadounidense'), ('2', 'Scarlett', 'Johnson', '1972-05-02', 'Estadounidense')	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
18	21:35:48	INSERT INTO peliculas_actores (idRelacion, Peliculas_idPeliculas, Actores_idActor, Personaje) VALUES (1, 3, 4, 'Luke Hobbs'), (2, 2, 3, 'María González'), (3, 5, 2, 'Capitana Vega'), (4, 4, 5, 'Lisa Johnson'), (5, 1, 1, 'Dr. Pikachu (voz)');	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

Configuración

Acerca de

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONROTEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FA880463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar

-- INSERCIÓN DE DATOS EN TABLA peliculas\_actores (NUEVA TABLA):

-- INSERT INTO: Comando para insertar en la nueva tabla de relación

-- (idRelacion, Peliculas\_idPeliculas, Actores\_idActor, Personaje): Columnas destino

-- VALUES: Precede los 5 registros que relacionan películas con actores

-- Cada registro tiene: id, referencia a película, referencia a actor y nombre de personaje

-- Las referencias deben coincidir con IDs existentes en tablas relacionadas

-- El punto y coma (;) finaliza la instrucción

INSERT INTO peliculas\_actores (idRelacion, Peliculas\_idPeliculas, Actores\_idActor, Personaje) VALUES

(1, 3, 4, 'Luke Hobbs'),

(2, 2, 3, 'María González'),

(3, 5, 2, 'Capitana Vega'),

(4, 4, 5, 'Lisa Johnson'),

(5, 1, 1, 'Dr. Pikachu (voz)');

MySQL Workbench

PreLaboratorio 1 - Warning ~

File Edit View Query Database Server Tools Scripting Help

Query 1

```

276 -- CONSULTA PARA MOSTRAR TODOS LOS DATOS:
277 -- SELECT * FROM: Comando para seleccionar todas las columnas de una tabla
278 -- Cada instrucción muestra el contenido completo de una tabla diferente
279 -- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,
280 -- inventario, renta, actores y peliculas_actores
281 -- Cada SELECT es una instrucción independiente terminada con punto y coma (;)
282 • SELECT * FROM direccion;
283 • SELECT * FROM categoria;
284 • SELECT * FROM peliculas;
285 • SELECT * FROM cliente;
286 • SELECT * FROM inventario;
287 • SELECT * FROM renta;
288 -- Adicion de 2 Tablas
289 • SELECT * FROM actores;
290 • SELECT * FROM peliculas_actores;
<
```

Result Grid | Filter Rows: Limit to 1000 rows | Edit | Export/Import | Wrap Cell Content |

	Direccion	Direccion	Ciudad	Pes
1	Calle Angel 123	Ciudad de México	12345	
2	Avenida Republica 456	Mérida	56789	
3	Boulevard Los Heroes 789	Monterrey	54321	
4	Calle Morelos 321	Puebla	98765	
5	Avenida Juárez 654	Tijuana	13579	
• NULL	NULL	NULL	NULL	NULL

dirección 1 x categoría 2 x películas 3 x cliente 4 x inventario 5 x renta 6 x actores 7 x películas\_actores 8

Action Output

#	Time	Action	Message
19	21:36:34	SELECT * FROM direccion LIMIT 0, 1000	5 row(s) returned
20	21:36:34	SELECT * FROM categoria LIMIT 0, 1000	5 row(s) returned
21	21:36:34	SELECT * FROM peliculas LIMIT 0, 1000	5 row(s) returned
22	21:36:35	SELECT * FROM cliente LIMIT 0, 1000	5 row(s) returned
23	21:36:35	SELECT * FROM inventario LIMIT 0, 1000	5 row(s) returned
24	21:36:35	SELECT * FROM renta LIMIT 0, 1000	5 row(s) returned
25	21:36:35	SELECT * FROM actores LIMIT 0, 1000	5 row(s) returned
26	21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned

Configuración

Acerca de

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONR0TEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FA880463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar

MySQL Workbench

PreLaboratorio 1 - Warning ~

File Edit View Query Database Server Tools Scripting Help

Query 1

```

276 -- CONSULTA PARA MOSTRAR TODOS LOS DATOS:
277 -- SELECT * FROM: Comando para seleccionar todas las columnas de una tabla
278 -- Cada instrucción muestra el contenido completo de una tabla diferente
279 -- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,
280 -- inventario, renta, actores y peliculas_actores
281 -- Cada SELECT es una instrucción independiente terminada con punto y coma (;)
282 • SELECT * FROM direccion;
283 • SELECT * FROM categoria;
284 • SELECT * FROM peliculas;
285 • SELECT * FROM cliente;
286 • SELECT * FROM inventario;
287 • SELECT * FROM renta;
288 -- Adicion de 2 Tablas
289 • SELECT * FROM actores;
290 • SELECT * FROM peliculas_actores;
<
```

Result Grid | Filter Rows: Limit to 1000 rows | Edit | Export/Import | Wrap Cell Content |

	idCategoria	Nombre
1		Acción
2		Comedia
3		Drama
4		Ciencia Ficción
5		Animación
• NULL	NULL	NULL

dirección 1 x categoría 2 x películas 3 x cliente 4 x inventario 5 x renta 6 x actores 7 x películas\_actores 8

Action Output

#	Time	Action	Message
19	21:36:34	SELECT * FROM direccion LIMIT 0, 1000	5 row(s) returned
20	21:36:34	SELECT * FROM categoria LIMIT 0, 1000	5 row(s) returned
21	21:36:34	SELECT * FROM peliculas LIMIT 0, 1000	5 row(s) returned
22	21:36:35	SELECT * FROM cliente LIMIT 0, 1000	5 row(s) returned
23	21:36:35	SELECT * FROM inventario LIMIT 0, 1000	5 row(s) returned
24	21:36:35	SELECT * FROM renta LIMIT 0, 1000	5 row(s) returned
25	21:36:35	SELECT * FROM actores LIMIT 0, 1000	5 row(s) returned
26	21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned

Configuración

Acerca de

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONR0TEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FA880463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar

MySQL Workbench

PreLaboratorio 1 - Warning ~

File Edit View Query Database Server Tools Scripting Help

Query 1

```

276 -- CONSULTA PARA MOSTRAR TODOS LOS DATOS:
277 -- SELECT * FROM: Comando para seleccionar todas las columnas de una tabla
278 -- Cada instrucción muestra el contenido completo de una tabla diferente
279 -- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,
280 -- inventario, renta, actores y peliculas_actores
281 -- Cada SELECT es una instrucción independiente terminada con punto y coma (;)
282 • SELECT * FROM direccion;
283 • SELECT * FROM categoria;
284 • SELECT * FROM peliculas;
285 • SELECT * FROM cliente;
286 • SELECT * FROM inventario;
287 • SELECT * FROM renta;
288 -- Adicion de 2 Tablas
289 • SELECT * FROM actores;
290 • SELECT * FROM peliculas_actores;
<
```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: |

	Nombre	Duración	Descripción	Año	Categoría_idCategoria
1	Avengers: Endgame	120	Aventuras de Pikachu y sus amigos	2019	5
2	El Vago y el Joven	95	Drama	2020	3
3	Pelados y Fumosos	110	Comedia lejópolis	2001	1
4	Risas Eternas	88	Comedia familiar	2019	2
5	Galaxia Lejana	135	Aventura espacial	2022	4
•	NULL	NULL	NULL	NULL	NULL

dirección 1 categoría 2 peliculas 3 x cliente 4 inventario 5 renta 6 actores 7 peliculas\_actores 8

Output:

Action Output

#	Time	Action	Message
19	21:36:34	SELECT * FROM direccion LIMIT 0, 1000	5 row(s) returned
20	21:36:34	SELECT * FROM categoria LIMIT 0, 1000	5 row(s) returned
21	21:36:34	SELECT * FROM peliculas LIMIT 0, 1000	5 row(s) returned
22	21:36:35	SELECT * FROM cliente LIMIT 0, 1000	5 row(s) returned
23	21:36:35	SELECT * FROM inventario LIMIT 0, 1000	5 row(s) returned
24	21:36:35	SELECT * FROM renta LIMIT 0, 1000	5 row(s) returned
25	21:36:35	SELECT * FROM actores LIMIT 0, 1000	5 row(s) returned
26	21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned

MySQL Workbench

PreLaboratorio 1 - Warning ~

File Edit View Query Database Server Tools Scripting Help

Query 1

```

276 -- CONSULTA PARA MOSTRAR TODOS LOS DATOS:
277 -- SELECT * FROM: Comando para seleccionar todas las columnas de una tabla
278 -- Cada instrucción muestra el contenido completo de una tabla diferente
279 -- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,
280 -- inventario, renta, actores y peliculas_actores
281 -- Cada SELECT es una instrucción independiente terminada con punto y coma (;)
282 • SELECT * FROM direccion;
283 • SELECT * FROM categoria;
284 • SELECT * FROM peliculas;
285 • SELECT * FROM cliente;
286 • SELECT * FROM inventario;
287 • SELECT * FROM renta;
288 -- Adicion de 2 Tablas
289 • SELECT * FROM actores;
290 • SELECT * FROM peliculas_actores;
<
```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: |

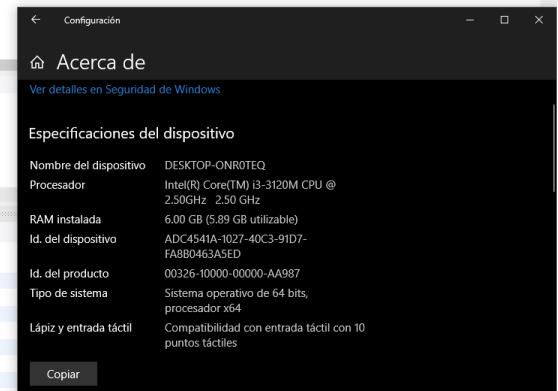
	Nombre	Apellido	Edad	Dirección_idDireccion
1	Juan	Pérez	25	1
2	Maria	Gómez	30	2
3	Juliana	Ramírez	22	3
4	Carlos	López	28	4
5	Ana	Morales	35	5
•	NULL	NULL	NULL	NULL

dirección 1 categoría 2 peliculas 3 x cliente 4 inventario 5 renta 6 actores 7 peliculas\_actores 8

Output:

Action Output

#	Time	Action	Message
19	21:36:34	SELECT * FROM direccion LIMIT 0, 1000	5 row(s) returned
20	21:36:34	SELECT * FROM categoria LIMIT 0, 1000	5 row(s) returned
21	21:36:34	SELECT * FROM peliculas LIMIT 0, 1000	5 row(s) returned
22	21:36:35	SELECT * FROM cliente LIMIT 0, 1000	5 row(s) returned
23	21:36:35	SELECT * FROM inventario LIMIT 0, 1000	5 row(s) returned
24	21:36:35	SELECT * FROM renta LIMIT 0, 1000	5 row(s) returned
25	21:36:35	SELECT * FROM actores LIMIT 0, 1000	5 row(s) returned
26	21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned



MySQL Workbench

PreLaboratorio 1 - Warning ~

File Edit View Query Database Server Tools Scripting Help

Query 1

```

276 -- CONSULTA PARA MOSTRAR TODOS LOS DATOS:
277 -- SELECT * FROM: Comando para seleccionar todas las columnas de una tabla
278 -- Cada instrucción muestra el contenido completo de una tabla diferente
279 -- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,
280 -- inventario, renta, actores y peliculas_actores
281 -- Cada SELECT es una instrucción independiente terminada con punto y coma (;)
282 • SELECT * FROM direccion;
283 • SELECT * FROM categoria;
284 • SELECT * FROM peliculas;
285 • SELECT * FROM cliente;
286 • SELECT * FROM inventario;
287 • SELECT * FROM renta;
288 -- Adicion de 2 Tablas
289 • SELECT * FROM actores;
290 • SELECT * FROM peliculas_actores;
<
```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: |

	Nombre	Apellido	Edad	Dirección_idDireccion
1	Juan	Pérez	25	1
2	Maria	Gómez	30	2
3	Juliana	Ramírez	22	3
4	Carlos	López	28	4
5	Ana	Morales	35	5
•	NULL	NULL	NULL	NULL

dirección 1 categoría 2 peliculas 3 x cliente 4 inventario 5 renta 6 actores 7 peliculas\_actores 8

Output:

Action Output

#	Time	Action	Message
19	21:36:34	SELECT * FROM direccion LIMIT 0, 1000	5 row(s) returned
20	21:36:34	SELECT * FROM categoria LIMIT 0, 1000	5 row(s) returned
21	21:36:34	SELECT * FROM peliculas LIMIT 0, 1000	5 row(s) returned
22	21:36:35	SELECT * FROM cliente LIMIT 0, 1000	5 row(s) returned
23	21:36:35	SELECT * FROM inventario LIMIT 0, 1000	5 row(s) returned
24	21:36:35	SELECT * FROM renta LIMIT 0, 1000	5 row(s) returned
25	21:36:35	SELECT * FROM actores LIMIT 0, 1000	5 row(s) returned
26	21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned

MySQL Workbench

PreLaboratorio 1 - Warning ~

File Edit View Query Database Server Tools Scripting Help

Query 1

```

276 -- CONSULTA PARA MOSTRAR TODOS LOS DATOS:
277 -- SELECT * FROM: Comando para seleccionar todas las columnas de una tabla
278 -- Cada instrucción muestra el contenido completo de una tabla diferente
279 -- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,
280 -- inventario, renta, actores y peliculas_actores
281 -- Cada SELECT es una instrucción independiente terminada con punto y coma (;)
282 • SELECT * FROM direccion;
283 • SELECT * FROM categoria;
284 • SELECT * FROM peliculas;
285 • SELECT * FROM cliente;
286 • SELECT * FROM inventario;
287 • SELECT * FROM renta;
288 -- Adicion de 2 Tablas
289 • SELECT * FROM actores;
290 • SELECT * FROM peliculas_actores;
<
```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: |

	Nombre	Apellido	Edad	Dirección_idDireccion
1	Juan	Pérez	25	1
2	Maria	Gómez	30	2
3	Juliana	Ramírez	22	3
4	Carlos	López	28	4
5	Ana	Morales	35	5
•	NULL	NULL	NULL	NULL

dirección 1 categoría 2 peliculas 3 x cliente 4 inventario 5 renta 6 actores 7 peliculas\_actores 8

Output:

Action Output

#	Time	Action	Message
19	21:36:34	SELECT * FROM direccion LIMIT 0, 1000	5 row(s) returned
20	21:36:34	SELECT * FROM categoria LIMIT 0, 1000	5 row(s) returned
21	21:36:34	SELECT * FROM peliculas LIMIT 0, 1000	5 row(s) returned
22	21:36:35	SELECT * FROM cliente LIMIT 0, 1000	5 row(s) returned
23	21:36:35	SELECT * FROM inventario LIMIT 0, 1000	5 row(s) returned
24	21:36:35	SELECT * FROM renta LIMIT 0, 1000	5 row(s) returned
25	21:36:35	SELECT * FROM actores LIMIT 0, 1000	5 row(s) returned
26	21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned

MySQL Workbench

Prelabatorio 1 - Warning ~

File Edit View Query Database Server Tools Scripting Help

Query 1

```

276 -- CONSULTA PARA MOSTRAR TODOS LOS DATOS:
277 -- SELECT * FROM: Comando para seleccionar todas las columnas de una tabla
278 -- Cada instrucción muestra el contenido completo de una tabla diferente
279 -- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,
280 -- inventario, renta, actores y peliculas_actores
281 -- Cada SELECT es una instrucción independiente terminada con punto y coma (;)
282 • SELECT * FROM direccion;
283 • SELECT * FROM categoria;
284 • SELECT * FROM peliculas;
285 • SELECT * FROM cliente;
286 • SELECT * FROM inventario;
287 • SELECT * FROM renta;
288 -- Adicion de 2 Tablas
289 • SELECT * FROM actores;
290 • SELECT * FROM peliculas_actores;
<
```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: |

idCopiasPeliculas	Pelicula_idPeliculas	Disponible
1	1	1
2	2	1
3	3	0
4	4	1
5	5	1
• NULL	NULL	NULL

dirección 1 categoría 2 peliculas 3 cliente 4 inventario 5 x renta 6 actores 7 peliculas\_actores 8

Action Output

#	Time	Action	Message
19	21:36:34	SELECT * FROM direccion LIMIT 0, 1000	5 row(s) returned
20	21:36:34	SELECT * FROM categoria LIMIT 0, 1000	5 row(s) returned
21	21:36:34	SELECT * FROM peliculas LIMIT 0, 1000	5 row(s) returned
22	21:36:35	SELECT * FROM cliente LIMIT 0, 1000	5 row(s) returned
23	21:36:35	SELECT * FROM inventario LIMIT 0, 1000	5 row(s) returned
24	21:36:35	SELECT * FROM renta LIMIT 0, 1000	5 row(s) returned
25	21:36:35	SELECT * FROM actores LIMIT 0, 1000	5 row(s) returned
26	21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned

Configuración

Acerca de

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONR0TEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FA880463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar

MySQL Workbench

Prelabatorio 1 - Warning ~

File Edit View Query Database Server Tools Scripting Help

Query 1

```

276 -- CONSULTA PARA MOSTRAR TODOS LOS DATOS:
277 -- SELECT * FROM: Comando para seleccionar todas las columnas de una tabla
278 -- Cada instrucción muestra el contenido completo de una tabla diferente
279 -- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,
280 -- inventario, renta, actores y peliculas_actores
281 -- Cada SELECT es una instrucción independiente terminada con punto y coma (;)
282 • SELECT * FROM direccion;
283 • SELECT * FROM categoria;
284 • SELECT * FROM peliculas;
285 • SELECT * FROM cliente;
286 • SELECT * FROM inventario;
287 • SELECT * FROM renta;
288 -- Adicion de 2 Tablas
289 • SELECT * FROM actores;
290 • SELECT * FROM peliculas_actores;
<
```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: |

iRenta	Fecha_Renta	Fecha_Entrega	Inventario_idCopiasPeliculas	Cliente_idCliente
1	2023-01-15	2023-01-20	1	1
2	2023-02-10	2023-02-15	2	2
3	2023-03-05	2023-03-10	3	3
4	2023-04-20	2023-04-25	4	4
5	2023-05-15	2023-05-20	5	5
• NULL	NULL	NULL	NULL	NULL

dirección 1 categoría 2 peliculas 3 cliente 4 inventario 5 x renta 6 actores 7 peliculas\_actores 8

Action Output

#	Time	Action	Message
19	21:36:34	SELECT * FROM direccion LIMIT 0, 1000	5 row(s) returned
20	21:36:34	SELECT * FROM categoria LIMIT 0, 1000	5 row(s) returned
21	21:36:34	SELECT * FROM peliculas LIMIT 0, 1000	5 row(s) returned
22	21:36:35	SELECT * FROM cliente LIMIT 0, 1000	5 row(s) returned
23	21:36:35	SELECT * FROM inventario LIMIT 0, 1000	5 row(s) returned
24	21:36:35	SELECT * FROM renta LIMIT 0, 1000	5 row(s) returned
25	21:36:35	SELECT * FROM actores LIMIT 0, 1000	5 row(s) returned
26	21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned

Configuración

Acerca de

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONR0TEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FA880463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar

MySQL Workbench

PreLaboratorio 1 - Warning -

File Edit View Query Database Server Tools Scripting Help

Query 1

```

276 -- CONSULTA PARA MOSTRAR TODOS LOS DATOS:
277 -- SELECT * FROM: Comando para seleccionar todas las columnas de una tabla
278 -- Cada instrucción muestra el contenido completo de una tabla diferente
279 -- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,
280 -- inventario, renta, actores y peliculas_actores
281 -- Cada SELECT es una instrucción independiente terminada con punto y coma (;)

282 • SELECT * FROM direccion;
283 • SELECT * FROM categoria;
284 • SELECT * FROM peliculas;
285 • SELECT * FROM cliente;
286 • SELECT * FROM inventario;
287 • SELECT * FROM renta;
288 -- Adicion de 2 Tablas
289 • SELECT * FROM actores;
290 • SELECT * FROM peliculas_actores;

```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: |

Actor	Nombre	Apellido	FechadNacimiento	Nacionalidad
1	Ronald	Reynor Jr.	1985-04-01	Estadounidense
2	Scarlett	Johansson	1984-11-22	Estadounidense
3	Juliana	Moore	1960-12-03	Estadounidense
4	Dwayne	Johnson	1972-05-02	Estadounidense
5	Margot	Robbie	1990-07-02	Australiana
•	NULL	NULL	NULL	NULL

dirección 1 categoría 2 películas 3 cliente 4 inventario 5 renta 6 actores 7 x pelícualas\_actores 8

Action Output

Time	Action	Message
19 21:36:34	SELECT * FROM direccion LIMIT 0, 1000	5 row(s) returned
20 21:36:34	SELECT * FROM categoria LIMIT 0, 1000	5 row(s) returned
21 21:36:34	SELECT * FROM peliculas LIMIT 0, 1000	5 row(s) returned
22 21:36:35	SELECT * FROM cliente LIMIT 0, 1000	5 row(s) returned
23 21:36:35	SELECT * FROM inventario LIMIT 0, 1000	5 row(s) returned
24 21:36:35	SELECT * FROM renta LIMIT 0, 1000	5 row(s) returned
25 21:36:35	SELECT * FROM actores LIMIT 0, 1000	5 row(s) returned
26 21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned

MySQL Workbench

PreLaboratorio 1 - Warning -

File Edit View Query Database Server Tools Scripting Help

Query 1

```

276 -- CONSULTA PARA MOSTRAR TODOS LOS DATOS:
277 -- SELECT * FROM: Comando para seleccionar todas las columnas de una tabla
278 -- Cada instrucción muestra el contenido completo de una tabla diferente
279 -- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,
280 -- inventario, renta, actores y peliculas_actores
281 -- Cada SELECT es una instrucción independiente terminada con punto y coma (;)

282 • SELECT * FROM direccion;
283 • SELECT * FROM categoria;
284 • SELECT * FROM peliculas;
285 • SELECT * FROM cliente;
286 • SELECT * FROM inventario;
287 • SELECT * FROM renta;
288 -- Adicion de 2 Tablas
289 • SELECT * FROM actores;
290 • SELECT * FROM peliculas_actores;

```

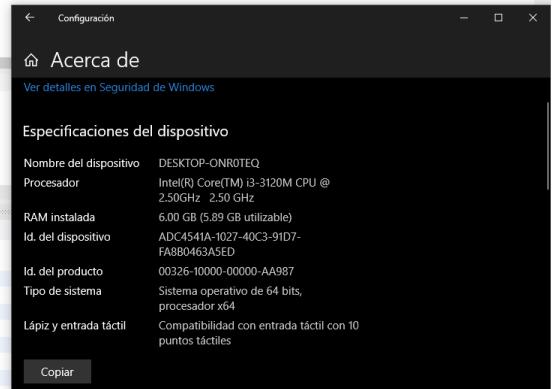
Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: |

IdResidencia	Peliculas_idPelicula	Actores_idActor	Personaje
1	2	4	Luke Hobbs
2	2	3	Maria Gonzales
3	5	2	Captana Vega
4	4	5	Lisa Johnson
5	1	1	Dr. Pikachu (voz)
•	NULL	NULL	NULL

dirección 1 categoría 2 películas 3 cliente 4 inventario 5 renta 6 actores 7 x pelícualas\_actores 8

Action Output

Time	Action	Message
19 21:36:34	SELECT * FROM direccion LIMIT 0, 1000	5 row(s) returned
20 21:36:34	SELECT * FROM categoria LIMIT 0, 1000	5 row(s) returned
21 21:36:34	SELECT * FROM peliculas LIMIT 0, 1000	5 row(s) returned
22 21:36:35	SELECT * FROM cliente LIMIT 0, 1000	5 row(s) returned
23 21:36:35	SELECT * FROM inventario LIMIT 0, 1000	5 row(s) returned
24 21:36:35	SELECT * FROM renta LIMIT 0, 1000	5 row(s) returned
25 21:36:35	SELECT * FROM actores LIMIT 0, 1000	5 row(s) returned
26 21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned



MySQL Workbench

PreLaboratorio 1 - Warning -

File Edit View Query Database Server Tools Scripting Help

Query 1

```

276 -- CONSULTA PARA MOSTRAR TODOS LOS DATOS:
277 -- SELECT * FROM: Comando para seleccionar todas las columnas de una tabla
278 -- Cada instrucción muestra el contenido completo de una tabla diferente
279 -- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,
280 -- inventario, renta, actores y peliculas_actores
281 -- Cada SELECT es una instrucción independiente terminada con punto y coma (;)

282 • SELECT * FROM direccion;
283 • SELECT * FROM categoria;
284 • SELECT * FROM peliculas;
285 • SELECT * FROM cliente;
286 • SELECT * FROM inventario;
287 • SELECT * FROM renta;
288 -- Adicion de 2 Tablas
289 • SELECT * FROM actores;
290 • SELECT * FROM peliculas_actores;

```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: |

IdResidencia	Peliculas_idPelicula	Actores_idActor	Personaje
1	2	4	Luke Hobbs
2	2	3	Maria Gonzales
3	5	2	Captana Vega
4	4	5	Lisa Johnson
5	1	1	Dr. Pikachu (voz)
•	NULL	NULL	NULL

dirección 1 categoría 2 películas 3 cliente 4 inventario 5 renta 6 actores 7 x pelícualas\_actores 8

Action Output

Time	Action	Message
19 21:36:34	SELECT * FROM direccion LIMIT 0, 1000	5 row(s) returned
20 21:36:34	SELECT * FROM categoria LIMIT 0, 1000	5 row(s) returned
21 21:36:34	SELECT * FROM peliculas LIMIT 0, 1000	5 row(s) returned
22 21:36:35	SELECT * FROM cliente LIMIT 0, 1000	5 row(s) returned
23 21:36:35	SELECT * FROM inventario LIMIT 0, 1000	5 row(s) returned
24 21:36:35	SELECT * FROM renta LIMIT 0, 1000	5 row(s) returned
25 21:36:35	SELECT * FROM actores LIMIT 0, 1000	5 row(s) returned
26 21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned

## -- CONSULTA PARA MOSTRAR TODOS LOS DATOS:

-- SELECT \* FROM: Comando para seleccionar todas las columnas de una tabla

-- Cada instrucción muestra el contenido completo de una tabla diferente

-- Se ejecutan secuencialmente para mostrar: direccion, categoria, peliculas, cliente,

-- inventario, renta, actores y peliculas\_actores

-- Cada SELECT es una instrucción independiente terminada con punto y coma (;)

```
SELECT * FROM direccion;
```

```
SELECT * FROM categoria;
```

```
SELECT * FROM peliculas;
```

```
SELECT * FROM cliente;
```

```
SELECT * FROM inventario;
```

```
SELECT * FROM renta;
```

## -- Adición de 2 Tablas

```
SELECT * FROM actores;
```

```
SELECT * FROM peliculas_actores;
```

The screenshot shows the MySQL Workbench interface with a query editor and a results grid.

**Query Editor:**

```
300 -- 1. El primer % = puede haber cualquier texto antes de "Juliana" (o nada)
301 -- 2. "Juliana" = debe aparecer exactamente así en el nombre
302 -- 3. El segundo % = puede haber cualquier texto después de "Juliana" (o nada)
303 -- UNION ALL: Combina con resultados de otra consulta similar (mantiene duplicados)
304 -- La segunda parte busca en películas donde el título contenga 'Juliana'
305 -- La tercera parte busca en actores cuyo nombre contenga 'Juliana'
306 -- Cada parte del UNION ALL debe tener el mismo número y tipo de columnas
307 -- El punto y coma (;) finaliza toda la instrucción compuesta
308 • SELECT 'Cliente' AS tipo, idCliente AS id, CONCAT(Nombre, ' ', Apellido) AS nombre
FROM cliente
WHERE Nombre LIKE '%Juliana%'

UNION ALL

314 SELECT 'Pelicula' AS tipo, idPeliculas AS id, Nombre
FROM peliculas
WHERE Nombre LIKE '%Juliana%'

UNION ALL

320 SELECT 'Actor/Actriz' AS tipo, idActor AS id, CONCAT(Nombre, ' ', Apellido) AS nombre
FROM actores
WHERE Nombre LIKE '%Juliana%';

```

**Result Grid:**

tipo	id	nombre
Cliente	3	Juliana Ramirez
Pelicula	2	El Viejito de Juliana
Actor/Actriz	3	Juliana Moore

**Action Output:**

#	Time	Action	Message
26	21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned
27	21:46:55	SELECT Cliente' AS tipo, idCliente AS id, CONCAT(Nombre, ' ', Apellido) AS nombre FROM cliente WHERE Nombre LIKE '%Juliana%' UNION ALL SELECT 'Pelicula' AS tipo, idPeliculas AS id, Nombre FROM peliculas WHERE Nombre LIKE '%Juliana%'	3 row(s) returned

**System Configuration Window:**

Acerca de

Nombre del dispositivo: DESKTOP-QNROTEQ

Procesador: Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz

RAM instalada: 6.00 GB (5.89 GB utilizable)

Id. del dispositivo: ADC4541A-1027-40C3-91D7-FAB80463A5ED

Id. del producto: 00326-10000-00000-AA987

Tipo de sistema: Sistema operativo de 64 bits, procesador x64

Lápiz y entrada táctil: Compatibilidad con entrada táctil con 10 puntos táctiles

## -- CONSULTA PARA BUSCAR "JULIANA" EN TABLAS RELEVANTES:

-- SELECT: Inicia una consulta para buscar coincidencias

-- 'Cliente' AS tipo: Crea columna temporal con valor constante 'Cliente'

-- idCliente AS id: Muestra el ID del cliente con alias 'id'

-- CONCAT(Nombre, ' ', Apellido) AS nombre: Combina nombre y apellido en una columna llamada nombre

-- FROM cliente: Especifica la tabla de origen

-- WHERE Nombre LIKE '%Juliana%': Filtra registros donde el nombre contenga 'Juliana'

- '%Juliana%' significa:
  - 1. El primer % = puede haber cualquier texto antes de "Juliana" (o nada)
  - 2. "Juliana" = debe aparecer exactamente así en el nombre
  - 3. El segundo % = puede haber cualquier texto después de "Juliana" (o nada)
- UNION ALL: Combina con resultados de otra consulta similar (mantiene duplicados)
- La segunda parte busca en películas donde el título contenga 'Juliana'
- La tercera parte busca en actores cuyo nombre contenga 'Juliana'
- Cada parte del UNION ALL debe tener el mismo número y tipo de columnas
- El punto y coma (;) finaliza toda la instrucción compuesta

```
SELECT 'Cliente' AS tipo, idCliente AS id, CONCAT(Nombre, ' ', Apellido) AS nombre
FROM cliente
WHERE Nombre LIKE '%Juliana%'
```

UNION ALL

```
SELECT 'Película' AS tipo, idPeliculas AS id, Nombre
FROM peliculas
WHERE Nombre LIKE '%Juliana%'
```

UNION ALL

```
SELECT 'Actor/Actriz' AS tipo, idActor AS id, CONCAT(Nombre, ' ', Apellido) AS nombre
FROM actores
WHERE Nombre LIKE '%Juliana%';
```

The screenshot shows the MySQL Workbench interface. The main window displays a SQL query in the 'Query 1' tab:

```

318 UNION ALL
319
320 SELECT 'Actor/Actriz' AS tipo, idActor AS id, CONCAT(Nombre, ' ', Apellido) AS nombre
321 FROM actores
322 WHERE Nombre LIKE '%Juliana%';
323
324 -- ELIMINACIÓN LÓGICA DE POKEMON1 USANDO UPDATE:
325 -- UPDATE: Comando para modificar registros existentes
326 -- peliculas: Tabla que contiene los registros a modificar
327 -- SET Nombre = null: Asigna como vacía al campo Nombre (eliminación lógica)
328 -- WHERE Nombre = 'POKEMON1': Condición que selecciona solo ese registro específico
329 -- Esto mantiene todos los demás campos y relaciones intactas
330 -- El punto y coma (;) finaliza la instrucción
331 • UPDATE peliculas
332 SET Nombre = null
333 WHERE Nombre = 'POKEMON1';
334
335 -- VERIFICACIÓN DEL CAMBIO:
336 -- SELECT * FROM peliculas: Muestra todo el contenido de la tabla peliculas
337 -- Permite confirmar visualmente que:
338 -- 1. El registro que era POKEMON1 ahora tiene nombre vacío
339 -- 2. Todos los demás campos (ID, duración, etc.) se mantienen igual
340 -- 3. La relación con categoría (ID 5 - Animación) se preserva
341 -- El punto y coma (;) finaliza la instrucción
342 • SELECT * FROM peliculas;
343
344 -- CONSULTA PARA ORDENAR CATEGORÍAS ASCENDENTEMENTE:
345 -- SELECT * FROM categoria: Selecciona todas las columnas de la tabla categoria
346 -- ORDER BY Nombre ASC: Ordena los resultados por el campo Nombre en orden A-Z
347 -- ASC es opcional (es el orden por defecto) pero se incluye para claridad

```

The status bar at the bottom shows two log entries:

- 27 21:46:55 SELECT 'Cliente' AS tipo, idCliente AS id, CONCAT(Nombre, ' ', Apellido) AS nombre FROM cliente WHERE Nombre LIKE '%Juliana%' UNION ALL SELECT 'Película' AS tipo, id AS id, Nombre AS nombre FROM peliculas WHERE Nombre = 'POKEMON1'
- 28 21:48:16 UPDATE peliculas SET Nombre = null WHERE Nombre = 'POKEMON1'

A configuration window titled 'Configuración' is open on the right side, showing device specifications:

Especificaciones del dispositivo	
Nombre del dispositivo	DESKTOP-ONROTEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FA880463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

## -- ELIMINACIÓN LÓGICA DE POKEMON1 USANDO UPDATE:

-- UPDATE: Comando para modificar registros existentes

-- peliculas: Tabla que contiene los registros a modificar

-- SET Nombre = null: Asigna como vacía al campo Nombre (eliminación lógica)

-- WHERE Nombre = 'POKEMON1': Condición que selecciona solo ese registro específico

-- Esto mantiene todos los demás campos y relaciones intactas

-- El punto y coma (;) finaliza la instrucción

UPDATE peliculas

SET Nombre = null

WHERE Nombre = 'POKEMON1';

The screenshot shows the MySQL Workbench interface. The main window displays a query editor with the following SQL code:

```

327 -- SET Nombre = null: Asigna como vacía al campo Nombre (eliminación lógica)
328 -- WHERE Nombre = 'POKEMON1': Condición que selecciona solo ese registro específico
329 -- Esto mantiene todos los demás campos y relaciones intactas
330 -- El punto y coma (;) finaliza la instrucción
331 • UPDATE peliculas
332   SET Nombre = null
333 WHERE Nombre = 'POKEMON1';
334
335 -- VERIFICACIÓN DEL CAMBIO:
336 -- SELECT * FROM peliculas; Muestra todo el contenido de la tabla peliculas
337 -- Permite confirmar visualmente que:
338 -- 1. El registro que era POKEMON1 ahora tiene nombre vacío
339 -- 2. Todos los demás campos (ID, duración, etc.) se mantienen igual
340 -- 3. La relación con categoría (ID 5 - Animación) se preserva
341 -- El punto y coma (;) finaliza la instrucción
342 • SELECT * FROM peliculas;
343
344 -- CONSULTA PARA ORDENAR CATEGORÍAS ASCENDENTEMENTE:
345 -- SELECT * FROM categoria; Selecciona todas las columnas de la tabla categoria
346 -- ORDER BY Nombre ASC: Ordena los resultados por el campo Nombre en orden A-Z
347 -- ASC es opcional (es el orden por defecto) pero se incluye para claridad
348 -- El punto y coma (;) finaliza la instrucción

```

The results grid shows the following data for the 'peliculas' table:

ID	Título	Duración	Descripción	Año	Categoría_idCategoría
1	NULL	120	Aventuras de Pikachu y sus amigos	1998	5
2	El Vaje de Juliana	95	Drama familiar	2020	3
3	Rápidos y Furiosos	110	Carreras ilegales	2001	1
4	Risas Eternas	88	Comedia familiar	2019	2
5	Galaxia Lejana	135	Aventura espacial	2022	4
6	NULL	NULL	NULL	NULL	NULL

The configuration window on the right shows the following details:

- Nombre del dispositivo: DESKTOP-ONROTEQ
- Procesador: Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
- RAM instalada: 6.00 GB (5.89 GB utilizable)
- Id. del dispositivo: ADC4541A-1027-40C3-91D7-F4880463A5ED
- Id. del producto: 00326-10000-00000-AA987
- Tipo de sistema: Sistema operativo de 64 bits, procesador x64
- Lápiz y entrada táctil: Compatibilidad con entrada táctil con 10 puntos táctiles

## -- VERIFICACIÓN DEL CAMBIO:

-- SELECT \* FROM peliculas: Muestra todo el contenido de la tabla peliculas

-- Permite confirmar visualmente que:

-- 1. El registro que era POKEMON1 ahora tiene nombre vacío

-- 2. Todos los demás campos (ID, duración, etc.) se mantienen igual

-- 3. La relación con categoría (ID 5 - Animación) se preserva

-- El punto y coma (;) finaliza la instrucción

SELECT \* FROM peliculas;

MySQL Workbench

PreLaboratorio 1 - Warning -

File Edit View Query Database Server Tools Scripting Help

Query 1

```

328 -- WHERE Nombre = 'POKEMON1': Condición que selecciona solo ese registro específico
329 -- Esto mantiene todos los demás campos y relaciones intactas
330 -- El punto y coma (;) finaliza la instrucción
331 • UPDATE peliculas
332 SET Nombre = null
333 WHERE Nombre = 'POKEMON1';
334
335 -- VERIFICACIÓN DEL CAMBIO:
336 -- SELECT * FROM peliculas: Muestra todo el contenido de la tabla peliculas
337 -- Permite confirmar visualmente que:
338 -- 1. El registro que era POKEMON1 ahora tiene nombre vacío
339 -- 2. Todos los demás campos (ID, duración, etc.) se mantienen igual
340 -- 3. La relación con categoría (ID 5 - Animación) se preserva
341 -- El punto y coma (;) finaliza la instrucción
342 • SELECT * FROM peliculas;
343
344 -- CONSULTA PARA ORDENAR CATEGORÍAS ASCENDENTEMENTE:
345 -- SELECT * FROM categoria: Selecciona todas las columnas de la tabla categoria
346 -- ORDER BY Nombre ASC: Ordena los resultados por el campo Nombre en orden A-Z
347 -- ASC es opcional (es el orden por defecto) pero se incluye para claridad
348 -- El punto y coma (;) finaliza la instrucción
349 • SELECT * FROM categoria ORDER BY Nombre ASC;
    
```

Result Grid

IDCategoria	Nombre
1	Acción
5	Animación
4	Ciencia Ficción
2	Comedia
3	Drama
6	Romántica

Output

Action Output

Time	Action	Message
29 21:48:50	SELECT * FROM peliculas LIMIT 0, 1000	5 row(s) returned
30 21:50:06	SELECT * FROM categoria ORDER BY Nombre ASC LIMIT 0, 1000	5 row(s) returned

Configuración

Acerca de

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONROTEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FA880463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

-- CONSULTA PARA ORDENAR CATEGORÍAS ASCENDENTEMENTE:

-- SELECT \* FROM categoria: Selecciona todas las columnas de la tabla categoria

-- ORDER BY Nombre ASC: Ordena los resultados por el campo Nombre en orden A-Z

-- ASC es opcional (es el orden por defecto) pero se incluye para claridad

-- El punto y coma (;) finaliza la instrucción

SELECT \* FROM categoria ORDER BY Nombre ASC;

MySQL Workbench

PreLaboratorio 1 - Warning -

File Edit View Query Database Server Tools Scripting Help

Query 1

```

335 -- VERIFICACIÓN DEL CAMBIO:
336 -- SELECT * FROM peliculas; Muestra todo el contenido de la tabla peliculas
337 -- Permite confirmar visualmente que:
338 -- 1. El registro que era POKEMONI ahora tiene nombre vacío
339 -- 2. Todos los demás campos (ID, duración, etc.) se mantienen igual
340 -- 3. La relación con categoría (ID 5 - Animación) se preserva
341 -- El punto y coma (;) finaliza la instrucción
342 • SELECT * FROM peliculas;
343
344 -- CONSULTA PARA ORDENAR CATEGORÍAS ASCENDENTEMENTE:
345 -- SELECT * FROM categoria; Selecciona todas las columnas de la tabla categoria
346 -- ORDER BY Nombre ASC: Ordena los resultados por el campo Nombre en orden A-Z
347 -- ASC es opcional (es el orden por defecto) pero se incluye para claridad
348 -- El punto y coma (;) finaliza la instrucción
349 • SELECT * FROM categoria ORDER BY Nombre ASC;
350
351 -- CONSULTA PARA ORDENAR PELÍCULAS POR AÑO DESCENDENTEMENTE:
352 -- SELECT * FROM peliculas; Selecciona todas las columnas de la tabla peliculas
353 -- ORDER BY Año DESC: Ordena los resultados por el campo Año en orden Z-A (nuevas primero)
354 -- DESC es necesario para invertir el orden por defecto (ascendente)
355 -- El punto y coma (;) finaliza la instrucción
356 • SELECT * FROM peliculas ORDER BY Año DESC;

```

Result Grid

#	peliculas	Nombre	Duración	Descripción	Año	Categoría_idCategoria
5	Galaxia Lejana	135	Aventura espacial	2022	4	
2	El Vaje de Juliana	95	Drama familiar	2020	3	
4	Risas Eternas	88	Comedia familiar	2019	2	
3	Rápidos y Furiosos	110	Carreras ilegales	2001	1	
1	NULL	120	Aventuras de Pikachu y sus amigos	1998	5	
•	NULL	NULL	NULL	NULL	NULL	NULL

películas 12 x

Output

Action Output

Time Action

30 21:50:06 SELECT \* FROM categoria ORDER BY Nombre ASC LIMIT 0, 1000  
31 21:51:43 SELECT \* FROM peliculas ORDER BY Año DESC LIMIT 0, 1000

Message 5 row(s) returned  
5 row(s) returned

Configuración

Acerca de

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONROTEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADC4541A-1027-40C3-91D7-FA880463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10 puntos táctiles

Copiar

-- CONSULTA PARA ORDENAR PELÍCULAS POR AÑO DESCENDENTEMENTE:

-- SELECT \* FROM peliculas: Selecciona todas las columnas de la tabla peliculas

-- ORDER BY Año DESC: Ordena los resultados por el campo Año en orden Z-A (nuevas primero)

-- DESC es necesario para invertir el orden por defecto (ascendente)

-- El punto y coma (;) finaliza la instrucción

SELECT \* FROM peliculas ORDER BY Año DESC;

MySQL Workbench

ReLaboratorio 1 - Warning ~

File Edit View Query Database Server Tools Scripting Help

Query 1

7 • CREATE DATABASE IF NOT EXISTS sistema\_renta\_peliculas;

8

9 -- SELECCIÓN DE LA BASE DE DATOS:

10 -- USE: Comando para especificar qué base de datos usaremos en las siguientes operaciones

Output

Action Output

#	Time	Action	Message
1	21:20:00	CREATE DATABASE IF NOT EXISTS sistema_renta_peliculas	1 row(s) affected
2	21:20:00	USE sistema_renta_peliculas	0 row(s) affected
3	21:21:00	CREATE TABLE direccion ( idDireccion INT(11) PRIMARY KEY, Direccion VARCHAR(45), Ciudad VARCHAR(45), Pais VARCHAR(45) )	0 row(s) affected
4	21:22:36	CREATE TABLE categoria ( idCategoria INT(11) PRIMARY KEY, Nombre VARCHAR(45) )	0 row(s) affected
5	21:23:16	CREATE TABLE pelicula ( idPelicula INT(11) PRIMARY KEY, Nombre VARCHAR(45), Duracion INT(11), Descripcion TEXT, Año INT(11), Categoria_idCategoria INT(11) )	0 row(s) affected
6	21:24:30	CREATE TABLE cliente ( idCliente INT(11) PRIMARY KEY, Nombre VARCHAR(45), Apellido VARCHAR(45), Edad INT(11), Direccion_idDireccion INT(11), FO... )	0 row(s) affected
7	21:25:13	CREATE TABLE inventario ( idCopiasPelicula INT(11) PRIMARY KEY, Peliculas_idPelicula INT(11), Disponible TINYINT(1) )	0 row(s) affected
8	21:25:41	CREATE TABLE renta ( idRenta INT(11) PRIMARY KEY, Fecha_Renta DATE, Fecha_Entrega DATE, Inventario_idCopiasPelicula INT(11), Cliente_idCliente IN... )	0 row(s) affected
9	21:26:25	CREATE TABLE actores ( idActor INT(11) PRIMARY KEY, Nombre VARCHAR(45), Apellido VARCHAR(45), FechaNacimiento DATE, Nacionalidad VARCHAR(45) )	0 row(s) affected
10	21:27:44	CREATE TABLE peliculas_actores ( idRelacion INT(11) PRIMARY KEY, Peliculas_idPelicula INT(11), Actores_idActor INT(11), Personaje VARCHAR(45), FOREIGN KEY (Peliculas_idPelicula) REFERENCES pelicula(idPelicula), FOREIGN KEY (Actores_idActor) REFERENCES actores(idActor) )	0 row(s) affected
11	21:29:40	INSERT INTO direccion (idDireccion, Direccion, Ciudad, Pais) VALUES (1, 'Calle Principal 123', 'Ciudad de México', '12345'), (2, 'Avenida Revolución 456', 'Guadalajara', '67890')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
12	21:29:36	INSERT INTO categoria (idCategoria, Nombre) VALUES (1, 'Acción'), (2, 'Comedia'), (3, 'Drama'), (4, 'Ciencia Ficción'), (5, 'Aventura')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
13	21:30:03	INSERT INTO pelicula (Nombre, Duracion, Descripcion, Año, Categoria_idCategoria) VALUES (1, 'POKEMON1', 120, 'Aventuras de Pikachu y sus amigos', 1998)	0 row(s) affected
14	21:30:33	INSERT INTO cliente (idCliente, Nombre, Apellido, Edad, Direccion_idDireccion) VALUES (1, 'Juan', 'Pérez', 25, 1), (2, 'María', 'Gómez', 30, 2), (3, 'Julieta', 'Ramírez', 22, 3), (4, 'Luis', 'García', 28, 4), (5, 'Ana', 'Gómez', 35, 5)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
15	21:31:14	INSERT INTO inventario (idCopiasPelicula, Peliculas_idPelicula, Disponible) VALUES (1, 1, 1), (2, 2, 1), (3, 3, 0), (4, 4, 1), (5, 5, 1)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
16	21:32:04	INSERT INTO renta (idRenta, Fecha_Renta, Fecha_Entrega, Inventario_idCopiasPelicula, Cliente_idCliente) VALUES (1, '2023-01-15', '2023-01-20', 1, 1), (2, '2023-02-10', 2, 2)	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
17	21:33:09	INSERT INTO actores (idActor, Nombre, Apellido, FechaNacimiento, Nacionalidad) VALUES (1, 'Robert Downey Jr.', '1965-04-04', 'Estadounidense'), (2, 'Scarlett Johansson', '1984-08-22', 3)	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
18	21:35:48	INSERT INTO peliculas_actores (idRelacion, Peliculas_idPelicula, Actores_idActor, Personaje) VALUES (1, 3, 4, 'Luke Hobbs'), (2, 2, 3, 'Maria González'), (3, 5, 2, 'Capitana Marvel')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0
19	21:36:34	SELECT * FROM direccion LIMIT 0, 1000	5 row(s) returned
20	21:36:34	SELECT * FROM categoria LIMIT 0, 1000	5 row(s) returned
21	21:36:34	SELECT * FROM pelicula LIMIT 0, 1000	5 row(s) returned
22	21:36:35	SELECT * FROM cliente LIMIT 0, 1000	5 row(s) returned
23	21:36:35	SELECT * FROM inventario LIMIT 0, 1000	5 row(s) returned
24	21:36:35	SELECT * FROM renta LIMIT 0, 1000	5 row(s) returned
25	21:36:35	SELECT * FROM actores LIMIT 0, 1000	5 row(s) returned
26	21:36:35	SELECT * FROM peliculas_actores LIMIT 0, 1000	5 row(s) returned
27	21:46:55	SELECT Cliente AS tipo, Cliente AS id, CONCAT(Nombre, ' ', Apellido) AS nombre FROM cliente WHERE Nombre LIKE '%Juliana%' UNION ALL SELECT Pelicula AS tipo, Pelicula AS id, Nombre AS nombre FROM pelicula WHERE Nombre LIKE '%Juliana%'	3 row(s) returned
28	21:48:16	UPDATE pelicula SET Nombre = null WHERE Nombre = 'POKEMON1'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
29	21:48:50	SELECT * FROM pelicula LIMIT 0, 1000	5 row(s) returned
30	21:50:06	SELECT * FROM categoria ORDER BY Nombre ASC LIMIT 0, 1000	5 row(s) returned
31	21:51:43	SELECT * FROM peliculas ORDER BY Año DESC LIMIT 0, 1000	5 row(s) returned

Configuración

Acerca de

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo: DESKTOP-ONR0TEQ

Procesador: Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz, 2.50 GHz

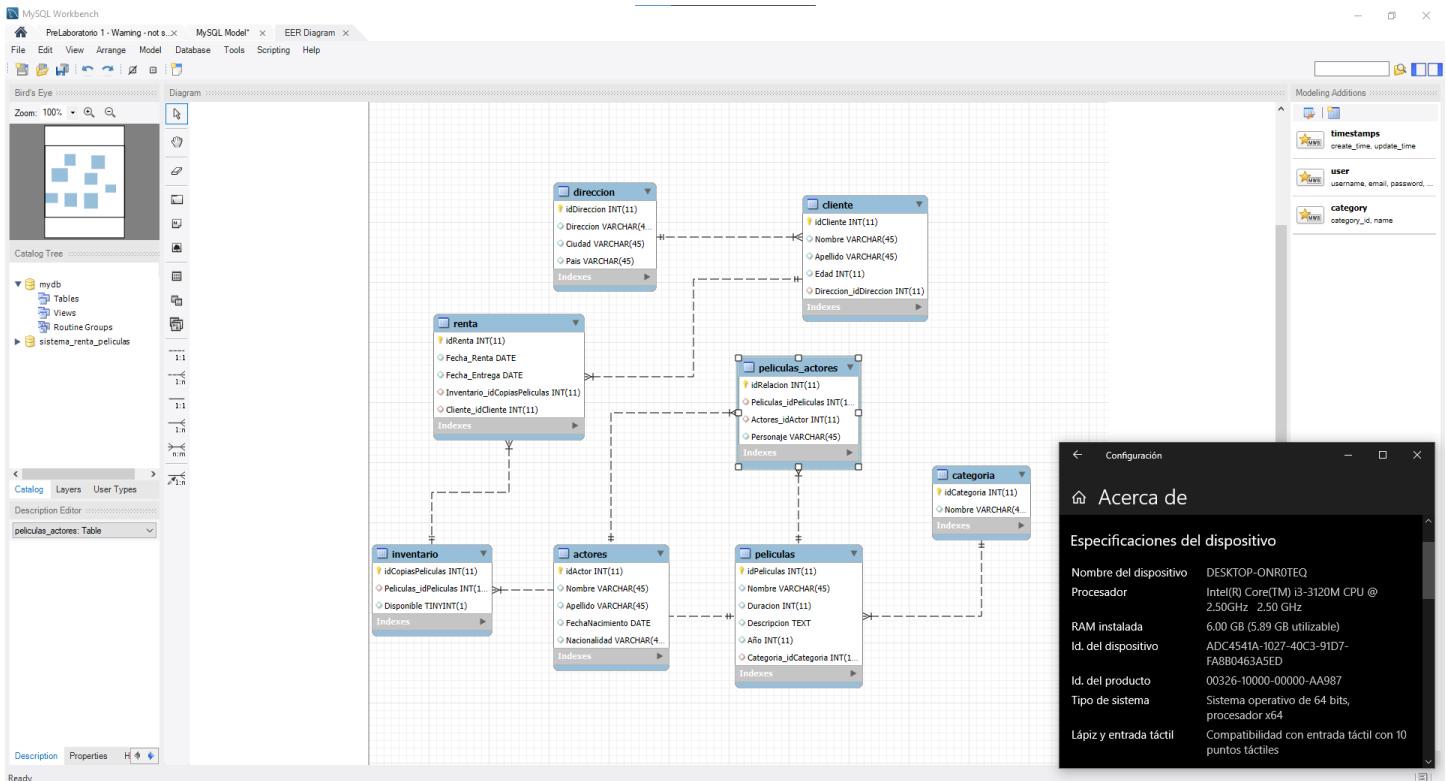
RAM instalada: 6.00 GB (5.89 GB utilizable)

Id. del dispositivo: ADC4541A-1027-40C3-91D7-FA8B0463A5ED

Id. del producto: 00326-10000-00000-AA987

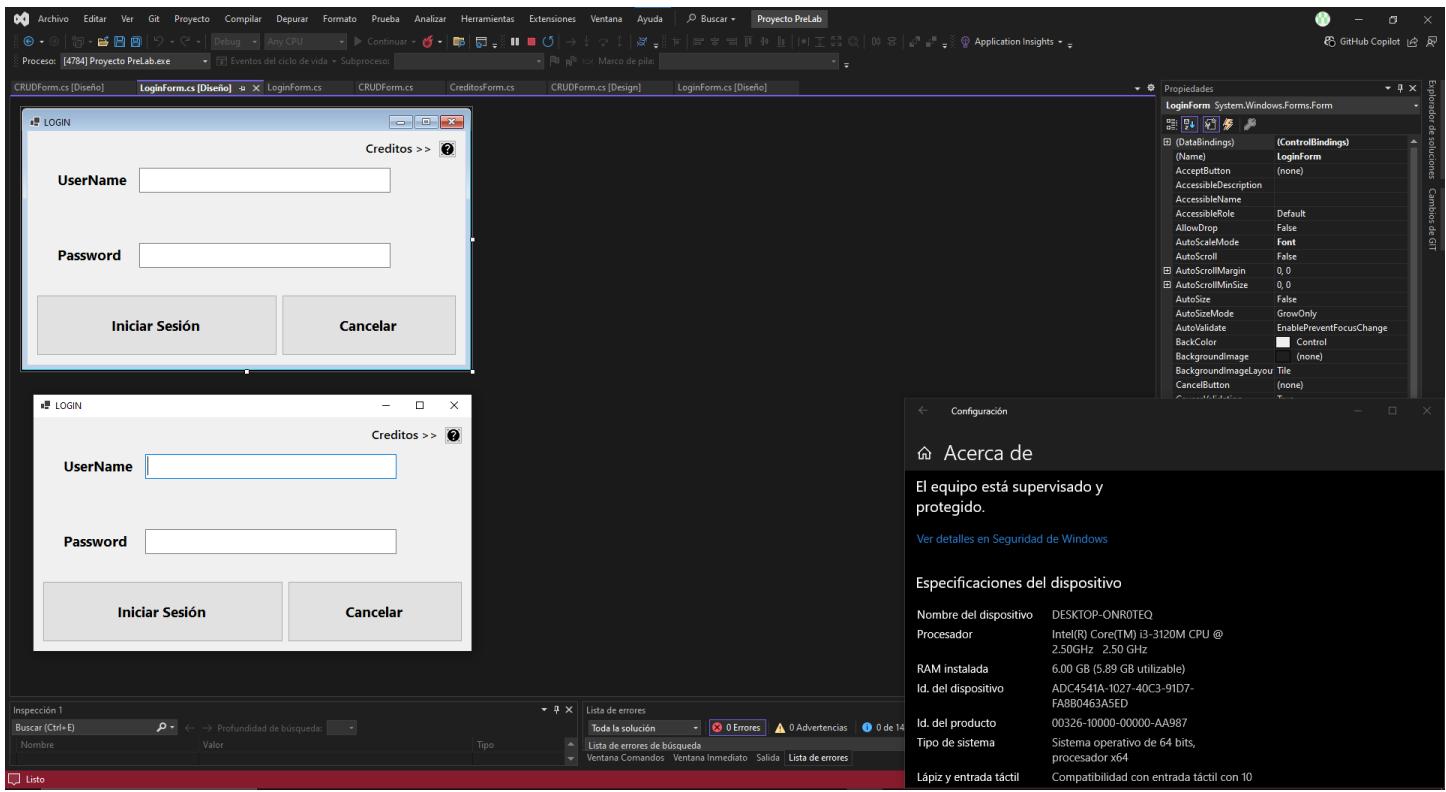
Tipo de sistema: Sistema operativo de 64 bits, procesador x64

Lápiz y entrada táctil: Compatibilidad con entrada táctil con 10 puntos táctiles



**VISUAL  
CRUD**

Q #



```
// Definición del espacio de nombres (namespace) Proyecto_PreLab
```

```
// -----
```

```
// Agrupa lógicamente todas las clases relacionadas con este proyecto
```

```
// Beneficios:
```

```
// 1. Organización lógica del código
```

```
// 2. Evita colisiones de nombres con otros componentes
```

```
// 3. Facilita el mantenimiento y la modularidad
```

```
namespace Proyecto_PreLab
```

```
{
```

```
// Declaración de la clase principal LoginForm que hereda de Form
```

```
// -----
```

```
// partial: Indica que la clase está dividida entre varios archivos
```

```
// (normalmente el código principal y el código generado por el diseñador)
```

```
// public: La clase es accesible desde otros assemblies (ensamblados)
```

```
public partial class LoginForm : Form
```

```
{  
    // Constructor de la clase CRUDForm  
  
    // -----  
  
    // Se ejecuta automáticamente al crear una instancia con 'new CRUDForm()'  
  
    // No recibe parámetros (constructor por defecto)  
  
    public LoginForm()  
  
{  
  
    // InitializeComponent(): Método generado automáticamente por el diseñador de Windows Forms  
  
    // Responsabilidades:  
  
    // 1. Instanciar todos los controles definidos visualmente en el diseñador  
  
    // 2. Configurar sus propiedades iniciales (tamaño, posición, texto, etc.)  
  
    // 3. Asignar manejadores de eventos básicos  
  
    // DEBE ser la primera línea en el constructor porque:  
  
    // - Asegura que todos los controles estén creados antes de cualquier otra operación  
  
    // - Los controles deben existir antes de poder manipularlos  
  
    InitializeComponent();  
  
  
    // txtUsuario.Select() es una referencia al TextBox definido en el diseñador  
  
    // .Select() hace:  
  
    // 1. Establece el foco (cursor) en ese control  
  
    // 2. Si hay texto, lo selecciona todo (útil para reemplazo rápido)  
  
    // Beneficios:  
  
    // - El usuario puede empezar a escribir inmediatamente  
  
    // - Guía visual sobre dónde comenzar la interacción  
  
    // - Mejora la eficiencia en formularios de entrada de datos  
  
    txtUsuario.Select();
```

```
}
```

```
// Manejador de eventos para el Click del botón Login
```

```
// -----
```

```
// Estructura típica de event handler en .NET:
```

```
// - private: Generalmente los handlers son privados
```

```
// - void: No retornan valores
```

```
// - Parámetros:
```

```
// * sender: El objeto que generó el evento (el botón en este caso)
```

```
// * e: Información adicional sobre el evento (EventArgs base vacío)
```

```
private void btnLogin_Click(object sender, EventArgs e)
```

```
{
```

```
// Obtención y limpieza de credenciales
```

```
// -----
```

```
// txtUsuario.Text - Accede al texto actual del TextBox
```

```
// .Trim() - Elimina espacios en blanco al inicio/final
```

```
// Beneficios:
```

```
// - Evita problemas con espacios accidentales
```

```
// - Mejora consistencia en comparaciones
```

```
// - Previene "contraseña invisible" con espacios finales
```

```
string usuario = txtUsuario.Text.Trim();
```

```
string contraseña = txtContraseña.Text.Trim();
```

```
// Validación 1: Ambos campos vacíos
```

```
// -----
```

```
// string.IsNullOrEmpty() vs == "";
```

```
// - También maneja casos null (aunque improbable en TextBox)
// - Más legible y expresivo
// - Ligeramente más eficiente

if (string.IsNullOrEmpty(usuario) && string.IsNullOrEmpty(contraseña))
{
    // MessageBox.Show() - Diálogo modal al usuario
    // -----
    // Parámetros:
    // 1. Mensaje principal
    // 2. Título de la ventana
    // 3. MessageBoxButtons: Opciones (OK, YesNo, etc.)
    // 4. MessageBoxIcon: Ícono visual (Warning, Error, etc.)

    // Características:
    // - Bloquea la interacción con otros forms
    // - Devuelve DialogResult para manejar respuestas

    MessageBox.Show("Debe ingresar usuario y contraseña",
        "Campos vacíos",
        MessageBoxButtons.OK,
        MessageBoxIcon.Warning);

    // Establecer foco en el primer campo
    txtUsuario.Focus();

    // Salir temprano del método (early return pattern)
    return;
}
```

```
// Validación 2: Solo usuario vacío

if (string.IsNullOrEmpty(usuario))

{
    MessageBox.Show("Debe ingresar un usuario",
                    "Usuario requerido",
                    MessageBoxButtons.OK,
                    MessageBoxIcon.Warning);

    txtUsuario.Focus();

    return;
}

// Validación 3: Solo contraseña vacía

if (string.IsNullOrEmpty(contraseña))

{
    MessageBox.Show("Debe ingresar una contraseña",
                    "Contraseña requerida",
                    MessageBoxButtons.OK,
                    MessageBoxIcon.Warning);

    txtContraseña.Focus();

    return;
}

// Validación 4: Credenciales incorrectas
// -----
// Credenciales hardcodeadas (en una app real usaría almacenamiento seguro)
```

```
// Uso de OR (||) para:  
  
// 1. Evaluación de cortocircuito (si usuario falla, no evalúa contraseña)  
  
// 2. Cualquier credencial incorrecta invalida el login  
  
if (usuario != "DonBosco" || contraseña != "donbosco")  
{  
  
    MessageBox.Show("Usuario o contraseña incorrectos",  
        "Error de autenticación",  
        MessageBoxButtons.OK,  
        MessageBoxIcon.Error);  
  
  
    // Limpieza segura de la contraseña  
  
    txtContraseña.Clear();  
  
  
    // Selección completa para fácil reemplazo  
  
    txtUsuario.SelectAll();  
  
  
    // Regresar foco para nuevo intento  
  
    txtUsuario.Focus();  
  
    return;  
}  
  
  
// Autenticación exitosa - Abrir formulario principal  
// -----  
  
// Creación del CRUDForm (formulario principal)  
  
// 'new' realiza:  
  
// 1. Asignación de memoria
```

```
// 2. Llamada al constructor  
  
// 3. Inicialización de campos  
  
CRUDForm mainForm = new CRUDForm();  
  
  
// ShowDialog() vs Show():  
  
// - Modal: Bloquea interacción con el padre  
  
// - Retorna sólo al cerrarse  
  
// - Ideal para flujos donde se necesita respuesta  
  
mainForm.ShowDialog();  
  
  
// Limpieza post-login (seguridad y usabilidad)  
  
txtContraseña.Clear();  
  
txtUsuario.Clear();  
  
}  
  
  
// Manejador para el botón Cancelar  
  
private void btnCancelar_Click(object sender, EventArgs e)  
{  
  
    // Application.Exit() vs this.Close()  
  
    // -----  
  
    // Application.Exit():  
  
    // - Termina toda la aplicación  
  
    // - Dispara eventos de cierre en todos los forms  
  
    // - Libera recursos  
  
    // this.Close():  
  
    // - Sólo cierra este formulario
```

```
// - Si es el main form, termina la app (depende de configuración)

// En login forms es común usar Application.Exit()

Application.Exit();

}

// Manejador para el botón Créditos

private void btnCreditos_Click(object sender, EventArgs e)

{

    // Creación del formulario de créditos

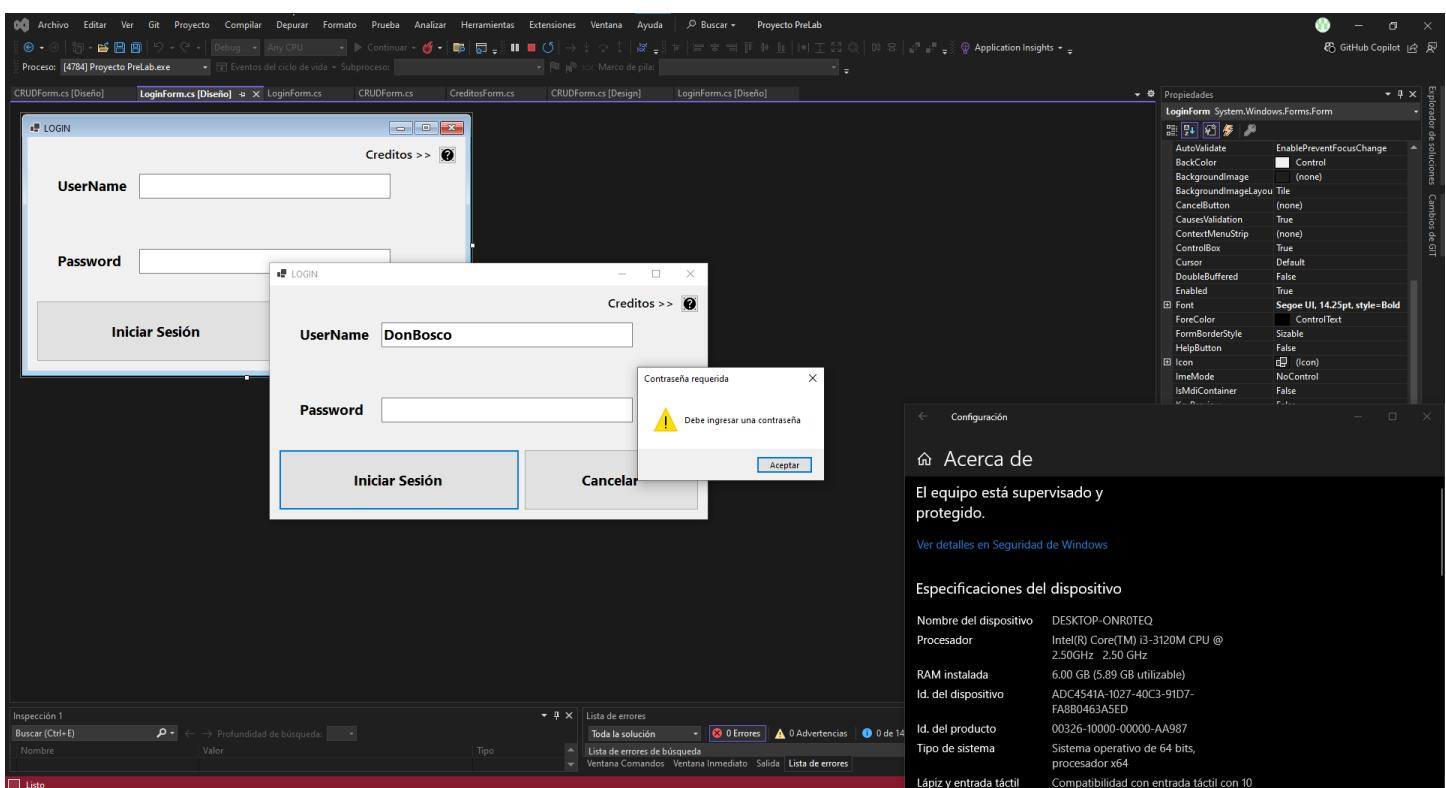
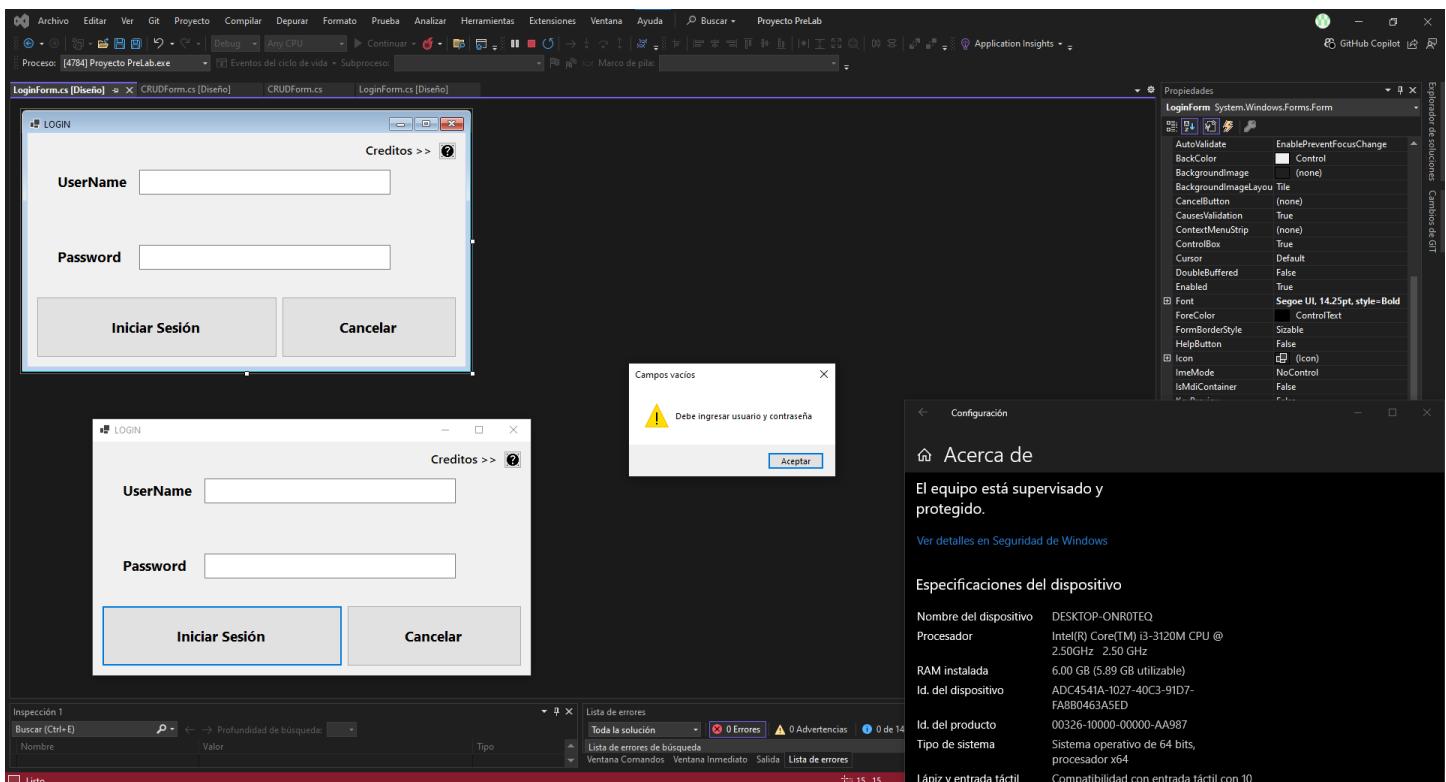
    CreditosForm credForm = new CreditosForm();

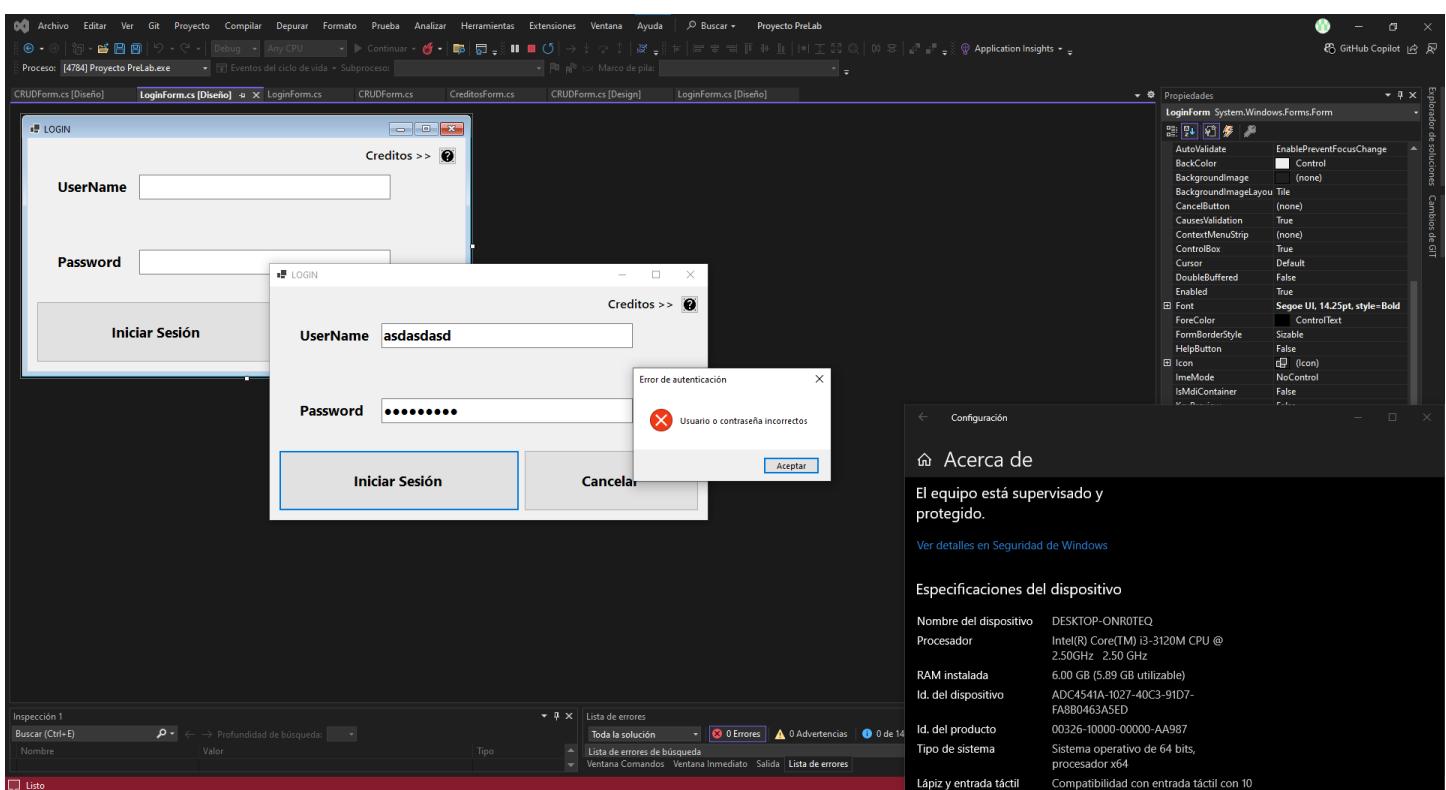
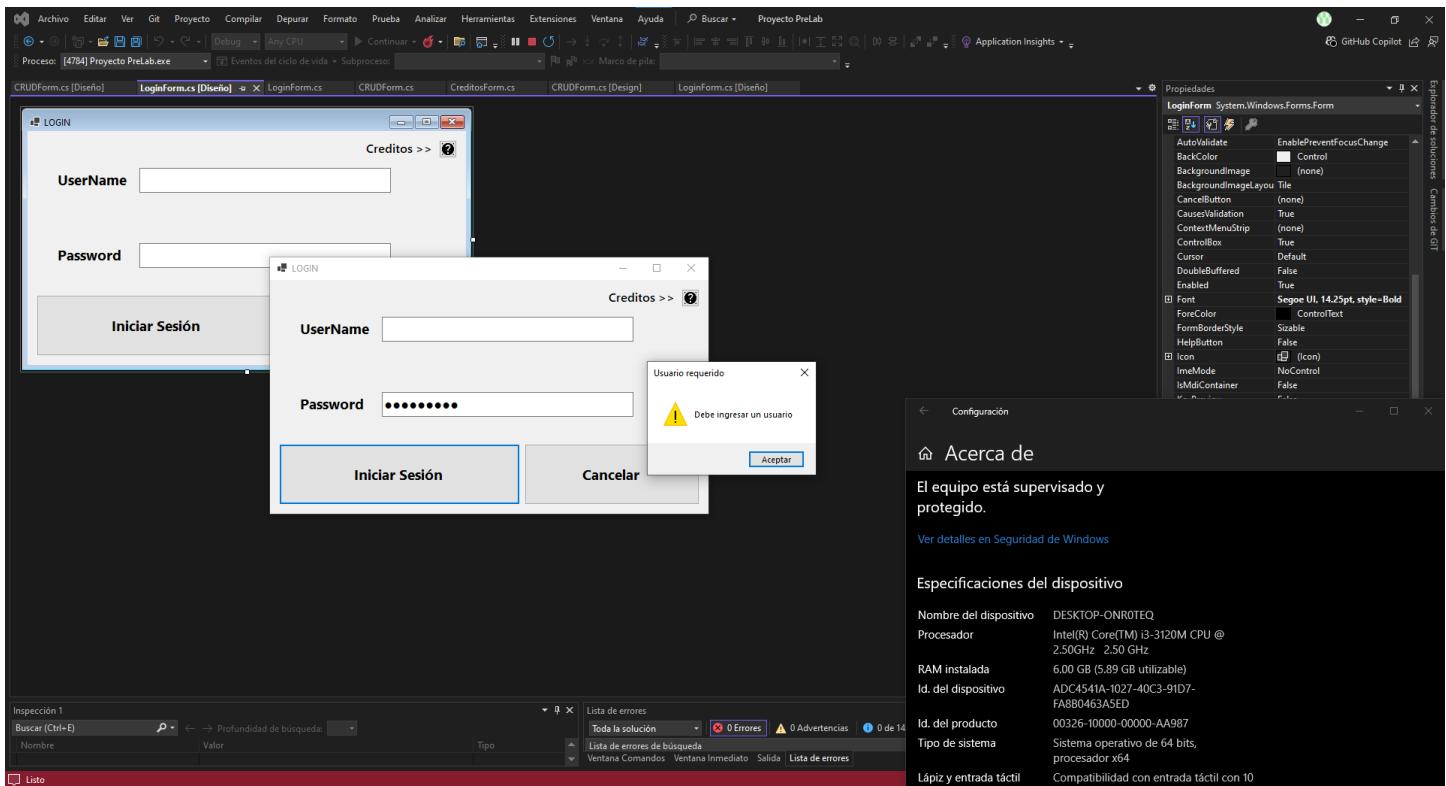
    // Mostrado modal para asegurar atención

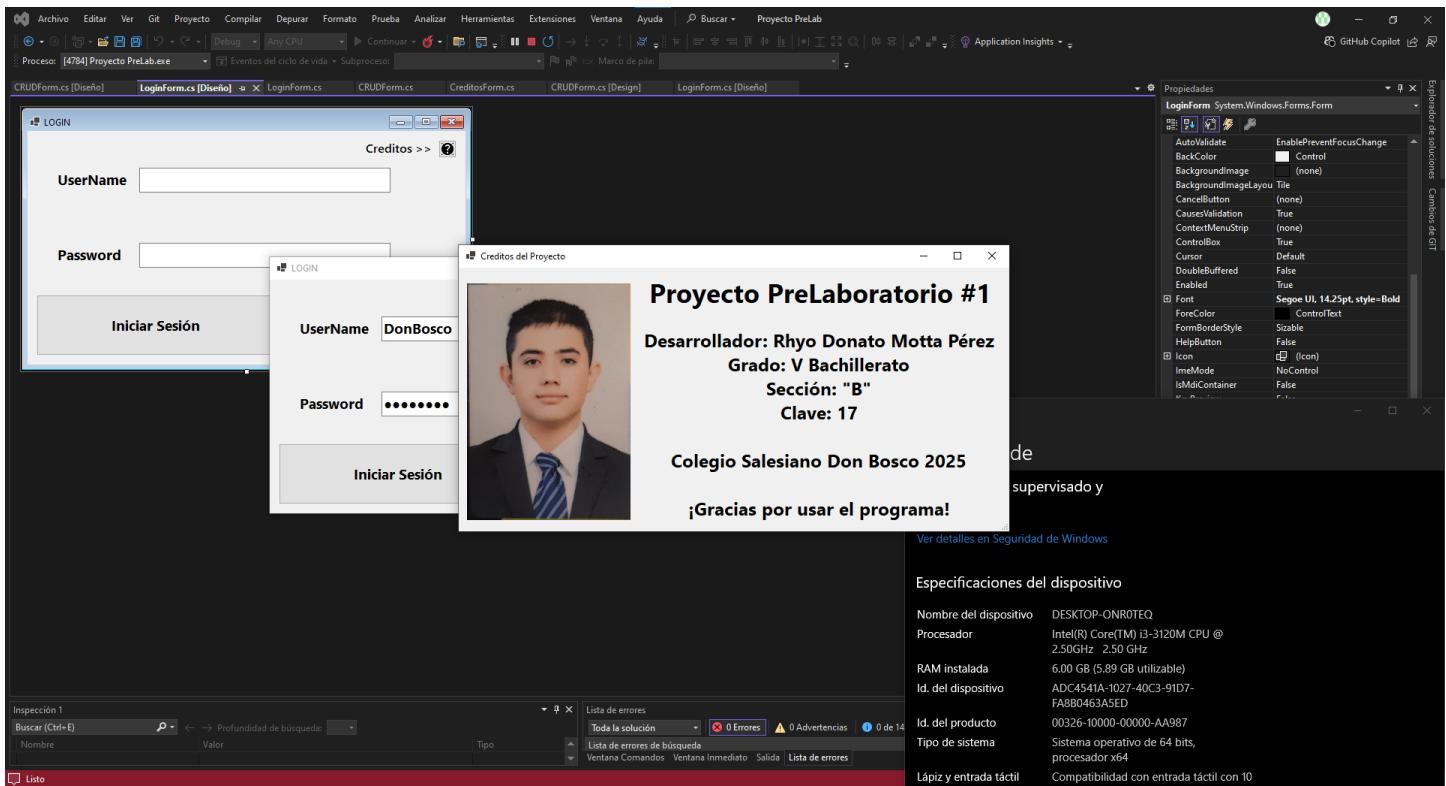
    credForm.ShowDialog();

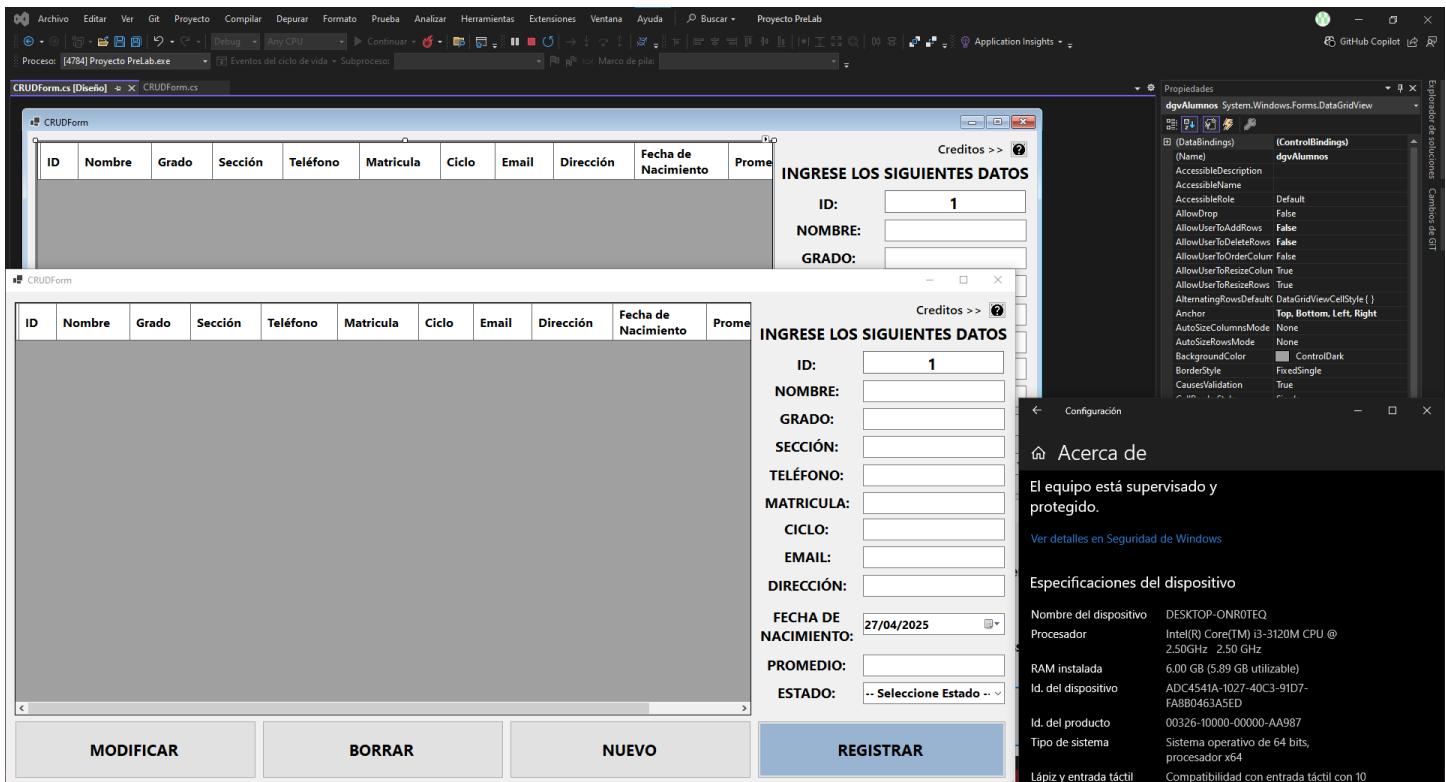
}

}
```









```
using System; // Proporciona clases fundamentales
```

```
using System.Collections.Generic; // Contiene colecciones genéricas como
```

```
using System.ComponentModel; // Proporciona funcionalidades para
```

```
using System.Data; // Clases para manejo de datos como
```

```
using System.Drawing; // Funcionalidades gráficas básicas
```

```
using System.Linq; // Soporte para LINQ (Language Integrated Query)
```

```
using System.Text; // Manipulación avanzada de cadenas
```

```
using System.Threading.Tasks; // Soporte para programación asíncrona
```

```
using System.Windows.Forms; // Clases para crear aplicaciones de Windows Forms
```

```
// Definición del espacio de nombres (namespace) Proyecto_PreLab
// -----
// Agrupa lógicamente todas las clases relacionadas con este proyecto
// Beneficios:
// 1. Organización lógica del código
// 2. Evita colisiones de nombres con otros componentes
// 3. Facilita el mantenimiento y la modularidad

namespace Proyecto_PreLab

{
    // Declaración de la clase principal CRUDForm que hereda de Form
    // -----
    // partial: Indica que la clase está dividida entre varios archivos
    // (normalmente el código principal y el código generado por el diseñador)
    // public: La clase es accesible desde otros assemblies (ensamblados)

    public partial class CRUDForm : Form
    {
        // Declaración de variables miembro de la clase
        // -----
        // code: Contador para generar IDs automáticos secuenciales
        // private: Solo accesible dentro de esta clase (encapsulación)
        // int: Tipo de dato entero de 32 bits con signo (-2,147,483,648 a 2,147,483,647)
        // Se inicia en 1 porque:
        // - Es más natural para IDs (0 suele significar "no asignado" o valor por defecto)
        // - Mejor legibilidad para el usuario final

        private int code = 1;

        // seleccionFila: Almacena el índice de la fila seleccionada en el DataGridView
        // static: La variable es compartida por todas las instancias de CRUDForm
```

```
// Esto podría ser discutible ya que normalmente cada formulario debería manejar su propia selección
// -1: Valor convencional que indica "ninguna selección"
// Se usa para:
// - Saber qué fila se está editando
// - Mantener el estado entre diferentes operaciones CRUD
private static int seleccionFila = -1;

// Constructor de la clase CRUDForm
// -----
// Se ejecuta automáticamente al crear una instancia con 'new CRUDForm()'
// No recibe parámetros (constructor por defecto)
public CRUDForm()
{
    // InitializeComponent(): Método generado automáticamente por el diseñador de Windows Forms
    // Responsabilidades:
    // 1. Instanciar todos los controles definidos visualmente en el diseñador
    // 2. Configurar sus propiedades iniciales (tamaño, posición, texto, etc.)
    // 3. Asignar manejadores de eventos básicos
    // DEBE ser la primera línea en el constructor porque:
    // - Asegura que todos los controles estén creados antes de cualquier otra operación
    // - Los controles deben existir antes de poder manipularlos
    InitializeComponent();
}

// Manejador del evento Load del formulario
// -----
// Se ejecuta cuando el formulario se carga en memoria, después del constructor
// pero antes de mostrarse al usuario
// sender: Referencia al objeto que generó el evento (en este caso, el formulario mismo)
// e: Instancia de EventArgs que contiene datos adicionales del evento (vacío en este caso)
```

```
private void CRUDForm_Load(object sender, EventArgs e)
{
    // Configurar el texto del Label lblCodigo con el valor actual del contador
    // .Text: Propiedad que contiene el texto mostrado en un control Label
    // .ToString(): Convierte el entero 'code' a su representación string
    // Esto muestra al usuario el ID que se asignará al próximo registro
    lblCodigo.Text = code.ToString();

    // Establecer la selección inicial del ComboBox cmbEstado al primer ítem
    // .SelectedIndex: Índice basado en 0 del ítem seleccionado (-1 si no hay selección)
    // Se establece en 0 para:
    // - Tener un valor por defecto
    // - Evitar que el ComboBox aparezca vacío al inicio
    cmbEstado.SelectedIndex = 0;
}
```

```
// Manejador del evento Click del botón Registrar (Create en CRUD)
// -----
private void btnReg_Click(object sender, EventArgs e)
{
    // VALIDACIÓN INICIAL:
    // ValidarCampos(): Método que verifica que todos los campos obligatorios estén completos
    // Devuelve true si todos los campos son válidos, false si hay errores
    // El operador ! niega el resultado (si es false, sale del método con return)
    // Esto previene que se registren datos incompletos o inválidos
    if (!ValidarCampos())
        return;

    // AGREGAR NUEVA FILA AL DATAGRIDVIEW:
    // dgvAlumnos: Control DataGridView que muestra los registros en formato tabular
```

```
// .Rows: Colección que contiene todas las filas del DataGridView  
// .Add(): Método para agregar una nueva fila, acepta valores para cada columna  
// Se pasan los valores en el orden de las columnas del DataGridView  
  
dgvAlumnos.Rows.Add(  
  
    code.ToString(),           // Columna 0: ID (convertido a string)  
    txtNombre.Text.Trim(),     // Columna 1: Nombre (sin espacios extras)  
    txtGrado.Text.Trim(),      // Columna 2: Grado  
    txtSeccion.Text.Trim(),    // Columna 3: Sección  
    txtTelefono.Text.Trim(),   // Columna 4: Teléfono  
    txtMatricula.Text.Trim(),  // Columna 5: Matrícula  
    txtCiclo.Text.Trim(),      // Columna 6: Ciclo Académico  
    txtEmail.Text.Trim(),      // Columna 7: Email  
    txtDireccion.Text.Trim(),  // Columna 8: Dirección  
   .dtpFechaNacimiento.Value // Columna 9: Fecha  
        .ToString(),          // .ToString() muestra formato regional  
    txtPromedio.Text.Trim(),    // Columna 10: Promedio  
    cmbEstado.Text            // Columna 11: Estado del ComboBox  
);
```

```
// MOSTRAR MENSAJE DE CONFIRMACIÓN:  
  
// MessageBox.Show(): Muestra un cuadro de diálogo modal al usuario  
// Parámetros:  
// 1. Mensaje principal ("Alumno registrado correctamente.")  
// 2. Título de la ventana ("Éxito")  
// 3. Botones a mostrar (OK)  
// 4. Icono (Information = ícono azul con 'i')  
  
// Esto proporciona feedback inmediato al usuario sobre la operación exitosa  
  
MessageBox.Show("Alumno registrado correctamente.",  
    "Éxito",  
    MessageBoxButtons.OK,
```

```

    MessageBoxIcon.Information);

// ACTUALIZAR CONTADOR DE ID:
// ++ operador de incremento (equivale a code = code + 1)
// Prepara el siguiente ID para un nuevo registro
code++;

// ACTUALIZAR LABEL CON NUEVO ID:
// Muestra al usuario el ID que se asignará al próximo registro
lblCodigo.Text = code.ToString();

// LIMPIAR CAMPOS PARA NUEVO REGISTRO:
// Prepara el formulario para una nueva entrada de datos
LimpiarCampos();

}

// Manejador del evento Click del botón Modificar (Update en CRUD)
// -----
private void btnMod_Click(object sender, EventArgs e)
{
    // VERIFICAR SELECCIÓN VÁLIDA:
    // seleccionFila < 0 significa que no hay fila seleccionada
    // Esto previene errores al intentar modificar sin selección
    if (seleccionFila < 0)
    {
        MessageBox.Show("Por favor, seleccione un alumno para modificar.",
            "Error",
            MessageBoxButtons.OK,
            MessageBoxIcon.Warning);
        return; // Salir del método si no hay selección
    }
}

```

```
}
```

```
// VALIDAR CAMPOS:
```

```
// Reutiliza la misma validación que para registro nuevo
```

```
if (!ValidarCampos())
```

```
    return;
```

```
// VERIFICAR ÍNDICE VÁLIDO:
```

```
// Comprueba que el índice esté dentro del rango válido
```

```
// Esto es defensivo contra posibles errores de índice
```

```
if (seleccionFila >= 0 && seleccionFila < dgvAlumnos.Rows.Count)
```

```
{
```

```
// ACTUALIZAR FILA SELECCIONADA:
```

```
// Acceder a cada celda por su índice y actualizar su valor
```

```
// Se actualizan todos los campos, incluso si no cambiaron
```

```
dgvAlumnos.Rows[seleccionFila].Cells[0].Value = lblCodigo.Text; // ID
```

```
dgvAlumnos.Rows[seleccionFila].Cells[1].Value = txtNombre.Text; // Nombre
```

```
dgvAlumnos.Rows[seleccionFila].Cells[2].Value = txtGrado.Text; // Grado
```

```
dgvAlumnos.Rows[seleccionFila].Cells[3].Value = txtSeccion.Text; // Sección
```

```
dgvAlumnos.Rows[seleccionFila].Cells[4].Value = txtTelefono.Text; // Teléfono
```

```
dgvAlumnos.Rows[seleccionFila].Cells[5].Value = txtMatricula.Text; // Matrícula
```

```
dgvAlumnos.Rows[seleccionFila].Cells[6].Value = txtCiclo.Text; // Ciclo
```

```
dgvAlumnos.Rows[seleccionFila].Cells[7].Value = txtEmail.Text; // Email
```

```
dgvAlumnos.Rows[seleccionFila].Cells[8].Value = txtDireccion.Text; // Dirección
```

```
dgvAlumnos.Rows[seleccionFila].Cells[9].Value = dtpFechaNacimiento.Value.ToString("yyyy-MM-dd"); // Fecha
```

```
dgvAlumnos.Rows[seleccionFila].Cells[10].Value = txtPromedio.Text; // Promedio
```

```
dgvAlumnos.Rows[seleccionFila].Cells[11].Value = cmbEstado.Text; // Estado
```

```
// MOSTRAR CONFIRMACIÓN:
```

```

MessageBox.Show("Datos del alumno actualizados correctamente.",
    "Éxito",
    MessageBoxButtons.OK,
    MessageBoxIcon.Information);

// LIMPIAR Y RESETEAR:
// Prepara el formulario para una nueva operación
LimpiarCampos();
seleccionFila = -1; // Resetea la selección
}

}

// Manejador del evento Click del botón Eliminar (Delete en CRUD)
// -----
private void btnDel_Click(object sender, EventArgs e)
{
    // VERIFICAR SELECCIÓN VÁLIDA:
    // Comprueba que haya una fila seleccionada y que el índice sea válido
    if (seleccionFila >= 0 && seleccionFila < dgvAlumnos.Rows.Count)
    {
        // PEDIR CONFIRMACIÓN:
        // MessageBox con botones Sí/No y icono de pregunta
        // Es importante confirmar antes de eliminar datos
        DialogResult confirmacion = MessageBox.Show(
            "¿Está seguro que desea eliminar este alumno?\nEsta acción no se puede deshacer.",
            "Confirmar eliminación",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Question);

        // SI CONFIRMA:

```

```
if (confirmacion == DialogResult.Yes)
{
    // ELIMINAR FILA:
    // RemoveAt elimina la fila en el índice especificado
    dgvAlumnos.Rows.RemoveAt(seleccionFila);

    // MOSTRAR CONFIRMACIÓN:
    MessageBox.Show("Alumno eliminado correctamente.",
        "Éxito",
        MessageBoxButtons.OK,
        MessageBoxIcon.Information);

    // LIMPIAR Y RESETEAR:
    seleccionFila = -1; // Resetea la selección
    LimpiarCampos(); // Prepara para nueva operación
}

else
{
    // MOSTRAR ERROR SI NO HAY SELECCIÓN:
    MessageBox.Show("Por favor, seleccione un alumno para eliminar.",
        "Error",
        MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
}

// Manejador del evento Click del botón Nuevo
// -----
private void btnNew_Click(object sender, EventArgs e)
```

```

{
    // LIMPIAR CAMPOS:
    // Restablece todos los controles a sus valores iniciales
    LimpiarCampos();

    // RESETEAR SELECCIÓN:
    // Indica que no hay ninguna fila seleccionada
    seleccionFila = -1;

}

// Manejador del evento CellClick del DataGridView
// -----
private void dgvAlumnos_CellClick(object sender, DataGridViewCellEventArgs e)
{
    // VERIFICAR QUE NO SEA CLIC EN ENCABEZADO:
    // e.RowIndex es -1 si se hizo clic en el encabezado de columna
    if (e.RowIndex >= 0)

    {
        // GUARDAR SELECCIÓN:
        // Almacena el índice de la fila clickeada para operaciones posteriores
        seleccionFila = e.RowIndex;

        // OBTENER REFERENCIA A LA FILA:
        // Accede a la fila específica que fue clickeada
        DataGridViewRow fila = dgvAlumnos.Rows[e.RowIndex];

        // CARGAR DATOS A CONTROLES:
        // Uso de operador ?. para manejo seguro de nulos (null-conditional)
        // Uso de ?? para proporcionar valor por defecto si es nulo (null-coalescing)
        lblCodigo.Text = fila.Cells[0].Value?.ToString() ?? "";
    }
}

```

```

txtNombre.Text = fila.Cells[1].Value?.ToString() ?? "";
txtGrado.Text = fila.Cells[2].Value?.ToString() ?? "";
txtSeccion.Text = fila.Cells[3].Value?.ToString() ?? "";
txtTelefono.Text = fila.Cells[4].Value?.ToString() ?? "";
txtMatricula.Text = fila.Cells[5].Value?.ToString() ?? "";
txtCiclo.Text = fila.Cells[6].Value?.ToString() ?? "";
txtEmail.Text = fila.Cells[7].Value?.ToString() ?? "";
txtDireccion.Text = fila.Cells[8].Value?.ToString() ?? "";

// MANEJO ESPECIAL PARA FECHA:
// TryParse intenta convertir el string a DateTime
// Si falla, asigna la fecha actual por defecto
if (fila.Cells[9].Value != null &&
    DateTime.TryParse(fila.Cells[9].Value.ToString(), out DateTime fecha))
{
    dtpFechaNacimiento.Value = fecha;
}
else
{
    dtpFechaNacimiento.Value = DateTime.Today;
}

// CARGAR RESTO DE CAMPOS:
txtPromedio.Text = fila.Cells[10].Value?.ToString() ?? "";
cmbEstado.Text = fila.Cells[11].Value?.ToString() ?? "";
}

// Método para validar campos del formulario
// -----

```

```
private bool ValidarCampos()
{
    // LISTA PARA CAMPOS VACÍOS/INVÁLIDOS:
    // List<string> es una colección dinámica que almacena los nombres de los campos inválidos
    List<string> camposVacios = new List<string>();

    // VERIFICAR CADA CAMPO:
    // string.IsNullOrWhiteSpace verifica: null, vacío o solo espacios
    // Agrega el nombre del campo a la lista si está vacío
    if (string.IsNullOrWhiteSpace(txtNombre.Text))
        camposVacios.Add("Nombre");
    if (string.IsNullOrWhiteSpace(txtGrado.Text))
        camposVacios.Add("Grado");
    if (string.IsNullOrWhiteSpace(txtSeccion.Text))
        camposVacios.Add("Sección");
    if (string.IsNullOrWhiteSpace(txtTelefono.Text))
        camposVacios.Add("Teléfono");
    if (string.IsNullOrWhiteSpace(txtMatricula.Text))
        camposVacios.Add("Matrícula");
    if (string.IsNullOrWhiteSpace(txtCiclo.Text))
        camposVacios.Add("Ciclo Académico");
    if (string.IsNullOrWhiteSpace(txtEmail.Text))
        camposVacios.Add("Email");
    if (string.IsNullOrWhiteSpace(txtDireccion.Text))
        camposVacios.Add("Dirección");
    if (string.IsNullOrWhiteSpace(txtPromedio.Text))
        camposVacios.Add("Promedio");

    // VALIDAR COMBOBOX:
    // SelectedIndex <= 0 significa no seleccionado o selección inicial
```

```

if (cmbEstado.SelectedIndex <= 0)
    camposVacios.Add("Estado");

// CASO 1: TODOS LOS CAMPOS VACÍOS:
// Si todos los campos están vacíos, muestra un mensaje general
if (camposVacios.Count == 10)
{
    MessageBox.Show("Debe completar todos los campos del formulario.",
        "Formulario incompleto",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);

txtNombre.Focus(); // Enfocar primer campo para facilitar la entrada
return false;
}

// CASO 2: ALGUNOS CAMPOS VACÍOS:
// Si solo algunos campos están vacíos, muestra un mensaje detallado
if (camposVacios.Count > 0)
{
    // CONSTRUIR MENSAJE DETALLADO:
    // string.Join concatena los elementos de la lista con separadores
    string mensaje = "Los siguientes campos son obligatorios:\n\n" +
        string.Join("\n", camposVacios);

    MessageBox.Show(mensaje,
        "Campos incompletos",
        MessageBoxButtons.OK,
        MessageBoxIcon.Warning);
}

```

```
// ENFOCAR PRIMER CAMPO VACÍO:  
// Establece el foco en el primer campo inválido encontrado  
// Esto mejora la experiencia de usuario al corregir los errores  
if (camposVacios.Contains("Nombre"))  
    txtNombre.Focus();  
else if (camposVacios.Contains("Grado"))  
    txtGrado.Focus();  
else if (camposVacios.Contains("Sección"))  
    txtSeccion.Focus();  
else if (camposVacios.Contains("Teléfono"))  
    txtTelefono.Focus();  
else if (camposVacios.Contains("Matrícula"))  
    txtMatricula.Focus();  
else if (camposVacios.Contains("Ciclo Académico"))  
    txtCiclo.Focus();  
else if (camposVacios.Contains("Email"))  
    txtEmail.Focus();  
else if (camposVacios.Contains("Dirección"))  
    txtDireccion.Focus();  
else if (camposVacios.Contains("Promedio"))  
    txtPromedio.Focus();  
else if (camposVacios.Contains("Estado"))  
    cmbEstado.Focus();  
  
return false;  
}
```

```
// VALIDACIÓN ADICIONAL PARA PROMEDIO (NUMÉRICO):  
// TryParse intenta convertir el texto a decimal  
// Si falla, muestra un mensaje de error
```

```
if (!decimal.TryParse(txtPromedio.Text, out decimal promedio))
{
    MessageBox.Show("El campo 'Promedio' debe contener un valor numérico válido.\n" +
        "Ejemplo: 15.5",
        "Error en promedio",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);

    txtPromedio.Focus();
    txtPromedio.SelectAll(); // Selecciona todo el texto para facilitar la corrección
    return false;
}

// VALIDACIÓN RANGO PROMEDIO (0-100):
// Verifica que el promedio esté dentro del rango aceptable
if (promedio < 0 || promedio > 100)
{
    MessageBox.Show("El promedio debe ser positivo y menor o igual 100.",
        "Error en promedio",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);

    txtPromedio.Focus();
    txtPromedio.SelectAll();
    return false;
}

return true; // Todas las validaciones pasaron
}
```

```
// Método para limpiar todos los controles del formulario
// -----
private void LimpiarCampos()
{
    // LIMPIAR TEXTBOXES:
    // .Clear() elimina todo el texto del control
    txtNombre.Clear();
    txtGrado.Clear();
    txtSeccion.Clear();
    txtTelefono.Clear();
    txtMatricula.Clear();
    txtCiclo.Clear();
    txtEmail.Clear();
    txtDireccion.Clear();
    txtPromedio.Clear();

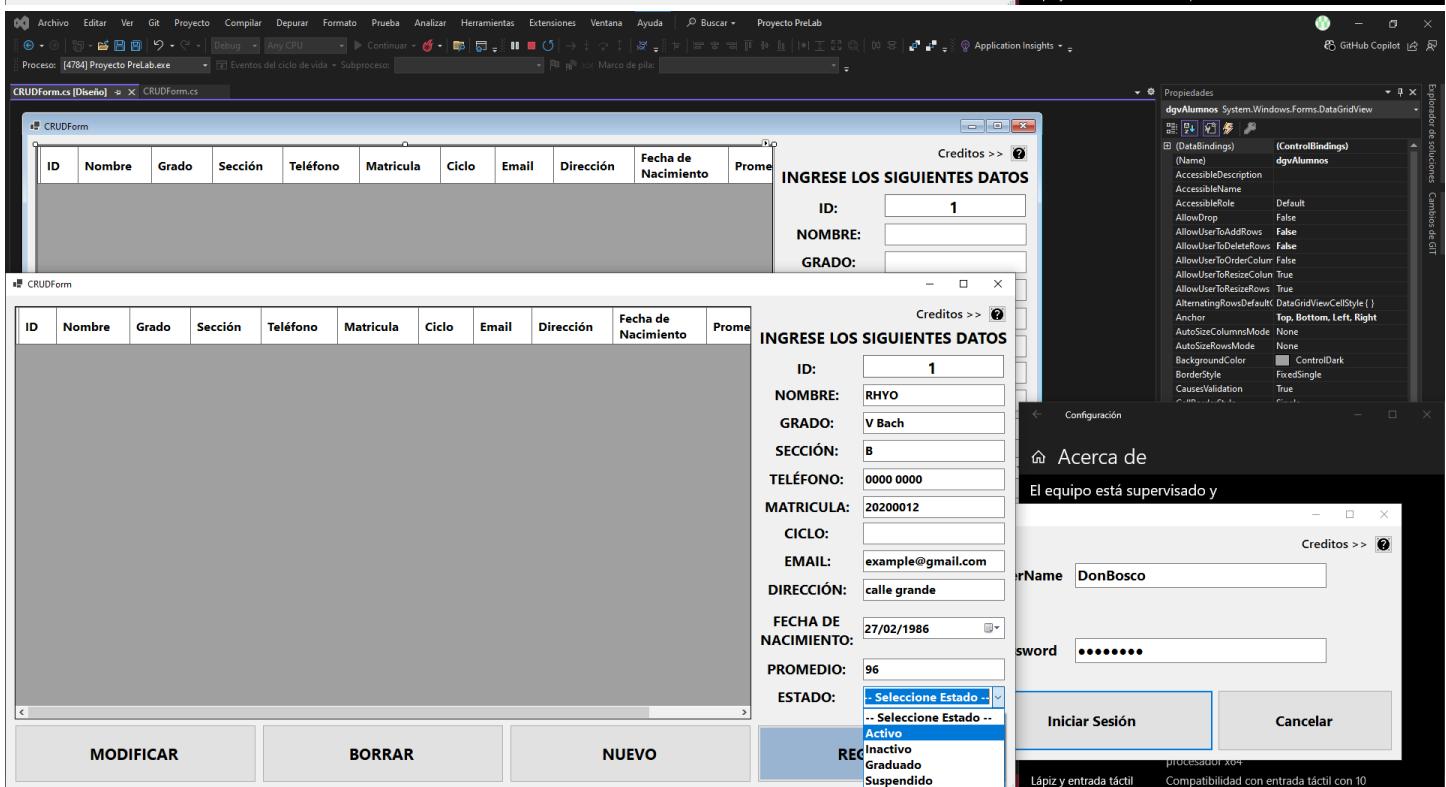
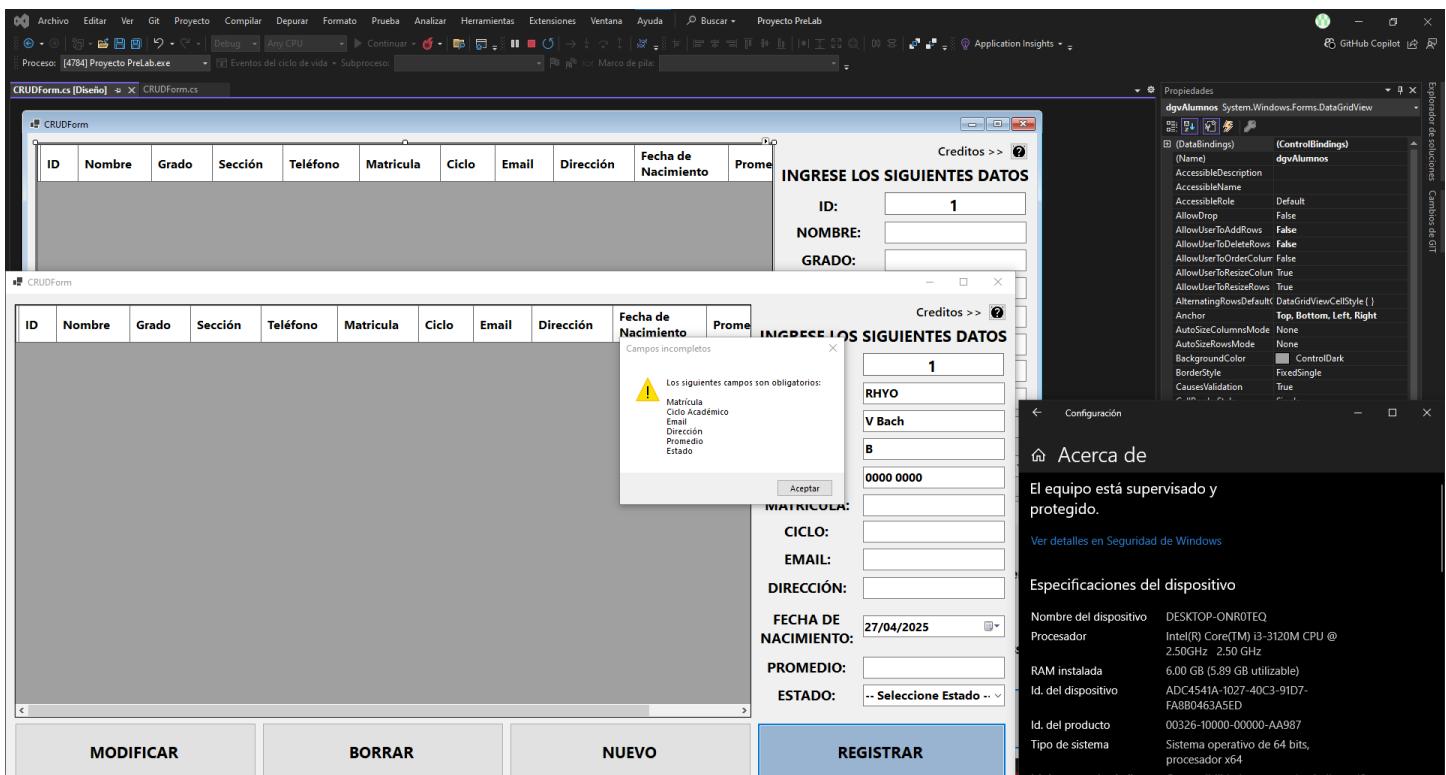
    // RESTABLECER DATETIMEPICKER:
    // Establece la fecha al día actual
    dtpFechaNacimiento.Value = DateTime.Today;

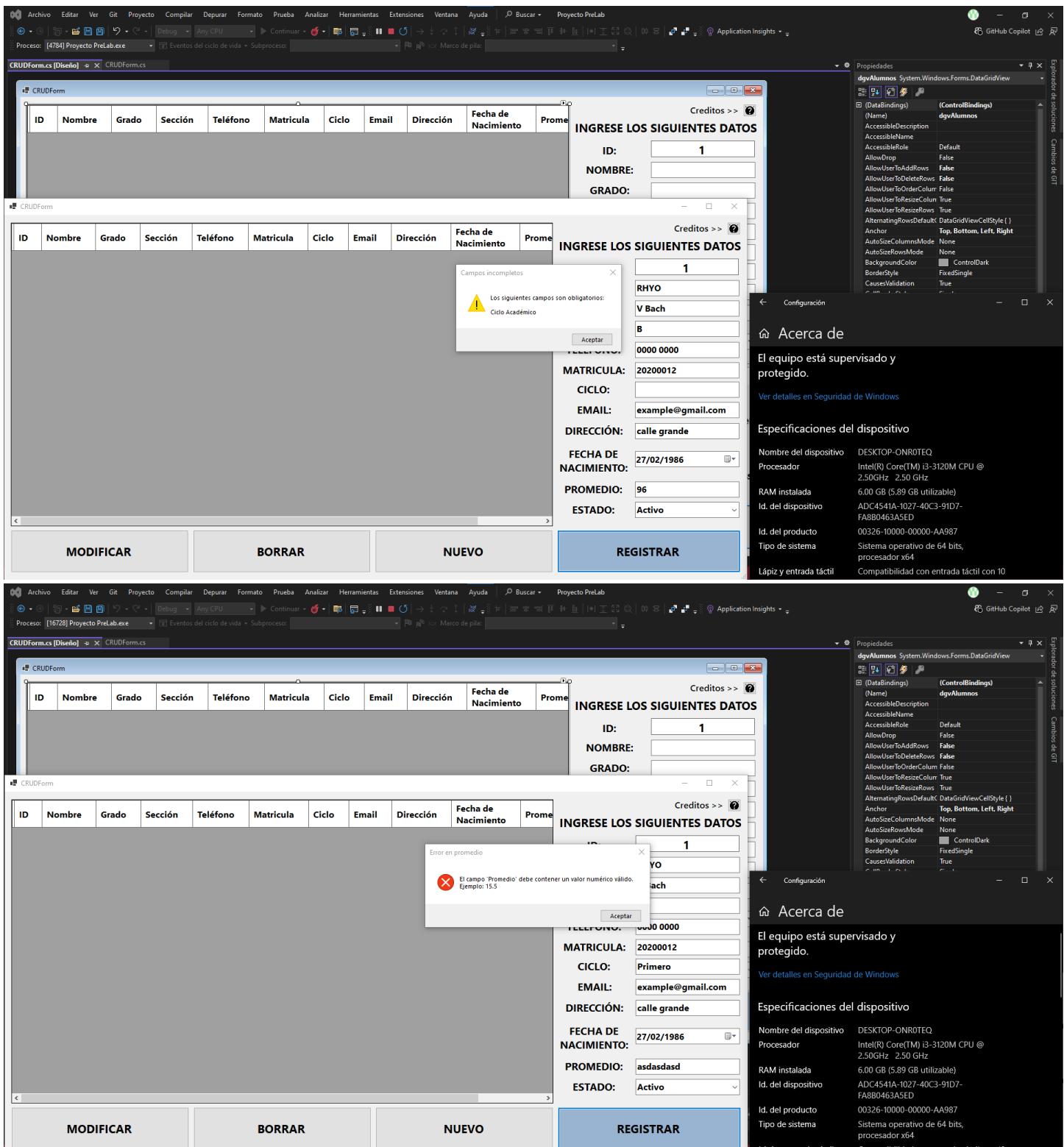
    // RESTABLECER COMBOBOX:
    // Vuelve a la selección inicial (índice 0)
    cmbEstado.SelectedIndex = 0;

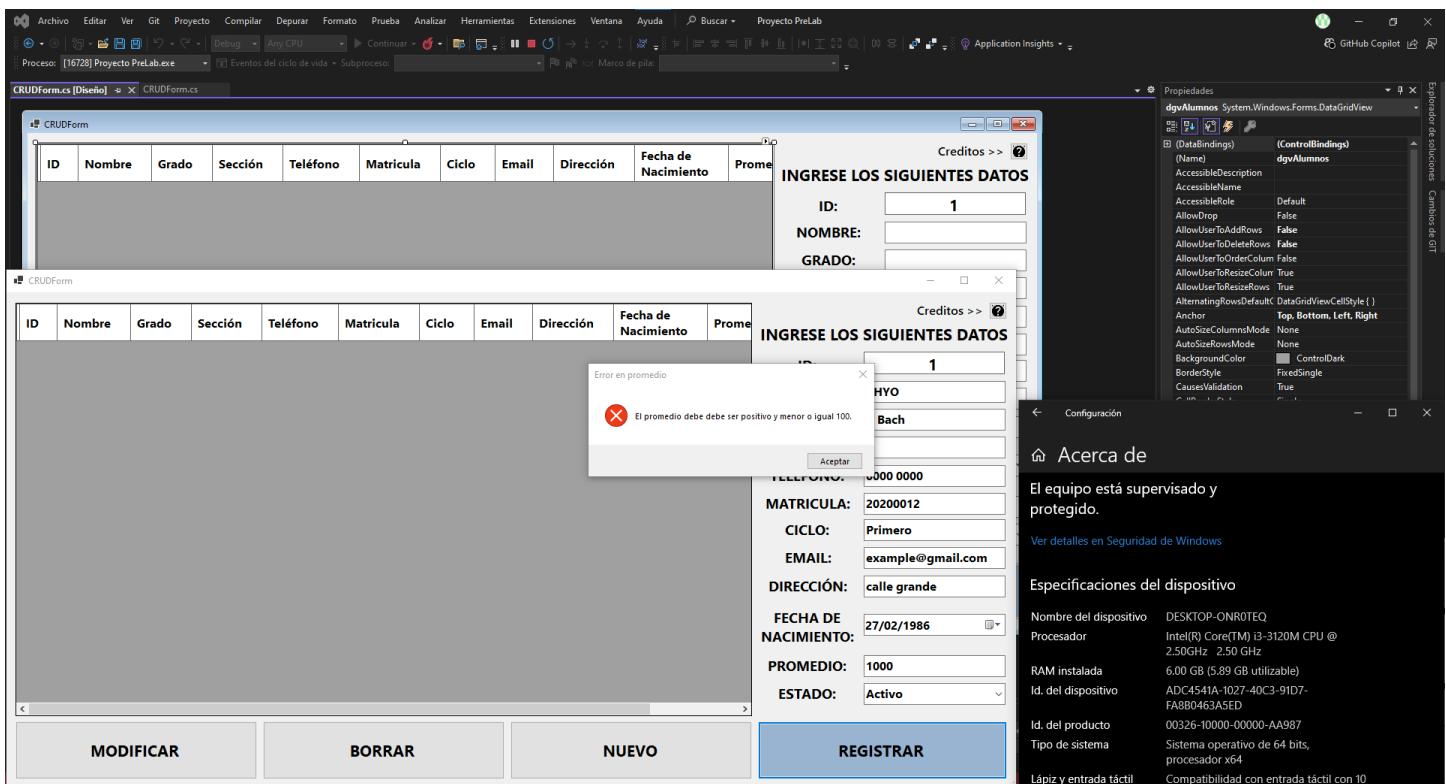
    // ACTUALIZAR LABEL DE CÓDIGO:
    // Muestra el próximo ID disponible
    lblCodigo.Text = code.ToString();

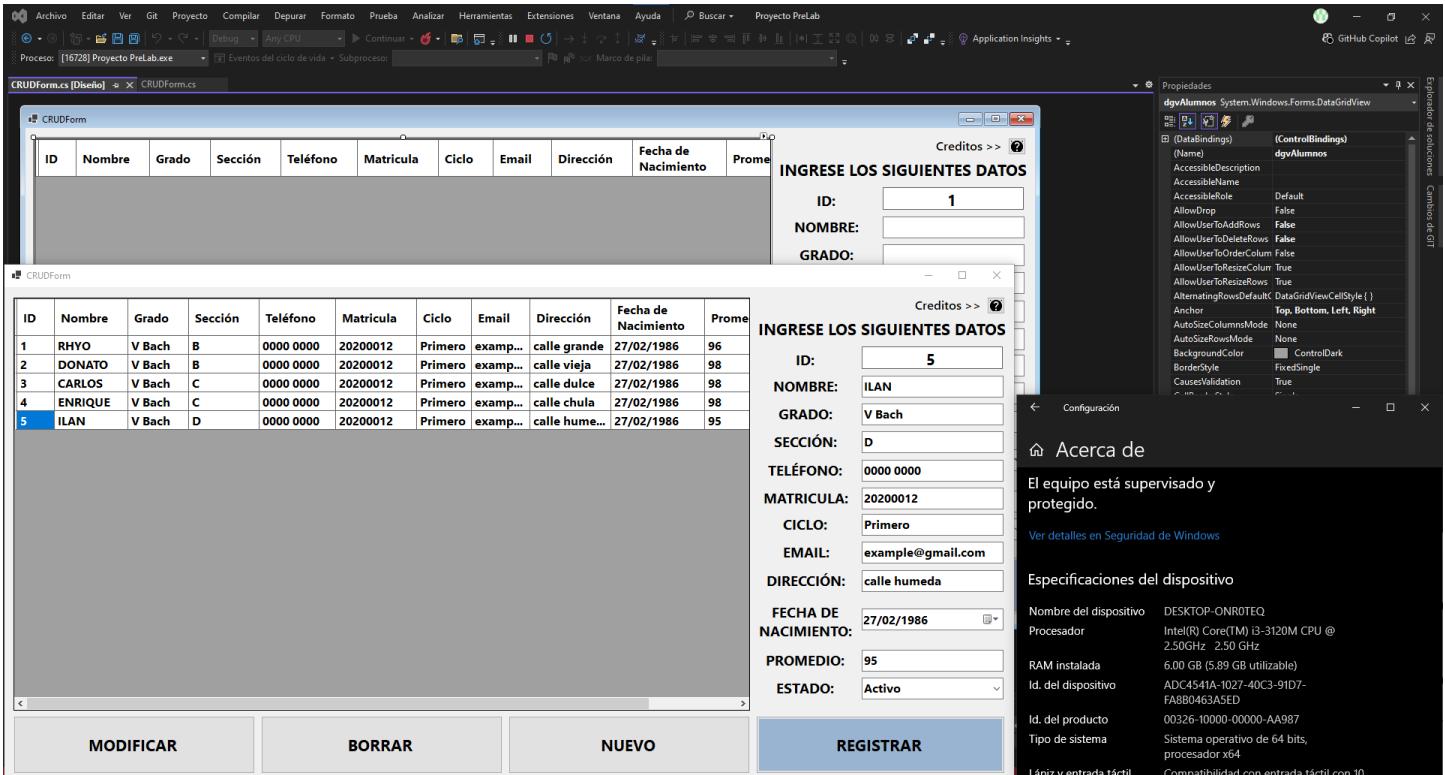
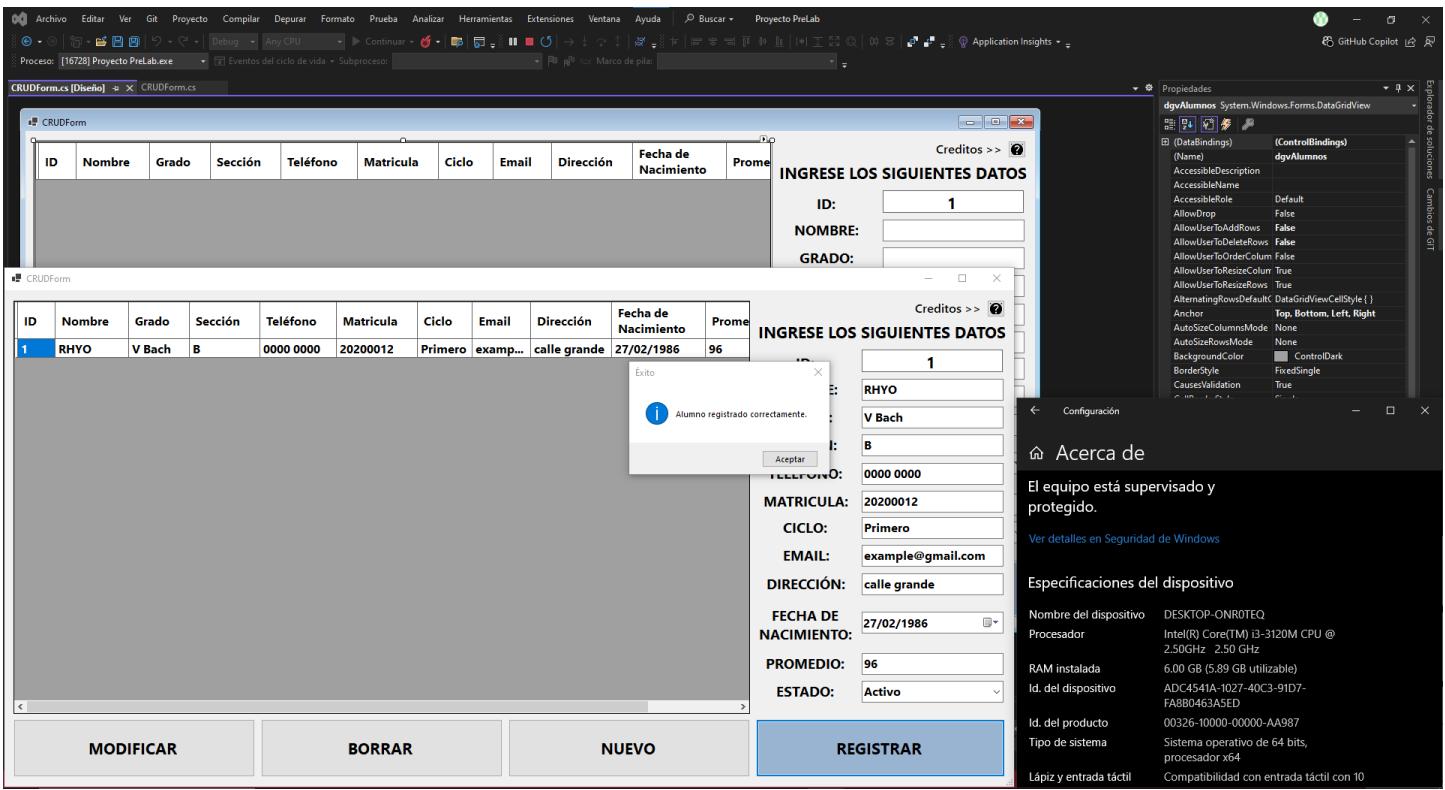
    // ENFOCAR CAMPO PRINCIPAL:
    // Establece el foco en el campo Nombre para facilitar la nueva entrada
```

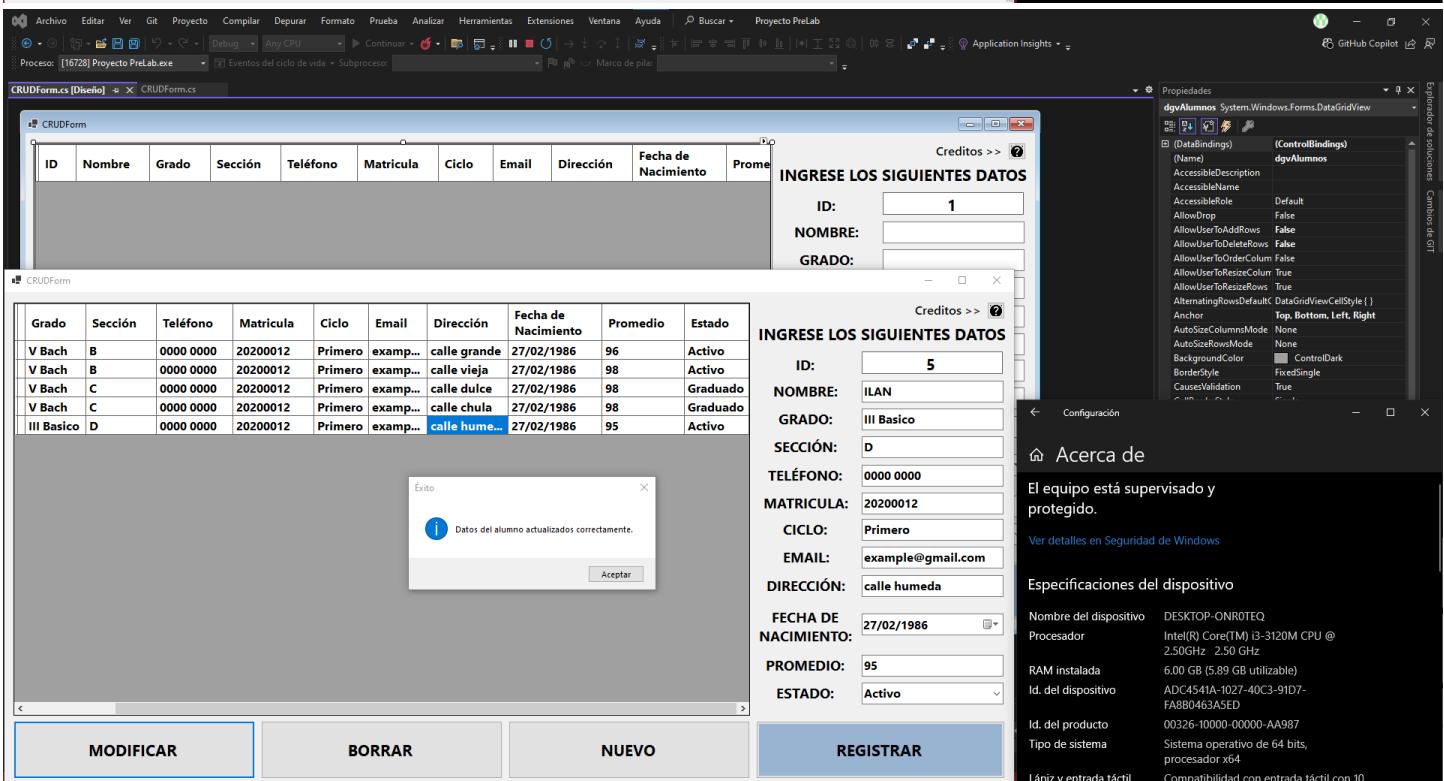
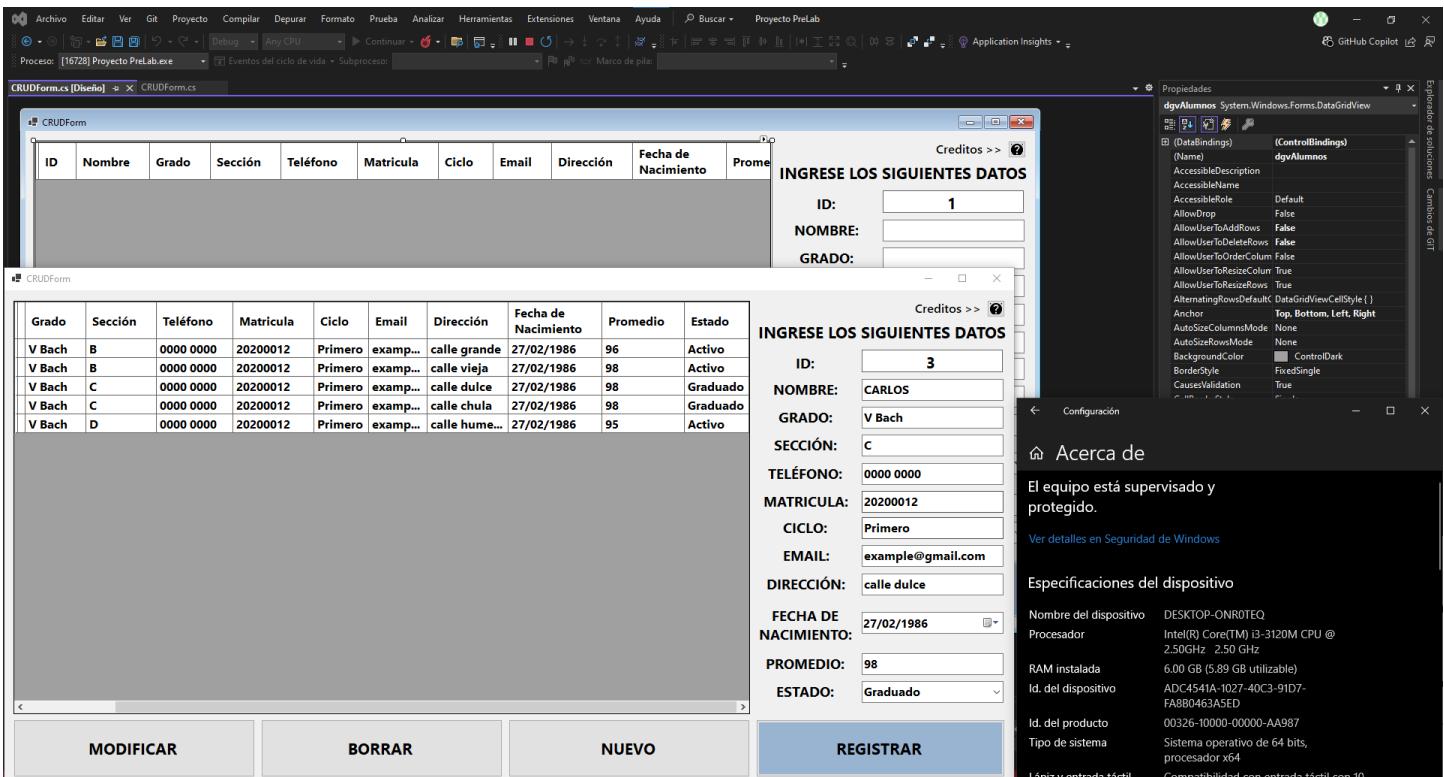
```
txtNombre.Focus();  
}  
  
// Manejador del evento Click del botón Créditos  
// -----  
private void btnCreditos_Click(object sender, EventArgs e)  
{  
    // CREAR INSTANCIA DEL FORMULARIO:  
    // CreditosForm es otro formulario que muestra información sobre los autores  
    CreditosForm formCreditos = new CreditosForm();  
  
    // MOSTRAR COMO DIÁLOGO MODAL:  
    // ShowDialog() muestra el formulario de forma modal:  
    // - Bloquea interacción con formulario padre hasta cerrarse  
    // - El código aquí se pausa hasta que se cierre el diálogo  
    // Esto asegura que el usuario atienda la información de créditos  
    formCreditos.ShowDialog();  
}  
}
```











Proyecto Prelab

Proceso: [16728] Proyecto Prelab.exe

Eventos del ciclo de vida > Subproceso:

Marco de pila: Application Insights GitHub Copilot Explorador de soluciones Cambiar de GIT

**CRUDForm.cs (Diseño)** + CRUDForm.cs

**CRUDForm**

ID	Nombre	Grado	Sección	Teléfono	Matrícula	Ciclo	Email	Dirección	Fecha de Nacimiento	Promedio
1	RHYO	V Bach	B	0000 0000	20200012	Primero	examp...	calle grande	27/02/1986	96
2	DONATO	V Bach	B	0000 0000	20200012	Primero	examp...	calle vieja	27/02/1986	95
3	CARLOS	V Bach	C	0000 0000	20200012	Primero	examp...	calle dulce	27/02/1986	95
4	ENRIQUE	V Bach	C	0000 0000	20200012	Primero	examp...	calle chula	27/02/1986	95
5	ILAN	III Basico	D	0000 0000	20200012	Primero	examp...	calle hume...	27/02/1986	95

**INGRESE LOS SIGUIENTES DATOS**

Creditos >> ?

ID:  NOMBRE:   
GRADO:

Confirmar eliminación

¿Está seguro que desea eliminar este alumno?  
Esta acción no se puede deshacer.

Sí No

**TELEFONO:**   
**MATRICULA:**   
**CICLO:**   
**EMAIL:**   
**DIRECCIÓN:**   
**FECHA DE NACIMIENTO:**   
**PROMEDIO:**   
**ESTADO:**

**REGISTRAR**

Acerca de

El equipo está supervisado y protegido.

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo: DESKTOP-ONR0TEQ  
Procesador: Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz, 2.50 GHz  
RAM instalada: 6.00 GB (5.89 GB utilizable)  
Id. del dispositivo: ADC4541A-1027-40C3-91D7-FA880463A5ED  
Id. del producto: 00326-10000-00000-AA987  
Tipo de sistema: Sistema operativo de 64 bits, procesador x64  
Lápiz y entrada táctil: Compatibilidad con entrada táctil con 10

Proyecto Prelab

Proceso: [16728] Proyecto Prelab.exe

Eventos del ciclo de vida > Subproceso:

Marco de pila: Application Insights GitHub Copilot Explorador de soluciones Cambiar de GIT

**CRUDForm.cs (Diseño)** + CRUDForm.cs

**CRUDForm**

ID	Nombre	Grado	Sección	Teléfono	Matrícula	Ciclo	Email	Dirección	Fecha de Nacimiento	Promedio
1	RHYO	V Bach	B	0000 0000	20200012	Primero	examp...	calle grande	27/02/1986	96
2	DONATO	V Bach	B	0000 0000	20200012	Primero	examp...	calle vieja	27/02/1986	95
3	CARLOS	V Bach	C	0000 0000	20200012	Primero	examp...	calle dulce	27/02/1986	95
4	ENRIQUE	V Bach	C	0000 0000	20200012	Primero	examp...	calle chula	27/02/1986	95

**INGRESE LOS SIGUIENTES DATOS**

Creditos >> ?

ID:  NOMBRE:   
GRADO:

Exito

Alumno eliminado correctamente.

Aceptar

**TELEFONO:**   
**MATRICULA:**   
**CICLO:**   
**EMAIL:**   
**DIRECCIÓN:**   
**FECHA DE NACIMIENTO:**   
**PROMEDIO:**   
**ESTADO:**

**REGISTRAR**

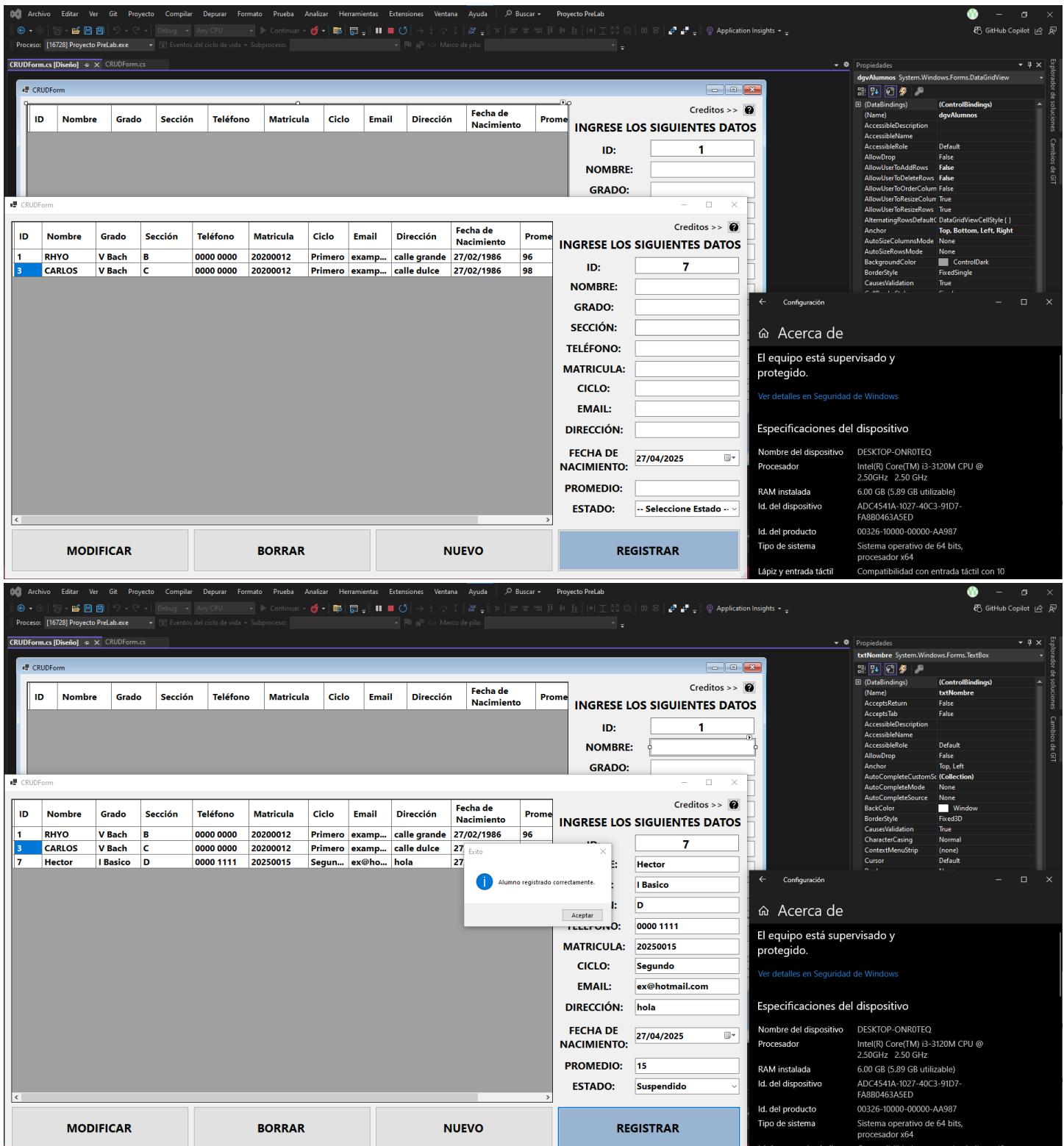
Acerca de

El equipo está supervisado y protegido.

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo: DESKTOP-ONR0TEQ  
Procesador: Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz, 2.50 GHz  
RAM instalada: 6.00 GB (5.89 GB utilizable)  
Id. del dispositivo: ADC4541A-1027-40C3-91D7-FA880463A5ED  
Id. del producto: 00326-10000-00000-AA987  
Tipo de sistema: Sistema operativo de 64 bits, procesador x64  
Lápiz y entrada táctil: Compatibilidad con entrada táctil con 10



Archivo Editar Ver Git Proyecto Compilar Depurar Formato Prueba Analizar Herramientas Extensiones Ventana Ayuda Buscar Proyecto PreLab

Proceso: [16728] Proyecto PreLab.exe Eventos del ciclo de vida > Subproceso: Application Insights GitHub Copilot

Propiedades: txtNombre System.Windows.Forms.TextBox

txtNombre (Name) txtNombre  
 AcceptsReturn False  
 AcceptsTab False  
 AccessibleDescription  
 AccessibleName  
 AccessibleRole Default  
 AllowDrop False  
 Anchor Top, Left  
 AutoCompleteCustomSc (Collection)  
 AutoCompleteMode None  
 AutoCompleteSource None  
 BackColor Window  
 BorderStyle Fixed3D  
 CausesValidation True  
 CharacterCasing Normal  
 ContextMenuStrip (none)  
 Cursor Default

INGRESE LOS SIGUIENTES DATOS

Creditos >> INGRESE LOS SIGUIENTES DATOS

ID: 1  
 NOMBRE:  
 GRADO:

Creditos >> INGRESE LOS SIGUIENTES DATOS

ID: 8  
 NOMBRE:  
 GRADO:  
 SECCIÓN:  
 TELÉFONO:  
 MATRICULA:  
 CICLO:  
 EMAIL:  
 DIRECCIÓN:  
 FECHA DE NACIMIENTO: 27/04/2025  
 PROMEDIO:  
 ESTADO: -- Seleccione Estado --

Configuración Acerca de

El equipo está supervisado y protegido.

Ver detalles en Seguridad de Windows

Especificaciones del dispositivo

Nombre del dispositivo	DESKTOP-ONR0TEQ
Procesador	Intel(R) Core(TM) i3-3120M CPU @ 2.50GHz, 2.50 GHz
RAM instalada	6.00 GB (5.89 GB utilizable)
Id. del dispositivo	ADG454HA-1027-40C3-91D7-FA880463A5ED
Id. del producto	00326-10000-00000-AA987
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil	Compatibilidad con entrada táctil con 10

CRUDForm.cs (Diseño) > CRUDForm.cs

CRUDForm

CRUDForm

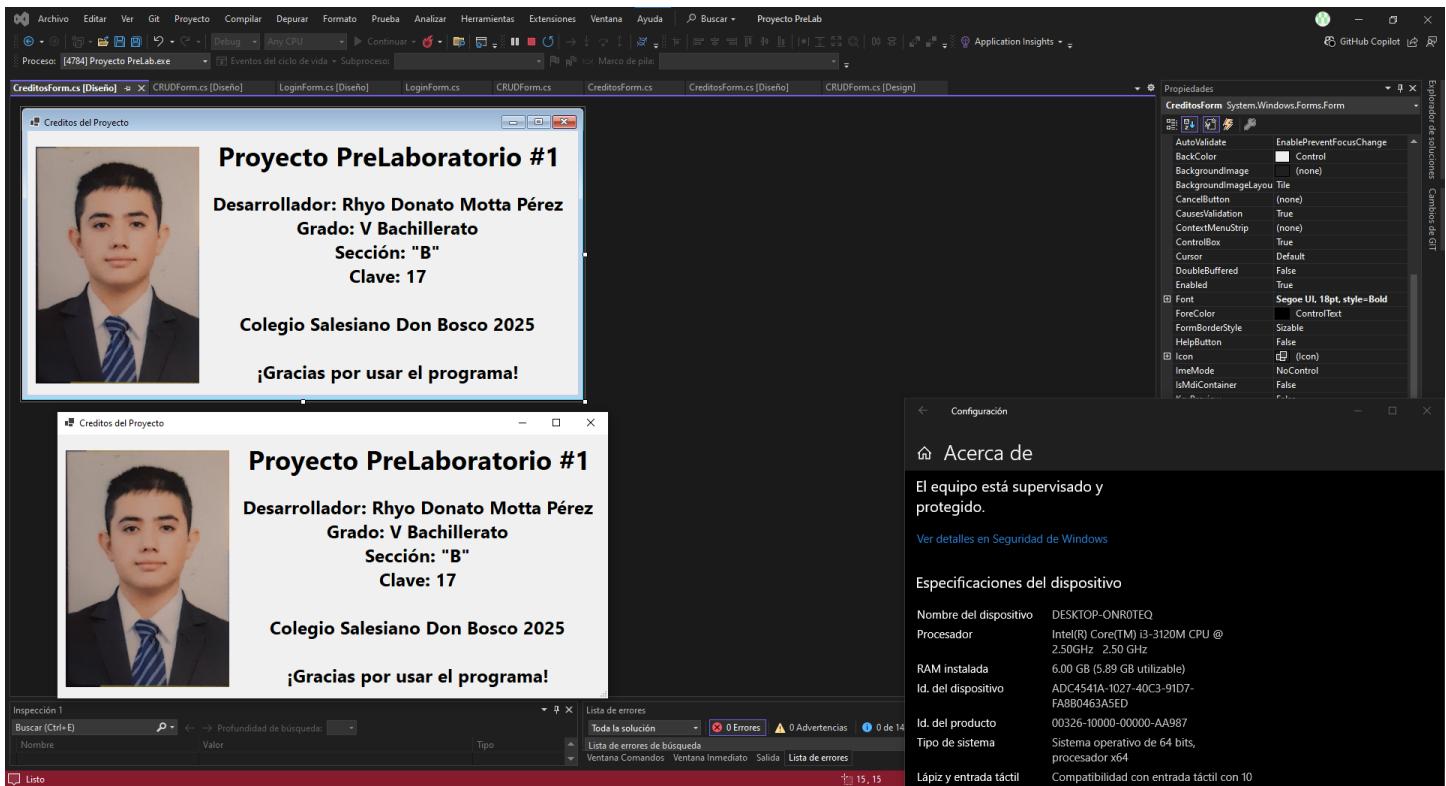
Proyecto PreLaboratorio #1

Desarrollador: Rhyo Donato Motta Pérez  
 Grado: V Bachillerato  
 Sección: "B"  
 Clave: 17

Colegio Salesiano Don Bosco 2025

¡Gracias por usar el programa!

MODIFICAR BORRAR NUEVO REGISTRAR



using System; // Proporciona clases fundamentales

using System.Collections.Generic; // Contiene colecciones genéricas como

using System.ComponentModel; // Proporciona funcionalidades para

using System.Data; // Clases para manejo de datos como

using System.Drawing; // Funcionalidades gráficas básicas

using System.Linq; // Soporte para LINQ (Language Integrated Query)

using System.Text; // Manipulación avanzada de cadenas

using System.Threading.Tasks; // Soporte para programación asíncrona

```
using System.Windows.Forms; // Clases para crear aplicaciones de Windows Forms

// Definición del espacio de nombres (namespace) Proyecto_PreLab
// -----
// Agrupa lógicamente todas las clases relacionadas con este proyecto

// Beneficios:
// 1. Organización lógica del código
// 2. Evita colisiones de nombres con otros componentes
// 3. Facilita el mantenimiento y la modularidad

namespace Proyecto_PreLab
{
    // Declaración de la clase principal CreditosForm que hereda de Form
    // -----
    // partial: Indica que la clase está dividida entre varios archivos
    // (normalmente el código principal y el código generado por el diseñador)
    // public: La clase es accesible desde otros assemblies (ensamblados)

    public partial class CreditosForm : Form
    {
        // Constructor de la clase CreditosForm
        // -----
        // Se ejecuta automáticamente al crear una instancia con 'new CreditosForm()'
```

```
// No recibe parámetros (constructor por defecto)

public CreditosForm()

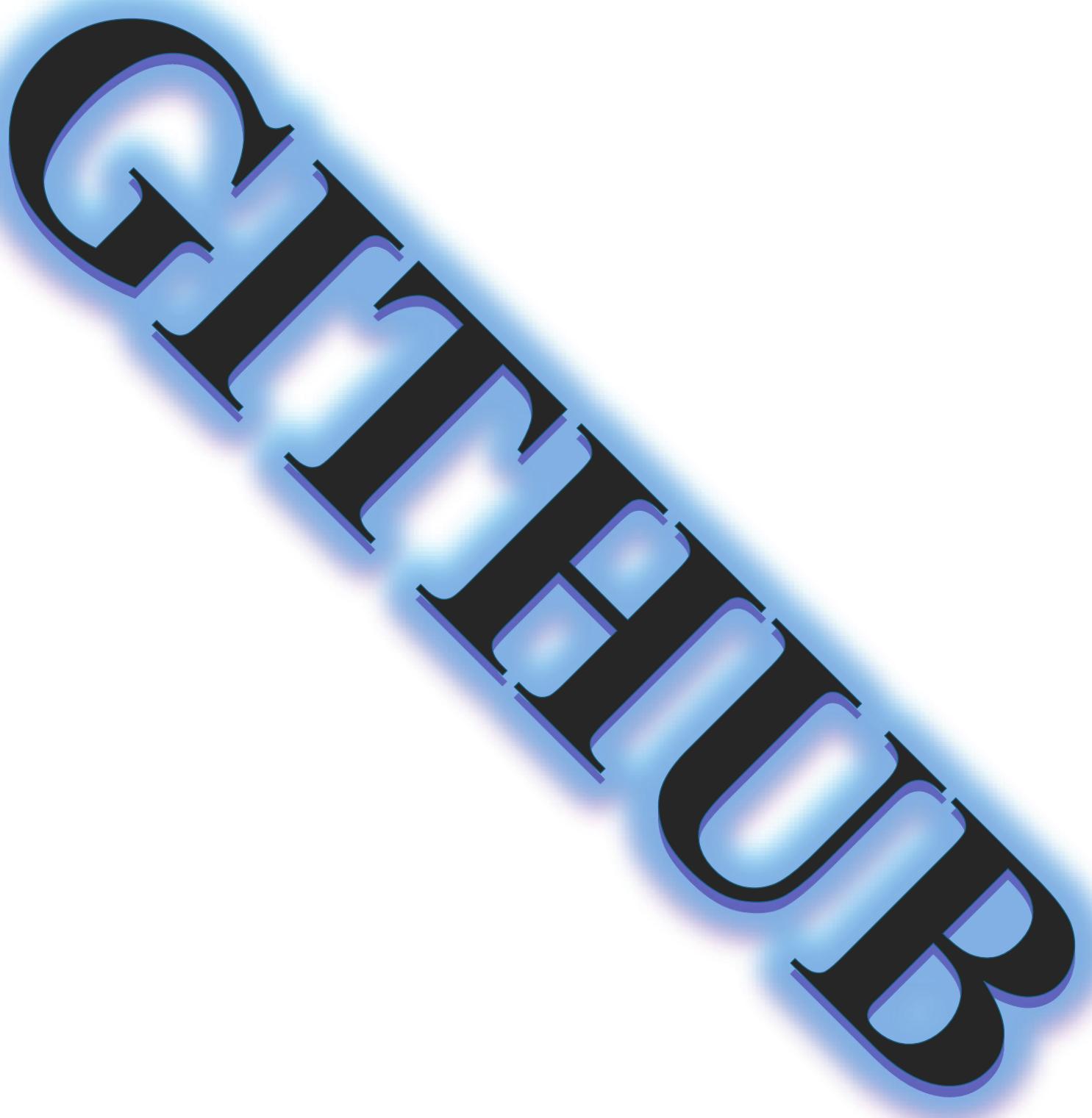
{
    // InitializeComponent(): Método generado automáticamente por el diseñador de Windows Forms

    // Responsabilidades:
    // 1. Instanciar todos los controles definidos visualmente en el diseñador
    // 2. Configurar sus propiedades iniciales (tamaño, posición, texto, etc.)
    // 3. Asignar manejadores de eventos básicos

    // DEBE ser la primera línea en el constructor porque:
    // - Asegura que todos los controles estén creados antes de cualquier otra operación
    // - Los controles deben existir antes de poder manipularlos

    InitializeComponent();
}

}
```



<https://github.com/2012002917-RhyoMotta/PRE->

LABORATORIO-I---PRACTICA-SUPERVISADA.git