

Ing. En sistemas computacionales

Taller De Bases De Datos Ago-Dic 2023 Imbo

Unidad 6: Conectividad a Bases de Datos

Proyecto Final: HEYDA

Equipo 6:

- Mena Méndez Heileen Carolina
  - Morales Garza Alan Yahir
- Muñoz Contreras Roberto Arturo
  - Palafox Soria Cristian Manuel

M.C. Juan Manuel Bernal Ontiveros

27 de noviembre de 2023

# Contenido

CAPÍTULO 1. INTRODUCCIÓN .....	4
1.1 Antecedentes del proyecto .....	4
1.2 Descripción de la problemática a resolver .....	4
1.3 Objetivos y metas.....	4
1.4 Justificación .....	4
CAPÍTULO 2. FUNDAMENTO TEÓRICO .....	5
Bases de datos: .....	5
Modelo relacional .....	5
Objetivos .....	5
Características .....	5
Definiciones .....	6
Servidor de Base de datos.....	6
¿Qué es? .....	6
Usos populares de servidores de bases de datos .....	6
¿Qué es un sistema de gestión de base de datos? .....	7
¿Qué permite el DBMS?.....	7
Tipos de sistemas de gestión de bases de datos .....	7
¿Qué es PostgreSQL? .....	8
Las características clave de PostgreSQL incluyen: .....	8
Instalación .....	9
CAPÍTULO 3. DESARROLLO .....	14
3.1 Requerimientos. ....	14
3.2 Diseño .....	14
3.2.1 Base de Datos .....	14
3.2.2 Diseño de clases.....	17
3.2.3 Diseño de interfaz de usuario .....	18
Menú inicio: .....	19
Menú de productos:.....	19
Menú de pedidos: .....	20
Sub ventana de productos: .....	20
Menú de empleados: .....	21
Menú para crear un nuevo empleado .....	21
Menú de proveedores .....	22
3.3 Codificación.....	23
Menús: .....	23
Menú de productos, proveedores y empleados:.....	24

CAPÍTULO 4. CONCLUSIONES ..... 26

Fuentes consultadas ..... 26

# CAPÍTULO 1. INTRODUCCIÓN

## 1.1 Antecedentes del proyecto

Los integrantes de este equipo cuentan con una amplia experiencia en cuanto se refiere a trabajar en el trato al cliente. En el caso de Heileen Mena, ella se desempeñó como encargada de una papelería por un periodo aproximado de 5 años.

Ella, al haber estado en un ámbito laboral donde se incurrían mucho los pedidos, ya fuera ordene y recoja o de consumo en el local, se encuentra altamente capacitada para encontrar los puntos de oportunidad y de mejora referentes a un sistema de producción relacionados a una empresa que brinda productos o servicios.

## 1.2 Descripción de la problemática a resolver

En la gran mayoría de las micro y pequeñas empresas existe una mala organización respecto a su sistema de pedidos de un producto y servicio, ya que en dichas empresas no consideran una necesidad el contar con un software especializado para cada tarea.

Debido a esto, es muy común de escuchar comentarios de parte de los trabajadores respecto a diversas inconformidades respecto a sus compañeros, puesto que no existe una buena comunicación a la hora de realizar un pedido, ya que se tiende a pasar poca o nula información al respecto.

## 1.3 Objetivos y metas

El objetivo de nuestro proyecto se basa en la generación de un software el cual permita un manejo de la información más fácil y rápido, logrando con eso, simplificar las labores relacionadas con la realización de pedidos para micro y pequeñas empresas.

## 1.4 Justificación

El proyecto que nos ocupa plantea mejorar y agilizar todo el proceso relacionado a los “pedidos para llevar” de una empresa, logrando implementar una bitácora para llevar un mejor seguimiento de los pedidos pendientes.

Para ello, se creará un software el cual mezclará un punto de venta con el formato de comandas que llega a implementar un restaurante, mostrándole a la persona encargada de preparar los pedidos todos los datos relacionados al mismo. Volviendo su trabajo mucho más fácil.

## CAPÍTULO 2. FUNDAMENTO TEÓRICO

### Bases de datos:

#### *Modelo relacional*

El modelo relacional, para el modelado y el arreglo de bases de datos, es un prototipo de datos basado en la dialéctica de predicados y en el parecer de conjuntos.

Tras individuo postuladas sus bases en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), nunca tardó en consolidarse como un reciente ejemplo en los modelos de fondo de datos.

Su representación esencial es la conveniencia de datos. Estos datos podrían considerarse en fase dialéctica como conjuntos de datos llamados tuplas. Pese a que esta es el parecer de las bases de datos relacionales creadas por Codd, la totalidad de las veces se conceptualiza de una práctica más obediente de imaginar, pensando en cada unión como si fuese un soporte que está compuesta por registros (cada línea del soporte sería un examen o “tupla”) y columnas (asimismo llamadas “campos”).

### Objetivos

Los objetivos que este modelo persigue son:

- **Independencia Física:** La forma de almacenar los datos no debe influir en su manipulación. Si el almacenamiento físico cambia, los usuarios que acceden a esos datos no tienen que modificar sus aplicaciones.
- **Independencia Lógica:** Las aplicaciones que utilizan la base de datos no deben ser modificadas por que se inserten, actualicen y eliminen datos.
- **Flexibilidad:** En el sentido de poder presentar a cada usuario los datos de la forma en que éste prefiera
- **Uniformidad:** Las estructuras lógicas de los datos siempre tienen una única forma conceptual (las tablas), lo que facilita la creación y manipulación de la base de datos por parte de los usuarios.
- **Sencillos:** Las características anteriores hacen que este Modelo sea fácil de comprender y de utilizar por parte del usuario final.

### Características

- Los datos son atómicos ó monovaluados;
- Los datos de cualquier columna son de un solo tipo.
- Cada columna posee un nombre único.
- El orden de las columnas no es de importancia para la tabla.
- Las columnas de una relación se conocen como atributos.
- Cada atributo tiene un dominio,
- No existen 2 filas en la tabla que sean idénticas.
- Las informaciones en las bases de datos son representadas como datos explícitos.
- Cada relación tiene un nombre específico y diferente al resto de las relaciones.
- Los valores de los atributos son atómicos: en cada tupla, cada atributo (columna) toma un solo valor. Se dice que las relaciones están normalizadas.
- El orden de los atributos no importa: los atributos no están ordenados. □ Cada tupla es distinta de las demás: no hay tuplas duplicadas

- El orden de las tuplas no importa: las tuplas no están ordenadas.
- Los atributos son atómicos: en cada tupla, cada atributo (columna) toma un solo valor. Se dice que las relaciones están normalizadas.

### Definiciones

- **Relación:** Tabla bidimensional para la representación de datos. Ejemplo: Estudiantes.
- **Tuplas:** Filas de una relación que contiene valores para cada uno de los atributos (equivale a los registros). Ejemplo: 34563, José, Martinez, 19, Masculino. Representa un objeto único de datos implícitamente estructurados en una tabla. Un registro es un conjunto de campos que contienen los datos que pertenecen a una misma entidad. □ **Atributos:** Columnas de una relación y describe las características particulares de cada campo. Ejemplo: id estudiante
- **Esquemas:** Forma de representar una relación y su conjunto de atributos. Ejemplo: Estudiantes (id estudiante, nombre(s), apellido(s), edad, género)
- **Claves:** Campo cuyo valor es único para cada registro. Principal, identifica una tabla, y Foránea, clave principal de otra tabla relacionada. Ejemplo: id estudiante.
- **Clave Primaria:** identificador único de una tupla.
- **Cardinalidad:** número de tuplas(m).
- **Grado:** número de atributos(n).
- **Dominio:** colección de valores de los cuales el atributo obtiene su atributo.

### Servidor de Base de datos

#### ¿Qué es?

Un servidor de base de datos, también conocido como servidor de base de datos o base de datos relacional RDBMS (sistema de administración de base de datos relacional), es un tipo de software de servidor que puede organizar información mediante tablas, índices y conjuntos de datos.

A nivel de hardware, un servidor de base de datos es un equipo de cómputo especializado para consultar clientes remotos o locales que solicitan información o modifican registros o tablas que residen en las bases de datos del sistema (a menudo, desde un servidor web o un servidor de aplicaciones).

### Usos populares de servidores de bases de datos

Existen muchos usos populares que se les ha dado a los servidores de base de datos desde su invención, por ejemplo:

- **Administración de registros de personas:** el uso de software para registros médicos, así como fichas de perfiles de clientes en clínicas, centros de odontología y más, es algo muy común de ver.
- **Administración de documentos:** sirve para organizar documentos de texto de diversa índole, algo muy usado en las diferentes empresas.
- **Administración contable e impositiva:** el uso de base de datos en estudios contables mejora el manejo de facturas, pagos de impuestos, gastos, ingresos y egresos.
- **Indexación de sitios web:** el caso más popular del mundo es Google con su inmensa base de datos para indexar, gestionar y modificar los aspectos de sitios webs en sus resultados.

- **Servir contenido dinámico:** con el auge del servidor PHP y la programación web, se comenzaron a usar las bases de datos para servir datos de foros, CMS, administraciones de usuarios, gestores de contenidos (como WordPress) y más.

### *¿Qué es un sistema de gestión de base de datos?*

Un sistema de gestión de base de datos (DBMS) es un conjunto de programas, invisibles para el usuario final, que se utilizan para administrar y administrar información, incluidas las bases de datos.

Data Manager o Database Manager le permite administrar todos los accesos a sus bases de datos. Esto se debe a que está destinado a actuar como una interfaz entre las bases de datos, los usuarios y las aplicaciones.

Sus funciones incluyen la capacidad de los usuarios comerciales para almacenar información, modificar datos y acceder a la base de conocimientos de la organización. De manera similar, el administrador de la base de datos procesa consultas y análisis para generar informes.

Además, un sistema de gestión de bases de datos puede entenderse como una colección de datos interrelacionados estructurados y organizados en un ecosistema formado por un conjunto de programas que acceden y gestionan los datos fácilmente.

### **¿Qué permite el DBMS?**

En otras palabras, el administrador de la base de datos controla todo lo que un usuario puede hacer en un BBDD. El desarrollo de esta funcionalidad generalmente requiere herramientas especiales como sistemas de búsqueda, sistemas de informes y varias aplicaciones. El administrador de base de datos también permite:

- Las interacciones con cada base de datos gestionada siempre se pueden desarrollar independientemente del programa o aplicación que las gestiona.
- Trabajar con bases de datos. Garantizamos seguridad, integridad y consistencia.
- Definición de bases de datos en diferentes niveles de abstracción.

### **Tipos de sistemas de gestión de bases de datos**

Los sistemas de gestión de bases de datos se tipifican según la forma en que se organizan los datos almacenados. (una fila consta de columnas para cada conjunto de datos y cada atributo) conectadas por correspondencia.

#### *Con base en modelos de datos*

- Sistema de red: Una estructura interrelacionada compleja ya que cada registro de datos se puede rastrear a través de una ruta diferente.
- Sistemas relacionales: almacenan datos en filas y columnas que forman tablas unidas por una clave común.

- Sistema multidimensional: una estructura similar a una tabla que almacena datos tal como se ven.
- Sistema de objetos: almacena todo tipo de datos gráficos como imágenes, sonidos y texto.

#### ***Basado en el número de usuarios:***

Depende del número de empleados necesarios para operar la base de datos. Por lo tanto, un sistema de administración de bases de datos se puede usar para permitir solo un usuario a la vez o para admitir múltiples usuarios.

Una ventaja es que puedes afinar los tipos de usuarios que pueden ver y navegar dentro de tu base de datos, siempre dependiendo de las necesidades de privacidad de tu empresa.

#### ***Basado en la distribución:***

- Sistema de gestión de datos centralizado: El sistema de gestión y la base de datos se almacenan y operan desde un solo lugar. Esto reúne la información de cada área y la herramienta de gestión de datos en una única plataforma.
- Sistema de gestión de datos distribuidos: Tanto la base de datos como el sistema de gestión están conectados a través de una red informática, pero se encuentran en ubicaciones diferentes.

#### ***¿Qué es PostgreSQL?***

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto. Es conocido comúnmente como "Postgres". Fue creado en la Universidad de California en Berkeley en la década de 1980 y desde entonces ha sido desarrollado y mantenido por una comunidad global de desarrolladores de software.

#### **Las características clave de PostgreSQL incluyen:**

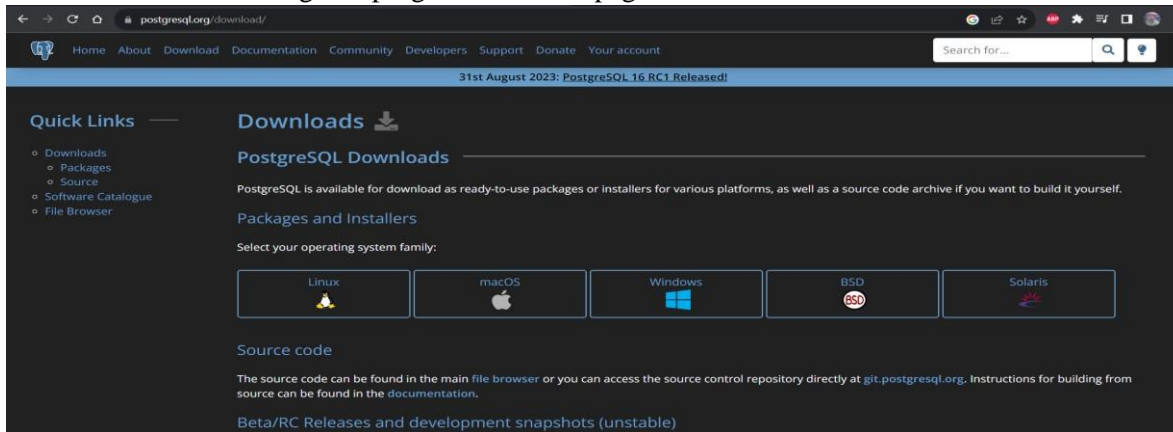
- **Modelo relacional:** PostgreSQL sigue el modelo de base de datos relacional, lo que significa que organiza los datos en tablas con filas y columnas, y permite establecer relaciones entre las tablas.
- **Transacciones:** Ofrece soporte para transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), lo que garantiza la integridad de los datos incluso en situaciones de fallo.
- **Extensibilidad:** PostgreSQL es altamente extensible, lo que permite a los desarrolladores agregar nuevas funciones y tipos de datos personalizados.
- **Lenguaje SQL:** Utiliza el lenguaje SQL (Structured Query Language) para consultar y manipular datos, lo que facilita la interacción con la base de datos.
- **Escalabilidad:** PostgreSQL es escalable y puede gestionar grandes cantidades de datos y cargas de trabajo de manera efectiva.
- **Soporte para tipos de datos avanzados:** Además de los tipos de datos comunes (enteros, cadenas, etc.), PostgreSQL admite tipos de datos avanzados como arrays, JSON, HSTORE (un almacén de pares clave-valor), y muchos otros.
- **Índices avanzados:** Ofrece una variedad de tipos de índices para acelerar las consultas, incluyendo índices B-tree, índices hash, índices GiST y más.
- **Compatibilidad con estándares:** Cumple con los estándares SQL y ofrece características adicionales que no están presentes en todos los sistemas de gestión de bases de datos relacionales.



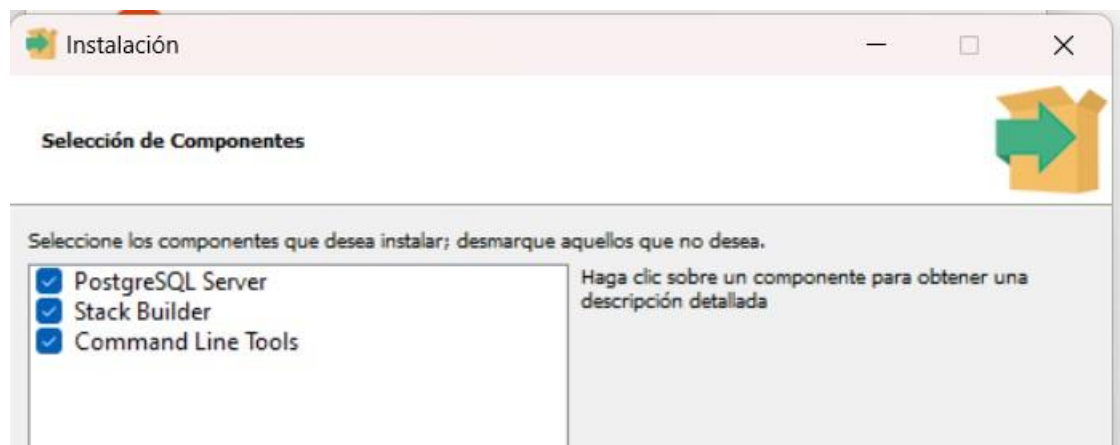
PostgreSQL es ampliamente utilizado en aplicaciones empresariales y proyectos de código abierto debido a su robustez, rendimiento y licencia de código abierto. Es una opción popular para aquellos que necesitan una base de datos relacional confiable y poderosa.

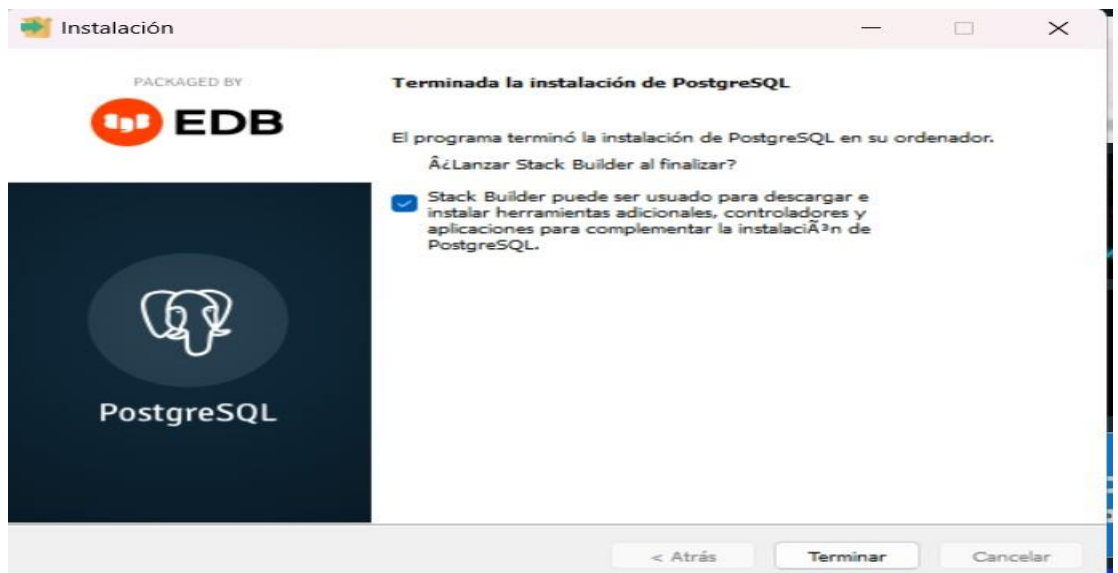
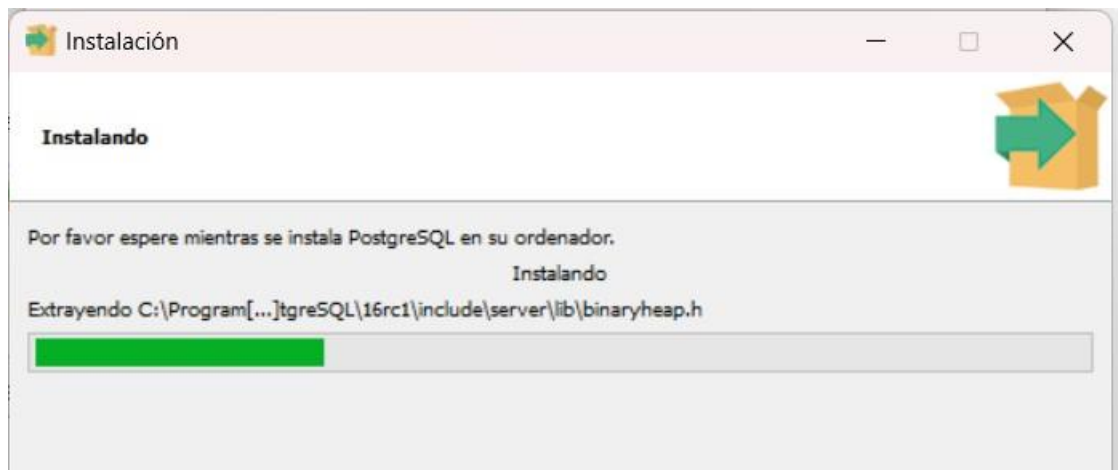
## Instalación

1. Procedemos a descargar el programa desde la página oficial.

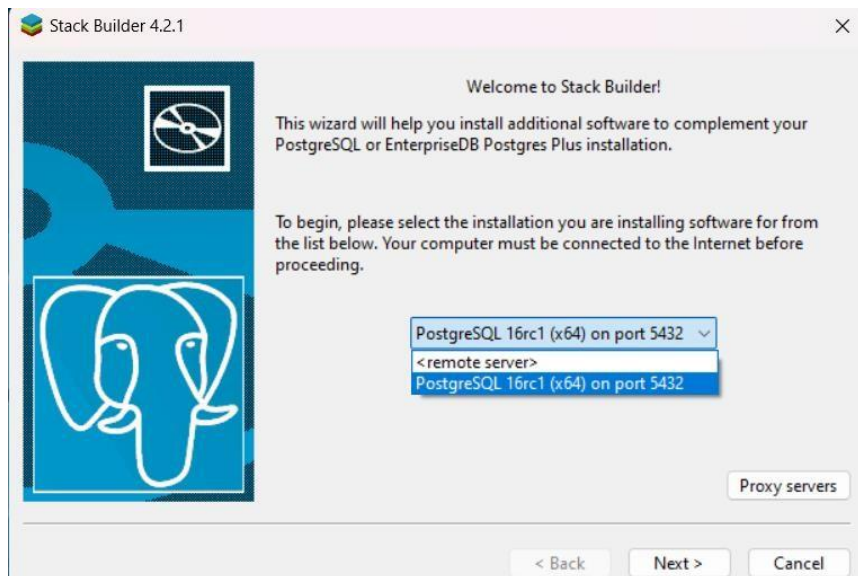


2. Seleccionamos los componentes a instalar y procedemos a toda la instalación.

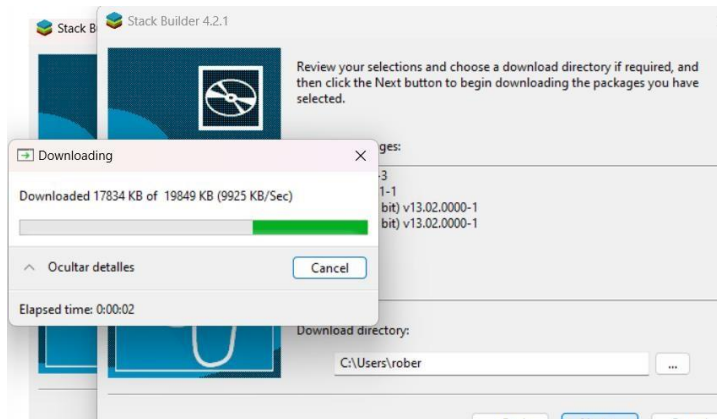
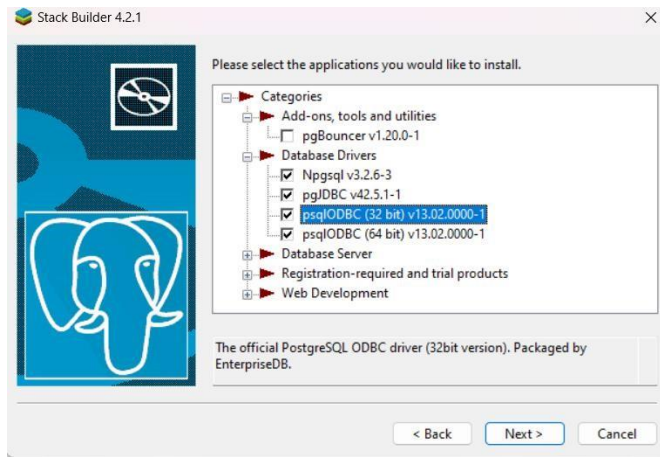




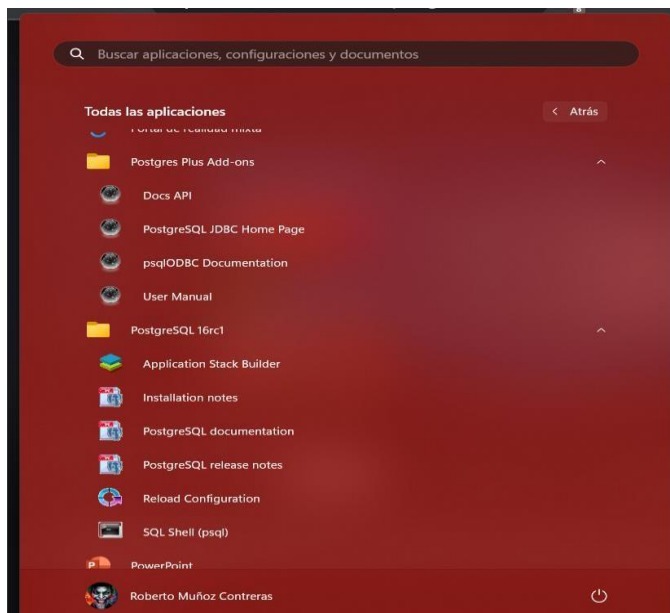
3. Aquí al inicializar el Builder seleccionamos el PostgreSQL que usaremos.



4. Seleccionamos todos los drivers de la Base de datos para evitar futuros conflictos.



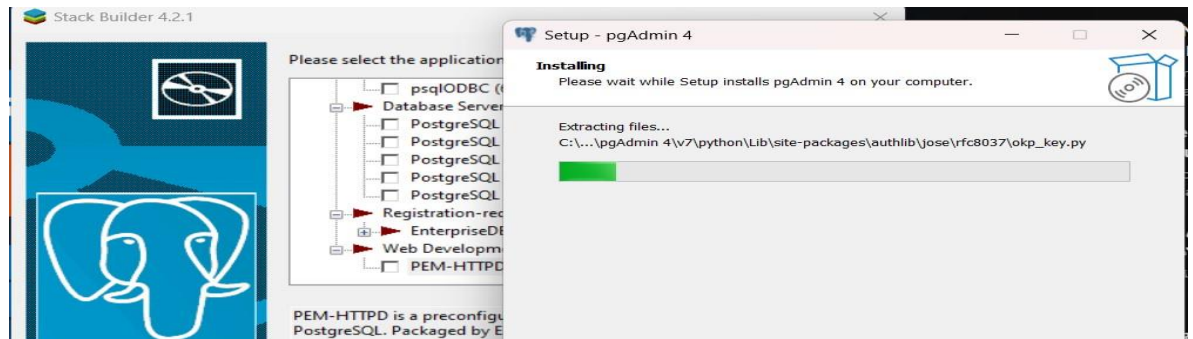
5. Después de descargar e instalar todos los drivers todo se va a guardar en la carpeta madre que tu le dijiste al instalador que toma el nombre de PostgreSQL en el Disco C:, ya nos aparecerá en nuestra maquina todo lo relacionado con la instalación



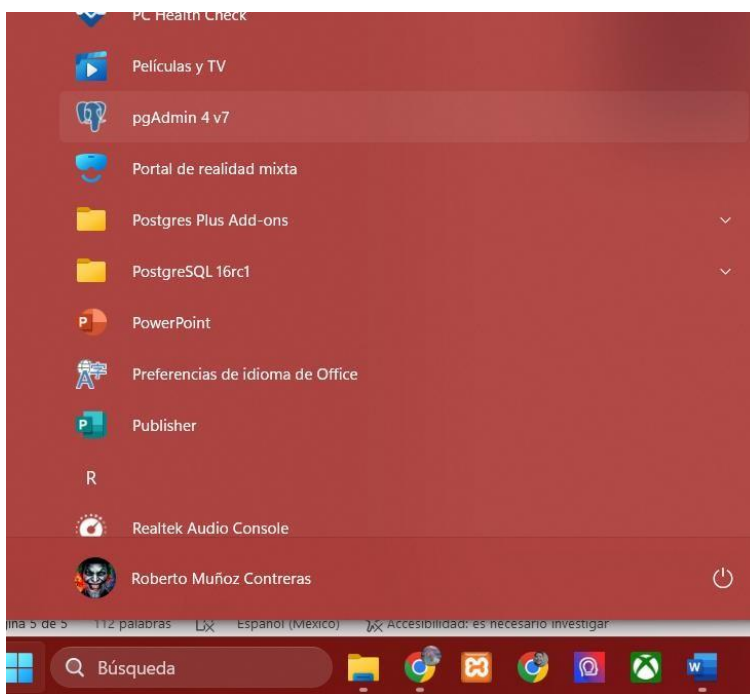
6. Abrimos el Shell e ingresamos la contraseña que definimos en la instalación (En este caso 1234)

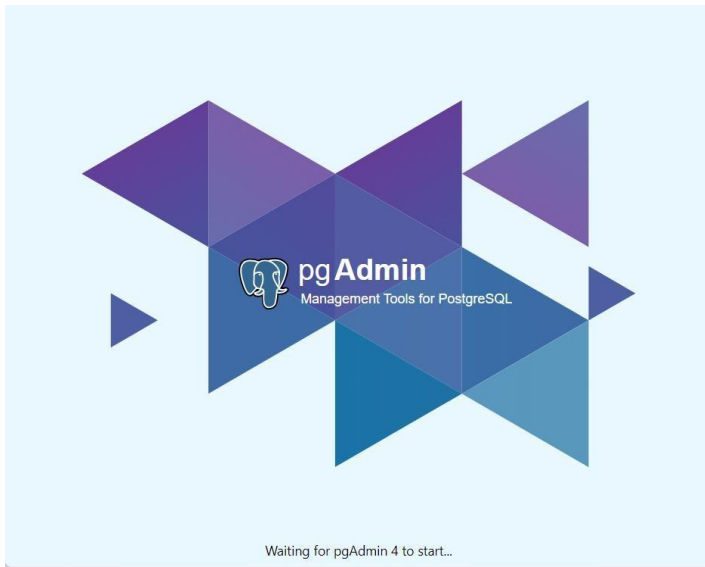
```
Selecionar SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:
psql (16rc1)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.
postgres=#
```

7. Procedemos agregar la instalación de phpMyAdmin



8. Una vez instalado abrimos el programa.





9. Esta es la interfaz, nos da acceso libre y con una interfaz grafica a las bases de dato, los catálogos, extensiones, eventos trágicos, entre otras muchas cosas mas, siendo muy atractivo a la vista y fácil de manejar.

The screenshot shows the pgAdmin 4 web interface. On the left is the "Object Explorer" tree, showing the hierarchy of the PostgreSQL 16rc1 database. The main area displays the "Dashboard" with several charts: "Database sessions" (Total, Active, Idle), "Transactions per second" (Transactions, Commits, Rollback), "Tuples in" (Inserts, Updates, Deletes), "Tuples out" (Fetched, Returned), and "Block I/O" (Reads, Hits). Below these charts is the "Database activity" section, which includes a table of active sessions.

Database activity							
Sessions							
			PID	User	Application	Client	Backend start
✖	■	▶	3756	postgr...	pgAdmin 4 - DB:post...	::1	2023-09-12 20:55:33...

At the bottom of the interface, a status bar shows the current server: "Servers > PostgreSQL 16rc1" and a message: "Auto-detected PostgreSQL 16rc1 installation with the data directory at C:\Program Files\PostgreSQL\16rc1\data".

## CAPÍTULO 3. DESARROLLO

### 3.1 Requerimientos.

Nuestro proyecto permite:

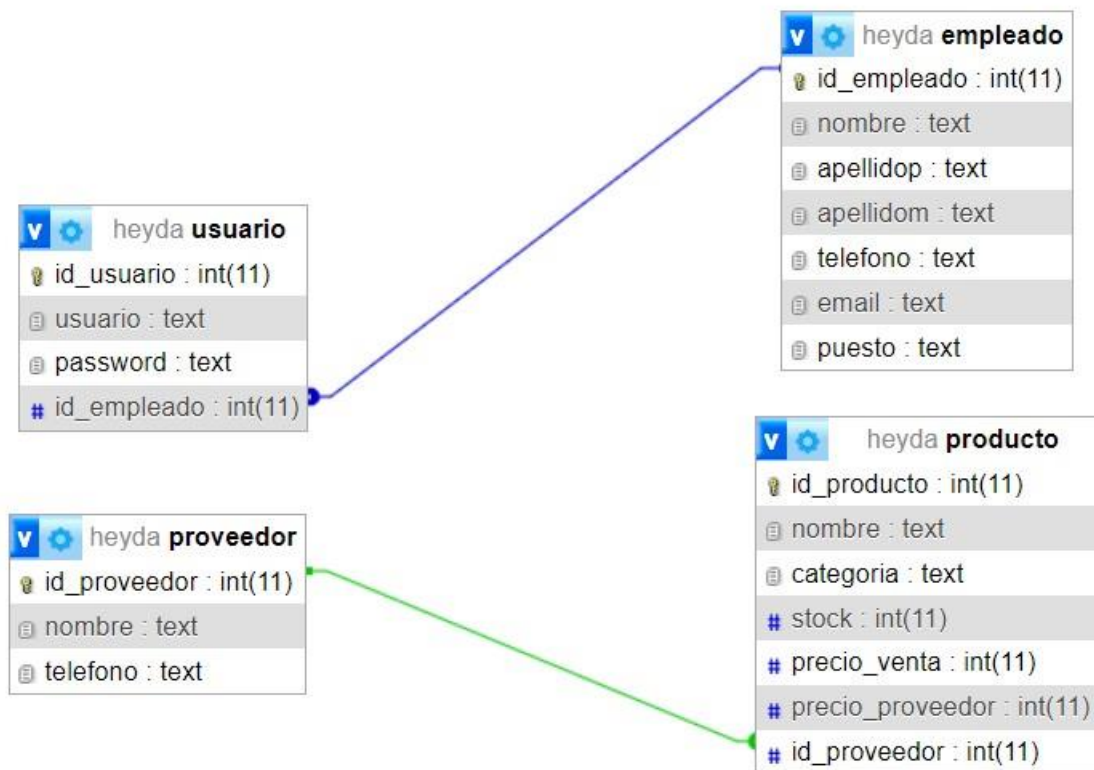
1. Creación del registro de nuevos empleados.
2. En base a los empleados habilitados, crearles un usuario para acceder a la aplicación.
3. Dar de alta productos y proveedores.
4. Realizar un reporte del pedido, en el cual se realiza en formato de ticket.
5. Manejo de inventario mediante transacciones.

### 3.2 Diseño

#### 3.2.1 Base de Datos

El proyecto terminó con 4 tablas, usuario, empleado, proveedor y producto, las cuales se encuentran relacionadas, tal como se aprecia en la Ilustración 1.

Por motivos de comodidad, la base de datos se creó primero con las clases con llave primaria, para luego anexar las tablas que contienen la llave foránea, y con ello no tener problemas con la inconsistencia de los datos.



*Ilustración 1 Relaciones de la base de datos*



empleado

General

Columns

Advanced

Constraints

Parameters

Security















SQL



Inherited from table(s)

Select to inherit from...

Columns

+

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
 	id_empleado	numeric			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
 	nombre	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 	apellidop	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 	apellidom	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 	telefono	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 	email	text			<input type="checkbox"/>	<input type="checkbox"/>	
 	puesto	text			<input type="checkbox"/>	<input type="checkbox"/>	

Close

Reset

Save

*Ilustración 2 tabla de empleados*

producto

General

Columns

Advanced

Constraints

Parameters

Security















SQL



Inherited from table(s)

Select to inherit from...

Columns

+

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
 	id_producto	numeric			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
 	nombre	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 	categoria	text			<input type="checkbox"/>	<input type="checkbox"/>	
 	stock	numeric			<input type="checkbox"/>	<input type="checkbox"/>	
 	precio_venta	numeric			<input type="checkbox"/>	<input type="checkbox"/>	
 	precio_proveedor	numeric			<input type="checkbox"/>	<input type="checkbox"/>	
 	id_proveedor	numeric			<input type="checkbox"/>	<input type="checkbox"/>	

Close

Reset

Save

*Ilustración 3 Tabla de productos*

proveedor

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s) 

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	id_proveedor	numeric			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	nombre	text			<input type="checkbox"/>	<input type="checkbox"/>	
	telefono	text			<input type="checkbox"/>	<input type="checkbox"/>	

*Ilustración 4 Tabla de proveedores*

usuario

General

Columns

Advanced

Constraints

Parameters

Security

SQL

Inherited from table(s) 

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	id_usuario	numeric			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	usuario	text			<input type="checkbox"/>	<input type="checkbox"/>	
	password	text			<input type="checkbox"/>	<input type="checkbox"/>	
	id_empleado	numeric			<input type="checkbox"/>	<input type="checkbox"/>	

Close

Reset

Save

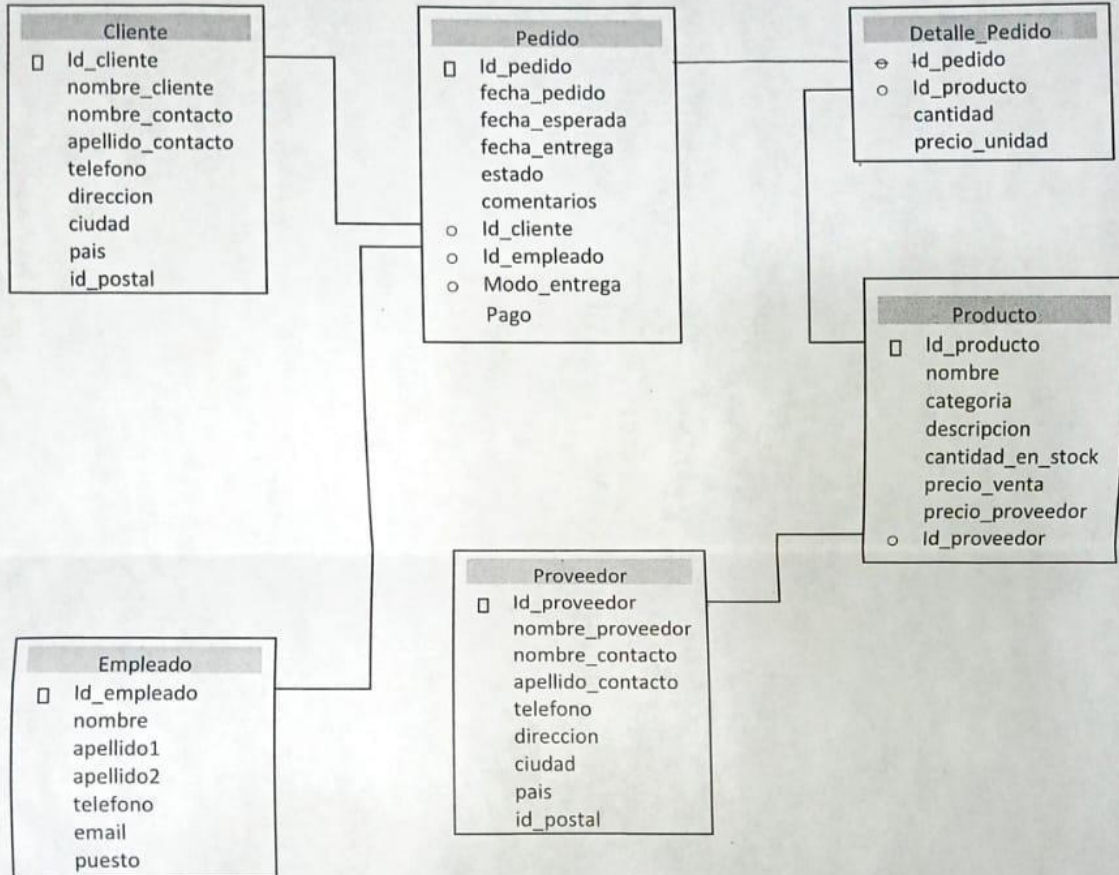
*Ilustración 5 Tabla de usuarios*



## HEYDA: Modelo relacional

### Prueba 2

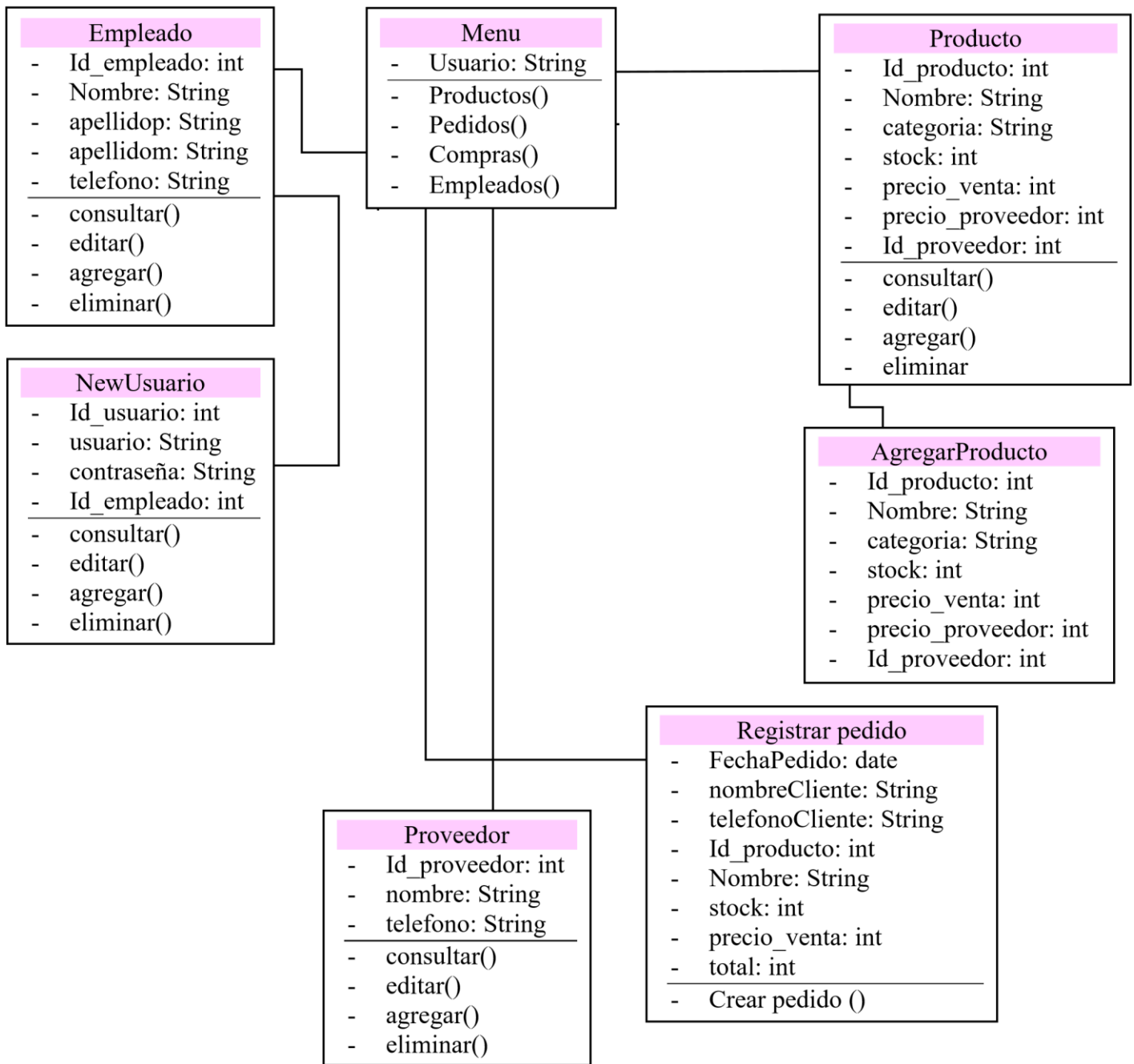
□ Llave primaria. ○ Llave foránea.



700000  
28-11-2023

Mena Méndez Heileen Carolina

### 3.2.2 Diseño de clases



### 3.2.3 Diseño de interfaz de usuario

Menú inicio:

Productos

Pedidos

Empleados

Crear Usuarios

Proveedores

Cerrar Sesión

10:00:00

lunes, 27 de noviembre de 2023

Menú de productos:

Productos

Pedidos

Empleados

Crear Usuarios

Proveedores

Cerrar Sesión

ID:

PRODUCTO:

CATEGORIA:

STOCK:

PRECIO VENTA:

PRECIO PROVEEDOR:

PROVEEDOR:

AGREGAR

ACTUALIZAR

ELIMINAR

PRODUCTOS

id_producto	nombre	categoria	stock	precio_venta	precio_proveedor	nombre1
8520	Pluma Bic Azul	Todo lo relacio...	21	15.78	8	BACO
9736	Pegamento Vin...	Artículos de so...	36	32	19	BIC
117596	Cuaderno 100 h...	Papeles y etiqu...	78	25	10	BACO
1478	Pluma bic negra	Todo lo relacio...	24	12	8	BIC

MODIFICAR

BUSCAR PROVEEDOR POR NOMBRE:

PROVEEDORES

id_proveedor	nombre	telefono
4	BARRILITO	6563698520
5	PAPER MATE	6567812450
1	BACO	6561087414
8	VINCI	6564817412
3	BIC	6561017412

## Menú de pedidos:

- Productos
- Pedidos
- Empleados
- Crear Usuarios
- Proveedores

### REGISTRAR PEDIDO

INFORMACIÓN VENTA:

FECHA PEDIDO: lunes, 27 de noviembre de 2023

FECHA ENTREGA: lunes, 27 de noviembre de 2023

INFORMACIÓN CLIENTE:

NOMBRE:

TÉLEFONO:

INFORMACIÓN PRODUCTO:

COD. PRODUCTO:

PRECIO:

PRODUCTO:

STOCK:

CANTIDAD:

	IdProducto	Producto	Precio	Cantidad	SubTotal
*					

TOTAL A PAGAR:

CREAR PEDIDO

## Sub ventana de productos:

AgregarProducto

PRODUCTOS

BUSCAR PRODUCTO POR NOMBRE:

id_producto	nombre	categoria	stock	precio_venta	precio_proveedor
740802	Pluma Baco Punto Fino Negra	Todo lo relacionado con la escritura	62	8	4
96385	Marcador Expo	Todo lo relacionado con la escritura	12	14	7
9630	Abaco	Artículos de sobremesa	25	28	14
96308520	Agenda	Organización personal	4	296	150

## Menú de empleados:

Productos

Pedidos

Empleados

Crear Usuarios

Proveedores

Cerrar Sesión

### EMPLEADOS

ID:

NOMBRE:

APELLIDO PATERNO:

APELLIDO MATERNO:

TÉLEFONO:

EMAIL:

PUESTO:

AGREGAR

GUARDAR

ELIMINAR

id_empleado	nombre	apellido	apellidom	telefono	email	puesto
3	Alan	Morales	Garza	6569632587	alan@gmail.com	VEN
1	Heileen	Mena	Mendez	6561015485	heileenmena41@gmail.com	ADM
4	Roberto	Munoz	Contreras	6561234567	Roberto.m@gmail.com	CON

MODIFICAR

## Menú para crear un nuevo empleado

Productos

Pedidos

Empleados

Crear Usuarios

Proveedores

Cerrar Sesión

### USUARIOS

ID:

USUARIO:

CONTRASEÑA:

ID EMPLEADO:

AGREGAR

GUARDAR

ELIMINAR

### USUARIOS CREADOS

id_usuario	usuario	password	id_empleado
1	ADMIN	ADMIN	1
2	ALAN	1230	0

MODIFICAR

BUSCAR EMPLEADO POR NOMBRE:

### EMPLEADOS HABILITADOS

id_empleado	nombre	apellido	apellidom	telefono	email	puesto
3	Alan	Morales	Garza	6569632587	alan@gmail.com	VEN
1	Heileen	Mena	Mendez	6561015485	heileenmena41@gmail.com	ADM
4	Roberto	Munoz	Contreras	6561234567	Roberto.m@gmail.com	CON

Menú de proveedores

Productos

Pedidos

Empleados

Crear Usuarios

Proveedores

PROVEEDORES

ID:

NOMBRE:

TELÉFONO:

AGREGAR

GUARDAR

ELIMINAR

MODIFICAR

id_proveedor	nombre	telefono
4	BARRILITO	6563698520
5	PAPER MATE	6567812450
1	BACO	6561087414
8	VINCI	6564817412
3	BIC	6561017412

### 3.3 Codificación

#### Menús:

En todos los menús recurre a la misma instancia, se trata de un form padre que invoca al form hijo, en este caso, la parte de fondo gris claro es el form hijo, al momento de pulsar cada botón del panel azul este va cambiando.

El panel de color azul fue creado con el objetivo de simular un menú, está conformado por 2 paneles, el panel principal y otro que se muestra una vez que se pulsa el botón “reportes”, logrando dar el efecto de un menú desplegable.

Los menús cuentan con una barra superior conformada por 4 botones cerrar, maximizar, minimizar y restaurar la pantalla, los botones maximizar y restaurar se encuentran uno sobre el otro, para lograr una imagen similar a los botones de cualquier software de computadora.



```
private void AbrirFormHija(object formhija)
{
    if (this.panelContenedor.Controls.Count > 0)
    {
        this.panelContenedor.Controls.RemoveAt(0);
    }
    Form fh = formhija as Form;
    fh.TopLevel = false;
    fh.Dock = DockStyle.Fill;
    this.panelContenedor.Controls.Add(fh);
    this.panelContenedor.Tag = fh;
    fh.Show();
}
```

### Menú de productos, proveedores y empleados:

La codificación de estos 3 formularios hija es idéntica, puesto que el objetivo de estas ventanas es lograr consultar, agregar, modificar o eliminar los datos de su categoría, para lo cual se han agregado las etiquetas y cuadros de texto correspondientes a los campos de dichas tablas de las bases de datos.

Siendo complementadas por un datagridview, en el cual, al seleccionar un campo de su interior mostrará en los cuadros de texto su información correspondiente, permitiendo el uso de los botones modificar y eliminar, ya que en la codificación de los mismos el Query está conformado por los datos que se encuentren en dichos cuadros de texto.

También, se le anexó en la parte interior un área para búsqueda por filtro, en donde se utiliza una sentencia LIKE de mysql. Este mismo filtro se utiliza en la sub ventana de productos y en la ventana para la creación de un nuevo usuario.

**PRODUCTOS**

id_producto	nombre	categoria	stock	precio_venta	precio_proveedor	nombre1
8520	Pluma Bic Azul	Todo lo relacio...	21	15.78	8	BACO
9736	Pegamento Vin...	Artículos de so...	36	32	19	BIC
117596	Cuaderno 100 h...	Papeles y etiqu...	78	25	10	BACO
1478	Pluma bic negra	Todo lo relacio...	24	12	8	BIC

**PROVEEDORES**

id_proveedor	nombre	telefono
4	BARRILITO	6563698520
5	PAPER MATE	6567812450
1	BACO	6561087414
8	VINCI	6564817412
3	BIC	6561017412

```
//Pasar los datos del DataGridView a los textbox
1 referencia
private void btnModificar_Click(object sender, EventArgs e)
{
    txtId.Texts = dgvProductos.CurrentRow.Cells[0].Value.ToString();
    txtProducto.Texts = dgvProductos.CurrentRow.Cells[1].Value.ToString();
    cbCategoria.Text = dgvProductos.CurrentRow.Cells[2].Value.ToString();
    txtStock.Texts = dgvProductos.CurrentRow.Cells[3].Value.ToString();
    txtPrecioVenta.Texts = dgvProductos.CurrentRow.Cells[4].Value.ToString();
    txtPrecioProveedor.Texts = dgvProductos.CurrentRow.Cells[5].Value.ToString();
}
```



```
private void txtBuscarProveedor__TextChanged(object sender, EventArgs e)
{
    string consulta = "SELECT * FROM `proveedor` WHERE `nombre` LIKE '" + txtBuscarProveedor.Texts + "%'";
    MySqlConnection MyConn2 = new MySqlConnection(MyConnection2);
    MySqlDataAdapter adaptador = new MySqlDataAdapter(consulta, MyConn2);
    MyConn2.Open();
    DataTable dt = new DataTable();
    adaptador.Fill(dt);
    dgvProveedores.DataSource = dt;
}
```

## CAPÍTULO 4. CONCLUSIONES

Después de muchas horas de trabajo y tazas de café, hemos llegado a la conclusión de que la implementación de una base de datos es una tarea fácil de hacer, pero muy extensa y con múltiples posibilidades de falla, lo cual lo vuelve una actividad bastante tediosa.

Pero a su vez, al terminar el trabajo, se siente una enorme sensación de felicidad, por haber completado exitosamente la actividad.

### Fuentes consultadas

Borges, E., & Borges, E. (2019). Servidor Base de Datos. Infranetworking.

<https://blog.infranetworking.com/servidor-base-de-datos/>

Da Silva, D. (2021, 13 julio). Open Source: qué es, ventajas y diferencias con open API. *Zendesk MX*.

<https://www.zendesk.com.mx/blog/que-es-open-source/>

Araneda, P. (2021, 31 marzo). *Capítulo 7 El Modelo Relacional | Base de Datos*.

<https://bookdown.org/paranedagarcia/database/el-modelo-relacional.html>

*Gestores de base de datos: qué son y qué tipos existen*. (s. f.). UNIR México.

<https://mexico.unir.net/ingenieria/noticias/gestores-de-base-de-datos/>

*Quickly Develop Cloud Applications with MySQL Heatwave*. (s. f.). <https://www.oracle.com/mysql/>

▷ *Ventajas y Desventajas de MYSQL*. (s. f.). <https://codigosql.top/ventajas-y-desventajas-de-mysql/>