

# **Actividad 1 - Creación de la Base de Datos.**

## **Lenguajes de Programación II.**

### **Ingeniería en Desarrollo de Software**

**Tutor: Miguel Ángel Rodríguez Vega**

**Alumno: Ramón Ernesto Valdez Felix**

**Fecha: 13/11/2023**

## Índice

Introducción .....	3
Descripción .....	3
Justificación .....	4
Desarrollo .....	4
Modelo Entidad-Relación .....	5
Modelo Lógico-Relacional .....	6
Base de Datos .....	6
Conclusión .....	9
Referencias .....	9

## Introducción

En la actividad 1 de la materia de lenguajes de programación II, se necesita realizar una estructura de clases a la empresa UNI donde pueda controlar sus diferentes tipos de empleados con sus datos personales y se correlaciones con las subclases a utilizar para obtener la información deseada por la empresa UNI usando como ejemplo las siguientes: (Persona, puesto y Centro) estas deben de utilizar atributos que es correlaciones entre si mostrando la información completa el empleado como “Nombre completo, RFC, fecha nacimiento entre otros atributos” esto permitiendo mostrar toda la información al ser solicitada en el código script de C++, adicionalmente se requiere realizar crear los diagramas de los métodos de entidad- relación y lógica-relacional para completar con dicha actividad principal.

## Descripción

En esta actividad 1 entregaremos el documento realizado de nombre “Creación de la base de datos ” esto nos dará el derecho a ser calificada para así obtenerte la puntuación de la calificación final de la materia impartida por el docente o maestro asignado a la materia de lenguajes de la programación II, ya que es necesario realizar la documentación para la actividad 1 donde se nos pide que trabajemos con tres cosas principales métodos de entidad-relación, lógica-relacional, la creación de la base de datos en SQL Server y el código script de C++ donde mostremos el uso de las clases para obtener el requerimiento solicitado por la empresa UNI para así tener el control de sus empleados esto fue proporcionado en la documentación de academia, ejecutarlo el script y tomar capturas de la evidencia así como anexar la evidencia de los métodos solicitados.

## Justificación

En esta actividad se trabajará los diagramas de métodos de entidad-relación, lógica-relacional y con la programación del script de C++ utilizando clases esto para obtener el resultado del funcionamiento correcto del script:

- Usar clases, herencia de clases y atributos.
- Subirlo al GitHub el documento realizado compartiendo el link para que pueda consultar el docente o maestro.
- Descargar e instala visual estudio para la creación del script de C++ a utilizar.
- Catálogo de puestos a utilizar.
- Analizar los datos que se presentan en la contextualización.
- Realizar un diagrama Entidad-Relación y un diagrama Lógico-Relacional.
- Crear la base de datos (BD) en Microsoft SQL Server para el sistema. El programa se desarrollará en lenguaje C++.

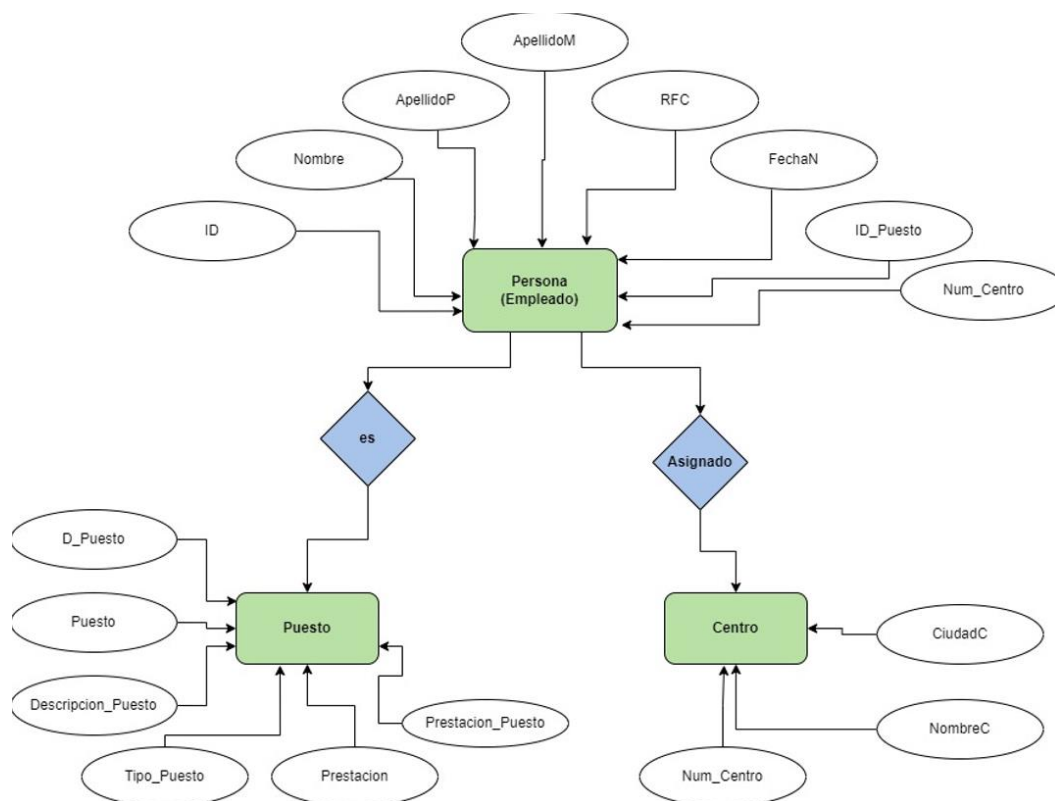
## Desarrollo:

En esta actividad realizaremos las actividades siguientes un diagrama Entidad-Relación, un diagrama Lógico-Relacional y la creación de la base de datos en SQL como también el programa en el lenguaje de programación de C++ utilizando la herramienta de Visual Studio y ejecutar en script para la toma de la evidencia a recolectara, así como la creación de la base en SQL Server.

**Link:** [GitHub](#)

## Modelo Entidad-Relación.

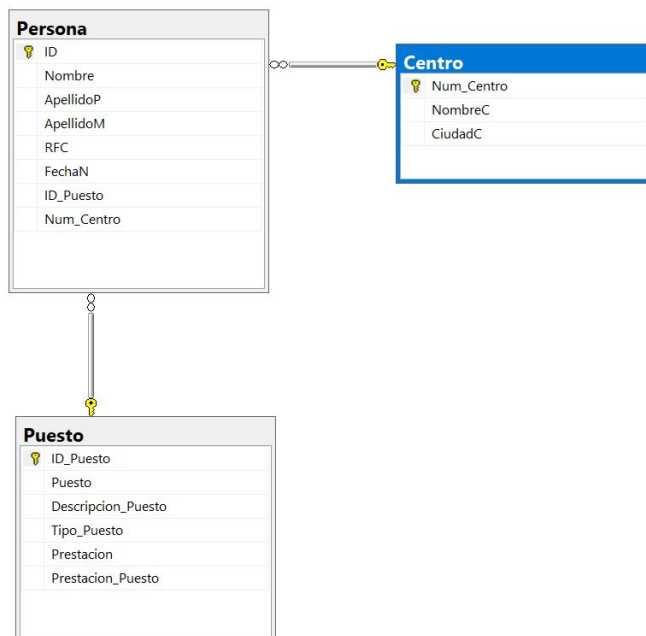
En este punto de la primera actividad de la materia, se nos pide la creación de recomienda diagrama de método de entidad-relación el cual nos da la idea de cómo realizar la correlación de la información a recabar y así tener el control para de dicha información que será administrada utilizada de forma correcta para el uso exclusivo de la empresa UNI asi teniendo una mejora en su administración de recursos o empleados:



Este modelo es una herramienta para el modelo de datos, la cual nos facilita el poder representar las entidades de una base de datos. los diagramas ER son un lenguaje gráfico para describir conceptos. Informalmente, son simples dibujos o gráficos que describen información que trata un sistema de información y el software que lo automatiza.

## Modelo Lógico-Relacional.

En esta actividad anexaremos las pantallas de evidencia de creación del diagrama del modelo lógico relacional de la base de datos creada en la actividad solicitada por la materia de Lenguajes de la Programación II:



## Base de Datos.

En esta actividad anexaremos las pantallas de evidencia de creación y ambientación de la base de datos que nos permitirá almacenar los datos solicitados en la actividad en cuestión:

- Creación de tablas y atributos de la base de datos de actividad 1.

Iniciamos creando la primera tabla que nos servirá como catálogo para el llenado de la base de datos, creamos la tabla de puestos e indicamos los tipos de puestos que existen, con sus diferentes atributos como mostramos en las imágenes de evidencia:

```
CREATE TABLE [dbo].[Puesto](
    [ID_Puesto] [int] NOT NULL PRIMARY KEY,
    [Puesto] [nvarchar](30) NOT NULL,
    [Descripcion_Puesto] [nvarchar](30) NOT NULL,
    [Tipo_Puesto] [bit] NOT NULL,
    [Prestacion] [nvarchar](4) NOT NULL,
    [Prestacion_Puesto] [nvarchar](15) NULL,
);
SET IDENTITY_INSERT dbo.Puesto ON;
INSERT INTO dbo.Puesto (ID_Puesto, Puesto, Descripcion_Puesto, Tipo_Puesto, Prestacion, Prestacion_Puesto)
VALUES (1, 'Gerente', 'Gerente Supervisor Depto. TDA', 1, 'Si', 'Combustible');
INSERT INTO dbo.Puesto (ID_Puesto, Puesto, Descripcion_Puesto, Tipo_Puesto, Prestacion, Prestacion_Puesto)
VALUES (2, 'Vendedor', 'Vendedor Deptos. TDA', 0, 'No', NULL);
INSERT INTO dbo.Puesto (ID_Puesto, Puesto, Descripcion_Puesto, Tipo_Puesto, Prestacion, Prestacion_Puesto)
VALUES (3, 'Cajero', 'Punto de Vta. TDA', 0, 'No', NULL);
```

	ID_Puesto	Puesto	Descripcion_Puesto	Tipo_Puesto	Prestacion	Prestacion_Puesto
1	1	Gerente	Gerente Supervisor Depto. TDA	1	Si	Combustible
2	2	Vendedor	Vendedor Deptos. TDA	0	No	NULL
3	3	Cajero	Punto de Vta. TDA	0	No	NULL


Query executed successfully.

Iniciamos creado la segunda tabla que también nos servirá como catalogo para el llenado de la base de datos, en este punto creamos la tabla de Centro e indicamos los tipos de puestos que existen, con sus diferentes atributos como mostramos en las imágenes de evidencia donde el atributo num\_centro declaramos que sea auto incremental y asignamos la numeración deseado que inicie su incremento:

```
CREATE TABLE [dbo].[Centro](
    [Num_Centro] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [NombreC] [nvarchar](30) NOT NULL,
    [CiudadC] [nvarchar](30) NOT NULL,
);

SET IDENTITY_INSERT dbo.Centro on;
INSERT INTO dbo.Centro (Num_Centro, NombreC, CiudadC)
VALUES (2001, 'Tienda La Primavera', 'Culiacan');
SET IDENTITY_INSERT dbo.Centro off;
INSERT INTO dbo.Centro (NombreC, CiudadC)
VALUES ('Tienda Humaya', 'Culiacan');
INSERT INTO dbo.Centro (NombreC, CiudadC)
VALUES ('Tienda Plaza Sur', 'Culiacan');
INSERT INTO dbo.Centro (NombreC, CiudadC)
VALUES ('Tienda Ruby', 'Culiacan');
INSERT INTO dbo.Centro (NombreC, CiudadC)
VALUES ('Tienda Obregon', 'Culiacan');
```

	Num_Centro	NombreC	CiudadC
1	2001	Tienda La Primavera	Culiacan
2	2002	Tienda Humaya	Culiacan
3	2003	Tienda Plaza Sur	Culiacan
4	2004	Tienda Ruby	Culiacan
5	2005	Tienda Obregon	Culiacan

 Query executed successfully.

Aquí creado la tercera y última tabla de la base de datos que nos servirá como la unión entre las tablas de catálogo de puesto y centro con la generación de las llaves foráneas a relacionar las tablas, se utiliza de igual forma que en la tabla de catálogo de centros un atributo autoincrementar:

```
CREATE TABLE [dbo].[Persona](
[ID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY,
[Nombre] [nvarchar](15) NOT NULL,
[ApellidoP] [nvarchar](20) NOT NULL,
[ApellidoM] [nvarchar](20) NOT NULL,
[RFC] [nvarchar](20) NOT NULL,
[FechaN] [date] NOT NULL,
[ID_Puesto] [int] NOT NULL,
[Num_Centro] [int] NOT NULL,
);

ALTER TABLE [dbo].[Persona] WITH CHECK ADD CONSTRAINT [FK_Persona_Puesto] FOREIGN KEY([ID_Puesto])
REFERENCES [dbo].[Puesto] ([ID_Puesto])
GO

ALTER TABLE [dbo].[Persona] WITH CHECK ADD CONSTRAINT [FK_Persona_Centro] FOREIGN KEY([Num_Centro])
REFERENCES [dbo].[Centro] ([Num_Centro])
GO

SET IDENTITY_INSERT dbo.Persona ON;
INSERT INTO dbo.Persona(ID,Nombre,ApellidoP,ApellidoM,RFC,FechaN,ID_Puesto,Num_Centro)
VALUES (5001,'Juan','Castro','Castro','CACJ830604','1983-06-04',1,2001);
SET IDENTITY_INSERT dbo.Persona OFF;
INSERT INTO dbo.Persona(Nombre,ApellidoP,ApellidoM,RFC,FechaN,ID_Puesto,Num_Centro)
VALUES ('Julian','Meza','Perez','MEPJ830605','1983-06-05',2,2002);
INSERT INTO dbo.Persona(Nombre,ApellidoP,ApellidoM,RFC,FechaN,ID_Puesto,Num_Centro)
VALUES ('Carlos','Lopez','Garcia','LOGC830606','1983-06-06',3,2003);
INSERT INTO dbo.Persona(Nombre,ApellidoP,ApellidoM,RFC,FechaN,ID_Puesto,Num_Centro)
VALUES ('Manuel','Lagos','Rosa','LARM830607','1983-06-07',1,2004);
INSERT INTO dbo.Persona(Nombre,ApellidoP,ApellidoM,RFC,FechaN,ID_Puesto,Num_Centro)
VALUES ('Oscar','Rosa','Meza','ROMO830608','1983-06-08',3,2005);
```

	ID	Nombre	ApellidoP	ApellidoM	RFC	FechaN	ID_Puesto	Num_Centro
1	5001	Juan	Castro	Castro	CACJ830604	1983-06-04	1	2001
2	5002	Julian	Meza	Perez	MEPJ830605	1983-06-05	2	2002
3	5003	Carlos	Lopez	Garcia	LOGC830606	1983-06-06	3	2003
4	5004	Manuel	Lagos	Rosa	LARM830607	1983-06-07	1	2004
5	5005	Oscar	Rosa	Meza	ROMO830608	1983-06-08	3	2005

- Unión de tablas para generación de reporte.

En este punto realizamos la unión entre las bases de datos creado en la referencia de las llaves foráneas generadas en la última tabla de la base de datos persona que da como referencia a los empleados de la empresa UNI y en este punto mostramos la información para la generación del reporte solicitada en la actividad A1:

```
select p1.ID, p1.Nombre, p1.ApellidoP, p1.ApellidoM, p1.RFC, p1.FechaN, p2.Puesto, p2.Descripcion Puesto, p2.Tipo Puesto, p2.Prestacion, p2.Prestacion Puesto, c.Num_Centro, c.NombreC
from Persona p1 LEFT JOIN Puesto p2 ON p1.ID_Puesto=p2.ID_Puesto
LEFT JOIN Centro C ON p1.Num_Centro=c.Num_Centro
order by tipo_puesto desc
```

	ID	Nombre	ApellidoP	ApellidoM	RFC	FechaN	Puesto	Descripcion_Puesto	Tipo_Puesto	Prestacion	Prestacion_Puesto	Num_Centro	NombreC
1	5001	Juan	Castro	Castro	CACJ830604	1983-06-04	Gerente	Gerente Supervisor Depto. TDA	1	Si	Combustible	2001	Tienda La Primavera
2	5004	Manuel	Lagos	Rosa	LARM830607	1983-06-07	Gerente	Gerente Supervisor Depto. TDA	1	Si	Combustible	2004	Tienda Ruby
3	5005	Oscar	Rosa	Meza	ROMO830608	1983-06-08	Cajero	Punto de Vta. TDA	0	No	NULL	2005	Tienda Obregon
4	5002	Julian	Meza	Perez	MEPJ830605	1983-06-05	Vendedor	Vendedor Deptos. TDA	0	No	NULL	2002	Tienda Humaya
5	5003	Carlos	Lopez	Garcia	LOGC830606	1983-06-06	Cajero	Punto de Vta. TDA	0	No	NULL	2003	Tienda Plaza Sur

En este punto creamos el script del procedimiento que mandaremos a ejecutar como reporte de la información para la generación del reporte solicitada en la actividad A1 utilizando las dos líneas siguientes EXEC [dbo].[UNI\_A1] go:

```
CREATE PROCEDURE dbo.UNI_A1
AS
BEGIN
WITH rep AS( select p1.ID, p1.Nombre, p1.ApellidoP, p1.ApellidoM, p1.RFC, p1.FechaN, p2.Puesto, p2.Descripcion Puesto, p2.Tipo Puesto, p2.Prestacion, p2.Prestacion Puesto,
from Persona p1 LEFT JOIN Puesto p2 ON p1.ID_Puesto=p2.ID_Puesto
LEFT JOIN Centro C ON p1.Num_Centro=c.Num_Centro)

SELECT REPORT = 'DATABASE TABLES',
ServerName = @@SERVERNAME,
DatabaseName = DB_NAME(),
*
FROM rep
ORDER BY 4, 5
END

EXEC [dbo].[UNI_A1]
GO
```

	REPORT	ServerName	DatabaseName	ID	Nombre	ApellidoP	ApellidoM	RFC	FechaN	Puesto	Descripcion_Puesto	Tipo_Puesto	Prestacion	Prestacion_Puesto	Num_Centro	NombreC
1	DATABASE TABLES	231444L90188024\SQLEXPRESS	DB_LPII_A1	5001	Juan	Castro	Castro	CACJ830604	1983-06-04	Gerente	Gerente Supervisor Depto. TDA	1	Si	Combustible	2001	Tienda La Primavera
2	DATABASE TABLES	231444L90188024\SQLEXPRESS	DB_LPII_A1	5002	Julian	Meza	Perez	MEPJ830605	1983-06-05	Vendedor	Vendedor Deptos. TDA	0	No	NULL	2002	Tienda Humaya
3	DATABASE TABLES	231444L90188024\SQLEXPRESS	DB_LPII_A1	5003	Carlos	Lopez	Garcia	LOGC830606	1983-06-06	Cajero	Punto de Vta. TDA	0	No	NULL	2003	Tienda Plaza Sur
4	DATABASE TABLES	231444L90188024\SQLEXPRESS	DB_LPII_A1	5004	Manuel	Lagos	Rosa	LARM830607	1983-06-07	Gerente	Gerente Supervisor Depto. TDA	1	Si	Combustible	2004	Tienda Ruby
5	DATABASE TABLES	231444L90188024\SQLEXPRESS	DB_LPII_A1	5005	Oscar	Rosa	Meza	ROMO830608	1983-06-08	Cajero	Punto de Vta. TDA	0	No	NULL	2005	Tienda Obregon



## Conclusión

En conclusión: Las bases de datos en las empresas se han convertido en un elemento imprescindible, siendo consideradas por algunos como una de las mayores aportaciones de la informática al mundo laboral. Las bases de datos son programas capaces de almacenar una gran cantidad de información relacionada entre sí y de una manera ordenada, lo que para las empresas es algo extraordinario, ya que les ahorra muchísimo tiempo y dinero. Esta facilidad de manejo de datos junto a las posibilidades de negocio que abren, hace que las bases de datos en las empresas sean realmente importantes y valiosas, hasta el punto en que los datos se han convertido hoy en día en una auténtica mina de oro. Tan solo hay que ver lo lejos que están llegando las redes sociales, que son expertas en recoger datos para ponerlos a disposición de sus anunciantes, un modelo de negocio basado en las bases de datos.

## Referencias

GitHub: Let's build from here. (n.d.)

Wikipedia contributors. (n.d.). *Modelo entidad-relación*. Wikipedia, The Free Encyclopedia.

[https://es.wikipedia.org/w/index.php?title=Modelo\\_entidad-relaci%C3%B3n&oldid=155006888](https://es.wikipedia.org/w/index.php?title=Modelo_entidad-relaci%C3%B3n&oldid=155006888)

*¿Por qué son tan importantes las bases de datos en las empresas?* (n.d.). Entel Comunidad Empresas. Retrieved November 24, 2023, from <https://ce.entel.cl/articulos/base-de-datos-de-empresas/>