

**TRABAJO FINAL: SISTEMA DE CONTROL DE PROYECTOS**

Integrantes del Grupo Nro. 3

* Rene Lanza Chevarria
* Nuryma Contreras Medina
* Paul Alvarado Bustos
* Nieves Coasaca Gayona

ÍNDICE

1. MODELO CONCEPTUAL
2. MODELO LÓGICO
3. MODELO FÍSICO
4. EXPORTACIÓN DEL MODELO A SQL SERVER
5. IMPLEMENTACIÓN DE LA BASE DE DATOS
6. CONSULTAS ( QUERY )

**SISTEMA DE CONTROL DE PROYECTOS**

Se plantea como requerimiento **Implementar un Sistema de Control de Proyectos que permitirá controlar el uso de recursos de un proyecto (Costo del uso), en base a un presupuesto.**

Un proyecto se define como un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. Un proyecto tiene un inicio y un fin claramente definidos y se lleva a cabo para alcanzar un objetivo específico dentro de un conjunto de restricciones, que pueden incluir limitaciones de tiempo, recursos y presupuesto.

**El Presupuesto del Proyecto estará constituido por tablas que definen una estructura de costos ( Componente >> Entregable >> Partida ) escalonada es decir; un conjunto de trabajos (partidas) forman un Entregable, a su vez; un grupo de trabajos (partidas) forman un Entregable uniendo todos los entregables del proyecto formamos un Componente.**

Los recursos utilizados en una unidad de construcción (partida), se clasifican en Mano de obra, materiales y equipos utilizados para realizar esta “partida”.

Para efectuar el control de recursos, se establecen formatos físicos, para cada grupo de los recursos ( Mano de Obra, Materiales y Equipos), estos formatos físicos (impresos); conforman las entidades en la base de datos formando tablas que se deben relacionar en una tabla única llamada “Partida”.

Al obtener el costo de una partida, podemos actualizar los costos de los entregables y componentes en cascada.

**DEFINICIONES:**

En el contexto del Project Management Institute (PMI), **un componente se refiere a un elemento individual que forma parte de un proyecto**. En el caso de un proyecto de construcción, un componente podría ser una estructura específica, como una pared, un techo o una escalera.

**Un entregable es un resultado tangible que se espera obtener en un proyecto**. En el caso de un proyecto de construcción, un entregable podría ser la finalización de una fase del proyecto, la entrega de un conjunto de planos arquitectónicos o la construcción completa de una estructura.

**Una partida se refiere a un elemento específico del presupuesto de un proyecto**. En un proyecto de construcción, una partida podría ser el costo de los materiales, mano de obra y equipos necesarios para efectuar la construcción de una estructura.

Es importante tener en cuenta que la definición exacta de estos términos puede variar según el contexto y el enfoque del PMI, pero en general, se utilizan para desglosar los elementos del proyecto y el presupuesto en partes más manejables y comprensibles.

**PROYECTO**

**PRESUPUESTO**

**MATERIALES**

**EQUIPOS**

**MANO DE OBRA**

**COMPONENTE**

**ENTREGABLE**

**PRESUPUESTO**

**PARTIDA=TRABAJO**

**MODELO CONCEPTUAL**

Un modelo conceptual en SQL es una representación visual y estructurada de las entidades y relaciones de una base de datos. Este modelo describe cómo se organiza la información en la base de datos y cómo se relacionan entre sí los diferentes conjuntos de datos.

ENTIDADES

Entidades Fuertes: No dependen de otras entidades

CLIENTE

UBIGEO

PROYECTO ( Parcialmente dependiente )

EQUIPO

PERSONAL

MATERIALES

Entidades Débiles: Dependen de otras entidades

COMPONENTE

ENTREGABLE

PARTIDA

Entidades de Relación o asociativas: Sirven para relacionar entidades.

PARTE EQUIPOS

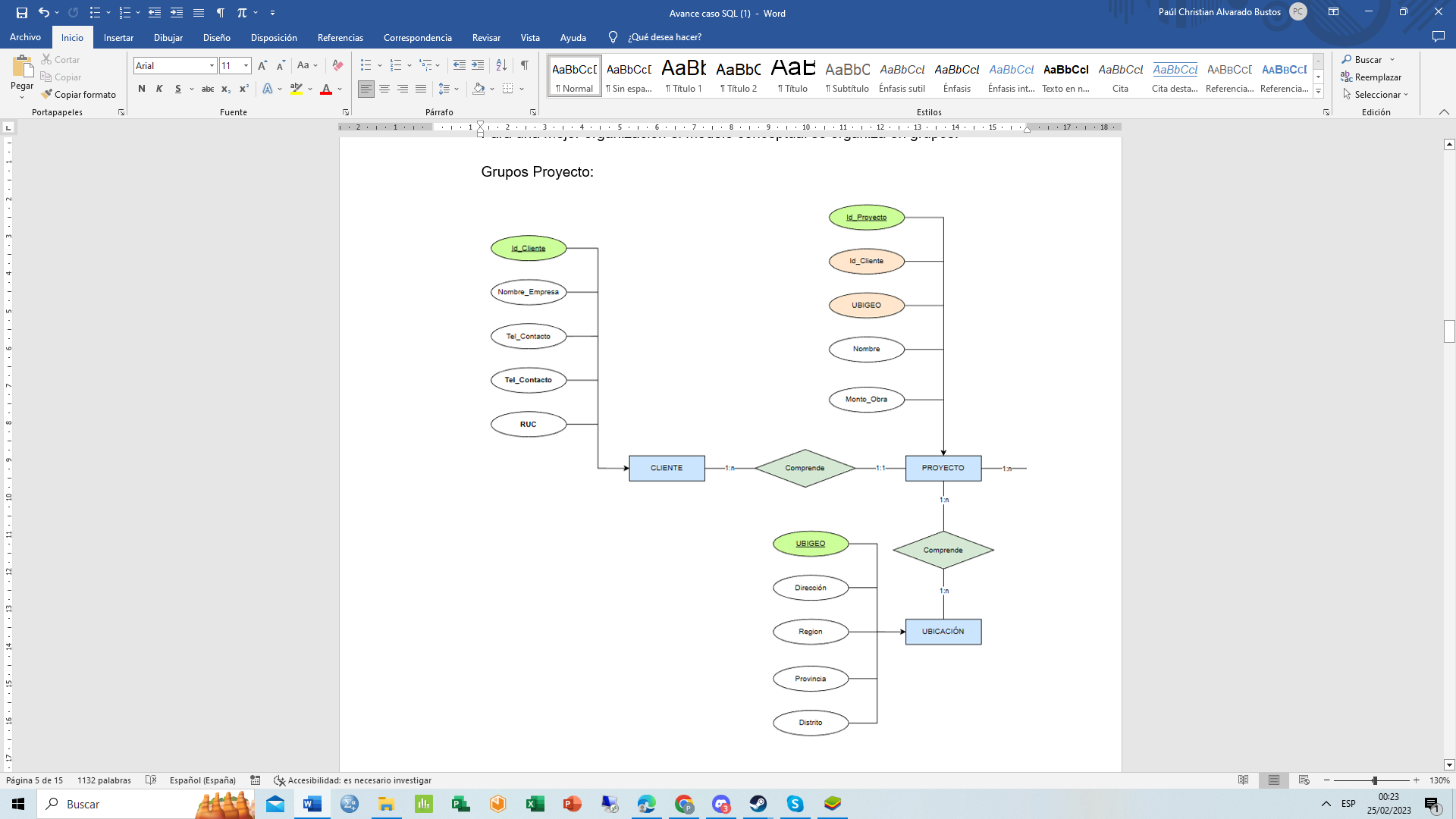
TAREOS

NOTA DE ALMACÉN

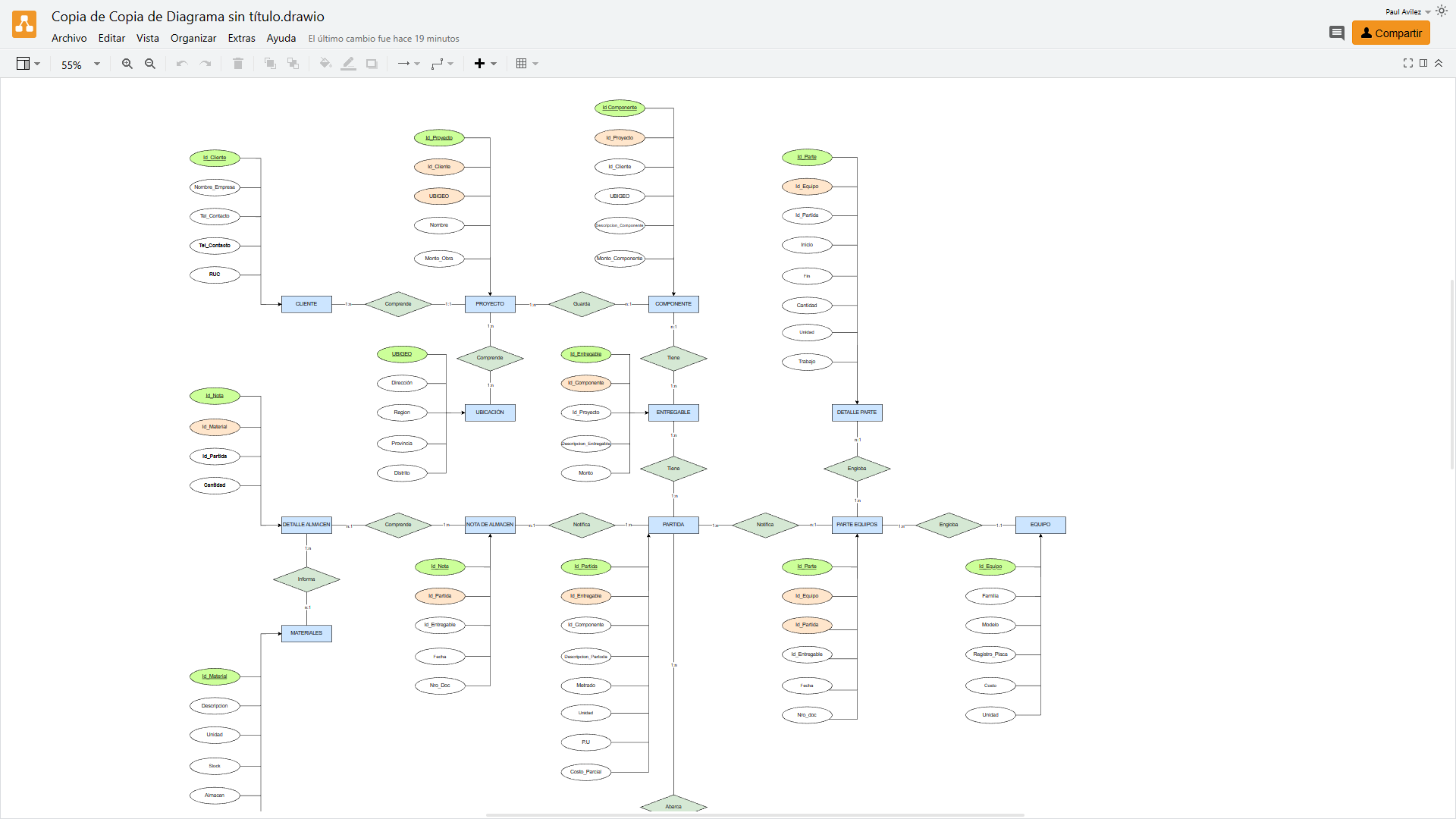
otras entidades asociativas.

Para una mejor organización el modelo conceptual se organiza en grupos:

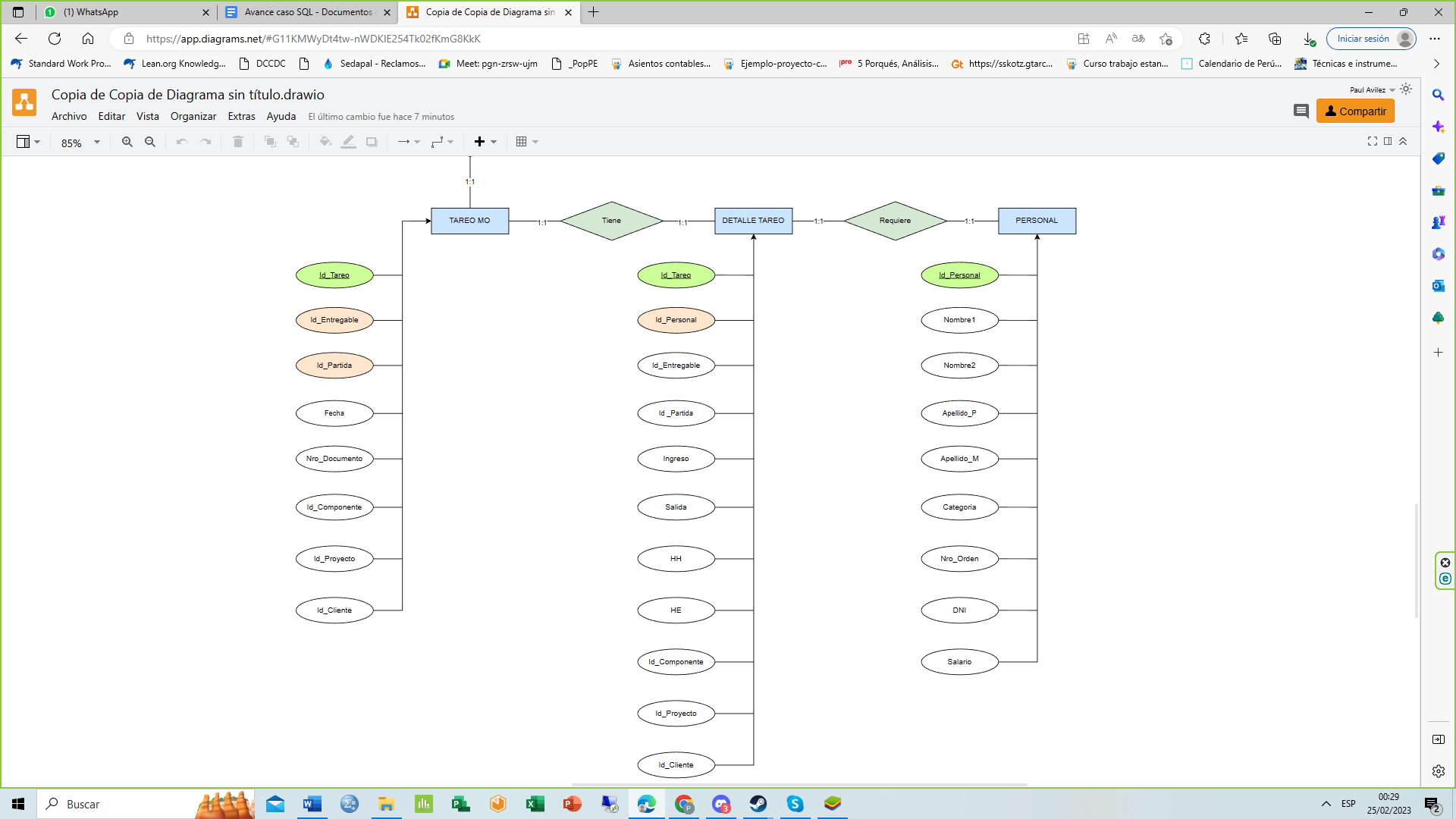
**Grupos Proyecto:**



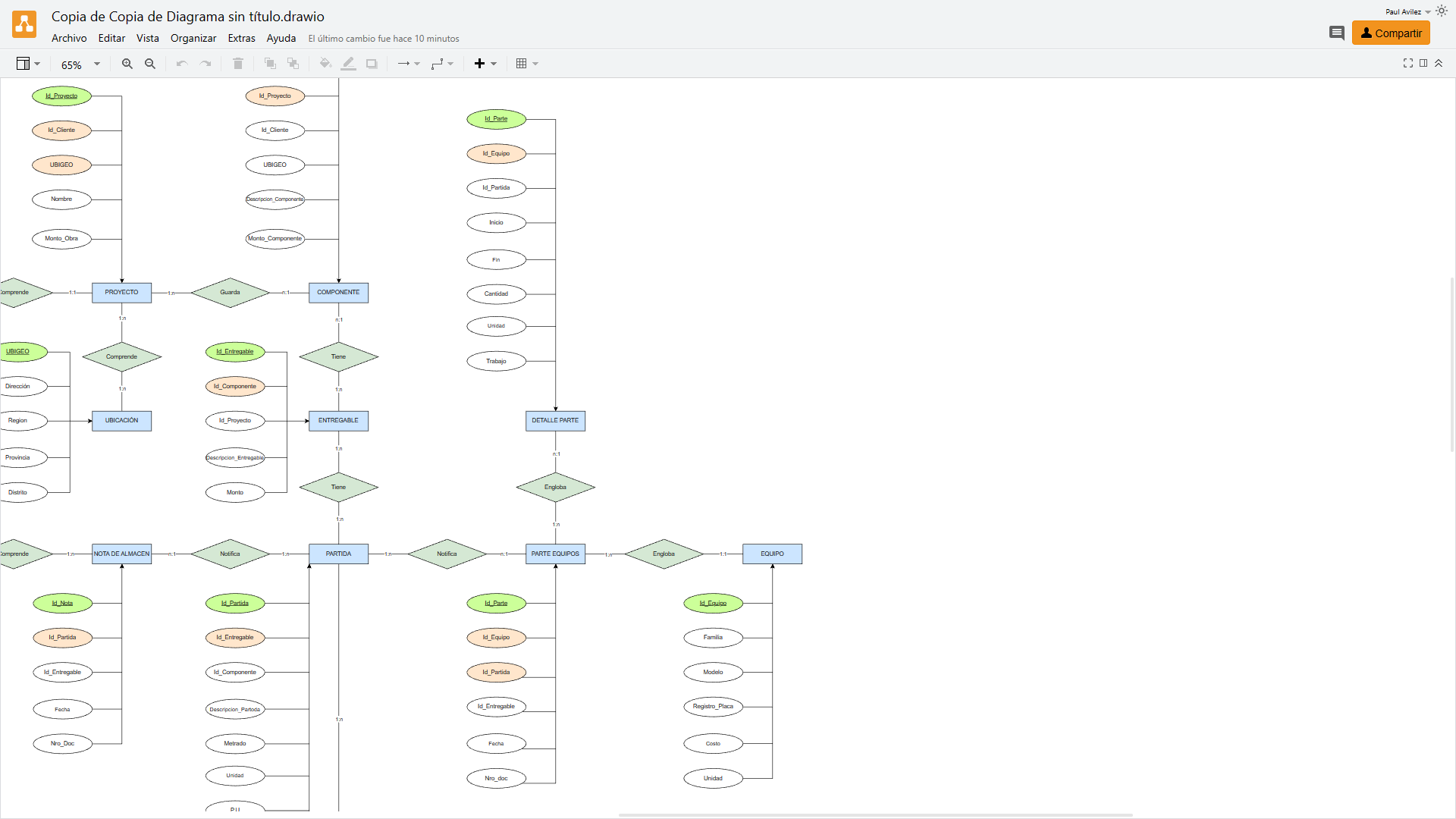
**Grupo Presupuesto**



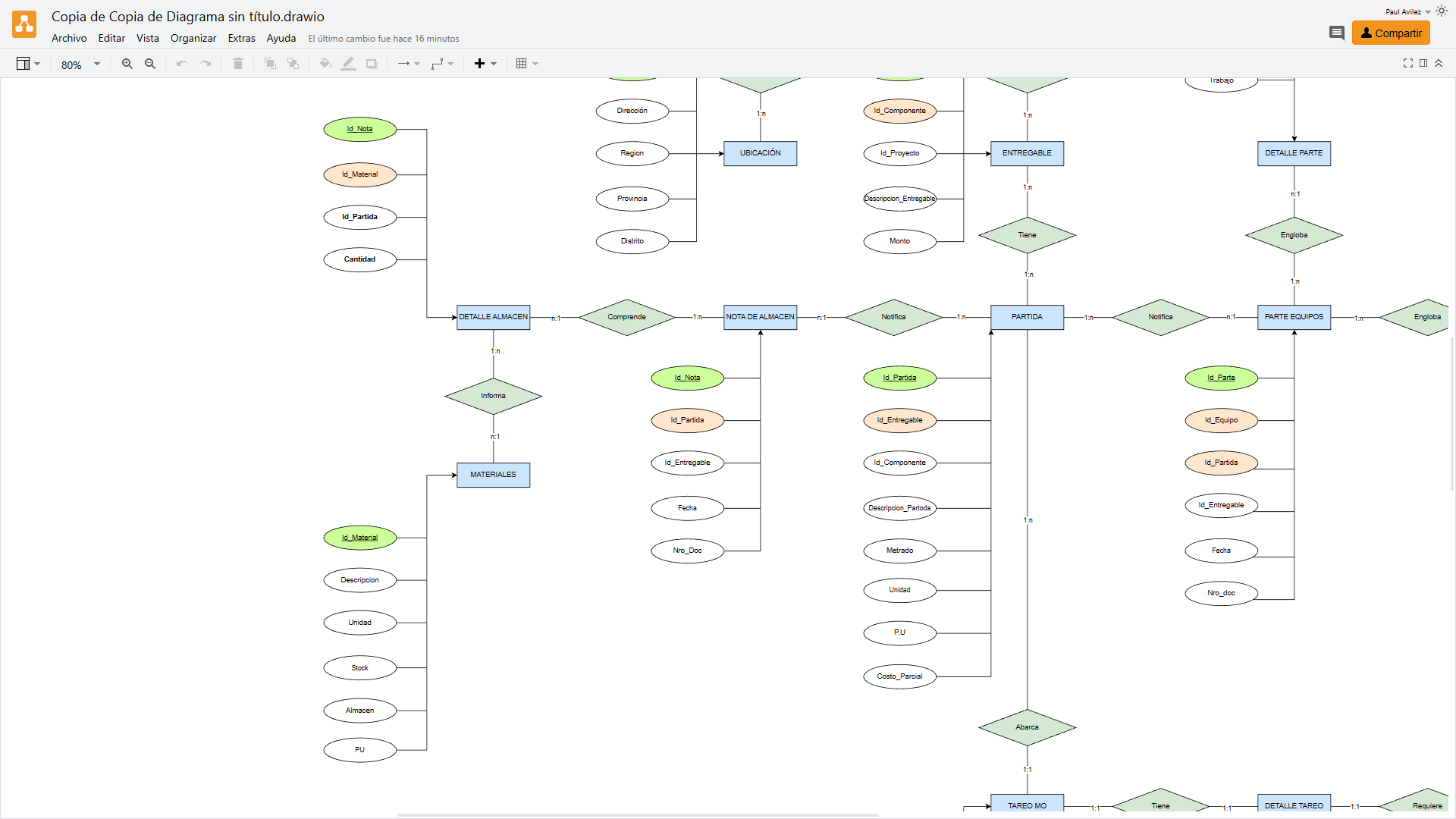
**Grupo Mano de Obra**



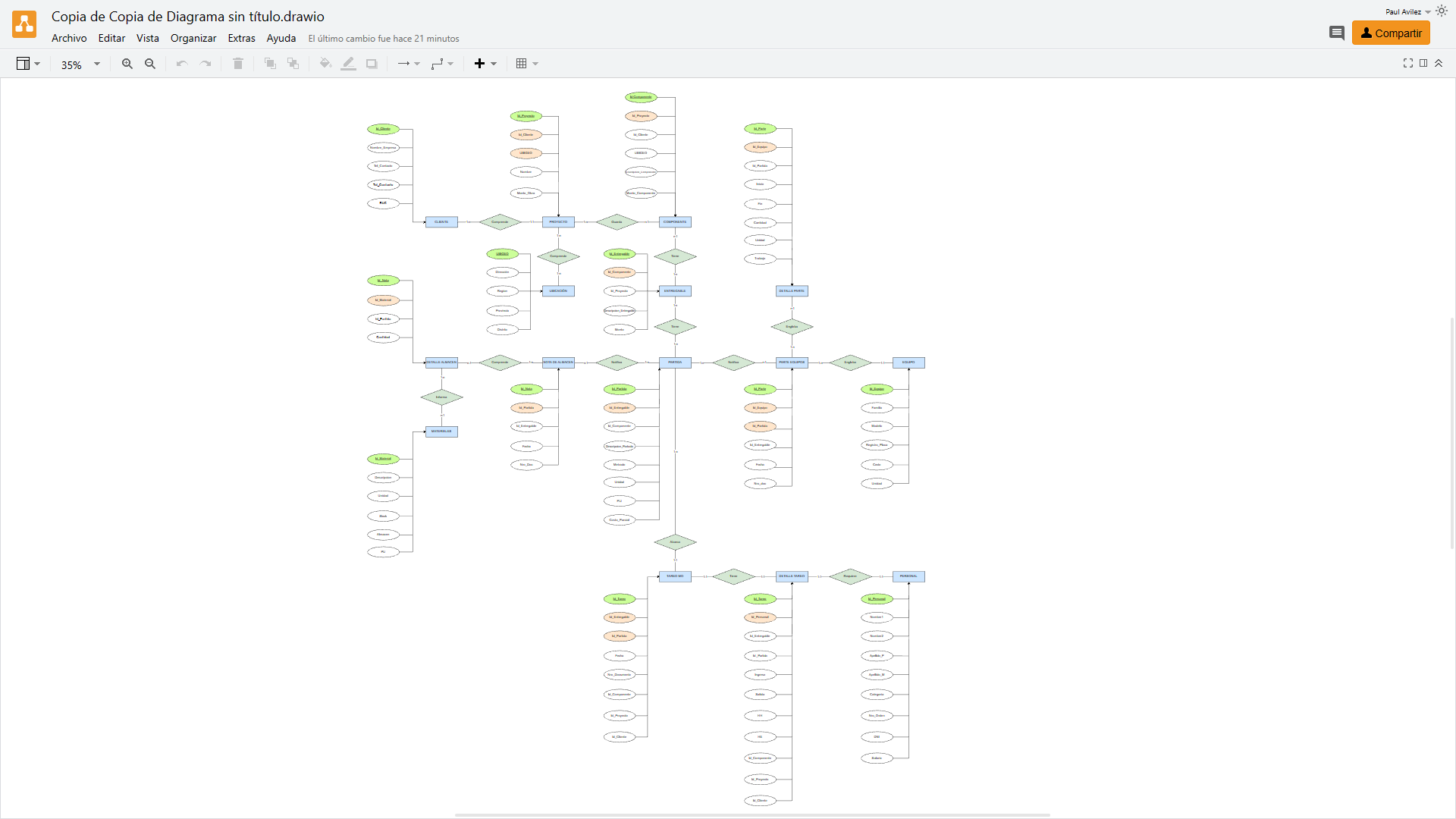
**Grupo Equipos**



**Grupo Materiales**



Una visión integral del sistema se muestra en el siguiente grafico

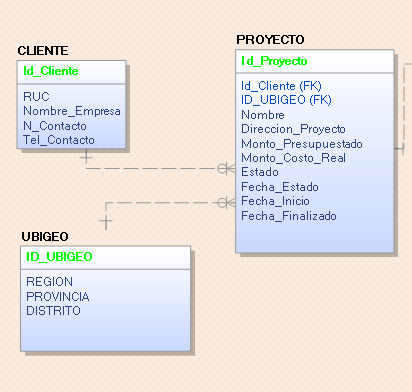


**MODELO LÓGICO**

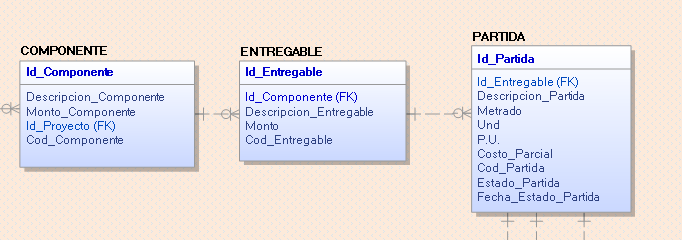
El modelo lógico de datos es un nivel intermedio entre el modelo conceptual y el modelo físico de datos. Mientras que el modelo conceptual se enfoca en la estructura de alto nivel de los datos y el modelo físico se enfoca en cómo se almacenan físicamente, el modelo lógico se enfoca en la estructura y las relaciones de los datos en un nivel más detallado y preciso que el modelo conceptual, pero sin llegar a la implementación física del modelo.

Al Igual que en el modelo anterior, representamos la estructura por grupos, para una mejor comprensión.

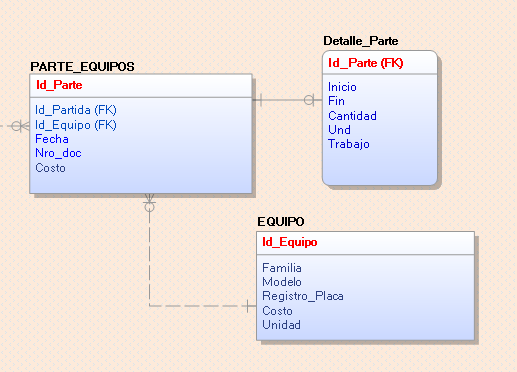
**Grupo proyecto**



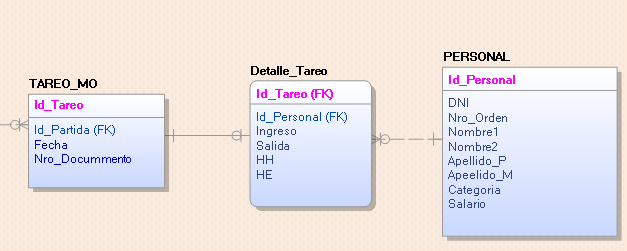
**Grupo Presupuesto**

****

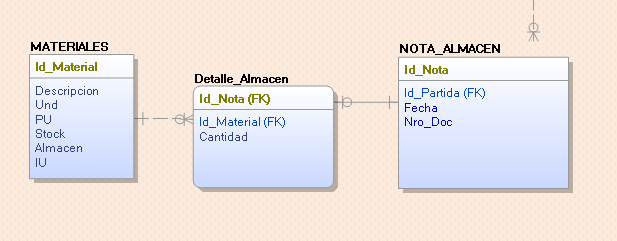
**Grupo Equipos**

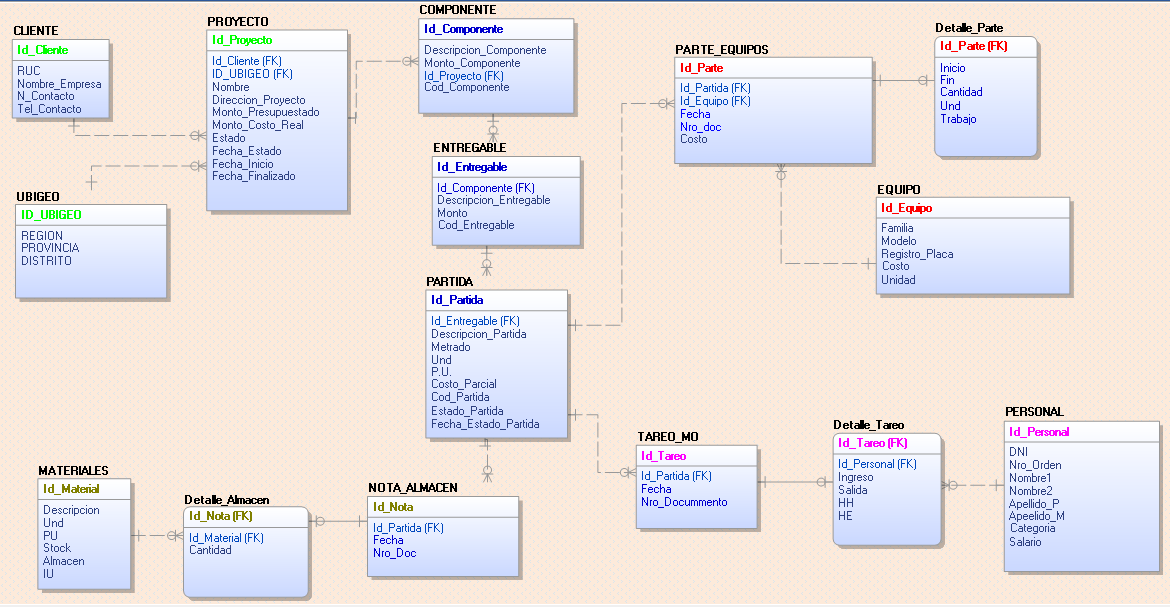
****

**Grupo Mano De Obra**

****

**Grupo Materiales**

****





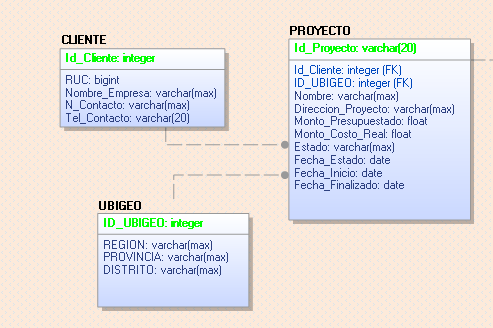
**MODELO FÍSICO:**

El modelo físico es el nivel más bajo de abstracción en la construcción de una base de datos y describe cómo los datos se almacenan en el sistema de gestión de base de datos (DBMS) específico. Por lo tanto, el modelo físico es específico para cada DBMS y puede variar dependiendo del sistema utilizado.

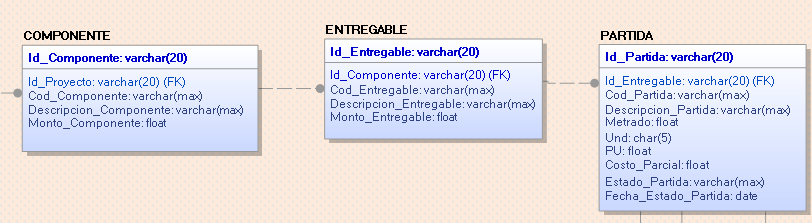
El modelo físico de datos se utiliza para diseñar la base de datos en un nivel de detalle muy alto, incluyendo cómo se almacenarán los datos y cómo se accederán a ellos. Este modelo es necesario para implementar la base de datos en el sistema de gestión de base de datos y para realizar ajustes de rendimiento en el sistema.

El modelo se desarrolla en grupos para mejor comprensión.

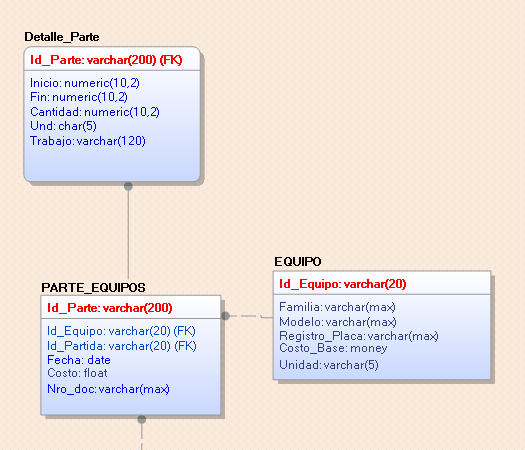
**Grupo proyecto**



**Grupo Presupuesto**



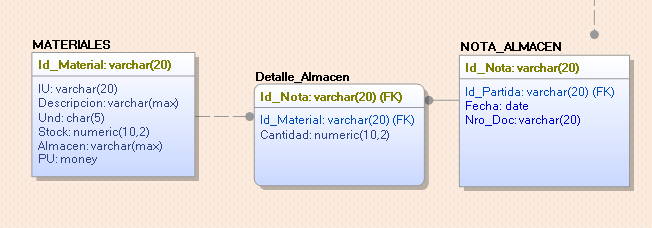
**Grupo Equipos**

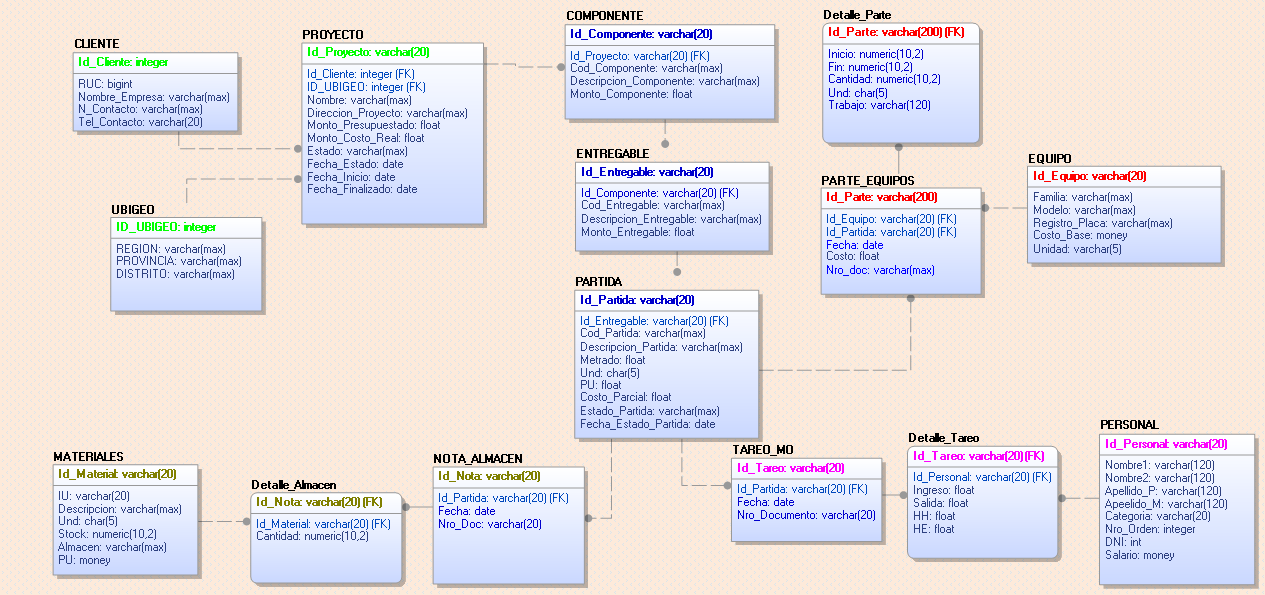


**Grupo Mano De Obra**



**Grupo Materiales**

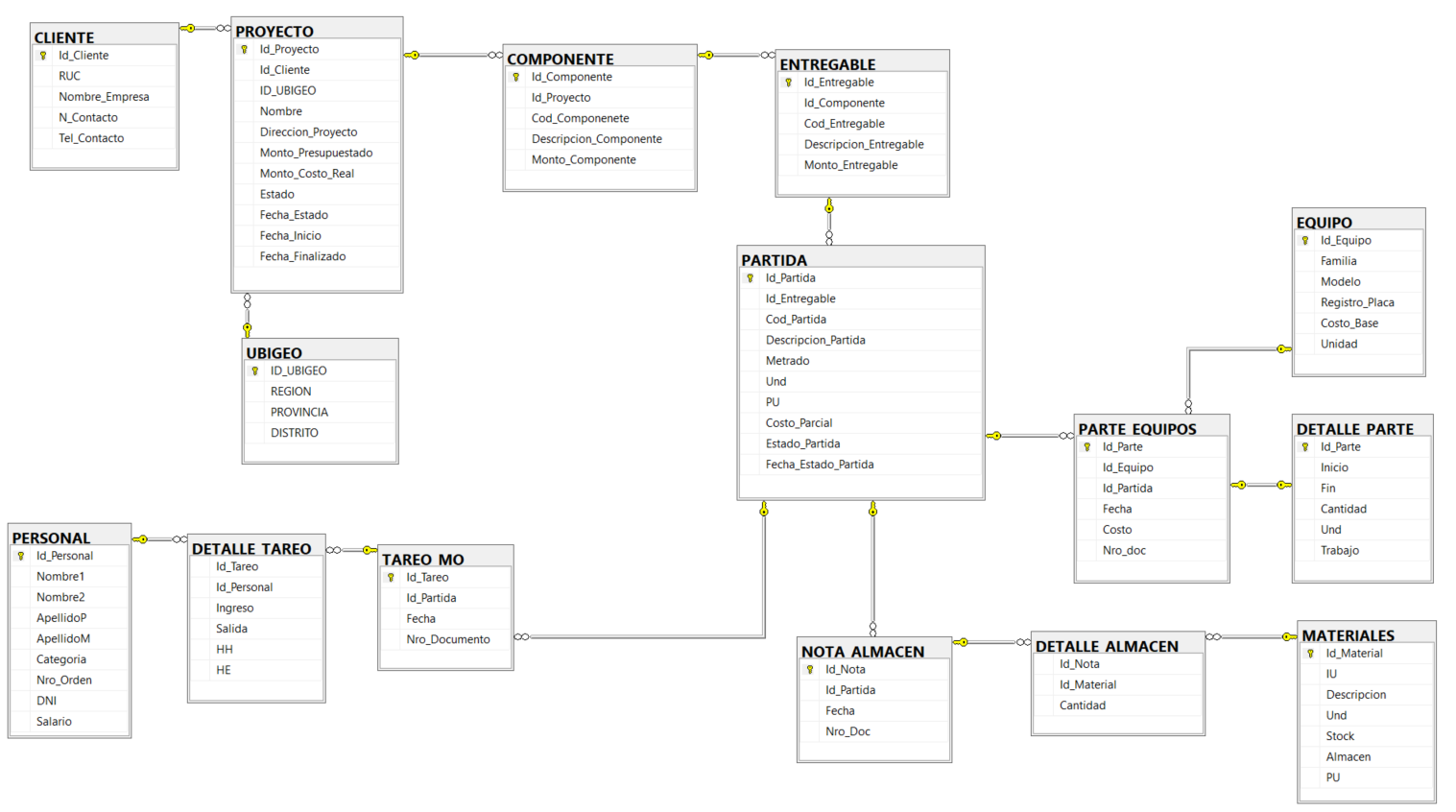






**MODELO E/R SQL SERVER**

El gráfico muestra el modelo exportado al SQL Server.

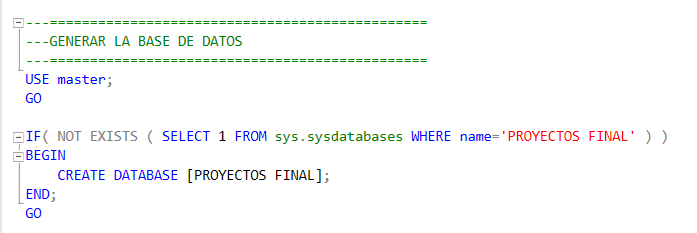


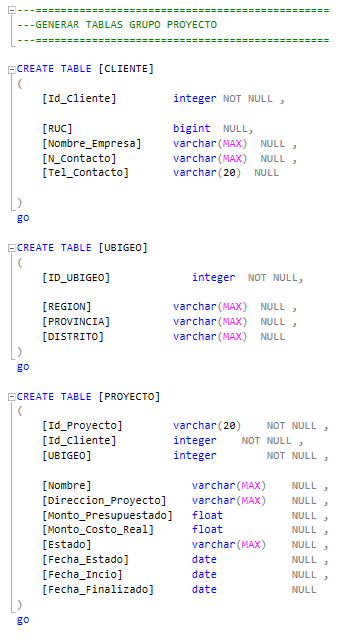
**IMPLEMENTACIÓN DE LA BASE DE DATOS**

La implementación de una base de datos es un proceso que involucra varias etapas, desde el diseño de la base de datos hasta la puesta en producción y el mantenimiento.

Presentamos los más destacados:

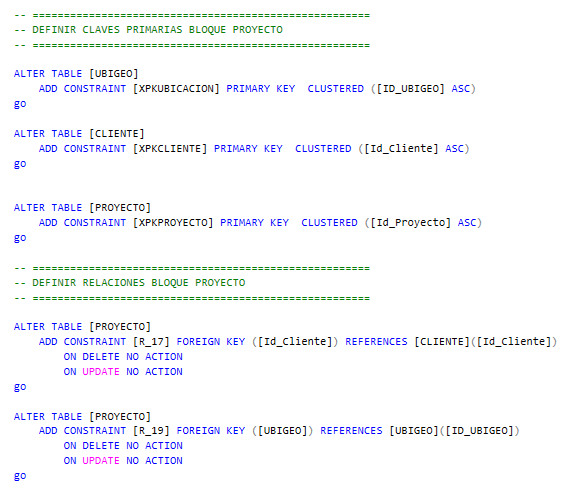
1. Creación de la estructura de la base de datos: en esta etapa se crean las tablas, campos y relaciones definidos en el diseño de la base de datos.





La creación del resto de tablas se muestra en el archivo .

Para cada tabla se creó su llave primaria ay sus llaves foráneas, por ejemplo para el grupo de tablas asociadas al proyecto:



El resto de la creación de las llaves primarias y foráneas se muestras en el siguiente archivo .

1. Carga de datos: se deben cargar los datos iniciales en la base de datos, ya sea a través de la importación de datos desde otro sistema o a través de la introducción manual de los datos.

Como implementación principal se generó la carga de datos al modelo mediante el asistente de importación desde Excel.

Previamente se ha definido las tablas y datos que se requieren para implementar la base de datos

1. Pruebas y ajustes: se deben realizar pruebas para asegurarse de que la base de datos está funcionando correctamente y ajustarla según sea necesario para optimizar el rendimiento y la eficiencia.

Se realizaron pruebas de consistencia mediante consultas query, como por ejemplo:

---=================================================================== ---ACTIVAR LA BASE DE DATOS

---===================================================================

USE [PROYECTOS FINAL]

GO

---===================================================================

--- Lista de proyectos: Cliente, Proyecto, Region, Provincia, Distrito, Estado del proyecto

---===================================================================

SELECT

CLI.Nombre\_Empresa AS CLIENTE,

Nombre AS [NOMBRE PROYECTO],

PRO.Estado AS [ESTADO DEL PROYECTO],

UBI.REGION,

UBI.PROVINCIA,

UBI.DISTRITO

FROM

PROYECTO AS PRO

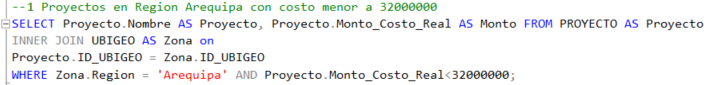
INNER JOIN CLIENTE AS CLI ON CLI.Id\_Cliente=PRO.Id\_Cliente

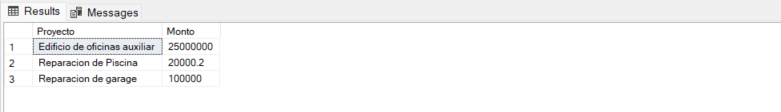
INNER JOIN UBIGEO AS UBI ON UBI.ID\_UBIGEO=PRO.UBIGEO

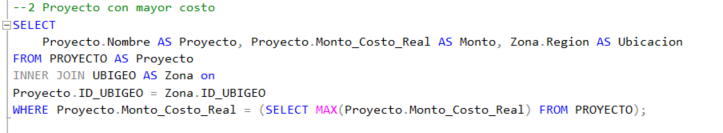
GO

**CONSULTAS SQL (QUERY)**

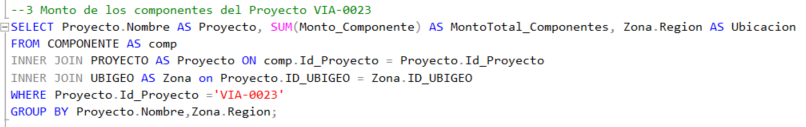
* Consultas SQL Nuryma Contreras:













* Consultas SQL Rene Lanza:

---====================================================================

--- CONSULTAR LOS PARTES DIARIOS DE USO DE EQUIPOS

---====================================================================

SELECT

PAR.Descripcion\_Partida AS PARTIDA,

EQP.Familia AS EQUIPO,

EQP.Registro\_Placa AS REGISTRO,

SUM(DET.Cantidad) AS CANTIDAD,

DET.Und AS UNIDAD,

PARTE.Costo AS PU,

SUM(DET.Cantidad\*PARTE.Costo) AS COSTO

FROM EQUIPO AS EQP

INNER JOIN PARTE\_EQUIPOS AS PARTE ON EQP.Id\_Equipo=PARTE.Id\_Equipo

INNER JOIN Detalle\_Parte AS DET ON PARTE.Id\_Parte=DET.Id\_Parte

INNER JOIN PARTIDA AS PAR ON PARTE.Id\_Partida=PAR.Id\_Partida

GROUP BY

PAR.Descripcion\_Partida,

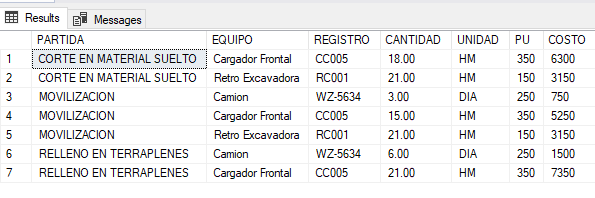
EQP.Familia,

EQP.Registro\_Placa,

DET.Und,

PARTE.Costo

GO



---====================================================================

--- CONSULTAR MATERIALES UTILIZADOS POR PARTIDA

---====================================================================

SELECT

NOTA.Id\_Partida,

PAR.Descripcion\_Partida,

ALM.Id\_Material,

MAT.Descripcion,

MAT.PU,

SUM(ALM.Cantidad) AS CANTIDAD,

SUM(MAT.PU\*ALM.Cantidad) AS COSTO

FROM NOTA\_ALMACEN AS NOTA

INNER JOIN Detalle\_Almacen AS ALM ON NOTA.Id\_Nota=ALM.Id\_Nota

INNER JOIN MATERIALES AS MAT ON ALM.Id\_Material=MAT.Id\_Material

INNER JOIN PARTIDA AS PAR ON PAR.Id\_Partida=NOTA.Id\_Partida

GROUP BY

ALM.Id\_Material,

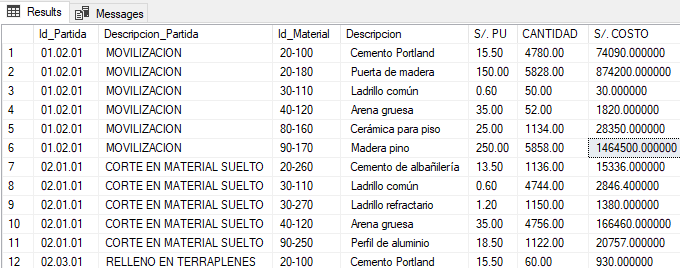
MAT.Descripcion,

NOTA.Id\_Partida,

PAR.Descripcion\_Partida,

MAT.PU;

GO



---====================================================================

--- Lista de proyectos: Cliente, Proyecto, Región, Provincia, Distrito,

Estado del proyecto

---====================================================================

SELECT

CLI.Nombre\_Empresa AS CLIENTE,

Nombre AS [NOMBRE PROYECTO],

PRO.Estado AS [ESTADO DEL PROYECTO],

UBI.REGION,

UBI.PROVINCIA,

UBI.DISTRITO

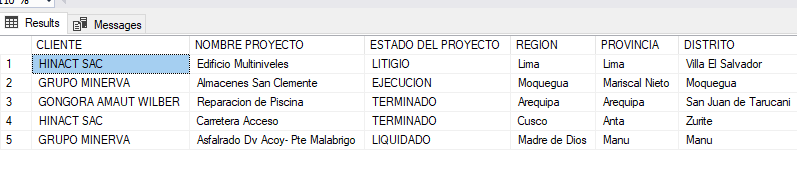
FROM

PROYECTO AS PRO

INNER JOIN CLIENTE AS CLI ON CLI.Id\_Cliente=PRO.Id\_Cliente

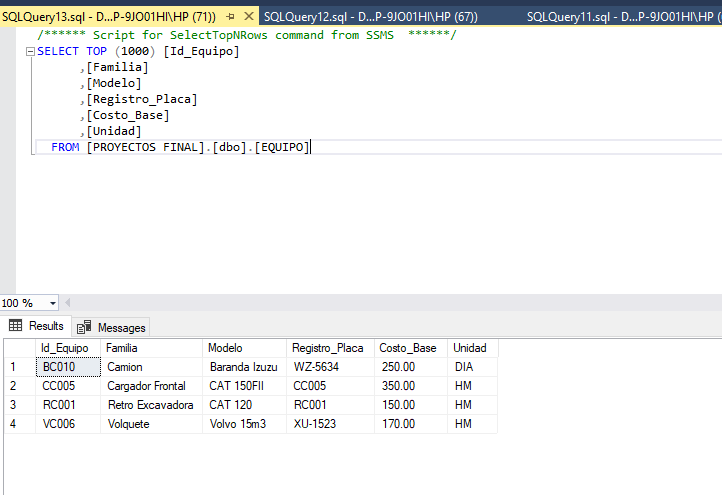
INNER JOIN UBIGEO AS UBI ON UBI.ID\_UBIGEO=PRO.UBIGEO

GO

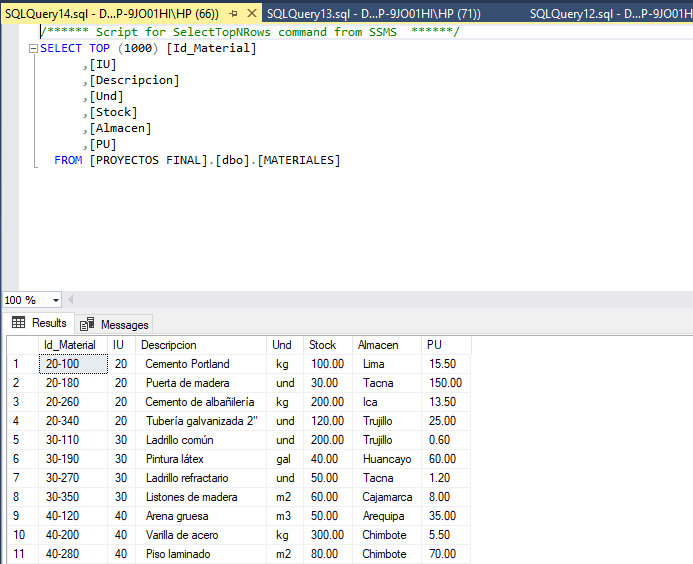


* Consultas SQL Nieves Coasaca:

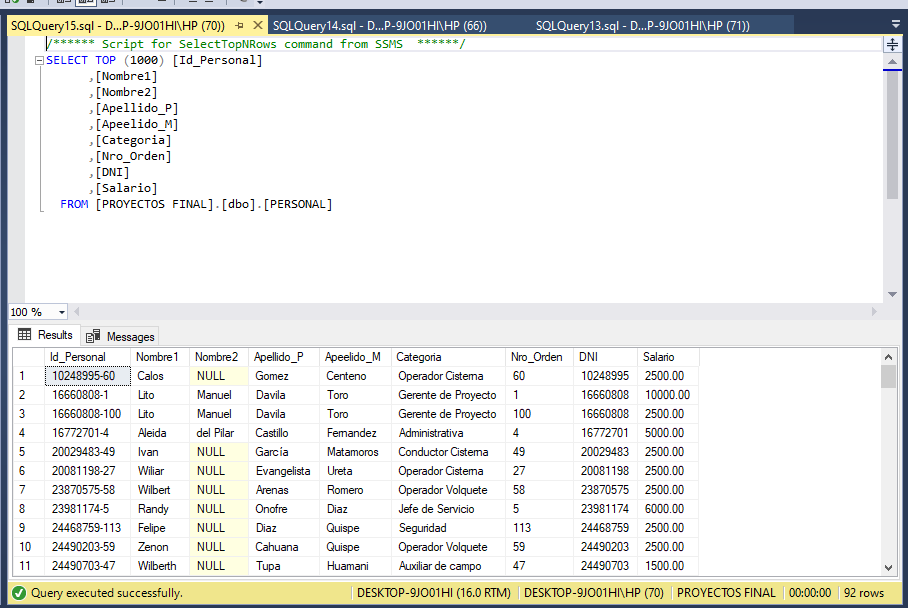
Consulta Equipos



Consulta Materiales



Consulta Personal



PAUL ALVARADO BUSTOS

---====================================================================

--- CONSULTAR EL PRESUPUESTO DE UN PROYECTO

---====================================================================

SELECT

PRO.Nombre AS PROYECTO,

COMP.Cod\_Componente AS CODIGO,

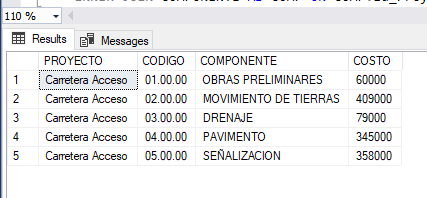
COMP.Descripcion\_Componente AS COMPONENTE,

COMP.Monto\_Componente AS COSTO

FROM PROYECTO AS PRO

INNER JOIN COMPONENTE AS COMP ON COMP.Id\_Proyecto=PRO.Id\_Proyecto

GO



---====================================================================

--- CONSULTAR PERSONAL POR PARTIDA

---====================================================================

SELECT

PAR.Descripcion\_Partida AS PARTIDA,

MO.Nro\_Docummento AS PLANILLA,

PER.Categoria AS Categoria,

SUM(-TAR.HH) AS HORAS,

ROUND(PER.Salario/30/8,2) AS [COSTO HORA],

ROUND(SUM(-TAR.HH\*PER.Salario/30/8),2) AS COSTO

FROM TAREO\_MO AS MO

INNER JOIN PARTIDA AS PAR ON MO.Id\_Partida=PAR.Id\_Partida

INNER JOIN Detalle\_Tareo AS TAR ON TAR.Id\_Tareo=MO.Id\_Tareo

INNER JOIN PERSONAL AS PER ON PER.Id\_Personal=TAR.Id\_Personal

GROUP BY

PER.Categoria ,

PAR.Descripcion\_Partida ,

MO.Nro\_Docummento ,

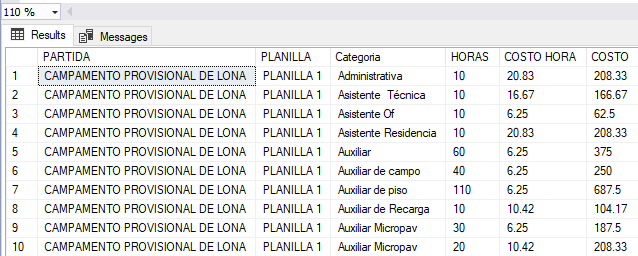
--TAR.Id\_Personal ,

PER.Salario

ORDER BY

PAR.Descripcion\_Partida

GO



---====================================================================

--- CONSULTAR EL COSTO TOTAL UTILIZADO DE LOS EQUIPOS POR PARTIDAS

---====================================================================

SELECT

PAR.Descripcion\_Partida AS PARTIDA,

SUM(DET.Cantidad\*PARTE.Costo) AS COSTO

FROM EQUIPO AS EQP

INNER JOIN PARTE\_EQUIPOS AS PARTE ON EQP.Id\_Equipo=PARTE.Id\_Equipo

INNER JOIN Detalle\_Parte AS DET ON PARTE.Id\_Parte=DET.Id\_Parte

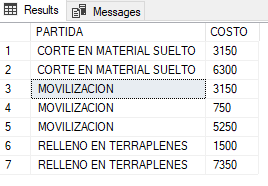
INNER JOIN PARTIDA AS PAR ON PARTE.Id\_Partida=PAR.Id\_Partida

GROUP BY

PAR.Descripcion\_Partida,

PARTE.Costo

GO



Fuentes:

Repo Github:

<https://github.com/R3natoky/SQL>