



Modelos

Entidad-Relación y Relacional

El modelo Relacional (Parte I)

***Elaborar un modelo de datos
de acuerdo a los estándares
de modelamiento para
resolver un problema
de baja complejidad***

- Unidad 1:
Bases de datos relacionales
- Unidad 2:
Manipulación de datos y
transaccionalidad en las
operaciones
- Unidad 3:
Definición de tablas
- Unidad 4:
Modelos Entidad-Relación y
Relacional



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Transforma un modelo conceptual a modelo lógico a partir de entidades y atributos.*

¿Por qué es importante
modelar?



¿A qué se refiere el
principio de abstracción
en el modelado de datos?



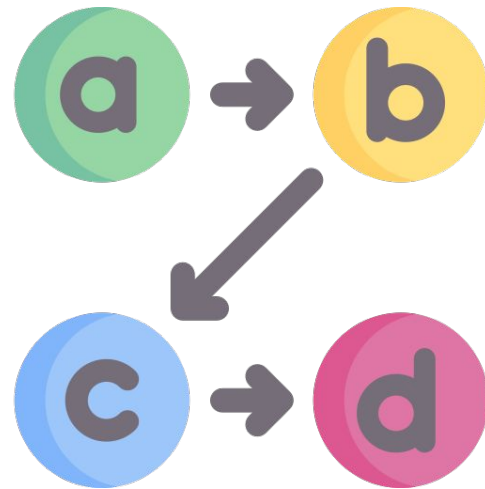
¿Es más eficiente
resolver un problema en
su conjunto o separarlo
en micro problemas?



**/* El modelo relacional y sus diferencias
con el modelo conceptual */**

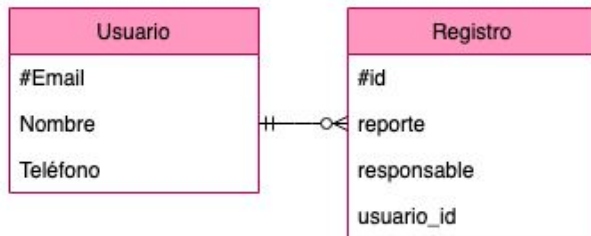
Modelo lógico

- Es el siguiente paso para la representación de las entidades y atributos que se definan durante el proceso de modelado conceptual.
- Define entidades transaccionales y operativas.
- Permite tener una visión más cercana al resultado final de la estructura lógica de una base de datos.
- Entrega una representación gráfica de cómo fluyen los datos en un sistema.

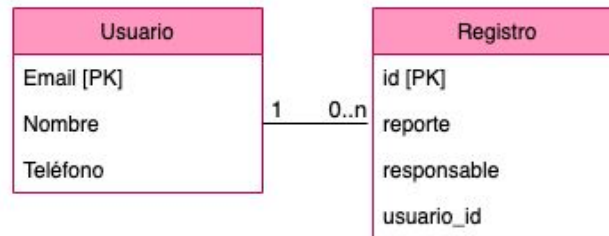


Diagramas de modelo lógico

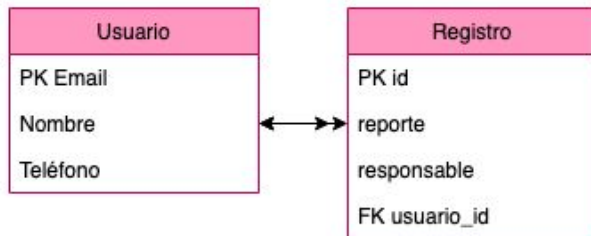
Martin /Crowfoot



UML



Bachman



Diagramas de modelo lógico

- La diferencia entre los tipos de notaciones principalmente cambian la forma de representar la cardinalidad y como anotar la clave primaria, foránea y atributos obligatorios.
- Utilizaremos la notación de UML.
- En esta, la cardinalidad de la relación se anota como mínimo..máximo, un único número indica que la cantidad es obligatoria.
- El nombre de la relación va indicado sobre la línea para indicar la relación de izquierda a derecha y bajo la línea para indicarla de derecha a izquierda.

Ejercicio guiado

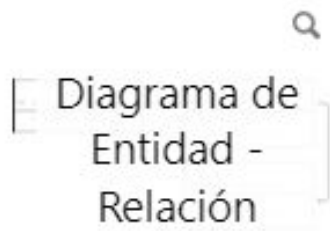
"Creando un diagrama UML"



Creando un diagrama UML

Contexto

A continuación, realizaremos un ejercicio de representación lógica de un modelo de datos. Para realizarlo, utilizaremos la herramienta de diagramas vista en la sesión anterior, deberás utilizar como plantilla para agilizar el proceso la opción que dice “Diagrama entidad-relación”.



Creando un diagrama UML







- **Paso 1:** Utilizando el modelo conceptual de la sesión anterior, implementemos ahora el modelo lógico. Iniciemos entonces con los clientes.

Cliente
Cliente_ID (PK)
Nombre
Apellido
Dirección
DNI

Compra
ID
Cliente_ID (FK)
Producto_ID (FK)

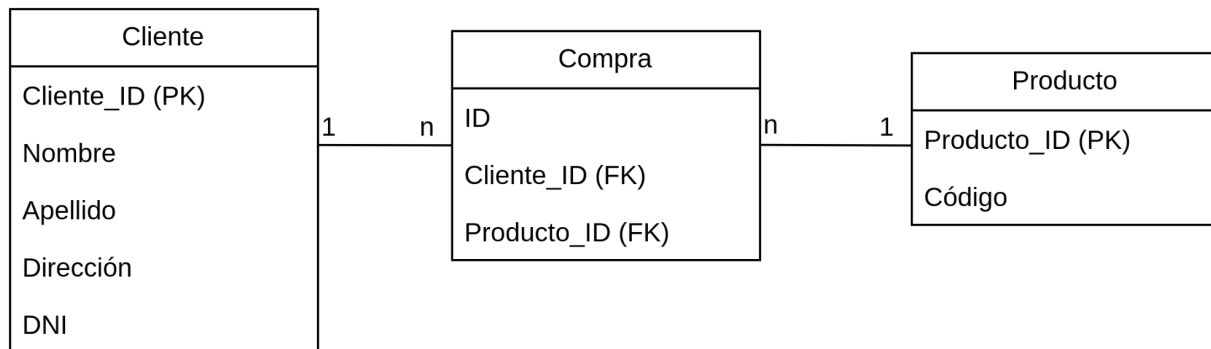
Producto
Producto_ID (PK)
Código

Repaso de tipos de relaciones

	Uno
	Muchos
	Uno y solo uno
	Cero o uno
	Uno o muchos
	Cero o muchos

Creando un diagrama UML

- **Paso 2:** Como sabemos, la relación entre Cliente y Producto es N:N. Apliquemos entonces los conectores.



/* Reglas de transformación */

Reglas básicas de transformación

Modelo conceptual a modelo lógico

Como vimos en la sesión anterior, un modelo conceptual representa en rectángulos las entidades y óvalos los atributos. Cuando pasamos del modelo conceptual a lógico debemos:

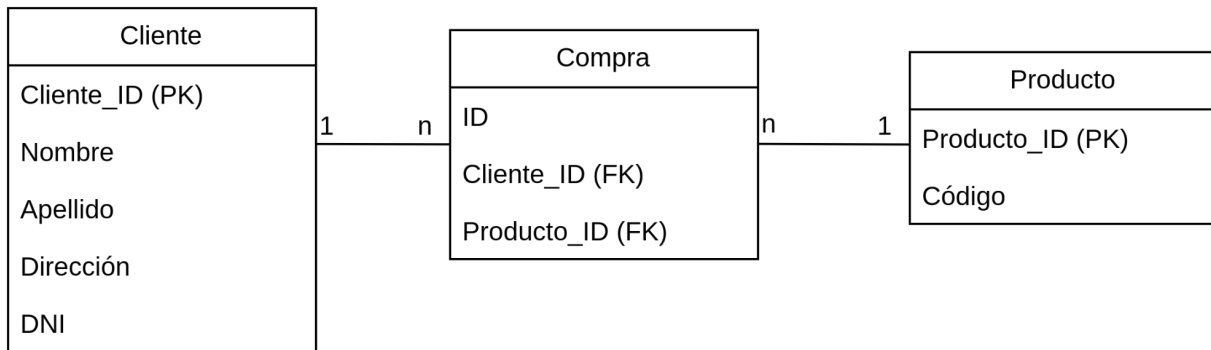
1. La entidad definida en rectángulo pasa a ser una tabla, ejemplo, la tabla cliente.
2. Los atributos en óvalos pasan a ser los campos, ejemplo, ID, Nombre, Apellido, Etc.
3. Debe existir un identificador único para la entidad o tabla, ejemplo, el ID.
4. El identificador único se transforma en clave Primaria.

Cliente
Cliente_ID (PK)
Nombre
Apellido

Reglas básicas de transformación

En relaciones N:N

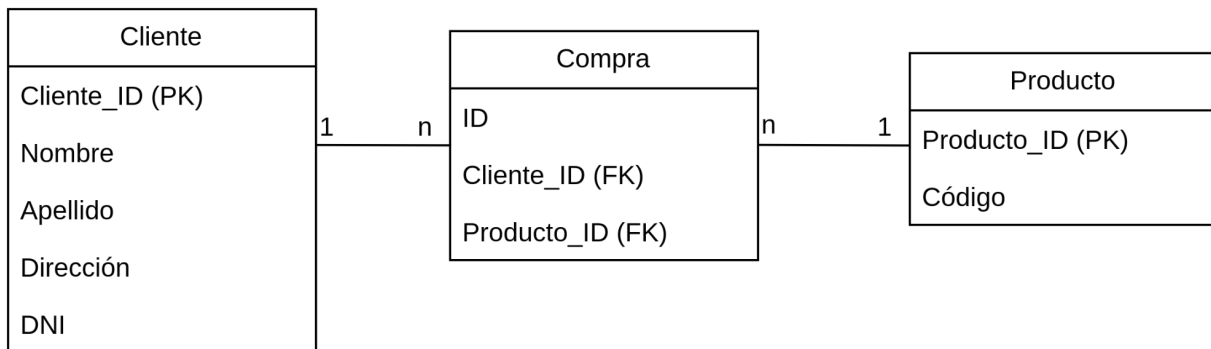
- Siempre que exista una relación muchos a muchos entre dos entidades debe existir una tabla intermedia que las interrelacione.



Reglas básicas de transformación

En relaciones N:N

- Se aplica un proceso de herencia.
- La tabla intermedia hereda de Cliente y Producto las claves primarias.
- Esta herencia es lo que genera la interrelación entre ambas tablas.



Consideración en reglas de transformación

Tomemos en consideración que estas reglas de transformación pueden ser amplias y abordan muchos más casos de implementación. Para los alcances de este curso vimos solo las fundamentales, sin embargo, en caso de querer profundizar más al respecto te recomendamos visitar documentaciones oficiales.

¿Qué tipo de relación
tienen las entidades
Estudiantes y Notas?



Ejercicio propuesto

"Genera el modelo
conceptual de
estudiantes y notas"



Genera el modelo conceptual de estudiantes y notas

Contexto

A continuación, deberás crear el modelo conceptual de dos entidades, Estudiantes y Notas. Para la entidad estudiantes deberán existir los atributos ID, Nombre, Apellido, Email y para la entidad de notas ID y resultado.

Luego, realiza el traspaso del modelo conceptual al lógico siguiendo las reglas de transformación mencionadas anteriormente.

Utiliza la [aplicación](#) de diagramas para realizar este ejercicio.



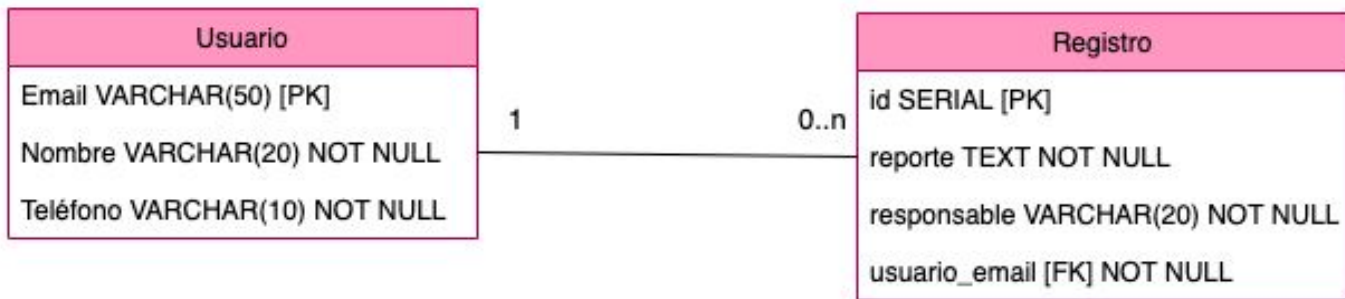
**/* Asignando tipos de datos y
restricciones al modelo */**

Modelo físico

- Responsable de representar cómo se construirá finalmente el modelo en nuestra base de datos.
- Incluye la estructuras de las tablas, definiendo cada uno de sus atributos y tipo de dato para cada atributo, las restricciones de las columnas, las claves primarias, claves foráneas y las relaciones entre las tablas.

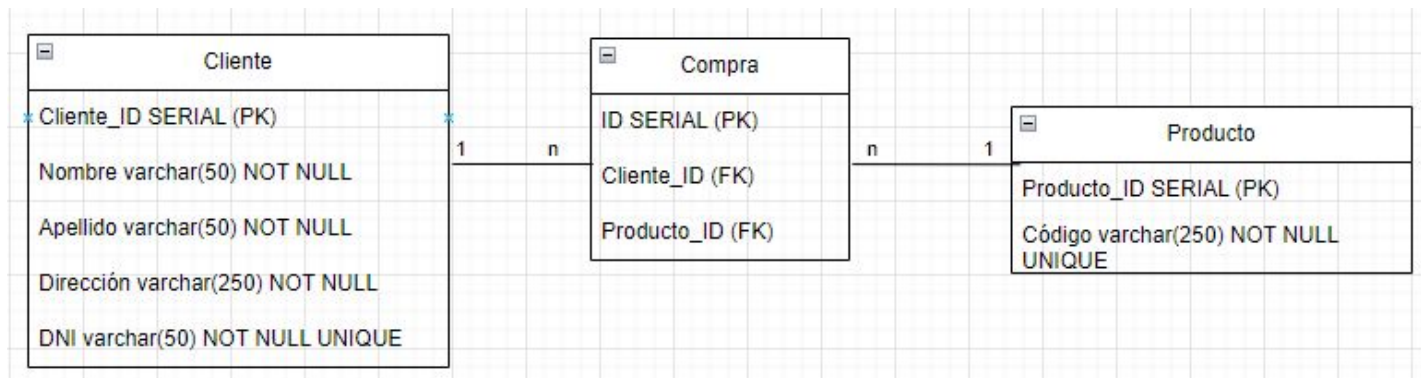
En síntesis, el modelo físico **es la versión final del modelo**, lo puedes interpretar como el modelo lógico con los metadatos declarativos de cada atributo.

Ejemplo de Modelo físico



Creando un diagrama UML

Seguiremos trabajando con el diagrama relacional de Clientes y Productos, en esta ocasión agregaremos tipado de datos y restricciones a cada uno de los atributos (campos) de nuestras tablas.



Creando un diagrama UML

Traspaso a SQL

Ya vimos varias formas de trabajar con SQL, por un lado desde la terminal o consola y además con una interfaz gráfica como DBeaver. A continuación, te mostraremos cómo realizar tus consultas y sentencias SQL desde Visual Studio Code para posteriormente cargarlo como fichero.

- **Paso 1:** Iniciamos el proceso creando la base de datos hacer la conexión a ella.

```
create database Productos_feria;
```

```
\c Productos_feria;
```



Creando un diagrama UML

Traspaso a SQL

- **Paso 2:** Descargar en la plataforma te dejaremos el script SQL desarrollado en Visual Studio Code con el nombre “**Material de apoyo - El modelo relacional (Parte I)**”.

Este, contiene el código de creación para las 3 tablas del modelo físico.

- **Paso 3:** Cargar el fichero con
`\i /ruta/ruta/nombre_del_archivo_sql;`



Ejecución desde ficheros

Comando \i

- Podemos crear ficheros con extensión `.sql` que nos permitirá escribir todos nuestros comandos SQL para poder cargarlos.
- El proceso de carga se realiza con el comando `\i`.
- La sintaxis de este comando es:

`\i <nombre_archivo.ext>` Ejecutar los comandos desde un archivo

Ejercicio propuesto

"Inserta registros en las
tablas desde un fichero SQL"



Contexto

Trabajando con la base de datos de productos de feria, deberás insertar algunos registros en cada tabla desde un fichero SQL.

- Genera los inserts en cada tabla.
- Ejecuta el fichero con el comando \i.



¿Pudiste cargar los inserts en las tablas?





Próxima sesión...

- *Guía de ejercicios*

{desafío}
latam_

*Academia de
talentos digitales*

