

# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS



## Proyecto Final: Diagrama Entidad-Relación

*Flores Martínez Andrés*  
*Vázquez Salcedo Eduardo Eder*  
*Sánchez Pérez Pedro Juan Salvador*  
*Concha Vázquez Miguel*

Trabajo presentado en cumplimiento con la asignatura de Fundamentos de  
Bases de Datos impartida por el profesor  
GERARDO AVILÉS ROSAS  
10 de enero de 2018



**A continuación detallaremos nuestras elecciones para la construcción del diagrama entidad-relación (E/R) del caso de uso de la taquería *Tacoste* planteado a partir de nuestro documento de análisis de requerimientos:**

Lo primero que hicimos fue seguir la recomendación de identificar en primera instancia a las entidades principales involucradas en el caso de uso, a pesar de que sabíamos que probablemente iban a irse viendo modificadas en el proceso. Evidentemente —a partir del documento de análisis de requerimientos— era importante considerar como entidades fuertes a los *clientes* de las distintas *sucursales* de las taquerías, así como también a los empleados que laboran en las mismas.

Sin embargo, desde un inicio notamos que había propiedades particulares que eran compartidas tanto por los clientes como por los empleados; estos atributos entonces fueron abstraídos en una super-entidad **Persona** y fueron: *email* (dirección de correo electrónico)<sup>1</sup>, *teléfono* (número telefónico)<sup>2</sup> y el *nombreCompleto*, siendo este último un atributo compuesto conformado por el *nombre*, *apellidoPaterno* y el *apellidoMaterno* del individuo. También añadimos un atributo compuesto con los datos de la *dirección* de la persona, constituido este último por los atributos del *CP* (código postal), *calle*, *colonia*, *municipio*, *estado* y *número*<sup>3</sup>. Como identificador incluimos una llave subrogada incremental que sería común a todo individuo que tuviera relación con las taquerías y que llamamos *taquiClave*.

Luego, aprovechando las capacidades ofrecidas por el modelo E/R extendido (EER), fue posible a través de la especialización definir dos sub-entidades que tuvieran asociados los atributos de **Persona** y que diera cabida también a establecer relaciones específicas adicionales entre las sub-entidades y otras que definiríamos más adelante; estas sub-entidades fueron justamente las de **Empleado** y **Cliente**.

---

<sup>1</sup>Consideramos que es suficiente contar con una sola dirección de correo por cada persona.

<sup>2</sup>Asimismo, suponemos que basta con tomar en cuenta un único número telefónico por cada persona.

<sup>3</sup>Decidimos que había que definir al atributo *número* como un atributo compuesto que incluyera un *numInterior* y *numExterior* para ser más precisos.

En la especialización definimos una restricción de disyunción para poder establecer que bajo nuestro esquema no era posible que un empleado fuera a su vez un cliente de la taquería<sup>4</sup>. Subsecuentemente establecimos también una restricción de completez de especialización total, pues forzosamente ocurría que todo individuo del cual íbamos a almacenar información era forzosamente un cliente o un empleado de la taquería.

En particular, para la sub-entidad **Empleado** agregamos como sus atributos el *CURP*, *RFC*, *tipo* (referente al tipo de empleado en cuestión), *tipoSangre*, *fechaNac* (fecha de nacimiento)<sup>5</sup>, *numEmergencia* (un número telefónico de alguna persona de contacto en caso de algún percance o emergencia) y la fecha de contratación del empleado para poder tener información de su antigüedad al que llamamos *fechaContratación*. En cuanto al conjunto entidad **Cliente**, incluimos como sus atributos *numPuntos* para poder determinar el número de puntos acumulados por cada cliente bajo el programa de **taquero mucho** y un atributo *fechaPrimerVisita* para poder determinar la primera vez que consumió en alguna sucursal; esto nos podría servir para poder —en algún momento— dar beneficios a los clientes más antiguos.

Para la entidad fuerte **Sucursal** que identificamos a partir del análisis de requerimientos, incluimos como sus atributos un identificador *idSucursal* que funcionaría como su llave primaria, un atributo compuesto *horario* que incluía a su vez la *horaApertura* y *horaCierre* de las sucursales, el atributo compuesto *dirección* que de nueva cuenta estaba conformado por *municipio*, *colonia*, *calle*, *CP*, *estado* y el atributo a su vez compuesto de *numInterior* y *numExterior* de *número*. Como pensamos que era importante prever la posibilidad de que una misma sucursal tuviera varios teléfonos de contacto, entonces agregamos un atributo multivaluado *teléfono* y también un atributo calculado de *numEmpleados* para poder conocer cuántos empleados laboran en cada sucursal de Tacoste.

---

<sup>4</sup>Dado el caso en que un posible empleado quisiera ir, por ejemplo, a comer el fin de semana con su familia a alguna sucursal, entonces sería todavía considerado como un empleado para evitar la duplicidad de la información en la base de datos. Esta posibilidad tiene cabida dentro del modelo planteado puesto que las relaciones de *pedidos* y demás ocupan el identificador de la **Persona**: la *taquiClave* que es común tanto a personas como a empleados.

<sup>5</sup>A partir de esta información podríamos posteriormente obtener la edad de cada empleado.

A partir de las entidades definidas hasta el momento fue posible definir dos relaciones entre la sub-entidad **Empleado** y la entidad **Sucursal** a partir del papel que cada una desempeña. Era importante conocer al gerente de cada sucursal a pesar de que no estaba especificado en el caso de uso, pero sí en nuestro análisis de requerimientos con tal de tener escalabilidad. Así, establecimos una relación *dirigir* entre las entidades previamente mencionadas, con un atributo de relación *fechaInicio* para poder llevar cuenta del tiempo en que el gerente ha desempeñado su papel.

Para esta relación binaria, la restricción de participación fue parcial del lado de **Empleado** teniendo en cuenta que no todos los empleados forzosamente eran gerentes, y fue total del lado de **Sucursal**, pues no pueden haber sucursales sin supervisores. Para la restricción de cardinalidad, la definimos de tipo muchos-uno ya que un empleado puede solamente estar asociado a una sucursal, no dando así cabida a que un gerente sea dirigente de varias sucursales a la vez; por otro lado, una misma sucursal consideramos que puede tener más de un gerente ya que puede ser que estén enfocados a dirigir áreas ajenas.

La segunda relación establecida entre **Sucursal** y **Empleado** fue la de *trabajar*, identificada a partir del análisis de requerimientos y evidente desde el caso de uso. Entonces, para poder identificar el salario de cada empleado, nos pareció oportuno incluir un atributo de relación *salario*. La restricción de participación fue parcial del lado del empleado en el sentido en que podrían haber personas como el señor Pepe que no necesariamente laboraran en una sucursal en el sentido de que llevan un control más global de la taquería; del lado de Sucursal fue total para indicar que no puede haber sucursales sin empleados. Para la restricción de cardinalidad, impusimos un esquema muchos-uno para indicar que los empleados pueden estar asociados a una única sucursal como indicamos en las restricciones solicitadas, pero que en una sucursal puede laborar más de un empleado.

También notamos que era necesario poder identificar a los jefes particulares de cada empleado, estableciendo así una nueva relación binaria de la entidad **Empleado** consigo misma que definimos con el nombre *supervisar*. A pesar de que no era obligatorio, decidimos etiquetar los roles de participación de ambas partes con “gerente” y “subordinado”, teniendo así una mayor claridad. La participación fue total de ambos lados, pues no puede haber empleados sin algún tipo de jefe, como tampoco jefes sin subordinados. Además, la restricción de cardinalidad fue de tipo uno-muchos, pues

un mismo gerente puede tener a varios empleados bajo su supervisión, pero cada empleado puede tener un solo empleado al cual responde directamente.

A pesar de que ya habíamos considerado un atributo para el tipo de empleado en el conjunto entidad de **Empleado** que nos permitiría definir a los parrilleros, meseros, taqueros, cajeros, tortilleros, repartidores y dando cabida a la escalabilidad, decidimos finalmente abstraer a los repartidores en una sub-entidad de los empleados que llamamos **TacoRider**. La razón principal fue la de poder almacenar información adicional de éstos últimos a raíz de los datos que tendrían que tener de los pedidos a domicilio, su licencia (cuando aplicara) y los datos de los vehículos utilizados por ellos. La restricción de completez en esta especialización fue parcial, pues no todo empleado es necesariamente un repartidor. No fue necesario establecer ninguna restricción de disyunción al tratarse de una única sub-entidad. Como atributo incluimos uno de tipo booleano que llamamos *¿estáDisponible?* para poder saber si está en condición de hacer un envío en un momento determinado.

Al ser la licencia algo opcional y que aplica en ocasiones dependiendo del tipo de vehículo utilizado por el repartidor, decidimos crear una nueva entidad fuerte **Licencia**. A primera vista pensamos que se trataba de una entidad débil forzosamente debía existir un repartidor para poder existir un registro de la licencia. No obstante, luego caímos en cuenta de que en emergencias sería posible hacer que algún otro empleado de otro tipo se convirtiera momentáneamente en **tacoRider** con tal de hacer entregas a domicilio. Bajo esta lógica, las licencias podrían existir aún cuando todavía no estuviera contemplado un empleado particular como un repartidor para poder decidir justamente quién era candidato a realizar una entrega en motocicleta. El único atributo que incluimos en la entidad **Licencia** fue el de su *código*, mismo que funge como llave de la entidad fuerte.

La relación binaria creada entre la **Licencia** y el **TacoRider** fue la de *tener*. La restricción de participación fue total del lado de la **Licencia** y parcial del lado de **TacoRider** para contemplar la posibilidad de que no tuviera una licencia al moverse únicamente en bicicleta como se especificaba en el documento de análisis de requerimientos. La restricción de cardinalidad fue de uno-uno porque un repartidor no puede tener más de una licencia legalmente, así como una licencia no puede pertenecer a más de un repartidor.

De igual manera identificamos una nueva relación *poseer* entre las dos entidades previamente descritas para poder tomar en cuenta a los medios de transporte. Así, también fue necesario crear una entidad **Transporte**. Decidimos que la entidad era fuerte en el sentido en que los transportes podrían existir inclusive cuando no hubiera repartidores porque serían de la taquería propiamente. Para dicha entidad consideramos como atributos una llave subrogada *idTransporte*, *marca*, *modelo* y *tipo*. Poder presentar el tipo del vehículo de esta manera ofrece escalabilidad ya que eventualmente los dueños de la taquería pueden plantearse la inclusión de vehículos de entrega de nuevo tipo, como automóviles.

En cuanto a la relación *poseer*, la restricción de participación fue total de ambos lados para indicar que todo repartidor debe tener un vehículo asignado, y todo vehículo debe ser usado por algún repartidor, pues sería un desperdicio para la taquería comparar vehículos sobrantes y desde el caso de uso se busca una optimización de recursos monetarios. Luego, la restricción de cardinalidad fue de muchos-muchos para indicar que un mismo vehículo puede ser eventualmente conducido por distintos repartidores, y que un mismo repartidor puede utilizar distintos medios de transporte para la entrega de los pedidos.

Como siguiente paso en el proceso del diseño del diagrama, comenzamos a definir entidades fuertes para **Pedido**, **Producto**, **Salsa** e **Ingrediente** como habíamos podido observar a través de lo definido en el análisis de requerimientos.

En lo que respecta a los productos vendidos en las taquerías Tacoste, decidimos incluir como atributos una llave primaria *idProducto*, el *nombre*, su *precio* y el número de puntos que genera el producto para poder asegurar escalabilidad en el sentido en como la empresa maneja el proyecto de *taquero mucho*. También consideramos la *taquegoría* de cada producto (categoría) y un atributo multivaluado para las *leyendas* que puede tener el producto y que se presentan en el menú como pueden ser light, vegano, orgánico, etcétera.

Para los ingredientes, de forma similar añadimos como atributos una llave *idIngrediente*, *nombre*, *marca*, *fechaCaducidad* (con tal de evitar potenciales desperdicios), y la *cantidadExistencia* para poder darse cuenta del momento en que resulta necesario la compra de más ingredientes de cierto tipo.

Para la entidad **Salsa**, notamos que tenía que tener los atributos de **Producto**, además de *presentación*, y *scoville* para contemplar el nivel de picor de cada salsa. Por ende, impusimos un esquema de especialización, considerando a la **Salsa** como sub-entidad de **Producto**. No hubo necesidad de imponer restricciones de disyunción al ser solamente un conjunto de entidades especializado; para la restricción de completez, la especificamos parcial ya que no todo producto es necesariamente una salsa.

Ahora, para la entidad fuerte de **Pedido** definimos como atributo llave el *numPedido*, pero también consideramos como atributos la *fechaPedido*, *promoción*<sup>6</sup>, así como un atributo calculado de *costoTotal*.

A partir de estas nuevas entidades fuertes definimos nuevas relaciones. Como los repartidores se encargan de entregar pedidos a domicilio, entonces creamos una relación con restricción de cardinalidad uno-mucho llamada **llevar** entre **TacoRider** y **Pedido**. El motivo fue que cada pedido es entregado por un solo repartidor, pero un mismo repartidor puede hacer entrega de múltiples pedidos. Ahora, para la restricción de participación se especificó como total del lado de **TacoRider** porque no puede haber repartidores que no hagan entregas; fue parcial del lado de **Pedido** ya que puede haber órdenes que tienen lugar dentro de las sucursales físicamente, así que no se tiene que hacer un envío a domicilio.

Se especificó también una relación *ordenar* con restricción de cardinalidad uno-muchos entre **Cliente** y **Pedido**, pues un mismo cliente puede hacer múltiples pedidos, pero no puede ocurrir que un mismo pedido corresponda a más de un cliente. De esta forma solucionamos el inconveniente de las mesas, pues estamos definiendo que cada pedido estará asociado en estos escenarios con algún cliente de dicha mesa que actúa como responsable de la compra. Como atributo de la relación incluimos *métodoPago* para especificar la forma en la que se efectuó el cobro del pedido (tarjeta de crédito o débito, efectivo, vales, cripto-moneda, etcétera). Finalmente, la restricción de participación fue total de ambos lados, pues forzosamente un cliente de la taquería efectúa pedidos por definición de cliente, así como los pedidos tiene que ser hechos por clientes.

---

<sup>6</sup>Estamos evitando a toda costa los valores nulos en nuestro diseño, por lo que en caso de que no aplique alguna promoción, tendremos que especificarlo explícitamente en la base de datos.



Como se llevan a cabo los pedidos en las sucursales, establecimos una relación *ocurrir* entre **Sucursal** y **Pedido**. La restricción de cardinalidad fue de uno-muchos, pues en una sucursal se pueden presentar varios pedidos, pero un pedido no puede tener lugar en más de una sucursal. Fue total del lado de **Pedido**, pues forzosamente tiene que ocurrir en una sucursal para poder definirse; en cuando a **Sucursal**, su participación es parcial, pues puede haber sucursales registradas que todavía no hayan registrado ningún pedido.

Luego, definimos una relación binaria de nombre *contener* entre los conjuntos de entidades fuertes **Producto** y **Pedido** con atributo de relación *cantidad* para poder registrar la cantidad de cada producto por pedido. La restricción de participación fue total de ambos lados, pues un pedido forzosamente tiene productos y un los productos necesariamente forman parte de algún pedido. Más aún, la restricción de cardinalidad la definimos de tipo muchos-muchos, pues un pedido puede consistir de más de un producto, así como un producto puede estar presente en varios pedidos.

La relación con nombre *tener* que definimos entre **Producto** e **Ingrediente** fue a raíz de nuestro documento de análisis de requerimientos y su restricción de cardinalidad de tipo muchos-muchos. Esto último se debió a que un ingrediente puede estar presente en varios productos, así como un producto puede llevar más de un ingrediente —y generalmente este es el caso. Ahora, la restricción de participación fue total del lado de **Producto** porque forzosamente se componen de ingredientes y parcial del otro lado, pues podemos tener ingredientes en una sucursal que todavía no hayan sido usados y en consiguiente no formen parte de algún producto. También añadimos un atributo de relación *cantidad* para poder justamente indicar la fracción del ingrediente presente en cada producto.

Dado que de acuerdo al caso de uso y a lo reflejado por el análisis de requerimientos en la sección de relaciones entre los datos era necesarios que las salsas recomendaron productos con qué acompañarlas, entonces definimos una relación *recomendar* entre **Producto** y **Salsa**. La restricción de participación fue parcial del lado de **Producto**, pues los productos no necesariamente recomiendan salsas; lo contrario no puede decirse, pues necesariamente ocurre y entonces fue total del lado del conjunto entidad **Salsa**. La restricción de cardinalidad fue muchos-muchos para contemplar la posibilidad de que un producto pueda recomendar muchas salsas y viceversa.

El siguiente paso en la creación del diagrama fue el establecimiento de una nueva entidad fuerte para los proveedores: **Proveedor**. Para poder hacer la diferenciación entre el tipo de bienes que proveen, creamos una entidad **Mobiliario** con atributos *tipo* y un *idMueble* como una llave. En cuanto a los proveedores, agregamos atributos de *inicioRelación*<sup>7</sup>, *RFC* como llave, *razónSocial*, teléfono, *email*<sup>8</sup> y el atributo compuesto correspondiente a la dirección: *CP*, *colonia*, *calle*, *municipio*, *estado* y el atributo compuesto de *numInterior* y *numExterior* del número.

Como siguiente paso, establecimos dos relaciones análogas con atributo *precio* entre la entidad **Proveedor** y las entidades respectivas de **Ingrediente** y **Mobiliario**. Las restricciones en ambos casos fueron las mismas: en cuando a la participación fue total de ambos lados y de cardinalidad muchos-muchos. Las razones fueron que forzosamente los proveedores tienen que hacer entrega de dichos bienes (según sea el caso) y estos bienes tienen que ser necesariamente entregados por los proveedores con tal de no romper potenciales alianzas comerciales. Además, un mismo proveedor puede hacer entrega de varios bienes, y un mismo bien puede ser provisto por distintos proveedores. A dichas relaciones las llamamos *proveerIng* en el caso de la relación binaria entre **Proveedor** e **Ingrediente**; para la relación binaria entre **Proveedor** y **Mobiliario** la llamamos *proveerMob*.

La última fase de traducción consistió en considerar a la entidad fuerte **Histórico** ya identificada en el análisis de requerimientos. Dicha entidad tenía como atributos el *idProducto*, *fechaActualización*, *precioPrevio* y *precioNuevo* tras la actualización. El identificador del producto funcionó como atributo llave de la entidad. Establecimos de igual forma una relación binaria entre **Histórico** y **Producto** que llamamos *conservar*, con una restricción de cardinalidad muchos-muchos y participación total. Esto se debe a que todos los productos deben mantener su histórico, y forzosamente los históricos contemplan productos. Por otro lado, un mismo producto puede llevar varios históricos y un histórico considerar a varios productos.

Notamos finalmente que no especificamos entidades débiles en el modelo,

---

<sup>7</sup>El motivo de esto fue para considerar la fiabilidad de cada proveedor a partir de la antigüedad de la relación para considerar la realización de alianzas comerciales como se había especificado en el caso de uso.

<sup>8</sup>Así como con las personas, consideramos suficiente almacenar un registro de un único número telefónico y dirección de correo electrónico como forma de contacto para cada proveedor.

pues cada entidad definida puede definirse y existir por sí sola sin tener que requerir de otras para tener un sentido completo.

El resultado de la creación del modelo a partir del caso de uso y el subsecuente reflejo del mismo en el documento del análisis de requerimientos está presente en la segunda página de este mismo documento.

## Referencias

- [1] *Modelo Entidad Relación*, Avilés Rosas Gerardo. UNAM, Facultad de Ciencias, págs. 1-27.