

TEMA 5  
MODELO DE DATOS EXTENDIDO



### 5.1. ENTIDADES SUPERTIPO Y SUBTIPO

Para entender estos 2 conceptos dentro del modelado de bases de datos, considerar el siguiente escenario. Suponer que se tiene la siguiente lista de entidades con sus respectivos atributos:

- Profesor (**nombre, apellido paterno, apellido materno, edad**, email, RFC, tipo profesor)
- Investigador (**nombre, apellido paterno, apellido materno, edad**, email, cédula, total\_articulos)
- Administrativo (**nombre, apellido paterno, apellido materno, edad**, departamento, horas\_extras)

Observaciones de las 3 entidades anteriores:

- Las 3 entidades tienen atributos en común: **nombre, apellido paterno, apellido materno y edad**.
- Cada entidad tiene atributos que son particulares, por ejemplo, el RFC solo aparece en la entidad profesor.

Para este escenario, existe una forma particular de realizar el modelado de datos aplicando una especie de “factorización” de los atributos comunes, similar al concepto de herencia dentro de la programación orientada a objetos (POO).

- Los atributos comunes se extraen y se asocian a una entidad llamada “**supertipo**”. En este caso, es posible crear una nueva entidad llamada *persona* que tendrá el rol de **supertipo**.
- Las entidades *profesor*, *investigador* y *administrativo*, se les conoce como “**subtipos**” de la entidad *persona*.

### 5.2. IDENTIFICACIÓN DE UNA RELACIÓN DE SUPERTIPO - SUBTIPOS

Existen 2 estrategias para realizar la identificación:



#### 5.2.1. Generalización: Proceso Bottom – Up

PROFESOR

|                |              |          |
|----------------|--------------|----------|
| ◆ NOMBRE       | VARCHAR(30)  | NOT NULL |
| ◆ AP_PATERNO   | VARCHAR(30)  | NOT NULL |
| ◆ AP_MATERNO   | VARCHAR(30)  | NOT NULL |
| ◆ EDAD         | NUMERIC(2,0) | NOT NULL |
| ◆ EMAIL        | VARCHAR(150) | NOT NULL |
| ◆ RFC          | VARCHAR(13)  | NOT NULL |
| ◆ ESPECIALIDAD | VARCHAR(30)  | NOT NULL |

INVESTIGADOR

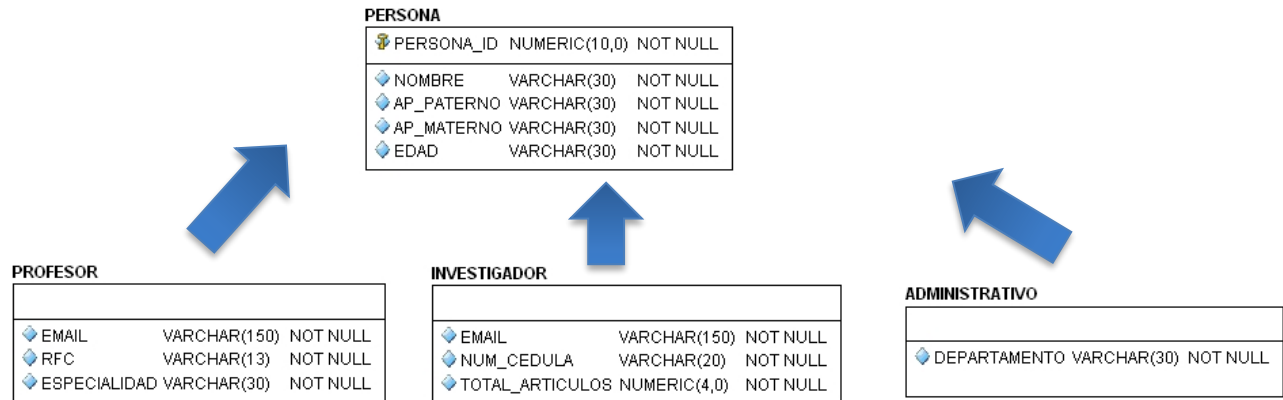
|                   |              |          |
|-------------------|--------------|----------|
| ◆ NOMBRE          | VARCHAR(30)  | NOT NULL |
| ◆ AP_PATERNO      | VARCHAR(30)  | NOT NULL |
| ◆ AP_MATERNO      | VARCHAR(30)  | NOT NULL |
| ◆ EDAD            | NUMERIC(2,0) | NOT NULL |
| ◆ EMAIL           | VARCHAR(150) | NOT NULL |
| ◆ NUM_CEDULA      | VARCHAR(20)  | NOT NULL |
| ◆ TOTAL_ARTICULOS | NUMERIC(4,0) | NOT NULL |

ADMINISTRATIVO

|                |               |          |
|----------------|---------------|----------|
| ◆ NOMBRE       | VARCHAR(30)   | NOT NULL |
| ◆ AP_PATERNO   | VARCHAR(30)   | NOT NULL |
| ◆ AP_MATERNO   | VARCHAR(30)   | NOT NULL |
| ◆ EDAD         | NUMERIC(30,0) | NOT NULL |
| ◆ DEPARTAMENTO | VARCHAR(30)   | NOT NULL |

- En este caso, se cuenta con la existencia de 2 o más entidades y se identifica la existencia de atributos en común.
- En este proceso, los atributos en común se extraen o se separan de las entidades y se agregan a una nueva entidad “general” que actuará como supertipo.

- La forma más sencilla es identificando los campos que se repiten en las 3 tablas. Al aplicar un proceso de generalización, se obtiene lo siguiente:



### 5.2.2. Especialización: Proceso Top - Down

- En este caso, se parte de la existencia de una sola entidad.
- Se observa que una entidad tiene ciertos atributos y/o interrelaciones que tienen sentido para ciertas instancias, pero no para otras, por lo que es conveniente definir uno o varios subtipos.

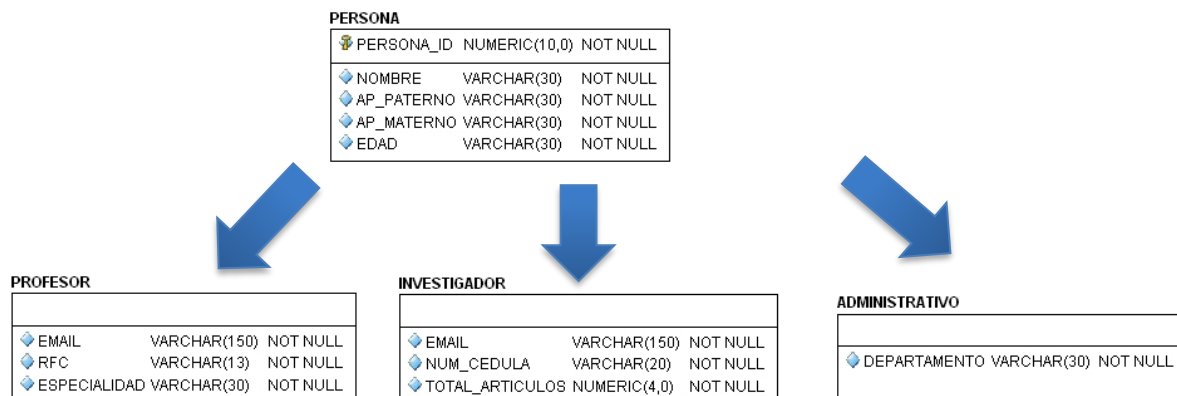
#### Ejemplo:

**PERSONA**

|                 |               |          |
|-----------------|---------------|----------|
| PERSONA_ID      | NUMERIC(10,0) | NOT NULL |
| NOMBRE          | VARCHAR(30)   | NOT NULL |
| AP_PATERNO      | VARCHAR(30)   | NOT NULL |
| AP_MATERNO      | VARCHAR(30)   | NOT NULL |
| EDAD            | VARCHAR(30)   | NOT NULL |
| EMAIL           | VARCHAR(150)  | NULL     |
| RFC             | VARCHAR(13)   | NULL     |
| ESPECIALIDAD    | VARCHAR(30)   | NULL     |
| NUM_CEDULA      | VARCHAR(20)   | NULL     |
| TOTAL_ARTICULOS | NUMERIC(4,0)  | NULL     |
| DEPARTAMENTO    | VARCHAR(30)   | NULL     |



- Observar que los campos particulares de cada subtipo se declaran como NULL. Al crear un registro, algunos campos pueden carecer de valor.
- En este proceso los campos se extraen y se crea una nueva tabla subtipo. Debe existir al menos un campo para que la tabla tenga sentido:

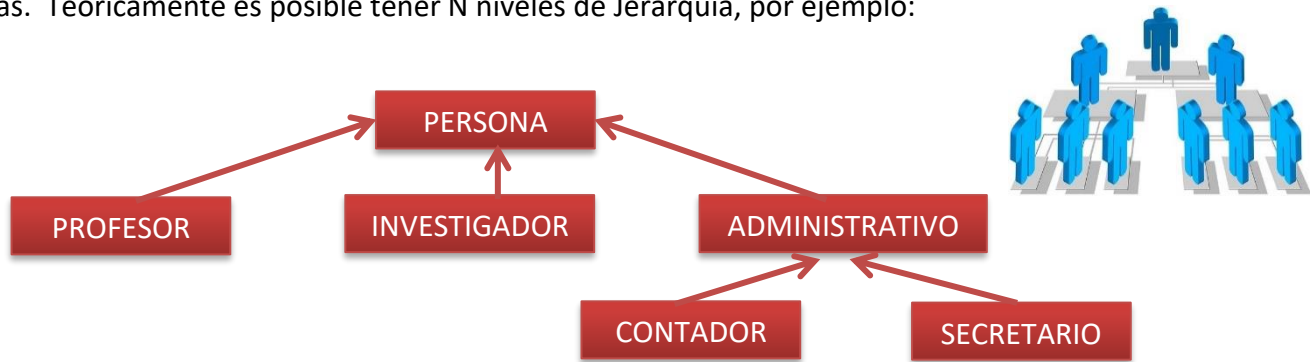


Para cualquiera de las 2 estrategias, verificar la correcta construcción del supertipo y sus subtipos, se deberán cumplir las siguientes condiciones:

- Todos los subtipos deberán contar con al menos un campo particular. No deben contener campos comunes.
- Al igual que en herencia, la existencia del subtipo se justifica aplicando la palabra “ES”. Por ejemplo, un profesor es una persona, un investigador es una persona, etc.

### 5.3. JERARQUÍAS DE ESPECIALIZACIÓN.

Observar que esta estrategia de diseño lleva a la construcción de una Jerarquía en este caso, jerarquía de tablas. Teóricamente es posible tener N niveles de Jerarquía, por ejemplo:



Con base a las 2 estrategias de identificación mencionadas anteriormente, se tiene un total de 3 posibles opciones de diseño:

#### 5.3.1. opción 1: Nivel alto de Jerarquía.

En esta estrategia, se crean tantos niveles y subtipos como sean necesarios. En el ejemplo anterior, se tienen 3 niveles: 2 supertipos (persona y administrativo) con sus respectivos subtipos.

#### Desventajas:

- A mayor nivel de Jerarquía, el desempeño de la base de datos puede verse afectado, ya que se requiere un mayor número de operaciones `join` para ligar (relacionar) tablas. Por ejemplo, para obtener los datos de un contador, se requieren realizar operaciones `join` entre 3 tablas: contador, administrador y persona.
- Típicamente un nivel máximo de 2 podría considerarse como razonable en una base de datos productiva o transaccional.

#### Ventajas:

- A nivel de diseño, esta estrategia representa la mejor opción. Los atributos particulares a cada subtipo se definen como `not null`, existe un mayor control de la integridad de los datos.
- Adecuada cuando se tenga un número de atributos aproximadamente similar entre el supertipo y cada uno de sus subtipos.
- Para aplicaciones que hacen uso de la programación orientada a objetos que acceden a la base de datos, un nivel alto de jerarquía permite establecer una relación 1 – 1 con respecto al modelo de clases. Por ejemplo, a nivel de la aplicación puede existir esta misma jerarquía de clases, por lo que habrá una correspondencia directa entre una clase y una tabla: A la clase `Persona` le corresponde la tabla `persona`, y así sucesivamente.

#### 5.3.2. Opción 2: Tabla simple sin subtipos:

En esta técnica se eliminan los subtipos y sus atributos son incorporados al supertipo como atributos opcionales, es decir, se aplica un proceso de **generalización**.

- Los atributos deben ser incorporados como nulos ya que no todas las instancias del supertipo estarán asociadas con cada subtipo: No todas las personas son profesores, o administradores, o investigadores.

#### Desventajas:

- Los campos particulares a cada subtipo se deberán definir como null. Esto puede generar problemas, por ejemplo, Un profesor se puede registrar sin su RFC. ¡La base de datos lo permite!

#### Ventajas:

- Se tiene el mejor resultado en cuanto a desempeño, no requiere ligar tablas.
- Adecuada cuando se tienen muy pocos atributos particulares por cada subtipo.

| PERSONA         |               |          |
|-----------------|---------------|----------|
| PERSONA_ID      | NUMERIC(10,0) | NOT NULL |
| NOMBRE          | VARCHAR(30)   | NOT NULL |
| AP_PATerno      | VARCHAR(30)   | NOT NULL |
| AP_MATerno      | VARCHAR(30)   | NOT NULL |
| EDAD            | VARCHAR(30)   | NOT NULL |
| EMAIL           | VARCHAR(150)  | NULL     |
| RFC             | VARCHAR(13)   | NULL     |
| ESPECIALIDAD    | VARCHAR(30)   | NULL     |
| NUM_CEDULA      | VARCHAR(20)   | NULL     |
| TOTAL_ARTICULOS | NUMERIC(4,0)  | NULL     |
| DEPARTAMENTO    | VARCHAR(30)   | NULL     |

### 5.3.3. Opción 3: Conjunto de tablas sin su supertipo:

En esta técnica se elimina el supertipo, sus atributos son incorporados a cada uno de los subtipos. Esto significa que existirá cierta redundancia de atributos en cada subtipo. En este escenario se aplica el concepto de **especialización**.

#### PROFESOR

|              |              |          |
|--------------|--------------|----------|
| NOMBRE       | VARCHAR(30)  | NOT NULL |
| AP_PATerno   | VARCHAR(30)  | NOT NULL |
| AP_MATerno   | VARCHAR(30)  | NOT NULL |
| EDAD         | NUMERIC(2,0) | NOT NULL |
| EMAIL        | VARCHAR(150) | NOT NULL |
| RFC          | VARCHAR(13)  | NOT NULL |
| ESPECIALIDAD | VARCHAR(30)  | NOT NULL |

#### INVESTIGADOR

|                 |              |          |
|-----------------|--------------|----------|
| NOMBRE          | VARCHAR(30)  | NOT NULL |
| AP_PATerno      | VARCHAR(30)  | NOT NULL |
| AP_MATerno      | VARCHAR(30)  | NOT NULL |
| EDAD            | NUMERIC(2,0) | NOT NULL |
| EMAIL           | VARCHAR(150) | NOT NULL |
| NUM_CEDULA      | VARCHAR(20)  | NOT NULL |
| TOTAL_ARTICULOS | NUMERIC(4,0) | NOT NULL |

#### ADMINISTRATIVO

|              |               |          |
|--------------|---------------|----------|
| NOMBRE       | VARCHAR(30)   | NOT NULL |
| AP_PATerno   | VARCHAR(30)   | NOT NULL |
| AP_MATerno   | VARCHAR(30)   | NOT NULL |
| EDAD         | NUMERIC(30,0) | NOT NULL |
| DEPARTAMENTO | VARCHAR(30)   | NOT NULL |

#### Ventajas:

- El desempeño no se afecta. Se cuenta únicamente con subtipos (un solo nivel).
- Los campos particulares se pueden definir como not null
- Adecuada para casos donde se tienen muy pocos atributos comunes.
- Adecuada cuando se tienen una relación de exclusión. Por ejemplo, una persona cuenta con un solo rol. Esto implica que no existirá redundancia de datos para los atributos en común, los cuales se encuentran en cada uno de los subtipos.

#### Desventajas:

- Los valores de los atributos comunes en cada tabla se repiten. Por ejemplo, suponer que una persona puede contar con varios roles a la vez, administrativo y profesor. Esto significa que existirá una instancia en la entidad administrativo y otra en profesor. Los valores de los atributos en común deberán repetirse en cada subtipo, por ejemplo, el nombre, apellidos, etc.

## 5.4. RELACIONES ENTRE UN SUPERTIPO Y SUS SUBTIPOS.

¿Qué tipo de relación existirá entre un supertipo y cada uno de sus subtipos?

- Una instancia de la tabla supertipo se asocia con 0 o máximo con un registro de alguno de sus subtipos. Por ejemplo, la persona con ID =1, no puede asociarse con 2 registros de la tabla *profesor*, ya que eso significaría que 2 profesores tienen la misma identidad. Por lo anterior, la relación de supertipo y subtipo es 1:1.

Existen 4 variantes para representar una relación entre un supertipo y su subtipo, se expresan a través de 2 tipos de restricciones:

#### **5.4.1. Restricción Excluyente/traslape (disjoint / overlapping).**

- **Excluyente:** Cada instancia del supertipo se asocia a lo más con una instancia de alguno de sus subtipos.
- **Traslape:** Cada instancia del supertipo puede asociarse con más de una instancia de sus subtipos

#### **5.4.2. Restricciones Parciales o totales (partial / complete).**

- **Total:** Cada instancia del supertipo se asocia por lo menos con una instancia de sus subtipos.
- **Parcial:** Cada instancia del supertipo puede o no asociarse con las instancias de sus subtipos.

Independiente a las restricciones anteriores, la cardinalidad entre un supertipo y cada uno de sus subtipos siempre será (0,1). Esto se revisa a detalle más adelante.

#### **5.4.3. Discriminante de subtipo.**

Considerar el siguiente escenario:

Se desea obtener todos los datos de una persona empleando como único criterio su identificador. Al emplear un diseño Supertipo -Subtipos surgen los siguientes puntos:

- ¿Cómo podríamos determinar con cuál de sus subtipos se relaciona cada instancia del supertipo? A partir del identificador no es posible determinar el subtipo(s) asociado(s) con la instancia del supertipo en cuestión. Se tendría que realizar una búsqueda del registro asociado en cada subtipo lo que puede generar inconvenientes de desempeño.
- Para resolver la pregunta anterior se emplea el concepto de “**discriminante**”.

El discriminante de subtipo puede ser implementado por uno o varios atributos, e inclusive por una entidad. Su estructura depende de las combinaciones entre las restricciones mencionadas anteriormente.

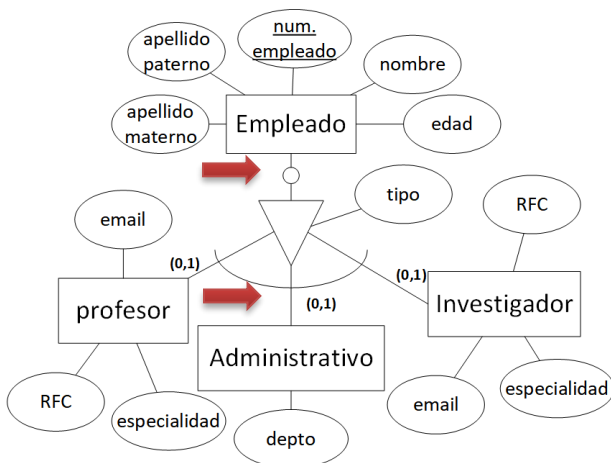


#### **Ejemplos:**

##### **Restricción excluyente / total (disjoint / complete)**

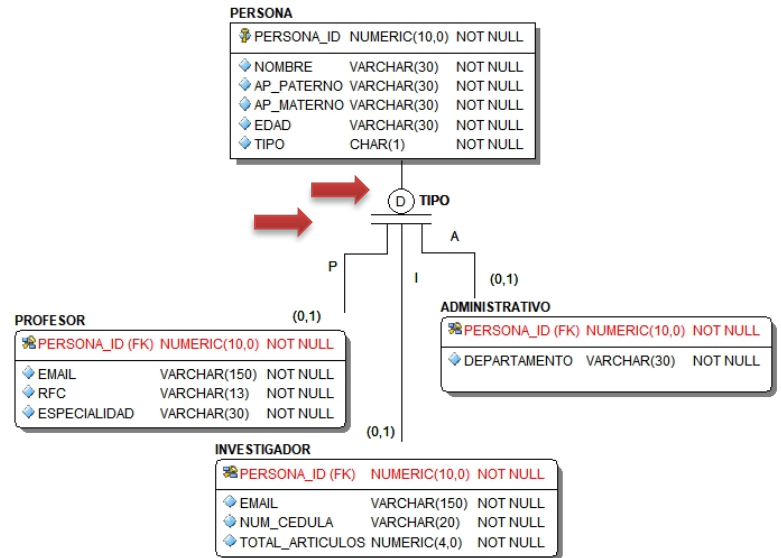
Una persona debe contar con un único rol: profesor, administrador o administrativo

## Diseño conceptual

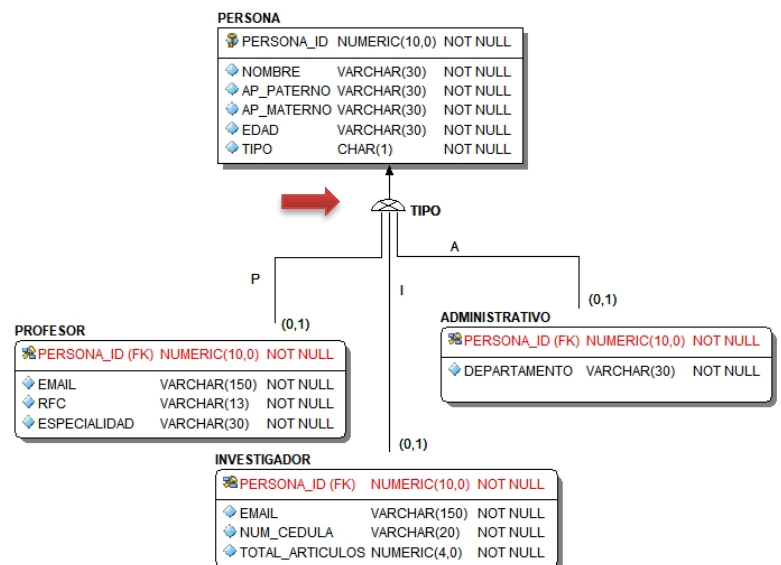


- Se emplea el arco para indicar una restricción excluyente. Se asocia al triángulo invertido.
- Se emplea un círculo para indicar una restricción total.
- El discriminante se representa por un atributo. Tipo = P, A, I

## Diseño lógico



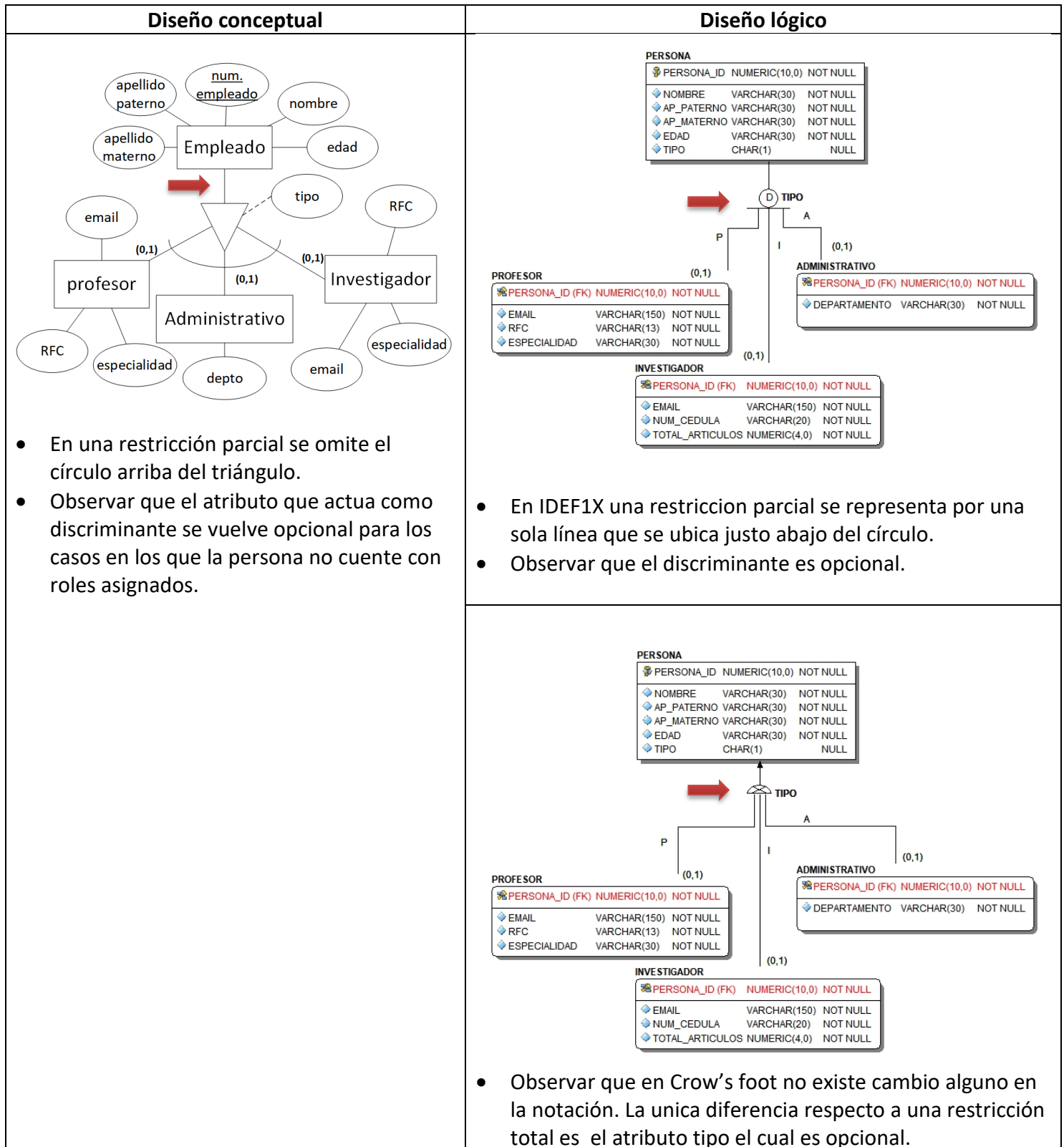
- En IDEF1X Se emplea un círculo y al interior una letra “D” para indicar una restricción excluyente (“Disjoint”).
- Se emplea doble línea horizontal justo abajo del círculo para indicar una restricción total.
- El discriminante se incorpora al supertipo.



- En Crow's foot se emplea un semi-círculo con una cruz al centro para representar una restricción excluyente.
- No existe notación para representar una restricción total.

## Restricción excluyente / parcial (disjoint / partial)

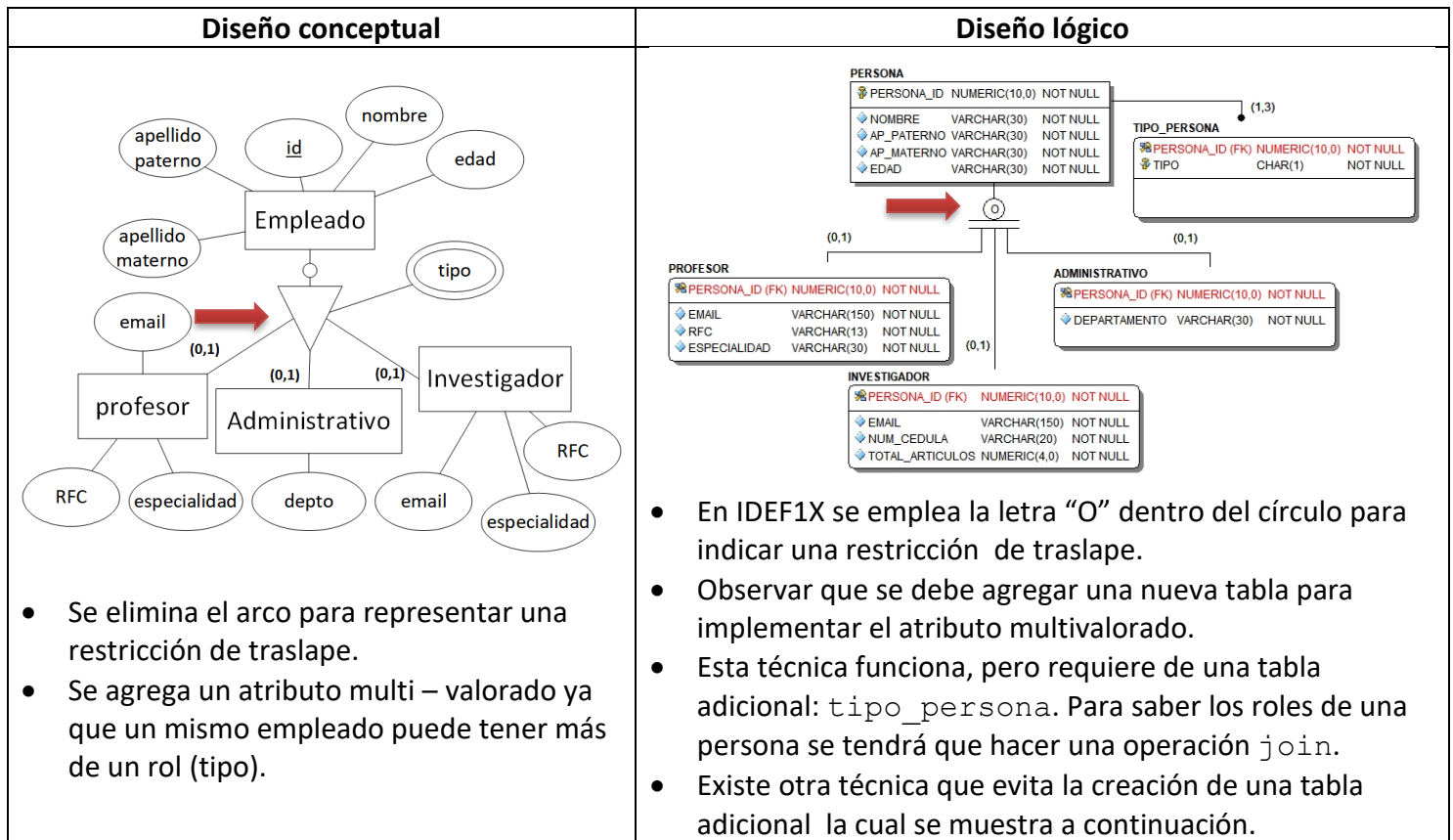
El rol de una persona puede ser opcional. En caso de contar con rol, debe ser solo uno.



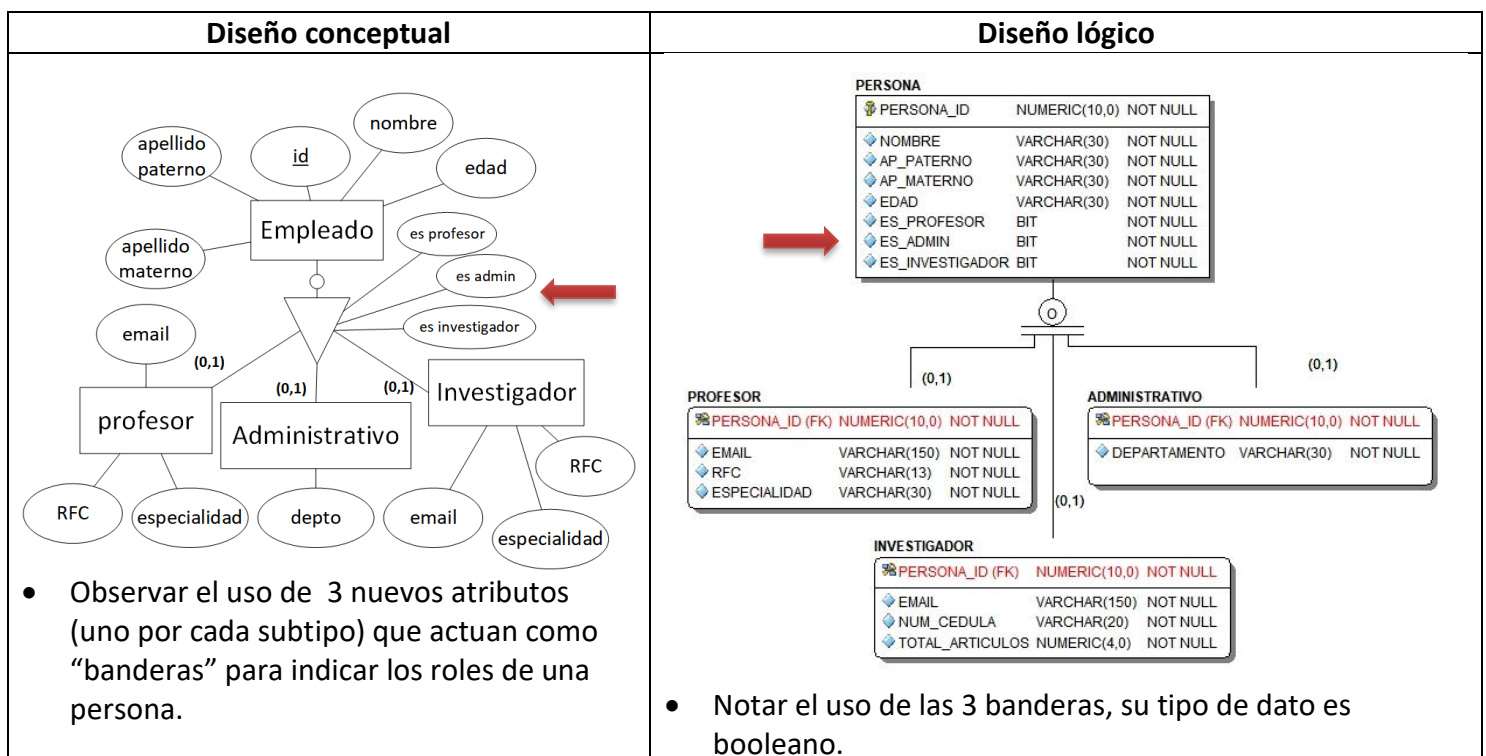


**Traslape / Total (Overlapping / complete):**

Una persona puede tener varios roles, mínimo uno.

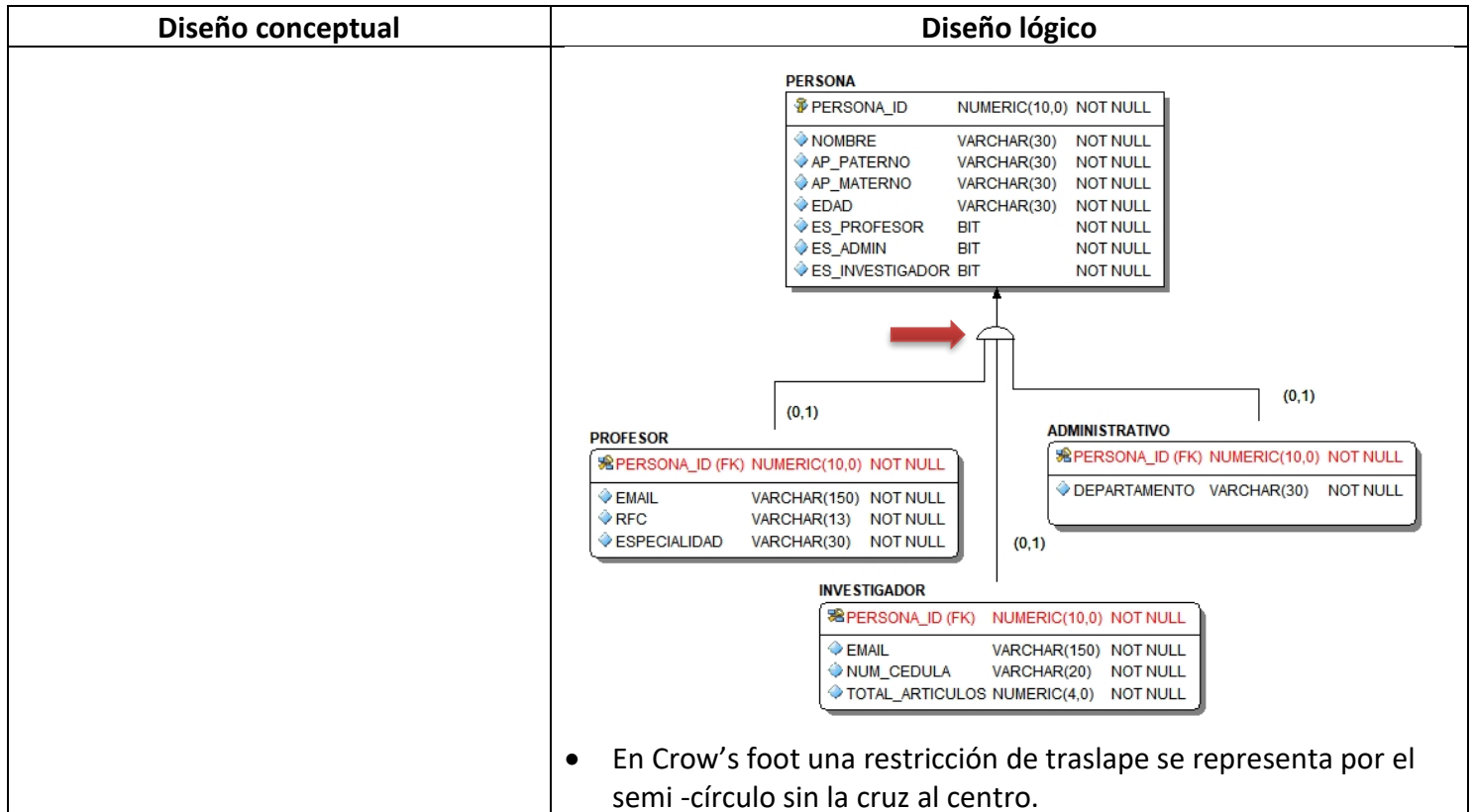


Opción 2: Eliminación de la tabla tipo\_persona.

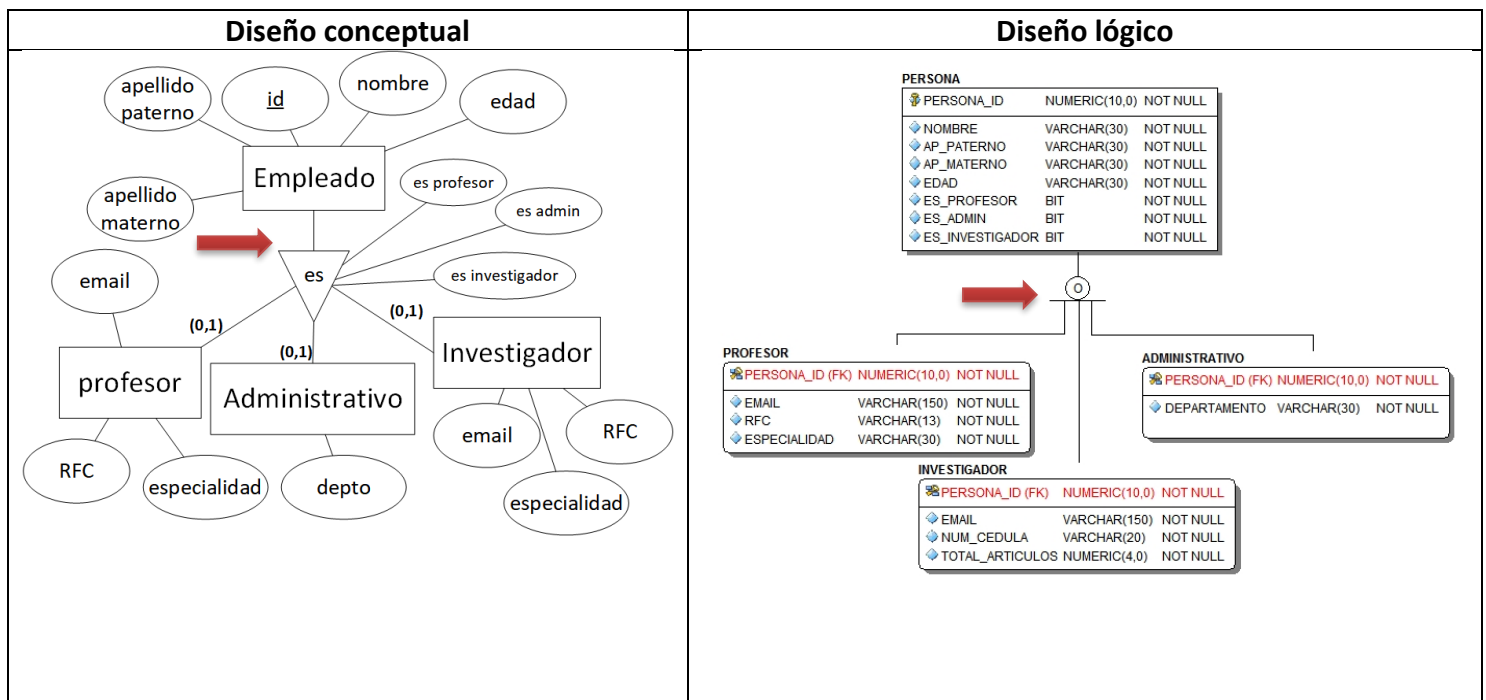


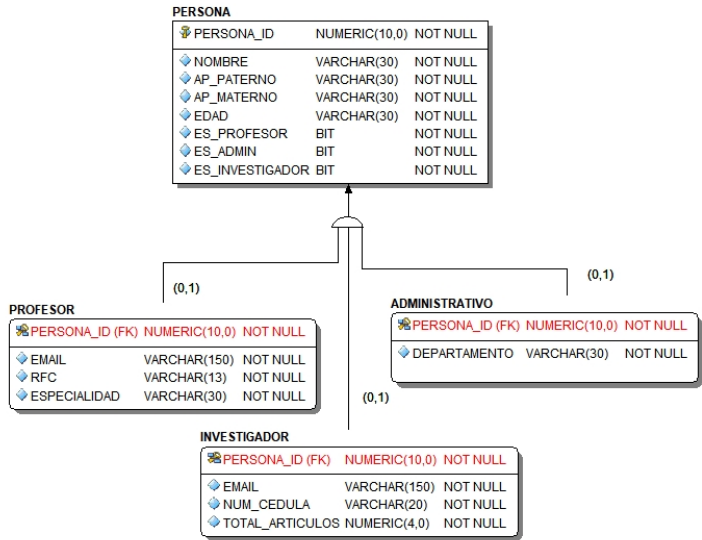


## Representación en Crow's foot

**Traslape / parcial (Overlapping / partial):**

Una persona puede o no tener roles. Pueden ser 0 o varios.



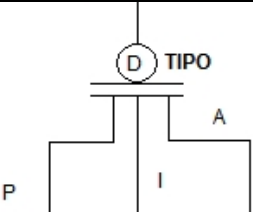
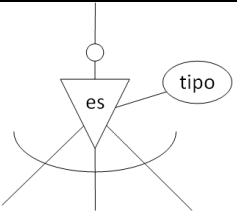
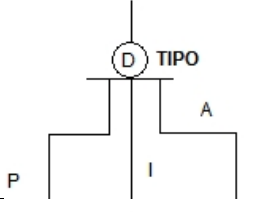
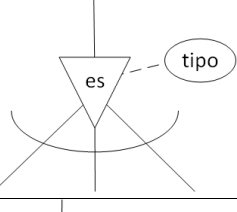
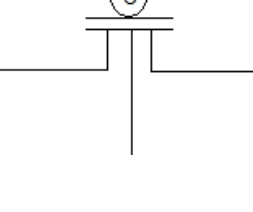
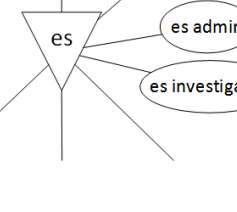
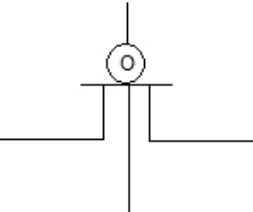
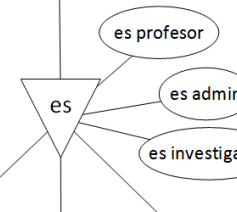
| Diseño conceptual   | Diseño lógico  |
|---|--|
| <ul style="list-style-type: none"> <li>Se elimina el círculo y el arco para indicar una restricción de traslape y parcial.</li> <li>Notar que las banderas siguen siendo obligatorias a pesar de ser una restricción parcial.</li> <li>Si se desea emplear la técnica con el atributo multivalorado, dicho atributo deberá ser opcional.</li> </ul> | <ul style="list-style-type: none"> <li>En IDEF1X se emplea una sola línea paralela para indicar una relación parcial.</li> <li>Se emplea la letra "O" para indicar una restricción de traslape.</li> </ul>   |
| En este caso, se omite el círculo.  |  <ul style="list-style-type: none"> <li>En Crow's foot no existe algún elemento gráfico para indicar esta combinación. El diagrama es el mismo sin importar si la restricción es parcial o total.</li> </ul> |



- Observar que, en todos los casos, la PK del supertipo se propaga hacia los subtipos empleando una relación identificativa.
- La cardinalidad entre una instancia y una instancia de sus subtipos es siempre (0,1) ¿por qué razón?

#### 5.4.4. Resumen de notaciones.

La siguiente tabla muestra el resumen de las diferentes combinaciones con sus respectivas notaciones y características del discriminante para cada caso.

| Tipo de relación  | Diseño lógico   | Diseño conceptual   | Discriminante.   |
|-------------------|---|---|--|
| Exclusiva/total   |    |    | <p>tipo char (1) not null</p> <ul style="list-style-type: none"> <li>• Observar las letras P, I, A, se suele especificar una letra (CHAR) como valor del campo para identificar a cada subtipo, se define como not null al ser una relación Total.</li> <li>• Observar la etiqueta “TIPO” que corresponde con el nombre del campo que actúa como discriminante.</li> </ul> |
| Exclusiva/parcial |    |    | <p>tipo_persona char (1) null</p> <p>Observar que el campo debe ser NULL ya que, en este caso, la persona puede o no tener un rol asignado (relación parcial).</p>   |
| Traslape/total    |   |   | <p>es_admin bool not null<br/>es_profesor bool not null<br/>es_investigador bool not null</p> <ul style="list-style-type: none"> <li>• Observar que en un tipo de relación de traslape se requiere un atributo booleano, uno por cada subtipo.</li> <li>• Si una persona tiene los 3 roles asignados, el valor de los 3 campos será true.</li> </ul>                       |
| Traslape/parcial  |  |  | <p>es_admin bool not null<br/>es_profesor bool not null<br/>es_investigador bool not null</p> <ul style="list-style-type: none"> <li>• Observar que, a pesar de ser una relación parcial, los 3 campos se definen como not null. Si una persona no tiene rol asignado, el valor de los 3 campos será false.</li> </ul>   |

### 5.5. MODELADO DE CATÁLOGOS Y DATOS CON HISTÓRICO.

En algunos casos, existen campos en una tabla que se actualizan con cierta frecuencia, y por cuestiones de control, se desea que se almacene cada uno de estos cambios, la fecha en la que se modificó, etc.

#### Ejercicio en clase 1:



Empleando el concepto de manejo y diseño de entidades e históricos realizar el diseño conceptual y lógico para el siguiente enunciado.

Una agencia de viajes aéreos desea llevar el control de los viajes que son adquiridos por sus clientes. Los datos del viaje son los siguientes:

- Lugar de salida (origen)
- Lugar de llegada (destino)
- Fecha y hora de salida

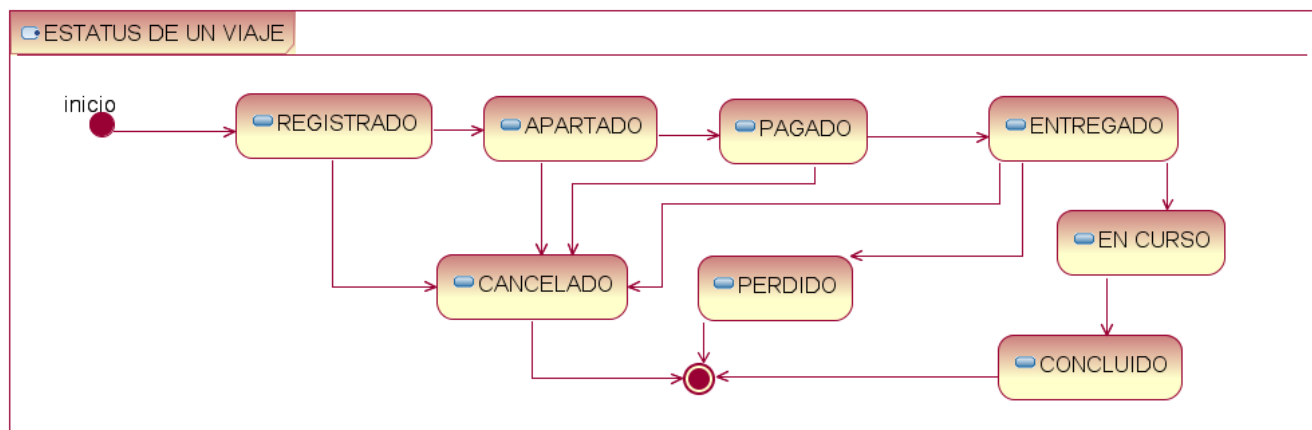
Cada destino y origen contiene una clave única de 3 caracteres, así como el nombre y descripción del lugar.

Ejemplo:

| lugar_id | clave_lugar | descripción    |
|----------|-------------|----------------|
| 1        | AGS         | AGUASCALIENTES |
| 2        | LCB         | LOS CABOS      |
| 3        | CAN         | CANCUN         |



Adicional a esta información, para que un viaje se realice, debe pasar por las siguientes etapas:



1. **REGISTRADO:** El cliente captura en el sistema los datos del viaje: origen, destino, nombre y apellidos de cada pasajero y la fecha de salida. El viaje se considera como REGISTRADO.
2. **APARTADO:** El sistema le da como tiempo máximo 1hr para que el cliente seleccione los números de asientos del avión para cada pasajero y para que el cliente capture los datos de su tarjeta de crédito. Mientras el cliente no proporcione los datos de su tarjeta, y no expire el tiempo antes mencionado, el viaje permanece como APARTADO.
3. **PAGADO:** Un viaje se considera PAGADO, cuando el cliente captura su número de tarjeta de crédito en el sistema, y es validado contra el banco correspondiente. Si el banco autoriza el pago, el viaje se considera como PAGADO. En caso contrario, se le notifica al cliente, y si el cliente no corrige el error, el viaje se considera como CANCELADO.
4. **ENTREGADO:** El cliente deberá acudir como máximo, un día antes a la agencia para recoger sus boletos. En este momento el viaje se considera como ENTREGADO. Si no acude el viaje se CANCELA.
5. **PERDIDO:** Ocurre cuando el cliente ha recogido sus boletos, pero este no se presenta al aeropuerto a tomar su vuelo.
6. **EN CURSO:** El viaje toma este valor cuando se le notifica a la agencia que el cliente ha tomado el vuelo. Permanece en este estado hasta que el viaje termina.
7. **CONCLUIDO.** Un viaje se considera concluido cuando:

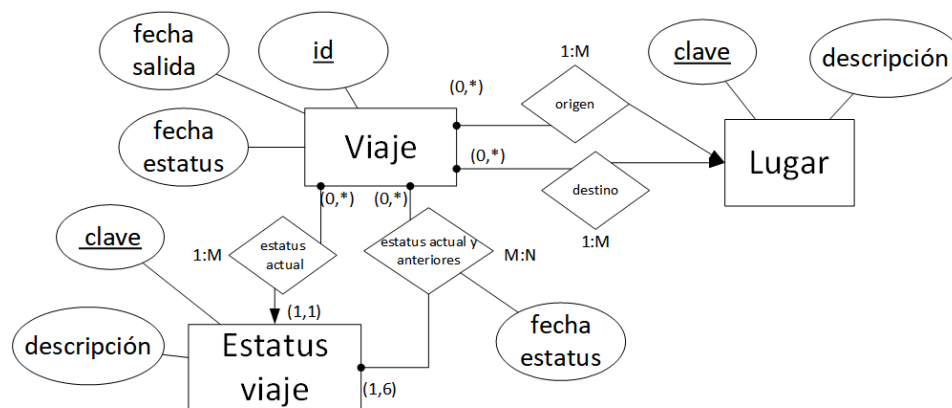
- a. El pasajero ha descendido del avión en el lugar destino para viajes sencillos.
  - b. El pasajero ha descendido del avión en el lugar origen para viajes redondos.
8. CANCELADO. Finalmente, el viaje se puede cancelar por las siguientes causas:
- a. El cliente no captura los datos de su tarjeta dentro del tiempo máximo permitido
  - b. El cliente no pudo comprobar su pago vía tarjeta de crédito.
  - c. La agencia cancela el viaje una vez que este ha sido ENTREGADO por alguna causa de fuerza mayor (fenómeno natural, fallas mecánicas, etc.)

La agencia requiere que se almacene en la base de datos, el estado actual que tiene cada viaje, así como la fecha en la que adquirió dicho estado.

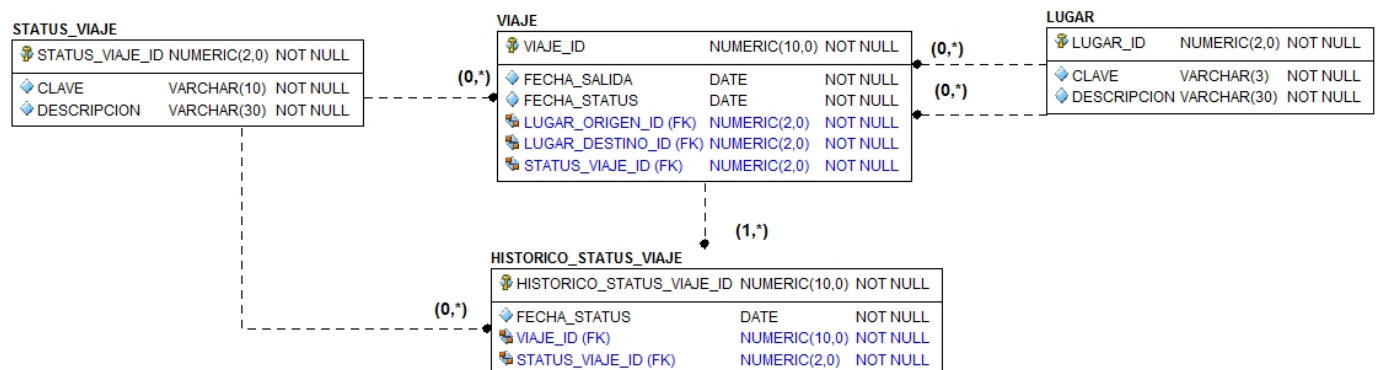
Se requiere adicionalmente almacenar toda la secuencia de estados que ha tenido un viaje, desde su inicio hasta que llegue a su estado terminal.

### Respuesta:

Modelo ER:



El modelo relacional para este ejemplo se muestra a continuación:



#### 5.5.1. Acerca de los catálogos.

- Observar el modelado de la tabla lugar. Se trata de una tabla que contiene los datos del catálogo de lugares. Este catálogo se considera como estático ya que sus valores no son modificados y tampoco se agregan nuevos.
- El uso de catálogos es adecuado cuando sus valores se emplean en varias tablas para mejorar consistencia, adicional a que cada valor del catálogo tiene sus propios atributos: clave y descripción.

- En cuanto a la cardinalidad, observar la participación opcional. Generalmente en una relación entre una entidad con un catálogo, la participación es opcional: la existencia de un elemento del catálogo no requiere la existencia de un registro en la tabla hija.

### 5.5.2. Acerca del histórico.

- Observar que en la tabla viaje se guarda el valor del “status” (`status_viaje_id`) actual del viaje y su fecha (`fecha_status`).
- Observar que estos 2 campos se duplican en `historico_status_viaje`
- Se emplea un catálogo estático para representar a cada uno de los estados del diagrama. A cada valor se le asigna su clave y su descripción. Se prefiere el uso de un catálogo ya que se emplea en 2 tablas.
- La forma en la que trabaja este modelo es la siguiente:
  - Cuando se crea un nuevo viaje, a este se le asocia su primer status y su fecha.
  - Una vez que el viaje ha sido creado, se agrega un nuevo registro en su histórico de status con los mismos valores asignados en la tabla VIAJE.
  - Cuando el viaje cambia de status, se realiza una actualización de los campos `status_viaje_id` y `fecha_status`.
  - Para no perder los valores de los campos actualizados, se agrega una nueva entrada al histórico. De esta forma se guardan los cambios generados para el campo `status_viaje_id`.
- En sentido estricto, ***existe redundancia*** entre los campos `status_viaje_id` y `fecha_status` de la tabla `viaje` con el último registro agregado en el histórico. A pesar de este detalle, el tener los valores en la tabla `viaje`, permite conocer de forma directa el valor actual del status del viaje y la fecha en la que se modificó.
- Notar que en cualquier requerimiento que implique la generación de un histórico, ***se debe considerar un campo de fecha*** para almacenar el instante en el tiempo en el que el campo o campos que requieren histórico se actualiza.

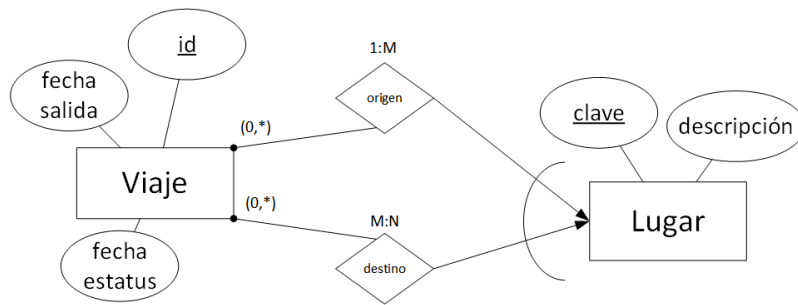
### 5.6. RELACIONES EXCLUSIVAS

- Suponer una entidad A que se relaciona con otras 2 entidades B y C.
- Una relación de exclusión se presenta cuando las instancias de A se deberán asociar con B o con C, pero no con ambas a la vez.
- Visto de otra forma, cada instancia de A participa en una relación con B o con C.

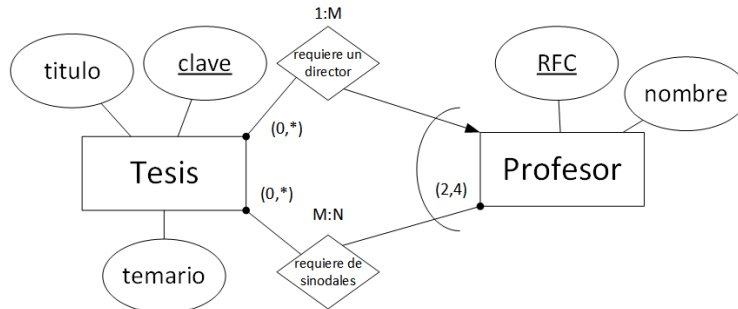
#### Ejemplo:

Considerando el ejemplo de la agencia de viajes anterior, la entidad Viaje se asocia 2 veces con la entidad Lugar. Un viaje tiene un destino y un origen. Cada instancia de Viaje se asocia con una instancia de Lugar para definir el origen del viaje, y con otra distinta para el destino. Lo anterior implica una relación de exclusividad ya que un viaje no puede tener el mismo origen y destino.

En el diseño conceptual la relación de exclusividad se representa el Arco (el mismo empleado en una relación supertipo -Subtipos).

**Ejemplo:**

- Un profesor puede participar como director en una Tesis. De serlo, no podrá participar como sinodal. Es decir, un profesor puede ser sinodal o director de una tesis, pero no ambos.



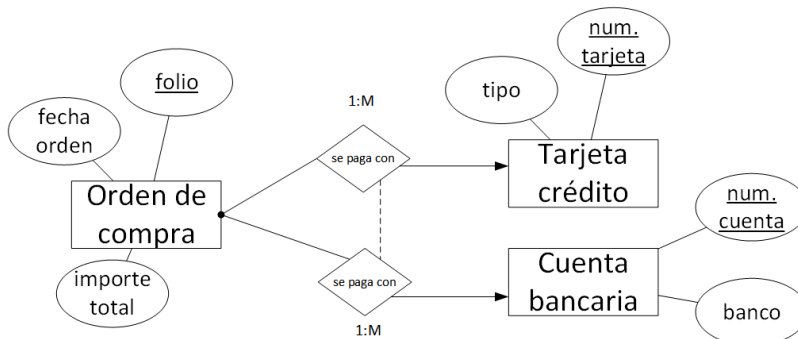
Observar que, en este tipo de relación, la exclusión se revisa del lado de la tabla padre. El arco se dibuja del lado de la tabla padre.

**5.7. RELACIONES DE EXCLUSIVIDAD.**

- En este caso, la relación de exclusividad se revisa del lado de la tabla hija.
- En estas situaciones, una entidad hija C se asocia con 2 entidades A y B. Cada instancia de C se asocia solo con una de las 2 instancias A o B, pero no ambas.
- En el diseño lógico, esto se refleja a través de la ocurrencia de 2 FKs, en donde para cada registro, solo una de ellas tendrá un valor asignado, por lo que ambas FKs deberán estar definidas como NULL

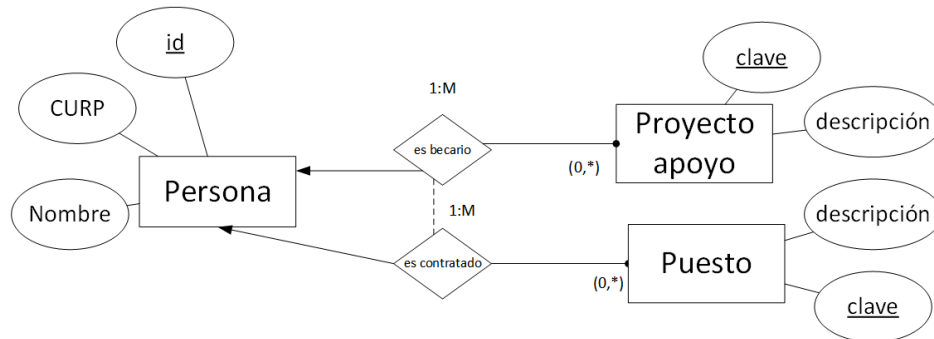
**Ejemplo:**

Una orden de compra puede ser pagada con tarjeta de crédito o a través de una cuenta bancaria, pero no ambas.

**Ejemplo:**

Cuando una persona entra a una empresa, puede ser contratada y por lo tanto se le asigna un puesto, o puede ser considerada como becario(a) por lo que, en lugar de asignarle un puesto, se le asigna a un programa social.





Observar que en este caso se traza una **línea punteada** entre las 2 relaciones para indicar la relación de **exclusividad**.

### 5.8. AGREGACIÓN

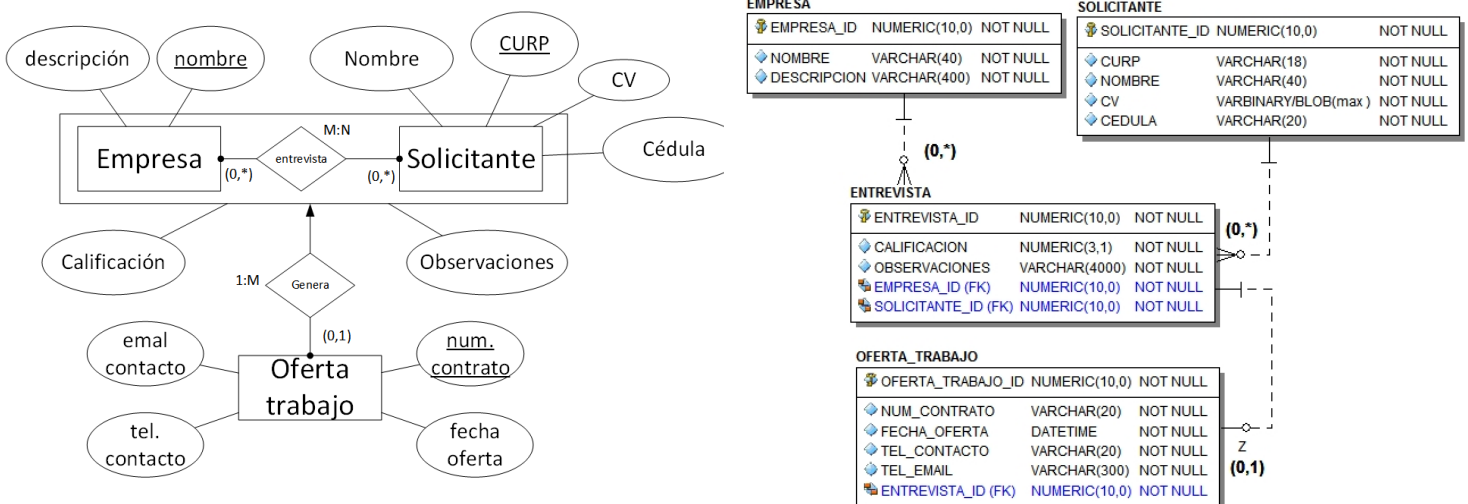
Este tipo de relación se presenta en especial con situaciones como la siguiente:

- Se tiene una relación M:N entre 2 entidades A y B.
- Se tiene una entidad C y se desea relacionarla con el resultado de relacionar a las entidades A y B.
- La condición anterior se puede ver de forma más clara en el diseño lógico: La entidad C se relaciona con la tabla intermedia que se produce al relacionar a las entidades A y B.

#### Ejemplo:

- En una feria de empleo las empresas realizan diversas entrevistas para reclutar a personas llamados solicitantes.
- Cada empresa registra sus datos en la feria: nombre, descripción y tipo.
- Cada solicitante registra sus datos: CURP, nombre, cedula profesional, CV. El solicitante puede participar en diversas entrevistas.
- Del resultado de las entrevistas se almacena la calificación de su examen y un texto que describe las observaciones que cada empresa encontró en el solicitante.
- Cuando el solicitante es aceptado en una empresa, se registra una nueva oferta de trabajo: fecha de la oferta, numero de contrato, email y teléfono del contacto para continuar con el proceso.
- Adicional a lo anterior, a la oferta de trabajo se le asocia el resultado de la entrevista que acredita al solicitante como apto o aprobado para cubrir dicha oferta.

Modelos:



Observar que la relación M:N se trata como una nueva entidad empleando el cuadro que las agrupa. A partir de ella se pueden crear nuevas relaciones con otras entidades.