

Soluciones a los ejercicios ternarios



GRUPO DE ESTRUCTURAS DE DATOS

CLASIFICACIÓN

- Clasificamos los enunciados como:
 - Simples.
 - Monotablas (poco interesantes): Generalmente acompañados de cortes horizontales y/o verticales en la relación: restricción/proyección
 - Multitablas:
 - JOIN
 - Identificación: en el enunciado aparece ‘al menos’ o ‘alguna’ (u otra forma equivalente).
 - Filtrante (se reconoce porque una de las relaciones de la reunión no aporta nada a la cabecera del resultado).
 - MINUS
 - Identificación: en el enunciado aparece ‘negación’ + verbo de la interrelación
 - DIV
 - Identificación: en el enunciado aparece ‘todos’ + verbo de la interrelación
 - Compuestos.
 - Cualquier mezcla de los simples, normalmente multitablas.

Ejercicio 1

al menos una

- Obtener los códigos de los proyectos que usan partes vendidas por el proveedor 'S2'. (pueden o no vender otras)

```
( SPJ JOIN  
  ( SPJ WHERE SN = 'S2' ) [PN]  
) [JN]
```

1. Seleccionamos las partes que vende 'S2'.
2. Obtenemos todas las ventas de las partes que vende 'S2'.
3. Proyectamos obteniendo los códigos de los proyectos.

Ejercicio 2

al menos una

- Obtener los códigos de los proveedores que venden partes vendidas por el proveedor 'S2'. (pueden o no vender otras)

```
( SPJ JOIN  
  ( SPJ WHERE SN = 'S2' ) [PN]  
) [SN]
```

1. Seleccionamos las partes que vende 'S2'.
2. Obtenemos todas las ventas de las partes que vende 'S2'.
3. Proyectamos obteniendo los códigos de los suministradores.



GRUPO DE ESTRUCTURAS DE DATOS

Puntos de articulación

- Aparecen en enunciados en lenguaje natural en los que se usa mas de un verbo relativo a la misma interrelación.
 - Los ejercicios 1 y 2 han sido ejemplos de enunciados articulados
 - Otro ejemplo: *Obtener los códigos de los proveedores que venden las partes usadas por el proyecto 'J1'.*
 - Solución algebraica incorrecta no-articulada: ~~(SPJ WHERE JN='J1') [SN]~~
- La solución no-articulada es incorrecta porque:
 - Si se diesen las siguientes tuplas en SPJ:
 - S1 P2 J1
 - S2 P2 J3
 - S3 P3 J3
 - El resultado debería obtener 'S1' y 'S2', pero solo obtiene 'S1'
 - En rosa la primera tupla completa con la que se detecta 'S1' en la solución algebraica no-articulada incorrecta mostrada anteriormente.
 - En verde se muestran las articulaciones que deberían permitir detectar 'S1' y 'S2' en una solución algebraica articulada correcta.
- La solución algebraica correcta debe construirse articulando, sobre una reunión, dos ocurrencias de la misma interrelación SPJ:
 - ((SPJ WHERE JN='J1')[PN] JOIN SPJ)[SN]

Punto de articulación

Ejercicio 3

➤ Obtener los códigos de los proveedores que no venden partes vendidas por el proveedor 'S2'.

S[SN]	MINUS		
	(SPJ JOIN		
	(SPJ WHERE SN='S2')	[PN])	[SN]

1. Seleccionamos las ventas de 'S2'.
2. Obtenemos las partes que vende S2.
3. Obtenemos todas las ventas de las partes que vende 'S2'.
4. Obtenemos los suministradores que suministran alguna parte de las que vende S2.
5. Obtenemos los códigos de todos los suministradores.
6. Quitamos los códigos obtenidos en 4.

Ejercicio 4

al menos una

➤ Obtener los códigos de los proveedores que venden partes vendidas por algún proveedor que no sea 'S2'.

```
( SPJ JOIN  
  ( SPJ WHERE SN < > 'S2' ) [PN]  
)[SN]
```



GRUPO DE ESTRUCTURAS DE DATOS

Ejercicio 5

➤ Obtener los códigos de los proveedores que venden partes no vendidas por el proveedor 'S2'.

```
( SPJ JOIN
  ( P[PN] MINUS
    ( SPJ WHERE SN = 'S2' ) [PN]
  )
)[SN]
```




Grupo de ESTRUCTURAS de DATOS

Ejercicio 6

➤ Obtener los códigos de los proveedores que no venden partes no vendidas por el proveedor 'S2'.

```
S[SN] MINUS
( SPJ JOIN
  ( P[PN] MINUS
    ( SPJ WHERE SN = 'S2' )[PN]
  )
)[SN]
```

Diferencias implícitas

- Existe un tipo especial de diferencias que no se pueden detectar en el enunciado en lenguaje natural mediante el método clásico de buscar un verbo asociado a una interrelación que aparezca negado.
 - Véanse las páginas 2 y 3 del solucionario de la entrega 1 de álgebra relacional:
<https://cv-aep.ulpgc.es/cv/ulpgctp20/mod/resource/view.php?id=192437>
- De ahora en adelante, a estas diferencias detectadas en el enunciado por el susodicho '*método clásico*' se las denominará **diferencias explícitas**.
- Al nuevo tipo especial de diferencias que no pueden ser detectadas por el método clásico aprendido en la referencia anteriormente citada, se las denominará **diferencias implícitas** y constituyen la materia sobre la que versarán las explicaciones que siguen.

Diferencias implícitas

En primer lugar, ¿cómo se detecta una diferencia implícita en un enunciado en lenguaje natural?

- En determinados enunciados en castellano aparece un **verbo asociado a una interrelación** acompañado de expresiones como:
 - **solo,**
 - **solamente,**
 - **exclusivamente,**
 - **en exclusiva,**
 - **de forma exclusiva,**
 - **nada mas que,**
 - o cualesquiera otras **expresiones sinónimas** que existan en dicha lengua.
- Ejemplos típicos podrían ser:
 - que venden solo tales partes,
 - que solamente usan tales partes,
 - que son vendidas exclusivamente por tales proveedores,
 - que son usadas en exclusiva en tales proyectos,
 - que le venden nada más que a tales proyectos,...
- Todos ellos son **enunciados que contienen una diferencia implícita.**



GRUPO DE ESTRUCTURAS DE DATOS

Diferencias implícitas

En segundo lugar, ¿cómo se resuelven en Álgebra Relacional?

Paso 1: Transformación del enunciado en lenguaje natural

- Una vez detectada en lenguaje natural la existencia de una diferencia implícita,
 - y siempre antes de resolver la expresión algebraica correspondiente,
- habrá que **transformar el enunciado** para poder visualizar su estructura.
- Para mejor comprensión, se escogerá un enunciado
 - simple y
 - paradigmático para una diferencia implícita en nuestra semántica habitual
- huyendo de combinar el caso que nos ocupa con otras operaciones diferentes.
- La transformación que se va a llevar a cabo constituye el paso más importante en la resolución de una diferencia implícita y es el que con más frecuencia induce a error al novicio
- **Enunciado original:** '*Obtener los códigos de los proveedores que solo venden (partes cuyo código no es p2)*'
 - Equivale a: '*Obtener los códigos de los proveedores que no venden lo contrario a (partes cuyo código no es p2)*'
- O sea, el **enunciado transformado** será: '*Obtener los códigos de los proveedores que no venden (la parte cuyo código es p2)*'
- Basta con entender el castellano para que la transformación se reduzca a una simple problema de lógica:
 - **Los proveedores que solo venden partes distintas de p2 son los proveedores que no venden p2.**
- Generalizando: se supone que
 - existe: un verbo de una interrelación acompañado de solo o de cualquier expresión sinónima (*solo venden*)
 - y a continuación un argumento restrictivo a cumplir por el complemento directo (*partes cuyo código no es p2*).
- Así que:
 - se sustituye por una negación explícita de dicho verbo (*no venden*)
 - y se invierte el argumento restrictivo poniendo el contrario (*la parte cuyo código es p2*).

Diferencias implícitas

Paso 2: Selección del minuendo

- Una vez transformado el enunciado se puede observar su estructura;
 - en este caso, se trata de **una sencilla diferencia explícita**:
 ~~$s[sn] \text{ minus } (spj \text{ where } pn='p2')[sn]$~~
 - pero con un error importante.
- La solución expuesta está mal porque es **estrictamente la del enunciado transformado** que se acaba de usar para poder resolver el ejercicio.
- Pero no se está resolviendo el enunciado transformado, **sino el original**.
- Y en el original, y en general **en los enunciados con diferencia implícita**, los **proveedores que no venden nada no deberían estar en el resultado**.
- Por ello, se dice que el **minuendo** de una **diferencia implícita** no está en la entidad ($s[sn]$, en este ejemplo) como ocurre en las diferencias explícitas, sino que **está en la interrelación** ($spj[sn]$).
- Así pues, la **solución correcta** sería:
- $spj[sn] \text{ minus } (spj \text{ where } pn='p2')[sn]$



GRUPO DE ESTRUCTURAS DE DATOS

Diferencias implícitas

- Ahora otro ejemplo, ya de forma abreviada, con una condición restrictiva conteniendo un verbo de una interrelación
- **Enunciado original:**
*Obtener los códigos de las partes que **se usan nada mas que** en (proyectos a los que **les vende** el proveedor cuyo código es s1)*
- **Enunciado transformado:**
*Obtener los códigos de las partes que **no se usan** en (proyectos a los que **no les vende** el proveedor cuyo código es s1)*
- **Solución algebraica:**
- **spj[pn] minus ((j[jn] minus (spj where sn='s1')[jn]) join spj)[pn]**

Ejercicio 7

Obtener los códigos de los proveedores que solo venden partes
vendidas por el proveedor 'S2'.

no

no

```
SPJ[SN] MINUS
( SPJ JOIN
  ( P[PN] MINUS
    ( SPJ WHERE SN = 'S2' )[PN]
  )
)[SN]
```



GRUPO DE ESTRUCTURAS DE DATOS

Ejercicio 9

- Enunciado original:
 - Obtener los códigos de los proveedores que solo venden partes no vendidas por algún proveedor que no sea 'S2'
- Enunciado transformado:
 - Obtener los códigos de los proveedores que no venden partes vendidas por algún proveedor que no sea 'S2'

```
SPJ[SN] MINUS  
( SPJ JOIN  
  ( SPJ WHERE SN ≠ 'S2' )[PN]  
)[SN]
```

Ejercicio 10

- Obtener los códigos de los proveedores que no venden partes vendidas por algún proveedor que no sea 'S2'.

```
S[SN] MINUS (SPJ JOIN (SPJ WHERE SN ≠ 'S2')[PN])[SN]
```


Ejercicio 11

➤ Obtener los códigos de los proveedores que solo venden partes vendidas por algún proveedor que no sea 'S2'.

$SPJ[SN] \text{ MINUS } (SPJ \text{ JOIN } (P[PN] \text{ MINUS } (SPJ \text{ WHERE } SN \neq 'S2') [PN])) [SN]$

Ejercicio 12

➤ Obtener los códigos de los proveedores que no venden partes no vendidas por algún proveedor que no sea 'S2'.

$S[SN] \text{ MINUS } (SPJ \text{ JOIN } (P[PN] \text{ MINUS } (SPJ \text{ WHERE } SN \neq 'S2') [PN])) [SN]$

Ejercicio 13

- Obtener los códigos de los proveedores que venden exclusivamente partes vendidas por el proveedor 'S2'

$SPJ[SN] \text{ MINUS } (SPJ \text{ JOIN } (P[PN] \text{ MINUS } (SPJ \text{ WHERE } SN = 'S2')[PN]))[SN]$

Ejercicio 14

- Enunciado original:
 - Obtener los códigos de los proveedores que venden partes vendidas solamente por (el proveedor 'S2')
- Enunciado transformado:
 - Obtener los códigos de los proveedores que venden partes no vendidas por (los proveedores que no sean 'S2')
- En este caso estamos ante una diferencia implícita de partes (en el segundo verbo) y se resuelve de la misma manera que cuándo la negación implícita está en el primer verbo:

$$(\text{SPJ JOIN } (\text{SPJ}[\text{PN}] \text{ MINUS } (\text{SPJ WHERE SN} \neq \text{'S2'})[\text{PN}]))[\text{SN}]$$
- Esta sería la solución algebraica según lo aprendido sobre las diferencias implícitas.
 - La resta de partes tiene el minuendo en la interrelación, como corresponde a una diferencia implícita, para evitar que en el resultado de dicha resta aparezcan las partes que no han sido vendidas.
- pero obsérvese que el resultado de la diferencia de partes **se somete a continuación a una reunión con SPJ**, de tal forma que sí en el resultado de la resta de partes hubiere alguna parte no vendida, la reunión con SPJ la haría desaparecer del resultado de la reunión y consecuentemente la solución algebraica que se expone a continuación también sería perfectamente válida:

$$(\text{SPJ JOIN } (\text{P}[\text{PN}] \text{ MINUS } (\text{SPJ WHERE SN} \neq \text{'S2'})[\text{PN}]))[\text{SN}]$$

Ejercicio 15

- Enunciado original:
 - Obtener los códigos de los proveedores que venden solamente (partes vendidas exclusivamente por el proveedor 'S2')
 - En este caso aparecen dos diferencias implícitas ¿Qué hacer?
 - Se ignora la segunda por ahora y se transforma el enunciado original, según lo aprendido, para la primera.
 - Equivale a:
 - Obtener los códigos de los proveedores que no venden lo contrario de (partes vendidas exclusivamente por el proveedor 'S2')
- Enunciado transformado:
 - Obtener los códigos de los proveedores que no venden (partes vendidas por los proveedores que no sean 'S2')
 - Obsérvese que valió la pena ignorar la segunda diferencia implícita, porque ahora ha desaparecido.
- Solución algebraica:

$$SPJ[SN] \text{ MINUS } (SPJ \text{ JOIN } (SPJ \text{ WHERE } SN \neq 'S2')[PN])[SN]$$

Ejercicio 16

- Enunciado original:
 - Obtener los códigos de los proveedores que solo venden (partes vendidas exclusivamente por proveedores que no son 'S2')
 - También en este caso aparecen dos diferencias implícitas.
 - Se ignora la segunda por ahora y se transforma el enunciado original, según lo aprendido, para la primera.
 - Equivale a:
 - Obtener los códigos de los proveedores que no venden lo contrario de (partes vendidas exclusivamente por proveedores que no son 'S2')
- Enunciado transformado:
 - Obtener los códigos de los proveedores que no venden (partes vendidas por el proveedor 'S2')
 - Obsérvese que volvió a valer la pena ignorar la segunda diferencia implícita, porque ahora también ha desaparecido.
- Solución algebraica:

$$SPJ[SN] \text{ MINUS } (SPJ \text{ JOIN } (SPJ \text{ WHERE } SN='S2'))[PN]][SN]$$

Identificación del cociente en el dividendo

- A continuación se repasa brevemente el problema que puede provocar no incluir en el dividendo la identificación del cociente
 - y que ya se expuso en la ‘Lección a aprender 4’ del documento PDF Solucionario Entrega 1 de Álgebra Relacional:

<https://cv-aep.ulpgc.es/cv/ulpgctp20/mod/resource/view.php?id=192437>

- a fin de que lo tengan presente para aquellos ejercicios que puedan necesitar el operador de división de entre los que siguen.
- Enunciado: Obtener los nombres de los proveedores que venden todas las partes.
- Solución algebraica incorrecta: $((SPJ \text{ JOIN } S)[SNOMBRE, PN]) \text{ DIV } P[PN]$
- Soluciones algebraicas correctas:

$$((SPJ[SN, PN] \text{ DIV } P[PN]) \text{ JOIN } S)[SNOMBRE]$$

y

$$(((SPJ \text{ JOIN } S)[SN, SNOMBRE, PN]) \text{ DIV } P[PN])[SNOMBRE]$$

- En la incorrecta se obtienen los nombres de los proveedores que, **entre todos los tocayos**, venden todas las partes.
 - Se ha perdido el identificador del proveedor.
- Generalizando: **Inclúyase siempre en la proyección del dividendo un identificador del cociente pretendido.**
 - Recuerdese que en el cociente se pretende tener una fila por cada proveedor que venda todas las partes, así que posibles identificadores correctos del cociente pretendido a incluir en el dividendo serían:
 - sn (simple, como en la primera solución correcta)
 - (sn, snombre) (compuesto, como en la segunda solución correcta)

Proyección en el dividendo

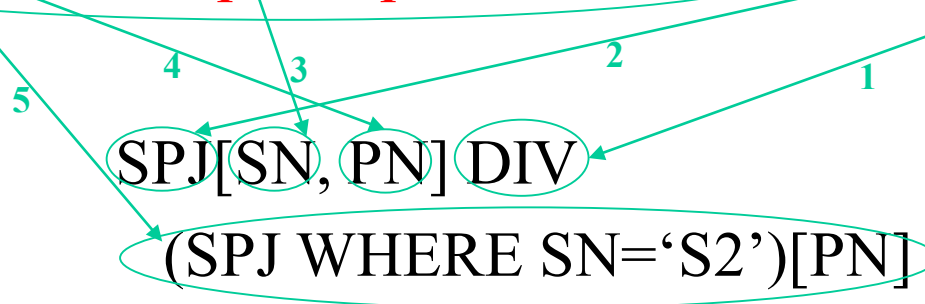
- A continuación se repasa brevemente el problema que puede provocar no proyectar adecuadamente el dividendo
 - y que ya se expuso en la ‘Lección a aprender 6’ del documento PDF Solucionario Entrega 1 de Álgebra Relacional:
<https://cv-aep.ulpgc.es/cv/ulpgctp20/mod/resource/view.php?id=192437>
 - a fin de que lo tengan presente para aquellos ejercicios que puedan necesitar el operador de división de entre los que siguen.
- Enunciado: Obtener los códigos de los proveedores que venden todas las partes.
- Solución algebraica correcta: $SPJ[SN, PN] \text{ DIV } P[PN]$
- Solución algebraica incorrecta: $(SPJ \text{ DIV } P[PN])[SN]$

Proyección en el dividendo

- Problema:
 - Si
 - $P[PN] = \{ 'P1', 'P2' \}$ y
 - $SPJ[SN, PN, CANT] = \{ ('S1', 'P1', 100), ('S1', 'P2', 100) \}$
 - Entonces ambas soluciones funcionan.
 - Pero si
 - $P[PN] = \{ 'P1', 'P2' \}$ y
 - $SPJ[SN, PN, CANT] = \{ ('S1', 'P1', 100), ('S1', 'P2', 200) \}$
 - La segunda **NO** funciona.
- Generalizando: **Proyéctese siempre el dividendo para que sólo incluya:**
 - **la cabecera completa del cociente**
 - que evidentemente constituirá un identificador (simple o compuesto) del cociente, y
 - **la cabecera completa del divisor,**
 - que evidentemente constituirá un identificador (simple o compuesto) del divisor.

Ejercicio 17

- Obtener los códigos de los proveedores que venden todas las partes vendidas por el proveedor 'S2'.



- El diagrama ilustra la forma de identificar en lenguaje natural las partes de la solución algebraica:
 - Se trata de una división
 - El dividendo es la relación SPJ
 - La cabecera del cociente es SN
 - La cabecera del divisor es PN
 - El divisor es $(SPJ \text{ WHERE } SN='S2')[PN]$



GRUPO DE ESTRUCTURAS DE DATOS

Ejercicio 18

- Obtener los códigos de los proveedores que venden todas las partes no vendidas por el proveedor 'S2'.

SPJ[SN, PN] DIV
(P[PN] MINUS
(SPJ WHERE SN='S2')[PN]
)

Ejercicio 19

- Obtener los códigos de los proveedores que venden todas las partes vendidas solamente por el proveedor 'S2'.

$$\begin{aligned}
 &SPJ[SN, PN] \text{ DIV} \\
 & \quad (SPJ[PN] \text{ MINUS} \\
 & \quad \quad (SPJ \text{ WHERE } SN \neq 'S2')[PN] \\
 & \quad)
 \end{aligned}$$

- Una **advertencia comparativa** con el ejercicio 14:
 - El minuendo de la resta implícita de partes del divisor **ha de ser obligatoriamente spj[pn]**
 - dado que dicha resta del divisor no está seguida en el ejercicio 19 de una reunión con la interrelación spj
 - como sí que ocurría en el caso del ejercicio 14, con lo cual se eliminaban las posibles partes no vendidas que pudieran aparecer en el resultado de la resta si se usaba el minuendo p[pn]
 - Si en el ejercicio 19 se usase el minuendo p[pn] y existiesen partes no vendidas
 - se causaría que ningún proveedor apareciera en el dividendo con todas las partes del divisor,
 - dado que en el divisor existirían partes no vendidas
 - y por tanto **el cociente estaría vacío**

Ejercicio 20:

Dividendo ternario

Divisor unario

Cociente binario

- Obtener los códigos de los proveedores que venden alguna parte a todos los proyectos
- No siempre aparece tan clara la estructura de la división en el enunciado en lenguaje natural,
 - sobre todo cuando el dividendo es ternario.
- Transformando el enunciado original con un poco de inspiración:
- Obtener las parejas de códigos de proveedores y partes que aparecen en ventas con todos los proyectos:
 - La cabecera del cociente es (sn, pn)
 - El divisor es el conjunto de todos los proyectos
 - La interrelación implicada es la de las ventas y se ha de proyectar en (sn, pn, jn):
 - (sn, pn) por ser la cabecera del cociente
 - junto con jn, por ser la cabecera del divisor
- Finalmente proyectamos las susodichas parejas en sn para averiguar que proveedores aparecen en ellas.

$$(SPJ[SN, PN, JN] \text{ DIV } J[JN])[SN]$$

Ejercicio 21:

Dividendo ternario

Divisor binario

Cociente unario

- Obtener los códigos de los proveedores que venden todas las partes a todos los proyectos
- Nuevamente se presenta un dividendo ternario.
- Transformando el enunciado original con otro poco de inspiración:
- Obtener los códigos de proveedores que aparecen en ventas con todas las parejas posibles de partes y proyectos:
 - La cabecera del cociente es (sn)
 - El divisor es el producto cartesiano de todas las posibles partes y todos los posibles proyectos (p[pn] times j[jn])
 - La interrelación implicada es la de las ventas y se ha de proyectar en (sn, pn, jn):
 - sn por ser la cabecera del cociente
 - junto con (pn, jn), por ser la cabecera del divisor

$SPJ[SN, PN, JN] \text{ DIV } (P[PN] \text{ TIMES } J[JN])$

Ejercicio 21:

División cuyo dividendo es otra división con

Dividendo ternario

Divisor unario

Cociente binario

- Obtener los códigos de los proveedores que venden todas las partes a todos los proyectos
- Transformando el enunciado original con otro poco más de inspiración:
- Obtener como cociente de la primera división las parejas de códigos de proveedores y partes que aparecen en ventas con todos los proyectos:
 - Nótese que es la misma división que la del ejercicio 20
 - La cabecera del cociente es (sn, pn)
 - El divisor es el conjunto de todos los proyectos
 - La interrelación implicada es la de las ventas y se ha de proyectar en (sn, pn, jn):
 - (sn, pn) por ser la cabecera del cociente
 - junto con jn, por ser la cabecera del divisor
- Ahora, el cociente de la primera división será el dividendo de la segunda:
 - La cabecera del cociente es (sn)
 - El divisor es el conjunto de todas las partes
 - La interrelación implicada es la de las ventas y se ha de proyectar en (sn, pn):
 - sn por ser la cabecera del cociente
 - junto con pn, por ser la cabecera del divisor
- O sea, basta con pedirle a los proveedores del ejercicio 20 que logren su objetivo con todas las partes posibles.

$$(\text{SPJ}[\text{SN}, \text{PN}, \text{JN}] \text{ DIV } \text{J}[\text{JN}]) \text{ DIV } \text{P}[\text{PN}]$$

- Por supuesto, esta otra solución algebraica también es correcta:

$$(\text{SPJ}[\text{SN}, \text{PN}, \text{JN}] \text{ DIV } \text{P}[\text{PN}]) \text{ DIV } \text{J}[\text{JN}]$$

Ejercicio 22:

División cuyo divisor es otra división

- Obtener los códigos de los proveedores que venden todas las (partes usadas en todos los proyectos)
 - Lo cual no plantea especiales problemas:

$$SPJ[SN,PN] \text{ DIV } (SPJ[PN, JN] \text{ DIV } J[JN])$$

Ejercicio 23

- Obtener los códigos de los proyectos que no usan partes vendidas por proveedores de Telde

J[JN] MINUS

(SPJ JOIN

(SPJ JOIN

(S WHERE CIUDAD='TELDE')[SN]

)[PN]

)[JN]

Ejercicio 24

- Enunciado original: Obtener los códigos de las partes usadas solamente en (proyectos a los que les vende algún proveedor de Telde)
- Enunciado transformado: Obtener los códigos de las partes no usadas en (proyectos a los que no les vende algún proveedor de Telde)

```

SPJ[PN] MINUS
  (SPJ JOIN
    (J[JN] MINUS
      (SPJ JOIN
        (S WHERE CIUDAD='TELDE')[SN]
      )[JN]
    )
  )[PN]

```

Ejercicio 25:

División cuyo divisor es un doble minus

- Enunciado original: Obtener los códigos de los proveedores que venden todas las (partes no usadas en proyectos a los que no les vende un proveedor de Telde)

```

SPJ[SN,PN] DIV
(P[PN] MINUS
  (SPJ JOIN
    (J[JN] MINUS
      (SPJ JOIN
        (S WHERE CIUDAD='TELDE')[SN]
      )[JN]
    )
  )
)[PN]
)

```

Ejercicio 26:

Diferencia cuyo sustraendo es una división

- Obtener los códigos de las partes no usadas en todos los proyectos.

$P[PN] \text{ MINUS } (SPJ[PN, JN] \text{ DIV } J[JN])$