Debilidad:

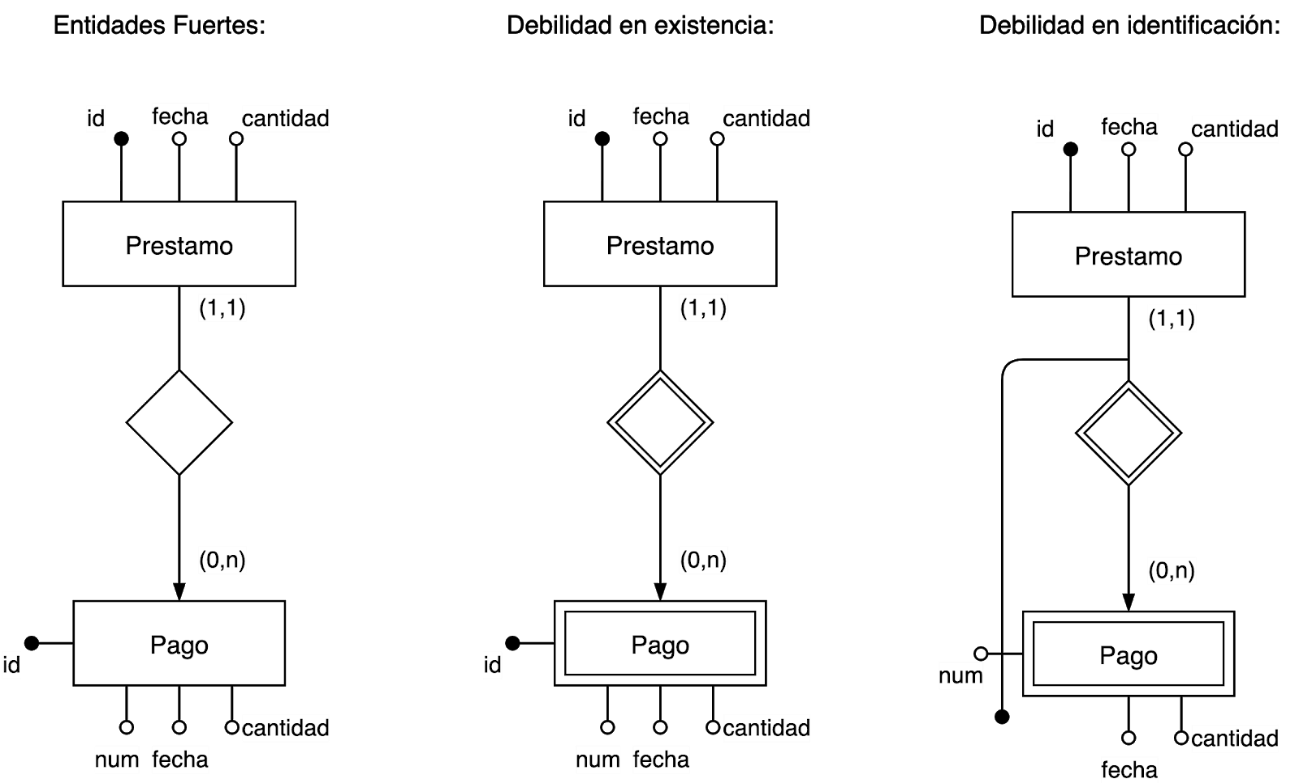
Una entidad débil es aquella cuya existencia depende de otra (considerada su entidad fuerte). Se trata de entidades totalmente supeditadas a otras, de modo que si un ejemplar de la entidad fuerte desaparece, todos los ejemplares de la entidad débil relacionados, desaparecerán también del sistema.

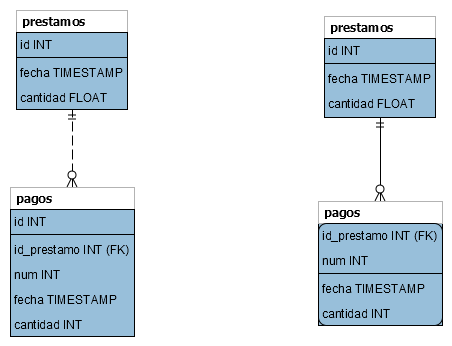
Las entidades débiles ocurren cuando hay una entidad más fuerte de la que dependen, en el sentido de que la propia existencia de la entidad débil está supeditada a la existencia de su entidad fuerte. Lógicamente tienen relación con esa entidad, y es esa relación la que marca el hecho de que una entidad es débil y la otra fuerte.

---------------

Supongamos que tenemos una entidad que registra préstamos realizados y otra para los pagos realizados para devolver ese préstamo. Se almacenará el número de cada pago (1er pago del préstamo, 2º pago del préstamo, etc)  
Omitiremos los datos del receptor del préstamo (sería otra entidad interrelacionada con préstamo).

Podemos modelarlo a nivel conceptual de 3 maneras:

[](https://app.diagrams.net/?page-id=GlJs3QuqFLX-6FHA9nxm&scale=auto#G1jYnYkJyBwsNtxmkupUbbgMhflO2Y8prk)

Entidades fuertes Debilidades en existencia y en identificación  
  


**Entidades fuertes**: considerando dos entidades fuertes:  
La tabla pagos tiene un id convencional autoincrementado.

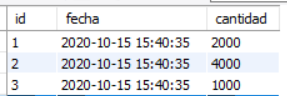
**Debilidades en existencia**: Igual que la anterior, pero “Pago” es una entidad débil dependiente de préstamo. Si un préstamo se elimina, todos los pagos de ese préstamo también se eliminan.

Ambas se convierten a relacional del mismo modo. La debilidad en existencia se implementará en SQL mediante la regla de Integridad Referencial “ON DELETE CASCADE” (la veremos más adelante)

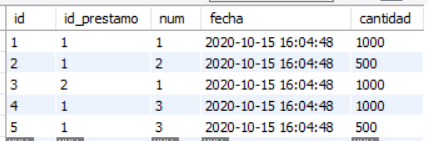
**Debilidad en identificación**: Se considera que un préstamo no puede estar identificado solo por sus atributos, sino que requiere el identificador de la entidad fuerte de la que depende.

El id de pagos desaparece, y la PK es compuesta por id\_prestamo (que también continúa siendo FK) junto con num (que pasará a considerarse “atributo discriminador” ya que discrimina entre los distintos pagos de un único préstamo. El conjunto de ambas identifica unívocamente cada pago.

Si tomamos por ejemplo esta tabla para los préstamos:



En los dos primeros casos, nada evitaría que pudiésemos repetir un pago con el mismo numero de préstamo (num) al mismo préstamo (id\_prestamo), como se puede ver aquí en la última fila:



Utilizando una relación de identificación, la columna id no existiría, y la última fila no se podría insertar. MySQL devolvería un error ya que la PK estaría duplicada.

[Implementación](https://www.edu.xunta.gal/centros/iesteis/aulavirtual/pluginfile.php/44948/mod_resource/content/1/debilidad_prestamos_pagos.sql)