

CONCEPTOS DOMAIN DRIVEN DESIGN

MAPEO DE CONTEXTOS

Daniel Blanco Calviño

DEPENDENCIAS ENTRE DISTINTOS CONTEXTOS

A pesar de que es positivo separar nuestro modelo en contextos acotados, la lógica de un sistema software complejo implica **interacción entre los distintos contextos**.

- Los contextos no son completamente independientes.
- Debemos tener clara la **interacción y dependencias** entre los mismos.

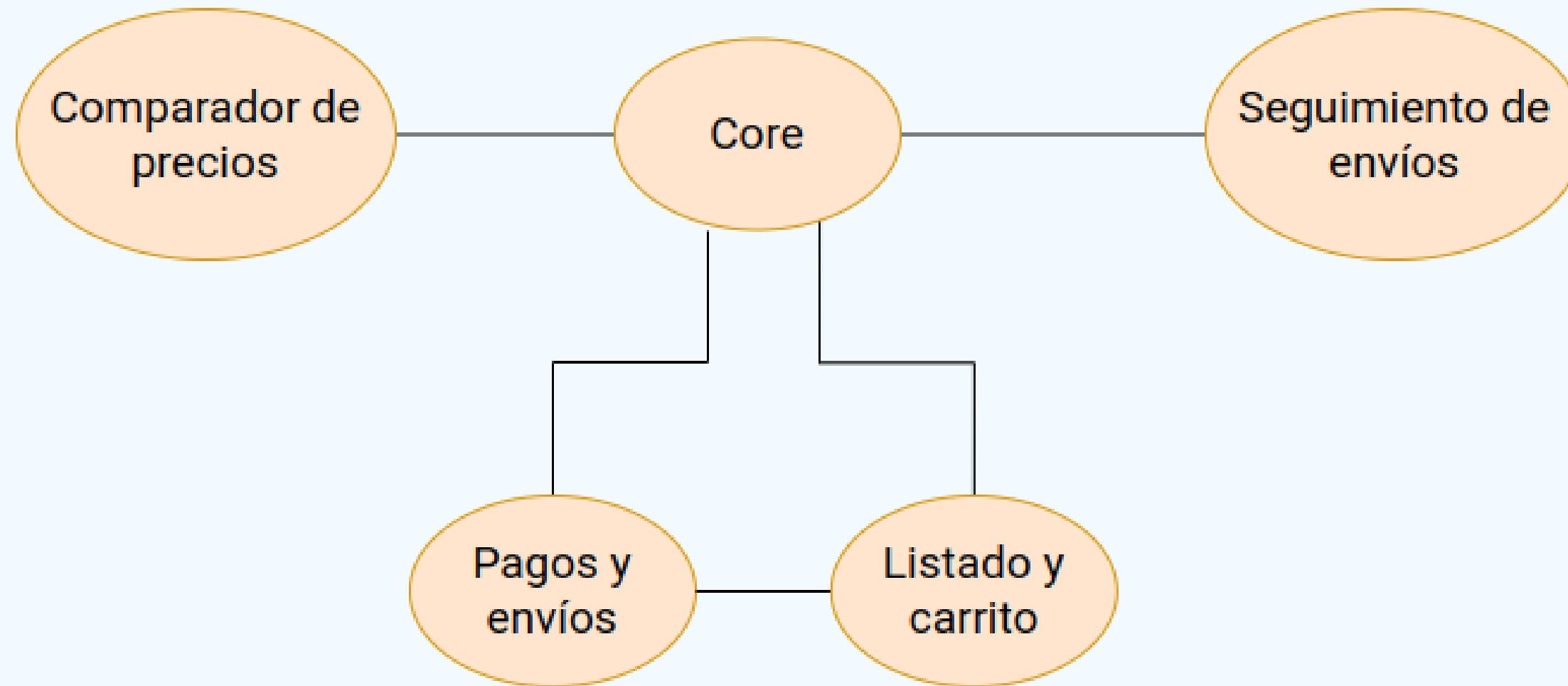
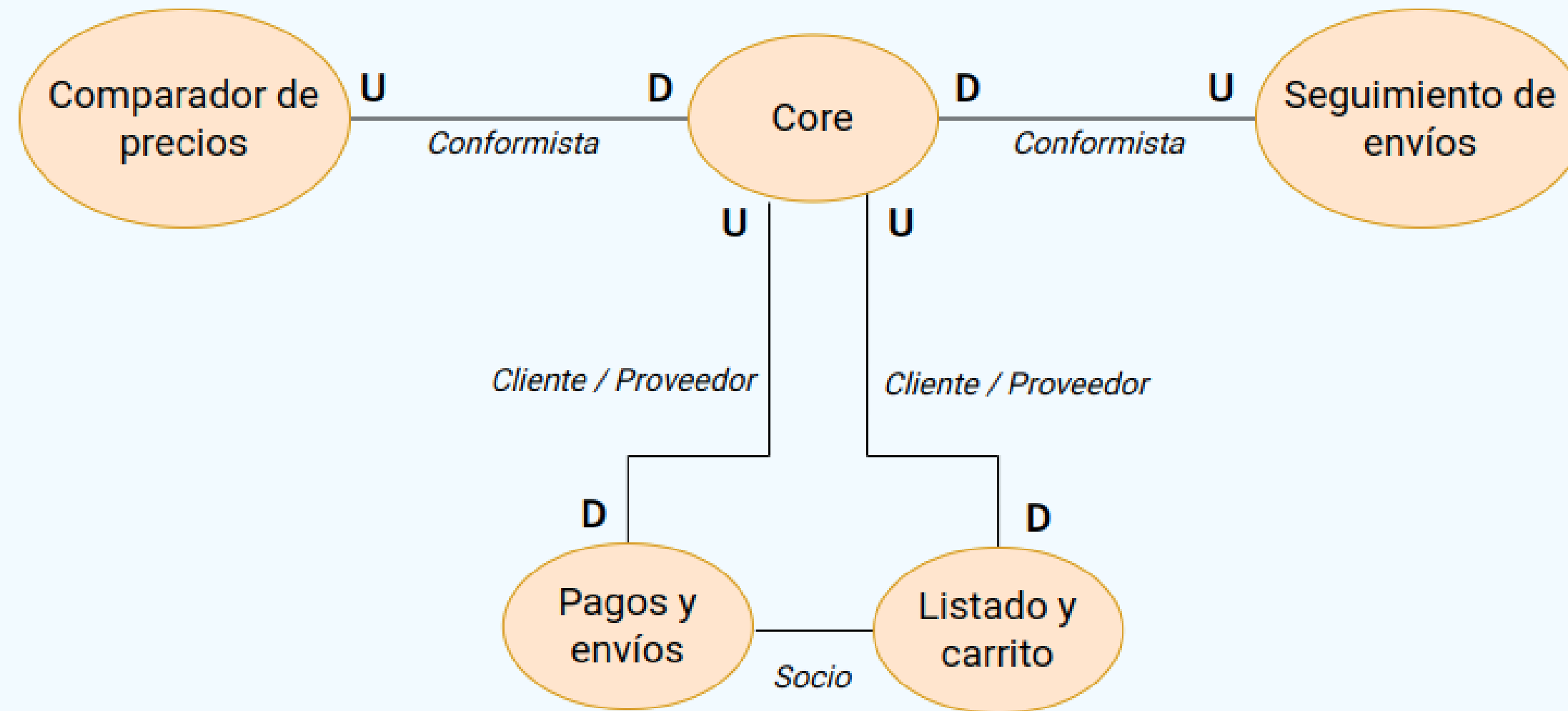


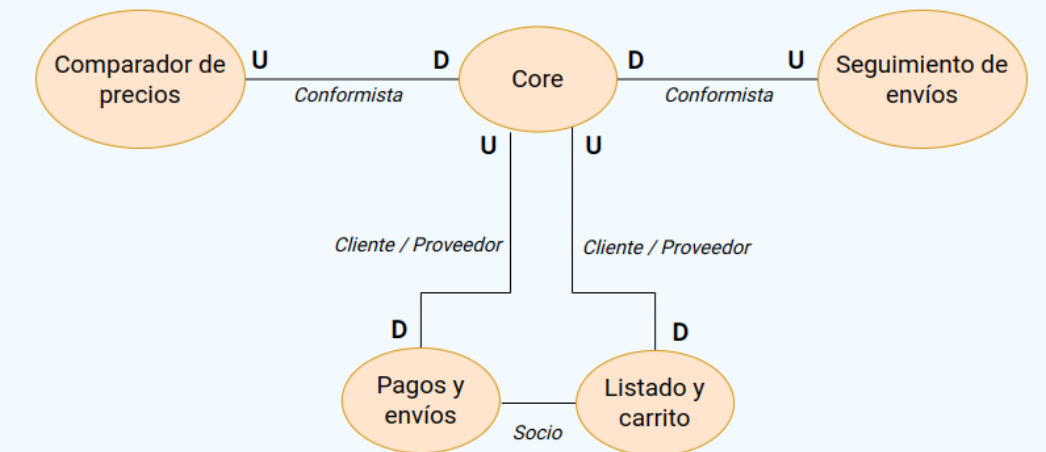
DIAGRAMA DE MAPEO DE CONTEXTO



El contexto **Upstream (U)** condiciona al **Downstream (D)**.

TIPOS DE RELACIONES

- **Conformista (Conformist).** No existe ninguna capacidad de negociación.
- **Cliente / Proveedor (Customer / Supplier).** Dependencia con cierto grado de negociación. Necesidades en el cliente pueden implicar cambios en el proveedor.
- **Socio (Partnership).** Ambos contextos **colaboran** por una meta en común, por lo que ambos lados de la relación tienen poder para influenciar al otro.
- **Núcleo compartido (Shared Kernel).** Dos o más contextos **comparten un mismo modelo**. Todos necesitan estar de acuerdo para realizar cambios. Difícil de mantener.



TIPOS DE RELACIONES

- **Capa anticorrupción (Anticorruption Layer).** Interfaz que utiliza Downstream para interactuar con Upstream, sin importar los cambios realizados en el último.
- **Open Host Service / Published Language.** Relaciones de tipo **conformista** en las que se provee de documentación al Downstream context. Además se proporcionan **versiones** y compatibilidades entre ellas.

CONCEPTOS DOMAIN DRIVEN DESIGN

MAPEO DE CONTEXTOS

Daniel Blanco Calviño