



Persistencia

Mapeo Objetos-Relacional

LUCIANO NICOLÁS STRACCIA

Versión Abril 2022

Persistencia

PERSISTENCIA

- La persistencia es la capacidad que tiene una aplicación de mantener el estado mas allá de su ejecución. Esto permite que ejecuciones posteriores puedan utilizar los datos generados en ejecuciones previas.

ELEMENTOS

- La persistencia implica:
- Medio persistente: es el soporte físico donde se almacenan los datos, puede ser un disco.
- Estructura de la base de datos: es el formato en el cual se guardan los datos.
- Base de datos o repositorio: conjunto de datos persistidos de la aplicación.
- Metadata: información acerca de los datos. Por ejemplo, el tipo de dato de un campo, su nombre, etc.
- Mecanismo para recuperar información de la base de datos
- Mecanismo para alterar la base de datos

ESTRUCTURA

- La persistencia puede ser realizada en diferentes estructuras:
- Archivos planos
- Archivos con formato estándar (XML, JSON, EDI-Fact, etc.)
- Base de datos relacionales
- Base de datos no relacionales

PERSISTENCIA RELACIONAL

- Elementos
- Claves
- Relaciones entre elementos
- SQL (Structured Query Language)

Mapeo objetos- relacional

MAPEO OBJETOS- RELACIONAL

- Si fuera un software basado en el modelo de programación orientada a objetos y una base de datos orientada a objetos:



SW OO



Objetos



BD OO

MAPEO OBJETOS- RELACIONAL

- Si fuera un software basado en el modelo de programación imperativo y una base de datos tradicional:



SW Tradicional



Registros



BD Tradicional

MAPEO OBJETOS- RELACIONAL

- Pero lo que más se utiliza es SW orientado a objetos y base de datos tradicional:



SW OO



BD Tradicional

- Es necesario un traductor: ORM (Object-Relational Mapping)

Impedance mismatch

- **Impedance Mismatch** (desajuste por impedancia) consiste en el problema o falta de concordancia objeto-relacional, esto consiste en un grupo de dificultades y problemas técnico-conceptuales a los que se enfrentan los diseñadores de bases de datos y los programadores. Estos problemas son generalmente la incompatibilidad entre los tipos de datos de las bases de datos y los tipos de datos del lenguaje de programación.

Herramientas ORM

- Se puede implementar en forma manual (implica muchas horas y un gran costo) o usar motores de ORM ya existentes



Técnicas de mapeo

Técnicas de mapeo

- Tener en cuenta:
 - Identidad
 - Relaciones (cardinalidad)
 - Herencia
 - Tipos de datos

Identidad

- Objetos:
 - Cada objeto sabe que es él y no otro
 - Cuando se referencia un objeto no se necesita identificarlo porque alguien nos pasó la referencia
 - Si no lo conocemos no podemos pedirle nada

Identidad

- Relacional:
- Necesidad de una clave que identifique unívocamente a cada registro de la tabla

Modelo relacional: Tabla Pedido

PEDIDO

PEDIDO_ID

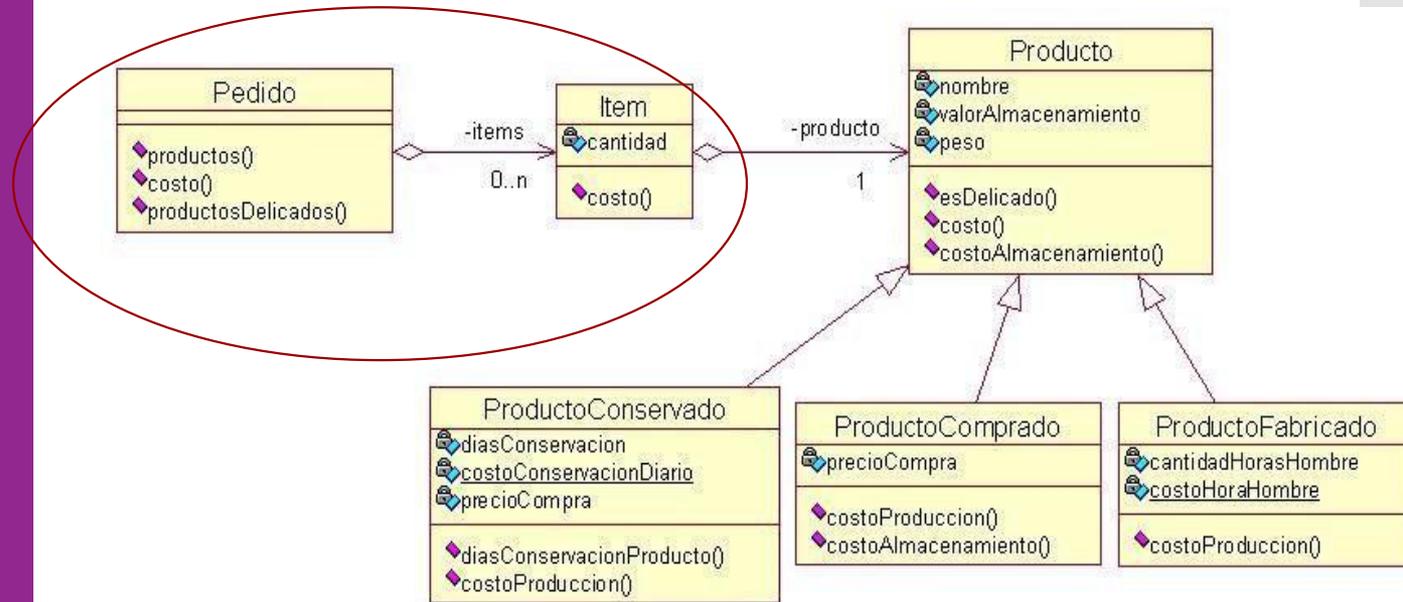
FECHA_PEDIDO

Modelo de objetos: Clase Pedido

Pedido

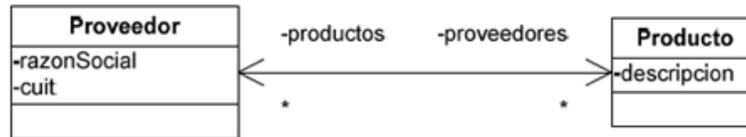
-fecha

Relaciones (cardinalidad)



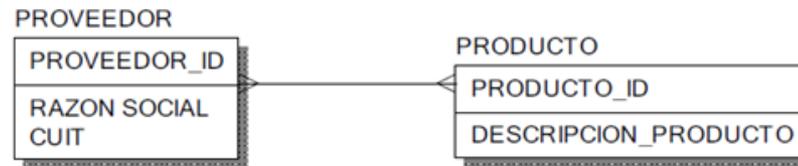
Relaciones (cardinalidad)

Objetos

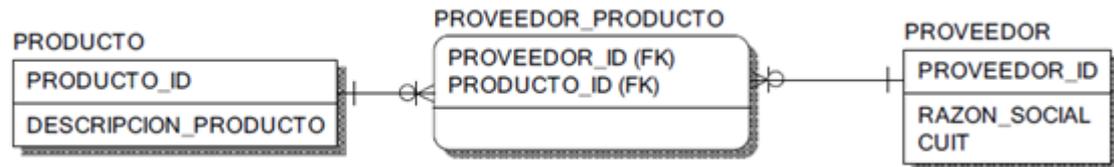


Colecciones en los objetos raíces

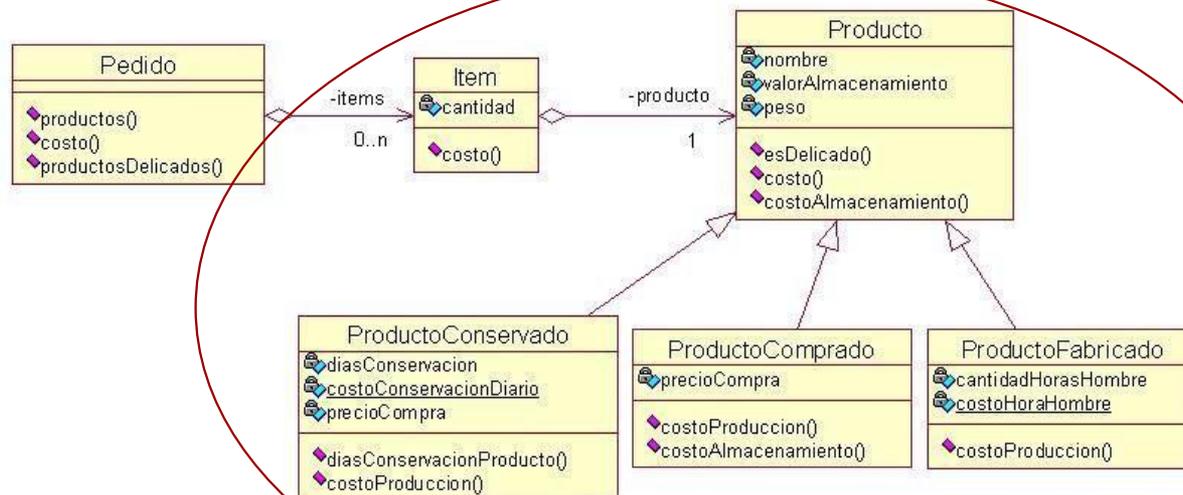
Relacional: modelo lógico



Relacional: modelo físico

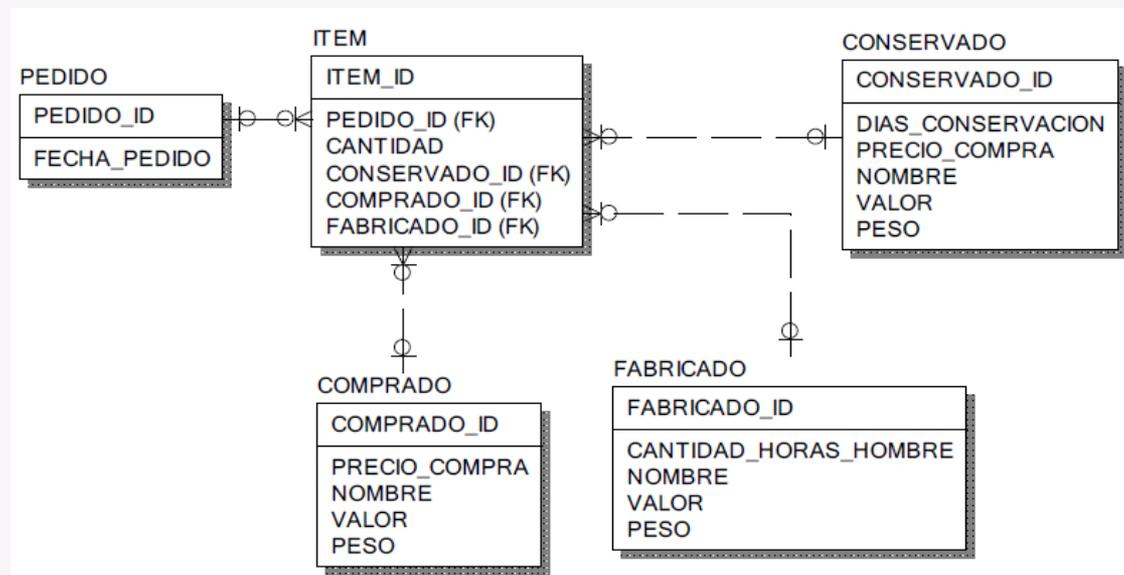


Herencia



Herencia

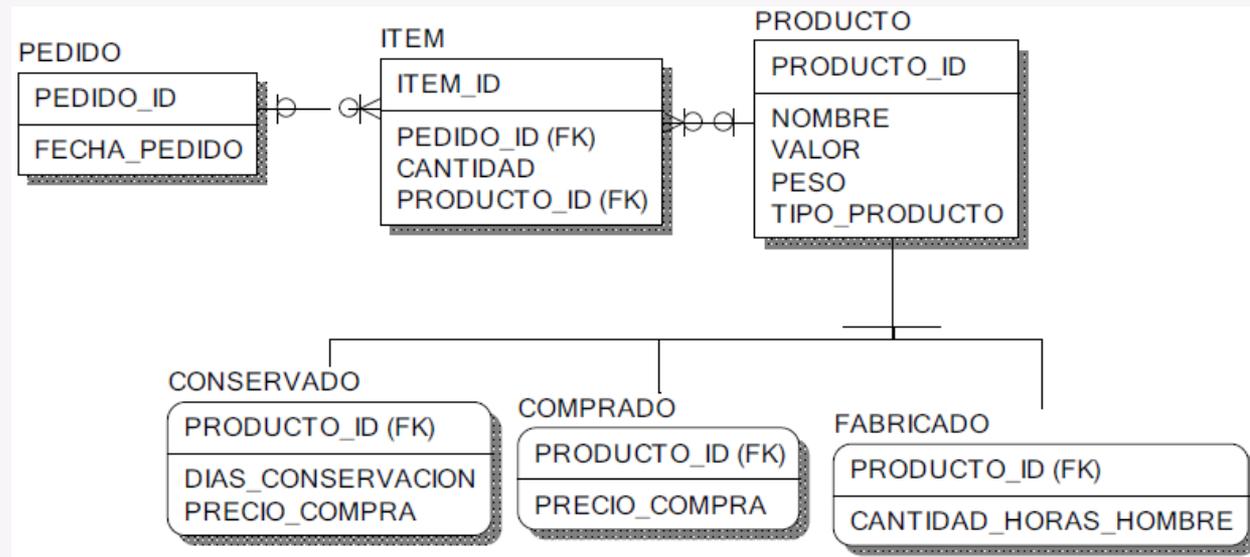
- Implementación 1: una tabla por subclase



- FK que pueden ser nulas / Atributos repetidos en subtipos

Herencia

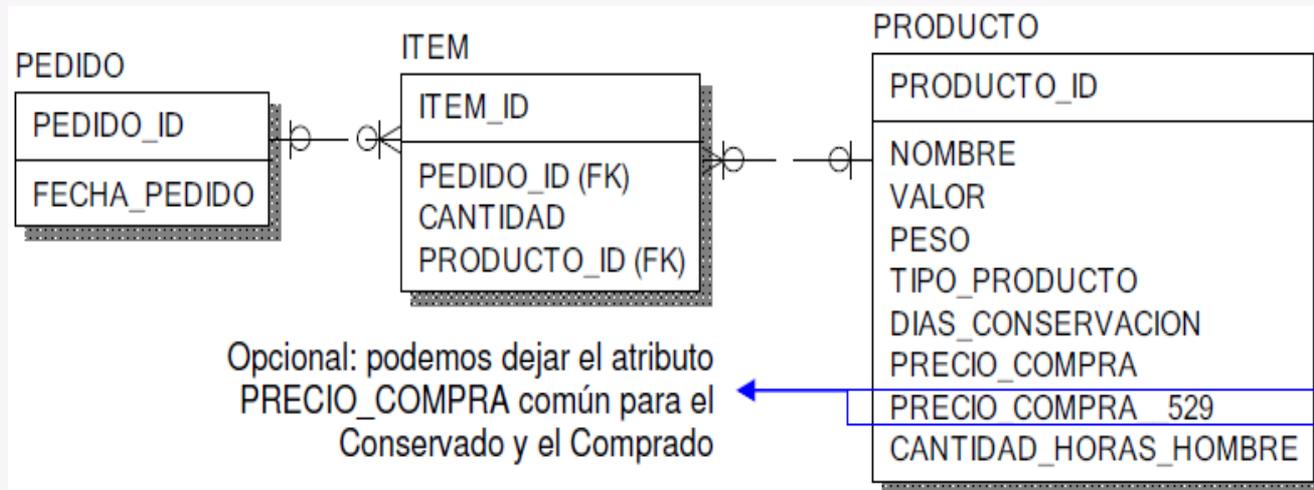
- Implementación 2: una tabla por clase



- Tipo de producto actúa como discriminante

Herencia

- Implementación 3: una única tabla

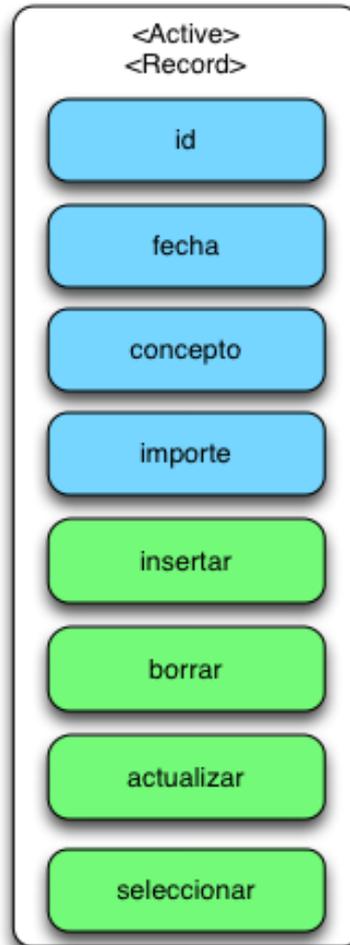


Tipos de datos

- Resolver problemas de conversión:
- String sin limitación de tamaño vs. Varchar/char que requiere definirle longitud
- Las fechas pueden tener tipos no siempre compatibles

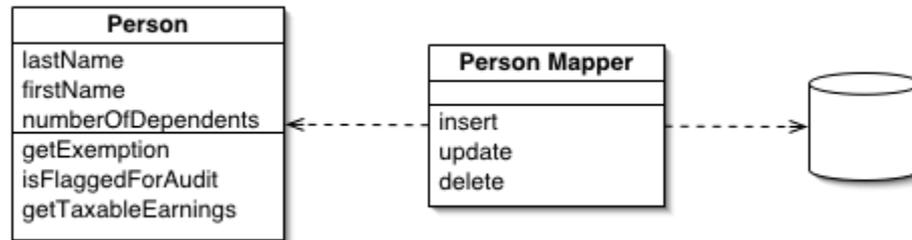
Patrones de acceso a datos

Active Record



<code><Tabla></code> Factura
<code>id</code>
<code>fecha</code>
<code>concepto</code>
<code>importe</code>

Data Mapper





PERSISTENCIA. ORM.
Luciano Nicolás Straccia
www.suriweb.com.ar