Persistencia en Sistemas O.O.

Taller de Programación

Instituto de Computación – Facultad de Ingeniería
Universidad de la República

Contenido

- Conceptos básicos
  - Definición y motivación de persistencia
  - Mecanismo de persistencia y framework
  - Base de datos
- Diseño de Persistencia
  - Arquitectura
  - Decisiones de diseño
  - Destinos de persistencia
  - Tipos de almacenamiento
- Mecanismos de Persistencia
  - Serialización de Objetos
  - Acceso directo a Base de datos
  - Mapeador Objeto Relacional
  - Generador de Código

Conceptos básicos

Conceptos Básicos (1)

- Persistencia de Información
  - Acción de preservar la información de forma permanente, de manera que pueda ser nuevamente utilizada.
- Persistencia de Objetos
  - Extender el tiempo de vida de un objeto más allá del tiempo de vida del proceso que lo creó.

Conceptos Básicos (2)

- Mecanismo de Persistencia
  - Técnica básica que permite resolver la persistencia de objetos.
- Framework
  - Estructura de soporte a un proyecto de desarrollo de software que permite unir y desarrollar diferentes componentes.
- Destino de Persistencia
  - Fuente de datos (Data Source) donde guardaremos la información a persistir de una aplicación (archivo, base de datos, etc.)

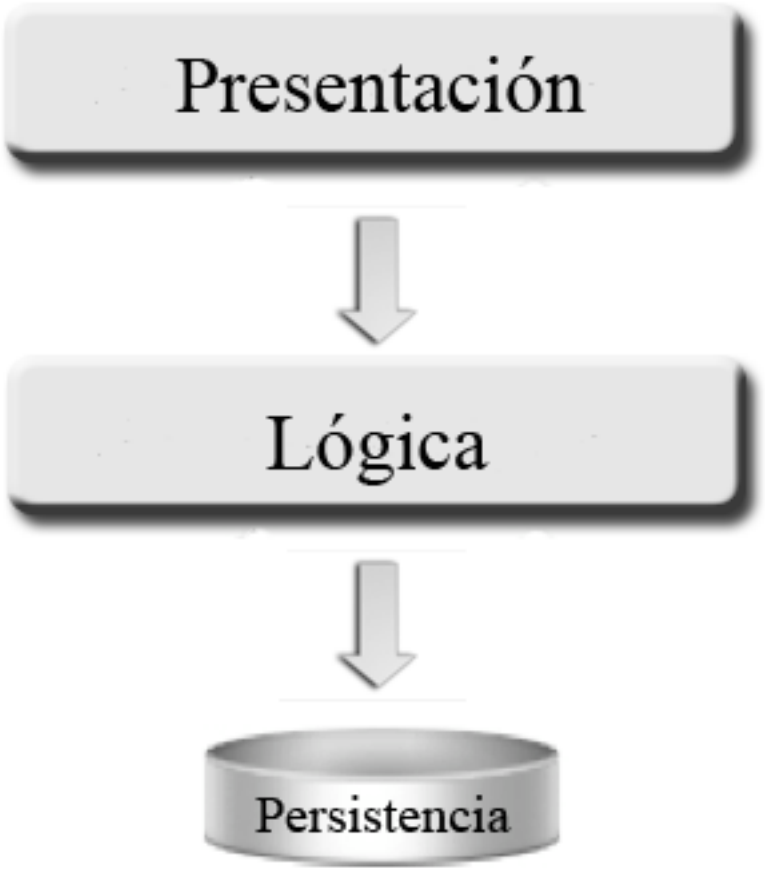
Conceptos Básicos (3)

- Base de Datos
  - Conjunto de datos relacionados con diferentes modos de organización.
- DBMS *(Database Management System)*
  - Grupo de Programas que sirven para definir, construir y manipular una base de datos.
- Metadata
  - Información sobre un conjunto particular de datos.

Diseño de Persistencia

Arquitectura en Capas (1)

- Partición de los componentes en capas según las responsabilidades de sus elementos.



Arquitectura en Capas (2)

- Los componentes de la capa lógica podrían ser los responsables de cargar y guardar los datos.
- Desventajas:
  - Alto acoplamiento entre los componentes y la fuente de datos.
  - Difícil mantenimiento.

Arquitectura en Capas (3)

- Una alternativa es contar con una capa acceso a la persistencia.
- La misma contiene uno o más controladores responsables de guardar y cargar los datos de la aplicación.
- Se encargan de toda la interacción con el destino de persistencia.

Arquitectura en Capas (4)

- La capa lógica contiene:
- Clases que permiten resolver los casos de uso de la aplicación
- Clases que permiten acceder a los datos persistidos.

Arquitectura en Capas (5)

- Patrón de Diseño DAO:
  - Encapsula todo el acceso a una fuente de datos utilizando la capa de acceso a datos.
  - Cada objeto de negocio a ser persistido crea un DAO con la información necesaria.
  - Luego utiliza el objeto DAO para obtener y guardar información del origen de datos.

Arquitectura en Capas (6)

- Ventajas:
  - Centralización del acceso a los datos en la nueva capa.
  - Mayor transparencia debido a que los objetos de negocio no conocen el destino de los datos.
  - Menor complejidad en objetos de negocio.

Decisiones de diseño (1)

- Autoescritura
  - Cada clase se encarga de persistirse.
  - No se necesitan clases adicionales para la persistencia.
  - No proporciona correcta separación entre capas.
  - Es poco flexible.

Decisiones de diseño (2)

- Wrappers
  - Crear wrappers (envoltorios) para cada clase a persistir y darle la responsabilidad de persistir su clase asociada.
  - Mayor independencia de la lógica.
  - Solución más compleja.

Decisiones de diseño (3)

- Datatypes
  - Utilización de un datatype por cada clase a persistir.
  - Sencillo de implementar
  - Independiente del diseño realizado
  - Flexible a cambios en diseño y requerimientos

Decisiones de diseño (4)

- Problemas comunes:
  - Doble visibilidad
  - Ciclos entre clases
  - Links a objetos aun no cargados
- Soluciones
  - Establecer orden de carga
  - Guardar información de los links entre objetos en fuentes de datos separadas.

Destinos de Persistencia (1)

- Existen distintas formas de persistir los datos de una aplicación:
  - Archivos:
    - Texto plano
    - XML
    - CSV
  - Bases de Datos:
    - Relacionales
    - Orientadas a Objetos
    - XML

Destinos de Persistencia (2)

- Archivos:
  - Cantidad de Archivos a utilizar:
    - Un único archivo para todos los datos del sistema.
    - Un archivo por clase fuerte del sistema.
    - Un archivo por clase a persistir.
  - Métodos de carga
    - Establecer un orden de carga.
    - Mantener archivos de relaciones entre objetos.
  - Creación de Archivos
    - Existen herramientas que asisten el pasaje de objetos al formato de archivo utilizado y viceversa.

Tipos de Almacenamiento (1)

- Se debe determinar en que momento se deben persistir u obtener los datos de la aplicación:
  - Estático
    - Se cargan todos los datos del sistema al iniciar la aplicación y se guardan los datos potencialmente modificados al finalizar su ejecución
  - Momento determinado
    - Cuando el usuario lo indique o cada cierto tiempo, se deben persistir todos los datos de la aplicación.

Tipos de Almacenamiento (2)

- Continuo
  - Se almacenan los datos a medida que son modificados en el sistema
  - No se necesita almacenar todos los datos al finalizar la ejecución de la aplicación
  - Mecanismo similar al utilizado por las bases de datos
  - Se deben manejar transacciones para que no se registren inconsistencia en los datos
  - Es más complejo de implementar

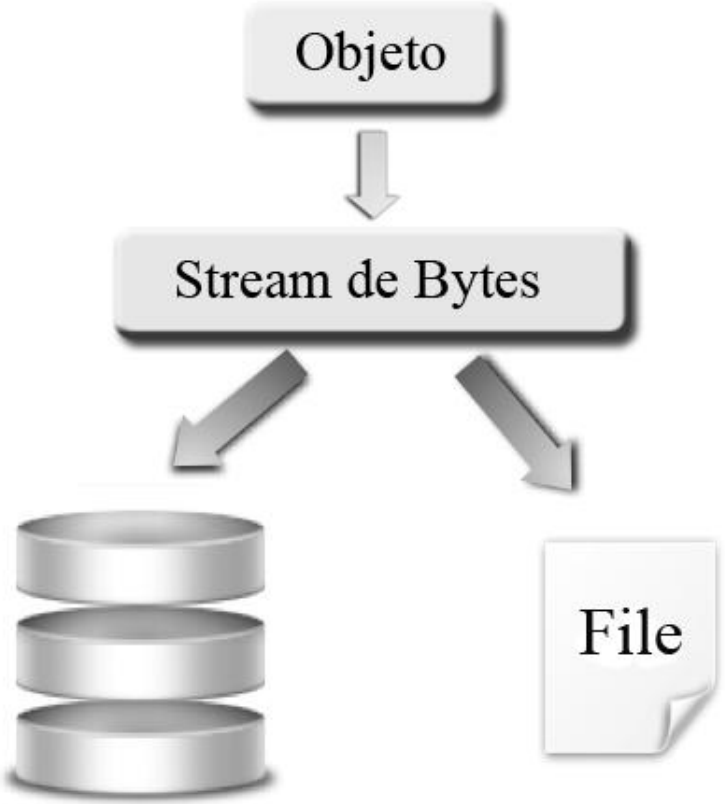
Mecanismos de Persistencia

Mecanismos de Persistencia(1)

- Serialización de Objetos
- Acceso directo a bases de datos
- Mapeadores Objeto - Relacional
- Generadores de Código

Mecanismos de Persistencia(2)

- Serialización de Objetos

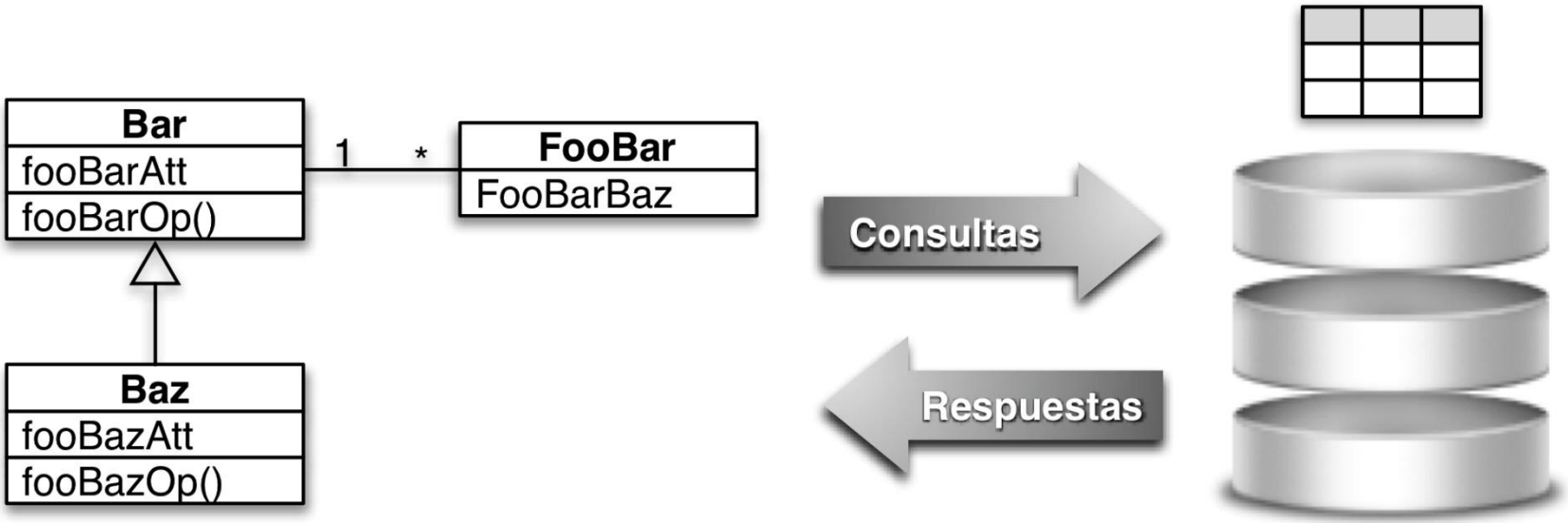


Mecanismos de Persistencia(3)

- Serialización de Objetos
  - Ventaja
    - Mecanismo simple de Persistencia
  - Desventaja
    - No soporta transacciones ni posee mecanismos de consulta de datos.
  - Java provee la serialización de Objetos.

Mecanismos de Persistencia(4)

- Acceso directo a base de datos

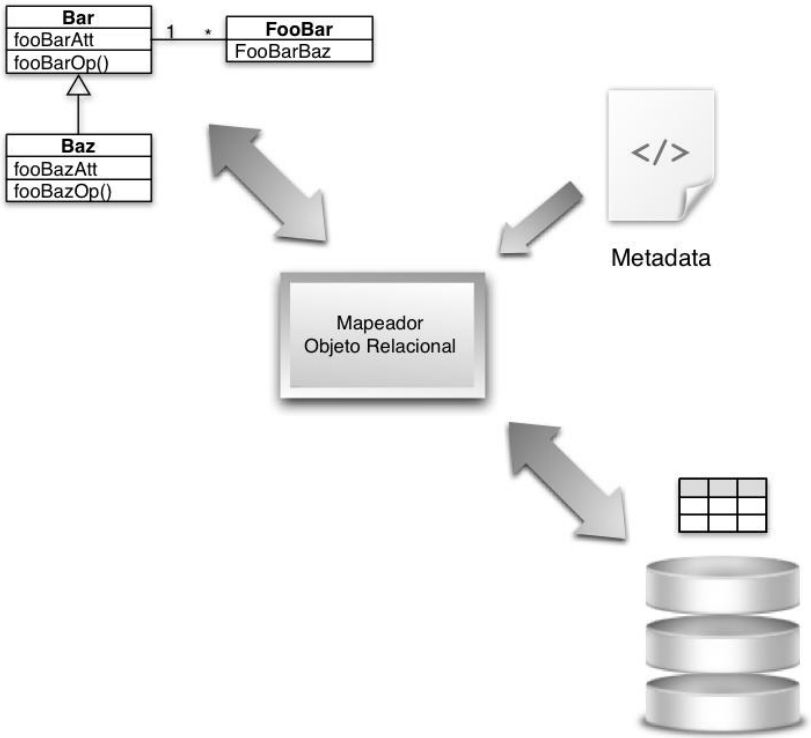


Mecanismos de Persistencia(5)

- Acceso directo a Base de Datos
  - Ventaja
    - Buen desempeño debido a que el acceso a la base se realiza en forma directa
  - Desventaja
    - Mayor complejidad para el diseño de la persistencia.
  - Herramienta Java
    - JDBC (Java Database Connectivity)

Mecanismos de Persistencia(6)

- Mapeador Objeto - Relacional

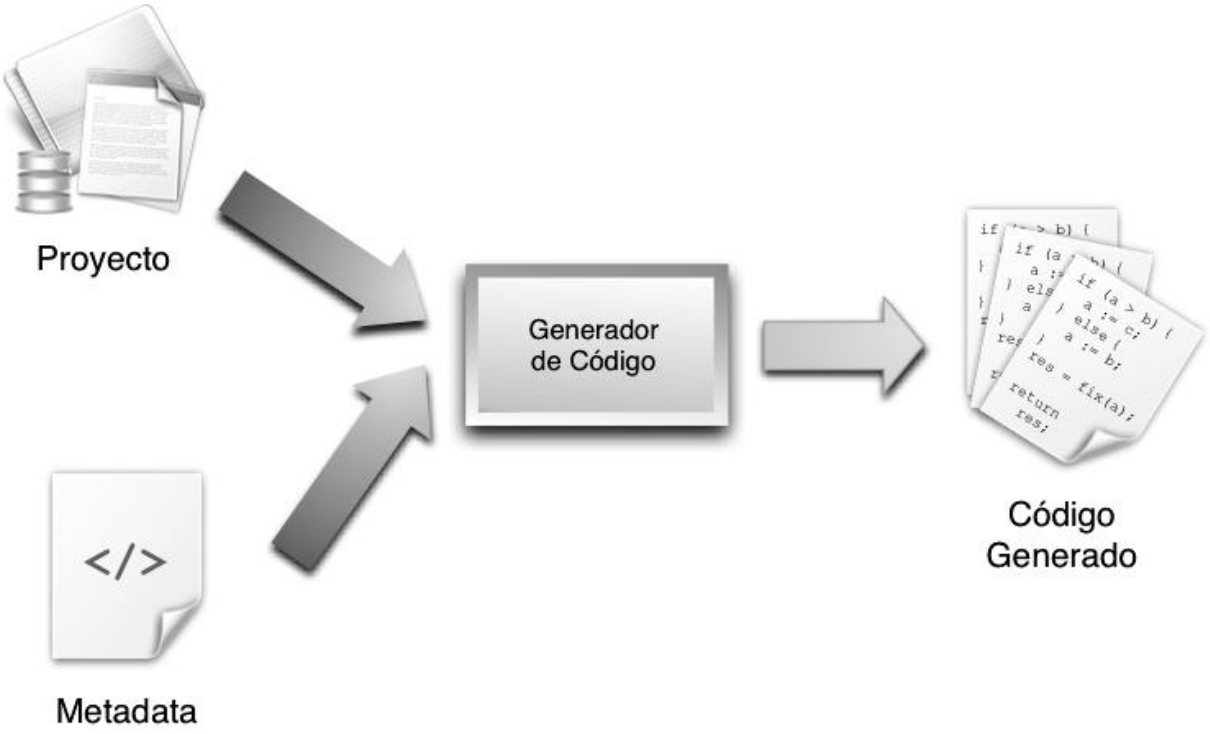


Mecanismos de Persistencia(7)

- Mapeador Objeto - Relacional
  - Ventaja
    - Reduce la cantidad de código necesario para lograr la persistencia. La aplicación resultante es fácil de mantener.
  - Desventaja
    - Requiere Mantener metadata sobre las entidades.
  - Herramienta Java
    - Hibernate

Mecanismos de Persistencia(8)

- Generador de Código



Mecanismos de Persistencia(9)

- Generador de Código
  - Ventaja
    - El código generado suele ser estable y libre de errores
  - Desventaja
    - Requiere inversión de tiempo en construir el esquema de datos previo a la generación de código.
  - Herramienta Java
    - FireStorm/DAO

Referencias

- Mecanismos de Persistencia
  - http://www.fing.edu.uy/inco/grupos/coal/uploads/Investigaci%f3n/mps07.pdf
- Patrón de diseño DAO
  - http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html
- Hibernate
  - http://www.hibernate.org/
- Serialización en Java
  - http://java.sun.com/developer/technicalArticles/Programming/serialization/