



Modelado de objetos de acceso a datos en dispositivos móviles

JULIA YAMILE GONZALEZ LUNA

DIEGO ARMANDO OCHOA MARTINEZ

ALAN OMAR LUNA FLORES

LUCERO MARTINEZ OLMOS

BRANDON JESUS ALEMAN ELIZALDE

AARÓN ERUBIEL ROBLES FLORES

LUIS FERNANDO PEREZ LUNA

PROGRAMACION PARA MOVILES II

GRUPO: 1922IS

DOCENTE: EMMANUEL TORRES SERVIN

Ingeniería en software | 28/06/22



INDICE

1. Introducción

2. Modelado de objetos en dispositivos móviles.

- 2.1. Características de los objetos de acceso a datos en dispositivos móviles.
- 2.2. Proceso de modelado de objetos de acceso a datos en dispositivos móviles.
- 2.3. Proceso de programación de objetos de acceso a datos en dispositivos móviles. Manipulación de datos en dispositivos móviles
- 2.4. Reconocer el concepto de conexión a bases de datos.
- 2.5. Describir la conexión a bases de datos estáticos, dinámicos, web y locales en dispositivos móviles.
- 2.6. Explicar el proceso de programación de conexión a bases de datos estáticos en dispositivos móviles.
- 2.7. Explicar el proceso de programación de conexión a bases de datos dinámicos en dispositivos móviles.
- 2.8. Explicar el proceso de programación de conexión a bases de datos locales en dispositivos móviles.

3. Persistencia de datos en los dispositivos móviles

- 3.1. Concepto de persistencia en dispositivos móviles.
- 3.2. Retos de la persistencia en los dispositivos móviles.
- 3.3. formas de persistencia en los sistemas operativos de los dispositivos móviles: preferencias, almacenamiento de archivos, datos estructurados.
- 3.4. Tipos de persistencia: local, remota y de Cacheo/Hoarding en dispositivos móviles.
- 3.5. Proceso de programación de persistencia en dispositivos móviles.

4. Mecanismos de tolerancia a fallos

- 4.1. Elementos para tomar en cuenta en el desarrollo de aplicaciones orientadas a móviles:
 - 4.1.1. lentitud en las consultas

4.1.2. alto consumo de batería

4.1.3. dificultad de sincronización

4.1.4. modo off-line

4.1.5. recuperación de conexión

4.2. Proceso de selección de los mecanismos de tolerancia a fallos en el desarrollo de aplicaciones de dispositivos móviles.

4.3. Proceso de programación de los mecanismos de tolerancia a fallos en el desarrollo de aplicaciones de dispositivos móviles.

INTRODUCCION

Un Objeto de Acceso a Datos o Data Access Object (DAO) son una serie de objetos que le permiten tener acceso y manipular datos mediante programación en bases de datos locales o remotos. Puede utilizar DAO para administrar bases de datos, así como sus objetos y su estructura.

Es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo. El término se aplica frecuentemente al Patrón de diseño Object.

Los Objetos de Acceso a Datos son un Patrón de Diseño Core J2EE y considerados una buena práctica.

Los Objetos de Acceso a Datos pueden usarse en Java para aislar a una aplicación de la tecnología de persistencia Java subyacente(API de Persistencia Java), la cual podría ser JDBC, JDO, EJB,CMP(Persistencia controlada por el Contenedor), TopLink, Hibernate, iBATIS, o cualquier otra tecnología de persistencia. Usando Objetos de Acceso de Datos significa que la tecnología subyacente puede ser actualizada o cambiada sin cambiar otras partes de la aplicación.

Para tener acceso y manipular datos mediante programa debe comprender la jerarquía de DAO. El orden de los objetos en DAO se conoce como su modelo de objetos. El modelo de objetos de DAO le permite escribir código que puede aprovechar la funcionalidad de la base de datos.

2. Modelado de objetos en dispositivos móviles

¿QUE ES?

El modelado de datos es el proceso de analizar y definir todos los diferentes datos que su empresa recopila y produce, así como las relaciones entre esos bits de datos. El proceso de modelado de sus datos crea una representación visual de datos a medida que se utilizan en su negocio, y el propio proceso es un ejercicio de conocimiento y aclaración de sus requisitos de datos.

CARACTERISTICAS DEL MODELADO DE DATOS

Al modelar sus datos, documentará los datos que tiene, cómo los usa y cuáles son sus requisitos relacionados con el uso, la protección y la gobernanza. Mediante el modelado de datos, su organización:

- Crea una estructura para la colaboración entre su equipo de TI y sus equipos comerciales.
- Expone oportunidades para mejorar los procesos comerciales, al definir las necesidades y los usos de los datos.
- Ahorra tiempo y dinero en TI y en inversiones en procesos, mediante una planificación adecuada por adelantado.
- Reduce los errores (y la entrada de datos redundantes propensa a errores), al tiempo que mejora la integridad de los datos.
- Aumenta la velocidad y el rendimiento de la recuperación y el análisis de datos, al planificar la capacidad y el crecimiento.

Por lo tanto, no se trata solo de lo que obtiene con el modelado de datos, sino también de cómo lo obtiene.

PROCESOS DE MODELADO DE OBJETOS DE ACCESO A DATOS EN DISPOSITIVOS MOVILES

Conceptual

Un modelo de datos conceptual define la estructura general de su negocio y sus datos. Se utiliza para organizar conceptos de negocio, según la definición de las partes interesadas en el negocio y los arquitectos de datos. Por ejemplo, puede tener datos de clientes, empleados y productos, y cada uno de esos depósitos de datos, conocidos como entidades, tiene relaciones con otras entidades. Tanto las entidades como las relaciones entre entidades se definen en su modelo conceptual.

Físico

Un modelo de datos físicos es su implementación específica del modelo de datos lógicos, y lo crean los administradores de la base de datos y los desarrolladores. Está desarrollado para una herramienta de base de datos específica o tecnología de almacenamiento de datos, y con conectores de datos para entregar los datos en todos los sistemas de negocio a los usuarios según sea necesario. Esta es la “cosa” a la que han estado conduciendo el resto de modelos: la implementación real de su estado de datos.

Lógico

Un modelo de datos lógicos se basa en el modelo conceptual con atributos específicos de datos dentro de cada entidad y relaciones específicas entre esos atributos. Por ejemplo, el cliente A compra el producto B al asociado de ventas C. Este es su modelo técnico de las reglas y estructuras de datos definidas por los arquitectos de datos y analistas de negocios, y ayudará a tomar decisiones sobre qué modelo físico requieren sus datos y necesidades de negocio.

MANIPULACION DE DATOS

La manipulación de datos es el proceso de cambiar o alterar datos para hacerlos más legibles y organizados. Por ejemplo, puede ordenar los datos alfabéticamente para acelerar el proceso de búsqueda de información útil

XML.

El Extensible Markup Language (XML) es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos en un formato que es a la vez legible y legible por máquina.

JSON.

JSON (/ dʒeɪsɒn / jah-soun, / dʒeɪsən / ja-hijo), o JavaScript Object Notation, es un formato estándar abierto que utiliza texto legible para transmitir objetos de datos que constan de pares atributo-valor. Se utiliza sobre todo para transmitir datos entre un servidor y aplicaciones web, como alternativa a XML.

Aunque en un principio derivado de la lengua scripting JavaScript, JSON es un formato de datos independiente del lenguaje, y el código para analizar y generar datos JSON está fácilmente disponible en una gran variedad de lenguajes de programación.

CONEXIÓN A BASE DE DATOS

Una conexión a base de datos es un archivo de configuración donde se especifica los detalles físicos de una base de datos como por ejemplo el tipo de base de datos y la versión, y los parámetros que permiten una conexión.

Describir la conexión a bases de datos estáticos, dinámicos, web y locales en dispositivos móviles

Una base de datos **estática** es aquella cuya función principal es el almacenamiento y registro de datos fijos. Es decir, guarda información que no se va a modificar ni editar con el tiempo.

Se trata de un tipo de bases de datos de solo lectura. Su implementación se suele realizar con el objetivo de registrar datos históricos para poder comparar su evolución a lo largo del tiempo.

En la clasificación entre bases de datos según la variabilidad de la información hay dos grandes grupos: las bases de datos estáticas y las dinámicas.

La principal diferencia entre ambas es que, mientras las bases de datos estáticas son de solo lectura y no permite modificar o añadir datos, las bases de datos dinámicas son mucho más flexibles y sí permiten editar, actualizar o borrar datos.

Por ello, las bases estáticas se suelen emplear para información que es inamovible en el tiempo, mientras que las dinámicas se emplean para crear bases con datos que son susceptibles de variar con el tiempo.

3. PERSISTENCIA DE DATOS EN MOVILES

La persistencia en Android consiste en tres tipos de almacenamientos con un propósito muy específico.

1. Persistencia en Android: Preferencias Compartidas o Shared Preferences

Con Shared Preferences podemos almacenar y recuperar en el formato clave-valor información como texto, booleanos y números; lo que lo convierte en potencial para almacenar configuraciones del usuario como: estilos, preferencias, etc.

Persistencia en Android: Almacenar archivos en memoria

- Este tipo de persistencia es uno de los más conocidos debido a que son soportados por la mayoría de los lenguajes de programación aparte de Java; consiste en guardar y recuperar la información en archivos; Android permite escribir y leer archivos que se encuentren ubicados en la propia Memoria Interna del dispositivo; al igual que con Shared Preferences.
- Almacenar archivos en Memoria Interna

- Hay que tener en cuenta que estos archivos son guardados en la Memoria Interna del dispositivo la cual puede ser limitada en el dispositivo y puede ralentizar el mismo.

Consideraciones

- No es necesario pedir permisos especiales en el Manifest.

Sobre su uso: Apertura y Escritura de archivos en Android

- Android proporciona el método `OutputStreamWriter` para escribir archivos en Android a través de la apertura del mismo con el método `openFileOutput`; el mismo recibe como parámetro:
 - El nombre del archivo.
 - El modo de acceso:
 - Es posible configurar los archivos para que solo puedan ser gestionados por la aplicación y por nadie ni nada más; ni siquiera el usuario; los modos de acceso son los siguientes:
 - `MODE_PRIVATE`: Solo es accesible por la aplicación y por nadie ni nada más (crea el archivo o lo sobrescribe si ya existe).
 - `MODE_APPEND`: Añade contenido a un archivo existente en el dispositivo.
 - `MODE_WORLD_READABLE`: Permite que otras aplicaciones puedan leer el archivo (deprecated desde el API 17).
 - `MODE_WORLD_WRITEABLE`: Permite que otras aplicaciones puedan escribir el archivo (deprecated desde el API 17).

4. MECANISMOS DE TOLERANCIA A FALLOS

Tolerancias a fallos en informática se determina a la capacidad de un sistema de almacenamiento de acceder a información o al recurso aún en caso de producirse algún fallo. Esta falla puede deberse a daños físicos (mal funcionamiento) en uno o más componentes de hardware lo que produce la pérdida de información almacenada. La tolerancia a fallos requiere para su implementación que el sistema

de almacenamiento guarde la misma información en más de un componente de hardware o en una máquina o dispositivo externos a modo de respaldo. De esta forma, si se produce alguna falla con una consecuente pérdida de datos, el sistema debe ser capaz de acceder a toda la información recuperando los datos faltantes desde algún respaldo disponible.

Nivel proceso

La latencia de tolerancia a fallas (FTL - Fault Tolerance Latency) es el tiempo requerido para completar todos los pasos secuenciales que se necesitan para recuperar un error. Esta definición puede volverse sumamente importante en sistemas de tiempo real, donde la latencia de cualquier política de manejo de errores (tales como detección, enmascaramiento y recuperación de errores, por ejemplo) no debe ser superior a la latencia requerida por la aplicación (ARL - Application Required Latency), que es un tiempo que depende del proceso.

Al evaluar la FTL considerando distintos mecanismos de tolerancia a fallas, se puede utilizar dicho FTL para comprobar si una determinada estrategia de tolerancia a fallas elegida para una aplicación puede cumplir las metas (debe ser $FTL \leq ARL$).

Nivel almacenamiento

Que el sistema de archivos sea tolerante a fallos implica que el Sistema de Archivos tenga un sistema de recuperación de transacciones, además de guardar varias copias de los archivos en distintas máquinas para garantizar la disponibilidad en caso de fallo del servidor principal. Además, se ha de aplicar un algoritmo que nos permita mantener todas las copias actualizadas de forma consistente, o un método alternativo que sólo nos permita acceder al archivo actualizado; como por ejemplo el invalidar el resto de copias cuando en cualquiera de ellas se vaya a realizar una operación de escritura. El uso de memorias caché para agilizar el acceso a los archivos también es recomendable, pero este caso requiere analizar con especial atención la consistencia del sistema.