

Trabajo de investigación

Dispositivos móviles



Profesora:
Torres Servín Emmanuel
Programación visual.

Grupo:
ingeniería en software
4322-IS

Estudiantes:
1321124285 | Hernández Silva Raúl Abraham.
1321124381 | Gutiérrez García Harold Guillermo.

Contenido

Modelado de objetos de acceso a datos en dispositivos móviles.	2
Características de los objetos de acceso a datos en dispositivos móviles.....	2
Proceso de modelado de objetos de acceso a datos en dispositivos móviles.....	3
Proceso de programación de objetos de acceso a datos en dispositivos móviles.	6
Manipulación de datos en dispositivos móviles.....	8
La conexión a bases de datos estáticos, dinámicos, web y locales en dispositivos móviles.....	8
Proceso de programación de conexión a bases de datos estáticos en dispositivos móviles.....	9
Proceso de programación de conexión a bases de datos dinámicos en dispositivos móviles.	11
Explicar el proceso de programación de conexión a bases de datos locales en dispositivos móviles.	12
Persistencia de datos en los dispositivos móviles.....	12
Retos de la persistencia en los dispositivos móviles.	12
formas de persistencia en los sistemas operativos de los dispositivos móviles..	14
Tipos de persistencia en dispositivos móviles.....	14
Proceso de programación de persistencia en dispositivos móviles.	15
Mecanismos de tolerancia a fallos	15
Proceso de selección de los mecanismos de tolerancia a fallos en el desarrollo de aplicaciones de dispositivos móviles.....	16
Proceso de programación de los mecanismos de tolerancia a fallos en el desarrollo de aplicaciones de dispositivos móviles.	16
Bibliografía.....	18

Modelado de objetos de acceso a datos en dispositivos móviles.

Características de los objetos de acceso a datos en dispositivos móviles.

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.

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.

La flexibilidad tiene un precio. Cuando se añaden DAOs a una aplicación, la complejidad adicional de usar otra capa de persistencia incrementa la cantidad de código ejecutado durante tiempo de ejecución. La configuración de las capas de persistencia requiere en la mayoría de los casos mucho trabajo.

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.

Los dispositivos móviles se están proliferando en la empresa a un ritmo exponencial. Ganar visibilidad y control sobre los dispositivos corporativos y los personales que se conectan a la red de su empresa y tienen acceso a recursos corporativos.

La solución de la administración de dispositivos móviles de AirWatch (MDM) le permite administrar las implementaciones a gran escala de los dispositivos móviles. Nuestra solución le brinda la habilidad de inscribir dispositivos rápidamente en su entorno empresarial, configurar y actualizar los ajustes del dispositivo over-the-air, imponer políticas de seguridad y conformidad, acceso seguro móvil a recursos corporativos y eliminar o bloquear dispositivos remotamente. Con AirWatch, se puede administrar un conjunto diverso de dispositivos Android, Apple iOS, BlackBerry, Mac OS X, Symbian, y Windows en una sola consola.

Proceso de modelado de objetos de acceso a datos en dispositivos móviles.

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. La ventaja de usar objetos de acceso a datos es que cualquier objeto de negocio (aquel que contiene detalles específicos de operación o aplicación) no requiere conocimiento directo del destino final de la información que manipula.

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.

La flexibilidad tiene un precio. Cuando se añaden DAOs a una aplicación, la complejidad adicional de usar otra capa de persistencia incrementa la cantidad de código ejecutado durante tiempo de ejecución. La configuración de las capas de persistencia requiere en la mayoría de los casos mucho trabajo.

- Manipulación de datos.
 - Transact-SQL proporciona sintaxis no especializada para las instrucciones INSERT, UPDATE o DELETE cuando se modifican datos de columnas de tipos definidos por el usuario (UDT). Las funciones CAST o CONVERT de Transact-SQL se utilizan para convertir tipos de datos nativos al tipo definido por el usuario.
- XML.
 - 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 . Se define en la especificación XML 1.0 producido por el W3C , y varias otras especificaciones relacionadas , todos los estándares abiertos libres.
 - Los objetivos de diseño de XML enfatizan la simplicidad, generalidad y facilidad de uso a través de Internet . Se trata de un formato de datos textual con un fuerte apoyo a través de Unicode para los idiomas del mundo . Aunque el diseño de XML se centra en los documentos , que es ampliamente utilizado para la representación de estructuras de datos arbitrarias , por ejemplo, en los servicios web

- Muchas interfaces de programación de aplicaciones (API) se han desarrollado para ayudar a los desarrolladores de software con el procesamiento de datos XML , y existen varios sistemas de esquema para ayudar en la definición de lenguajes basados en XML .
- A partir de 2009 , cientos de formatos de documentos utilizando la sintaxis XML se han desarrollado , incluyendo RSS, Atom , SOAP y XHTML . Formatos basados en XML se han convertido en la opción predeterminada para muchas herramientas de oficina de productividad, incluyendo Microsoft Office (Office Open XML) , OpenOffice.org y LibreOffice (OpenDocument) , e iWork de Apple. XML también ha sido empleado como el idioma base para los protocolos de comunicación , tales como XMPP .
- 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.
 - El formato JSON se especificó originalmente por Douglas Crockford, y se describe en RFC 4627 y ECMA-404. El tipo oficial de los medios de comunicación de Internet para JSON es application / json. La extensión de archivo es JSON. Json

Proceso de programación de objetos de acceso a datos en dispositivos móviles.

Problemática.

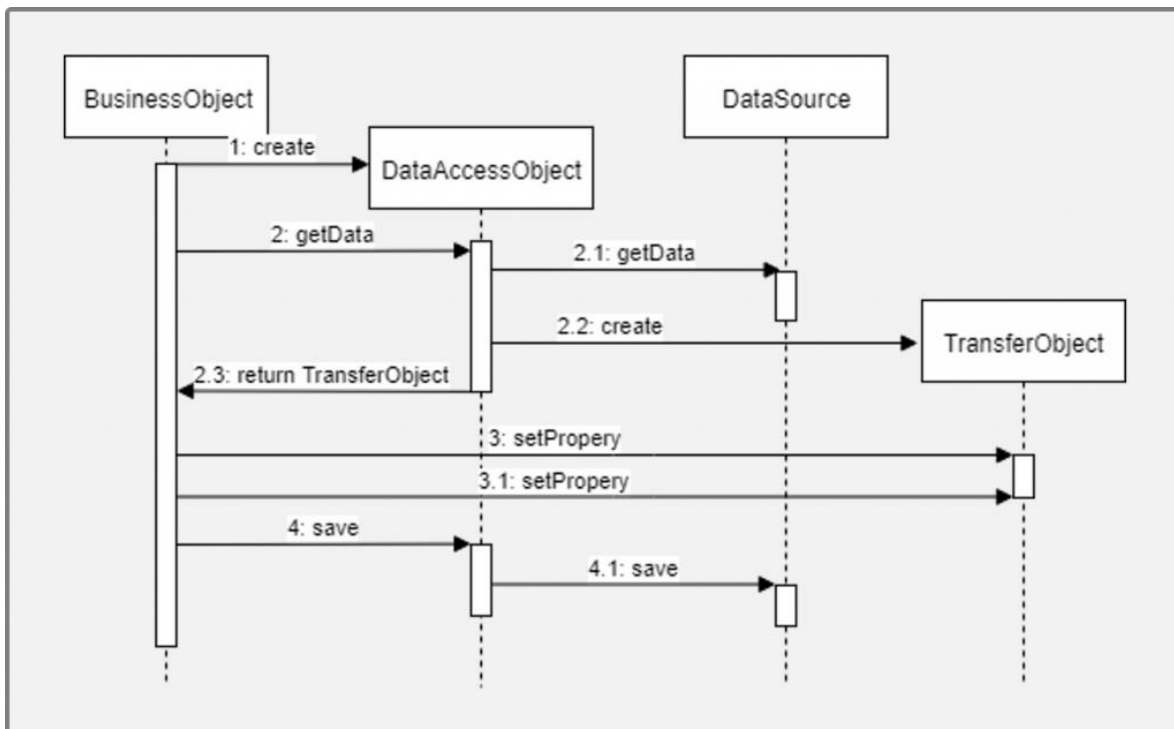
- Una de las grandes problemáticas al momento de acceder a los datos, es que la implementación y formato de la información puede variar según la fuente de los datos. Implementar la lógica de acceso a datos en la capa de lógica de negocio puede ser un gran problema, pues tendríamos que lidiar con la lógica de negocio en sí, más la implementación para acceder a los datos, adicional, si tenemos múltiples fuentes de datos o estas pueden variar, tendríamos que implementar las diferentes lógicas para acceder las diferentes fuentes de datos, como podrían ser: bases de datos relacionales, No SQL, XML, archivos planos, Webservices, etc).

Solución.

- Dado lo anterior, el patrón DAO propone separar por completo la lógica de negocio de la lógica para acceder a los datos, de esta forma, el DAO proporcionará los métodos necesarios para insertar, actualizar, borrar y consultar la información; por otra parte, la capa de negocio solo se preocupa por lógica de negocio y utiliza el DAO para interactuar con la fuente de datos.
- Los componentes que conforman el patrón son:
 - **BusinessObject**: representa un objeto con la lógica de negocio.
 - **DataAccessObject**: representa una capa de acceso a datos que oculta la fuente y los detalles técnicos para recuperar los datos.
 - **TransferObject**: este es un objeto plano que implementa el patrón Data Transfer Object (DTO), el cual sirve para transmitir la información entre el DAO y el Business Service.

- **DataSource:** representa de forma abstracta la fuente de datos, la cual puede ser una base de datos, Webservices, LDAP, archivos de texto, etc.

El siguiente diagrama muestra mejor la forma en que funciona el patrón, pues muestra de forma secuencial la forma en que se ejecutaría el patrón.



El diagrama se interpreta de la siguiente manera:

1. El *BusinessObject* crea u obtiene una referencia al *DataAccessObject*.
2. El *BusinessObject* solicita información al *DataAccessObject*
 - El *DataAccessObject* solicita la información al *DataSource*
 - El *DataAccessObject* crea una instancia del *TransferObject* con los datos recuperados del *DataSource*
 - El *DataAccessObject* response con el *TransferObject* creado en los pasos anteriores.

3. El *BusinessObject* actualiza algún valor del *TransferObject*
 - Más actualizaciones
4. El *BusinessObject* solicita el guardado de los datos actualizados al *DataAccessObject*.
 - El *DataAccessObject* guarda los datos en el *DataSource*.

Manipulación de datos en dispositivos móviles

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 JDBC desde el IBM® Integration Toolkit a la base de datos.

La conexión a bases de datos estáticos, dinámicos, web y locales en dispositivos móviles.

- Las bases de datos estáticas almacenan datos fijos que no se modifican con el tiempo. Normalmente se trata de datos históricos que pueden ser estudiados para ver su evolución en el tiempo y realizan proyecciones o tomar decisiones en base a dicha evolución.
- Por su parte, las bases de datos dinámicas almacenan información que se modifica con el tiempo. Los datos se editan y actualizan a medida que van cambiando. Por ejemplo, una base de datos con los precios de un comercio,
- Bases de datos web. El método DBaaS (Database as a Service) se basa en la contratación de proveedores de servicios para el almacenamiento de los datos. En este caso, la información se guarda en los servidores del proveedor, pudiendo acceder a ella a través de internet.

Entre las principales ventajas de las bases de datos en la nube está el ahorro de espacio físico, la disminución de los costes, o la posibilidad de acceder a los datos desde cualquier lugar o dispositivo con acceso a internet.

- Por su parte, una base de datos en local utiliza una red local (LAN), de manera que la infraestructura y la gestión de dicha base de datos se realiza en la propia organización. Solo pueden acceder a la información los equipos que estén conectados a la red local.

Proceso de programación de conexión a bases de datos estáticos en dispositivos móviles.

Son bases de datos de solo lectura. Es decir, están diseñadas para agregar datos fijos que no se pueden modificar con el tiempo. Se utilizan fundamentalmente para almacenar datos históricos o hechos invariables. Se suelen combinar diferentes bases de datos estáticas realizadas en diferentes periodos para analizar la evolución de los datos en el tiempo. Por ello, son muy usadas para hacer estudios de mercado, investigaciones estadísticas y otros proyectos relacionados con el Business Intelligence.

- La tabla estática es un objeto de proyecto de datos que define una tabla y sus fichas en tiempo de edición, que en tiempo de ejecución no se pueden modificar por el usuario.
- Para crear una tabla estática usaremos la sub-opción del mismo nombre de la opción nuevo objeto del menú objetos o a través de la galería de objetos (Tecla Insert).
- En el objeto tabla estática se define tanto las propiedades de la tabla como los registros que contiene, ya que únicamente se pueden definir los registros en tiempo de edición. En ejecución, el usuario no puede hacer alta, modificación o baja de los registros, debiendo trabajar con aquellos que haya definido el programador.

- La tabla estática se crea pulsando el botón nueva tabla estática, o la opción de menú objetos: nuevo objeto/tabla estática.
 - Propiedades.
 - Identificador

Etiqueta alfanumérica que identifica de forma unívoca una tabla estática dentro del proyecto de datos. Este identificador será el que usemos en fórmulas y para referenciarlo en otras propiedades.

Constará de mayúsculas y números exclusivamente. Al identificar de forma unívoca una tabla estática no puede haber duplicidad.
 - Nombre

Etiqueta alfanumérica que servirá como descriptor de la tabla estática. Se usará para presentar información de la tabla estática en objetos y en los inspectores.

Podemos definir una etiqueta por cada idioma presente en el proyecto.
 - Estilos

Podemos definir el estilo.
 - Privado

Limita el acceso del usuario final al objeto desde puntos donde no se haya programado el acceso al mismo.

- Comentarios

Esta propiedad nos permite documentar el uso de la tabla estática.

Proceso de programación de conexión a bases de datos dinámicos en dispositivos móviles.

Son bases de datos relacionales, es decir, en ellas se establecen relaciones entre los registros y sus campos. Están orientadas al almacenamiento de información que podría cambiar en el tiempo. Permiten añadir, modificar o eliminar la información presente en los campos de la base de datos. Se opone a las bases de datos estáticas, las cuáles no permiten editar la información.

Crear tabla dinámica.

- Para crear una tabla dinámica a partir de una tabla de datos primero debemos asegurarnos de que la tabla origen tiene encabezados de columna y de que no hay ninguna fila en blanco. Podemos crear una tabla dinámica desde cero o a partir de una serie de tablas dinámicas que Excel nos recomienda.
- Para crear una tabla dinámica desde cero usamos el comando Insertar >>> Tablas >>> Tabla dinámica.
- Para agregar o quitar un campo a la tabla dinámica en la sección Campos de la tabla dinámica, activa las casillas de los campos que deseas agregar y desactiva las casillas de los campos que deseas quitar.
- Para agregar un campo a un área específica (Filas, Columnas, Filtros, Valores), haz clic con el botón secundario en el campo y después selecciona Agregar a filtro de informe, Agregar a etiquetas de columna, Agregar a etiquetas de fila o Agregar a valores para colocar el campo en dicha área de la sección de áreas o arrastralo al área correspondiente.

- Para quitar un campo directamente de la sección de áreas, haz clic en el campo y, a continuación, haz clic en Quitar campo o arrástralo desde la sección de áreas fuera de la lista de campos.

Explicar el proceso de programación de conexión a bases de datos locales en dispositivos móviles.

Es la habilidad de recuperar la información de los sistemas de computación y/o repositorios de información sobre los dispositivos móviles en cualquier momento en cualquier lugar.

Este es un sistema distribuido que soporta conectividad móvil posee todas las capacidades de un sistema de base de datos y permiten a las unidades móviles

- Se debe tomar en cuenta
 - Desconexiones
 - Movilidad
 - Errores
 - Fallas en el dispositivo móvil
 - Se debe mantener la autonomía y la consistencia local del SMBD.

Persistencia de datos en los dispositivos móviles

La persistencia es, almacenar los datos de la aplicación para que estén disponibles al reiniciar el software Retos de la persistencia en los dispositivos móviles.

Retos de la persistencia en los dispositivos móviles.

Espacio limitado

- Almacenamiento local
 - Memoria interna/Memoria externa
 - Varios GB (no está mal)

- Seleccionar datos a persistir

Computación y memoria limitada

- Consultas complejas lentas
 - Mejor preprocesar datos en el servidor
 - Desnormalizar BBDD si es posible

- Alto consumo de batería

Restricciones de comunicación

- Dificultad para sincronizar
 - Minimizar número de conexiones
 - Sólo datos relevantes para la aplicación
 - Sólo modificaciones a los datos
 - Preprocesar datos

Desconexiones habituales

- Modo off-line. ¿Qué ocurre?
 - No funciona nada
 - Funcionamiento limitado (sólo lectura)
 - Funcionamiento completo (conflictos)
- Recuperación conexión
 - Sincronización de datos
 - Resolución de conflictos

formas de persistencia en los sistemas operativos de los dispositivos móviles.

Preferencias

- Parejas: clave -> valor
- Configuraciones, recordar acciones, etc.

Almacenamiento de archivos

- Almacenamiento interno o externo
- Archivos arbitrarios (imágenes, json, xml, texto, binarios, etc.)

Datos estructurados

- Base de datos SQL
- Colecciones de datos estructurados y relacionados

Tipos de persistencia en dispositivos móviles.

Local

- Almacenamiento en el propio móvil
- Espacio limitado. Acceso rápido. Alto coste consultas complejas

Remota

- El móvil sólo muestra datos, que lee de un servidor
- Cada acceso a datos requiere un consulta
- Alta latencia. Muy sensible a desconexiones

Cacheo/Hoarding

- Copia local de alguna información del servidor
- Comunicación con el servidor para sincronizar

- Funcional (más o menos) desconectado. Latencia según acierto.

Proceso de programación de persistencia en dispositivos móviles.

Diseño de la persistencia

- El camino sin reflexionar
 - Diseñar y crear la BBDD
 - Escribir SQL donde haga falta leer o modificar datos
 - Utilizar cursores para recorrer los resultados
- Con el paso del tiempo
 - ¿Y si cambio el nombre o tipo de una columna?
 - ¿Y si añado una tabla nueva?
 - ¿Y si cambio BBDD local por servicios web externos?

Mecanismos de tolerancia a fallos

Se deben tomar en cuenta los siguientes aspectos para el desarrollo de aplicaciones móviles:

- lentitud en las consultas
- alto consumo de batería
- dificultad de sincronización
- modo off-line
- recuperación de conexión

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

Las características básicas de la tolerancia a fallos:

- Ni un solo punto de falla - Si un sistema experimenta un fracaso, debe continuar funcionando sin interrumpirse durante el proceso de reparación
- Aislamiento de fallos en el componente que está fallando, cuando se produce un error, el sistema debe ser capaz de aislar la falla a la reincidencia. Esto requiere la adición de mecanismos de detección de fracaso dedicados que existen solamente para el propósito del aislamiento de falla. La recuperación de una condición de falla requiere la clasificación de la avería o falla en el componente. El Instituto Nacional de Estándares y Tecnología (NIST) clasifica los fallos según la localidad, causa, duración y efecto.
- La contención de fallos para evitar propagación del fallo - algunos mecanismos de fallo pueden causar fallos en el sistema mediante la propagación del fallo al resto del sistema. Un ejemplo de este tipo de fallo es el transmisor "Bribón" que puede colapsar la comunicación legítima en un sistema global y causar fallos en el sistema. Mecanismos que protejan el sistema y que aíslan un transmisor bribón o las componentes que fallan son requeridos.
- Disponibilidad de modos de reversión

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

El hardware con tolerancia a fallos requiere a veces que las piezas rotas pueden ser extraídas y reemplazadas por nuevas piezas mientras el sistema sigue funcionando (en informática conocido como sustitución en caliente). Un sistema de

este tipo con una sola copia de seguridad se conoce como único punto tolerante y representa la gran mayoría de sistemas tolerantes de fallos. En este tipo de sistemas el tiempo medio entre fallos debe ser lo suficientemente largo como para que los operadores puedan arreglar los dispositivos rotos (tiempo medio de reparación) antes de que la copia de seguridad también falle. Se recomienda que el tiempo entre fallos sea lo más largo posible, pero no es estrictamente necesario en un sistema tolerante de fallos.

La tolerancia a fallos funciona muy bien en las aplicaciones informáticas. La primera computadora con tolerancia a fallos fue SAPO en la República Checa. La empresa Tandem Computers ha basado todo su negocio en este tipo de equipos, que utilizan un único punto de tolerancia para crear sus sistemas NonStop, cuyos periodos de funcionamiento pueden medirse en años.

Los programas informáticos también pueden utilizar arquitecturas libres de fallos, por ejemplo en la replicación de procesos.

Los formatos de datos también pueden ser diseñados para degradarse correctamente. El lenguaje HTML por ejemplo, está diseñado para ser compatible, lo cual permite a los navegadores Web ignorar las nuevas entidades html que no entienden sin provocar que el documento sea inutilizable.

Bibliografía.

1.4 tecnologías de clientes ligeros. (s. f.). DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MOVILES. <http://aplicacionesdispmoviles.blogspot.com/2015/12/14-tecnologias-de-clientes-ligeros.html>

Bases de datos en dispositivos móviles - Introducción. (s. f.). Jose Manuel Ayala Wilson. https://jmaw.blogspot.com/2012/07/bases-de-datos-en-dispositivos-moviles_23.html

ciberseg1922. (2021, 16 de diciembre). *Tolerancia a fallos, qué es y técnicas* | Ciberseguridad. Ciberseguridad. <https://ciberseguridad.com/guias/prevencion-proteccion/tolerancia-fallos/>

(s. f.-a).

(s. f.-b).

(s. f.-c).

Tablado, F. (2020a, 7 de septiembre). *Bases de datos dinámicas. ¿qué son? Ejemplos* | ayuda ley protección datos. Ayuda Ley Protección Datos. <https://ayudaleyprotecciondatos.es/bases-de-datos/dinamicas/>

Tablado, F. (2020b, 7 de septiembre). *Bases de datos estáticas. ¿qué son? Ejemplos* | ayuda ley protección datos. Ayuda Ley Protección Datos. <https://ayudaleyprotecciondatos.es/bases-de-datos/estaticas/#:~:text=de%20datos%20estáticas?-,Definición,de%20datos%20de%20solo%20lectura.>

Tablado, F. (2020c, 7 de septiembre). *Bases de datos. ¿Qué es? Tipos y ejemplos* | AyudaLeyDatos. Ayuda Ley Protección Datos. <https://ayudaleyprotecciondatos.es/bases-de-datos/#:~:text=acceso%20a%20internet.-,Local,conectados%20a%20la%20red%20local.>

Unidad 4: Administración de datos en dispositivos móviles. (s. f.). Share and Discover Knowledge on SlideShare. https://es.slideshare.net/juan_anaya/unidad-4-administracin-de-datos-en-dispositivos-mviles

Unidad 5 ADMINISTRACIÓN DE DATOS EN DISPOSITIVOS MÓVILES. (s. f.). Desarrollo De Aplicaciones Para Dispositivos Moviles. <http://androidittla.blogspot.com/2014/05/unidad-5-administracion-de-datos-en.html>