

# Universidad Politécnica de Tecámac

## Tareas Móviles – Tarea 1

Torres Servín Emmanuel

4322IS

- Cosmes Santes Carlos Gerardo
- Martínez Osorio Brayan Javier
- Rivero Montijo Maribel

## Introducción

En el siguiente documento se pretende mostrar el resultado de múltiples investigaciones acerca de diversos temas relacionados con la materia de programación visual.

La investigación tiene suma relevancia al ser parte del desarrollo de los conocimientos para el futuro de la carrera, ya que estos serán conceptos fundamentales para el desarrollo en múltiples áreas como lo son los dispositivos movibles, manejo de datos, fallos en diversas áreas de desarrollo con manejo de datos.

La información es la base de las aplicaciones pues sin información ¿qué es una aplicación? De igual manera que existen montones de aplicaciones para dispositivos móviles también existen diferentes formas de manipular datos por eso existe la necesidad de la investigación pues se necesita reconocer conceptos como la conexión de base de datos los tipos de conexión los procesos de conexión a diferentes tipos de base de datos como los son la base de datos dinámicas y locales.

Pero aquí no acaba la cosa pues si existe el manejo de datos a gran escala se obtienen múltiples problemas que sean solucionados a lo largo del tiempo es fundamental conocer cómo es que el manejo de datos puede persistir en los dispositivos móviles, las formas en la que se hace y los tipos de persistencia existen como la local, remota cache etc.

Como se mencionó anterior mente el manejo de una gran cantidad de datos puede generar fallos en dichos datos ejemplo corrupción en los datos, estos problemas causan una gran pérdida tanto para el usuario como para las compañías desarrolladoras por eso existen los mecanismos de tolerancia a fallos los cuales podrían llegar a solucionar este gran problema si se implementan de forma correcta.

## Modelado de objetos de acceso a datos en dispositivos móviles

### Modelado de objetos 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.

Todas las bases de datos móviles tienen una arquitectura similar, donde debemos distinguir una serie de elementos principales característicos de este tipo de sistemas:

Servidor de base de datos corporativo y SMBD que gestiona y almacena los datos corporativos y proporciona aplicaciones corporativas. Se correspondería con el SMBD y el servidor que se utiliza en la organización.

Base de datos remota y SMBD que gestiona y almacena los datos móviles. Son las bases de datos que deben estar implementadas en los dispositivos móviles.

Plataforma de base de datos móvil, que puede ser un ordenador portátil, PDA u otro dispositivo de acceso a Internet, es decir, los dispositivos móviles en cuestión.

Enlaces de comunicación bidireccionales entre el SMBD corporativo y el SMBD móvil. Que pueden ser redes inalámbricas de distinta naturaleza, comunicaciones vía satélite, etc.

Las tecnologías de la información y la comunicación favorecen el acceso universal a la educación, la igualdad en el ejercicio de la enseñanza, el aprendizaje de calidad y el desarrollo profesional de las personas y dentro de ellos están los cursos introductorios a ciencias de la computación,

La solución de la administración de dispositivos móviles de AirWatch (MDM) le permite administrar las implementaciones a gran escala de los dispositivos móviles.

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

En el software informático, un objeto de acceso a datos (DAO) es un componente de software que proporciona una interfaz común entre una aplicación y uno o más dispositivos de almacenamiento de datos, como bases de datos o archivos.

- Servidor de base de datos corporativo y SMBD que gestiona y almacena los datos corporativos y proporciona aplicaciones corporativas. Se correspondería con el SMBD y el servidor que se utiliza en la organización.

- Base de datos remota y SMBD que gestiona y almacena los datos móviles.

Son las bases de datos que deben estar implementadas en los dispositivos móviles.

- Plataforma de base de datos móvil, que puede ser un ordenador portátil, PDAAu otro dispositivo de acceso a Internet, es decir, los dispositivos móviles en cuestión.

- Enlaces de comunicación bidireccionales entre el SMBD corporativo y el SMBD móvil. Que pueden ser redes inalámbricas de distinta naturaleza, comunicaciones vía satélite,etc

## **DB2 Everyplace**

Solución de IBM para procesos de informática distribuida Los profesionales que se desplazan con frecuencia pueden tener acceso a datos vitales que necesitan mientras están lejos de su centro de trabajo

El Adaptador de Archivos le permite distribuir archivos y aplicaciones hacia usuarios portátiles.

Permite una empresa sin cables para sus clientes y socios mediante soluciones de software de integración y desarrollo, infraestructura móvil.

Incluyendo procedimientos almacenados, triggers, integridad referencial, bloqueo a nivel de fila, recuperación automática, y mucho más.

También es muy eficiente a la hora de consumir recursos ya que necesita sólo aproximadamente 1MB RAM, y 2KB por cada conexión cliente

### **SQL SERVER CE**

Permite implementar la funcionalidad de una base de datos relacional y tiene la capacidad de brindar Acceso a los datos de una manera flexible y similar a

SQL Server, al mismo tiempo que permite conectarse a un servidor SQL Server central y mantener actualizada la información

### **SQLite**

Es un sistema de gestión de bases de datos relacional compatible con ACID

la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo.

El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones

reduciendo la latencia en el acceso a las BD En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB

- 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.

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.

Dentro de los dispositivos móviles distintos tipos de métodos para el acceso a la información desde

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

procesamiento de datos que caben en la palma de la mano, mismos que son conocidos como dispositivos móviles. Entre estos dispositivos se destacan los PDA (ayudante personal digital por sus siglas en inglés) y los Smartphones.

Los dispositivos móviles constituyen poderosas herramientas, que aunadas con recursos humano se vuelve eficientes para el ser humana, en su día a día.

Interfaz de programación de aplicaciones o API es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las "librerías".

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.

## Manipulación de datos en dispositivos móviles

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.

- Reconocer el concepto de conexión a bases de datos.
  - Las bases de datos contienen datos empresariales que graban los flujos de mensaje desplegados y a los que se accede a través de los mismos. Debe crear conexiones desde el nodo de integración a la base de datos utilizando ODBC o JDBC.
  - es la forma que un servidor de base de datos y su software cliente se comunican entre sí. El cliente y el servidor
  - El cliente utiliza una conexión a base de datos para enviar comando y recibir respuestas del servidor. Una base de datos es almacenada como un fichero o como un conjunto de ficheros en un disco.



- Describir la conexión a bases de datos estáticos, dinámicos, web y locales en dispositivos móviles.
- La conexión a bases de datos estáticos
  - conexión a base de datos es un archivo de configuración donde se especifica los detalles físicos de una base de datos
  - Son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos
- La conexión a bases de datos dinámicos
  - Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta.
- La conexión a bases de datos web
  - Todos los sitios web, incluso los más pequeños, utilizan datos, normalmente a través de bases de datos MySQL, para generar dinámicamente contenido en sus páginas.
  - Almacena la conexión en una variable llamada \$connection: `$connection = mysql_connect("servidor," "usuario," "contraseña") or die(mysql_error());` Cambia "servidor" por el nombre del servidor de tu sitio web (normalmente es "localhost" pero no siempre) y cambia "usuario" y "contraseña" por el nombre de usuario y contraseña de la base de datos.
- La conexión a bases de datos locales
  - Puede utilizar un DSN (Nombre de fuente de datos, Data Source Name) o una cadena de conexión para conectarse a la base de datos.
  - contiene codificada toda la información referente a la IP o nombre del servidor y los nombres de la(s) base(s) de datos a las que cada programa tendrá acceso, a modo que, una vez creado el acceso, podrás acceder a la base de datos en forma automática.

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

Base de datos móvil es una base de datos que se puede conectar a un

dispositivo informático móvil a través de una red móvil (o red inalámbrica). bases de datos móviles basadas en Android, bases de datos móviles basadas en:

iOS, etc. Ejemplos comunes de bases de datos son Couch base Lite, Object Box, etc.

- Hosts fijos:
  - Las transacciones y las funciones de gestión de datos
- Estaciones base:
  - instalación de radios bidireccionales en ubicaciones fijas
- Explicar el proceso de programación de conexión a bases de datos dinámicos en dispositivos móviles.
  - Unidades móviles
    - Son computadoras portátiles que se mueven por una región geográficas
  - a API difiere de una interfaz de aplicación binaria (ABI) en la que el primero es de código abierto basado mientras que el segundo es una interfaz binaria.
  - Bases de datos locales o regionales
  - La unidad virtual, es la unidad lógica que se le asigna a un disco virtual (virtual drive).
- Explicar el proceso de programación de conexión a bases de datos locales en dispositivos móviles.

**Máquina Virtual Dalvik**, que permite traducir a bytecode (compilación de código intermedio antes del código binario o de máquina) y ejecutar aplicaciones programadas en Java, lenguaje que se utiliza para programar en Android.

- Las **bases de datos** contienen **datos** empresariales que graban los flujos **de** mensaje desplegados y a los que se accede a través **de** los mismos
- Patrón de Diseño Core J2EE y considerados una buena práctica.
- usarse en Java para aislar a una aplicación de la tecnología

## Persistencia de datos en los dispositivos móviles

Las plataformas de desarrollo para móviles ofrecen distintas alternativas para la persistencia de datos, que permiten almacenar desde las preferencias de la aplicación hasta documentos o datos complejos. No solo necesitaremos almacenar los datos en el móvil, en algunos casos nos interesará guardarlos en un servidor

- Concepto de persistencia en dispositivos móviles.

La persistencia en el ámbito de una aplicación indistintamente si es una aplicación Android o de cualquier otro tipo consiste en que los datos manipulados por la aplicación "sobrevivan" a la ejecución de la misma en el tiempo; en otras palabras; consiste en almacenar los datos en un medio secundario, no volátil para posterior reconstrucción y utilización; por lo tanto, son independientes en el tiempo del proceso que los creó.

- Retos de la persistencia en los dispositivos móviles.

- Formas de persistencia en los sistemas operativos de los dispositivos móviles: preferencias, almacenamiento de archivos, datos estructurados.

- Persistencia básica en iOS
  - *property lists* o bien el sistema de preferencias de iOS.
  - directorios dentro de su *sandbox*.
  - a estructura "estándar" del *sandbox* y luego el API para abrir, crear y modificar archivos y directorios.
- Persistencia en Android
  - el método `OutputStreamWriter` para escribir archivos en Android a través de la apertura del mismo con el método `openFileOutput`;
  - `MODE_PRIVATE`, `MODE_APPEND`, `MODE_WORLD_READABLE`
  - solicitados en el Manifest

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

- Tipos de persistencia: local, remota y de Cacheo/Hoarding en dispositivos móviles.
  - Tipos de persistencia Local
    - consultas o búsquedas podemos usar APIs
    - Memoria Externa (en la mayoría de los casos, la memoria SD) con los permisos solicitados en el Manifest
  - Tipos de persistencia Remota
    - Directorios, *sandbox*
    - Servidor en ambos casos
  - Tipos de persistencia Cacheo
    - caché electrónico son solo una copia
    - Core Data
  - Tipos de persistencia Hoarding
    - crear *fetch request* que devuelvan todas las instancias de una entidad
    - NSPredicate
- Proceso de programación de persistencia en dispositivos móviles.
  - MODE\_PRIVATE: Sólo nuestra aplicación tiene acceso a estas preferencias.
  - MODE\_WORLD\_READABLE: Todas las aplicaciones pueden leer estas preferencias, pero sólo la nuestra puede modificarlas (deprecated desde el API 17).
  - MODE\_WORLD\_WRITEABLE: Todas las aplicaciones pueden leer y modificar estas preferencias (deprecated desde el API 17).
  - ```
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Función `Environment.getExternalStorageStatus()`:

- `Environment.MEDIA_MOUNTED`:
- Indica que la memoria está disponible y "montada" en el dispositivo; por lo tanto, podemos acceder a la misma.
- `Environment.MEDIA_MOUNTED_READ_ONLY`:
- Indica que la memoria está disponible y "montada" en el dispositivo; pero sólo podemos leer en ella.
- `Environment.MEDIA_UNMOUNTED`, `Environment.MEDIA_REMOVED`...
- Los demás estatus indican que la memoria no está disponible y que por lo tanto no podemos ni leer ni escribir en la misma.

## Mecanismos de tolerancia a fallos

"Tolerancia Frente a Fallos", o Fault Tolerance en inglés, tiene su origen en el procesamiento de datos y describe la posibilidad de que un sistema siga funcionando de forma continua incluso en caso de entrada incorrecta o fallo.

permite su buen funcionamiento incluso después de que ocurre una falla en algunos de sus componentes. Un diseño tolerante a fallas puede causar una reducción en el nivel de productividad o un mayor tiempo de respuesta. Sin embargo, se asegura de que todo el sistema no falle.

- Elementos para tomar en cuenta en el desarrollo de aplicaciones orientadas a móviles:

Podemos definir el desarrollo de aplicaciones móviles como el procedimiento de creación de software para dispositivos inalámbricos electrónicos (smartphone o tablets).

### ➤ Lentitud en las consultas:

- Un plan no óptimo generado por SQL Query Optimizer puede ser la causa de un rendimiento lento de las consultas.
- los problemas relacionados con la ejecución normalmente están relacionados con problemas de compilación, lo que produce un plan de consulta que no es óptimo o problemas de ejecución relacionados con un número insuficiente de recursos o una sobreutilización de estos.

### ➤ Alto consumo de batería:

- inmediato de la batería es por culpa de la **CPU**.
- El alto consumo de la CPU provoca el alto consumo de batería, aparte de que el teléfono se caliente mucho

### ➤ Dificultad de sincronización:

- tu información en todos tus dispositivos. Si la sincronización deja de funcionar o se detiene, prueba estas soluciones.

➤ **Modo off-line:**

- El modo sin conexión es un complemento que le permite desconectar fácilmente su navegador de Internet.
- Incluido en muchas aplicaciones que dependen de una conexión a Internet, el modo fuera de línea es un método para usar un servicio sin conexión a Internet. Por ejemplo, los servicios de transmisión, como Netflix, Spotify y YouTube, transmiten contenido desde sus servidores y dependen de una conexión constante a Internet.
- El botón de la barra de herramientas sirve como un interruptor ON|OFF. Cuando el complemento está activo, el ícono de la barra de herramientas es gris oscuro y cuando está inactivo, el botón es gris claro. Al presionar el botón de la barra de herramientas se alternará el modo sin conexión. Cuando el modo fuera de línea está activo

➤ **Recuperación de conexión:**

- Una *consulta correlacionada* se utiliza para recuperar una o más filas de una conexión de datos para un campo de objeto de negocio, basándose en una consulta completa SQL SELECT o una llamada de procedimiento almacenada.
- La restauración de datos mientras la base de datos está en línea se denomina *restauración en línea*. Se considera que una base de datos está en línea siempre que el grupo de archivos principal esté en línea, aunque alguno de los grupos de archivos secundarios esté sin conexión.

- Proceso de selección de los mecanismos de tolerancia a fallos en el desarrollo de aplicaciones de dispositivos móviles.
- La tecnología, los sistemas tolerantes a fallas se diseñaron para dar alarmas al usuario o al operador sobre la posible falla. Se suponía que el operador actuaría sobre la alarma y aclararía las cosas antes de que ocurriera una avería importante.
- Los sistemas, ya sean hardware o software, están diseñados para resolver problemas de forma independiente sin mucha interferencia humana, a menos que sea un problema importante que requiera atención inmediata.
- Sistemas de hardware respaldados por sistemas idénticos o equivalentes. Por ejemplo, un servidor puede ser tolerante a fallas utilizando un servidor idéntico que se ejecuta en paralelo, con todas las operaciones reflejadas en el servidor de respaldo.
- Sistemas de software respaldados por otras instancias de software. Por ejemplo, una base de datos con información del cliente se puede replicar continuamente en otra máquina. Si la base de datos principal deja de funcionar, las operaciones se pueden redirigir automáticamente a la segunda base de datos.
- Fuentes de energía que se hacen tolerantes a fallas utilizando fuentes alternativas. Por ejemplo, muchas organizaciones tienen generadores de energía que pueden asumir el control en caso de que falle la línea principal de electricidad.



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

los sistemas tolerantes a fallas se diseñaron para dar alarmas al usuario o al operador sobre la posible falla.

La tolerancia a fallas se refiere a la capacidad de un sistema (computadora, red, clúster de nube, etc.)

La alta disponibilidad se refiere a la capacidad de un sistema para evitar la pérdida de servicio minimizando el tiempo de inactividad.

- Tiempo de inactividad: un sistema de alta disponibilidad tiene un nivel mínimo permitido de interrupción del servicio.
- Alcance: la alta disponibilidad se basa en un conjunto compartido de recursos que se utilizan en conjunto para administrar fallas y minimizar el tiempo de inactividad.
- La tolerancia a fallas se basa en las copias de seguridad de la fuente de alimentación, así como en el hardware o software que puede detectar fallas y cambiar instantáneamente a componentes redundantes.
- Coste: un sistema tolerante a fallas puede ser costoso, ya que requiere la operación y el mantenimiento continuos de componentes adicionales redundantes.
- Por lo general, la alta disponibilidad forma parte de un paquete general a través de un proveedor de servicios (por ejemplo, un proveedor de equilibrador de carga).

La tolerancia a fallas de hardware para el software requiere lo siguiente.

- Respaldo Redundancia
- La copia de seguridad

mantiene información en caso de que deba reemplazarse el hardware, esto se puede hacer de dos formas.

Copia de seguridad programada automática mediante software

Copia de seguridad manual en un horario regular

Restauración de información

## Referencias

Alvarez, M. A. (2022, 31 julio). *13. Que es el Acceso a datos en los dispositivos móviles?*

*Explique qué es RMS, Controles de lista. Enlace a datos. ADO.Net. Proveedor de datos bajo entorno móvil.* Acceso de datos. Recuperado 2022, de

[https://maricelaariasistencia.blogspot.com/2015/08/13-que-es-el-acceso-datos-en-los\\_25.html](https://maricelaariasistencia.blogspot.com/2015/08/13-que-es-el-acceso-datos-en-los_25.html)

C. (2021, 16 diciembre). *Tolerancia a fallos, qué es y técnicas.* Ciberseguridad. Recuperado

2022, de <https://ciberseguridad.com/guias/prevencion-proteccion/tolerancia-fallos/>

Computer Hope. (2020, 30 junio). *What is Offline Mode?* Mecanismos Tolerancia. Recuperado

2022, de <https://www.computerhope.com/jargon/o/offline-mode.htm>

Hall, J. (2021). *6.- Base de Datos para Dispositivos Moviles - SMBD MÃ³viles - Ingenieria en*

*SIstemas Computacionales.* StuDocu. Recuperado 2022, de <https://www.studocu.com/es-mx/document/instituto-tecnologico-superior-de-alvarado/ingenieria-de-sistemas/6-base-de-datos-para-dispositivos-moviles-smbd-moviles-ingenieria-en-sistemas-computacionales/22806072>

M. (2022a, junio 7). *Restauración con conexión (SQL Server) - SQL Server.* Microsoft Docs.

Recuperado 2022, de <https://docs.microsoft.com/es-es/sql/relational-databases/backup-restore/online-restore-sql-server?view=sql-server-ver16>

Microsoft. (2022). *Problemas de sincronización con el cliente de sincronización de OneDrive*

*durante la carga de la carpeta.* Recuperado 2022, de <https://support.microsoft.com/es->

[es/office/problemas-de-sincronizaci%C3%B3n-con-el-cliente-de-sincronizaci%C3%B3n-de-onedrive-durante-la-carga-de-la-carpeta-a9b5ce42-6097-4503-9c0a-f28a50e48b9d](https://office.problemas-de-sincronizaci%C3%B3n-con-el-cliente-de-sincronizaci%C3%B3n-de-onedrive-durante-la-carga-de-la-carpeta-a9b5ce42-6097-4503-9c0a-f28a50e48b9d)

N. (2022b, mayo 24). *Tipos de problemas de rendimiento de consultas - Azure SQL Database & Azure SQL Managed Instance*. Microsoft Docs. Recuperado 2022, de <https://docs.microsoft.com/es-es/azure/azure-sql/identify-query-performance-issues?view=azuresql>

*Objeto de acceso a datos - Ensayos universitarios - 554 Palabras*. (2021). Buenas Tareas. Recuperado 30 de julio de 2022, de <https://www.buenastareas.com/ensayos/Objeto-De-Acceso-a-Datos/2141318.html>

Obregón, H. (2006). *Desarrollo de Aplicaciones Móviles. Elementos a Considerar*. SG Buzz. Recuperado 2022, de <https://sg.com.mx/content/view/490>

*Recuperación de datos de una conexión de datos utilizando una consulta correlacionada*. (2020). © Copyright IBM Corp. 2003, 2012. Recuperado 2022, de <https://www.ibm.com/docs/es/odm/8.5?topic=rdfdcbum-retrieving-data-from-data-connection-by-using-mapped-query>

*Solucionar problemas con la sincronización en Chrome - Ayuda de Google Chrome*. (2022). Ayuda Google Chrome. Recuperado 30 de julio de 2022, de <https://support.google.com/chrome/answer/9175737?hl=es>

Z. (2018, 28 noviembre). *¿Consumo de batería excesivo? Explicación y solución*. Androidsis. Recuperado 2022, de <https://www.androidsis.com/consumo-de-bateria-excesivo-explicacion-y-solucion/#:%7E:text=Para%20resumir%2C%20el%20principal%20culpable%20de%20es>

[e%20consumo,parte%20de%20que%20el%20tel%C3%A9fono%20se%20caliente%20m  
ucho.](#)

# Universidad Politecnica de Tecámac

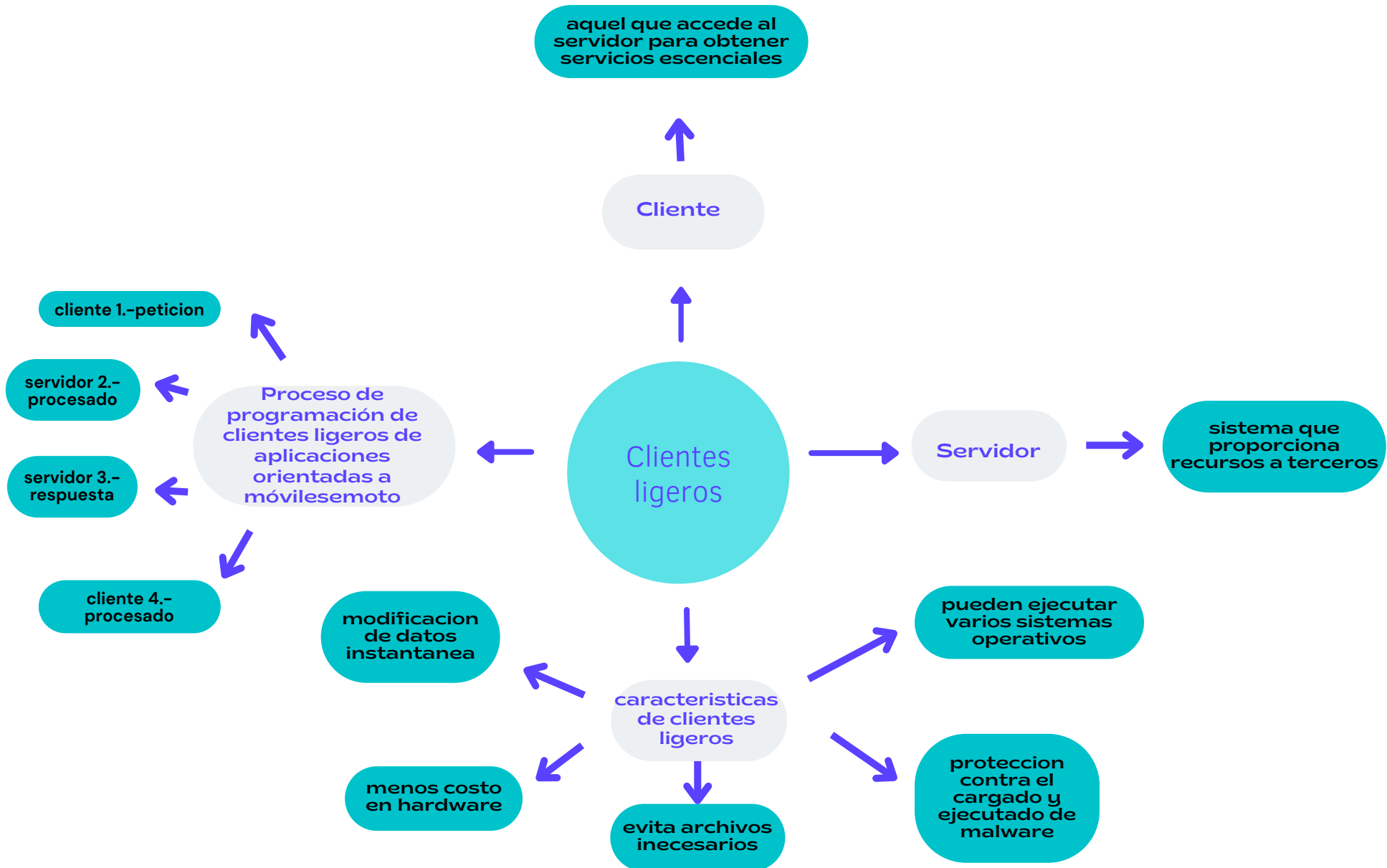
- Cosmes Santes Carlos Gerardo
- Martinez Osorio Brayan Javier
- Rivero Montijo Marivel

4322IS

Torres Servin Emmanuel

## Tarea 2

# Clientes ligeros



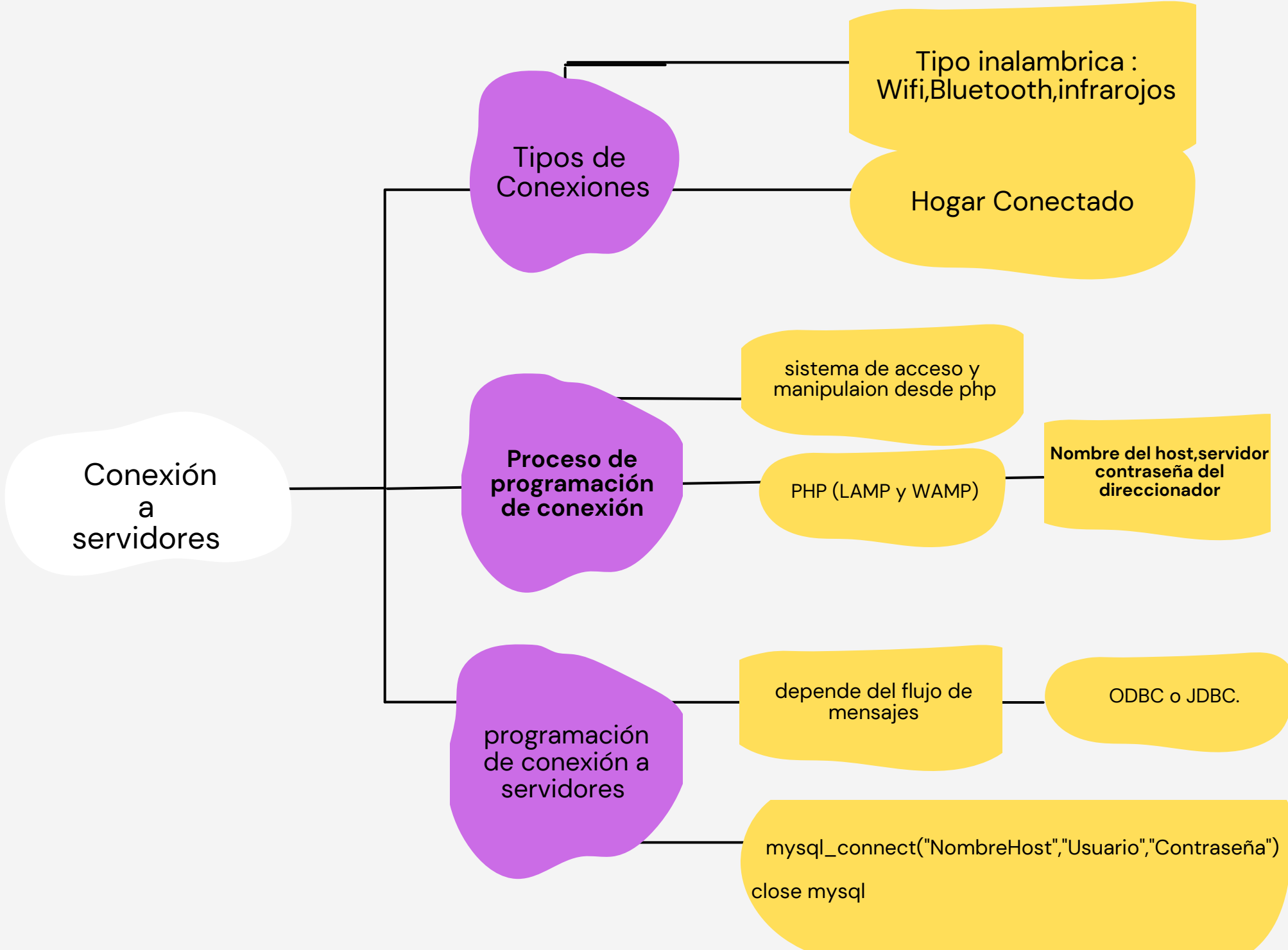
# Universidad Politecnica de Tecámac

- Cosmes Santes Carlos Gerardo
- Martinez Osorio Brayan Javier
- Rivero Montijo Marivel

4322IS

Torres Servin Emmanuel

## Tarea 3





# Universidad Politecnica de Tecámac

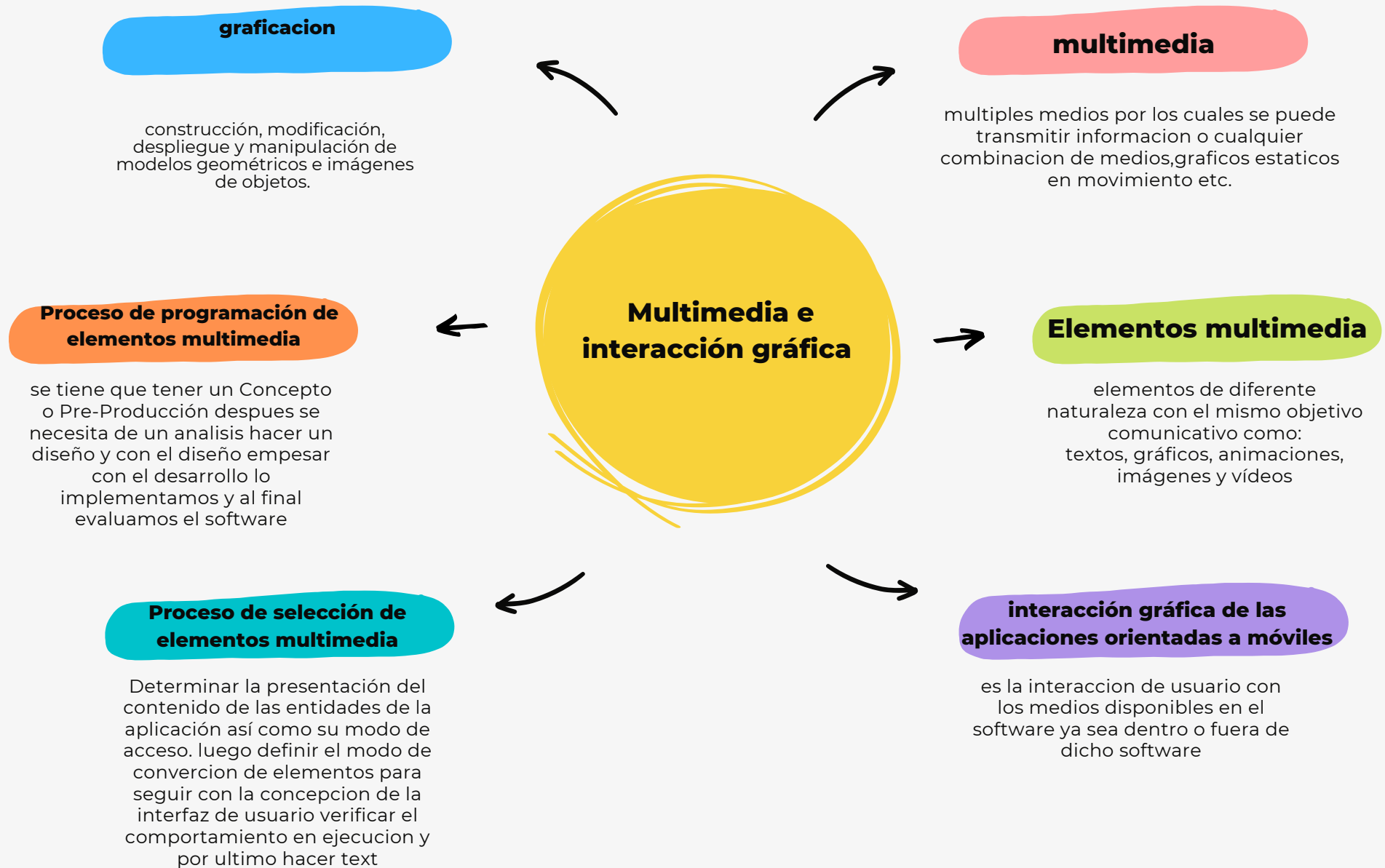
- Cosmes Santes Carlos Gerardo
- Martinez Osorio Brayan Javier
- Rivero Montijo Marivel

4322IS

Torres Servin Emmanuel

## Tarea 4

# Gráfico de Mapa Mental



# Tarea 5. Aplicaciones móviles dependientes del contexto y sensores embebidos en los dispositivos móviles

---

COSME SANTES CARLOS GERARDO  
MARTINEZ OSORIO BRAYAN JAVIER  
RIVERO MONTIJO MARIBEL

## APLICACIONES MOVILES DEPENDIENTES DEL CONTEXTO

El desarrollo en las innovaciones móviles avanza muy rápido. La evolución ha demostrado que un dispositivo es cada vez más necesario, tanto en los avances respecto al hardware que tiene cada dispositivo, como en la capacidad de almacenar información, capacidad de procesamiento entre otras cosas.

- Son cada vez más sofisticadas / necesarias
- Se busca una innovación
- Gama alta en pantallas táctiles
- GPS
- Sensores de movimiento

Se busca tener ámbitos para el desarrollo del trabajo y la administración de servicios de mantenimiento para tener un mejor balanceo y almacenamiento de información, con las TI se enfrentan retos para crear soluciones y hacer una correcta elección de elementos, así como el proceso de agregarlas con el contexto.

Conectividad es una solución móvil, con una extensión de sistemas empresariales como ERPs o CRMs, es fundamental las opciones de conectividad disponibles en el mercado y el impacto que tenga sobre la aplicación.



Sincronización de datos aunque es precisamente para una aplicación conectada, se vuelve fundamental por las buenas estrategias con las que cuenta, ya que nos brinda infraestructura de sincronización en motores de base de datos, es muy completo para el desarrollo de aplicaciones dependientes en los móviles, además de encargarse de

resolver los problemas más importantes y que estén afectando para que este trabajo de una manera óptima, así como el plus de monitorear el estatus de la sincronización.

Además de ofrecer una forma de programar basada en SQL en el dispositivo móvil.

Soporte la mayoría de usuarios están dispersos geográficamente y es más complicado darles un soporte, como posibles actualizaciones de las aplicaciones o bien otorgar el apoyo para resolver dudas, lo que puede complicarlo y hacerlas costosas, es por eso que es indispensable que se cuente con una estrategia de

soporte desde el inicio basada en herramientas que permitan administrar fácilmente la ubicación geográfica de los dispositivos y brindarles un buen servicio, atender remotamente a los usuarios.

Interfaz de usuario, de los errores más comunes es cuando el programador viene de computador a abordar un proyecto de aplicaciones para dispositivos móviles, ya que subestiman las diferencias de interfaz lo que hace que en el dispositivo móvil están restringidos en la pantalla y en las formas en las que aceptan entradas a los usuarios, lo que se busca es que se deba pensar en una interfaz muy sencilla y parecida a las demás aplicaciones que ya existen en un dispositivo móvil, así como minimizar el número de entradas por parte del usuario, ya que la aplicación debe de estar preparada para cualquier movimiento de parte del usuario sin que se pierda la información.

Plataforma, en general las plataformas mas comunes para el desarrollo de aplicaciones móviles son J2Me (Java 2 Micro Edition) y el .NET Compact Framework para Windows Mobile, ya que ofrecen ventajas sobre alternativas disponibles:

- Capacidad de rehusar el conocimiento de desarrollo existente en lenguajes .NET para el escritorio.
- Excelente desempeño y velocidad de desarrollo
- Facilidad para interactuar con aplicaciones corporativas gracias a una infraestructura muy completa para el manejo de XML y el desarrollo de clientes SOAP.
- Integración simple con SQL Server CE
- Posibilidad de desarrollar en Visual Studio .NET

Todo hardware necesita de un software que lo gestione. Sin un sistema operativo adecuado se desperdicia la gran capacidad potencial que ofrecen los teléfonos móviles.



Los dispositivos móviles cada día son mas avanzados y cuentan con capacidades cada vez más inteligentes, la geolocalización es ocupada para conocer la posición geográfica del dispositivo, a través de coordenadas de un satélite.

Las industrias móviles cuentan con infinitudes de geolocalizaciones y hay diversas maneras en las que la función ayude al dispositivo. Pues bien, ahora esta es muy

ocupada en nuestra vida cotidiana cuando queremos ordenar comida, comprar algún objeto o en la búsqueda de información general, esto con el fin de encontrar comercios cercanos, restaurantes, cines, gasolineras, tiendas departamentales, etc.



Dentro de las aplicaciones mas comunes que son basadas en servicios de geolocalización esta: Google Maps, Apple Maps, Waze o Google Earth, basadas a ofrecer información del trafico en una ciudad, las mejores rutas, tiempos, distintos transportes para llegar, etc.

En la actualidad las aplicaciones en móviles son mas necesarias, para el contexto del usuario y el desarrollo de aplicaciones dependientes del contexto en los dispositivos móviles.

Se busca el software y hardware que se adapte a las aplicaciones moviles, con el soporte y la interfaz del usuario

## **Sensores embebidos en los dispositivos móviles**

### **• Elementos y características de los sensores embebidos en los dispositivos móviles**

#### **Característica**

Se emplea uno o varios procesadores digitales (CPUs) en formato microprocesador, microcontrolador o DSP lo que le permite aportar 'inteligencia' al sistema anfitrión al que ayuda a gobernar y del que forma parte desarrolla funciones específicas.



Proporcionan información que ayudan a otros componentes generando un sistema más complejo

Se utilizan para poder determinar el estado del proceso donde están instalados. Ellos transforman las variaciones de la magnitud a medir en una señal eléctrica acondicionada de tal manera que pueda ser recibida en su destino.

Los sensores se caracterizan en función del valor de algunos de los parámetros

- Características de entrada
- Características de transferencia
- Características de salida

**Distancia:** Es un valor máximo o mínimo de lo físico que es la variable que el sensor puede medir o detectar

**Lapso:** es el intervalo con el que se detecta dicha variable

**Exactitud:** los errores en la medición se determinan en términos de presión, y son la diferencia entre el valor medio y el valor verdadero se define en una escala de infinitos elementos

**Presión:** es la cercanía entre un conjunto de valores

**Sensibilidad:** es la relación entre el cambio de salida y el de entrada esto se puede expresar como

$$S = \frac{dY}{dX}$$

**Linealidad:** es la desviación de los valores de un sensor entre los valores medios y la curva ideal

**Histéresis:** es la diferencia en la salida cuando los valores de entrada varían (disminuye o aumentan)

**Resolución:** es el cambio mínimo que se puede detectar con el sensor

**Reproductibilidad:** la capacidad de sensor para enviar los mismos datos de salida al recibir los mismos datos de entrada



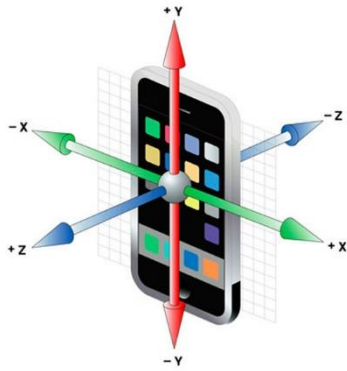
**Repetibilidad:** como en el punto anterior solo que esta vez se toma en cuenta otros factores como el ambiente, condiciones físicas de medición etc.

**Tiempo de respuesta:** es el momento en el que la salida alcanza un cierto porcentaje de su valor final, en respuesta de un cambio en la entrada.

## **Tipos**

Existen dos tipos los moduladores y los generadores según el aporte de energía. En los moduladores o activos, la energía de la señal de energía procede en, su mayor parte, de una fuente de energía auxiliar. La entrada solo controla la salida. En cambio, en los generadores o pasivos, la energía de salida es suministrada por la entrada.

Los sensores moduladores requieren más hilos que los generadores pues la energía de alimentación suele suministrarse mediante hilos distintos a los empleados para las señales. Por contra, su sensibilidad se puede modificar a través de la señal de alimentación, lo que no permiten los sensores generadores.



Según la señal de salida, los sensores se clasifican en analógicos o digitales. En los analógicos la salida varía de forma continua y la información está en la amplitud. Si es en forma de frecuencia, se denominan casi digitales, por la facilidad con que se puede convertir en una salida digital.

En los sensores digitales la salida varía en forma de saltos o pasos discretos. No requieren conversión A/D y la transición de su salida es más fácil. Tienen mayor fidelidad y fiabilidad, y muchas veces mayor exactitud, pero no hay modelos digitales para muchas de las magnitudes físicas de mayor interés.

Atendiendo al modo de funcionamiento, los sensores pueden ser de deflexión o de comparación. En los sensores por deflexión, la magnitud medida produce algún defecto físico, que engendra algún defecto similar, pero opuesto, en alguna parte del instrumento, y que está relacionado con alguna variable útil.

En los sensores que funcionan por comparación se intenta mantener nula la deflexión mediante la aplicación de un efecto bien conocido, opuesto al generado por la magnitud a medir. Hay un detector del desequilibrio y un medio para restablecerlo.

Las medidas por comparación suelen ser más exactas porque el efecto conocido opuesto se puede calibrar con un patrón o magnitud de referencia de calidad. El detector de desequilibrio solo mide alrededor de cero y puede ser muy sensible sin necesidad de estar calibrado, pero tiene menor respuesta dinámica y no tan rápida como en los de deflexión.





- **Proceso de programación de aplicaciones de acuerdo con sensores embebidos de móviles.**

Definir la aplicación: se necesita tener en claro cuál es el futuro trabajo del sensor, pues el sensor puede realizar diferentes funciones así que es importante tener en claro que se va a realizar.

Propuesta del proyecto: para plantearnos el inicio de cualquier proyecto es necesario que exista una propuesta o iniciativa que fije un objetivo. En esta fase se debe realizar un análisis del coste-beneficio que nos va a aportar el proyecto; además, se deben estimar los esfuerzos implicados en el proyecto.

Aprobación del proyecto: una vez propuesto, se debe seguir el proceso establecido por cada organización para la aprobación del mismo por parte de sus responsables.

Inicio del proyecto: en este punto se deben definir las actividades del proceso, se deben planificar las mismas en un cronograma y se debe realizar la asignación presupuestaria.

Ejecución del proyecto: durante esta etapa se deben llevar a cabo las actividades del proceso según lo previsto en el plan de ejecución. Es importante realizar un seguimiento del avance de las mismas, así como valorar y solucionar los problemas que aparezcan hasta culminar en la entrega y aceptación de los resultados. De todas maneras, aunque a veces no se le dé la importancia que realmente tiene, uno de los puntos más importantes que tiene lugar en esta fase es la consecución exhaustiva de los requerimientos del proyecto.

Cierre del proyecto: en este punto se evaluará la actuación de los participantes y los resultados, y se elaborará un informe final.

Dentro de la fase de ejecución del proyecto se realiza la secuencia de tareas que darán como resultado la aplicación informática definida en el proyecto. Las fases principales durante la ejecución del proyecto son:

Recogida de requerimientos: en esta fase los analistas se reúnen con el responsable funcional de la aplicación (RFA) para recoger las necesidades que debe satisfacer la aplicación. Como resultado se obtiene una matriz con todos los requerimientos ordenados y numerados donde se recoge el grado de cumplimiento de la aplicación para cada uno de ellos (total / parcial / nulo). Este documento tiene que ser validado por el RFA antes de iniciar la siguiente fase.

Análisis funcional: en esta fase se elabora un documento llamado análisis funcional donde se explica la funcionalidad que ofrecerá el software a partir de la matriz de

requerimientos, sin entrar en detalles técnicos. En este documento también aparecen los casos de uso del desarrollo, donde se puede validar la forma de integrar a los diferentes actores que van a intervenir en la aplicación. Este documento debe ser aprobado por el responsable de la aplicación antes de pasar a la siguiente fase.



**Diseño técnico:** una vez validado el análisis funcional se inicia esta fase, que tiene como objetivo el diseño técnico de la aplicación, que conseguirá resolver las necesidades funcionales planteadas en la fase anterior. En esta fase tiene lugar el diseño de las estructuras de datos necesarias (bases de datos), módulos o partes en las que se dividirá la aplicación o, incluso, la solución tecnológica necesaria para ponerla en marcha (lenguaje a utilizar, tipo de gestor de base de datos, servidores web, etc.). Este documento no debe ser revisado por el usuario, puesto que la parte técnica corresponde a los equipos de sistemas.

**Construcción:** una vez dividida en módulos la aplicación, en esta fase se asigna cada uno de estos módulos (con las especificaciones de lo que debe hacer y cómo lo debe hacer) a un equipo de programadores para su construcción. En esta fase se necesitará el compilador o intérprete del lenguaje escogido para llevar a cabo la aplicación (ver Tipos de lenguajes) para obtener el ejecutable, así como la ayuda de un depurador (debug) para detectar el origen de los errores que puedan surgir en las pruebas de cada uno de los módulos.

**Pruebas integradas:** una vez desarrollados todos los módulos por separado se unen en una única aplicación y se comprueba el correcto funcionamiento de todos ellos según las especificaciones solicitadas. En caso de detectarse cualquier error, se identifica el módulo donde se encuentra y se notifica al equipo de programadores para su corrección.

**Validación:** antes de poner en marcha la aplicación, se convoca una reunión con el RFA para llevar a cabo la prueba del producto y validarlo. De este modo tenemos una garantía de que lo desarrollado por los equipos de sistemas es realmente lo requerido por el usuario.

Producción: una vez obtenida la validación del usuario, se debe poner en marcha la producción de la aplicación. Previamente, suelen realizarse sesiones de formación de los usuarios para garantizar su correcta utilización por parte de los destinatarios del desarrollo.

Seguimiento y garantía: durante un determinado intervalo de tiempo posterior a la puesta en marcha de la aplicación, debe realizarse un seguimiento de su funcionamiento para detectar posibles errores no detectados en las fases anteriores.

## REFERENCIAS

<https://sg.com.mx/content/view/490>

<https://ceroideas.es/principales-caracteristicas-de-los-tipos-de-aplicaciones-moviles/>

Dialnet-InferenciaDeContextoEnAplicacionesMovilesIntelligen-6043106.pdf

B, D. (2021, 10 diciembre). *Algoritmos de Consenso Raft y Paxos* - Diego B. Medium. <https://medium.com/@juan.baranowa/algor%C3%ADmos-de-consenso-raft-y-paxos-b252e51e911a>

Agarwal, S .; Starobinski, D .; Ari Trachtenberg (2002). "Sobre la escalabilidad de los protocolos de sincronización de datos para PDA y dispositivos móviles". *Red IEEE* . **16** (4): 22-28. CiteSeerX 10.1.1.12.5572 . doi : 10.1109 / MNET.2002.1020232 .