

Base de datos para aplicaciones móviles híbridas

**Breve descripción:**

El estudio del componente favorecerá afianzarse en el diseño e implementación de bases de datos dentro del desarrollo de aplicaciones móviles. Además, reconocer diferentes herramientas vigentes para realizar tal proceso y definir la opción ideal según la naturaleza del proyecto por realizar; supliendo necesidades de rendimiento, seguridad, adaptabilidad y compatibilidad necesarias, para el correcto funcionamiento de la app.

**Septiembre 2023**

Tabla de contenido

[Introducción 1](#_Toc144906796)

[1. Diseño de bases de datos 3](#_Toc144906797)

[2. Sistema gestor de base de datos (SGBD) 5](#_Toc144906798)

[2.1. “Data management system” DNMS 5](#_Toc144906799)

[2.2. Componente de un SGBD 6](#_Toc144906800)

[2.3. Propiedad de un SGBD 7](#_Toc144906801)

[2.4. Tipos de SGBD utilizados en “apps” móviles 8](#_Toc144906802)

[3. Gestores de bases de datos para “apps” móviles y relacionales 12](#_Toc144906803)

[4. Creación de base de datos PostgreSQL 15](#_Toc144906804)

[5. Sistemas gestores de bases de datos no relacionales (NoSQL) 17](#_Toc144906805)

[6. ¿Por qué no utilizar conexión directa a base de datos? 20](#_Toc144906806)

[7. Beneficios de un servicio web 21](#_Toc144906807)

[8. Configuración de Firebase a una “app” de móvil híbrida. 27](#_Toc144906808)

[9. Vinculación de Firebase a la aplicación móvil 31](#_Toc144906809)

[10. SQLite para aplicaciones livianas 32](#_Toc144906810)

[Síntesis 34](#_Toc144906811)

[Material complementario 35](#_Toc144906812)

[Glosario 36](#_Toc144906813)

[Referencias bibliográficas 37](#_Toc144906814)

[Créditos 38](#_Toc144906815)

Introducción

Los dispositivos móviles generan nuevos requerimientos y con ellos aparecen nuevas herramientas que se adaptan a las necesidades emergentes. La demanda de bases de datos para dispositivos móviles ha creado un ecosistema de aplicaciones y servicios, organizando una gran comunidad de diseñadores, desarrolladores y grandes empresas que trabajan y enfocan todos sus esfuerzos en suplir necesidades del mercado de aplicaciones. Este constante desarrollo tecnológico ha llevado a crear diversos métodos de almacenamiento de información en dispositivos móviles. El siguiente video presenta, de manera genérica, lo que se desarrollará alrededor de la estructuración de la base de datos para aplicaciones móviles híbridas.

1. Bases de datos para aplicaciones móviles híbridas



[**Enlace de reproducción del video**](https://www.youtube.com/watch?v=PgnkvZl0EGQ)

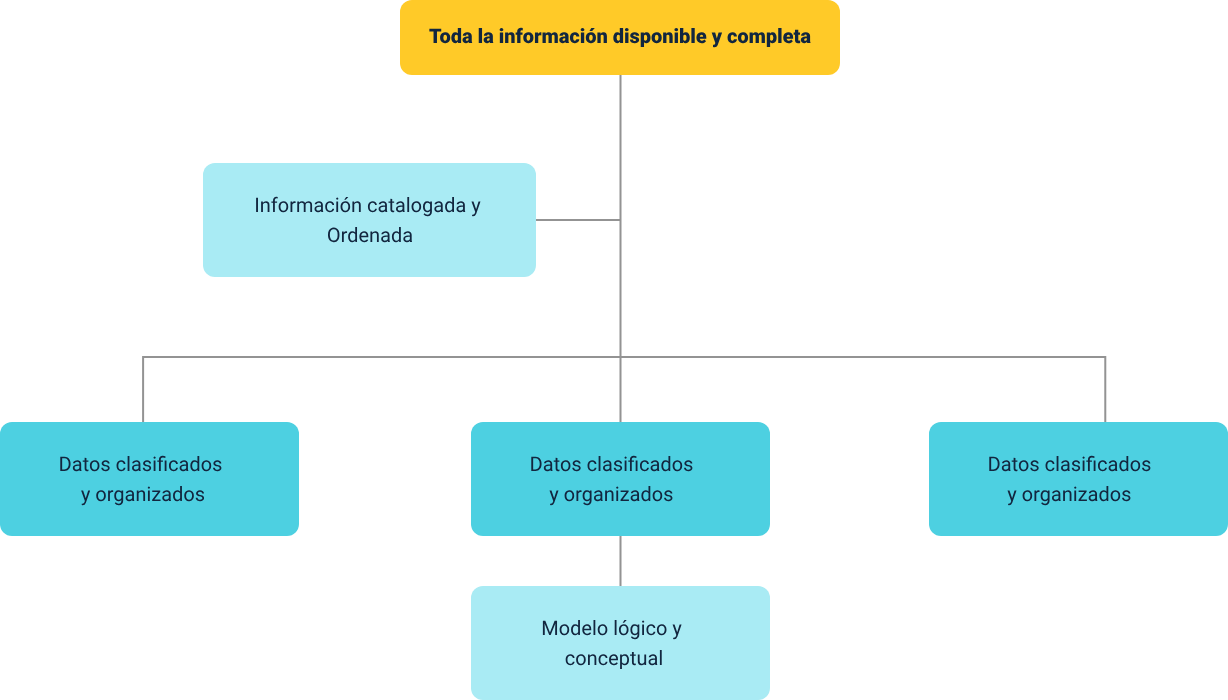
|  |
| --- |
| **Síntesis del video: Bases de datos para aplicaciones móviles híbridas** |
| El diseño e implementación de bases de datos para el desarrollo de aplicaciones móviles requiere del reconocimiento de las herramientas vigentes para ese proceso. Estas facilitan el modelar estructuras de datos según requisitos técnicos para luego vincularlas e integrarlas con la aplicación móvil.  Entre estas herramientas se encuentran los sistemas de gestión de bases de datos (SGBD), los cuales a su vez se diversifican para modelos relacionales (SQL) o no relacionales (NoSQL).  Finalmente, una vez estructurada la base de datos con el sistema gestor adecuado, se propone utilizar un “web server” como Firebase, que ofrece a los desarrolladores y administradores de bases de datos opciones para crear aplicaciones de plataforma web, Android e iOS, funcionales y versátiles; o un SQLite como herramienta de uso libre para aplicaciones livianas. |

Tal necesidad ha provocado, también, que el almacenamiento de los datos en estos dispositivos haya mejorado, tanto en capacidad como en herramientas. Por dicha razón, en la actualidad, se cuenta con diversas opciones de manejadores de bases de datos para aplicaciones móviles que permiten una administración desde las bases de datos más pequeñas y fáciles de implementar, por lo general de manera local, hasta las más robustas que requieren todo un equipo de trabajo (Silberschatz, Korth y Sudarshan, 2006).

# Diseño de bases de datos

Este diseño se realiza cuando se cuenta con toda la información disponible del proyecto; esta información debe estar catalogada y ordenada de manera clara para iniciar la clasificación de datos y su organización, con lo que se genera el modelo lógico y conceptual, basado y fundamentado en el modelo físico que se obtuvo como resultado de la investigación y levantamiento de la información, tal como se aprecia a continuación.

1. Base de datos relacional



Algunos parámetros en la generación del diseño de la base de datos son:

* Determinar el propósito de la base de datos.
* Buscar y organizar la información.
* Dividir la información en tablas.
* Convertir los elementos de información en columnas.
* Especificar las claves principales.
* Establecer las relaciones de tablas.
* Aplicar las reglas de normalización.

# Sistema gestor de base de datos (SGBD)

Toda la información que circula en la red, la recibida y la que se envía, sale o se aloja en una base de datos. Internet no sería lo que se conoce sin las bases de datos que proporcionan la posibilidad de guardar la información.

Para lograr la creación, gestión y administración, de tan grande cantidad de información, se utilizan los denominados sistemas gestores de base de datos (SGBD), que evolucionan a medida que lo requiere la tecnología. Los SGBD definen la estructura para el almacenamiento y búsqueda de información, de la manera más eficiente posible.

## “Data management system” DBMS

Este se puede definir como un *software* que estructura y organiza la información, bajo un modelo de almacenamiento que cuenta con ciertas características particulares, al momento de almacenar configurar, administrar y utilizar la información.

Los siguientes cuatro elementos, engloban los requisitos más importantes de un sistema de administración de bases de datos (SGBD):

* **Lectura y escritura**. El sistema de gestión de la base de datos define permisos de lectura y escritura.
* **Interfaces y lenguaje**. Estos procesos se definen por medio de las interfaces específicas de la aplicación y están determinados por el lenguaje de definición de datos correspondiente.
* **Lenguaje SQL**. El más conocido, usado y trabajado de estos lenguajes es SQL (“Structured Query Language”), que se define como un lenguaje de consulta estructurado.
* **Funciones y propiedades**. Las funciones y propiedades de las transacciones de un DBMS (sistema administrador de bases de datos) son: Atomicidad (“atomicity”), consistencia (“consistency”), aislamiento (“isolation”) y permanencia (“durability”).

## Componente de un SGBD

Un sistema de gestión de base de datos consta de varios componentes, todos los cuales contribuyen al buen funcionamiento del “software”. Los elementos básicos que lo conforman son tres: 1) el diccionario de datos, 2) el lenguaje de definición de datos y 3) el lenguaje de manipulación de datos.

En la tabla 1 se muestran los componentes que necesita manejar una base de datos para una aplicación móvil; se sigue la misma estructura que se utiliza en las páginas web.

1. Componentes de un SGBD

| Componentes | Manejo |
| --- | --- |
| Diccionario de datos | El diccionario de datos suministra las características y toda la información importante sobre los datos almacenados. Sin importar su diversidad, también notifica los permisos de uso de cada registro. |
| Lenguaje de definición de datos | Utiliza un DDL (“Data Definition Language”) que estructura el contenido de la base de datos; por medio de este lenguaje se puede crear, modificar y eliminar. |
| Lenguaje de manipulación de datos | Utiliza un DML (“Data Manipulation Language”), que permite registrar, eliminar, modificar, comprimir y extraer datos. |

## Propiedad de un SGBD

Por lo general, todas las aplicaciones móviles cuentan con una base de datos que puede ser pequeña y simple o muy grande y robusta; pero, por más pequeña que esta sea, debe contar con ciertas tareas, funciones y propiedades que harán de la base de datos un elemento aportante al buen desempeño de la aplicación (Silberschatz, Korth y Sudarshan, 2006).

En la tabla 2 se indican tales propiedades y la manera como estas logran cualificar el sistema de administración de bases de datos.

1. Propiedades de un SGBD

| Propiedades | Tareas y funciones |
| --- | --- |
| Almacenamiento de datos | Se debe permitir el almacenamiento de datos en varios formatos como texto, documentos, contraseñas, imágenes, estos también se deben poder consultar. |
| Edición de datos | Los elementos almacenados se deben poder modificar y editar directamente. |
| Eliminación de datos | Se pueden eliminar los datos almacenados y editados solo por los perfiles de usuario que permiten esta actividad. |
| Administración de metadatos | Crear, leer, actualizar y borrar datos. Realizar el CRUD organizando, facilitando la función de búsqueda. |
| Seguridad de los datos | Evitar el acceso a los datos de individuos no autorizadas y mantener la seguridad de los datos, implementando un método de cifrado eficaz para evitar que los datos se pierdan o sean manipulados. |
| Integridad de los datos | En las bases de datos relacionales se utilizan cuatro conceptos para mantener la integridad y coherencia de los datos: integridad de dominio, integridad de entidad, integridad referencial y coherencia lógica. |
| Modo multiusuario | Permitir el ingreso de varios o muchos usuarios a la vez y lograr mantener la seguridad de los datos, la coherencia y no afectar el rendimiento de la aplicación. |
| Optimización de consulta | La lógica y la estructura de la base de datos deben garantizar el buen funcionamiento y la optimización de recursos. |
| “Triggers” y “Stored procedures” | Son procedimientos que se encargan de activar automáticamente (“trigger”) como respuesta a algunas acciones. |

## Tipos de SGBD utilizados en “apps” móviles

El principal objetivo de la instalación de un sistema gestor de base de datos es desarrollar y administrar la información de la base de datos, de la mejor manera posible. Existen diferentes modelos que se diferencian, fundamentalmente, en la manera en la cual está almacenada y es consultada la información. Por lo tanto, decidirse por un DBMS ha de estar más ligado al tipo de aplicación que se quiere desarrollar y la manera en que se deban almacenar y consultar los datos dentro de la aplicación; para ello, conocer los modelos de datos y sus características, permitirá saber, de forma sencilla, cuál es la opción más adecuada para implementar.

A continuación, se muestran los tipos de sistemas de administración de bases de datos más utilizados y sus principales características y funcionalidades.

1. Tipos de SGBD

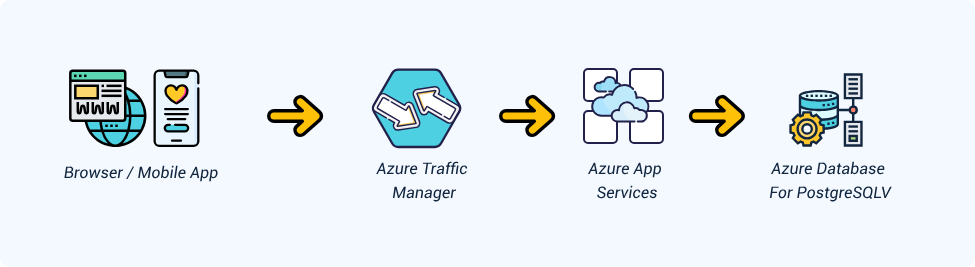
| Tipos de SGBD | Características y funcionalidades | Ejemplos |
| --- | --- | --- |
| Relacional | El más común y popular.  Los datos se estructuran en filas de tabla.  La posibilidad de crear diferentes relaciones entre las filas y presentarlas en columnas.  Una tabla puede tener muchos registros y cada registro puede tener muchos campos.  Utiliza el lenguaje SQL.  SQL se utiliza en, prácticamente, todas las bases de datos relacionales.  Esto permite una mayor seguridad e integración con otras bases de datos. | Microsoft Access.  Microsoft SQL Server.  MySQL.  Db2 de IBM.  Oracle Database.  IBM Informix.  MariaDB.  Sybase ASE.  Firebird. |
| Jerárquica | Es similar a la estructura de un árbol y muy parecida a la estructura de Microsoft Windows que organiza las carpetas y archivos.  El enlace es anidado conservando los datos organizados en un orden particular en un mismo nivel de lista.  Esto reduce el requerimiento de memoria y también aumenta la eficiencia. | IMS de IBM. |
| De red | En un modelo de red, la característica determinante es que almacena un registro con un enlace a otros registros formando una red.  Los componentes de una base de datos de red pueden tener varias relaciones padre-hijo.  Dado que siempre hay un vínculo entre un padre y un hijo, se mantiene la integridad de los datos. | PostgreSQL.  InterSystems Caché.  InterSystems IRIS. |
| Orientada a objetos | Utiliza el concepto de la herencia: esto representa que los objetos pueden transferir algunos de sus atributos a otros objetos, lo que se regula a través del SGBD. | PostgreSQL.  InterSystems Caché (combina relacional y orientado a objetos).  InterSystems IRIS (combina relacional y orientado a objetos). |
| Orientada a documentos | Almacena los registros de datos en diferentes documentos. | OrientDB.  CouchDB.  MongoDB. |
| El modelo de base de datos plana | En un modelo de base de datos plano, hay dos dimensiones.  Todos los registros se almacenan en un solo lugar.  Funciona bien para bases de datos pequeñas y tiene requisitos mínimos de “software” y “hardware”. | Excel. |

# Gestores de bases de datos para “apps” móviles y relacionales

Los gestores de bases de datos se dividen según la manera en que se administran los datos. En la actualidad existen dos modelos: las relacionales (SQL) y las no relacionales (NoSQL). En este punto, se enfocará la mirada hacia los gestores de bases de datos relacionales (SQL). Este modelo se caracteriza por establecer vínculos y relaciones entre los datos, utilizando una tabla integradora por cada relación existente con sus propios registros y atributos.

En la figura 2 se muestra un ejemplo de arquitectura del modelo relacional.

1. Ejemplo de arquitectura utilizando Azure y PostgreSQL



Nota. Tomada de Microsoft (2021).

A continuación, se indican algunos de los sistemas de administración de bases de datos más utilizados del modelo relacional.

* **MySQL.** Es un SGBD multihilo y multiusuario, de uso libre, siendo el más utilizado en el desarrollo de páginas web y aplicaciones móviles. Se entrega bajo la licencia GNU GPL.
* **MariaDB**. Es una consecuencia de MySQL. Está compuesto por varias extensiones su entorno gráfico de desarrollo, es amigable. Se deriva a partir de la compra de MySQL por Oracle para seguir la filosofía del “open source”, es totalmente compatible con MySQL. Algunas de sus principales características son el aumento de motores de almacenamiento, la gran escalabilidad y la buena seguridad y rapidez en transacciones. Cuenta con las nuevas extensiones que permiten el trabajo y desarrollo de aplicación para bases de datos NoSQL.
* **SQLite**. Es una biblioteca escrita en C que implementa un SGBD y que permite transacciones sin necesidad de un servidor ni configuraciones; es muy utilizada en la creación y gestión de pequeñas bases de datos locales; además, es útil para las aplicaciones móviles que no requieren conexión a servidores remotos y maneja un pequeño flujo de datos local.
* **Principales características.** Una de sus principales características es su tamaño reducido; al tratarse de una biblioteca no requiere de muchos recursos como sí lo hace un SGBD, a pesar de su tamaño reducido es de gran estabilidad pues está diseñado sobre los principios de ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), su desventaja es que no soporta bases de datos muy grandes.
* **Microsoft SQL Server**. Es un gestor de base de datos relacionales que utiliza el lenguaje Transact-SQL. Una de sus grandes características es que brinda la posibilidad de manejar gran cantidad de información y usuarios, generando un flujo fuerte y constante, que soporta sin inconvenientes. Está soportado, solamente, por Microsoft. Presenta una excelente seguridad, estabilidad y escalabilidad. Cuenta con la opción de realizar la cancelación de consultas y también con un excelente entorno gráfico que facilita su administración. Utiliza y se centra sobre los comandos DDL y DML. Como todos los productos de Microsoft, cuenta con una versión “Express”, pero presenta grandes limitantes. Si se requiere una versión completa, se debe pagar un alto precio por su uso completo “Standard”, “Developer”, “Enterprise” o SQL Azure, la versión de SQL Server en la nube.
* **Oracle**. Es el SGBD más utilizado en el ámbito empresarial y su recorrido y evolución le posicionan como uno de los mejores en el mercado; se destaca por su gran soporte de transacciones, su estabilidad y escalabilidad, su soporte multiplataforma. Su gran inconveniente o desventaja es que cuenta con una versión gratuita: “Edition Express”, que proporciona ciertas características, pero muchas limitaciones.
* **PostgreSQL**. Es, en la actualidad, uno de los gestores de base de datos relacional más usados en el mercado, tiene licencia BSD y es de uso libre; además, presenta grandes ventajas para los desarrolladores de aplicaciones móviles y administradores de bases de datos. Presenta un control de concurrencias multiversión (MVCC) y flexibilidad en cuanto a lenguajes de programación y ofrece una excelente estabilidad, robustez y eficiencia en todos los procesos principales con bases de datos robustas. Su principal falencia es, aunque no parezca lógico, la lentitud para realizar la administración de bases de datos pequeñas: este gestor no se debe tener en cuenta, o tomar como una opción, cuando la aplicación desarrollada utiliza una base de datos con poca información y bajo flujo de datos; se subutilizará la herramienta y no se tendrán buenos resultados. Dispone de una herramienta de administración muy fácil e intuitiva para la gestión de las bases de datos.

# Creación de base de datos PostgreSQL

Este gestor de base de datos relacional, fundamentado sobre el modelo orientado a objetos, es de uso libre y cuenta con un excelente soporte para crear y administrar bases de datos de diferente tamaño y servicios. En este punto del componente, se visualizarán los pasos de la creación de una base de datos y su alojamiento sobre un servidor que provee la plataforma, monitoreando todas las actividades y el flujo de información que se genera, en tiempo real, sobre la aplicación móvil.

A continuación, se indican los pasos y orientaciones de creación de base de datos, en PostgreSQL.

1. **Página de inicio para la creación de base de datos PostgreSQL.** Para crear y administrar una base de datos PostgreSQL con un entorno gráfico amigable, se puede utilizar la herramienta pgAdmin. Para trabajar con este entorno, se puede ingresar en el sitio oficial de la herramienta <https://www.pgadmin.org/> Allí hay la posibilidad de realizar una demostración del funcionamiento SGBD en línea; también, es posible descargar la aplicación o crear un usuario y trabajar sobre la plataforma web.
2. **Bienvenida al entorno web de la aplicación.** Esta herramienta cuenta con un “sandbox” en el cual se puede realizar una práctica básica guiada, familiarizándose con el entorno de trabajo pgAdmin.
3. **Ingreso en la plataforma.** El siguiente procedimiento permite crear y trabajar una base de datos utilizando el entorno en línea, realizando un registro en la plataforma por medio de un usuario y una contraseña.
4. **Entorno de trabajo de PGAdmin.** Este es el entorno de administración proporcionado PGAdmin. Se puede realizar el trabajo de administrar un nuevo servidor, realizar la configuración del entorno de trabajo; también es posible ingresar, directamente, a la página web de PGAdmin. Así mismo, se puede encontrar e ingresar a la comunidad y acceder a toda la documentación de soporte de PostgreSQL.
5. **Creación de un servidor.** PGAdmin permite realizar la creación de un servidor de base de datos sobre el cual se creará la base de datos que, posteriormente, será vinculada a la aplicación o al programa que se realice; también, posibilita realizar una configuración básica de administración y soporte.
6. **Contraseña de ingreso.** Para realizar la configuración y creación de la base de datos es importante la seguridad; por tal razón, siempre pedirá la contraseña para realizar cualquier cambio importante sobre el servidor.
7. **Base de datos creada y monitoreada.** Después de la vinculación de la base de datos al servidor y a la aplicación, es posible realizar la administración y el seguimiento de todos los procesos, actividades y rendimiento de la base de datos.

Para profundizar en la creación de bases de datos PostgreSQL, se sugiere revisar los videos ubicados en el material complementario.

# Sistemas gestores de bases de datos no relacionales (NoSQL)

Una base de datos no relacional (NoSQL) no requiere de estructuras de datos fijas, como tablas, ni garantiza completamente las características ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad); las bases de datos no relacionales son utilizadas en ambientes distribuidos que deben estar siempre disponibles y activos, gestionando un enorme flujo de información.

Para administrar este tipo de arquitectura de bases de datos, existen también unos gestores que facilitan el trabajo a los desarrolladores y administradores; los principales sistemas (SGBD NoSQL) que se puede encontrar son, a saber:

* **MongoDB**. Mongo es el gestor de datos más utilizado para la creación y administración de base de datos no relacionales. Está enfocado en el manejo de ficheros que permiten almacenar la información en estructuras BSON, con un esquema dinámico que facilita el proceso de integración.

**Características.** Algunas de sus características principales es que permite la indexación y réplica de los datos, realiza un balanceo de carga, todo el almacenamiento se realiza sobre ficheros que permiten realizar las consultas “ad hoc”, es de uso libre y presenta una completa plataforma de administración. Es importante resaltar que grandes empresas, como Google, Facebook, eBay, Cisco o Adobe, utilizan este gestor como su principal herramienta para administrar sus bases de datos; ello indica que su soporte es bien robusto y permite el manejo de grandes volúmenes de datos y grandes flujos de información.

* **Redis**. Se basa en el método de almacenamiento clave-valor. Se podría comparar con el manejo de un vector o vectores que almacenan todos los datos, sin importar su tipo o formato. Se pueden almacenar cadenas, listas, enteros, “float”, etc. El principal uso de este SGBD es el almacenamiento en memoria caché y la administración de sesiones. Sus características más relevantes son la atomicidad y persistencia, su gran velocidad en tiempo de respuesta y la simplicidad en sus procesos.
* **Cassandra**. Este SGBD utiliza el método clave-valor para el almacenamiento. Trabaja sobre el modelo NoSQL distribuido y masivamente escalable. Una de sus principales características es que es multiplataforma, cuenta con su propio lenguaje de consultas, (CQL) “Cassandra Query Languaje”. Utiliza una arquitectura “peer-to-peer”, algunas de las plataformas o compañías que lo utilizan son Facebook, Twitter, Instagram, Spotify o Netflix, entre otras
* **Existen algunos otros SGBD**. Para elegir el SGBD adecuado, se debe empezar principalmente definiendo qué tipo de datos se almacenarán y cómo se van a administrar. Entre los sistemas gestores de base de datos mencionados es posible encontrar el que mejor se adapte a las necesidades y a la capacidad de inversión a realizar, volumen de información a almacenar, tipo de consultas a realizar, flujo y tamaño de información a procesar.
  + Azure Cosmos DB.
  + RavenDB.
  + ObjectDB.
  + Apache CouchDB.
  + Neo4j.
  + Google BigTable.
  + Apache Hbase.
  + Amazon DynamoDB.

# ¿Por qué no utilizar conexión directa a base de datos?

No se recomienda conectar la base de datos, por medio de una conexión persistente, directamente a la aplicación, ya que esto podría traer complicaciones cuando hay mala conectividad o fallas de conexión móvil. En cambio, sí es recomendable utilizar un método de solicitudes HTTP, sin estado (“stateless HTTP requests”). Este diseño sin estado, simplifica el esquema del servidor porque no hay necesidad de asignar, dinámicamente, almacenamiento para tratar las conversaciones en curso. Si un cliente desaparece, en medio de la transacción, ninguna parte del sistema tiene que ser responsable de limpiar el estado actual del servidor.

A continuación, se indican otras razones por las que no es recomendable la conexión directa a base de datos.

**Saturación y retrasos**. Otro de los inconvenientes que se pueden llegar a presentar es la gran cantidad de conexiones que se realizan desde las aplicaciones móviles a la base de datos; esto puede generar problemas al momento de realizar el monitoreo de los procesos y limitar la capacidad de respuesta, saturando la base de datos y generando retardos en las respuestas.

**Riesgos de seguridad**. Los resultados enviados desde la DB a la aplicación consumen gran ancho de banda y energía de la batería, y otro de los grandes problemas al proceder directamente a la base de datos es la exposición de la base de datos de manera directa y se corren riesgos de seguridad por no contar con protocolos que medien la conexión y sirvan como filtros seguros de autenticación.

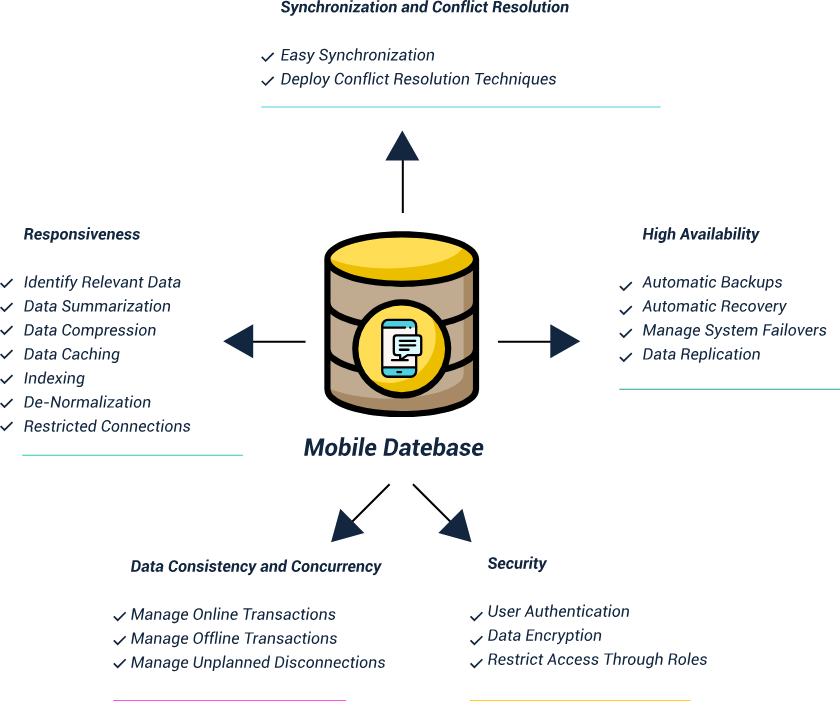
# Beneficios de un servicio web

Los principales beneficios que se obtienen del uso de los servicios web, para una aplicación móvil, es que se puede lograr una conexión de corta duración, con un estado mínimo, permitiendo volver al estado inicial si la transacción o el proceso no finalizó correctamente. El dispositivo cambia de redes wifi a red móvil y viceversa, perdiendo la conexión por unos segundos, pero retoma el requerimiento.

De acuerdo con Rodríguez, Martín, Valenzuela y Chávez (2009) distintas aplicaciones de “software”, desarrolladas en lenguajes de programación diferentes y ejecutadas sobre cualquier plataforma, pueden utilizar el “Web Service” para intercambiar información en redes de computadoras, como internet, evitando la pérdida total de la información y la conexión (p. 2).

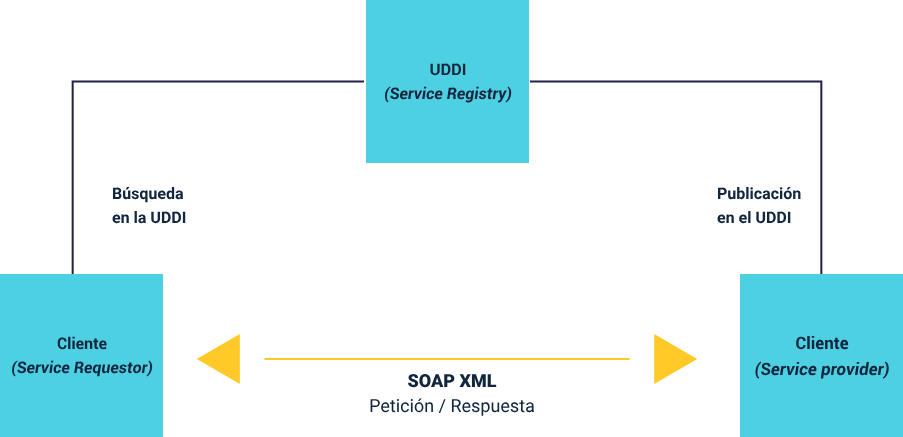
En las figuras que se muestran, se invita a reconocer los elementos clave para el diseño de una base de datos móvil y los beneficios del uso de servidores web para las aplicaciones móviles.

1. Elementos clave para el diseño de una base de datos móvil



|  |
| --- |
| **Descripción de la figura: Elementos clave para el diseño de una base de datos móvil** |
| Los elementos necesarios de una base de datos móvil son:  **“Synchronization and Conflict Resolution”**   * “Easy Synchronization” * “Deploy Conflict Resolution Techniques”   **“Responsiveness”**   * “Identify Relevant Data” * “Data Summarization” * “Data Compression” * “Data Caching” * “Indexing” * “De-Normalization” * “Restricted Connections”   **“High Availability”**   * “Automatic Backups” * “Automatic Recovery” * “Manage System Failovers” * “Data Replication”   **“Data Consistency and Concurrency”**   * “Manage Online Transactions” * “Manage Offline Transactions” * “Manage Unplanned Disconnections”   **“Security”**   * “User Authentication” * “Data Encryption” * “Restrict Access Through Roles” |

1. “Webs Services”



|  |
| --- |
| **Descripción de la figura: “Webs Services”** |
| El “Web Services” inicia con el servicio de registro. Luego, se divide dos caminos: En la búsqueda en la UDDI o en la publicación en el UDDI.  El primero llega al cliente (“service requestor”) y el segundo, al “service provider”; ambos a su vez generan petición y/o respuesta. |

Se invita a dar clic sobre los términos que aquí se proponen y descubrir en ellos algunos otros beneficios y aspectos funcionales del uso de servicios web.

* **Alojamiento en servidor**. Una buena práctica es estructurar y diseñar la lógica de base de datos de la aplicación y alojarla en un servidor de aplicaciones accesible desde la web y comunicarse con ese servidor de aplicaciones, por medio de protocolos HTTP + JSON, SOAP, XML-RPC o similares, dependiendo del lenguaje y “frameware” que se está utilizando para el desarrollo. Esto hará mucho más eficiente el uso del ancho de banda y la aplicación será más tolerante a los problemas con la conectividad. También evitará exponer la base de datos directamente a internet.
* **SOAP – “Simple Object Access Protocol”.** SOAP es un protocolo escrito en XML para el intercambio de información entre aplicaciones. Es un formato para enviar mensajes, diseñado especialmente para servir de comunicación en Internet, pudiendo extender los HTTP “headers”. Es una forma de definir qué información se envía y cómo, mediante XML. Básicamente es un protocolo para acceder a un “Web Service” (Lázaro, 2018).
* **WSDL – “Web Services Description Language”**. WSDL es un lenguaje basado en XML para describir los servicios web y cómo acceder a ellos. Es el formato estándar para describir un “Web Service”, y fue diseñado por Microsoft e IBM. WSDL es una parte integral del estándar UDDI y es el lenguaje que este utiliza (Lázaro, 2018).
* **UDDI –** **“Universal Description, Discovery and Integration”**. UDDI es un estándar XML para describir, publicar y encontrar servicios web. Es un directorio donde las compañías pueden registrar y buscar servicios web. Es un directorio de interfaces de servicios web descritos en WSDL que se comunican mediante SOAP (Lázaro, 2018).

# Configuración de Firebase a una “app” de móvil híbrida.

Una buena opción para trabajar bases de datos en aplicaciones móviles híbridas es utilizar un “web server” como Firebase, que ofrece a los desarrolladores y administradores de bases de datos opciones para crear aplicaciones de plataforma web, Android e iOS, funcionales y versátiles. Firebase se puede comprender como una plataforma que integra y presta varios servicios esenciales para el desarrollo de aplicaciones móviles; es un producto soportado por Google con grandes complementos y funciones para desarrollar, manejar, administrar y potenciar las aplicaciones móviles y “software” en general.

A continuación, se muestra cómo Firebase es una recopilación de herramientas en las que los diseñadores, desarrolladores y analistas pueden trabajar y apoyarse buscando mejor rendimiento, creando aplicaciones.

**Firebase y su historia.**

Firebase inició en el año 2011 como una idea de Andrew Lee y James Tamplin; ofrecía una API para la integración de chat en línea para sitios web, pero este chat estaba siendo utilizado para transmitir mensajes que no eran de chat. De esa experiencia nació una plataforma para la sincronización de datos en tiempo real. El primer producto Firebase lanzado fue la base de datos Firebase en tiempo real. Se trató de una API que permitía la sincronización de datos de aplicaciones implementadas en dispositivos Android, web e iOS. En el 2014 Google compró la compañía y la fusionó con Divso, otra empresa que contaba con una plataforma de alojamiento web.

**Funcionalidades de Firebase.** Lo primero a considerar es que Firebase no es gratuito y tiene algunos servicios que son básicos y permiten realizar actividades y operaciones básicas. Se denomina plan “Spark*”* de Firebase por sus características y sirve para ayudar a los desarrolladores a comenzar; este plan ofrece buenas posibilidades de lectura y escritura de la base de datos con restricciones, pero con suficientes posibilidades de trabajar una aplicación de un tamaño medio.

**Velocidad en el desarrollo**. Una de las grandes ventajas de utilizar Firebase consiste en reducir el tiempo de desarrollo de aplicaciones. Soluciona el acceso a servidores evitando el mantenimiento de bases de datos y servicios “backend”. Permite a desarrolladores de “frontend” realizar la administración de todo el trabajo, gracias a los servicios de almacenamiento de datos, la autenticación, las notificaciones, el análisis. El proceso de integración a las aplicaciones es sencillo, cuenta con SDK que genera el enlace directo entre la aplicación y Firebase; es muy fácil de instalar y configurar.

**Almacenamiento en la nube.** Cuenta con dos bases de datos que pueden ser la solución a muchos problemas de conexión y almacenamiento: Firestore y Firebase's Realtime Database. Permite el almacenamiento en la nube sin esfuerzo y admite el desarrollo de aplicaciones sin servidor a través de funciones en la nube integradas. Firebase ofrece soluciones para todo el ciclo de desarrollo de aplicaciones, creando, liberando y monitoreando aplicaciones. El soporte y la integración de la plataforma es excelente, pues Google genera la integración fácilmente a todos sus demás servicios, por ejemplo, Google Maps se puede integrar de manera sencilla en una aplicación móvil.

**Rendimiento óptimo**. No ofrece un servidor, trabaja sobre un clúster de base de datos optimizando el rendimiento, utilizando una arquitectura sin servidor que requiere que los usuarios paguen en función de las solicitudes llamadas respuestas, es decir, por el flujo de información que maneja la aplicación. Los desarrolladores obtienen la libertad de adoptar un enfoque práctico para controlar DevOps, la configuración, la infraestructura y la planeación de la capacidad. Firebase viene con un kit de ML con API fácilmente disponibles para diferentes plataformas móviles, como identificación de texto, detección de rostros, etiquetado de imágenes, escaneo de códigos de barras, servicios que se prestan de manera individual y tienen un costo diferencial. ML puede ser utilizado por los desarrolladores de iOS y Android, independientemente de su nivel de experiencia.

**Funciones en la nube**. Firebase Cloud Functions hace referencia a un marco sin servidor que permite a los desarrolladores ejecutar código “backend” para responder a los eventos desencadenados por características HTTPS y Firebase. Google Cloud se utiliza para almacenar código TypeScript o JavaScript y se puede ejecutar dentro de un entorno administrado. Firebase soluciona uno de los problemas más complejos al momento de desarrollo y ofrece buena seguridad sobre los datos y disponibilidad óptima de estos con la ayuda de copias de seguridad periódicas. Las aplicaciones se encuentran seguras frente a las posibilidades de fallos y pérdida de datos utilizando el sistema de copia de seguridad automática de la plataforma.

**Autenticación y mensajería**. La autenticación es una característica de Firebase que ofrece bibliotecas de interfaz de usuario listas para usar, backend y SDK convenientes para la autenticación de aplicaciones de usuario. La autenticación se admite a través de números de teléfono, contraseñas y proveedores como Google, Twitter, Facebook y otros. Firebase Cloud Messaging o FCM se refiere a un servicio de mensajes multiplataforma para mensajería gratuita. Esta característica permite a los propietarios de aplicaciones, notificar a las aplicaciones cliente, si se van a sincronizar los datos o los correos electrónicos.

# Vinculación de Firebase a la aplicación móvil

El primer paso para realizar la vinculación de Firebase a la aplicación móvil, es crear un proyecto en Firebase definiendo las características necesarias para su funcionamiento. Esta vinculación es importante porque es una de las plataformas más usadas en el desarrollo de “apps” móviles por sus ventajas y recursos ofrecidos.

Firebase presenta la posibilidad de utilizar un “hosting” gratuito y una base de datos que puede ser administrada; también permite hacer monitoreo de las actividades y procesos de la base de datos.

**Nota**. Para asimilar y aplicar la vinculación de Firebase al proyecto de aplicación móvil, visite y estudie el documento **Vinculacion-de-Firebase**, ubicado en la carpeta anexos. Desarrolle los pasos que allí se muestran y refuerce los aspectos más importantes de este desarrollo.

# SQLite para aplicaciones livianas

Esta herramienta de uso libre permite almacenar y administrar información en dispositivos móviles como los “smartphones” y también en pequeños dispositivos utilizados en IoT, de manera sencilla, eficaz y rápida; utiliza y soporta el lenguaje SQL sin ninguna limitación, implementando el estándar SQL92.

**Historia**

Aparece en el año 2000 gracias a su desarrollador Richard Hip, bajo licencia GPL, su código es de dominio público y puede ser modificado. Su evolución es muy grande gracias a la cantidad de colaboradores que generan librerías que permiten ser utilizado en casi cualquier sistema y es soportado por muchos lenguajes de programación permitiendo una correcta integración. Es posible encontrar una gran cantidad de componentes, librerías y “drivers” para interactuar con SQLite desde una gran diversidad de lenguajes y plataformas de programación. Se adapta a lenguajes modernos como Java, Perl, Python, PHP, Ruby, C#, y a lenguajes más viejos como Pascal, SmallTalk, Clipper, para todos es posible librerías y ejemplos de código para SQLite (Rómmel, s. f.).

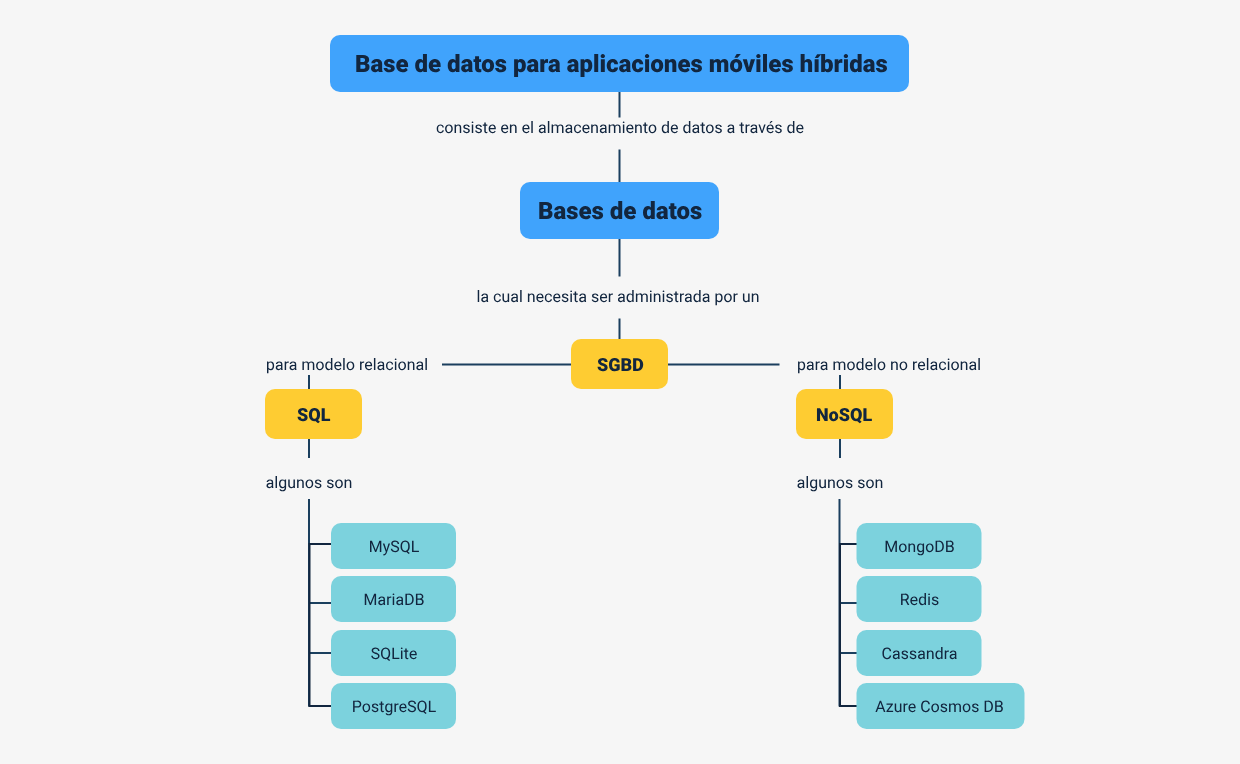
**Principales características**

SQLite promueve a trabajar con proyectos livianos. Su base de datos se encuentra en un solo archivo y su funcionamiento se puede limitar al solo uso de la memoria RAM, no tiene dependencias externas todos sus procesos y actividades están autocontenidas, utiliza librerías de acceso que permite la integración con múltiples lenguajes de programación. Soporta texto en formato UTF-8 y UTF-16, así como datos numéricos de 64 bits. Trabaja con funciones SQL definidas por el usuario (UDF). Su código fuente es de dominio público y se encuentra muy bien documentado (Rómmel, s. f.).

**Migraciones a diversas plataformas**. Fue diseñado con lenguaje de programación C, por esa sencilla razón es soportado para realizar migraciones sencillas a diversas plataformas de sistemas operativos y a dispositivos electrónicos que requieran bases de datos pequeñas, SQLite es una base de datos muy eficaz para cualquier desarrollo en ambientes embebidos, pues ofrece un alto rendimiento, eficacia, seguridad, estandarización e inter tolerabilidad. Todo esto la ha catapultado a convertirse en la base de datos de facto para desarrollos móviles (Rómmel, s. f.).

Síntesis

A continuación, se presenta a modo de diagrama las temáticas desarrolladas en este componente formativo.



Material complementario

|  |  |  |  |
| --- | --- | --- | --- |
| Tema | Referencia | Tipo de material | Enlace del recurso |
| 1. Diseño de bases de datos | Casas, J. (s.f.). Diseño conceptual de bases de datos. UOC. | Página web | <http://cv.uoc.edu/annotation/cb826b689abc472d8fb5b2519840058b/699689/PID_00213704/PID_00213704.html> |
| 3. Gestores de bases de datos para apps móviles y relacionales | Hdeleon.net (2019). ¿Cómo hacer una base de datos para una app tipo Uber? | Video | <https://www.youtube.com/watch?v=1yAC3m7f5Pk> |
| 4. Creación de base de datos PostgreSQL | Database Star. (2022). pgAdmin Tutorial - How to Use pgAdmin. | Video | <https://www.youtube.com/watch?v=WFT5MaZN6g4> |
| 4. Creación de base de datos PostgreSQL | Reyes, P. (2021). Crear base de datos PostgreSQL - PGAdmin 4. | Video | <https://www.youtube.com/watch?v=B_lqjZGwViI> |
| 8. Configuración de Firebase a una app móvil híbrida | Cano, J. (2017). Nativescript, login con firebase | Video | <https://www.youtube.com/watch?v=WKfgnzl0p3E> |
| 10. SQLite para aplicaciones livianas | Henao, C. (2017). 50. Cómo crear una Base de Datos SQLite en Android | Video | <https://www.youtube.com/watch?v=9WiyqIcffe0> |

Glosario

**“Peer-to-peer”:** conexión con una arquitectura destinada a la comunicación entre aplicaciones. Esto permite a las personas o a los ordenadores compartir información y archivos, de uno a otro, sin necesidad de intermediarios.

**“Sandbox”:** plataforma aislada para ejecutar programas con seguridad y de manera separada. Se utiliza para ejecutar código nuevo o de dudosa procedencia.

**SDK**: recopilación de un grupo de herramientas de desarrollo de “software” que complementa o facilita la comunicación e interacción con otra plataforma o con otro “software”.

**Sistema Gestor de Base de Datos (SGBD):** servicio principal para almacenar, procesar y proteger los datos. El SGBD proporciona acceso controlado y procesamiento de transacciones rápido, para cumplir con los requisitos de las aplicaciones consumidoras de datos más exigentes de una empresa.

**“Web Services”:** tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Referencias bibliográficas

Lázaro, D. (2018). Introducción a los web services. <https://diego.com.es/introduccion-a-los-web-services>

Microsoft.com. (2021). Aplicaciones web y móviles escalables mediante Azure Database for PostgreSQ. <https://docs.microsoft.com/es-es/azure/architecture/solution-ideas/articles/scalable-web-and-mobile-applications-using-azure-database-for-postgresql>

Rodríguez, N., Martín, A., Valenzuela, A. y Chávez, S. (2009). Consumo de web service desde dispositivos móviles heterogéneos. <http://sedici.unlp.edu.ar/bitstream/handle/10915/19809/Documento_completo.pdf%3Fsequence%3D1>

Rómmel, F. (s.f.). SQLite: la base de datos embebida. <https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida>

Silberschatz, A., Korth., H., y Sudarshan, S. (2006). Fundamentos de bases de datos. McGraw-Hill.

Yu, W. & Sharma, S. (2007). A Mobile Database Design Methodology for Mobile Software Solutions [Figure 6. Mobile database issues and solutions]. Semantic Scholar. <https://www.semanticscholar.org/paper/A-Mobile-Database-Design-Methodology-for-Mobile-Yu-Sharma/41e3949c8aca76725184c740d782e79ca28efddc>

Créditos

| Nombre | Cargo | Regional y Centro de Formación |
| --- | --- | --- |
| Claudia Patricia Aristizábal | Responsable del Ecosistema | Dirección General |
| Rafael Neftalí Lizcano Reyes | Responsable de Línea de Producción | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Wilson Andrés Cuervo Nieto | Experto Temático | Centro de Diseño y Metrología - Regional Distrito Capital |
| Fabián Leonardo Correa Díaz | Diseñador Instruccional | Centro agropecuario La Granja - Regional Tolima |
| Alix Cecilia Chinchilla Rueda | Evaluador instruccional | Centro de Gestión Industrial - Regional Distrito Capital |
| José Gabriel Ortiz Abella | Corrector de estilo | Centro para la Industria de la Comunicación Gráfica - Regional Distrito Capital |
| Miroslava González Hernández | Diseñadora Instruccional | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Carmen Alicia Martínez Torres | Animador y Productor Multimedia | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Wilson Andrés Arenales Cáceres | Storyboard e ilustración | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Camilo Andrés Bolaño Rey | Locución | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Blanca Flor Tinoco Torres | Diseñador de Contenidos Digitales | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Andrea Paola Botello De la Rosa | Desarrollador Full-Stack | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Andrea Paola Botello De la Rosa | Actividad didáctica | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Zuleidy María Ruíz Torres | Validación de Recursos Educativos Digitales | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Luis Gabriel Urueta Álvarez | Validación de Recursos Educativos Digitales | Centro Industrial del Diseño y la Manufactura - Regional Santander |
| Daniel Ricardo Mutis Gómez | Evaluador para Contenidos Inclusivos y Accesibles | Centro Industrial del Diseño y la Manufactura - Regional Santander |