

# FRAMEWORK PARA EL USO DE SINCRONIZACIÓN DE INFORMACIÓN EN TELÉFONOS MÓVILES SIN CONEXIÓN A INTERNET

Bienvenido, este manual de usuario lo guiará en el desarrollo de aplicaciones para que sean capaces de realizar procesos de sincronización de información por medio de SMS cuando la conexión vía Internet falle.

## Requerimientos:

Se requieren dos teléfonos móviles con sistema Android 2.2 en adelante. Uno para realizar las operaciones del lado del cliente y otro para manejar las peticiones del lado del servidor.

Se requiere un servidor web y un gestor de base de datos MySQL.

Para la aplicación del lado del servidor, se necesita que se encuentre en la misma red local que el servidor web, en caso contrario debe estar conectado a una red con conexión a internet muy estable.

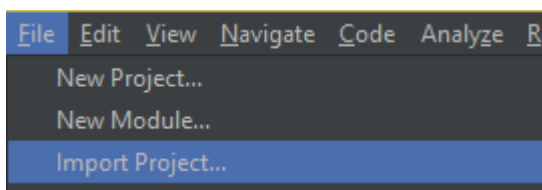
## Paso 1:

Clonar el repositorio de GitHub en su equipo o descárguelo en un ZIP.

## Paso 2:

Una vez descargado el repositorio puede empezar su propio proyecto.

Usando Android Studio ( <http://developer.android.com/sdk/index.html#top> ) seleccione la opción *Import Project*:



Descomprima el ZIP o busque el directorio en el que se guardó el repositorio, importe el proyecto *inventario* y guarde los cambios.

Puede repetir el paso para importar el proyecto *SMSTerver*, o puede instalar directamente la apk en su teléfono Android, esta apk se encuentra en el directorio:

SMSTerver\app\build\outputs\apk con el nombre *app-debug.apk*.

## Paso 3:

Importe el script SQL llamado *android\_connect* en su base de datos MySQL y coloque el directorio *androidconnect* en su servidor web, este directorio será la API web que recibirá las peticiones HTTP y regresará los resultados en formato JSON.

#### Paso 4:

Configure el archivo **db\_config** que se encuentra en la raíz del directorio **android\_connect**, este archivo contiene los valores para acceder a su base de datos.

#### Paso 5:

Configure su aplicación servidor **SMServer**: una vez instalada la app, ejecute y presione el botón **Configurar Servidor**. Se mostrará la siguiente ventana:

En el primer campo escriba el nombre de su dominio incluyendo el prefijo **www**.

Si está trabajando en una red local, ingrese la dirección IP local en la cual está escuchando el **servidor web** en ambos campos. Si el **SMServer** no puede estar en la misma red local que el servidor entonces colocar el nombre del dominio en ambos campos.

Los dos campos deben estar configurados, luego, seleccione **GUARDAR**.



MOVILNET 11:50

Nombre del dominio (Ej: www.example.com)  
www.ejemplo.com

Direccion IP del servidor web (local) (Ej: 192.168.0.100)  
192.168.0.100

GUARDAR

#### Paso 6:

Configure su aplicación cliente **inventario** con AndroidStudio:

- En la clase **IncomingSMS**, línea 26 y 27, configurar las variables **ServerNumber** y **CodePlusServerNumber** con el número de teléfono y el número de teléfono más el código del país respectivamente del servidor móvil en el cual se encontrara la aplicación servidor **SMServer**.
- De igual manera configurar en la clase **JSONParser** en la línea 34, la variable **ServerNumber**. (puede ser necesario colocar el código del país).
- Editar en la clase **EditarProducto** línea 42,45 y 48, clase **NuevoProducto** línea 33 y clase **TodosLosProductos** línea 39, todas las URLs correspondientes a sus operaciones en el API web.

#### Paso 7:

Usted está listo para realizar las operaciones por defecto de la aplicación cliente correspondientes a un inventario, estas operaciones podrán ser realizadas sin necesidad de establecer una conexión a Internet. Para probarlo, desactive el Wi-Fi y las redes de datos del teléfono.

## Crear su propia aplicación:

Si usted ya posee una aplicación o quiere crear una nueva y resolver sus problemas de sincronización usando SMS, usted debe instalar el **SMSTServer** como ya se describió anteriormente, luego, usar las clases **JSONParser** e **IncomingSMS** en su aplicación.

Debe configurar estas clases de la misma manera en la que se describe en el paso 6, además de esto, para que **JSONParser** funcione correctamente también debe incluir la clase **ConnectionDetector**.

Cada vez que vaya a hacer una petición HTTP al servidor para realizar una sincronización debe usar la clase **JSONParser** con su método **makeHttpRequest()**, pasando por parámetros el **Context** de la aplicación, la **url** a la cual va la petición en el API web, el **método** (GET o POST), los **parámetros** y finalmente un valor **verdadero** para realizar la conexión vía SMS si el Internet falla o **falso** para realizar la conexión normal.

Todas las peticiones llegan al servidor web, por lo que debe programar las consultas a la base de datos que se adapten a su aplicación y retornar los resultados en formato JSON para su posterior procesado. De igual manera la aplicación cliente debe contar con una base de datos para guardar la información que se haya consultado y así poder decir que la información se encuentra sincronizada.

Esta es la estructura final que debe tener el sistema:

