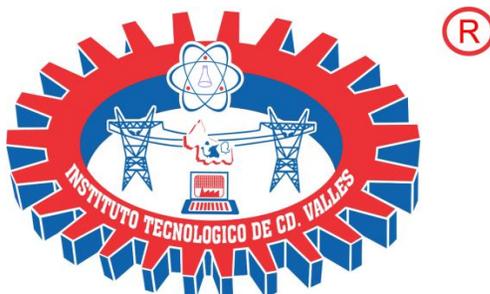


DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA  
INSTITUTO TECNOLÓGICO DE CD. VALLES



**MANUAL TECNICO**

NENEK-SAAC sistema adaptable para el almacenaje de contenidos producidos por comunidades virtuales de hablantes nativos (CVLI)

Ingeniería en Sistemas Computacionales

Emmanuel Domínguez Izaguirre

09690078

Gerardo Alfredo Vázquez Martínez

09690139

Selene Vega Guerrero

09690102

ASESOR INTERNO

ASESOR EXTERNO

Dr. José Luis González Compeán

Dra. Anuschka Johanna Van ' t Hooft



## Índice

1. Introducción.....	1
2. Aspectos del Análisis.....	2
3. Especificaciones Técnicas.....	3
4. Diccionario de Datos .....	4
5. Instalación de aplicaciones.....	9
6. Configuración de aplicaciones .....	13
7. Funcionamiento de la aplicación para ambientes móviles .....	18
1. Funcionamiento de la aplicación nativa de Android .....	40

# 1. Introducción

El presente manual técnico tiene como finalidad describir el diseño del prototipo para la gestión de Nenek-SAAC en ambientes móviles, así como también la aplicación nativa de Android 2.3 GingerBread a 4.1 JellyBean.

La implementación de Nenek-SAAC en ambientes móviles (web) se basa en una adaptación para teléfonos móviles ya que se ha comprobado que conforme avanza la tecnología estos dispositivos cada vez adquieren mejor funcionamiento y mayor adaptabilidad a las diferentes tecnologías en este caso el web, por ello muchas personas han preferido este tipo de dispositivos ya que son más prácticos y fácil de llevar debido a su tamaño y peso.

La implementación de la aplicación nativa de Android cuenta con una protección de la información por lado del servidor, así como también se toma en cuenta los posibles ataques que pueden surgir en este nuevo medio, y la utilización de la misma es responsabilidad de cada usuario. Resulta ser bastante fácil de implementar puesto solo es necesario instalar los paquetes básicos disponibles en cualquier distribución de Ubuntu.

La interconexión se centra en la gestión de los contenidos producidos por las demás aplicaciones que se encuentran en la plataforma de Nenek.mx. Con esto se tendrá un control de las aplicaciones y se pueden generar estadísticas respecto a las preferencias de los usuarios.

## **2. Aspectos del Análisis**

### **2.1. Herramientas tecnológicas**

Los ambientes móviles abarcan un amplio mercado para los potenciales usuarios que se internan a las nuevas tecnologías específicamente hablando, las comunidades virtuales de hablantes de la lengua indígena (CVLI). Estos últimos, con el paso del tiempo desarrollaron un particular interés por los antes mencionados con forme pasan los años. Estas nuevas tecnologías involucran nuevos paradigmas de programación, modelado de contenidos, herramientas multimedia, herramientas de conectividad a bases de datos, herramientas de seguridad y hardware.

La aplicación web se puede utilizar en dispositivos móviles que soporte las nuevas tecnologías de programación web en este caso html5, css3 y el framework bootstrap 2.8. En estos tiempos en que la tecnología ha avanzado la mayoría de los dispositivos móviles soportan este tipo de tecnología. Es por ello que esta aplicación va orientada a ese tipo de dispositivos. Una de las ventajas que se presentan en este tipo de tecnología es que tiene mayor portabilidad y por eso muchos usuarios la utilizan.

En su totalidad la aplicación nativa para Android solo puede operar en teléfonos con sistemas operativos Android que comprenden desde la versión 2.3 (GingerBread) a la versión 4.1 (JellyBean). El motivo de usar este sistema operativo, se debe a la creciente popularidad la cual incrementa a un ritmo exponencial que conlleva esta plataforma respecto a sus competidores, también otra de las grandes ventajas que tiene esta plataforma es que se habla de un software libre.

### 3. Especificaciones Técnicas

**Sistema Operativo:** GNU/Linux Ubuntu 12.04 Precise Pangolin o versiones superiores.

**Manejador de Base de Datos:** Postgresql 9.1.

**Lenguaje de Programación:** PHP 5.4, HTML5, CSS3, Framework (Bootstrap 2.8), JavaScript y Java JDK 7.

**Servidor de Aplicaciones:** Servidor Web Apache 2.0, Servidor.

**Navegador Web:** Cualquier versión de Google Chrome (para trabajar en la aplicación) y funciona en todos los exploradores Web.

**IDE:** Android Developer Tools Plugin para Eclipse IDE (Eclipse IDE, SDK Tools)

**Teléfono Celular:** Sistema operativo Android 2.3 GingerBread a 4.1 JellyBean.

## 4. Diccionario de Datos

Los datos a utilizar se manejarán en las tablas ya existentes de la base de datos de Nenek-SAAC las cuales son users, file\_user, tareas, user\_grupo, files, grupos, file\_grupo, mail, tareas.

**Tabla 1. Estructura de la tabla users en la base de datos en Nenek-SAAC**

Campo	Tipo	A utilizar
id_user	Text	X
name	Text	X
password	Text	X
avatar	Text	X
hora	time without time zone	X
fecha	Date	X
sesion	Text	X
email	Text	X
folder_user	Text	X

**Tabla 2. Estructura de la tabla file\_user en la base de datos Nenek-SAAC**

Campo	Tipo	A utilizar
id_grupo	text	X
Id_user	text	X
autorizado	text	X
Id	text	X
fecha	text	X

**Tabla 3. Estructura de la tabla tareas en la base de datos Nenek-SAAC**

Campo	Tipo	A utilizar
id_tarea	text	X
director	text	X
tarea	text	X
inicio	text	X
fin	text	X
descripción	text	X
id_user	text	X
id_grupo	text	X
allowed	text	X
status	text	X

**Tabla 4. Estructura de la tabla grupos en la base de datos de Nenek-SAAC**

Campos	Tipo	A utilizar
id_grupo	text	X
nombre	text	X
dueño	text	X
open	text	X
fecha	text	X

**Tabla 5. Estructura de la tabla files en la base de datos de Nenek-SAAC**

Campos	Tipo	A utilizar
id_file	text	X
id_user	text	X
file_name	text	X
ip	text	X
public	text	X
size_file	text	X
fecha	date	X
hora	time without time zone	X
padre	text	X
type	text	X
subpadre	text	X
url_movil	text	X
type_archivo	text	X
nombre_file	text	X

**Tabla 6. Estructura de la tabla file\_users de la base de datos de Nenek-SAAC**

Campo	Tipo	A utilizar
id_user	text	X
id_file	text	X

**Tabla 7. Estructura de la tabla file\_grupo de la base de datos de Nenek-SAAC**

Campo	Tipo	A utilizar
id_grupo	text	X
id_file	text	X
fecha	text	X

**Tabla 8. Estructura de la tabla mail en la base de datos de Nenek-SAAC**

Campos	Tipo	A utilizar
id_msg	text	X
envia	text	X
recibe	text	X
asunto	text	X
status	text	X
cuerpo	text	X
hora	text	X
fecha	text	X
dueno	text	X

**Tabla 9. Estructura de la tabla tareas en la base de datos de Nenek-SAAC**

Campos	Tipo	A utilizar
id_tarea	text	X
director	text	X
tarea	text	X
inicio	text	X
fin	text	X
descripcion	text	X
Id_user	text	X
Id_grupo	text	X
allowed	text	X
status	text	X

A continuación se describe brevemente la función de la aplicación para ambientes móviles (web).

Nombre	Función	Localización	Tamaño	No. De Líneas	Incluye
<b>profile.php</b>	Archivo que permite visualizar los archivos del usuario, también poder subir archivos.	/www/var/nenek/cenenek/	28,4 kb	704	sesion.php, dbconnect.php, common.php
<b>mail.php</b>	Permite visualizar los mensajes enviados y recibidos por el usuario o grupos de trabajo.	/www/var/nenek/cenenek/	20,6 kb	573	sesion.php, dbconnect.php, common.php
<b>Working_group.php</b>	En este archivo se pueden crear grupos de trabajo y también	/www/var/nenek/cenenek/	18,4 kb	512	sesion.php, dbconnect.php, common.php

	entrar en ellos y visualizar los archivos compartidos en cada grupo.				
<b>colaborativo.php</b>	Archivo en el cual se visualizan las tareas asignadas a cada usuario así como también las tareas personales.	/www/var/nenek/cenenek/	12,1 kb	314	sesion.php, dbconnect.php, common.php
<b>galeria.php</b>	Se obtienen las imágenes subidas por el usuario y se visualizan en el presente archivo.	/www/var/nenek/cenenek/	4,7 kb	147	sesion.php, dbconnect.php, common.php
<b>edit_profile.php</b>	En este archivo se pueden actualizar los datos del usuario.	/www/var/nenek/cenenek/	12,8 kb	305	sesion.php, dbconnect.php, common.php
<b>download.php</b>	Archivo en cual se genera la función para poder descargar los archivos de cada usuario.	/www/var/nenek/cenenek/	3,8 kb	79	dbconnect.php, config.php
<b>enviar.php</b>	Archivo que contienen las funciones para enviar mensajes a usuarios o grupos de trabajo.	/www/var/nenek/cenenek/	3,5 kb	94	dbconnect.php
<b>new_grupo.php</b>	Permite crear y comprobar si existen grupos de usuarios en la base de datos.	/www/var/nenek/cenenek/	1,3 kb	47	sesion.php, dbconnect.php
<b>new_tarea.php</b>	Archivo que permite asignar tareas a usuarios y grupos registrados en Nenek-SAAC.	/www/var/nenek/cenenek/	2,9 kb	91	dbconnect.php
<b>new_user.php</b>	Permite registrar y verificar si existe algún usuario registrado en Nenek-SAAC.	/www/var/nenek/cenenek/	10,1 kb	245	dbconnect.php
<b>busca_usuario.php</b>	Permite realizar una búsqueda de los usuarios registrados.	/www/var/nenek/cenenek/	2,2 kb	79	dbconnect.php
<b>forgotpass.php</b>	Permite recuperar la contraseña del usuario registrado.	/www/var/nenek/cenenek/	4,5 kb	126	dbconnect.php, common.php
<b>abandonar_grupo.php</b>	Archivo que contiene la función para eliminar a un usuario	/www/var/nenek/cenenek/	2,3 kb	48	dbconnect.php

<b>p</b>	de un grupo de trabajo.				
<b>aceptar_solicitud.php</b>	Permite aceptar usuarios a grupos de trabajo.	/www/var/nenek/cenenek/	3,1 kb	65	dbconnect.php

A continuación se describe brevemente la función de la aplicación nativa para Android.

<b>Nombre</b>	<b>Función</b>	<b>Localización</b>	<b>Tamaño</b>	<b>No. De Líneas</b>	<b>Incluye</b>
<b>Archivos.class</b>	Despliega una lista con los archivos de cada usuario	/src/com/	23,6 kb	726	
<b>ArchivoElegido.class</b>	Muestra al usuario un archivo específico	/src/com/	18,5 kb	475	
<b>Grupos.class</b>	Interfaz para los grupos de trabajo	/src/com/	62,2 kb	1656	
<b>Inicio.class</b>	Presentación de la aplicación	/src/com/	2,7 kb	111	
<b>Lenguaje.class</b>	Interfaz para realizar cambio de idioma	/src/com/	1,6 kb	59	
<b>MainActivity.class</b>	Interfaz login	/src/com/	11,5 kb	366	
<b>Perfil.class</b>	Interfaz para poder cambiar los datos personales del usuario	/src/com/	16,6 kb	483	
<b>Registro.class</b>	Interfaz para registrar nuevos usuarios	/src/com/	9,9 kb	280	
<b>Solicitudes.class</b>	Interfaz para aceptar peticiones de usuarios para entrar al grupo del usuario	/src/com/	13,4 kb	361	
<b>Subir.class</b>	Interfaz para subir contenidos a Nenek-SAAC	/src/com/	13,2 kb	374	
<b>SubirFotoPerfil.class</b>	Interfaz para cambiar la imagen de perfil del usuario	/src/com/	13,4 kb	432	
<b>DialogCompartir.class</b>	Dialogo para compartir archivos	/src/com/	11 kb	304	

	a un grupo autorizado			
<b>DialogoList.class</b>	Dialogo para listar las tareas de un usuario	/src/com/	14,3 kb	400
<b>JSONParser.class</b>	Constructor de JSON	/src/com/helpers/	5,4 kb	159
<b>SesionManager.class</b>	Clase gestora de sesiones	/src/com/helpers/	5,6 kb	214

## 5. Instalación de aplicaciones

Para implementar de manera correcta Nenek-SAAC en un servidor se deben primero instalar ciertos programas, los cuales necesitan de una conexión a internet de forma obligatoria.

La manera típica de instalar programas en el sistema operativo Ubuntu es mediante una terminal o mediante el centro de software de ubuntu, sin embargo, algunos programas no requieren una instalación previa puesto que vienen preparados para funcionar tan pronto su descarga termina como es el caso de Eclipse.

Para abrir una terminal se puede presionar la combinación de teclas: **Ctrl + Alt + T** y en instantes aparecerá una terminal para realizar operaciones. En la terminal se procede a entrar como super usuario tecleando `sudo -s` y posteriormente introduciendo la contraseña que el equipo solicita.

Una vez estando en modo "root", se procede a instalar y configurar Apache2, Php5, Postgresql para tener la aplicación de Nenek-SAAC en el servidor lista y así permitir la entrada de usuarios. Después se procederá a instalar ADT Plugin para Eclipse.

### 5.1. Instalación de Apache2

Con una terminal en modo super usuario se procede a instalar el servidor web de apache2, tecleando en la terminal el comando descrito a continuación y seguido de la tecla enter.

```
# apt-get install apache2
```

```
root@emmanuel-pc:/home/emmanuel# sudo apt-get install apache2
```

En seguida se muestra el proceso de instalación y la terminal nos pide nuestra autorización para continuar con la instalación de apache2. Para autorizar basta con presionar la tecla S (Y en el lenguaje ingles), ó N para cancelar.

Para comprobar que apache2 se ha instalado correctamente basta con abrir cualquier navegador web y teclear “localhost” en la barra de direcciones o URL, y deberá aparecer el mensaje “It Works! this is the default web page for this server”

## **5.2. Instalación de Php5**

El proceso para instalar el lenguaje de servidores Php5 es muy similar al anterior descrito, la diferencia radica en que se debe cambiar el comando de instalación por:

```
# apt-get install php5
```

```
root@emmanuel-pc:/home/emmanuel# sudo apt-get install php5
```

Al igual que en apache2 se mostrará una ventana que nos pide la autorización del sistema para continuar con la instalación. Nuevamente se presiona la tecla S para permitir este proceso.

## **5.3. Instalación de Postgresql**

Continuamos ahora con la instalación del sistema gestor de bases de datos Postgresql 9.1 el cual es muy similar a la instalación de php5 se debe tener una terminal con modo super usuario, y se procede a teclear el siguiente comando:

```
# apt-get install postgresql
```

```
root@emmanuel-pc:/home/emmanuel# sudo apt-get install postgresql
```

Nuevamente aceptamos el proceso de instalación presionando la tecla S al momento de que la terminal nos pida nuestra autorización.

## **5.4. Conectando Apache2 con Postgresql y Php**

Ahora que se tienen instalados Apache2, Postgresql y Php5 se deben conectar todas estas aplicaciones, esto se logra instalando ciertas librerías y/o módulos como son pg\_sql, libmod-apache2-php5, para comenzar se procederá a instalar php5-pgsql el cual conecta el sistema gestor de bases de datos Postgresql con el lenguaje de programación Php5.

Para instalar pgsq1 tecleamos en una terminal la siguiente instrucción:

```
# apt-get install php5-pgsql
```

```
root@emmanuel-pc:/home/emmanuel# sudo apt-get install php5-pgsql
```

Ahora se instala libmod-apache2-php5, este modulo conecta el servidor apache2 con el lenguaje de servidores php5, para instalarlo se teclea en una terminal la siguiente instrucción:

```
#apt-get install libmod-apache2-php5
```

```
root@emmanuel-pc:/home/emmanuel# sudo apt-get install libapache2-mod-php5
```

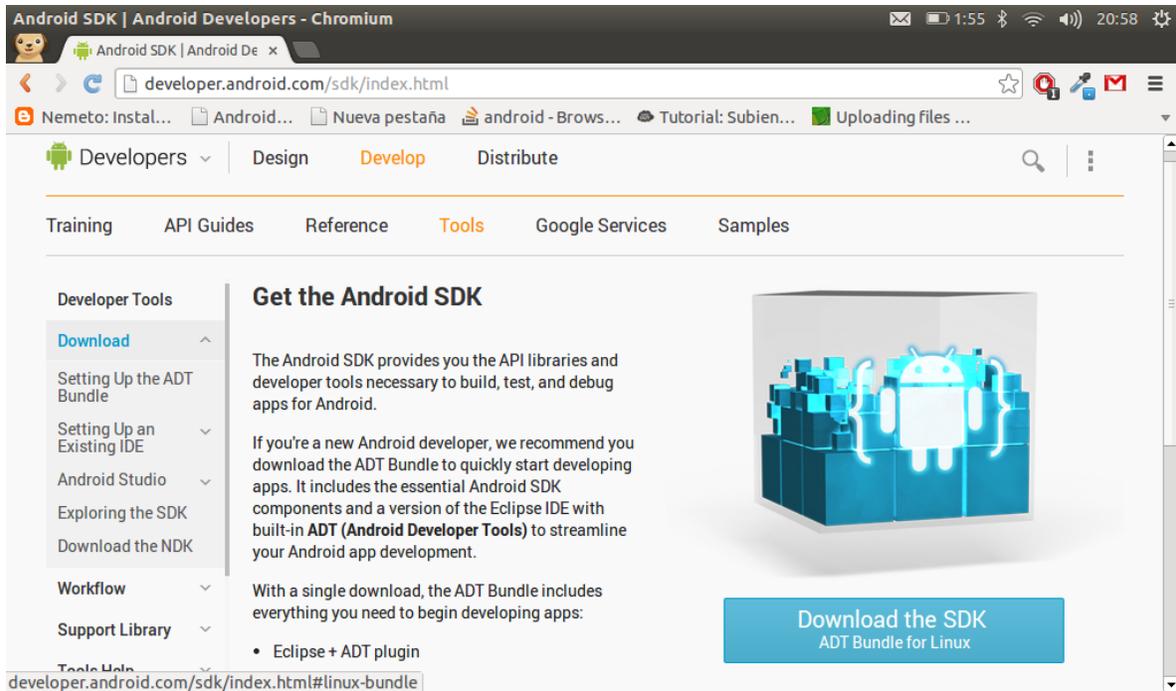
## 5.5. Instalación de ADT plugin para Eclipse

La instalación del IDE para desarrollar en el sistema operativo Android tiene como nombre ADT, existen diversas formas de instalar este IDE, pero en este apartado se explica la forma más sencilla de obtener este IDE listo para desarrollar. Este plugin consta del IDE Eclipse, un entorno de Java y el Software Development Kit (SDK) para Android.

Se comienza abriendo un navegador diferente a internet explorer 9 y se teclea la siguiente dirección en su URL:

<http://developer.android.com/tools/index.html>

Después ahí acceder a la sección Download y presionamos el botón que dice "Download the SDK ADT Bundle for Linux". Este IDE tiene un peso aproximado de 430 megabytes por lo demorara un poco en conexiones de internet lentas.



**Imagen de la página de descargas de Android developer.**

Una vez descargado el ADT, se crea una carpeta nueva con el nombre “Android” en el lugar donde se desee colocar este IDE y se busca dentro de la carpeta Eclipse el archivo eclipse. Si se realiza desde una terminal, usted puede ejecutar el comando dentro de la carpeta en la que se encuentre el archivo.

```
# ./eclipse
```

En caso de usar el modo gráfico usted puede dar doble clic sobre el archivo que se muestra a continuación.

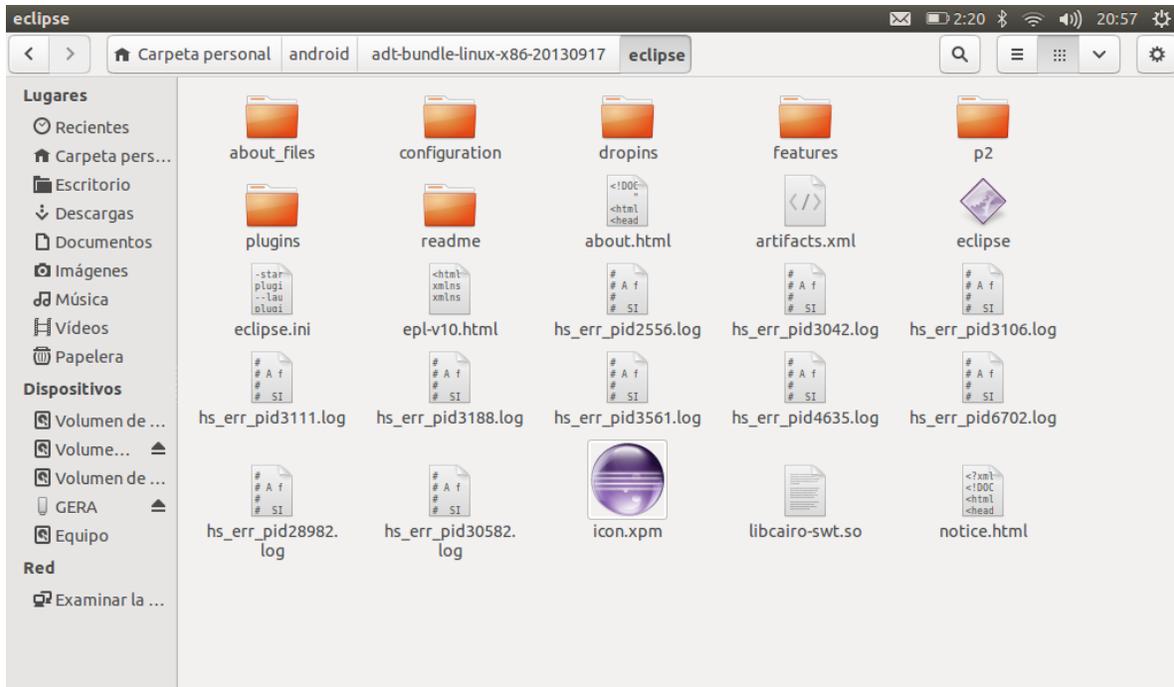


Imagen del archivo eclipse ejecutable desde modo gráfico.

## 6. Configuración de aplicaciones

Para que las aplicaciones cumplan las funciones esperadas se deben configurar ciertos archivos internos de las mismas. Para esto accederemos a una terminal en modo super usuario.

### 6.1. Configuración de apache2

La configuración de apache2 consta en modificar el directoryIndex para cambiar el orden de ejecución de archivos, este se encuentra en el archivo dir.conf, para esto desde una terminal se debe teclear el siguiente comando

```
# nano /etc/apache2/mods-enabled/dir.conf
```

```
root@emmanuel-pc:/home/emmanuel# nano /etc/apache2/mods-enabled/dir.conf
```

Ahí se encuentra el orden de ejecución de archivos por tanto se debe poner el archivo index.php como primer archivo a ejecutar, esto deberá quedar así:

```
DirectoryIndex index.php index.cgi index.pl index.html index.xhtml index.htm
```

```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.cgi index.pl index.html index.xhtml index.htm
</IfModule>
```

En seguida se guardan los cambios y se realiza un reinicio en el servidor con el comando:

```
# /etc/init.d/apache2 restart
```

```
root@emmanuel-pc:/home/emmanuel# /etc/init.d/apache2 restart
```

## 6.2. Configuración de Postgresql

Para configurar el sistema gestor de bases de datos primero se debe editar el archivo postgresql.conf tecleando en una terminal la siguiente línea:

```
# nano /etc/postgresql/9.1/main/postgresql.conf
```

```
root@emmanuel-pc:/home/emmanuel# nano /etc/postgresql/9.1/main/postgresql.conf
```

En este archivo se debe descomentar la línea no.59 la cual indica

```
#listen addresses='localhost'
```

```
#-----
# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -
#listen_addresses = 'localhost'          # what IP address(es) to listen on;
#                                         # comma-separated list of addresses;
#                                         # defaults to 'localhost', '*' = all
#                                         # (change requires restart)
port = 5432                              # (change requires restart)
max_connections = 100                    # (change requires restart)
# Note: Increasing max_connections costs ~400 bytes of shared memory per
# connection slot, plus lock space (see max_locks_per_transaction).
```

Y escribir:

```
listen_addresses='*'
```

En seguida se guarda el documento y se reinicia el servicio de postgresql tecleando

```
# /etc/init.d/postgresql restart
```

```
root@emmanuel-pc:/home/emmanuel# /etc/init.d/postgresql restart
```

Ahora se cuentan con la base de datos de Nenek-SAAC la cual tiene por nombre cenenek por lo que ahora se le deben de dar los permisos a los usuarios de la base de datos, se debe teclear en una terminal la siguiente instrucción:

```
# nano /etc/postgresql/9.1/main/pg_hba.conf
```

```
root@emmanuel-pc:/home/emmanuel# nano /etc/postgresql/9.1/main/pg_hba.conf
```

E incluir ahí la instrucción que dará de alta a la base de datos local para que el servidor pueda acceder con libertad a ella mediante la siguiente línea en la línea 14 del archivo:

**Host cenenek postgres 127.0.0.1/32 trust**

```
host  cenenek postgres  127.0.0.1/32  trust
```

Después de guardar este archivo, nuevamente se reinicia el servicio de postgresql para que los cambios realizados surtan efecto con el siguiente comando:

```
# /etc/init.d/postgresql restart
```

```
root@emmanuel-pc:/home/emmanuel# /etc/init.d/postgresql restart
```

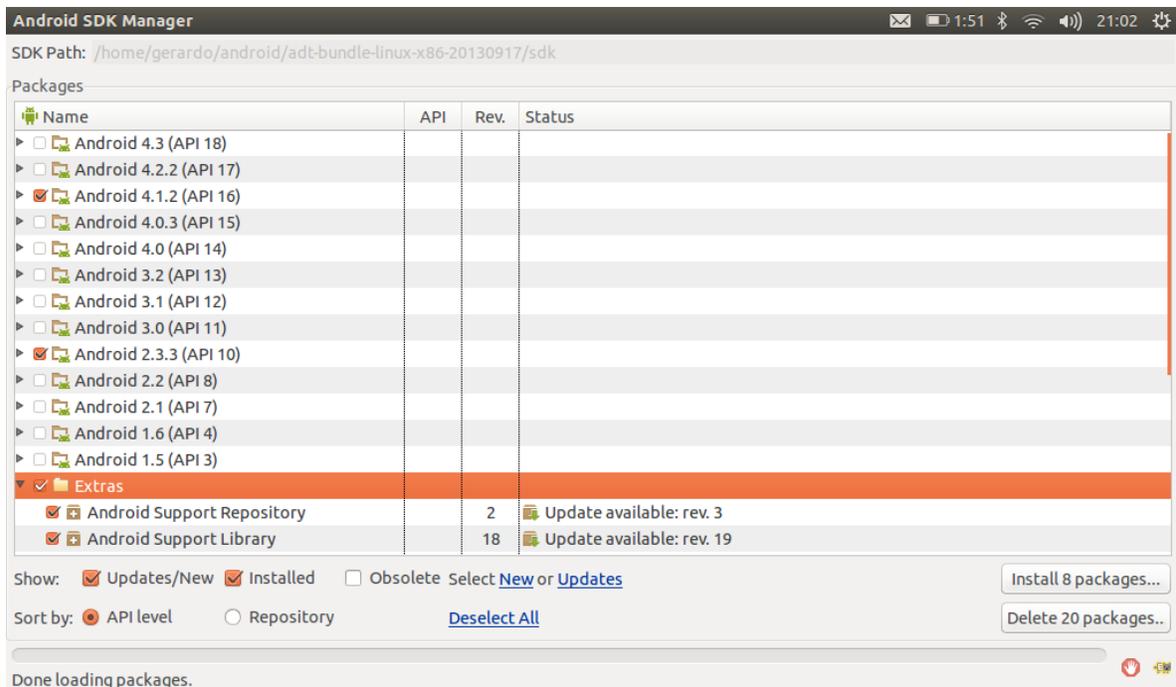
### 6.3. Configuración de SDK

Ahora que se tiene eclipse instalado se proceden a instalar los distintos emuladores de cada uno de los sistemas operativos con los que cuenta Android. Hacer click en el icono de Android SDK y seleccionar los paquetes necesarios para instalar.



Imagen del icono de SDK

En nuestro caso se deberá instalar como mínimo todo el kit de Android 4.1 y 2.3 así como también todas las google API's que se ofertan.



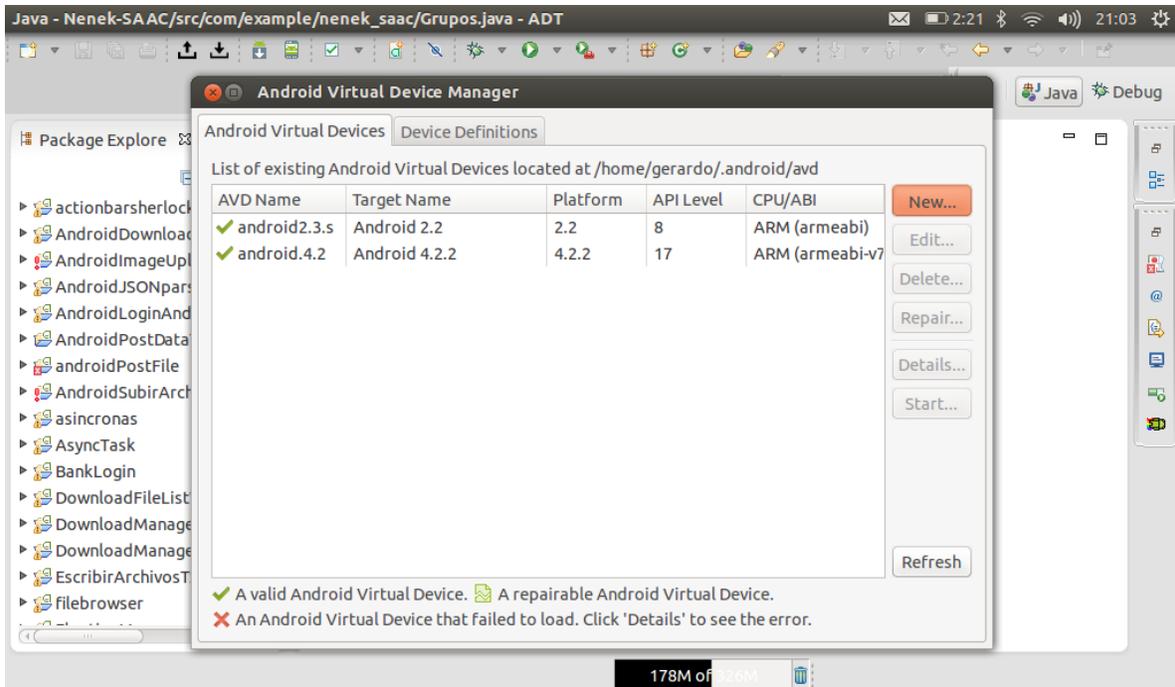
**Imagen del SDK Manager.**

Ahora se debe crear un Android Virtual Device (AVD) el cual consiste de un emulador de un dispositivo celular. Para ello se debe hacer click en el icono de Android AVD.



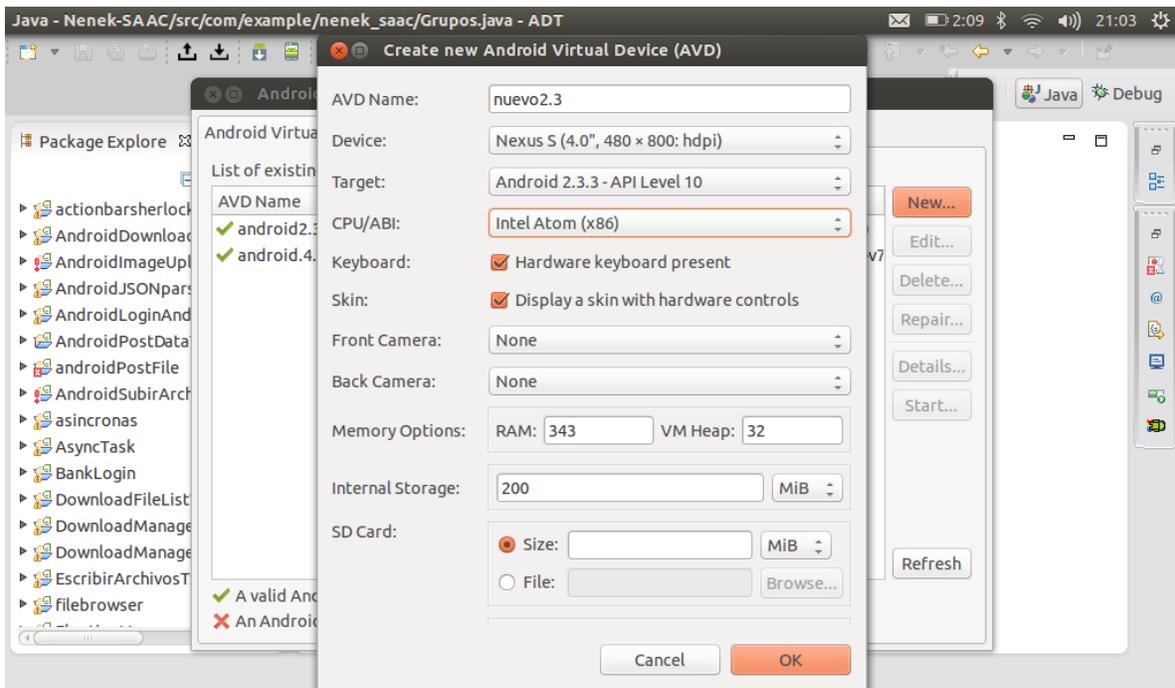
**Imagen del icono de AVD.**

Hacer click en “new device” y aparecerá la siguiente ventana, donde se debe elegir desde la versión de Android que emulara el dispositivo, la marca del teléfono, su procesador etcétera.



**Android Virtual Device Manager, crear nuevo AVD.**

Nota: Entre mayor sea la capacidad de su AVD, mas recursos consumirá de su computadora. Se recomienda crear un dispositivo 2.3 y uno 4.1 para realizar las pruebas.



**Creación de un nuevo AVD.**

## 7. Funcionamiento de la aplicación para ambientes móviles

En los siguientes funcionamientos se explica solo lo esencial para que funcionen las aplicaciones de la plataforma como por ejemplo: realizar un registro, control de acceso a la plataforma, listado de archivos, compartimiento de archivos, etc.

### 1.1. Funcionamiento del sistema de login

En el sistema del login el usuario envía sus datos y se obtienen mediante un archivo que verifica si existe o no el usuario, si existe permite el acceso al perfil.

#### ARCHIVO INDEX.PHP

```
<!--IniciaPHP-->
<?php
session_start();
//Se verifica si ya hay una sesión activa, si existe alguna, lo redirige al archivo del perfil, si no se
queda en el index.php
if(isset($_SESSION['usuario'])){
    header('Location: profile.php');
} else {
    header('index.php');
}
//Verifica el tipo de lenguaje seleccionado
include_once 'common.php';
//Archivo que contiene la conexión a la base de datos
include 'dbconnect.php';
include 'log.php';
//Comprueba si se envía algo en el formulario del login, si se envía algo inicia la comprobación de
usuario.
if(isset($_POST['entrar'])){
//Se crean las variables que se obtienen por el método POST.
$id_user = htmlspecialchars($_POST['email'], ENT_QUOTES);
$password = htmlspecialchars($_POST['password'], ENT_QUOTES);

//Se realiza la consulta para verificar si coinciden los datos enviados por el usuario.
$query = "SELECT * FROM users WHERE email = '$id_user' AND password = '$password'";
//se obtienen los resultados de la búsqueda y se almacenan en la variable result.
$result = pg_query($conn,$query);
//Si no encuentra resultados verdaderos en la consulta nos imprime un error de consulta.
if(!$result){
    echo "Fallo query: $query.<br/>";
} else {
    //asocia los resultados a una variable num
    $num = pg_num_rows($result);
    //comprueba si el número de resultados es menor o igual que cero imprime un mensaje de error
de usuario o contraseña.
    if($num <= 0) {
        $message = "<small style='font-size:100%;' title='Esta mal escrito tu usuario o
contrase&ntilde;a.'>".$lang['LOGIN_USR_CTN_MAL']."</small>";
    }
    //si los resultados son mayor que cero, realiza las siguientes funciones
```

```

else {
    //asocia resultados obtenidos a una variable row
        $row = pg_fetch_assoc($result);
        $activacion = $row['sesion'];
    //se crea una variable con la id del usuario.
    $id_user = $row['id_user'];
    //Se crean variables de sesion
    $_SESSION['connected'] = 1;
    $_SESSION['usuario'] = $id_user;

    //se hace un update a la tabla users el cual cambia el valor de la sesion a si
    $query = "UPDATE users SET sesion ='si' where id_user='$id_user'";
        $result = pg_query($conn,$query) or die("Fallo1 query: $query.<br/>");
    //Nos redirije a la pagina profile.php
        header("Location: profile.php");
    }
}
}

```

## 1.2. Funcionamiento del sistema de registro

En el sistema de registro el usuario tendrá que enviar sus datos mediante un formulario html, estos datos se envían por método POST y se verifica si los datos coinciden procede a registrar el usuario.

### ARCHIVO NEW\_USER.PHP

```

<?php
//Se verifica si en realidad se envia algo del formulario de registro.
if(isset($_POST['entrar']))){
    //De la misma se comprueba si se envia algo de los campos de usuario contraseña y nombre.
    if(isset($_POST['id_user']) && isset($_POST['password']) && isset($_POST['name'])){
        //Se incluye el archivo de conexion a la base de datos
        include 'dbconnect.php';
        //se crea la variable id_user apartir del dato enviado por POST
        $id_user = $_POST['id_user'];
        $id_user = filter_var($id_user,FILTER_SANITIZE_STRING);
        //se crea la variable name apartir del dato enviado por POST
        $name = $_POST['name'];
        $name = filter_var($name,FILTER_SANITIZE_STRING);
        $name = htmlentities($name);
        //se crea la variable password apartir del dato enviado por POST
        $password = $_POST['password'];
        //se crea la variable nemail apartir del dato enviado por POST
        $nemail = $_POST['email'];
        //se crea la variable avatar apartir del dato enviado por POST
        $avatar = "images/nueva.png";

        //se realiza una consulta a la tabla usuarios enviando el dato de usuario.
        $query_1 = "SELECT * FROM users WHERE id_user='$id_user'";
        $result_1 = pg_query($conn,$query_1) or die("Fallo query: $query.<br/>");
        //se crean las variables hora, fecha y activacion.
        $hora= date("H:i:s");
    }
}

```

```

$fecha = date("Y/n/d");
$activacion=uniqid();
//Verifica el resultado obtenido apartir de la consulta.
//Si no existe algun usuario existente en la BD procede a realizar las siguientes funciones.
if(pg_num_rows($result_1)<=0){

    //En esta parte se hace una modificacion al campo de usuario.
    //Eliminan los caracteres asignados en el arreglo.
    $arreglo=array(
    ", . : * / < > ? " ; \ " ' " : . " ! " | " " " @ " % " + " ~ " # " & " = " ] " } " { " [ " $ " " " ^ " )";
    //se crea la variable id_user ya limpia de caracteres especiales.
    $id_user=str_replace($arreglo,"",$id_user);

    $time=time();//fecha del registro del usuario
    $allowed="no";//no se permite entrar hasta que revise su correo y valla al link
    $type_user = "Normal";//tipo de usuario
    $avatar2="resources/img/nenek.png";//imagen por default
    //se crea la carpeta del usuario con un md5 e uniqid
    $name_folder_user= md5(uniqid());//carpeta del usuario

    //se realiza una insercion de datos a la tabla users
    $query = "INSERT INTO users
VALUES('$id_user','$name','$password','$avatar','$hora','$fecha','$activacion','$nemail','$name_folder_user');"
    $result = pg_query($conn,$query);

    $query2 = "INSERT INTO usuarios
(id_user,password,nombre,email,folder_user,avatar,time,tipo_usuario) VALUES
('$id_user','$password','$name','$nemail','$name_folder_user','$avatar2','$time','$type_user)";
    // $query2 = "INSERT INTO usuarios VALUES('$id_user','$password',)";
    $result = pg_query($nemekcon,$query2);

    //Parte donde se envia un mensaje al usuario donde se le da la bienvenida al sistema Nenek-
    SAAC
    $id_msg=uniqid();
    $fecha = date("Y/m/d");
    $hora = date("H:i:s");

    $asunto="Bienvenido Nenek-SAAC te da la Bienvenida";
    $msg="Bienvenido ".$id_user." tu contrasena es: ".$password." Nenek-SAAC te da la
    Bienvenida, y te invita a que uses las diferentes opciones de trabajo:
    Grupos de trabajo,Mensageria Interna y Almacenamiento de tus archivos personales";
    $query = "insert into mail values
('$id_msg','TreeStore','$id_user','$asunto','0','$msg','$hora','$fecha','$id_user)";
    $result = pg_query($conn,$query);

    $destinatario= $nemail;
    $asunto= "Activacion de cuenta Nenek-SAAC";
    $link= "http://www.adaptivez.org.mx/TreeStore/activacion.php?activacion=".$activacion."";
    $link = "<a href=$link>activar mi cuenta</a>";
    $cuerpo= "$msg <br/> Para activar tu cuenta en TreeStore entra a la siguiente direccion,
    ".$link;
    $headers = "MIME-Version: 1.0\r\n";
    $headers .= "Content-type: text/html; charset=iso-8859-1\r\n";
    $headers .= "From: Nenek-SAAC <TreeStore@adaptivez.org.mx >\r\n";

```

```

mail($destinatario,$asunto,$cuerpo,$headers);

if(!$result){
  echo "Fallo query: $query<hr/>";
}

if(!$result){
  echo "Fallo query:$query<br/>";
} else {

  $ruta = "$profiles/$name_folder_user";
  $file1="index2.php";
  $file2="index.php";
  mkdir("$ruta", 0777, true);
  chmod("$ruta", 0777);
  copy("$file1", "$ruta/$file2");

  //include 'disperse_folders.php';
  header("Location: index.php?msg=$id_user");
}
} else {
  echo '
<script language="javascript" type="text/javascript">
  alert("Ya existe un usuario con ese ID.");
</script>';
}
} else {
  echo '
<script language="javascript" type="text/javascript">
  alert("Debes llenar todos los campos!");
</script>';
}
}
?>

```

### 1.3. Funcionamiento del sistema de listado de archivos

El listado de archivos se muestra en una tabla la cual contiene archivos con diferentes acciones como descargar archivo, eliminar, o compartirlo.

#### ARCHIVO PROFILE.PHP

```

//Lista de Archivos mostrados en el perfil
if(($mis_archivos==1 || ($mostar_list==2 || $mostar_list==1)) && ($grupos!=1)) {
  echo "<h3> <p title='Mis Archivos'>".$lang['CUERPO_MIS_ARCHIVOS'].</p></h3>";
  $opcion=1;

  //Funcion que muestra la lista de archivos en el perfil de usuario
  list_files($id_user,$mostar_list,$file_search,$extension_search,$inicio,$redundancia,$compartido,$id_grupo,$status,$dueno,$opcion,$padre,$lang);
}

```

```
}
```

## ARCHIVO CONFIG.PHP

//función en la cual se generan las consultas y se crea la tabla en la cual se enlistan los archivos del usuario.

```
function
```

```
list_files($id_user,$mostar_list,$file_search,$extension_search,$inicio,$redundancia,$compartido,$id_grupo,$s  
tatus,$dueno,$opcion,$padre,$lang) {
```

```
//Se incluye el archivo de conexion con la bd
```

```
include 'dbconnect.php';
```

```
if ($mostar_list==1) {
```

```
if ($inicio=="")
```

```
    $inicio=0;
```

```
if($padre!="") {
```

```
    $query = "SELECT * FROM files WHERE id_user = '$id_user' and padre='$padre' order by type,
```

```
id_file desc limit 15 offset '$inicio';
```

```
    } else if($compartido==1) {
```

```
        $d=1;
```

```
        $query = "select *from files, file_grupo where file_grupo.id_grupo='$id_grupo' and
```

```
file_grupo.id_file=files.id_file order by files.type, files.id_file;";
```

```
    } else {
```

```
        $query = "SELECT * FROM files WHERE id_user = '$id_user' and padre='/' order by type, id_file desc  
limit 15 offset '$inicio';
```

```
    }
```

```
    $result = pg_query($conn,$query);
```

```
    if(!$result) {
```

```
        echo "Fallo query: $query<hr/>";
```

```
    } else {
```

```
        if(pg_num_rows($result) == 0) {
```

```
            echo $lang['TABLA_SIN_ARCHIVOS'];
```

```
        } else {
```

```
            echo '<table class="table table-bordered"><thead><tr><th
```

```
title="Archivos">'. $lang['TABLA_ARCHIVOS']. '</th><th colspan="4"
```

```
title="Opciones">Opciones</th><th>Logs</th></tr></thead>';
```

```
            while($row = pg_fetch_assoc($result)) {
```

```
                $id_file = $row['id_file'];
```

```
                $folder = $row['type'];
```

```
                $ip = $row['ip'];
```

```
                $id_user2 = $row['id_user'];
```

```
                $file = $row['file_name'];
```

```
                $file = utf8_decode($file);
```

```
                $public = $row['public'];
```

```
                echo "<tr><td>";
```

```
                if($folder==1) {
```

```
                    echo '&nbsp;';
```

```
                }
```

```
                else {
```

```
                    $type = pathinfo($file);
```

```
                    $extension = $type['extension'];
```

```
                }
```

```
                include 'extension_mime.php';
```

```
                echo basename($file). '</td>';
```

```
                $fn=basename($file);
```

```
                if($dueno==$id_user) {
```

```

        $ver=1;
        $oculta12=2;
    } else
    if($status=='yes')
    {
        $ver=1;
        $oculta12=2;
    } else
        if($status=='no')
        {
            $ver=2;
            $oculta12=2;
        }

    if($ver==1)
    {
        if($folder==1)
        {
            echo '<td><a href="profile.php?id='.$_SESSION['id_user'].'&padre='.$file.'" title="Abrir"><i
onclick="refreshlt()" class="fa fa-arrow-right fa-1x" title="Abrir"></i>&nbsp;'.LANG['TABLA_ABRIR'].'</a></td>';

            $nfl=1; }
            else
            {
                echo '<td><a
href="download.php?id='.$_SESSION['id_file'].'&id_user='.$_SESSION['id_user'].'&redundancia='.$redundancia.'&d=perfil"><i
onclick="refreshlt()" class="fa fa-download fa-1x"
title="Descargar"></i>&nbsp;'.LANG['TABLA_DESCARGAR'].'</a></td>';
            }
            echo '<td><a href="delete.php?id='.$_SESSION['id_user'].'&u='.$_SESSION['id_file'].'&d='.$d.'" title="Eliminar"><i
class="fa fa-times fa-1x"></i>&nbsp;'.LANG['TABLA_ELIMINAR'].'</a></td>';

            if($opcion ==1) {
                if($public == 1){
                    echo '<td><a class="popup" href="compartir.php?id='.$_SESSION['id_user'].'&f='.$_SESSION['id_file'].'&fn='.$fn.'" <i
class="fa fa-share fa-1x"
title="Compartir"></i>&nbsp;'.LANG['TABLA_COMPARTIR'].'</a></td><td>P&uacute;blico</td>';
                }
                else {
                    echo '<td><a class="popup" href="compartir.php?id='.$_SESSION['id_user'].'&f='.$_SESSION['id_file'].'&fn='.$fn.'"
title="Compartir"><i class="fa fa-share fa-1x"
title="Compartir"></i>&nbsp;'.LANG['TABLA_COMPARTIR'].'</a></td><td>Privado</td>';
                }
                echo '<td><a class="popup" href="info.php?id='.$_SESSION['id_user'].'&u='.$_SESSION['id_file'].'"></a></td>';
            }
            else {
                echo '<td>sin permiso</td>';
                echo '<td></td>';
                echo '<td></td>';
                echo '<td></td>';
                echo '<td></td>';
                echo '<td>sin permiso</td>';
            }
            echo "</tr>";
        }
        if($compartido!=1)
        {
            echo '<tr>
            <th colspan="6" align="center">
            <A HREF="inicio_paginacion.php?id='.$_SESSION['id_user'].'&inicio='.$inicio.'" > <img class="rooll"

```



```

//RUTA PARA LA APK y lo JQuery
$ip = $_SERVER['SERVER_ADDR'];
$ruta_movil = "http://".$ip."/nenek/cenenek/$profiles/$user_folder/".$file;
$tipo_movil = $_FILES['upload']['type'];
$nombre_movil = $file;
$size_file = $_FILES['upload']['size'];
//termina

//se genera la ruta del archivo completa.
$target_path = $target_path . $file;
$type = pathinfo($file);
//Verifica si la extension del archivo es .exe .bin .php, si es asi el archivo no se sube y envia un
mensaje de archivo no permitido.
if($extension == "exe" || $extension == "bin" || $extension == "php" || $extension == "")
{
    echo '<script language="javascript" type="text/javascript"> alert("Tipo de archivo no
Permitido!"); </script>';
} else {
//Verifica si ya existe el archivo
if(file_exists($target_path)) {
    $ok = "<small style='font-size:100%;' title='Ya existe'> Ta kuajtits </small> $file.";
} else {
//si no existe el archivo lo mueve a la ruta de la carpeta del usuario
if(move_uploaded_file($_FILES['upload']['tmp_name'], $target_path)) {
    while(!file_exists($target_path)){ /*Hold on 'til it exists*/ }
    if(file_exists($target_path)){
        // se genera una variable a la cual se le asigna el resultado de la funcion en la cual se va
hacer el registro del archivo
        // a la tabla files.
        $id_f= save_file($target_path,$id_user,$client_ip,$padre,$ruta_movil,$tipo_movil,
$nombre_movil, $size_file);
        //Se inicio la subida del archivo
        upload_start($id_user,$id_f);
        GetSize($id_f,$target_path,1);
        //Mensaje de archivo subido con exito
        $ok = "<small style='font-size:100%;' title='El Archivo'>An t&apos;olab
</small> .basename($_FILES['upload']['name'])."<small style='font-size:100%;' title='fue subido
con &eacute;xito'> k&apos;adhij alwa&apos;</small>";
        //se le da permisos a la ruta de la carpeta del usuario
        chmod($target_path,0777);
    }
} else { $error = "<small style='font-size:100%;' title='Error al subir el archivo, intente de
nuevo!'> Yab alwa&apos;k&apos;adhij an t&apos;olab,ka taka&apos; junil</small>" ; }
}
unset($_POST);
upload_end($id_user);
}
}

```

## 1.5. Funcionamiento para la descarga de archivos

Para poder descargar archivos se necesita hacer una consulta en la que se comprueba el id del usuario así como también la id del archivo. Si estas coinciden se procede a realizar la descarga mediante el archivo download.php que contiene las funciones esenciales para poder realizar la descarga.

### ARCHIVO CONFIG.PHP

```
//Se presenta solo un fragmento del código completo del archivo ya que solo en esta parte se genera la descarga del archivo.
```

```
//Función que genera la lista de archivos.
```

```
/*FUNCIÓN QUE MUESTRA LA LISTA DE ARCHIVOS*/
```

```
function
```

```
list_files($id_user,$mostar_list,$file_search,$extension_search,$inicio,$redundancia,$compartido,$id_grupo,$status,$dueno,$opcion,$padre,$lang) {
```

```
    //Se incluye el archivo de conexion con la bd
    include 'dbconnect.php';
```

```
//Se generan las consultas para extraer los archivos
```

```
if($padre!="") {
```

```
    $query = "SELECT * FROM files WHERE id_user = '$id_user' and
```

```
padre='$padre' order by type, id_file desc limit 15 offset '$inicio';
```

```
    } else if($compartido==1) {
```

```
        $d=1;
```

```
        $query = "select *from files, file_grupo where
```

```
file_grupo.id_grupo='$id_grupo' and file_grupo.id_file=files.id_file order by files.type,
```

```
files.id_file;";
```

```
    } else {
```

```
        $query = "SELECT * FROM files WHERE id_user = '$id_user' and padre='/'
```

```
order by type, id_file desc limit 15 offset '$inicio';
```

```
    }
```

```
//parte del codigo completo de la tabla donde se genera la lista de archivos, en esta parte es donde se genera el enlace de descarga.
```

```
else {
```

```
    echo '<td><a
```

```
href="download.php?id='.$id_file.'&id_user='.$id_user.'&redundancy='.$redundancia.'&d=p
```

```
erfil"><i onclick="refreshIt()" class="fa fa-download fa-1x"
```

```
title="Descargar"></i>&nbsp;'.$lang['TABLA_DESCARGAR'].'</a></td>';
```

```
}
```

### ARCHIVO DOWNLOAD.PHP

```
//Se crean las variables que se obtienen por el metodo GET.
```

```
$grupo = $_GET['d'];
```

```
$id_user=$_GET['id_user'];
```

```

Sid_file = $_GET['id'];

//Se genera la consulta a la tabla files apartir de la variable id_file obtenido por el metodo
GET.
$query = "SELECT * FROM files WHERE id_file = '$id_file' LIMIT 1;";
$result = pg_query($conn,$query);
//si no se obtienen resultados de la consulta que marco fallo de consulta.
if(!$result){
echo "Fallo query: $query<br/>";
}
else {
//si se generan resultados se crea la variable row asociando los resultados de la consulta.
$row = pg_fetch_assoc($result);
$ip = $row['ip'];
$filename = $row['file_name'];
$filename = utf8_decode($filename);
$file = $filename;
//Inicia la descarga mediante la funcion down_start
down_start($id_user,$id_file,$filename);
GetSize_down($id_file,$file);

if(file_exists($filename)){
    $f = $filename;
    header('Content-Type: application/force-download');
    header('Content-Disposition: attachment; filename="'.basename($f).'"');
    header("Content-type: application/octet-stream");
    $fp = fopen("$f", "r");
    fpassthru($fp);
down_end($id_user);

//Envio notificacion al email de todos los q pertenecen al grupo
$queryg="select * from file_grupo where id_file='$id_file'";
$consultag=pg_query($conn,$queryg);
$columnag=pg_fetch_assoc($consultag);
$id_grupo=$columnag['id_grupo'];
$querym="select * from user_grupo where id_user='$id_grupo'";
$consultam=pg_query($conn,$querym);
while ($columnam=pg_fetch_assoc($consultam))
{
$senduser=$columnam['id_user'];

$query="select * from users where id_user='$senduser'";
$consulta=pg_query($conn,$query);
$columna=pg_fetch_assoc($consulta);
$email=$columna['email'];
$destinatario= $email;
$asunto="Eliminaron un archivo";
$link= "http://www.adaptivez.org.mx/TreeStore";
$cuero= "El usuario $dueno dejo de compartir $filename del grupo $grupo<br/>".$link;
$headers = "MIME-Version: 1.0\r\n";
$headers .= "Content-type: text/html; charset=iso-8859-1\r\n";

```

```

mail($destinatario,$asunto,$cuerpo,$headers);
$query = "insert into email values ('$destinatario','$id_user','$asunto');";
$result = pg_query($conn,$query);

}

header("Refresh: 0; url='./archivos_compartidos.php?g=$grupo', TRUE, 302");

if ($_GET['d']=="perfil") {
    header("Refresh: 0; url='./profile.php', TRUE, 302");
}

}

}

```

## 1.6. Funcionamiento para compartir archivos

Para compartir archivos primeramente se selecciona el archivo de la lista donde se encuentra, y como siguiente paso se de la al botón compartir, donde mandará a llamar las siguientes funciones para poder compartir el archivo.

### ARCHIVO CONFIG.PHP

//se presenta un fragmento del código que es el que contiene la función para compartir los archivos.

```

if($opcion ==1) {
    if($public == 1){
        echo '<td><a class="popup"
href="compartir.php?id='.$id_user.'&f='.$id_file.'&fn='.$fn.'" > <i class="fa fa-share fa-1x"
title="Compartir"></i>&nbsp;'. $lang['TABLA_COMPARTIR'].'</a></td><td>P&uacute;blico
</td>';
    } else {
        echo '<td><a class="popup"
href="compartir.php?id='.$id_user.'&f='.$id_file.'&fn='.$fn.'" title="Compartir"><i class="fa
fa-share fa-1x"
title="Compartir"></i>&nbsp;'. $lang['TABLA_COMPARTIR'].'</a></td><td>Privado</td>';
    }
    echo '<td><a class="popup" href="info.php?id='.$id_user.'&u='.$id_file.'"></a></td>';
}
}

```

### ARCHIVO COMPARTIR.PHP

//En este archivo se presenta la función que hace que comparta el archivo con un grupo de trabajo.

```
<?php
```

```

//se incluye la conexión a la base de datos
include 'dbconnect.php';
//Se genera la consulta a tabla grupos
$query = "select * from grupos, user_grupo where grupos.id_grupo=user_grupo.id_grupo
and user_grupo.id_user='$id_user' and user_grupo.autorizado='si' ";

$result = pg_query($conn,$query);
//si no existe ningún resultado que envíe un error de consulta
if(!$result) {
    echo "Fallo query: $query<hr/>";
} else {
    //verifica si eres miembro de un grupo
    if(pg_num_rows($result) == 0) {
        echo "Aun no es miembro de ningun grupo de trabajo";
    }
    //genera una tabla con la lista de los grupos a los que perteneces y a los cuales puedes
    compartir el archivo seleccionado
    else {
        echo '<table class="table table-bordered"><tr><th>Grupo
        &nbsp;&nbsp;&nbsp;</th><th>Compartir &nbsp;&nbsp;&nbsp;</th><th>Dueño</th></tr>';
        while($row = pg_fetch_assoc($result)) {
            $grupo = $row['nombre'];
            $id_grupo = $row['id_grupo'];
            $dueno = $row['dueno'];

            echo '<tr><td>'.$grupo.'</td>';
            echo '<td><a
            href="compartir_grupo.php?id='.$id_user.'&g='.$id_grupo.'&file='.$id_file.'&fn='.$file.'&gr='.$
            $grupo.'"></a></td>';
            echo '<td>'.$dueno.'</td>';
        }
    }
    echo '<table/>';
?>

```

## 1.7. Funcionamiento para buscar archivos

Para generar la búsqueda de archivos se utilizan funciones con JavaScript, HTML y PHP.

### ARCHIVO PROFILE.PHP

```

//Fragmento de código el cual genera la búsqueda de los archivos en el archivo profile.php
//Lista en la cual está el enlace el cual manda a llamar una función javascript la cual
muestra un formulario donde se genera la consulta del archivo.
<ul class="nav nav-pills">
    <li><A HREF=""<?php echo

```

```

$_SERVER['PHP_SELF']. "?id=$id_user&mis_archivos=1";?>" title='Mis Archivos'<i
class="fa fa-archive fa-1x">&nbsp;</i><?php echo
$lang['CUERPO_MIS_ARCHIVOS'];?></A></li>
<li><A HREF="<?php echo $_SERVER['PHP_SELF']. "?id=$id_user&grupos=1";?>"
title='Mis Grupos de trabajo'><i class="fa fa-users fa-1x">&nbsp;</i><?php echo
$lang['CUERPO_MIS_GRUPOS_TRABAJO'];?></A></li>
<?php
if($grupos!=1)
{
?>
<li><a style='cursor: pointer;' onclick="muestra_oculta('contenido_a_mostrar)" title="
Buscar Archivos"><i class="fa fa-search fa-1x">&nbsp;</i><?php echo
$lang['CUERPO_BUSCAR_ARCHIVOS'];?></a></li>
<li><a style='cursor: pointer;' onclick="muestra_oculta('contenido_a_mostrar2)"
title="Nueva Carpeta"><i class="fa fa-folder fa-1x">&nbsp;</i><?php echo
$lang['CUERPO_NUEVA_CARPETA'];?></a></li>
</ul>

//Formulario en el cual se genera la consulta del archivo a buscar
<!--Buscar Archivo-->
<div id="contenido_a_mostrar">
<form class="well" action="<?php echo
$_SERVER['PHP_SELF']. "?id=$id_user&padre=$padre";?>&new=1" method="POST"
name="search" id="search">
<fieldset>
<legend title='Buscar'><?php echo $lang['CUERPO_BUSCAR_FILE'];?></legend>
<div class="d-form">
<label for="file" title='Archivo'> <?php echo $lang['CUERPO_ARCHIVO'];?>:</label>
</div>
<div class="d-form">
<input type="text" id="file" name="file" size="50px"/>
</div>
<div class="d-form">
<label for="extension" title='Extension'>Extension:</label>
</div>
<div class="d-form">
<input type="text" id="extension" name="extension" title="Ej: doc, xls, jpg, png"/>
</div>
<div class="d-form">
<button id="buscar" name="buscar" type="submit" value="Buscar" class="form
awesome btn btn-primary btn-small" title='Buscar'><?php echo
$lang['CUERPO_BUSCAR_FILE'];?></button>
</div>
</fieldset>
</form>
<hr/>
</div>

//Fragmento de código php para realizar la búsqueda de los datos que se envían del
formulario, en el cual se genera una función en donde se envían los datos del archivo que
se va a buscar.

```

```

$query_user = "SELECT * FROM files WHERE file_name = '$padre' LIMIT 1;";
$result_user = pg_query($conn,$query_user) or die("Fallo query: $query.<br/>");
$row = pg_fetch_assoc($result_user);
$atrasp = $row['padre'];

echo '<h3 title="Mis Archivos -
Carpeta:">'. $lang['TABLA_MIS_ARCHIVOS_CARPETA'].':&nbsp;'. $padre.'</h3>';
echo '<a href="profile.php?id='.$id_user.'&padre='.$atrasp.'"><i class="fa fa-reply fa-
1x">&nbsp;</i>'. $lang['SOLICITUDES_ATRAS'].</a><br><hr>';

$opcion=1;

list_files($id_user,$mostar_list,$file_search,$extension_search,$inicio,$redundancia,$com
partido,$id_grupo,$status,$dueno,$opcion,$padre,$lang);

```

## 1.8. Funcionamiento para mostrar el listado de mensajes

Para realizar un listado de los mensajes de cada usuario se hace una consulta a la base de datos a la tabla mail, donde se verifica su id\_user y se extraen los campos de la tabla donde se encuentran los mensajes del usuario.

### ARCHIVO MAIL.PHP

//Fragmento del código PHP que genera la consulta en la base de datos donde extrae los mensajes de cada usuario y los muestra en una tabla.

```

$query = "select * from mail where recibe='$id_user' and (dueno='$id_user' and
status!='2') order by fecha desc, hora desc limit 15 offset '$inicio';";
$result = pg_query($conn,$query);

```

```

if(!$result){ echo "Fallo query: $query<hr/>"; }
else {
    if(pg_num_rows($result) == 0){
        echo '<p title="No hay
mensajes">'. $lang['CUERPO_TABLA_NO_MENSAJES'].</p>';
    }
    else {
        echo "<table class='table table-bordered'>
<thead>
<tr>
<th>&nbsp;</th>
<th title='Enviado por'>". $lang['CUERPO_TABLA_ENVIADO']. "</th>
<th title='Asunto'>". $lang['CUERPO_TABLA_ASUNTO']. "</th>
<th title='Ir'>". $lang['CUERPO_TABLA_IR']. "</th>
<th title='Borrar'>". $lang['CUERPO_TABLA_BORRAR']. "</th>
<th title='Fecha'>". $lang['CUERPO_TABLA_FECHA']. "</th>

```

```

    <th title='Hora'>".$lang['CUERPO_TABLA_HORA'].</th>
</tr></thead>";

while($row = pg_fetch_assoc($result)) {
    $id_msg = $row['id_msg'];
    $envia = $row['envia'];
    $recibe = $row['recibe'];
    $asunto = $row['asunto'];
    $status = $row['status'];
    $cuerpo = $row['cuerpo'];
    $fecha = $row['fecha'];
    $Hora = $row['hora'];

if($status=='0') {
    echo '<tr><td></td>';
    echo '<td><b>'.$envia.</td>';
    echo '<td ><b>'.$asunto.</td>';
    echo '<td><a
href="mail.php?id_user='.$id_user.'&id_msg='.$id_msg.'&msg=1&type=1&mostrar=6";?
title="ir">
    <i onclick="refreshIt()" class="fa fa-arrow-right fa-1x"
title="K&apos;alej"></i>&nbsp;'.$lang['CUERPO_TABLA_IR'].'</a></td>';

/*Mensaje no visto eliminar*/
    echo '<td><a
href="eliminar_msg.php?id_user='.$id_user.'&id_msg='.$id_msg.'&msg=1&type=1&mostra
r='.$mostrar.'";? title="Borrar">
<i class="fa fa-times fa-1x"
title="Pakuw"></i>&nbsp;'.$lang['CUERPO_TABLA_BORRAR'].'</a></td>';
    echo '<td><b>'.$fecha.</td>';
    echo '<td ><b>'.$Hora.</td>';
}
else
    {
    echo '<tr><td></td>';
    echo '<td>'.$envia.</td>';
    echo '<td >'.$asunto.</td>';
    echo '<td><a
href="mail.php?id_user='.$id_user.'&id_msg='.$id_msg.'&msg=1&type=1&mostrar=6";?
title="ir">
    <i onclick="refreshIt()" class="fa fa-arrow-right fa-1x"
title="K&apos;alej"></i>&nbsp;'.$lang['CUERPO_TABLA_IR'].'</a></td>';

/*Mensaje visto eliminar*/
    echo '<td><a
href="eliminar_msg.php?id_user='.$id_user.'&id_msg='.$id_msg.'&msg=1&type=1&mostra
r='.$mostrar.'";? title="Borrar">
<i class="fa fa-times fa-1x"
title="Pakuw"></i>&nbsp;'.$lang['CUERPO_TABLA_BORRAR'].'</a></td>';
    echo '<td>'.$fecha.</td>';
}
}

```

```

        echo '<td >'.$Hora.'</td>';
    }
}

    echo '<tr><th colspan="7" align=right align="center">
<A
HREF="inicio_paginacion.php?id='.$id_user.'&inicio='.$inicio.'&pag_mail='.$pag_mail.'&mostrar='.$mostrar.'"> </A>
<A
HREF="atras_paginacion.php?id='.$id_user.'&inicio='.$inicio.'&pag_mail='.$pag_mail.'&mostrar='.$mostrar.'"> </A>
<A
HREF="adelante_paginacion.php?id='.$id_user.'&inicio='.$inicio.'&pag_mail='.$pag_mail.'&mostrar='.$mostrar.'"> </A>
<A
HREF="fin_paginacion.php?id='.$id_user.'&inicio='.$inicio.'&pag_mail='.$pag_mail.'&mostrar='.$mostrar.'"> </A>
</th>
</tr>';
echo "</tr>";

}
}
echo '</table>';
}

```

## 1.9. Funcionamiento para crear grupos de trabajo

El funcionamiento para crear grupos de trabajo se basa en que cada usuario registrado pueda crear un grupo o tambien agregarse a otro grupo creado por otra persona registrada en la plataforma Nenek-SAAC, y asi poder compartir archivos.

### ARCHIVO WORKING\_GROUP.PHP

//En este fragment de codigo se genera una lista de opciones entre la cual esta la de crear nuevo grupo.

```

<ul class="nav nav-pills">
    <li><A HREF="<?php echo
$_SERVER['PHP_SELF']. "?id_user=$id_user&mostrar=1";?>" title="Nuevo Grupo"><i
class="fa fa-plus-square fa-1x" title="Nuevo Grupo"></i>&nbsp;<?php echo
$lang['GRUPOS_TRABAJO_NUEVO_GRUPO'];?></A></li>

    <li><A HREF="<?php echo

```

```

$_SERVER['PHP_SELF']. "?id_user=$id_user&mostrar=2";?>" title="Agregar usuarios a
mis grupos"><i class="fa fa-plus fa-1x" title="Nuevo Grupo"></i>&nbsp;<i class="fa fa-user
fa-1x" title="Nuevo Grupo"></i>&nbsp;<?php echo
$lang['GRUPOS_ADD_USUARIOS_GRUPO'];?></A></li>

<li><A HREF="<?php echo
$_SERVER['PHP_SELF']. "?id_user=$id_user&mostrar=3";?>" title="Agregar a otros
grupos"><i class="fa fa-plus-square fa-1x" title="Nuevo Grupo"></i>&nbsp;<?php echo
$lang['GRUPOS_ADD_OTROS_GRUPOS'];?></A></li>

<li><A HREF="<?php echo
$_SERVER['PHP_SELF']. "?id_user=$id_user&mostrar=4";?>" title="Mis grupos de
trabajo"><i class="fa fa-group fa-1x">&nbsp;</i><?php echo
$lang['GRUPOS_MIS_GRUPOS_TRABAJO'];?></A></li>
<!--<A HREF="<?php echo
$_SERVER['PHP_SELF']. "?id_user=$id_user&mostrar=5";?>">--Add me to Organization--
</A-->
</ul>

//Formulario para crear un grupo de trabajo

<!--Form para crear un grupo nuevo-->
<form class="form-inline" action="new_grupo.php" method="POST" name="login"
id="login" enctype="multipart/form-data">
<small style="font-size:100%;" title="Nuevo grupo de trabajo "><?php echo
$lang['GRUPOS_NUEVO_GRUPO_TRABAJO']?></small> <br /><br />
<fieldset class="well form-m">
<?php
    if ($_GET['existe']=='si') {
        echo "<small style='font-size:18px' title='Un grupo ya existe con ese
nombre'>". $lang['GRUPOS_GRUPO_EXISTENTE'] . "</small>" ;echo '<br /> <br />';
    }
    if ($_GET['existe']=='no') {
        echo "<small style='font-size:18px' title='Grupo
creado'>". $lang['GRUPOS_GRUPO_CREADO'] . "</small>";echo '<br /><br />';
    }
?>
<br>
<div class="d-form">
<small style="font-size:20px" title="Nombre"><?php echo
$lang['CUERPO_NOMBRE_CARPETA']?></small>
</div>
<div class="d-form">
    <input type="text" id="Enviar" name="grupo" size="50px"/>
</div>
<div class="d-form">
    <button class="btn btn-primary" id="Enviar" name="Enviar" type="submit"
value="Enviar" title="Enviar"><?php echo
$lang['CUERPO_MENSAJES_ENVIAR']?></button>
</div>
</fieldset>

```

```

<hr />
</form>

//ARCHIVO NEW_GRUPO.PHP

<?php
include('dbconnect.php');
include ('sesion.php');

// Se crea la variable $id_user a partir de la variable de sesion
$id_user = $_SESSION['usuario'];

// $id_user = $_GET['id_user'];
//$id_user = utf8_encode($id_user);
$grupo = $_POST['grupo'];
//$password = md5($password);
//echo $id_user, $grupo;exit;
$id_grupo=uniqid();

$query = "SELECT * FROM grupos WHERE nombre='$grupo' and dueno='$id_user'";
$result = pg_query($conn,$query) or die("Fallo query: $query.<br/>");

if(pg_num_rows($result)<=0){

    $existe="no";
    $fecha = date("Y/m/d");
    //Se insertan datos en la tabla grupos
    $query = "INSERT INTO grupos
VALUES('$id_grupo','$grupo','$id_user','yes','$fecha)";

    $result = pg_query($conn,$query);
    if(!$result){
        echo "Fallo query:$query<br/>";exit;
    }

    $query = "SELECT * FROM user_grupo";
    $result = pg_query($conn,$query) or die("Fallo 4query: $query.<br/>");
    $num=pg_num_rows($result)+1;
    //Se insertan datos en la tabla user_grupo
    $query = "INSERT INTO user_grupo
VALUES('$id_grupo','$id_user','si','$num','$fecha)";
    $result = pg_query($conn,$query);

    if(!$result){
        echo "Fallo query:$query<br/>";
        exit;
    }
}
else {
    $existe="si";
}
}

```

```
    echo
"<script>document.location.href='working_group.php?existe=$existe&mostrar=1';</script>\n";
?>
```

## 2.0. Funcionamiento para editar el perfil de usuario

El usuario podrá cambiar sus datos personales como la contraseña, email, nombre e imagen de perfil.

### ARCHIVO EDIT\_PROFILE.PHP

```
//formulario para cambiar la contraseña
<form class="well" action="<?php echo $_SERVER['PHP_SELF'].'?id=$id_user';?>"
method="POST" name="pass">
    <p title='Cambiar Contraseña'><?php echo
$lang['EDITAR_CAMBIAR_CONTRASENA'];?></p>
    <div>
        <div class="d-form">
            <label title='Actual Contraseña'><?php echo
$lang['EDITAR_ACTUAL_CONTRASENA'];?></label>
        </div>
        <div class="d-form">
            <input type="password" id="password" name="password"/>
        </div>
    </div>
</div>
<div>
    <div class="d-form">
        <label title='Nueva contraseña'><?php echo
$lang['EDITAR_NUEVA_CONTRASENA'];?></label>
    </div>
    <div class="d-form">
        <input type="password" id="password" name="npassword"/>
    </div>
</div>
    <div class="d-form">
        <button name="Apply" type="submit" value="Apply" title='Guardar' class="btn
btn-primary"><?php echo $lang['CUERPO_BTN_CARPETA'];?></button>&nbsp;
    </div>
<a href="edit_profile.php?id=<?php echo $id_user;?>" title="Atr&aacute;s">&nbsp;<?php echo
$lang['SOLICITUDES_ATRAS'];?></a></p>
</form>

//funcion php donde comprueba que se envia algo del formulario y despues se haga una
consulta a la base de datos donde se actualiza la nueva contraseña.

if(isset($_POST['password']) && isset($_POST['npassword']))
```

```

{
    $password1 = $_POST['password'];
    $password2 = $_POST['npassword'];
    if($password1==$pass)
    {
    $query = "update users set password='$password2' where password='$password1' and
id_user='$id_user'";
    $result = pg_query($conn,$query);
    echo '<script language="javascript" type="text/javascript">
alert("Itmejits a Tsinat japixtalabil");
</script>';
    } else
    {
    echo '<script language="javascript" type="text/javascript">
alert(" A tsinat japixtalabil yab jununul");
</script>';

    }
}

//Formulario donde se cambia el nombre del usuario

<form class="well" action="<?php echo $_SERVER['PHP_SELF']. "?id=$id_user";?>"
method="POST" name="name">
    <p>ka jalk'uy a Bij</p>
<div>
    <div class="d-form">
        <label title='Nombre'><?php echo
$lang['CUERPO_NOMBRE_CARPETA'];?></label>
    </div>
    <div class="d-form">
        <input value="<?php echo $name;?>" readonly="readonly" id="name"
name="name"/>
    </div>
</div>
<div>
    <div class="d-form">
        <label title='Nuevo Nombre'><?php echo
$lang['EDITAR_NUEVO_NOMBRE'];?></label>
    </div>
    <div class="d-form">
        <input id="nname" name="nname"/>
    </div>
</div>
    <button name="Apply" type="submit" value="Apply" title='Guardar' class="btn btn-
primary"><?php echo $lang['CUERPO_BTN_CARPETA'];?></button>
    <a href="edit_profile.php?&id=<?php echo $id_user;?>" title="Atr&aacute;s">&nbsp;<?php echo
$lang['SOLICITUDES_ATRAS'];?></a></p>
</form>

```

```

//Funcion php que actualiza el nombre del usuario

if(isset($_POST['nname']))
{
    $nname = $_POST['nname'];
    $query = "update users set name='$nname' where id_user='$id_user';";
    $result = pg_query($conn,$query);
    echo '<script language="javascript" type="text/javascript">
        alert("¡tmejits a Bij!");
    </script>';
}

//Formulario para cambiar el email del usuario.

<form class="well" action="<?php echo $_SERVER['PHP_SELF']. "?id=$id_user";?>"
method="POST" name="email">
    <p title='Cambiar Email'><?php echo $lang['EDITAR_CAMBIAR_EMAIL'];?></p>
<div>
    <div class="d-form">
        <label title='Email'><?php echo $lang['LOGIN_EMAIL'];?></label>
    </div>
    <div class="d-form">
        <input value="<?php echo $email;?>" readonly="readonly" id="email"
name="email"/>
    </div>
</div>

<div>
    <div class="d-form">
        <label title='Nuevo Correo'><?php echo
$lang['EDITAR_NUEVO_EMAIL'];?></label>
    </div>
    <div class="d-form">
        <input id="nemail" name="nemail"/>
    </div>
</div>
    <button class="btn btn-primary" name="Apply" type="submit" value="Apply"
title='Guardar'><?php echo $lang['CUERPO_BTN_CARPETA'];?></button>
    <a href="edit_profile.php?&id=<?php echo $id_user;?>" title='Atr&aacute;s'>&nbsp;<?php echo
$lang['SOLICITUDES_ATRAS'];?></a></p>
</form>

//Funcion php que actualiza el email del usuario.

if(isset($_POST['nemail']))
{
    $nemail = $_POST['nemail'];
    $query = "update users set email='$nemail' where id_user='$id_user';";
    $result = pg_query($conn,$query);
}

```

```

echo '<script language="javascript" type="text/javascript">
      alert(" Itmejits a Bukudh kaw");
    </script>';
$activacion=uniqid();
$query = "update users set sesion='si' where id_user='$id_user'";
$result = pg_query($conn,$query);

$destinatario= $nemail;
$sasunto= "Bienvenido a TreeStore";
$link= "http://www.adaptivez.org.mx/TreeStore";
$cuerpo= "Bienvenido a TreeStore, tu almacenamiento de archivos en la nube.";
$headers = "MIME-Version: 1.0\r\n";
$headers .= "Content-type: text/html; charset=iso-8859-1\r\n";
$headers .= "From: TreeStore <TreeStore@adaptivez.org.mx >\r\n";
mail($destinatario,$sasunto,$cuerpo,$headers);
echo "<script> document.location.href='profile.php?id=$id_user'; </script>\n";
}

//Formulario donde se cambia la imagen del usuario

<form action="<?php echo $_SERVER['PHP_SELF']. "?id=$id_user";?>" method="POST"
name="login" id="login" enctype="multipart/form-data">
<fieldset>
<legend title='Subir Archivo'><?php echo
$lang['EDITAR_SUBIR_ARCHIVO_TITLE'];?></legend>
  <input type="hidden" name="MAX_FILE_SIZE" value=" <?php echo
$value_size;?>" /><!--5MB-->
  <div class="d-form">
    <label><?php echo $lang['CUERPO_ARCHIVO'];?></label>
  </div>
  <div class="d-form">
    <input type="file" name="upload" />
  </div>
  <div class="d-form">
    <button id="subir" name="subir" type="submit" value="Subir" class="form awesome
btn btn-primary" title='Subir'><?php echo $lang['EDITAR_SUBIR_ARCHIVO'];?></button>
  </div>
</fieldset>
</form>

//Funcion php que permite actualizar la imagen del perfil.

if(isset($_POST['subir'])){
  $ruta = ".$profiles/$user_folder/";
  $full_file = $_FILES['upload']['name'];

  $tipo = $_FILES['upload']['type'];
  $file = basename($full_file);
  $target_path = $ruta . $file;

  $type = pathinfo($file);

```

```

    $extension = $type['extension'];

    if($extension == "exe" || $extension == "bin"){
        $error = "<small style='font-size:100%;' title='Tipo de archivo no permitido'>Axe xi
t&apos;olab yab a wit&apos;al ka dhya&apos;! </small>";
    }
    else {
        if(move_uploaded_file($_FILES['upload']['tmp_name'], $target_path)) {
            while(!file_exists($target_path)){ /*Hold on 'til it exists*/ }
            if(file_exists($target_path)){

                $query = "update users set avatar='$target_path' where
id_user='$id_user'";
                $result = pg_query($conn,$query);

                $ok = "<small style='font-size:100%;' title='El archivo'>An
t&apos;olab </small> ".basename($_FILES['upload']['name'])."<small style='font-
size:100%;' title='fue subido con &eacute;xito'> k&apos;adhij alwa&apos;</small>";
                chmod($target_path,0777);
                header('Location: edit_profile.php');
            }
        } else {
            $error = "<small style='font-size:100%;' title='Error al subir el archivo, intente de nuevo!'>
Yab alwa&apos;k&apos;adhij an t&apos;olab,ka taka&apos; junil</small>";
        }
    }
}

```

## 1. Funcionamiento de la aplicación nativa de Android

Con todo ahora funcionando se explicará ahora el funcionamiento de la aplicación de nativa de Android la cual se maneja en Java.

### 1.1. Funcionamiento del sistema de login

El login consta de la obtención de datos del usuario, para posteriormente validarlo y por ultimo permitirle el acceso al usuario. El siguiente código muestra la lógica para el login:

```

//Tarea ejecutada por el login
private class MyAsyncTask extends AsyncTask<String, Integer, String>{

private ProgressDialog progressDialog = new ProgressDialog(MainActivity.this);
    InputStream is = null ;
    String result = "";
    protected void onPreExecute() {
        progressDialog.setMessage(getString(R.string.Validando));
        progressDialog.show();
        progressDialog.setOnCancelListener(new OnCancelListener() {
            @Override

```

```

        public void onCancel(DialogInterface arg0) {
            MyAsyncTask.this.cancel(true);
        }
    });
}
@Override
protected String doInBackground(String... params) {
    // TODO Auto-generated method stub
    System.out.println("MyAsyncTask: Iniciando Login");
    result = postData(params[0]);
    return result;
}

protected void onPostExecute(String result){
    this.progressDialog.dismiss();
    id_user = result.trim();
    System.out.println("MyAsyncTask: Recibiendo");
    //A validar la respuesta del servidor
    if(result.equals("ErrorInternet")){
        Toast.makeText(getApplicationContext(), getString(R.string.InternetError),
            Toast.LENGTH_LONG).show();

        value2.setText("");
        return;
    }
    if(id_user.equals("invalido")){
        Toast.makeText(getApplicationContext(), getString(R.string.VerificaDatos),
            Toast.LENGTH_LONG).show();

        value.setText("");
        value2.setText("");
        return;
    }else{
        //iniciando la obtencion de el usuario
        System.out.println("MyAsyncTask: Iniciando UserTask");
        new UserTask().execute(userUrl, id_user);
    }
}

public String postData(String valueIWantToSend) {
    System.out.println("MyAsyncTask: Aqui");
    HttpParams httpParameters = new BasicHttpParams();
    HttpProtocolParams.setContentCharset(httpParameters, HTTP.UTF_8);
    HttpProtocolParams.setHttpElementCharset(httpParameters, HTTP.UTF_8);
    // Create a new HttpClient and Post Header
    HttpClient httpclient = new DefaultHttpClient();
    HttpPost httppost = new HttpPost(getString(R.string.ServicioURL)
        +getString(R.string.LoginService));

    System.out.println("MyAsyncTask: Pasando url");
    try {
        // Add your data
        List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();
        nameValuePairs.add(new BasicNameValuePair("myHttpData", nombre));
        nameValuePairs.add(new BasicNameValuePair("password", pass));
        httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs, HTTP.UTF_8));
        System.out.println("MyAsyncTask: Ejecutando envio de entidades");
        // Execute HTTP Post Request
        HttpResponse response = httpclient.execute(httppost);
    }
}

```



archivo

```
try {
    JSONObject c = json.getJSONObject(i);
    String varchivo = c.getString(TAG_FILE);
    String vtamano = c.getString(TAG_SIZE);
    String vtipo = c.getString(TAG_TYPE);
    String vfecha = c.getString(TAG_DATE);
    String vhora = c.getString(TAG_TIME);
    String vurl = c.getString(TAG_URL);
    HashMap<String, String> map = new HashMap<String, String>();

    int in = new Integer(vtamano);
    in= (in/1024);
    vtamano = String.valueOf(in);

    // adding each child node to HashMap key => value

    map.put(TAG_FILE, varchivo);
    map.put(TAG_SIZE, vtamano+" Kb");
    map.put(TAG_TYPE, vtipo);
    map.put(TAG_DATE, vfecha+" ");
    map.put(TAG_TIME, vhora);

    //obteniendo variables para trabajar

    map.put("name", varchivo); //obteniendo directo el nombre del

    map.put("tamano", vtamano); //tamaño
    map.put("tipo", vtipo); //tipo
    map.put("fechas", vfecha);
    map.put("hora", vhora);
    map.put("url", vurl);

    jsonlist.add(map);

}catch (JSONException e) {
    // TODO Auto-generated catch block
    Toast.makeText(getApplicationContext(),
getString(R.string.ErrorArchivos), Toast.LENGTH_SHORT).show();
    e.printStackTrace();
} //fin catch
} //fin for

adapter = new SpecialAdapter(context, jsonlist,
R.layout.archivos_item, new String[] { TAG_FILE, TAG_SIZE,
TAG_DATE, TAG_TIME }, new int[] {
R.id.vehicleType, R.id.vehicleColor,
R.id.treadType, R.id.horaType});

// We need notify the adapter that the data have been changed
setListAdapter(adapter);
((BaseAdapter)adapter).notifyDataSetChanged();
System.out.println("Progres task: Adaptador colocado");
super.onPostExecute(json);
// selecting single ListView item
lv = getListView();
```

```

lv.setTextFilterEnabled(true);
lv.setOnItemClickListener(new OnItemClickListener(){

@Override
public void onItemClick(AdapterView<?> parent,View v,int posicion, long
id) {

// TODO Auto-generated method stub
@SuppressWarnings("unchecked")
HashMap<String, String> o = (HashMap<String, String>)
lv.getItemAtPosition(posicion);

//Obteniendo variables
name_archivo = o.get("name");
size = o.get("tamano");
types = o.get("type");
date = o.get("fechas");
time = o.get("hora");
urlfile = o.get("url");

//Nueva actividad
Intent i = new Intent(v.getContext(), ArchivosElegido.class); //add activity
Bundle b = new Bundle();
i.putExtra("file", name_archivo); //archivo
i.putExtra("id_user", id); //usuario
i.putExtra("tam", size); //tamaño
i.putExtra("tipo", types); //tipo
i.putExtra("fech", date); //fecha
i.putExtra("horas", time);
i.putExtra("urlfile", urlfile);
v.getContext().startActivity(i);

}

});

}else{
Toast.makeText(getApplicationContext(),
getString(R.string.ErrorArchivos)+" 2", Toast.LENGTH_SHORT).show();
}
pullToRefreshView.onRefreshComplete();

}

protected JSONArray doInBackground(final String... args) {
if (isCancelled()) {
return null;
}
JSONParser jParser = new JSONParser();

// getting JSON string from URL
JSONArray json = jParser.getJSONFromUrl2(url, id_user,"1");
System.out.println("ProgresTask: Json pasandose");
return json;

}

```

```
}//fin Progresstask
```

## 1.3. Funcionamiento de los grupos de trabajo

Los grupos de trabajo se dividen en 3 secciones: mis grupos, archivos en los grupos, trabajo colaborativo.

### 1.3.1. Mis grupos de trabajo

En esta sección se detalla la lógica de los grupos que contiene un usuario, para agregar usuarios y crear nuevos grupos.

```
private class GruposTask extends AsyncTask<String, Void, JSONArray> {

    private ProgressDialog dialog;

    private ListActivity activity;

    // private List<Message> messages;
    public GruposTask(ListActivity activity) {
        this.activity = activity;
        context = activity;
        dialog = new ProgressDialog(context);
    }

    /** application context. */
    private Context context;

    protected void onPreExecute() {
        agregarAlGrupo.setVisibility(View.GONE);
        progres.setVisibility(View.VISIBLE);
    }

    @Override
    protected void onPostExecute(JSONArray json) {
        progres.setVisibility(View.GONE);
        if(json != null){
            grupos.clear();

            for (int i = 0; i < json.length(); i++) {
                try {
                    JSONObject c = json.getJSONObject(i);
                    String vnombre_grupo = c.getString(TAG_NOMBRE_GRUPO);

                    grupos.add(vnombre_grupo);

                }catch (JSONException e) {
                    // TODO Auto-generated catch block
                    Toast.makeText(getApplicationContext(),
                        getString(R.string.ErrorGrupos), Toast.LENGTH_SHORT).show();
                    e.printStackTrace();
                }
            }
        }
    }
}
```

```

        } //fin catch
    } //fin for

    adapter = new ArrayAdapter<String>( context,
    android.R.layout.simple_spinner_dropdown_item, grupos );

    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

    // We need notify the adapter that the data have been changed
    SpinnerGrupo.setAdapter( adapter );
    autoCompleteGrupo.setAdapter(adapter);
    adapter.notifyDataSetChanged();
    System.out.println("GruposTask: Lista hecha");
    super.onPostExecute(json);

} else{
    grupos.add("No tienes un grupo de trabajo");
}

}
protected JSONArray doInBackground(final String... args) {
    if (isCancelled()) {
        return null;
    }
    JSONParser jParser = new JSONParser();

    // getting JSON string from URL
    JSONArray json = jParser.getJSONFromUrl(url, id_user);
    System.out.println("GruposTask: Json pasandose");
    return json;
}
} //Fin GruposTask

```

### 1.3.2. Archivos en los grupos

Este apartado detalla una lista con los archivos que se encuentran en cada grupo de trabajo en el que un usuario se encuentre autorizado.

```

// We need notify the adapter that the data have been changed
archivosGrupoElegidoAdapter = new SpecialAdapter(context, archivosGrupoElegido,
    R.layout.archivos_item, new String[] { TAG_FILE, TAG_ID,
    TAG_DATE, TAG_TIME }, new int[] {
    R.id.vehicleType, R.id.vehicleColor,
    R.id.treadType, R.id.horaType});

// We need notify the adapter that the data have been changed
setListAdapter(archivosGrupoElegidoAdapter);
((BaseAdapter)archivosGrupoElegidoAdapter).notifyDataSetChanged();
System.out.println("GroupFileTask: Adaptador colocado");
super.onPostExecute(jsonUs);
// selecting single ListView item
lv = getListView();
lv.setTextFilterEnabled(true);

```

```

lv.setOnItemClickListener(new OnItemClickListener(){

    @Override
    public void onItemClick(AdapterView<?> parent,View v,int posicion, long id) {
        // TODO Auto-generated method stub
        @SuppressWarnings("unchecked")
        HashMap<String, String> o = (HashMap<String, String>) lv.getItemAtPosition(posicion);

        //Obteniendo variables
        name_archivo = o.get("nombre_file");
        file_url = o.get("url_movil");

        AlertDialog.Builder alt_bld = new AlertDialog.Builder(Grupos.this);
        alt_bld.setMessage(getString(R.string.aDescargar))
        .setCancelable(false)
        .setPositiveButton(getString(R.string.aSi),new DialogInterface.OnClickListener() {
            //en la alerta le dices que si
            public void onClick(DialogInterface dialog, int id) {

                // Aqui ponemos el codigo a ejecutar
                // al pulsar el boton "guardar"
                dm = (DownloadManager) getSystemService(DOWNLOAD_SERVICE);
                Request request = new Request(
                Uri.parse(file_url));
                //Restringe los tipos de conexion en las cuales se procedera la descarga.
                request.setAllowedNetworkTypes
                (DownloadManager.Request.NETWORK_WIFI |
                DownloadManager.Request.NETWORK_MOBILE);
                //Elige si la descarga se hara en una conexion con roaming.
                request.setAllowedOverRoaming(false);
                //Descripcion de la descarga, se mostrara en la notificacion (si esta activada)
                request.setDescription(getString(R.string.Descarga));
                //La ubicacion donde se guardara el archivo en algun lugar de la memoria
                externa
                request.setDestinationInExternalPublicDir("/NENEK-SAAC", name_archivo);
                //Colocando que la notificacion sea visible siempre que se termine la descarga
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
                    request.setNotificationVisibility
                    (DownloadManager.Request.VISIBILITY_VISIBLE_NOTIFY_COMPLETED);
                }else {
                    request.setShowRunningNotification(true);
                }

                enqueue = dm.enqueue(request);
            }
        })
        .setNegativeButton(getString(R.string.aMasTarde), new DialogInterface.OnClickListener() {
            //aquí mas tarde...
            public void onClick(DialogInterface dialog, int id) {
                // Aqui ponemos el codigo a ejecutar
                // al pulsar el boton "Cancelar"
            }
        });
        AlertDialog alert = alt_bld.create();
    }
});

```

```
alert.setTitle(getString(R.string.aListaGrupo)); // Aqui ponemos el titulo de la ventana
alert.setIcon(R.drawable.indicator_arrow); // Aqui ponemos el icono de la ventana
alert.show();
```

### 1.3.3. Trabajo Colaborativo

Esta sección se utiliza para asignar tareas a los usuarios en un grupo de trabajo.

```
System.out.println("asignarTareaTask: Mandando informacion");
HttpParams httpParameters = new BasicHttpParams();
HttpProtocolParams.setContentCharset(httpParameters, HTTP.UTF_8);
HttpProtocolParams.setHttpElementCharset(httpParameters, HTTP.UTF_8);
    // Create a new HttpClient and Post Header
    HttpClient httpClient = new DefaultHttpClient();
    HttpPost httpPost = new
HttpPost(getString(R.string.ServicioURL)+getString(R.string.AsignarTarea));
System.out.println("asignarTareaTask: Observando si");
    try {
        System.out.println("asignarTareaTask: Se detiene");
        // Add your data
        List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();
        nameValuePairs.add(new BasicNameValuePair("id_user", id_user));
        nameValuePairs.add(new BasicNameValuePair("tarea", stringTituloTarea));
        nameValuePairs.add(new BasicNameValuePair("asignar", asignar));
        nameValuePairs.add(new BasicNameValuePair("fin", stringFecha));

        nameValuePairs.add(new BasicNameValuePair("desc", stringDescripcionTarea));
        //user que recibe la tarea
        nameValuePairs.add(new BasicNameValuePair("assign", stringTareaUsuario));
        //grupo que recibe la tarea
        nameValuePairs.add(new BasicNameValuePair("grupoassign", stringTareaGrupo));
        httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs, HTTP.UTF_8));

        // Execute HTTP Post Request
        HttpResponse response = httpClient.execute(httpPost);
        HttpEntity enty = response.getEntity();
        String text = EntityUtils.toString(enty);
        return text;
    }
```

### 1.4. Funcionamiento de editar perfil

Para editar el perfil un usuario puede subir una imagen de perfil, cambiar su contraseña, nombre y correo. La lógica para cambiar la información de un usuario se muestra a continuación:

```
public String postData(String valueWantToSend) {
    HttpParams httpParameters = new BasicHttpParams();
    HttpProtocolParams.setContentCharset(httpParameters, HTTP.UTF_8);
    HttpProtocolParams.setHttpElementCharset(httpParameters, HTTP.UTF_8);
        // Create a new HttpClient and Post Header
        HttpClient httpClient = new DefaultHttpClient();
```

```

        HttpPost httppost = new
HttpPost(getString(R.string.ServicioURL)+getString(R.string.PerfilService));

    try {
        // Add your data
        List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>();
        System.out.println("Variables " + id_user + " "+ us + " "+ pass + " "+ newpass + " "+ email);
        nameValuePairs.add(new BasicNameValuePair("id", id_user));
        nameValuePairs.add(new BasicNameValuePair("usuario", us));
        nameValuePairs.add(new BasicNameValuePair("password", pass));
        nameValuePairs.add(new BasicNameValuePair("nuevoPassword", newpass));
        nameValuePairs.add(new BasicNameValuePair("email", email));
        httppost.setEntity(new UrlEncodedFormEntity(nameValuePairs, HTTP.UTF_8));

                // Execute HTTP Post Request
                HttpResponse response = httpclient.execute(httppost);
                HttpEntity enty = response.getEntity();
                String text = EntityUtils.toString(enty);
                return text;
    } catch (ClientProtocolException e) {
        // TODO Auto-generated catch block
        return "ErrorServidor";
    } catch (IOException e) {
        // TODO Auto-generated catch block
        return "ErrorInternet";
    }
}

```

La lógica para cambiar una imagen de perfil se puede apreciar en el punto 8.7.

## 1.5. Funcionamiento de registro

En el registro los nuevos usuarios pueden acceder a la plataforma de Nenek-SAAC la forma en la que ellos ingresan se detalla a continuación.

## 1.6. Funcionamiento de despliegue de tareas

Al momento de recibir o asignar una tarea en grupos, el usuario las puede visualizar en esta clase. La lógica de muestra se aprecia en el siguiente código.

```

if(json != null){
    tareaslist.clear();
    if(TAG_DIRECTOR.equals("id_user")){
        TAG_STAT = "status";
        TAG_USER = "director";
    }else{
        TAG_STAT = "allowed";
        TAG_USER= "id_user";
    }
    for (int i = 0; i < json.length(); i++) {
        try {
            JSONObject c = json.getJSONObject(i);

```

```

String vtarea = c.getString(TAG_TITULO);
String vuser = c.getString(TAG_USER);
String vdesc = c.getString(TAG_DESC);
String vfecha = c.getString(TAG_FECHA);
String vstado = c.getString(TAG_STAT);
String vinicio = c.getString(TAG_INICIO);
String vid = c.getString(TAG_ID);
HashMap<String, String> map = new HashMap<String, String>();

// adding each child node to HashMap key => value
map.put(TAG_TITULO, vtarea);
map.put(TAG_USER, vuser);
map.put(TAG_DESC, vdesc);
map.put(TAG_FECHA, vfecha);
map.put(TAG_STAT, vstado); //Cambia por allowed o status
map.put(TAG_INICIO, vinicio);
map.put(TAG_ID, vid);

//obteniendo variables para trabajar
//map.put("id", String.valueOf(i)); //numero de posicion dentro

listview

map.put("user", vuser); //obteniendo el nombre del usuario
map.put("desc", vdesc); //descripcion de la tarea
map.put("fecha", vfecha); //fecha de solicitud
map.put("titulo", vtarea); //nombre de la tarea
map.put("inicio", vinicio); //inicio
map.put("estado", vstado); //estado
map.put("id_tarea", vid);
tarealist.add(map);

}catch (JSONException e) {
// TODO Auto-generated catch block
Toast.makeText(getApplicationContext(), getString(R.string.ErrorArchivos),
Toast.LENGTH_SHORT).show();
e.printStackTrace();
} //fin catch
} //fin for

adapter = new SpecialAdapter(context, tarealist,
R.layout.archivos_item, new String[] { TAG_TITULO, TAG_DESC, TAG_USER,
TAG_FECHA }, new int[] {
R.id.vehicleType, R.id.vehicleColor,
R.id.treadType, R.id.horaType});

// We need notify the adapter that the data have been changed
setListAdapter(adapter);
((BaseAdapter)adapter).notifyDataSetChanged();

System.out.println("TareasTask: Adaptador colocado");
super.onPostExecute(json);

```

## 1.7. Funcionamiento de subir y descargar archivos

En este apartado se detalla la función principal de Nenek-SAAC para subir y descargar archivos, ya sean imágenes, audio, archivos comprimidos, etcétera.

### 1.7.1. Lógica para subir archivos

La subida de archivos se detalla a continuación:

```
//obtenemos el tipo de extension ejemplo si es (foto.jpg = "image/jpeg") etc.
String extension = local_path.substring(local_path.lastIndexOf("."));
String mimeTypeMap =MimeTypeMap.getFileExtensionFromUrl(extension);
String mimeType = MimeTypeMap.getSingleton()

.getMimeTypeFromExtension(mimeTypeMap);

//Bien empecemos con la subida
HttpParams httpParameters = new BasicHttpParams();
HttpProtocolParams.setContentCharset(httpParameters, HTTP.UTF_8);
HttpProtocolParams.setHttpElementCharset(httpParameters, HTTP.UTF_8);
//Haciendo la conexion
HttpClient httpClient = new DefaultHttpClient();
HttpContext localContext = new BasicHttpContext();
HttpPost httpPost = new HttpPost(getString(R.string.ServicioURL)      +
getString(R.string.SubirService));

    MultipartEntity entity = new MultipartEntity(

    HttpMultipartMode.BROWSER_COMPATIBLE);

    ByteArrayOutputStream bos = new ByteArrayOutputStream();

    byte[] data = bos.toByteArray();
    entity.addPart("uploaded", new FileBody(new File(local_path), mimeType));
    entity.addPart("us", new StringBody(userT.getText().toString()));
    httpPost.setEntity(entity);

    HttpResponse response = httpClient.execute(httpPost, localContext);
    BufferedReader reader = new BufferedReader(new
InputStreamReader(response.getEntity().getContent(), "UTF-8"));

        String sResponse = reader.readLine();
        System.out.println(sResponse);
        return sResponse;
```

### 1.7.2. Lógica para descargar archivos

Para descargar archivos se utiliza una de las clases reservadas de Android: Download Manager esta funcionalidad se limita a versiones de API level 9 Android GingerBread (2.3) en adelante.

```

dm = (DownloadManager) getSystemService(DOWNLOAD_SERVICE);
Request request = new Request(
    Uri.parse(file_url));
//Restringe los tipos de conexion en las cuales se procedera la descarga.

request.setAllowedNetworkTypes(DownloadManager.Request.NETWORK_WIFI |
    DownloadManager.Request.NETWORK_MOBILE);
//Elige si la descarga se hara en una conexion con roaming.
request.setAllowedOverRoaming(false);
//Descripcion de la descarga, se mostrara en la notificacion (si esta activada)
request.setDescription(getString(R.string.Descarga));
//La ubicacion donde se guardara el archivo en algun lugar de la memoria externa
request.setDestinationInExternalPublicDir("/NENEK-SAAC", archivo);
//Colocando que la notificacion sea visible siempre que se termine la descarga
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
    request.setNotificationVisibility(
        DownloadManager.Request.VISIBILITY_VISIBLE_NOTIFY_COMPLETED);
}else {
    request.setShowRunningNotification(true);
}

enqueue = dm.enqueue(request);

```