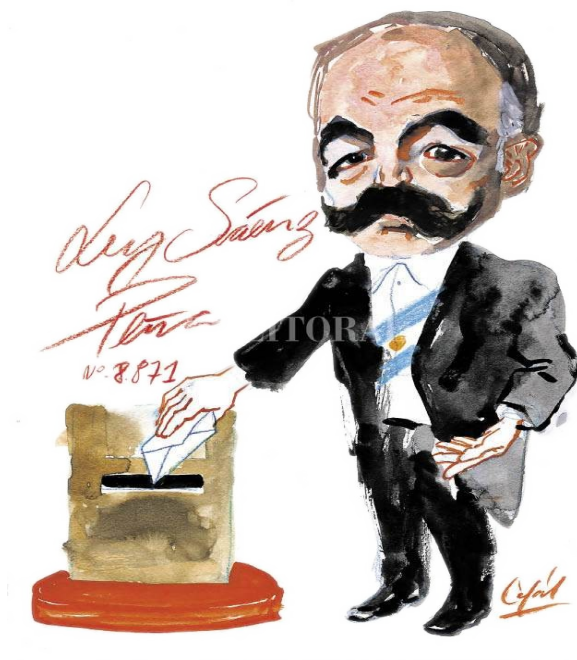




## ***Asistente Electoral: App Sáenz Peña***



### **Proyecto Integrador**

## ***Desarrollo de Aplicaciones en Dispositivos Móviles***

### **Alumno:**

Cajide Maidana, Manuel

127-407-7

[manuel.cajide.maidana@ieee.org](mailto:manuel.cajide.maidana@ieee.org)

**Docente:** Ing. Alejandro Breitman

### **Objetivo**

Aplicar los conocimientos adquiridos en la materia para desarrollar una aplicación para teléfonos móviles y tablets que asista a las autoridades de mesa para verificar la información necesaria y realizar el escrutinio de una votación.

Todo el código se encuentra disponible en: <https://github.com/envido32/AppSaenzPena>

### **Uso básico**

#### *Login:*

Existen tres tipos de usuarios cada uno con permisos diferentes de acción:

- Votante
- Fiscal
- Administrador

#### *Bienvenida:*

La aplicación le dará la bienvenida al usuario y le ofrecerá las siguientes opciones en pantalla:

##### *- Candidatos:*

Disponible a todos los usuarios.

Muestra una lista de los candidatos e información básica.

Haciendo clic en cada uno puede accederse a información extra.

Solo disponible al administrador: Agregar, editar y eliminar.

##### *- Padrón:*

Permite verificar el lugar de votación de un votante con sus datos, facilitado leyendo mediante código de barras del DNI.

##### *- Votar:*

No disponible ya que es imposible realizar un sistema de votación electrónico que garantice integridad, auditabilidad y secreto del voto simultaneamente.

##### *- Escrutinio:*

Muestra las mismas listas de partidos disponibles en Candidatos pero con un contador numérico. Luego permite verificar la contabilización total.

##### *- Agregar Usuario:*

Solo disponible al administrador: Permite agregar un usuario con su contraseña y tipo (votante, fiscal o administrador).

##### *- Configuración:*

Solo disponible al administrador: Accede a las opciones de configuración.

Cambia el formato de colores de la aplicación.

Activa el modo Dialog en algunas actividades.

Permite configurar el distrito donde se realiza la votación manualmente o por GPS.

## Explicación técnica

### **Splash Screen Activity**

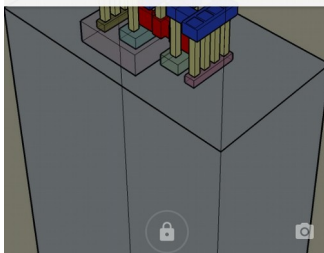
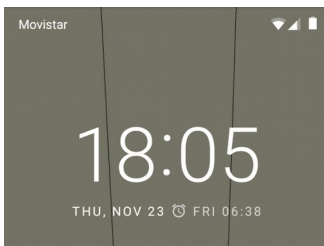
Durante el Splash Screen se cargan las bases de datos necesarias y en caso de no estar disponibles se las generan. Esto incluye las bases de los candidatos y usuarios.

La base de datos SQLite “DB\_Votacion” contiene dos tablas: “Candidatos” y “Usuarios”.

```
VotacionSQLiteHelper dbVotacionHelper =  
    new VotacionSQLiteHelper(this, "DB_Votacion", null, 1);  
SQLiteDatabase dbVotacion = dbVotacionHelper.getWritableDatabase();  
dbVotacionHelper.onUpgrade(dbVotacion, 1, 1);  
  
int codigo = 1; // 0 = invalido, 1 = admin, 2 = fiscal, 3 = votante  
String nombre = "admin";  
String pass = "admin";  
ContentValues nuevoRegistro = new ContentValues();  
nuevoRegistro.put("codigo", codigo);  
nuevoRegistro.put("nombre", nombre);  
nuevoRegistro.put("pass", pass);  
//Insertamos el registro en la base de datos  
dbVotacion.insert("Usuarios", null, nuevoRegistro);
```

Se utiliza el mismo método para generar los demás usuarios y la lista de candidatos. Luego la actividad pasara directamente a la pantalla de Login.

```
TimerTask task = new TimerTask(){  
    @Override  
    public void run(){  
        // Start the next activity  
        Intent mainIntent = new Intent().setClass(getBaseContext(),  
LoginActivity.class);  
  
        startActivity(mainIntent);  
        // Terminamos la actividad para que el usuario no  
        // pueda volver para atrás con el botón de back  
        finish();  
    }  
};  
// Simulamos con un timer un tiempo de espera definido en  
// una constante al comienzo  
Timer timer = new Timer();  
timer.schedule(task, SPLASH_SCREEN_DELAY);
```



Para garantizar estabilidad del sistema esto se realiza mediante un Async Task y al finalizar envía una notificación al Sistema Operativo.

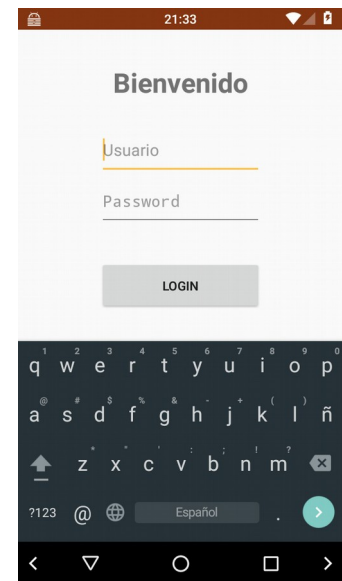
```
mBuilder =  
    new NotificationCompat.Builder(SplashActivity.this)  
  
.setSmallIcon(android.R.drawable.stat_sys_warning)  
.setLargeIcon(((BitmapDrawable)getResources()
```

```
        .getDrawable(R.drawable.saenzpena)).getBitmap()))
        .setContentView(R.layout.alerta)
        .setTitle("Mensaje de Alerta")
        .setSubtitle("Base de datos cargada!.")
        .setInfo("4")
        .setTicker("Alerta!");
notIntent = new Intent(SplashActivity.this, LoginActivity.class);
contIntent = PendingIntent.getActivity(SplashActivity.this, 0, notIntent, 0);
mBuilder.setContentIntent(contIntent);
mNotificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
```

## Login Activity

Actividad básica de login, verificara la contraseña con la base de datos y pasara al MainActivity de ser valida.

```
String[] campos = new String[] {"codigo", "nombre", "pass"};
String[] args = new String[] {usrMail.getText().toString()};
Cursor dbUsuariosCursor = dbVotacion.query("Usuarios",
    campos, "nombre=?", args, null, null, null);
//Nos aseguramos de que existe al menos un registro
if (dbUsuariosCursor.moveToFirst()) {
    do {
        int codigo = dbUsuariosCursor.getInt(0);
        String nombre = dbUsuariosCursor.getString(1);
        String pass = dbUsuariosCursor.getString(2);
        //Verifican login
        if (usrPass.getText().toString().equals(pass)) {
            usrType = codigo;
        }while(dbUsuariosCursor.moveToNext());
    }
}
```



Por SharedPreferences guardara el tipo de usuario (admin, fiscal o votante) usado durante toda la sesión.

```
SharedPreferences prefs =
    getSharedPreferences("MyPref", Context.MODE_PRIVATE);
SharedPreferences.Editor editor = prefs.edit();
editor.putInt("usrType", usrType);
editor.apply();
```

Por bundle le pasara el nombre de usuario y lanza el Main.

```
//Creamos el Intent
Intent intent =
    new Intent(getBaseContext(), MainActivity.class);
//Creamos la información a pasar entre actividades
Bundle bundleInfo = new Bundle();
bundleInfo.putString("usrMail", usrMail.getText().toString());
//Añadimos la información al intent
intent.putExtras(bundleInfo);
//Iniciamos la nueva actividad
startActivity(intent);
```



## Main Activity

La actividad posee cuadro botones principales que acceden a otras actividades y dos botones mas en el Toolbar.

En la bienvenida muestra el nombre del usuario pasada por Bundle.

```
//Recuperamos la información pasada en el intent
Bundle bundle = this.getIntent().getExtras();
textSaludos.setText("Bienvenido: " +
    bundle.getString("usrMail"));
```

Y de acuerdo al tipo de usuario accedido estarán disponibles los botones correspondientes.

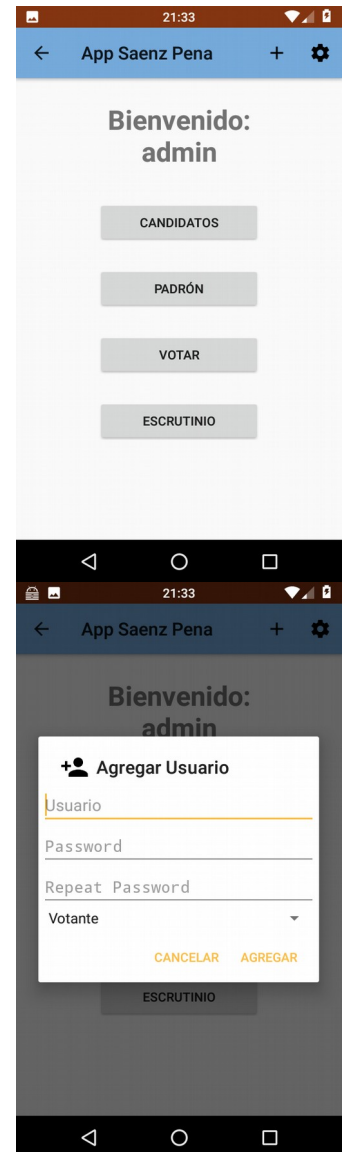
```
prefs = getSharedPreferences("MyPref", Context.MODE_PRIVATE);
usrType = prefs.getInt("usrType", 0);
```

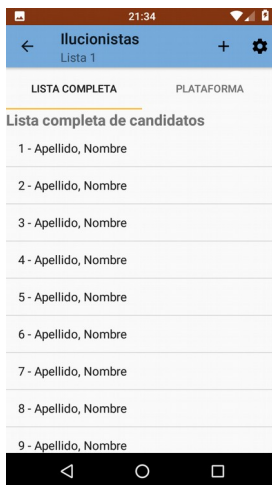
```
switch (usrType) {
    case 1: { // Admin
        btnEscrutineio.setVisibility(View.VISIBLE);
    }break;
    case 2: { // Fiscal
        btnEscrutineio.setVisibility(View.VISIBLE);
    }break;
    case 3: { // Votante
        btnEscrutineio.setVisibility(View.GONE);
    }break;
}
```

Y luego con el toolbar

```
public boolean onCreateOptionsMenu (Menu menu) {
    if(usrType==1) {
        getMenuInflater().inflate(R.menu.main, menu);
    }
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    return true;
}
```

Donde, en caso de ser administrador, estará disponible la opción de agregar mas usuarios con una interfaz tipo Dialog y el botón para acceder a las configuraciones.

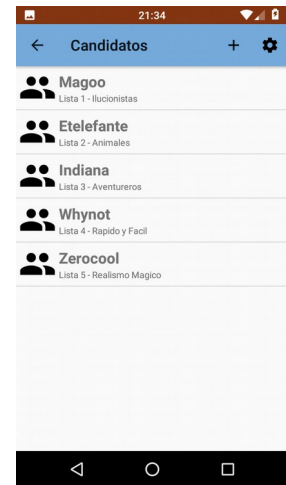




## Candidatos Activity

Aquí se harán disponibles las listas de candidatos disponibles en formato ListView hecho a medida con una Adaptador cargadas de la base de datos SQLite con ayuda de un `ArrayList<CandidatosClass>`.

`CandidatosClass` posee la información del nombre del candidato, nombre del partido y numero de lista.



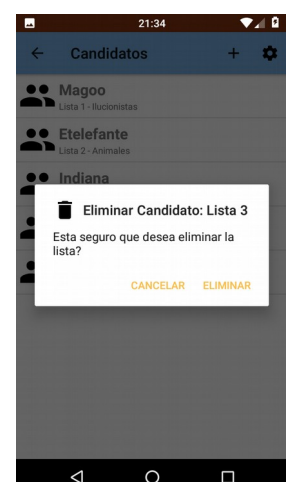
```
class AdaptadorCandidatos extends ArrayAdapter<CandidatoClass> {
    public ArrayList<CandidatoClass> datos;
    public AdaptadorCandidatos(Context context, ArrayList<CandidatoClass>
datos_candidatos) {
        super(context, R.layout.listitem_candidato, datos_candidatos);
        datos = datos_candidatos;
    }
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = LayoutInflater.from(getContext());
        View item = inflater.inflate(R.layout.listitem_candidato, null);
        TextView lblNombre = (TextView)item.findViewById(R.id.LblNombre);
        lblNombre.setText(datos.get(position).getNombre());
        TextView lblPartido = (TextView)item.findViewById(R.id.LblPartido);
        lblPartido.setText("Lista " + datos.get(position).getLista() + " - " +
datos.get(position).getPartido());
        return(item);
    }
}
```



Haciendo clic sobre el candidato se accede a mayor información de la lista y el partido en un formato Tabs.

En caso de ser administrador permitirá las siguientes opciones:

- El botón + en el Toolbar permite agregar un candidato nuevo.
- Manteniendo apretado el candidato permite eliminarlo.





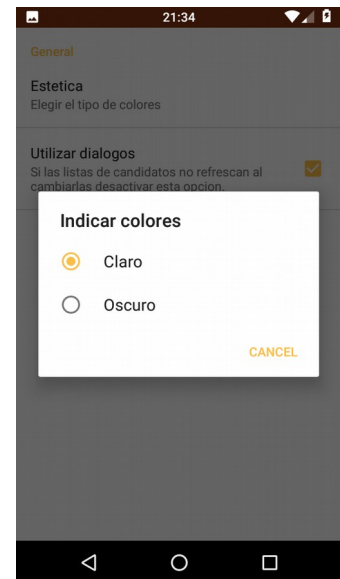


## Settings Activity

Solo el administrador puede acceder a esta opción. Utilizando el formato de Preferences hay tres opciones disponibles:

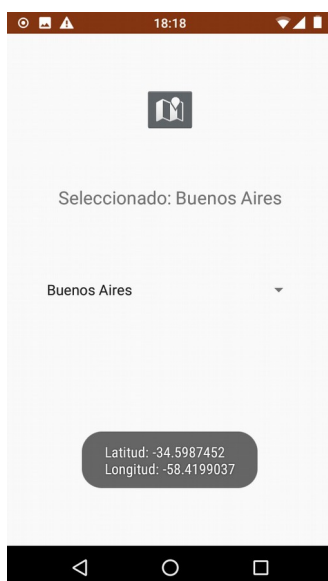
- Un List con dos opciones de combinación de colores para la aplicación (claro y oscuro).
- Un CheckBox que permite desactivar los Dialogs (y utilizar Activities tradicionales) en caso de tener inconvenientes de compatibilidad.
- Una opción para seleccionar el distrito por GPS o manual.

```
<ListPreference
    android:key="opcTheme"
    android:title="Estetica"
    android:summary="Elegir el tipo de colores"
    android:dialogTitle="Indicar colores"
    android:entries="@array/themetype"
    android:entryValues="@array/themecode" />
<CheckBoxPreference
    android:key="opcDialog"
    android:title="Utilizar dialogs"
    android:summary="@strings/dialogo_change" />
<Preference
    android:key="opcGPS"
    android:summary="Elegir distrito por GPS"
    android:title="Distrito" >
    <intent android:targetClass="com.electiva.envido32.appsaenzpena.DistritoActivity"
        android:targetPackage="com.electiva.envido32.appsaenzpena">
    </intent>
</Preference>
```



## Distrito Activity

Permite cargar la ubicación geográfica mediante GPS o manualmente en ayuda de un Spinner . Dato guardado en SharedPreferences y el distrito por GPS utiliza Google Play API.



```
apiClient = new GoogleApiClient.Builder(this)
    .enableAutoManage(this, this)
    .addConnectionCallbacks(this)
    .addApi(LocationServices.API)
    .build();
```

```
Location lastLocation =
```

```
LocationServices.FusedLocationApi.getLastLocation(apiClient);
lat = lastLocation.getLatitude();
lng = lastLocation.getLongitude();
```

```
List<Address> addresses = gcd.getFromLocation(lat, lng, 1);
String countryName = addresses.get(0).getAdminArea();
```

## Padrón Activity

Mediante WebView se muestra en pantalla la pagina web oficial de la Cámara Nacional Electoral, pueden llenarse los datos a mano o con la cámara leer el código de barras en el frente del DNI a través de la librería Zxing y luego volcados en la pagina web por comandos JavaScript.



```
myWebView = (WebView) findViewById(R.id.webview);
myWebView.setWebViewClient(new MyWebViewClient());
myWebView.getSettings().setJavaScriptEnabled(true);
myWebView.loadUrl("https://www.padron.gob.ar/");
```

```
IntentIntegrator integrator = new IntentIntegrator(this);
integrator.setPrompt("Lea el código de barras al frente del DNI");
integrator.setOrientationLocked(true);
```

@Override

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    IntentResult result = IntentIntegrator.parseActivityResult(
        requestCode, resultCode, data);
```

```
        myWebView.setVisibility(View.VISIBLE);
        String barcode = result.getContents();
        String[] info = barcode.split("@");
        String orden = info[0];
        String apellido = info[1];
        String nombre = info[2];
        switch (info[3]) {
            case "M": sexo = 0; break;
            case "F": sexo = 1; break;
            default:
        }
    }
```

```
        dni = Integer.parseInt(info[4]);
        String emicion = info[5];
        String vencimiento = info[6];
        SharedPreferences prefs = getSharedPreferences("MyPref",
            Context.MODE_PRIVATE);
        distrito = prefs.getInt("opcDistrito", -1);
        String js = "javascript:" +
            "document.getElementById('matricula').value='"+dni+"';" +
            "document.getElementById('distrito').value='"+distrito+"';" +
            "document.getElementsByName('sexo')[ '"+sexo+"'].checked=true;";
```

```
        myWebView.evaluateJavascript(js, new ValueCallback<String>());
```





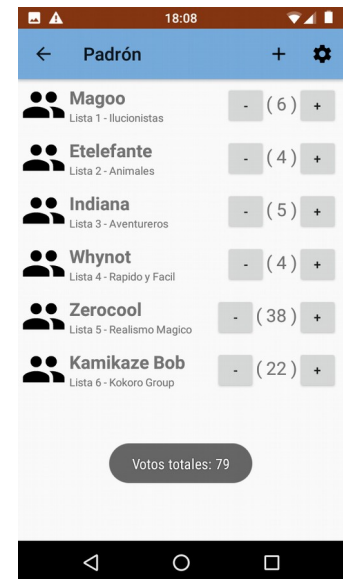


## Escrutinio Activity

Mediante RecyclerView muestra en pantalla las listas habilitadas en “Candidatos Activity” y mediante un contador interactivo permite asistir a las autoridades de mesa a realizar el conteo y finalmente realiza la sumatoria total.

```
rv = (RecyclerView)findViewById(R.id.RecView);
rv.setHasFixedSize(true);
LinearLayoutManager llm = new
LinearLayoutManager(getBaseContext());
rv.setLayoutManager(llm);
adaptador = new AdaptadorEscrutineo(datos_candidatos);
rv.setAdapter(adaptador);
```


```
class AdaptadorEscrutineo extends
RecyclerView.Adapter<AdaptadorEscrutineo.EscrutineoViewHolder> {
    public ArrayList<CandidatoClass> datos;
    public static class EscrutineoViewHolder extends
RecyclerView.ViewHolder {
        public TextView escPartido;
        public TextView escNombre;
        public TextView escCount;
        public Button addBtn;
        public Button lessBtn;
        EscrutineoViewHolder(final View itemView) {
            super(itemView);
            escPartido = (TextView)itemView.findViewById(R.id.LblPartido);
            escNombre = (TextView)itemView.findViewById(R.id.LblNombre);
            escCount = (TextView)itemView.findViewById(R.id.LblCuenta);
            addBtn = (Button)itemView.findViewById(R.id.buttonAdd);
            addBtn.setVisibility(View.VISIBLE);
            lessBtn = (Button)itemView.findViewById(R.id.buttonLess);
            lessBtn.setVisibility(View.VISIBLE);
        }
    }
    public AdaptadorEscrutineo(ArrayList<CandidatoClass> datos_candidatos) {
        datos = datos_candidatos;
    }
    @Override
    public EscrutineoViewHolder onCreateViewHolder(ViewGroup viewGroup, int i) {
        View v = LayoutInflater.from(viewGroup.getContext()).inflate(
R.layout.listitem_candidato, viewGroup, false);
        EscrutineoViewHolder evh = new EscrutineoViewHolder(v);
        return evh;
    }
    @Override
    public void onBindViewHolder(
final EscrutineoViewHolder candidatoViewHolder, final int position) {
        candidatoViewHolder.escNombre.setText(datos.get(position).getNombre());
        candidatoViewHolder.escPartido.setText(
"Lista " + datos.get(position).getLista() + " - " + datos.get(position).getPartido());
        candidatoViewHolder.escCount.setText(
"( " + datos.get(position).getCount() + " )");
```





```
Button buttonAdd = candidatoViewHolder.addBtn;
buttonAdd.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        datos.get(position).addCount();
        candidatoViewHolder.escCount.setText(
            "( " + datos.get(position).getCount() + " )");
    }
});
Button buttonLess = candidatoViewHolder.lessBtn;
buttonLess.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        datos.get(position).lessCount();
        candidatoViewHolder.escCount.setText(
            "( " + datos.get(position).getCount() + " )");
    }
});
}
```

## Publicar en Play Store

Campo	Descripción
Nombre	App Saens Pena
Descripción breve	Asistente electoral para autoridades de mesa, fiscales y votantes.
Descripción completa	App Sáens Peña es una aplicación de ayuda para el día electoral. Permite ver información de los candidatos y sus partidos políticos, verificar el lugar de votación en el padrón y para las autoridades de mesa un asistente de conteo para confeccionar las actas disminuyendo los errores de carga.
Categoría	Herramientas
Clasificación	Apto para todo publico
Icono	
Distribución	Gratuita – Ad runned – A pedido de gobiernos
Localización	Argentina – Expantible a países de habla hispana
Idioma	Español – Expandible a países de habla inglesa