

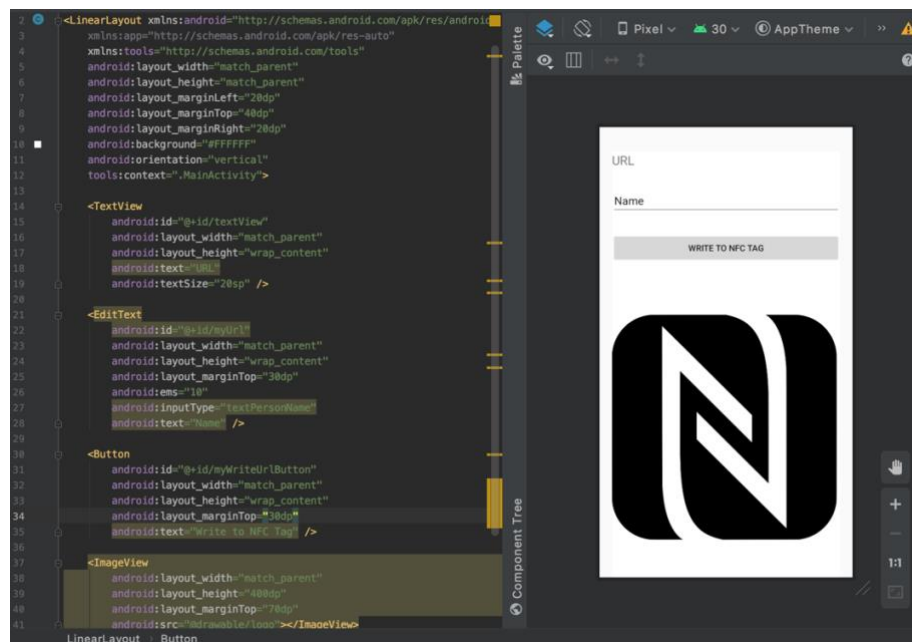
Introducción

Diseñar una aplicación móvil que permita la comunicación vía NFC para el intercambio de información entre un dispositivo móvil y una etiqueta NFC, pasiva o activa. En forma alternativa, se puede realizar la comunicación entre dos dispositivos móviles que posean, ambos, el sensor NFC. Transferir alguna información textual.

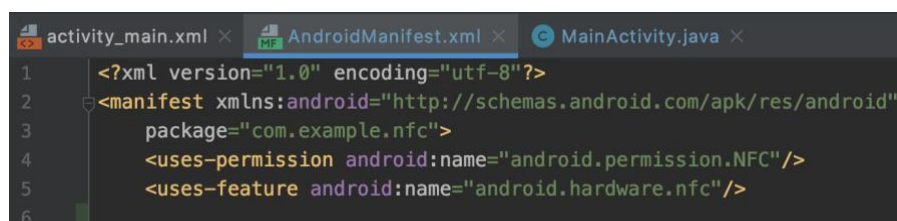
Desarrollo

Para la implementación de este proyecto se uso una etiqueta NFC como se muestra en la ilustración 1, así como un dispositivo que cuente con NFC (Ilustración 2). En cuestión del desarrollo, la aplicación puede leer, así como escribir una etiqueta NFC.

Primero se ha implementado la interfaz, editando el archivo `activity_main`, el cual cuenta con un `TextView`, un `Edittext`, en el cual vamos a poner el mensaje, en nuestro ejemplo una dirección web, posteriormente tenemos un botón el cual nos va a servir para poder escribir el mensaje que se ha escrito anteriormente en la etiqueta NFC y por último tenemos una imagen con el logo de NFC.



De igual manera se tienen que agregar los permisos necesarios en el Manifest del proyecto, para soportar el uso de NFC.



En cuestión del código primero se programó la parte la cual pudiera escribir en una etiqueta al momento de reconocerla y una vez que se ha reconocida saltara un cuadro para poder empezar a escribir en la etiqueta.

```
case DIALOG_WRITE_URL:
    return new AlertDialog.Builder( context: this).setTitle("Write URL to tag")
        .setMessage("Touch tag to start writing")
        .setCancelable(true)
        .setNegativeButton(android.R.string.cancel, (dialog, which) -> {
            dialog.cancel();
        })
        .setOnCancelListener((dialogInterface) -> {
            writeUrl = false;
        }).create();
```

Posteriormente la aplicación empezara a escribir el mensaje en ella. Esta parte se consideran dos casos en específico, el primero esta dentro de un if el cual va a definir si la etiqueta que se va a escribir ya esta pre formateada, en su caso se procede a escribir. En caso de que no este formateada, se incluye un else para que de esta forma se pueda escribir en ella.

```
152         if (ndefTag != null) {
153             try {
154                 ndefTag.connect();
155                 ndefTag.writeNdefMessage(msg);
156             } catch (Exception e) {
157             } finally {
158                 try {
159                     ndefTag.close();
160                 } catch (Exception e) {
161                 }
162             }
163         }
164     } else {
165         NdefFormatable ndefFormatableTag = NdefFormatable.get(tag);
166         if (ndefFormatableTag != null) {
167             try {
168                 ndefFormatableTag.connect();
169                 ndefFormatableTag.format(msg);
170             } catch (Exception e) {
171             } finally {
172                 try {
173                     ndefFormatableTag.close();
174                 } catch (Exception e) {
175                 }
176             }
177         }
178     }
```

La otra parte de la aplicación que se ha programado es la de poder leer una etiqueta NFC, primero se detecta la etiqueta que se quiere leer, esto mostrando un cuadro de dialogo que se ha podido hacer.

```

case DIALOG_NEW_TAG:
    return new AlertDialog.Builder( context: this).setTitle("Tag Detected")
        .setCancelable(true)
        .setNeutralButton(android.R.string.ok, (dialog, which) -> {
            dialog.dismiss();
        }).create();

180     StringBuilder tagInfo = new StringBuilder();
181     byte[] uid = tag.getId();
182     tagInfo.append("UUID: ")
183         .append(toHexString(uid))
184         .append("\n\n");

```

Una vez que se detecto se procede a leer el mensaje que hay en le etiqueta NFC, en donde se vana encontrar los registros y por ultimo se muestran los datos encontrados.

```

activity_main.xml x AndroidManifest.xml x MainActivity.java x
181     byte[] uid = tag.getId();
182     tagInfo.append("UUID: ")
183         .append(toHexString(uid))
184         .append("\n\n");
185     Parcelable[] ndefRaw = data.getParcelableArrayExtra(NfcAdapter.EXTRA_NDEF_MESSAGES);
186     NdefMessage[] ndefMsgs = null;
187     if (ndefRaw != null) {
188         ndefMsgs = new NdefMessage[ndefRaw.length];
189         for (int i = 0; i < ndefMsgs.length; i++) {
190             ndefMsgs[i] = (NdefMessage) ndefRaw[i];
191         }
192     }
193     if (ndefMsgs != null) {
194         for (int i = 0; i < ndefMsgs.length; i++) {
195             NdefRecord[] records = ndefMsgs[i].getRecords();
196             if (records != null) {
197                 for (int j = 0; j < records.length; j++) {
198                     if ((records[j].getTnf() == NdefRecord.TNF_WELL_KNOWN) && Arrays.equals(records[j].getType(),
199                         NdefRecord.RTD_URI)) {
200                         byte[] payload = records[j].getPayload();
201                         String uri = new String(Arrays.copyOfRange(payload, from: 1, payload.length),
202                             Charset.forName("UTF-8"));
203                         tagInfo.append("URI: ").append(uri).append("\n");
204                     }
205                 }
206             }
207         }
208     }

```

Y una vez que se encuentran se procede a visitar la pagina web o hacer la acción correspondiente. Para poder validar que todo se ha hecho correctamente, de igual manera se ha usado el lector de una etiqute NFC del celular. Y como se muestra todo ha salido correctamente. Al final se muestra la pagina de la escuela, como se escribió en la etiqueta NFC.

Por ultimo se muestran las imágenes de como se ve nuestra aplicación en un dispositivo.



Ilustración 1 Etiqueta NFC

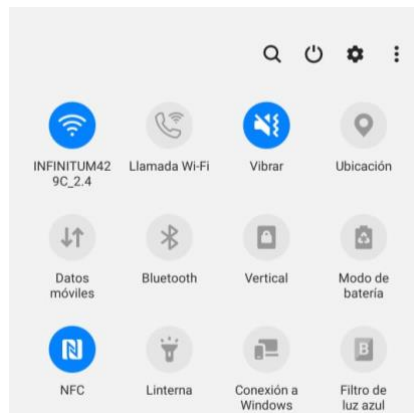


Ilustración 2 Dispositivo con NFC

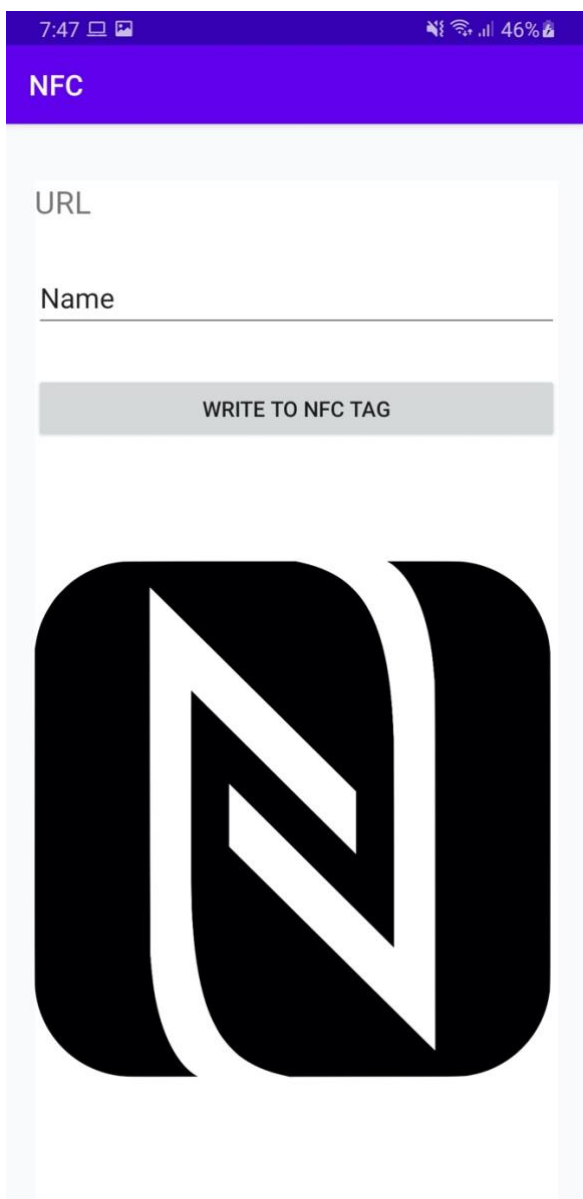


Ilustración 3 Interfaz de la app



Ilustración 4 Ingresar mensaje

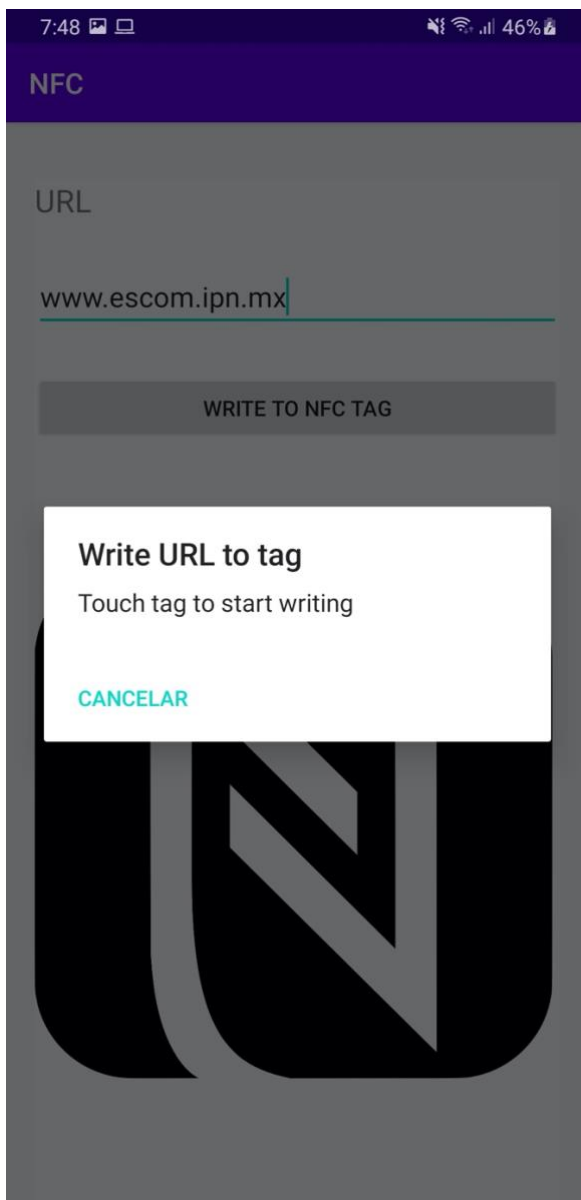


Ilustración 5 Mensaje de escritura

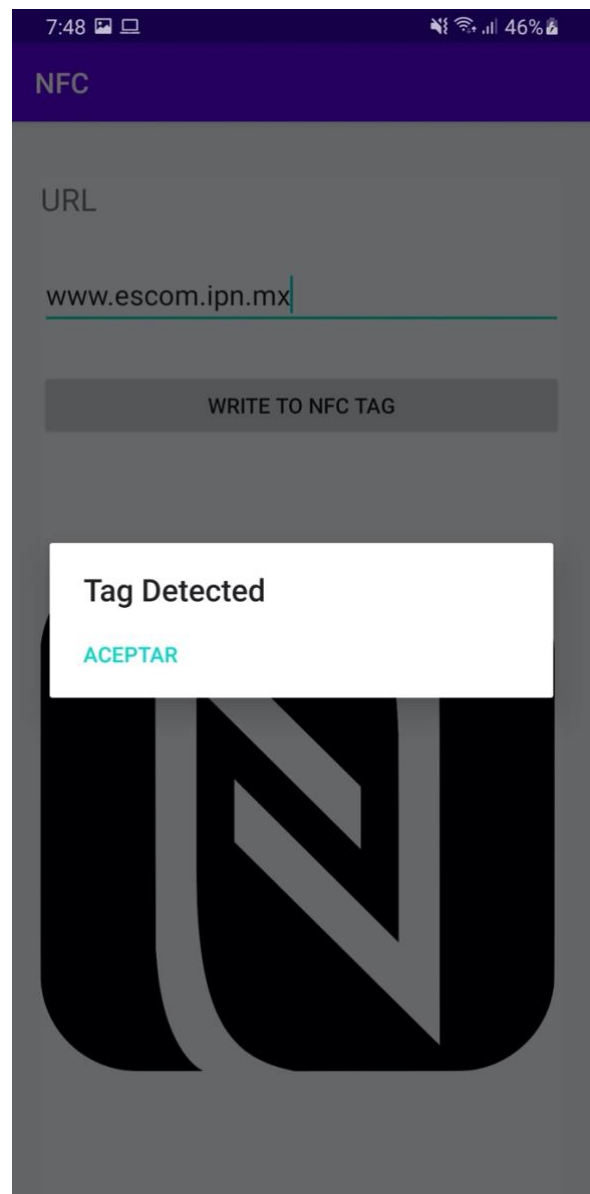


Ilustración 6 Mensaje de escritura

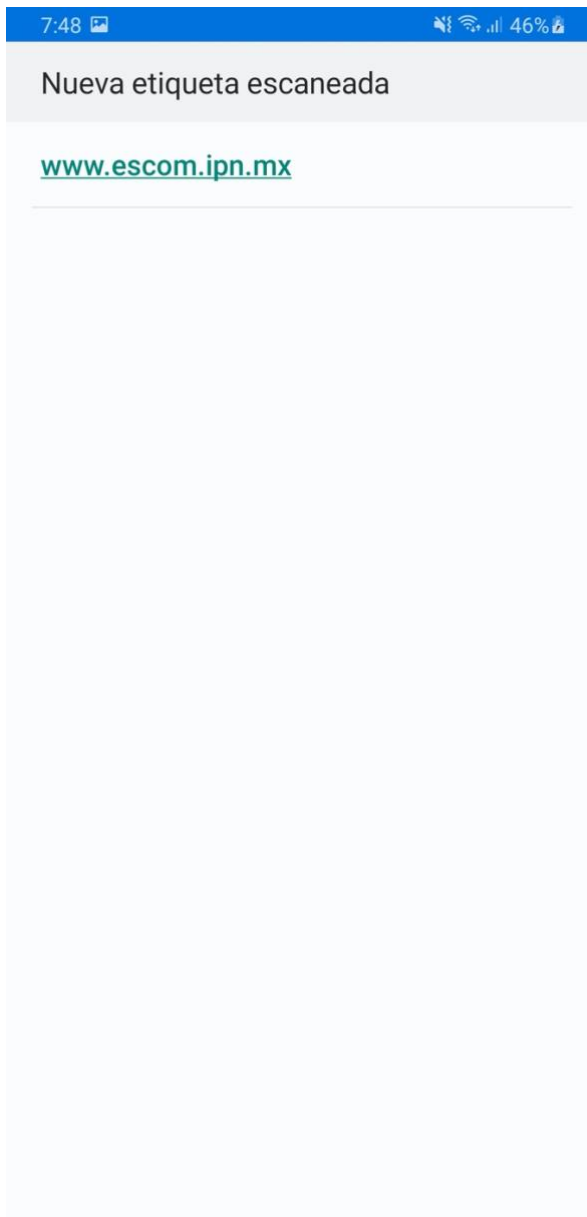


Ilustración 7 Validación de mensaje



Ilustración 8 Pagina de escom

Referencias

- [1] Cifeuntes. A, Application Development for Mobile Devices (2020).