

APIs Nativas: Almacenamiento de Datos

Joel Ortíz

Octubre 18 de 2013

@flashinstructor



- BlackBerry Elite Member & Community Manager
- Adobe Certified Instructor, Expert & Community Manager
- StarWars FanBoy
- BlackBerry, Activ
Developing Experiences

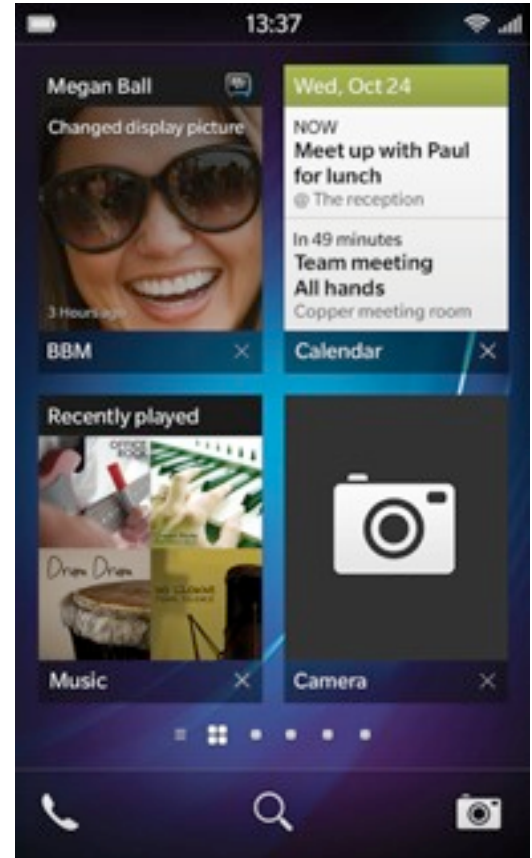


Archivos y Sistema de Archivos

No es tan aburrido como parece

Datos, Datos y Más Datos

- No es sorpresa:
 - ▶ Nuestras aplicaciones requieren datos siempre
 - ▶ Es como si fuera la comida de las aplicaciones XD
- Cascades y aplicaciones nativas tienen diferentes medios para almacenar y obtener datos
- Si es sorpresa:
 - ▶ Tendremos perímetros para el acceso a datos

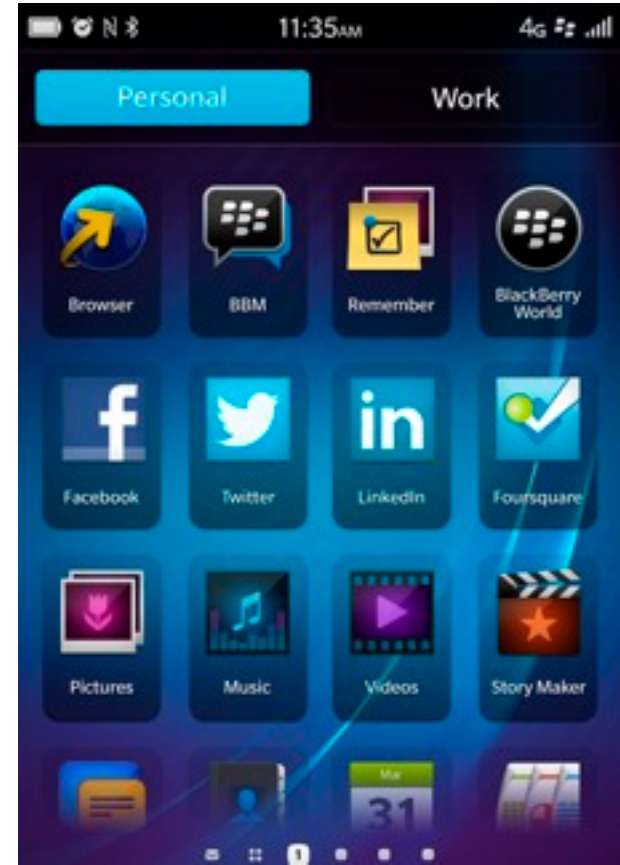


- Las aplicaciones tiene la capacidad de:
 - ▶ Leer y escribir hacia y desde el sistema de archivos
 - ▶ Crear archivos y carpetas
 - ▶ Crear bases de datos
 - ▶ Crear consultas a bases de datos
 - ▶ Obtener datos desde un servidor remoto o red local

BlackBerry Balance



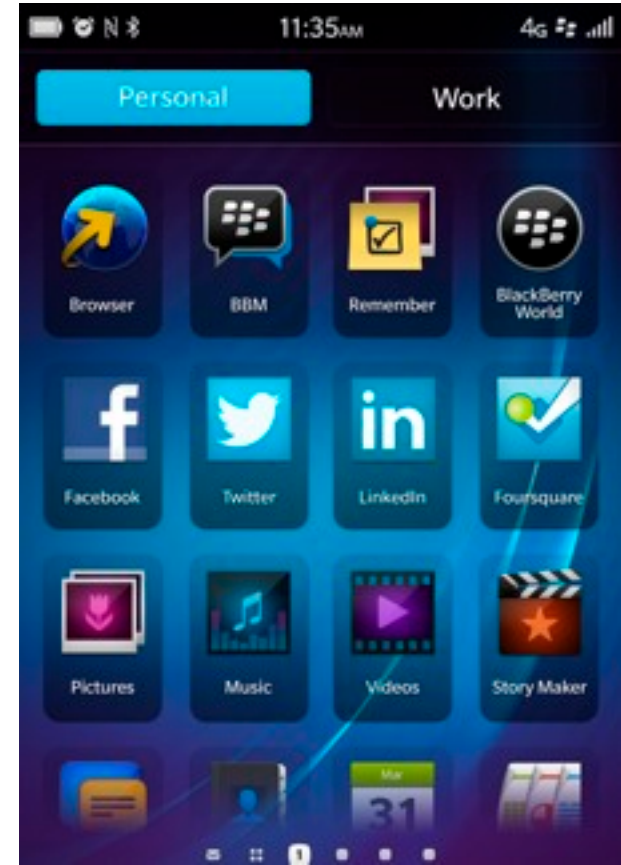
- Aplicaciones de **trabajo/empresariales**:
 - ▶ Pueden acceder a datos de trabajo y visualizar datos personales
 - ▶ Pueden adjuntar archivos personales a emails o entradas del calendario de trabajo
 - ▶ Pueden acceder a intranets usando BlackBerry Bridge, VPNs corporativas o WiFi
- Aplicaciones **personales**:
 - ▶ No pueden acceder a datos de trabajo
 - ▶ No pueden adjuntar archivos de trabajo a emails o entradas de calendario personales



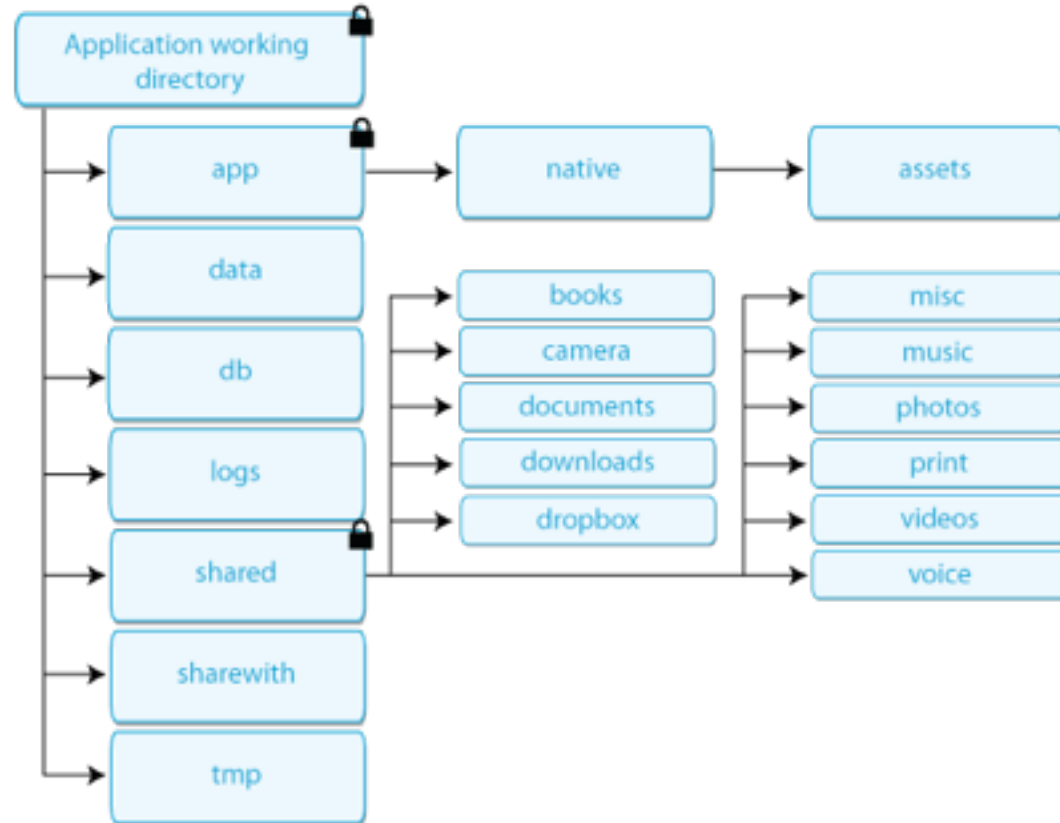
BlackBerry Balance



Aplicaciones de Trabajo	Aplicaciones de Trabajo/Personales	Aplicaciones Personales
BES email, contactos, calendario, tareas etc.	Visor de fotos	Social (Facebook, Twitter, etc.)
BlackBerry Bridge & Viewer	Media Player	App World
BlackBerry Browser	Visor de documentos (Acrobat, Doc To Go)	Web Browser
Aplicaciones de Trabajo	Aplicaciones de Archivos	Video Chat



- Las aplicaciones corren sobre un “sandbox”
- El “sandbox” tiene directorios tales como:
 - Datos
 - Activos
 - Tu aplicación compilada
 - Etc.
- No todos los directorios son para nuestro uso




```
FILE *fp;  
fp = fopen( "report.dat", "r" );  
if( fp != NULL )  
{  
    /* rest of code goes here */  
    fclose( fp );  
}
```

```
QFile textfile("data/files/text/newfile.txt");  
textfile.open(QIODevice::WriteOnly | QIODevice::Text);  
QTextStream out(&textfile);  
out << "This is a text file\n";  
textfile.close();
```

- Cascades tiene una clase que nos permite el acceso al “sandbox”:

`QDir`

- Además provee funciones que nos permiten el acceso a diferentes rutas del sistema de archivos

`QDir::currentPath()` – ruta al directorio de trabajo de la app

`QDir::homePath()` – retorna el directorio de datos de la app

`QDir::tempPath()` – accede al directorio temporal de la app

- No todos los directorios visibles son accesibles:
 - ▶ app
 - Aplicación compilada, activos, source
 - ▶ data
 - Este es donde se almacenan tus datos. Variable `$HOME`
 - ▶ db
 - Archivos de bases de datos de la aplicación
 - ▶ logs
 - Sistema de logs para una aplicación
 - ▶ shared
 - Subfolders que contienen datos compartidos agrupados por tipos. Todas las aplicaciones pueden leer este directorio. Una aplicación puede escribir en este directorio especificando el permiso `access_shared`
 - ▶ tmp

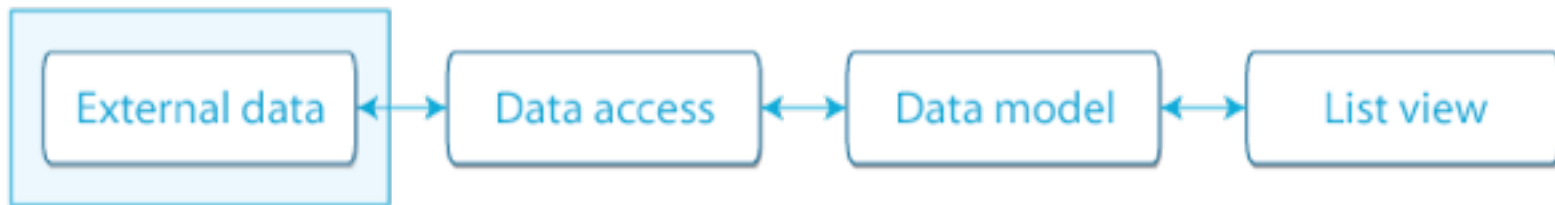
1 Demo

5 Minutos

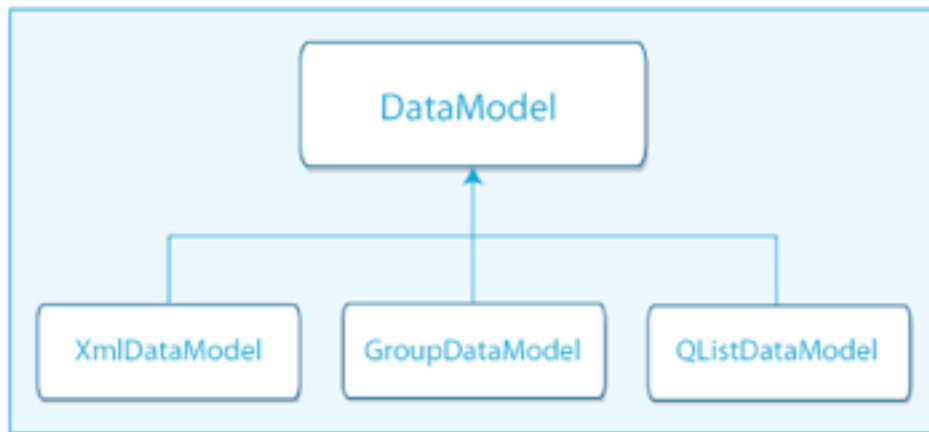
Cascades: Modelo y Acceso a Datos

¡MVC! si, como lo escuchas ¡MVC!

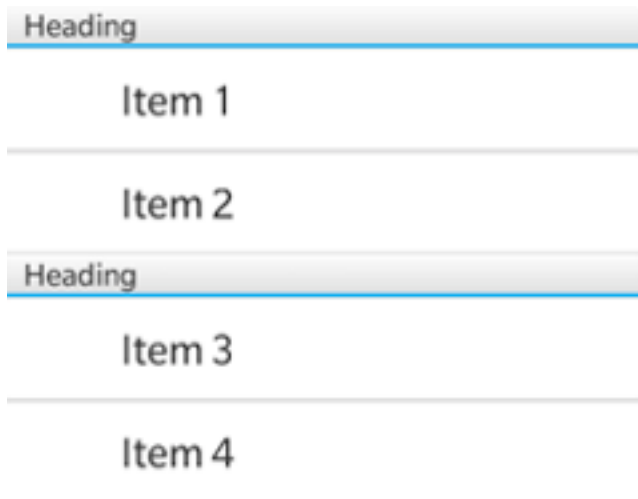
- Cascades ofrece un “data model” que permite el consumo y presentación de datos de forma fácil
- Los datos externos (JSON o bases de datos) son cargados por un objeto DataAccess
- Los datos cargados son organizados por la clase DataModel
- Cascades provee clases “ListView” para presentar los DataModel



- Un “data model” provee la información para ser desplegada en una lista
- El DataModel puede ser extendido por una representación personalizada de los datos



- Un ListView determina como serán desplegados los datos en tu aplicación
- Pueden ser asociados con un DataModel



2 Demo

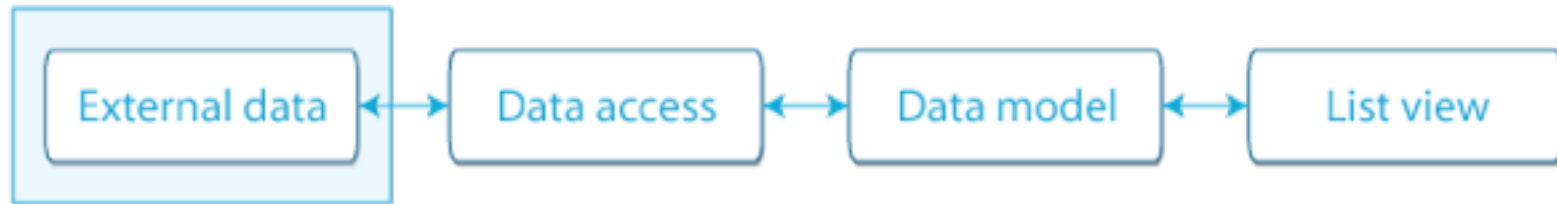
5 Minutos

Bases de Datos

¡Integremos Bases de Datos!



- Empaquetad con las herramientas SQLite
- Definido como libsqlite
- Esta disponible, pero se recomienda usar QSql si se esta empleando C++
- Para aplicaciones creadas en Cascades también existe una interfaz para interactuar con bases de datos



Código QtSQL... ¿fácil?



```
#include <QSqlDatabase>

...

QSqlDatabase db = QSqlDatabase::addDatabase("QPSQL");

db.setHostName("myserver");
db.setDatabaseName("mydb");
db.setUserName("myusername");
db.setPassword("mypassword");

bool ok = db.open();
if(ok)
{
    QSqlQuery query("SELECT * FROM *");
}
```

3 Demo

5 Minutos

APIs Nativas: NFC

Joel Ortíz

Octubre 18 de 2013

@flashinstructor



Near Field Communications es una tecnología mágica

Con solo chocar los dispositivos NFC puedes...

*Disparar
Acciones*

*Transferir
Estados*

*Transferir
Objetos*

*Ejecutar
Transacciones*

*Buscar llegadas
de Bus*

*Juegos... ¡Tu
Turno!*

*Fotos,
Contactos, etc.*

Pagos

Enriquece tus aplicaciones con NFC

Táctiles

Sin Gestos

Conveniente

Diversión

Seguro

“NFC Móvil... Espera un crecimiento de usuarios 16.7 M a más de 250 M en 4 años (2011 - 2015)”

*Disparar
Acciones*

*Transferir
Estados*

*Transferir
Objetos*

Tags

Punto a Punto

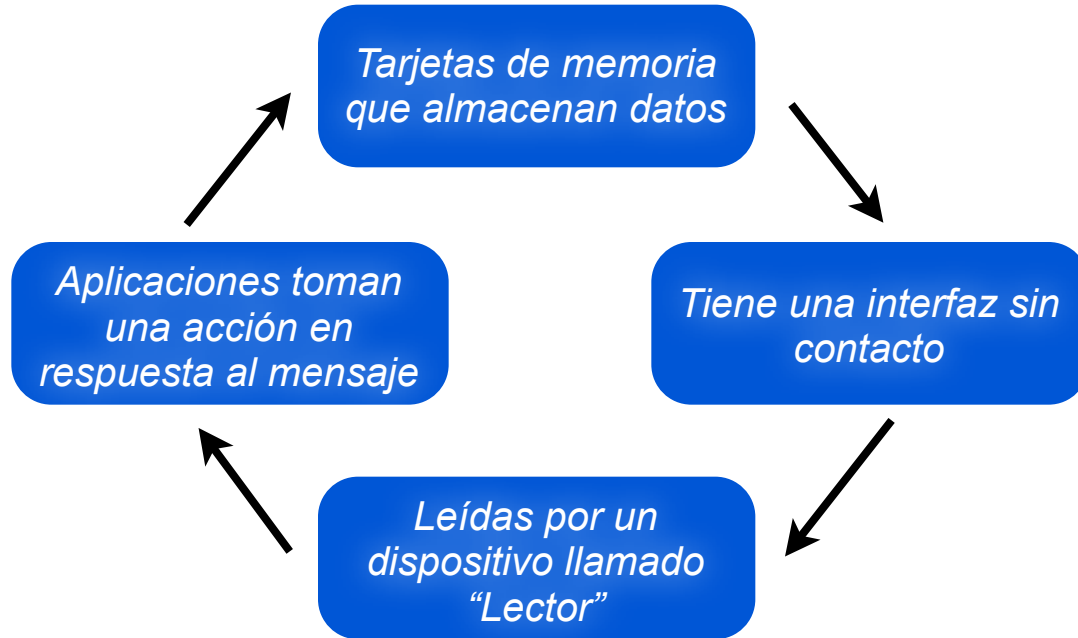


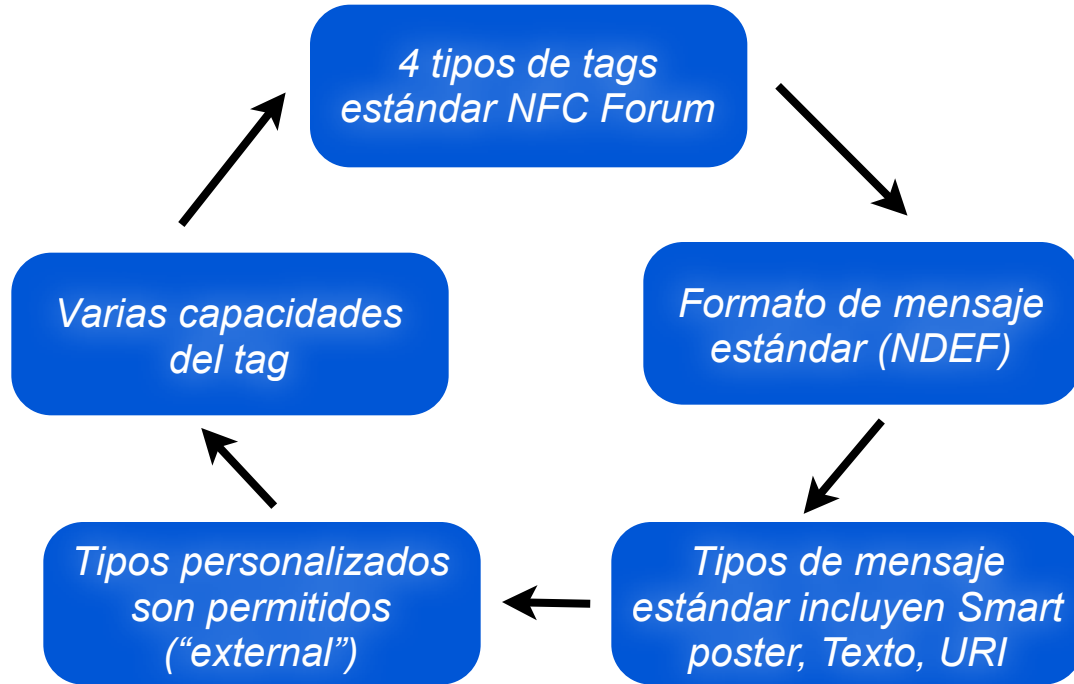
Para compartir los datos, alinee la parte posterior del dispositivo con otro dispositivo o etiqueta NFC.



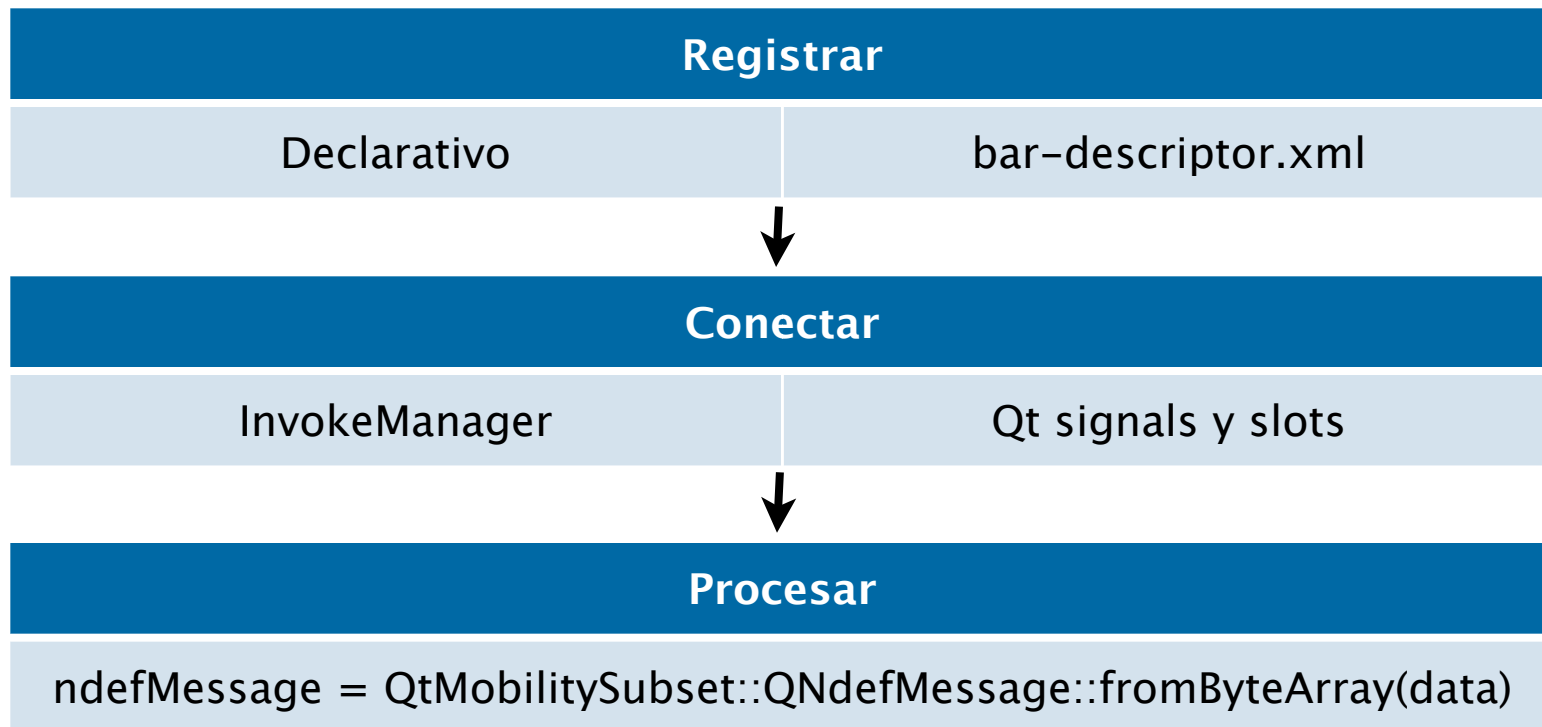
Lectura de NFC Tags

NFC Tags

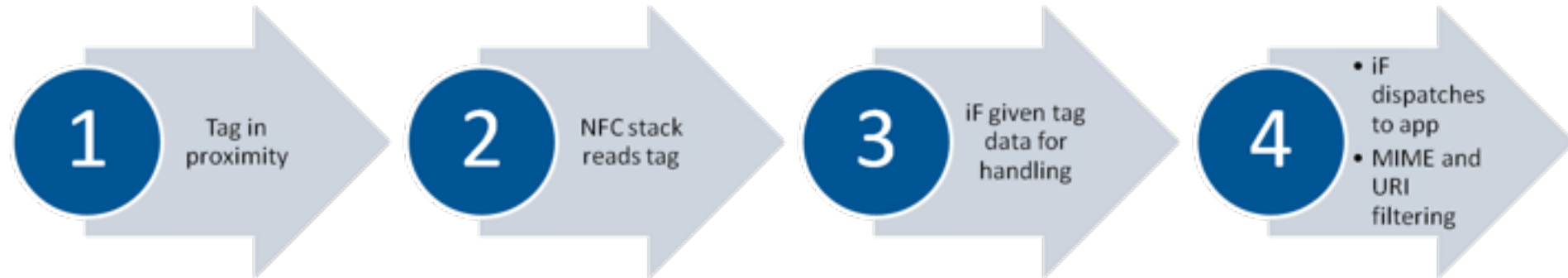




El Proceso de 3 Pasos



```
<invoke-target id="com.example.NfcRaceTime">
  <require-source-permissions>invoke_across_perimeters</require-
source-permissions>
  <type>APPLICATION</type>
  <filter>
    <action>bb.action.OPEN</action>
    <mime-type>application/vnd.rim.nfc.ndef</mime-type>
    <property var="uris" value="ndef://4/my.rim.com/myrecordtype"/
  >
  </filter>
</invoke-target>
```

```
bb::system::InvokeManager* _invokeManager =  
    new bb::system::InvokeManager();  
  
QObject::connect(_invokeManager,  
    SIGNAL(  
        invoked(const bb::system::InvokeRequest&)),  
        this,  
    SLOT(  
        receivedInvokeRequest(  
            const bb::system::InvokeRequest&)));
```

```
// SLOT method
void App::receivedInvokeRequest(
    const bb::system::InvokeRequest& request) {

    QByteArray data = request.data();

    QtMobilitySubset::QNdefMessage ndefMessage =
        QtMobilitySubset::QNdefMessage::fromByteArray(data);

    handleNdefRequest(ndefMessage);
}
```

```
// look for NDEF records of our special type
QList<QtMobilitySubset::QNdefRecord>::const_iterator ndefRecord;

for ( ndefRecord = ndefMessage.begin(); ndefRecord !=
ndefMessage.end(); ndefRecord++) {

    if (ndefRecord->typeNameFormat() ==
        QtMobilitySubset::QNdefRecord::ExternalRtd) {

        if (QString(
            ndefRecord->type()).compare("my.rim.com:myrecordtype") == 0 ) {
            emit raceTagDetected(QString(ndefRecord->payload()));
        }
    }
}
```

NFC Modo Punto a Punto

Transferir datos de un dispositivo a otro

Sin Gestos

Comparte

Interactivo

Comparte solo tocando ambos dispositivos

Crea experiencias interactivas/físicas

SNEP

*Protocolo de
intercambio simple de
NDEF*

*Orientado a
mensajes*

Higher level

LLCP

*Protocolo de control
lógico de link*

*Orientado a
conexiones*

Lower level

Invocation Framework

Menos código

*Otra app proporciona
el servicio a la tuya*

*Puede proveer UI
para el usuario*

Uso directo de la API

*Algunas API son
llamadas en
iteraciones de eventos*

*Todas las opciones
disponibles para tu
app*

*Control completo
encima del UI*


```
Button {
  id: shareButton
  text: "Share vCard - Press and Hold!"
  contextActions: [
    ActionSet {
      title: "Share NDEF Data"
      InvokeActionItem {
        id: sharedNdefData
        query {
          mimeType: "application/vnd.rim.nfc.ndef"
          invokeActionId: "bb.action.SHARE"
        }
      }
    }
  ]
}
```

```
Button {  
    ....  
    contextMenuHandler: ContextMenuHandler {  
        onShowing: {  
            // make NDEF message as byte array in C++  
            sharedNdefData.data =  
                _ndefFactory.getNdefVcardMessage(sendVcard.ndefFirstName,  
                sendVcard.ndefLastName, sendVcard.ndefAddress,  
                sendVcard.ndefEmail, sendVcard.ndefMobile);  
        }  
    }  
}
```

4 Demo

5 Minutos

Gracias por su atención

¿Preguntas?

@flashinstructor

<http://blackberrydeveloper.mx>

joelortiz@blackberrydeveloper.mx