

UNIVERSIDAD NACIONAL DEL SANTA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas e Informática



MANUAL:

MANUAL DE DEPLOYMENT

con consideraciones de seguridad para producción

IX CICLO – III UNIDAD

SEMESTRE 2025 – 01

Asignatura:

Aplicaciones Móviles.

Alumno:

VASQUEZ RAMOS, Jose Manuel.

Docente:

Ms. Johan Max Alexander LOPEZ HEREDIA.

Nuevo Chimbote – Perú

Agosto, 2025

Manual de Deployment

1. Requisitos Previos

| Elemento | Versión Recomendada |
|------------------------|---------------------|
| Android Studio | Narwhal o superior |
| SDK mínimo | API 29 (Android 10) |
| JDK | JDK 17 |
| Gradle Wrapper | Incluido |
| Emulador o dispositivo | Android 10+ |

2. Clonar el Proyecto

Desde Android Studio:

1. Ve a **File > New > Project from Version Control...**
2. En el campo de URL, coloca:

```
https://github.com/josevasquezramos/seguridad_priv_a.git
```

3. Presiona **Clone** y espera que se configure el proyecto.

O por terminal:


```
git clone https://github.com/josevasquezramos/seguridad_priv_a.git
cd seguridad_priv_a
```

3. Configurar Android Studio

1. Asegúrate de que el **SDK Manager** tenga instalado Android 10+ (API 29+).
2. Verifica que `gradle.properties`, `build.gradle` y `local.properties` no expongan secretos.
3. Android Studio detectará automáticamente los módulos y sincronizará Gradle.

4. **Compilación y Pruebas**

Ejecutar en modo Debug:

- Usa el botón  **Run** para ejecutar en emulador o dispositivo.
- Verifica que las actividades respondan correctamente a los permisos y simulaciones.

Verificación de Logs de Seguridad:

- Ve a la actividad **Protección de Datos**.
- Asegúrate de que los accesos estén siendo registrados correctamente.

5. **Consideraciones de Seguridad**

Permisos en Tiempo de Ejecución

Cada actividad maneja sus permisos individualmente con:

- Solicitud explicativa (`shouldShowRequestPermissionRationale`)
- Control UI según estado de permisos
- Registro en logs de accesos

Encriptación de Datos

- Usa `EncryptedSharedPreferences` con AES-256-GCM.
- Firma HMAC para integridad.
- Derivación de clave por usuario (PBKDF2).

Autenticación Biométrica

- Implementada con `BiometricPrompt`.
- Requerida para acceder a logs y gestión de datos.
- Alternativa: credenciales del dispositivo.

Rotación de Clave Criptográfica

- Se rota automáticamente cada 30 días.
- Se re-encriptan los datos almacenados.
- Se registra la rotación como log de seguridad.

Anonimización

- Se anonimizan datos de contactos, llamadas, ubicaciones, imágenes.

Forense y Evidencias

- Evidencias de seguridad se almacenan localmente y tienen cadena de custodia.
- Logs en blockchain local (hash SHA-256).

6. Generar APK para Producción

Paso 1: Configura Firma

1. Ve a **Build > Generate Signed Bundle / APK...**
2. Elige **APK > Next**
3. Selecciona un **keystore seguro**:
 - O crea uno con Android Studio.
 - Protege este archivo y no lo subas a git.
4. Completa:
 - **Key alias**
 - **Password**
 - **Destination folder**
 - **Build type: release**

Paso 2: ProGuard / R8








Asegúrate de que esté activo en build.gradle:

```
gradle
CopiarEditar
buildTypes {
    release {
        minifyEnabled true
        shrinkResources true
        proguardFiles getDefaultProguardFile(
            'proguard-android-optimize.txt'),
            'proguard-rules.pro'
    }
}
```

Paso 3: Generar APK

- Presiona **Finish**.
- El APK estará en:
app/build/outputs/apk/release/app-release.apk

7. Checklist de Seguridad Final

| Componente | Estado |
|---------------------------------|--|
| Solicitud explícita de permisos |  Implementada |
| Almacenamiento seguro local |  AES-256-GCM |
| Logs de acceso |  Registrados |
| Autenticación biométrica |  Obligatoria |
| Anonimización de datos |  Aplicada |
| Control de sesión (timeout) |  Implementado |
| Firma y ofuscación del APK |  Configurada |