Evidencia de Aprendizaje – taller acerca de integración, tecnologías emergentes y disruptivas

GA8-220501096-AA2-EV01

Fase – 3 Ejecución – Área técnica

Por:

Santiago Arango Rodriguez

Centro de la Tecnología del Diseño y la Productividad Empresarial

Regional Cundinamarca – SENA - Girardot

Análisis y Desarrollo de Software

Ficha 2977481

Instructor: Milton Ivan Barbosa

6 de octubre del 2025

Contenido

Imágenes

1. Introducción

En este taller se desarrolló una aplicación en App Inventor enfocada a la interfaz de un prototipo movil con un sistema básico de registro e inicio de sesión de usuarios, como parte del aprendizaje sobre el funcionamiento de las interfaces, almacenamiento de datos y flujo lógico de una app.

Durante el proceso se construyeron pantallas funcionales que permiten registrar nuevos usuarios, guardar sus datos localmente y posteriormente iniciar sesión para acceder a una pantalla principal donde se muestran sus datos personales y el saldo disponible.

El objetivo principal de este ejercicio fue fortalecer las habilidades prácticas en el uso de componentes visuales, manejo de eventos y almacenamiento con Firebase, comprendiendo cómo se estructura un flujo de autenticación dentro de una aplicación Android.

2. Justificación

El desarrollo de esta aplicación permitió poner en práctica conceptos fundamentales del diseño de software en entornos visuales como App Inventor, donde la lógica se representa mediante bloques en lugar de código textual. A través de este taller se buscó entender de forma más clara cómo se gestionan los datos de los usuarios, cómo se aplican condiciones lógicas y de validación, y cómo se enlazan las diferentes pantallas dentro de un proyecto móvil.

Este tipo de ejercicios son muy útiles en la formación técnica, ya que ayudan a relacionar la teoría con la práctica y a comprender los pasos que sigue un sistema real al procesar el

registro y la autenticación de usuarios. Además, favorece el desarrollo del pensamiento lógico y la capacidad de resolución de problemas, aspectos esenciales en la carrera de análisis y desarrollo de software.

3. Objetivos

3.1 Objetivo general

Desarrollar una aplicación móvil en App Inventor que permita registrar, almacenar y autenticar usuarios mediante un sistema de inicio de sesión funcional, aplicando principios básicos de diseño de interfaces, manejo de datos y lógica de programación.

3.2 Objetivos específicos

1. Diseñar las pantallas de registro, inicio de sesión y menú principal utilizando componentes gráficos adecuados para una experiencia de usuario clara e intuitiva.
2. Implementar el almacenamiento local de información mediante el componente TinyDB, garantizando la correcta gestión de los datos de los usuarios registrados.
3. Aplicar condiciones lógicas que permitan validar el acceso de los usuarios y mostrar información personalizada tras el inicio de sesión exitoso.

4. Sección 1 Taller

4.1 ¿Qué es Android?

Android es un sistema operativo de código abierto diseñado principalmente para dispositivos móviles (teléfonos, tablets, algunos wearables y televisores). Se basa en un núcleo Linux modificado y provee una plataforma completa para desarrollar, ejecutar y distribuir aplicaciones.

En términos prácticos: Android es la capa que permite que una app use el hardware (pantalla, cámara, sensores) y los servicios (notificaciones, almacenamiento, red) del dispositivo.

En un tono más personal: pienso en Android como la «base» que hace que mi juego, aplicación o herramienta pueda comunicarse con el teléfono; sin Android la app no podría sonar, mostrar interfaces ni acceder a Internet.

4.2 ¿Qué es un APK?

APK significa **Android Package** (paquete Android). Es el archivo que contiene una aplicación Android lista para instalar: incluye el código compilado, recursos (imágenes, xml, sonidos), y el manifiesto que describe permisos y componentes.

Puntos clave:

- Un APK es un archivo empaquetado (similar a un .zip) que el sistema Android instala.
- Debe estar firmado digitalmente por el desarrollador para que el sistema lo acepte.
- Actualmente, además de APK existe el formato **AAB (Android App Bundle)** que Google Play utiliza para generar APKs optimizados por dispositivo en el momento de la descarga (reduce tamaño de descarga para el usuario).

4.3 ¿Qué es el Android SDK?

El Android SDK (Software Development Kit) es el conjunto de herramientas, bibliotecas y utilidades que necesitas para desarrollar aplicaciones Android. Incluye:

- **APIs y bibliotecas** para interactuar con Android.
- **Build-tools** (herramientas de compilación) y el compilador.
- **Emuladores** (AVD) para probar apps sin un dispositivo físico.
- **Platform-tools** como adb (Android Debug Bridge) para depuración y transferencia de archivos.
- **SDK Manager**, que permite instalar diferentes versiones de la plataforma y componentes.

El SDK es la caja de herramientas oficial que permite escribir, compilar, depurar y probar aplicaciones.

4.4 ¿Cuál es el lenguaje utilizado para desarrollar aplicaciones en Android?

Hoy en día los lenguajes principales son:

- **Kotlin**: lenguaje moderno, expresivo y oficial recomendado por Google desde 2019. Es conciso y reduce muchos errores comunes de Java.
- **Java**: histórico y ampliamente usado; todavía soportado y común en proyectos grandes o heredados.
- **C/C++**: se usan con el **NDK (Native Development Kit)** cuando se necesita rendimiento nativo crítico (por ejemplo en motores de juegos o procesamiento intensivo).
- **Otros enfoques multiplataforma**: frameworks como Flutter (Dart), React Native (JavaScript/TypeScript), Xamarin (C#) permiten desarrollar apps que corren en Android y otras plataformas a partir de un solo código.

4.5 ¿Qué IDEs de desarrollo existen para codificar?

Los entornos de desarrollo más relevantes:

- **Android Studio**: IDE oficial (basado en IntelliJ IDEA). Ofrece integración total con el SDK, emuladores, herramientas de profiling, y soporte nativo para Kotlin y Java.
- **IntelliJ IDEA**: la versión completa de JetBrains es compatible y comparte muchas características con Android Studio.
- **Visual Studio + Xamarin**: para desarrollo en C# con integración multiplataforma.
- **Visual Studio Code**: editor ligero, usado con extensiones y herramientas para Flutter, React Native, o comandos CLI; no es un IDE completo pero es muy popular.
- **Unity / Unreal Engine**: para desarrollo de videojuegos que se exportan a Android.
- **Eclipse (histórico)**: antiguamente se usaba mucho; hoy en día está en desuso para desarrollo Android nativo.

4.6 Android multiusuario

Concepto: Android multiusuario permite que un mismo dispositivo sea usado por varias cuentas de usuario con perfiles independientes (aplicaciones y configuraciones separadas). Es útil en tablets familiares, dispositivos compartidos y entornos educativos.

Breve descripción:

- Cada usuario tiene su propio espacio de aplicaciones, configuraciones y almacenamiento (aunque hay excepciones para apps y datos compartidos administrativamente).
- Existen tipos de perfiles: usuario principal, usuarios adicionales y perfiles restringidos (útiles para control parental o kioscos).
- No todos los dispositivos o versiones de Android implementan exactamente las mismas capacidades multiusuario; en teléfonos suele estar más limitado que en tablets.

4.7 Principio de mínimo privilegio (mínimo privilegio)

Definición: El principio de mínimo privilegio establece que una aplicación, proceso o usuario debe tener solo los permisos absolutamente necesarios para realizar su función, y nada más.

Aplicación práctica en Android:

- Solicitar solo los permisos que la app necesita (por ejemplo: solicitar acceso a ubicación solo si la funcionalidad lo requiere).
- Evitar permisos peligrosos innecesarios y explicar al usuario por qué se piden.
- Diseñar la app para degradar funciones si el permiso no es concedido, en lugar de fallar por completo.

Beneficios: reduce la superficie de ataque, protege la privacidad del usuario y mejora la seguridad general del sistema.

4.8 Componentes esenciales de una aplicación Android

Una app Android típica está compuesta por varios tipos de componentes, cada uno con responsabilidades claras:

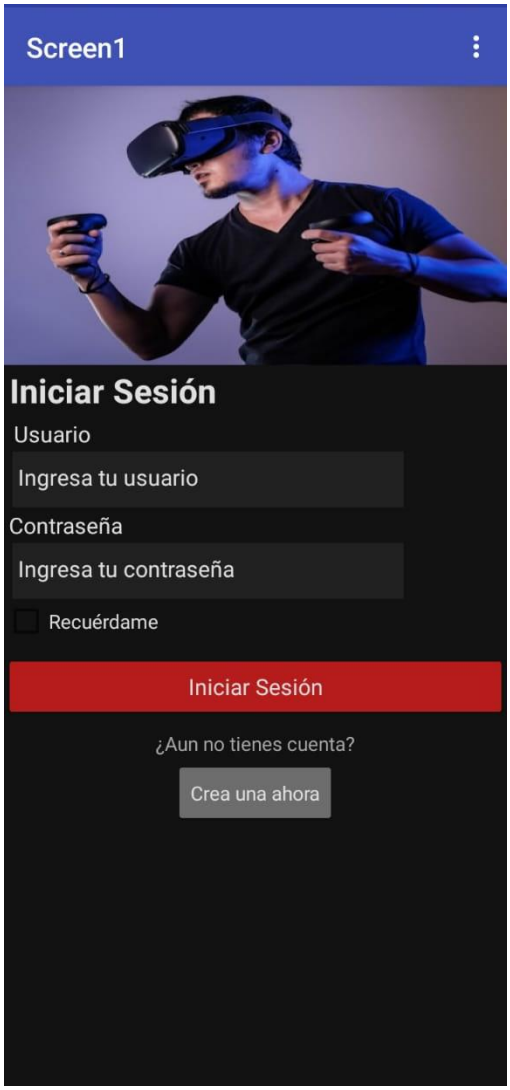
1. **Activities (Actividades):** pantallas o interfaces con las que interactúa el usuario. Una app puede tener múltiples actividades y un flujo entre ellas.
2. **Services (Servicios):** procesos que se ejecutan en segundo plano para tareas de larga duración (por ejemplo reproducir música o sincronizar datos).
3. **Broadcast Receivers (Receptores de difusión):** componentes que escuchan mensajes del sistema o de otras apps (por ejemplo: evento de batería baja, SMS recibido).
4. **Content Providers (Proveedores de contenido):** permiten compartir datos entre aplicaciones (por ejemplo: contactos, multimedia) mediante un contrato estándar.
5. **Resources (Recursos):** archivos estáticos como layouts XML, imágenes, cadenas de texto, estilos y temas.
6. **AndroidManifest.xml:** archivo esencial que declara los componentes de la app, permisos requeridos, actividades principales y otros metadatos.
7. **Gradle / Sistema de construcción:** scripts y configuraciones que definen cómo compilar y empaquetar la app (dependencias, variantes de build, firmas).
8. **Librerías y runtime:** bibliotecas externas (AndroidX, support libraries) y la máquina virtual (ART) que ejecuta el código.

5. Seccion 2 Taller Primera aplicación Android en MIT App Inventor

5.1 Flujo de la aplicación

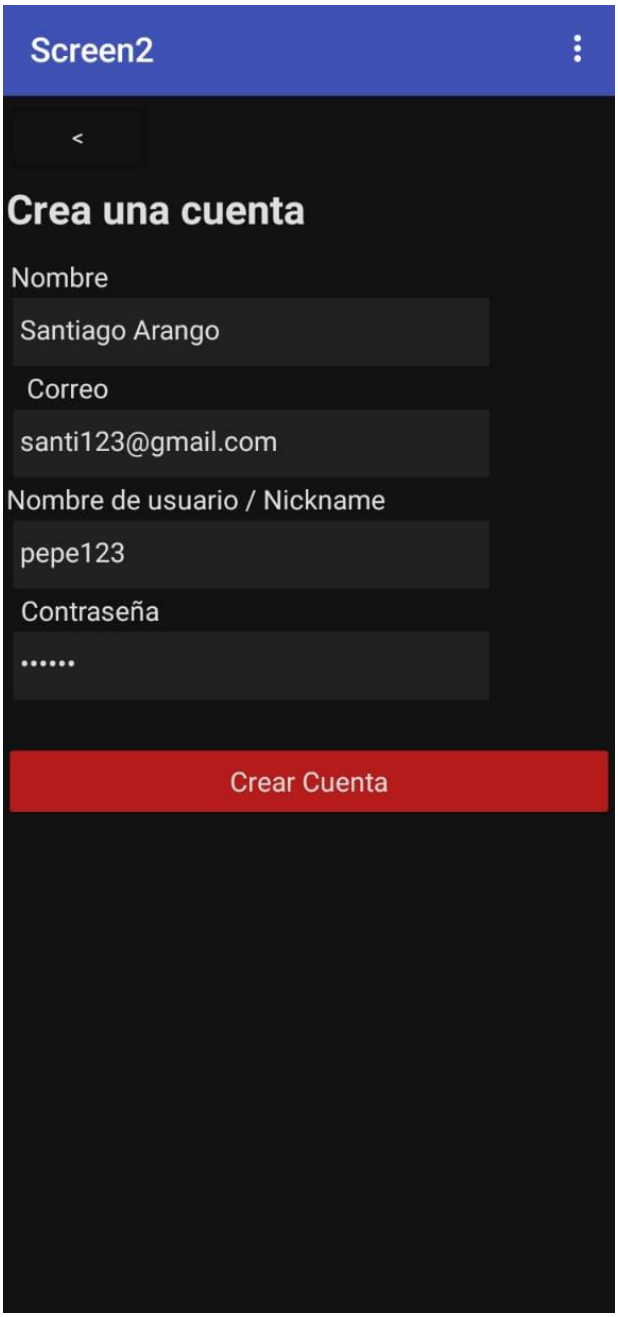
1. **Pantalla de inicio / Login**: muestra campos Usuario y Contraseña, checkbox Recuérdame, botón rojo Iniciar Sesión, y botón gris Crea una ahora para ir a registro.

Imagen 1 Captura de pantalla APK 1



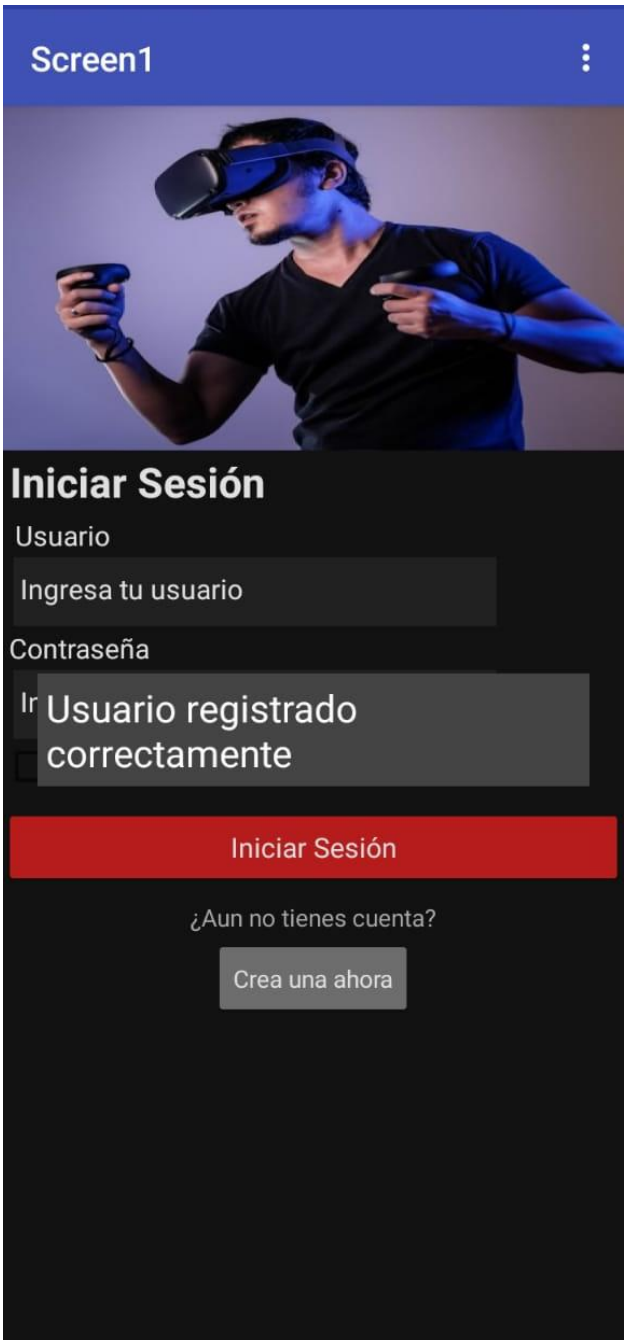
2. **Pantalla de registro**: campos Nombre, Correo, Nombre de usuario / Nickname, Contraseña y botón Crear Cuenta.

Imagen 2 Captura de pantalla APK 2



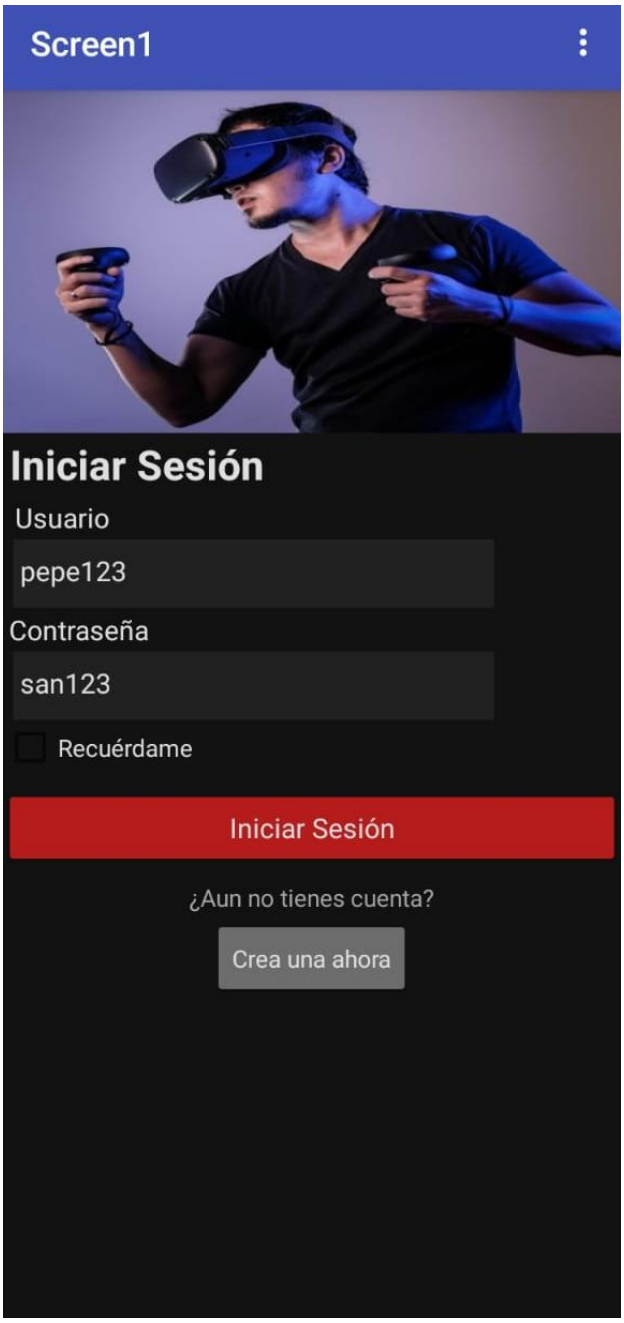
3. **Notificación**: al crear la cuenta aparece un mensaje flotante: *“Usuario registrado correctamente”*.

Imagen 3 Captura de pantalla APK 3



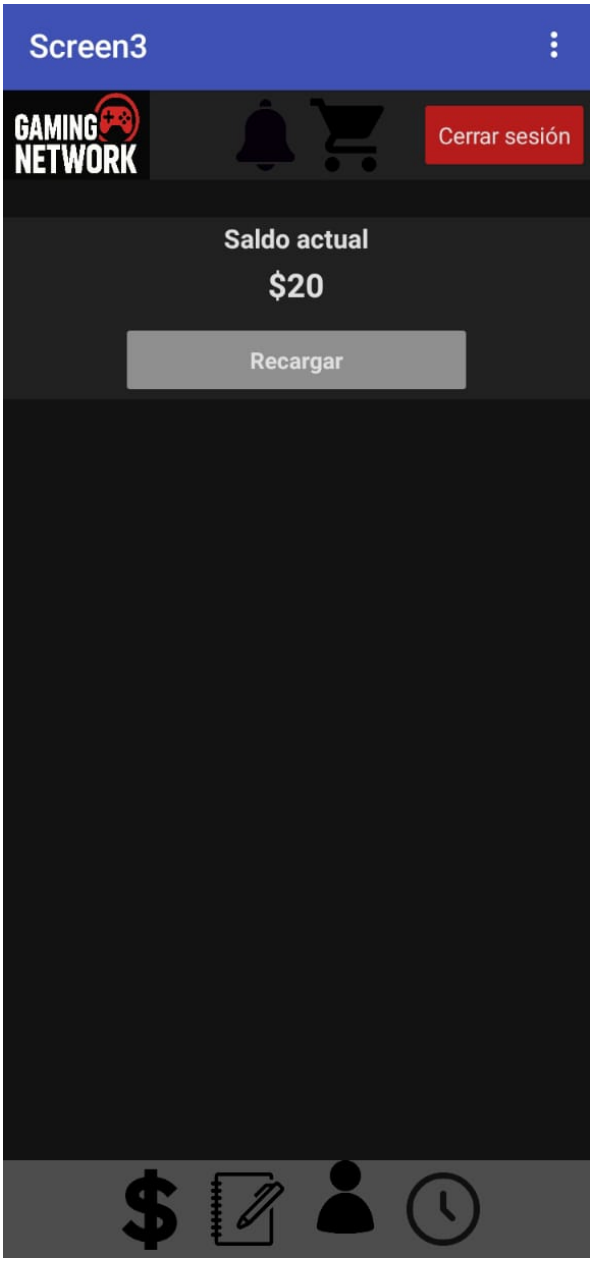
4. **Login con datos ingresados**: el usuario escribe pepe123 / san123 en los campos de login y pulsa Iniciar Sesión.

Imagen 4 Captura de pantalla APK 4



5. **Pantalla principal después del login:** Se abre la pantalla principal y se muestra algo básico como el saldo del cliente + un botón de Recargar, además de una barra de navegación (no funcional).

Imagen 5 Captura de pantalla APK 5



6. Conclusiones

El desarrollo de este taller permitió comprender de manera práctica cómo funcionan los procesos de registro e inicio de sesión dentro de una aplicación móvil creada en App Inventor. A través del uso de componentes visuales y del manejo de datos con TinyDB, se logró construir un flujo funcional que demuestra los principios básicos de persistencia de información y validación de usuarios.

Además, este ejercicio fortaleció las habilidades en el diseño de interfaces y en la organización lógica de los bloques, fomentando una mejor comprensión sobre cómo se estructuran los sistemas móviles. En general, la actividad representó una experiencia significativa para consolidar conocimientos esenciales en el desarrollo de software, conectando la teoría con la práctica y permitiendo visualizar cómo los conceptos aprendidos se aplican en escenarios reales.

7. Referencias

Developers, A. (2025). *Conceptos básicos de Android*. Obtenido de Google Developers: https://developer.android.com/guide

Developers, G. (2025). *Introducción al Android SDK*. Obtenido de Android Developers: https://developer.android.com/studio/intro

Inventor, M. A. (2024). *¿Qué es una APK y cómo funciona?* Obtenido de MIT App Inventor Documentation: https://appinventor.mit.edu/explore/resources

Project, A. O. (2024). *Estructura y componentes de una aplicación Android*. Obtenido de Android Source: https://source.android.com/docs

Team, A. I. (2025). *What is MIT App Inventor?* Obtenido de MIT App Inventor: https://appinventor.mit.edu