



Semana 5

## Desarrollo de Aplicaciones Móviles (DSY2204)

Formato de respuesta

<b>Nombre estudiante:</b> Katlheen Rodriguez	
<b>Asignatura:</b> Desarrollo de Aplicaciones Móviles	<b>Carrera:</b> Ingeniería en Desarrollo de Software
<b>Profesor:</b> MIGUEL EMILIO PUEBLA CUERO	<b>Fecha:</b> 15/09/2025

# Descripción de la actividad

En esta quinta semana, se realizará la actividad sumativa 2, llamada "Integrando Kotlin a la aplicación móvil con Android Studio", donde deberán integrar a la app creada en el framework Android Studio, las funcionalidades básicas y avanzadas del lenguaje de programación Kotlin, con una entrega de encargo que contenga la documentación del desarrollo y código del proyecto comprimido en ZIP.

## Instrucciones específicas

A continuación, se detallan las instrucciones y requisitos de la actividad sumativa 2, deberás leer cada uno de ellos y apuntar al cumplimiento.

### Características del producto:

- Adaptar al lenguaje de programación Kotlin la aplicación móvil de accesibilidad para personas con discapacidad sensorial visual desarrollada en la experiencia de aprendizaje 1.
- La aplicación móvil deberá ser desarrollada específicamente para personas con discapacidad sensorial visual, facilitando a los usuarios su comunicación (escribir y leer) en su entorno cotidiano, mejorando su calidad de vida al transformar sus dispositivos móviles en una herramienta que los integra al mundo real y digital.

### Requerimientos generales:

Para el desarrollo de la entrega solicitada considerar los siguientes aspectos:

- Implementación el lenguaje de programación Kotlin en cada componente UI de Material Design y Jetpack Compose integrado en entrega de la experiencia de aprendizaje 1 (input, botones, tablas, grillas, vínculos, textos, combo box, check list y radio buttons). Adaptar las views de Login, Registro y Recuperar contraseña al lenguaje de programación Kotlin.
- Generar un array en Kotlin que almacene los datos de 5 usuarios y sus contraseñas, registrados previamente desde el formulario de la view de Registro.

### Requerimientos técnicos

- Desarrollar la aplicación con el Framework Android Studio.

- Generar y compartir el repositorio Git del proyecto.
- Integrar el lenguaje de programación Kotlin al código fuente de la app móvil.

### **Entrega de encargo con documentación**

Esta actividad tendrá dos partes y se entregará todo en **un solo ZIP**.

**En la primera parte**, una vez desarrollado el proyecto desde Android Studio, se deberá comprimir la carpeta del código del proyecto completo en formato .zip desde Git y adjuntarlo en espacio indicado en el AVA.

**En la segunda parte**, se deberá documentar todo el proceso de desarrollo del proyecto, describiendo el rendimiento de las views diseñadas, restricciones y el seguimiento de los 5 riesgos técnicos del proyecto definidos en la experiencia de aprendizaje 1, además de los accionables de mitigación de cada riesgo definido. Todo ello **en este mismo documento de formato de respuesta, a continuación de estas instrucciones** y subirlo en el ZIP que entregarás con el desarrollo del Proyecto y este documento del formato de respuesta debe subirse en formato PDF.

- Entregar el producto totalmente finalizado, según el avance solicitado en esta entrega.
- No se aceptarán entregas fuera de plazo ni por otro medio que no sea AVA.
- Todos los productos desarrollados con IA serán evaluados con nota mínima.

### **Restricciones en esta entrega**

La aplicación móvil debe ser adaptativa a múltiples dispositivos Android que cuenten con una conexión activa a internet.

### **Formato del texto:**

- Fuente: Arial.
- Tamaño de fuente: 12 puntos.
- Interlineado: 1,5 líneas.
- Alineación del texto: a la izquierda.

**Iniciar documentación desde la siguiente página.**

# Datos del documento

## Histórico de revisiones

Versión	Fecha	Descripción/cambio	Autor
1.0	30/08/2025	Creación del documento inicial.	Katlheen Rodriguez
1.1	15/09/2025	Se organiza la aplicación en paquetes, se agrega página de bienvenida con cierre de sesión.	Katlheen Rodriguez

## Información del proyecto

Organización	DUOC
Sección	002A
Proyecto (nombre)	ChromaAssist
Fecha de inicio	30/08/2025
Fecha de término	15/09/2025
Patrocinador principal	DUOC
Docente	MIGUEL EMILIO PUEBLA CUERO

## ***Alcance del proyecto.***

*Problema, solución propuesta y contexto del proyecto.*

### **Problema:**

Las personas con daltonismo enfrentan dificultades para distinguir colores en su vida diaria, lo que puede afectar su aprendizaje, la interpretación de señales visuales, la navegación en entornos digitales y la interacción social. En aplicaciones móviles o sitios web, la falta de accesibilidad para daltónicos genera barreras en la comprensión de información esencial, reduciendo su autonomía y usabilidad de las tecnologías.

### **Solución propuesta:**

Se desarrollará una aplicación móvil inclusiva denominada ChromaAssist, diseñada específicamente para personas con daltonismo. La aplicación se centra en mejorar la accesibilidad y facilitar la interacción con interfaces digitales mediante los siguientes elementos:

**Interfaz con colores adaptados:** La aplicación utiliza un esquema de colores y botones accesibles para daltónicos, garantizando que la información visual sea clara y legible para todos los tipos de daltonismo.

**Pantallas de login, bienvenida y registro accesibles:** Cada pantalla cuenta con contraste elevado entre texto, fondos y botones, con tipografía legible y disposición intuitiva, minimizando confusiones visuales.

**Recuperación de contraseña y navegación simple:** Funciones como “Recuperar contraseña” están diseñadas para ser claras y directas, evitando depender únicamente de colores para transmitir información.

**Componentes visuales coherentes:** Iconos y botones cuentan con descripciones semánticas (contentDescription) que permiten la compatibilidad con lectores de pantalla, reforzando la accesibilidad.

**Navegación por secciones:** La app permite moverse entre login, bienvenida, registro y recuperación mediante navegación intuitiva (NavHost), asegurando que los usuarios comprendan dónde se encuentran y qué acciones realizar.

## Contexto del proyecto:

El proyecto surge como respuesta a la necesidad de promover la accesibilidad digital y la inclusión social en el uso de tecnologías móviles. Se enmarca dentro de un contexto académico, donde se busca aplicar conocimientos en desarrollo de software móvil y diseño inclusivo, con el objetivo de ofrecer una solución práctica que mejore la calidad de vida de un grupo de usuarios frecuentemente desatendido.

## ***Funciones esperadas del software a desarrollar.***

- **Pantalla de Inicio de Sesión (Login)**

Campos: Usuario y Contraseña.

Botón “Ingresar” con colores accesibles.

Validación de usuario y contraseña existentes.

Mensajes claros de error mediante Toast.

Iconos accesibles y semántica para lectores de pantalla.

Redirección a la pantalla de bienvenida en caso de login correcto.

- **Pantalla de Bienvenida**

Muestra mensaje de bienvenida al usuario.

Permite cerrar sesión, redireccionando al login.

- **Pantalla de Registro de Usuario**

Campos: Usuario, Contraseña y Confirmar Contraseña.

Validaciones:

Campos obligatorios.

Contraseñas coincidentes.

Máximo de 5 usuarios registrados.

Usuario no duplicado.

Botón “Registrar” con colores contrastantes.

Navegación de regreso al Login tras registro exitoso.

Botón “Atrás” para retroceder al login.

- **Pantalla de Recuperación de Contraseña**

Campo: Usuario.

Botón “Enviar” para mostrar contraseña si el usuario existe.

Mensajes de usuario no encontrado o contraseña mostrada.

Colores y tipografía accesibles.

Navegación Intuitiva

Flujo: Login ↔ Registro ↔ Recuperación de contraseña.

Botón “Atrás” para retroceder al login.

- **Interfaz Adaptada para Daltónicos**

Esquema de colores accesibles.

Tipografía legible (Arial, tamaño 14–20 sp).

Contraste adecuado entre fondo, texto e iconos.

Iconos con contentDescription para compatibilidad con lectores de pantalla.

### ***Rendimiento esperado del software a desarrollar.***

#### **Velocidad de respuesta:**

La aplicación debe cargar cada pantalla (Login, Registro, Recuperación, Bienvenida) en menos de 3 segundos en un dispositivo promedio Android (API 21+).

La navegación entre pantallas debe ser fluida, sin retrasos ni congelamientos perceptibles.

#### **Consumo de recursos:**

La app debe mantener un uso mínimo de CPU y memoria, compatible con dispositivos de gama media.

Optimización de imágenes y recursos gráficos para evitar sobrecarga.

#### **Compatibilidad:**

Funcionamiento correcto en dispositivos con Android 5.0 (API 21) en adelante.

Escalabilidad de la interfaz a distintos tamaños de pantalla (smartphones y tablets).

Mantener accesibilidad para usuarios con daltonismo y compatibilidad con lectores de pantalla.

### **1. Login Screen:**

Campos: Usuario, Contraseña.

Botones: Ingresar, Registrarse, Recuperar contraseña.

Funcionalidad:

Validar credenciales.

Mostrar mensajes de error/success con Toast.

Navegar a Registro o Recuperación.

Accesibilidad: Colores contrastantes, tipografía legible, iconos con contentDescription.

### **2. Register Screen:**

Campos: Usuario, Contraseña, Confirmar Contraseña.

Botón: Registrar.

Funcionalidad:

Validar campos obligatorios.

Verificar coincidencia de contraseñas.

Evitar duplicados y límite de usuarios.

Redirigir a Login tras registro exitoso.

Accesibilidad: Colores y contraste adaptados para daltónicos, tipografía consistente.

### **3. Recover Password Screen**

Campo: Usuario.

Botón: Enviar.



Funcionalidad:

Buscar usuario existente.

Mostrar contraseña mediante Toast o mensaje de error.

Navegación simple de regreso al Login.

Accesibilidad: Contraste de colores y semántica para lectores de pantalla.

#### **4. Welcome Screen**

Funcionalidad:

Dar la bienvenida al usuario cuando el login es correcto.

Cierre de sesión simple de regreso al Login.

#### ***Restricciones del proyecto.***

##### **Tecnológicas:**

La aplicación se desarrollará únicamente para Android, con compatibilidad desde API 21 (Android 5.0) en adelante.

Se usará Kotlin y Jetpack Compose como framework principal de UI.

El proyecto estará limitado a uso local, sin integración de bases de datos remotas o nube por el momento. Se usarán únicamente recursos gráficos optimizados, evitando imágenes pesadas que afecten el rendimiento.

##### **Tiempo:**

El desarrollo, pruebas y documentación deberán completarse dentro del plazo establecido del proyecto académico.

Las fases de implementación de nuevas funcionalidades están limitadas a la planificación inicial, evitando cambios mayores fuera del alcance.

##### **De diseño y accesibilidad:**

Los colores y contrastes de la interfaz están restringidos a paletas seguras para personas con daltonismo.

La tipografía y tamaño de texto deben ser consistentes y legibles en todas las pantallas.

Todos los elementos interactivos deben tener descripciones accesibles (contentDescription) para compatibilidad con lectores de pantalla.

#### **De funcionalidad:**

No se incluirán funcionalidades avanzadas de sincronización, notificaciones o almacenamiento en la nube.

Solo se permitirá un máximo de 5 usuarios registrados de manera local para pruebas.

#### **De hardware y dispositivos:**

El software debe funcionar en dispositivos de gama media y baja, con memoria limitada y resolución mínima de 720p.

No se garantiza soporte para emuladores con aceleración de hardware deshabilitada.

#### **De seguridad:**

La aplicación no implementa cifrado de contraseñas ni autenticación segura en esta versión; los datos se almacenan en memoria mientras la app está abierta.

No se permite el acceso externo ni la exportación de credenciales por el momento.

#### ***Estructura de desglose de trabajo.***

*Definición de Diagrama y diccionario EDT.*

#### **Diagrama EDT:**

**Nivel 0:** Proyecto ChromaAssist – App para daltónicos

#### **Nivel 1:**

1. Planificación del proyecto
2. Análisis y diseño
3. Desarrollo de la aplicación
4. Pruebas y aseguramiento de calidad
5. Documentación

## **Nivel 2:**

### **1. Planificación del proyecto**

- 1.1 Definición de alcance
- 1.2 Cronograma y asignación de recursos
- 1.3 Gestión de riesgos

### **2. Análisis y diseño**

- 2.1 Investigación sobre daltonismo y accesibilidad
- 2.2 Diseño de interfaces accesibles (mockups)
- 2.3 Selección de paleta de colores y tipografía
- 2.4 Especificación de funcionalidades

### **3. Desarrollo de la aplicación**

- 3.1 Configuración del proyecto Android Studio
- 3.2 Implementación de la pantalla de Login
- 3.3 Implementación de la pantalla de Registro
- 3.4 Implementación de la pantalla de Recuperación de Contraseña
- 3.5 Integración de navegación entre pantallas
- 3.6 Aplicación de paleta y contrastes accesibles
  
- 3.7 Implementación de la pantalla de Bienvenida
- 3.8 Estructuración del proyecto en paquetes (organización modular del código)

### **4. Pruebas y aseguramiento de calidad**

- 4.1 Pruebas unitarias de componentes
- 4.2 Pruebas de navegación
- 4.3 Pruebas de accesibilidad y contraste
- 4.4 Corrección de errores

### **5. Documentación**

- 5.1 Manual de usuario
- 5.2 Documentación técnica
- 5.3 Presentación final

### Cambios respecto a la primera entrega:

- Se agregó la pantalla de Bienvenida (3.7).
- Se estructuró el proyecto en paquetes para mejorar la organización del código (3.8).

### Diccionario EDT

Código	Actividad	Descripción	Responsable	Duración Estimada
1	Planificación del proyecto	Organización inicial, definición de alcance y cronograma	Equipo	1 día
1.1	Definición de alcance	Documentar objetivos y funcionalidades	Líder de proyecto	1 día
1.2	Cronograma y recursos	Asignación de tareas y tiempos	Líder de proyecto	1 día
1.3	Gestión de riesgos	Identificación de posibles problemas	Equipo	1 día
2	Análisis y diseño	Investigación y diseño de interfaces	Equipo de UX/UI	1 día
2.1	Investigación sobre daltonismo	Recopilación de necesidades de usuarios daltónicos	UX	1 día
2.2	Diseño de interfaces	Crear mockups accesibles	UX/UI	1 día
2.3	Selección de paleta y tipografía	Garantizar accesibilidad y contraste	UX/UI	1 día

2.4	Especificación de funcionalidades	Listado de vistas y botones	Equipo	1 día
3	Desarrollo de la aplicación	Codificación de pantallas y lógica	Desarrollador	1 día
3.1	Configuración del proyecto	Android Studio + Kotlin + Compose	Dev	1 día
3.2	Pantalla de Login	Login con validación de usuario	Dev	1 día
3.3	Pantalla de Registro	Registro local de usuarios	Dev	1 día
3.4	Pantalla de Recuperación	Recuperación de contraseña	Dev	1 día
3.5	Integración de navegación	Navegación entre pantallas	Dev	1 día
3.6	Aplicación de paleta accesible	Contrastes y colores daltonicos	UX/UI	1 día
4	Pruebas y QA	Verificación de funcionamiento y accesibilidad	QA	1 día
4.1	Pruebas unitarias	Test de componentes	QA	1 día
4.2	Pruebas de navegación	Revisar flujo entre pantallas	QA	1 día
4.3	Pruebas de accesibilidad	Contraste, legibilidad, contentDescription	QA	1 día

4.4	Corrección de errores	Ajustes finales post-testing	Dev	1 día
5	Documentación	Elaboración de manuales y reportes	Equipo	1 día
5.1	Manual de usuario	Guía para operar la app	Equipo	1 día
5.2	Documentación técnica	Explicación de código y arquitectura	Dev	1 día
5.3	Presentación final	Preparar material para evaluación	Equipo	1 día
6	Cambios solicitados	Aplicar cambios para mejorar la aplicación.	Desarrollador	1 día
6.1	Estructura del proyecto.	Reorganización de la estructura del proyecto en paquetes para mejorar modularidad y mantenibilidad del código.	Dev	1 día
6.2	Pantalla de Bienvenida	Implementación de pantalla de Bienvenida que se muestra al usuario después de un login exitoso.	Dev	0.5 día

### **Resumen de riesgos**

*Listado de riesgos relacionado al desarrollo de S.W. Indicar riesgo, etapa o fase en que se presenta, la probabilidad de que ocurra, magnitud o impacto de este riesgo por etapa en el proceso.*

**Probabilidad:** alta, media, baja.

**Impacto:** alto, significativo, moderado, inferior y bajo.

<b>Riesgo</b>	<b>Fase</b>	<b>Probabilidad</b>	<b>Impacto</b>	<b>Acción de mitigación</b>
Problemas de compatibilidad con dispositivos de API 21	Desarrollo / Pruebas	Media	Alto	Probar la aplicación en emuladores y dispositivos reales con API 21 desde etapas tempranas.
Errores en la navegación entre pantallas (login, registro, recuperación, bienvenida)	Desarrollo / Integración	Media	Alto	Realizar pruebas unitarias y de flujo de usuario, corregir errores detectados durante cada sprint.
Paleta de colores no adecuada para todos los tipos de daltonismo	Diseño / Validación	Baja	Alto	Revisar y ajustar colores según estándares de accesibilidad y realizar pruebas con usuarios con daltonismo.
Fallos de registro o recuperación de usuarios (pérdida de datos)	Desarrollo / Pruebas	Baja	Alto	Implementar validaciones de entrada y pruebas, respaldos de datos durante la fase de pruebas.

Conflictos o errores por reorganización de paquetes en el proyecto	Desarrollo / Integración	Media	Significativo	Revisar dependencias, probar módulos de forma independiente y realizar integración progresiva.
--	--------------------------	-------	---------------	--

### **Definición de artefactos.**

*Listar y describir los artefactos o entregables que serán administrados y entregados durante el desarrollo del proyecto.*

<b>Artefacto</b>	<b>Descripción</b>
Código fuente	Proyecto Android Studio en Kotlin con Jetpack Compose, estructurado en paquetes.
Aplicación APK	Compilación final instalable en dispositivos Android.
Documentación del proyecto	Manual de usuario, documentación técnica y presentación final.

### **Condiciones de aceptación para cierre del proyecto**

*Condiciones que se deben cumplir para dar termino al proyecto y margen de tolerancia de aceptación de defectos.*

### **Funcionalidad completa de la aplicación**

La app debe incluir las pantallas de Login, Bienvenida, Registro y Recuperación de contraseña funcionando correctamente.

La navegación entre pantallas debe ser fluida y sin errores.

Todos los botones y campos de texto deben ser accesibles y responder a la interacción del usuario.

### **Accesibilidad para daltónicos**

La paleta de colores y los contrastes deben permitir que personas con daltonismo distingan los elementos de la interfaz.

Todos los elementos interactivos deben incluir contentDescription para soporte de tecnologías de asistencia.



**Margen de tolerancia de defectos**

Se permite hasta 5% de defectos menores, tales como ajustes estéticos o mejoras de UX, siempre que no afecten la funcionalidad principal ni la accesibilidad.

0% de defectos críticos: errores que impidan iniciar sesión, registrar usuarios o recuperar contraseñas no serán aceptables.

**Entrega final**

Código subido a repositorio y funcional en Android Studio.

APK generada y verificada en dispositivos con API 21.



Reservados todos los derechos Fundación Instituto Profesional Duoc UC. No se permite copiar, reproducir, reeditar, descargar, publicar, emitir, difundir, de forma total o parcial la presente obra, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio (electrónico, mecánico, fotocopia, grabación u otros) sin autorización previa y por escrito de Fundación Instituto Profesional Duoc UC. La infracción de dichos derechos puede constituir un delito contra la propiedad intelectual.