

EVALUACIÓN

HERRAMIENTAS
PROGRAMACIÓN MOVIL
Semana 8

DE

Michel Brevis
07-12-2024

Técnico en Análisis y
Programación Computacional



DESARROLLO:

Eres un desarrollador de aplicaciones móviles por la empresa Desarrollos Full, la cual realiza diversas aplicaciones a medida. Uno de sus clientes cuenta con una galería de arte, la cual quiere exponer ciertas obras al público general, es por esto que se han fijado los siguientes requerimientos para la aplicación:

- El nombre del proyecto debe ser su **nombre_apellido_fecha_seccioncurso** (la fecha en formato yyyyymmdd).
- La aplicación debe ser desarrollada en lenguaje Kotlin (aplicativo en lenguaje Java no será considerado).
- La aplicación debe ser capaz de mostrar, al menos, cinco imágenes de la galería de arte.
- La aplicación tendrá la opción de mostrar una miniatura de las imágenes de la galería de arte directamente desde la pantalla principal del equipo móvil (celular).

De acuerdo con lo planteado, a continuación, realiza la siguiente actividad:

1. Realiza una tabla comparativa donde consideres las características más importantes entre tareas en segundo plano, notificaciones y AppWidgets. Además, indica y justifica cuál es el adecuado para la situación planteada.

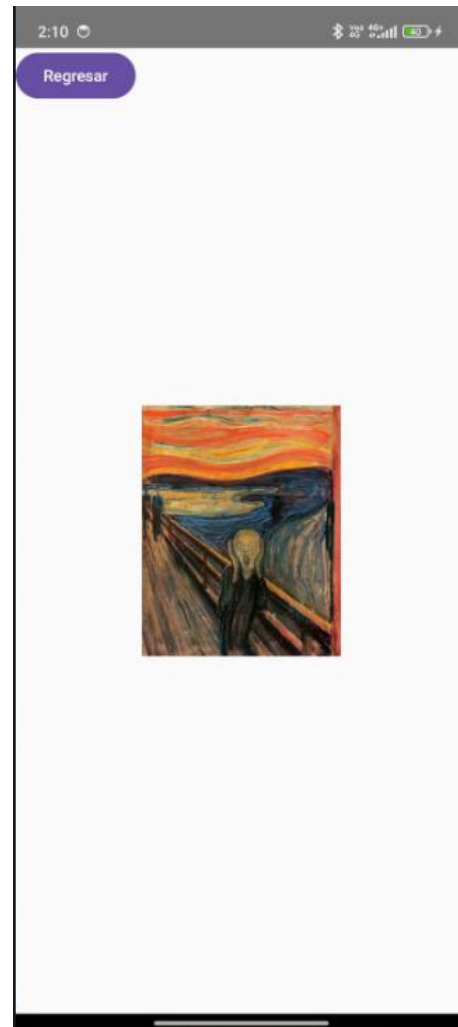
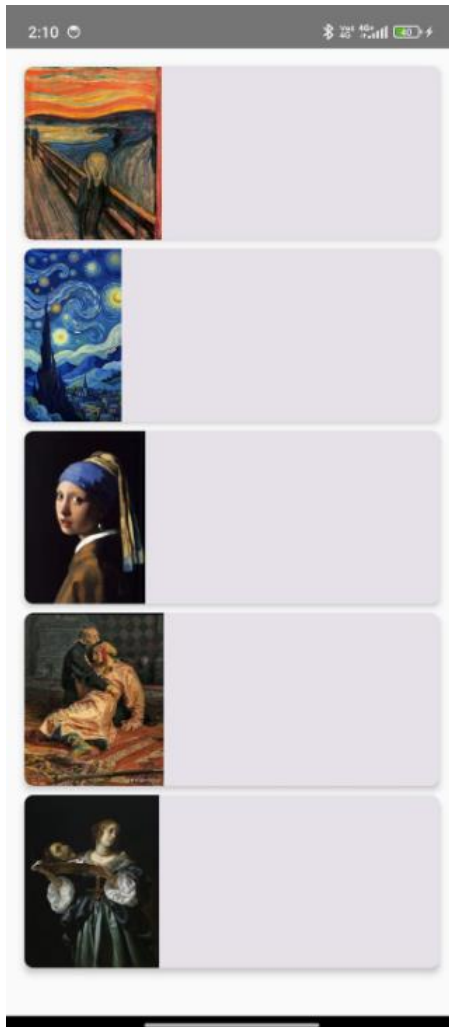
Característica	Tareas en Segundo Plano	Notificaciones	AppWidgets
Propósito	Ejecutar procesos que no requieren interacción del usuario	Informar al usuario de eventos o cambios en la aplicación	Mostrar información persistente y accesible desde el inicio del dispositivo
Interacción del Usuario	Generalmente no interactiva	Interactiva (puede incluir botones y acciones rápidas)	Interactiva (puede incluir botones o gestos básicos)
Persistencia en la Interfaz	Invisible para el usuario	Temporal (hasta que el usuario interactúa o la descarta)	Permanente mientras el widget esté en la pantalla de inicio
Uso de Recursos	Consume recursos dependiendo de la tarea ejecutada	Consume pocos recursos	Consume más recursos (para actualizaciones frecuentes o complejas)
Ventajas	Ideal para sincronizaciones o tareas largas	Notifica eventos importantes de manera inmediata	Acceso rápido a información clave sin abrir la app
Limitaciones	Difícil de monitorear por el usuario	Pueden ser ignoradas o descartadas	Limitado en espacio y funcionalidad
Ejemplo de Uso Común	Descargar archivos, sincronización en segundo plano	Recordatorios, mensajes nuevos	Mostrar el clima, agenda diaria o actualizaciones rápidas

Según lo desarrollado, el recurso que usaría para la aplicación de la galería de imágenes serían las notificaciones, destacando lo siguiente:

- La aplicación muestra imágenes en miniatura en la pantalla principal y permite navegar a una vista de pantalla completa al seleccionar una miniatura.
- Si se quisiera implementar una función adicional para notificar al usuario sobre nuevas imágenes disponibles o recordarle visitar la galería, **notificaciones** serían la elección adecuada.
- Las notificaciones pueden incluir acciones rápidas, como "Ver imagen destacada", y redirigir directamente a la galería o a una imagen específica.

2. Desarrolla la aplicación solicitada basada en la elección y justificación de la pregunta anterior

Capturas de la APP desarrollada:



Se implementó un sistema de **notificaciones programadas** utilizando **WorkManager**, una herramienta de Android diseñada para ejecutar tareas en segundo plano de manera confiable. La funcionalidad principal es:

1. Programación de la Notificación al Cerrar la Aplicación:

Al cerrar la aplicación, se utiliza WorkManager para programar una notificación que se mostrará después de un retraso específico (en este caso, 5 segundos).

2. Notificación Informativa:

La notificación invita al usuario a volver a visitar la galería de arte de la aplicación. Incluye un título, un mensaje descriptivo y un icono representativo.

3. Confiabilidad del Sistema:

WorkManager asegura que la notificación se envíe incluso si la aplicación está cerrada completamente o el dispositivo se reinicia, respetando las políticas de ahorro de batería de Android.

4. Compatibilidad:

Se maneja de manera automática la ejecución en dispositivos modernos, adaptándose a restricciones como las de Android 12 y superiores, mediante la creación de canales de notificación y el manejo de permisos.

3. Describe y ejemplifica cada uno de los pasos necesarios para publicar la aplicación solicitada en Google Play Store.

1. Crear una Cuenta de Desarrollador en Google Play

1. Accede al sitio de Google Play Console.
 2. Registra una cuenta de desarrollador pagando la tarifa única de \$25 USD.
 3. Completa los detalles requeridos:
 - Nombre del desarrollador.
 - Dirección de contacto.
 - Dirección de correo electrónico.
-

2. Preparar la Aplicación

Antes de subir la aplicación, asegurarse de que esté correctamente optimizada y lista para su distribución.

2.1 Generar el Archivo Firmado (APK o AAB)

1. Abre Android Studio.
2. Ve a **Build > Generate Signed Bundle/APK**.
3. Selecciona:
 - **APK**: Archivo compatible con dispositivos individuales.
 - **AAB (Android App Bundle)**: Recomendado, optimiza la entrega para múltiples dispositivos.
4. Genera un Keystore si no tienes uno:
 - Define una contraseña, alias y ubicación del archivo.
 - Guarda estos detalles en un lugar seguro.
5. Firma tu aplicación con el Keystore y genera el archivo .apk o .aab.

2.2 Validar y Optimizar

1. Usar herramientas como **Lint** en Android Studio para identificar errores comunes.
 2. Activa **ProGuard** o **R8** para reducir el tamaño del archivo y proteger tu código.
-

3. Registrar la Aplicación en Google Play Console

1. Accede a Google Play Console.
 2. Haz clic en **Crear aplicación**.
 3. Rellena los campos iniciales:
 - Nombre de la aplicación (ejemplo: "Galería Michel").
 - Idioma predeterminado.
 - Categoría (ejemplo: "Arte y Diseño").
 - Decide si la aplicación será gratuita o de pago.
-

4. Completar la Información de la Aplicación

Rellenar los datos obligatorios para que Google pueda validar la aplicación.

4.1 Descripción

- Breve descripción (máximo 80 caracteres) y una descripción larga que explique las funciones principales de la aplicación.

4.2 Capturas de Pantalla

1. Proporciona al menos 2 capturas de pantalla del diseño de tu aplicación:
 - Ejemplo: Pantalla principal y vista de pantalla completa.
2. Respetar las dimensiones requeridas por Google Play (mínimo 320px, máximo 3840px).

4.3 Icono y Gráficos Promocionales

1. Subir un icono de alta resolución (512x512 px).
2. Opcional: Proporciona un gráfico promocional (1024x500 px).

4.4 Políticas de Privacidad

1. Si tu aplicación recopila datos de los usuarios, proporciona un enlace a tu política de privacidad.
-

5. Subir el Archivo Firmado

1. Ve a la sección **Producción** en Google Play Console.
2. Crea una nueva versión.
3. Sube el archivo generado (.apk o .aab).
4. Agrega notas de la versión (cambios o mejoras).

6. Configurar Precio y Distribución

1. Selecciona si la aplicación será **gratuita** o **de pago**.
 - **Nota:** Una aplicación gratuita no puede cambiarse a pagada después de su publicación.
 2. Define los países en los que estará disponible.
 - Puedes seleccionar "Todo el mundo" o países específicos.
-

7. Probar la Aplicación

1. Realiza pruebas internas o cerradas antes de la publicación pública.
 - Invita testers para garantizar que la aplicación funciona correctamente en diferentes dispositivos.
 2. Revisa los resultados en Google Play Console.
-

8. Enviar para Revisión

1. Ir a la sección **Revisión** en Google Play Console.
 2. Asegurarse de haber completado todos los campos requeridos.
 3. Haz clic en **Enviar para Revisión**.
 - Google verificará tu aplicación. Este proceso puede tardar entre unas horas y varios días.
-

9. Publicar la Aplicación

1. Una vez aprobada, ve a la sección **Producción**.
 2. Haz clic en **Publicar ahora**.
 - La aplicación estará disponible en Google Play Store en unas horas.
-

10. Monitorear y Actualizar

1. Usa Google Play Console para:
 - Analizar descargas y métricas de rendimiento.
 - Leer reseñas de los usuarios.
 - Identificar problemas técnicos reportados.
2. Publica actualizaciones para corregir errores o mejorar funcionalidades.

Ejemplo de Publicación

- Nombre de la aplicación: **Galería arte**
- Categoría: **Arte y Diseño**
- Descripción breve: "Explora imágenes de arte únicas con Galería Michel."
- Países disponibles: **Todo el mundo**
- Versión inicial: **1.0.0**
- Notas de la versión: "Primera versión con vistas de galería y pantalla completa."

REFERENCIAS BIBLIOGRÁFICAS:

- **Ejemplo texto de lectura de IACC:**

IACC. (2024). *Herramientas de programación móvil*
Semana 8