

TALLER 4

Examen Final

**PROGRAMACIÓN II**

# Unidades de Competencia

*Aprendizaje esperado correspondiente a la unidad y la actividad:*

* *U1 - Utilizar colecciones y arrays en Kotlin para el almacenamiento y manipulación de datos.*
* *U1 - Diseñar y desarrollar funciones en Kotlin para dividir y reutilizar funcionalidades.*
* *U1 - Aplicar los principios de la programación orientada a objetos (POO) en la escritura de código Kotlin para resolver los requerimientos del usuario.*
* *U1 - Gestionar los eventos de usuario en aplicaciones Android para agregar funcionalidad.*
* *U2 - Utilizar recursos de texto y drawables para mejorar la interfaz de usuario de una aplicación Android.*
* *U2 - Configurar y gestionar dependencias de Gradle para incluir bibliotecas externas en un proyecto Android.*
* *U2 - Implementar ViewModels y utilizar variables de estado para mantener datos de manera eficiente.*
* *U2 - Desarrollar listas dinámicas utilizando LazyColumn o RecyclerView para mostrar y gestionar datos de manera eficiente.*
* *U2 - Implementar navegación entre pantallas utilizando el componente de navegación de Android.*
* *U2 - Utilizar corrutinas para realizar operaciones asincrónicas de manera eficiente.*
* *U3 - Diseñar y gestionar bases de datos SQLite para el almacenamiento de información en aplicaciones Android.*
* *U3 - Utilizar la biblioteca Room para simplificar la interacción con bases de datos SQLite.*

## Instrucciones generales

Una vez realizada la lectura comprensiva de la unidad I, II y III del material de estudio, es fundamental la realización de este taller, el que tiene por objetivo medir la correcta internalización y aplicación de los conceptos abordados en las mencionadas unidades.

El desarrollo de este trabajo es individual, no se permitirán entregas de talleres en duplas o en grupos.

**Contenidos asociados al taller:**

1. Colecciones y arreglos en Kotlin

2. POO en Kotlin

3 .Gestión de eventos

4. Recursos de texto y gráficos en Android

5. Construcción de interfaces de usuario (Compose o Vistas)

6. Gradle

7. ViewModels

8. Variables de estado

9. Listas dinámicas (LazyColumn o RecyclerView)

10. Componente de Navegación

11. Programación Asincrónica (Corrutinas)

12. Persistencia en Base de Datos SQLite

## Instrucciones específicas

El taller propuesto requiere articular contenidos abordados en la Unidad I, II y III, a través de las siguientes acciones:



**Al final de este documento podrás encontrar la pauta de evaluación y los indicadores de logro asociados a cada actividad.**

# Actividad 1

### Instrucciones

*La empresa ha decidido abordar las preocupaciones sobre cobros injustificados y alzas en los servicios básicos en Chile, desarrollando una aplicación que permita a los usuarios registrar las lecturas de los medidores de agua, luz y gas en sus hogares. Esta primera etapa se centra en la creación de una aplicación inicial con las siguientes características técnicas:*

*:*

1. *Diseñar y programar la solución usando Programación Orientada a Objetos en Kotlin. La aplicación debe registrar al menos el tipo de gasto, el valor indicado por el medidor respectivo y la fecha.*
2. *Diseñar y codificar 2 pantallas en la aplicación, puede utilizar Compose o Vistas XML:*
   1. *Una que muestre el listado de mediciones registradas por los medidores*
   2. *Una que muestre un formulario para ingresar el registro*
3. *Implementar la gestión de eventos para el botón que permite navegar hacia el formulario y el botón que guarda un nuevo registro.*
4. *Utilizar recursos drawables o íconos para representar los diferentes tipos de registro (agua, luz, gas).*
5. *Utilice recursos de texto para hacer que los botones y etiquetas soporten inglés y español.*
6. *Implementar al menos un ViewModel para gestionar la lógica de presentación y datos.*
7. *Utilizar listas dinámicas, como LazyColumn o RecyclerView, para mostrar los registros en la pantalla de listado*
8. *Utilice el componente de navegación para moverse de una pantalla a otra*
9. *Utilizar la biblioteca ROOM para persistir los datos en una base de datos SQLite, asegurando que los registros se mantengan incluso después de cerrar la aplicación.*
10. *Emplear corrutinas donde sea necesario para evitar bloquear el hilo principal de la aplicación y garantizar una experiencia de usuario fluida.*

*A continuación, se muestran las pantallas sugeridas para solucionar el caso:*

|  |  |
| --- | --- |
| *A screenshot of a phone  Description automatically generated*  Figura 1 Solución sugerida para listado de registro de medidores | *A screenshot of a computer  Description automatically generated*  Figura 2 Solución sugerida de formulario para el ingreso de lecturas de medidores |

# Entregables

Se solicitan los siguientes entregables para poder evaluar el taller:

1. **Informe en Word o PDF sin comprimir**: el informe debe ser redactado utilizando sus propias palabras para explicar el trabajo realizado. La claridad en la exposición de su trabajo contribuirá a una mejor evaluación y retroalimentación.
   1. Debe estar en formato Word o PDF
   2. Se requiere cargar el trabajo en la plataforma sin comprimir, de manera que pueda someterse a las herramientas de detección de plagio. De lo contrario, se aplicarán penalizaciones en forma de resta de puntos. Este procedimiento es crucial para asegurar la integridad académica y garantizar una evaluación justa y precisa del trabajo.
2. **Código fuente del proyecto comprimido en ZIP o vía enlace:**
   1. En caso de adjuntar el código fuente comprimido, debe realizar una limpieza previa del proyecto utilizando la función "**Clean Project**" de Android Studio, la cual se encuentra en el menú "Build". Esta práctica garantiza la eliminación de archivos innecesarios.
   2. En el caso de utilizar GitHub u otra plataforma similar, es su responsabilidad asegurarse de que el enlace proporcionado sea correcto y que el repositorio sea visible públicamente.
   3. En el caso de utilizar alguna plataforma de almacenamiento, es su responsabilidad compartir un enlace que permita la visibilidad pública de los archivos.

### Recomendaciones generales

### Se recomienda una lectura minuciosa de las instrucciones proporcionadas y una revisión exhaustiva de la pauta de evaluación. Asegúrese de abordar cada uno de los puntos indicados en ambas fuentes, ya que estas contienen detalles cruciales para el desarrollo y la evaluación exitosa de su proyecto.

### Se sugiere que el informe se mantenga conciso, centrándose específicamente en los puntos relevantes para la evaluación. Dirija su atención a los aspectos clave del desarrollo que se están evaluando, evitando incluir información superflua. Para cualquier requisito adicional, es posible revisar el código fuente. Este enfoque permite una mejor retroalimentación.

### Cuando escriba su código, es recomendable emplear nombres significativos tanto para variables, atributos, funciones, métodos como para clases. Opte por nombres descriptivos que reflejen claramente el propósito y contenido de la entidad que representan. Por ejemplo, en lugar de utilizar nombres genéricos como "a1", considere nombrar sus variables de manera explícita, por ejemplo, "montoAPagar". Esto no solo mejora la claridad y la legibilidad del código, sino que también facilita la comprensión tanto para usted como para otros colaboradores que puedan revisar o trabajar en el proyecto en el futuro. Además, incluir comentarios explicativos puede proporcionar una guía adicional sobre la función y la lógica subyacente.

**Información Importante**

Esta actividad será evaluada por el profesor de la asignatura, por lo que se sugiere, para su realización, considerar los Criterios de Evaluación detallados al final de este documento. El puntaje total de este taller es de 100 puntos, y el puntaje mínimo de aprobación es de 60 puntos.

# Formalidades

* Tipo de letra: Arial 12
* Interlineado: 1,15
* Entregables:
  + 1) Entrega del taller en archivo Word o PDF (sin comprimir), que lleve como nombre: SuNombre\_SuApellido.
  + 2) Código fuente de su proyecto comprimido en ZIP o a través de enlace (Github, Google Drive, OneDrive, o similar)
* Subir archivo a la plataforma

# Pauta de evaluación

|  |  |  |
| --- | --- | --- |
| **Criterios de evaluación** | **Indicadores de logro** | **Puntaje** |
| Comprende los conceptos fundamentales de Kotlin para el desarrollo de aplicaciones | Utiliza correctamente variables mutables e inmutables (val y var). | 5 |
| Utiliza colecciones y arrays en Kotlin para el almacenamiento y manipulación de datos. | Utiliza correctamente listas para almacenar los datos de la aplicación. | 5 |
| Aplica los principios de la programación orientada a objetos (POO) en la escritura de código Kotlin para resolver los requerimientos del usuario. | Estructura y distribuye las funcionalidades necesarias para la aplicación a través de clases. Se destaca por una organización coherente y eficiente, donde cada clase desempeña su función específica de manera clara y contribuye al conjunto general de la solución del problema. | 5 |
| Diseña interfaces de usuario utilizando Compose o Vistas XML | Organiza la interfaz utilizando Compose o Vistas XML. Crea un diseño ordenado, aplicando espacios y alineaciones entre elementos. | 5 |
| Gestiona los eventos de usuario en aplicaciones Android para agregar funcionalidad. | Gestiona eventos correctamente para dar funcionalidad a la aplicación | 5 |
| Uso adecuado de recursos de texto y drawables para mejorar la apariencia de la interfaz de usuario. | Incorpora y utiliza al menos 3 recursos drawable o iconos | 4 |
| Crea los recursos de texto indicados en inglés y español | 5 |
| Utiliza los recursos de texto en composables o vistas | 5 |
| Configuración correcta de dependencias de Gradle para incluir bibliotecas externas relevantes en el proyecto. | Agrega y modifica correctamente las dependencias indicadas en el o los archivos de Gradle | 5 |
| Implementación de navegación entre pantallas utilizando el componente de navegación de Android. | Implementa el componente de navegación para estructurar el proyecto | 5 |
| Vincula las pantallas usando el componente de navegación | 5 |
| Creación y uso efectivo de ViewModels para almacenar y gestionar datos de la aplicación. | Crea al menos un ViewModel | 5 |
| Utiliza MutableState o FlowState correctamente dentro del ViewModel | 5 |
| Desarrollo de listas dinámicas (LazyColumn o RecyclerView) que permitan al usuario ver y gestionar datos de manera eficiente. | Utiliza LazyColumn o RecyclerView para mostrar la o las listas indicadas en el ejercicio | 5 |
| Diseñó y creó una base de datos SQLite que persiste la información requerida por el negocio. | Crea clase entidad o similar con todos los campos requeridos | 5 |
| Escoge correctamente los tipos de datos | 5 |
| Crea interfaz DAO con los métodos solicitados | 5 |
| Implementa correctamente los métodos DAO | 5 |
| Utiliza Corrutinas al momento de trabajar con la BD | 5 |
|  | Evidencia un código ordenado mediante una correcta indentación. Usa nomenclaturas apropiadas para nombrar clases, variables y funciones, contribuyendo así a la claridad y legibilidad del código. Además, escribe comentarios explicativos, y emplea nombres significativos que facilitan la interpretación de clases, variables y funciones | 2 |
| Adjunta el código fuente del proyecto en formato ZIP o proporciona un enlace visible públicamente. El proyecto está limpio, libre de archivos innecesarios. | 2 |
| Cumple con éxito el plazo de entrega, presentando un informe en formato Word o PDF sin comprimir. El informe está completo, redactado con buena ortografía y explica con las propias palabras del estudiante el proceso seguido. Se enfoca en los puntos esenciales de la unidad a evaluar. | 2 |
| **PUNTAJE TOTAL** | | **100** |

