Portafolio de Evidencias Agosto – diciembre 2025

DESARROLLO DE APLICACIÓN MOVIL I

JOSE ARTURO BUSTAMANTE LAZCANO

Nombre del Alumno: Jimena Hdez Villa

Séptimo Semestre

Contenido

[Evaluación Diagnostica de Desarrollo de Aplicación Móvil I 2](#_Toc216314972)

[# Semana 1 – Introducción al desarrollo de aplicaciones 4](#_Toc216314973)

[# Semana 2 – Profundización en fundamentos de desarrollo de aplicaciones 11](#_Toc216314974)

[# Semana 3 – Interfaces de usuario y primeras pantallas de la aplicación 14](#_Toc216314975)

[# Semana 4 – Navegación entre pantallas, eventos y estructura lógica de la app 18](#_Toc216314976)

[# Semana 5 – Manejo de datos, listas y pantallas dinámicas en aplicaciones 23](#_Toc216314977)

[# Semana 6 – Formularios, validaciones y retroalimentación al usuario 28](#_Toc216314978)

[# Semana 7 – Persistencia de datos local y manejo básico del ciclo de vida de la app 34](#_Toc216314979)

[# Semana 8 – Consumo de servicios web (APIs) y sincronización básica de datos 40](#_Toc216314980)

[# Semana 9 – Proyecto integrador I: prototipo funcional de la aplicación móvil 47](#_Toc216314981)

[# Semana 10 – Cierre del curso, pruebas, documentación y entrega del proyecto integrador 55](#_Toc216314982)

[# Semana 11 – Retroalimentación, mejora continua y portafolio profesional de aplicaciones móviles 63](#_Toc216314983)

[# Semana 12 – Publicación, despliegue y mantenimiento de aplicaciones móviles 68](#_Toc216314984)

[# Semana 13 – Evaluación final, refuerzo, continuidad y cierre académico del curso 73](#_Toc216314985)

[# Semana 14 – Proyección, temas avanzados, investigación y cierre profesional del curso 80](#_Toc216314986)

[# Semana 15 – Síntesis global, acreditación y cierre formal del repositorio del curso 85](#_Toc216314987)

[# Semana 16 – Recuperación, regularización, consolidación total y cierre administrativo del curso 92](#_Toc216314988)

# Evaluación Diagnostica de Desarrollo de Aplicación Móvil I

Nombre: Jimena Hdez Villa

**1. ¿Qué es una aplicación móvil?**

a) Un documento de Word que se abre en el celular  
b) Un programa diseñado para ejecutarse en dispositivos móviles  
c) Una página web que solo funciona en computadoras  
d) Un archivo PDF interactivo

**2. ¿Cuál de los siguientes es un sistema operativo móvil?**

a) Windows 7  
b) Linux Mint  
c) Android  
d) macOS

**3. ¿Qué es un APK en Android?**

a) Un lenguaje de programación  
b) El archivo instalador de una aplicación Android  
c) Un editor de código  
d) Un emulador de Android

**4. ¿Qué significa “UI” en el contexto de apps móviles?**

a) Universal Input  
b) User Interface (Interfaz de Usuario)  
c) Unique Identifier  
d) Ultra Internet

**5. ¿Cuál de las siguientes NO es una característica típica de una app móvil?**

a) Uso de sensores (GPS, giroscopio, cámara)  
b) Interacción táctil  
c) Dependencia total de un mouse  
d) Pantallas adaptadas a tamaños pequeños

**6. ¿Qué es un emulador en desarrollo móvil?**

a) Un virus de Android  
b) Un dispositivo físico  
c) Un programa que simula un dispositivo móvil en tu computadora  
d) Un tipo de APK especial

**7. ¿Qué lenguaje se usa comúnmente para desarrollar apps nativas en Android?**

a) JavaScript solamente  
b) Java o Kotlin  
c) PHP  
d) SQL

**8. ¿Qué es una “actividad” (Activity) en Android?**

a) Una base de datos  
b) Una pantalla o interfaz con la que el usuario interactúa  
c) Un archivo de configuración del sistema  
d) Un tipo de sensor

**9. ¿Qué es un “layout” en desarrollo móvil?**

a) El diseño visual de la interfaz (cómo se acomodan los elementos)  
b) El código que conecta la base de datos  
c) Un tipo de sensor  
d) Una notificación emergente

**10. ¿Qué es una app híbrida?**

a) Una app escrita solo en C  
b) Una app que se ejecuta únicamente en iOS  
c) Una app hecha con tecnologías web (HTML, CSS, JS) empaquetada para móvil  
d) Una app que no usa internet

# # Semana 1 – Introducción al desarrollo de aplicaciones

> Carpeta: `Semana1` del repositorio `desarrolloapps1`

> Periodo: Agosto – diciembre 2025

La carpeta \*\*Semana1\*\* está pensada como el punto de partida del curso de \*\*Desarrollo de Aplicaciones\*\*, donde se colocan los primeros materiales, ejercicios y ejemplos prácticos.

Este documento describe de forma extensa cuál es el \*\*propósito de la semana\*\*, qué tipo de \*\*archivos\*\* se esperan, cómo se relacionan entre sí y cómo puedes aprovecharlos para estudiar o documentar tu trabajo en la materia.

## 1. Objetivo general de la Semana 1

La \*\*Semana 1\*\* tiene como finalidad:

- Presentar la \*\*visión general del curso\*\* de desarrollo de aplicaciones.

- Introducir al estudiante a las \*\*herramientas básicas\*\* que se utilizarán (entorno de desarrollo, repositorio, compiladores, editores, etc.).

- Revisar los \*\*conceptos fundamentales\*\* de la programación y la estructura de una aplicación.

- Establecer el \*\*flujo de trabajo\*\* con el repositorio GitHub:

- Cómo descargar el código.

- Cómo organizar archivos por semana.

- Cómo documentar avances y prácticas.

En esta fase no se busca aún construir una aplicación compleja, sino \*\*sentar las bases\*\*: entender la organización del proyecto, familiarizarse con la estructura de carpetas y comprender la lógica de cómo se irá trabajando semana a semana.

## 2. Organización lógica de la carpeta `Semana1`

Aunque el nombre exacto de cada archivo puede variar, la carpeta \*\*Semana1\*\* suele agrupar contenido de los siguientes tipos:

1. \*\*Apuntes y documentación teórica\*\*

Archivos de texto (por ejemplo `.md`, `.txt` o incluso `.pdf`) donde se explican:

- El propósito del curso.

- El temario general.

- Conceptos básicos de desarrollo de aplicaciones.

- Instrucciones para instalar herramientas o configurar el entorno.

2. \*\*Ejemplos de código base\*\*

Pequeños programas o proyectos de ejemplo que muestran:

- Cómo estructurar un proyecto sencillo.

- Cómo compilar o ejecutar una aplicación “hola mundo”.

- Cómo organizar directorios y archivos fuente.

3. \*\*Ejercicios prácticos / tareas de la semana\*\*

Archivos que contienen:

- Enunciados de prácticas.

- Instrucciones paso a paso.

- Actividades que el estudiante debe realizar para reforzar lo visto en clase.

4. \*\*Recursos adicionales (opcional)\*\*

Material de apoyo como:

- Diapositivas exportadas.

- Imágenes o diagramas.

- Enlaces de referencia a documentación oficial o tutoriales.

La idea es que, al abrir la carpeta `Semana1`, el estudiante pueda identificar \*\*rápidamente\*\* qué leer primero (teoría), qué ejemplo revisar (código base) y qué ejercicios debe realizar (práctica).

## 3. Contenido típico por tipo de archivo

A continuación, se describe con mayor detalle qué suele contener cada tipo de archivo que podría estar dentro de la carpeta \*\*Semana1\*\*.

### 3.1. Apuntes teóricos

En los apuntes teóricos se suele incluir:

- \*\*Presentación del curso\*\*

- Nombre de la materia (Desarrollo de Aplicaciones / Desarrollo de Apps 1).

- Nombre del docente.

- Periodo (Agosto–Diciembre 2025).

- Modalidad de trabajo (clases presenciales, prácticas en laboratorio, uso de repositorio, etc.).

- \*\*Objetivos de la materia\*\*

- Desarrollar aplicaciones funcionales paso a paso.

- Aprender a usar herramientas modernas de desarrollo.

- Aplicar buenas prácticas de programación y documentación.

- \*\*Temario general (visión global)\*\*

- Semana 1: Introducción y entorno de trabajo.

- Semanas posteriores: estructuras de control, manejo de datos, interfaces, conexión con servicios, etc.

(En el documento de Semana1 normalmente solo se bosqueja el panorama general.)

- \*\*Entorno de trabajo\*\*

- Herramientas recomendadas (IDE o editor de código).

- Gestión de versiones con Git y GitHub.

- Recomendaciones de organización de carpetas por semana y por proyecto.

Estos apuntes sirven como \*\*marco de referencia\*\* para todo el semestre y suelen ser el primer archivo que el estudiante debería revisar al entrar a la carpeta `Semana1`.

### 3.2. Ejemplos de código base

La carpeta puede contener uno o varios proyectos o archivos con código sencillo, cuyo objetivo principal es:

- Ver en la práctica cómo luce una \*\*estructura mínima de proyecto\*\*.

- Mostrar un ejemplo tipo \*\*“Hola Mundo”\*\* en el lenguaje o tecnología que se utilizará.

- Demostrar el proceso de:

- Abrir el proyecto.

- Compilar o ejecutar.

- Ver el resultado en consola, navegador o dispositivo.

Algunos elementos que pueden incluir los ejemplos:

- \*\*Archivos fuente\*\*

Por ejemplo:

- Un archivo principal donde se encuentra la función principal (`main`) o el punto de entrada de la aplicación.

- Archivos auxiliares que ilustran cómo separar lógica en distintas unidades.

- \*\*Comentarios en el código\*\*

Para explicar:

- Qué hace cada sección.

- Cómo compilar o ejecutar desde la línea de comandos o desde el IDE.

- Buenas prácticas básicas de estilo de código (nombres de variables, indentación, etc.).

- \*\*Archivos de configuración (opcional)\*\*

Dependiendo de la herramienta, podría haber archivos de configuración del proyecto (por ejemplo, archivos de proyecto del IDE, configuraciones de compilación o archivos de build).

Estos ejemplos no buscan ser complejos, sino \*\*claros y didácticos\*\*, ayudando al estudiante a entender la estructura básica sin abrumarlo con muchos detalles técnicos desde el primer día.

### 3.3. Ejercicios prácticos de la Semana 1

Además de ejemplos listos, la carpeta `Semana1` suele incluir \*\*ejercicios o tareas\*\* que el estudiante debe realizar. En esta semana, típicamente los ejercicios se centran en:

- \*\*Familiarizarse con el repositorio\*\*

- Clonar o descargar el repositorio desde GitHub.

- Navegar por las carpetas de las diferentes semanas.

- Ubicar el código de ejemplo y los apuntes.

- \*\*Primeros programas sencillos\*\*

- Crear una versión propia de “Hola Mundo”.

- Modificar ligeramente el ejemplo proporcionado (cambiar mensajes, agregar variables, etc.).

- Compilar y ejecutar la aplicación, verificando que los cambios se reflejan.

- \*\*Documentación básica\*\*

- Escribir en un archivo (por ejemplo `.md` o `.txt`) qué se hizo en el ejercicio.

- Anotar problemas encontrados y cómo se resolvieron.

- Documentar los comandos básicos utilizados (si se usó línea de comandos o Git).

Los enunciados de estos ejercicios pueden estar en uno o varios archivos, por ejemplo:

- Un archivo que explique \*\*la actividad 1\*\* (configurar el entorno).

- Otro archivo con \*\*la actividad 2\*\* (crear y ejecutar el primer programa).

- Instrucciones adicionales para entregar o subir evidencia.

### 3.4. Recursos adicionales

Finalmente, es posible que la carpeta incluya algunos recursos de apoyo, tales como:

- \*\*Presentaciones o notas exportadas\*\*

- Diapositivas introductorias a la materia.

- Resumen de los temas que se verán durante el semestre.

- Reglas de trabajo en clase y criterios de evaluación.

- \*\*Imágenes o diagramas\*\*

- Diagramas de flujo simples que muestran el ciclo de vida de una aplicación.

- Esquemas del proceso de desarrollo (análisis → diseño → implementación → pruebas → despliegue).

- \*\*Enlaces de referencia\*\*

- Referencias a documentación oficial del lenguaje o framework que se utilizará.

- Tutoriales básicos recomendados.

- Videos introductorios sugeridos por el docente.

Estos materiales no siempre son obligatorios, pero ayudan a que el estudiante tenga un contexto más amplio y recursos adicionales para reforzar lo visto en clase.

## 4. Cómo utilizar el contenido de `Semana1`

La carpeta `Semana1` no solo almacena archivos, sino que marca el \*\*inicio del flujo de trabajo\*\* que se repetirá cada semana. Una forma recomendada de usar este contenido es:

1. \*\*Leer primero la documentación / apuntes\*\*

- Entender el objetivo de la semana.

- Revisar qué herramientas se necesitan.

- Identificar qué se espera que el estudiante logre al finalizar la semana.

2. \*\*Explorar los ejemplos de código\*\*

- Abrir el proyecto o archivo de ejemplo en el entorno de desarrollo.

- Compilar o ejecutar el código.

- Realizar pequeñas modificaciones para comprobar que todo funciona.

3. \*\*Realizar los ejercicios propuestos\*\*

- Seguir las instrucciones de las prácticas.

- Crear o modificar archivos según se indique.

- Probar continuamente el programa para comprobar resultados.

4. \*\*Documentar los avances\*\*

- Registrar en un archivo de notas lo aprendido.

- Anotar errores y soluciones.

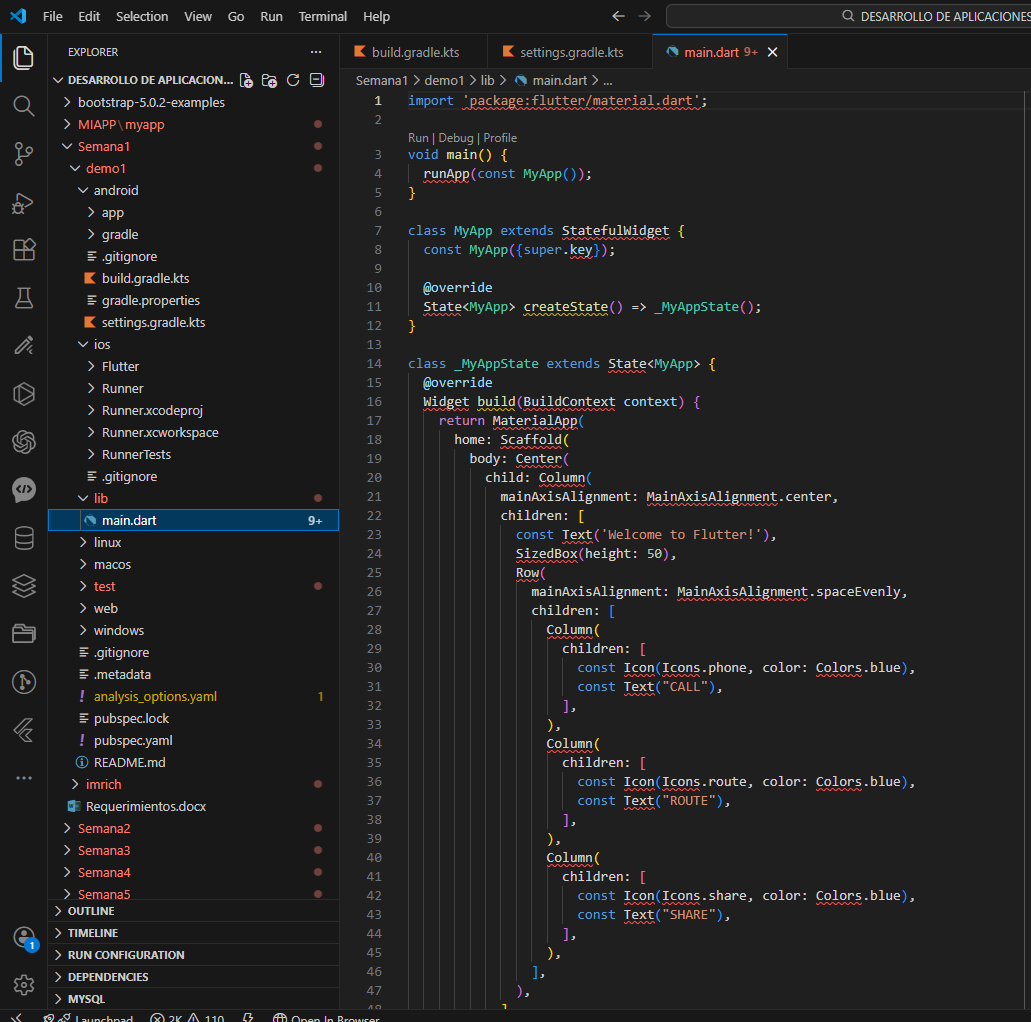
- Dejar comentarios en el propio código cuando algo sea importante o no obvio.

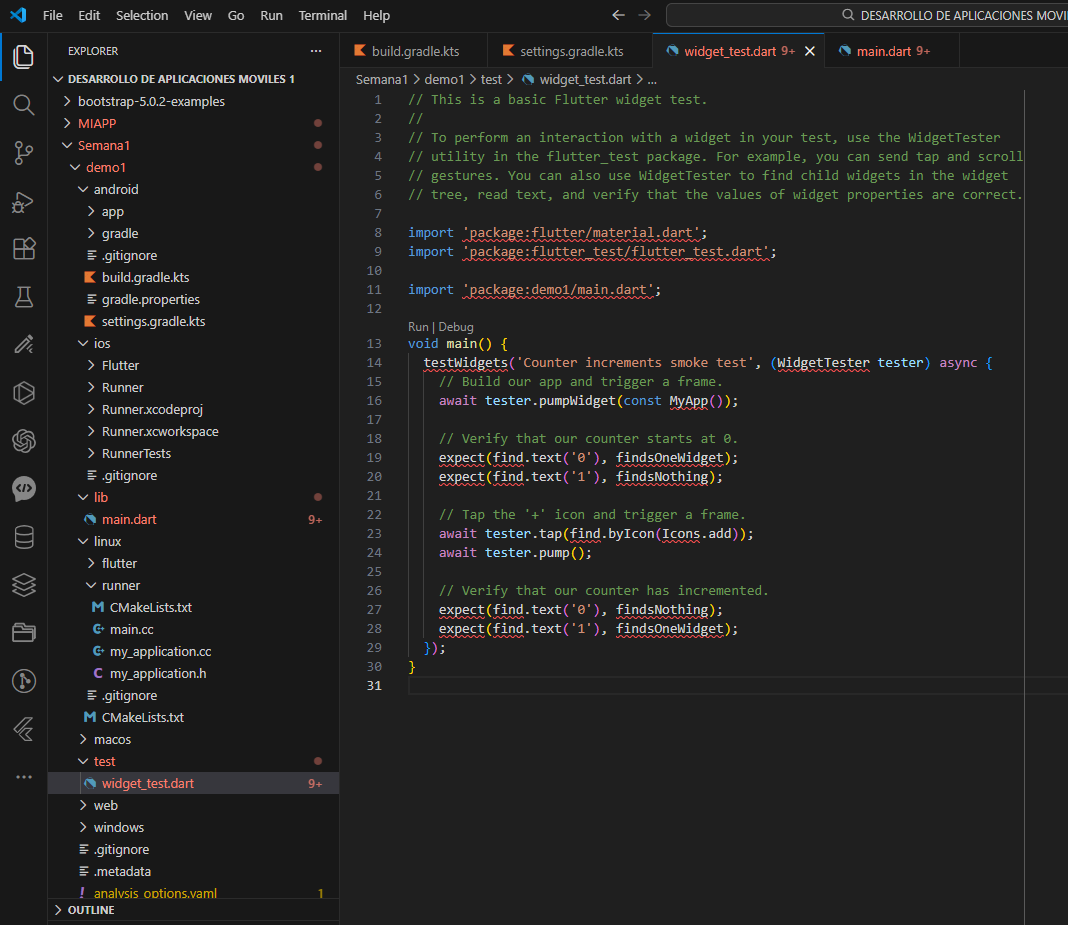
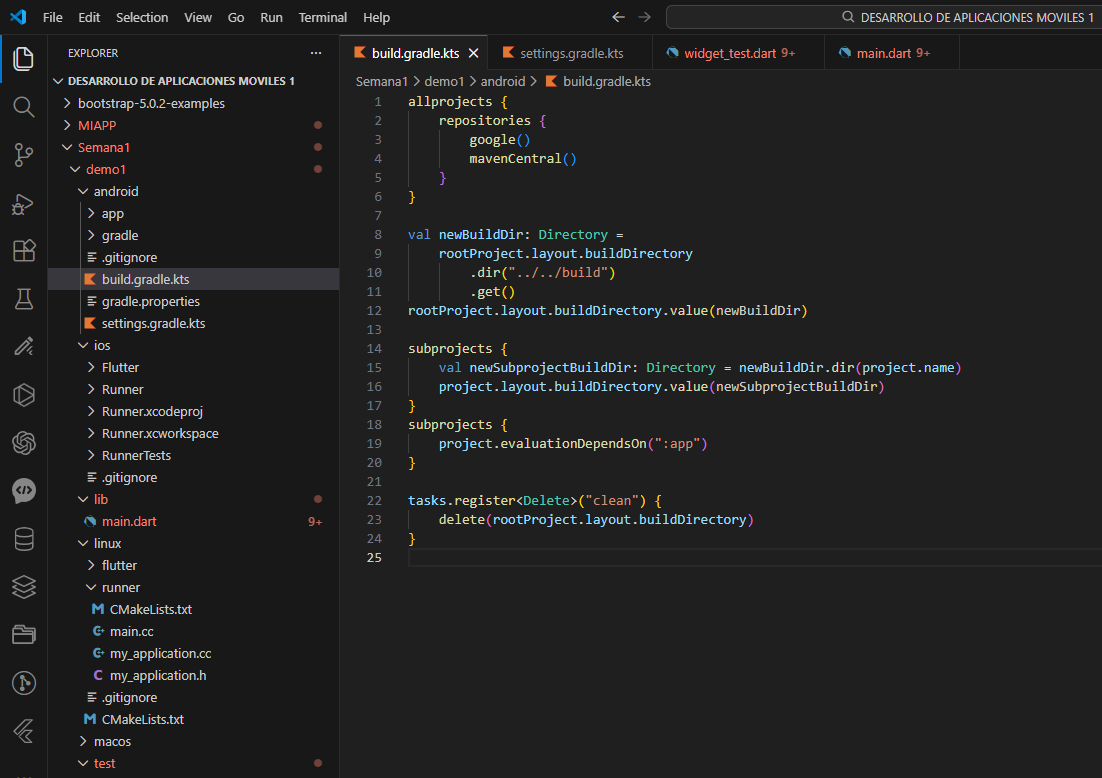
5. \*\*Prepararse para la siguiente semana\*\*

- Ver qué temas se tocarán después de la Semana 1.

- Verificar que se cumplieron los objetivos planteados.

- Dejar el entorno listo (repositorio actualizado, código ordenado, apuntes guardados).





# # Semana 2 – Profundización en fundamentos de desarrollo de aplicaciones

> Carpeta: `Semana2` del repositorio `desarrolloapps1`

> Periodo: Agosto–Diciembre 2025

La carpeta \*\*Semana2\*\* contiene el material correspondiente a la \*\*segunda semana del curso de Desarrollo de Aplicaciones\*\*.

En esta etapa se busca reforzar lo visto en la Semana 1 y avanzar hacia conceptos más prácticos de \*\*lógica de programación, estructura de una aplicación\*\* y \*\*organización del código\*\*.

Este documento describe de forma detallada el propósito de la Semana 2, el tipo de contenido que agrupa esta carpeta y cómo se espera que el estudiante trabaje con los archivos.

## 1. Objetivo general de la Semana 2

En la \*\*Semana 2\*\* se pretende que el estudiante:

- Consolide los \*\*conceptos básicos de programación\*\* introducidos en la primera semana.

- Comprenda mejor la \*\*estructura de un proyecto de aplicación\*\* (módulos, archivos fuente, recursos).

- Practique el uso del entorno de desarrollo (IDE/editor) con ejemplos más elaborados.

- Aplique \*\*estructuras de control\*\* (condicionales, ciclos) en pequeños programas o componentes de la app.

- Comience a pensar en la \*\*organización del código\*\* para que sea mantenible y escalable.

La idea principal es dejar de ver solo ejemplos muy simples (como “Hola Mundo”) y empezar a escribir \*\*lógica que resuelva pequeñas tareas reales\*\*, preparando el camino para las siguientes semanas.

## 2. Organización general de la carpeta `Semana2`

Aunque los nombres concretos de los archivos pueden variar, es común que la carpeta `Semana2` se organice en bloques como:

1. \*\*Documentación teórica / notas de clase\*\*

2. \*\*Ejemplos de código guiados por el docente\*\*

3. \*\*Ejercicios prácticos para el estudiante\*\*

4. \*\*Recursos adicionales y apoyo\*\*

La estructura puede verse conceptualmente así (ejemplo genérico):

```text

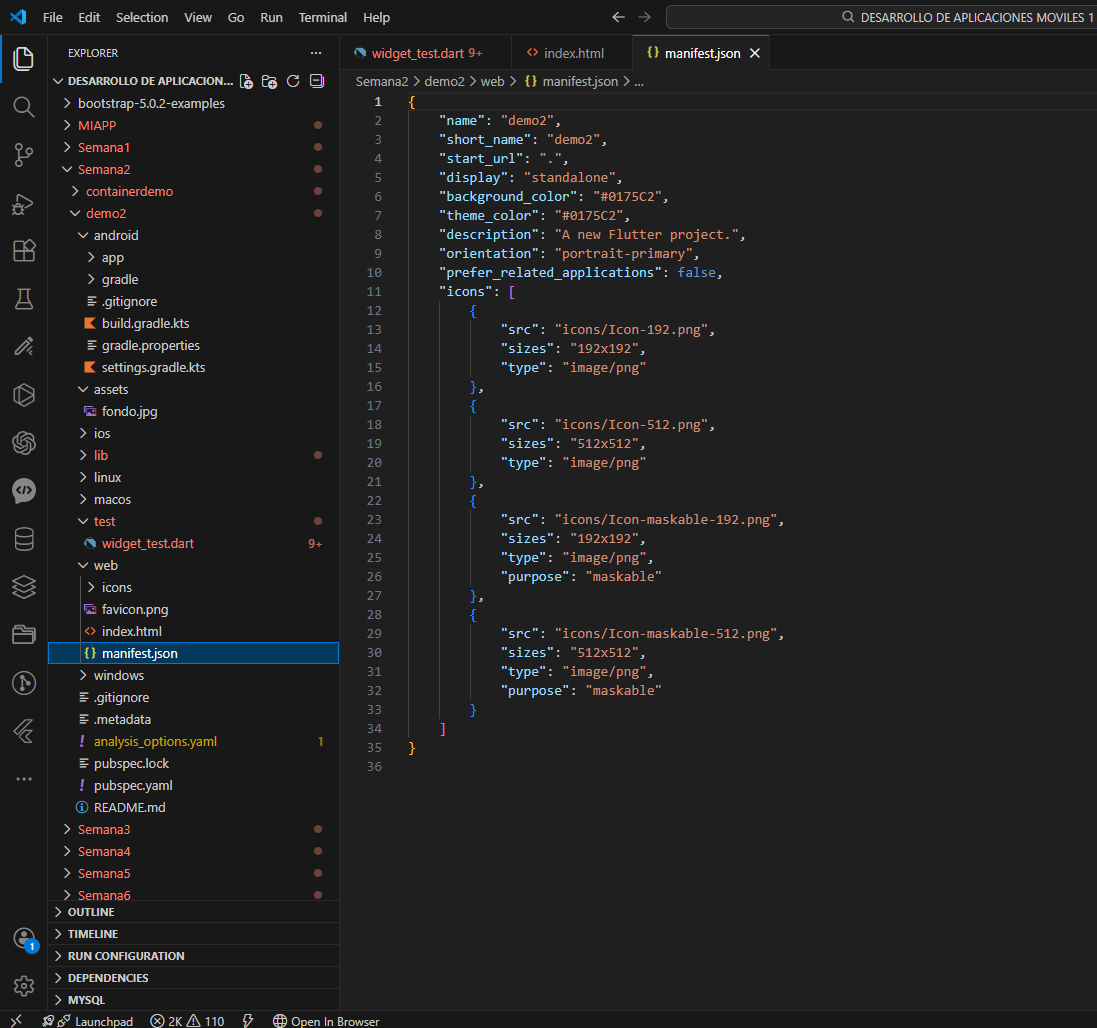
Semana2/

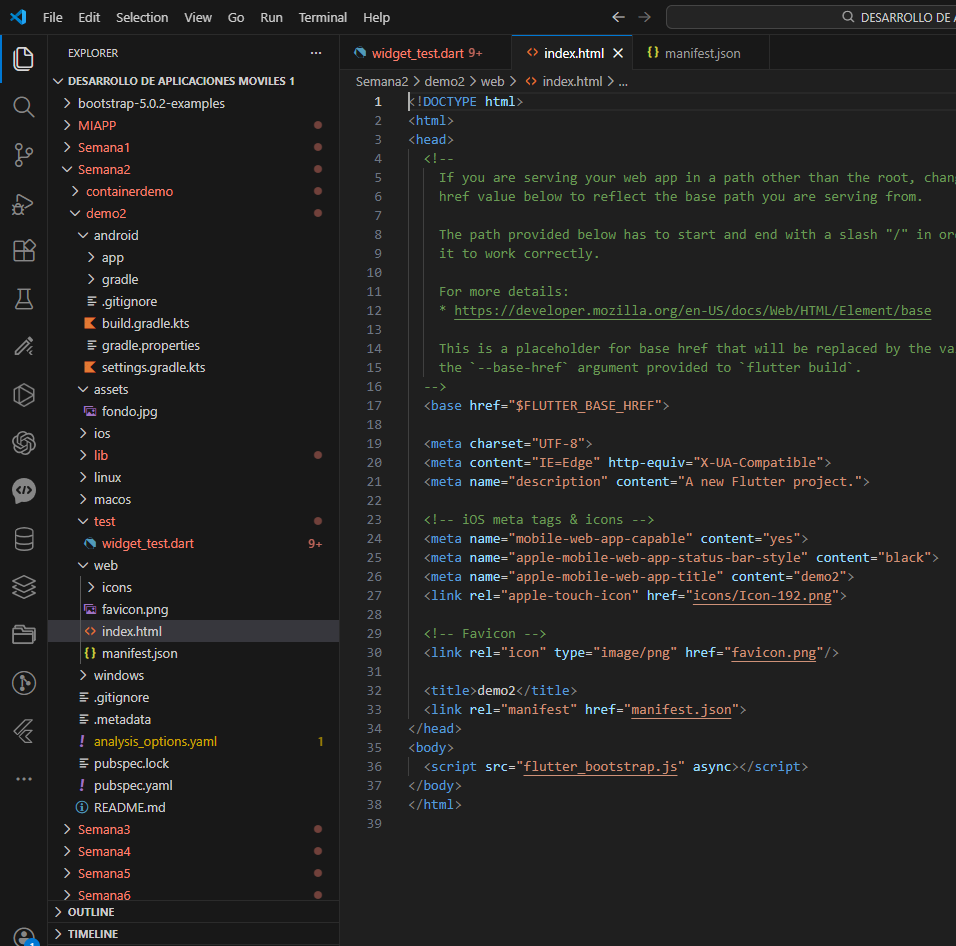
├─ docs/ # Documentos y apuntes teóricos de la semana

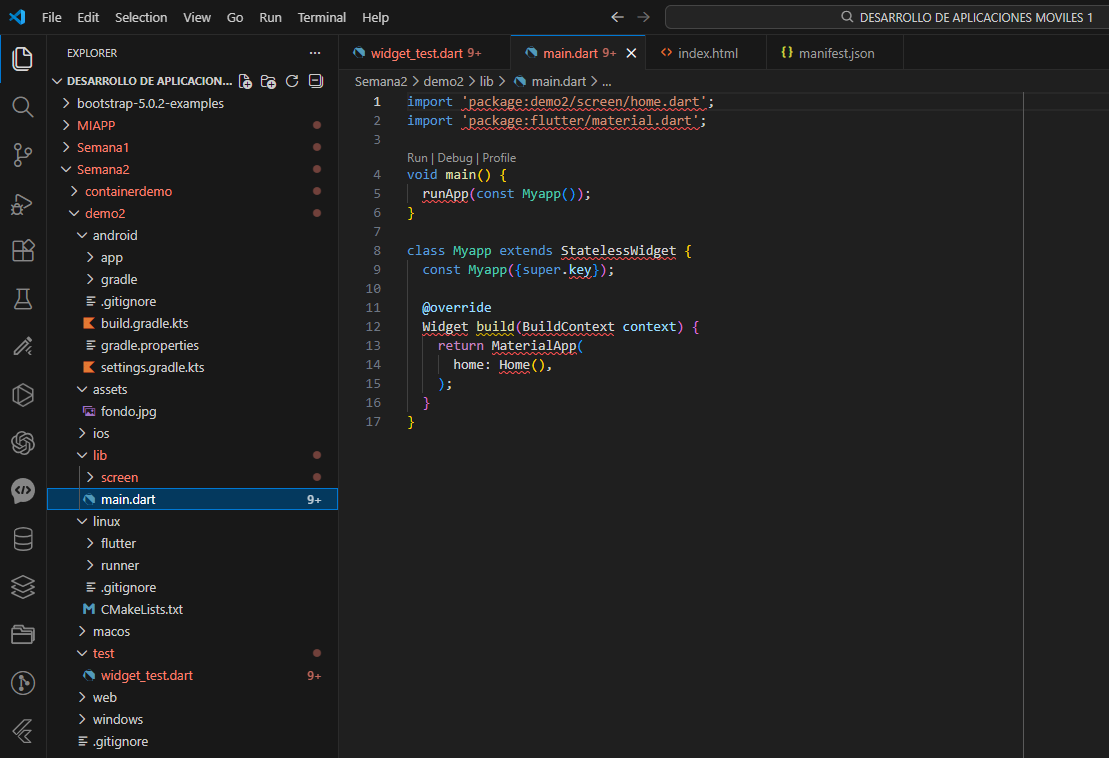
├─ ejemplos/ # Programas de ejemplo explicados en clase

├─ ejercicios/ # Actividades que el alumno debe resolver

└─ recursos/ # Imágenes, diagramas u otros archivos de apoyo







# # Semana 3 – Interfaces de usuario y primeras pantallas de la aplicación

> Carpeta: `Semana3` del repositorio `desarrolloapps1`

> Periodo: Agosto–Diciembre 2025

> Materia: Desarrollo de Aplicaciones / Desarrollo de Aplicación Móvil I

La carpeta \*\*Semana3\*\* contiene el material correspondiente a la \*\*tercera semana de trabajo\*\* del curso.

En este punto el estudiante ya revisó:

- Semana 1: Introducción al curso, entorno de desarrollo, “Hola mundo”.

- Semana 2: Fundamentos de lógica, estructuras de control, pequeños programas con decisiones y ciclos.

En \*\*Semana 3\*\* el enfoque se mueve hacia algo mucho más cercano a una app real:

el \*\*diseño de interfaces de usuario (UI)\*\*, la \*\*estructura de pantallas\*\* y la \*\*interacción básica\*\* con el usuario.

Este documento describe a detalle:

- El \*\*objetivo formativo\*\* de la semana.

- La \*\*organización interna\*\* de la carpeta `Semana3`.

- El tipo de \*\*código fuente\*\* que se espera encontrar.

- Los \*\*ejercicios y prácticas\*\* que el alumno realiza.

- Las \*\*evidencias\*\* que se recomiendan generar (capturas, código, etc.).

> Si en tu repositorio los nombres son distintos, puedes editar este archivo y reemplazarlos.

## 1. Objetivo general de la Semana 3

En \*\*Semana 3\*\* se busca que el estudiante:

1. Comprenda los \*\*conceptos básicos de interfaz de usuario (UI)\*\* en aplicaciones.

2. Reconozca la importancia de la \*\*experiencia de usuario (UX)\*\* desde el inicio del desarrollo.

3. Sea capaz de \*\*estructurar una pantalla básica\*\*:

- Título o encabezado.

- Zona de contenido principal.

- Botones o elementos interactivos.

4. Implemente \*\*interacción simple\*\*:

- Botones que reaccionan a un click/toque.

- Campos de texto que capturan información básica del usuario.

- Mensajes o resultados que cambian según la entrada del usuario.

5. Reforce el uso del repositorio (`desarrolloapps1`) como herramienta de:

- Organización de proyectos.

- Almacenamiento de ejemplos.

- Entrega de prácticas y evidencias.

El resultado ideal es que, al terminar esta semana, el estudiante ya sea capaz de construir y ejecutar \*\*al menos una pantalla funcional\*\*, aunque sea sencilla, que muestre una interfaz \*\*clara, legible y usable\*\*.

## 2. Organización general de la carpeta `Semana3`

Una organización típica para `Semana3` puede verse así:

```text

Semana3/

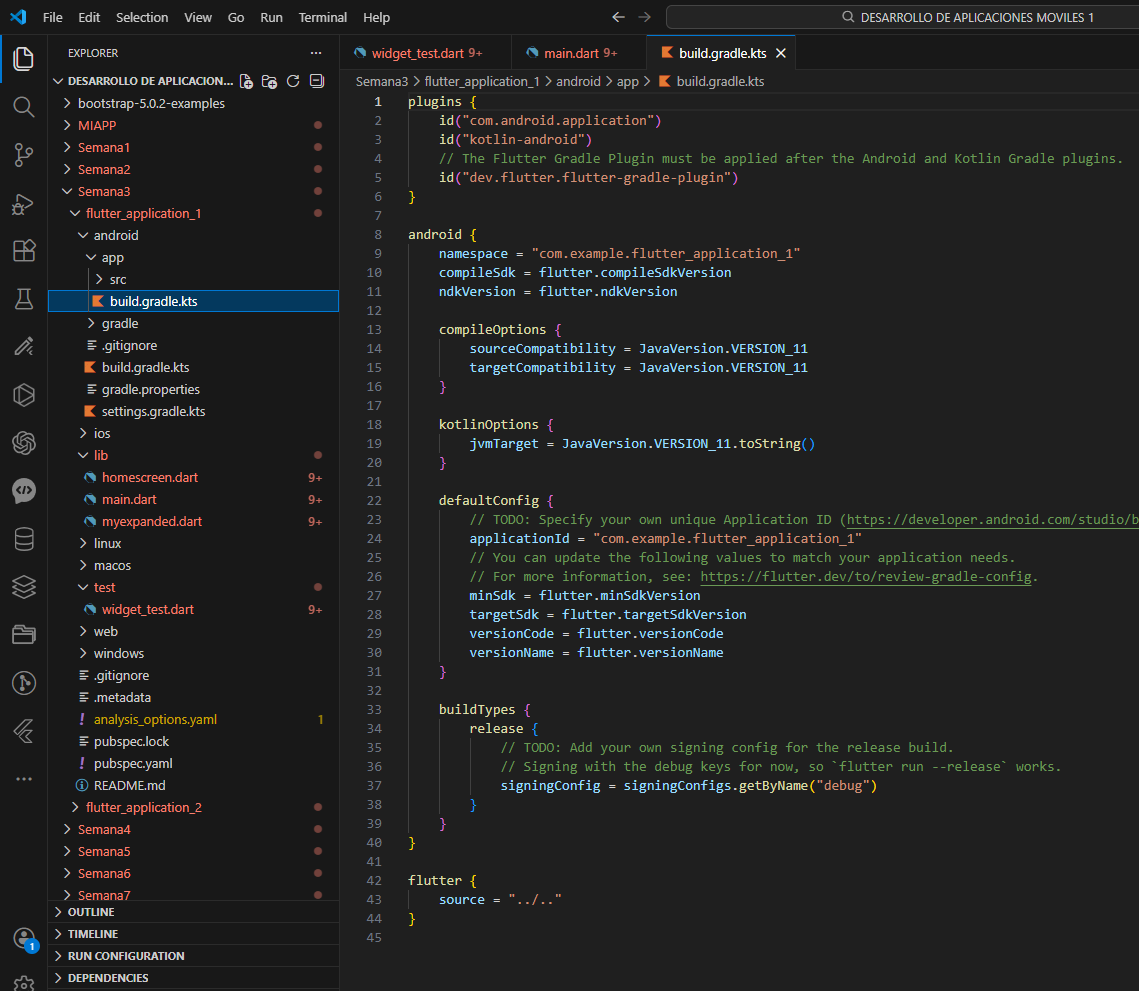
├─ docs/ # Apuntes, resúmenes y enunciados de la semana

├─ ejemplos\_ui/ # Ejemplos guiados de interfaces

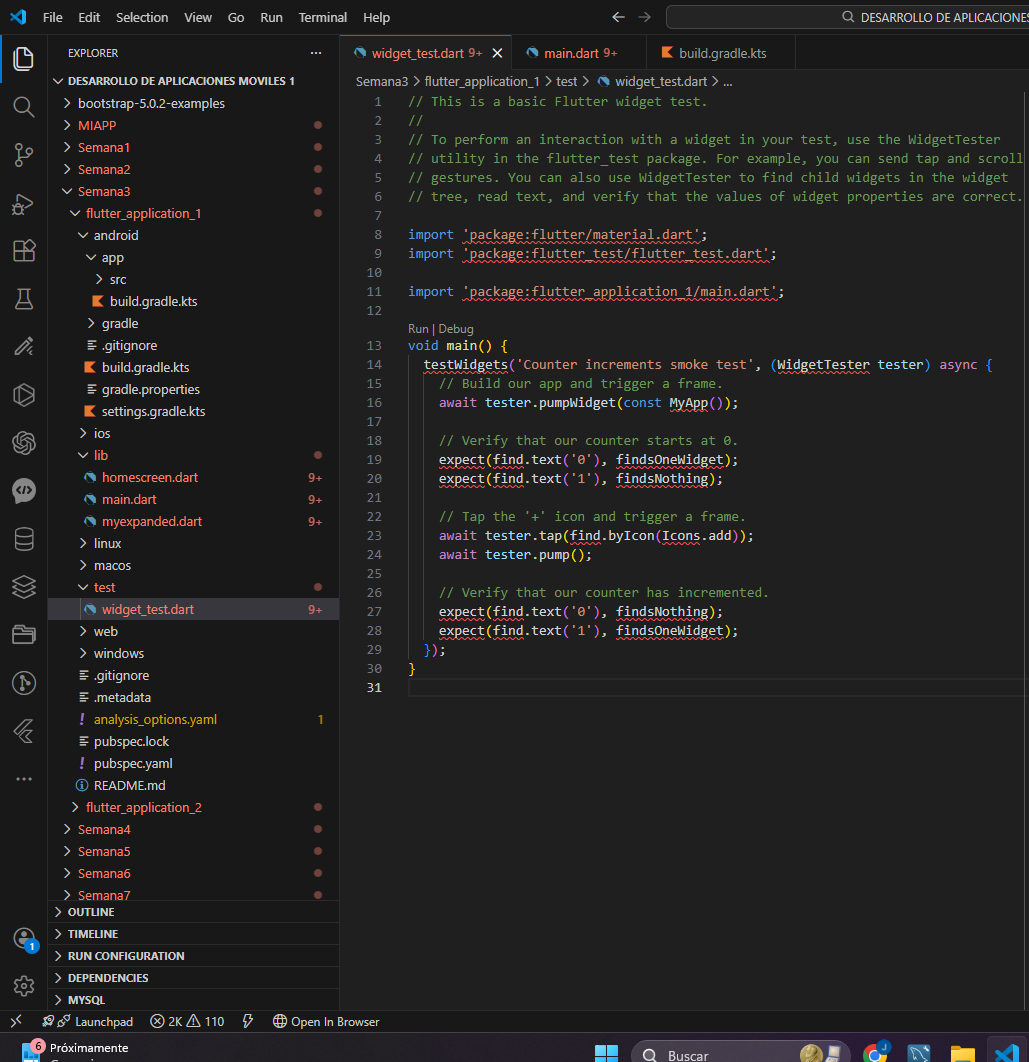
├─ proyecto\_base/ # Proyecto/base de app para la semana

├─ ejercicios/ # Ejercicios que el alumno debe completar

└─ evidencias/ # Capturas de pantalla y archivos de entrega







# # Semana 4 – Navegación entre pantallas, eventos y estructura lógica de la app

> Carpeta: `Semana4` del repositorio `desarrolloapps1`

> Materia: Desarrollo de Aplicación Móvil I

> Periodo: Agosto–Diciembre 2025

La carpeta \*\*Semana4\*\* corresponde a la cuarta semana de trabajo del curso de \*\*Desarrollo de Aplicaciones / Desarrollo de Aplicación Móvil I\*\*.

En este punto del semestre, el estudiante ya ha:

- Conocido el \*\*entorno de desarrollo\*\* y el repositorio (Semana 1).

- Practicado \*\*lógica básica y estructuras de control\*\* (Semana 2).

- Diseñado \*\*primeras interfaces de usuario\*\* y pantallas sencillas (Semana 3).

En \*\*Semana 4\*\* el enfoque se centra en dar un paso muy importante:

trabajar la \*\*navegación entre pantallas\*\*, el manejo de \*\*eventos (acciones del usuario)\*\* y comenzar a pensar en la \*\*estructura lógica de la aplicación\*\* como un todo, no solo como pantallas aisladas.

Este documento describe a detalle:

- Los \*\*objetivos formativos\*\* de la semana.

- La \*\*organización recomendada\*\* de la carpeta `Semana4`.

- El tipo de \*\*documentos y código\*\* que se espera encontrar.

- Las \*\*actividades y ejercicios\*\* clave.

- Las \*\*evidencias sugeridas\*\* para documentación y evaluación.

- Recomendaciones de trabajo, buenas prácticas y relación con semanas posteriores.

## 1. Objetivo general de la Semana 4

El objetivo global de la \*\*Semana 4\*\* es que el estudiante logre:

> \*\*Construir y comprender un flujo de navegación básico entre varias pantallas de una aplicación, manejando eventos del usuario (clicks, entradas de texto, cambios de estado) y organizando el código de forma coherente.\*\*

De este objetivo se derivan varios \*\*objetivos específicos\*\*:

1. Comprender el concepto de \*\*navegación\*\*:

- Pantalla inicial, pantallas secundarias, regreso hacia atrás.

- Rutas, intents o controladores de navegación (según la tecnología utilizada).

2. Manejar \*\*eventos de usuario\*\*:

- Click en botones.

- Cambios en campos de texto.

- Selecciones en listas u otros componentes.

3. Entender la idea de \*\*estado\*\*:

- Datos que cambian dentro de una pantalla.

- Cómo una acción del usuario modifica lo que se muestra.

4. Reforzar la \*\*organización del proyecto\*\*:

- Separación de archivos por pantalla o módulo.

- Nombrado adecuado de funciones y clases.

- Comentarios básicos en el código para explicar la lógica.

5. Producción de \*\*evidencias claras\*\*:

- Capturas de código.

- Capturas de la app en ejecución.

- Breve documentación de lo realizado.

## 2. Organización general sugerida de la carpeta `Semana4`

Aunque los nombres concretos de los archivos pueden variar, una estructura típica para `Semana4` podría ser:

```text

Semana4/

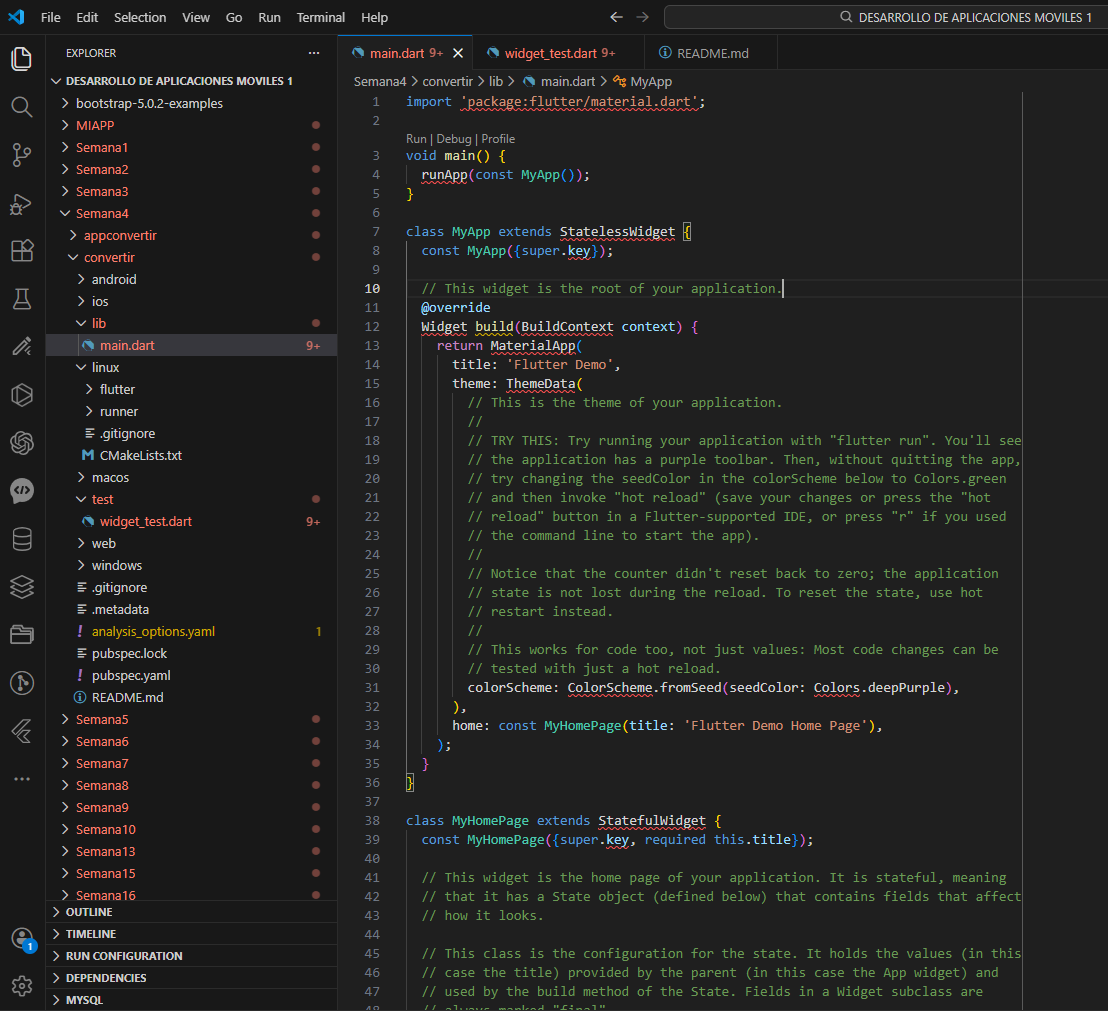
├─ docs/ # Documentación teórica y guías de la semana

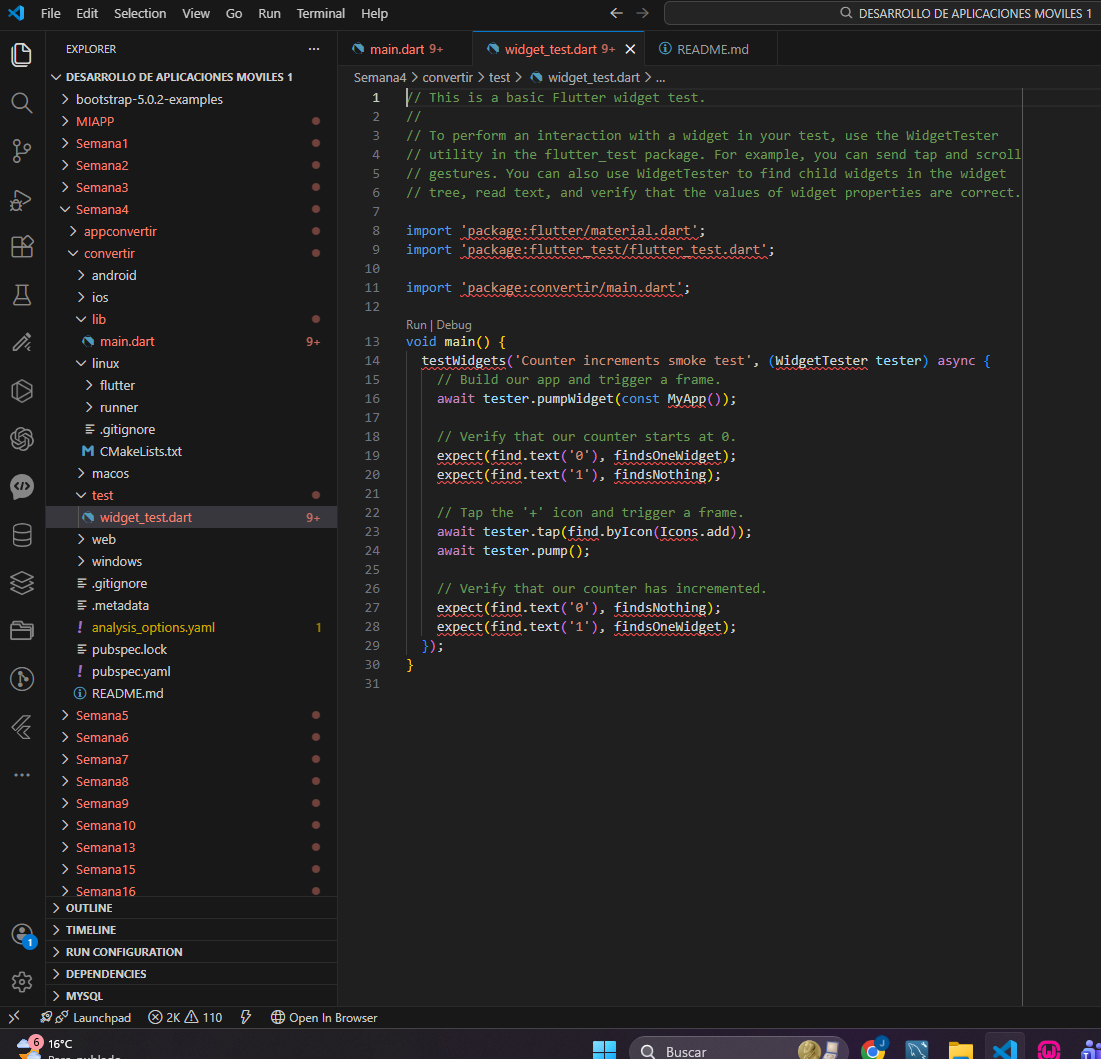
├─ ejemplos\_navegacion/ # Ejemplos guiados de navegación y eventos

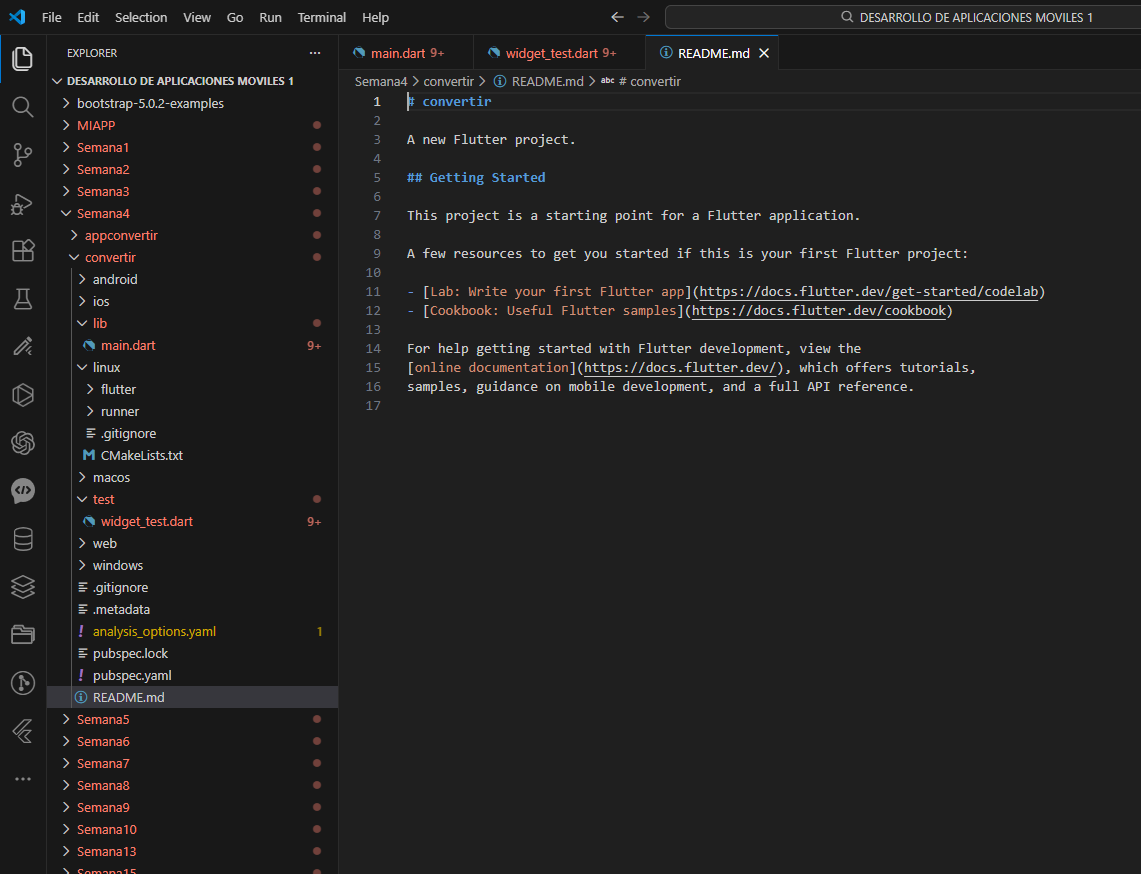
├─ proyecto\_semana4/ # Proyecto base o proyecto integrador de la semana

├─ ejercicios/ # Actividades que el alumno debe resolver

└─ evidencias/ # Capturas, reportes y archivos para entregar







# **Semana 1 – Introducción al desarrollo de aplicaciones**

# Carpeta: Semana1 del repositorio desarrolloapps1 Periodo: Agosto – diciembre 2025

# La carpeta **Semana1** sirve como punto de inicio del curso de **Desarrollo de Aplicaciones**, donde se incluyen los primeros materiales, ejemplos y ejercicios prácticos. Este documento explica detalladamente el **propósito de la semana**, los **tipos de archivos** que se esperan encontrar, cómo se relacionan entre sí y cómo aprovecharlos para estudiar o registrar tu trabajo en la materia.

# **1. Propósito general de la Semana 1**

# Durante la **Semana 1**, se pretende:

# Dar una **visión general del curso** de desarrollo de aplicaciones.

# Familiarizar al estudiante con las **herramientas iniciales** (IDE, repositorio, compiladores, editores, etc.).

# Revisar los **conceptos esenciales de programación** y la estructura básica de una aplicación.

# Enseñar el **uso del repositorio GitHub**, incluyendo:

# Cómo descargar y clonar proyectos.

# Cómo organizar archivos por semana.

# Cómo documentar avances y prácticas.

# El objetivo en esta etapa no es desarrollar aplicaciones complejas, sino **sentar las bases**: entender la estructura de carpetas, la organización del proyecto y la lógica de trabajo semanal.

# **2. Estructura de la carpeta Semana1**

# Aunque los nombres de los archivos pueden variar, la carpeta **Semana1** generalmente incluye:

# **Apuntes y documentación teórica** Archivos en .md, .txt o .pdf donde se explican:

# La finalidad del curso.

# Temario general.

# Conceptos básicos de desarrollo de aplicaciones.

# Guías de instalación y configuración del entorno.

# **Ejemplos de código iniciales** Pequeños programas o proyectos de referencia que muestran:

# Cómo organizar un proyecto sencillo.

# Cómo ejecutar un “Hola Mundo”.

# Distribución de directorios y archivos fuente.

# **Ejercicios prácticos** Archivos que incluyen:

# Enunciados de prácticas.

# Instrucciones paso a paso.

# Actividades para reforzar los contenidos vistos en clase.

# **Recursos complementarios (opcional)** Material adicional como:

# Diapositivas o presentaciones.

# Diagramas o imágenes.

# Enlaces a documentación oficial o tutoriales recomendados.

# De esta forma, al abrir la carpeta Semana1, el estudiante puede identificar rápidamente **qué leer primero**, **qué ejemplos revisar** y **qué ejercicios realizar**.

# **3. Contenido habitual de cada tipo de archivo**

# **3.1. Apuntes teóricos**

# Los apuntes suelen contener:

# **Presentación del curso**

# Nombre de la materia: Desarrollo de Aplicaciones / Desarrollo de Apps 1.

# Docente a cargo.

# Periodo académico.

# Modalidad de trabajo (clases presenciales, prácticas de laboratorio, uso del repositorio, etc.).

# **Objetivos del curso**

# Aprender a desarrollar aplicaciones funcionales paso a paso.

# Conocer y utilizar herramientas modernas de desarrollo.

# Aplicar buenas prácticas de programación y documentación.

# **Temario general**

# Semana 1: Introducción y configuración del entorno.

# Semanas siguientes: estructuras de control, manejo de datos, interfaces, conexión con servicios, entre otros.

# **Entorno de trabajo**

# Herramientas recomendadas (IDE o editor).

# Uso de Git y GitHub.

# Organización de carpetas y proyectos por semana.

# Estos apuntes funcionan como **referencia para todo el semestre** y suelen ser el primer archivo a revisar en la carpeta Semana1.

# **3.2. Ejemplos de código base**

# En esta carpeta se incluyen proyectos o archivos con código sencillo, cuyo objetivo es:

# Mostrar la **estructura mínima de un proyecto**.

# Presentar un ejemplo tipo **“Hola Mundo”** en el lenguaje o tecnología usada.

# Enseñar el flujo de trabajo: abrir, compilar y ejecutar un proyecto.

# Algunos elementos que suelen aparecer:

# **Archivos fuente**

# Archivo principal con la función main o punto de entrada de la aplicación.

# Archivos auxiliares que ejemplifican la separación de lógica en módulos.

# **Comentarios en el código**

# Explicando cada sección.

# Indicando cómo ejecutar o compilar.

# Buenas prácticas de nomenclatura e indentación.

# **Archivos de configuración (opcional)**

# Dependiendo del entorno, pueden incluir configuraciones de compilación o archivos del IDE.

# El objetivo es que estos ejemplos sean **claros y didácticos**, sin complicar al estudiante con detalles avanzados.

# **3.3. Ejercicios prácticos**

# En esta semana, los ejercicios están enfocados en:

# **Familiarizarse con el repositorio**

# Clonar o descargar proyectos.

# Navegar entre carpetas y localizar ejemplos y apuntes.

# **Primeros programas simples**

# Crear su propia versión de “Hola Mundo”.

# Modificar ejemplos (cambiar mensajes, agregar variables, etc.).

# Ejecutar y verificar resultados.

# **Documentación básica**

# Registrar en un archivo lo realizado.

# Anotar problemas y soluciones.

# Documentar comandos usados (Git, terminal o IDE).

# Los enunciados pueden estar separados por actividad: por ejemplo, **Actividad 1: Configurar entorno**, **Actividad 2: Primer programa**, etc.

# **3.4. Recursos complementarios**

# Opcionalmente, la carpeta puede incluir:

# **Presentaciones o notas exportadas**

# Resumen de temas del semestre.

# Normas de trabajo y evaluación.

# **Diagramas o imágenes**

# Diagramas de flujo del ciclo de vida de la aplicación.

# Esquemas de desarrollo (análisis → diseño → implementación → pruebas → despliegue).

# **Enlaces de referencia**

# Documentación oficial de la tecnología.

# Tutoriales y videos introductorios recomendados.

# Estos recursos ayudan a tener un **contexto más amplio** y refuerzan el aprendizaje.

# **4. Cómo aprovechar la carpeta Semana1**

# Se recomienda seguir estos pasos:

# **Revisar primero la teoría**

# Entender los objetivos de la semana.

# Identificar las herramientas necesarias.

# Saber qué resultados se esperan.

# **Explorar los ejemplos de código**

# Abrir proyectos en el IDE.

# Ejecutar y modificar el código para probarlo.

# **Realizar los ejercicios**

# Seguir las instrucciones de los enunciados.

# Crear o modificar archivos según las indicaciones.

# Verificar resultados constantemente.

# **Registrar avances**

# Documentar lo aprendido y errores solucionados.

# Dejar comentarios útiles en el código.

# **Prepararse para la siguiente semana**

# Revisar los próximos temas.

# Asegurar que los objetivos de la Semana 1 se cumplieron.

# Mantener el entorno y repositorio listos.

# **Semana 2 – Profundización en fundamentos de desarrollo de aplicaciones**

# Carpeta: Semana2 del repositorio desarrolloapps1 Periodo: Agosto–Diciembre 2025

# La carpeta **Semana2** contiene el material de la segunda semana del curso. En esta etapa se refuerzan los conceptos de la Semana 1 y se avanza hacia **lógica de programación más elaborada, organización de código y estructura de proyectos**.

# **1. Propósito general de la Semana 2**

# Al finalizar esta semana, el estudiante deberá:

# Consolidar los **conceptos básicos de programación**.

# Entender la **estructura de un proyecto de software** (módulos, archivos fuente y recursos).

# Practicar el entorno de desarrollo con ejemplos más complejos.

# Aplicar **estructuras de control** como condicionales y ciclos en pequeños programas.

# Comenzar a organizar el código de forma que sea **legible y mantenible**.

# El objetivo es dejar de lado los ejemplos muy simples y empezar a resolver **problemas reales de manera programática**.

# **2. Organización típica de la carpeta Semana2**

# Semana2/

# ├─ docs/ # Apuntes y documentación de la semana

# ├─ ejemplos/ # Ejemplos de código explicados por el docente

# ├─ ejercicios/ # Actividades a desarrollar por el alumno

# └─ recursos/ # Diagramas, imágenes u otros apoyos

# **Semana 3 – Interfaces de usuario y primeras pantallas**

# Carpeta: Semana3 del repositorio desarrolloapps1 Periodo: Agosto–Diciembre 2025

# En **Semana3** el enfoque se traslada al **diseño de interfaces (UI)** y la **interacción básica** con el usuario.

# El estudiante ya revisó:

# Semana 1: Introducción, entorno de desarrollo, “Hola Mundo”.

# Semana 2: Fundamentos de programación y estructuras de control.

# **1. Propósito general de la Semana 3**

# Comprender conceptos de **interfaz de usuario (UI)**.

# Valorar la **experiencia del usuario (UX)** desde el inicio.

# Construir **pantallas básicas**: encabezado, contenido y botones.

# Implementar **interacciones simples**: botones, campos de texto y mensajes dinámicos.

# Reforzar el uso de GitHub para organizar proyectos, ejemplos y evidencias.

# Al terminar, el estudiante debe ser capaz de crear **al menos una pantalla funcional y clara**.

# **2. Organización típica de Semana3**

# Semana3/

# ├─ docs/ # Apuntes y resúmenes

# ├─ ejemplos\_ui/ # Ejemplos guiados de UI

# ├─ proyecto\_base/ # Proyecto inicial para la semana

# ├─ ejercicios/ # Prácticas a realizar

# └─ evidencias/ # Capturas y entregables

# **Semana 4 – Navegación entre pantallas y eventos**

# Carpeta: Semana4 del repositorio desarrolloapps1 Periodo: Agosto–Diciembre 2025

# En esta etapa, el estudiante ya ha:

# Conocido el entorno de desarrollo y el repositorio (Semana 1).

# Practicado lógica básica (Semana 2).

# Diseñado primeras interfaces (Semana 3).

# **1. Propósito general de la Semana 4**

# **Desarrollar y comprender un flujo de navegación básico entre varias pantallas de una app, manejando eventos de usuario y organizando el código de manera lógica y coherente.**

# Objetivos específicos:

# Entender la **navegación**: pantallas principales y secundarias, retorno y rutas/intents.

# Manejar **eventos de usuario**: clicks, entradas de texto y selecciones.

# Comprender el **estado de la aplicación** y cómo las acciones del usuario lo modifican.

# Organizar el proyecto: separación de módulos, nombres claros y comentarios explicativos.

# Generar **evidencias claras**: capturas de código, screenshots y documentación breve.

# **2. Organización recomendada de Semana4**

# Semana4/

# ├─ docs/ # Documentación y guías de la semana

# ├─ ejemplos\_navegacion/ # Ejemplos de navegación y eventos

# ├─ proyecto\_semana4/ # Proyecto integrador de la semana

# ├─ ejercicios/ # Prácticas a desarrollar

# └─ evidencias/ # Capturas, reportes y entregables

# # Semana 7 – Persistencia de datos local y manejo básico del ciclo de vida de la app

> Carpeta: `Semana7` del repositorio `desarrolloapps1`

> Materia: Desarrollo de Aplicación Móvil I

> Periodo: Agosto–Diciembre 2025

La carpeta \*\*Semana7\*\* corresponde a la séptima semana de trabajo del curso de \*\*Desarrollo de Aplicación Móvil I\*\*.

Hasta este punto, el estudiante ya ha recorrido el siguiente camino:

- \*\*Semana 1:\*\* Entorno de desarrollo, repositorio, primer programa / app de prueba.

- \*\*Semana 2:\*\* Lógica básica, estructuras de control, decisiones y ciclos.

- \*\*Semana 3:\*\* Construcción de interfaces gráficas y componentes básicos.

- \*\*Semana 4:\*\* Navegación entre pantallas y manejo de eventos.

- \*\*Semana 5:\*\* Listas, colecciones y pantallas dinámicas.

- \*\*Semana 6:\*\* Formularios, validaciones y retroalimentación al usuario.

En \*\*Semana 7\*\* se introduce un tema clave para que las aplicaciones dejen de ser “temporales” y empiecen a \*\*recordar información\*\*:

> \*\*Persistencia de datos local y relación con el ciclo de vida básico de la aplicación.\*\*

Es decir, se aborda cómo hacer que los datos \*\*no se pierdan\*\* cuando el usuario cierra la app, cambia de pantalla o incluso apaga el dispositivo.

## 1. Propósito general de la Semana 7

El objetivo central de esta semana es que el estudiante sea capaz de:

> \*\*Diseñar e implementar mecanismos sencillos de persistencia de datos a nivel local (en el dispositivo), integrados con formularios y pantallas ya desarrolladas, respetando el ciclo de vida básico de la aplicación.\*\*

De este propósito general se desprenden varios objetivos específicos:

1. Comprender el concepto de \*\*persistencia\*\*:

- Diferencia entre datos en memoria (temporales) y datos almacenados (persistentes).

2. Conocer tipos de \*\*almacenamiento local\*\* comunes en aplicaciones:

- Preferencias simples (clave–valor).

- Archivos de texto o almacenamiento ligero.

- Introducción a bases de datos locales (según el alcance del curso).

3. Integrar la persistencia con:

- Formularios que guardan información.

- Listas que se llenan con datos previamente guardados.

4. Relacionar la persistencia con el \*\*ciclo de vida de la app\*\*:

- Qué pasa cuando se abre, se cierra, se pausa, se reanuda.

- En qué momentos conviene guardar o recuperar datos.

5. Generar evidencias claras de:

- Código que guarda/carga información.

- Pantallas que muestran datos persistidos.

- Breves explicaciones del proceso.

## 2. Competencias que se fortalecen en Semana 7

### 2.1. Competencias técnicas

- Uso de \*\*APIs de almacenamiento local\*\* del framework/plataforma (según se haya visto en clase).

- Lectura y escritura de datos en:

- Preferencias / configuración simple.

- Archivos o estructuras básicas.

- Integración de persistencia con:

- Formularios (guardar lo que el usuario captura).

- Listas (mostrar datos guardados anteriormente).

### 2.2. Competencias de diseño y arquitectura

- Separar lógicamente:

- \*\*Capa de datos\*\* (dónde y cómo se guardan).

- \*\*Capa de presentación\*\* (cómo se muestran en la interfaz).

- Pensar en estructuras mínimas de capas (por ejemplo, Modelo–Vista).

- Diseñar pantallas que consideren que los datos pueden venir de:

- Entradas del usuario.

- Datos anteriormente guardados.

### 2.3. Competencias de experiencia de usuario (UX)

- Mostrar al usuario \*\*indicadores claros\*\* cuando:

- Se ha guardado la información.

- Se ha producido un error al guardar.

- Evitar que el usuario pierda información sin aviso:

- Guardar al confirmar.

- Avisar antes de descartar cambios.

- Mantener la app \*\*consistente\*\*: que los datos que el usuario ve tengan sentido con lo que guardó.

### 2.4. Competencias de documentación y trabajo profesional

- Describir por escrito:

- Qué datos se guardan.

- Dónde se guardan (a nivel conceptual).

- En qué momento se guardan y se recuperan.

- Generar bitácoras simples de pruebas:

- Qué se probó.

- Qué funcionó y qué no.

- Cómo se resolvieron problemas.

## 3. Organización general sugerida de la carpeta `Semana7`

Una estructura ordenada para la carpeta `Semana7` podría ser:

```text

Semana7/

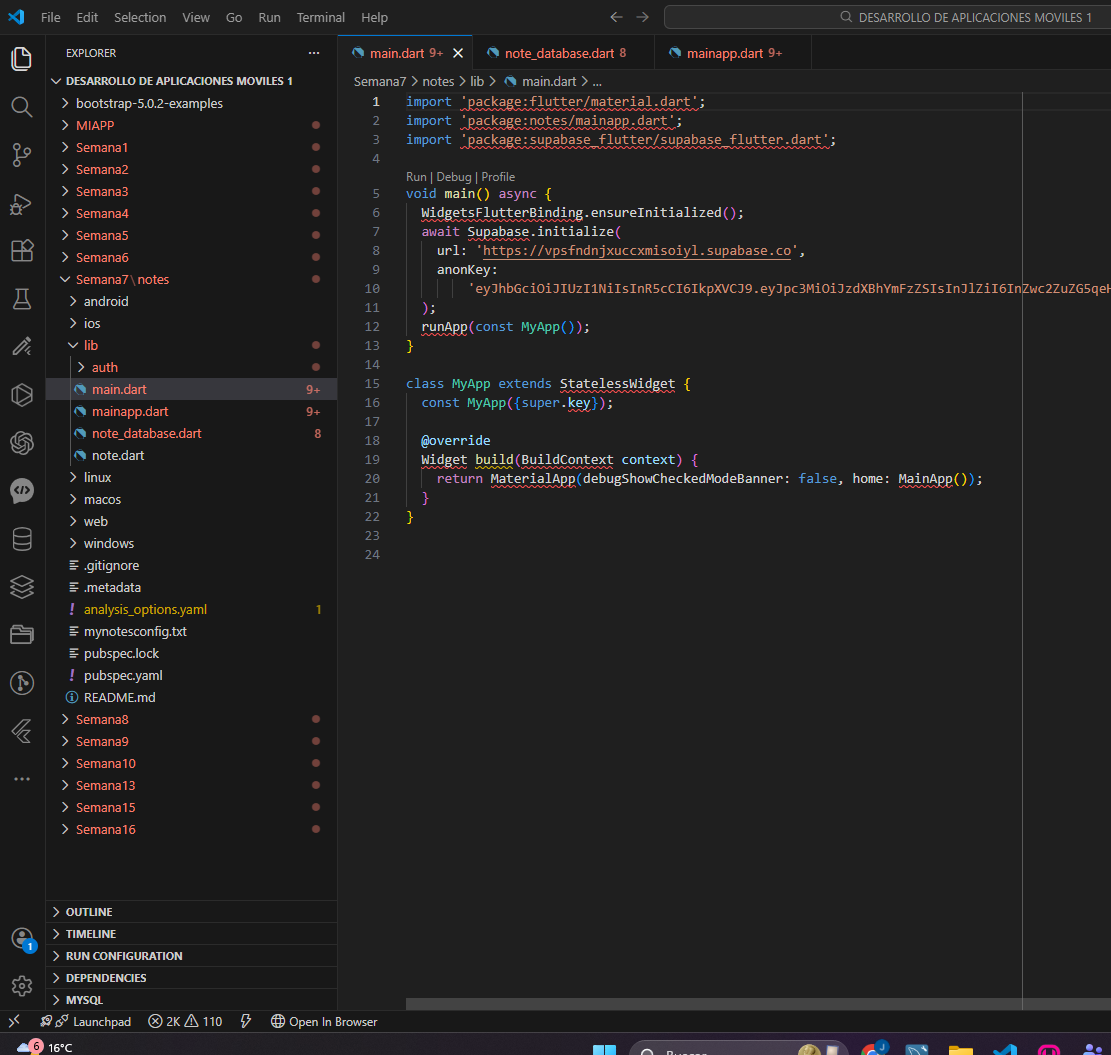
├─ docs/ # Documentación teórica, guías y apuntes de la semana

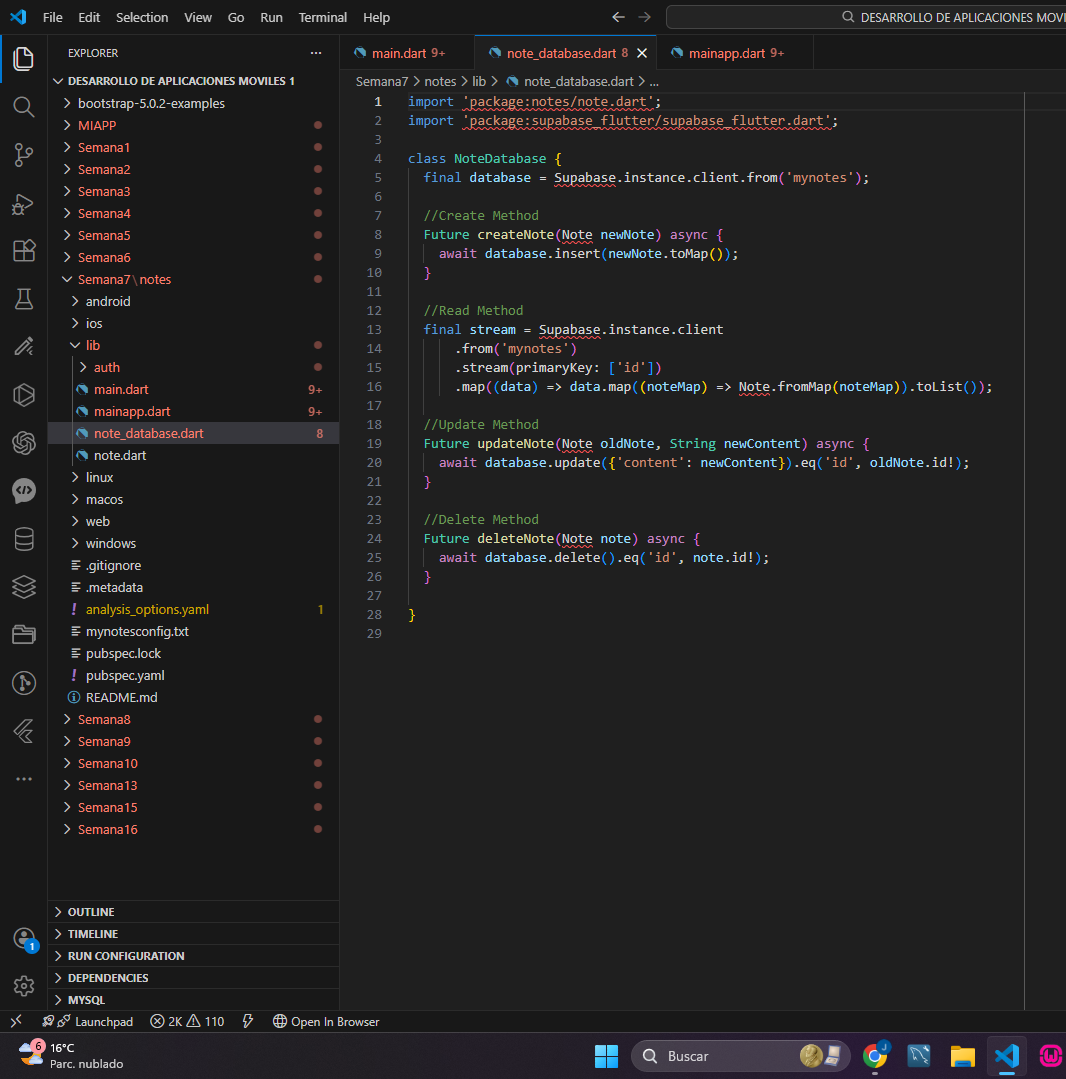
├─ ejemplos\_persistencia/ # Ejemplos guiados de almacenamiento local

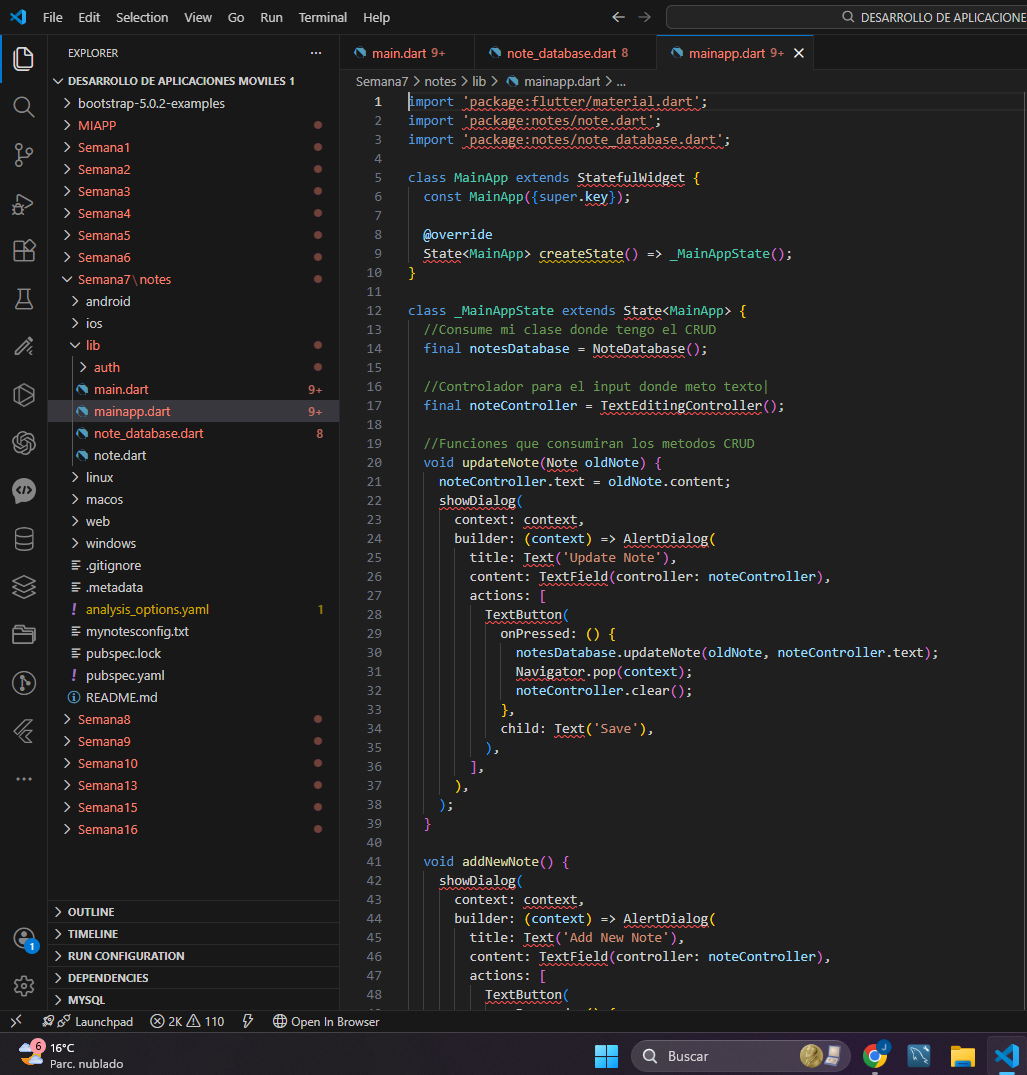
├─ proyecto\_semana7/ # Proyecto integrador de la semana

├─ ejercicios/ # Actividades individuales y/o por equipo

└─ evidencias/ # Capturas, reportes y entregas de Semana 7







# # Semana 8 – Consumo de servicios web (APIs) y sincronización básica de datos

> Carpeta: `Semana8` del repositorio `desarrolloapps1`

> Materia: Desarrollo de Aplicación Móvil I

> Periodo: Agosto–Diciembre 2025

La carpeta \*\*Semana8\*\* corresponde a la octava semana de trabajo del curso de \*\*Desarrollo de Aplicación Móvil I\*\*.

Hasta este punto, el estudiante ya ha recorrido el siguiente camino:

- \*\*Semana 1:\*\* Entorno de desarrollo, organización del repositorio y primer programa / app de prueba.

- \*\*Semana 2:\*\* Lógica básica, estructuras de control y decisiones.

- \*\*Semana 3:\*\* Interfaces gráficas iniciales y componentes básicos de UI.

- \*\*Semana 4:\*\* Navegación entre pantallas y manejo de eventos.

- \*\*Semana 5:\*\* Listas, colecciones de datos y pantallas dinámicas.

- \*\*Semana 6:\*\* Formularios, validaciones y retroalimentación al usuario.

- \*\*Semana 7:\*\* Persistencia de datos local y nociones del ciclo de vida de la app.

En \*\*Semana 8\*\* se da un salto importante hacia el mundo “real” de las aplicaciones conectadas:

> \*\*Se aborda el consumo de servicios web (APIs), el intercambio de datos en formato JSON y la sincronización básica entre información local y remota.\*\*

El objetivo es que la aplicación deje de trabajar solo con datos “quemados” en el código o almacenados localmente, y comience a \*\*leer información desde un servidor\*\*, procesarla y mostrarla al usuario de forma amigable.

## 1. Rol de la Semana 8 dentro del curso

La \*\*Semana 8\*\* marca el inicio del trabajo con \*\*redes y servicios web\*\* dentro del contexto del desarrollo móvil.

A partir de aquí, el estudiante empieza a entender cómo:

- Las apps se conectan a internet.

- Consumen información desde APIs (por ejemplo, listados, detalles, configuraciones).

- Envían datos al servidor (en versiones más avanzadas).

- Combinan datos \*\*locales\*\* con datos \*\*remotos\*\*.

Esta semana es una preparación directa para:

- Temas de \*\*integración con backends\*\*.

- Manejo de \*\*autenticación\*\* (login contra un servidor).

- Sincronización de datos más compleja (offline/online).

## 2. Objetivos de aprendizaje de la Semana 8

### 2.1. Objetivo general

> \*\*Que el estudiante sea capaz de consumir un servicio web sencillo desde su aplicación móvil, interpretar la respuesta (usualmente en JSON) y mostrar los datos recibidos en la interfaz de manera ordenada y comprensible.\*\*

### 2.2. Objetivos específicos

Al finalizar Semana 8, se espera que el estudiante:

1. Comprenda los conceptos básicos de \*\*HTTP\*\* y \*\*APIs REST\*\*:

- Qué es una URL / endpoint.

- Qué es un método HTTP (GET, POST, etc.).

- Qué es un código de respuesta (200, 404, 500, etc.).

2. Sea capaz de realizar una \*\*petición HTTP GET\*\* desde la app:

- Hacia un endpoint definido (público o de prueba).

- Controlando la ejecución (inicio, éxito, error).

3. Interprete y procese respuestas en \*\*formato JSON\*\*:

- Identificar llaves y valores.

- Mapear la respuesta a un modelo de datos (clases, estructuras).

4. Muestre la información recibida en la \*\*interfaz gráfica\*\*:

- Listas que se llenan con datos remotos.

- Pantallas de detalle que muestran información de un elemento recibido.

5. Maneje casos de \*\*error de red y estados de carga\*\*:

- Mostrar una indicación de “Cargando…” mientras llega la respuesta.

- Mostrar un mensaje de error si no hay conexión o la respuesta es inválida.

6. Documente y evidencie:

- El endpoint utilizado.

- La estructura del JSON (al menos de forma básica).

- Capturas de la app mostrando datos reales provenientes de un servicio.

## 3. Competencias que se refuerzan en Semana 8

### 3.1. Competencias técnicas

- Uso de \*\*librerías o APIs de red\*\* del framework/plataforma (según el curso).

- Manejo de \*\*peticiones HTTP\*\* (especialmente GET).

- Interpretación de respuestas en \*\*JSON\*\*:

- Conversión de texto JSON a estructuras de datos nativas.

- Manejo de listas y objetos anidados.

- Integración de datos remotos con la interfaz:

- Llenado de listas.

- Actualización dinámica de pantallas.

### 3.2. Competencias de arquitectura y organización

- Separar, al menos de forma sencilla:

- Lógica de red (servicios, clientes).

- Lógica de datos (modelos).

- Lógica de presentación (pantallas).

- Empezar a utilizar “capas” o módulos:

- Un archivo/clase para el \*\*servicio de API\*\*.

- Un archivo/clase para el \*\*modelo de datos remoto\*\*.

- Un archivo/clase para la \*\*pantalla que consume esos datos\*\*.

### 3.3. Competencias de experiencia de usuario (UX)

- Mostrar al usuario estados de:

- \*\*Carga:\*\* indicador mientras se espera la respuesta.

- \*\*Error:\*\* mensaje comprensible cuando algo sale mal.

- \*\*Resultado:\*\* datos bien formateados cuando todo sale bien.

- Evitar congelamientos aparentes:

- No dejar la pantalla “vacía” sin explicación mientras se carga.

- Informar al usuario sobre la necesidad de estar conectado a internet (si aplica).

### 3.4. Competencias de documentación y pruebas

- Registrar:

- La URL del endpoint utilizado.

- Ejemplo de la respuesta JSON (aunque sea recortada).

- Los pasos para reproducir la funcionalidad.

- Realizar pruebas manuales:

- App con internet → datos cargan correctamente.

- App sin internet → se muestra un mensaje de error.

## 4. Organización general sugerida de la carpeta `Semana8`

Una organización ordenada para la carpeta `Semana8` podría ser:

```text

Semana8/

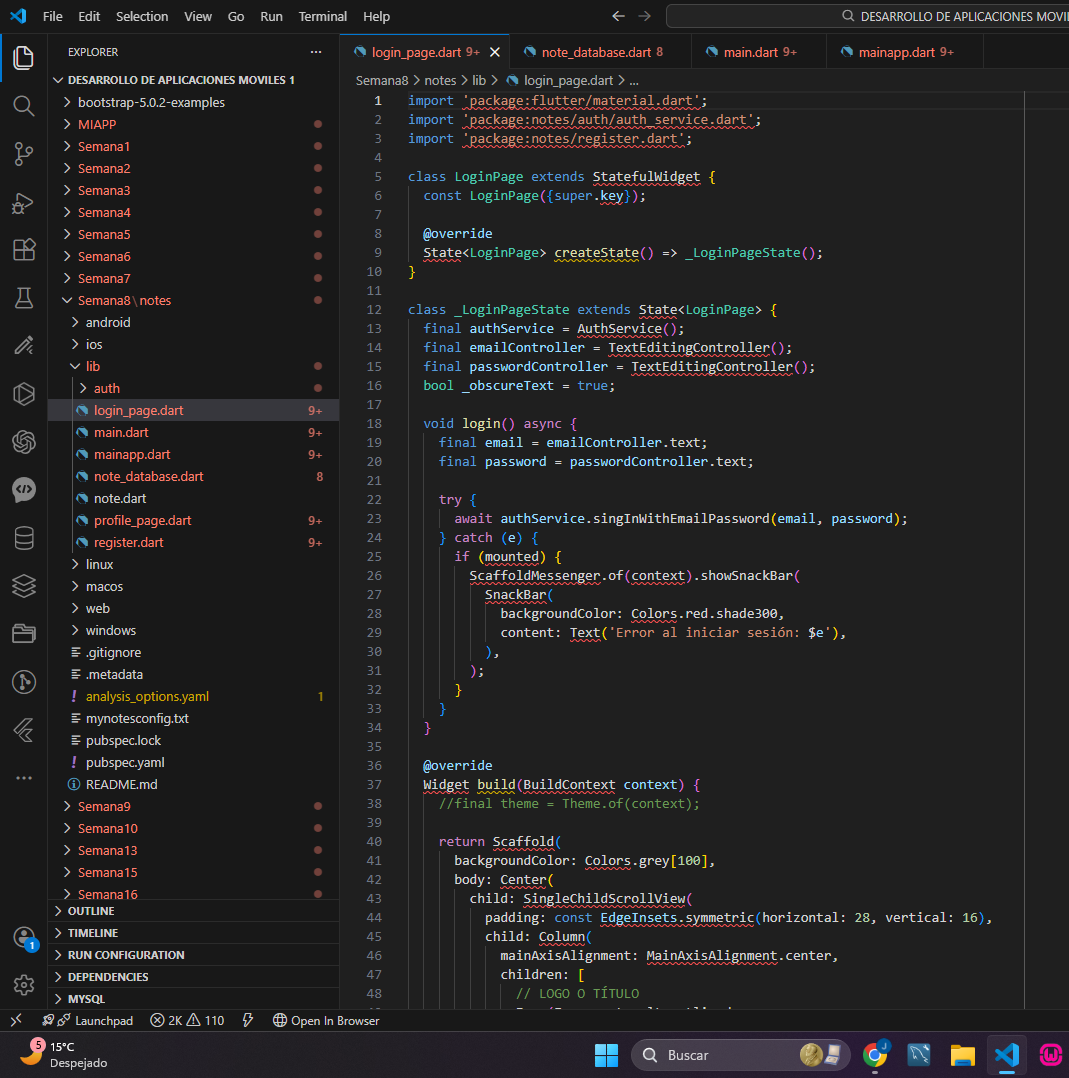
├─ docs/ # Teoría de APIs, HTTP, JSON, guías de práctica

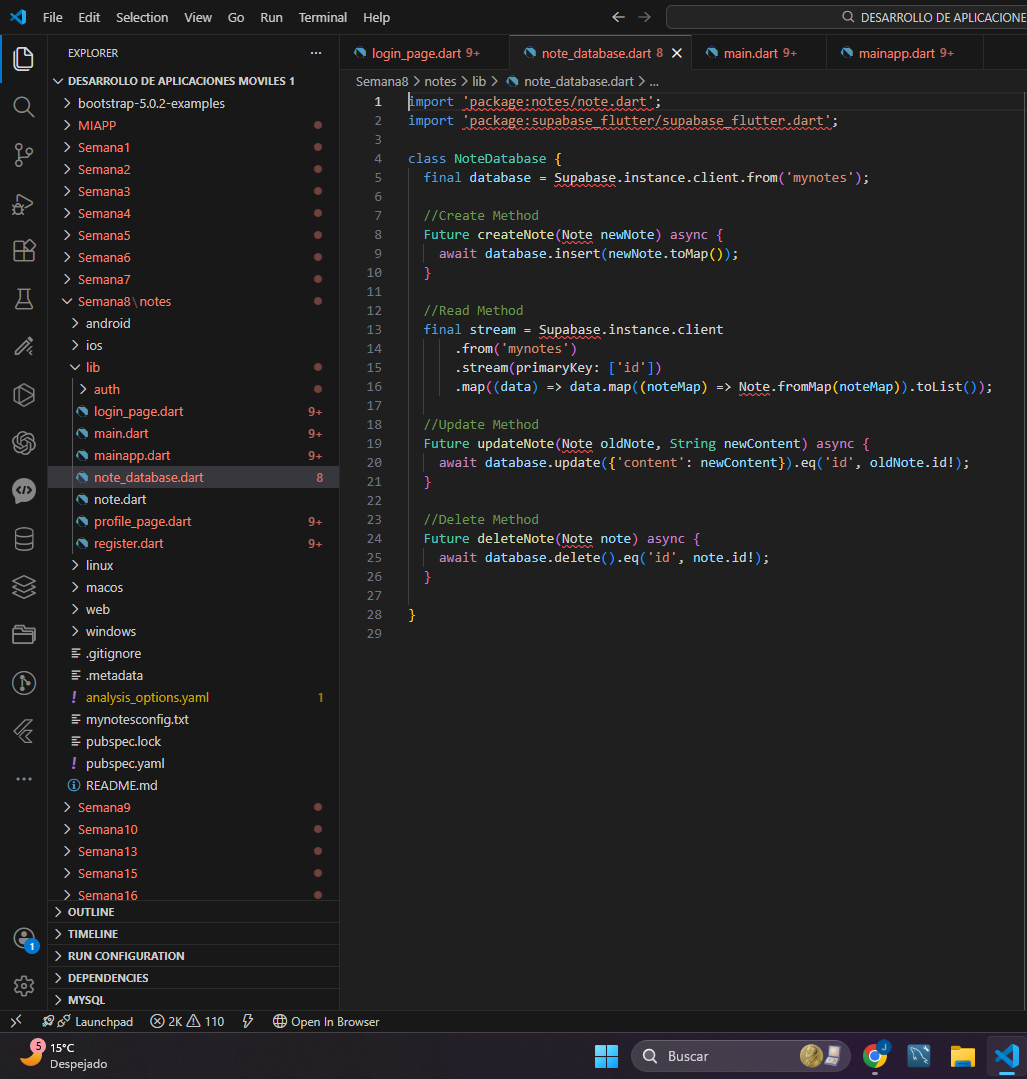
├─ ejemplos\_api/ # Ejemplos guiados de consumo de servicios web

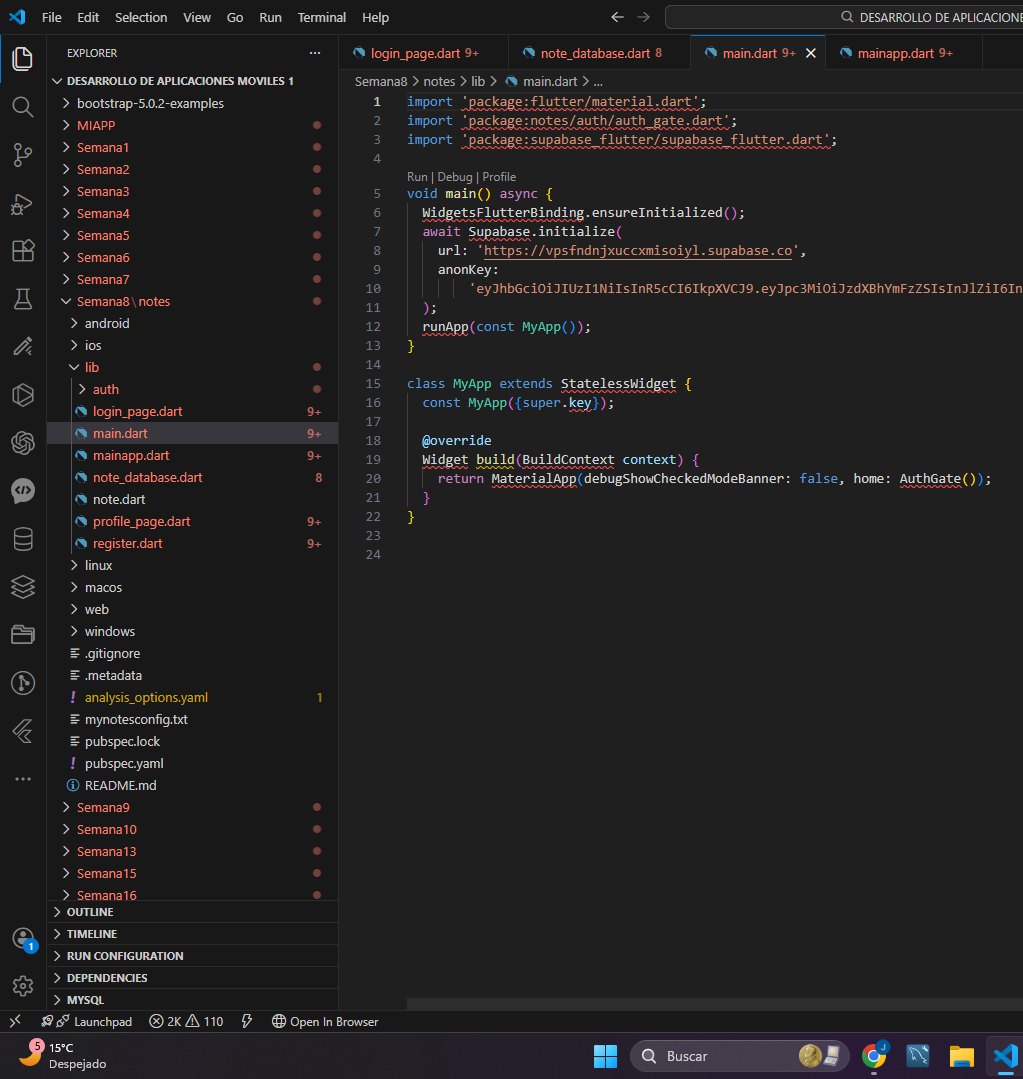
├─ proyecto\_semana8/ # Proyecto integrador con datos remotos

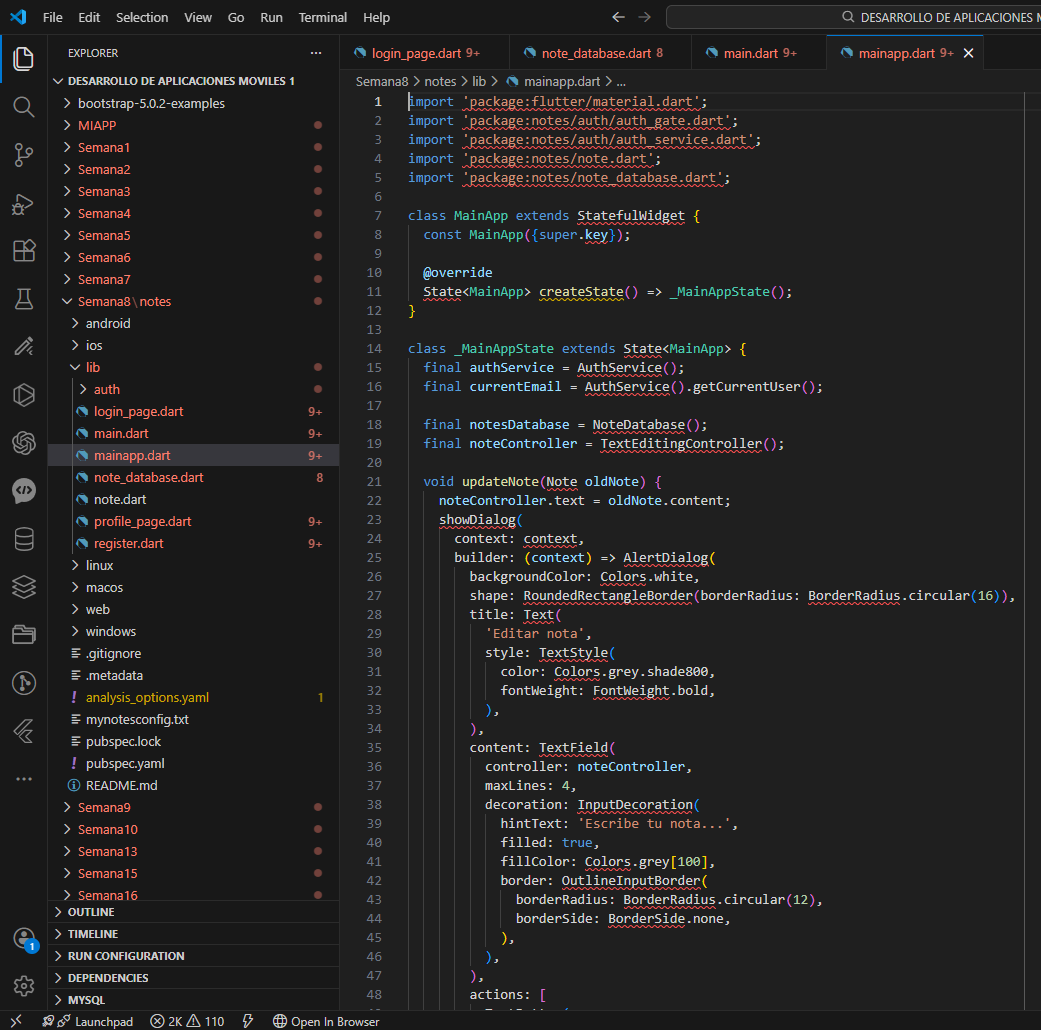
├─ ejercicios/ # Actividades prácticas a realizar

└─ evidencias/ # Capturas, reportes y entregas de Semana 8









# # Semana 9 – Proyecto integrador I: prototipo funcional de la aplicación móvil

> Carpeta: `Semana9` del repositorio `desarrolloapps1`

> Materia: Desarrollo de Aplicación Móvil I

> Periodo: Agosto–Diciembre 2025

La carpeta \*\*Semana9\*\* marca un punto clave dentro del curso de \*\*Desarrollo de Aplicación Móvil I\*\*:

es el momento en el que todos los conceptos vistos en semanas anteriores comienzan a integrarse en un \*\*proyecto más completo y coherente\*\*, que se parece ya a una aplicación real lista para usarse.

Hasta este momento, el recorrido ha sido:

- \*\*Semana 1:\*\* Entorno de desarrollo, organización del repositorio y primer programa / app básica.

- \*\*Semana 2:\*\* Lógica de programación, estructuras de control y decisiones.

- \*\*Semana 3:\*\* Interfaces gráficas iniciales y componentes básicos de UI.

- \*\*Semana 4:\*\* Navegación entre pantallas y manejo de eventos.

- \*\*Semana 5:\*\* Listas, colecciones y pantallas dinámicas.

- \*\*Semana 6:\*\* Formularios, validaciones y retroalimentación al usuario.

- \*\*Semana 7:\*\* Persistencia de datos local y ciclo de vida básico de la app.

- \*\*Semana 8:\*\* Consumo de servicios web (APIs) y uso de datos remotos en la aplicación.

En \*\*Semana 9\*\*, el objetivo es \*\*conectar todas esas piezas\*\* en un \*\*prototipo funcional\*\* de aplicación, al que llamamos:

> \*\*Proyecto integrador I – Versión inicial funcional (MVP / prototipo)\*\*

La carpeta `Semana9` se utiliza como contenedor del \*\*primer gran avance del proyecto final\*\*, donde el alumno debe demostrar que es capaz de combinar:

- Pantallas y navegación.

- Formularios y validaciones.

- Listas y detalle de información.

- Manejo de datos locales y/o remotos.

- Persistencia básica donde aplique.

## 1. Rol de la Semana 9 dentro del curso

La \*\*Semana 9\*\* funciona como un \*\*punto de consolidación\*\*:

- Deja de haber ejercicios totalmente aislados (pantalla por aquí, lista por allá) y pasa a haber \*\*un solo proyecto central\*\*.

- Se comienza a trabajar con la idea de \*\*“producto mínimo viable” (MVP)\*\*:

- Una app que \*\*no está terminada al 100 %\*\*, pero que:

- Compila,

- Se ejecuta,

- Permite realizar ciertas acciones reales (capturar, listar, consultar datos),

- Tiene un flujo lógico desde inicio hasta alguna funcionalidad concreta.

En esta semana el enfoque principal ya no es solo “probar un concepto”, sino \*\*construir una versión inicial completa\*\* del proyecto que se seguirá mejorando en semanas posteriores.

## 2. Objetivos de aprendizaje de la Semana 9

### 2.1. Objetivo general

> \*\*Construir un primer prototipo funcional de la aplicación móvil integradora, que incluya varias pantallas, navegación, formularios, uso de listas y manejo básico de datos (locales y/o remotos), organizado en un proyecto limpio y entendible.\*\*

### 2.2. Objetivos específicos

Al finalizar Semana 9, se espera que el estudiante:

1. \*\*Integre\*\* en un solo proyecto los elementos trabajados por separado:

- Navegación.

- Formularios.

- Listas.

- Persistencia y/o consumo de datos.

2. Diseñe un \*\*flujo de uso coherente\*\*:

- Pantalla inicial → menú / login / portada.

- Acceso a módulos específicos (listas, formularios, consultas).

- Salida o cierre del flujo.

3. Estructure el proyecto en \*\*módulos o capas simples\*\*:

- Archivos por pantalla.

- Archivos o clases por modelo de datos.

- Opcionalmente, archivos para servicios (API, almacenamiento, etc.).

4. Aplique \*\*buenas prácticas mínimas\*\*:

- Nombres descriptivos en archivos y clases.

- Comentarios breves donde sea necesario.

- Código legible (indentación, separación de secciones).

5. Genere \*\*evidencias claras\*\*:

- Capturas de pantalla de las principales pantallas.

- Capturas del código principal.

- Descripción de qué hace el prototipo en esta versión.

## 3. Competencias que se refuerzan en Semana 9

### 3.1. Competencias técnicas

- Integración de varias funcionalidades en un solo proyecto:

- Formularios con validación.

- Listas que muestran datos.

- Navegación entre múltiples pantallas.

- Posible uso de datos remotos o persistentes.

- Uso ordenado del entorno de desarrollo (IDE) para:

- Organizar carpetas y archivos.

- Compilar y ejecutar el proyecto repetidamente.

- Detectar y corregir errores de integración.

### 3.2. Competencias de diseño y arquitectura

- Pensar la aplicación en términos de \*\*módulos\*\*:

- Pantallas como componentes independientes.

- Modelos de datos reutilizables.

- Servicios o utilidades agrupados.

- Empezar a visualizar la app como un sistema:

- ¿Qué parte se encarga de la lógica?

- ¿Qué parte se encarga de la interfaz?

- ¿Dónde se reciben y se almacenan datos?

### 3.3. Competencias de experiencia de usuario (UX)

- Definir un \*\*flujo de navegación claro\*\*:

- El usuario debe entender, sin explicación externa, qué hace la app.

- Botones con textos claros (no solo “OK”, “Aceptar” genérico).

- Evitar pantallas “muertas” sin opciones de regreso o avance.

- Unificar el estilo visual básico:

- Tipografías, colores, tamaños de títulos y textos.

- Ubicación de botones principales.

### 3.4. Competencias de documentación y trabajo profesional

- Crear documentación mínima del proyecto:

- Descripción del objetivo de la app.

- Lista de pantallas implementadas.

- Funcionalidades que ya funcionan y las que están en desarrollo.

- Desarrollar el hábito de:

- \*\*Versionar\*\* el proyecto por semanas (Semana7, Semana8, Semana9…).

- Mantener un orden de avances que luego pueda revisarse o presentar como evidencia.

## 4. Organización general sugerida de la carpeta `Semana9`

Aunque la estructura real puede variar, una organización recomendada para `Semana9` es:

```text

Semana9/

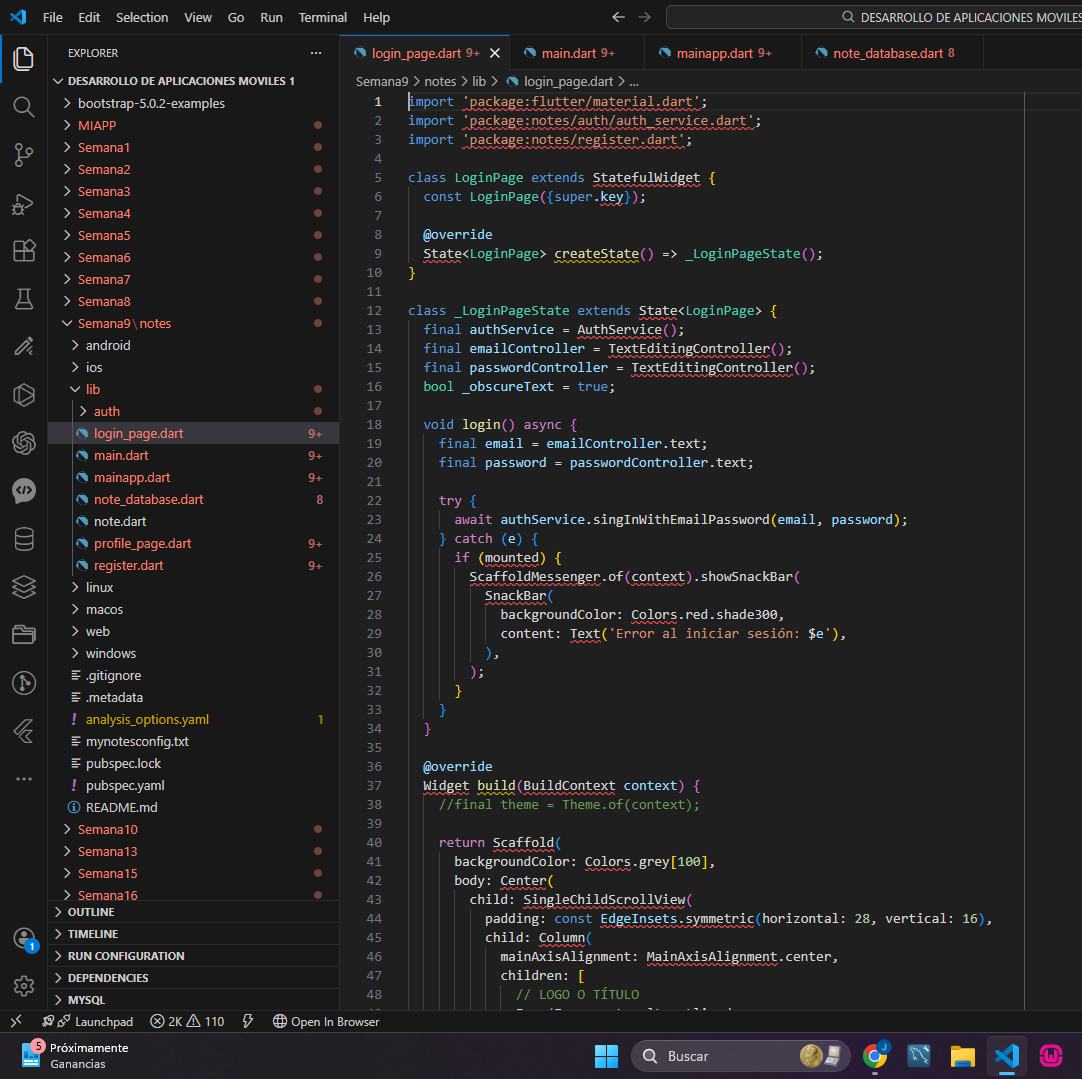
├─ docs/ # Documentación de diseño y descripción del proyecto integrador

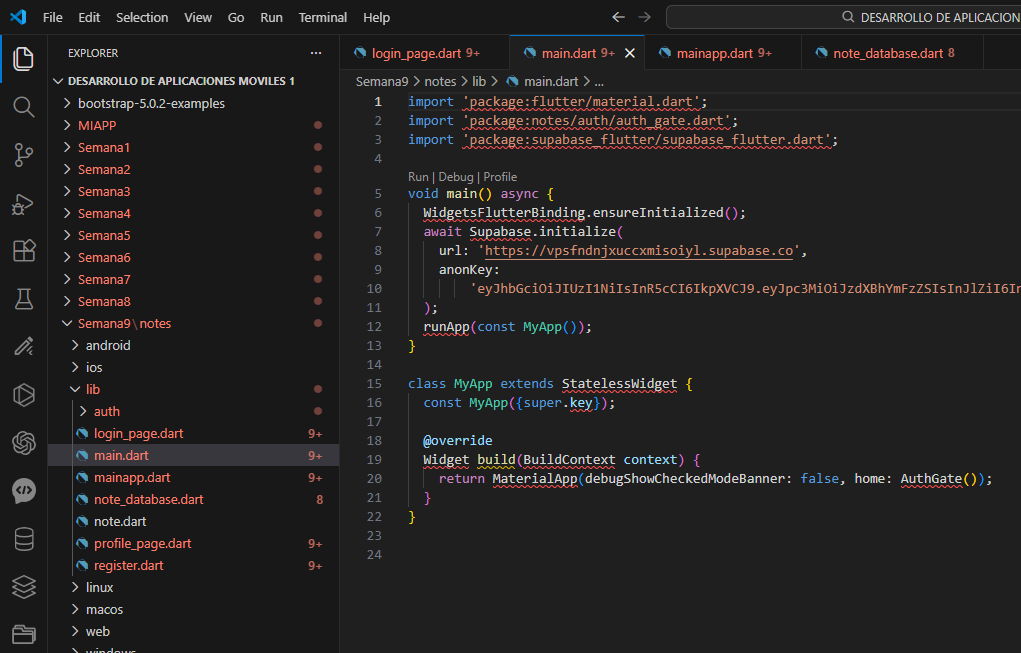
├─ proyecto\_integrador/ # Código fuente del prototipo funcional

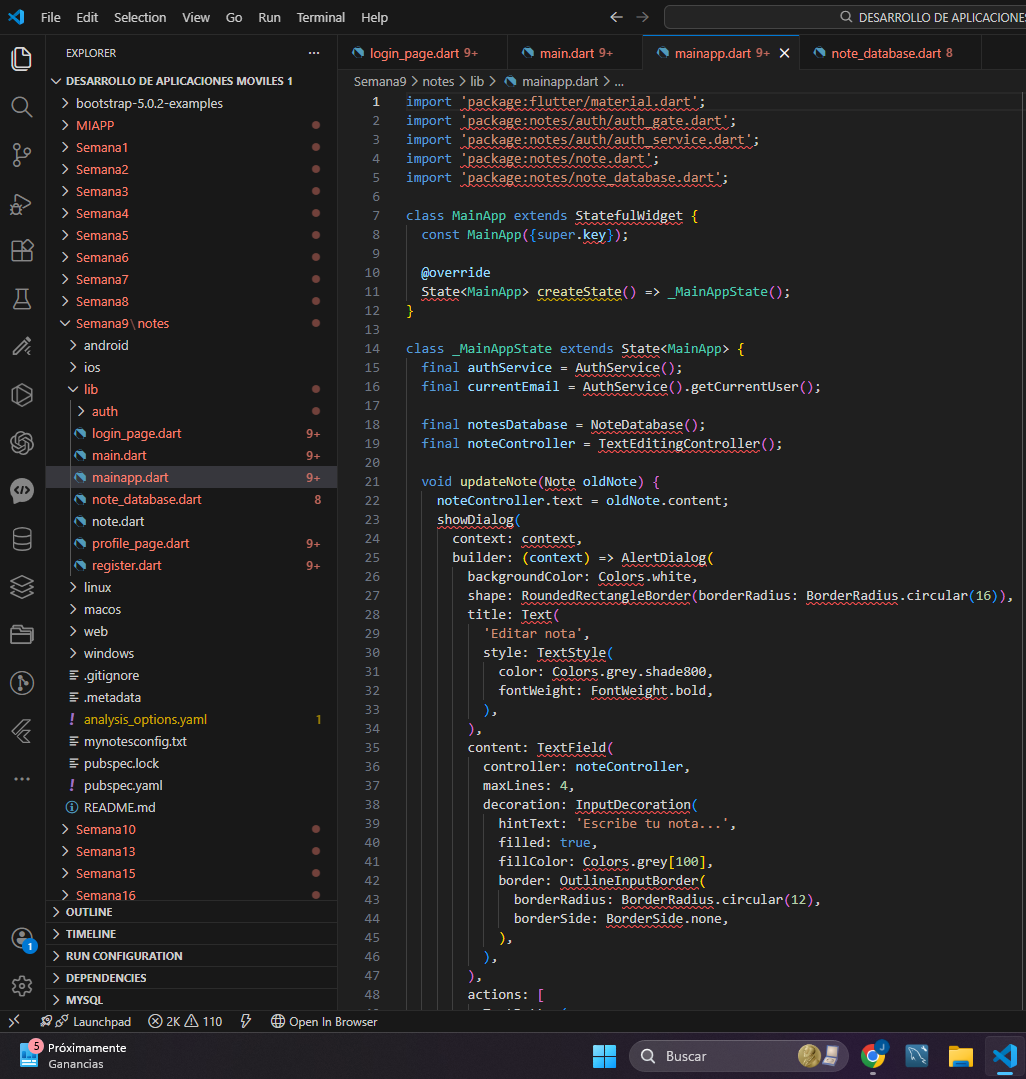
├─ recursos\_ui/ # Imágenes, íconos, mockups o prototipos de diseño (opcional)

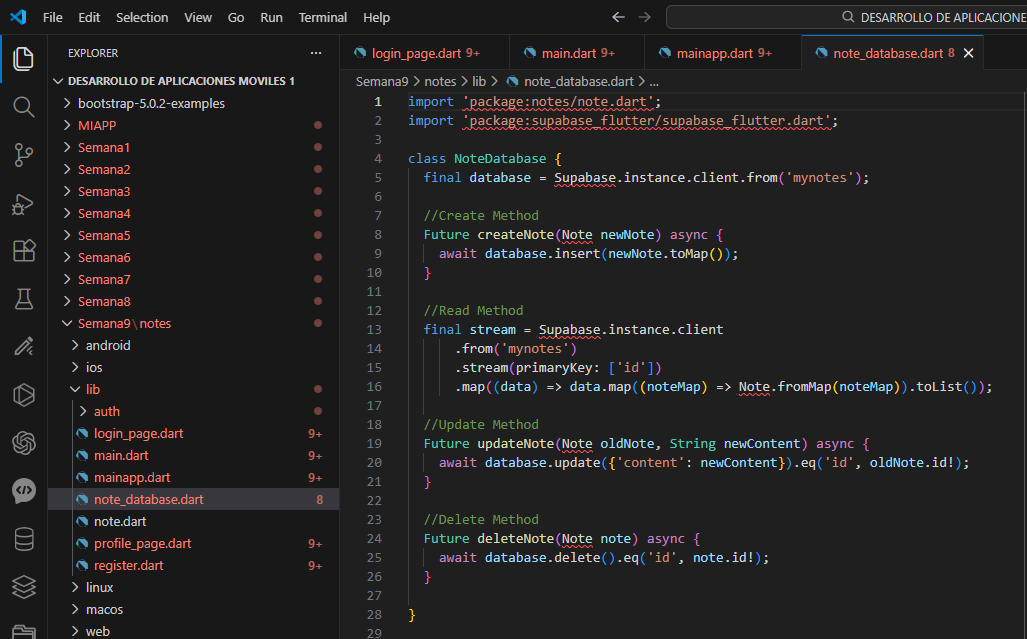
├─ ejercicios\_apoyo/ # Tareas pequeñas relacionadas con el proyecto (opcional)

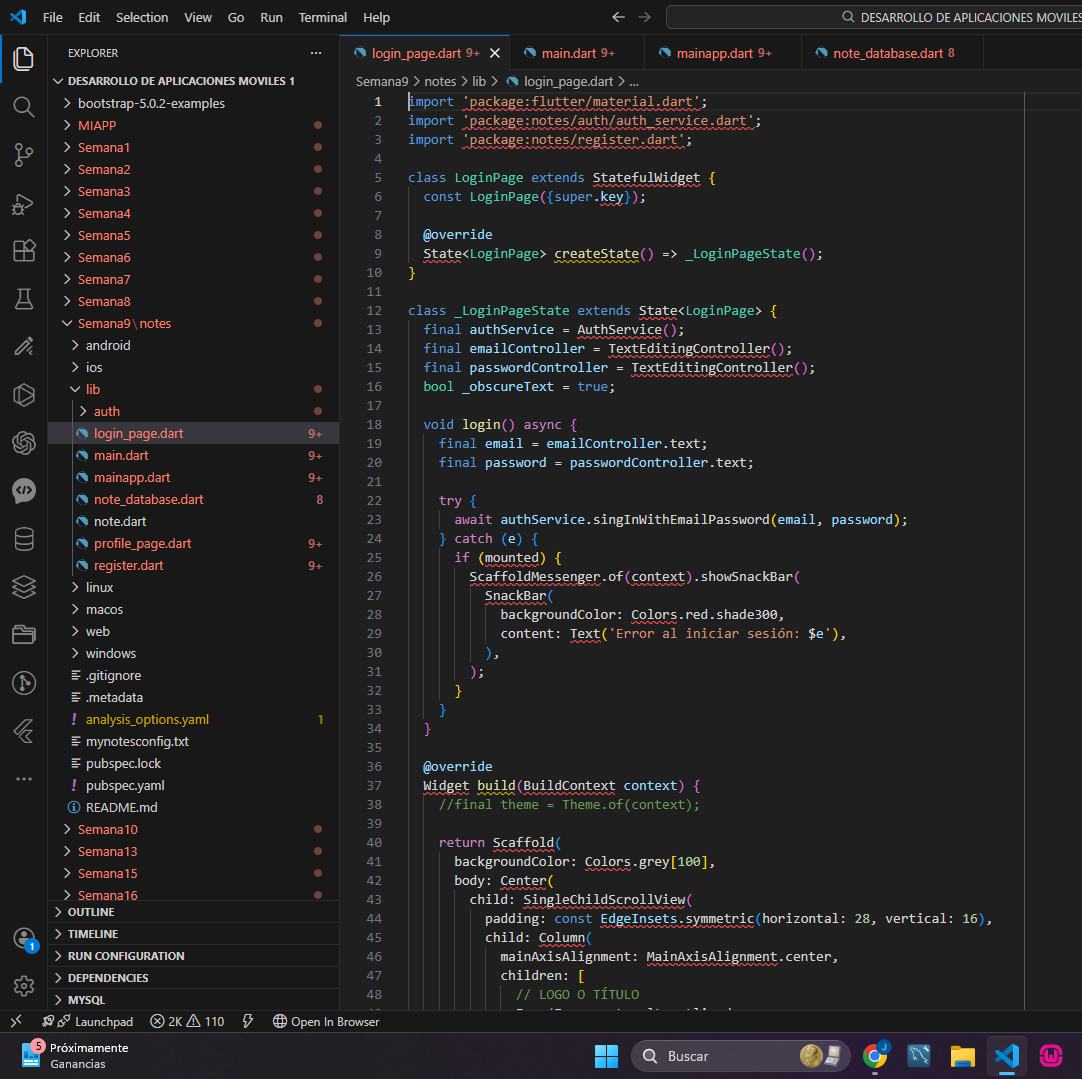
└─ evidencias/ # Capturas de pantalla, reportes y resúmenes de la semana











# # Semana 10 – Cierre del curso, pruebas, documentación y entrega del proyecto integrador

> Carpeta: `Semana10` del repositorio `desarrolloapps1`

> Materia: Desarrollo de Aplicación Móvil I

> Periodo: Agosto–Diciembre 2025

La carpeta \*\*Semana10\*\* marca la \*\*fase de cierre\*\* del curso de \*\*Desarrollo de Aplicación Móvil I\*\*.

En este punto, el estudiante ya ha trabajado, a lo largo de las semanas anteriores, en:

- \*\*Semana 1:\*\* Entorno de desarrollo y organización del repositorio.

- \*\*Semana 2:\*\* Lógica básica y estructuras de control.

- \*\*Semana 3:\*\* Interfaces de usuario y componentes gráficos iniciales.

- \*\*Semana 4:\*\* Navegación entre pantallas y manejo de eventos.

- \*\*Semana 5:\*\* Listas y manejo de colecciones de datos.

- \*\*Semana 6:\*\* Formularios, validaciones y retroalimentación al usuario.

- \*\*Semana 7:\*\* Persistencia de datos local.

- \*\*Semana 8:\*\* Consumo de servicios web (APIs) e integración de datos remotos.

- \*\*Semana 9:\*\* Proyecto integrador I – Prototipo funcional de la aplicación.

En \*\*Semana 10\*\*, el enfoque principal es:

> \*\*Pulir, probar, documentar y preparar la entrega final de la aplicación móvil integradora.\*\*

Es decir, se pasa de “solo hacer que funcione” a \*\*cerrar el ciclo completo de desarrollo\*\*, con:

- Pruebas básicas (funcionales, de flujo, de validaciones).

- Corrección de errores y pequeños detalles.

- Documentación técnica y de usuario.

- Preparación de evidencias y presentación final del proyecto.

## 1. Rol de la Semana 10 dentro del curso

La \*\*Semana10\*\* funciona como la etapa de:

- \*\*Consolidación final\*\* de los conocimientos y habilidades adquiridas.

- \*\*Cierre del proyecto integrador\*\*, donde se asegura que:

- La app compila sin errores.

- El flujo de uso es coherente.

- Las principales funcionalidades definidas al inicio están implementadas.

- \*\*Formalización de la entrega\*\*, con:

- Documentación.

- Evidencias.

- (Opcional) Presentación ante el grupo o docente.

En lugar de introducir muchos conceptos nuevos, Semana10 se centra en que el estudiante:

- Revise lo que ya hizo.

- Mejore la calidad del código y de la experiencia de usuario.

- Aprenda la importancia de \*\*documentar y presentar\*\* su trabajo profesionalmente.

---

## 2. Objetivos de aprendizaje de la Semana 10

### 2.1. Objetivo general

> \*\*Completar, probar y documentar la versión final del proyecto integrador de la aplicación móvil, de forma que pueda considerarse un producto mínimo viable (MVP) listo para ser presentado y evaluado.\*\*

### 2.2. Objetivos específicos

Al finalizar Semana10, se espera que el estudiante:

1. Revise y \*\*ajuste el flujo de navegación\*\* de la app:

- Pantalla inicial → menús → módulos principales → salida.

- Posibilidad de regresar desde pantallas secundarias sin “atascos”.

2. Verifique el correcto funcionamiento de:

- Formularios (captura y validación de datos).

- Listas (visualización, selección de elementos).

- Persistencia (datos que se guardan y se recuperan).

- Consumo de datos remotos (si el proyecto usa APIs).

3. Aplique un \*\*plan de pruebas básicas\*\*:

- Pruebas de caso feliz (cuando todo se hace correctamente).

- Pruebas de errores comunes (datos incorrectos, campos vacíos, sin internet, etc.).

4. Mejore la \*\*experiencia de usuario (UX)\*\*:

- Mensajes claros.

- Botones bien nombrados.

- Diseño mínimamente uniforme (colores, tipografías, espaciado).

5. Prepare la \*\*documentación del proyecto\*\*:

- Descripción general de la app.

- Lista de pantallas y funcionalidades.

- Requerimientos básicos para ejecutar el proyecto.

- Problemas conocidos y posibles mejoras futuras.

6. Genere \*\*evidencias finales\*\*:

- Capturas de pantalla representativas.

- (Opcional) Video corto de uso.

- Reporte o README técnico.

## 3. Competencias que se refuerzan en Semana 10

### 3.1. Competencias técnicas

- Depuración (debugging) y resolución de errores.

- Ajustes finales de código:

- Eliminación de funciones o variables sin uso.

- Corrección de pequeños bugs en validaciones y navegación.

- Preparación del proyecto para:

- Ser ejecutado por el docente.

- Ser compartido con otros (compañeros, futuros cursos).

### 3.2. Competencias de diseño y UX

- Coherencia visual:

- El estilo de la app debe ser homogéneo.

- Evitar pantallas “diferentes” sin intención (colores aleatorios, textos desalineados).

- Claridad en la interacción:

- Botones con textos descriptivos.

- Mensajes sencillos, sin tecnicismos.

- Eliminación de elementos confusos:

- Botones que no hacen nada.

- Pantallas vacías o de prueba que ya no se usan.

### 3.3. Competencias de documentación

- Creación de un \*\*README\*\* o documento de proyecto que incluya:

- Nombre y descripción de la app.

- Objetivo general y problema que resuelve.

- Tecnologías utilizadas.

- Instrucciones para ejecutar el proyecto.

- Descripción de pantallas y funcionalidades.

- Limitaciones y mejoras futuras.

- Preparación de informes o reportes:

- Resumen de avances realizados desde la versión inicial (Semana9) hasta la final (Semana10).

- Principales cambios y correcciones.

### 3.4. Competencias de presentación y comunicación

- Explicar el proyecto de forma breve y clara:

- ¿Qué hace la app?

- ¿Para quién está pensada?

- ¿Qué módulos o secciones tiene?

- Responder preguntas básicas:

- ¿Cómo se implementó cierta funcionalidad?

- ¿Qué tecnologías o librerías se usaron?

## 4. Organización general sugerida de la carpeta `Semana10`

Una organización ordenada de `Semana10` podría ser:

```text

Semana10/

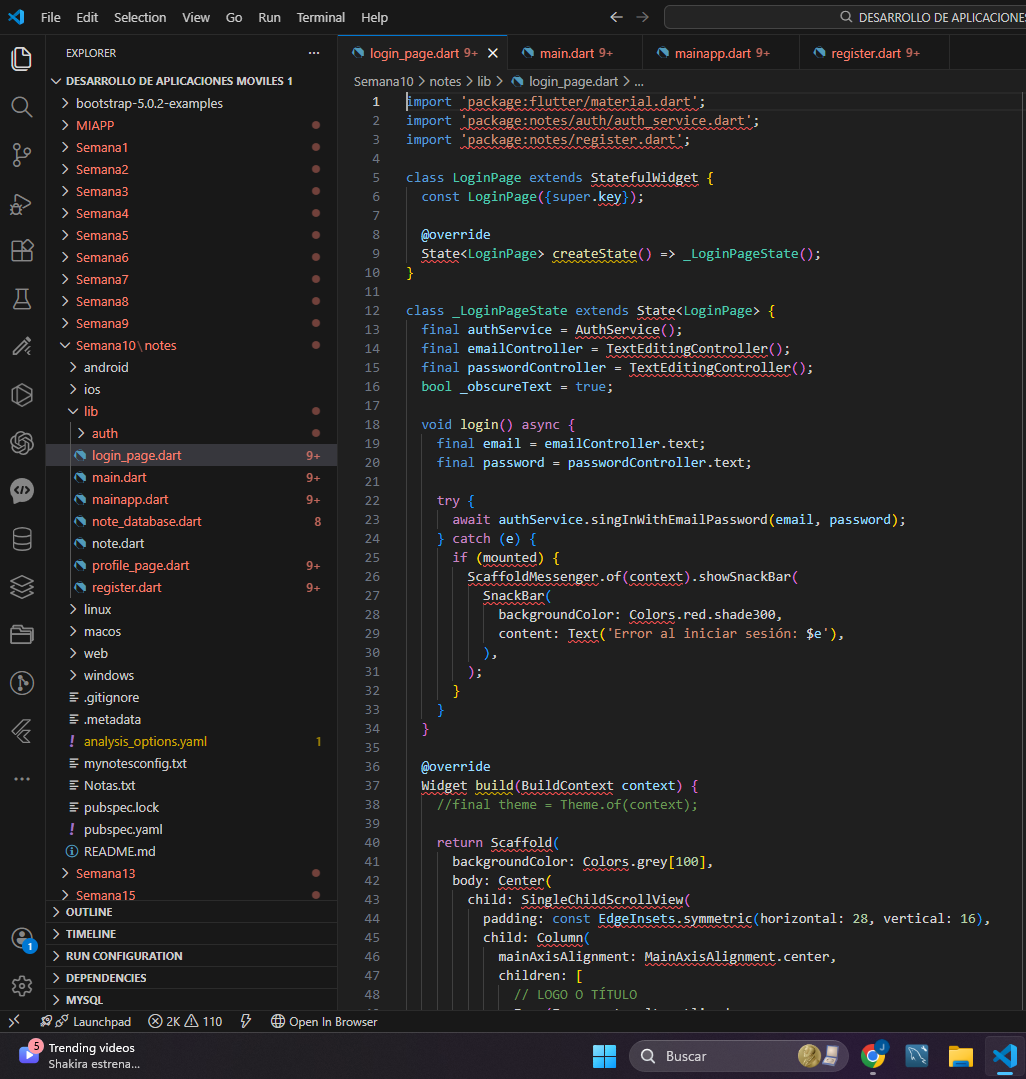
├─ docs/ # Documentos de descripción, diseño y reporte final

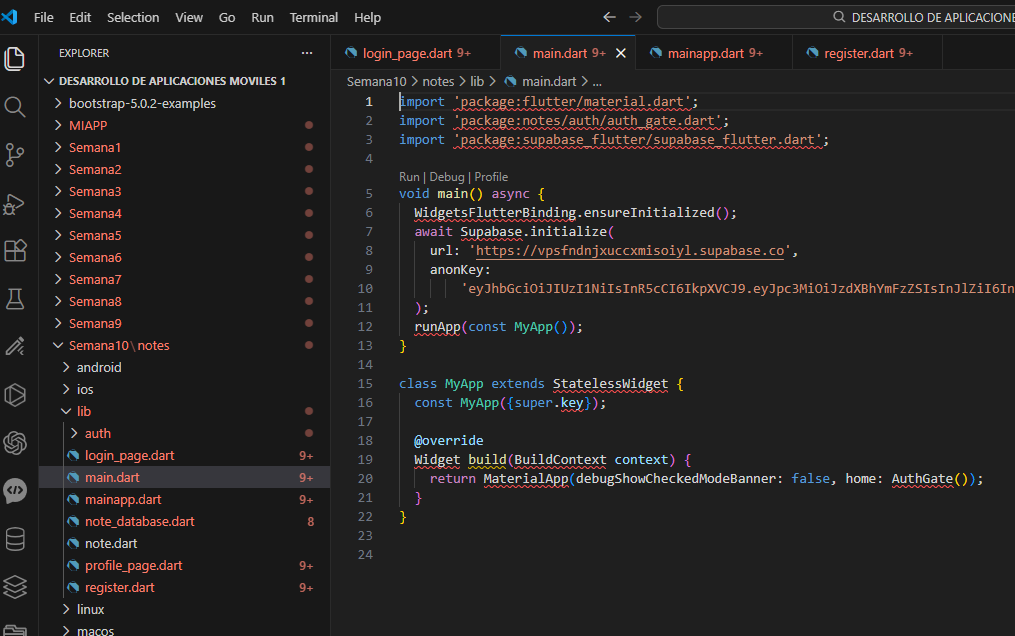
├─ proyecto\_final/ # Versión final del proyecto integrador

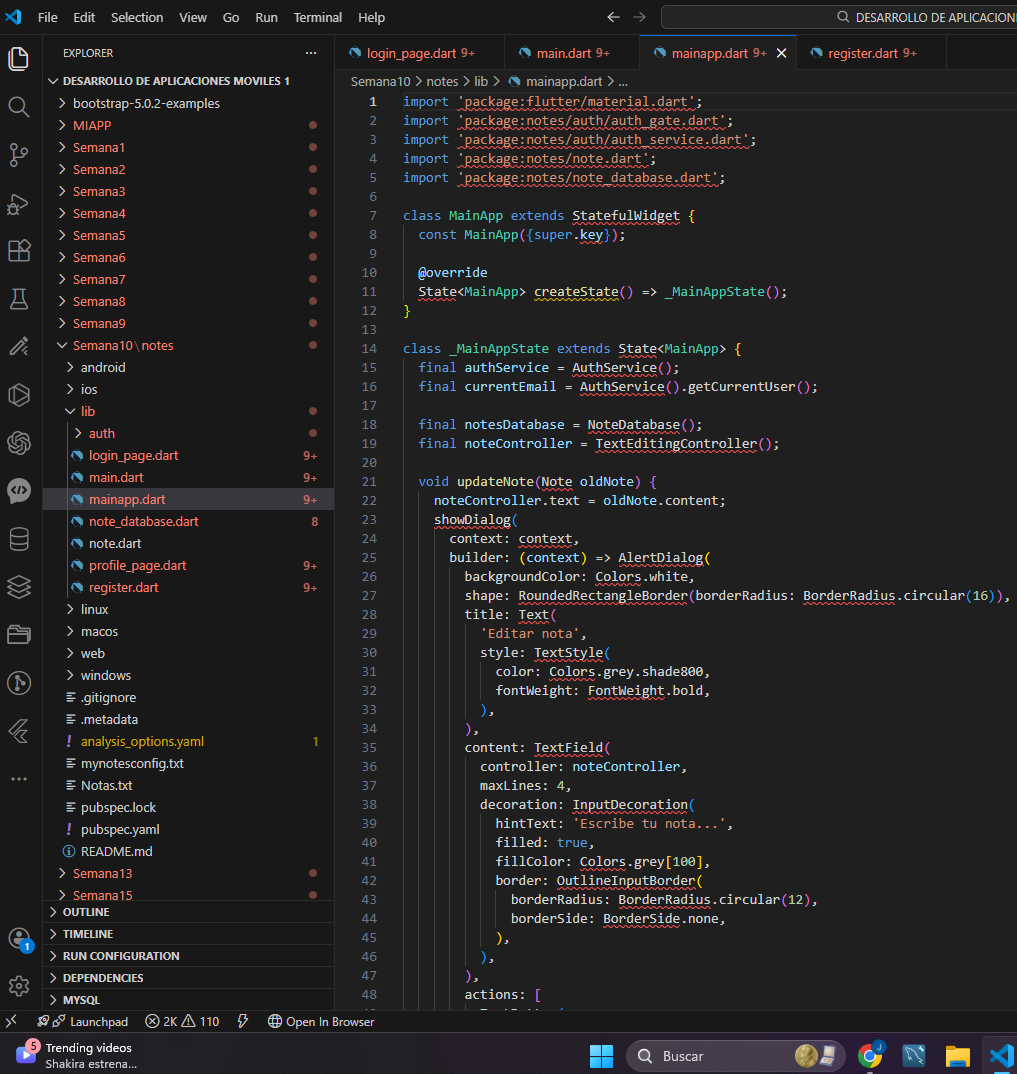
├─ pruebas/ # Planes y resultados de pruebas (opcional)

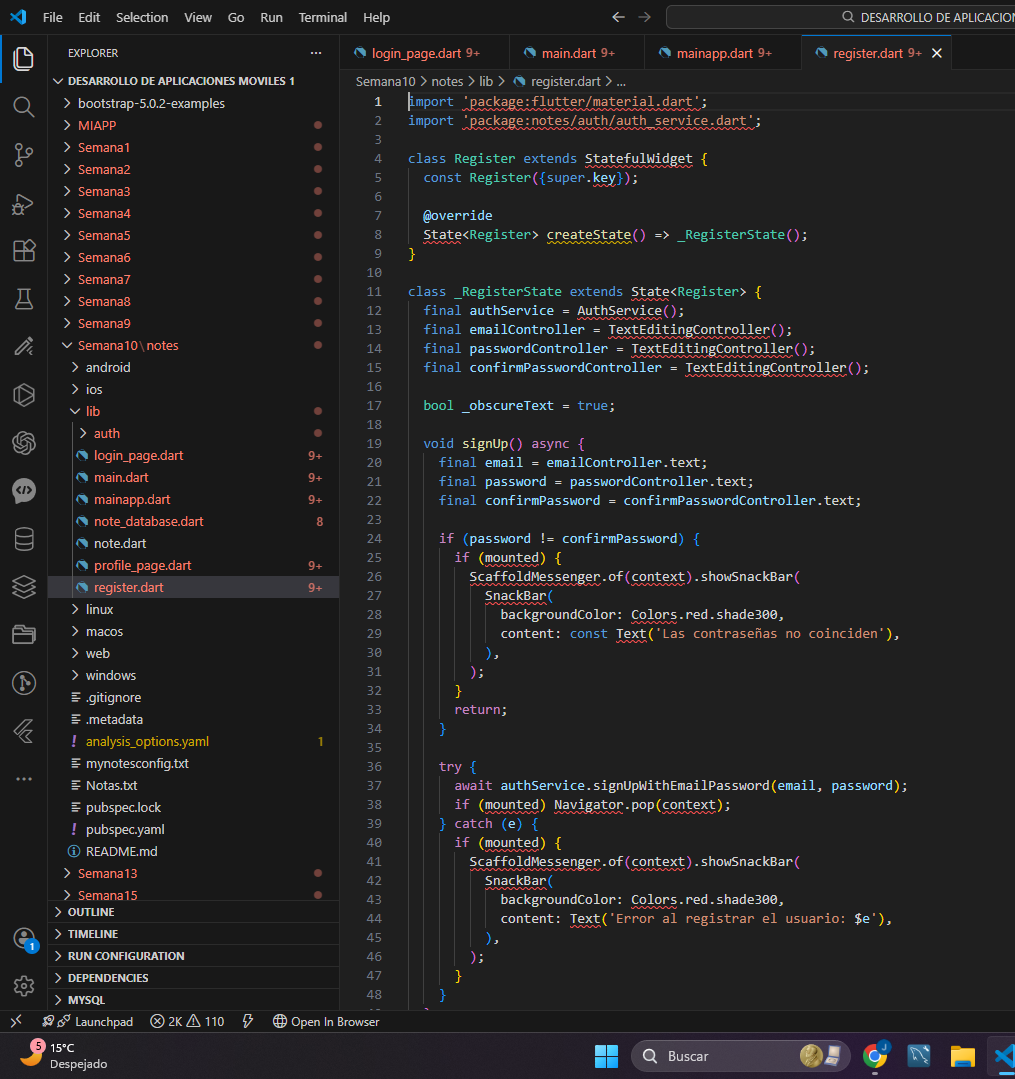
├─ presentacion/ # Materiales para exponer el proyecto (opcional)

└─ evidencias/ # Capturas de pantalla, videos, reportes y README final









# # Semana 11 – Retroalimentación, mejora continua y portafolio profesional de aplicaciones móviles

> Carpeta: `Semana11` del repositorio `desarrolloapps1`

> Materia: Desarrollo de Aplicación Móvil I

> Periodo: Agosto–Diciembre 2025

La carpeta \*\*Semana11\*\* corresponde a una etapa de \*\*cierre ampliado, retroalimentación y consolidación\*\* dentro del curso de \*\*Desarrollo de Aplicación Móvil I\*\*.

Aunque la estructura principal del curso se completa formalmente en las primeras 10 semanas, esta semana adicional se dedica a:

- Revisar críticamente el trabajo realizado.

- Mejorar la calidad técnica y estética de la aplicación.

- Construir un \*\*portafolio profesional\*\* a partir del proyecto integrador.

- Ordenar la evidencia del curso de forma clara y presentable.

En esta etapa, el objetivo deja de ser “seguir agregando más cosas a la app” y pasa a ser:

> \*\*Convertir el trabajo de semestre en un producto presentable, bien documentado y que pueda servir como evidencia académica y profesional.\*\*

## 1. Contexto general de la Semana 11 en el curso

Antes de entrar en detalle, es importante recordar el camino recorrido:

- \*\*Semana 1:\*\* Entorno de desarrollo, estructura del repositorio, primer programa.

- \*\*Semana 2:\*\* Lógica básica, estructuras de control, decisiones y ciclos.

- \*\*Semana 3:\*\* Interfaces gráficas iniciales, componentes UI básicos.

- \*\*Semana 4:\*\* Navegación entre pantallas y manejo de eventos.

- \*\*Semana 5:\*\* Listas, colecciones y pantallas dinámicas.

- \*\*Semana 6:\*\* Formularios, validaciones y retroalimentación básica.

- \*\*Semana 7:\*\* Persistencia de datos local y ciclo de vida de la app.

- \*\*Semana 8:\*\* Consumo de servicios web/APIs y datos remotos.

- \*\*Semana 9:\*\* Proyecto integrador I – Prototipo funcional.

- \*\*Semana 10:\*\* Proyecto integrador final, pruebas y documentación básica.

En \*\*Semana 11\*\*, todo lo anterior se retoma desde una perspectiva de:

- \*\*Reflexión:\*\* ¿Qué aprendí realmente? ¿Qué mejoraría si volviera a empezar?

- \*\*Pulido:\*\* Mejoras pequeñas que suben bastante la calidad percibida.

- \*\*Presentación:\*\* Cómo mostrar el trabajo a un profesor, reclutador o compañero.

## 2. Propósito general de la Semana 11

El propósito central de la carpeta `Semana11` es:

> \*\*Transformar el proyecto integrador y las evidencias del curso en un portafolio organizado, bien documentado y con una versión pulida de la aplicación, listo para ser mostrado como logro académico y comienzo de portafolio profesional.\*\*

De este propósito se desprenden varios objetivos específicos:

1. Revisar el proyecto integrador con enfoque de \*\*calidad\*\* (no solo funcionalidad).

2. Aplicar \*\*refactorización básica\*\* al código:

- Mejorar organización.

- Eliminar redundancias.

- Comentar partes críticas.

3. Completar y mejorar la \*\*documentación técnica y de usuario\*\*.

4. Organizar \*\*evidencias del curso\*\* por semana o por tema.

5. Elaborar un \*\*portafolio resumido\*\* con:

- Descripción de la app.

- Capturas clave.

- Aspectos técnicos aprendidos.

- Plan de mejoras futuras.

## 3. Competencias que se fortalecen en Semana 11

### 3.1. Competencias técnicas

- Refactorizar código:

- Renombrar variables, funciones y archivos para mayor claridad.

- Extraer funciones repetidas en utilidades comunes.

- Separar responsabilidades (por ejemplo, separar lógica de negocio de lógica de interfaz).

- Mejorar aspectos técnicos:

- Manejo un poco más fino de errores.

- Limpieza de warnings del IDE.

- Organización de imports, dependencias y comentarios.

### 3.2. Competencias de diseño y UX

- Uniformar la apariencia de la app:

- Mismo estilo de botones (color, tamaño, texto).

- Mismo estilo de títulos y encabezados.

- Espaciado adecuado entre elementos.

- Revisar la experiencia del usuario:

- Flujo claro desde inicio hasta la funcionalidad principal.

- Mensajes de error amigables y útiles.

- Minimizar clics innecesarios para lograr objetivos comunes.

### 3.3. Competencias de documentación

- Elaborar y/o pulir documentación que incluya:

- Descripción general del proyecto.

- Objetivo y problema que resuelve.

- Tecnologías utilizadas.

- Instrucciones para ejecutar la app.

- Lista de funcionalidades implementadas.

- Posibles mejoras futuras.

- Preparar resumen por semana o por módulo:

- Qué se aprendió.

- Qué se implementó.

- Cómo se conectó con el proyecto integrador.

### 3.4. Competencias de presentación profesional

- Construir un \*\*portafolio básico\*\* que pueda adjuntarse a:

- Curriculum vitae.

- Entrevistas técnicas básicas.

- Evidencias de titulación o prácticas.

- Aprender a “contar la historia” del proyecto:

- ¿De dónde nació la idea?

- ¿Qué retos técnicos se enfrentaron?

- ¿Qué se logró al final del curso?

## 4. Organización sugerida de la carpeta `Semana11`

Una estructura organizada para `Semana11` podría ser:

```text

Semana11/

├─ docs/ # Documentos de reflexión, mejora y portafolio

├─ proyecto\_ajustes\_finales/ # Versión refinada del proyecto con pequeños cambios

├─ portafolio/ # Material para portafolio (docs, imágenes, descripciones)

├─ rubricas\_y\_checklists/ # Criterios de evaluación, listas de verificación

└─ evidencias/ # Capturas finales, comparaciones antes/después

# # Semana 12 – Publicación, despliegue y mantenimiento de aplicaciones móviles

> Carpeta: `Semana12` del repositorio `desarrolloapps1`

> Materia: Desarrollo de Aplicación Móvil I

> Periodo: Agosto–Diciembre 2025

La carpeta \*\*Semana12\*\* representa la \*\*fase final y de cierre técnico\*\* del curso de \*\*Desarrollo de Aplicación Móvil I\*\*, enfocada especialmente en:

- Entender \*\*cómo se empaqueta una aplicación móvil\*\*.

- Conocer el proceso general de \*\*despliegue / publicación\*\*.

- Reflexionar sobre el \*\*mantenimiento, actualización y ciclo de vida largo\*\* de una app.

- Dejar el proyecto en un estado que pueda ser usado como \*\*base real para versiones futuras\*\* o para un curso de continuación (por ejemplo, Desarrollo de Aplicaciones Móviles II).

Mientras que en:

- \*\*Semana 9\*\* se armó el \*\*primer prototipo integrador\*\*,

- \*\*Semana 10\*\* se orientó a \*\*pruebas y documentación técnica\*\*,

- \*\*Semana 11\*\* se centró en \*\*retroalimentación, mejora continua y portafolio\*\*,

la \*\*Semana 12\*\* se enfoca en el paso que muchas veces se olvida en el aula, pero que es crucial en el mundo real:

> \*\*¿Cómo llevo mi app desde el entorno de desarrollo hasta las manos del usuario final, y cómo la mantengo viva en el tiempo?\*\*

## 1. Contexto de la Semana 12 dentro del curso

Para ubicar mejor el rol de `Semana12`, conviene recordar brevemente el camino del curso:

- \*\*Semana 1:\*\* Entorno de desarrollo, repositorio y primer programa.

- \*\*Semana 2:\*\* Lógica y estructuras de control.

- \*\*Semana 3:\*\* Interfaces y componentes básicos.

- \*\*Semana 4:\*\* Navegación y eventos.

- \*\*Semana 5:\*\* Listas y vistas dinámicas.

- \*\*Semana 6:\*\* Formularios y validaciones.

- \*\*Semana 7:\*\* Persistencia de datos local.

- \*\*Semana 8:\*\* Consumo de servicios web (APIs).

- \*\*Semana 9:\*\* Proyecto integrador I – Prototipo funcional.

- \*\*Semana 10:\*\* Cierre funcional, pruebas y documentación.

- \*\*Semana 11:\*\* Retroalimentación, pulido y portafolio profesional.

Sobre esa base, \*\*Semana 12\*\* busca:

- Completar el ciclo: \*\*desarrollo → prueba → documentación → empaquetado → despliegue → mantenimiento\*\*.

- Introducir, aunque sea a nivel conceptual/práctico básico, lo que implica:

- Generar un \*\*build de producción\*\*.

- Configurar \*\*versionado\*\*.

- Entender el flujo general para publicar en una \*\*tienda de apps\*\* (Google Play, etc.), aunque no siempre se haga la publicación real en el curso.

- Pensar en \*\*mantenimiento, actualizaciones y soporte\*\*.

## 2. Propósito general de la Semana 12

El propósito central de la carpeta `Semana12` es:

> \*\*Guiar al estudiante en los pasos finales necesarios para empaquetar su aplicación, entender el proceso de despliegue/publicación y reflexionar sobre el mantenimiento y evolución futura del proyecto.\*\*

De este propósito se desprenden algunos \*\*objetivos específicos\*\*:

1. Conocer el flujo general para \*\*generar un ejecutable o paquete instalable\*\* (por ejemplo, un APK en Android).

2. Entender aspectos básicos de \*\*firmado, versión y configuración de build\*\*.

3. Revisar consideraciones de \*\*rendimiento, tamaño y permisos\*\* antes de publicar.

4. Elaborar una \*\*versión final “para usuario”\*\* del proyecto integrador, con:

- Mensajes limpios.

- Errores críticos corregidos.

- Elementos de prueba temporales eliminados.

5. Documentar el \*\*proceso de empaquetado y despliegue\*\*:

- Qué comandos o botones del IDE se usaron.

- Qué archivos resultantes se generaron.

6. Reflexionar sobre el \*\*mantenimiento a largo plazo\*\*:

- Cómo se manejarían futuras versiones.

- Cómo se atendería retroalimentación de usuarios.

- Qué cambios se priorizarían si la app siguiera creciendo.

## 3. Competencias que se fortalecen en Semana 12

### 3.1. Competencias técnicas

- Comprensión de la diferencia entre:

- \*\*Versión de desarrollo\*\* (debug).

- \*\*Versión de producción\*\* (release).

- Manejo básico de:

- Configuración de \*\*versionCode / versionName\*\* o equivalente.

- Generación de un paquete instalable (APK, AAB, etc., según el entorno).

- Identificación de:

- Permisos usados por la app.

- Posibles advertencias del IDE antes de compilar para producción.

### 3.2. Competencias de ingeniería de software

- Entender que una app no “termina” cuando compila:

- Debe mantenerse, actualizarse y monitorearse.

- Desarrollar una visión de \*\*ciclo de vida largo\*\*:

- Versiones v1.0, v1.1, v2.0, etc.

- Registro de cambios (changelog).

- Manejo de bugs reportados.

### 3.3. Competencias de documentación y entrega

- Documentar el \*\*proceso de empaquetado\*\*:

- Pasos claros para generar el build.

- Requerimientos mínimos para compilar.

- Crear documentación \*\*orientada a usuario final\*\*:

- Mensajes simples de instalación.

- Recomendaciones de uso.

- Notas de seguridad si aplica (por ejemplo, uso de internet, almacenamiento, etc.).

### 3.4. Competencias de proyección profesional

- Entender el valor de decir:

- “No solo desarrollé una app, sino que llegué hasta la etapa de \*\*empaquetado y preparación para publicación\*\*.”

- Dejar el proyecto listo para:

- Mostrarlo en entrevistas.

- Continuarlo en cursos posteriores.

- Extenderlo como proyecto personal.

## 4. Organización general sugerida de la carpeta `Semana12`

Una estructura ordenada para la carpeta `Semana12` podría ser:

```text

Semana12/

├─ docs/ # Guías de empaquetado, publicación y mantenimiento

├─ build\_y\_despliegue/ # Archivos o notas específicas de builds generados

├─ proyecto\_final\_12/ # Versión final del proyecto lista para empaquetar

├─ ejercicios\_aplicados/ # Actividades puntuales sobre empaquetado/versión

└─ evidencias/ # Capturas, reportes y resultados de empaquetado

# # Semana 13 – Evaluación final, refuerzo, continuidad y cierre académico del curso

> Carpeta: `Semana13` del repositorio `desarrolloapps1`

> Materia: Desarrollo de Aplicación Móvil I

> Periodo: Agosto–Diciembre 2025

La carpeta \*\*Semana13\*\* representa la etapa de \*\*cierre académico ampliado\*\* del curso de \*\*Desarrollo de Aplicación Móvil I\*\*, donde se concentran:

- Actividades de \*\*evaluación final\*\* (teórica y práctica).

- Material de \*\*refuerzo\*\* para temas que resultaron más complejos.

- Espacios para \*\*extender\*\* el proyecto integrador más allá de lo mínimo requerido.

- Orientaciones para la \*\*continuidad\*\* del aprendizaje (tanto en la materia como en cursos posteriores).

- Organización final de \*\*evidencias\*\* y del repositorio como “cierre de ciclo”.

Mientras que:

- En \*\*Semana 9–10\*\* se pulió el \*\*proyecto integrador\*\*,

- En \*\*Semana 11\*\* se trabajó la \*\*retroalimentación, portafolio y mejora continua\*\*,

- En \*\*Semana 12\*\* se revisó el \*\*empaquetado, despliegue y mantenimiento\*\*,

la \*\*Semana 13\*\* está pensada como un espacio para:

> \*\*Evaluar lo aprendido, reforzar lo necesario, consolidar evidencias y dejar trazado el camino hacia el futuro académico/profesional del estudiante en desarrollo de aplicaciones móviles.\*\*

## 1. Contexto de la Semana 13 dentro del curso

Para ubicar el rol de `Semana13`, conviene recordar la ruta formativa global:

- \*\*Semana 1:\*\* Entorno de desarrollo, estructura del repositorio, primer programa/app.

- \*\*Semana 2:\*\* Fundamentos de lógica, estructuras de control, decisiones y ciclos.

- \*\*Semana 3:\*\* Interfaces gráficas iniciales y componentes básicos de UI.

- \*\*Semana 4:\*\* Navegación entre pantallas y manejo de eventos.

- \*\*Semana 5:\*\* Manejo de listas, colecciones y pantallas dinámicas.

- \*\*Semana 6:\*\* Formularios, validaciones y retroalimentación al usuario.

- \*\*Semana 7:\*\* Persistencia de datos local y ciclo de vida básico de la app.

- \*\*Semana 8:\*\* Consumo de servicios web (APIs), JSON y datos remotos.

- \*\*Semana 9:\*\* Proyecto integrador I – Prototipo funcional.

- \*\*Semana 10:\*\* Pruebas, corrección de errores y documentación del proyecto.

- \*\*Semana 11:\*\* Retroalimentación, pulido y construcción de portafolio.

- \*\*Semana 12:\*\* Empaquetado, despliegue conceptual y mantenimiento.

Sobre esa base, \*\*Semana 13\*\* se concibe como:

- Una \*\*capa final de cierre\*\*, que combina:

- \*\*Evaluación\*\* (exámenes, rúbricas, autoevaluación).

- \*\*Refuerzo\*\* (material de repaso, ejercicios extra).

- \*\*Proyección\*\* (qué sigue después del curso, rutas de aprendizaje).

- Un punto en el que el estudiante puede:

- Revisar en conjunto todo lo aprendido.

- Ordenar su repositorio como \*\*evidencia institucional\*\*.

- Ver su proyecto integrador como un \*\*logro importante\*\* y no solo como una tarea más.

## 2. Propósito general de la Semana 13

El propósito central de la carpeta `Semana13` es:

> \*\*Concentrar las actividades de evaluación final, refuerzo de contenidos, documentación de evidencias y orientación hacia la continuidad del aprendizaje, completando el ciclo formativo del curso Desarrollo de Aplicación Móvil I.\*\*

De este propósito se desprenden varios \*\*objetivos específicos\*\*:

1. Aplicar una \*\*evaluación diagnóstica final\*\*:

- Teórica (conceptos clave).

- Práctica (pequeños retos de código y diseño de UI).

2. Ofrecer \*\*material de refuerzo\*\*:

- Actividades adicionales para quienes necesiten consolidar temas específicos.

- Ejemplos más claros de puntos que generaron dudas durante el curso.

3. Permitir una \*\*extensión del proyecto integrador\*\*:

- Nuevas ideas o funcionalidades opcionales.

- Ajustes experimentales que no eran obligatorios para la calificación, pero sí enriquecen el aprendizaje.

4. Organizar y \*\*empaquetar evidencias\*\*:

- Por semana.

- Por tema.

- Por tipo de trabajo (código, capturas, reportes).

5. Proporcionar una \*\*guía de continuidad\*\*:

- Recomendaciones para cursos siguientes (por ejemplo, Desarrollo de Aplicación Móvil II).

- Recursos para autoaprendizaje (documentación oficial, cursos en línea, proyectos personales sugeridos).

## 3. Competencias que se fortalecen en Semana 13

### 3.1. Competencias técnicas

- Repaso y consolidación de:

- Estructuras de control.

- Diseño de interfaces.

- Navegación y eventos.

- Listas, formularios, validaciones.

- Persistencia y consumo de servicios web (si el curso lo cubrió plenamente).

- Capacidad de \*\*resolver problemas integrales\*\*:

- No solo ejercicios aislados, sino actividades que involucren varios conceptos combinados.

### 3.2. Competencias de autoevaluación y metacognición

- Reconocer:

- En qué temas se siente más fuerte el estudiante.

- En qué temas aún hay dudas.

- Aprender a construir un \*\*plan personal de mejora\*\*:

- Qué repasar.

- Qué proyectos practicar.

- Qué tecnologías explorar a futuro.

### 3.3. Competencias de documentación y evidencia

- Finalizar la \*\*organización del repositorio\*\*:

- Carpetas por semana completas.

- README o documentos de explicación por semana/proyecto.

- Preparar \*\*evidencias ordenadas\*\*:

- Útiles para el docente.

- Útiles para trámites escolares.

- Útiles para el propio estudiante como archivo personal.

### 3.4. Competencias de proyección académica y profesional

- Entender que el curso:

- Es un punto de partida, no el final del camino.

- Ser capaz de responder:

- ¿Qué puedo mostrar de este curso en mi CV/portafolio?

- ¿Qué me gustaría aprender después (frameworks, lenguajes, plataformas)?

## 4. Organización general sugerida de la carpeta `Semana13`

Una estructura ordenada para `Semana13` podría ser:

```text

Semana13/

├─ docs/ # Instrucciones, rúbricas, guías de estudio, orientación futura

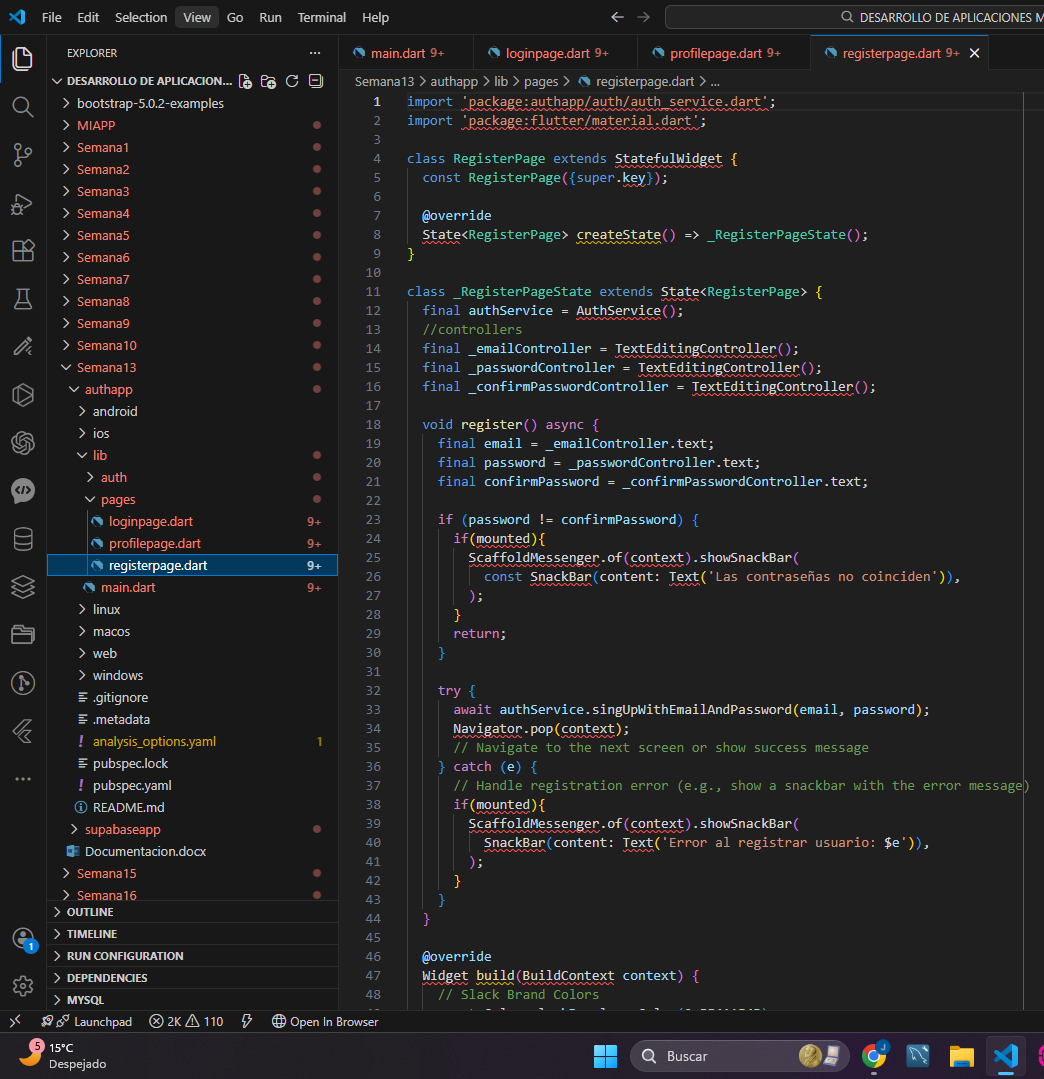
├─ evaluacion\_final/ # Exámenes, cuestionarios, retos de evaluación

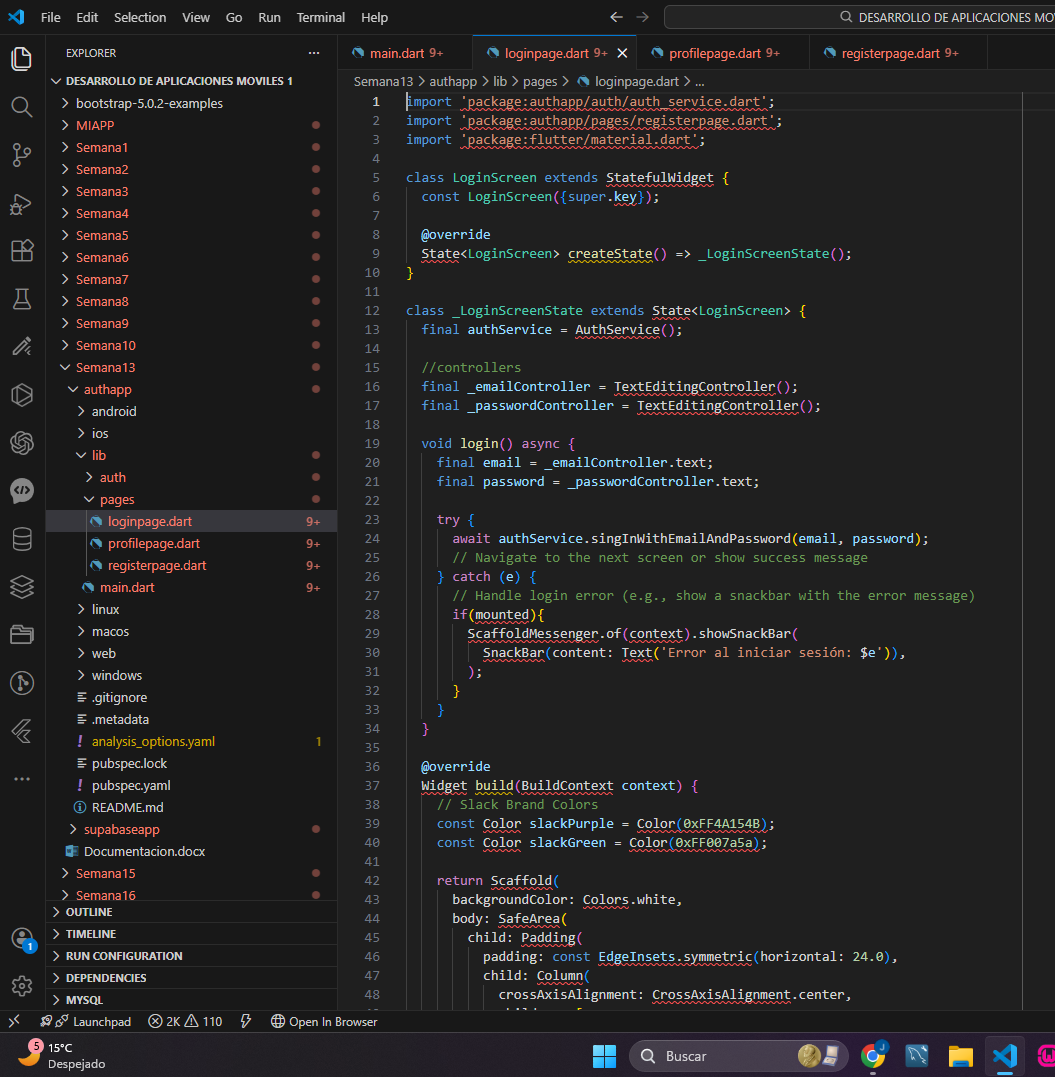
├─ refuerzo\_y\_repaso/ # Material de repaso y ejercicios adicionales

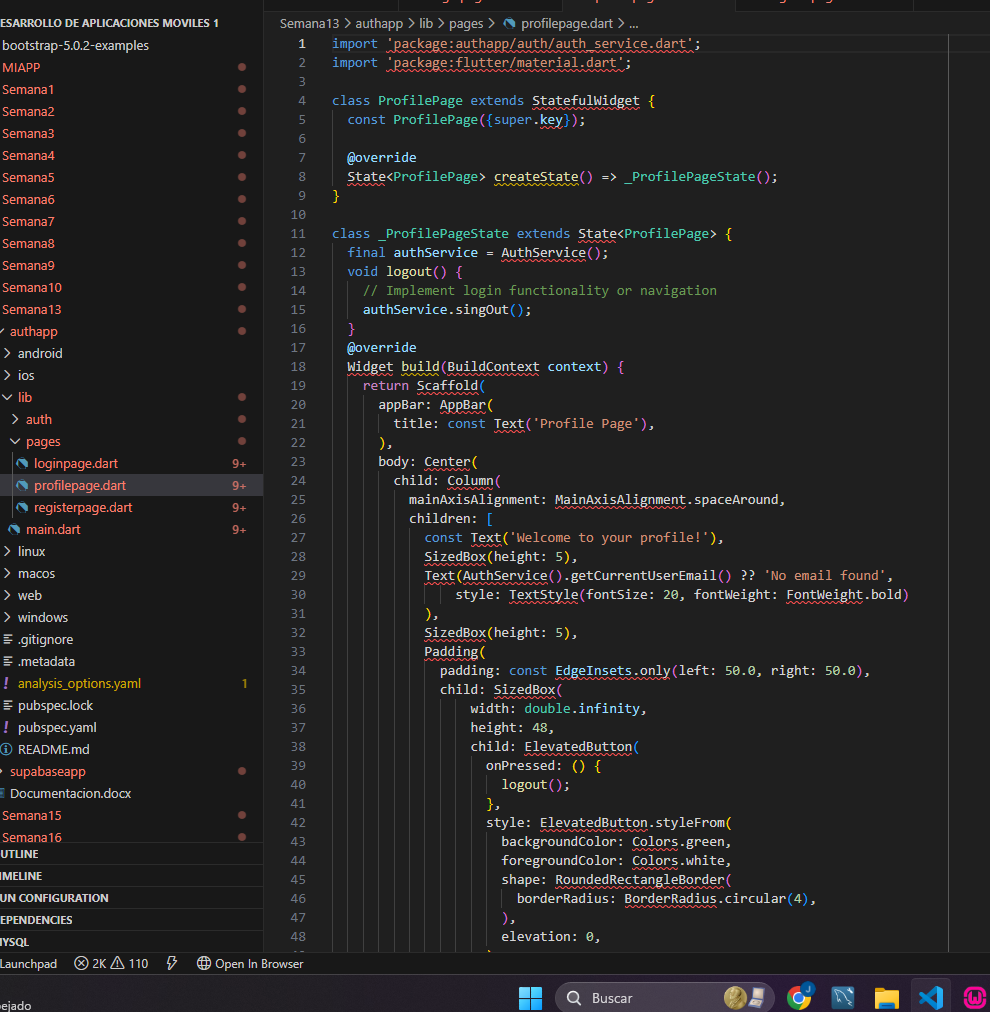
├─ proyecto\_extendido/ # Extensión opcional del proyecto integrador

├─ portafolio\_y\_resumen/ # Resumen de evidencias, síntesis del curso

└─ evidencias/ # Capturas finales, reportes y entregables de Semana 13







# # Semana 14 – Proyección, temas avanzados, investigación y cierre profesional del curso

> Carpeta: `Semana14` del repositorio `desarrolloapps1`

> Materia: Desarrollo de Aplicación Móvil I

> Periodo: Agosto–Diciembre 2025

La carpeta \*\*Semana14\*\* representa la \*\*fase de cierre ampliado y de proyección profesional\*\* del curso de \*\*Desarrollo de Aplicación Móvil I\*\*.

A diferencia de las primeras semanas, cuyo foco estaba en aprender conceptos nuevos y construir el proyecto integrador, en Semana14 se busca:

- Mirar \*\*hacia adelante\*\*: ¿qué sigue después de este curso?

- Explorar \*\*temas avanzados\*\* e ideas de mejora que van más allá del programa mínimo.

- Plantear \*\*líneas de investigación\*\* o profundización en desarrollo móvil.

- Dejar el repositorio y el proyecto integrador en un estado \*\*presentable y reutilizable\*\* (portafolio, entrevistas, futuras materias).

- Conectar lo aprendido con el \*\*mundo laboral\*\* y con la formación continua.

En otras palabras:

> \*\*Semana14 no es solo la “última semana de clase”, sino el puente entre lo que ya aprendiste y lo que podrías lograr si sigues construyendo sobre este curso.\*\*

## 1. Contexto de la Semana 14 dentro del curso

Para entender el papel de `Semana14`, conviene ver el “mapa completo” del recorrido del curso:

- \*\*Semana 1:\*\* Entorno de desarrollo, estructura del repositorio, primer programa / app.

- \*\*Semana 2:\*\* Fundamentos de lógica, estructuras de control, decisiones y ciclos.

- \*\*Semana 3:\*\* Interfaces de usuario básicas y componentes gráficos iniciales.

- \*\*Semana 4:\*\* Navegación entre pantallas y manejo de eventos.

- \*\*Semana 5:\*\* Listas, colecciones de datos y pantallas dinámicas.

- \*\*Semana 6:\*\* Formularios, validaciones y retroalimentación al usuario.

- \*\*Semana 7:\*\* Persistencia de datos local y ciclo de vida básico.

- \*\*Semana 8:\*\* Consumo de servicios web (APIs), JSON y datos remotos.

- \*\*Semana 9:\*\* Proyecto integrador I – Prototipo funcional.

- \*\*Semana 10:\*\* Pruebas, corrección de errores y documentación funcional.

- \*\*Semana 11:\*\* Retroalimentación, pulido del proyecto y construcción de portafolio.

- \*\*Semana 12:\*\* Empaquetado, despliegue conceptual y mantenimiento.

- \*\*Semana 13:\*\* Evaluación final, refuerzo y cierre académico del curso.

Con todo eso como base, \*\*Semana14\*\* se diseñaría para:

- Profundizar en \*\*temas que el programa oficial solo alcanza a mencionar\*\*.

- Dar \*\*orientaciones concretas\*\* sobre qué aprender después (frameworks, herramientas, buenas prácticas avanzadas).

- Cerrar el repositorio de forma que, meses o años después, el estudiante pueda \*\*volver a este proyecto y seguir construyendo\*\*.

## 2. Propósito general de la Semana 14

El propósito central de la carpeta `Semana14` es:

> \*\*Brindar al estudiante un marco de proyección y continuidad, presentando temas avanzados, líneas de investigación, mejoras posibles a la app y recomendaciones para convertir el proyecto integrador y el repositorio en un verdadero punto de partida profesional.\*\*

De este propósito se desprenden varios \*\*objetivos específicos\*\*:

1. Presentar \*\*temas avanzados de desarrollo móvil\*\* que no se pudieron abordar a profundidad en el curso, pero que son relevantes (arquitecturas, pruebas automatizadas, publicación profesional, etc.).

2. Sugerir \*\*mejoras y extensiones\*\* concretas para el proyecto integrador:

- Nuevos módulos.

- Uso de patrones de diseño.

- Integración con servicios adicionales.

3. Orientar al estudiante sobre \*\*cómo usar este curso como portafolio\*\*:

- Qué mostrar en entrevistas.

- Cómo explicar lo que hizo.

- Cómo seguir mejorando su app por cuenta propia.

4. Dejar una \*\*guía de estudio y recursos adicionales\*\*:

- Documentación oficial.

- Cursos, libros, tutoriales recomendados.

- Temas de investigación para proyectos futuros (residencias, tesis, etc.).

5. Organizar en la carpeta `Semana14` materiales que sirvan como:

- Cierre conceptual.

- Mapa de ruta para el futuro.

## 3. Competencias que se refuerzan en Semana 14

Aunque a estas alturas el foco ya no es tanto el “código nuevo”, Semana14 fortalece competencias muy importantes:

### 3.1. Competencias técnicas (visión avanzada)

- Comprender a nivel conceptual:

- Arquitecturas comunes en apps móviles modernas (por ejemplo, MVC, MVVM, Clean Architecture).

- Uso de servicios en segundo plano (notificaciones, sincronización).

- Patrones de diseño aplicados a apps (Singleton, Repository, Observer, etc., según el nivel del grupo).

- Identificar en qué puntos del proyecto integrador:

- Podría aplicarse una arquitectura más formal.

- Podrían separarse más claramente las capas (datos, presentación, lógica de negocio).

### 3.2. Competencias de ingeniería de software

- Valorar la importancia de:

- Control de versiones (Git) como herramienta profesional.

- Código limpio, mantenible y legible.

- Documentación constante a lo largo del ciclo de vida del proyecto.

- Entender la diferencia entre:

- Un prototipo de clase.

- Una app lista para escalar y mantenerse en producción.

### 3.3. Competencias de proyección profesional

- Preparar al estudiante para:

- Hablar de su proyecto en entrevistas.

- Mostrar capturas, código y documentación de forma profesional.

- Reconocer cómo “vender” su trabajo:

- Qué destacar (logros técnicos, aprendizaje, solución propuesta).

- Qué aceptar como limitación actual (lo que falta, pero se puede mejorar).

### 3.4. Competencias de autoaprendizaje y continuidad

- Saber \*\*por dónde seguir\*\*:

- Temas complementarios: bases de datos avanzadas, APIs más complejas, autenticación, seguridad, etc.

- Temas de siguiente nivel: apps multiplataforma, integración con hardware, análisis de rendimiento.

- Desarrollar una \*\*mentalidad de aprendizaje permanente\*\*:

- Reconocer que las tecnologías móviles cambian rápido.

- Aprender a leer documentación oficial y mantenerse actualizado.

## 4. Organización general sugerida de la carpeta `Semana14`

Una estructura ordenada y útil para `Semana14` podría ser:

```text

Semana14/

├─ docs/ # Documentos de temas avanzados, proyección y guías

├─ ideas\_mejoras\_proyecto/ # Propuestas de extensión y rediseño del proyecto integrador

├─ investigacion\_y\_tendencias/ # Notas y resúmenes sobre tendencias actuales en desarrollo móvil

├─ recursos\_adicionales/ # Links, referencias, materiales extra recomendados

└─ evidencias\_finales/ # Documentos de cierre, síntesis y evidencias adicionales

# # Semana 15 – Síntesis global, acreditación y cierre formal del repositorio del curso

> Carpeta: `Semana15` del repositorio `desarrolloapps1`

> Materia: \*\*Desarrollo de Aplicación Móvil I\*\*

> Periodo: \*\*Agosto–Diciembre 2025\*\*

La carpeta \*\*Semana15\*\* representa el \*\*cierre formal y administrativo\*\* del curso de \*\*Desarrollo de Aplicación Móvil I\*\*, así como la \*\*síntesis global\*\* de todo lo trabajado a lo largo del semestre.

En las semanas anteriores se han ido construyendo:

- Conocimientos teóricos de programación y desarrollo móvil.

- Habilidades prácticas para crear interfaces, manejar datos, consumir servicios, etc.

- Un \*\*proyecto integrador\*\* que evoluciona desde un prototipo simple hasta una app funcional y presentable.

- Evidencias y documentación que dan cuenta del proceso.

En \*\*Semana15\*\* el enfoque gira en torno a:

> \*\*Acreditar formalmente el curso, cerrar el repositorio de manera ordenada, sintetizar los aprendizajes clave y dejar listos los materiales para futuras consultas académicas o profesionales.\*\*

Es la semana donde se deja “amarrado” todo:

- Calificaciones finales.

- Entregas definitivas.

- Organización del repositorio.

- Documentos de cierre y reflexión final.

## 1. Rol de la Semana 15 dentro del curso

Para comprender la importancia de `Semana15`, conviene recordar el recorrido general:

- \*\*Semanas 1–4:\*\*

Bases de programación, primeras interfaces, navegación y eventos.

- \*\*Semanas 5–8:\*\*

Manejo de listas, formularios, validaciones, persistencia de datos y consumo de APIs.

- \*\*Semanas 9–10:\*\*

Construcción y cierre técnico del \*\*proyecto integrador\*\*.

- \*\*Semanas 11–12:\*\*

Portafolio, empaquetado, despliegue conceptual y mantenimiento.

- \*\*Semana 13:\*\*

Evaluación final, refuerzo de contenidos y organización de evidencias.

- \*\*Semana 14:\*\*

Proyección profesional, temas avanzados y líneas de investigación.

Sobre esta base, \*\*Semana15\*\* se concibe como:

- Una \*\*capa final de cierre institucional y personal\*\*.

- El espacio para:

- Registrar resultados definitivos (académicos).

- Cerrar el ciclo de trabajo del repositorio.

- Elaborar una \*\*síntesis final\*\* de aprendizajes, logros y retos.

- Formalizar el curso como una experiencia completa, no solo una acumulación de tareas.

## 2. Propósito general de la Semana 15

El propósito central de la carpeta `Semana15` es:

> \*\*Concentrar la síntesis global del curso, documentar la acreditación y cierre, y dejar el repositorio preparado como evidencia formal de lo aprendido en Desarrollo de Aplicación Móvil I.\*\*

De este propósito se desprenden varios \*\*objetivos específicos\*\*:

1. Documentar de forma clara:

- Qué se logró terminar en el proyecto integrador.

- Qué contenidos del programa se alcanzaron.

- Qué temas quedan como puertas abiertas para cursos posteriores.

2. Organizar y asegurar la \*\*completitud del repositorio\*\*:

- Verificar que todas las carpetas de semanas anteriores contengan:

- Códigos.

- Documentos.

- Evidencias.

- Corregir nombres, agregar README faltantes, etc.

3. Integrar una \*\*memoria final del curso\*\*:

- Resumen de actividades clave.

- Principales dificultades técnicas y cómo se resolvieron.

- Aprendizajes personales y profesionales.

4. Dejar constancia de la \*\*acreditación\*\*:

- Documentos (si se usan) de rúbricas, criterios y resultados.

- Constancias o resúmenes que indiquen el cumplimiento de objetivos de la materia.

5. Preparar el repositorio para su uso futuro:

- Como \*\*portafolio\*\*.

- Como referencia para \*\*Desarrollo de Aplicación Móvil II\*\* u otras materias.

- Como base para proyectos personales o de titulación.

## 3. Competencias que se refuerzan en Semana 15

Aunque la parte “técnica” del código se concentra en semanas anteriores, Semana15 refuerza competencias clave de \*\*ingeniería de software y vida profesional\*\*.

### 3.1. Competencias de síntesis y documentación

- Capacidad de \*\*resumir un curso completo\*\* en un conjunto de documentos claros.

- Elaboración de:

- Síntesis de contenidos.

- Descripción de logros y retos.

- Sección de “aprendizajes clave”.

- Organización de un \*\*repositorio académico\*\*:

- Estructura limpia.

- README por semana o módulo.

- Documentos de explicación general.

### 3.2. Competencias de cierre de proyecto

- Entender que un proyecto no solo se “acaba” cuando deja de tener errores, sino cuando:

- Está \*\*documentado\*\*.

- Es \*\*comprensible\*\* para alguien más.

- Tiene \*\*evidencias claras\*\* de que se probó y se usó.

- Formalizar el fin de una fase:

- Cierre de “versión 1.0” del proyecto integrador.

- Preparación para posibles versiones futuras en otros cursos o contextos.

### 3.3. Competencias de evaluación y autoevaluación

- Interpretar rúbricas y criterios de evaluación.

- Analizar:

- ¿En qué áreas se obtuvo mejor desempeño?

- ¿En qué áreas hubo más retos?

- Desarrollar \*\*honestidad académica\*\*:

- Reconocer errores.

- Valorar logros sin exagerarlos ni minimizarlos.

### 3.4. Competencias de proyección personal

- Visualizar cómo este curso:

- Se traduce en habilidades para el mercado laboral.

- Se suma a un currículum o portafolio.

- Planear qué sigue:

- ¿Quiero seguir en desarrollo móvil?

- ¿Qué frameworks o lenguajes me interesa aprender a partir de aquí?

- ¿Podría usar esta app como punto de partida para un proyecto real?

## 4. Organización general sugerida de la carpeta `Semana15`

Una estructura ordenada y útil para `Semana15` podría ser:

```text

Semana15/

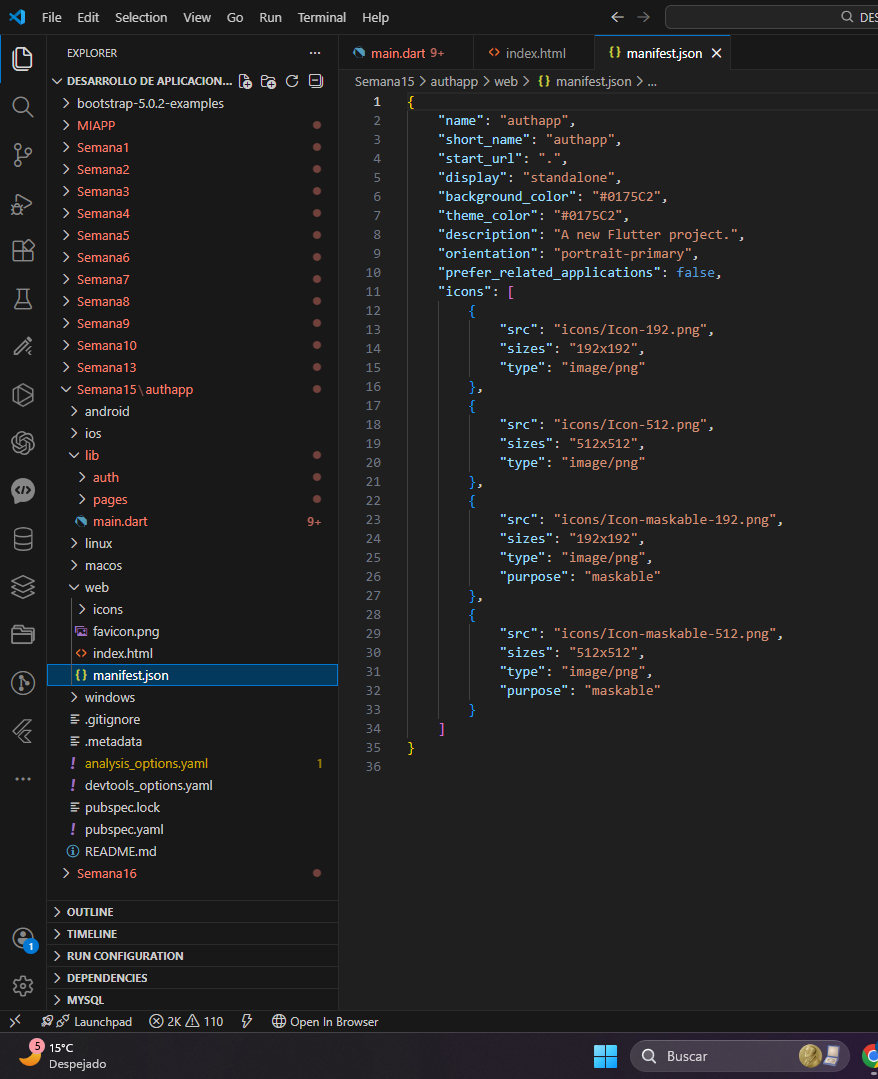
├─ docs\_cierre/ # Documentos de síntesis, cierre y acreditación

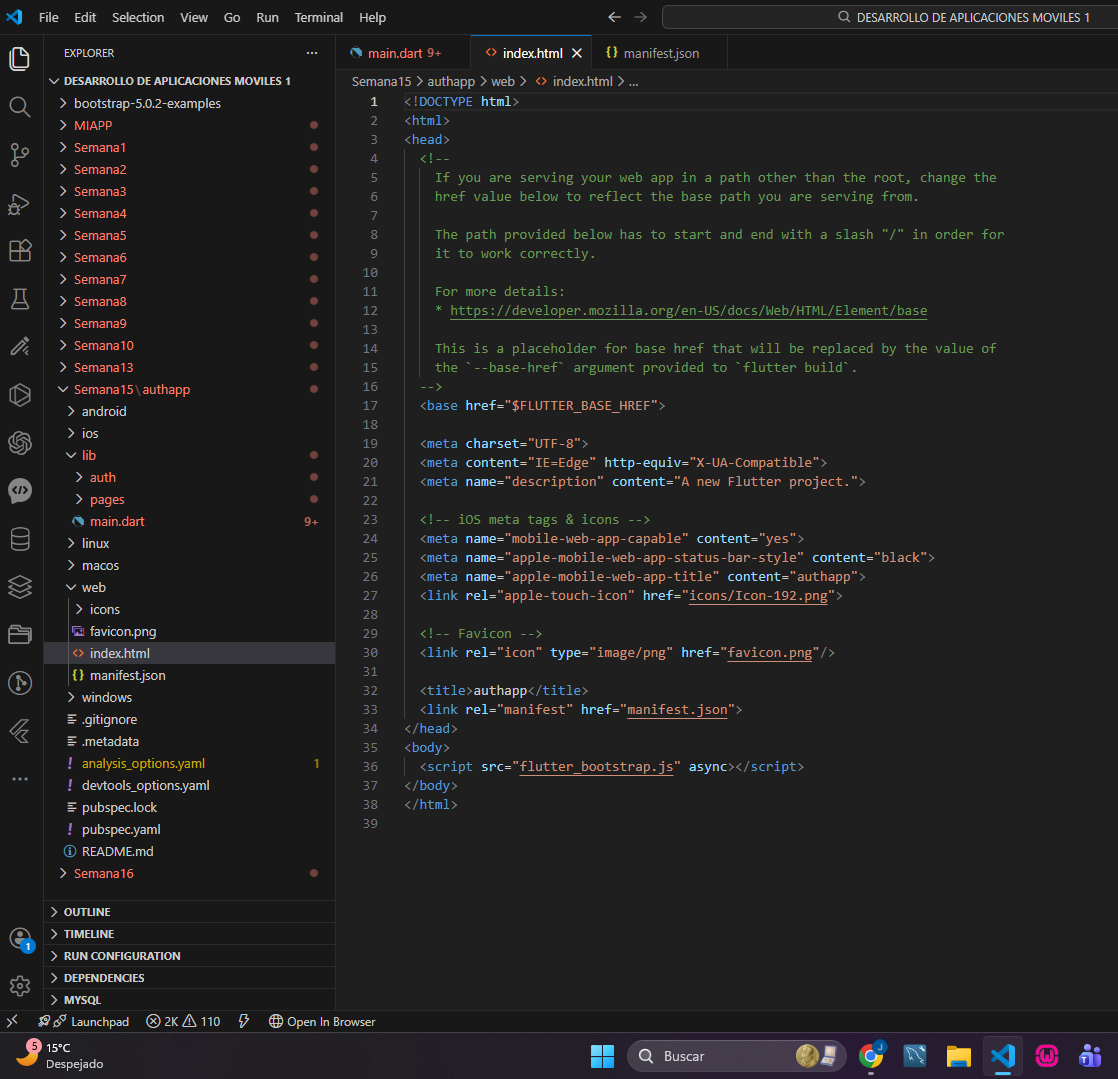
├─ resumen\_curso/ # Síntesis global de lo aprendido en la materia

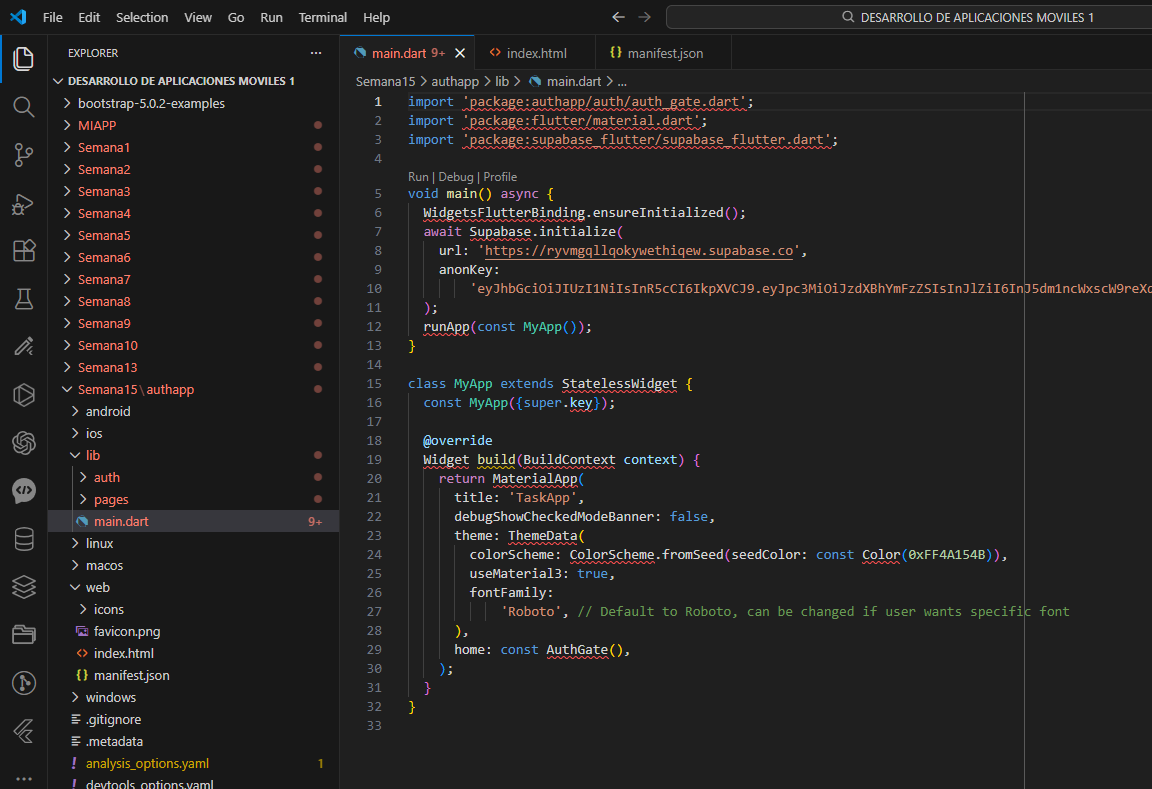
├─ consolidacion\_repositorio/ # Notas de revisión y checklist del repositorio completo

├─ proyecto\_integrador\_final/ # (Opcional) enlace o referencia a la versión definitiva del proyecto

└─ evidencias\_cierre/ # Evidencias finales y documentos de respaldo







# # Semana 16 – Recuperación, regularización, consolidación total y cierre administrativo del curso

> Carpeta: `Semana16` del repositorio `desarrolloapps1`

> Materia: \*\*Desarrollo de Aplicación Móvil I\*\*

> Periodo: \*\*Agosto–Diciembre 2025\*\*

La carpeta \*\*Semana16\*\* puede considerarse la \*\*última capa de cierre académico, técnico y administrativo\*\* del curso de \*\*Desarrollo de Aplicación Móvil I\*\*.

En muchas planeaciones, esta semana se reserva para:

- \*\*Recuperación y regularización\*\* de estudiantes que aún tienen entregas pendientes.

- \*\*Correcciones finales\*\* a actividades clave (sobre todo al proyecto integrador).

- \*\*Ajustes administrativos\*\* (calificaciones finales, revisiones, aclaraciones).

- \*\*Consolidación total del repositorio\*\*, asegurando que todo el material del semestre quede completo, ordenado y utilizable a futuro.

Mientras que:

- \*\*Semana 15\*\* se centra en la \*\*síntesis global y cierre académico formal\*\*,

- \*\*Semana 16\*\* funciona como un \*\*“colchón” de cierre operativo\*\*, donde se terminan de acomodar detalles, se registran correcciones, se regularizan entregas atrasadas y se deja todo listo para archivar el curso.

## 1. Contexto de la Semana 16 dentro del curso

Para ubicar `Semana16` en el mapa global del curso, conviene recordar el camino completo:

- \*\*Semanas 1–4\*\*

Bases: entorno de desarrollo, lógica, primeras interfaces, navegación y eventos.

- \*\*Semanas 5–8\*\*

Nivel intermedio: listas, formularios, validaciones, persistencia local, consumo de APIs.

- \*\*Semanas 9–10\*\*

Proyecto integrador: prototipo funcional y posterior estabilización, pruebas y documentación.

- \*\*Semana 11\*\*

Retroalimentación y pulido del proyecto, portafolio y mejoras.

- \*\*Semana 12\*\*

Empaquetado conceptual, generación de builds y mantenimiento.

- \*\*Semana 13\*\*

Evaluación final, refuerzo y organización de evidencias.

- \*\*Semana 14\*\*

Proyección profesional, temas avanzados y líneas de investigación futuras.

- \*\*Semana 15\*\*

Síntesis global, acreditación y cierre formal del repositorio.

A partir de eso, \*\*Semana16\*\* se entiende como:

- El \*\*espacio final de maniobra\*\*:

- Para completar evidencias faltantes.

- Para registrar correcciones sobre avances ya hechos.

- Para cerrar cualquier aspecto administrativo pendiente.

- El punto donde se valida que el repositorio `desarrolloapps1`:

- Queda \*\*completo\*\*, legible y bien estructurado.

- Puede ser consultado en el futuro como \*\*evidencia académica y profesional\*\*.

## 2. Propósito general de la Semana 16

El propósito central de la carpeta `Semana16` es:

> \*\*Servir como módulo de recuperación, regularización y consolidación final del curso, dejando tanto el repositorio como el proyecto integrador en un estado completo, corregido y listo para cierre administrativo y archivo.\*\*

De este propósito general derivan varios \*\*objetivos específicos\*\*:

1. Brindar un espacio de \*\*recuperación y regularización\*\*:

- Permitir la entrega o corrección de actividades clave.

- Ajustar aquellos detalles mínimos que impiden considerar completo el proyecto integrador.

2. Consolidar y \*\*verificar la integridad del repositorio\*\*:

- Confirmar que todas las carpetas de Semana1 a Semana16 contienen el material esperado.

- Comprobar que todos los README o documentos de descripción se encuentran presentes y legibles.

3. Registrar \*\*correcciones finales\*\*:

- Sobre el proyecto integrador.

- Sobre documentación incompleta o incorrecta.

- Sobre evidencias faltantes o mal organizadas.

4. Generar \*\*documentos de cierre administrativo\*\*:

- Notas de regularización.

- Listados finales de actividades entregadas.

- Registro de alumnos que completaron exitosamente el curso.

5. Dejar una \*\*constancia final de conclusiones y recomendaciones\*\*:

- Para el propio estudiante (qué debería reforzar a futuro).

- Para futuras ediciones del curso (sugerencias de mejora).

## 3. Competencias que se refuerzan en Semana 16

Aunque esta semana no siempre introduce código nuevo, sí refuerza competencias muy relevantes en el ámbito académico y profesional.

### 3.1. Competencias de organización y cierre de proyectos

- Comprender que un proyecto no solo es:

- Escribir código,

- Sino también \*\*dejarlo cerrado correctamente\*\*.

- Desarrollar habilidades para:

- Revisar que no falten componentes.

- Asegurar que la estructura de archivos tiene sentido.

- Documentar adecuadamente el estado final de un proyecto.

### 3.2. Competencias de responsabilidad académica

- Cumplir con \*\*entregas finales\*\* y procesos de regularización.

- Asumir responsabilidad por:

- Actividades que estuvieron incompletas.

- Mejoras que se debían aplicar.

- Demostrar compromiso para \*\*terminar lo que se comenzó\*\*.

### 3.3. Competencias de revisión crítica

- Revisar el propio trabajo con mirada crítica:

- Qué partes del proyecto son más fuertes.

- Qué partes quedaron “apenas suficientes”.

- Qué cosas se mejorarían si hubiera más tiempo.

- Documentar estas reflexiones en forma de:

- Notas.

- Listas de mejoras pendientes.

- Comentarios sobre la experiencia del curso.

### 3.4. Competencias administrativas básicas

- Entender el rol de:

- Entregas finales.

- Evidencias.

- Calificaciones y rúbricas.

- Aprender a \*\*organizar documentación académica\*\*, útil si se requiere:

- Para trámites escolares.

- Para demostrar competencias en otras instancias (prácticas, residencias, etc.).

## 4. Organización general sugerida de la carpeta `Semana16`

Una estructura ordenada para `Semana16` podría ser la siguiente:

```text

Semana16/

├─ docs\_recuperacion/ # Instrucciones y documentos de regularización/extraordinarios

├─ ajustes\_proyecto/ # Correcciones finales del proyecto integrador

├─ consolidacion\_repositorio/ # Checklists y reportes de revisión global del repo

├─ reportes\_finales/ # Documentos de cierre, registros y conclusiones

└─ evidencias\_adicionales/ # Evidencias complementarias o corregidas

