Desarrollo de Interfaces

Proyecto Entregable 03 - Primera aplicación Completa en Flutter

Autor: Sergi García

Actualizado Septiembre 2025

Licencia

Nomenclatura

A lo largo de este tema se utilizarán diferentes símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

📖 **Importante**

❕ **Atención**

💬 **Interesante**

ÍNDICE

Proyecto Entregable 03 - Primera aplicación completa en Flutter

1. Objetivo del Proyecto

Desarrollar una aplicación funcional y completa en Flutter que sirva como consolidación de los conceptos fundamentales del framework. El alumnado demostrará su capacidad para estructurar un proyecto, implementar una interfaz de usuario coherente, gestionar el estado de la aplicación y crear una experiencia de usuario dinámica e interactiva.

2. Descripción General

Cada estudiante diseñará y programará de forma individual una aplicación móvil y/o web con Flutter que cumpla con una serie de requisitos técnicos y de diseño preestablecidos. El proyecto culminará con una defensa oral en la que se presentará la aplicación y se demostrará el dominio de los conceptos utilizados.

3. Requisitos técnicos obligatorios

La aplicación debe incluir de manera imprescindible los siguientes elementos:

- **Estructura Multipantalla:**
  - Mínimo 3 pantallas con navegación fluida entre ellas (por ejemplo, usando Navigator).
  - Pantalla de Bienvenida: Debe incluir un título, un logotipo/icono representativo y una breve descripción del propósito de la aplicación.
- **Interactividad y Gestión de Estado:**
  - Uso de al menos 3 tipos diferentes de widgets de entrada de datos (ej: TextField, Checkbox/Switch, DropdownButton, Slider, selección de fecha/hora).
  - Uso de al menos 2 tipos diferentes de widgets de salida de datos (ej: ListView/ListView.builder, Card, AlertDialog, SnackBar, DataTable).
  - La aplicación debe permitir al usuario realizar acciones dinámicas que modifiquen el estado de la app (ej: añadir, editar, eliminar o marcar elementos como completados).
- **Diseño y Experiencia de Usuario (UX/UI):**
  - Diseño coherente y cuidado, utilizando una paleta de colores y tipografía uniforme en toda la aplicación.
  - La interfaz debe ser intuitiva, clara y estar organizada de forma lógica.
- **Documentación del Código:**
  - El código fuente debe estar profusamente comentado con explicaciones claras en:
    - La cabecera de las clases y widgets principales (propósito y funcionalidad).
    - Las variables de estado clave.
    - Los métodos más complejos, describiendo la lógica de implementación.

4. Tema de la aplicación

El tema es libre, pero la aplicación debe tener un propósito y una coherencia temática claros. Algunas ideas orientativas:

- Mini-agenda de contactos.
- Aplicación de recetas o lista de la compra.
- Diario personal o blog simple.
- Un quiz o juego de preguntas sencillo.
- Trackeador de hábitos o gastos personales.

5. Condiciones de desarrollo

- Permitido:
  - Consultar la documentación oficial de Flutter y Dart.
  - Utilizar ejemplos de código de repositorios oficiales (ej: samples de pub.dev), tutoriales o ejemplos de clase como inspiración, nunca como copia literal.
  - Usar asistentes de IA (Copilot, ChatGPT, etc.) solo como ayuda para entender conceptos, como apoyo en situaciones particulares o para depurar errores específicos de sintaxis. Queda terminantemente prohibido generar bloques de código completos o la estructura de la aplicación con estas herramientas.
- Prohibido:
  - Copiar código completo de compañeros, de internet o generado por IA.
  - Recibir ayuda directa en la escritura del código de otra persona (compañero, externo, etc.). El trabajo debe ser auténtico e individual.

6. Entregables

1. Código Fuente: El proyecto completo de Flutter, subido a un repositorio Git privado y compartido con el profesorado.
2. Defensa Oral - Presentación Pechakucha:
   - El alumnado realizará una presentación ante el profesorado en formato Pechakucha (20 diapositivas x 20 segundos cada una, 6:40 min total).
   - La presentación debe incluir:
     - Introducción y objetivo de la app.
     - Demostración en vivo de la aplicación y sus funcionalidades.
     - Explicación de las decisiones de diseño y técnicas más relevantes.
   - Tras la presentación, el profesorado podrá:
     - Solicitar modificaciones o mejoras específicas en el código en tiempo real.
     - Plantear preguntas sobre cualquier aspecto de la implementación para evaluar la comprensión profunda del proyecto.

7. Criterios de evaluación

La calificación final se basará en los siguientes aspectos:

Criterio	Peso	Descripción
Funcionamiento Técnico	30%	La aplicación funciona sin crashes. Todas las pantallas y navegaciones operan correctamente.

Implementación de Widgets	25%	Uso correcto, variado y efectivo de los widgets de entrada y salida requeridos.
Diseño y UX	20%	Interfaz coherente, atractiva, intuitiva y con una identidad visual definida.
Dinamismo y gestión de Estado	15%	La aplicación responde correctamente a las interacciones del usuario, con una gestión de estado robusta.
Documentación y Defensa	10%	Calidad de los comentarios en el código, claridad de la memoria técnica y capacidad de explicación y adaptación durante la defensa oral.

Consejo: Enfoca tu esfuerzo en entender la lógica detrás de cada elemento que implementes. Durante la defensa, se valorará mucho más que demuestres por qué has hecho las cosas de una determinada manera y que seas capaz de modificarlo, a que simplemente hayas copiado un código funcional sin entenderlo.