

Diseño aplicado a interfaces web con Flutter

Unidad 05. Caso práctico 01



Autor: Sergi García

Actualizado Noviembre 2025

Licencia



Reconocimiento - No comercial - CompartirlGual (BY-NC-SA): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se ha de hacer con una licencia igual a la que regula la obra original.

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

De una interfaz básica a una versión profesional - Curriculum vitae	3
Material proporcionado	4
Objetivo	4
¿Qué debes hacer?	4
¿Y si tengo problemas técnicos?	4
¿Qué entregar?	4
Opción 1 — Entrega visual	5
Opción 2 — Entrega explicativa	5
Propuesta mejoras para mejorar un Curriculum Vitae hecho en Flutter	5
◆ 1. Añadir una foto de perfil — Humanizar y centrar la atención	5
◆ 2. Colores suaves y fondo distinto del blanco — Profesionalismo moderno	5
◆ 3. Separar secciones en tarjetas o bloques — Estructura visual clara	6
◆ 4. Iconografía funcional — Reforzar la comprensión inmediata	7
◆ 5. Jerarquía tipográfica — Diseñar para el escaneo, no la lectura completa	7
◆ 6. Títulos de sección con iconos — Crear anclas visuales	8
◆ 7. Espaciado y ritmo visual — Airear el contenido	8
◆ 8. Contenido técnico preciso — Profesionalismo en la redacción	9
◆ 9. Diseño responsive — Accesible y adaptable	9
◆ 10. Composición centrada vs. alineada a la izquierda — Estructura y equilibrio	10

UNIDAD 05. CASO PRÁCTICO 01

DE UNA INTERFAZ BÁSICA A UNA VERSIÓN PROFESIONAL - CURRICULUM VITAE

En esta actividad trabajaremos en la mejora del diseño visual y estructural de una aplicación de currículum desarrollada con Flutter.

A continuación se muestran **dos imágenes** correspondientes a diferentes versiones de la misma pantalla:

- La **primera imagen** representa una versión funcional, pero con un diseño básico, plano y con poca jerarquía visual.
- La **segunda imagen** muestra una versión mucho más cuidada, moderna y profesional, con atención al detalle y una experiencia visual mejorada.

Imagen 1 — Versión inicial (básica y funcional):

Curriculum de Juan Pérez

Información Personal

Nombre: Juan Pérez García
Email: juan.perez@example.com
Teléfono: +34 600 123 456
Ciudad: Valencia, España

Experiencia Laboral

Desarrollador Flutter – Empresa XYZ (2022 - Actualidad)
• Desarrollo de aplicaciones móviles multiplataforma.
• Implementación de interfaces responsivas y lógicas de negocio.
• Colaboración con equipos de diseño y backend.

Técnico Informático – Empresa ABC (2020 - 2022)
• Mantenimiento de equipos y soporte técnico.
• Instalación y configuración de software y redes locales.

Educación

Grado Superior en Desarrollo de Aplicaciones Multiplataforma
IES Serra Perenixia – 2018 - 2020

Habilidades

Flutter Dart Firebase Git SQL Python

Idiomas

• Español – Nativo
• Inglés – Intermedio (B1)

Contacto

juan.perez@example.com
 +34 600 123 456

Imagen 2 — Versión mejorada (más profesional y refinada):

Curriculum de Juan Pérez

Juan Pérez García
Desarrollador Flutter

juan.perez@example.com +34 600 123 456

Experiencia Laboral

Desarrollador Flutter
Empresa XYZ – 2022 - Actualidad
• Desarrollo de aplicaciones móviles multiplataforma.
• Implementación de interfaces responsivas.
• Integración con APIs REST y Firebase.

Técnico Informático
Empresa ABC – 2020 - 2022
• Soporte técnico y mantenimiento de equipos.
• Configuración de redes y sistemas Windows/Linux.

Educación

Grado Superior en Desarrollo de Aplicaciones Multiplataforma
IES Serra Perenixia – 2018 - 2020

 MATERIAL PROPORCIONADO

Para ayudarte en este ejercicio, te entregamos:

- El **código fuente completo y comentado** de la versión inicial.
- El **código fuente completo y comentado** de la versión final mejorada.

Estos archivos te permitirán explorar cómo se organiza el código, cómo se implementan elementos de diseño visual, y qué técnicas se aplican para estructurar mejor la interfaz.

 OBJETIVO

Analizar las diferencias entre ambas versiones, revisar el código proporcionado y los consejos de diseño del documento adjunto, y aplicar **al menos dos mejoras** en tu propia versión de la aplicación.

Este módulo está especialmente orientado al **diseño visual y la experiencia de usuario** más que a la complejidad técnica. Por tanto:

- No es necesario replicar toda la versión final.
- Lo importante es que **identifiques dos mejoras significativas** y las apliques correctamente o expliques cómo lo harías.

 ¿QUÉ DEBES HACER?

1. Estudia ambas imágenes y revisa los códigos proporcionados.
2. Lee los **consejos de diseño explicados en el documento** (iconografía, jerarquía, espaciado, tarjetas, fondo, etc.).
3. Elige **dos mejoras** que puedas aplicar a tu versión del CV.
4. Usa el código base de la versión inicial como punto de partida.
5. Aplica esas mejoras en tu código.
6. Manda capturas de pantalla con las mejoras, o si tienes problemas, documento explicando que has intentado (se puede obtener el apto, ya que nos centramos más en diseño).

 ¿Y SI TENGO PROBLEMAS TÉCNICOS?

Este ejercicio está centrado en el **pensamiento visual y estructural**, no en el dominio total de Flutter. Si alguna de las mejoras propuestas no logras implementarla, pero explicas:

- qué intentaste,
- cómo pensaste resolverlo,
- o qué parte del código exploraste,

se considerará **válidamente trabajado**.

 ¿QUÉ ENTREGAR?

Puedes elegir entre las siguientes formas de entrega, según tu nivel de avance:

Opción 1 — Entrega visual

- Capturas de pantalla mostrando las dos mejoras que aplicaste en tu versión.
- No es necesario entregar el código fuente.
- Puedes incluir una breve descripción debajo de cada captura explicando qué mejora aplicaste.

Opción 2 — Entrega explicativa

- Si no lograste implementar las mejoras, puedes entregar un documento en el que expliques:
 - Qué intentaste hacer.
 - Qué elementos del código analizaste.
 - Qué mejoras querías aplicar y por qué.
 - Qué dificultad encontraste.

 **Ambas opciones son válidas y consideradas como superación del ejercicio.**

Lo importante es demostrar que **has reflexionado sobre el diseño** y que entiendes el propósito de las mejoras propuestas.

PROPUESTA MEJORAS PARA MEJORAR UN CURRICULUM VITAE HECHO EN FLUTTER

1. Añadir una foto de perfil — Humanizar y centrar la atención

Diseño:

Una imagen de perfil transforma un currículum genérico en algo personal y memorable. Funciona como un “punto focal”, es decir, lo primero que el ojo busca. En diseño visual, tener un punto focal claro mejora la estética y la navegación.

Recomendaciones:

- Usa una foto circular (más amigable que cuadrada).
- Colócala centrada arriba del nombre.
- Que tenga buen contraste con el fondo (fondo blanco sobre fondo lavanda, por ejemplo).

En Flutter:

```
CircleAvatar(
  radius: 40,
  backgroundImage: AssetImage('assets/profile.jpg'),
)
```

2. Colores suaves y fondo distinto del blanco — Profesionalismo moderno

Diseño:

El blanco absoluto puede parecer sin terminar. Un fondo gris muy claro, lavanda o azul suave da más sofisticación y menos fatiga visual. Además, permite destacar las tarjetas o contenedores blancos por encima del fondo, creando profundidad.

Psicología del color:

- Lavanda transmite calma, creatividad y enfoque.
- Grises suaves dan neutralidad y limpieza.

En Flutter:

```
Scaffold(  
  backgroundColor: Color(0xFFFF4F3FB), // Lavanda suave  
)
```

3. Separar secciones en tarjetas o bloques — Estructura visual clara

Diseño:

Los humanos leemos en bloques. Si agrupas “Experiencia”, “Educación” y “Habilidades” dentro de tarjetas con sombra y borde redondeado, estás ayudando al lector a navegar.

Esto genera **jerarquía visual**: lo que está dentro de una tarjeta se percibe como relacionado.

Recomendaciones:

- Usa tarjetas o contenedores con margen y padding.
- Borde redondeado + sombra leve = efecto moderno y limpio.

En Flutter:

```
Container(  
  margin: EdgeInsets.symmetric(vertical: 12),  
  padding: EdgeInsets.all(16),  
  decoration: BoxDecoration(  
    color: Colors.white,  
    borderRadius: BorderRadius.circular(10),  
    boxShadow: [  
      BoxShadow(color: Colors.black12, blurRadius: 4, offset: Offset(0,  
2))  
    ],  
  ),  
  child: Column(  
    crossAxisAlignment: CrossAxisAlignment.start,  
    children: [Text("Experiencia Laboral"), ...],  
  ),  
)
```

◆ 4. Iconografía funcional — Reforzar la comprensión inmediata

Diseño:

El uso de iconos junto a texto ( Teléfono,  Email,  Educación) acelera la lectura y mejora la comprensión.

En diseño visual se conoce como "**doble codificación**": una misma información representada por texto + imagen (ícono), lo que facilita la lectura incluso para personas con poco tiempo.

Recomendaciones:

- Usa íconos pequeños y alineados verticalmente al texto.
- No abuses de colores: mantenlos grises o de acento suave.

En Flutter:

```
Row(
  children: [
    Icon(Icons.email, size: 16, color: Colors.grey[700]),
    SizedBox(width: 6),
    Text('juan.perez@example.com'),
  ],
)
```

◆ 5. Jerarquía tipográfica — Diseñar para el escaneo, no la lectura completa

Diseño:

Un reclutador o lector no lee el CV, lo **escanea**. Para ello, necesitas jerarquía:

- **Nombre** → grande, negrita.
- **Puesto** → mediano, color secundario.
- **Secciones** → con separación clara.
- **Detalles** → texto gris más pequeño.

Esto guía al lector sin necesidad de forzarlo a procesar todo el contenido.

En Flutter:

```
Text(
  'Juan Pérez García',
  style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold),
),
Text(
  'Desarrollador Flutter',
  style: TextStyle(fontSize: 16, color: Colors.grey[600]),
),
```

◆ 6. Títulos de sección con iconos — Crear anclas visuales

Diseño:

Los títulos con iconos (กระเป๋า Experiencia, 🎓 Educación) funcionan como **puntos de referencia visual**.

Permiten que el usuario se oriente rápidamente al escanear el documento.

Cada ícono actúa como una “*bandera visual*” que marca dónde empieza una nueva idea.

Recomendaciones:

- Mantén un mismo estilo de ícono (por ejemplo, todos Material o todos Outline).
- Usa un solo color de acento (azul, violeta o gris oscuro).
- Añade un pequeño Divider debajo para separar visualmente el título del contenido.

En Flutter:

```
Row(
  children: [
    Icon(Icons.work, color: Colors.indigo),
    SizedBox(width: 8),
    Text(
      'Experiencia Laboral',
      style: TextStyle(fontWeight: FontWeight.bold, fontSize: 16),
    ),
  ],
),
Divider(thickness: 1, height: 16),
```

◆ 7. Espaciado y ritmo visual — Airar el contenido

Diseño:

El espacio *no es vacío*, es una herramienta de organización.

Un error común de principiantes es llenar todo el espacio con texto sin dejar respiración visual.

El “aire” entre secciones permite al ojo descansar y agrupar la información correctamente.

Reglas útiles:

- Usa más espacio entre secciones (p. ej. 24px).
- Menos espacio entre título y texto relacionado (p. ej. 8–12px).
- Sé consistente: el mismo espaciado entre secciones debe repetirse en todo el documento.

En Flutter:

```
SizedBox(height: 24), // separación entre secciones
SizedBox(height: 8), // separación dentro de La misma sección
```

◆ 8. Contenido técnico preciso — Profesionalismo en la redacción

Diseño de contenido:

El diseño no solo es visual, también textual.

Un currículum con texto genérico (“mantenimiento de equipos”) transmite menos valor que uno con lenguaje técnico específico (“soporte técnico y configuración de redes Windows/Linux”).

Esto también tiene impacto visual: un texto más estructurado, con viñetas bien alineadas, genera un **ritmo de lectura limpio**.

Recomendaciones:

- Usa frases cortas y precisas.
- Empieza con verbos de acción (“Desarrollo”, “Implementación”, “Diseño”).
- Evita bloques grandes de texto.

En Flutter:

```
Column(  
  mainAxisAlignment: MainAxisAlignment.start,  
  children: const [  
    Text('• Implementación de interfaces responsivas.'),  
    Text('• Integración con APIs REST y Firebase.'),  
  ],  
)
```

◆ 9. Diseño responsive — Accesible y adaptable

Diseño:

Un CV digital debería verse bien tanto en una laptop como en un móvil. El diseño responsive no es solo cuestión técnica, sino también estética: mantener el equilibrio entre los elementos al cambiar el ancho de pantalla.

Recomendaciones:

- Usa anchuras relativas (porcentaje o MediaQuery).
- Centra el contenido principal.
- Evita que los textos largos se corten o se salgan del viewport.

En Flutter:

```
LayoutBuilder(  
  builder: (context, constraints) {  
    double maxWidth = constraints.maxWidth > 600 ? 600 :  
    constraints.maxWidth;  
    return Center(  
      child: Container(  
        width: maxWidth,
```

```
        child: Column(children: [...]),
    ),
);
},
)
```

◆ 10. Composición centrada vs. alineada a la izquierda — Estructura y equilibrio

Diseño:

- **Centrado** → transmite estabilidad, serenidad y foco (ideal para el encabezado con foto y nombre).
- **Alineado a la izquierda** → da estructura y profesionalismo (ideal para el cuerpo del CV: experiencia, educación, etc.).

Equilibrio recomendado:

Encabezado centrado + cuerpo alineado a la izquierda = diseño armonioso y natural.

En Flutter:

```
Column(
  mainAxisAlignment: MainAxisAlignment.start, // para secciones
  children: [
    Center(child: Text('Curriculum de Juan Pérez', style: ...)), // 
encabezado centrado
    SizedBox(height: 24),
    Text('Experiencia Laboral', style: ...), // cuerpo alineado a la
izquierda
  ],
)
```