

Diseño aplicado a interfaces web con Flutter

Unidad 06. Caso práctico 01



Autor: Sergi García

Actualizado Noviembre 2025

Licencia



Reconocimiento - No comercial - CompartirlIgual (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 - Pantalla de registro de formulario	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
Propuestas de mejora para una pantalla de registro de formulario en Flutter	5
1. Estilizar el encabezado — Jerarquía visual fuerte y propósito claro	5
2. Añadir iconos funcionales a campos — Refuerzo visual e intuitivo	6
3. Colores de validación claros y consistentes — Feedback visual inmediato	6
4. Fondo suave lavanda y área principal centrada — Profesionalismo moderno	7
5. Claridad en campos opcionales vs. obligatorios — Guía de lectura	8
6. Diseño del checkbox más visible y accesible	8
7. Botón de envío desactivado con retroalimentación visual	9
8. Espaciado vertical y ritmo visual — Diseño respirable	9
9. Pie de página informativo y sutil	10
10. Composición centrada y responsive — Escalable y legible en cualquier dispositivo	10
11. Imagen institucional con bordes redondeados y marco negro — Profesionalismo y estructura	11

UNIDAD 06. CASO PRÁCTICO 01

DE UNA INTERFAZ BÁSICA A UNA VERSIÓN PROFESIONAL - PANTALLA DE REGISTRO DE FORMULARIO

En esta actividad trabajaremos en la mejora del diseño visual y estructural de un formulario de registro en una bolsa de trabajo, 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):

Bolsa de Empleo del Instituto



Registro Bolsa de Empleo

Nombre completo *	<input type="text"/>	
Este campo es obligatorio		
Correo electrónico *	<input type="text"/>	
Introduce un correo válido		
Teléfono *	<input type="text"/>	
Debe tener 9 dígitos numéricos		
Perfil de LinkedIn (opcional)	<input type="text"/>	
Puedes dejarlo en blanco		
<input type="checkbox"/> Acepto la política de privacidad *		
Enviar		

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

Bolsa de Empleo del Instituto



Registro Bolsa de Empleo

Completa tus datos para formar parte de nuestra bolsa de empleo

Nombre completo *	<input type="text"/>	
Revisa este campo		
Correo electrónico *	<input type="text"/>	
Revisa este campo		
Teléfono *	<input type="text"/>	
Revisa este campo		
Perfil de LinkedIn (opcional)	<input type="text"/>	
Campo válido		
<input type="checkbox"/> Acepto la política de privacidad *		
► Enviar formulario		

 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 formulario.
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.

PROPUESTAS DE MEJORA PARA UNA PANTALLA DE REGISTRO DE FORMULARIO EN FLUTTER

1. Estilizar el encabezado — Jerarquía visual fuerte y propósito claro

Diseño:

En la versión mejorada se ha reforzado el encabezado aplicando **tamaño, colores y estructura centrada** para destacar el objetivo del formulario.

- Se incorpora un **título principal claro**, un **subtítulo explicativo** y el **logo centrado**.
- Se usa color de acento (Colors.indigo) en el título para reforzar identidad institucional.
- El subtexto explica claramente qué se va a hacer, lo cual orienta al usuario.

En Flutter:

```
Column(  
  children: [  
    Image.asset('assets/logo.png', height: 64),  
    const SizedBox(height: 12),  
    Text(  
      'Registro Bolsa de Empleo',  
      style: TextStyle(  
        fontSize: 20,  
        fontWeight: FontWeight.bold,  
        color: Colors.indigo,  
      ),  
    ),  
    const SizedBox(height: 4),  
    Text(  
      'Ingresa tus datos para registrarte',  
      style: TextStyle(  
        color: Colors.indigo,  
        fontSize: 16,  
        fontWeight: FontWeight.bold,  
      ),  
    ),  
  ],  
);
```

```
'Completa tus datos para formar parte de nuestra bolsa de
empleo',
style: TextStyle(fontSize: 14, color: Colors.grey[600]),
),
],
),
),
```

◆ 2. Añadir iconos funcionales a campos — Refuerzo visual e intuitivo

Diseño:

La versión mejorada añade **iconos dentro de los campos de entrada** (`TextField`) que representan su función:

 para nombre,  para email,  para teléfono,  para LinkedIn.

Esto ayuda a identificar de forma inmediata el propósito del campo, incluso sin leer el texto.

En Flutter:

```
TextField(
decoration: InputDecoration(
prefixIcon: Icon(Icons.email),
labelText: 'Correo electrónico *',
border: OutlineInputBorder(),
),
)
```

◆ 3. Colores de validación claros y consistentes — Feedback visual inmediato

Diseño:

El formulario ahora **resalta en verde los campos válidos** y en **rojo los inválidos**, con bordes suaves y un diseño visualmente claro.

Esto mejora el feedback del usuario al instante, alineándose con buenas prácticas de accesibilidad.

En Flutter:

```
TextField(
validator: (value) {
if (value == null || value.isEmpty) return 'Campo obligatorio';
return null;
},
decoration: InputDecoration(
prefixIcon: Icon(Icons.person),
border: OutlineInputBorder(),
errorStyle: TextStyle(color: Colors.red),
enabledBorder: OutlineInputBorder(
borderSide: BorderSide(color: Colors.grey.shade400),
```

```
  ),
  focusedBorder: OutlineInputBorder(
    borderSide: BorderSide(color: Colors.indigo),
  ),
  errorBorder: OutlineInputBorder(
    borderSide: BorderSide(color: Colors.red),
  ),
  focusedErrorBorder: OutlineInputBorder(
    borderSide: BorderSide(color: Colors.redAccent),
  ),
),
)
```

◆ 4. Fondo suave lavanda y área principal centrada — Profesionalismo moderno

Diseño:

El uso de un **fondo color lavanda claro** (en AppBar y fondo general) le da un toque institucional y agradable. Esto sustituye al fondo blanco plano, que se percibe como más crudo o genérico.

Además, todo el formulario está centrado, con buen espaciado lateral, lo que mejora la legibilidad.

En Flutter:

```
Scaffold(
  backgroundColor: Color(0xFFFF4F3FB), // Fondo Lavanda
  appBar: AppBar(
    backgroundColor: Color(0xFFE6E6FA), // Tono más fuerte para AppBar
    title: Text('Bolsa de Empleo del Instituto'),
    centerTitle: true,
  ),
  body: Center(
    child: ConstrainedBox(
      constraints: BoxConstraints(maxWidth: 600),
      child: Padding(
        padding: const EdgeInsets.all(16),
        child: Form(...),
      ),
    ),
  ),
)
```

◆ 5. Claridad en campos opcionales vs. obligatorios — Guía de lectura

Diseño:

Ahora los campos obligatorios están marcados con * y los opcionales como "(opcional)".

Además, los mensajes de ayuda debajo de cada campo son más claros: "Campo válido" o "Revisa este campo".

Esto mejora **la experiencia cognitiva** del usuario al eliminar dudas.

En Flutter:

```
TextField(  
  decoration: InputDecoration(  
    labelText: 'Perfil de LinkedIn (opcional)',  
    helperText: 'Puedes dejarlo en blanco',  
    prefixIcon: Icon(Icons.link),  
    border: OutlineInputBorder(),  
  ),  
)
```

◆ 6. Diseño del checkbox más visible y accesible

Diseño:

El checkbox se ha alineado mejor, usando una fuente clara y espacio apropiado.

Además, se integra mejor visualmente con el resto del formulario, permitiendo una lectura más cómoda.

Esto es importante porque el **aceptar política de privacidad es obligatorio**, por lo tanto debe estar bien visible pero no agresivo.

En Flutter:

```
CheckboxListTile(  
  value: isChecked,  
  onChanged: (value) => setState(() => isChecked = value!),  
  title: Text('Acepto la política de privacidad *'),  
  controlAffinity: ListTileControlAffinity.leading, // Icono a La  
izquierda  
)
```

◆ 7. Botón de envío desactivado con retroalimentación visual



Diseño:

El botón “Enviar formulario” tiene un aspecto más **moderno y claro**, y cambia su apariencia según el estado:

- **Desactivado:** gris claro, texto en gris.
- **Activado:** color principal (azul/índigo) + ícono de envío + contraste adecuado.

Esto mejora la **accesibilidad** y evita confusión visual sobre si el botón es interactivo o no.



En Flutter:

```
ElevatedButton.icon(  
    onPressed: isFormValid ? _enviarFormulario : null, // Se activa sólo  
    si el formulario es válido  
    icon: Icon(Icons.send),  
    label: Text('Enviar formulario'),  
    style: ElevatedButton.styleFrom(  
        minimumSize: Size.fromHeight(48),  
        shape: StadiumBorder(),  
    ),  
)
```

◆ 8. Espaciado vertical y ritmo visual — Diseño respirable



Diseño:

Se ha aplicado un **mejor ritmo visual vertical**, permitiendo que cada campo tenga espacio suficiente arriba y abajo. El resultado es un formulario que **no abruma**, aunque contenga muchos elementos. Esto es esencial en diseño visual: el espacio bien usado mejora la experiencia tanto como los colores o tipografía.



En Flutter:

```
Column(  
    children: [  
        SizedBox(height: 16),  
        TextFormField(...),  
        SizedBox(height: 12),  
        TextFormField(...),  
        SizedBox(height: 20),  
        CheckboxListTile(...),  
        SizedBox(height: 24),  
        ElevatedButton(...),  
    ],  
)
```

◆ 9. Pie de página informativo y sutil

Diseño:

Se añadió un **footer** al final con una línea legal o institucional:
“© 2025 IES Tu instituto — Bolsa de Empleo”.

Este tipo de texto transmite **profesionalismo y credibilidad**. Además, está alineado con el tono visual de la app (color gris claro, pequeño).

En Flutter:

```
Center(  
    child: Text(  
        '(c) 2025 IES Tu instituto -- Bolsa de Empleo',  
        style: TextStyle(fontSize: 12, color: Colors.grey[600]),  
    ),  
)
```

◆ 10. Composición centrada y responsive — Escalable y legible en cualquier dispositivo

Diseño:

La estructura ahora está centrada en pantalla, con un **ancho máximo limitado**, lo que permite que el contenido:

- No se estire en pantallas grandes.
- No quede pegado a los bordes en pantallas pequeñas.

Esto asegura una **experiencia visual óptima en desktop y móvil**.

En Flutter:

```
Center(  
    child: ConstrainedBox(  
        constraints: BoxConstraints(maxWidth: 600), // Máximo ancho para  
        evitar estiramiento  
        child: Padding(  
            padding: const EdgeInsets.all(16),  
            child: Form(...),  
        ),  
    ),  
)
```

◆ 11. Imagen institucional con bordes redondeados y marco negro — Profesionalismo y estructura

Diseño:

Una imagen (como el logo del instituto) sin bordes ni integración visual puede parecer flotante o informal.

Al añadir **esquinas redondeadas** y un **borde negro fino**, se genera una sensación de estructura, elegancia y cohesión con el diseño general del formulario.

Esto también delimita visualmente la zona del encabezado, haciendo que la parte superior del formulario se perciba más cuidada y profesional.

Recomendaciones:

- Usa un radio de 16 para bordes suaves pero notorios.
- Asegúrate de que la imagen se recorte con ClipRRect para que no sobresalga del borde.
- Mantén el borde del mismo color que el texto o título para coherencia.

En Flutter:

```
Container(  
  decoration: BoxDecoration(  
    border: Border.all(  
      color: Colors.black, // Color del borde exterior  
      width: 2,           // Grosor del borde  
    ),  
    borderRadius: BorderRadius.circular(16), // Bordes redondeados del  
    contenedor  
  ),  
  child: ClipRRect(  
    borderRadius: BorderRadius.circular(16), // Bordes redondeados  
    aplicados a la imagen  
    child: Image.asset(  
      'assets/images/logo.png',  
      height: 100,  
      fit: BoxFit.cover,  
    ),  
  ),  
,
```