**HeroTextButtonMedia**

El componente **HeroTextButtonImg** es un elemento reusable para la sección "Hero" de una landing page. Combina un bloque de contenido textual (título, subtítulo, lista de viñetas y un botón) con un bloque de contenido multimedia, el cual puede ser una imagen o un video, según se especifique mediante las props.

**Características**

* **Responsive:** Se adapta a diferentes tamaños de pantalla utilizando utilidades de Tailwind CSS.
* **Animaciones:** Integra animaciones de entrada con Framer Motion.
* **Contenido Multimedia Condicional:** Permite mostrar ya sea una imagen (usando Next.js Image) o un video (<video> HTML) según la prop mediaType.
* **Reutilizable:** Diseñado para ser configurado mediante props y utilizado en distintas secciones o páginas.

**Props**

| **Prop** | **Tipo** | **Valor por defecto** | **Descripción** |
| --- | --- | --- | --- |
| **title** | string | — | Texto principal que se muestra en la parte izquierda del componente. |
| **subtitle** | string | — | Subtítulo que refuerza el mensaje principal, se muestra debajo del título. |
| **bulletPoints** | string[] | [] | Array de strings que se renderizan en una lista con viñetas. Cada elemento describe una ventaja o beneficio. |
| **buttonText** | string | — | Texto que se muestra en el botón. |
| **buttonLink** | string | — | URL a la que redirige el botón. Por ejemplo, un enlace a WhatsApp. |
| **mediaSrc** | string | — | Ruta o URL del recurso multimedia (imagen o video). |
| **mediaType** | string | "image" | Define el tipo de contenido a mostrar. Acepta "image" (por defecto) o "video". |

**Estructura y Funcionamiento**

El componente se divide en dos columnas principales:

1. **Columna Izquierda (Contenido Textual):**
   * **Título (h1):** Muestra el título principal.
   * **Subtítulo (h2):** Muestra el subtítulo.
   * **Lista de Viñetas (ul > li):** Lista las ventajas o beneficios.
   * **Botón (a):** Botón interactivo que redirige a la URL especificada en buttonLink.
2. **Columna Derecha (Contenido Multimedia):**
   * **Media Condicional:** Si mediaType es "video", se renderiza un elemento <video> con controles; en caso contrario, se utiliza el componente Image de Next.js para mostrar una imagen.

Ambas columnas incorporan animaciones de entrada desde direcciones opuestas (izquierda para el contenido textual y derecha para la multimedia) usando Framer Motion.

**Ejemplo de Implementación**

**Código del Componente**

jsx

Copiar

"use client";

import React from "react";

import Image from "next/image";

import { motion } from "framer-motion";

const HeroTextButtonImg = ({

title,

subtitle,

bulletPoints = [],

buttonText,

buttonLink,

mediaSrc,

mediaType = "image", // "image" o "video"

}) => {

return (

<section className="py-spacing-6 bg-gray-100 items-start justify-between gap-4">

<div className="w-full px-spacing-2 lg:px-spacing-4 flex flex-col lg:flex-row items-center">

{/\* Columna Izquierda: Textos y Botón \*/}

<motion.div

className="w-full lg:w-1/2 mb-6 lg:mb-0"

initial={{ opacity: 0, x: -50 }}

whileInView={{ opacity: 1, x: 0 }}

viewport={{ once: true }}

transition={{ duration: 0.8, ease: "easeInOut" }}

>

{/\* Título \*/}

<h1 className="section-h5 text-primary">{title}</h1>

{/\* Subtítulo \*/}

<h2 className="section-title text-primary mb-spacing-3">{subtitle}</h2>

{/\* Lista con Viñetas \*/}

<ul className="text-primary list-disc pl-6 mb-spacing-4">

{bulletPoints.map((point, index) => (

<li key={index} className="text-dark section-h5">

{point}

</li>

))}

</ul>

{/\* Botón \*/}

<div className="flex justify-center">

<a href={buttonLink} className="main-button">

{buttonText}

</a>

</div>

</motion.div>

{/\* Columna Derecha: Media (Imagen o Video) \*/}

<motion.div

className="w-full lg:w-1/2 flex justify-center"

initial={{ opacity: 0, x: 50 }}

whileInView={{ opacity: 1, x: 0 }}

viewport={{ once: true }}

transition={{ duration: 0.8, ease: "easeInOut" }}

>

<div className="relative w-full lg:max-w-[800px] h-[300px] sm:h-[500px] lg:h-[600px]">

{mediaType === "video" ? (

<video

controls

className="w-full h-full object-contain rounded-lg shadow-lg"

>

<source src={mediaSrc} type="video/mp4" />

Tu navegador no soporta la reproducción de video.

</video>

) : (

<Image

src={mediaSrc}

alt="Media"

layout="fill"

objectFit="contain"

className="rounded-lg shadow-lg"

/>

)}

</div>

</motion.div>

</div>

</section>

);

};

export default HeroTextButtonImg;

**Ejemplo de Uso**

A continuación se muestra un ejemplo de cómo utilizar el componente en una página, mostrando tanto un ejemplo con imagen como otro con video. En este caso, el enlace del botón se dirige a WhatsApp.

jsx

Copiar

import React from "react";

import HeroTextButtonImg from "../components/HeroTextButtonImg";

export default function Home() {

const bulletPoints = [

"Acceso instantáneo a tu información desde cualquier dispositivo.",

"Integración directa con redes sociales y plataformas profesionales.",

"Actualizaciones en tiempo real para mantener tu perfil siempre actualizado.",

];

// Enlace a WhatsApp: actualiza el número y el mensaje según corresponda.

const whatsappLink =

"https://wa.me/1234567890?text=Hola,%20me%20interesa%20conocer%20más%20sobre%20AnvetCard";

return (

<div className="min-h-screen bg-gray-50 space-y-12">

{/\* Uso del componente con imagen \*/}

<HeroTextButtonImg

title="Transforma tu red de contactos"

subtitle="Descubre el poder de la tarjeta digital moderna"

bulletPoints={bulletPoints}

buttonText="Contáctanos por WhatsApp"

buttonLink={whatsappLink}

mediaType="image"

mediaSrc="/assets/images/hero\_image.jpg"

/>

{/\* Uso del componente con video \*/}

<HeroTextButtonImg

title="Experiencia visual única"

subtitle="Mira cómo AnvetCard revoluciona tu forma de presentarte"

bulletPoints={bulletPoints}

buttonText="Chatea por WhatsApp"

buttonLink={whatsappLink}

mediaType="video"

mediaSrc="/assets/videos/hero\_video.mp4"

/>

</div>

);

}

**Notas Adicionales**

* **Media Responsivo:**  
  Asegúrate de que las rutas especificadas en mediaSrc sean correctas y que los archivos existan en el directorio público de tu proyecto.
* **Personalización:**  
  Puedes ajustar las clases de Tailwind CSS (por ejemplo, py-spacing-6, px-spacing-2, section-h5, etc.) para que se adapten al estilo de tu proyecto.
* **Accesibilidad:**  
  Considera agregar atributos aria-label o role adicionales si es necesario para mejorar la accesibilidad.