

```
document.addEventListener("DOMContentLoaded", () => {
  /*-- Variables --*/
  const boton = document.createElement("button");
  const img_container = document.createElement("figure");
  const img = document.createElement("img");
  const leyenda = document.createElement("legend");
  const imagenes = [
    "./assets/ejercicio1/img1.jpg",
    "./assets/ejercicio1/img2.jpg",
    "./assets/ejercicio1/img3.jpg",
    "./assets/ejercicio1/img4.jpg"
  ];
  let imgActualIndex = 0;

  /*-- Establece propiedades y añade los elementos al documento --*/
  boton.innerText = "Mostrar imagen";
  img_container.id = "img_container"; // Esto es para que se aplique el estilo CSS
  document.body.append(boton);

  /*-- Controla el evento clic del botón --*/
  boton.addEventListener("click", () => {
    img.src = imagenes[imgActualIndex];
    leyenda.innerText = imagenes[imgActualIndex].substring(imagenes[imgActualIndex].lastIndexOf("/") + 1);

    document.body.append(img_container);
    img_container.append(img);
    img_container.append(leyenda);
    boton.remove();

    document.addEventListener("keydown", handleKeyDown);

    /*-- (EXTRA) Funciones táctiles: Lo he intentado pero no he logrado hacerlo funcionar --*/
    // document.body.addEventListener("touchstart", handleTouchStart);
    // document.body.addEventListener("touchmove", handleTouchMove);
    // document.body.addEventListener("touchend", handleTouchEnd);
  });

  /*-- Controla el evento keyDown del teclado desde document --*/
  function handleKeyDown(event) {
    if (event.code === "ArrowRight") {
```

```

        imgActualIndex = (imgActualIndex + 1) % imagenes.length;
    } else if (event.code === "ArrowLeft") {
        imgActualIndex = (imgActualIndex - 1 + imagenes.length) % imagenes.length;
    }
    /*-- Asigna la imagen nueva --*/
    img.src = imagenes[imgActualIndex];
    leyenda.innerHTML = imagenes[imgActualIndex].substring(imagenes[imgActualIndex].lastIndexOf("/"));
}

/*-- (EXTRA) Controla eventos de las pantallas táctiles (lo he intentado pero no funciona) --*/
let toques;

function handleTouchStart() {
    toques = new Array();
}

function handleTouchMove(event) {
    toques[toques.length] = {x:event.changedTouches[0].clientX, y:event.changedTouches[0].clientY};
}

function handleTouchEnd(event) {
    if (toques.length < 100) { // Si la cantidad de toques es menor que 100
        return;
    }

    let vector = {
        x: toques[toques.length - 1].x - toques[0].x,
        y: toques[toques.length - 1].y - toques[0].y,
    };

    let angulo = Math.atan(vector.y / vector.x);
    // if (vector.y < 0 || vector.x < 0) {

    // }

    // if (vector.y < 0 ? !(vector.x < 0) : vector.x < 0) { //OR exclusivo: o y es negativo, o x es negativo -> 2º y 3er cuadrante
    //     angulo += Math.PI;
    // } else if (vector.y < 0 && vector.x < 0) { // Cuarto cuadrante
    //     angulo += Math.PI * 2;
    // }

```

```
/*-- Determina si ha deslizado a la izquierda o a la derecha --*/
if (angulo > 0) { // Ha deslizado hacia la derecha
    imgActualIndex = (imgActualIndex + 1) % imagenes.length;
} else { // Ha deslizado hacia la izquierda
    imgActualIndex = (imgActualIndex - 1 + imagenes.length) % imagenes.length;
}
// if (angulo > Math.PI / 2) { // Izquierda
//     imgActualIndex = (imgActualIndex - 1 + imagenes.length) % imagenes.length;

/*-- Asigna la imagen nueva --*/
img.src = imagenes[imgActualIndex];
leyenda.innerText = imagenes[imgActualIndex].substring(imagenes[imgActualIndex].lastIndexOf("/"));
}
});
```