

```

// ==UserScript==
// @name          Capturar curso Podia V3 lesson-content
// @namespace     local.curso.podia.lessoncontent
// @version       3.0
// @description   Captura lecciones Podia usando #lesson-content
// @match         URL_DEL_CURSO*
// @match         URL_DEL_CURSO/*
// @grant         none
// ==/UserScript==

(function () {
    "use strict";

    const COURSE_SLUG = "curso-descargado";
    const STORAGE_KEY = "yg_curso_v3_" + COURSE_SLUG;
    const WAIT_MS = 2500;

    const sleep = ms => new Promise(r => setTimeout(r, ms));

    function txt(s) {
        return (s || "").replace(/\s+/g, " ").trim();
    }

    function getState() {
        try {
            return JSON.parse(localStorage.getItem(STORAGE_KEY) || "{}");
        } catch {
            return {};
        }
    }

    function setState(state) {
        localStorage.setItem(STORAGE_KEY, JSON.stringify(state));
    }

    function clearState() {
        localStorage.removeItem(STORAGE_KEY);
    }

    function escapeHtml(s) {
        return (s || "")
            .replaceAll("&", "&amp;")
            .replaceAll("<", "&lt;")
            .replaceAll(">", "&gt;")
            .replaceAll("'", "&quot;")
            .replaceAll('"', "&#039;");
    }

    function crearId(url) {
        const u = new URL(url, location.href);
        return "l-" + u.pathname.split("/").filter(Boolean).pop()
            .toLowerCase()
            .replace(/^[^a-z0-9-]/g, "-");
    }

```

```

function esLeccion(url) {
  const u = new URL(url, location.href);
  const parts = u.pathname.split("/").filter(Boolean);

  return (
    parts[0] === "view" &&
    parts[1] === "courses" &&
    parts[2] === COURSE_SLUG &&
    parts.length >= 5
  );
}

function recogerEnlaces(max = Infinity) {
  return
[...document.querySelectorAll(`a[href*="/view/courses/${COURSE_SLUG}"]`)]
    .map(a => ({
      url: new URL(a.getAttribute("href"), location.href).href,
      titulo: txt(a.innerText)
    }))
    .filter(x => esLeccion(x.url))
    .filter((x, i, arr) => arr.findIndex(y => y.url === x.url) === i)
    .slice(0, max);
}

async function esperarLessonContent() {
  for (let i = 0; i < 15; i++) {
    const el = document.querySelector("#lesson-content");

    if (el && txt(el.innerText).length > 80) {
      return el;
    }

    await sleep(1000);
  }

  return document.querySelector("#lesson-content");
}

function limpiarContenido(el) {
  const clon = el.cloneNode(true);

  clon.querySelectorAll("script, style, noscript, svg, iframe, button, form").forEach(x => x.remove());

  clon.querySelectorAll("a[href]").forEach(a => {
    try {
      const abs = new URL(a.getAttribute("href"), location.href);
      a.setAttribute("href", abs.href);
    } catch {}
  });

  clon.querySelectorAll("img[src]").forEach(img => {

```

```

        try {
            img.setAttribute("src", new URL(img.getAttribute("src"),
location.href).href);
        } catch {}
    });

    clon.querySelectorAll("*").forEach(node => {
        [...node.attributes].forEach(attr => {
            if (!["href", "src", "alt", "title"].includes(attr.name)) {
                node.removeAttribute(attr.name);
            }
        });
    });

    return clon.innerHTML.trim();
}

async function capturarPaginaActual() {
    const state = getState();

    if (!state.active || !state.links || !state.links.length) return;

    const actualPath = location.pathname;
    const indexActual = state.links.findIndex(l => new
URL(l.url).pathname === actualPath);

    if (indexActual === -1) return;

    mostrarEstado(`Capturando ${indexActual + 1}/${state.links.length}`);

    await sleep(WAIT_MS);

    const contenido = await esperarLessonContent();

    if (!contenido) {
        console.warn("No encuentro #lesson-content en esta página:",
location.href);
    }

    const html = contenido ? limpiarContenido(contenido) : "<p><strong>No
se encontró contenido.</strong></p>";

    const titulo =
        state.links[indexActual].titulo ||
        txt(document.querySelector("h1")?.innerText) ||
        txt(document.title) ||
        `Lección ${indexActual + 1}`;

    state.pages[indexActual] = {
        id: crearId(location.href),
        titulo,
        url: location.href,
        html
    };
};

```

```

    state.index = indexActual + 1;
    setState(state);

    console.log(`Capturada ${indexActual + 1}/${state.links.length}:
    ${titulo}`);
    console.log("Preview:", txt(contenido?.innerText).slice(0, 300));

    if (state.index >= state.links.length) {
        descargarFinal(state);
    } else {
        location.href = state.links[state.index].url;
    }
}

function descargarFinal(state) {
    const paginas = (state.pages || []).filter(Boolean);

    const indice = paginas.map((p, i) => `
        <li><a href="#${p.id}">${i + 1}. ${escapeHtml(p.titulo)}</a></li>
    `).join("");

    const cuerpo = paginas.map((p, i) => `
        <section id="${p.id}">
            <p class="volver"><a href="#indice">↑ Volver al índice</a></p>
            <h2>${i + 1}. ${escapeHtml(p.titulo)}</h2>
            <p class="origen">Original: ${escapeHtml(p.url)}</p>
            <div class="contenido">
                ${p.html}
            </div>
        </section>
    `).join("\n");

    const salida = `<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Curso local - ${COURSE_SLUG}</title>
<style>
    body {
        font-family: system-ui, -apple-system, BlinkMacSystemFont, "Segoe
UI", sans-serif;
        max-width: 980px;
        margin: 40px auto;
        padding: 0 20px;
        line-height: 1.65;
        color: #111;
        background: #fff;
    }
    h1 { font-size: 2rem; }
    h2 { margin-top: 4rem; padding-top: 2rem; border-top: 1px solid #ddd; }
    h3 { margin-top: 2rem; }
    a { color: #0066cc; }
    img { max-width: 100%; height: auto; }

```

```

code, pre {
    background: #f4f4f4;
    padding: 2px 5px;
    border-radius: 4px;
}
pre {
    padding: 16px;
    overflow-x: auto;
}
#indice {
    background: #f7f7f7;
    padding: 20px;
    border-radius: 12px;
}
.origen, .volver { font-size: .9rem; color: #666; }
.contenido {
    background: #fff;
    padding: 4px 0 20px;
}
</style>
</head>
<body>
<h1>Curso local: ${COURSE_SLUG}</h1>

<nav id="indice">
    <h2>Índice</h2>
    <ol>${indice}</ol>
</nav>

${cuerpo}

</body>
</html>`;

```

```

    const blob = new Blob([salida], { type: "text/html;charset=utf-8" });
    const a = document.createElement("a");
    a.href = URL.createObjectURL(blob);
    a.download = `curso-${COURSE_SLUG}-lesson-content.html`;
    document.body.appendChild(a);
    a.click();
    a.remove();

    clearState();
    alert(`Curso generado. Lecciones capturadas: ${paginas.length}`);
}

function iniciar(max) {
    const links = recogerEnlaces(max);

    if (!links.length) {
        alert("No encuentro lecciones. Abre la página índice online del curso.");
        return;
    }
}

```

```

const state = {
  active: true,
  startedAt: new Date().toISOString(),
  index: 0,
  links,
  pages: []
};

setState(state);
alert(`Empieza captura de ${links.length} lecciones. No cierres la
pestaña.`);
location.href = links[0].url;
}

function mostrarEstado(texto) {
  const status = document.querySelector("#yg-panel-captura-curso .yg-
status");
  if (status) status.textContent = texto;
}

function crearPanel() {
  if (document.getElementById("yg-panel-captura-curso")) return;

  const panel = document.createElement("div");
  panel.id = "yg-panel-captura-curso";
  panel.innerHTML = `
    <div style="
      position: fixed;
      z-index: 999999;
      top: 12px;
      right: 12px;
      background: #111;
      color: white;
      padding: 12px;
      border-radius: 10px;
      font-family: system-ui, sans-serif;
      font-size: 14px;
      box-shadow: 0 4px 18px rgba(0,0,0,.25);
    ">
      <strong>Captura #lesson-content</strong><br>
      <div class="yg-status" style="margin:6px
0;color:#ddd;">Listo</div>
      <button id="yg-test-curso">Test 3</button>
      <button id="yg-full-curso">Todo</button>
      <button id="yg-clear-curso">Borrar estado</button>
    </div>
  `;

  document.body.appendChild(panel);

  document.getElementById("yg-test-curso").onclick = () => iniciar(3);
  document.getElementById("yg-full-curso").onclick = () =>
iniciar(Infinity);

```

```
document.getElementById("yg-clear-curso").onclick = () => {  
    clearState();  
    localStorage.removeItem("yg_curso_claude-para-pymes");  
    localStorage.removeItem("yg_curso_texto_claude-para-pymes");  
    alert("Estado borrado.");  
};  
}  
  
crearPanel();  
capturarPaginaActual();  
})();
```