

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1,viewport-fit=cover" />
  <title>Provincias de Argentina – Juego</title>
  <meta name="theme-color" content="#0ea5e9">
<style>
:root {
  --ok: #22c55e;
  --bad: #ef4444;
  --idle: #f1f5f9;
  --line: #1e293b;
  --line-strong: #0f172a;
  --accent: #0ea5e9;
}
* { box-sizing: border-box; }
body {
  margin: 0; font-family: system-ui, -apple-system, "Segoe UI", Roboto, Ubuntu, Cantarell, "Helvetica Neue", Arial, "Noto Sans";
  background: #f8fafc; color: #0f172a;
}
.wrap { max-width: 1000px; margin: 0 auto; padding: 16px; display: grid; gap: 16px; }
header h1 { margin: 0 0 6px; font-size: 1.35rem; }
header p { margin: 0; color: #475569; font-size: .95rem; }
.card {
  background: #fff; border: 1px solid #e2e8f0; border-radius: 16px;
  padding: 14px 14px 10px; box-shadow: 0 4px 16px rgba(2,6,23,.04);
}
.grid { display: grid; gap: 16px; }
@media (min-width: 900px) { .grid { grid-template-columns: 2fr 1fr; } }
.mapBox {
  width: 100%; aspect-ratio: 4/5;
  border-radius: 12px; overflow: hidden; border: 1px solid #cbd5e1; background: white;
  position: relative;
}
svg { width: 100%; height: 100%; display: block; touch-action: none; }
.legend { display: flex; gap: 12px; align-items: center; margin-top: 8px; color: #475569; font-size: .9rem; }
.dot { width: 12px; height: 12px; border-radius: 3px; display: inline-block; vertical-align: middle; }
.row { display: flex; gap: 8px; }
input[type="text"], input[type="url"] {
  width: 100%; padding: 10px 12px; border-radius: 12px;
  border: 1px solid #cbd5e1; outline: none; font-size: 1rem;
}
input[type="file"] { font-size: .95rem; }
input:focus { border-color: var(--accent); box-shadow: 0 0 0 3px rgba(14,165,233,.2); }
```

```

button {
    border: none; background: var(--accent); color: white; font-weight: 600;
    padding: 10px 14px; border-radius: 12px; cursor: pointer;
}
button.ghost { background: white; color: #0f172a; border: 1px solid #cbd5e1; font-weight: 500; }
.muted { color: #475569; font-size: .9rem; }
.badge { display:inline-block; background:#ecfdf5; color:#065f46; padding:4px 8px; border-radius:10px; font-size:.9rem; font-weight:600; }
.msg { margin-top: 8px; color:#1f2937; font-size: .95rem; min-height: 1.2em; }
.zoomBtns { position:absolute; right:10px; top:10px; display:flex; flex-direction:column; gap:8px; z-index:10; }
.zoomBtns button { background: #fffffcc; color:#0f172a; border:1px solid #cbd5e1; }

</style>
</head>
<body>
<div class="wrap">
    <header class="card">
        <h1> Provincias de Argentina – Juego</h1>
        <p>1) Tocá una provincia del mapa. 2) Escribí su nombre. 3) Correcto = <b style="color:var(--ok)">verde</b>, incorrecto = <b style="color:var(--bad)">rojo</b>.</p>
    </header>

    <div class="grid">
        <!-- Mapa -->
        <section class="card">
            <div id="map" class="mapBox">
                <div class="zoomBtns">
                    <button id="btnPlus" aria-label="Acercar">+</button>
                    <button id="btnMinus" aria-label="Alejar">-</button>
                    <button id="btnReset" aria-label="Restablecer">○</button>
                </div>
            </div>
            <div class="legend">
                <span><span class="dot" style="background:var(--ok)"></span> Correcta</span>
                <span><span class="dot" style="background:var(--bad)"></span> Incorrecta</span>
                <span><span class="dot" style="background:var(--idle)"></span> Sin responder</span>
            </div>
        </section>

        <!-- Controles -->
        <aside class="card">
            <div class="row">
                <input id="answer" type="text" placeholder="¿Cómo se llama la provincia?" autocomplete="off" />
                <button id="checkBtn">Comprobar</button>
            </div>
        </aside>
    </div>
</body>

```

```
<div class="row" style="margin-top:8px; gap: 12px; align-items:center;">
  <span class="badge">Aciertos: <span id="hits">0</span>/<span
id="total">--</span></span>
  <button class="ghost" id="resetBtn">Reiniciar</button>
</div>
<div id="msg" class="msg"></div>

<hr style="margin:12px 0; border:none; border-top:1px solid #e2e8f0;" />
<div class="muted" style="margin-bottom:6px;"><b>Opciones avanzadas</b></div>

<label class="muted" for="geoUrl">URL GeoJSON (recomendado)</label>
<div class="row" style="margin-top:6px;">
  <input id="geoUrl" type="url" placeholder="Pegá la URL y toca Cargar" />
  <button id="loadBtn">Cargar</button>
</div>
<div id="err" class="muted" style="margin-top:6px;color:#b91c1c;"></div>

<div class="muted" style="margin-top:10px;">Archivo local (.geojson / .json)</div>
<input id="geoFile" type="file" accept=".geojson,.json,application/json" />
</aside>
</div>

<footer class="muted" style="text-align:center;">
  Hecho con D3. GeoJSON por defecto: Datos Argentina.
</footer>
</div>

<!-- D3 -->
<script src="https://unpkg.com/d3@7/dist/d3.min.js"></script>

<script>
// ====== Config ======
const DEFAULT_GEOJSON_URL =
"https://apis.datos.gob.ar/georef/api/provincias?formato=geojson";

// ====== Utils ======
function stripDiacritics(str) {
  try { return str.replace(new RegExp("\\\\p{Diacritic}+", "gu"), ""); }
  catch { return str.replace(/[\u0300-\u036f]+/g, ""); }
}
function normalize(s) {
  return stripDiacritics(String(s).toLowerCase().normalize("NFD"))
    .replace(/[^a-z\s]/g, "").replace(/\s+/g, " ").trim();
}
const PROV_SYNONYMS = {
  "caba": "Ciudad Autónoma de Buenos Aires",
  "capital federal": "Ciudad Autónoma de Buenos Aires",
  "buenos aires": "Buenos Aires",
}
```

```

"catamarca": "Catamarca",
"chaco": "Chaco",
"chubut": "Chubut",
"cordoba": "Córdoba",
"corrientes": "Corrientes",
"entre rios": "Entre Ríos",
"formosa": "Formosa",
"jujuy": "Jujuy",
"la pampa": "La Pampa",
"la rioja": "La Rioja",
"mendoza": "Mendoza",
"misiones": "Misiones",
"neuquen": "Neuquén",
"rio negro": "Río Negro",
"salta": "Salta",
"san juan": "San Juan",
"san luis": "San Luis",
"santa cruz": "Santa Cruz",
"santa fe": "Santa Fe",
"santiago del estero": "Santiago del Estero",
"tierra del fuego": "Tierra del Fuego, Antártida e Islas del Atlántico Sur",
"tucuman": "Tucumán",
};

// ===== Estado =====
let features = [];
let status = {};
let selectedId = null;
let selectedName = "";
let hits = 0;

// ===== DOM =====
const mapBox = document.getElementById("map");
const answer = document.getElementById("answer");
const checkBtn = document.getElementById("checkBtn");
const resetBtn = document.getElementById("resetBtn");
const hitsEl = document.getElementById("hits");
const totalEl = document.getElementById("total");
const msgEl = document.getElementById("msg");
const geoUrlInput = document.getElementById("geoUrl");
const loadBtn = document.getElementById("loadBtn");
const errEl = document.getElementById("err");
const fileInput = document.getElementById("geoFile");
const btnPlus = document.getElementById("btnPlus");
const btnMinus = document.getElementById("btnMinus");
const btnReset = document.getElementById("btnReset");

// ===== SVG / D3 =====

```

```

let svg, g, path, projection, zoom, currentTransform;
function initSvg() {
    mapBox.innerHTML = mapBox.innerHTML; // deja los botones
    const svgEl = d3.select(mapBox).append("svg");
    svg = svgEl;
    g = svg.append("g");
    projection = d3.geoMercator();
    path = d3.geoPath(projection);
    currentTransform = d3.zoomIdentity;

    const w = mapBox.clientWidth, h = mapBox.clientHeight;
    svg.attr("viewBox", `0 0 ${w} ${h}`);

    zoom = d3.zoom().scaleExtent([0.8, 8]).on("zoom", (ev) => {
        currentTransform = ev.transform;
        g.attr("transform", currentTransform);
    });
    svg.call(zoom, currentTransform).on("dblclick.zoom", null);
}

function drawMap(geojson) {
    features = geojson.features || [];
    totalEl.textContent = features.length;

    // Ajuste inicial
    const w = mapBox.clientWidth, h = mapBox.clientHeight;
    projection.fitSize([w, h], geojson);

    const areas = g.selectAll("path").data(features, d => d.properties?.id || d.id);
    areas.enter().append("path")
        .attr("d", path)
        .attr("fill", d => colorFor(d))
        .attr("stroke", "#1e293b")
        .attr("stroke-width", 1.2)
        .attr("vector-effect", "non-scaling-stroke")
        .style("cursor", "pointer")
        .on("click", (_, d) => {
            selectedId = d.properties?.id || d.id;
            selectedName = d.properties?.nombre || d.properties?.name || "";
            msgEl.textContent = `Provincia seleccionada: ${selectedName}.`;
            answer.focus(); updateColors();
        });
}

updateColors();
}

function colorFor(d) {
    const id = d.properties?.id || d.id;

```

```

const st = status[id];
if (st === "correct") return "var(--ok)";
if (st === "wrong") return "var(--bad)";
if (id === selectedId) return "#93c5fd";
return "var(--idle)";
}
function updateColors() {
g.selectAll("path")
.attr("fill", d => colorFor(d))
.attr("stroke", d => (status[d.properties?.id || d.id] ? "var(--line-strong)" : "var(--line)"));
}
function resetGame() {
status = {}; selectedId = null; selectedName = "";
hits = 0; hitsEl.textContent = hits; msgEl.textContent = ""; answer.value = "";
updateColors();
// reset zoom
svg.transition().duration(200).call(zoom.transform, d3.zoomIdentity);
}

// ===== Comprobación =====
function checkAnswer() {
if (!selectedId) { msgEl.textContent = "Tocá primero una provincia del mapa.";
answer.focus(); return; }
const userNorm = normalize(answer.value);
const canonical = PROV_SYNONYMS[userNorm] || answer.value.trim();
const isCorrect = normalize(canonical) === normalize(selectedName);
const prev = status[selectedId];
status[selectedId] = isCorrect ? "correct" : "wrong";
if (isCorrect) {
if (prev !== "correct") { hits++; hitsEl.textContent = hits; }
msgEl.textContent = `¡Correcto! Es ${selectedName}.`;
} else { msgEl.textContent = "No. Intentá de nuevo.";}
answer.value = ""; answer.focus(); updateColors();
if (hits === features.length) msgEl.textContent = "¡Felicitaciones! Completaste todas las provincias.";
}
}

// ===== Carga de datos =====
async function loadFromUrl(url) {
errEl.textContent = "";
try {
const res = await fetch(url, { cache: "no-store" });
if (!res.ok) throw new Error(`HTTP ${res.status}`);
const data = await res.json();
if (!data || !data.type) throw new Error("Formato no válido");
if (data.type === "Topology") { throw new Error("Subí/pegá un GeoJSON (TopoJSON no soportado aquí)."); }
initSvg(); drawMap(data);
}

```

```

} catch (e) { errEl.textContent = e.message || "No se pudo cargar el archivo/URL."; }
}
function loadFromFile(file) {
  errEl.textContent = "";
  const reader = new FileReader();
  reader.onload = () => {
    try {
      const data = JSON.parse(reader.result);
      if (!data || !data.type) throw new Error("Archivo JSON no es GeoJSON válido.");
      if (data.type === "Topology") throw new Error("Archivo TopoJSON detectado. Convertí a GeoJSON.");
      initSvg(); drawMap(data);
    } catch (e) { errEl.textContent = e.message || "Archivo inválido."; }
  };
  reader.onerror = () => errEl.textContent = "No se pudo leer el archivo.";
  reader.readAsText(file);
}

// ===== Eventos =====
checkBtn.addEventListener("click", checkAnswer);
answer.addEventListener("keydown", (e) => { if (e.key === "Enter") { e.preventDefault(); checkAnswer(); }});
resetBtn.addEventListener("click", resetGame);
loadBtn.addEventListener("click", () => { const u = geoUrlInput.value.trim(); if (u) loadFromUrl(u); });
geoUrlInput.addEventListener("keydown", (e) => {
  if (e.key === "Enter") { e.preventDefault(); const u = geoUrlInput.value.trim(); if (u) loadFromUrl(u); }
});
fileInput.addEventListener("change", (e) => { const f = e.target.files?.[0]; if (f) loadFromFile(f); });

// Zoom buttons
btnPlus.addEventListener("click", () => svg.transition().duration(200).call(zoom.scaleBy, 1.3));
btnMinus.addEventListener("click", () => svg.transition().duration(200).call(zoom.scaleBy, 1/1.3));
btnReset.addEventListener("click", () => svg.transition().duration(200).call(zoom.transform, d3.zoomIdentity));

// Redibujar al rotar/resize
window.addEventListener("resize", () => {
  if (!features.length) return;
  const w = mapBox.clientWidth, h = mapBox.clientHeight;
  svg.attr("viewBox", `0 0 ${w} ${h}`);
  projection.fitSize([w, h], { type:"FeatureCollection", features });
  g.selectAll("path").attr("d", path);
}, { passive: true });

```

```
// Init
(async function start(){ await loadFromUrl(DEFAULT_GEOJSON_URL); })();

// Tests rápidos en consola
(function runDevTests() {
  const tests = [
    { in: "Córdoba", out: "cordoba" },
    { in: "San Juan", out: "san juan" },
    { in: " Río Negro!!! ", out: "rio negro" },
    { in: "Tucumán", out: "tucuman" },
    { in: "Ciudad Autónoma de Buenos Aires", out: "ciudad autonoma de buenos aires" },
  ];
  tests.forEach((t, i) => {
    const got = normalize(t.in);
    if (got !== t.out) console.error(`Test normalize #${i+1} FALLÓ: esperado='${t.out}', obtenido='${got}'`);
    else console.log(`Test normalize #${i+1} OK`);
  });
})();
</script>
</body>
</html>
```