<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8" />

<title>Pushutropa Cheats Sims 4</title>

<meta name="viewport" content="width=device-width,initial-scale=1" />

<style>

:root {

--bg:#f0f2ff;

--card:#ffffff;

--radius:14px;

--shadow:0 20px 50px -10px rgba(0,0,0,0.1);

--accent:#7c5aff;

--text:#1f2340;

--muted:#6e7391;

--code-bg:#f5f5fa;

--transition:.35s cubic-bezier(.4,.2,.2,1);

}

[data-theme="dark"] {

--bg:#0f111a;

--card:#1f2335;

--text:#f0f2ff;

--muted:#8a92b2;

--accent:#7c5aff;

--code-bg:rgba(255,255,255,0.05);

}

\*{box-sizing:border-box;}

body {

margin:0;

background:radial-gradient(ellipse at bottom left,#e8eaff 0%,var(--bg) 80%);

font-family: system-ui,-apple-system,BlinkMacSystemFont,"Segoe UI",Roboto,sans-serif;

color: var(--text);

min-height:100vh;

padding:1rem;

transition: background var(--transition), color var(--transition);

}

.container {

max-width: 1024px;

margin:0 auto;

display:grid;

gap:1rem;

}

header {

display:flex;

flex-wrap:wrap;

gap:0.5rem;

align-items:center;

justify-content: space-between;

}

h1 {

margin:0;

font-size:1.9rem;

line-height:1.1;

}

.subtitle {

font-size:0.9rem;

color: var(--muted);

margin-top:4px;

}

.controls {

display:flex;

flex-wrap:wrap;

gap:10px;

align-items:center;

margin-top:6px;

}

.search-wrapper {

position:relative;

flex:1;

min-width:220px;

}

.search-wrapper input {

width:100%;

padding:12px 16px;

border:none;

border-radius:999px;

font-size:1rem;

outline:none;

background:var(--code-bg);

color: var(--text);

box-shadow: inset 0 6px 18px rgba(0,0,0,0.08);

transition: background var(--transition);

}

.tag-list {

display:flex;

flex-wrap:wrap;

gap:8px;

margin:8px 0 4px;

}

.tag {

background:rgba(124,90,255,0.1);

padding:6px 12px;

border-radius:999px;

font-size:0.7rem;

cursor:pointer;

display:inline-flex;

gap:6px;

align-items:center;

transition:filter .2s;

border:2px solid transparent;

}

.tag.active {

background: var(--accent);

color:#fff;

border-color:#fff;

}

.toggle, .lang-switch {

background: none;

border: 2px solid var(--accent);

padding:6px 14px;

border-radius:999px;

cursor:pointer;

font-size:0.75rem;

font-weight:600;

display:inline-flex;

align-items:center;

gap:6px;

transition: background var(--transition), color var(--transition);

}

.grid {

display:grid;

gap:1rem;

grid-template-columns: repeat(auto-fit,minmax(300px,1fr));

margin-top:8px;

}

.card {

background: var(--card);

padding:16px 18px;

border-radius: var(--radius);

position:relative;

overflow:hidden;

box-shadow: var(--shadow);

display:flex;

flex-direction:column;

gap:10px;

transition: transform .2s;

}

.card:hover { transform: translateY(-3px); }

.card h3 {

margin:0;

font-size:1.15rem;

display:flex;

align-items:center;

gap:8px;

}

.category-label {

font-size:0.55rem;

text-transform: uppercase;

letter-spacing:1px;

background:rgba(124,90,255,0.08);

padding:5px 12px;

border-radius:999px;

display:inline-block;

margin-bottom:4px;

font-weight:600;

}

.code-block {

background: var(--code-bg);

padding:10px 14px;

border-radius:8px;

font-family: "SFMono-Regular",Consolas,"Andale Mono",monospace;

position:relative;

user-select: all;

overflow-wrap: anywhere;

display:flex;

justify-content: space-between;

align-items:center;

gap:10px;

}

.code-text {

flex:1;

margin:0;

font-size:0.9rem;

word-break: break-all;

}

.actions {

margin-top:auto;

display:flex;

gap:8px;

flex-wrap:wrap;

align-items:center;

}

.btn {

cursor:pointer;

padding:8px 14px;

border:none;

border-radius:8px;

font-weight:600;

font-size:0.75rem;

display:inline-flex;

align-items:center;

gap:6px;

background: var(--accent);

color:#fff;

transition:filter .2s;

}

.btn.outline {

background: none;

color: var(--accent);

border:2px solid var(--accent);

}

.favorite {

margin-left:auto;

cursor:pointer;

font-size:1.2rem;

line-height:1;

user-select:none;

}

.small {

font-size:0.65rem;

padding:6px 10px;

}

.pill {

background: rgba(0,0,0,0.05);

padding:6px 12px;

border-radius:999px;

font-size:0.55rem;

display:inline-block;

}

.no-results {

grid-column:1/-1;

text-align:center;

padding:60px 20px;

background:rgba(255,255,255,0.05);

border-radius:10px;

font-size:1rem;

}

.credit {

font-size:0.65rem;

text-align:center;

margin-top:20px;

color: var(--muted);

}

.copy-feedback {

position:absolute;

top:8px;

right:8px;

background:rgba(0,0,0,0.3);

padding:6px 10px;

border-radius:6px;

font-size:0.6rem;

opacity:0;

pointer-events:none;

transition:opacity .25s;

}

.suggestion-box {

background: var(--card);

padding:14px 16px;

border-radius:12px;

box-shadow: var(--shadow);

display:flex;

flex-direction:column;

gap:10px;

}

.suggestion-list {

max-height:180px;

overflow:auto;

background: var(--code-bg);

border-radius:8px;

padding:8px;

font-size:0.8rem;

}

.favorite-badge {

background: #ffe066;

color: #1f2340;

padding:4px 10px;

border-radius:999px;

font-size:0.55rem;

margin-right:6px;

display:inline-block;

}

.lang-toggle {

display:inline-flex;

gap:4px;

}

.tag svg, .icon-small { width:16px; height:16px; display:inline-block; vertical-align:middle; }

.category-icon { width:18px; height:18px; margin-right:6px; }

.flex { display:flex; gap:6px; align-items:center; }

@media (max-width:1000px){

h1 { font-size:1.5rem; }

}

</style>

</head>

<body>

<div class="container" id="app">

<header>

<div>

<h1 data-i18n="title">Pushutropa Cheats Sims 4</h1>

<div class="subtitle" data-i18n="subtitle">Busca por categoría, palabra o guarda tus favoritos.</div>

</div>

<div class="controls">

<div class="search-wrapper">

<input id="search" placeholder="Buscar: dinero, habilidades..." aria-label="Buscar cheats" />

</div>

<button class="toggle" id="theme-toggle" aria-label="Cambiar tema">🌙 <span id="theme-label">Dark</span></button>

<div class="lang-switch">

<select id="lang-select" aria-label="Idioma">

<option value="es">ES</option>

<option value="en">EN</option>

</select>

</div>

</div>

</header>

<div class="tag-list" id="tag-container"></div>

<div class="grid" id="cards"></div>

<div class="suggestion-box">

<div style="display:flex; justify-content:space-between; align-items:center;">

<div class="flex">

<div class="favorite-badge" data-i18n="suggestions\_title">Sugerencias</div>

<div style="font-size:0.85rem;" data-i18n="suggestions\_desc">Tus seguidores pueden proponer nuevos cheats.</div>

</div>

<button class="btn small" id="clear-suggestions">Limpiar</button>

</div>

<div style="display:flex; gap:12px; flex-wrap:wrap;">

<input id="suggestion-input" placeholder="Ej: cheat de felicidad" style="flex:1; padding:10px; border-radius:8px; border:1px solid rgba(0,0,0,0.1);" />

<button class="btn small" id="add-suggestion">Agregar</button>

</div>

<div class="suggestion-list" id="suggestion-list"></div>

<div style="font-size:0.6rem; color: var(--muted);" data-i18n="suggestions\_note">

Las sugerencias quedan guardadas en el navegador. Si sos tú quien administra la página, puedes copiarlas y meterlas manualmente.

</div>

</div>

<div class="credit" data-i18n="footer">

Hecho para tu tropa. Compartí el link. Podés editar los cheats en el código fácilmente.

</div>

</div>

<template id="card-template">

<div class="card">

<div class="category-label"><span class="category-icon"></span><span class="cat-text"></span></div>

<h3><span class="title-text"></span></h3>

<div class="code-block">

<div class="code-text"></div>

<div style="display:flex; gap:8px; align-items:center;">

<button class="btn small copy-btn" data-i18n="copy">Copiar</button>

<div class="favorite" aria-label="Favorito">☆</div>

</div>

</div>

<div class="actions">

<div class="pill desc-text"></div>

<div style="margin-left:auto; display:flex; gap:6px;">

<div class="lang-badge" style="font-size:0.55rem;"></div>

</div>

</div>

<div class="copy-feedback">Copiado</div>

</div>

</template>

<script>

const i18n = {

es: {

title: "Pushutropa Cheats Sims 4",

subtitle: "Busca por categoría, palabra o guarda tus favoritos.",

copy: "Copiar",

suggestions\_title: "Sugerencias",

suggestions\_desc: "Tus seguidores pueden proponer nuevos cheats.",

suggestions\_note: "Las sugerencias quedan guardadas en el navegador. Si sos tú quien administra la página, puedes copiarlas y meterlas manualmente.",

footer: "Hecho para tu tropa. Compartí el link. Podés editar los cheats en el código fácilmente."

},

en: {

title: "Pushutropa Sims 4 Cheats",

subtitle: "Search by category, keyword or save favorites.",

copy: "Copy",

suggestions\_title: "Suggestions",

suggestions\_desc: "Your followers can propose new cheats.",

suggestions\_note: "Suggestions are kept in the browser. If you're managing the page you can copy them manually.",

footer: "Made for your crew. Share the link. You can edit cheats easily in the code."

}

};

const cheats = [

{

title: {es:"Motherlode", en:"Motherlode"},

code: "motherlode",

category: "Dinero",

tags: ["dinero","money","simoleones","rich"],

description: {es:"Agrega 50,000 simoleones.", en:"Adds 50,000 simoleons."},

icon: "💰"

},

{

title: {es:"Kaching", en:"Kaching"},

code: "kaching",

category: "Dinero",

tags: ["dinero","money"],

description: {es:"Agrega 1,000 simoleones.", en:"Adds 1,000 simoleons."},

icon: "💵"

},

{

title: {es:"Casas gratis", en:"Free Real Estate"},

code: "freerealestate on",

category: "Construcción",

tags: ["construcción","casa","hogar","vivienda"],

description: {es:"Casas gratis al mudarse.", en:"Free houses when moving in."},

icon: "🏠"

},

{

title: {es:"Subir lógica a 10", en:"Max Logic Skill"},

code: "stats.set\_skill\_level Major\_Logic 10",

category: "Habilidades",

tags: ["habilidades","skills","lógica","inteligencia"],

description: {es:"Sube lógica a nivel 10.", en:"Raises logic to level 10."},

icon: "🧠"

},

{

title: {es:"Resetear Sim", en:"Reset Sim"},

code: "resetSim Nombre Apellido",

category: "Misc",

tags: ["reset","bug","arreglar"],

description: {es:"Resetea al Sim si se traba.", en:"Resets a stuck Sim."},

icon: "🔄"

},

{

title: {es:"Mover objetos libremente", en:"Move Objects"},

code: "bb.moveobjects",

category: "Construcción",

tags: ["construcción","decorar","objetos"],

description: {es:"Permite colocar objetos sin restricciones.", en:"Place objects freely."},

icon: "🛠️"

},

{

title: {es:"XP rápido (ejemplo)", en:"Fast XP (example)"},

code: "xp Boost",

category: "Carrera / Progreso",

tags: ["trabajo","promoción","experiencia"],

description: {es:"Ejemplo de cheat de progreso.", en:"Example progression cheat."},

icon: "📈"

}

];

const categoryIcons = {

"Dinero": "💰",

"Construcción": "🏗️",

"Habilidades": "🧠",

"Misc": "🔧",

"Carrera / Progreso": "🚀"

};

let activeTag = null;

let currentLang = localStorage.getItem("lang") || "es";

let theme = localStorage.getItem("theme") || "dark";

const favorites = JSON.parse(localStorage.getItem("favorites") || "[]");

const suggestions = JSON.parse(localStorage.getItem("suggestions") || "[]");

const tagContainer = document.getElementById("tag-container");

const cardsContainer = document.getElementById("cards");

const searchInput = document.getElementById("search");

const langSelect = document.getElementById("lang-select");

const themeToggle = document.getElementById("theme-toggle");

const themeLabel = document.getElementById("theme-label");

const suggestionInput = document.getElementById("suggestion-input");

const addSuggestionBtn = document.getElementById("add-suggestion");

const suggestionListEl = document.getElementById("suggestion-list");

const clearSuggestionsBtn = document.getElementById("clear-suggestions");

function applyI18n() {

document.querySelectorAll("[data-i18n]").forEach(el => {

const key = el.getAttribute("data-i18n");

if (i18n[currentLang] && i18n[currentLang][key]) {

el.textContent = i18n[currentLang][key];

}

});

}

function setLang(l) {

currentLang = l;

localStorage.setItem("lang", l);

langSelect.value = l;

applyI18n();

renderCards();

}

function applyTheme() {

document.documentElement.setAttribute("data-theme", theme);

themeLabel.textContent = theme === "dark" ? (currentLang==="es"?"Oscuro":"Dark") : (currentLang==="es"?"Claro":"Light");

themeToggle.textContent = theme === "dark" ? "🌙 " + themeLabel.textContent : "☀️ " + themeLabel.textContent;

}

function toggleTheme() {

theme = theme === "dark" ? "light" : "dark";

localStorage.setItem("theme", theme);

applyTheme();

}

function matchesFilter(cheat) {

const q = searchInput.value.trim().toLowerCase();

let textMatch = (

cheat.title[currentLang].toLowerCase().includes(q) ||

cheat.category.toLowerCase().includes(q) ||

cheat.description[currentLang].toLowerCase().includes(q) ||

cheat.tags.some(t => t.toLowerCase().includes(q))

);

if (q === "") textMatch = true;

let tagMatch = true;

if (activeTag) {

const lower = activeTag.toLowerCase();

tagMatch =

cheat.category.toLowerCase() === lower ||

cheat.tags.some(t => t.toLowerCase() === lower) ||

cheat.title[currentLang].toLowerCase().includes(lower);

}

return textMatch && tagMatch;

}

function setupTags() {

tagContainer.innerHTML = "";

const allBtn = document.createElement("div");

allBtn.className = "tag" + (activeTag===null ? " active" : "");

allBtn.textContent = currentLang === "es" ? "Todo" : "All";

allBtn.addEventListener("click", () => {

activeTag = null;

updateTagActive();

renderCards();

});

tagContainer.appendChild(allBtn);

const favBtn = document.createElement("div");

favBtn.className = "tag" + (activeTag==="\_\_fav" ? " active" : "");

favBtn.innerHTML = `⭐ ${currentLang==="es"?"Favoritos":"Favorites"}`;

favBtn.addEventListener("click", () => {

activeTag = "\_\_fav";

updateTagActive();

renderCards();

});

tagContainer.appendChild(favBtn);

const cats = [...new Set(cheats.map(c => c.category))];

cats.forEach(cat => {

const btn = document.createElement("div");

btn.className = "tag" + (activeTag===cat ? " active" : "");

const icon = categoryIcons[cat] || "📁";

btn.innerHTML = `<span>${icon}</span> ${cat}`;

btn.addEventListener("click", () => {

if (activeTag === cat) activeTag = null;

else activeTag = cat;

updateTagActive();

renderCards();

});

tagContainer.appendChild(btn);

});

}

function updateTagActive() {

document.querySelectorAll(".tag").forEach(t => t.classList.remove("active"));

document.querySelectorAll(".tag").forEach(t => {

const text = t.textContent.trim().toLowerCase();

if (activeTag === null && (currentLang==="es"?text==="todo":text==="all")) t.classList.add("active");

else if (activeTag === "\_\_fav" && text.includes("favoritos")) t.classList.add("active");

else if (activeTag === t.textContent.trim()) t.classList.add("active");

});

}

function renderCards() {

cardsContainer.innerHTML = "";

let filtered;

if (activeTag === "\_\_fav") {

filtered = cheats.filter(c => favorites.includes(c.code));

} else {

filtered = cheats.filter(matchesFilter);

}

if (filtered.length === 0) {

const no = document.createElement("div");

no.className = "no-results";

no.innerHTML = currentLang==="es"

? `No se encontraron cheats para <strong>"${searchInput.value}"</strong>.`

: `No cheats found for <strong>"${searchInput.value}"</strong>.`;

cardsContainer.appendChild(no);

return;

}

filtered.forEach(c => {

const tpl = document.getElementById("card-template");

const clone = tpl.content.cloneNode(true);

clone.querySelector(".title-text").textContent = c.title[currentLang];

clone.querySelector(".cat-text").textContent = c.category;

clone.querySelector(".code-text").textContent = c.code;

clone.querySelector(".desc-text").textContent = c.description[currentLang];

const copyBtn = clone.querySelector(".copy-btn");

const feedback = clone.querySelector(".copy-feedback");

const favEl = clone.querySelector(".favorite");

const langBadge = clone.querySelector(".lang-badge");

const iconSpan = clone.querySelector(".category-icon");

iconSpan.textContent = categoryIcons[c.category] || c.icon || "❖";

favEl.textContent = favorites.includes(c.code) ? "★" : "☆";

favEl.title = currentLang==="es" ? "Agregar/quitar favorito" : "Toggle favorite";

favEl.addEventListener("click", () => {

const idx = favorites.indexOf(c.code);

if (idx === -1) favorites.push(c.code);

else favorites.splice(idx,1);

localStorage.setItem("favorites", JSON.stringify(favorites));

renderCards();

});

copyBtn.textContent = i18n[currentLang].copy;

copyBtn.addEventListener("click", () => {

navigator.clipboard.writeText(c.code).then(() => {

feedback.textContent = currentLang==="es" ? "Copiado" : "Copied";

feedback.style.opacity = "1";

setTimeout(() => { feedback.style.opacity = "0"; }, 1000);

}).catch(() => {

feedback.textContent = currentLang==="es" ? "Error" : "Error";

feedback.style.opacity = "1";

setTimeout(() => { feedback.style.opacity = "0"; }, 1000);

});

});

langBadge.textContent = currentLang === "es" ? "ES" : "EN";

cardsContainer.appendChild(clone);

});

}

function renderSuggestions() {

suggestionListEl.innerHTML = "";

if (suggestions.length === 0) {

suggestionListEl.textContent = currentLang==="es" ? "Aún no hay sugerencias." : "No suggestions yet.";

return;

}

suggestions.slice().reverse().forEach(s => {

const d = document.createElement("div");

d.style.display="flex";

d.style.justifyContent="space-between";

d.style.padding="6px 8px";

d.style.borderBottom="1px solid rgba(0,0,0,0.05)";

d.innerHTML = `<div>${escapeHtml(s.text)}</div><div style="font-size:0.6rem; opacity:.7;">${new Date(s.date || Date.now()).toLocaleString()}</div>`;

suggestionListEl.appendChild(d);

});

}

function escapeHtml(text) {

const div = document.createElement("div");

div.textContent = text;

return div.innerHTML;

}

searchInput.addEventListener("input", () => renderCards());

langSelect.addEventListener("change", e => setLang(e.target.value));

themeToggle.addEventListener("click", toggleTheme);

addSuggestionBtn.addEventListener("click", () => {

const raw = suggestionInput.value.trim();

if (!raw) return;

suggestions.push({text: raw, date: Date.now()});

localStorage.setItem("suggestions", JSON.stringify(suggestions));

suggestionInput.value = "";

renderSuggestions();

});

clearSuggestionsBtn.addEventListener("click", () => {

if (confirm(currentLang==="es"?"Limpiar todas las sugerencias?":"Clear all suggestions?")) {

suggestions.length=0;

localStorage.setItem("suggestions", JSON.stringify(suggestions));

renderSuggestions();

}

});

setLang(currentLang);

applyTheme();

setupTags();

renderSuggestions();

renderCards();

const observer = new MutationObserver(() => updateTagActive());

observer.observe(tagContainer, {childList:true, subtree:true});

</script>

</body>

</html>