SELVA — Technical Whitepaper (v1.0)

Chain: Base (Mainnet)

Standard: ERC-20 (18 decimals)

Token Address: `0x5bD472E9c0fE7A6986Bc8E661BBc092f716133f2`

Primary Market: Uniswap v3 (Base) — SELVA/USDC (0.3% / 3000)

Pool Address: `0xF109456223621006e35A66f4Fb5f934E0E63de09`

> This document describes SELVA’s technical design, market plumbing (DEX/Liquidity), tooling, and operational practices. It is informational and does not constitute financial advice.

1. Overview

SELVA is an ERC-20 token deployed on **Base** designed for practical utility within the SELVA ecosystem—centering on **DEX-native liquidity** (Uniswap v3), **wallet compatibility**, and **automation tooling** for market operations (quotes, approvals, swaps, and LP fee accrual strategies).

Design principles:

- **Simplicity & Compatibility:** standard ERC-20 with **no transfer fees** and **no restrictive hooks**; compatible with Uniswap v3, routers, and Permit2/Universal Router flows.
- **DEX-first Liquidity:** primary pair **SELVA/USDC** na Uniswap v3 (fee tier 0.3%).
- **Transparent Tooling:** open scripts for approvals, previews, swaps (single-hop) e **autoswap** parametrizado (intervalos aleatórios dentro de limites definidos).

2. Contract & Addresses

- **SELVA (ERC-20):**

`0x5bD472E9c0fE7A6986Bc8E661BBc092f716133f2` (18 decimals)

- **Uniswap v3 (Base) — operational addresses used by our tooling:**
- **SwapRouter (v3):** `0x2626664c2603336E57B271c5C0b26F421741e481`
- **Quoter (v2/quoter):** `0x3d4e44Eb1374240CE5F1B871ab261CD16335B76a`
- **Primary Pool (SELVA/USDC, 0.3%):** `0xF109456223621006e35A66f4Fb5f934E0E63de09`
- **Permit2 (canonical):** `0x000000000022D473030F116dDEE9F6B43aC78BA3`
- **Universal Router (used by the bot for UR flows):** address referenced by scripts (subject to protocol deployments); the bot prints the exact addresses it uses at runtime for visibility.

> Note: Some protocol addresses are ecosystem-standard and subject to upstream deployments. The bot prints the exact addresses it uses at runtime for visibility.

3. Token Specification

- **Standard:** ERC-20
- **Decimals:** 18
- **Transfer Fees:** **None** (0%).
- **DEX Compatibility:** confirmed on Uniswap v3 (Base), including **single-hop** USDC■SELVA via fee tier **3000 (0.3%)**.
- **Restrictions / Blacklists / Cooldowns:** none implemented in the deployed token (tested via static calls and live swaps).
- **Ownership & Admin:** if any owner/multisig/timelock is configured, details should be disclosed here with on-chain addresses and privileges. *(Update if applicable.)*

4. Market Structure (Uniswap v3)

- **Primary Pair:** SELVA/USDC (fee 0.3% / 3000); pool: `0xF109...de09`.
- **Concentrated Liquidity:** Uniswap v3 concentrates liquidity in ticks. Fee accrual happens when trades cross active liquidity.
- **Routing:** **Single-hop** USDC■SELVA preferred for price clarity and minimal routing risk. If necessary, multi-hop could be explored provided intermediate pools exist with liquidity.

Operational implications:

- If the price moves **outside** active LP ranges, quotes may revert or return near-zero amounts; the bot includes **pre-flight diagnostics** (pool inspection, quote simulation) to prevent wasteful txs.
- The token is **not fee-on-transfer**, avoiding the common failure mode of v3 swaps.

5. Tooling & Automation

The repository provides **auditable scripts** used during development and operations:

- **Balance & Approval:**
- `approve.js` — handles ERC-20 `approve` to the router; also supports **Permit2** and the Universal Router path (when using UR).
- `index.js` — holds the **ADDRESS** (public wallet) and uses `.env` for the `PRIVATE_KEY`, ensuring secrets stay out of git.*

- **Swap & Quote:**
- `swap.js` — end-to-end swap engine with:
- **Preview** (`swap:preview` / `swap:preview-simple`): simula rota, path, minOut, slippage.
- **Buy/Sell** (`swap:buy-simple`, `swap:sell-simple`): execução single-hop com parâmetros **simplificados** (apenas a quantidade), e **default config** para fee/slippage/deadline/autoapprove.
- **Universal Router Flow** com **Permit2** (approve to Permit2 + Permit2.approve to Universal Router), quando acionado.

- **Autoswap (LP fee accrual strategy tool):**
- `swap.js autoswap --iters N --delay-min A --delay-max B --usdc-min X --usdc-max Y --selva-min U --selva-max V`

Alterna **compras e vendas** em quantias **pseudo-aleatórias** dentro de faixas definidas, com **delays aleatórios** (e.g., 2–3s) para dispersar requisições e reduzir risco de rate-limit no RPC.

Esta rotina visa **induzir volume controlado** na faixa ativa de liquidez, contribuindo para **acúmulo de fees** no LP (desde que haja LP constituída no par e a faixa esteja cobrindo o preço).

> *Secrets handling*: `.env` (com `PRIVATE_KEY`/RPCs) é **excluído do git** via `.gitignore`. O repositório publica apenas **código e ABIs** necessários, nunca chaves.

6. LP Fee Accrual: Considerações Técnicas

- **Taxa do par:** 0.3% (3000).
- **Princípio:** cada swap que cruza a faixa ativa **acumula taxa** proporcional ao volume.
- **Parâmetros operacionais recomendados (exemplo):**
- **Delays (autoswap):** `--delay-min 2 --delay-max 3` segundos (equilíbrio entre dispersão e limites do RPC).
- **Quantidade (USDC/SELVA):** valores variados dentro de **mín/máx** compatíveis com o tamanho de sua **TVL** e do **par** (evite ordens que esgotem a faixa).
- **Risco de slippage:** o bot aplica **slippage padrão** (ex.: 0.5%) e checagens de **minOut**; é possível forçar `minOut=0` para debug, mas **não recomendado** em produção.
- **TVL & Range:** fees dependem do **preço permanecer na faixa** e do **volume**. Uma faixa muito estreita rende mais quando ativa, mas **sai do preço** com facilidade; uma faixa muito larga mantém cobertura, mas **capta menos fees por unit of liquidity**.

7. Wallet & Ecosystem Integration

- **Wallets:** compatível com MetaMask e Base Wallet; token **importável** via address e decimals (18).
- **Metadata & Logos:** SVG/PNG padronizados (transparent background, square 512×512), mantendo **simetria e proporção áurea** no ícone.
- **Coinbase/Base Listing UX:** via Deform/Wallet update, **PNG** funciona bem; **SVG** é aceito/útil, mas não estritamente obrigatório para Wallet (diferente de Coinbase CEX).

8. Security Posture

- **No fee-on-transfer, no honeypot behavior:** testado via `callStatic`/transfers e swaps reais.
- **Approvals & Permit2:** o fluxo Universal Router exige **duas aprovações**:

1) `ERC20.approve(PERMIT2, …)`

2) `Permit2.approve(token, UniversalRouter, …)`

O bot automatiza (`--autoapprove`) se ainda não configurado.

- **Key Management:** a `PRIVATE_KEY` é mantida **somente** em `.env` local; RPC keys (Alchemy) **não** vão para o git.
- **Audits:** se houver auditorias externas, adicione os relatórios/links. Caso contrário, declarar **“No external audit at this time”**.

9. Roadmap (High-Level)

- **Phase 1 — Liquidity & Wallets:**

Publicar logo, metadados, form CoinGecko, melhorar documentação e UX dos scripts.

- **Phase 2 — LP Ops & Analytics:**

Métricas de faixa (ticks), monitoramento de **fee growth**, rebalance on-chain, e simulação de ranges.

- **Phase 3 — Ecosystem Utilities:**

Integrações adicionais (analytics dashboard, alerts), par alternativo (ex.: SELVA/USDbC) conforme oportunidades de liquidez.

10. Disclaimers

This whitepaper is for **informational purposes only** and does not constitute financial, investment, or legal advice. Digital assets are volatile and involve risk of loss. Always **DYOR** and consider your risk tolerance. Protocol addresses and integrations may evolve over time; verify on-chain data and official sources before transacting.

11. Quick Links

- **Token (BaseScan):** *add your BaseScan contract URL*
- **Pool (Uniswap v3, 0.3%):** *add your Uniswap pool URL*
- **Repository / Docs:** *your GitHub or site*
- **Contact:** *email or contact form*
- **Logo Assets:** *link to SVG/PNGs hosted in repo or site*