

# EZ-OS: Documento Fundador (Consolidação)

---

## 1. Definição do Sistema

O EZ-OS (Easy Operating System) é um sistema operacional leve, offline e agnóstico de hardware, projetado especificamente para consoles portáteis baseados em Linux, como o R35S e dispositivos similares. Diferente de interfaces de lançamento convencionais (launchers), o EZ-OS define-se como um Sistema Operacional de Memória Lúdica. Sua função primordial não é a gestão de desempenho gráfico ou a organização estética de bibliotecas de jogos, mas sim o registro factual de eventos de uso e a expressão simbólica desse histórico.

## 2. Princípios Não Negociáveis

A integridade do EZ-OS repousa sobre cinco pilares fundamentais que regem tanto o seu desenvolvimento quanto a sua operação:

- **Factualidade da Memória:** O sistema registra estritamente eventos reais de jogo. Nenhuma narrativa ou dado é inventado ou simulado; o histórico é uma derivação direta da realidade de uso.
- **Estado Padrão de Silêncio:** A interface de usuário opera sob o princípio da mínima interrupção. O sistema apenas comunica informações quando há relevância factual, evitando notificações desnecessárias ou pressões por engajamento.
- **Identidade Procedural Determinística:** A variação visual e a identidade do mascote são geradas a partir de uma semente (seed) única do usuário. Este processo é determinístico: a forma é individual, mas a lógica de evolução é universal e estável.
- **Resiliência à Degradação:** O sistema é projetado para manter total funcionalidade e legibilidade em condições gráficas mínimas, incluindo ASCII

puro, 2-bit e terminais de texto básicos.

- **Isolamento de IA:** Processamentos de inteligência artificial de alta carga nunca são executados localmente no console. O dispositivo limita-se ao registro e à expressão; interpretações profundas são delegadas a camadas externas opcionais.

## 3. Arquitetura do Sistema

---

O EZ-OS é estruturado em quatro domínios isolados que se comunicam exclusivamente através de arquivos JSON validados, garantindo estabilidade e auditabilidade.

Domínio	Função	Responsabilidade Técnica
UI (Interface)	Leitura Simbólica	Renderização da TUI e exibição do mascote ASCII.
Memória	Grafo Factual	Armazenamento e gestão do histórico em estrutura de grafo.
Governança	Regras e Limites	Validação de dados e aplicação das restrições do sistema.
Execução	Integração de Jogos	Interface com emuladores (RetroArch) e registro de sessões.
Galeria	Catálogo de Memória	Organização simbólica e técnica das ROMs baseada em ressonância.
Simbiose	Agente de Hardware	Diagnóstico inteligente e adaptação do sistema ao dispositivo físico.

## 4. Grafo de Memória

---

O núcleo informational do sistema é um grafo de memória armazenado em formato JSON. Esta estrutura assegura que o histórico seja imutável e temporalmente coerente.

- **Nós (Nodes)**: Compreendem a identidade do usuário, metadados de jogos indexados e eventos específicos (Início, Fim, Conquistas).
- **Relações (Edges)**: Estabelecem vínculos de causalidade e cronologia, conectando eventos aos jogos correspondentes e ordenando-os no tempo.

O grafo permite que o sistema recupere o estado exato da experiência do usuário sem a necessidade de processamento heurístico.

## 5. Mascote (Companion)

---

O mascote é a representação visual e simbólica do estado do grafo de memória. Ele atua estritamente como um leitor de dados, nunca como um agente decisor ou simulador de emoções.

- **Representação**: Renderizado em ASCII Art como forma canônica.
- **Evolução**: O mascote atravessa fases de desenvolvimento (Ovo, Jovem, Adulto, Sábio) baseadas exclusivamente no volume e na natureza dos eventos registrados no grafo.
- **Comportamento**: Caracteriza-se pela ausência de falas longas ou simulações de personalidade, limitando-se a expressar o tempo vivido pelo sistema.

## 6. Interface TUI (Terminal User Interface)

---

A interface do EZ-OS é baseada em terminal, priorizando a longevidade e a eficiência em hardware modesto.

- **Layout**: Organização fixa em proporção 4:3, com menu lateral minimalista e área central reservada para o mascote e logs de memória.
- **Estética**: Estilo retro-futurista de alto contraste, utilizando cores planas e evitando efeitos visuais que demandem aceleração de hardware.
- **Filosofia**: A interface valoriza a repetição voluntária e o abandono sem julgamento, respeitando o ritmo do usuário.

## 7. EZ-Gallery: O Catálogo de Memória

---

A EZ-Gallery é a camada de interface entre os arquivos brutos (ROMs) e o grafo factual. Ela transforma uma lista estática de arquivos em um catálogo vivo e simbólico.

- **Ressonância de Memória:** Os jogos não são listados apenas por ordem alfabética, mas sim por sua “ressonância” — um valor derivado da frequência e recência de uso registrado no grafo.
- **Identificação Técnica:** Cada arquivo é identificado via hash (SHA1), garantindo que a memória seja vinculada à identidade correta do jogo, independentemente do nome do arquivo.
- **Estados de Visualização:** A galeria expressa visualmente quais jogos são “vivos” (com muitas memórias) e quais são “dormentes” (ainda não integrados ao grafo).

## 8. EZ-Symbiosis: A Inteligência de Hardware

---

O EZ-OS introduz a camada de **Simbiose**, onde o mascote atua como um elo entre o software e as capacidades físicas do dispositivo.

- **Consciência de Hardware:** O sistema mapeia recursos subutilizados (ex: botões extras, núcleos de CPU ociosos, scripts de otimização) e os integra à lógica de uso.
- **Insights de Mascote:** Através de uma “simbiose” com o dispositivo, o mascote gera tutoriais contextuais e silenciosos, ensinando o usuário a extrair o máximo potencial do seu hardware.
- **Menu Adaptativo:** A interface TUI reorganiza-se dinamicamente para oferecer atalhos e funções que façam sentido para o hardware específico detectado, otimizando o fluxo de trabalho do usuário.

## 9. Governança e Indexação de Dados

---

O EZ-OS opera sob uma política rigorosa de gestão de dados para garantir segurança técnica e conformidade.

- **Indexação:** O sistema utiliza bases de metadados canônicas (Libretro, No-Intro, Redump) para identificar arquivos fornecidos pelo usuário via hash (SHA1/CRC).
- **Metadados:** São utilizados apenas para normalização de nomes, plataformas e publishers.
- **Restrições:** O sistema não hospeda, distribui ou facilita o download de ROMs. Ele reconhece apenas o conteúdo que o usuário já possui localmente.
- **Privacidade:** O funcionamento é 100% offline, com total controle do usuário sobre a ativação de módulos de enriquecimento semântico.

## 8. Estado Atual do Projeto

---

O EZ-OS não é um conceito teórico, mas um sistema implementado e funcional.

- **Execução:** O núcleo do sistema, incluindo o motor do grafo de memória, o gerador procedural do mascote e a interface TUI, está codificado e operacional em ambiente Python.
- **Integração:** Scripts de launcher já realizam o ciclo completo de indexação, execução simulada e registro de eventos.
- **Portabilidade:** A arquitetura agnóstica permite que os artefatos atuais sejam portados para sistemas baseados em ARM (como o R35S) com esforço de adaptação mínimo.

O EZ-OS não tenta prender o jogador. Ele apenas lembra quando ele volta.