



PLANO DE TRABALHO - 7 DIAS ATÉ O LANÇAMENTO



CRONOGRAMA COMPLETO



DIA 1 (HOJE) - VALIDAÇÃO SNES

MANHÃ (9h-12h) - 3 horas

09:00 - 09:30 | Setup & Preparação

- [] Backup completo do projeto
- [] Git commit: "Pre-launch testing phase"
- [] Criar pasta: /test-results/day1/
- [] Preparar planilha de testes (Excel/Google Sheets)

09:30 - 10:00 | Teste #1: Super Mario World

- [] Abrir GUI
- [] Selecionar SNES
- [] Carregar: Super Mario World.smc
- [] Extrair texto
- [] Capturar screenshot do log
- [] Validar extracted_text.txt
- [] Contar strings extraídas: _____
- [] Traduzir (Gemini ou Ollama)
- [] Validar tradução
- [] Tempo total: _____ min
- [] RESULTADO: ☒ PASS / ☒ FAIL

10:00 - 10:30 | Teste #2: Donkey Kong Country 2

- [] Repetir processo acima
- [] Comparar performance vs. Mario
- [] Anotar diferenças
- [] RESULTADO: ☒ PASS / ☒ FAIL

10:30 - 11:30 | Teste #3: Legend of Zelda

- [] Repetir processo acima
- [] Validar textos longos (diálogos)
- [] Verificar quebras de linha preservadas
- [] Tempo de tradução: ____ min
- [] RESULTADO: ☒ PASS / ☒ FAIL

11:30 - 12:00 | Análise de Resultados

- [] Criar relatório: test_results_day1.md
- [] Listar bugs encontrados (se houver)
- [] Priorizar correções
- [] Decidir: passar pro Dia 2 ou corrigir bugs?

ALMOÇO (12h-13h) 🍽️

TARDE (13h-17h) - 4 horas

13:00 - 14:00 | Correção de Bugs (se necessário)

- [] Bug #1: _____
Solução: _____
- [] Bug #2: _____
Solução: _____
- [] Testar novamente após correções

14:00 - 15:00 | Testes Adicionais SNES

- [] Teste opcional #4: Final Fantasy VI
- [] Teste opcional #5: Chrono Trigger
- [] Teste opcional #6: (sua escolha)

Objetivo: Validar robustez do sistema

15:00 - 16:00 | Documentação

- [] Atualizar README.md
- [] Criar: SNES_SUPPORTED_GAMES.md
- [] Listar jogos testados com sucesso

- [] Documentar configurações funcionais
- [] Capturar screenshots dos sucessos

16:00 - 17:00 | Preparação PS1

- [] Baixar ISO PS1 para teste (sugestão: Castlevania SOTN)
- [] Instalar ferramentas: 7-Zip, CDMage (se Windows)
- [] Estudar estrutura ISO PS1
- [] Revisar ps1_config.json
- [] Ajustar offsets baseado em pesquisa

NOITE (19h-21h) - 2 horas *(opcional, se tiver energia)*

19:00 - 20:00 | Pesquisa de Mercado

- [] Visitar Gumroad: pesquisar "rom translation"
- [] Analisar preços de ferramentas similares
- [] Ler reviews de competidores
- [] Anotar pontos fortes/fracos
- [] Definir seu diferencial

20:00 - 21:00 | Planejamento Dia 2

- [] Listar ISOs PS1 disponíveis
- [] Preparar ambiente de teste
- [] Revisar código relacionado a PS1
- [] Git commit: "Day 1 complete - SNES validated"

DIA 2 (AMANHÃ) - VALIDAÇÃO PS1

MANHÃ (9h-13h) - 4 horas

09:00 - 10:00 | Setup PS1

- [] Extrair ISO para pasta de trabalho
- [] Identificar arquivos de texto no ISO
- [] Usar hex editor: localizar strings manualmente
- [] Anotar offsets importantes
- [] Atualizar ps1_config.json com dados reais

10:00 - 11:30 | Teste #1: PS1 ISO Simples

- [] Carregar ISO na GUI
- [] Tentar extrair texto
- [] Analisar log de erros (provavelmente haverá)
- [] Ajustar configuração
- [] Retry até funcionar
- [] RESULTADO: ☒ PASS / ☒ FAIL

11:30 - 13:00 | Debugging & Ajustes

Se falhou:

- [] Verificar encoding (Shift-JIS vs ASCII)
- [] Ajustar offsets
- [] Verificar estrutura de ponteiros
- [] Testar extração manual primeiro
- [] Integrar na GUI depois

Se funcionou:

- [] Testar tradução completa
- [] Validar output
- [] Capturar evidências

ALMOÇO (13h-14h) 🕒

TARDE (14h-18h) - 4 horas**14:00 - 16:00 | Testes PS1 Adicionais**

- [] ISO #2: (diferente do primeiro)
- [] ISO #3: (se der tempo)
- [] Comparar estruturas entre ISOs
- [] Documentar padrões encontrados
- [] Criar template de config genérico

16:00 - 17:00 | Comparação SNES vs PS1

- [] Performance: tempo de extração
- [] Qualidade: textos capturados
- [] Limitações: o que não funciona
- [] Criar tabela comparativa

17:00 - 18:00 | Relatório Dia 2

- [] Documentar descobertas PS1
- [] Atualizar PS1_SUPPORTED_GAMES.md
- [] Listar ISOs testados
- [] Criar FAQ com problemas comuns
- [] Git commit: "Day 2 complete - PS1 validated"

NOITE (19h-21h) - 2 horas**19:00 - 20:00 | Decisão: Adicionar N64/GBA?**

- [] Analisar tempo restante (5 dias)
- [] Avaliar complexidade N64
- [] Decidir: foco em polir SNES+PS1 OU adicionar plataformas?

Recomendação: POLIR O QUE FUNCIONA

- 2 plataformas sólidas > 4 plataformas bugadas
- Você pode adicionar N64 em v1.1 (pós-launch)

20:00 - 21:00 | Planejamento Dias 3-7

- [] Revisar roadmap
- [] Ajustar timeline se necessário
- [] Listar tarefas críticas vs opcionais
- [] Priorizar: o que é ESSENCIAL para lançar?

🧠 DIA 3 - POLIMENTO UX/UI**MANHÃ (9h-13h) - 4 horas****09:00 - 10:30 | Melhorias de Interface**

- [] Adicionar splash screen
 - Logo + "Loading..."
 - Versão + Copyright
- [] Melhorar barra de progresso
 - Mostrar % exato
 - Tempo estimado restante
 - Velocidade (textos/seg)

- [] Adicionar ícones
 - Botões com ícones (extração, tradução, etc)
 - Status indicators (☒ ☒ ☒)

10:30 - 12:00 | Mensagens de Erro User-Friendly

Substituir:

☒ "Exception: FileNotFoundError at line 234"

Por:

☒ "Arquivo ROM não encontrado.
Por favor, verifique se o caminho está correto.
Caminho esperado: /path/to/file.smc"

- [] Revisar TODAS as mensagens de erro
- [] Adicionar dicas de solução
- [] Traduzir para português (se GUI em PT)
- [] Testar cada erro propositalmente

12:00 - 13:00 | Tutorial de Primeiro Uso

- [] Criar wizard de boas-vindas
 - "Bem-vindo ao ROM Translator!"
 - "Passo 1: Selecione uma plataforma"
 - "Passo 2: Carregue uma ROM"
 - "Passo 3: Extraia o texto"
 - Checkbox: "Não mostrar novamente"
- [] Salvar preferência em config.json

ALMOÇO (13h-14h) 🕒

TARDE (14h-18h) - 4 horas

14:00 - 15:30 | Features Extras (se der tempo)

- [] Histórico de traduções
 - Mostrar últimas 10 ROMs processadas
 - Botão "Carregar novamente"
- [] Preview de texto extraído
 - Janela modal com primeiras 50 linhas
 - Scroll para ver mais

- ☐ Estatísticas
 - Total de textos extraídos: _____
 - Tempo de processamento: _____
 - Custo estimado (se API paga): \$_____

15:30 - 17:00 | Tema Dark/Light (se der tempo)

- ☐ Implementar toggle dark/light
- ☐ Testar legibilidade em ambos
- ☐ Salvar preferência
- ☐ Aplicar ao iniciar

17:00 - 18:00 | Testes de Usabilidade

- ☐ Pedir para alguém testar (amigo, familiar)
- ☐ Observar onde eles travam
- ☐ Anotar confusões/dúvidas
- ☐ Ajustar com base no feedback
- ☐ Git commit: "Day 3 complete - UX polished"

DIA 4 - EMPACOTAMENTO

MANHÃ (9h-13h) - 4 horas

09:00 - 11:00 | Build Windows (.exe)

- ☐ Instalar PyInstaller: `pip install pyinstaller`
- ☐ Criar spec file customizado
- ☐ Incluir todos assets (ícones, configs)
- ☐ Build: `pyinstaller --onefile --windowed gui_translator.py`
- ☐ Testar .exe em máquina limpa (VM ou PC diferente)
- ☐ Verificar tamanho (< 100MB ideal)
- ☐ Se muito grande: otimizar (excluir dependências não usadas)

11:00 - 13:00 | Build Linux (AppImage)

- ☐ Instalar appimage-builder
- ☐ Criar AppImage recipe
- ☐ Build AppImage
- ☐ Testar em Ubuntu 22.04
- ☐ Testar em Debian 11
- ☐ Verificar dependências incluídas

TARDE (14h-18h) - 4 horas

14:00 - 15:30 | Redução de Tamanho

```
Se executável > 100MB:  
[ ] Usar UPX compressor  
[ ] Excluir debug symbols  
[ ] Remover imports não usados  
[ ] Considerar pip install --no-deps  
[ ] Meta: < 50MB final
```

15:30 - 17:00 | Instalador (opcional, mas recomendado)

```
Windows:  
[ ] Usar Inno Setup (free)  
[ ] Criar script de instalação  
[ ] Adicionar atalho no Desktop  
[ ] Adicionar entrada no Menu Iniciar  
[ ] Opção de desinstalação  
  
Linux:  
[ ] AppImage já é portátil (OK)  
[ ] Ou criar .deb package (Ubuntu/Debian)
```

17:00 - 18:00 | Validação Final de Builds

```
[ ] Testar instalação completa (Windows)  
[ ] Testar AppImage (Linux)  
[ ] Verificar ativação de licença funciona  
[ ] Testar em máquina SEM Python instalado  
[ ] Documentar requisitos mínimos de sistema  
[ ] Git commit: "Day 4 complete - Builds ready"
```

DIA 5 - DOCUMENTAÇÃO & MARKETING

MANHÃ (9h-13h) - 4 horas

09:00 - 10:30 | README.md Profissional

```
[ ] Seções:  
  - Badges (license, version, platform)
```


- Screenshot principal
- Features list
- Quick start (5 comandos)
- Installation (Windows, Linux, Mac)
- Usage examples
- FAQ
- Contributing
- License
- Support

- [] Usar Markdown avançado (collapsible sections, tables)
- [] Adicionar GIFs demonstrativos

10:30 - 12:00 | Guias Específicos

- [] INSTALLATION.md
 - Windows: passo-a-passo com prints
 - Linux: comandos + troubleshooting
 - Mac: instruções alternativas
- [] USER_GUIDE.md
 - Tour pela interface
 - Como extrair texto
 - Como traduzir
 - Configurações avançadas
- [] TROUBLESHOOTING.md
 - "ROM não carrega" → solução
 - "Tradução falha" → solução
 - "GUI não abre" → solução
 - Logs de debug

12:00 - 13:00 | Documentação Técnica

- [] API_REFERENCE.md (para desenvolvedores)
- [] CONTRIBUTING.md (como contribuir)
- [] CODE_OF_CONDUCT.md
- [] CHANGELOG.md (histórico de versões)

TARDE (14h-18h) - 4 horas

14:00 - 15:30 | Material Visual

- [] Logo profissional
 - Canva.com (free)

- 512x512px PNG
- Fundo transparente
- Estilo retro/pixel art

[] Screenshots (8-10)

- Interface principal
- Extração em progresso
- Tradução completa
- Configurações
- Antes/depois texto

[] GIF animado (< 10MB)

- Screencast: carregar ROM → traduzir → sucesso
- Giphy Capture ou LICEcap
- Loop infinito

15:30 - 17:00 | Vídeo Demo (YouTube)

[] Roteiro:

- 0:00 - Intro (problema que resolve)
- 0:30 - Interface tour
- 1:00 - Demo: traduzir Super Mario World
- 2:00 - Demo: traduzir jogo PS1
- 3:00 - Comparação com ferramentas antigas
- 3:30 - Pricing & onde comprar
- 4:00 - Conclusão + CTA

[] Gravar com OBS Studio

[] Editar com DaVinci Resolve (free)

[] Música de fundo: epidemic sound ou free music archive

[] Thumbnail atrativo (YouTube requirements)

17:00 - 18:00 | Copy de Vendas

- [] Escrever descrição para Gumroad (ver template no LAUNCH_STRATEGY.md)
- [] Criar bullet points persuasivos
- [] Adicionar prova social (se tiver beta testers)
- [] Enfatizar benefícios, não features
- [] Call-to-action claro
- [] Git commit: "Day 5 complete - Documentation ready"

🕒 DIA 6 - SETUP GUMROAD & MARKETING

MANHÃ (9h-13h) - 4 horas

09:00 - 10:30 | Criar Conta Gumroad

- [] Registrar em gumroad.com
- [] Verificar identidade (pode demorar 24-48h)
- [] Configurar pagamento (PayPal ou Stripe)
- [] Definir informações fiscais
- [] Criar perfil de vendedor atrativo

10:30 - 12:30 | Criar Listing do Produto

- [] Nome: "ROM Translation Framework - AI-Powered Translator"
- [] URL: gumroad.com/l/rom-translator
- [] Descrição: (usar template do LAUNCH_STRATEGY.md)
- [] Upload de screenshots (8-10 imagens)
- [] Upload de arquivos:
 - Windows .exe ou .zip
 - Linux AppImage
 - User Guide PDF
 - Quick Start PDF
 - Config templates
- [] Configurar tiers:
 - Free (link para GitHub)
 - Indie: \$19.99
 - Pro: \$49.99
 - Enterprise: \$199/ano
- [] Ativar código de desconto: LAUNCH25 (25% off)
- [] Configurar email pós-compra customizado

12:30 - 13:00 | Testar Fluxo de Compra

- [] Fazer compra teste (modo sandbox)
- [] Verificar email de confirmação
- [] Testar download do produto
- [] Verificar ativação de licença funciona
- [] Ajustar se necessário

TARDE (14h-18h) - 4 horas

14:00 - 15:00 | Setup Social Media

- [] Twitter/X:
 - Bio: "Building AI-powered ROM translation tools"
 - Header: Banner do produto

- Pin tweet: Anúncio de lançamento

- [] Reddit account preparado
- [] YouTube channel (se ainda não tem)
- [] GitHub repo público (free version)

15:00 - 16:30 | Preparar Conteúdo de Launch

- [] Rascunhar posts para Reddit:
 - r/Roms: "I built an AI ROM translator"
 - r/RomHacking: "[Tool] New GUI translator"
 - r/emulation: "ROM Translation Made Easy"
 - r/SideProject: "Launched my ROM tool"
- [] Preparar tweet thread (10 tweets):
 - Tweet 1: "I spent 6 months building..."
 - Tweet 2: "The problem: Old tools are CLI-only..."
 - Tweet 3: "My solution: Modern GUI with AI..."
 - Tweet 4-9: Features, demo GIF, screenshots
 - Tweet 10: "Available now: [link]"
- [] Rascunhar email para lista (se tiver)

16:30 - 18:00 | Preparar Press Kit

- [] Criar pasta: /press-kit/
 - Product_Sheet.pdf
 - Screenshots (high-res)
 - Logo (SVG + PNG)
 - Press_Release.md
 - Contact info
- [] Lista de contatos:
 - Gaming journalists
 - YouTubers (retro gaming)
 - Blogs (emulation, translation)
- [] Preparar email pitch (curto e direto)
- [] Git commit: "Day 6 complete - Marketing ready"

DIA 7 - LANÇAMENTO!

MANHÃ (9h-12h) - 3 horas

09:00 - 09:30 | Checklist Final

- ☐ Gumroad listing: público e ativo
- ☐ Todos os arquivos uploadados
- ☐ Códigos de desconto ativos
- ☐ Video demo: público no YouTube
- ☐ GitHub repo: público
- ☐ Redes sociais: prontas
- ☐ Support email: funcionando
- ☐ Licença: activation testada

09:30 - 10:00 | GO LIVE! 🦊

- ☐ Publicar produto no Gumroad
- ☐ Post em Reddit (todos os subs relevantes)
- ☐ Tweet thread
- ☐ Post em Discord servers
- ☐ Atualizar GitHub README com link de compra
- ☐ Email para lista (se tiver)

10:00 - 12:00 | Engajamento Ativo

- ☐ Responder TODOS os comentários no Reddit
- ☐ Responder mentions no Twitter
- ☐ Agradecer primeiros compradores
- ☐ Retweet feedback positivo
- ☐ Fix bugs urgentes se reportados

TARDE (14h-18h) - 4 horas

14:00 - 15:00 | Post Launch #2

- ☐ Cross-post em mais comunidades
- ☐ Post em Facebook groups
- ☐ Post em LinkedIn (se tiver)
- ☐ Compartilhar em Discord DMs (amigos)

15:00 - 17:00 | Análise Inicial

- ☐ Verificar analytics Gumroad
- ☐ Quantas visitas: _____
- ☐ Quantas vendas: _____
- ☐ Taxa de conversão: _____%

- ☐ Feedback dos primeiros usuários
- ☐ Bugs críticos reportados

17:00 - 18:00 | Ajustes Rápidos

- ☐ Fix bugs críticos (se houver)
- ☐ Atualizar FAQ baseado em perguntas
- ☐ Melhorar descrição se conversão baixa
- ☐ Agradecer primeiros 10 clientes pessoalmente

NOITE (19h-21h) - 2 horas

19:00 - 20:00 | Post em Product Hunt (opcional)

- ☐ Criar listing no ProductHunt
- ☐ Pedir upvotes de amigos (legalmente)
- ☐ Responder todos os comentários
- ☐ Objetivo: Top 10 do dia (mais visibilidade)

20:00 - 21:00 | Celebração & Reflexão 🥳

- ☐ Comemorar! Você lançou um produto! 🥂
- ☐ Anotar lições aprendidas
- ☐ Planejar próxima semana (suporte, v1.1)
- ☐ Dormir cedo (você merece) 😴

MÉTRICAS PARA ACOMPANHAR (Pós-Launch)

Diariamente (Primeira Semana):

- Vendas: _____
- Visitas ao Gumroad: _____
- Taxa de conversão: _____%
- Suporte tickets: _____
- Bugs reportados: _____
- Reviews/feedback: _____

Semanalmente:

- MRR (Monthly Recurring Revenue)
- CAC (Customer Acquisition Cost)
- LTV (Lifetime Value)
- Churn rate
- NPS (Net Promoter Score)

EXPECTATIVAS REALISTAS

Primeiras 24h:

Conservador: 5-10 vendas (\$100-\$200)
Realista: 10-20 vendas (\$200-\$500)
Otimista: 20-50 vendas (\$500-\$1,500)

Primeira Semana:

Conservador: 15-30 vendas (\$300-\$800)
Realista: 30-75 vendas (\$800-\$2,000)
Otimista: 75-150 vendas (\$2,000-\$5,000)

Primeiro Mês:

Conservador: 50-100 vendas (\$1,000-\$2,500)
Realista: 100-200 vendas (\$2,500-\$5,000)
Otimista: 200-500 vendas (\$5,000-\$15,000)

PLANO B (Se as vendas forem lentas)

Ações Imediatas:

- [] Reduzir preço temporariamente (50% off)
- [] Oferecer versão trial (7 dias grátis)
- [] Fazer giveaway (10 licenças grátis)
- [] Investir em ads (Reddit, Twitter - \$50-100)
- [] Pedir reviews honestas
- [] Criar conteúdo: tutoriais, case studies
- [] Parceria com influencers (enviar licenças)
- [] Refinar value proposition (testar copy diferente)

☑ CHECKLIST MESTRE

DIA 1: [] SNES validado
DIA 2: [] PS1 validado
DIA 3: [] UX polido
DIA 4: [] Builds prontos
DIA 5: [] Docs completas
DIA 6: [] Marketing pronto
DIA 7: [] LANÇADO! 🚀

BOA SORTE, FUTURO EMPREENDEDOR! 💰 🎮

Lembre-se: **Shipping > Perfection**

É melhor lançar um MVP (Minimum Viable Product) funcional do que esperar 6 meses pelo produto perfeito.

Você pode sempre lançar v1.1, v1.2, v2.0 depois! 🚀