

Arte ANSI: história, exemplos, ferramentas, repositórios e recursos

Pablo Murad

Abril de 2026

Sumário

Arte ANSI: história, exemplos, ferramentas, repositórios e recursos	1
Como usar este documento	1
História curta da arte ANSI	2
Conceitos que você precisa dominar	2
Exemplo visual mínimo	2
Exemplos para rodar e editar	2
PowerShell: bloco de cor ANSI mínimo	2
PowerShell: banner animado pequeno	3
CMD / Batch: texto colorido	3
Bash: rampa de 256 cores	3
Python + Colorama: bloco em movimento	3
Converter um .ANS para PNG	3
Roteiro de aprendizado	3
Recursos, links e repositórios	4
História, arquivos e preservação	4
Editores e ferramentas de criação	4
Visualizadores, conversores e formatos	5
Referências ANSI e programação de terminal	5
Bibliotecas de código e exemplos executáveis	5
Projetos de imagem, vídeo e conversão	6
Aplicativos de terminal e inspiração visual	7
Projetos iniciais que valem mais do que 100 abas abertas	7
Cuidados e bom senso	7
Fontes consultadas	7

Arte ANSI: história, exemplos, ferramentas, repositórios e recursos

Autor: Pablo Murad

Data: Abril de 2026

Um guia prático para aprender arte ANSI e *textmode art* editando, rodando, convertendo e publicando: história, exemplos, comandos, repositórios, arquivos, ferramentas e bibliotecas.

Como usar este documento

- Leia a história curta para entender o que você está estudando.
- Rode os exemplos pequenos antes de baixar dezenas de repositórios.
- Escolha uma ferramenta de desenho: Moebius, PabloDraw ou Durdraw.
- Baixe um artpack, abra uma arte .ANS, altere uma área pequena e exporte.

- Crie uma peça sua: logo, tela de menu, animação curta ou conversor.

Atalho brutal: se você tentar aprender tudo ao mesmo tempo, vira colecionador de links. Ordem recomendada: PowerShell ANSI -> Moebius/PabloDraw -> AnsiLove -> Durdraw -> uma biblioteca de código, como Rich, Asciiatics ou TerminalTextEffects -> seu próprio mini artpack.

História curta da arte ANSI

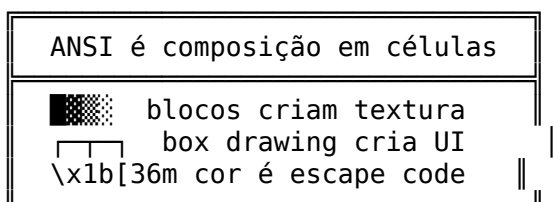
A arte ANSI nasceu de uma combinação de tecnologia e cultura: terminais que aceitavam sequências de escape, PCs com caracteres gráficos em Code Page 437, BBSes que precisavam de identidade visual e grupos de artistas competindo por estilo. O resultado não era apenas texto colorido; era design em células, com limite, disciplina e personalidade.

- **Controle de terminal:** sequências ANSI/VT movem cursor, limpam tela e aplicam cor.
- **IBM PC e CP437:** blocos, linhas, cantos e sombreamentos viraram o alfabeto visual da arte ANSI.
- **BBSes:** a arte aparecia em telas de login, menus, anúncios de boards, loaders e assinaturas.
- **Artpacks:** grupos como ACiD, iCE, Aces of ANSI Art, Mistigris, Fire e Blocktronics distribuíam coleções em ZIP/RAR.
- **Preservação:** Sixteen Colors, Artpacks.org, Defacto2, Textfiles e scene.org hoje são essenciais para estudar a cena.
- **Renascimento moderno:** a estética volta em terminal apps, demoscene, telnet BBSes, retro-computing, arte generativa e bibliotecas TUI.

Conceitos que você precisa dominar

- **ASCII art:** arte em texto simples, normalmente sem cor.
- **ANSI art:** arte textual com sequências ANSI para cor, estilo e cursor, geralmente em arquivos .ANS.
- **Textmode art:** termo guarda-chuva: ANSI, ASCII, BIN, XBin, Amiga ASCII, PETSCII, Teletext, Shift_JIS AA, Unicode/Braille art.
- **SGR:** *Select Graphic Rendition*, família de comandos como ESC[31m para vermelho e ESC[0m para reset.
- **Code Page 437:** conjunto de caracteres original do IBM PC, indispensável para ANSI clássico.
- **SAUCE:** metadados no fim do arquivo: título, autor, grupo, data, dimensões e formato.
- **iCE colors:** modo histórico que troca *blink* por 16 cores de fundo.

Exemplo visual mínimo



Exemplos para rodar e editar

PowerShell: bloco de cor ANSI mínimo

```
# PowerShell 7+ recomendado
$esc = "`e"
$reset = "$esc[0m"
$colors = 31,32,33,34,35,36,91,92,93,94,95,96
foreach ($c in $colors) {
    Write-Host "$esc[{$c}m█ Cor ANSI $c $reset" -NoNewLine
}
```

```
    Start-Sleep -Milliseconds 80
}
Write-Host ""
```

PowerShell: banner animado pequeno

```
$esc = "`e"
$frames = @( " ANSI ", " ANSI ", " ANSI ", " ANSI ")
for ($i = 0; $i -lt 40; $i++) {
    $frame = $frames[$i % $frames.Count]
    Write-Host "$esc[2J$esc[H$esc[96m$frame$esc[0m"
    Start-Sleep -Milliseconds 90
}
```

CMD / Batch: texto colorido

```
@echo off
REM Em Windows Terminal / Windows 10+, ANSI normalmente funciona.
REM Este truque cria o caractere ESC:
for /F "delims=" %%e in ('echo prompt $E ^| cmd') do set "ESC=%%e"
echo %ESC%[31mVermelho%ESC%[0m %ESC%[32mVerde%ESC%[0m %ESC%[36mCiano%ESC%[0m
```

Bash: rampa de 256 cores

```
#!/usr/bin/env bash
for i in {16..231}; do
    printf "\e[48;5;%sm %3s \e[0m" "$i" "$i"
    (( (i-15) % 12 == 0 )) && echo
done
```

Python + Colorama: bloco em movimento

```
from colorama import init
import time, sys
init()
ESC = "\033"
for x in range(1, 40):
    sys.stdout.write(f"{ESC}[2J{ESC}[{5}];{x}H{ESC}[96m█{ESC}[0m")
    sys.stdout.flush()
    time.sleep(0.03)
print()
```

Converter um .ANS para PNG

```
# Linux/macOS/WSL depois de instalar ansilove
ansilove arte.ans -o arte.png

# Ver/tocar no Durdraw
durdraw -p arte.ans
```

Roteiro de aprendizado

1. **Semana 1 - Terminal básico:** cor, reset, limpar tela, mover cursor, esconder cursor, loop de tempo.
2. **Semana 2 - CP437 e composição:** abra um .ANS no Moebius/PabloDraw, troque paleta, redesenhe uma caixa, assine.

3. **Semana 3 - Conversão:** use AnsiLove para PNG, ANSIFilter/ansi2html para HTML e VHS para GIF.
4. **Semana 4 - Produzindo:** crie logo, tela estilo BBS/menu e animação de 5 frames. Publique em um repositório.

Recursos, links e repositórios

História, arquivos e preservação

1. **ANSI art - Break Into Chat / BBS Wiki** - Visão geral bem focada no contexto BBS: CP437, ANSI.SYS, arquivos .ANS, cores e ANSImation.
2. **ANSI art - Wikipedia** - Referência ampla para terminologia, cena, artpacks, grupos e relação entre ASCII, ANSI, BBS e demoscene.
3. **Sixteen Colors / 16colo.rs** - Arquivo público essencial de artpacks ANSI e ASCII. Permite navegar por ano, grupo, artista, tags e tipo de arquivo.
4. **Artpacks.org** - Arquivo de packs BBS, ANSI, ASCII e artscene desde o início dos anos 1990.
5. **ACiD Productions - official information** - Núcleo histórico de uma das organizações mais influentes da cena ANSI/artscene. Bom para rastrear ferramentas, padrões e arquivos da ACiD.
6. **ACiD Productions on Sixteen Colors** - Página do grupo com linha do tempo, contagens, tipos de arquivo e artes da ACiD para navegar.
7. **Aces of ANSI Art - Defacto2** - Coleção histórica da Aces of ANSI Art, um dos primeiros grupos organizados da cena BBS e precursor importante.
8. **Defacto2 - text art scene links** - Porta de entrada curada para arquivos da text art scene como 16colors, Blocktronics, Art Scene/Textfiles e recursos de BBS.
9. **scene.org artpacks directory** - Arquivo de packs relacionados a cena. Bom para baixar material histórico e comparar como os packs eram distribuídos.
10. **Binary Passion - textmode art resources** - Página pequena, mas bem curada, com recursos de textmode art, arquivos, ferramentas e cultura relacionada.
11. **ASCII Art Archive** - Grande arquivo ASCII. Não é ANSI puro, mas é excelente para estudar forma, silhueta, tipografia e textura de caracteres.
12. **Mozz ASCII / Textmode archive** - Coleção e espelho de ASCII, ANSI de BBS, ANSI chinês de BBS e arte textual japonesa em Shift_JIS.
13. **Textfiles.com Art Scene** - Arquivo clássico de preservação da art scene, com packs, arquivos e material histórico.

Editores e ferramentas de criação

1. **Moebius** - Editor moderno de arte ANSI e ASCII para macOS, Linux e Windows. Use como editor principal para trabalho sério em .ANS.
2. **PabloDraw** - Editor/visualizador ANSI/ASCII e RIPscrip com recursos multiusuário. Historicamente importante e ainda útil para aprender o fluxo tradicional.
3. **Durdraw** - Editor de terminal para ASCII, Unicode e ANSI em sistemas Unix-like, com animação por frames, 16/256 cores, conversão CP437 e playback.
4. **Textual Paint** - Editor TUI inspirado no MS Paint que edita arte ANSI e imagens raster. Ótimo para estudar padrões de edição em terminal.
5. **Sixteen Colors Draw** - Experimento de editor ANSI no navegador ligado ao ecossistema Sixteen Colors. Útil quando você quer desenhar sem instalar app desktop.
6. **ASCII Art Paint** - Editor e conversor de gráficos ASCII que roda no navegador e pode funcionar offline. Melhor para experimentos ASCII/poster do que ANSI BBS clássico.

7. **MonoSketch** - Aplicativo para desenho de grafos ASCII. Bom para treinar disciplina de layout em texto.

Visualizadores, conversores e formatos

1. **AnsiLove/C** - Conversor em C de arte ANSI e ASCII para PNG. Suporta ANS, PCB, BIN, ADF, IDF, TND, XB, SAUCE, fontes DOS/Amiga e iCE colors.
2. **libansilove** - Versao biblioteca do AnsiLove para embutir renderizacao ANSI/artscene nas suas proprias ferramentas.
3. **ANSIFilter** - Utilitario para remover ou interpretar codigos ANSI e converter saida ANSI para HTML, RTF, TeX, LaTeX, BBCode e Pango markup.
4. **ansi2html.py** - Pequena ferramenta Python para converter saida colorida ANSI/VT100 em HTML.
5. **Fancy ANSI** - Biblioteca JavaScript para converter sequencias ANSI em HTML seguro para navegador, com variaveis CSS customizaveis.
6. **SAUCE metadata specification** - Formato de metadados usado em arquivos de textmode art para guardar titulo, autor, grupo, data, dimensoes e informacoes de formato.
7. **XBin format specification** - Especificacao de imagens textmode eXtended Binary: dimensoes, paleta, fonte e compressao alem de ANSI/BIN simples.
8. **Code Page 437 reference** - Conjunto de caracteres essencial para arte textmode IBM PC/MS-DOS: box drawing, sombreamentos, simbolos e glifos legados.
9. **Box-drawing characters** - Referencia para linhas, cantos, blocos e sombreamentos usados tanto em ANSI classico quanto em arte terminal Unicode moderna.

Referências ANSI e programação de terminal

1. **PowerShell about_ANSI_Terminals** - Referencia oficial do PowerShell para renderizacao ANSI, \$PSStyle, suporte de terminal e comportamento no PowerShell moderno.
2. **Windows Console Virtual Terminal Sequences** - Referencia oficial da Microsoft para movimento de cursor, cores, SGR e ativacao de processamento VT em apps de console Windows.
3. **VT100.net technical manual** - Referencia historica de controle de terminal. Ensina o modelo mental por tras de movimento de cursor e sequencias de controle/escape.
4. **Bash Hackers Wiki - terminal codes** - Introducao pratica a codigos de controle ANSI/VT100, tput, posicionamento de cursor e trabalho seguro no shell.
5. **ANSI escape code reference** - Referencia para cores SGR, modos 8/16/256 cores, truecolor, movimento de cursor e vocabulario de controle terminal.
6. **GitHub topic: ascii-art** - Pagina viva para descobrir repositorios ativos de ASCII/terminal art em Python, Go, JavaScript, C++, Rust e outros.
7. **GitHub topic: ascii-art-generator** - Bom ponto de busca para conversores imagem-para-texto, editores e experimentos.

Bibliotecas de código e exemplos executáveis

1. **Rich** - Biblioteca Python para texto rico no terminal, syntax highlight, tabelas, Markdown, barras de progresso, paineis e estilo ANSI.
2. **Colorama** - Camada Python de compatibilidade que faz sequencias ANSI funcionarem melhor em terminais Windows.
3. **Termcolor** - Biblioteca Python simples para saida colorida no terminal. Boa para primeiros experimentos antes de Rich ou Textual.

4. **Asciiatics** - Toolkit Python para TUIs fullscreen e animacoes ASCII com cores, cursor, sprites, efeitos e widgets.
5. **TerminalTextEffects** - Engine Python de efeitos visuais no terminal com cores RGB/256, movimento de caracteres, gradientes, cenas e caminhos de animacao.
6. **ascii-art for Node.js** - Biblioteca Node.js para codigos ANSI, fontes FIGlet, ASCII art, cores e graficos de terminal.
7. **cfonts** - Fontes coloridas para console no ecossistema Node/Rust. Boa para logos, telas de abertura e identidade visual CLI.
8. **picocolors** - Formatador ANSI minimo em JavaScript. Bom quando voce quer cor no terminal com quase nenhum peso de dependencia.
9. **kleur** - Biblioteca Node.js rapida para formatacao ANSI no terminal, com estilo encadeado e aninhado.
10. **Pterm** - Biblioteca Go para saida terminal bonita: cores, texto grande, caixas, graficos, heatmaps, spinners, prompts e temas.
11. **fatih/color** - Pacote Go popular para saida colorida no terminal. Util para pequenos CLIs e aprendizado de formatacao ANSI em Go.
12. **Spectre.Console** - Biblioteca .NET para apps de console bonitos e multiplataforma, com markup, paineis, tabelas, grids, progresso e estilos.
13. **rust-ansi-term** - Biblioteca Rust para cores ANSI e estilos de texto. Boa para aprender primitivas de formatacao terminal.
14. **Crossterm** - Biblioteca Rust para manipulacao de terminal multiplataforma: cursor, input, cores, raw mode e controle terminal.
15. **blessed-contrib** - Dashboards JavaScript de terminal usando ASCII/ANSI art, graficos, mapas, widgets e layouts dinamicos.
16. **Charmbracelet Lip Gloss** - Biblioteca Go de estilo para layouts de terminal, bordas, padding, cores e composicao TUI responsiva.
17. **Charmbracelet Gum** - Componentes TUI prontos para scripts shell: caixas estilizadas, prompts, filtros, spinners, tabelas e renderizacao Markdown.
18. **Charmbracelet VHS** - Transforma sessoes de terminal roteirizadas em GIF/video/texto. Perfeito para gravar demos ANSI e experimentos de terminal art.

Projetos de imagem, vídeo e conversão

1. **ascii-image-convert** - CLI multiplataforma para converter imagens e GIFs em ASCII ou Braille art, com saida colorida opcional.
2. **tim** - Visualizador de imagem/video no terminal com Sixel, Kitty, iTerm2 ou fallback em blocos Unicode e cor 24-bit.
3. **tplay** - Player de midia ASCII no terminal para imagens, GIFs, videos, webcam e fontes do YouTube.
4. **video-to-ascii** - Pacote Python para tocar videos no terminal usando caracteres como pixels.
5. **image-to-ascii** - Modulo Node.js para converter imagens em ASCII art. Bom para experimentos de imagem em CLI.
6. **ASCII-generator** - Projeto Python/OpenCV para converter imagens e video em representacoes ASCII.
7. **figlet-fonts** - Grande colecao de fontes FIGlet/TOIlet para tipografia de terminal e banners.
8. **FIGfonts by Philippe Majerus** - Colecao de FIGfonts que experimenta com cores ANSI/VT e caracteres Unicode estendidos.

9. **pyfiglet** - Implementação Python do FIGlet. Util para gerar banners dentro de scripts Python.

Aplicativos de terminal e inspiração visual

1. **CMatrix** - Efeito Matrix rain clássico em C/ncurses. Estude temporização, colunas, fluxos de glifos e repaint do terminal.
2. **cxxmatrix** - Brinquedo visual C++ de terminal com Matrix rain, Game of Life, Mandelbrot, banners e troca de cenas.
3. **TMatrix** - Implementação moderna de Matrix rain em Rust/terminal. Boa para estudar performance e configuração.
4. **MapSCII** - Mapa mundial em Braille e ASCII no console. Excelente exemplo de visualização espacial em texto.
5. **wtrr.in** - Previsão do tempo no terminal com saída em estilo ASCII/ANSI. Estude apresentação terminal prática e saída amigável para curl.
6. **ansiweather** - Script shell para clima com cores ANSI e símbolos Unicode. Bom exemplo compacto de scripting.
7. **astroterm** - Planetário renderizado no terminal. Estude renderização de cena, movimento e dados astronômicos em interface textual.
8. **zero** - Renderizador 3D em JavaScript exibido no terminal. Boa ponte entre matemática gráfica e saída textual.
9. **FTXUI** - Biblioteca C++ para TUI. Não é arte ANSI em si, mas é excelente para criar apps visuais interativos no terminal.
10. **Diagon** - Gerador interativo de diagramas ASCII para grafos, árvores, tabelas e diagramas técnicos.

Projetos iniciais que valem mais do que 100 abas abertas

1. Um banner PowerShell de 16 cores que anima por 10 segundos.
2. Uma tela .ANS de 80 colunas com moldura, logo e assinatura.
3. Um PNG gerado com AnsiLove a partir da sua tela .ANS.
4. Uma animação Python/Colorama com cursor e blocos.
5. Um GIF gravado com VHS mostrando seu terminal rodando a animação.

Cuidados e bom senso

- Não rode .EXE, .COM ou binários antigos fora de DOSBox, VM ou sandbox.
- Prefira código-fonte, .ANS e .ASC para estudo.
- Cheque licença antes de reutilizar arte. Preservação não é permissão comercial automática.
- Use fonte monospace com suporte a CP437/box drawing.
- Finalize scripts com reset ESC[0m. Terminal possuído é divertido uma vez; depois vira perda de tempo.

Fontes consultadas

BBS Wiki, Sixteen Colors, Microsoft Learn, VT100.net, GitHub, Artpacks.org, Defacto2, AnsiLove, Moebius, PabloDraw, Durdraw e demais repositórios listados neste documento.