

HOME - PROTOTIPO INTERATIVO

Danilo Novais Portfolio Homepage

Technical Documentation & Design Specifications

Technical Documentation & Design Specifications

Version: 3.0

Last Updated: January 03, 2026

Status:  Ready for Implementation

1. PROJECT OVERVIEW

1.1 Vision & Goals

Create a premium institutional portfolio that demonstrates design excellence not just through showcased work, but through the site's own execution. The homepage must:

- Establish immediate credibility through atmospheric visual design and editorial sophistication
- Guide visitors intuitively from brand introduction → work showcase → contact
- Feel distinctive and memorable without sacrificing usability or accessibility
- Perform flawlessly across all devices and connection speeds
- Reflect the designer's philosophy: "You don't see design. But it sees you."

Success looks like:

- Visitors spending 2+ minutes exploring the portfolio
- High engagement with featured projects (50%+ click-through)
- Contact form conversions from qualified leads
- Zero accessibility violations (WCAG AA minimum)
- Lighthouse scores: Performance 90+, Accessibility 100, Best Practices 100

1.2 Target Audience

Primary:

- Brand managers and marketing directors at mid-to-large companies seeking design partners
- Creative agencies looking for freelance brand designers or collaborators
- Startups and scale-ups needing brand identity and campaign work

Secondary:

- Design recruiters and HR professionals
- Fellow designers and creative community (peer recognition)
- Potential collaborators for interdisciplinary projects

User Needs:

- Quickly understand what Danilo does and his areas of expertise
- See evidence of high-quality work across branding, campaigns, and digital
- Assess cultural fit and working style
- Easily initiate contact

1.3 Key Success Metrics

Engagement: Average session duration, scroll depth, interaction rate

Conversion: Contact form submissions, portfolio page visits

Technical: Page load time <3s, Core Web Vitals passing, 0 console errors

Accessibility: WCAG AA compliance, keyboard navigation coverage

1.4 Technical Constraints

- No localStorage/sessionStorage in artifacts (Claude.ai environment restriction)
- Self-hosted assets where possible to avoid external dependencies
- Graceful degradation for WebGL/3D features (fallback to static alternatives)
- Mobile-first responsive design (majority of traffic expected from mobile)
- Performance budget: <2MB initial page weight, <5s time to interactive

2. DESIGN SYSTEM

2.1 Color Palette

Token	Value	Uso
bluePrimary	`#0048ff`	Cor primária da marca, CTAs, links, elementos interativos
blueAccent	`#4fe6ff`	Destaques secundários, brilhos "ghost"/atmosfera
purpleDetails	`#8705f2`	Pequenos detalhes e highlights
pinkDetails	`#f501d3`	Pequenos detalhes, ênfases pontuais
background	`#040013`	Fundo escuro principal
backgroundLight	`#f0f0f0`	Seções claras (forms, blocos alternados)
text	`#fcffff`	Texto principal em fundo escuro
textInverse	`#0e0e0e`	Texto em fundos claros
textEmphasis	`#2E85F2`	Palavras destacadas no meio do texto
textHighlight	`#4fe6ff`	Destaques curtos, intros breves
textSecondary	`#a1a3a3`	Infos secundárias, metadata
neutral	`#0b0d3a`	Gradientes, fundos sutis
neutralLight	`#F5F5F5`	Fundos de seções secundárias

> Obs: `textEmphasis` estava com `#2E85F2` e `textHilight` com typo – normalizei para `textHighlight`.

2.2 Typography

Fonte primária: TT Norms Pro (self-hosted, fallback: `ui-sans-serif, system-ui`)

Tokens de texto **responsivos** (usando `clamp`) para manter coerência em todos os breakpoints:

Token	Mobile (~<640px)	Desktop (~≥1024px)	Peso	Uso
display	2.5rem (40px)	4.5rem (72px)	Black	Frases grandes no meio da página, não-semânticas (Big Phrase)
h1	2rem (32px)	3.5rem (56px)	Bold	Hero headlines, títulos principais
h2	1.5rem (24px)	2.5rem (40px)	Bold	Títulos de seção
h3	1.25rem (20px)	1.75rem (28px)	Medium	Títulos de cards, subtítulos
body	1rem (16px)	1.125rem (18px)	Regular	Texto corrido
small	0.875rem (14px)	0.875rem (14px)	Reg/Med	Labels, legendas
micro	0.75rem (12px)	0.75rem (12px)	Mono	Tags, infos de sistema

Tokens em CSS com `clamp()`

```
[ 'css
:root {
  --font-display: clamp(2.5rem, 5vw, 4.5rem);
  --font-h1:      clamp(2rem, 4vw, 3.5rem);
  --font-h2:      clamp(1.5rem, 3vw, 2.5rem);
  --font-h3:      clamp(1.25rem, 2vw, 1.75rem);
  --font-body:    clamp(1rem, 1.2vw, 1.125rem);
  --font-small:   0.875rem;
  --font-micro:   0.75rem;
}

body {
  font-family: "TT Norms Pro", ui-sans-serif, system-ui, -apple-system,
  BlinkMacSystemFont,
  "Segoe UI", sans-serif;
}

.display-text {
  font-size: var(--font-display);
  font-weight: 700;
  line-height: 1.1;
}
```

```
.h1 {
  font-size: var(--font-h1);
  font-weight: 700;
  line-height: 1.1;
}

.h2 {
  font-size: var(--font-h2);
  font-weight: 700;
  line-height: 1.15;
}

.h3 {
  font-size: var(--font-h3);
  font-weight: 500;
  line-height: 1.2;
}

body {
  font-size: var(--font-body);
  font-weight: 400;
  line-height: 1.5;
}

.small {
  font-size: var(--font-small);
}

.micro {
  font-size: var(--font-micro);
  font-family: ui-monospace, SFMono-Regular, Menlo, Monaco, Consolas,
    "Liberation Mono", "Courier New",
    monospace;
}
```

Versão conceitual em Tailwind

```
// tailwind.config.js
module.exports = {
  theme: {
    extend: {
      fontFamily: {
        sans: ['"TT Norms Pro"', 'ui-sans-serif', 'system-ui'],
      },
      fontSize: {
        display: [
          'clamp(2.5rem, 5vw, 4.5rem)',
          { lineHeight: '1.1', fontWeight: '700' },
        ],
        h1: [
          'clamp(2rem, 4vw, 3.5rem)',
          { lineHeight: '1.1', fontWeight: '700' },
        ],
        h2: [
          'clamp(1.5rem, 3vw, 2.5rem)',
```

```
        { lineHeight: "1.15", fontWeight: "700" },
    ],
h3: [
    "clamp(1.25rem, 2vw, 1.75rem)",
    { lineHeight: "1.2", fontWeight: "500" },
],
body: [
    "clamp(1rem, 1.2vw, 1.125rem)",
    { lineHeight: "1.5", fontWeight: "400" },
],
small: ["0.875rem", { lineHeight: "1.4" }],
micro: ["0.75rem", { lineHeight: "1.4" }],
},
},
},
},
}];'
```

2.3 Spacing & Grid

Container

- max-width: 1680px
 - Padding horizontal: clamp(24px, 5vw, 96px)

Ritmo Vertical

- Seções: py-16 md:py-24
 - Componentes: gap-8 md:gap-12
 - Elementos internos: qap-4 md:qap-6

Grid (Tailwind)

- Mobile (até md):
 - Layout: 1 coluna (grid-cols-1 ou flex flex-col)
 - w-full
 - Alinhamento:
 - text-center para todos os textos
 - items-center e justify-center para stacks verticais (flex-col)
 - Tablet (md:):
 - Cards em md:grid-cols-2
 - Hero / destaques podem continuar 1 coluna
 - Textos podem voltar a text-left se fizer sentido
 - Desktop (lg:+):
 - Distribuição customizada por seção
 - Textos geralmente alinhados à esquerda para leitura longa

Regra de alinhamento para mobile (base do sistema):

Breakpoint padrão: < 768px

Regra:

- Todos os títulos (display, h1, h2, h3), parágrafos e CTAs usam text-align: center.
 - Componentes em coluna usam align-items: center.
 - Imagens e ícones principais centralizados (margin-inline: auto).

Exemplo padrão de seção:

```
<section className="flex flex-col items-center text-center md:items-start  
md:text-left">  
  {/* conteúdo */}  
</section>
```

2.4 Animation Principles

Filosofia: animações orgânicas e intencionais, nunca gratuitas.

Core Library: Framer Motion

Diretrizes:

- Animar apenas transform e opacity (performance)
- Easing: cubic-bezier(0.22, 1, 0.36, 1) (easeOutExpo)
- Duração: 300–700ms na maioria das transições
- Stagger: 60–120ms entre elementos sequenciais
- Respeitar prefers-reduced-motion: desabilitar animações não essenciais

Padrões comuns:

```
// Scroll reveal  
<motion.div  
  initial={{ opacity: 0, y: 24 }}  
  whileInView={{ opacity: 1, y: 0 }}  
  transition={{ duration: 0.6, ease: [0.22, 1, 0.36, 1] }}  
>  
  
// Hover (botões, cards)  
<motion.button  
  whileHover={{ scale: 1.02, y: -2 }}  
  transition={{ duration: 0.3 }}  
>  
  
// Staggered children  
const variants = {  
  hidden: { opacity: 0 },  
  visible: {  
    opacity: 1,  
    transition: { staggerChildren: 0.08 },  
  },  
};
```

Em mobile, como tudo é centralizado e o fluxo é vertical, as entradas preferenciais vêm de baixo (y: 24 → 0) acompanhando o scroll.

2.5 Display Text / Big Phrases (Frases em destaque)

Frases grandes no meio da página, com grande destaque visual, mas sem função de título semântico.

Token: display

Diretrizes de uso:

- Quando usar:
- Frases de impacto, statements da marca, quotes, promessas fortes de seção.
- Semântica:
- Usar como `<p>` ou `` com classe específica:
- `className="display-text"` ou `className="text-display"`
- Exemplo:

```
<p className="text-display">  
  Construímos experiências digitais que parecem magia, mas são guiadas por  
  dados.  
</p>
```

- Alinhamento:
- Mobile: sempre centralizado, com largura limitada:
- Ex.: `className="text-display max-w-2xl mx-auto text-center"`
- Desktop: pode ser centralizado ou seguir a grid da seção (recomendado manter centralizado em blocos de destaque).
- Espaçamento:
- Mais respiro que títulos normais:
- Ex.: `mt-16 mb-12` (ajustar conforme a seção).
- Cores:
- Base: `text (#fcffff)`
- Palavras-chave com `textEmphasis` e `textHighlight`.

Exemplo em JSX/Tailwind:

```
<section className="py-16 flex flex-col items-center text-center">  
  <p className="text-display max-w-2xl mx-auto">  
    Criamos produtos que parecem  
    <span className="text-textHighlight"> magia</span>, mas são construídos  
    com  
    <span className="textEmphasis"> engenharia séria</span>.  
  </p>  
</section>
```

2.6 Global Assets

Logos:

- Favicon - Usar no browser:
`'https://aymuvxysygrwoicsjgxj.supabase
.co/storage/v1/object/public/logo_site/Favicon.svg'`
- Favicon Light - Usar no browser:
`'https://aymuvxysygrwoicsjgxj.supabase
.co/storage/v1/object/public/logo_site/FaviconLight.svg'`

- Logo Light (full - Usar no header):
`https://aymuvxysygrwoiccsjgxj.supabase.co/storage/v1/object/public/logo_site/LogoLight.svg`
- Logo Dark (full - Usar no header):
`https://aymuvxysygrwoiccsjgxj.supabase.co/storage/v1/object/public/logo_site/LogoDark.svg`

2.7 Fonts:

- font-family: 'TT Norms Pro';
src:
 url('https://aymuvxysygrwoiccsjgxj.supabase.co/storage/v1/object/public/assets/fonts/TT%20Norms%20Pro%20Thin.woff2') format('woff2');
font-weight: 100;
font-style: normal;
font-display: swap;
- font-face {
font-family: 'TT Norms Pro';
src:
 url('https://aymuvxysygrwoiccsjgxj.supabase.co/storage/v1/object/public/assets/fonts/TT%20Norms%20Pro%20Light.woff2') format('woff2');
font-weight: 300;
font-style: normal;
font-display: swap;
- font-face {
font-family: 'TT Norms Pro';
src:
 url('https://aymuvxysygrwoiccsjgxj.supabase.co/storage/v1/object/public/assets/fonts/TT%20Norms%20Pro%20Regular.woff2') format('woff2');
font-weight: 400;
font-style: normal;
font-display: swap;
- font-face {
font-family: 'TT Norms Pro';
src:
 url('https://aymuvxysygrwoiccsjgxj.supabase.co/storage/v1/object/public/assets/fonts/TT%20Norms%20Pro%20Medium.woff2') format('woff2');
font-weight: 500;
font-style: normal;
font-display: swap;
- font-face {
font-family: 'TT Norms Pro';
src:
 url('https://aymuvxysygrwoiccsjgxj.supabase.co/storage/v1/object/public/assets/fonts/TT%20Norms%20Pro%20Bold.woff2') format('woff2');
font-weight: 700;

```

font-style: normal;
font-display: swap;

- font-face {
  font-family: 'TT Norms Pro';
  src:
    url('https://aymuvxysygrwoicsjgxj.supabase
.co/storage/v1/object/public/assets/fonts/TT%20Norms%20Pro%20Black
.woff2') format('woff2');
  font-weight: 900;
  font-style: normal;
  font-display: swap;
}

/* Fonte Mono para Tags */
@font-face {
  font-family: 'PPSupplyMono';
  src: url('https://assets.codepen.io/7558/PPSupplyMono-Variation.woff2')
    format('woff2');
  font-weight: 100 900;
  font-style: normal;
  font-display: swap;
}

```

Videos:

- Manifesto Video:
`<https://aymuvxysygrwoicsjgxj.supabase
.co/storage/v1/object/public/project-videos/VIDEO-APRESENTACAO-PORTFOLIO
.mp4>`

Client Logos:

- 12 monochromatic SVG logos: `client1.svg` through `client12.svg`
- Base URL:
`<https://aymuvxysygrwoicsjgxj.supabase
.co/storage/v1/object/public/client-logos/>`

3. SITE ARCHITECTURE

3.1 Information Architecture

Homepage

- Header (persistent navigation)
- Hero + Manifesto Video
- Portfolio Showcase (3 categories)
- Featured Projects (4 highlighted works)
- Clients/Brands (logo grid)
- Contact (form + info)
- Footer

Other Pages (linked from homepage)

- /sobre (About page)
- /portfolio (Full portfolio with filters)
- /portfolio/[slug] (Individual project pages)

3.2 Navigation Structure

Primary Navigation (Header):

- Home → `/` or `#hero`
- Sobre (About) → `/sobre`
- Portfólio → `/portfolio`
- Contato (Contact) → `#contact` (always anchors to contact section)

Secondary Navigation (Footer):

- Same as primary navigation
- Additional: Social media links

3.3 Section Flow

The homepage follows a narrative arc:

1. Header: Establishes brand identity and provides navigation
2. Hero: Creates emotional impact and communicates positioning
3. Manifesto Video: Deepens understanding through motion storytelling
4. Portfolio Showcase: Introduces work categories with editorial rhythm
5. Featured Projects: Demonstrates quality through curated examples
6. Clients/Brands: Builds credibility through association
7. Contact: Provides clear conversion path
8. Footer: Reinforces brand and provides supplementary navigation

**4. COMPONENT SPECIFICATIONS

4.1 Header

Purpose: Provide persistent, accessible navigation using a clean, static interface.

Desktop (≥1024px): Static Glass Header

Layout:

- Position: `position: sticky`, `top: 24px` (or `top: 0` with padding), `z-index: 40`
- Width: Partial container (centered), or Full-width (based on preference, images suggest clean alignment)
- Height: `64px`
- Style: Pill-shaped or Bar, translucent glass effect (`backdrop-filter: blur(12px)`), subtle border.
- **Behavior:** COMPLETELY STATIC. No movement tracking, no physics.

Content:

- Left: Logo (SVG) -
https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/logo_site/FaviconLight.svg
- Right: Navigation Links (Home, Sobre, Portfólio, Contato)

Interaction – Contrast Adaptation (Scroll):

Essential for visibility on the white 'Contato' section shown in designs.

- **Trigger:** When the header overlaps a light-colored background (e.g., the white Contact section).
 - **Action:**
 - Text/Icon color switches to **Blue** (Primary Brand Color).
 - Background may increase opacity for readability.
 - **Transition:** `transition: all 0.3s ease`.
-

****Desktop ($\geq 1024px$): Fluid Glass Header****

- Layout:
 - Position: `position: sticky`, `top: 0`, `z-index: 40`
 - Width: Partial container (not full-width), horizontally centered
 - Height: `56–72px`
 - Style: Pill-shaped with rounded corners, translucent glass effect (blur + subtle gradient)
- Content:
 - Left: Logo (Favicon Light)

https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/logo_site/FaviconLight.svg
 - Right: Horizontal navigation list (Home, Sobre, Portfólio, Contato)
- Interaction - Fluid Glass Effect:
 - The header responds to cursor movement along the X-axis:
 - Follow behavior: Subtle horizontal translation (`maxTranslateX: 40–60px`)
 - Spring physics: `followDamping: 0.08–0.12s`, gentle overshoot
 - Scale: Slight horizontal stretch (`maxScaleX: 1.05`) and vertical compression (`maxScaleY: 1.02`)
 - Visual: Refraction distortion, chromatic aberration on edges, backdrop blur
- Fallback (no WebGL / reduced motion):
 - Same layout and typography
 - Static positioning with standard backdrop-filter blur or solid/gradient background
 - No cursor-following behavior

****Mobile & Tablet ($\leq 1023px$): Staggered Menu Header****

- Layout:
 - Position: Fixed bar at top, full-width
 - Height: `48–64px`
 - Content: Logo (left), Hamburger icon (right)
- Menu Overlay (when open):
 - Fullscreen or near fullscreen overlay
 - Gradient background (primary to neutral)
 - Navigation items in vertical column, large text, generous spacing
 - Social media icons at bottom
- Animation:
 - Open: Overlay fades in (`opacity: 0 → 1`, 200–250ms), panel slides in from right (`translateX: 100% → 0`, 260–320ms), hamburger morphs to X
 - Items appear with stagger: `opacity: 0 → 1`, `translateY: 16px → 0`, `staggerDelay: 100ms`
 - Close: Reverse sequence, items disappear in reverse order
- Interactions:
 - Tap X icon: Close menu
 - Tap navigation item: Close menu + navigate

- Tap overlay background: Close menu
 - Accessibility:
 - `aria-label` on hamburger/X icon
 - `aria-expanded` state
 - Focus trap within open menu
 - ESC key closes menu
 - Mobile-Specific Implementation Notes:
 - The header does not have the glass effect on mobile
 - The logo is positioned flush left with no additional spacing
 - The hamburger menu icon is positioned flush right with 16px padding
 - When open, the menu covers the entire viewport
-

**4.2 Hero + Manifesto

✨ Objetivo

- Quero trazer um impacto na primeira impressão para quem entra na minha pagina, gerar curiosidade. Criar uma experiência hero imersiva e responsiva, com atmosfera 3D, manifesto em vídeo e animação de entrada impactante. A sessão além da animação 3D (Inspirado em: CodePen: <https://codepen.io/danilonovaisv/pen/YPWyrdW>) com a frase de impacto contará um video manifesto com resumo poético do meu trabalho (Animação scroll do video baseada na referencia do site ('<https://loandbehold.studio/>')).
- A ação principal na sessão é CTA que leva para sessão SOBRE e/ou entrar em contato.

🎨 2. Identidade Visual

- Color Palette:

Token	Value	Uso
bluePrimary	`#0048ff`	Cor primária da marca, CTAs, links, elementos interativos
background	`#040013`	Fundo escuro principal
text	`#fcffff`	Texto principal em fundo escuro

- Typography:

Fonte primária: TT Norms Pro (self-hosted, fallback: `ui-sans-serif, system-ui`)

Tokens de texto **responsivos** (usando `clamp`) para manter coerência em todos os breakpoints:

Token	Mobile (~<640px)	Desktop (~≥1024px)	Peso	Uso
-------	------------------	--------------------	------	-----

display	2.5rem (40px)	4.5rem (72px)	Black	Frases grandes no meio da página, não-semânticas (Big Phrase)
- src:	url('https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/assets/fonts/TT%20Norms%20Pro%20Black.woff2') format('woff2');			
h1	2rem (32px)	3.5rem (56px)	Bold	Hero headlines, títulos principais
- src:	url('https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/assets/fonts/TT%20Norms%20Pro%20Bold.woff2') format('woff2');			
h2	1.5rem (24px)	2.5rem (40px)	Bold	Títulos de seção
- src:	url('https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/assets/fonts/TT%20Norms%20Pro%20Bold.woff2') format('woff2');			
h3	1.25rem (20px)	1.75rem (28px)	Medium	Títulos de cards, subtítulos
- src:	url('https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/assets/fonts/TT%20Norms%20Pro%20Medium.woff2') format('woff2');			
body	1rem (16px)	1.125rem (18px)	Regular	Texto corrido
- src:	url('https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/assets/fonts/TT%20Norms%20Pro%20Regular.woff2') format('woff2');			

📝 **3. Conteúdo da Hero**

Conteúdo:

- Tag: `'[BRAND AWARENESS]` (mono, 19px)
- Display: "Você não vê / o design." (2 e 3 linhas)
- H2: "Mas ele vê você." (destacado)

CTA: "step inside →" (link: `/sobre`, hover animado)

Design Visual:

- **Formato:** Compósito (Pílula à esquerda + Círculo à direita).
- **Cor:** Azul Primário (`#0048ff`). Texto Branco.
- **Texto:** Uppercase, tracking médio, padding `px-6 py-3`.
- **Ícone:** Seta (→) centralizada no círculo.

Estilo:

- Centralizado verticalmente
- Cor: `#d9dade`

🎥 4. Animações e Interações

🎬 Entrada TExtos (Page Load)

```
```js
initial: {
 opacity: 0,
 scale: 0.92,
 translateY: 60,
 filter: "blur(10px)"
},
animate: {
 opacity: 1,
 scale: [1.02, 1],
 translateY: 0,
 filter: "blur(0px)"
},
duration: 1.2s,
easing: [0.25, 0.46, 0.45, 0.94]
```

```

CTA - Interações e Animações:

1. **Hover (Desktop):** O botão inteiro sobe 1px (`translateY(-1px)`).
2. **Seta (Opcional):** Desliza 4px para a direita no hover.
3. **Click (Mobile):** Efeito de compressão (`scale(0.98)`).
4. **Foco (Teclado):** Outline de 2px sólido cor `#4fe6ff` com offset de 4px.

- [] Qual tipo de animação deseja usar? (Framer Motion: fade-in, slide, spring?; Three.js para fundo animado?)
- [] Efeitos ao scroll ou load?
- [] Interação no hover dos CTAs?

🖼 5. Imagens / Elementos Visuais

Animação Ghost - Elementos visuais:

```
### Background / atmosfera
aspecto	observação
**Cores e textura**	O fundo da hero é um **gradiente bem escuro** (`#0a0a0a` → `#1a1a1a`) aplicado ao `body` e ao `preloader`. Após a transição, o canvas do WebGL é transparente e um plano em Three.js (Mesh com `PlaneGeometry(300,300)`) recebe um **shader próprio** chamado `_atmosphere_. Esse shader controla a opacidade e cria um **halo circular** em torno do fantasma. Ele usa `revealRadius`, `fadeStrength`, `baseOpacity` e `revealOpacity` para revelar gradualmente a região perto do personagem. Esse fundo não é uma imagem, mas sim um **material customizado** em Three.js.
```

| ****Efeito de película analógica**** | Após o render da cena, um **passo de pós-processamento** (ShaderPass) aplica `_grain_, _bleeding_, _scanlines_, _vignette_ e jitter` usando uniforms (``uAnalogGrain``, ``uAnalogBleeding``, ``uAnalogVSync``, etc.), criando um visual vintage de monitor. Esse shader é opcional e pode ser parametrizado via GUI. |

| ****Preloader**** | Antes de exibir a cena 3D, um preloader em HTML/CSS ocupa toda a tela. Ele usa um **SVG de fantasma em 2D** que flutua e pisca (animações `@keyframes ghostFloat` e `eyePulse`). Uma barra de progresso (``.progress-bar``) anima enquanto os assets são carregados. |

Fantasma e objetos flutuantes
| elemento | implementação |
|---|---|

| **Fantasma 3D** | Criado com `THREE.SphereGeometry(2,40,40)`; vértices da parte inferior são deslocados por ruído sinusoidal para formar a "saia" ondulada. O material é um `MeshStandardMaterial` com **alta emissividade** (`emissiveIntensity`) e cor controlada via parâmetro `bodyColor`. Rim lights direcionais adicionam contorno azulado. |

| **Olhos** | Olhos são um `Group` com esferas menores (`SphereGeometry(0.3)`) e glows maiores (`SphereGeometry(0.525)`) com materiais transparentes. A opacidade aumenta de acordo com o movimento do usuário; há também um controle de `_fade_` (`eyeGlowDecay`) que suaviza a iluminação ao parar de mover o mouse. |

| **Fireflies** | 20 "vagalumes" são gerados como pequenas esferas amarelas e um `PointLight`. Cada vagalume possui uma esfera de glow (`SphereGeometry(0.08)`) com `BackSide` e opacidade 0.4. Um objeto `userData` controla velocidade, fase e intensidade; eles se movem suavemente dentro de um volume (`±30` x, ±20` y, ±15` z`) e pulsam em brilho. |

| **Partículas** | Cada partícula é uma pequena forma (esfera, tetraedro ou octaedro) clonada de um `_pool_`. Ao mover o mouse, partículas nascem a partir da posição do fantasma, possuem escala randômica, rotação lenta e decaem (`life` e `decayRate`). A cor é fluorescente (ex.: `violet`) e pode ser ajustada via parâmetros. |

| **Interação com o mouse** | O fantasma segue o cursor suavemente. A coordenada do mouse é convertida de `screen` para `world` (`x: (event.clientX/window.innerWidth)*2 - 1`, `y: ...`). Um `followSpeed` governa a aproximação; além disso, ocorrem **oscilações** (senos e cossenos) para dar vida mesmo sem movimento. A intensidade do brilho dos olhos e a taxa de geração de partículas dependem da velocidade do mouse. |

Tipografia e layout

* Todo o conteúdo é centralizado com Flexbox (`display:flex; flex-direction:column; justify-content:center; align-items:center;`), ocupando `100vh`.

THUMB VIDEO MANIFESTO – Posição na página, comportamento, interação e scroll

1.1 Onde o "thumb video" fica

O *thumb video* pertence à **Home Hero** (`home-hero`), dentro de um wrapper identificado por classe **`.video-wrapper`**.

Padrão de layout (deduzido pelo HTML + animações):

- **Desktop ($\geq 768px$)**: `video-wrapper` fica **absoluto no canto inferior direito** da hero (ex.: `md:absolute md:bottom... md:right...`).
- **Mobile ($< 768px$)**: `video-wrapper` é **relative** e entra no fluxo normal do layout (classe começa com `relative`), com **aspect ratio vertical** (`aspect-[9/14]`), sugerindo thumbnail "cartaz"/vertical.

1.2 Interação (click/tap)

O `` tem um handler de click:

- **Desktop ($\geq 768px$)**: ao clicar no vídeo, ele **não alterna o mute**; ele **faz scroll** até o final da área "pinned" da hero:
 - Chamada observada: `scroll(distanceToBottom(... pin-spacer ...) - 1)`
 - Resultado: força o usuário a "revelar" a segunda fase da hero (quando o vídeo já ocupou área maior e overlays aparecem).
- **Mobile ($< 768px$)**: ao tocar no vídeo, ele **alterna o estado de mute**:
 - `showreelMuted = !showreelMuted`
 - Ou seja: no mobile, o vídeo é consumido como mídia "inline" e o tap controla áudio.

1.3 Scroll: pin + scrub (comportamento principal)

No desktop, a hero é **pinned** (fixada) com **scrub**.

Configuração observada no bundle:

- `scrollTrigger: { trigger: ".home-hero", pin: true, start: "top top", end: "+=800", scrub: 1, ... }`

Isso significa:

- A hero "segura" o scroll por ~800px de progresso virtual.
- A animação do vídeo é sincronizada com o scroll ("scrub"), em vez de ser uma animação com tempo fixo.

1.4 Transformações do thumb (desktop)

Durante a animação/pinning, o vídeo sofre **morphing**:

Transição 1 (desktop):

- `home-hero .video-wrapper`
 - **from**: `width: "219px"`, `height: "131px"`, `right: 30`, `bottom: 30`, `borderRadius: 5`
 - **to**: `width: "100%"`, `height: "100%"`, `right: 0`, `bottom: 0`, `borderRadius: 0`
 - easing/duração: `ease: "expo.inOut"`, `duration: 0.7`

Transição complementar:

- `home-hero video`
 - `borderRadius: 5` → `borderRadius: 0` (`ease: "power4.out"`, `duration: 0.9`)

> Interpretação: começa como **thumb pequeno** no canto inferior direito e se transforma em **vídeo fullscreen** dentro do container da hero.

```

### 1.5 Estados por progresso do scroll (gatilhos)
O `onUpdate` do ScrollTrigger define thresholds:

- `progress <= 0.03`
  - remove `no-hover` do `.video-wrapper`
- caso contrário
  - adiciona `no-hover` (bloqueia alguns hovers durante transição inicial)

- `progress >= 0.75` (e ainda não executado)
  - adiciona `.show` em:
    - `.home-hero .video-text`
    - `.home-hero .toggle-sound`
    - `.home-hero .video-overlay`
  - altera `showreelMuted` para **false** (desmuta) **em desktop**
    (observado `store("showreelMuted", !1)`)

- `progress < 0.75`
  - remove `.show` dos elementos acima
  - força `showreelMuted = true` (muta novamente)

```

1.6 Hover/overlay (desktop)

O vídeo tem estilo de hover (observado por utilitários):

- `group-hover:scale-105` + `transition duration-500 ease-in-out`

E existem camadas:

- `.video-overlay` (provável layer de gradiente/sombra)
- `.video-text` (texto e metadados que aparecem com `.show`)
- `.toggle-sound` (botão de som, aparece com `.show`)

📱 6. Responsividade

**Comportamento Responsivo Textos

Viewport:

- **Desktop**:

H1: "Você não vê" (linha 01)
 "o design." (linha 02)
`_Fonte: TT Norms Pro Black, 6–9rem, tracking-tight_`

- **Tablet**:

H1: "Você não vê" (linha 01)
 "o design." (linha 02)
`_Fonte: TT Norms Pro Black, 6–9rem, tracking-tight_`

- **Mobile**:

H1: "Você não"
 "vê o"
 "design."
`_Fonte: TT Norms Pro Black, 6–9rem, tracking-tight_`

---- ### **Animação Ghost - Responsividade**

- * A implementação original é **fluida**: o WebGL renderer dimensiona para `window.innerWidth`/`window.innerHeight`; um listener de `resize` recalcula a razão de aspecto, atualiza `camera`, `renderer`, `composer` e uniformes de shaders. Portanto, ele se adapta a qualquer resolução.
- * A tipografia utiliza unidades `vw` (viewport width) para o tamanho do título e mantém margens em `vh`. Esse ajuste natural dispensa media queries explícitas, mas para uma aplicação real recomenda-se:
 - * **Mobile-first** – defina a base de fontes com `clamp()` ou classes Tailwind (`text-4xl md:text-6xl lg:text-8xl`) para controlar legibilidade em telas pequenas.
 - * **Breakpoints** – use breakpoints Tailwind (`sm`, `md`, `lg`, `xl`) para ajustar espaçamentos, tamanho do container e possivelmente reduzir a quantidade de partículas/vagalumes em dispositivos com performance limitada.
- * **Fallback touch** – em touch devices onde `mousemove` não ocorre, mantenha o fantasma centralizado e rode apenas a animação de flutuação. Detecte `pointer: coarse` e reduza efeitos opcionais para preservar bateria.

---- ### **THUMB VIDEO MANIFESTO - Responsividade:**

Desktop

Estado Inicial:

- Posição: `bottom-right`, `30vw` largura
- Estilo: `rounded`, `shadow`, `aspect-video`
- Props: autoplay, loop, muted, `playsInline`

Scroll Behavior:

- Vídeo fixo no viewport (não rola com página)
- Animação:
 - scale: `[0.3, 1]`
 - posX/posY: `["100%", "50%"]`
 - borderRadius: `["16px", "0px"]`
 - editorial opacity: `[1, 0]`

Lógica Scroll:

- Quando atinge fullscreen:
 - `hold 2s` com scroll travado
 - vídeo se **desmuta**
- Ao sair do Hero: volta a ser **muted**

Click Behavior:

- Clica → vai direto ao fullscreen
- Triggera mesma lógica de áudio/scroll

****Hover**:**

- Scale `1 → 1.05`, `500ms`

****(Mobile)****

****Layout**:**

- Seção fullscreen logo após a Hero – colado as paredes da pagina
- `aspect-video` ,

****Vídeo**:**

- autoplay, loop, muted, `playsInline`
- botão visível para som (tap = unmute)
- Ao sair da seção → mutar de novo

****Animação**:**

```
```js
initial: { opacity: 0, scale: 0.95, y: 20 }
animate: { opacity: 1, scale: 1, y: 0 }
transition: { duration: 0.6, ease: [0.22, 1, 0.36, 1] }
```
```

****Acessibilidade**:**

- Controles claros
- Modo focável visível
- Sem som automático

🧠 7. Acessibilidade & SEO

**Animação Ghost – Acessibilidade e SEO**

- * ****Semântica**:** mantenha o texto principal dentro de um `

` e a linha secundária como `` ou ` `. Evite usar ` ` dentro do ``; em vez disso, use CSS (`display:block` para separar linhas) ou `Framer Motion` para animação de cada linha.
- * ****Contraste**:** a cor `#e0e0e0` em fundo `#0a0a0a` garante contraste suficiente (> 7:1) para WCAG 2.1. Teste as cores do brilho do fantasma contra o fundo; dê alternativa de tema sem efeitos para usuários sensíveis a brilho/interferência.
- * ****Canvas acessível**:** o canvas de Three.js é puramente decorativo. Adicione `aria-hidden="true"` e `role="presentation"` ao contêiner do canvas. Forneça uma descrição alternativa da animação via `

` ou `aria-label` ("Fantasma flutuante com partículas luminosas") para leitores de tela.
- * ****Carregamento**:** o preloader deve ser anunciável; inclua `aria-live="polite"` e texto visível indicando progresso. O progresso é visual e há uma barra; forneça também `aria-valuenow`/`aria-valuemax` nos momentos de carregamento se possível.

* **SEO**: como a hero é a primeira seção, inclua meta tags (`<title>`, `<meta name="description">`, `<meta property="og:image">`) no layout Next.js. Utilize Next.js `Image` para pré-carregar versões estáticas do fantasma ou do fundo como `_placeholder_`.

THUMB VIDEO MANIFESTO – Acessibilidade & SEO:

3.1 Problemas típicos do padrão original (e como corrigir)

O padrão "vídeo clicável" costuma ser um `<video>` com handler de click. Para acessibilidade:

Checklist obrigatório:

- [] Não usar o vídeo como único "botão".
 - ✓ Envolver com `<button>` ou `<a>` com `role="button"` e `aria-label`.
- [] Botão de som (`toggle-sound`) com:
 - `aria-label="Ativar/Desativar som do vídeo"
 - `aria-pressed={muted ? "false" : "true"}`
- [] `playsInline` no mobile (já existe) para evitar fullscreen indesejado.
- [] **Respeitar prefers-reduced-motion**: reduzir/encurtar morphing e remover "scrub" intenso.
- [] Headline principal da hero como `<h1>` (sem pular heading levels).
- [] Contraste: overlay/gradiente no vídeo para garantir leitura do texto.

SEO:

- O vídeo é decorativo/experiencial; não deve substituir o conteúdo textual importante.
 - Se a thumb tem CTA, garantir link navegável e texto indexável fora do canvas.
-

🔧 8. Extras Técnicos (pra sua stack):

Estrutura (Desktop – Z-Index Stack)

1. **Z-50** Preloader (SVG ghost + texto)
2. **Z-30** Manifesto Video Thumbnail (flutuante, bottom-right)
3. **Z-20** Ghost Atmosphere (WebGL Canvas) – O Ghost sempre acima da camada de texto
4. **Z-10** Editorial Text Block (centralizado)
5. **Z-0** Fundo Gradient: `#040013 → #0b0d3a`

**Animação Ghost – Arquitetura recomendada (Next.js App Router)

A estrutura abaixo divide a hero em componentes reutilizáveis. Todos os componentes podem ser organizados dentro de `app/components` para o App Router.

1. Estrutura de arquivos

```
```
app/
 page.tsx → Home
 components/
 Hero.tsx → Componente principal da hero (texto + canvas)
 GhostScene.tsx → Cena 3D com Three.js/R3F
 Preloader.tsx → Pré-carregador e animação inicial
 HeroText.tsx → Container de texto e CTA
 lib/
 useMouse.ts → Hook para rastrear movimento do mouse
 styles/
 globals.css → Importa Tailwind e fontes customizadas
 tailwind.config.ts → Extensão de tema (cores, fontes)
````
```

2. Tailwind e configuração

- * **Instalação**: o projeto Next.js deverá estar com Tailwind já instalado (`pnpm add tailwindcss @types/tailwindcss`).
- * **Extensões de tema**: configure fontes e cores fluorescentes para reutilização:

```
```ts
// tailwind.config.ts
import type { Config } from 'tailwindcss'

export default <Config>{
 content: [
 './app/**/*.{js,ts,jsx,tsx}',
],
 theme: {
 extend: {
 fontFamily: {
 boldonse: ['var(--font-boldonse)', 'sans-serif'],
 monospace: ['PPSupplyMono', 'monospace'],
 },
 colors: {
 background: '#0a0a0a',
 highlight: '#e0e0e0',
 fluorescent: {
 cyan: '#00ffff',
 lime: '#00ff00',
 magenta: '#ff00ff',
 yellow: '#ffff00',
 orange: '#ff4500',
 pink: '#ff1493',
 purple: '#9400d3',
 blue: '#0080ff',
 green: '#00ff80',
 }
 }
 }
 }
}
```

```

 red: '#ff0040',
 teal: '#00ffaa',
 violet: '#8a2be2',
 }
},
screens: {
 xs: '375px',
},
keyframes: {
 ghostFloat: {
 '0%,100%': { transform: 'translateY(0)' },
 '50%': { transform: 'translateY(-8px)' },
 },
 eyePulse: {
 '0%,100%': { transform: 'scale(1)' },
 '50%': { transform: 'scale(1.3)' },
 },
},
animation: {
 ghostFloat: 'ghostFloat 3s ease-in-out infinite',
 eyePulse: 'eyePulse 2s ease-in-out infinite',
},
},
},
plugins: [],
}
```

```

3. Componentes React

```

#### `Hero.tsx`


```tsx
// app/components/Hero.tsx
import dynamic from 'next/dynamic';
import { Suspense } from 'react';
import HeroText from './HeroText';
import Preloader from './Preloader';

// Carregar cena 3D somente no cliente (desativa SSR)
const GhostScene = dynamic(() => import('./GhostScene'), { ssr: false });

export default function Hero() {
 return (
 <section className="relative h-screen w-full bg-background
text-highlight overflow-hidden">
 {/* Preloader controla exibição inicial */}
 <Preloader />
 {/* Texto sobreposto */}
 <HeroText />
 {/* Canvas 3D e pós-processamento */}
 <Suspense fallback={null}>
 <GhostScene />
 </Suspense>
 </section>
)
}
```

```

```

);
}

```
// app/components/HeroText.tsx
import { motion } from 'framer-motion';

export default function HeroText() {
 const container = {
 hidden: { opacity: 0 },
 show: { opacity: 1, transition: { delay: 1.5, duration: 1.2 } },
 };

 return (
 <motion.div
 className="absolute inset-0 flex flex-col justify-center items-center
 text-center pointer-events-none"
 variants={container}
 initial="hidden"
 animate="show"
 >
 <span className="font-monospace text-xs uppercase tracking-widest
 mb-2">[Brand awareness]
 <h1 className="font-bold font-semibold uppercase leading-tight
 mb-4 text-[10vw] md:text-6xl lg:text-8xl">
 Você não vê o design
 </h1>
 <h2 className="font-bold font-medium uppercase text-[6vw]
 md:text-4xl lg:text-5xl">
 Mas ele vê você
 </h2>
 </motion.div>
);
}
```

```

*Note: O uso de `pointer-events:none` impede interferência com o canvas; as tags `<h1>` e `<h2>` garantem estrutura semântica.

```

```
// app/components/Preloader.tsx
import { useState, useEffect } from 'react';
import { motion, AnimatePresence } from 'framer-motion';

export default function Preloader() {
 const [complete, setComplete] = useState(false);

 useEffect(() => {
 // Simule carga ou escute eventos de carregamento (ex.: fonts/webgl)
 const timer = setTimeout(() => setComplete(true), 3000);
 return () => clearTimeout(timer);
 });
}
```

```

}, []);

return (
 <AnimatePresence>
 {!complete && (
 <motion.div
 className="fixed inset-0 z-50 flex flex-col justify-center items-center bg-gradient-to-br from-[#0a0a0a] via-[#1a1a1a] to-[#0a0a0a]"
 initial={{ opacity: 1 }}
 exit={{ opacity: 0, transition: { duration: 1 } }}>
 <motion.div
 initial={{ y: 0 }}
 animate={{ y: [-10, 10, -10] }}
 transition={{ duration: 3, repeat: Infinity, ease: 'easeInOut' }}>
 <!-- Ícone SVG do fantasma -->
 <svg className="w-20 h-20" viewBox="0 0 512 512"
 fill="currentColor" aria-hidden="true">
 <path d="m508.374
 432.802s-46.6-39.038-79.495-275.781c-8.833-87.68-82.856-156
 .139-172.879-156.139-90.015 0-164.046 68.458-172.879
 156.138-32.895 236.743-79.495 275.782-79.495 275.782-15.107
 25.181 20.733 28.178 38.699 27.94 35.254-.478 35.254 40.294
 70.516 40.294 35.254 0 35.254-35.261 70.508-35.261s37.396
 45.343 72.65 45.343 37.389-45.343 72.651-45.343c35.254 0
 35.254 35.261 70.508 35.261s35.27-40.772
 70.524-40.294c17.959.238 53.798-2.76 38.692-27.94z"/>
 <circle cx="208" cy="225" r="22" />
 <circle cx="297" cy="225" r="22" />
 </svg>
 </motion.div>
 </motion.div>
 <motion.p className="font-monospace text-xs uppercase tracking-widest mt-4"
 initial={{ opacity: 1 }}
 animate={{ opacity: [1, 0.2, 1] }}
 transition={{ duration: 2, repeat: Infinity }}>
 Summoning spirits...
 </motion.p>
 </motion.div>
)}>
</AnimatePresence>
);
}
```

```

`GhostScene.tsx`

```

```tsx
// app/components/GhostScene.tsx
'use client';
import { Canvas, useFrame, useThree } from '@react-three/fiber';
import { useEffect, useRef, useState, Suspense } from 'react';
import { Sphere, PointLight, Plane, Float } from '@react-three/drei';

```

```

import * as THREE from 'three';

// Shader para o efeito de atmosfera (baseado no script original)
const AtmosphereMaterial = {
 uniforms: {
 ghostPosition: { value: new THREE.Vector3() },
 revealRadius: { value: 20 },
 fadeStrength: { value: 1.7 },
 baseOpacity: { value: 0.9 },
 revealOpacity: { value: 0.05 },
 time: { value: 0 },
 },
 vertexShader: `
 varying vec2 vUv;
 varying vec3 vWorldPosition;
 void main() {
 vUv = uv;
 vec4 worldPos = modelMatrix * vec4(position, 1.0);
 vWorldPosition = worldPos.xyz;
 gl_Position = projectionMatrix * modelViewMatrix * vec4(position,
 1.0);
 }
 `,
 fragmentShader: `
 uniform vec3 ghostPosition;
 uniform float revealRadius;
 uniform float fadeStrength;
 uniform float baseOpacity;
 uniform float revealOpacity;
 uniform float time;
 varying vec2 vUv;
 varying vec3 vWorldPosition;
 void main() {
 float dist = distance(vWorldPosition.xy, ghostPosition.xy);
 float dynamicRadius = revealRadius + sin(time * 2.0) * 5.0;
 float reveal = smoothstep(dynamicRadius * 0.2, dynamicRadius, dist);
 reveal = pow(reveal, fadeStrength);
 float opacity = mix(revealOpacity, baseOpacity, reveal);
 gl_FragColor = vec4(0.001, 0.001, 0.002, opacity);
 }
 `,
 transparent: true,
};

function Ghost() {
 const group = useRef<THREE.Group>(null!);
 const body = useRef<THREE.Mesh>(null!);
 const [hover, setHover] = useState(false);
 const { viewport, mouse } = useThree();

 // Atualiza posição para seguir o mouse
 useFrame(({ clock }) => {
 const t = clock.getElapsedTime();
 // movimento flutuante constante
 const floatY = Math.sin(t * 1.5) * 0.05 + Math.cos(t * 0.7) * 0.03;
 ...
 });
}

```

```

group.current.position.y = floatY;
// segue mouse suavemente
const targetX = (mouse.x ?? 0) * viewport.width * 0.5;
const targetY = (mouse.y ?? 0) * viewport.height * 0.3;
group.current.position.x += (targetX - group.current.position.x) * 0.05;
group.current.position.y += (targetY - group.current.position.y) * 0.05;

// ajuste do emissive intensity pulsando
const pulse = Math.sin(t * 1.6) * 0.6 + Math.sin(t * 0.6) * 0.12;
if (body.current.material instanceof THREE.MeshStandardMaterial) {
 body.current.material.emissiveIntensity = 5.8 + pulse;
}
});

// cria geometria do fantasma com "saia" ondulada
useEffect(() => {
 if (!body.current) return;
 const geo = body.current.geometry as THREE.SphereGeometry;
 const pos = geo.attributes.position as THREE.BufferAttribute;
 const arr = pos.array as Float32Array;
 for (let i = 0; i < arr.length; i += 3) {
 const y = arr[i + 1];
 if (y < -0.2) {
 const x = arr[i];
 const z = arr[i + 2];
 const noise1 = Math.sin(x * 5) * 0.35;
 const noise2 = Math.cos(z * 4) * 0.25;
 const noise3 = Math.sin((x + z) * 3) * 0.15;
 arr[i + 1] = -2.0 + noise1 + noise2 + noise3;
 }
 }
 geo.computeVertexNormals();
}, []);

return (
<group ref={group} scale={2.4}>
 <mesh ref={body}>
 <sphereGeometry args={[2, 40, 40]} />
 <meshStandardMaterial
 color="#0f2027"
 roughness={0.02}
 metalness={0}
 transparent
 opacity={0.88}
 emissive="#0080ff"
 emissiveIntensity={5.8}
 />
 </mesh>
 {/* Olhos */}
 <group>
 <mesh position={[-0.7, 0.6, 2.0]}>
 <sphereGeometry args={[0.3, 12, 12]} />
 <meshBasicMaterial color="#8a2be2" transparent opacity={0.0} />
 </mesh>
 <mesh position={[0.7, 0.6, 2.0]}>

```

```

 <sphereGeometry args={[0.3, 12, 12]} />
 <meshBasicMaterial color="#8a2be2" transparent opacity={0.0} />
 </mesh>
</group>
/* Vagalumes e partículas podem ser componentes separados */
</group>
);
}

export default function GhostScene() {
const planeRef = useRef<THREE.Mesh>(null!);
const ghostRef = useRef<THREE.Group>(null!);

useFrame(({ clock }) => {
const t = clock.getElapsedTime();
if (planeRef.current) {
const mat = planeRef.current.material as any;
mat.uniforms.time.value = t;
mat.uniforms.ghostPosition.value.copy(ghostRef.current?.position ??
new THREE.Vector3());
}
});

return (
<Canvas
 className="absolute inset-0"
 gl={{ antialias: true, alpha: true }}
 camera={{ position: [0, 0, 20], fov: 75 }}
>
 <ambientLight color="#0a0a2e" intensity={0.08} />
 /* Rim lights */
 <directionalLight position={[−8, 6, −4]} color="#4a90e2"
 intensity={1.8} />
 <directionalLight position={[8, −4, −6]} color="#50e3c2"
 intensity={1.26} />

 /* Atmosfera */
 <mesh ref={planeRef} position={[0, 0, −50]} renderOrder={−100}>
 <planeGeometry args={[300, 300]} />
 <shaderMaterial attach="material" args={[AtmosphereMaterial]} />
 </mesh>

 <Suspense fallback={null}>
 <Ghost />
 </Suspense>
</Canvas>
);
}
```

```

**Observações*:*

- A cena utiliza `Canvas` do `@react-three/fiber` com `className="absolute inset-0"` para ocupar toda a tela.

- O shader de atmosfera foi transposto para `shaderMaterial`. Para o shader analógico (grain/vignette), pode-se criar outro `ShaderPass` usando `@react-three/postprocessing` se necessário.
- Para adicionar **vagalumes** e **partículas**, crie componentes que iteram listas de objetos e usam `useFrame` para atualizar posição e opacidade. Reduza a contagem em dispositivos móveis.

Animação Ghost - Integração com o restante da stack:

- * **App Router**: as rotas ficam em `app/page.tsx`. O componente `Hero` pode ser importado ali. Qualquer rota subsequente (blog, contato) poderá ser carregada via links normais. Como o canvas é puramente do cliente, o componente `GhostScene` deve ser marcado com `use client` e importado dinamicamente.
- * **Firebase Hosting**: compile com `next build` e publique usando `firebase deploy`. Garanta que as variáveis de ambiente da hospedagem estejam definidas em `.env` e leia-as via `process.env` apenas no servidor.
- * **Supabase Storage**: armazene assets como fontes personalizadas, modelos glb ou texturas no Supabase. Carregue-os em tempo de build ou no cliente usando a URL gerada. Defina um mecanismo de fallback no preloader caso o download falhe.

**Animação Ghost - Conclusão

A animação "Spectral Ghost" combina **WebGL/Three.js**, um **shader de película analógica**, partículas responsivas e um preloader estilizado. Para reconstruí-la em Next.js com React Three Fiber é necessário separar a lógica em componentes reutilizáveis, cuidar da **responsividade** com unidades relativas e breakpoints Tailwind, e garantir **acessibilidade** com marcas semânticas e canvas marcado como decorativo. Com as configurações e exemplos acima, um agente de construção de sites pode replicar o efeito de forma fiel e escalável.

THUMB VIDEO MANIFESTO - Extras técnicos (stack solicitada):

4.1 Divisão de componentes recomendada Estrutura (App Router):

- `<Hero />`
 - `<GhostCanvasBackground />` (R3F)
 - `<HeroCopy />` (h1, subtítulo, CTA)
 - `<ShowreelThumb />` (thumb vídeo + overlay + mute + scroll progress)
 - `<Preloader />` (se necessário)

4.2 App Router: client-side ou server-side?

- **Hero** precisa ser **Client Component** por:
 - WebGL + R3F
 - listeners de mouse/scroll
 - controle de vídeo (mute/play)
 - framer-motion `useScroll`

Então:

- `app/page.tsx` pode ser server, mas renderiza ``:
 - `const HeroClient = dynamic(() => import("./HeroClient"), { ssr: false })`

4.3 Three.js / R3F: fallback SSR

- Use `dynamic(..., { ssr:false })` para o canvas.
- Renderize um fallback estático:
 - background gradient + textura (CSS) + ghost SVG/PNG + thumb vídeo.

4.4 Tailwind: theme extend (tokens úteis)

Sugestão mínima (exemplo):

- `colors.blue = "#0B63FF"` (ou o azul neon do projeto)
- `spacing.gutter = "20px"` (e `md: 30px`, `lg: 40px`)
- `borderRadius.videoThumb = "5px"'

4.5 Framer Motion: declarativas ou variantes?

Para recriar fielmente o scrub do scroll:

- Use **Framer Motion** + `useScroll` + `useTransform`** (declarativo)
 - mantém stack "pura" sem GSAP
- Alternativa: **GSAP ScrollTrigger**** (fica idêntico ao original), mas adiciona dependência.

Recomendação: **Framer Motion** (porque está na sua stack), com fallback "CSS sticky".

THUMB VIDEO MANIFESTO – Implementação sugerida (Framer Motion, sem GSAP):

5.1 Conceito

- Hero com `position: relative` e `height: 200svh` (mobile) / `height: 100vh` (desktop).
- Conteúdo textual fica central.
- Thumb fica:
 - mobile: dentro do fluxo com aspect vertical
 - desktop: `absolute bottom-gutter right-gutter`, inicia pequeno e, conforme scroll, vira fullscreen.

5.2 Código (ShowreelThumb.tsx)

```
> Este componente implementa:  
> - morph por scroll (219x131 → 100%)  
> - borderRadius 5 → 0  
> - threshold 75% para mostrar overlay + botão de som  
> - click desktop = scroll-to-reveal; click mobile = toggle mute  
  
```tsx  
'use client';
```

```
import { useEffect, useMemo, useRef, useState } from 'react';
import { motion, useScroll, useTransform } from 'framer-motion';
```

```
function useMediaQuery(query: string) {
 const [matches, setMatches] = useState(false);
```

```

useEffect(() => {
 const m = window.matchMedia(query);
 const onChange = () => setMatches(m.matches);
 onChange();
 m.addEventListener('change', onChange);
 return () => m.removeEventListener('change', onChange);
}, [query]);
return matches;
}

type Props = {
 heroRef: React.RefObject<HTMLElement>;
 src: string; // Supabase Storage URL (mp4)
};

export function ShowreelThumb({ heroRef, src }: Props) {
 const isDesktop = useMediaQuery('(min-width: 768px)');
 const videoRef = useRef<HTMLVideoElement | null>(null);

 // Estado de áudio (equivale ao store showreelMuted)
 const [muted, setMuted] = useState(true);

 // Scroll progress dentro da hero
 const { scrollYProgress } = useScroll({
 target: heroRef,
 offset: ['start start', 'end start'], // ajusta o "scrub"
 });

 // Morphing desktop (valores do bundle: 219x131 → 100%)
 const w = useTransform(scrollYProgress, [0, 1], ['219px', '100%']);
 const h = useTransform(scrollYProgress, [0, 1], ['131px', '100%']);
 const right = useTransform(scrollYProgress, [0, 1], ['30px', '0px']);
 const bottom = useTransform(scrollYProgress, [0, 1], ['30px', '0px']);
 const radius = useTransform(scrollYProgress, [0, 1], ['5px', '0px']);

 // Mostrar overlays após 75%
 const overlayOpacity = useTransform(scrollYProgress, [0.74, 0.75], [0, 1]);

 // Controlar mute por threshold (desktop)
 useEffect(() => {
 if (!isDesktop) return;
 const unsub = scrollYProgress.on('change', (p) => {
 if (p >= 0.75) setMuted(false);
 else setMuted(true);
 });
 return () => unsub();
 }, [isDesktop, scrollYProgress]);

 useEffect(() => {
 if (!videoRef.current) return;
 videoRef.current.muted = muted;
 // tentativa de autoplay (silencioso no mobile)
 videoRef.current.play().catch(() => {});
 }, [muted]);
}

```

```
const onClick = async () => {
 if (!heroRef.current) return;

 if (isDesktop) {
 // desktop: scroll para "revelar"
 const rect = heroRef.current.getBoundingClientRect();
 const top = window.scrollY + rect.top;
 const end = top + heroRef.current.offsetHeight - window.innerHeight +
 1;
 window.scrollTo({ top: end, behavior: 'smooth' });
 } else {
 // mobile: toggle mute
 setMuted((m) => !m);
 }
};

return (
 <motion.div
 className={[
 'video-wrapper z-10',
 'relative aspect-[9/14] overflow-hidden rounded-[5px]',
 'md:absolute md:aspect-auto',
 'cursor-pointer group',
].join(' ')}
 style={
 isDesktop
 ? { width: w, height: h, right, bottom, borderRadius: radius,
 position: 'absolute' as const }
 : undefined
 }
 onClick={onClick}
 role="button"
 aria-label={isDesktop ? 'Revelar vídeo (scroll)' : muted ? 'Ativar som do vídeo' : 'Desativar som do vídeo'}
 tabIndex={0}
 onKeyDown={(e) => (e.key === 'Enter' || e.key === ' ') && onClick()}
 >
 <motion.video
 ref={videoRef}
 className="w-full h-full object-cover transition duration-500
 ease-in-out group-hover:scale-105"
 src={src}
 autoPlay
 loop
 playsInline
 muted={muted}
 />

 {/* Overlay escuro/gradiente */}
 <motion.div
 className="video-overlay absolute inset-0 pointer-events-none"
 style={{ opacity: overlayOpacity }}
 />
);
```

```

/* Texto/legenda */
<motion.div
 className="video-text absolute bottom-0 left-0 w-full p-4
 pointer-events-none"
 style={{ opacity: overlayOpacity }}
>
 <p className="text-white text-sm opacity-80">Showreel</p>
 <p className="text-white text-base font-medium">Strategy • Branding
 • Motion</p>
</motion.div>

/* Toggle som (só visível no desktop, após 75%) */
<motion.button
 type="button"
 className="toggle-sound absolute top-3 right-3 w-9 h-9 rounded-full
 bg-black/60 text-white backdrop-blur flex items-center
 justify-center"
 style={{ opacity: isDesktop ? overlayOpacity : 0, pointerEvents:
 isDesktop ? 'auto' : 'none' }}
 onClick={(e) => {
 e.stopPropagation();
 setMuted((m) => !m);
 }}
 aria-label={muted ? 'Ativar som do vídeo' : 'Desativar som do
 vídeo'}
 aria-pressed={!muted}
>
 {muted ? '🔇' : '🔊'}
</motion.button>
</motion.div>
);
```

```

5.3 CSS/Tailwind para overlay

No Tailwind, você pode adicionar utilitários, mas um mínimo em CSS global resolve:

```

```css
.video-overlay {
 background: radial-gradient(120% 120% at 70% 30%, rgba(0,0,0,0) 0%,
 rgba(0,0,0,.55) 70%, rgba(0,0,0,.75) 100%);
}
```

```

THUMB VIDEO MANIFESTO – Integração na Hero (App Router):

```

### 6.1 HeroClient.tsx
```tsx
'use client';

import { useRef } from 'react';

```

```

import { ShowreelThumb } from './ShowreelThumb';
import dynamic from 'next/dynamic';

const GhostCanvasBackground = dynamic(() =>
 import('./GhostCanvasBackground'), { ssr: false });

export default function HeroClient() {
 const heroRef = useRef<HTMLElement>(null);

 return (
 <section ref={heroRef} className="home-hero relative w-full h-[200svh]
 md:h-screen overflow-hidden bg-neutral-950">
 <GhostCanvasBackground />

 <div className="relative z-20 h-screen flex items-center
 justify-center text-center px-5">
 <div className="max-w-4xl">
 <p className="text-white/60 text-xs tracking-widest uppercase
 mb-4">[BRAND AWARENESS]</p>
 <h1 className="text-white text-5xl md:text-7xl font-semibold
 tracking-tight leading-tight">
 Você não vê o design.
 Mas ele vê
 você.
 </h1>
 </div>
 </div>

 <ShowreelThumb
 heroRef={heroRef}
 src={'https://YOUR_SUPABASE_PUBLIC_URL/LB>Showreel-2025.mp4'}
 />
 </section>
);
}
```
```

```

### ### \*\*THUMB VIDEO MANIFESTO – Recomendações

- ### Desktop
  - Implementar "pin" real:
    - **Opção A (sem GSAP)**: `section` com `height: 200vh` e inner sticky (100vh). O scroll "simula" o pin.
    - **Opção B (GSAP ScrollTrigger)**: reproduz exatamente. (Se o agente puder adicionar dependência.)
- ### Tablet touch
  - Decidir se "click = scroll" faz sentido.
  - Alternativa: no tablet, usar comportamento do mobile (toggle mute), porque o usuário está em touch.
- ### Performance
  - `preload="metadata"` para o vídeo.

- `playsInline` e `muted` no primeiro paint para garantir autoplay.
- Lazy-load do vídeo (IntersectionObserver) se a hero não está imediatamente visível.

---

### **### \*\*THUMB VIDEO MANIFESTO – Checklist de validação (para o agente autônomo)**

#### **\*\*Funcional\*\***

- [ ] Thumb inicia pequeno no canto inferior direito (desktop) e cresce para fullscreen com scroll.
- [ ] Border radius 5px → 0px durante a transição.
- [ ] Overlay + texto + toggle-sound aparecem após 75% do progresso.
- [ ] Desktop click faz scroll para o final da hero (reveal).
- [ ] Mobile click alterna mute.
- [ ] Estado de mute volta a true quando scroll retorna < 75%.

#### **\*\*A11y\*\***

- [ ] Wrapper focável + Enter/Space acionam.
- [ ] Botão de som com `aria-label` e `aria-pressed`.
- [ ] Respeitar `prefers-reduced-motion`.

#### **\*\*SEO\*\***

- [ ] Conteúdo textual em `

# ` e ` ` fora do canvas.
- [ ] Vídeo não é a única fonte de informação.

---

### **### \*\*THUMB VIDEO MANIFESTO – Onde o arquivo de vídeo entra (Supabase Storage)**

- Manifesto Video:  
`<https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/project-videos/VIDEO-APRESENTACAO-PORTFOLIO.mp4>`
- Usar URL pública no `src`.

Exemplo (pseudo):

```
```ts
const src =
  `${process.env
    .NEXT_PUBLIC_SUPABASE_URL}/storage/v1/object/public/assets/LB>Showreel-2025
  .mp4`;
```

```

## # 4.3 Portfolio Showcase – Protótipo Interativo

### ## Purpose

Apresentar as principais categorias de trabalho com \*\*sofisticação editorial\*\*, usando movimento, hierarquia tipográfica e interação progressiva para \*\*guiar o usuário naturalmente\*\* até áreas específicas do portfólio – replicando o ritmo, layout e comportamento da sessão equivalente na home do site de referência.

---

## ## Layout & Estrutura

### ### Desktop ( $\geq 1024\text{px}$ )

#### #### Estrutura Geral

- Headline centralizada:
  - \*\*"portfólio showcase"\*\*
    - "portfólio" em branco
    - "showcase" em `#0048ff`
  - Label flutuante contextual:
    - Texto: \*\*[what we love working on]\*\*
    - Cor: `#0048ff`
    - Posicionamento: absoluto, alinhado à esquerda, alinhado a esquerda e ao intem 'Brand & Campaigns' dentro da faixa
  - Três faixas interativas horizontais ['#8705f2'] (\*accordion-style stripes\*), com alinhamento alternado:
    1. \*\*Brand & Campaigns\*\* – alinhada à direita
    2. \*\*Videos & Motions\*\* – centralizada
    3. \*\*Web Campaigns, Websites & Tech\*\* – alinhada à esquerda
      - Quebra de linha após a vírgula
  - CTA centralizado abaixo das faixas:
    - \*\*"let's build something great →"\*\*

#### #### Estrutura de Cada Stripe

Cada faixa contém:

- \*\*Thumbnail de vídeo/imagem\*\*
  - Largura: `288px`
  - Aspect ratio: ~16:9
  - Bordas levemente arredondadas
  - Oculta por padrão
- \*\*Título da categoria\*\*
  - Tipografia grande (2x1–5x1)
  - Peso médio
  - Font-family: `TT Norms Pro Normal`
- \*\*Ícone de ação\*\*
  - Badge circular azul
  - Ícone de seta interna

## ## Interações & Animações

### ### Scroll Reveal (Desktop)

- Trigger: quando 30% da seção entra na viewport

- Animação:

```
```js
opacity: 0 → 1
translateY: 24px → 0
duration: 0.8s
easing: ease-out
stagger: 120ms entre faixas
```
```

- Durante a entrada, os títulos transitam para `#0057FF`, reforçando hierarquia visual.

---

#### ### Hover sobre a Stripe (Desktop)

##### #### 1. Revelação da Thumbnail

```
```js
width: 0 → 288px
opacity: 0 → 1
duration: 700ms
easing: cubic-bezier(0.22, 1, 0.36, 1)
```
```

##### #### 2. Ajuste de Espaçamento Interno

```
```js
gap: gap-7 → gap-10
duration: 300ms
```
```

##### #### 3. Ícone de Seta

```
```js
rotation: -45deg → 0deg
duration: 500ms
```
```

> A interação é **progressiva e silenciosa**, sem sobreposição agressiva ou quebra de layout.

---

#### ### Click

- Navegação para `/portfolio`
- Categoria correspondente aplicada via filtro (slug).

---

## ## Responsividade & Adaptação de Conteúdo

#### ### Mobile & Tablet ( $\leq 1023\text{px}$ )

##### #### Layout

- Cards verticais full-width
- Conteúdo texto alinhado a esquerda da página

- Ícone de Seta alinhado a direita da página
- Label flutuante removida
- CTA centralizado

#### #### Comportamento

- Sem hover
- Thumbnails ocultas ou estáticas
- Ícones de seta à direita

----

#### #### Categories & Assets

##### 1. \*\*Brand & Campaigns\*\*

- Slug: `brand-campaigns`
- Thumbnail:  
`<https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/project-images/Branding-Project.webp>`

##### 2. \*\*Videos & Motions\*\*

- Slug: `videos-motions`
- Thumbnail:  
`<https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/project-images/webdesigner-2%202.gif>`

##### 3. \*\*Web Campaigns, Websites & Tech\*\*

- Slug: `websites-webcampaigns-tech`
- Thumbnail:  
`[https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/project-images/WelcomeAd\\_800x500px.webp](https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/project-images/WelcomeAd_800x500px.webp)`

#### #### CTA Button

**Text:** "let's build something great →"  
**Destination:** `/#contact`  
**Estado**, Propriedade, Valor / Classe Tailwind, Duração, Easing  
**Idle**, Translação Y, translate-y-0, -, -  
**Hover**, Translação Y, -translate-y-px (Levitação sutil), 200ms, ease-out  
**Hover**, Background, bg-light-blue `#0048ff` (Iluminação), 300ms, default  
(`translateX: 4px`)  
**Optional:** Subtle looping animation on arrow in idle state (`translateX: 0 → 4px → 0`)

----

## ## Resultado Esperado

- Experiência editorial fluida
- Movimento como reforço de significado
- Consistência total entre desktop e mobile

-----

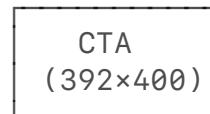
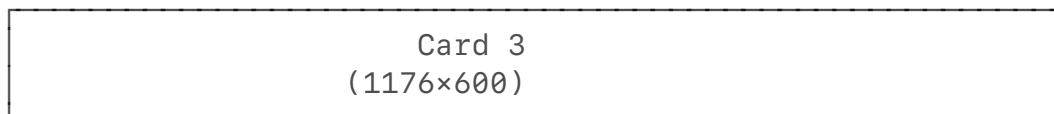
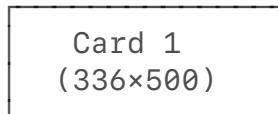
## # \*\*4.4 Featured Projects

\*\*Purpose:\*\* Showcase curated, high-quality work examples in an editorial, magazine-style layout (Bento Grid).

#### Layout (Desktop)

Irregular grid resembling a magazine spread:

```



```

\*\*Grid Implementation (Tailwind):\*\*

```
```javascript
// Row 1
<div className="md:col-span-5"><ProjectCard /></div>
<div className="md:col-span-7"><ProjectCard /></div>

// Row 2
<div className="md:col-span-12"><ProjectCard /></div>

// Row 3
<div className="md:col-span-8"><ProjectCard /></div>
<div className="md:col-span-4"><CTAProjectCard /></div>
````
```

#### Project Cards

\*\*Structure:\*\*

- \*\*Image/Video:\*\* Covers card, object-fit cover
- \*\*Pills (tags):\*\* Positioned absolute, top-right, semi-transparent background (`#E6EFEF` at 70% opacity), small text
- \*\*Info Block (below image):\*\*
  - Title (H3, medium weight)
  - Meta: Client • Year (`#6B7280`, small text)
  - Arrow icon in blue circle (translates right on hover)

\*\*Interaction (Desktop):\*\*

```
```javascript
// Hover on card
image: { scale: 1.03, translateY: -1 }
duration: 500ms

arrow: { translateX: 20px }
duration: 700ms ease-out

shadow: { shadow-xl shadow-blue-500/10 }
````
```

### \*\*Scroll Reveal:\*\*

```
```javascript
container: { opacity: 0 → 1, y: 40px → 0 }
cards: { scale: 0.96 → 1, staggerChildren: 0.12 }
duration: 0.7s ease-out
````
```

## #### Featured Projects Content

### 1. \*\*Magic – devolvendo a magia ao rádio\*\*

- Slug: `magic-radio-branding`
- Category: branding & campanha
- Client: Magic
- Year: 2023
- Image:  
`<https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/project-images/Brand-Identity%20copy.webp>`

### 2. \*\*Uma marca ousada e consistente\*\*

- Slug: `branding-project-01`
- Category: branding
- Client: Cliente confidencial
- Year: 2022
- Image:  
`<https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/project-images/Branding-Project.webp>`

### 3. \*\*Key visual para campanha sazonal\*\*

- Slug: `key-visual-campaign`
- Category: campanha
- Client: Cliente confidencial
- Year: 2021
- Image:  
`<https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/project-images/Key-Visual.webp>`

### 4. \*\*Experiência web em movimento\*\*

- Slug: `webdesigner-motion`
- Category: web & motion
- Client: Cliente confidencial
- Year: 2023
- Image:  
`<https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/project-images/webdesigner-2%202.gif>`

#### #### CTA Card

##### \*\*Content:\*\*

- Headline: "Like what you see?" - font normal - (on hover, text becomes `#0057FF`)
- Button: "view projects →" with arrow icon
- Background: `#040013` (on hover, text becomes `#0057FF`)

##### \*\*Interaction:\*\*

```
```javascript
// Hover
background: { #040013 }
text: { white → #0057FF }
arrow: { translateX: 4px }
duration: 300ms
```
```

##### \*\*Destination:\*\* `/portfolio`

#### #### Layout (Mobile)

- All cards stacked vertically
- Full-width, heights adapted to content/aspect ratio
- CTA card appears as last item

#### \*\*Desktop (≥1024px)\*\*

- Layout:
  - Irregular grid resembling a magazine spread (Bento Grid)
  - 4 project cards with varying sizes and positions
  - CTA card in bottom-right position
- Grid Implementation (Tailwind):

```
```jsx
// Row 1
<div className="md:col-span-5"> <ProjectCard /> </div>
<div className="md:col-span-7"> <ProjectCard /> </div>

// Row 2
<div className="md:col-span-12"> <ProjectCard /> </div>

// Row 3
<div className="md:col-span-8"> <ProjectCard /> </div>
<div className="md:col-span-4"> <CTAProjectCard /> </div>
```

```
- Project Cards:
  - Structure: Image/Video, Pills (tags), Info Block
  - Interaction: Hover effects on image, arrow, and shadow
  - Scroll Reveal: Container and cards with staggered animation

#### \*\*Mobile & Tablet (≤1023px)\*\*

- Layout:
  - All cards stacked vertically
  - Full-width, heights adapted to content/aspect ratio

- CTA card appears as last item
  - Specific Implementation Details:
    - Each project card takes 100% of the container width
    - Cards have consistent vertical spacing (32px)
    - Project images maintain aspect ratio but scale to fit width
    - Tags/pills are positioned at the top of the card
    - The CTA card is simplified to a full-width button with text "view projects →"
    - The grid layout is replaced with a simple vertical stack
    - Project titles and metadata are center-aligned
  - Content Adaptation:
    - The irregular bento grid is replaced with a clean vertical list
    - Project cards have consistent height on mobile
    - The CTA card no longer has a separate section but is integrated as a full-width button
    - The hover effects are replaced with touch-friendly tap effects
- 

## # \*\*4.5 Clients/Brands

**\*\*Purpose:\*\*** Build credibility by displaying recognizable brands previously worked with.

### #### Layout

**\*\*Desktop & Mobile:\*\***

- Full-width blue bar: `bg-[#0048ff]`
- Centered headline: "marcas com as quais já trabalhei" (white, bold, 2x1)
- Grid of logos: Responsive grid (3–4 columns on mobile, 6+ on desktop)
- Logos: White (apply `filter: brightness(0) invert` to SVGs)

### #### Interaction

**\*\*Hover (Desktop):\*\***

```
```javascript
logo: { scale: 1.04, brightness: 1.1 }
duration: 300ms
```

```

**\*\*Scroll Reveal:\*\***

```
```javascript
title: { opacity: 0 → 1, y: 16 → 0 }
logos: {
  staggerChildren: 0.03,
  each: { opacity: 0 → 1, y: 12 → 0, scale: 0.9 → 1 }
}
```

```

### #### Client Logos (12 total)

- `client1.svg` through `client12.svg`
- Base URL:  
`https://aymuvxysygrwoicsjgxj.supabase.co/storage/v1/object/public/client-logos/`

#### #### Accessibility

- Logos have descriptive `alt` text (e.g., "Logo da empresa X")
- Keyboard navigable (if logos link anywhere)
- Respect `prefers-reduced-motion` (disable entrance stagger)

#### \*\*Desktop & Mobile\*\*

- Layout:
  - Full-width blue bar: `bg-[#0048ff]`
  - Centered headline: "marcas com as quais já trabalhei"
  - Grid of logos: Responsive grid (3-4 columns on mobile, 6+ on desktop)
  - Logos: White (apply `filter: brightness(0) invert` to SVGs)
- Interaction:
  - Hover (Desktop): logo scale and brightness increase
  - Scroll Reveal: title and logos with staggered animation
- Mobile-Specific Details:
  - Grid changes to 2 columns on small mobile, 3 columns on larger mobile/tablet
  - Logos are scaled to 70% of their desktop size on mobile
  - Vertical spacing between logo rows is 24px on mobile
  - The headline is smaller (1.5rem vs 2rem on desktop)
  - Logo grid has 16px padding on all sides on mobile
  - The blue background extends full width with no horizontal container constraints

----

## # \*\*4.6 Contact

\*\*Purpose:\*\* Provide clear contact information and enable message submission.

#### #### Layout (Desktop)

Two-column layout:

- \*\*Left column:\*\* Contact info + social media links
- \*\*Right column:\*\* Contact form

#### #### Layout (Mobile)

Single column, vertically stacked:

1. Headline + subheadline
2. Contact info
3. Contact form
4. Social media links

#### #### Content

\*\*Headline:\*\* "contato"

\*\*Subheadline:\*\* "Tem uma pergunta ou quer trabalhar junto?"

#### #### Contact Information

\*\*Direct Channels:\*\*

- \*\*Phone:\*\* `+55 (11) 98396-6838`
  - Icon: Phone
  - Link: `tel:+5511983966838`
- \*\*Email (primary):\*\* `danilo@portfoliodanilo.com`
  - Icon: Envelope
  - Link: `mailto:danilo@portfoliodanilo.com`
- \*\*Email (secondary):\*\* `dannovaisv@gmail.com`
  - Icon: Envelope
  - Link: `mailto:dannovaisv@gmail.com`

\*\*Interaction:\*\*

- Text color: `#111111`
- Hover: Underline + color change to `#0057FF`
- Icons match text color

\*\*Social Media & Portfolio:\*\*

- Icons only (no text labels)
- Platforms: Instagram, Facebook, LinkedIn, Portfolio site, Twitter
- Color: `#111111`, hover: `#0057FF` + `scale(1.1)`
- All open in new tab (`target="\_blank"`, `rel="noopener noreferrer"`)

\*\*Links:\*\*

- Instagram: `https://instagram.com/danilo\_novais`
- Facebook: `https://facebook.com/danilonovaisvilela`
- LinkedIn: `https://linkedin.com/in/danilonovais`
- Portfolio: `https://portfoliodanilo.com`
- Twitter: `https://twitter.com/danilo\_novais`

#### #### Contact Form

\*\*Fields:\*\*

- Name (text input, required)
- Email (email input, required)
- Message (textarea, required)

\*\*Submit Button:\*\*

- Label: "Enviar Mensagem"
- Background: `#0048ff`
- Text: White
- Hover: Slight elevation (`translateY: -1px`), `scale(1.02)`
- Tap: `scale(0.98)`

\*\*Form Action:\*\*

- Method: POST
- Endpoint: `https://formsubmit.co/danilo@portfoliodanilo.com`

- States:**
  - Focus on input:** Border color `#0057FF`, shadow `ring-2 ring-blue-500`
  - Error:** Show validation message below field
  - Success:** Show success message, clear form
  - Loading:** Disable button, show spinner

#### #### Interactions & Animations

##### \*\*Scroll Reveal:\*\*

```
```javascript
section: { opacity: 0 → 1, y: 24 → 0 }
form fields: { staggerChildren: 60ms }
duration: 0.6s
```

```

##### \*\*Form Interactions:\*\*

```
```javascript
// Input focus
ring-2 ring-blue-500 ring-offset-2

// Button hover
{ scale: 1.02, y: -1 }

// Button tap
{ scale: 0.98 }
```

```

##### \*\*Desktop (≥1024px):\*\*

- Layout:
  - Two-column layout: Left column (contact info), Right column (contact form)
- Content:
  - Headline: "contato"
  - Subheadline: "Tem uma pergunta ou quer trabalhar junto?"
  - Contact Information: Direct channels (phone, emails), Social media
  - Contact Form: Name, Email, Message fields, Submit button

##### \*\*Mobile & Tablet (≤1023px):\*\*

- Layout:
  - Single column, vertically stacked:
    - Headline + subheadline
    - Contact info
    - Contact form
    - Social media links
- Specific Implementation Details:
  - All elements are full-width with appropriate spacing
  - Form fields have larger touch targets (minimum 48x48px)
  - The contact info section has increased vertical spacing between items
  - Social media icons are displayed in a single row at the bottom
  - The form submit button is full-width with increased height
  - The headline and subheadline are center-aligned
  - The contact information is simplified to a vertical list
- Content Adaptation:
  - The two-column layout collapses to a single column
  - Social media icons are reduced to just the icons (no text)

- Form fields have increased padding for touch interaction
- The "Enviar Mensagem" button has a minimum width of 100% on mobile

#### #### Accessibility

- All inputs have associated `<label>` elements
- Keyboard navigable
- Focus indicators (`focus-visible`)
- Error messages programmatically associated with fields
- Minimum touch target size: 48x48px (mobile)

---

## # \*\*4.7 Footer

**\*\*Purpose:\*\*** Provide legal information, supplementary navigation, and social links while respecting the overall editorial aesthetic.

#### #### Desktop ( $\geq 1024\text{px}$ )

##### **\*\*Layout:\*\***

- Fixed bar at bottom of viewport
- Persistent (always visible)
- Horizontal layout: Copyright (left) | Navigation + Social (right)
- Height: `48–64px`

##### **\*\*Behavior:\*\***

- `position: fixed`, `bottom: 0`, `z-index: 10`
- Does not scroll away

#### #### Mobile & Tablet ( $\leq 1023\text{px}$ )

##### **\*\*Layout:\*\***

- Static section in document flow (last element on page)
- Vertical stack: Copyright → Navigation → Social
- **Never fixed**
- **Never overlaps content**

##### **\*\*Spacing:\*\***

- Generous vertical padding: `py-10`
- Spacing between blocks: `space-y-6`
- Minimum touch target: 48x48px

#### #### Content

##### **\*\*Copyright:\*\***

- "© 2025 Danilo Novais Vilela – todos os direitos reservados"
- Color: White
- Small text

##### **\*\*Navigation Links:\*\***

- Home → `#hero`
- Portfólio Showcase → `#portfolio-showcase`
- Sobre → `/sobre`
- Contato → `#contact`

#### **\*\*Social Media:\*\***

- Same icons and links as Contact section
- White icons, hover: slight opacity reduction (desktop) or focus indicator (mobile)

#### **#### Background**

- Solid blue: `bg-[#0057FF]`
- Text: White

#### **#### Interactions**

##### **\*\*Desktop:\*\***

```
```
// Hover on links
opacity: 1 → 0.8
underline animation
duration: 200ms
```

```
// Hover on social icons
scale: 1.05
duration: 200ms
```
```

##### **\*\*Mobile:\*\***

- No hover dependencies
- Feedback only on `:active` / `:focus-visible`

#### **\*\*Desktop ( $\geq 1024px$ )\*\***

- Layout:
  - Fixed bar at bottom of viewport
  - Persistent (always visible)
  - Horizontal layout: Copyright (left) | Navigation + Social (right)
  - Height: `48–64px`
- Behavior:
  - `position: fixed`, `bottom: 0`, `z-index: 10`
  - Does not scroll away

#### **\*\*Mobile & Tablet ( $\leq 1023px$ )\*\***

- Layout:
  - Static section in document flow (last element on page)
  - Vertical stack: Copyright → Navigation → Social
  - Never fixed, never overlaps content
- Specific Implementation Details:
  - The footer is the last element on the page, not fixed to bottom
  - Content is center-aligned with generous vertical padding
  - Navigation links are displayed in a single row
  - Social media icons are displayed in a single row below navigation
  - Copyright text is smaller (0.875rem) on mobile

- The blue background extends full width with no container constraints
  - Spacing between elements is increased (32px) for touch accessibility
  - Content Adaptation:
    - The fixed position behavior is removed on mobile
    - The horizontal layout is converted to a vertical stack
    - The navigation and social links are consolidated into fewer rows
    - The height is variable based on content rather than fixed
- 

#### #### Accessibility

- All links have `aria-label` where needed
- Keyboard navigable
- Logical tab order
- Contrast: White on `#0057FF` meets WCAG AA
- Touch targets meet minimum 48x48px

#### #### Non-Negotiables

- Footer is **\*\*fixed only on desktop\*\***
  - Footer is **\*\*not fixed on mobile\*\***
  - No competition with Hero, Manifesto, or CTAs
  - Clean, unobtrusive design
- 

## # \*\*5. TECHNICAL IMPLEMENTATION

### ## 1. Visão Geral da Tecnologia (Tech Stack)

#### ### Frontend Core

- **Framework:** React 18+ (com Hooks)
- **Meta-framework:** Next.js 13+ (App Router)
- **Linguagem:** TypeScript (implícito pela stack moderna)

#### ### Estilização e UI

- **CSS:** Tailwind CSS (apenas classes utilitárias core)
- **Animações de Interface:** Framer Motion
- **3D / WebGL:** React Three Fiber + Three.js

#### ### Infraestrutura e Assets

- **Formulários:** FormSubmit.co
  - **Assets:** Fontes self-hosted; Assets externos via Supabase Storage
  - **Build/Deploy:** Vercel, Netlify ou Cloudflare Pages
- 

### ## 2. Requisitos de Performance e Acessibilidade

#### ### 2.1 Orçamento de Performance (Performance Budget)

- **Peso Inicial da Página:** < 2MB

- **Time to Interactive (TTI):** < 5s (em 3G)
- **First Contentful Paint (FCP):** < 2s
- **Cumulative Layout Shift (CLS):** < 0.1

### ### 2.2 Estratégias de Otimização

- **Imagens:** Formato WebP, lazy loading, `srcset` responsivo.
- **Vídeos:** Comprimidos, autoplay mudo, lazy load (abaixo da dobra).
- **Fontes:** Self-hosted, preloaded, \*subsetting\* se possível.
- **JavaScript:** Code splitting, imports dinâmicos para WebGL.
- **CSS:** Purge de classes Tailwind não utilizadas em produção.
- **WebGL:** Max DPR de 2, antialias desativado.

### ### 2.3 Padrões de Acessibilidade (WCAG 2.1 Level AA)

- **Contraste:** Mínimo 4.5:1 para texto normal.
- **Navegação:** Totalmente operável via teclado (Focus Indicators visíveis).
- **Semântica:** Uso correto de HTML semântico e atributos ARIA.
- **Movimento:** Respeitar a preferência `prefers-reduced-motion`.
- **Mobile:** Áreas de toque mínimas de 48x48px.

---

## ## 3. Especificação de Componentes Críticos

### ### 3.1 Botão CTA (Primary Call-to-Action)

#### **Design Visual:**

- **Formato:** Compósito (Pílula à esquerda + Círculo à direita).
- **Cor:** Azul Primário (`#0048ff`). Texto Branco.
- **Texto:** Uppercase, tracking médio, padding `px-6 py-3`.
- **Ícone:** Seta (→) centralizada no círculo.

#### **Interações e Animações:**

1. **Hover (Desktop):** O botão inteiro sobe 1px (`translateY(-1px)`).
2. **Seta (Opcional):** Desliza 4px para a direita no hover.
3. **Click (Mobile):** Efeito de compressão (`scale(0.98)`).
4. **Foco (Teclado):** Outline de 2px sólido cor `#4fe6ff` com offset de 4px.

#### **Exemplo de Implementação (Framer Motion):**

```
```ts
import { motion } from 'framer-motion';
import { ArrowRight } from 'lucide-react';

export const CTAButton = ({ href, children, variant = 'primary' }) => {
  return (
    <motion.a
      href={href}
      className="inline-flex items-stretch rounded-full bg-[#000022]
        text-white group focus-visible:outline-2
        focus-visible:outline-[#4fe6ff]"
      whileHover={{ y: -1 }}
      whileTap={{ scale: 0.98 }}
      transition={{ duration: 0.2, ease: 'easeOut' }}>
      {children}
    </motion.a>
  );
}
```

```

>
  <span className="px-6 py-3 font-medium uppercase tracking-wide">{children}</span>
  <motion.span
    className="flex items-center justify-center w-12 h-12 rounded-full"
    whileHover={{ x: 4 }}
  >
    <ArrowRight className="w-5 h-5" />
  </motion.span>
</motion.a>
);
};

```

```

### ### 3.2 Atmosfera "Ghost" (WebGL)

**Propósito:** Camada visual atmosférica na seção Hero (decorativo).

**Regras Críticas:**

- \* Carregamento via **Dynamic Import** (apenas client-side).
- \* Fallback para gradiente estático se falhar ou se `prefers-reduced-motion` estiver ativo.
- \* Não deve controlar layout ou bloquear texto.

**Arquitetura de Pastas Sugerida:**

```

```text
components/
  -- canvas/
    |   GhostCanvas.tsx
    |   Ghost.tsx
    |   Particles.tsx
    |   Fireflies.tsx
    |
    |   postprocessing/
```
```
---
```

4. Estrutura de Conteúdo e Páginas

4.1 Inventário de Conteúdo

1. **Hero:** Manchete editorial, submanchete, CTA principal, Atmosfera WebGL.
2. **Portfolio Showcase:** Título, lista de categorias, labels flutuantes, CTA.
3. **Featured Projects:** Grid Bento, metadados dos projetos.
4. **Clients/Brands:** Grid de logotipos.
5. **Contact:** Formulário (2 colunas desktop, 1 mobile), info de contato, sociais.
6. **Footer:** Copyright 2025, navegação.

4.2 Comportamento Responsivo

- * **Header:**
- * Desktop: Efeito "Glass" fluido.
- * Mobile: Menu "Hambúrguer" com animação escalonada (staggered).

* **Footer:**

- * Desktop: Fixo na base (se o design pedir reveal) ou estático.
- * Mobile: Sempre estático no fluxo do documento (nunca fixo).

5. Build, Deploy e Variáveis

Comando de Build: `pnpm run build`

Variáveis de Ambiente (`.env.local`):

```
```bash
NEXT_PUBLIC_GA_ID=UA-XXXXXXXXXX-X
NEXT_PUBLIC_SUPABASE_URL=
[https://aymuvxysygrwoicsjgxj.supabase.co](https://aymuvxysygrwoicsjgxj
.supabase.co)
NEXT_PUBLIC_SUPABASE_ANON_KEY=<anon_key>
````
```

Checklist Pré-Deployment:

- * [] Auditoria Lighthouse (Score > 90).
- * [] Teste em dispositivos móveis reais (iOS/Android).
- * [] Validação de envio de formulários.
- * [] Verificação de links externos (abrir em nova aba).
- * [] Teste de vídeos (autoplay mudo).
- * [] Teste de `prefers-reduced-motion`.

6. Checklist de Garantia de Qualidade (QA)

Visual

- * [] **Hero:** Texto legível sobre o WebGL.
- * [] **Manifesto:** Vídeo inicia pequeno e expande no scroll.
- * [] **Projetos:** Grid Bento proporcional no desktop, pilha vertical no mobile.
- * [] **Logos:** Grid adapta colunas (2-3 no mobile).

Funcionalidade

- * [] Links de navegação (âncoras e rotas) funcionais.

- * [] Formulário envia dados corretamente.
- * [] Ícones sociais abrem em novas abas.
- * [] Sem erros no console do navegador.
- * [] Sem scroll horizontal indesejado no mobile.

Performance & Acessibilidade

- * [] Carregamento < 3s em 3G.
- * [] Navegação completa via Tab (Teclado).
- * [] Screen Readers leem o conteúdo corretamente.
- * [] CLS < 0.1 (Layout estável).

```

---