

📄 GUÍA DE IMPLEMENTACIÓN DEFINITIVA - ANCLORA PRESS UI FASE 1

🎯 OBJETIVO: Implementar Fase 1 con Guía de Estilos Anclora Metaform Oficial

PRERREQUISITOS

- Repositorio: `https://github.com/TonilAPro73/AncloraPress-ui`
- Node.js 16+ instalado
- Git configurado

FASE A: PREPARACIÓN DEL ENTORNO

A1. Clonar y Preparar Repositorio

```
```bash
```

```
Clonar repositorio (si no lo tienes)
```

```
git clone https://github.com/TonilAPro73/AncloraPress-ui.git
```

```
cd AncloraPress-ui
```

```
Crear rama para la implementación
```

```
git checkout -b feat/fase1-fundacion-tecnica-ancora
```

```
Instalar dependencias existentes
```

```
npm install
```

```
Instalar nuevas dependencias necesarias
```

```
npm install lucide-react @tailwindcss/forms @tailwindcss/typography
```

```
```
```

A2. Estructura de Carpetas Base

```
```bash
```

```
Crear estructura según guía Anclora
```

```
mkdir -p styles/base
```

```
mkdir -p styles/components
```

```
mkdir -p styles/layout
```

```
mkdir -p styles/utilities
```

```
mkdir -p components/ui
```

```
mkdir -p components/core
```

```
mkdir -p components/layout
```

```
mkdir -p lib
```

```
```
```

FASE B: CONFIGURACIÓN TAILWIND CON GUÍA ANCLORA

B1. Configurar tailwind.config.js

```
```bash
```

```
Hacer backup de configuración existente
cp tailwind.config.js tailwind.config.js.backup
``
```

```
Reemplazar `tailwind.config.js` con:
```

```
```javascript
```

```
/** @type {import('tailwindcss').Config} */
```

```
module.exports = {
```

```
  content: [
```

```
    './pages/**/*.js,ts,jsx,tsx,mdx',
```

```
    './components/**/*.js,ts,jsx,tsx,mdx',
```

```
    './app/**/*.js,ts,jsx,tsx,mdx',
```

```
    './src/**/*.js,ts,jsx,tsx,mdx',
```

```
  ],
```

```
  theme: {
```

```
    extend: {
```

```
      // Colores exactos de la guía Anclora Metaform
```

```
      colors: {
```

```
        primary: {
```

```
          DEFAULT: '#006EE6', // Azul Anclora
```

```
          dark: '#0050A7', // Azul Profundo
```

```
        },
```

```
        secondary: {
```

```
          DEFAULT: '#00B8D9', // Cian Anclora
```

```
          dark: '#00829B', // Cian Profundo
```

```
        },
```

```
        success: '#28A745', // Verde Éxito
```

```
        warning: '#FFC107', // Ámbar Advertencia
```

```
        danger: '#DC3545', // Rojo Error
```

```
        gray: {
```

```
          100: '#F5F7FA', // Gris Claro
```

```
          200: '#E4E7EB', // Gris Medio
```

```
          400: '#8492A6', // Gris
```

```
          700: '#3E4C59', // Gris Oscuro
```

```
          900: '#1F2933', // Negro Anclora
```

```
        },
```

```
        white: '#FFFFFF',
```

```
      },
```

```
      // Tipografía Inter según guía
```

```
      fontFamily: {
```

```
        sans: ['Inter', '-apple-system', 'BlinkMacSystemFont', 'Segoe UI', 'Roboto', 'Oxygen',  
'Ubuntu', 'Cantarell', 'Open Sans', 'Helvetica Neue', 'sans-serif'],
```

```
      },
```

```
      // Escalas tipográficas exactas
```

```
      fontSize: {
```

```
        'h1': ['2rem', { lineHeight: '1.2', fontWeight: '700' }], // 32px
```

```

'h2': ['1.75rem', { lineHeight: '1.2', fontWeight: '700' }], // 28px
'h3': ['1.5rem', { lineHeight: '1.3', fontWeight: '600' }], // 24px
'h4': ['1.25rem', { lineHeight: '1.3', fontWeight: '600' }], // 20px
'subtitle': ['1.125rem', { lineHeight: '1.4', fontWeight: '500' }], // 18px
'body': ['1rem', { lineHeight: '1.5', fontWeight: '400' }], // 16px
'small': ['0.875rem', { lineHeight: '1.5', fontWeight: '400' }], // 14px
'micro': ['0.75rem', { lineHeight: '1.5', fontWeight: '400' }], // 12px
},

// Sistema de espaciado Anclora (múltiplos de 4px)
spacing: {
  '1': '4px', // --space-1
  '2': '8px', // --space-2
  '3': '12px', // --space-3
  '4': '16px', // --space-4
  '5': '24px', // --space-5
  '6': '32px', // --space-6
  '7': '48px', // --space-7
  '8': '64px', // --space-8
},

// Bordes redondeados según guía
borderRadius: {
  DEFAULT: '6px',
  'lg': '8px',
},

// Sombras exactas de la guía
boxShadow: {
  'sm': '0 1px 2px 0 rgba(0, 0, 0, 0.05)',
  DEFAULT: '0 2px 8px 0 rgba(0, 0, 0, 0.08)',
  'md': '0 4px 16px 0 rgba(0, 0, 0, 0.1)',
  'lg': '0 10px 25px 0 rgba(0, 0, 0, 0.15)',
},

// Breakpoints de la guía
screens: {
  'xs': '0px',
  'sm': '640px',
  'md': '768px',
  'lg': '1024px',
  'xl': '1280px',
  '2xl': '1536px',
},
},
},
plugins: [
  require('@tailwindcss/forms'),

```

```
    require('@tailwindcss/typography'),
  ],
}
```

FASE C: ARCHIVOS CSS BASE SEGÚN GUÍA ANCLORA

C1. Crear `styles/base/_variables.css`

```
``css
/* Variables CSS Anclora Metaform - Guía Oficial */
:root {
  /* Colores primarios */
  --color-primary: #006EE6;
  --color-secondary: #00B8D9;

  /* Colores secundarios */
  --color-primary-dark: #0050A7;
  --color-secondary-dark: #00829B;

  /* Colores de estado */
  --color-success: #28A745;
  --color-warning: #FFC107;
  --color-danger: #DC3545;

  /* Colores neutros */
  --color-white: #FFFFFF;
  --color-gray-100: #F5F7FA;
  --color-gray-200: #E4E7EB;
  --color-gray-400: #8492A6;
  --color-gray-700: #3E4C59;
  --color-gray-900: #1F2933;

  /* Tipografía */
  --font-primary: 'Inter', -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen,
  Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif;

  /* Tamaños de fuente - Escritorio */
  --font-size-h1: 2rem; /* 32px */
  --font-size-h2: 1.75rem; /* 28px */
  --font-size-h3: 1.5rem; /* 24px */
  --font-size-h4: 1.25rem; /* 20px */
  --font-size-subtitle: 1.125rem; /* 18px */
  --font-size-body: 1rem; /* 16px */
  --font-size-small: 0.875rem; /* 14px */
  --font-size-micro: 0.75rem; /* 12px */
}
```

```

/* Pesos de fuente */
--font-weight-regular: 400;
--font-weight-medium: 500;
--font-weight-semibold: 600;
--font-weight-bold: 700;

/* Line heights */
--line-height-tight: 1.2;
--line-height-snug: 1.3;
--line-height-normal: 1.4;
--line-height-relaxed: 1.5;

/* Sistema de espaciado (múltiplos de 4px) */
--space-1: 4px; /* Espaciado mínimo */
--space-2: 8px; /* Espaciado estándar */
--space-3: 12px; /* Espaciado medio */
--space-4: 16px; /* Espaciado base */
--space-5: 24px; /* Espaciado amplio */
--space-6: 32px; /* Espaciado grande */
--space-7: 48px; /* Espaciado muy grande */
--space-8: 64px; /* Espaciado máximo */
}

/* Media query para móviles */
@media (max-width: 768px) {
  :root {
    --font-size-h1: 1.75rem; /* 28px */
    --font-size-h2: 1.5rem; /* 24px */
    --font-size-h3: 1.25rem; /* 20px */
    --font-size-h4: 1.125rem; /* 18px */
    --font-size-subtitle: 1rem; /* 16px */
  }
}
...

### **C2. Crear `styles/base/_typography.css`**
```css
/* Tipografía Anclora Metaform - Guía Oficial */

/* Importar fuente Inter */
@import
url('https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&display=swap');

body {
 font-family: var(--font-primary);
 font-size: var(--font-size-body);
 font-weight: var(--font-weight-regular);
 line-height: var(--line-height-relaxed);

```

```
color: var(--color-gray-900);
background-color: var(--color-white);
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;
}
```

```
/* Clases utilitarias según guía */
```

```
.text-h1 {
 font-size: var(--font-size-h1);
 font-weight: var(--font-weight-bold);
 line-height: var(--line-height-tight);
 color: var(--color-gray-900);
}
```

```
.text-h2 {
 font-size: var(--font-size-h2);
 font-weight: var(--font-weight-bold);
 line-height: var(--line-height-tight);
 color: var(--color-gray-900);
}
```

```
.text-h3 {
 font-size: var(--font-size-h3);
 font-weight: var(--font-weight-semibold);
 line-height: var(--line-height-snug);
 color: var(--color-gray-900);
}
```

```
.text-h4 {
 font-size: var(--font-size-h4);
 font-weight: var(--font-weight-semibold);
 line-height: var(--line-height-snug);
 color: var(--color-gray-900);
}
```

```
.text-subtitle {
 font-size: var(--font-size-subtitle);
 font-weight: var(--font-weight-medium);
 line-height: var(--line-height-normal);
 color: var(--color-gray-700);
}
```

```
.text-body {
 font-size: var(--font-size-body);
 font-weight: var(--font-weight-regular);
 line-height: var(--line-height-relaxed);
 color: var(--color-gray-900);
}
```

```
.text-small {
 font-size: var(--font-size-small);
 font-weight: var(--font-weight-regular);
 line-height: var(--line-height-relaxed);
 color: var(--color-gray-700);
}
```

```
.text-micro {
 font-size: var(--font-size-micro);
 font-weight: var(--font-weight-regular);
 line-height: var(--line-height-relaxed);
 color: var(--color-gray-400);
}
...
```

### \*\*C3. Reemplazar `styles/globals.css`\*\*

```
```css
```

```
/* Estilos principales AncloraPress - Guía Anclora Metaform */
```

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

```
/* Importar estilos base */
```

```
@import './base/_variables.css';
```

```
@import './base/_typography.css';
```

```
@layer base {
  html {
    scroll-behavior: smooth;
  }
}
```

```
body {
  font-family: var(--font-primary);
  background-color: var(--color-gray-100);
}
```

```
/* Enfoque accesible según guía Anclora */
```

```
:focus {
  outline: 2px solid var(--color-primary);
  outline-offset: 2px;
}
```

```
/* Enlaces */
```

```
a {
  color: var(--color-primary);
  text-decoration: none;
```

```

    transition: all 0.2s ease;
}

a:hover {
    color: var(--color-primary-dark);
    text-decoration: underline;
}
}

@layer components {
/* Sistema de contenedores Anclora */
.container {
    width: 100%;
    max-width: 1200px;
    margin: 0 auto;
    padding: 0 var(--space-4);
}

/* Botones base según guía */
.btn {
    display: inline-flex;
    align-items: center;
    justify-content: center;
    padding: var(--space-2) var(--space-4);
    border-radius: 6px;
    font-weight: var(--font-weight-medium);
    font-size: var(--font-size-body);
    line-height: var(--line-height-normal);
    transition: all 0.2s ease;
    cursor: pointer;
    border: none;
}

.btn-primary {
    background-color: var(--color-primary);
    color: var(--color-white);
}

.btn-primary:hover {
    background-color: var(--color-primary-dark);
}

.btn-secondary {
    background-color: transparent;
    color: var(--color-primary);
    border: 1px solid var(--color-primary);
}
}

```



```
.btn-secondary:hover {  
  background-color: rgba(0, 110, 230, 0.1);  
}
```

```
.btn:disabled {  
  opacity: 0.5;  
  cursor: not-allowed;  
  pointer-events: none;  
}
```

/* Cards según guía */

```
.card {  
  background-color: var(--color-white);  
  border-radius: 8px;  
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.08);  
  overflow: hidden;  
}
```

```
.card-header {  
  padding: var(--space-4);  
  border-bottom: 1px solid var(--color-gray-200);  
}
```

```
.card-body {  
  padding: var(--space-4);  
}
```

```
.card-footer {  
  padding: var(--space-4);  
  border-top: 1px solid var(--color-gray-200);  
  background-color: var(--color-gray-100);  
}
```

```
.card-title {  
  margin: 0;  
  font-size: var(--font-size-h4);  
  font-weight: var(--font-weight-semibold);  
  color: var(--color-gray-900);  
}
```

/* Inputs según guía */

```
.input {  
  width: 100%;  
  padding: var(--space-3);  
  border: 1px solid var(--color-gray-200);  
  border-radius: 6px;  
  font-size: var(--font-size-body);  
  line-height: var(--line-height-normal);  
}
```

```

    color: var(--color-gray-900);
    transition: all 0.2s ease;
  }

  .input:hover {
    border-color: var(--color-gray-400);
  }

  .input:focus {
    outline: none;
    border-color: var(--color-primary);
    box-shadow: 0 0 0 3px rgba(0, 110, 230, 0.15);
  }
}

/* Accesibilidad - Reducción de movimiento */
@media (prefers-reduced-motion: reduce) {
  * {
    animation-duration: 0.01ms !important;
    animation-iteration-count: 1 !important;
    transition-duration: 0.01ms !important;
    scroll-behavior: auto !important;
  }
}
...

---

## **FASE D: COMPONENTES UI SEGÚN GUÍA ANCLORA**

### **D1. Crear `lib/utils.js`**
```javascript
// Función utilitaria para combinar clases CSS
export function cn(...classes) {
 return classes.filter(Boolean).join(' ');
}
...

D2. Crear `components/ui/Button.jsx`
```javascript
import React from 'react';
import { Loader2 } from 'lucide-react';
import { cn } from '@lib/utils';

const Button = React.forwardRef(({
  className,
  variant = 'primary',
  size = 'default',

```

```

loading = false,
disabled = false,
children,
...props
}, ref) => {

  // Variantes según guía Anclora
  const variants = {
    primary: 'bg-primary text-white hover:bg-primary-dark focus:ring-primary/20',
    secondary: 'bg-transparent text-primary border border-primary hover:bg-primary/10
focus:ring-primary/20',
    tertiary: 'bg-transparent text-gray-700 hover:bg-gray-100 focus:ring-gray-400/20',
    danger: 'bg-danger text-white hover:bg-red-700 focus:ring-danger/20',
    success: 'bg-success text-white hover:bg-green-700 focus:ring-success/20'
  };

  // Tamaños según guía
  const sizes = {
    sm: 'px-3 py-1.5 text-small',
    default: 'px-4 py-2 text-body',
    lg: 'px-6 py-3 text-subtitle'
  };

  const baseClasses = 'inline-flex items-center justify-center rounded-md font-medium
transition-all duration-200 focus:outline-none focus:ring-2 focus:ring-offset-2
disabled:opacity-50 disabled:cursor-not-allowed';

  return (
    <button
      ref={ref}
      className={cn(
        baseClasses,
        variants[variant],
        sizes[size],
        loading && 'relative text-transparent',
        className
      )}
      disabled={disabled || loading}
      {...props}
    >
      {loading && (
        <div className="absolute inset-0 flex items-center justify-center">
          <Loader2 className="h-4 w-4 animate-spin text-current" />
        </div>
      )}
      {children}
    </button>
  );

```

```
});
```

```
Button.displayName = 'Button';
```

```
export { Button };
```

```
...
```

```
### **D3. Crear `components/ui/Card.jsx`**
```

```
```javascript
```

```
import React from 'react';
```

```
import { cn } from '@lib/Utils';
```

```
const Card = React.forwardRef(({ className, ...props }, ref) => (
```

```
 <div
```

```
 ref={ref}
```

```
 className={cn('card', className)}
```

```
 {...props}
```

```
 />
```

```
));
```

```
Card.displayName = 'Card';
```

```
const CardHeader = React.forwardRef(({ className, ...props }, ref) => (
```

```
 <div
```

```
 ref={ref}
```

```
 className={cn('card-header', className)}
```

```
 {...props}
```

```
 />
```

```
));
```

```
CardHeader.displayName = 'CardHeader';
```

```
const CardTitle = React.forwardRef(({ className, children, ...props }, ref) => (
```

```
 <h3
```

```
 ref={ref}
```

```
 className={cn('card-title', className)}
```

```
 {...props}
```

```
 >
```

```
 {children}
```

```
 </h3>
```

```
));
```

```
CardTitle.displayName = 'CardTitle';
```

```
const CardBody = React.forwardRef(({ className, ...props }, ref) => (
```

```
 <div
```

```
 ref={ref}
```

```
 className={cn('card-body', className)}
```

```
 {...props}
```

```
 />
```

```
));
```

```

CardBody.displayName = 'CardBody';

const CardFooter = React.forwardRef(({ className, ...props }, ref) => (
 <div
 ref={ref}
 className={cn('card-footer', className)}
 {...props}
 />
));
CardFooter.displayName = 'CardFooter';

export { Card, CardHeader, CardTitle, CardBody, CardFooter };
...

```

```

D4. Crear `components/ui/index.js`
```javascript
export { Button } from './Button';
export { Card, CardHeader, CardTitle, CardBody, CardFooter } from './Card';
...

```

```
---
```

FASE E: APLICACIÓN PRINCIPAL ANCLORAPRESS

```

### **E1. Crear `components/AncloraPressApp.jsx`**
```javascript
import React, { useState, useCallback, useRef } from 'react';
import { Upload, FileText, Settings, Eye, Download, Zap, CheckCircle, AlertCircle, Loader2,
Save } from 'lucide-react';
import { Button } from '@components/ui/Button';
import { Card, CardHeader, CardTitle, CardBody, CardFooter } from '@components/ui/Card';

```

```

// Utilidad para simular procesamiento IA
const simulateAIProcessing = (duration) => {
 return new Promise(resolve => setTimeout(resolve, duration));
};

```

```

const AncloraPressApp = () => {
 const [activeStep, setActiveStep] = useState(1);
 const [uploadedFile, setUploadedFile] = useState(null);
 const [processingState, setProcessingState] = useState('idle');
 const [conversionProgress, setConversionProgress] = useState(0);
 const [editorContent, setEditorContent] = useState("");
 const [previewFormat, setPreviewFormat] = useState('epub');
 const [outputFormats, setOutputFormats] = useState({
 epub: true,
 mobi: true,
 pdf: true
 });

```

```
});
```

```
const steps = [
 { id: 1, title: 'Importar Contenido', icon: Upload },
 { id: 2, title: 'Editor Inteligente', icon: FileText },
 { id: 3, title: 'Vista Previa', icon: Eye },
 { id: 4, title: 'Exportar', icon: Download }
];
```

```
const supportedFormats = [
 { type: 'word', name: 'Microsoft Word', extensions: ['.docx', '.doc'], icon: '📄' },
 { type: 'gdocs', name: 'Google Docs', extensions: ['.gdoc'], icon: '📝' },
 { type: 'notion', name: 'Notion Export', extensions: ['.zip'], icon: '📁' },
 { type: 'markdown', name: 'Markdown', extensions: ['.md', '.markdown'], icon: '📖' },
 { type: 'text', name: 'Texto Plano', extensions: ['.txt'], icon: '📄' }
];
```

```
// Componente de barra de progreso
```

```
const ProgressBar = ({ progress, status = 'processing' }) => {
 const statusColors = {
 processing: 'bg-primary',
 completed: 'bg-success',
 error: 'bg-danger'
 };
};
```

```
return (
 <div className="w-full bg-gray-200 rounded-full h-2 mb-4">
 <div
 className={`h-2 rounded-full transition-all duration-500 ease-out
${statusColors[status]} }`
 style={{ width: `${progress}%` }}
 />
 </div>
);
};
```

```
// Componente de subida de archivos
```

```
const AIFileUploader = () => {
 const fileInputRef = useRef(null);

 const handleFileUpload = useCallback(async (file) => {
 setUploadedFile(file);
 setProcessingState('processing');
 setConversionProgress(0);

 const progressSteps = [
 { progress: 20, message: 'Analizando estructura del documento...' },
 { progress: 40, message: 'Optimizando formato con IA...' },
]
```

```

 { progress: 60, message: 'Procesando elementos multimedia...' },
 { progress: 80, message: 'Generando metadatos inteligentes...' },
 { progress: 100, message: 'Conversión completada exitosamente' }
];

 for (const step of progressSteps) {
 await simulateAIProcessing(800);
 setConversionProgress(step.progress);
 }

 setProcessingState('completed');
 setEditorContent(`# ${file.name.split('.')[0]}\n\nContenido procesado automáticamente
por IA...\n\nEste es un ejemplo de cómo el contenido se estructura automáticamente
después del procesamiento inteligente.`);
}, []);

const handleDrop = useCallback((e) => {
 e.preventDefault();
 const files = Array.from(e.dataTransfer.files);
 if (files.length > 0) {
 handleFileUpload(files[0]);
 }
}, [handleFileUpload]);

const handleDragOver = useCallback((e) => {
 e.preventDefault();
}, []);

return (
 <div className="space-y-6">
 <div
 className="border-2 border-dashed border-gray-200 rounded-lg p-8 text-center
hover:border-primary/50 transition-colors cursor-pointer"
 onDrop={handleDrop}
 onDragOver={handleDragOver}
 onClick={() => fileInputRef.current?.click()}
 >
 <input
 ref={fileInputRef}
 type="file"
 className="hidden"
 accept=".docx,.doc,.md,.txt,.zip"
 onChange={(e) => e.target.files?.[0] && handleFileUpload(e.target.files[0])}
 />

 <div className="flex flex-col items-center space-y-4">
 <div className="w-16 h-16 bg-primary/10 rounded-full flex items-center
justify-center">

```

```

 <Upload className="w-8 h-8 text-primary" />
 </div>
 <div>
 <h3 className="text-h3 text-gray-900 mb-2">
 Arrastra tu archivo aquí o haz clic para seleccionar
 </h3>
 <p className="text-body text-gray-400">
 Admitimos Word, Google Docs, Notion, Markdown y más
 </p>
 </div>
</div>
</div>

{/* Formatos soportados */}
<div className="grid grid-cols-2 md:grid-cols-5 gap-4">
 {supportedFormats.map((format) => (
 <Card key={format.type} className="text-center p-3 hover:shadow-md
transition-shadow">
 <div className="text-2xl mb-2">{format.icon}</div>
 <div className="text-small font-medium text-gray-900">{format.name}</div>
 <div className="text-micro text-gray-400">
 {format.extensions.join(', ')}
 </div>
 </Card>
))}
</div>

{/* Estado de procesamiento */}
{processingState === 'processing' && (
 <Card className="border-primary/20 bg-primary/5">
 <CardBody>
 <div className="flex items-center space-x-3 mb-3">
 <Loader2 className="w-5 h-5 text-primary animate-spin" />

 Procesando con IA Avanzada...

 </div>
 <ProgressBar progress={conversionProgress} status="processing" />
 <p className="text-small text-primary">
 Optimizando formato y estructura automáticamente
 </p>
 </CardBody>
 </Card>
)}

{/* Estado completado */}
{processingState === 'completed' && uploadedFile && (
 <Card className="border-success/20 bg-success/5">

```



```

 <CardBody>
 <div className="flex items-center space-x-3">
 <CheckCircle className="w-5 h-5 text-success" />
 <div>
 <div className="font-medium text-success">
 ¡Archivo procesado exitosamente!
 </div>
 <div className="text-small text-success/80">
 {uploadedFile.name} - Listo para editar
 </div>
 </div>
 </div>
 </CardBody>
 </Card>
)}
</div>
);
};

// Editor inteligente
const IntelligentEditor = () => {
 const [editorMode, setEditorMode] = useState('visual');
 const [aiSuggestions, setAiSuggestions] = useState(false);

 return (
 <div className="space-y-4">
 {/* Toolbar */}
 <Card>
 <CardBody className="flex items-center justify-between">
 <div className="flex items-center space-x-2">
 <Button
 variant={editorMode === 'visual' ? 'primary' : 'secondary'}
 size="sm"
 onClick={() => setEditorMode('visual')}
 >
 Visual
 </Button>
 <Button
 variant={editorMode === 'code' ? 'primary' : 'secondary'}
 size="sm"
 onClick={() => setEditorMode('code')}
 >
 Código
 </Button>
 </div>

 <Button
 variant={aiSuggestions ? 'primary' : 'tertiary'}

```

```

 size="sm"
 onClick={() => setAiSuggestions(!aiSuggestions)}
 >
 <Zap className="w-4 h-4 mr-2" />
 IA
 </Button>
 </CardBody>
 </Card>

 {/ Editor */}
 <div className="grid grid-cols-1 lg:grid-cols-3 gap-6">
 <div className="lg:col-span-2">
 <Card>
 <CardBody className="p-0">
 <textarea
 value={editorContent}
 onChange={(e) => setEditorContent(e.target.value)}
 className="w-full h-96 p-6 border-none outline-none resize-none text-body"
 placeholder="Comienza a escribir tu contenido aquí..."
 />
 </CardBody>
 </Card>
 </div>

 {/ Panel lateral */}
 <div className="space-y-4">
 {aiSuggestions && (
 <Card className="border-primary/20 bg-primary/5">
 <CardHeader>
 <CardTitle className="text-h4 text-primary">Sugerencias IA</CardTitle>
 </CardHeader>
 <CardBody className="space-y-2">
 <Button variant="tertiary" size="sm" className="w-full justify-start">
 Mejorar estructura del capítulo
 </Button>
 <Button variant="tertiary" size="sm" className="w-full justify-start">
 Optimizar para legibilidad
 </Button>
 <Button variant="tertiary" size="sm" className="w-full justify-start">
 Añadir elementos visuales
 </Button>
 </CardBody>
 </Card>
)}

 <Card>
 <CardHeader>
 <CardTitle className="text-h4">Estadísticas</CardTitle>

```

```

 </CardHeader>
 <CardBody className="space-y-1 text-small text-gray-700">
 <div>Palabras: {editorContent.split(' ').filter(w => w).length}</div>
 <div>Caracteres: {editorContent.length}</div>
 <div>Páginas estimadas: {Math.ceil(editorContent.length / 2000)}</div>
 </CardBody>
 </Card>
</div>
</div>
</div>
);
};

```

// Vista previa

```

const PreviewPanel = () => {
 const previewOptions = [
 { id: 'epub', name: 'EPUB3', icon: '📖' },
 { id: 'mobi', name: 'MOBI/Kindle', icon: '📱' },
 { id: 'pdf', name: 'PDF', icon: '📄' }
];

```

```

 return (
 <div className="space-y-4">
 <div className="flex space-x-2">
 {previewOptions.map((option) => (
 <Button
 key={option.id}
 variant={previewFormat === option.id ? 'primary' : 'secondary'}
 onClick={() => setPreviewFormat(option.id)}
 className="flex items-center space-x-2"
 >
 {option.icon}
 {option.name}
 </Button>
))}
 </div>

```

```

 <Card>
 <CardHeader>

 Vista previa - {previewOptions.find(o => o.id === previewFormat)?.name}

 </CardHeader>
 <CardBody className="min-h-96">
 <div className="max-w-md mx-auto bg-white shadow-lg rounded-lg
overflow-hidden">
 <div className="bg-gradient-to-r from-primary to-secondary h-32 flex items-center
justify-center">

```

```

 <h1 className="text-white text-subtitle font-bold">Título del Libro</h1>
 </div>
 <div className="p-4 space-y-3">
 <div className="h-4 bg-gray-200 rounded w-3/4"></div>
 <div className="h-4 bg-gray-200 rounded w-full"></div>
 <div className="h-4 bg-gray-200 rounded w-2/3"></div>
 <div className="h-32 bg-gray-100 rounded flex items-center justify-center">
 Contenido del libro
 </div>
 </div>
 </div>
 </CardBody>
</Card>
</div>
);
};

```

// Panel de exportación

```

const ExportPanel = () => {
 const [isExporting, setIsExporting] = useState(false);
 const [exportProgress, setExportProgress] = useState(0);

```

```

 const handleExport = async () => {
 setIsExporting(true);
 setExportProgress(0);

```

```

 const exportSteps = [
 { progress: 25, message: 'Generando EPUB3...' },
 { progress: 50, message: 'Creando archivo MOBI...' },
 { progress: 75, message: 'Optimizando PDF...' },
 { progress: 100, message: 'Exportación completada' }
];

```

```

 for (const step of exportSteps) {
 await simulateAIProcessing(1000);
 setExportProgress(step.progress);
 }

```

```

 setIsExporting(false);
 };

```

```

 const enabledFormats = Object.entries(outputFormats)
 .filter(([_, enabled]) => enabled)
 .map(([format, _]) => format);

```

```

 return (
 <div className="space-y-6">
 <Card>

```

```

<CardHeader>
 <CardTitle className="text-h3">Listo para Exportar</CardTitle>
</CardHeader>

<CardBody>
 <div className="grid grid-cols-1 md:grid-cols-2 gap-4 mb-6">
 <div>
 <h4 className="font-medium text-gray-700 mb-2">Formatos
seleccionados:</h4>
 <div className="space-y-1">
 {enabledFormats.map(format => (
 <div key={format} className="flex items-center space-x-2">
 <CheckCircle className="w-4 h-4 text-success" />
 {format.toUpperCase()}
 </div>
))}
 </div>
 </div>

 <div>
 <h4 className="font-medium text-gray-700 mb-2">Estadísticas:</h4>
 <div className="space-y-1 text-small text-gray-700">
 <div>Páginas: {Math.ceil(editorContent.length / 2000)}</div>
 <div>Palabras: {editorContent.split(' ').filter(w => w).length}</div>
 <div>Tamaño estimado: {Math.ceil(editorContent.length / 1024)} KB</div>
 </div>
 </div>
 </div>

 <Button
 onClick={handleExport}
 disabled={isExporting || enabledFormats.length === 0}
 loading={isExporting}
 className="w-full"
 >
 <Download className="w-5 h-5 mr-2" />
 Exportar Libros ({enabledFormats.length} formatos)
 </Button>
</CardBody>
</Card>

{isExporting && (
 <Card className="border-primary/20 bg-primary/5">
 <CardBody>
 <ProgressBar progress={exportProgress} status="processing" />
 <p className="text-small text-primary">
 Generando archivos optimizados con IA...
 </p>
 </CardBody>
 </Card>
)}

```

```

 </CardBody>
 </Card>
 })
 </div>
);
};

const renderActiveStep = () => {
 switch (activeStep) {
 case 1: return <AIFileUploader />;
 case 2: return <IntelligentEditor />;
 case 3: return <PreviewPanel />;
 case 4: return <ExportPanel />;
 default: return <AIFileUploader />;
 }
};

return (
 <div className="min-h-screen bg-gray-100">
 { /* Header */ }
 <header className="bg-white shadow border-b border-gray-200">
 <div className="container">
 <div className="flex items-center justify-between h-16">
 <div className="flex items-center space-x-3">
 <div className="w-8 h-8 bg-primary rounded-lg flex items-center justify-center">
 <FileText className="w-5 h-5 text-white" />
 </div>
 <h1 className="text-subtitle font-bold text-gray-900">AncloraPress</h1>

 Fase 1: Fundación Técnica

 </div>
 </div>

 <div className="flex items-center space-x-4">
 <Button variant="tertiary" size="sm">
 <Settings className="w-4 h-4 mr-2" />
 Configuración
 </Button>
 <Button variant="tertiary" size="sm">
 <Save className="w-4 h-4 mr-2" />
 Guardar
 </Button>
 </div>
 </div>
 </div>
 </div>
 </div>
 </div>
 </div>
 { /* Navegación de pasos */ }

```

```

<div className="bg-white border-b border-gray-200">
 <div className="container">
 <nav className="flex space-x-8 py-4">
 {steps.map((step) => {
 const Icon = step.icon;
 const isActive = activeStep === step.id;
 const isCompleted = step.id < activeStep || (step.id === 1 && processingState ===
'completed');

 return (
 <button
 key={step.id}
 onClick={() => setActiveStep(step.id)}
 className={`flex items-center space-x-2 px-3 py-2 rounded-lg transition-all ${
 isActive
 ? 'bg-primary/10 text-primary border-2 border-primary/30'
 : isCompleted
 ? 'text-success hover:bg-success/5'
 : 'text-gray-700 hover:bg-gray-100'
 }`}
 >
 {isCompleted ? (
 <CheckCircle className="w-5 h-5" />
) : (
 <Icon className="w-5 h-5" />
)}
 {step.title}
 </button>
);
 })}
 </nav>
 </div>

 {/* Contenido principal */}
 <main className="container py-8">
 <Card>
 <CardBody className="p-6">
 {renderActiveStep()}
 </CardBody>
 </Card>
 </main>

 {/* Footer */}
 <footer className="bg-gray-900 text-white py-6">
 <div className="container">
 <div className="grid grid-cols-1 md:grid-cols-3 gap-6">
 <div>

```

```

 <h4 className="font-semibold mb-2">🚀 Conversor IA Universal</h4>
 <p className="text-small text-gray-400">
 Procesa Word, Google Docs, Notion y Markdown en menos de 5 minutos
 </p>
 </div>
 <div>
 <h4 className="font-semibold mb-2">✨ Editor Inteligente</h4>
 <p className="text-small text-gray-400">
 Corrección automática y optimización con sugerencias de IA
 </p>
 </div>
 <div>
 <h4 className="font-semibold mb-2">📖 Multi-formato</h4>
 <p className="text-small text-gray-400">
 Exporta a EPUB3, MOBI y PDF optimizados automáticamente
 </p>
 </div>
</div>

<div className="border-t border-gray-700 mt-6 pt-4 text-center">
 <p className="text-small text-gray-400">
 AncloPress - Democratizando la creación de e-books con IA • Fase 1: Fundación
 Técnica
 </p>
</div>
</div>
</footer>
</div>
);
};

export default AncloPressApp;
...

FASE F: CONFIGURACIÓN DE RUTAS Y IMPORTS

F1. Configurar imports absolutos

Crear o actualizar `jsconfig.json` (JavaScript):
```json
{
  "compilerOptions": {
    "baseUrl": ".",
    "paths": {
      "@/*": ["/.*"],
      "@components/*": ["/components/*"],

```



```

    "@/lib/*": ["/lib/*"],
    "@/styles/*": ["/styles/*"]
  }
}
}
...

```

****O actualizar `tsconfig.json` (TypeScript):****

```

```json
{
 "compilerOptions": {
 "baseUrl": ".",
 "paths": {
 "@/*": ["/"],
 "@components/*": ["/components/*"],
 "@lib/*": ["/lib/*"],
 "@styles/*": ["/styles/*"]
 }
 }
}
}
...

```

**### \*\*F2. Crear página del editor\*\***

**\*\*Para Pages Router - crear `pages/editor.js`:\*\***

```

```javascript
import AncloraPressApp from '@components/AncloraPressApp';

export default function EditorPage() {
  return <AncloraPressApp />;
}
...

```

****Para App Router - crear `app/editor/page.js`:****

```

```javascript
import AncloraPressApp from '@components/AncloraPressApp';

export default function EditorPage() {
 return <AncloraPressApp />;
}
...

```

**### \*\*F3. Actualizar importación de estilos\*\***

**\*\*Para Pages Router - actualizar `pages/\_app.js`:\*\***

```

```javascript
import '@styles/globals.css'

```

```
export default function App({ Component, pageProps }) {  
  return <Component {...pageProps} />  
}
```

****Para App Router - actualizar `app/layout.js`:****

```
``javascript  
import '@styles/globals.css'  
  
export default function RootLayout({ children }) {  
  return (  
    <html lang="es">  
      <body>{children}</body>  
    </html>  
  )  
}
```

**FASE G: TESTING Y VERIFICACIÓN**

**G1. Ejecutar servidor de desarrollo**

```
``bash  
npm run dev  
``
```

**G2. Verificar funcionalidades**

1. Ir a `http://localhost:3000/editor`
2. Verificar carga de estilos Anclora
3. Probar drag & drop de archivos
4. Verificar navegación entre pasos
5. Confirmar animaciones y transiciones

**G3. Build de producción**

```
``bash  
npm run build  
npm start  
``
```

**FASE H: COMMIT Y DESPLIEGUE**

**H1. Commit de cambios**

```
``bash  
git add .  
git commit -m "feat: implementar Fase 1 - Fundación Técnica con guía Anclora"
```

- Conversor IA Universal con procesamiento automático
- Editor inteligente con sugerencias IA
- Vista previa multi-formato (EPUB3, MOBI, PDF)
- Exportación optimizada con barra de progreso
- Componentes UI siguiendo guía oficial Ancloga Metaform
- Paleta de colores exacta (#006EE6, #00B8D9, etc.)
- Tipografía Inter con escalas definidas
- Sistema de espaciado múltiplos de 4px
- Accesibilidad WCAG 2.1 AA compliant"

H2. Push al repositorio

```
``bash
git push origin feat/fase1-fundacion-tecnica-ancloga
```

H3. Crear Pull Request

- Título: "Fase 1: Fundación Técnica - Implementación completa con guía Ancloga"
- Descripción: Lista de funcionalidades implementadas

**CHECKLIST FINAL 

- [] Tailwind configurado con colores exactos Ancloga
- [] Variables CSS implementadas según guía oficial
- [] Tipografía Inter cargada con escalas correctas
- [] Componentes Button y Card funcionando
- [] Aplicación principal navegable (4 pasos)
- [] Drag & drop para subida de archivos
- [] Barras de progreso animadas
- [] Estados de loading/success/error
- [] Vista previa multi-formato
- [] Estilos responsive
- [] Accesibilidad por teclado
- [] Build de producción exitoso

RESULTADOS ESPERADOS

Al completar estos pasos tendrás:

1. **Aplicación completamente funcional** siguiendo la guía Ancloga
2. **4 funcionalidades core** de la Fase 1 operativas
3. **UI/UX profesional** con colores y tipografía oficial
4. **Base sólida** para implementar Fases 2-5

5. ****Arquitectura escalable**** preparada para crecimiento

****Tiempo estimado de implementación: 3-4 horas****

La implementación resultante será una aplicación de nivel profesional que cumple con los objetivos estratégicos de reducir las barreras técnicas que afectan al 90% de creadores, estableciendo la base para revolucionar la industria editorial.