

## TPS STUDIO

**Primera Entrega (2 de Febrero de 2026)**

**Alumno:** David Gutiérrez Ortiz

**Ciclo:** Desarrollo de Aplicaciones Multiplataforma (DAM) - 2º Curso

**Proyecto:** Intermodular

**Fecha de Entrega:** 2 de febrero de 2026

---

### 1. DESCRIPCIÓN BREVE DEL PROYECTO

TPS Studio es una aplicación de escritorio especializada en el diseño y preimpresión de tarjetas plásticas CR80 (formato estándar de tarjeta de crédito: 85.60 × 53.98 mm) y acreditaciones profesionales. La herramienta está dirigida a técnicos de preimpresión, diseñadores gráficos y empresas de impresión que requieren un control preciso sobre las medidas, márgenes de seguridad y sangrado de corte.

Problema que resuelve

En el sector de la impresión de tarjetas plásticas, los diseñadores suelen utilizar software genérico (Adobe Illustrator, CorelDRAW) que no está optimizado para este formato específico. Esto genera problemas recurrentes:

- **Errores de medidas:** Conversiones manuales de milímetros a píxeles que generan imprecisiones.
- **Márgenes incorrectos:** Falta de guías automáticas para zona de seguridad (3mm) y sangrado (2mm).
- **Flujo de trabajo ineficiente:** Necesidad de configurar plantillas manualmente en cada proyecto.
- **Exportación inadecuada:** Archivos que no cumplen con las especificaciones técnicas de las impresoras de tarjetas.

TPS Studio automatiza estos aspectos críticos, proporcionando un entorno de trabajo específico para CR80 con medidas precisas, guías visuales automáticas y exportación optimizada para preimpresión.

Público objetivo

- **Imprentas especializadas** en tarjetas plásticas y acreditaciones.
- **Departamentos de diseño** de empresas que gestionan eventos, ferias o sistemas de identificación corporativa.
- **Técnicos de preimpresión** que preparan archivos para impresión offset o digital en plástico.
- **Diseñadores freelance** que trabajan con clientes del sector de identificación y acreditación.

## 2. REQUISITOS FUNCIONALES (RF)

### RF01 - Gestión de Proyectos

El sistema debe permitir crear, abrir, guardar y gestionar proyectos de diseño de tarjetas CR80. Cada proyecto incluye metadatos (nombre, cliente, fechas) y puede contener diseños para frente y dorso.

### RF02 - Creación de Plantilla CR80

El sistema debe proporcionar un lienzo de trabajo con las dimensiones exactas de una tarjeta CR80 (85.60 × 53.98 mm), escalado a 4 píxeles por milímetro para visualización óptima.

### RF03 - Gestión de Fondo

El usuario debe poder añadir una imagen de fondo a la tarjeta con dos modos de ajuste:

- **Modo Sangrado (BLEED):** La imagen se extiende 2mm más allá del borde de corte.
- **Modo Final (FINAL):** La imagen se ajusta exactamente al tamaño de corte final.

### RF04 - Añadir Elementos de Texto

El sistema debe permitir insertar elementos de texto con las siguientes propiedades configurables:

- Contenido del texto
- Fuente tipográfica
- Tamaño de fuente
- Color
- Alineación (izquierda, centro, derecha)
- Estilo (negrita, cursiva)

### RF05 - Añadir Elementos de Imagen

El usuario debe poder insertar imágenes (PNG, JPG, JPEG, GIF) como elementos flotantes sobre la tarjeta, con control de:

- Posición (X, Y)
- Dimensiones (ancho, alto)
- Opacidad
- Mantener proporción al redimensionar

#### **RF06 - Selección y Manipulación de Elementos**

El sistema debe permitir seleccionar elementos mediante clic, mostrando un marco de selección con tiradores (handles) para:

- **Mover:** Arrastrar el elemento a una nueva posición.
- **Redimensionar:** Estirar desde las esquinas (imágenes) o desde el lateral derecho (textos).

#### **RF07 - Panel de Propiedades Dinámico**

El sistema debe mostrar un panel lateral derecho que cambia dinámicamente según el elemento seleccionado, permitiendo editar todas sus propiedades en tiempo real.

#### **RF08 - Sistema de Capas**

El sistema debe mostrar una lista de capas (elementos) en el panel izquierdo, permitiendo:

- Visualizar todos los elementos del lado actual (frente/dorso)
- Seleccionar elementos desde la lista
- Identificar elementos bloqueados con icono

#### **RF09 - Bloqueo de Elementos**

El usuario debe poder bloquear/desbloquear elementos para prevenir modificaciones accidentales. Los elementos bloqueados no pueden moverse ni redimensionarse.

#### **RF10 - Gestión de Frente y Dorso**

El sistema debe permitir alternar entre el diseño del frente y el dorso de la tarjeta, manteniendo elementos independientes para cada cara.

#### **RF11 - Guías Visuales**

El sistema debe mostrar/ocultar guías visuales:

- **Borde de corte final** (línea gris sólida)
- **Margen de seguridad** (línea verde discontinua, 3mm interior)
- **Zona de sangrado** (línea roja discontinua, 2mm exterior)

#### **RF12 - Control de Zoom**

El usuario debe poder ajustar el nivel de zoom del lienzo para trabajar con mayor o menor detalle.

#### **RF13 - Edición Externa de Fondos**

El sistema debe permitir abrir la imagen de fondo en un editor externo configurado (ej: Photoshop) y recargar automáticamente los cambios.

#### **RF14 - Recarga de Fondos**

El usuario debe poder recargar la imagen de fondo desde el disco para actualizar cambios realizados externamente.

#### **RF15 - Etiquetado de Elementos**

El sistema debe permitir asignar etiquetas descriptivas a los elementos (ej: "NOMBRE", "FOTO\_CARNET") para facilitar su identificación.

#### **RF16 - Gestión de Datos del Cliente**

El sistema debe permitir asociar información del cliente al proyecto (nombre, empresa, email, teléfono).

#### **RF17 - Persistencia de Proyectos**

El sistema debe guardar proyectos en formato .tps (JSON) incluyendo:

- Metadatos del proyecto
- Configuración de elementos
- Referencias a archivos de imágenes (copiados a carpetas internas del proyecto)

#### **RF18 - Lista de Proyectos Recientes**

El sistema debe mantener un historial de proyectos recientemente abiertos para acceso rápido.

### **3. REQUISITOS NO FUNCIONALES (RNF)**

#### **RNF01 - Rendimiento**

El sistema debe renderizar el lienzo a 60 FPS durante operaciones de arrastre y redimensionamiento, garantizando una experiencia fluida incluso con múltiples elementos.

#### **RNF02 - Precisión de Medidas**

Todas las medidas deben mantener una precisión de  $\pm 0.01$  mm respecto a las especificaciones CR80 estándar, crítico para cumplir con tolerancias de impresión industrial.

#### **RNF03 - Usabilidad**

La interfaz debe seguir convenciones estándar de software de diseño:

- Selección con clic izquierdo
- Menú contextual con clic derecho
- Arrastre intuitivo
- Feedback visual inmediato

#### **RNF04 - Portabilidad**

El sistema debe ejecutarse en Windows 10/11 sin dependencias externas más allá de Java Runtime Environment 21.

#### **RNF05 - Mantenibilidad**

El código debe seguir arquitectura MVVM con separación clara de responsabilidades:

- **Model:** Entidades de negocio (Proyecto, Elemento, etc.)
- **View:** Interfaces FXML y controladores
- **ViewModel:** Estado de la vista
- **Managers:** Lógica especializada (Canvas, Propiedades, Modos)

#### **RNF06 - Escalabilidad**

La arquitectura debe permitir añadir nuevos tipos de elementos (formas, códigos QR, campos variables) sin modificar el núcleo del sistema.

#### **RNF07 - Seguridad de Archivos**

El sistema debe evitar bloqueos de archivos al abrir imágenes, permitiendo edición externa simultánea mediante carga de imágenes por proxy (sin mantener handles abiertos).

#### **RNF08 - Gestión de Memoria**

El sistema debe implementar caché de imágenes con límite de memoria para evitar consumo excesivo en proyectos con múltiples imágenes de alta resolución.

#### **RNF09 - Consistencia Visual**

La interfaz debe mantener un tema oscuro profesional coherente en todos los paneles y diálogos, con paleta de colores definida:

- Fondo principal: #1e1b1c
- Texto primario: #e8e6e7
- Acentos: #4a9b7c (verde azulado)

#### **RNF10 - Tolerancia a Errores**

El sistema debe validar entradas del usuario y mostrar mensajes de error claros en caso de:

- Archivos de imagen corruptos
- Rutas inválidas
- Valores numéricos fuera de rango

#### **RNF11 - Compatibilidad de Formatos**

El sistema debe soportar los formatos de imagen más comunes en preimpresión: PNG, JPG, JPEG, GIF.

#### **RNF12 - Persistencia Robusta**

El guardado de proyectos debe ser atómico (escritura completa o rollback) para evitar corrupción de archivos en caso de cierre inesperado.

#### **RNF13 - Documentación de Código**

Todo el código debe incluir comentarios en español explicando la lógica de negocio, especialmente en secciones críticas como cálculo de coordenadas y renderizado.

#### **RNF14 - Tiempo de Respuesta**

Las operaciones de guardado/carga de proyectos deben completarse en menos de 2 segundos para proyectos de tamaño medio (< 20 elementos, imágenes < 5MB).

## **4. CASOS DE USO**

### **4.1 Explicación de los Casos de Uso**

CU01 - Crear Nuevo Proyecto

**Actor:** Usuario/Diseñador

**Descripción:** El usuario crea un nuevo proyecto de diseño de tarjeta CR80 desde cero.

**Flujo Principal:**

1. El usuario hace clic en "Nuevo Proyecto" desde el menú o pantalla de inicio.
2. El sistema muestra un diálogo solicitando nombre del proyecto y datos del cliente (opcional).
3. El usuario introduce el nombre del proyecto (ej: "Acreditación Feria 2026").
4. El usuario opcionalmente completa datos del cliente (nombre, empresa, email, teléfono).
5. El sistema crea la estructura de carpetas del proyecto y abre el editor con un lienzo CR80 vacío.
6. El sistema muestra el modo Diseño con herramientas activas.

**Flujos Alternativos:**


- **3a.** Si el usuario cancela el diálogo, el sistema vuelve a la pantalla anterior sin crear proyecto.
- **3b.** Si el nombre está vacío, el sistema muestra error y solicita un nombre válido.

## CU02 - Añadir Fondo a la Tarjeta

**Actor:** Usuario/Diseñador

**Descripción:** El usuario establece una imagen de fondo para la cara actual de la tarjeta.

### Flujo Principal:

1. El usuario hace clic en el botón "  Fondo" del panel de herramientas.
2. El sistema abre un selector de archivos filtrado por imágenes (PNG, JPG, JPEG, GIF).
3. El usuario selecciona un archivo de imagen desde su sistema.
4. El sistema copia el archivo a la carpeta

Fondos/ del proyecto.

5. El sistema ajusta automáticamente la imagen al modo BLEED (con sangrado de 2mm).
6. El sistema renderiza la imagen de fondo en el lienzo.

### Flujos Alternativos:

- **3a.** Si el usuario cancela la selección, no se añade fondo.
  - **4a.** Si el archivo no es una imagen válida, el sistema muestra error y solicita otro archivo.
  - **5a.** El usuario puede cambiar posteriormente el modo de ajuste a FINAL desde el panel de propiedades.
- 

## CU03 - Añadir Elemento de Texto

**Actor:** Usuario/Diseñador

**Descripción:** El usuario inserta un campo de texto en la tarjeta.

### Flujo Principal:

1. El usuario hace clic en el botón "T Texto" del panel de herramientas.
2. El sistema crea un elemento de texto con valores por defecto (contenido "Texto", fuente Arial, tamaño 16, color negro).
3. El sistema posiciona el texto en el centro del lienzo.
4. El sistema selecciona automáticamente el nuevo elemento.
5. El sistema muestra el panel de propiedades de texto en el lateral derecho.
6. El usuario edita el contenido, fuente, tamaño y demás propiedades según necesidad.

**Flujos Alternativos:**

- **6a.** El usuario puede mover y redimensionar el texto arrastrándolo en el lienzo.
  - **6b.** El usuario puede asignar una etiqueta descriptiva (ej: "NOMBRE\_TITULAR").
- 

**CU04 - Mover y Redimensionar Elemento**

**Actor:** Usuario/Diseñador

**Descripción:** El usuario ajusta la posición y tamaño de un elemento en el lienzo.

**Flujo Principal:**

1. El usuario hace clic sobre un elemento en el lienzo.
2. El sistema selecciona el elemento y muestra el marco de selección con tiradores.
3. El usuario arrastra el elemento a la nueva posición deseada.
4. El sistema actualiza las coordenadas X, Y en tiempo real.
5. El usuario arrastra un tirador de esquina (imágenes) o lateral (textos) para redimensionar.
6. El sistema actualiza las dimensiones ancho/alto manteniendo proporción si está activado.

**Flujos Alternativos:**

- **1a.** Si el elemento está bloqueado, el sistema no permite moverlo ni redimensionarlo.
  - **3a.** El usuario puede introducir coordenadas exactas en el panel de propiedades.
  - **5a.** En imágenes con "Mantener proporción" activo, el redimensionamiento es proporcional.
- 

**CU05 - Configurar Modo de Ajuste del Fondo**

**Actor:** Usuario/Diseñador

**Descripción:** El usuario cambia entre modo sangrado y modo final para el fondo de la tarjeta.

**Flujo Principal:**

1. El usuario selecciona el elemento de fondo desde la lista de capas o haciendo clic en el lienzo.
2. El sistema muestra el panel de propiedades del fondo en el lateral derecho.



3. El usuario selecciona el radio button "Con sangre (CR80 + sangrado)" o "Corte final (CR80 exacto)".
4. El sistema recalcula automáticamente las dimensiones del fondo según el modo elegido.
5. El sistema reposiciona la imagen para que esté centrada respecto a la tarjeta.
6. El sistema repinta el lienzo mostrando el nuevo ajuste.

**Flujos Alternativos:**

- **4a.** Si la imagen original es más pequeña que el tamaño requerido, puede aparecer pixelada (el sistema no escala hacia arriba).
- 

**CU06 - Editar Fondo en Editor Externo**

**Actor:** Usuario/Diseñador

**Descripción:** El usuario abre la imagen de fondo en un editor externo (ej: Photoshop) para realizar modificaciones avanzadas.

**Flujo Principal:**

1. El usuario selecciona el fondo y hace clic en "Editor Externo" en el panel de propiedades.
2. El sistema verifica si hay un editor configurado en las preferencias.
3. El sistema lanza el editor externo mediante proceso desacoplado (sin bloqueo de archivo).
4. El usuario realiza modificaciones en el editor externo y guarda el archivo.
5. El usuario regresa a TPS Studio y hace clic en "Recargar desde disco".
6. El sistema recarga la imagen actualizada y repinta el lienzo.

**Flujos Alternativos:**

- **2a.** Si no hay editor configurado, el sistema muestra diálogo para seleccionar el ejecutable del editor.
- **4a.** Si el archivo se elimina externamente, el sistema muestra error al intentar recargar.

## CU07 - Alternar entre Frente y Dorso

**Actor:** Usuario/Diseñador

**Descripción:** El usuario cambia entre el diseño del frente y el dorso de la tarjeta.

### Flujo Principal:

1. El usuario hace clic en el botón "Frente/Dorso" de la barra de herramientas.
2. El sistema guarda el estado actual del lado visible.
3. El sistema cambia el flag interno mostrando Frente al valor opuesto.
4. El sistema carga los elementos correspondientes al nuevo lado.
5. El sistema actualiza el panel de capas con los elementos del nuevo lado.
6. El sistema repinta el lienzo mostrando el nuevo lado.

### Flujos Alternativos:

- **3a.** El sistema muestra indicador visual "FRENTE" o "DORSO" en la parte superior del lienzo.
- 

## CU08 - Guardar Proyecto

**Actor:** Usuario/Diseñador

**Descripción:** El usuario guarda el estado actual del proyecto en disco.

### Flujo Principal:

1. El usuario hace clic en "Guardar Proyecto" desde el menú.
2. El sistema serializa todos los elementos del proyecto a formato JSON.
3. El sistema guarda el archivo .tps en la carpeta del proyecto.
4. El sistema actualiza la fecha de modificación en los metadatos.
5. El sistema añade el proyecto a la lista de recientes si es nuevo.
6. El sistema muestra confirmación visual en la barra de estado.

### Flujos Alternativos:

- **2a.** Si hay errores de serialización, el sistema muestra mensaje de error y no sobrescribe el archivo anterior.
- **3a.** Si no hay permisos de escritura, el sistema solicita otra ubicación.

## CU09 - Abrir Proyecto Existente

**Actor:** Usuario/Diseñador

**Descripción:** El usuario carga un proyecto previamente guardado.

### Flujo Principal:

1. El usuario hace clic en "Abrir Proyecto" o selecciona un proyecto de la lista de recientes.
2. El sistema muestra un selector de archivos filtrado por .tps (si no es desde recientes).
3. El usuario selecciona el archivo del proyecto.
4. El sistema deserializa el JSON y reconstruye la estructura del proyecto.
5. El sistema carga las imágenes referenciadas desde las carpetas del proyecto.
6. El sistema abre el editor mostrando el lado frente por defecto.

### Flujos Alternativos:

- **4a.** Si el archivo está corrupto, el sistema muestra error y no carga el proyecto.
- **5a.** Si alguna imagen referenciada no existe, el sistema muestra advertencia pero carga el resto del proyecto.

## CU10 - Exportar a PDF (Futuro)

**Actor:** Usuario/Diseñador

**Descripción:** El usuario exporta el diseño finalizado a formato PDF listo para impresión.

### Flujo Principal:

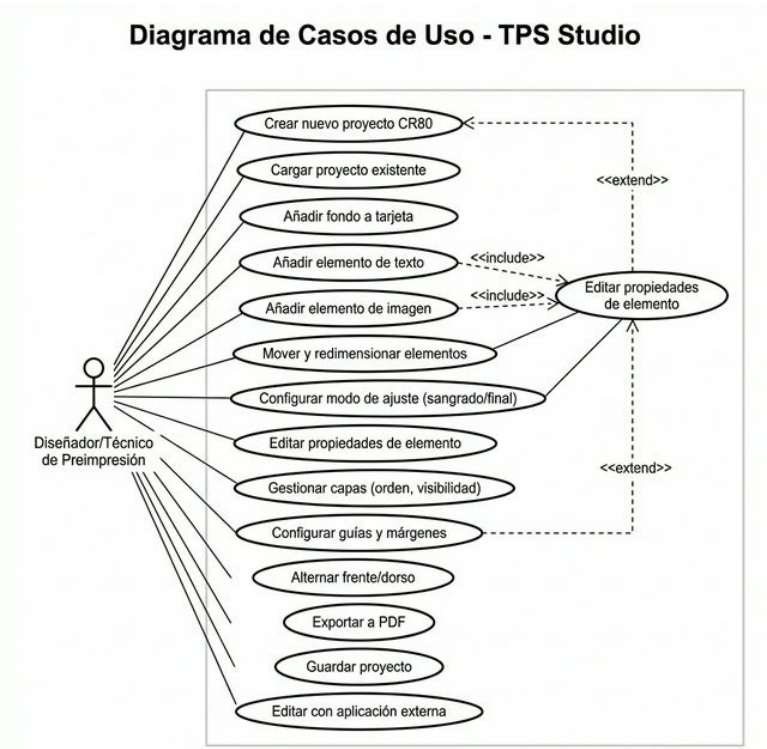
1. El usuario hace clic en "Exportar" desde el panel de producción.
2. El sistema muestra opciones de exportación (DPI, incluir guías, frente/dorso/ambos).
3. El usuario configura las opciones y selecciona ubicación de guardado.
4. El sistema genera el PDF con las especificaciones técnicas correctas.
5. El sistema guarda el archivo en la ubicación seleccionada.
6. El sistema muestra confirmación y opción de abrir el PDF.

### Flujos Alternativos:

- **Nota:** Esta funcionalidad está prevista pero no implementada en la primera iteración.

4.2 Diagrama de Casos de Uso

INSTRUCCIONES PARA CREAR EL DIAGRAMA:



5. DIAGRAMA DE GANTT

Planificación Temporal – Primera Iteración

El desarrollo de TPS Studio comenzó en diciembre de 2025 y se ha organizado en una primera iteración de aproximadamente 12 semanas, con entregas parciales y revisión continua. La planificación realista queda estructurada de la siguiente forma:

Semanas	Fase	Tareas Principales
1–2 (Dic 2025)	Análisis y Diseño	Estudio de requisitos, definición de arquitectura MVVM, planteamiento inicial del modelo de proyecto
3–4 (Dic 2025)	Configuración del Proyecto	Setup Maven, configuración JavaFX, estructura de paquetes
5–6 (Ene 2026)	Implementación UI Base	Pantalla de login, ventana principal, estructura de paneles
6–8 (Ene 2026)	Desarrollo Canvas y Editor	Sistema de renderizado, coordenadas, zoom, guías visuales
8–9 (Final Ene 2026)	Gestión de Elementos	Clase Elemento base, TextoElemento, ImagenElemento, ImagenFondoElemento

Semanas	Fase	Tareas Principales
<b>9-10 (Inicio Feb 2026)</b>	Sistema de Propiedades y Capas	Panel dinámico de propiedades, lista de capas, sistema de selección
<b>10-11 (Actual)</b>	Funcionalidades de Exportación	Preparación base para exportación PDF (estructura inicial, no completada)
<b>11-12 (En curso)</b>	Pruebas y Documentación	Testing funcional, corrección de bugs, documentación técnica inicial

## 6. ESQUEMA ENTIDAD-RELACIÓN Y NORMALIZACIÓN

Nota Importante

**En esta primera iteración, TPS Studio NO utiliza base de datos relacional.** La persistencia se realiza mediante serialización JSON de objetos a archivos .tps, con referencias a archivos de imágenes almacenados en carpetas del proyecto.

Sin embargo, se presenta a continuación un **modelo conceptual propuesto** para una futura evolución del sistema que requiera almacenamiento en base de datos (por ejemplo, para gestión multiusuario, versionado o integración con sistemas empresariales).

### 6.1 Modelo Entidad-Relación Propuesto

Entidades

#### PROYECTO

- id\_proyecto (PK, INT, AUTO\_INCREMENT)
- nombre (VARCHAR(100), NOT NULL)
- fecha\_creacion (DATETIME, NOT NULL)
- fecha\_modificacion (DATETIME, NOT NULL)
- ruta\_archivo (VARCHAR(255), UNIQUE)
- id\_cliente (FK, INT, NULLABLE)

#### CLIENTE

- id\_cliente (PK, INT, AUTO\_INCREMENT)
- nombre (VARCHAR(100), NOT NULL)
- empresa (VARCHAR(100), NULLABLE)
- email (VARCHAR(100), NULLABLE)
- telefono (VARCHAR(20), NULLABLE)

## DOCUMENTO

- id\_documento (PK, INT, AUTO\_INCREMENT)
- id\_proyecto (FK, INT, NOT NULL)
- tipo\_cara (ENUM('FRENTE', 'DORSO'), NOT NULL)
- ancho\_mm (DECIMAL(6,2), NOT NULL, DEFAULT 85.60)
- alto\_mm (DECIMAL(6,2), NOT NULL, DEFAULT 53.98)

## ELEMENTO

- id\_elemento (PK, INT, AUTO\_INCREMENT)
- id\_documento (FK, INT, NOT NULL)
- tipo (ENUM('TEXTO', 'IMAGEN', 'FONDO'), NOT NULL)
- etiqueta (VARCHAR(50), NULLABLE)
- x (DECIMAL(8,2), NOT NULL)
- y (DECIMAL(8,2), NOT NULL)
- ancho (DECIMAL(8,2), NOT NULL)
- alto (DECIMAL(8,2), NOT NULL)
- z\_index (INT, NOT NULL, DEFAULT 0)
- visible (BOOLEAN, NOT NULL, DEFAULT TRUE)
- bloqueado (BOOLEAN, NOT NULL, DEFAULT FALSE)

## ELEMENTO\_TEXTO (Especialización de ELEMENTO)

- id\_elemento (PK, FK, INT)
- contenido (TEXT, NOT NULL)
- fuente (VARCHAR(50), NOT NULL, DEFAULT 'Arial')
- tamaño (INT, NOT NULL, DEFAULT 16)
- color (VARCHAR(7), NOT NULL, DEFAULT '#000000')
- alineacion (ENUM('LEFT', 'CENTER', 'RIGHT'), DEFAULT 'LEFT')
- negrita (BOOLEAN, DEFAULT FALSE)
- cursiva (BOOLEAN, DEFAULT FALSE)

### **ELEMENTO\_IMAGEN** (Especialización de ELEMENTO)

- id\_elemento (PK, FK, INT)
- ruta\_archivo (VARCHAR(255), NOT NULL)
- opacidad (DECIMAL(3,2), NOT NULL, DEFAULT 1.00)
- mantener\_proporcion (BOOLEAN, NOT NULL, DEFAULT TRUE)

### **ELEMENTO\_FONDO** (Especialización de ELEMENTO)

- id\_elemento (PK, FK, INT)
- ruta\_archivo (VARCHAR(255), NOT NULL)
- modo\_ajuste (ENUM('BLEED', 'FINAL'), NOT NULL, DEFAULT 'BLEED')

### Relaciones

1. **PROYECTO - CLIENTE:** N:1 (Muchos proyectos pueden pertenecer a un cliente, un cliente puede tener muchos proyectos)
2. **PROYECTO - DOCUMENTO:** 1:N (Un proyecto tiene exactamente 2 documentos: frente y dorso)
3. **DOCUMENTO - ELEMENTO:** 1:N (Un documento contiene múltiples elementos)
4. **ELEMENTO - ELEMENTO\_TEXTO:** 1:0..1 (Herencia/Especialización)
5. **ELEMENTO - ELEMENTO\_IMAGEN:** 1:0..1 (Herencia/Especialización)
6. **ELEMENTO - ELEMENTO\_FONDO:** 1:0..1 (Herencia/Especialización)

### 6.2 Claves

#### **Claves Primarias (PK):**

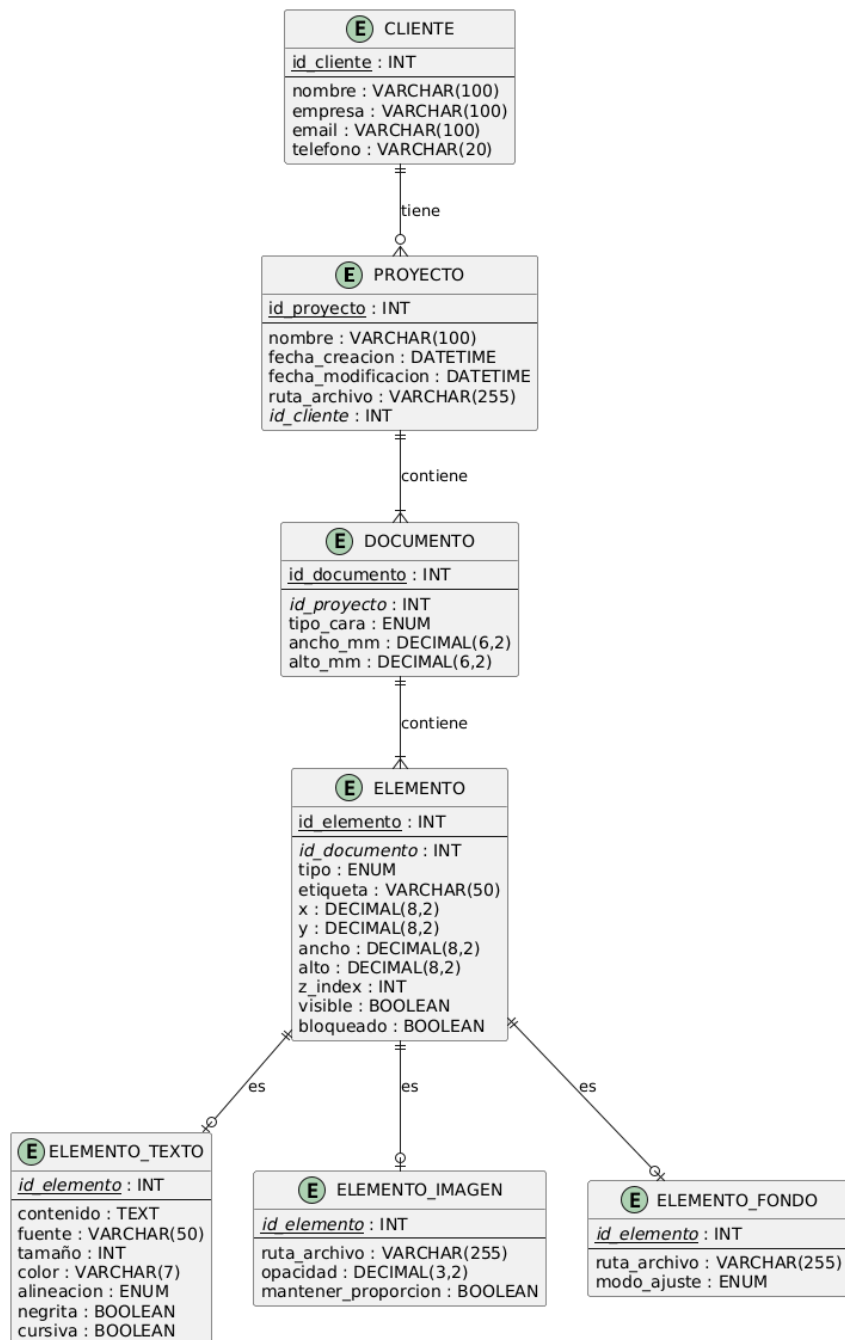
- Todas las entidades principales tienen id\_[entidad] como clave primaria auto-incremental.
- Las tablas de especialización (ELEMENTO\_TEXTO, ELEMENTO\_IMAGEN, ELEMENTO\_FONDO) usan el mismo id\_elemento como PK y FK.

#### **Claves Foráneas (FK):**

- PROYECTO.id\_cliente → CLIENTE.id\_cliente
- DOCUMENTO.id\_proyecto → PROYECTO.id\_proyecto
- ELEMENTO.id\_documento → DOCUMENTO.id\_documento
- ELEMENTO\_TEXTO.id\_elemento → ELEMENTO.id\_elemento
- ELEMENTO\_IMAGEN.id\_elemento → ELEMENTO.id\_elemento
- ELEMENTO\_FONDO.id\_elemento → ELEMENTO.id\_elemento

### Restricciones de Integridad:

- ON DELETE CASCADE en DOCUMENTO.id\_proyecto (si se elimina un proyecto, se eliminan sus documentos)
- ON DELETE CASCADE en ELEMENTO.id\_documento (si se elimina un documento, se eliminan sus elementos)
- ON DELETE SET NULL en PROYECTO.id\_cliente (si se elimina un cliente, los proyectos quedan sin cliente asignado)





## 7. ANÁLISIS DAFO

### Fortalezas (Strengths)

- **Nicho Especializado:** TPS Studio se enfoca exclusivamente en el formato CR80, un mercado con necesidades específicas no cubiertas por software genérico.
- **Experiencia Real en el Sector:** El desarrollo está basado en conocimiento directo del flujo de trabajo de preimpresión de tarjetas plásticas.
- **Precisión de Medidas:** Control milimétrico de dimensiones, sangrado y márgenes de seguridad, crítico para cumplir especificaciones de impresión industrial.
- **Interfaz Enfocada en Productividad:** Herramientas específicas para el caso de uso, sin funcionalidades innecesarias que distraigan al usuario.
- **Arquitectura MVVM Escalable:** Código bien estructurado que facilita mantenimiento y adición de nuevas funcionalidades.
- **Sin Dependencias de Licencias:** Software de código propio sin dependencias de librerías de pago o con restricciones de licencia.

### Debilidades (Weaknesses)

- **Proyecto Individual:** Desarrollo realizado por una sola persona, lo que limita la velocidad de implementación y cobertura de testing.
- **Primera Versión:** Funcionalidad básica en esta iteración, falta de características avanzadas (exportación PDF, campos variables, integración con bases de datos).
- **Sin Base de Datos Implementada:** Persistencia mediante archivos JSON, lo que limita capacidades multiusuario y consultas complejas.
- **Alcance Inicial Reducido:** Solo soporta formato CR80, no otros formatos de tarjetas o acreditaciones de tamaños personalizados.
- **Curva de Aprendizaje JavaFX:** Tecnología menos popular que frameworks web, lo que puede dificultar encontrar colaboradores o soporte comunitario.
- **Falta de Integración:** No se conecta aún con sistemas de gestión de clientes o flujos de impresión automatizados.

### Oportunidades (Opportunities)

- **Mercado de Imprentas Locales:** Empresas pequeñas y medianas que buscan alternativas económicas a software profesional costoso (Adobe, CorelDRAW).
- **Clientes Corporativos:** Empresas con necesidades recurrentes de acreditaciones para eventos, ferias, control de acceso.
- **Eventos y Ferias Comerciales:** Sector en crecimiento post-pandemia con alta demanda de identificación visual profesional.
- **Integración con Flujos de Impresión:** Posibilidad de conectar con RIPs (Raster Image Processors) y sistemas de impresión directa.

- **Expansión a Otros Formatos:** Futuras versiones podrían soportar formatos personalizados, tarjetas de visita, pases VIP, etc.
- **Automatización de Procesos:** Implementación de campos variables conectados a bases de datos para impresión masiva personalizada.
- **Modelo Freemium:** Versión básica gratuita con funcionalidades premium de pago (exportación avanzada, plantillas prediseñadas).

#### Amenazas (Threats)

- **Software Establecido:** Competencia de soluciones consolidadas como Adobe Illustrator, CorelDRAW, que aunque no especializadas, son estándar de la industria.
- **Soluciones Web:** Aparición de herramientas online que no requieren instalación y funcionan en cualquier plataforma.
- **Cambios en Estándares de Impresión:** Evolución de tecnologías de impresión que podrían requerir adaptaciones significativas del software.
- **Dependencia de Tecnología Java:** JavaFX tiene menor adopción que tecnologías web modernas, lo que puede limitar la comunidad de desarrollo.
- **Necesidad de Mantenimiento Continuo:** Actualizaciones de sistemas operativos y Java requieren testing y adaptaciones periódicas.
- **Resistencia al Cambio:** Usuarios acostumbrados a flujos de trabajo existentes pueden ser reacios a adoptar nueva herramienta.
- **Piratería de Software Profesional:** Acceso ilegal a software de pago reduce el incentivo de adoptar alternativas gratuitas.

#### Tabla DAFO

##### FORTALEZAS

- Nicho especializado en preimpresión
- Experiencia real en el sector
- Medidas precisas CR80 estándar
- Interfaz enfocada en productividad
- Control total de sangrado y márgenes
- Arquitectura MVVM escalable

##### OPORTUNIDADES

- Mercado de imprentas locales
- Clientes corporativos (acreditaciones)
- Eventos y ferias comerciales
- Integración con flujos de impresión
- Expansión a otros formatos
- Automatización de procesos

##### DEBILIDADES

- Proyecto individual (recursos limitados)
- Primera versión (funcionalidad básica)
- Sin base de datos implementada
- Alcance inicial reducido
- Curva de aprendizaje JavaFX
- Falta de integración

##### AMENAZAS

- Software establecido (Adobe, CorelDRAW)
- Competencia de soluciones web
- Cambios en estándares de impresión
- Dependencia de tecnología Java
- Necesidad de mantenimiento continuo
- Resistencia al cambio

## 8. NOTA SOBRE TECNOLOGÍA Y MULTIPLATAFORMA

Tecnología Actual: JavaFX Desktop

TPS Studio ha sido desarrollado como **aplicación de escritorio nativa** utilizando Java 21 y JavaFX 21.0.4. Esta decisión tecnológica se basa en los siguientes criterios:

### Justificación de la elección:

1. **Enfoque en Herramienta de Preimpresión:** El público objetivo (técnicos de preimpresión, diseñadores en imprentas) trabaja principalmente en estaciones de trabajo Windows con monitores de alta resolución, donde las aplicaciones de escritorio ofrecen mejor rendimiento y control.
2. **Precisión y Rendimiento:** El renderizado de elementos gráficos con medidas milimétricas requiere control preciso del canvas y capacidad de procesamiento local, aspectos donde JavaFX destaca sobre soluciones web.
3. **Integración con Sistema de Archivos:** La gestión de proyectos con carpetas de imágenes, edición externa en Photoshop y exportación a PDF se beneficia del acceso directo al sistema de archivos local.
4. **Madurez de JavaFX:** Tecnología estable con amplia documentación y soporte para aplicaciones empresariales de escritorio.

---

### Sobre Flutter y Migración Multiplataforma

En la documentación inicial del proyecto se mencionaba la posibilidad de utilizar **Flutter** para desarrollo multiplataforma (Android, macOS, iOS). Sin embargo, tras el análisis técnico y de mercado, se ha determinado que:

### No aplica en esta iteración:

- Flutter está orientado principalmente a aplicaciones móviles y web, mientras que TPS Studio requiere capacidades de escritorio avanzadas (gestión de archivos, integración con editores externos, renderizado de alta precisión).
- El caso de uso principal (preimpresión profesional) se realiza en entornos de escritorio Windows, no en dispositivos móviles.
- La complejidad de mantener dos bases de código (JavaFX y Flutter) no se justifica en la fase inicial del proyecto.

**Posible línea futura:** Si en iteraciones posteriores se identifica demanda de:

- **Versión móvil** para previsualización de diseños en campo (eventos, ferias)
- **Versión macOS** para diseñadores que trabajan en ecosistema Apple
- **Versión web** para colaboración remota

Se podría evaluar:

1. **Migración completa a Flutter Desktop** (que soporta Windows, macOS, Linux)
2. **Desarrollo de app complementaria** en Flutter para móvil, manteniendo JavaFX para escritorio
3. **Arquitectura híbrida** con backend compartido y frontends específicos por plataforma

Por el momento, la estrategia es **consolidar la versión Windows de escritorio** antes de considerar expansión multiplataforma.

## 9. CONCLUSIÓN

Estado Actual del Proyecto

TPS Studio ha completado su **primera iteración funcional** con los componentes core implementados:

- Arquitectura MVVM con separación clara de responsabilidades
- Sistema de gestión de proyectos con persistencia JSON
- Editor visual con canvas CR80 de medidas precisas
- Gestión completa de elementos (texto, imagen, fondo)
- Panel de propiedades dinámico
- Sistema de capas con selección y bloqueo
- Guías visuales de sangrado y márgenes de seguridad
- Integración con editores externos (Photoshop)
- Interfaz de usuario con tema oscuro profesional

Funcionalidades Pendientes

Para la **segunda iteración** se priorizarán:

1. **Exportación a PDF:** Generación de archivos PDF con especificaciones técnicas para impresión (CMYK, sangrado, marcas de corte).
2. **Campos Variables:** Sistema de plantillas con campos dinámicos conectables a fuentes de datos (CSV, base de datos). La aplicación permitirá importar datos variables desde fuentes externas (por ejemplo, Excel/CSV y bases de datos como Access/MySQL) para rellenar automáticamente campos etiquetados en el diseño (p. ej. NOMBRE, ID\_SOCIO, CÓDIGO\_QR). El usuario podrá mapear columnas/campos de origen con las etiquetas del proyecto y generar previsualizaciones o exportaciones por lotes
3. **Historial de Cambios:** Implementación de undo/redo para mejorar la experiencia de edición.

4. **Biblioteca de Plantillas:** Colección de diseños predefinidos para acelerar el flujo de trabajo.
5. **Optimización de Rendimiento:** Mejoras en el sistema de caché de imágenes y renderizado para proyectos complejos.

#### Próximos Pasos Inmediatos

Tras esta entrega del 2 de febrero, el desarrollo continuará con:

1. **Validación con Usuarios Reales:** Pruebas con técnicos de preimpresión para identificar mejoras de usabilidad.
2. **Implementación de Exportación PDF:** Prioridad máxima para la siguiente iteración, ya que es funcionalidad crítica para el flujo de producción.
3. **Refinamiento de UI/UX:** Ajustes basados en feedback de usuarios beta.
4. **Documentación de Usuario:** Creación de manual de usuario y tutoriales en video.

TPS Studio demuestra viabilidad técnica y potencial de mercado como herramienta especializada para preimpresión de tarjetas CR80, con una base sólida para evolución futura.