

## Análisis Funcional y Matemático de la Página de Contratación

Esta página es una aplicación web interactiva diseñada para que los clientes configuren y soliciten un pedido de códigos QR personalizados, con un cálculo de precios en tiempo real. A continuación se desglosa su funcionamiento.

### 1. Estructura y Estilo General

- **Tecnologías Principales:** El sitio está construido con HTML, y utiliza **Tailwind CSS** para un diseño moderno y adaptable (responsive). La interactividad se maneja con **JavaScript** puro, sin dependencias de frameworks como React o Vue.
- **Diseño Visual:**
  - **Tema Oscuro:** La página utiliza una paleta de colores oscura por defecto (`<html class="dark">`), lo que le da un aspecto sofisticado y profesional.
  - **Fondo Dinámico:** Un fondo con una imagen de alta calidad (D3-LTA-Fondo.jpg) se mantiene fijo en el viewport. Sobre él, se aplica un efecto de desenfoque (`backdrop-filter: blur(6px)`) y un velo semitransparente (`rgba(10, 10, 15, 0.9)`) para asegurar que el contenido del formulario sea legible y destaque.
  - **Tipografía:** Se utiliza una selección de más de 120 familias de **Google Fonts** para ofrecer una personalización amplia. Las fuentes principales son 'Barlow Condensed' para títulos y 'Poppins' para el cuerpo del texto, definidas mediante variables CSS para una fácil gestión y consistencia en todo el sitio.

### 2. Selector de Moneda (ARS / USD)

- **Interfaz:** Consta de dos botones de radio (ARS, USD) con un marcador visual que se desliza suavemente para indicar la selección activa, proporcionando una experiencia de usuario fluida e intuitiva.
- **Mecanismo de Conversión:**
  1. **Obtención de Tasas de Cambio:** Al cargar la página, la función `fetchExchangeRates` realiza una llamada asíncrona a una API externa (<https://open.er-api.com/v6/latest/CHF>) para obtener las tasas de cambio más recientes, utilizando el Franco Suizo (CHF) como moneda base para todos los precios internos. Este enfoque asegura que los precios se mantengan actualizados globalmente.
  2. **Mecanismo de Respaldo (Fallback):** En caso de que la API falle o no responda, el sistema recurre a tasas de cambio predefinidas (ARS: 1250, USD: 1.1) para garantizar que la página siga siendo funcional en todo momento.
  3. **Cálculo en Tiempo Real:** La función `convert(valueInChf)` toma cualquier precio en CHF y lo multiplica por la tasa de cambio de la moneda

seleccionada (`appState.currentCurrency`), que está almacenada en el estado de la aplicación.

4. **Formateo de Moneda:** La función `formatCurrency(value)` utiliza la API de internacionalización de JavaScript (`Intl.NumberFormat`) para dar un formato adecuado al valor numérico según la moneda elegida. Para ARS, aplica un redondeo especial para mostrar cifras más limpias (`Math.round(value / 1000) * 1000`), una práctica común para la moneda local, mientras que para USD, redondea al entero más cercano.

### 3. Paso 1: Nivel de Diseño y Funcionalidad

- **Nivel de Diseño del QR:**
  - **Funcionalidad:** El cliente elige entre tres niveles: Básico, Estándar y Premium. Cada uno tiene un precio base fijo en CHF (40, 62, 110 respectivamente).
  - **Impacto en el Sistema:** La selección del nivel de diseño es una variable fundamental que actualiza el `baseDesignCost` en el motor de precios. Además, tiene un efecto en cascada: determina la cantidad máxima de colores que se pueden elegir en la paleta del Paso 2 y ajusta el costo base del servicio de "Entrega Express".
- **Plan de Funcionalidad (URL Dinámica):**
  - **Funcionalidad:** Ofrece cuatro planes (Ninguno, Starter, Business, Scale) que definen cuántas "versiones de destino" pueden ser dinámicas, es decir, editables a futuro.
  - **Precios y Renovación:** Cada plan tiene un costo inicial (`price`) y una tarifa de renovación anual (`renewal`) asociados. Por ejemplo, el plan "Starter" tiene un costo de 35 CHF y una renovación de 15 CHF.
  - **Límites de URLs:** Cada plan define un límite de URLs dinámicas permitidas (0, 3, 12, o 30). Este límite es crucial para la lógica de la interfaz en el Paso 3.
  - **Impacto en el Costo Total:** La selección de un plan actualiza las variables `functionalityFee` (el costo único) y `renewalFee` (el costo anual recurrente), que se muestran por separado en el resumen del pedido.

### 4. Paso 2: Información Creativa

- **Identidad Visual:**
  - **Carga de Archivos:** Los botones de "Seleccionar archivo" están diseñados para ser visualmente atractivos y funcionales. La función `handleFileInputChange` gestiona la carga de archivos del logo y de la imagen de referencia para colores, utilizando `FileReader` para leer el archivo localmente y mostrar una previsualización instantánea, reemplazando el texto de marcador de posición.

- **Sugerencia de Tipografías:** Inicialmente, se muestra un solo selector de fuentes (addFontSelector). El cliente puede añadir hasta dos más con un botón. La función removeFontSelector asegura que el botón de eliminar solo sea visible si hay más de un selector, evitando dejar la sección sin opciones.
- **Paleta de Colores:** La función updateColorPickerUI se ejecuta cada vez que cambia el "Nivel de Diseño". Limpia y regenera dinámicamente la cantidad de selectores de color (<input type="color">) permitidos por el plan elegido (3, 5 o 7).
- **Palabra Clave del Vendedor:**
  - **Funcionalidad y Lógica:** El campo de texto activa la función handleKeywordValidation en cada pulsación de tecla. Esta función compara el texto ingresado (convertido a mayúsculas y sin espacios) con el código predefinido "VENDEDOR01". Si hay una coincidencia, se activa un descuento del 10% (appState.sellerDiscount.active = true). Este descuento se aplica como un multiplicador (\* 0.90) al final de todos los cálculos de costos.

## 5. Paso 3: Configuración de Destinos

- **Versiones de Destino:** El usuario puede crear múltiples "versiones" de un mismo QR. Cada versión puede apuntar a un destino diferente. La función addLinkEntry clona una plantilla HTML para crear nuevas entradas de forma dinámica. La opción "Autenticación de dos factores" ha sido eliminada de los tipos de destino para simplificar la selección.
- **Lógica de URL Dinámica (Función Clave):**
  - handleDynamicUrlToggle: Cuando se activa el checkbox "Convertir en URL Dinámica":
    1. La función almacena el tipo de destino original en un atributo dataset para poder restaurarlo.
    2. Oculta el selector de "Tipo de Destino" (<select>) usando la clase invisible, lo que preserva el espacio del elemento y evita que el layout se desplace.
    3. Muestra un div superpuesto con el texto "URL (Dinámica)", que tiene la misma altura y padding que el input adyacente, manteniendo la consistencia visual.
    4. Reemplaza el campo de entrada de la URL por un cuadro informativo. Si la opción original era "URL Estática", reemplaza solo el input dentro de su contenedor (.url-input-wrapper); de lo contrario, reemplaza todo el contenedor de valor (.link-value-container).
  - updateDynamicUrlCheckboxesState: Esta es la función central que gestiona los límites del plan.
    1. **Verificación de Límite:** Al cambiar el "Plan de Funcionalidad", esta

función se ejecuta y compara el número de URLs dinámicas ya activadas (.link-dynamic-url-toggle:checked) con el limit del nuevo plan.

2. **Desactivación Automática:** Si el número de URLs activas excede el nuevo límite (al bajar de plan), la función desactiva automáticamente las últimas que se agregaron hasta cumplir con el cupo. Para ello, invoca a handleDynamicUrlToggle en los checkboxes sobrantes, restaurando su estado original.
3. **Bloqueo de Opciones:** Deshabilita los checkboxes de las demás versiones si ya se ha alcanzado el límite, y cambia el color del texto de la etiqueta a gris para indicar visualmente que la opción no está disponible.

## 6. Paso 4 y 5: Adicionales y Servicio de Impresión

- **Mecanismos Matemáticos de Precios Dinámicos:**
  - **Entrega Express (getExpressDeliveryCostChf):** El precio de este servicio se compone de una tarifa base más un costo variable. La tarifa base depende de la combinación del "Nivel de Diseño" y el "Plan de Funcionalidad". El costo variable se calcula con una **interpolación lineal** que disminuye el precio por versión a medida que se añaden más, desde CHF 4.0 para la primera versión hasta CHF 2.5 al llegar a 30 versiones. Esto incentiva los pedidos de mayor volumen.
  - **Impresión de Stickers (getStickerPricePerUnit):** El precio por sticker se calcula mediante una **interpolación lineal** del costo *total*, no del costo unitario. Se toman dos puntos de referencia: el costo total para 1 sticker (total\_at\_1) y el costo total para 1000 stickers (total\_at\_1000). La fórmula  $total\_at\_1 + ((total\_at\_1000 - total\_at\_1) / 999) * (cantidad - 1)$  calcula el costo total para una cantidad X, y luego lo divide por X para obtener el precio por unidad. Esto resulta en un costo por unidad que decrece a medida que aumenta la cantidad. Si se solicita el recorte individual, se aplica un multiplicador del 160% (cut\_multiplier: 1.6) sobre el costo de impresión.

## 7. Resumen del Pedido y Envío Final

- **Desglose de Costos:** La barra lateral, actualizada por la función updateSummary, muestra un desglose claro y transparente de los costos, separando la tarifa de diseño base de las versiones adicionales, la funcionalidad, los adicionales y la impresión.
- **Gráfico Dona:** El gráfico de Chart.js se actualiza en tiempo real con cada cambio, proporcionando una representación visual inmediata de la proporción de cada categoría de costo sobre el total.
- **Modal de Contacto:**

- **Diseño de Botones:** Los botones de "Vía de contacto preferida" (WhatsApp, Telegram, SMS) han sido rediseñados para ser más compactos, con el icono sobre el texto (flex-col).
- **Efecto de Selección:** El botón activo se resalta con un fondo (bg-blue-600/40), un borde azul (border-blue-400) y una sombra (shadow-lg), proporcionando un feedback visual claro e inequívoco al usuario.
- **Recopilación de Datos (submitForm):** Al confirmar el pedido, el sistema recopila todos los valores del formulario y los estructura en un objeto JSON integral. Este objeto contiene metadatos (fecha, moneda), información de contacto, detalles completos del proyecto (descripción, logo, fuentes, colores), la selección de servicios, un array detallado con cada una de las versiones de destino, los addons seleccionados y un desglose completo de la cotización final. Este JSON está listo para ser enviado a un servidor o API para su procesamiento.

En resumen, la página combina un diseño de interfaz de usuario claro y reactivo con un motor de precios robusto y flexible que utiliza mecanismos matemáticos como la interpolación lineal para ofrecer una tarificación dinámica, transparente y justa al cliente.