

## **4. Manual de la aplicación (organizado por funciones)**

El presente manual describe las principales funcionalidades implementadas en la aplicación web TechSolutions, organizada en módulos y vistas. Cada función cumple un papel esencial dentro del flujo de trabajo, garantizando la interacción fluida entre el usuario, el administrador y el sistema.

### **Módulo 1: Navegación y estructura general**

#### **Función 1. Carga del sitio web (index.html)**

Al ingresar al dominio principal, el sistema carga la página de inicio que incluye la barra de navegación, el slider y la información principal de la empresa. Esta función inicializa los componentes visuales y configura la estructura base del sitio web.

#### **Función 2. Navegación entre secciones**

Permite que el usuario se desplace entre las vistas “Inicio”, “Servicios”, “Contacto” y “Login Admin” mediante la barra de navegación (navbar). Se implementa con enlaces HTML y control de rutas en JavaScript, garantizando una navegación fluida sin recargar completamente la página.

#### **Función 3. Desplazamiento interno (Scroll y anclas)**

Implementa el desplazamiento suave hacia las secciones internas del Home, como “Nosotros” o “Servicios destacados”, mejorando la experiencia de usuario y la accesibilidad.

#### **Función 4. Slider dinámico**

Muestra imágenes rotativas con mensajes promocionales en la página principal. Utiliza JavaScript para cambiar automáticamente las imágenes cada cierto intervalo o mediante botones de navegación manual.

### **Módulo 2: Gestión y visualización de servicios**

#### **Función 5. Carga dinámica de servicios**

Recupera los datos de los servicios almacenados en un archivo servicios.json o base de datos SQL y los renderiza automáticamente en la vista “Listado de servicios”. Utiliza fetch() para leer el archivo y DOM manipulation para construir dinámicamente las tarjetas de servicios.

#### **Función 6. Renderizado de tarjetas de servicio**

Cada servicio se representa en una tarjeta que incluye imagen, nombre y precio. El sistema genera estas tarjetas automáticamente a partir de los datos del archivo JSON, garantizando consistencia entre el contenido y la interfaz.

### **Función 7. Filtrado y búsqueda de servicios**

Permite al usuario aplicar filtros por nombre o categoría (si está implementado) mediante un campo de búsqueda. La función recorre el arreglo de servicios y muestra solo aquellos que coinciden con el criterio.

### **Función 8. Redirección al detalle del servicio**

Cuando el usuario hace clic en “Ver más” en una tarjeta, la aplicación redirige a la vista detalle.html, pasando el ID del servicio como parámetro. El sistema usa este valor para cargar la información correspondiente.

### **Función 9. Carga de detalle de servicio**

Obtiene del archivo JSON el registro cuyo ID coincide con el seleccionado. Muestra su imagen principal, nombre, precio, descripción, cantidad disponible y estado de promoción.

Si el campo EnPromocion es verdadero, la interfaz muestra una etiqueta visual especial.

### **Función 10. Botón “Volver al listado”**

Permite al usuario regresar al listado de servicios sin perder la sesión de navegación. Se implementa mediante una función de redirección o `window.history.back()`.

## **Módulo 3: Autenticación y seguridad**

### **Función 11. Validación del formulario de login**

Verifica que los campos de usuario y contraseña no estén vacíos antes de enviarse. Si faltan datos, muestra un mensaje de advertencia al usuario.

### **Función 12. Autenticación de administrador**

Envía los datos de login al backend (`/auth/login`) o los compara localmente con los registros almacenados en el archivo `usuarios.json`.

Si las credenciales son válidas, se genera una sesión activa y se redirige al panel administrativo (`admin.html`).

### **Función 13. Control de sesión**

Mantiene activa la sesión del administrador durante su navegación en el panel. Si se detecta inactividad o cierre de sesión, redirige automáticamente al login.

### **Función 14. Cierre de sesión (Logout)**

Permite al administrador salir de la aplicación eliminando los datos de sesión del almacenamiento local y redirigiendo al formulario de inicio de sesión.

## **Módulo 4: Panel administrativo (CRUD de servicios)**

### **Función 15. Visualización de servicios en la tabla administrativa**

Muestra todos los servicios existentes en una tabla HTML. Los datos se obtienen del archivo JSON o la base de datos mediante una consulta GET al endpoint `/api/servicios`.

### **Función 16. Creación de nuevos servicios**

Permite registrar un nuevo servicio mediante un formulario con campos: nombre, imagen, precio, cantidad, descripción y promoción.

Al guardar, los datos se envían al backend con una petición POST y se almacenan en la base de datos o se actualiza el archivo JSON.

### **Función 17. Edición de servicios existentes**

Carga los datos del servicio seleccionado en un formulario editable.

Al modificar y guardar, se actualiza la información en la base de datos mediante una petición PUT.

### **Función 18. Eliminación de servicios**

Permite eliminar servicios mediante un botón “Eliminar” junto a cada registro.

Antes de eliminar, el sistema solicita confirmación. La eliminación puede ser lógica (cambio de estado Activo = 0) o física (borrado del registro).

### **Función 19. Actualización en tiempo real del listado**

Cada vez que se crea, edita o elimina un servicio, la tabla se actualiza automáticamente sin recargar la página completa, usando técnicas de manipulación dinámica del DOM.

### **Función 20. Validación de formularios en CRUD**

Comprueba que los campos del formulario (nombre, precio, cantidad, etc.) contengan valores válidos antes de enviar la información al backend.

Evita el registro de datos vacíos o inconsistentes, mejorando la integridad del sistema.

## **Módulo 5: Funciones adicionales y de soporte**

### **Función 21. Control de errores**

Detecta errores en la carga de archivos JSON o en las peticiones al backend.

Si ocurre un error (por ejemplo, fallo de conexión), muestra un mensaje al usuario y evita la interrupción completa de la aplicación.

### **Función 22. Diseño responsive**

Ajusta automáticamente la disposición de los elementos visuales según el tamaño de la pantalla.

Implementado con CSS Flexbox y media queries, garantiza una visualización correcta tanto en computadores de escritorio como en dispositivos móviles.

### **Función 23. Carga de imágenes de servicios**

Permite al administrador subir o definir una URL para la imagen de cada servicio. El sistema almacena la ruta en el campo `ImagenUrl` de la base de datos o archivo JSON.

### **Función 24. Control de promoción**

Incluye un campo tipo checkbox en el formulario de creación/edición de servicios. Si está marcado, se activa el campo `EnPromocion = true`, lo cual se refleja visualmente con una etiqueta de color en el frontend.

### **Función 25. Sincronización con archivo JSON o base de datos**

Permite exportar o importar la información de los servicios desde el backend hacia el archivo `servicios.json`, garantizando que la información se mantenga actualizada en caso de desconexión del servidor SQL.

### **Función 26. Footer dinámico**

Muestra automáticamente el año actual y enlaces activos a las redes sociales de la empresa.

Se actualiza mediante una función JavaScript que obtiene la fecha del sistema.

### **Función 27. Enlaces externos y contacto**

Desde el Home o el Footer, el usuario puede acceder a enlaces externos de redes sociales o enviar un correo electrónico de contacto mediante `mailto:.`

## **Resumen general**

Este conjunto de funciones permite que la aplicación cumpla su propósito de ser una plataforma integral para la visualización y gestión de servicios tecnológicos.

El sistema combina una interfaz amigable con una lógica de negocio sólida, garantizando la sincronización entre los datos, la navegación intuitiva y la gestión administrativa eficiente.