# Especificaciones del Sistema Talonario

## Introducción

El sistema Talonario es una plataforma diseñada para gestionar ingresos y egresos asociados a cuentas de Convenio y Aporte Socio. Permite a los usuarios de bienestar crear y consultar registros, y a los administradores supervisar y gestionar las operaciones.

## Roles del Sistema

### 1. Administrador

Funciones del Administrador:  
- Gestionar usuarios del sistema (crear, editar, eliminar).  
- Supervisar registros creados por los usuarios de bienestar.  
- Consultar y auditar cualquier registro en el sistema.  
- Generar reportes generales del talonario.  
- Tener acceso completo a todos los registros y configuraciones.

### 2. Usuario de Bienestar

Funciones del Usuario de Bienestar:  
- Iniciar sesión en el sistema.  
- Crear registros en el talonario (ingreso o egreso) bajo las cuentas de Convenio o Aporte Socio.  
- Adjuntar los documentos requeridos para cada tipo de registro.  
- Consultar los registros creados.  
- Modificar o actualizar un registro, si es necesario, antes de finalizarlo.

## Flujo del Sistema

1. Inicio de Sesión: El usuario (con rol de bienestar o administrador) se logea con sus credenciales.  
2. Opciones tras el Login:  
 - Crear Registro: Permite iniciar el proceso de creación de un registro.  
 - Consultar Registro: Permite buscar registros existentes en el sistema.  
3. Creación de Registro: El usuario selecciona entre dos opciones de cuenta:  
 - Convenio: Puede ser Ingreso o Egreso.  
 - Aporte Socio: Puede ser Ingreso o Egreso.  
4. Formularios de Registro: Cada formulario incluye campos comunes y específicos según el tipo de operación.  
5. Finalización del Registro: Se guarda la información, se adjuntan los documentos requeridos, y se genera un número de folio único.

## Especificaciones de los Formularios

Todos los registros en el sistema comparten los siguientes campos a rellenar:  
- Fecha de creación (automática por el sistema).  
- Nombre del registro.  
- Concepto.  
- Monto.  
- Cheque N°.  
- Banco.  
- Detalle.  
- Debe.  
- Haber.  
- Total.  
Además, cada formulario incluye adjuntos específicos según el tipo de cuenta y operación:  
1. Convenio Ingreso: Giro electrónico, Orden de ingreso.  
2. Convenio Egreso: Comprobante de transferencia bancaria.  
3. Aporte Socio Ingreso: Giro electrónico, Orden de ingreso.  
4. Aporte Socio Egreso: Documento que origina el gasto.

## Numeración de Folios

El número de folio es continuo entre años y tiene el formato número\_correlativo/año. Por ejemplo:  
- Si el último folio del año 2024 es 45/2024, el primer folio del año 2025 será 46/2025.  
Este número es generado automáticamente por el sistema al guardar un registro.

**Tecnología**

**Backend:**

1. **Framework: Django (Python)**
   * **Por qué Django**:
     + Es robusto y viene con muchas herramientas integradas para manejo de formularios, validación de datos y control de usuarios.
     + Ideal para aplicaciones donde se necesita un sistema administrativo o con reglas complejas como un talonario digital.
     + Escalable y con una comunidad activa.
   * **Complementos útiles**:
     + Django REST Framework (DRF): Si necesitas una API para consumo desde frontend separado o móvil.
     + Django Admin: Para gestionar usuarios, auditorías, y operaciones básicas de los datos.
2. **Base de datos: PostgreSQL**
   * **Por qué PostgreSQL**:
     + Soporta operaciones financieras precisas con tipos de datos como NUMERIC para cálculos monetarios.
     + Compatible con Django y tiene excelente manejo de integridad referencial.

**Frontend:**

1. **Framework: React.js**
   * **Por qué React**:
     + Ideal para formularios dinámicos y componentes reutilizables (como validaciones interactivas de los campos del tanolario).
     + Permite un diseño moderno y una experiencia de usuario fluida.
   * Alternativa: Si prefieres algo más sencillo, **Django Templates** puede ser suficiente para un inicio.
2. **UI Framework: Tailwind CSS o Bootstrap**
   * **Por qué**:
     + Simplifican el diseño y ofrecen estilos predefinidos para que la plataforma se vea profesional sin invertir mucho tiempo en el diseño desde cero.

**Autenticación y Autorización:**

1. **Sistema integrado de Django**:
   * Django viene con un sistema de autenticación sólido que puedes extender para manejar roles (administradores, usuarios normales, etc.).
   * Complementa con **django-allauth** si necesitas autenticación social o manejo avanzado de usuarios.
2. **Tokens o Sesiones**:
   * Usa sesiones de Django para autenticar usuarios en un sistema basado en formularios.
   * Si usas una API, opta por **JWT** (JSON Web Tokens) con Django REST Framework.

**Almacenamiento de Archivos Adjuntos:**

1. **AWS S3 o Google Cloud Storage**:
   * Para guardar documentos como talonarios, comprobantes de transferencias, etc.
   * Alternativa local: Manejar los documentos en el servidor usando **Django FileField**, aunque no será tan escalable.

**Infraestructura:**

1. **Despliegue**:
   * Usa Docker para contenerizar tu aplicación y asegurar portabilidad.
   * Despliega en servicios cloud como **AWS**, **Heroku** o **DigitalOcean**.
   * Configura HTTPS con **Let's Encrypt** para garantizar seguridad en la transferencia de datos.
2. **Manejo de Logs y Errores**:
   * Implementa **Sentry** o **Rollbar** para monitorear errores.
   * Configura logs estructurados con herramientas como **Loguru** o el sistema de logging nativo de Python.

**Razones para esta combinación:**

* **Django + React**: Combinan un backend sólido con un frontend interactivo y moderno.
* **PostgreSQL**: Es ideal para manejar datos financieros con integridad.
* **Almacenamiento escalable**: AWS o Google Cloud te preparan para manejar grandes volúmenes de documentos.