# Manual Técnico TReDS – Estado al 11 de julio de 2025

## 1. Arquitectura General del Sistema

El sistema TReDS está construido sobre FastAPI como framework backend, utilizando Jinja2 para el renderizado del frontend.   
La base de datos es SQLite durante el desarrollo local y gestionada mediante SQLAlchemy ORM, con versiones controladas por Alembic.

## 2. Estructura de Archivos y Módulos

- main.py: punto de entrada del servidor FastAPI.  
- routers/: contiene archivos por rol (proveedor, pagador, financiador, configuración, admin).  
- models.py: define los modelos SQLAlchemy.  
- templates/: vistas HTML renderizadas con Jinja2.  
- alembic/: migraciones de la base de datos.

## 3. Modelos Principales

- FacturaDB: representa facturas subidas por proveedores.  
- OfertaFinanciamiento: ofertas de financiadores sobre facturas.  
- Financiador: incluye login seguro y configuración de condiciones automáticas.  
- CondicionesPorPagador: lógica específica de oferta por RUT pagador.

## 4. Lógica de Usuarios y Rutas

Proveedores:  
- Registro, login.  
- Carga de facturas, solicitud de confirming.  
- Aceptación de ofertas.  
  
Pagadores:  
- Registro, login.  
- Confirmación de facturas.  
- Rechazo/modificación de vencimientos.  
  
Financiadores:  
- Registro/login con clave hasheada.  
- Visualización marketplace.  
- Creación de ofertas manuales o automáticas.  
- Gestión de costo de fondos (CF mensual).  
- Gestión de condiciones por pagador.  
  
Administrador:  
- Asignación de permisos `es\_admin` a usuarios financiadores.  
- Solo admin puede acceder a carga de CF.

## 5. Seguridad y Control de Acceso

- Sistema de login con contraseñas hasheadas (bcrypt).  
- Control de acceso por sesión (session["financiador\_id"], etc.).  
- CF sólo modificable si el usuario es admin (`es\_admin=True`).

## 6. Estado Actual (11 de julio)

✅ Login y registro operativo para todos los roles.  
✅ Proveedores pueden cargar XML y ZIP con facturas.  
✅ Pagadores confirman facturas (pasan a marketplace).  
✅ Financiadores ven marketplace personalizado y ofertan.  
✅ Funcionalidad para ingresar CF mensual.  
✅ Restricción para que solo admin acceda al formulario CF.  
✅ Migraciones Alembic implementadas para CF mensual y `es\_admin`.

## 7. Archivos Modificados Recientemente

- models.py  
- routers/financiador.py  
- routers/configuracion.py  
- templates/costo\_fondos.html  
- templates/base.html  
- templates/usuarios\_financiador.html  
- alembic/versions/\*

## 8. Próximos Pasos

- Mejoras en visualización de ofertas adjudicadas.  
- Validación de tasas (no permitir tasas 0).  
- Mejora en UX y feedback en botones de acción.  
- Control de múltiples usuarios simultáneos con sesiones reales.  
- Implementación de auditoría y logs de actividad por usuario.