**📚 DOCUMENTACIÓN COMPLETA - SISTEMA DE COMPROMISOS DE GESTIÓN**

**📖 MANUAL DE USUARIO COMPLETO**

**1. 🚀 INICIO RÁPIDO**

**Primer Acceso al Sistema**

1. Abrir el navegador web e ingresar a la URL del sistema
2. **Credenciales iniciales**:
   * Usuario: admin
   * Contraseña: admin
3. Cambiar contraseña inmediatamente en el módulo de Administración

**Navegación Principal**

* **Menú lateral**: Acceso a todos los módulos
* **Header superior**: Información del usuario y logo OPP
* **Contenido central**: Área de trabajo principal

**2. 📋 MÓDULO DE ACUERDOS**

**Crear Nuevo Acuerdo**

1. Ir a "Acuerdos" en el menú lateral
2. Click en "➕ Crear nuevo acuerdo"
3. Completar datos básicos:
   * Tipo de compromiso
   * Organismo
   * Año de vigencia
   * Tipo de organismo
4. **Código automático**: El sistema genera código único

**Editar Acuerdo Existente**

1. Buscar acuerdo en la lista
2. Click en "📂 Abrir"
3. Completar secciones:
   * Información general
   * Objeto del acuerdo
   * Partes firmantes
   * Cláusulas (automáticas por tipo)

**Gestión de Fichas**

**Crear ficha manual:**

* Click en "➕ Crear Ficha Manual"
* Completar datos de la ficha:
  + Nombre descriptivo
  + Tipo de meta
  + Responsables
  + Objetivo e indicadores

**Carga masiva vía CSV:**

1. Descargar plantilla CSV
2. Completar datos en Excel
3. Subir archivo completado
4. Verificar importación

**Gestión de Metas**

Por cada ficha:

1. Click en "➕ Agregar meta"
2. Completar:
   * Descripción clara
   * Unidad de medida
   * Valor objetivo
   * Sentido (≥, ≤, ==)
   * Frecuencia (Mensual, Trimestral, etc.)
   * Ponderación (suma total 100%)

**Rangos de Cumplimiento**

Configurar para cada meta:

* Mínimo y máximo por rango
* Porcentaje asociado
* Sistema calcula automáticamente

**3. 📊 MÓDULO DE SEGUIMIENTO**

**Carga de Resultados**

1. Ir a "Seguimiento de Indicadores"
2. Seleccionar "🎯 Cargar Resultados por Meta"
3. Elegir acuerdo → ficha → meta específica
4. Ingresar valor alcanzado
5. Sistema calcula cumplimiento automático

**Dashboard de Indicadores**

* Vista gráfica de progreso
* Métricas principales
* Tablas detalladas
* Filtros por período

**Generación de Reportes**

**Reporte HTML:**

* Click en "📄 Generar Reporte"
* Vista previa inmediata
* Descarga en formato HTML
* Optimizado para impresión

**Reporte Simple:**

* Versión básica rápida
* Formato tabla simple
* Ideal para análisis rápido

**4. 📈 MÓDULO DE REPORTES**

**Reportes Consolidados**

1. Seleccionar año y filtros
2. Click en "📊 Generar Reporte Consolidado"
3. Descargar en Excel para análisis

**Vista para Imprimir**

* Formato optimizado impresión
* Incluye todas las firmas
* Diseño profesional

**Exportación Completa**

* Paquete ZIP con múltiples formatos
* Incluye JSON, CSV, Excel
* Ideal para backups

**5. ⚙️ FUNCIONES AVANZADAS**

**Flujo de Aprobación**

1. **Borrador**: Edición completa
2. **Enviar a revisión**: Click en cambiar estado
3. **Aprobaciones sucesivas**: Según rol
4. **Aprobado final**: Solo lectura

**Sistema de Firmas**

1. En edición de acuerdo, sección "📝 Firmas"
2. Subir imagen de firma escaneada
3. Completar datos del firmante
4. Aparece automáticamente en reportes

**Archivos Adjuntos**

* Subir hasta 5 archivos simultáneos
* Formatos: PDF, Word, Excel, imágenes
* Descarga individual o grupal

**👥 GUÍAS DE PROCEDIMIENTO POR ROL**

**🎯 RESPONSABLE DE ACUERDO**

**Funciones Principales**

* Crear y editar acuerdos en estado "Borrador"
* Gestionar fichas y metas
* Cargar resultados de cumplimiento
* Enviar acuerdos a revisión

**Procedimiento Típico**

**Tareas Mensuales**

1. **Primera semana**: Revisar metas vencidas
2. **Segunda semana**: Cargar resultados del período
3. **Tercera semana**: Verificar cálculos automáticos
4. **Última semana**: Preparar reportes para supervisión

**🔍 SUPERVISOR OPP**

**Funciones Principales**

* Revisar acuerdos en "Pendiente de Revisión"
* Validar o rechazar acuerdos
* Supervisar cumplimiento global
* Generar reportes ejecutivos

**Procedimiento de Revisión**

1. **Revisión inicial**: Verificar integridad datos
2. **Validación técnica**: Coherencia de indicadores
3. **Aprobación condicionada**: Solicitar ajustes si corresponde
4. **Envío a Comisión CG**: Para aprobación final

**Indicadores Clave a Monitorear**

* Cumplimiento promedio por organismo
* Metas críticas con bajo desempeño
* Tendencias temporales
* Comparativas entre organismos

**🏛️ COMISIÓN CG**

**Funciones Principales**

* Aprobación final de acuerdos
* Archivado de acuerdos cumplidos
* Análisis estratégico de resultados
* Toma de decisiones basada en datos

**Procedimiento de Aprobación**

1. **Revisión global**: Cumplimiento de requisitos
2. **Análisis de métricas**: Indicadores de gestión
3. **Aprobación/Rechazo**: Decisión formal
4. **Archivado**: Cuando se cumple ciclo completo

**Sesiones de Análisis**

* **Trimestral**: Revisión de tendencias
* **Semestral**: Ajuste de estrategias
* **Anual**: Evaluación completa y planificación

**👨‍💼 ADMINISTRADOR**

**Funciones Principales**

* Gestión completa de usuarios
* Mantenimiento del sistema
* Resolución de problemas técnicos
* Backup y recuperación

**Tareas Diarias**

1. Verificar estado del sistema
2. Revisar logs de auditoría
3. Atender solicitudes de usuarios
4. Realizar backup incremental

**Tareas Semanales**

1. Limpieza de archivos temporales
2. Verificación de integridad de datos
3. Actualización de documentación
4. Revisión de permisos y roles

**🔧 DOCUMENTACIÓN TÉCNICA PARA ADMINISTRADORES**

**1. 🏗️ ARQUITECTURA DEL SISTEMA**

**Tecnologías Utilizadas**

text

Frontend: Streamlit (Python)

Almacenamiento: JSON files + File System

Autenticación: Hash SHA256 + Salt

Cache: Streamlit caching system

**Estructura de Archivos**

text

sistema\_cg\_data/

├── users.json (usuarios y permisos)

├── agreements.json (acuerdos completos)

├── counters.json (contadores automáticos)

├── audit.json (logs de auditoría)

├── naturaleza\_map.json (mapeo naturalezas jurídicas)

└── uploads/ (archivos adjuntos)

└── [acuerdo\_id]/

└── [archivos\_subidos]

**2. ⚙️ CONFIGURACIÓN Y MANTENIMIENTO**

**Variables de Configuración**

python

*# En app.py - Líneas principales*

DATA\_DIR = os.path.join(tempfile.gettempdir(), "sistema\_cg\_data")

USERS\_FILE = os.path.join(DATA\_DIR, "users.json")

AGREEMENTS\_FILE = os.path.join(DATA\_DIR, "agreements.json")

**Roles del Sistema**

python

ROLES\_SISTEMA = [

"Administrador",

"Responsable de Acuerdo",

"Supervisor OPP",

"Comisión CG",

"Consulta"

]

**3. 🔐 ADMINISTRACIÓN DE USUARIOS**

**Crear Usuario Nuevo**

1. Ir a "Administración del Sistema"
2. Sección "➕ Crear Nuevo Usuario"
3. Completar:
   * Usuario (único)
   * Nombre completo
   * Contraseña temporal
   * Rol asignado

**Resetear Contraseña**

1. Buscar usuario en lista
2. Click en "🔐 Cambiar Contraseñas"
3. Ingresar nueva contraseña
4. Confirmar cambios

**Gestión de Permisos**

* **Administrador**: Acceso completo
* **Responsable**: Crear/editar sus acuerdos
* **Supervisor**: Revisar y validar
* **Comisión CG**: Aprobar y archivar
* **Consulta**: Solo lectura

**4. 🛠️ HERRAMIENTAS DE DIAGNÓSTICO**

**Panel de Diagnóstico**

Accesible desde sidebar cuando hay problemas:

**Verifica:**

* ✅ Permisos de escritura
* ✅ Archivos críticos existentes
* ✅ Espacio en disco
* ✅ Estados del sistema

**Comandos de Mantenimiento**

python

*# En la consola de Streamlit*

check\_permissions() *# Verificar permisos*

limpiar\_caches() *# Limpiar caché*

reset\_counters() *# Reiniciar contadores*

**5. ❌ RESOLUCIÓN DE PROBLEMAS COMUNES**

**Problema: No se guardan los datos**

**Solución:**

1. Verificar permisos de escritura en DATA\_DIR
2. Usar herramienta "🔧 Intentar reparar permisos"
3. Verificar espacio en disco

**Problema: Usuario no puede acceder**

**Solución:**

1. Verificar que usuario esté "activo"
2. Revisar contraseña en "Cambiar Contraseñas"
3. Verificar rol asignado

**Problema: Cálculos incorrectos**

**Solución:**

1. Verificar rangos de cumplimiento en metas
2. Revisar valores objetivo vs alcanzados
3. Usar "Calcular cumplimiento" manual

**Problema: Lentitud del sistema**

**Solución:**

1. Limpiar caché con limpiar\_caches()
2. Verificar tamaño de archivos JSON
3. Eliminar archivos temporales antiguos

**💾 PROTOCOLOS DE BACKUP Y RECUPERACIÓN**

**1. 📅 PROTOCOLO DE BACKUP DIARIO**

**Backup Automático**

**Archivos a respaldar:**

* users.json (usuarios y permisos)
* agreements.json (acuerdos completos)
* counters.json (contadores)
* audit.json (logs auditoría)
* Directorio uploads/ (archivos adjuntos)

**Procedimiento Manual**

bash

*# Crear directorio de backup con fecha*

BACKUP\_DIR="backup\_$(date +%Y%m%d\_%H%M%S)"

mkdir -p $BACKUP\_DIR

*# Copiar archivos críticos*

cp sistema\_cg\_data/\*.json $BACKUP\_DIR/

*# Copiar uploads (si existen)*

cp -r sistema\_cg\_data/uploads $BACKUP\_DIR/ 2>/dev/null || true

*# Comprimir backup*

tar -czf ${BACKUP\_DIR}.tar.gz $BACKUP\_DIR

*# Verificar integridad*

tar -tzf ${BACKUP\_DIR}.tar.gz

**2. 🔄 PROTOCOLO DE RECUPERACIÓN**

**Recuperación Completa**

**Prerrequisitos:**

* Sistema reinstalado o limpio
* Misma versión de la aplicación
* Permisos de escritura en directorio

**Procedimiento:**

1. **Detener** la aplicación
2. **Extraer** backup: tar -xzf backup\_YYYYMMDD\_HHMMSS.tar.gz
3. **Restaurar archivos**:

bash

cp backup/\*.json sistema\_cg\_data/

cp -r backup/uploads sistema\_cg\_data/ 2>/dev/null || true

1. **Verificar permisos**: chmod 664 sistema\_cg\_data/\*.json
2. **Reiniciar** aplicación

**Recuperación Parcial**

**Caso: Pérdida de acuerdos específicos**

1. Extraer backup
2. Identificar acuerdo en agreements.json del backup
3. Copiar objeto del acuerdo al archivo actual
4. Reindexar contadores si es necesario

**3. 📊 VERIFICACIÓN DE INTEGRIDAD**

**Verificación Post-Recuperación**

1. **Usuarios**: Login con diferentes roles
2. **Acuerdos**: Abrir varios acuerdos y verificar datos
3. **Cálculos**: Verificar cumplimientos automáticos
4. **Archivos**: Descargar archivos adjuntos de prueba

**Script de Verificación**

python

def verificar\_integridad\_backup():

"""Verifica integridad después de recuperación"""

*# Verificar archivos críticos*

archivos\_criticos = [USERS\_FILE, AGREEMENTS\_FILE, COUNTERS\_FILE]

for archivo in archivos\_criticos:

if not os.path.exists(archivo):

return f"❌ Faltante: {archivo}"

*# Verificar estructura JSON*

try:

users = load\_json(USERS\_FILE, {})

agreements = load\_json(AGREEMENTS\_FILE, {})

if not isinstance(users, dict) or not isinstance(agreements, dict):

return "❌ Estructura JSON corrupta"

except:

return "❌ Error leyendo archivos JSON"

return "✅ Integridad verificada correctamente"

**4. 🗓️ CALENDARIO DE BACKUP**

**Frecuencias**

* **Backup Diario**: Lunes a Viernes - 20:00 hs
* **Backup Semanal**: Viernes - 21:00 hs (completo)
* **Backup Mensual**: Primer día del mes - 22:00 hs
* **Verificación**: Primer lunes de cada mes

**Retención**

* **Diarios**: 7 días
* **Semanales**: 4 semanas
* **Mensuales**: 12 meses
* **Anuales**: Permanente (fin de año)

**5. 🚨 PROTOCOLO DE EMERGENCIA**

**Escenario: Pérdida Total**

1. **Comunicar**: Notificar a todos los usuarios
2. **Evaluar**: Identificar último backup válido
3. **Recuperar**: Usar protocolo de recuperación completa
4. **Verificar**: Validar integridad de datos
5. **Comunicar**: Notificar restauración completada

**Contactos de Emergencia**

* **Responsable Técnico**: [Nombre] - [Teléfono]
* **Backup Técnico**: [Nombre] - [Teléfono]
* **Dirección OPP**: [Nombre] - [Teléfono]

**6. 📝 CHECKLIST DE BACKUP**

**Checklist Diario**

* Backup automático ejecutado
* Verificación de archivos críticos
* Espacio en disco suficiente
* Logs sin errores críticos

**Checklist Semanal**

* Backup completo comprimido
* Verificación de integridad
* Rotación de backups antiguos
* Revisión de logs de sistema

**Checklist Mensual**

* Prueba de recuperación en ambiente de prueba
* Actualización de documentación
* Revisión de políticas de retención
* Reporte de estado a dirección

**📞 SOPORTE Y CONTACTO**

**Canales de Soporte**

**Horarios de Soporte**

**Documentación Generada**: {fecha\_actual}  
**Versión del Sistema**: 2.0  
**Última Revisión**: {fecha\_actual}