**Sistema de Compromisos de Gestión - Uruguay**

Sistema integral para la gestión, seguimiento y evaluación de compromisos de gestión del Estado Uruguayo. Desarrollado con Streamlit para una interfaz web moderna y accesible.

1. **Características Principales**
   1. **Gestión de Acuerdos**

* **Creación y edición** de acuerdos de compromisos de gestión
* **Múltiples tipos** de compromiso (CG Institucional, CG Funcional, EEPP, etc.)
* **Gestión completa** de organismos y naturaleza jurídica
* **Sistema de versionado** con historial de cambios
* **Flujo de aprobación** con roles y permisos
  1. **Fichas y Metas**
* **Fichas de compromiso** con objetivos e indicadores
* **Metas asociadas** con ponderación y seguimiento
* **Sistema de rangos** de cumplimiento configurables
* **Cálculo automático** de porcentajes de cumplimiento
* **Carga masiva** mediante archivos CSV
  1. **Seguimiento y Evaluación**
* **Dashboard interactivo** con métricas en tiempo real
* **Cálculo automático** de cumplimientos ponderados
* **Sistema de estados** para metas (No Iniciada, En Progreso, Cumplida, etc.)
* **Indicadores visuales** de progreso
* **Alertas y notificaciones** de cumplimiento
  1. **Reportes y Análisis**
* **Reportes ejecutivos** en múltiples formatos (HTML, Excel, PDF)
* **Vistas imprimibles** optimizadas
* **Métricas globales** de cumplimiento
* **Análisis comparativos** por período
* **Exportación completa** de datos

1. **Tecnologías Utilizadas**

* **Frontend:** Streamlit 1.28+
* **Backend:** Python 3.8+
* **Almacenamiento:** JSON (estructurado en sistema de archivos)
* **Autenticación:** Sistema propio con hash SHA256
* **Procesamiento de datos:** Pandas, NumPy
* **Exportación:** Excel, HTML, CSV, ZIP

1. **Instalación**
   1. **Prerrequisitos**

* Python 3.8 o superior
* pip (gestor de paquetes de Python)
  1. **Pasos de instalación**

1. **Clonar el repositorio**

bash

git clone https://github.com/tu-usuario/sistema-compromisos-gestion.git

cd sistema-compromisos-gestion

1. **Crear entorno virtual (recomendado)**

bash

python -m venv venv

source venv/bin/activate *# Linux/Mac*

*# o*

venv\Scripts\activate *# Windows*

1. **Instalar dependencias**

bash

pip install -r requirements.txt

1. **Ejecutar la aplicación**

bash

streamlit run app.py

* 1. **Dependencias principales**

txt

streamlit>=1.28.0

pandas>=1.5.0

openpyxl>=3.0.0

xlsxwriter>=3.0.0

1. **Roles del Sistema**
   1. **Administrador**

* Gestión completa de usuarios
* Administración del sistema
* Acceso a todas las funcionalidades
* Reseteo de contadores y correcciones
  1. **Supervisor**
* Supervisión de acuerdos y cumplimientos
* Aprobación y revisión de estados
* Acceso a reportes ejecutivos
* Monitoreo global del sistema
  1. **Responsable de Acuerdo**
* Creación y edición de acuerdos
* Gestión de fichas y metas
* Carga de resultados de seguimiento
* Seguimiento de cumplimientos
  1. **Comisión CG**
* Revisión y validación de acuerdos
* Análisis de cumplimientos
* Generación de reportes institucionales
* Seguimiento de metas críticas
  1. **Usuario de Consulta**
* Visualización de acuerdos aprobados
* Acceso a reportes públicos
* Consulta de estados de cumplimiento

1. **Estructura del Sistema**
   1. **Arquitectura de Datos**

text

sistema\_cg\_data/

├── users.json # Usuarios y permisos

├── agreements.json # Base de datos de acuerdos

├── counters.json # Contadores globales

├── audit.json # Log de auditoría

├── naturaleza\_map.json # Mapeo de naturalezas jurídicas

└── uploads/ # Archivos adjuntos

└── [acuerdo\_id]/

└── [archivos]

* 1. **Estructura de un Acuerdo**

json

{

"id": "AC\_0001\_2024",

"tipo\_compromiso": "CG - Institucional",

"organismo\_nombre": "Nombre del Organismo",

"anio": 2024,

"estado": "Borrador",

"fichas": [

{

"id": "F\_0001\_AC0001\_2024",

"nombre": "Nombre de la ficha",

"metas": [

{

"id": "M\_0001\_F0001\_AC0001\_2024",

"descripcion": "Descripción de la meta",

"ponderacion": 25.0,

"cumplimiento\_calc": 85.5,

"rangos\_cumplimiento": {

"cumplido": 90,

"parcial": 60

}

}

]

}

]

}

**6. Flujos de Trabajo**

**6.1. Creación de Acuerdos**

text

Crear Acuerdo → Completar Información → Agregar Fichas → Definir Metas → Validar → Enviar a Aprobación

**6.2. Seguimiento de Metas**

text

Cargar Resultados → Calcular Cumplimiento → Clasificar Estado → Generar Reportes → Análisis Ejecutivo

**6.3. Flujo de Aprobación**

text

Borrador → Pendiente de Revisión → En Revisión OPP → En Revisión Comisión CG → Aprobado/Rechazado

* 1. **Sistema de Autenticación**

**7.1.Credenciales Iniciales**

* **Usuario:** admin
* **Contraseña:** admin

**7.2.Seguridad**

* Contraseñas hasheadas con SHA256 y salt
* Sesiones manejadas por Streamlit
* Auditoría de acciones críticas
* Validación de permisos por rol
  1. **Módulos Principales**

**8.1. Dashboard de Indicadores**

* Métricas principales de cumplimiento
* Gráficos de tendencia
* Tablas de seguimiento
* Análisis ejecutivo automático

**8.2. Gestión de Acuerdos**

* Creación y edición completa
* Sistema de versionado
* Flujo de aprobación
* Gestión de adjuntos

**8.3. Sistema de Reportes**

* Reportes consolidados por año
* Exportación múltiple formatos
* Vistas imprimibles
* Métricas globales

**8.4. Administración**

* Gestión de usuarios
* Diagnóstico del sistema
* Limpieza y mantenimiento
* Configuración de rangos

**9. Estados del Sistema**

**9.1. Estados de Acuerdo**

* **Borrador:** En edición inicial
* **Pendiente de Revisión:** Listo para revisión
* **En Revisión OPP:** En evaluación por OPP
* **En Revisión Comisión CG:** En evaluación por Comisión
* **Aprobado:** Aprobado formalmente
* **Rechazado:** Requiere modificaciones
* **Archivado:** Finalizado o histórico

**9.2. Estados de Meta**

* **No Iniciada:** Meta sin comenzar
* **En Progreso:** En ejecución
* **Cumplida:** 90%+ de cumplimiento
* **Parcialmente Cumplida:** 60%-89% de cumplimiento
* **No Cumplida:** Menos del 60% de cumplimiento
* **Verificada:** Cumplimiento confirmado

**10. Configuración**

**10.1. Variables de Entorno**

El sistema utiliza configuración por defecto, pero puede extenderse con:

* DATA\_DIR: Directorio de almacenamiento de datos
* LOG\_LEVEL: Nivel de logging del sistema

**10.2. Personalización**

* Rangos de cumplimiento configurables por meta
* Tipos de organismo personalizables
* Formatos de reporte adaptables
* Flujos de aprobación modificables

**11. Métricas y Cálculos**

**11.1. Cálculo de Cumplimiento**

text

Cumplimiento (%) = (Valor Alcanzado / Valor Objetivo) × 100

**11.2. Cumplimiento Ponderado**

text

Cumplimiento Ponderado = Σ(Cumplimiento Meta × Ponderación Meta) / Σ(Ponderaciones)

**11.3. Clasificación por Rangos**

* **✅ Cumplido:** ≥ 90%
* **🟡 Parcial:** 60% - 89%
* **🔴 No Cumplido:** < 60%

**12. Mantenimiento**

**12.1. Tareas Periódicas**

* **Respaldos:** Copia de seguridad de sistema\_cg\_data/
* **Limpieza:** Archivos temporales y cache
* **Actualización:** Dependencias de Python
* **Auditoría:** Revisión de logs del sistema

**12.2. Solución de Problemas Comunes**

1. **Problemas de permisos:**

bash

chmod -R 755 sistema\_cg\_data/

1. **Reinicio de contadores:**
   * Usar función "Reset Counters" en administración
2. **Usuario bloqueado:**
   * Editar users.json y establecer "active": true

**12.3. Desarrollo**

1. Fork del repositorio
2. Crear rama de feature (git checkout -b feature/nueva-funcionalidad)
3. Commit de cambios (git commit -am 'Agregar nueva funcionalidad')
4. Push a la rama (git push origin feature/nueva-funcionalidad)
5. Crear Pull Request

**13. Contexto Institucional**

Sistema desarrollado para el seguimiento de Compromisos de Gestión del Estado Uruguayo, alineado con las normativas y procesos de la Oficina de Planeamiento y Presupuesto (OPP) y la Comisión de Compromisos de Gestión.