

PROMPT MAESTRO PARA GITHUB COPILOT

Contexto general del proyecto

Estás trabajando dentro de un **editor VS Code** sobre un sistema ya existente llamado **SIGE – Sistema de Gestión de Eventos Científicos**. El sistema gestiona eventos científicos completos: registro de participantes, envío y revisión de resúmenes, gestión comercial, talleres, encuestas, transporte, acreditaciones, certificados, seguridad y administración.

El sistema tiene **dos grandes componentes**: 1. **Sitio Web (Drupal 10.x)** – orientado a participantes, ponentes, árbitros y comité organizador. 2. **Sistema de Gestión Interno (Backoffice)** – orientado a roles administrativos (SuperAdmin, Comercial, Organización, etc.).

Tu objetivo como agente (GitHub Copilot) es **completar, corregir, refactorizar o generar funcionalidades faltantes de forma autónoma, fiable y coherente**, respetando estrictamente el comportamiento funcional descrito en este documento.

Reglas estrictas para Copilot

- ❌ NO inventes funcionalidades
 - ❌ NO cambies flujos de negocio
 - ❌ NO elimines validaciones existentes
 - ✅ Respetar estados, roles y transiciones
 - ✅ Usar nombres semánticos coherentes
 - ✅ Priorizar consistencia antes que creatividad
 - ✅ Todo cambio debe alinearse con este manual
-

Arquitectura conceptual

Roles principales

Sitio Web (Drupal) - Participante - Ponente - Póster - Conferencista - Árbitro - Comité Organizador

Sistema de Gestión - SuperAdmin - Comercial - Administrador - Operador

Cada rol tiene **permisos estrictos** definidos por: - Grupos de recursos - Recursos - Roles

Flujo de vida del participante

1. Registro en el sitio web

2. Estado inicial: Registrado
 3. Revisión administrativa
 4. Aprobación → Aprobado
 5. Confirmación de pago → Confirmado
 6. Acreditación (manual o QR) → Acreditado
 7. Emisión de certificados
-

Estados globales (NO modificar)

Usuario

- Registrado
- Aprobado
- Confirmado
- Acreditado
- Cancelado

Resumen científico

- En proceso
- Aprobado
- Aprobado con cambios
- Rechazado

Ómnibus

- Creado
 - Transportando
 - Finalizado
-

Módulo Gestión Comercial

Registros

- Tabla dinámica con columnas configurables
- Filtros por columna
- Estados, transporte, estadía

Confirmar pago

- Opciones:
 - Pendiente de pago (con observaciones)
 - Pagado (con comprobante obligatorio)
 - Al confirmar → estado Confirmado

Acreditar

- Solo usuarios
- Individual o múltiple
- Compatible con escaneo QR

Credenciales

- Generación masiva en PDF
- Solo usuarios o

Sincronización

- Actualiza datos desde el sitio web Drupal
-

Módulo Distribución (Ómnibus)

- Crear ómnibus con:
 - Identificador
 - Capacidad
 - Fecha y hora
 - Asignar pasajeros:
 - Solo
 - Solo con transporte solicitado
 - Editar / eliminar solo si estado
 - Cambios de estado automáticos según escaneo
-

Módulo Talleres

- Crear talleres con:
 - Identificador
 - Nombre
 - Descripción
 - Sesiones:
 - Horario único
 - Horario por día
 - Control de asistencia
 - Registro de reacciones
-

Módulo Encuestas

- Crear / editar / eliminar encuestas
- Tipos de preguntas:

- Selección simple
 - Selección múltiple
 - Texto largo
 - Numérica
-

Módulo Administración

Seguridad

Recursos

- Agrupación lógica de permisos

Roles

- Asignación de grupos de recursos
- Activación / desactivación

Usuarios

- CRUD completo
 - Asignación de roles
 - Cambio de contraseña
-

Gestión de Resúmenes Científicos (Drupal)

Envío de resumen

Campos obligatorios: - Título - Simposio - Resumen (≤ 3000 palabras) - Palabras clave (≥ 3)

Opcionales: - ORCID - Autores secundarios

Versionado

- Cambios de contenido \rightarrow versión +1
 - Cambios de título/simposio \rightarrow resumen nuevo
-

Arbitraje

- Asignación manual por comité
- Notificación automática por correo
- Evaluación con motivo obligatorio si:
- Aprobado con cambios

- Rechazado
-

Notificaciones

- Todos los cambios de estado generan correo
 - Plantillas configurables
 - Cola de correos persistente
-

Reglas de implementación

- Validaciones en frontend y backend
 - Estados como ENUM o constantes
 - Auditoría de acciones
 - Logs visibles en tiempo real
 - Tareas programadas configurables
-

Arquitectura CMS tipo Drupal (obligatorio)

El sistema **SIGE** debe comportarse como un **CMS completo**, similar a Drupal, donde el **contenido**, la **estructura** y los **permisos** están desacoplados del código.

Esto implica que el **administrador NO depende de desarrollo** para: - Crear páginas - Editar menús - Publicar artículos - Gestionar bloques - Controlar visibilidad por rol

Jerarquía de permisos (muy importante)

SuperAdmin (control total)

El **SuperAdmin** es el rol con **máximo nivel de autoridad** y puede:

- Gestionar todos los módulos
- Crear / editar / eliminar:
 - Páginas
 - Artículos
 - Menús
 - Bloques
 - Idiomas
 - Roles
 - Recursos
- Plantillas de correo
- Acceder a configuración avanzada del sistema

- Forzar cambios de estado
- Ver y eliminar logs

⚠ Ningún otro rol puede modificar estructura CMS.

Estructura CMS (modelo Drupal)

1. Tipos de contenido

El sistema debe soportar **tipos de contenido configurables**, por ejemplo:

- Página básica
- Artículo / Noticia
- Convocatoria
- Taller
- Evento
- Simposio
- Patrocinador
- Página informativa

Cada tipo de contenido tiene: - Campos dinámicos - Estado (Borrador / Publicado) - Idioma - Autor - Fecha

2. Páginas

Las **páginas** representan rutas del sistema.

Ejemplos: - /inicio - /convocatoria - /programa - /inscripcion - /contacto

Funciones del admin: - Crear página - Editar contenido - Definir URL (slug) - Definir visibilidad por rol - Publicar / despublicar

3. Artículos / Noticias

Funcionan como contenido dinámico:

- Asociados a categorías
- Con imagen destacada
- Compartibles en redes
- Listables automáticamente

El sistema debe permitir: - CRUD completo - Programación de publicación

4. Menús

Sistema de menús jerárquico tipo Drupal.

Características: - Menú principal - Menú secundario - Menú footer

Cada ítem: - Texto - Ruta interna o externa - Orden - Rol visible

5. Bloques

Los bloques son componentes reutilizables:

Ejemplos: - Banner - Slider - Texto HTML - Accesos rápidos - Noticias recientes

Funciones: - Crear bloque - Asignar región - Controlar visibilidad por rol y página

Gestión de Idiomas

Sistema multilenguaje: - Idiomas configurables - Traducción de contenidos - Traducción de menús - Recursos de localización

Integración CMS + Sistema de Gestión

El CMS **alimenta visualmente** al sistema:

- Los eventos se muestran como páginas
 - Los talleres se listan dinámicamente
 - Los resúmenes generan vistas públicas
 - Los certificados se publican por perfil
-

Flujo técnico recomendado

- Backend desacoplado
 - API REST
 - Frontend consume contenido dinámico
 - Estados controlados por backend
-

Objetivo final

Este sistema debe permitir que:

- El **SuperAdmin** administre TODO sin código
- Los **admins** gestionen operación
- Los **usuarios** interactúen con contenido
- El sistema sea escalable y reutilizable

Este documento sigue siendo la **fuentes de verdad absoluta**.

Si existe conflicto entre código y documentación → **gana este archivo**.