DOCUMENTACIÓN DE PROMPTS

MÓDULO: **Redacción de los requerimientos del sistema**

#	PROMPT
1	Ayudame a redactar de manera más limpia y entendible los siguientes requerimientos, tambien podrias complementar con tus ideas para que la aplicación tenga un plus más significativo ########################## REQUERIMIENTOS ############################## Desarrollar una aplicación web que lleve el control de los becados de IRSI Es decir que esta aplicación servirá como un histórico de todas las personas que enviaron la solicitud para la beca y los que fueron becados, llevará el registro de los estudiantes, el registro de los estudiantes que fueron aceptados, el registro de los estudiantes que desertaron de la beca La beca es para dos cursos, ciberseguridad y desarrollador junior Esta beca está disponible para los estudiantes de latinoamérica Es posible que estudiantes años después soliciten de nuevo la solicitud, hay que manejar eso, llevar un histórico de eso La aplicación constará de los siguientes roles admin, asistente, director, consulta la aplicación web se estará desarrollando con python, pyodbc, flask y mssql server llevará el área de repostería se podrán cargar los estudiantes por medio de archivos de excel, csv la aplicación contará con estándares de seguridad muy altos, como lo es la inyección de código, OWASP, y otros estándares de seguridad debe de contar con una interfaz bonita utilizando bootstrap, esta aplicación será usada solo por administrativos, no por estudiantes becados y solicitantes, los datos de los solicitantes serán ingresados por el administrador y ayudantes
2	Ahora convierte ese documento generado en un documento profesional entregable, agrega toda la estructura necesaria

MÓDULO: **Documentación básica, diagramas de secuencia, actividades, casos de uso, HU**

#	PROMPT
1	En base al documento de requerimientos quiero que generes la siguiente documentación Diagramas de casos de uso (crea el script con PlantUML genera cada caso de uso por módulos) Diagramas de Secuencia (crea el script con PlantUML) Diagramas de actividades (crea el script con PlantUML) Historias de Usuario

	Se profesional

MÓDULO: **ARQUITECTURA Y CONFIGURACIÓN INICIAL**

#	PROMPT
1	Necesito crear la estructura de directorios para un sistema web de gestión de becas usando Flask + SQLAlchemy + Bootstrap. Requerimientos: - Patrón MVC - Módulos: usuarios, solicitantes, becados, reportes, configuración - Stack: Python 3.9+, Flask 2.3+, SQLAlchemy, Bootstrap 5.3+ - Base de datos: SQL Server Genera: 1. Estructura completa de directorios 2. Archivos __init__.py necesarios 3. Archivo requirements.txt con versiones específicas 4. Configuración básica de Flask app.py 5. Archivo .env.example con variables de entorno
2	Necesito configurar SQLAlchemy para conectar con SQL Server usando PyODBC en un proyecto Flask. Contexto: - Sistema de gestión de becas con múltiples módulos - Necesidad de auditoría y logging - Configuración flexible para desarrollo/producción Genera: 1. Clase de configuración (config.py) con diferentes entornos 2. Inicialización de SQLAlchemy con Flask 3. Configuración de conexión SQL Server con PyODBC 4. Setup básico para migraciones 5. Ejemplo de conexión y manejo de errores

MÓDULO: **MODELADO DE DATOS**

#	PROMPT
1	Necesito crear los modelos SQLAlchemy para el sistema de usuarios con 4 roles específicos. Especificaciones: - Roles: Administrador, Director, Asistente, Consulta - Autenticación con hash de contraseñas (bcrypt) - Control de sesiones y bloqueo por intentos fallidos - Campos: id, nombre, email, password_hash, rol, activo, fecha_creacion, ultimo_acceso, intentos_fallidos Genera: 1. Modelo User con todos los campos necesarios 2. Métodos para hash y verificación de contraseña 3. Método para verificar roles y permisos 4. Modelo AuditoriaLogin para logs de acceso 5. Relaciones necesarias entre modelos
2	Necesito crear el modelo SQLAlchemy para solicitantes de becas con información completa. Campos requeridos: - Datos personales: nombre, documento, fecha_nacimiento, género, país, ciudad, teléfonos, emails - Info académica: nivel_educativo, institución, promedio, experiencia_tech - Datos socioeconómicos: situación_laboral, ingresos, acceso_tecnología, dependientes - Info programa: programa_solicitado, modalidad, disponibilidad, motivación, objetivos - Control: estado, fecha_registro, fecha_actualizacion Genera: 1. Modelo Solicitante completo 2. Enums para estados, países, programas 3. Validaciones de campos críticos 4. Métodos de búsqueda y filtrado 5. Modelo para historial de estados
3	Necesito completar los modelos para becados y sistema de auditoría. Especificaciones Becados: - Conversión desde solicitante aprobado - Estados: Activo, En pausa, Desertor, Graduado, Suspendido - Datos adicionales: cohorte, fecha_inicio, modalidad, sede - Referencia a plataforma académica externa

	Especificaciones Auditoría: - Log completo de actividades críticas - Cambios en datos (antes/después) - Información de sesión (IP, timestamp, usuario) Genera: 1. Modelo Becado con relaciones 2. Modelo AuditoriaActividad completo 3. Modelo EstadoBecado para historial 4. Triggers/métodos automáticos de auditoría 5. Índices para optimización de consultas

MÓDULO: **AUTENTICACIÓN Y AUTORIZACIÓN**

#	PROMPT
1	Necesito implementar el sistema completo de autenticación para Flask. Requerimientos: - Login/logout con validación - Hash de contraseñas con bcrypt - Control de sesiones (2 horas timeout) - Bloqueo por intentos fallidos (5 intentos, 30 min) - Recuperación de contraseña por email Genera: 1. Formularios de login/registro con WTForms 2. Rutas de autenticación (/login, /logout, /reset) 3. Decoradores para proteger rutas 4. Middleware para manejo de sesiones 5. Funciones de utilidad para validación
2	Necesito implementar el sistema de autorización basado en roles. Matriz de permisos: - Admin: Control total - Director: Supervisión y aprobaciones - Asistente: Operaciones diarias - Consulta: Solo lectura de reportes Funcionalidades por rol según documento adjunto. Genera: 1. Decorador @require_role para proteger rutas

	2. Funciones de verificación de permisos 3. Middleware para inyectar permisos en templates 4. Sistema de navegación dinámico por rol 5. Manejo de errores 403 personalizado

MÓDULO: **MÓDULOS PRINCIPALES**

#	PROMPT
1	Necesito crear el módulo completo de gestión de usuarios internos. Funcionalidades: - CRUD de usuarios con validaciones - Cambio de roles con confirmación - Gestión de estados (activo/inactivo) - Panel de actividad y auditoría - Reseteo de contraseñas Genera: 1. Controlador users.py con todas las rutas 2. Templates HTML con Bootstrap (listado, formularios, detalle) 3. Formularios WTForms con validaciones 4. JavaScript para confirmaciones y AJAX 5. Integración con sistema de auditoría
2	Necesito el backend completo para gestión de solicitantes. Funcionalidades críticas: - Registro de solicitantes con formulario multi-paso - Gestión de estados con workflow - Detección automática de re-aplicaciones - Búsqueda avanzada con múltiples filtros - Carga de documentos con validación Genera: 1. Controlador solicitantes.py con todas las rutas 2. Servicios para lógica de negocio compleja 3. Utilidades para detección de duplicados 4. Funciones de búsqueda y filtrado avanzado 5. Manejo de archivos y validaciones
3	Necesito las interfaces para el módulo de solicitantes. Requerimientos UI: - Dashboard con estadísticas visuales

	- Formulario multi-paso responsive - Tabla con búsqueda avanzada y filtros - Vista de comparación entre aplicaciones - Carga de archivos drag & drop Genera: 1. Templates HTML con Bootstrap 5.3 2. JavaScript para formulario multi-paso 3. Componente de búsqueda avanzada 4. Visualizaciones con Chart.js 5. Interfaz de carga de archivos moderna
	Necesito el módulo completo de gestión de becados. Funcionalidades: - Conversión automática desde solicitante aprobado - Control de estados administrativos - Timeline de actividades - Panel de comunicaciones internas - Análisis de deserción Genera: 1. Controlador becados.py completo 2. Servicio de conversión solicitante→becado 3. Templates para gestión de estados 4. Interface de comunicaciones internas 5. Reportes de seguimiento y deserción

MÓDULO: **REPORTERÍA Y ANALYTICS**

#	PROMPT
1	Necesito el motor de reportería completo. Tipos de reportes: - Operativos: estadísticas de solicitantes, control de becados - Estratégicos: dashboard ejecutivo, análisis de impacto - Predictivos: proyecciones, identificación de riesgos Funcionalidades: - Filtros dinámicos y personalizables - Exportación múltiples formatos (PDF, Excel, CSV) - Reportes programados con email - Cache inteligente para performance

	Genera: 1. Controlador reportes.py con lógica compleja 2. Servicios de generación de reportes 3. Utilidades de exportación (PDF, Excel) 4. Sistema de cache y optimización 5. Scheduler para reportes automáticos
2	Necesito crear dashboards interactivos con métricas en tiempo real. Visualizaciones requeridas: - KPIs principales con semáforos de alerta - Gráficos de tendencias temporales - Mapas geográficos de distribución - Análisis de cohortes y deserción - Comparativas entre períodos Genera: 1. Templates de dashboard responsivo 2. JavaScript para gráficos interactivos (Chart.js) 3. Componentes de KPIs con alertas 4. Mapas interactivos para distribución geográfica 5. Sistema de actualización automática de datos

MÓDULO: **IMPORTACIÓN/EXPORTACIÓN**

#	PROMPT
1	Necesito implementar importación masiva de datos robusta. Requerimientos: - Soporte Excel (.xlsx) y CSV con diferentes encodings - Validación en tiempo real con preview - Detección de duplicados automática - Mapeo flexible de columnas - Procesamiento por lotes para archivos grandes - Rollback completo en caso de errores Genera: 1. Servicio de importación con pandas/openpyxl 2. Validadores de datos y duplicados 3. Interface de mapeo de columnas 4. Sistema de preview y confirmación

	5. Logs detallados y manejo de errores
2	Necesito un sistema completo de exportación de datos. Funcionalidades: - Exportación selectiva con filtros avanzados - Múltiples formatos (Excel, CSV, PDF, JSON) - Reportes formateados para presentación - Exportación programada automática - Compresión para archivos grandes Genera: 1. Servicios de exportación por formato 2. Generador de PDFs con reportlab 3. Sistema de filtros y selección de columnas 4. Scheduler para exportaciones automáticas 5. Interface de descarga y gestión de archivos

MÓDULO: **TESTING Y DOCUMENTACIÓN**

#	PROMPT
1	Necesito crear una suite completa de testing. Tipos de pruebas: - Unit tests para modelos y servicios - Integration tests para APIs - Functional tests para workflows críticos - Security tests para vulnerabilidades - Performance tests para carga Genera: 1. Tests unitarios con pytest 2. Tests de integración para APIs críticas 3. Tests funcionales para workflows 4. Tests de seguridad automatizados 5. Configuración de CI/CD básica

MÓDULO: **DISEÑOS DE VENTANAS**

#	PROMPT
1	Para mejorar el diseño de las ventanas se hizo con claude y se siguió el siguiente prompt: A partir de este template html quiero que me mejores el diseño utilizando bootstrap, que sea profesional, utiliza colores asociados con la empresa IRSI /*Incorporar el template/ Puedes agregar información si crees necesaria y conveniente para que le dé más valor a la aplicación, te dejo las rutas y el modelo asociados al template /*Incorporar rutas y el modelo*/