

Documento de Diseño de Software (DDS) - EC-02

Proyecto: Sistema de Áreas de Conocimiento y Planificación Académica

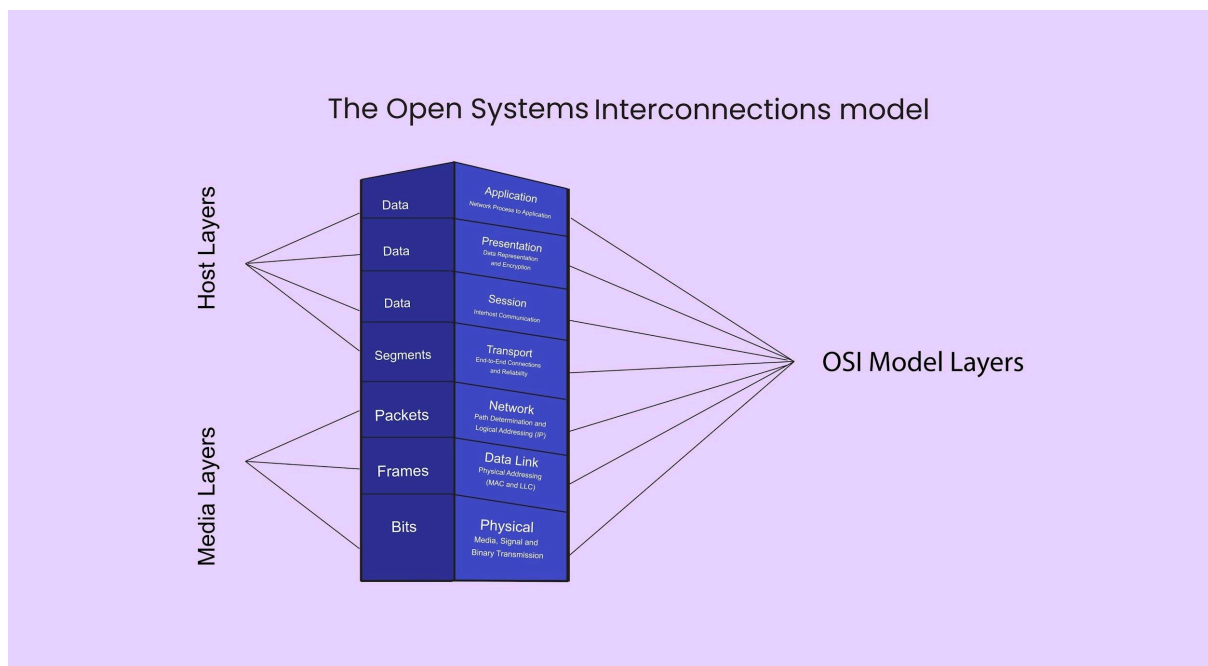
Versión: 1.0

Responsable: Arquitecto de Software

1. Arquitectura del Sistema

Se propone una **Arquitectura de Microservicios** o **Monolito Modular** (dependiendo de la escala) basada en capas para separar la lógica de negocio de la interfaz de usuario.

- **Capa de Presentación (Frontend):** Desarrollada en React o Angular, consumiendo servicios REST.
- **Capa de Aplicación (Backend):** Implementada en Node.js o Spring Boot.
- **Capa de Datos:** PostgreSQL para la persistencia de datos relacionales (NRS, docentes, horarios).



2. Diseño de la Base de Datos

El modelo debe ser relacional para garantizar la integridad de las reglas de negocio (ej. no exceder las 22 horas de un docente).

Entidades Principales:

- **Docente:** ID, Nombre, Especialidad, Carga Horaria Total.
 - **Materia:** Código, Nombre, Área de Conocimiento.
 - **Curso (NRS):** ID_Único, Materia_ID, Docente_ID, Paralelo, Horario.
 - **Aula/Laboratorio:** ID, Capacidad, Tipo.
-

3. Diseño de Interfaz (Mockups)

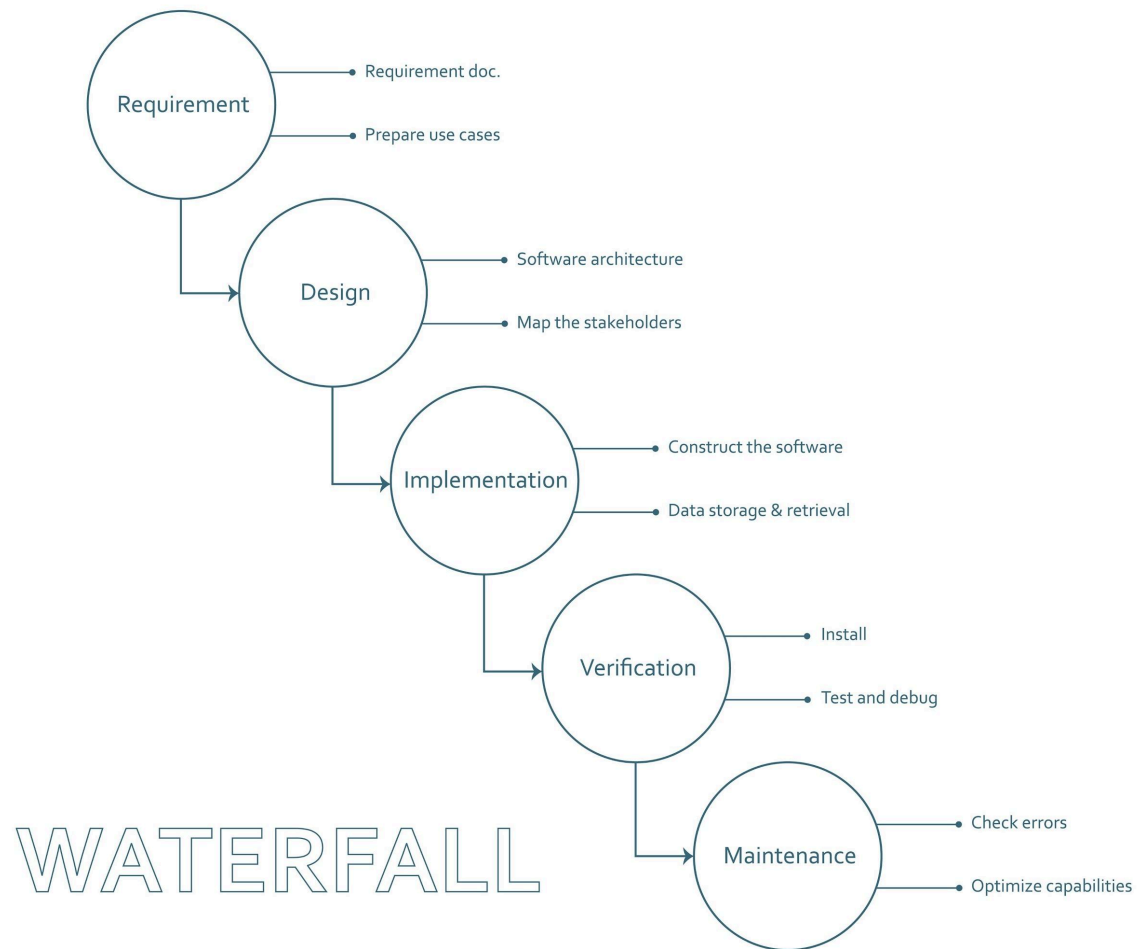
Se deben definir los componentes clave de la UI:

- **Dashboard del Coordinador:** Vista general de paralelos creados y docentes asignados.
 - **Módulo de Asignación:** Interfaz de "arrastrar y soltar" o selección filtrada de docentes según especialidad.
 - **Validador de Reglas:** Alertas visuales cuando un docente supera las horas permitidas o hay cruce de horarios en una misma aula.
-

4. Diagrama de Componentes

Describe cómo interactúan los módulos internos del sistema:

- **Módulo de Autenticación:** Gestión de roles (Director, Coordinador, Docente).
- **Motor de Reglas:** Lógica encargada de verificar la disponibilidad de docentes y aulas.
- **Generador de Reportes:** Exportación de la planificación académica en PDF/Excel.



5. Definición de API (Endpoints)

Para la comunicación entre el frontend y el backend:

- **GET /api/docentes**: Lista docentes filtrados por especialidad.
- **POST /api/nrs**: Crea un nuevo identificador de curso.
- **PUT /api/asignacion/{id}**: Vincula un docente a un NRS específico.

6. Estrategia de Seguridad

- **Autenticación**: Uso de JWT (JSON Web Tokens).
- **Autorización**: Control de acceso basado en roles (RBAC). El Coordinador puede editar, el Docente solo puede visualizar su horario.
- **Protección de datos**: Cifrado de contraseñas y backups automáticos diarios (como se menciona en el plan de GCS).

7. Trazabilidad de Requisitos

ID Requisito	Componente de Diseño	Descripción
RF-01 (Carga Horaria)	Motor de Reglas / Backend	Validación de límite de 22 horas.
RF-02 (Gestión NRS)	Módulo de Estructuración	Generación de identificadores únicos.
RF-03 (Especialidad)	Base de Datos / Filtros	Vinculación docente-materia por perfil.