**ÉPICA 1: MODELADO DE BASES DE DATOS**

**SQL Server:**

1. Analizar requerimientos para el modelo relacional
2. Diseñar diagrama entidad-relación (notación Chen)
3. Definir entidades y relaciones para cursos y semestres
4. Definir entidades y relaciones para estructura de carpetas y documentos
5. Definir entidades y relaciones para sistema de evaluación
6. Definir entidades y relaciones para noticias
7. Crear script para tablas de cursos y semestres
8. Crear script para tablas de documentos y carpetas
9. Crear script para tablas de evaluaciones y rubros
10. Crear script para tablas de noticias
11. Implementar validación de integridad referencial
12. Crear script de población con datos de prueba para cursos
13. Crear script de población con datos de prueba para evaluaciones

**MongoDB:**

1. Analizar requerimientos para modelo no relacional para MongoDB
2. Diseñar estructura de documentos para información personal en MongoDB
3. Definir esquema para estudiantes para MongoDB
4. Definir esquema para profesores para MongoDB
5. Definir esquema para administradores para MongoDB
6. Implementar sistema de encriptación MD5 para contraseñas
7. Crear script de población con datos de prueba para estudiantes
8. Crear script de población con datos de prueba para profesores

**ÉPICA 2: DESARROLLO DE APIS**

**API SQL Server:**

1. Configurar proyecto WebAPI en C#
2. Implementar Entity Framework para SQL Server
3. Crear modelos de datos para cursos y semestres
4. Crear modelos de datos para estructura documental
5. Crear modelos de datos para sistema de evaluación
6. Crear modelos de datos para noticias
7. Implementar controlador para gestión de cursos
8. Implementar controlador para gestión de semestres
9. Implementar controlador para gestión documental
10. Implementar controlador para gestión de evaluaciones
11. Implementar controlador para gestión de noticias
12. Crear pruebas unitarias para la API SQL Server
13. Configurar despliegue en IIS

**API MongoDB:**

1. Configurar proyecto WebAPI en C#
2. Implementar driver de MongoDB para C#
3. Crear modelos de datos para estudiantes en MongoDB
4. Crear modelos de datos para profesores en MongoDB
5. Crear modelos de datos para administradores en MongoDB
6. Implementar controlador para gestión de estudiantes con MongoDB
7. Implementar controlador para gestión de profesores
8. Implementar controlador para autenticación
9. Crear pruebas unitarias para la API MongoDB
10. Configurar despliegue en IIS

**Integración de APIs:**

1. Diseñar arquitectura de comunicación entre APIs
2. Implementar servicio de consulta cruzada para información personal
3. Implementar manejo de errores y excepciones

**ÉPICA 3: DESARROLLO FRONTEND (BASE)**

**Configuración inicial:**

1. Crear proyecto base React
2. Configurar integración con Bootstrap
3. Implementar sistema de estilos CSS global

**Componentes comunes:**

1. Crear componente de login
2. Crear componentes para manejo de tablas
3. Crear componentes para formularios
4. Crear componentes para carga de archivos
5. Crear componentes para visualización de documentos
6. Crear componentes para manejo de fechas

**Servicios frontend:**

1. Implementar servicio de autenticación
2. Implementar servicio para conexión con API SQL Server
3. Implementar servicio para conexión con API MongoDB
4. Crear servicio para manejo de archivos
5. Crear servicio para generación de reportes

**ÉPICA 4: VISTA ADMINISTRADOR**

**Autenticación:**

1. Implementar pantalla de login de administrador
2. Desarrollar validación de credenciales
3. Implementar manejo de sesiones
4. Crear página principal del administrador
5. Implementar menú de navegación específico

**Gestión de cursos:**

1. Crear interfaz para listar cursos genéricos
2. Desarrollar formulario para crear cursos
3. Implementar funcionalidad para deshabilitar cursos
4. Crear componente de búsqueda de cursos
5. Implementar validaciones de datos de cursos

**Inicialización de semestre:**

1. Crear interfaz para crear nuevo semestre
2. Desarrollar formulario para selección de cursos por semestre
3. Implementar asignación de grupos y profesores
4. Crear interfaz para matricular estudiantes por carnet
5. Desarrollar lógica para creación automática de estructura de carpetas
6. Implementar inicialización automática de rubros de evaluación

**Carga masiva:**

1. Crear interfaz para carga de archivo Excel
2. Implementar servicio de procesamiento de Excel
3. Crear visualización de resultados de importación

**ÉPICA 5: VISTA PROFESOR**

**Autenticación:**

1. Implementar pantalla de login de profesor
2. Desarrollar validación de credenciales
3. Implementar manejo de sesiones
4. Crear página principal del profesor
5. Implementar selector de cursos asignados

**Gestión documental:**

1. Crear interfaz de explorador de documentos
2. Implementar visualización de estructura de carpetas
3. Desarrollar funcionalidad para crear carpetas personalizadas
4. Implementar subida de documentos
5. Crear visualizador de documentos
6. Implementar descarga de documentos
7. Crear sistema de eliminación de documentos
8. Implementar restricciones para carpetas predefinidas

**Gestión de rubros:**

1. Crear interfaz para visualizar rubros del curso
2. Desarrollar formulario para crear nuevos rubros
3. Implementar edición de rubros existentes
4. Crear lógica para validación de porcentajes (total 100%)
5. Implementar eliminación de rubros
6. Desarrollar visualización de distribución de rubros

**Evaluaciones:**

1. Crear interfaz para listar evaluaciones
2. Desarrollar formulario para crear evaluaciones
3. Implementar asignación de rubros y porcentajes
4. Crear selector de tipo de evaluación (individual/grupal)
5. Implementar gestión de fechas límite
6. Desarrollar subida de especificaciones
7. Crear sistema para conformación de grupos

**Calificaciones:**

1. Crear interfaz para listar entregables
2. Implementar descarga de entregables
3. Desarrollar sistema de asignación de notas
4. Crear entrada para observaciones
5. Implementar subida de archivos de retroalimentación
6. Desarrollar sistema para guardar/publicar notas
7. Crear indicador visual de estado de publicación

**Noticias:**

1. Crear interfaz para listar noticias del curso
2. Desarrollar formulario para crear noticias
3. Implementar edición de noticias existentes
4. Crear sistema de eliminación de noticias
5. Implementar ordenamiento por fecha

**ÉPICA 6: VISTA ESTUDIANTE**

**Autenticación:**

1. Implementar pantalla de login de estudiante
2. Desarrollar validación de credenciales
3. Implementar manejo de sesiones
4. Crear página principal del estudiante
5. Implementar selector de cursos matriculados

**Documentos:**

1. Crear interfaz de explorador de documentos
2. Implementar navegación por estructura de carpetas
3. Desarrollar visualizador de documentos
4. Crear sistema de descarga de documentos
5. Implementar indicadores de documentos nuevos

**Evaluaciones:**

1. Crear interfaz para listar evaluaciones pendientes
2. Implementar visualización de especificaciones
3. Desarrollar sistema de subida de entregables
4. Crear validación de fechas límite
5. Implementar visualización de entregas anteriores
6. Desarrollar sistema para ver conformación de grupos
7. Crear visualización de notas publicadas

**Noticias:**

1. Crear interfaz para listar noticias del curso
2. Implementar ordenamiento por fecha (más reciente primero)
3. Desarrollar visualización detallada de noticias

**ÉPICA 7: REPORTES**

**Reporte de notas (profesor):**

1. Diseñar interfaz de reporte de notas
2. Implementar cálculo de notas por rubro
3. Desarrollar cálculo de nota final
4. Crear visualización tabular por estudiante
5. Implementar filtros y búsquedas
6. Desarrollar generación de PDF
7. Crear funcionalidad de descarga de reporte

**Reporte de estudiantes matriculados:**

1. Diseñar interfaz de reporte de estudiantes
2. Implementar visualización de datos personales
3. Crear sistema de filtrado y búsqueda
4. Desarrollar generación de PDF
5. Desarrollar generación de PDF

**Reporte de notas (estudiante):**

1. Diseñar interfaz personal de notas para estudiantes
2. Implementar visualización de notas por evaluación
3. Desarrollar cálculo de notas por rubro
4. Crear visualización de nota final para estudiantes
5. Implementar generación de PDF personal
6. Crear funcionalidad de descarga para los estudiantes

**Distribución de responsabilidades por rol**

**DBA SQL Server**

* Modelado y creación de la base de datos SQL Server
* Scripts de inicialización y población para SQL Server
* Soporte al equipo de backend para integración SQL

**DBA MongoDB**

* Diseño y creación de la base de datos MongoDB
* Scripts para MongoDB
* Soporte al equipo de backend para integración MongoDB

**Backend 1 (SQL Server)**

* API para gestión con SQL Server
* Integración entre APIs
* Lógica de negocio principal
* Generación de reportes PDF

**Backend 2 (MongoDB)**

* API para gestión con MongoDB
* Encriptación de contraseñas
* Integración de autenticación
* Funcionalidades relacionadas con datos personales

**Frontend**

* Implementación de todas las vistas (Admin, Profesor, Estudiante)
* Diseño con Bootstrap
* Implementación de subida/descarga de archivos
* Visualización de reportes