**SISTEMA DE GESTIÓN ACADÉMICA DOCUMENTACIÓN COMPLETA**

**Versión 1.0**

**Fecha: [29/12/2024]**

**CONTROL DE VERSIONES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Responsable** | **Descripción** |
| 1.0 | 29/12/2024 | Equipo renacido | Versión inicial del documento |

**TABLA DE CONTENIDOS**

1. VISIÓN DEL PROYECTO
2. ALCANCE Y OBJETIVOS
3. ARQUITECTURA DEL SISTEMA
4. HISTORIAS DE USUARIO
5. ANEXOS
6. **VISIÓN DEL PROYECTO**
   1. **DESCRIPCIÓN GENERAL**

El Sistema de Gestión Académica (SGA) es una herramienta que permite automatizar y optimizar la administración de estudiantes, docentes y aulas en la CEPRE UNU. La solución facilita procesos como el registro, seguimiento académico, asignación de aulas y generación de reportes.

**OBJETIVOS PRINCIPALES**

* Automatizar procesos de matrícula y gestión académica
* Centralizar y digitalizar la información de estudiantes, docentes y aulas.
* Mejorar la experiencia del usuario mediante una interfaz intuitiva y funcional.
  1. **STAKEHOLDERS**
* Estudiantes
* Docentes
* Personal administrativo
* Directivos académicos

1. **ALCANCE Y OBJETIVOS**
   1. **MÓDULOS DEL SISTEMA**
      1. **Gestión de Estudiantes**

* Registro y actualización de datos personales.
* Visualización del aula asignada.
  + 1. **Gestión de Docentes**
* Registro de datos personales y profesionales.
* Gestión de áreas de asignación.
  + 1. **Gestión de Aulas**
* Visualización de aulas y asignación de estudiantes.
* Configuración del sistema
  + 1. **Gestión de Horarios**
* Visualización de horarios disponibles
* Configuración del sistema
  + 1. **Gestión de Reportes**
* Visualización de reportes con respecto a estudiantes, docentes y aulas
* Configuración del sistema

1. **ARQUITECTURA DEL SISTEMA**
   1. **STACK TECNOLÓGICO**

**Aplicación Monolítica**

**Framework base:**

* Spring Boot: Para la gestión de la lógica de negocio y renderización de vistas (si aplica).

**Frontend:**

* Thymeleaf (o HTML puro, si no usas plantillas dinámicas): Para el renderizado de vistas directamente desde Spring Boot.
* CSS y JS: Para estilos y lógica en el cliente, utilizando frameworks como Bootstrap, CSS según sea necesario.

**Backend:**

* **Spring Framework**: Para controladores, servicios y repositorios.
* **JPA/Hibernate**: Para la interacción con la base de datos.

**Base de Datos**:

* MySQL
  1. **REQUISITOS NO FUNCIONALES**

**Disponibilidad:**

* Uptime: 99.9%
* Tiempo máximo de inactividad: 8.76 horas/año

**Rendimiento:**

* Tiempo de respuesta < 2 segundos
* Throughput: 1000 requests/segundo
* Latencia máxima: 200ms

**Seguridad:**

* Encriptación de datos sensibles
* Auditoría de acciones

**Escalabilidad:**

* Horizontal para componentes stateless
* Vertical para base de datos
* Auto-scaling basado en carga

1. **HISTORIAS DE USUARIO**
   1. **EPIC 1: Gestión de Usuarios**
      1. **HU-001: Registro de Estudiantes**

**Como** nuevo estudiante **Quiero** poder registrarme en el sistema **Para** acceder a los servicios académicos

**Criterios de Aceptación:**

1. El formulario debe incluir:
   * Datos personales (nombre, apellidos, DNI/pasaporte)
   * Fecha de nacimiento
   * Dirección
   * Contacto (email, teléfono)
   * Datos del contacto de emergencia
2. Validar formato de email y teléfono
3. Verificar que el DNI/pasaporte no esté duplicado

**Estimación: 5 puntos  
Prioridad: Alta**

* + 1. **HU-002: Registro de Docentes**

**Como** nuevo docente  
**Quiero** poder registrar mi perfil en el sistema  
**Para** gestionar mis cursos y estudiantes

Criterios de Aceptación:

1. El formulario debe incluir:
   * Datos personales completos
   * Información profesional
   * Áreas de especialización

**Estimación: 5 puntos  
Prioridad: Alta**

* + 1. **HU-003: Gestión de Perfiles**

**Como** usuario registrado  
**Quiero** poder actualizar mi información personal  
**Para** mantener mis datos actualizados

Criterios de Aceptación:

1. Permitir editar:
   * Datos de contacto
   * Dirección
   * Información profesional
2. Mantener historial de cambios

Estimación: 3 puntos  
Prioridad: Media

* 1. **EPIC 2: Gestión de Matrículas**
     1. **HU-004: Proceso de Matrícula Online**

**Como** estudiante  
**Quiero** poder matricularme en el centro pre universitario  
**Para** Prepararme para mi ingreso a la universidad

Criterios de Aceptación:

1. Mostrar:

* Datos personales requeridos: Nombre completo, DNI, fecha de nacimiento, dirección, teléfono y correo electrónico.
* Opciones de carreras disponibles: modalidad y horarios.

1. Validar conflictos de horario

* Datos personales: Que estén completos y correctos.
* Duplicados: Evitar que un estudiante ya registrado pueda registrarse nuevamente.

1. Permitir:

* Selección de carrera, modalidad y horarios.

Estimación: 8 puntos  
Prioridad: Alta

* 1. **EPIC 3: Gestión Académica**
     1. **HU-005: Asignación de Aulas**

**Como** Administrador   
**Quiero** Asignar aulas a estudiantes y docentes

**Para** Organizar las clases de forma eficiente y evitar conflictos de espacio

Criterios de Aceptación:

1. Mostrar:

* Listado de aulas disponibles con capacidad máxima.
* Horarios de uso de cada aula.

1. Permitir:

* Relacionar estudiantes y docentes con las aulas asignadas.

1. Generar:

* Reporte de aulas asignadas con horarios, docentes y estudiantes.

Estimación: 8 puntos  
Prioridad: Alta

* + 1. **HU-006: Control de Asistencia**

**Como** docente  
**Quiero** registrar la asistencia de los estudiantes  
**Para** hacer seguimiento de la participación

Criterios de Aceptación:

1. Permitir:
   * Registro por fecha
   * Múltiples estados (presente, tardanza, ausente)
   * Justificaciones
   * Reportes de asistencia
2. Cálculo automático de porcentajes
3. Alertas por inasistencias excesivas
4. Exportación de reportes

Estimación: 5 puntos  
Prioridad: Media

* + 1. **HU-007: Gestión de Horarios**

**Como** administrador   
**Quiero** poder gestionar los horarios de clases  
**Para** organizar el proceso académico

Criterios de Aceptación:

1. Permitir:
   * Creación de horarios
   * Asignación de aulas
   * Gestión de conflictos
2. Notificar cambios a afectados
3. Generar vista calendario

Estimación: 8 puntos  
Prioridad: Alta

* 1. **EPIC 4: Reportes y Análisis**
     1. **HU-008: Reportes Académicos**

**Como** director académico  
**Quiero** generar reportes de rendimiento de los estudiantes  
**Para** analizar el desempeño académico y tomar decisiones estratégicas

Criterios de Aceptación:

1. Generar reportes de:

* Estadísticas de aprobación: Porcentajes de estudiantes aprobados y desaprobados en cada evaluación.
* Tendencias de notas: Evolución de las notas a lo largo del proceso académico.
* Asistencia global: Porcentaje de asistencia por estudiante

1. Filtros personalizables
2. Exportación a múltiples formatos
3. Gráficos interactivos

Estimación: 8 puntos  
Prioridad: Media

**Notas Técnicas**

**Definition of Ready (DoR):**

* Historia claramente descrita
* Criterios de aceptación definidos
* Estimación realizada
* Dependencias identificadas
* Recursos necesarios disponibles

**Definition of Done (DoD):**

* Código implementado y probado
* Pruebas automatizadas escritas
* Documentación actualizada
* Code review completado
* Desplegado en ambiente de pruebas
* Aprobado por Product Owner

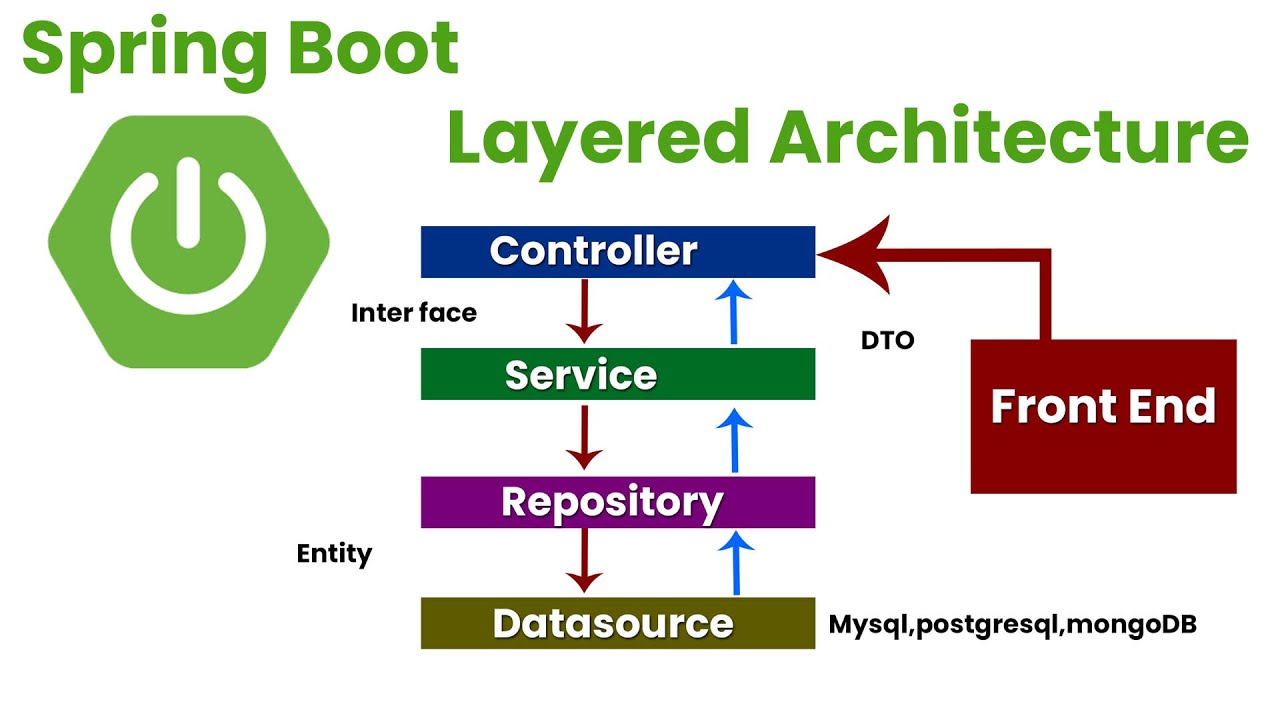
**Priorización:**

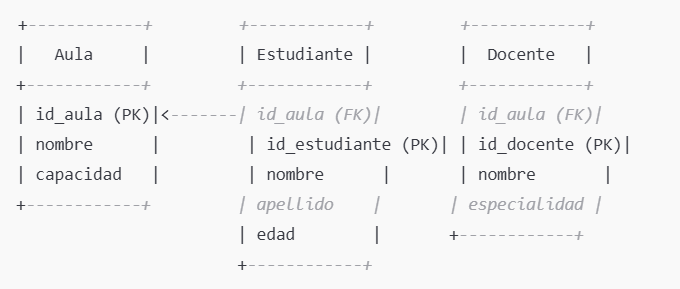
* Alta: Crítico para el MVP
* Media: Importante pero no bloqueante
* Baja: Deseable, pero puede esperar

**Estimación:**

* 1-2 puntos: Tarea simple
* 3-5 puntos: Complejidad media
* 8 puntos: Tarea compleja
* 13+ puntos: Requiere división

1. **ANEXOS**
   1. **DIAGRAMAS**

* Arquitectura del sistema:
  + En capas:
* Modelo de datos:



* 1. **MOCKUPS**
* Interfaces principales

**GITHUB**

* FRONT
* BACK
* DOCUMENTACION
* INFORME
* BD (SCRIPT)
* MANUAL DE INSTALACIÓN Y CONFIGURACIÓN

**APROBACIONES**

Product Owner Scrum Master

Fecha: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Fecha: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tech Lead Stakeholder Principal

Fecha: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Fecha: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_