

---

## **Especificación de requisitos de software**

**Proyecto:**  
Gestión curricular

# Índice

<b>1. Introducción</b>	<b>3</b>
1.1. Propósito	3
1.2. Ámbito del Sistema	3
1.3. Definiciones, Acrónimos y Abreviaturas	4
1.4. Referencias	4
1.5. Visión General del Documento	5
<b>2. Descripción General</b>	<b>6</b>
2.1. Perspectiva del Producto	6
2.2. Funciones del Producto	6
2.3. Características de los Usuarios	7
2.4. Restricciones	8
2.5. Suposiciones y Dependencias	8
2.6. Requisitos Futuros	9
<b>3. Requisitos Específicos</b>	<b>9</b>
3.1. Requisitos funcionales	10
3.2. Funciones	22
3.3. Requisitos de Rendimiento	23
3.4. Restricciones de Diseño	23
3.5. Atributos del Sistema	24
3.6. Otros Requisitos	24
<b>4. Apéndices</b>	<b>24</b>

# **1. Introducción**

El presente documento especifica los requisitos del sistema de Gestión Curricular desarrollado para la Universidad de los Llanos. Este sistema tiene como objetivo centralizar y automatizar la administración de los planes de estudio, microcurrículos y guías de laboratorio de los programas académicos.

Diseñado como una aplicación de escritorio desarrollada en Java con Spring Boot, el sistema permite a distintos tipos de usuarios —como decanos, directores de programa, docentes, comités de programa, invitados y directores de escuela— gestionar la estructura curricular, validar propuestas académicas y visualizar de manera gráfica la evolución del pénsum.

Entre sus funcionalidades clave se encuentran: la carga y validación de archivos Excel con información curricular, la generación automática del plan de estudios, la visualización interactiva del currículo, y un robusto sistema de control de acceso y firmas digitales para garantizar la trazabilidad y validez académica.

Este documento describe tanto los requisitos funcionales como no funcionales, además de las restricciones, interfaces, atributos de calidad y elementos visuales.

## **1.1. Propósito**

Este documento tiene como propósito describir detalladamente los requisitos funcionales y no funcionales del sistema de Gestión Curricular que será desarrollado como una aplicación de escritorio en Java utilizando Spring Boot. Está dirigido a los desarrolladores, diseñadores, evaluadores, usuarios finales (decano, directores de programa, directores de escuela, docentes, comité de programa, invitados) y a los responsables de validar y aprobar el sistema.

## **1.2. Ámbito del Sistema**

El sistema de Gestión Curricular tiene como objetivo principal permitir la digitalización, visualización y administración de los planes de estudio, microcurrículos y guías de laboratorio de los programas académicos de la Universidad de los Llanos. Sus funciones clave incluyen:

- Carga de archivos Excel con información del pénsum académico.
- Generación automática del plan de estudios a partir del Excel cargado.
- Visualización gráfica de los planes de estudio y prerrequisitos.
- Gestión de usuarios con distintos roles y permisos.
- Carga, gestión y descarga de microcurrículos y guías de laboratorio.

- Validación y firma de propuestas curriculares por los distintos actores del proceso.

El sistema no incluirá funcionalidades de matrícula, notas, ni procesos administrativos externos al ámbito curricular.

### 1.3. Definiciones, Acrónimos y Abreviaturas

Término / Acrónimo	Definición
Pénsun	Conjunto estructurado de asignaturas de un programa académico.
Microcurrículo	Documento que contiene los objetivos, contenidos, metodología , evaluación, etc de una asignatura.
RFN	Requerimiento No Funcional
RF	Requerimiento Funcional
HU	Historia de Usuario
CRUD	Crear, Leer, Actualizar y Eliminar (operaciones básicas sobre datos)
UI	Interfaz de Usuario

Tabla 1. Definiciones términos y acrónimos

### 1.4. Referencias

- IEEE - Institute of Electrical and Electronics Engineers. 1990. Estandar IEEE 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology. URL: <https://standards.ieee.org/ieee/610.12/855/>.
- Sommerville I. 2005. Ingeniería del Software. Pearson Addison-Wesley. 7ma edición. Madrid.

## **1.5. Visión General del Documento**

Este documento se organiza en las siguientes secciones:

- La Sección 2 describe el contexto del sistema, sus funciones generales, usuarios y restricciones.
- La Sección 3 especifica detalladamente los requisitos funcionales y no funcionales.
- La Sección 4 contiene apéndices y elementos complementarios relevantes para la implementación y validación del sistema.

## **2. Descripción General**

### **2.1. Perspectiva del Producto**

El sistema de Gestión Curricular es una aplicación de escritorio independiente que será utilizada por actores institucionales de la Universidad de los Llanos para administrar eficientemente la información curricular de los programas académicos.

Este sistema no depende de otras plataformas externas, aunque puede integrarse con bases de datos internas de la universidad. Es parte de un ecosistema institucional más amplio, pero funciona como un módulo autónomo centrado exclusivamente en la gestión curricular.

El producto interactúa con los siguientes elementos:

- Base de datos institucional: para almacenar usuarios, currículos, microcurrículos y propuestas académicas.
- Archivos Excel y PDF: como medios de carga, visualización y exportación de información.
- Sistema operativo local: ya que es una aplicación de escritorio que debe ejecutarse en equipos autorizados.

### **2.2. Funciones del Producto**

A nivel general, el sistema provee las siguientes funciones:

- Carga de pénsum académico: A través de archivos Excel estructurados.
- Generación automática del plan de estudios a partir de los datos cargados.
- Visualización gráfica del currículo.
- Gestión y validación de usuarios: con control de roles y permisos.
- Carga, gestión de microcurrículos y guías de laboratorio.
- Propuestas y firmas de cambios curriculares: incluyendo flujos de aprobación entre docentes, directores, comités y directores de escuela.
- Acceso público restringido para consulta general del currículo.

### 2.3. Características de los Usuarios

Tipo de Usuario	Características Principales
Decano	Usuario con privilegios máximos para asignar directores de programa, directores de escuela, comité de programa y docentes.
Director de Programa	Administra el currículo, visualiza las propuestas curriculares enviadas y firma propuestas finales, asignación de docentes a cursos, puede editar la información de los cursos.
Docente	Propone microcurrículos y revisa contenidos relacionados con su área.
Comité de Programa	Revisa, acepta o rechaza propuestas curriculares.
Director de Escuela	Revisa y firma propuestas finales.
Invitado	Usuario sin registro que puede visualizar el currículo general sin permisos de edición.

Tabla 2. Tipos de Usuario y características principales

## **2.4. Restricciones**

- El sistema debe desarrollarse exclusivamente en Java con el framework Spring Boot.
- Debe funcionar únicamente como aplicación de escritorio.
- Debe respetar la identidad visual de la Universidad (paleta: rojo, blanco y negro).
- Solo los usuarios autorizados pueden modificar los permisos del sistema.

## **2.5. Suposiciones y Dependencias**

- Se asume que los usuarios cuentan con credenciales válidas previamente registradas.
- El sistema operará sobre infraestructura local o institucional controlada (PCs, red interna).
- El sistema asumirá un formato estándar de archivo excel o csv para la carga de información curricular.
- Se espera que los usuarios conozcan la estructura curricular institucional y los criterios de validación académica.



## **2.6. Requisitos Futuros**

- Posible extensión a una versión web o móvil del sistema.
- Integración con otros sistemas de gestión institucional (académica, administrativa).
- Generación automática de reportes estadísticos sobre currículo, propuestas y aprobaciones.

## **3. Requisitos Específicos**

### 3.1. Requisitos funcionales

- El sistema debe permitir la carga de un archivo Excel con los planes de estudio.
- El sistema debe generar automáticamente el plan de estudios a partir de un archivo Excel cargado previamente.
- Gestión y validación de usuarios.
- El sistema debe poder presentar los planes de estudio de forma gráfica.
- El sistema debe permitir cargar microcurriculum y guías de laboratorio para cada curso.

1. El sistema debe permitir la carga de un archivo Excel con los planes de estudio.

Código: <b>HU-01-1</b>	Usuario: <b>Director de programa</b>
Título: <b>Carga del p�nsun acad�mico</b>	
Prioridad: <b>Alta</b>	Tipo: <b>Historia</b>
<b>Descripci�n:</b> Como <b>Director de programa</b> , quiero poder cargar los datos del p�nsun acad�mico desde un archivo Excel, para que la estructura curricular se almacene correctamente en el sistema y est� lista para su gesti�n y consulta.	
<b>Criterios de aceptaci�n</b> <ul style="list-style-type: none"><li>• <i>El archivo subido debe ser de tipo Excel (.csv, .xls, .xlsx).</i></li><li>• <i>El archivo debe ser escaneado para verificar que est� libre de virus.</i></li><li>• <i>Se debe mostrar un mensaje de confirmaci�n al completar la carga con �xito.</i></li><li>• <i>Los datos deben validarse para asegurar que est�n completos y correctos.</i></li><li>• <i>Debe existir una interfaz para cargar el archivo del p�nsun a trav�s de un bot�n.</i></li><li>• <i>Debe estar disponible un archivo de ejemplo con la estructura requerida para la carga del p�nsun (campos como: c�digo del curso, nombre, cr�ditos, intensidad horaria, tipo, prerrequisitos, etc.).</i></li></ul>	

2. El sistema debe generar autom ticamente el plan de estudios a partir de un archivo Excel cargado previamente. ident ficate en esta monda

Código: <b>HU-02-1</b>	Usuario: <b>Decano</b>
Título: <b>Automatización del plan de estudios</b>	
Prioridad: <b>Alta</b>	Tipo: <b>Historia</b>
<b>Descripción:</b> Como <b>Decano</b> , quiero que el sistema lea y procese automáticamente un archivo Excel cargado previamente, organizando los cursos según su código, créditos, intensidad horaria y prerequisites, para garantizar una generación eficiente del plan de estudios.	
<b>Criterios de aceptación</b> <i>Generación Automática del Plan de Estudios</i> <ul style="list-style-type: none"> <li>• El sistema debe poder leer y procesar correctamente el archivo Excel cargado.</li> <li>• Los datos extraídos deben almacenarse en la base de datos.</li> <li>• El plan de estudios generado debe respetar la linealidad y los prerequisites definidos.</li> <li>• Si el archivo contiene datos incompletos o incorrectos, el sistema debe mostrar un mensaje de error claro.</li> <li>• Si el archivo no cumple con el formato esperado, el sistema debe rechazarlo y notificar al usuario.</li> </ul>	

### 3. Gestión y validación de usuarios.

#### **Decano:**

Código: <b>HU-03-1</b>	Usuario: <b>Decano</b>
Título: <b>Gestión de Decano</b>	
Prioridad: <b>Alta</b>	Tipo: <b>Historia</b>
<b>Descripción:</b> Como <b>Decano</b> quiero acceder al sistema para visualizar los docentes y usuarios disponibles, con el fin de asignar a los Directores de Programa, comité de programa y directores de escuela responsables de administrar los currículos correspondientes.	

### Criterios de aceptación

#### Gestión de asignaciones:

- El sistema debe permitir visualizar los docentes que están registrados y disponibles para ser asignados como Directores de Programa.
- El sistema debe permitir asignar docentes como Directores de Programa
- El sistema debe impedir la eliminación del Decano.
- El sistema debe registrar la fecha y hora de cada asignación realizada.

#### Validación y seguridad:

- El Decano debe iniciar sesión en el sistema con usuario y contraseña válidos.
- Las credenciales deben validarse contra una base de datos segura.
- El sistema debe cerrar la sesión del Decano tras 10 minutos de inactividad.
- Solo el Decano puede realizar la asignación de Directores de Programa.
- El sistema debe garantizar que un programa tenga únicamente un Director asignado.

#### Interfaz:

- Debe existir un panel administrativo exclusivo para el Decano.
- El panel debe mostrar una lista filtrable de docentes, director de programa y director de escuela disponibles (por nombre).
- Debe mostrar mensajes de confirmación y éxito al realizar cada acción.

### *Director de programa:*

Código: <b>HU-03-2</b>	Usuario: <b>Director de programa</b>
Título: <b>Gestión de Director de Programa</b>	
Prioridad: <b>Alta</b>	Tipo: <b>Historia</b>
<b>Descripción:</b> Como <b>Director de Programa</b> , quiero acceder al sistema para gestionar y administrar el currículo del programa académico, incluyendo elementos como perfiles, competencias, resultados de aprendizaje, plan de estudios y cursos académicos, con el fin de mantener actualizada y disponible la información curricular. Además, necesito revisar, aprobar y firmar propuestas curriculares enviadas por los docentes, para garantizar su validez antes de ser evaluadas por el Comité de Programa	

### **Criterios de aceptación**

#### **Gestión curricular:**

- Puede subir y actualizar el plan de estudios, incluyendo los cursos académicos.
- Debe poder configurar los tipos de curso (teórico, práctico, teórico-práctico) y sus atributos (créditos, intensidad horaria, apoyos didácticos y microcurrículos).
- Puede asignar docentes a cursos.

#### **Visualización y publicación:**

- Puede visualizar las propuestas generadas por los docentes y enviarlas al Comité de programa
- Cuando el Director de Programa acepta una propuesta, el sistema la reenvía automáticamente al Comité de Programa para su confirmación.
- Puede dejar comentarios, evaluaciones y notificaciones sobre propuestas , sin modificarlas directamente

#### **Validación y restricciones:**

- No puede reasignarse a otro programa ni editar asignaciones hechas por el Decano.
- El Director de programa debe iniciar sesión en el sistema con usuario y contraseña válidos.
- Las credenciales deben validarse contra una base de datos segura.
- El sistema debe cerrar la sesión del Director de programa tras 10 minutos de inactividad.

#### **Validación y firma:**

- Puede visualizar las propuestas curriculares enviadas para su revisión.
- Puede verificar que cuenten con la firma del Director de Escuela antes de firmarlas.
- Puede firmar la propuesta para su aprobación final.
- Puede rechazar la firma y devolver la propuesta con observaciones.

#### **Interfaz:**

- Debe contar con un panel de gestión curricular estructurado por secciones (perfiles, competencias, resultados, plan, cursos).
- El sistema debe mostrar alertas si hay elementos incompletos o inconsistentes dentro del currículo.
- Debe incluir herramientas gráficas o visuales para la visualización del plan de estudios (como diagramas o líneas de tiempo).

**Docente:**

Código: <b>HU-03-3</b>	Usuario: <b>Docente</b>
Título: <b>Gestión de Docentes</b>	
Prioridad: <b>Alta</b>	Tipo: <b>Historia</b>
<b>Descripción:</b> Como <b>Docente</b> , quiero ingresar al sistema para proponer microcurrículos al Comité de programa, revisar los elementos curriculares relacionados con mi área, y descargar la información del currículo completo, con el fin de contribuir al mejoramiento académico del programa.	
<b>Criterios de aceptación</b>  <b>Consulta y descarga:</b> <ul style="list-style-type: none"> <li>● Pueden consultar los perfiles, competencias y resultados de aprendizaje asociados a su curso.</li> <li>● Pueden descargar el currículo completo.</li> <li>● No pueden modificar directamente el currículo general ni los cursos asignados.</li> </ul> <b>Restricciones de acceso:</b> <ul style="list-style-type: none"> <li>● No tienen permisos para asignar roles ni modificar datos de otros usuarios.</li> <li>● No pueden eliminar propuestas ni registros curriculares anteriores.</li> </ul> <b>Propuesta:</b> <ul style="list-style-type: none"> <li>● El docente puede crear propuestas curriculares cargando los documentos de soporte necesarios.</li> </ul> <b>Interfaz:</b> <ul style="list-style-type: none"> <li>● Debe existir una sección clara de “<i>Manejar mis Cursos</i>” donde el docente pueda crear y subir.</li> <li>● El sistema debe mostrar el estado de la propuesta y posibles observaciones del Director de Programa.</li> </ul>	

### Comité de programa:

Código: <b>HU-03-4</b>	Usuario: <b>Comité de programa</b>
Título: <b>Gestión de Comité de programa</b>	
Prioridad: <b>Alta</b>	Tipo: <b>Historia</b>

<p><b>Descripción:</b></p> <p>Como <b>Comité de programa</b>, quiero acceder al sistema para revisar y gestionar las propuestas de microcurrículos enviadas por los Directores de programa, tomar decisiones sobre su aprobación, con el fin de garantizar la calidad académica y la coherencia del currículo del programa.</p>
<p><b>Criterios de aceptación</b></p> <p><b>Revisión de propuestas:</b></p> <ul style="list-style-type: none"> <li>• El sistema debe permitir al Comité de programa visualizar todas las propuestas enviadas por los Directores de programa.</li> <li>• Cada propuesta debe poder ser aprobada o rechazada.</li> <li>• El historial de decisiones debe quedar registrado con la fecha, estado actual y usuario que realizó la acción.</li> </ul> <p><b>Restricciones de acceso:</b></p> <ul style="list-style-type: none"> <li>• Solo pueden acceder a las propuestas enviadas.</li> <li>• No pueden modificar el contenido original de los microcurrículos.</li> <li>• No pueden editar perfiles, competencias u otros componentes curriculares directamente.</li> </ul> <p><b>Interfaz:</b></p> <ul style="list-style-type: none"> <li>• El sistema debe permitir ver los detalles de cada propuesta y registrar observaciones fácilmente.</li> </ul>

## Invitado:

Código: <b>HU-03-5</b>	Usuario: <b>Invitado</b>
Título: <b>Visualización del currículo como usuario invitado</b>	
Prioridad: <b>Alta</b>	Tipo: <b>Historia</b>
<p><b>Descripción:</b></p> <p>Como <b>Invitado</b>, quiero acceder al sistema con permisos limitados para visualizar el currículo de forma general, con el fin de conocer la oferta académica y estructura curricular sin requerir autenticación completa ni afectar la integridad del sistema.</p>	

### Criterios de aceptación

#### Acceso y seguridad:

- El sistema debe permitir el acceso como invitado sin necesidad de registro o autenticación avanzada.
- No se debe permitir la edición ni descarga del contenido curricular.
- La navegación debe estar restringida a consulta y visualización de información pública.

#### Consulta y visualización:

- Puede acceder al currículo general aprobado, incluyendo perfiles, competencias, resultados de aprendizaje y plan de estudios.
- Puede visualizar cursos académicos, tipo de curso (teórico o teórico-práctico), intensidad horaria, créditos y microcurrículos aceptados.

#### Interfaz:

- Debe existir una vista pública amigable y resumida del currículo académico.
- El diseño debe ir acorde con los colores de la universidad.
- El diseño debe ser intuitivo y enfocado en la lectura.

### Director de escuela:

Código: <b>HU-03-6</b>	Usuario: <b>Director de escuela</b>
Título: <b>Gestión Director de escuela</b>	
Prioridad: <b>Alta</b>	Tipo: <b>Historia</b>
<b>Descripción:</b> Como <b>Director de Escuela</b> , necesito revisar y firmar las propuestas curriculares que hayan sido aprobadas previamente por el Comité de Programa, con el fin de garantizar que cumplen con los lineamientos académicos y administrativos antes de ser incorporadas al sistema curricular vigente.	



### **Criterios de aceptación**

#### **Validación y firma:**

- Puede visualizar las propuestas curriculares enviadas para su revisión.
- Puede verificar que cuenten con la firma del Director de Programa antes de firmarlas.
- Puede firmar la propuesta para su aprobación final.
- Puede rechazar la firma y devolver la propuesta con observaciones.

#### **Restricciones de acceso:**

- No puede crear ni editar propuestas curriculares.
- Solo se puede acceder a propuestas que hayan sido aprobadas por el Comité de Programa.
- No puede modificar observaciones previas realizadas por otros actores del proceso.

#### **Propuesta:**

- Una vez la propuesta ha sido aprobada por el Comité de Programa, el sistema habilita la firma del Director de Escuela.
- Si el Director de Escuela no firma, la propuesta se considera rechazada y se notifica al docente.

#### **Interfaz:**

- El sistema debe mostrar una lista de propuestas pendientes de firma, con estado detallado (quién aprobó, quién firmó, observaciones).
- Debe haber una opción clara para visualizar la propuesta, firmar electrónicamente o devolver con comentarios.
- El sistema debe registrar la fecha y hora de la firma.

### Propuestas de microcurrículos:

Código: <b>HU-03-7</b>	Usuario: <b>Docente</b>
Título: <b>Peticiones para cambios de contenidos</b>	
Prioridad: <b>Alta</b>	Tipo: <b>Historia</b>
<b>Descripción:</b> Como Docente, quiero tener la oportunidad de compartir mi opinión sobre los contenidos dictados en las asignaturas presentes correspondientes a mi perfil, con la intención de poder proponer mejoras y modificaciones teniendo en cuenta el enfoque del programa.	
<b>Criterios de aceptación</b>  <b>Propuestas académicas:</b> <ul style="list-style-type: none"><li>● El sistema debe permitir a los Docentes cargar propuestas de microcurrículos, asociados a cursos existentes.</li><li>● Cada propuesta debe quedar registrada con la fecha y el estado (pendiente, en revisión, aceptada, rechazada).</li></ul> <b>Consulta y visualización:</b> <ul style="list-style-type: none"><li>● Debe haber una sección correspondiente a las peticiones donde los docentes puedan observar el progreso de la petición realizada.</li><li>● Debe mostrar o tener la opción de descargar el ejemplo de un formato, con el fin de realizar la solicitud de forma correcta.</li><li>● Las peticiones deben ser respondidas por el director de programa y el comité de programa para su aprobación, la cual debe mostrarse como aprobada en su sección correspondiente.</li></ul> <b>Interfaz:</b> <ul style="list-style-type: none"><li>● Debe haber una sección de carga de peticiones para enviar la petición correspondiente.</li></ul>	

4. El sistema debe poder presentar los planes de estudio de forma gráfica.

Código: <b>HU-04</b>	Usuario: <b>Usuario sin rol específico</b>
Título: <b>Planes de estudio de forma gráfica.</b>	
Prioridad: <b>Alta</b>	Tipo: <b>Historia</b>
<b>Descripción:</b> Como <b>usuario</b> , quiero que el sistema me muestre el plan de estudios de forma gráfica, representando las asignaturas organizadas por semestres y sus prerrequisitos, para comprender mejor la estructura curricular y la progresión de los cursos. Además, deseo acceder al histórico de los anteriores planes de estudio para comparar cambios curriculares.	
<b>Criterios de aceptación</b> <i>Visualización del Plan de Estudios</i> <ul style="list-style-type: none"><li>● El sistema debe generar una representación gráfica del plan de estudios, mostrando las asignaturas organizadas por semestre.</li><li>● Se debe representar la relación entre asignaturas y sus prerrequisitos mediante conexiones visuales.</li></ul> <i>Histórico de Planes de Estudio</i> <ul style="list-style-type: none"><li>● El sistema debe ofrecer una opción para consultar versiones anteriores del plan de estudios.</li><li>● Al seleccionar una versión anterior, esta debe visualizarse gráficamente con el mismo formato del plan actual.</li></ul> <i>Validaciones y Errores</i> <ul style="list-style-type: none"><li>● La vista gráfica debe actualizarse automáticamente cuando se realicen modificaciones en el plan de estudios.</li></ul> <i>Interfaz:</i> <ul style="list-style-type: none"><li>● Al hacer clic sobre una asignatura, redirigirá a una pestaña en donde se mostrará información detallada de la asignatura como nombre, código, créditos, tipo de curso e intensidad horaria, guías de laboratorio.</li></ul>	

5. El sistema debe permitir cargar microcurrículum y guías de laboratorio para cada asignatura.

Código: <b>HU-05</b>	Usuario: <b>Director de programa</b>
Título: <b>Cargar microcurrículos</b>	
Prioridad: <b>Alta</b>	Tipo: <b>Historia</b>
<b>Descripción:</b> Como <b>Director de Programa</b> , quiero poder cargar y actualizar los microcurrículos y guías de laboratorio de cada asignatura dentro del sistema, para asegurar que los estudiantes y profesores tengan acceso a la documentación académica actualizada. Además, deseo que el sistema conserve un histórico de versiones anteriores, para facilitar el seguimiento de cambios y la consulta de versiones previas cuando sea necesario.	
<b>Criterios de aceptación</b> <i>Carga de Documentos</i> <ul style="list-style-type: none"><li>● El sistema debe permitir subir archivos en formatos PDF, DOCX o similares.</li><li>● Se debe poder asociar cada documento a una asignatura específica.</li><li>● Una vez cargado, el sistema debe mostrar un mensaje de confirmación y permitir la consulta inmediata.</li></ul> <i>Gestión de Documentos</i> <ul style="list-style-type: none"><li>● Se debe poder cargar y actualizar.</li><li>● Los directores de programa pueden gestionar los documentos de todas las asignaturas.</li><li>● Se debe permitir la descarga de los documentos por parte de los usuarios autorizados.</li></ul> <i>Histórico de Documentos</i> <ul style="list-style-type: none"><li>● El sistema debe mantener un histórico de versiones anteriores de microcurrículos asociados a los planes de estudio anteriores.</li></ul> <i>Validaciones y Errores</i> <ul style="list-style-type: none"><li>● Si un archivo supera el tamaño permitido (30MB), el sistema debe mostrar un mensaje de error.</li><li>● Si el formato del archivo no es válido, se debe rechazar la carga con una notificación.</li></ul>	

IA:

Código: <b>HU-06</b>	Usuario: <b>Cualquier usuario (independiente del rol)</b>
Título: <b>Interacción con agente de IA</b>	
Prioridad: <b>Alta</b>	Tipo: <b>Historia</b>
<b>Descripción:</b> Como <b>usuario con cualquier rol</b> dentro del sistema, quiero poder interactuar con un agente de inteligencia artificial para resolver dudas frecuentes sobre el uso de la app, obtener información de cursos y programas, comparar planes de estudio y conocer las funciones que tiene cada rol, de forma que pueda orientarse rápidamente y aprovechar mejor las funcionalidades de la plataforma.	
<b>Criterios de aceptación</b>  <b>Consultas generales:</b> <ul style="list-style-type: none"><li>• El sistema debe permitir a cualquier usuario interactuar con un agente de IA dentro del sistema.</li><li>• El agente debe poder responder preguntas frecuentes sobre el manejo y navegación del sistema.</li></ul> <b>Conocimiento del sistema:</b> <ul style="list-style-type: none"><li>• El agente de IA debe conocer las funciones asignadas a cada rol (por ejemplo: estudiante, docente, director de programa).</li><li>• El usuario puede preguntar por las responsabilidades y acciones permitidas según su rol.</li></ul> <b>Planes de estudio y cursos:</b> <ul style="list-style-type: none"><li>• El agente de IA debe poder proporcionar la información de un curso o asignatura.</li><li>• Debe ser posible obtener información de un programa académico completo.</li><li>• El agente debe poder comparar planes de estudio diferentes bajo solicitud del usuario.</li></ul> <b>Limitaciones y mensajes:</b> <ul style="list-style-type: none"><li>• El agente de IA solo debe responder preguntas relacionadas con el sistema.</li><li>• Si la consulta no está relacionada con el funcionamiento del sistema, el agente debe mostrar un mensaje claro indicando que no puede responder y limitarse únicamente a temas de la plataforma.</li><li>• Si el usuario no tiene permisos para acceder a cierta información, el agente debe informar y explicar por qué.</li></ul>	

### **3.2. Funciones**

#### **Decano**

- Iniciar sesión con credenciales.
- Asignar Directores de Programa.
- Evitar reasignación o eliminación del rol Decano.
- Acceso exclusivo al panel administrativo.

#### **Director de Programa**

- Subir, editar y validar planes de estudio.
- Cargar y gestionar microcurrículos y guías.
- Firmar o rechazar propuestas curriculares.
- Visualizar propuestas y enviarlas al comité.
- Asignar docentes a cursos.
- No puede reasignarse ni editar asignaciones del Decano.

#### **Docente**

- Crear y enviar propuestas curriculares.
- Consultar y descargar elementos del currículo.
- Ver estado de las propuestas.
- No puede editar currículo general ni asignaciones.

#### **Comité de Programa**

- Revisar propuestas de microcurrículos.
- Aprobar, rechazar o dejar en revisión.
- Registrar decisiones con fecha y usuario.
- Notificar automáticamente al docente.

#### **Director de Escuela**

- Visualizar propuestas aprobadas por el Comité.
- Firmar o rechazar propuestas.
- No puede crear ni editar propuestas.

#### **Invitado**

- Acceder al currículo general sin autenticación.
- Navegar por el plan de estudios sin permisos de edición ni descarga.

### **3.3. Requisitos de Rendimiento**

- El sistema debe responder a las acciones del usuario en un tiempo entre 500 ms y 1 segundo.
- Debe soportar más de 500 usuarios simultáneamente sin comprometer su rendimiento.

### **3.4. Restricciones de Diseño**

- El desarrollo debe realizarse exclusivamente en Java, utilizando el framework Spring Boot.
- La aplicación debe ser de escritorio, no web ni móvil.
- Se deben utilizar colores institucionales: rojo, blanco y negro.
- El sistema debe integrarse con el ícono y banner institucional.
- Toda funcionalidad de autenticación debe estar basada en credenciales en base de datos segura.

### **3.5. Atributos del Sistema**

- Seguridad: uso de autenticación por usuario y contraseña; cierre de sesión tras 10 minutos de inactividad.
- Fiabilidad: la validación de archivos y datos será rigurosa para evitar errores.
- Mantenibilidad: el sistema debe estar modularizado para facilitar actualizaciones futuras.
- Portabilidad: aunque es de escritorio, debe poder instalarse en distintos entornos con Java compatible.
- Usabilidad: interfaz clara, amigable y adaptada al perfil del usuario.

### **3.6. Otros Requisitos**

- Toda propuesta debe llevar trazabilidad: estado, fechas y responsables.
- Debe haber archivos de ejemplo disponibles para carga de p sum y plantillas de propuestas.
- Validaciones como tipo de archivo, peso m ximo (30MB), formato y virus ser n obligatorias.

## **4. Ap ndices**

### **4.1 Formatos de entrada y salida**

#### **a. Formato de archivo para carga del p sum (Excel):**

El archivo Excel que se cargue debe contener los siguientes campos como encabezado:

- C digo del curso
- Nombre del curso
-  rea
- ciclo
- Tipo de curso (te rico / pr ctico / te rico-pr ctico)
- Cr ditos
- Requisitos
- relaci n

*Nota:* Se proporcionar  un archivo de ejemplo con esta estructura.



#### **b. Formato de salida gráfica del plan de estudios:**

Se generará un gráfico que ilustre el plan de estudios de un programa académico (en este caso, Ingeniería de Sistemas) donde:

- Las asignaturas están organizadas por semestre.
- Cada asignatura tiene un código único.
- Las líneas de colores representan prerrequisitos entre asignaturas.
- Al hacer clic sobre una asignatura, el sistema debe mostrar información detallada (nombre, créditos, tipo de curso, intensidad horaria, etc.).

#### **4.3 Restricciones gráficas y visuales**

- El sistema debe seguir la identidad visual de la Universidad de los Llanos, lo cual incluye:
  - Paleta de colores: rojo , blanco, negro
  - Integración de logotipo institucional