
Manual Técnico

Proyecto:
Gestión curricular

Índice

1. Introducción	3
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
2. Descripción General	5
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
3. Requisitos Específicos	9
3.1. Requisitos funcionales	9
3.2. Funciones	22
3.3. Requisitos de Rendimiento	23
3.4. Restricciones de Diseño	23
3.5. Atributos del Sistema	23
3.6. Otros Requisitos	24
4. Apéndices	24
5. Arquitectura Del Software	25
5.1. Diagrama De Arquitectura	25
6. Modelamiento De Software	26
6.1. Diagrama De UML	26
6.2. Mockups	57
6.3. Modelo de datos	63
7. Modelo De Negocios	63
8. Documentación de Pruebas	63
9. Conclusiones	63

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 pésum.

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 pésum académico.
- Generación automática del plan de estudios a partir del Excel cargado.
- Visualización gráfica de los planes de estudio y prerequisitos.
- 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
Pénsum	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: [IEEE SA - IEEE 610.12-1990](#)
- 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 pénum 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: HU-01-1	Usuario: Director de programa
Título: Carga del pénsum académico	
Prioridad: Alta	Tipo: Historia
Descripción: Como Director de programa , quiero poder cargar los datos del pénsum académico desde un archivo Excel, para que la estructura curricular se almacene correctamente en el sistema y esté lista para su gestión y consulta.	
Criterios de aceptación <ul style="list-style-type: none"> • <i>El archivo subido debe ser de tipo Excel (.csv, .xls, .xlsx).</i> • <i>El archivo debe ser escaneado para verificar que esté libre de virus.</i> • <i>Se debe mostrar un mensaje de confirmación al completar la carga con éxito.</i> • <i>Los datos deben validarse para asegurar que estén completos y correctos.</i> • <i>Debe existir una interfaz para cargar el archivo del pénsum a través de un botón.</i> • <i>Debe estar disponible un archivo de ejemplo con la estructura requerida para la carga del pénsum (campos como: código del curso, nombre, créditos, intensidad horaria, tipo, prerequisitos, etc.).</i> 	

2. El sistema debe generar automáticamente el plan de estudios a partir de un archivo Excel cargado previamente. identifícate en esta monda

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

3. Gestión y validación de usuarios.

Decano:

Código: HU-03-1	Usuario: Decano
Título: Gestión de Decano	
Prioridad: Alta	Tipo: Historia
<p>Descripción: Como Decano 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.</p>	

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: HU-03-2	Usuario: Director de programa
Título: Gestión de Director de Programa	
Prioridad: Alta	Tipo: Historia
Descripción: Como Director de Programa , 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: HU-03-3	Usuario: Docente
Título: Gestión de Docentes	
Prioridad: Alta	Tipo: Historia
<p>Descripción: Como Docente, 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.</p>	
<p>Criterios de aceptación</p> <p>Consulta y descarga:</p> <ul style="list-style-type: none"> • Pueden consultar los perfiles, competencias y resultados de aprendizaje asociados a su curso. • Pueden descargar el currículo completo. • No pueden modificar directamente el currículo general ni los cursos asignados. <p>Restricciones de acceso:</p> <ul style="list-style-type: none"> • No tienen permisos para asignar roles ni modificar datos de otros usuarios. • No pueden eliminar propuestas ni registros curriculares anteriores. <p>Propuesta:</p> <ul style="list-style-type: none"> • El docente puede crear propuestas curriculares cargando los documentos de soporte necesarios. <p>Interfaz:</p> <ul style="list-style-type: none"> • Debe existir una sección clara de “<i>Manejar mis Cursos</i>” donde el docente pueda crear y subir. • El sistema debe mostrar el estado de la propuesta y posibles observaciones del Director de Programa. 	

Comité de programa:

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

Descripción:

Como **Comité de programa**, 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.

Criterios de aceptación**Revisión de propuestas:**

- El sistema debe permitir al Comité de programa visualizar todas las propuestas enviadas por los Directores de programa.
- Cada propuesta debe poder ser aprobada o rechazada.
- El historial de decisiones debe quedar registrado con la fecha, estado actual y usuario que realizó la acción.

Restricciones de acceso:

- Solo pueden acceder a las propuestas enviadas.
- No pueden modificar el contenido original de los microcurrículos.
- No pueden editar perfiles, competencias u otros componentes curriculares directamente.

Interfaz:

- El sistema debe permitir ver los detalles de cada propuesta y registrar observaciones fácilmente.

Invitado:

Código: HU-03-5	Usuario: Invitado
Título: Visualización del currículo como usuario invitado	
Prioridad: Alta	Tipo: Historia
Descripción:	
Como Invitado , 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.	

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: HU-03-6	Usuario: Director de escuela
Título: Gestión Director de escuela	
Prioridad: Alta	Tipo: Historia
Descripción: Como Director de Escuela , 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:	
<ul style="list-style-type: none">• 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: HU-03-7	Usuario: Docente
Título: Peticiones para cambios de contenidos	
Prioridad: Alta	Tipo: Historia
Descripción: 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.	

Criterios de aceptación

Propuestas académicas:

- El sistema debe permitir a los Docentes cargar propuestas de microcurrículos, asociados a cursos existentes.
- Cada propuesta debe quedar registrada con la fecha y el estado (pendiente, en revisión, aceptada, rechazada).

Consulta y visualización:

- Debe haber una sección correspondiente a las peticiones donde los docentes puedan observar el progreso de la petición realizada.
- 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.

Interfaz:

- Debe haber una sección de carga de peticiones para enviar la petición correspondiente.

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

Código: HU-04	Usuario: Usuario sin rol específico
Título: Planes de estudio de forma gráfica.	
Prioridad: Alta	Tipo: Historia
Descripción: Como usuario , 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.	

Criterios de aceptación

Visualización del Plan de Estudios

- El sistema debe generar una representación gráfica del plan de estudios, mostrando las asignaturas organizadas por semestre.
- Se debe representar la relación entre asignaturas y sus prerequisitos mediante conexiones visuales.

Histórico de Planes de Estudio

- El sistema debe ofrecer una opción para consultar versiones anteriores del plan de estudios.
- Al seleccionar una versión anterior, esta debe visualizarse gráficamente con el mismo formato del plan actual.

Validaciones y Errores

- La vista gráfica debe actualizarse automáticamente cuando se realicen modificaciones en el plan de estudios.

Interfaz:

- 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.

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

Código: HU-05	Usuario: Director de programa
Título: Cargar microcurrículos	
Prioridad: Alta	Tipo: Historia
Descripción: Como Director de Programa , 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.	

Criterios de aceptación

Carga de Documentos

- El sistema debe permitir subir archivos en formatos PDF, DOCX o similares.
- Se debe poder asociar cada documento a una asignatura específica.
- Una vez cargado, el sistema debe mostrar un mensaje de confirmación y permitir la consulta inmediata.

Gestión de Documentos

- Se debe poder cargar y actualizar.
- Los directores de programa pueden gestionar los documentos de todas las asignaturas.
- Se debe permitir la descarga de los documentos por parte de los usuarios autorizados.

Histórico de Documentos

- El sistema debe mantener un histórico de versiones anteriores de microcurrículos asociados a los planes de estudio anteriores.

Validaciones y Errores

- Si un archivo supera el tamaño permitido (30MB), el sistema debe mostrar un mensaje de error.
- Si el formato del archivo no es válido, se debe rechazar la carga con una notificación.

IA:

Código: HU-06	Usuario: Cualquier usuario (independiente del rol)
Título: Interacción con agente de IA	
Prioridad: Alta	Tipo: Historia
Descripción: Como usuario con cualquier rol 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.	

Criterios de aceptación

Consultas generales:

- El sistema debe permitir a cualquier usuario interactuar con un agente de IA dentro del sistema.
- El agente debe poder responder preguntas frecuentes sobre el manejo y navegación del sistema.

Conocimiento del sistema:

- El agente de IA debe conocer las funciones asignadas a cada rol (por ejemplo: estudiante, docente, director de programa).
- El usuario puede preguntar por las responsabilidades y acciones permitidas según su rol.

Planes de estudio y cursos:

- El agente de IA debe poder proporcionar la información de un curso o asignatura.
- Debe ser posible obtener información de un programa académico completo.
- El agente debe poder comparar planes de estudio diferentes bajo solicitud del usuario.

Limitaciones y mensajes:

- El agente de IA solo debe responder preguntas relacionadas con el sistema.
- 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.
- Si el usuario no tiene permisos para acceder a cierta información, el agente debe informar y explicar por qué.

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 pénum y plantillas de propuestas.
- Validaciones como tipo de archivo, peso máximo (30MB), formato y virus serán obligatorias.

4. Apéndices

4.1 Formatos de entrada y salida

a. Formato de archivo para carga del pénum (Excel):

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

- Código del curso
- Nombre del curso
- Área
- ciclo
- Tipo de curso (teórico / práctico / teórico-práctico)
- Créditos
- Requisitos
- relación

Nota: Se proporcionará 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:

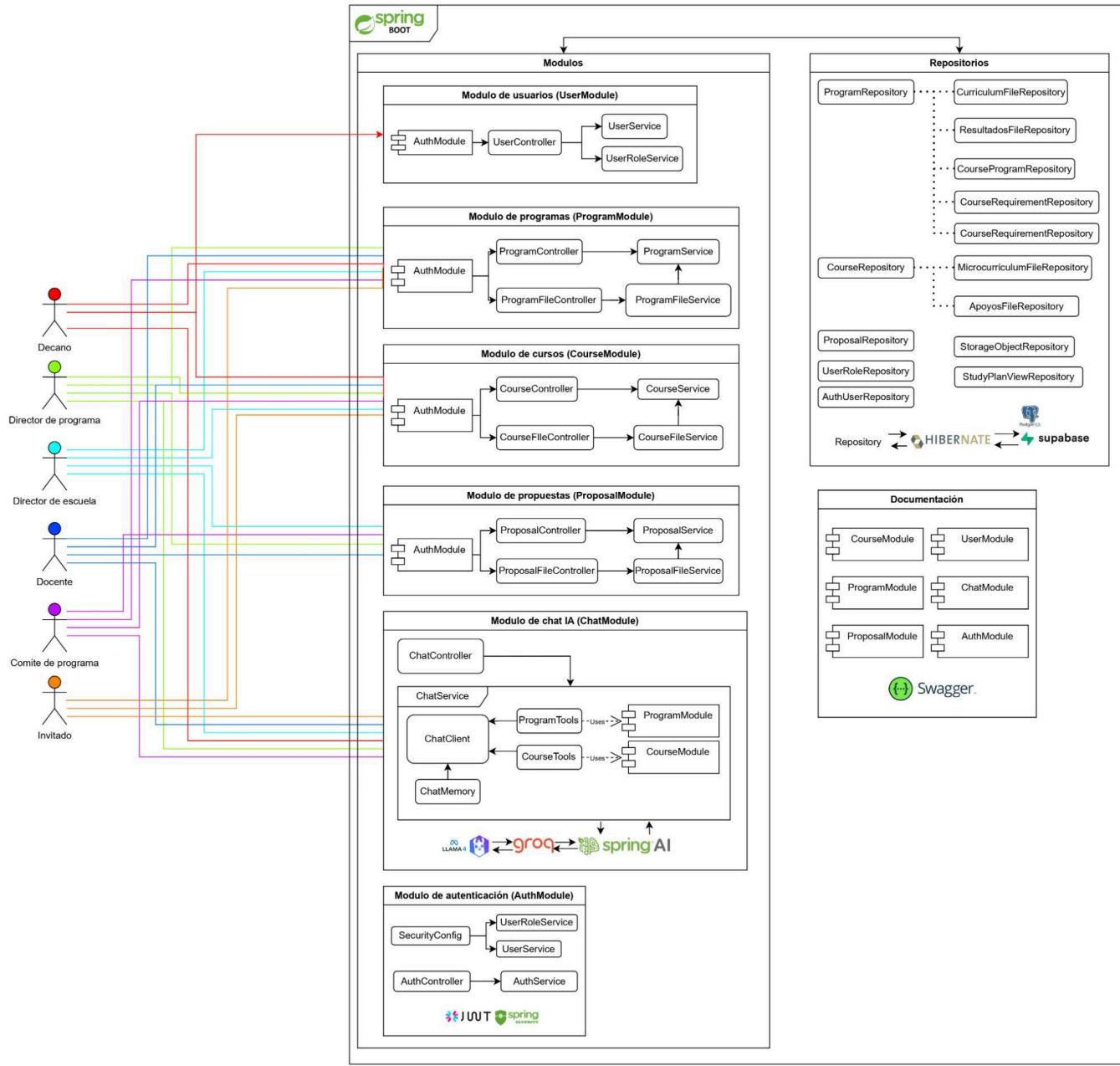
- Las asignaturas están organizadas por semestre.
- Cada asignatura tiene un código único.
- Las líneas de colores representan prerequisitos entre asignaturas.
- Al hacer clic sobre una asignatura, el sistema debe mostrar información detallada (nombre, créditos, tipo de curso, intensidad horaria, etc.).

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

5. Arquitectura Del Software

5.1. Diagrama De Arquitectura

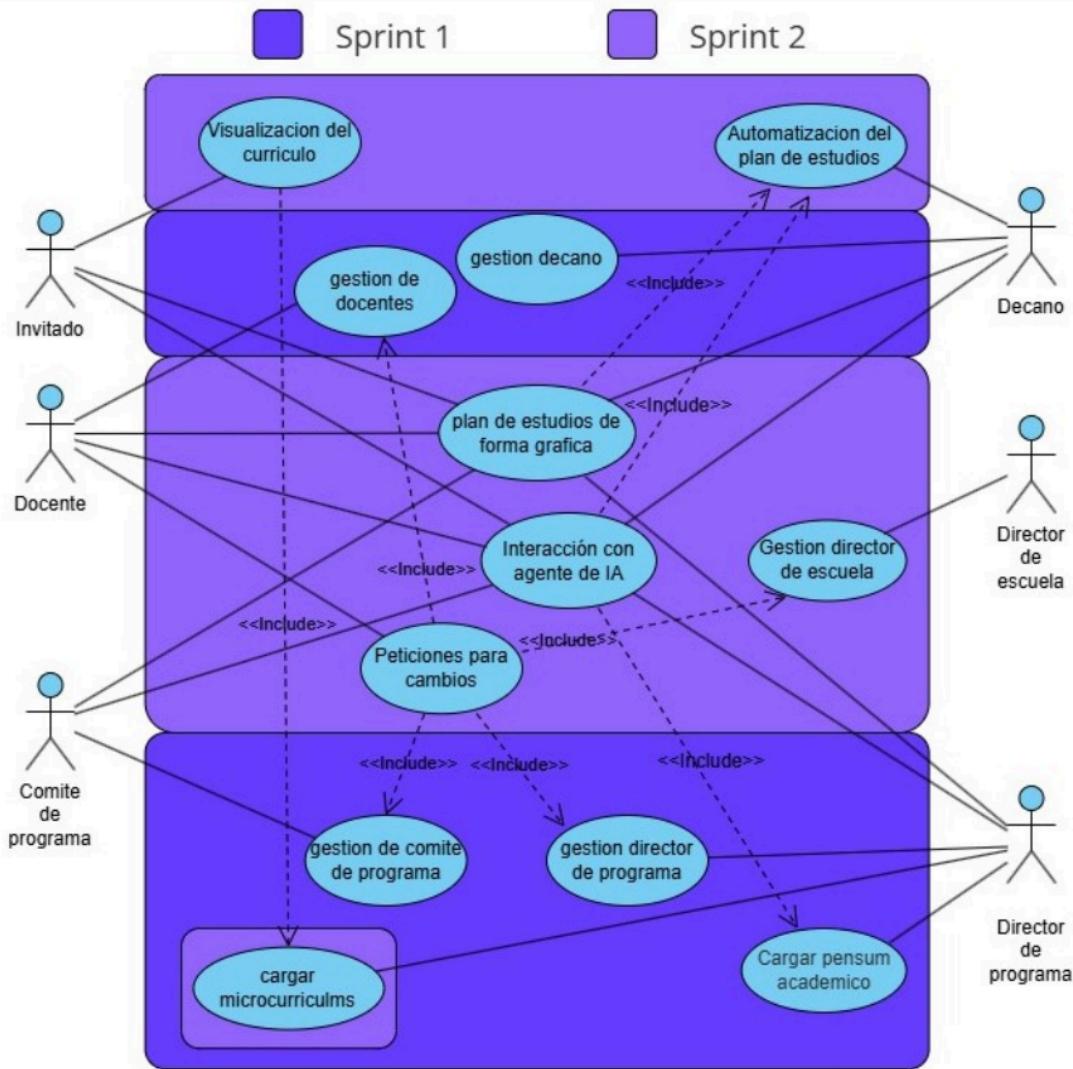


6. Modelamiento De Software

6.1. Diagrama De UML

6.1.0. Diagramas De Comportamiento

6.1.0.0. Diagrama Casos De Uso



Convención de colores

- Azul oscuro(rectángulos de fondo): funcionalidades comprometidas para Sprint 1.
- Violeta claro(rectángulos “superpuestos”): funcionalidades añadidas o ampliadas en Sprint 2.
- Todos los óvalos en un área heredan el color del sprint al que pertenecen.

Actores externos

- Actor Rol en el sistema
- Invitado Usuario sin credenciales, solo consulta.
- Docente Profesor que administra asignaturas y peticiones.
- Comité de programa Órgano que revisa micro-curículos.
- Decano Nivel directivo superior, aprueba planes.
- Director de escuela Supervisa varias carreras.
- Director de programa Administra un programa académico.

Las líneas sólidas conectan a cada actor con los casos de uso que puede iniciar.

Casos de uso por sprint

Sprint 1 (azul oscuro)

- Gestión de docentes – Los Docentes crean/actualizan su perfil académico.

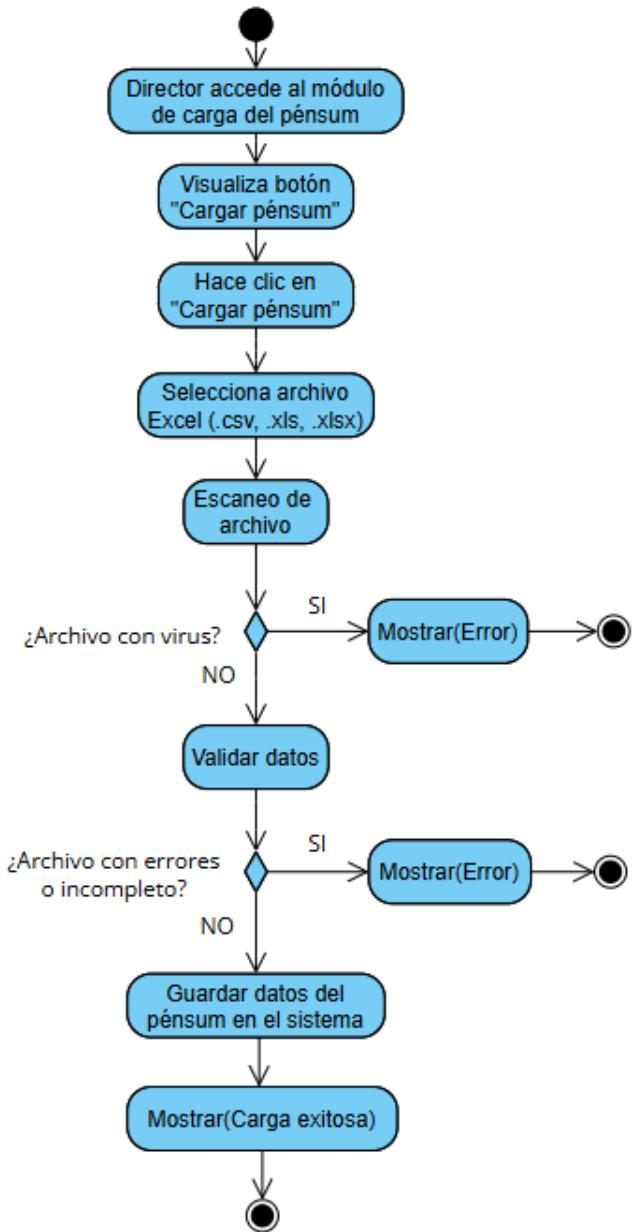
- · Gestión comité de programa – El Comité mantiene su composición.
- · Carga pensum académico – Sube el listado oficial de materias; incluye “Gestión de información adicional”.
- · Gestión decano – Perfil administrativo del Decano.
- · Gestión director de programa – Gestionar el currículo completo: perfiles, competencias, resultados, cursos, tipo de curso.

Sprint 2 (violeta claro)

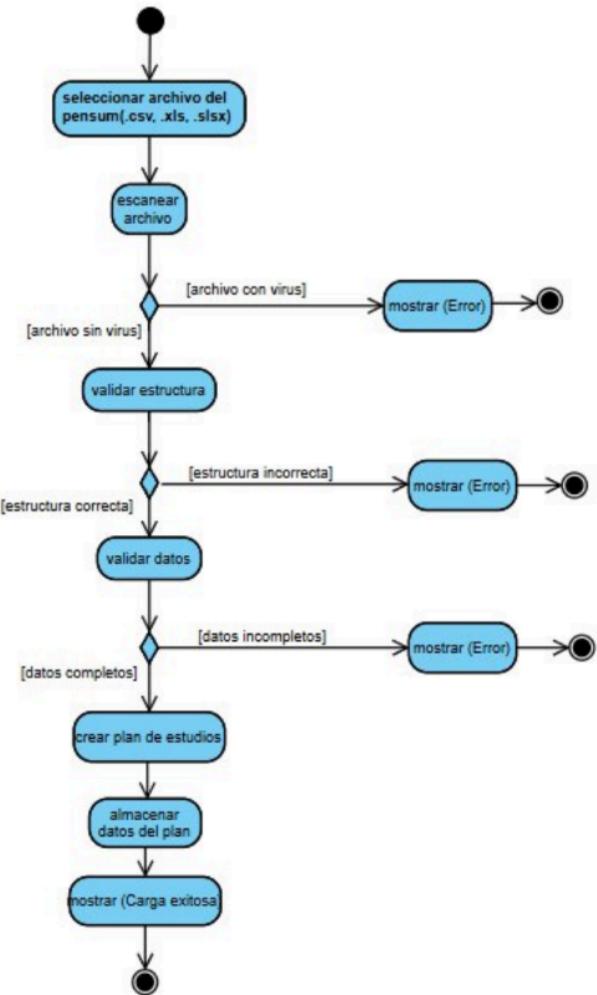
- · Automatización del plan de estudios – Genera secuencias de cursos.
- · Gestión director de escuela – Similar, para Director de escuela.
- · Visualización del currículo – Cualquier Invitado puede ver el plan vigente.
- · Plan de estudios de forma gráfica – Docente y Director de escuela visualizan un mapa gráfico de cursos.
- · Carga de microcurrículos – Sub-flujo incluido («include») desde “Gestión de comité...”.
- · Peticiones para cambios – Docentes proponen modificaciones; dispara flujo de aprobación.
- · Interacción con agente de IA – Facilita y mejora la experiencia de uso de la aplicación para todos los usuarios según su rol.

6.1.0.1. Diagrama De Actividades

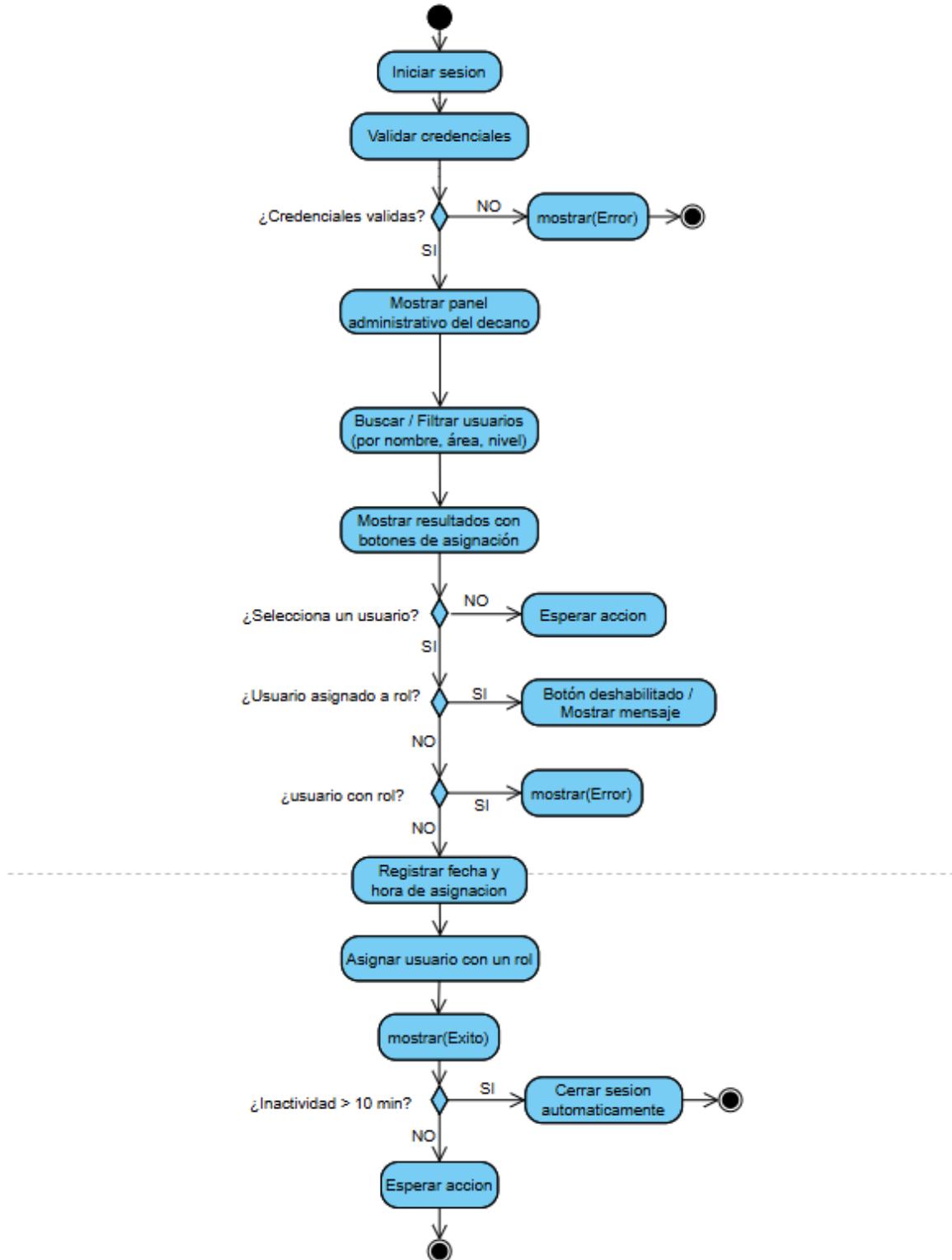
Carga del pesum academico



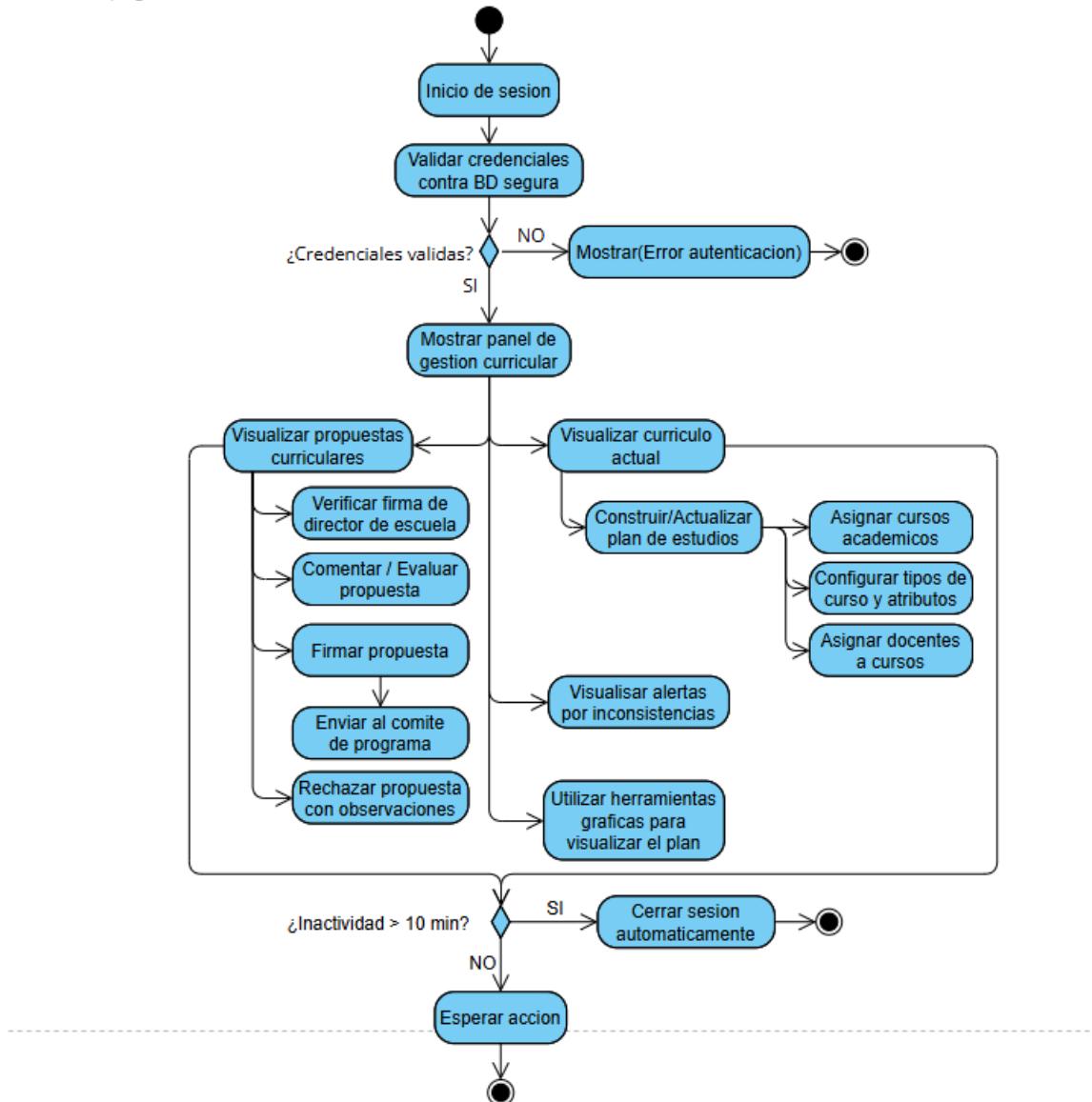
Automatizacion del plan de estudios



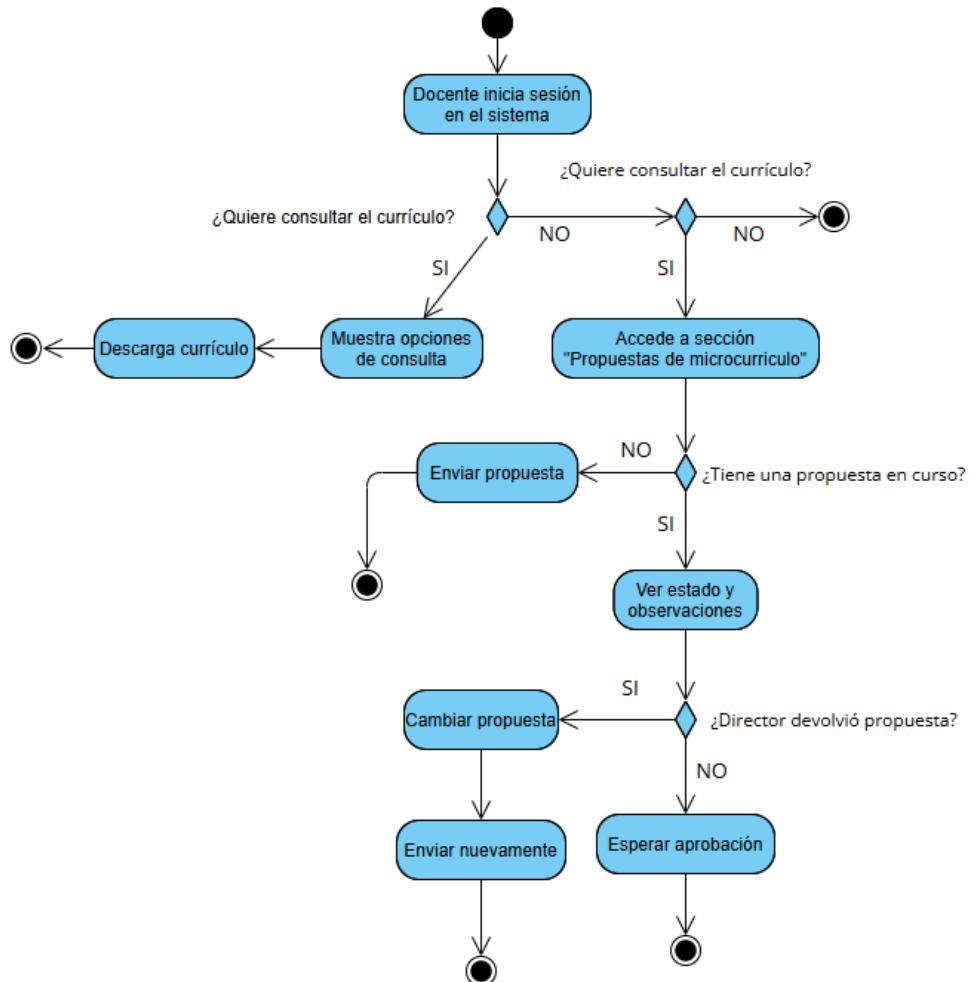
Decano



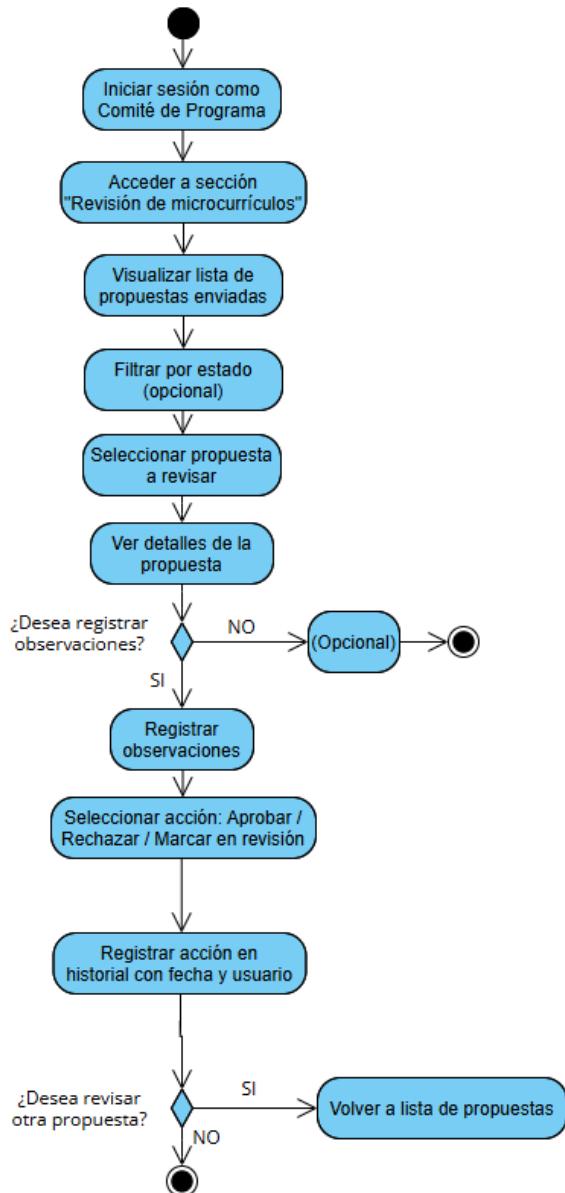
Director de programa



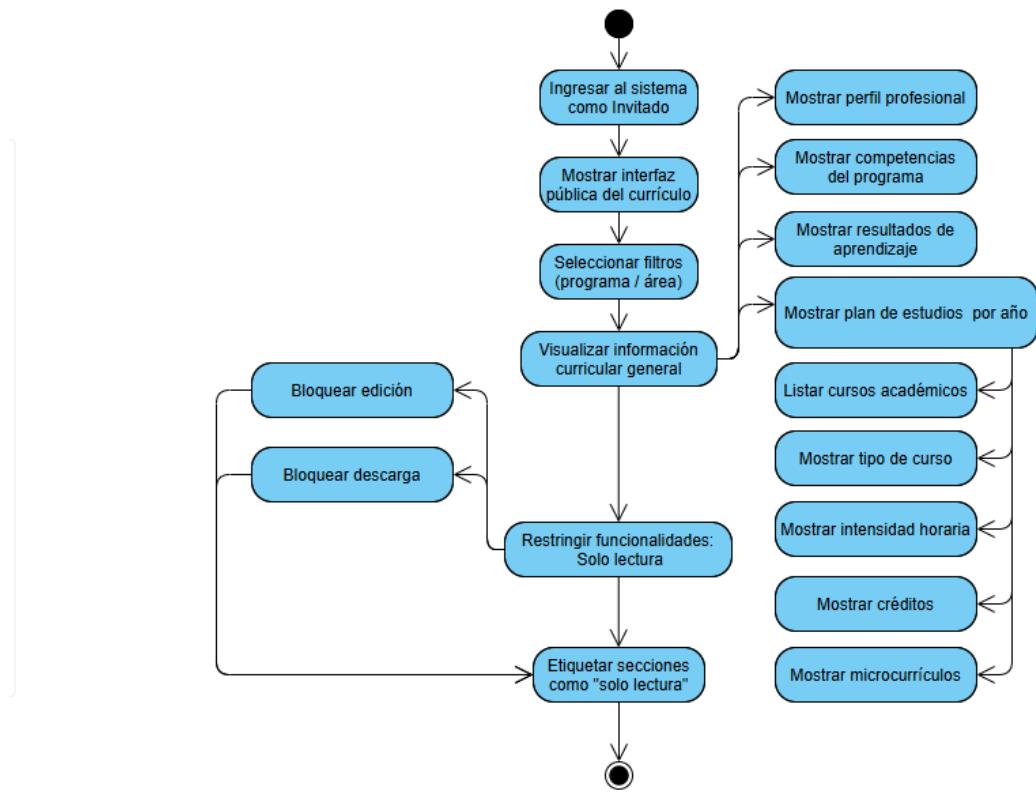
Gestion de docentes



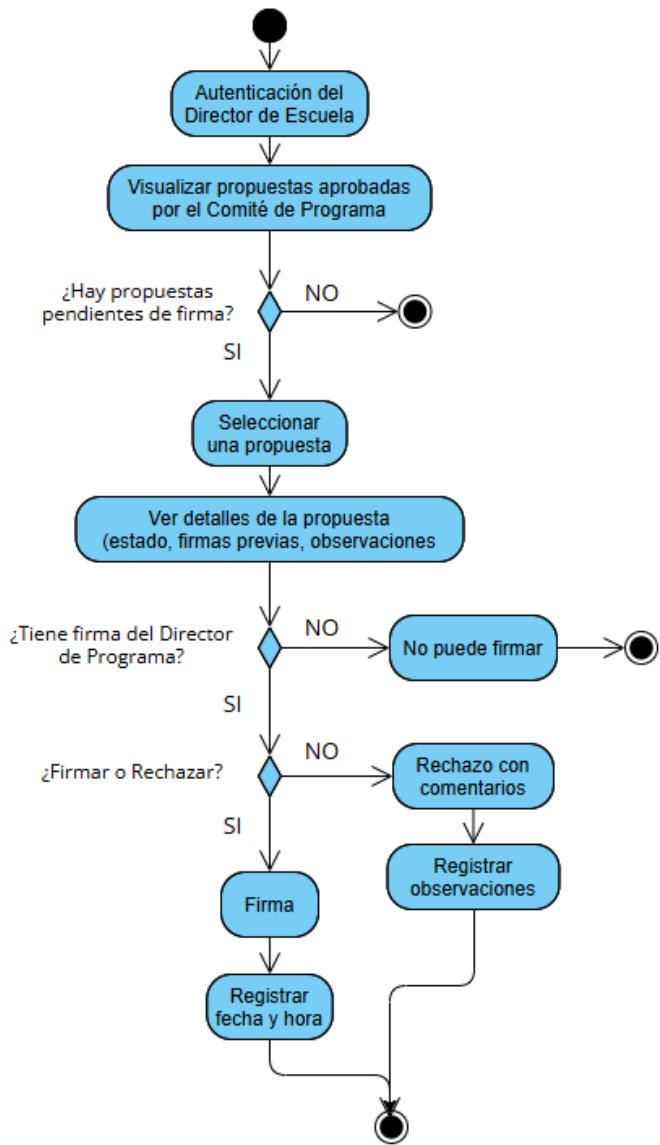
Gestion de comite de programa



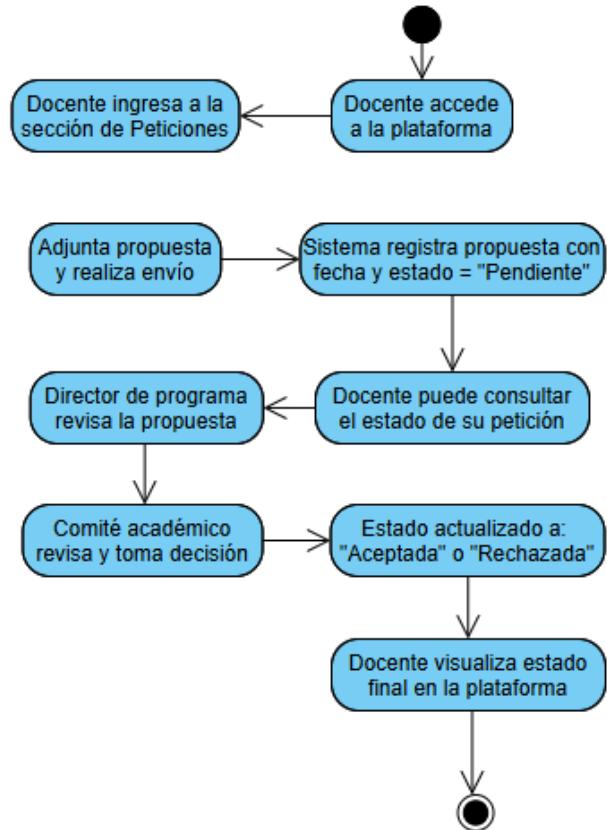
Visualización del Currículo como Usuario Invitado



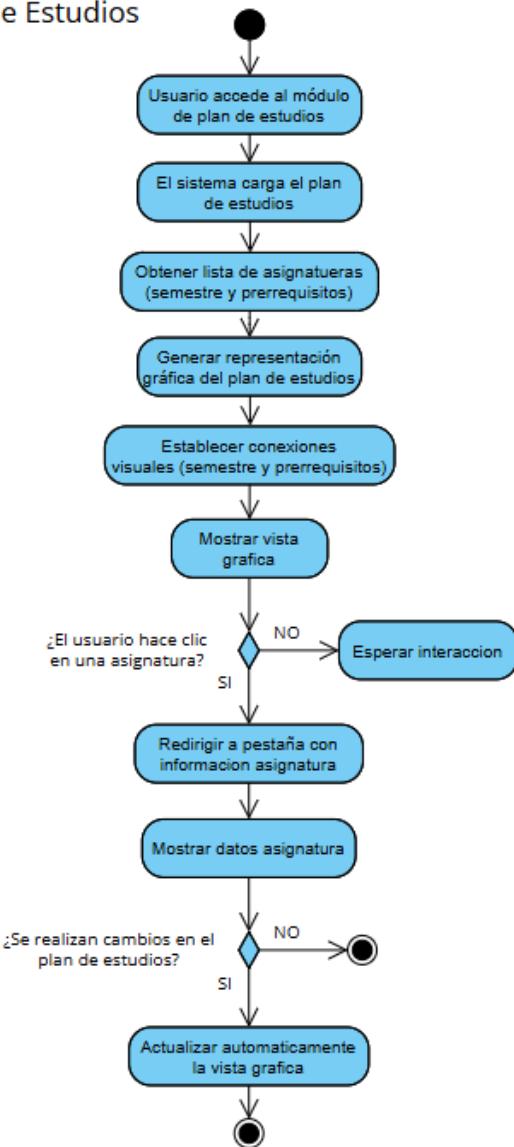
Gestión director de escuela



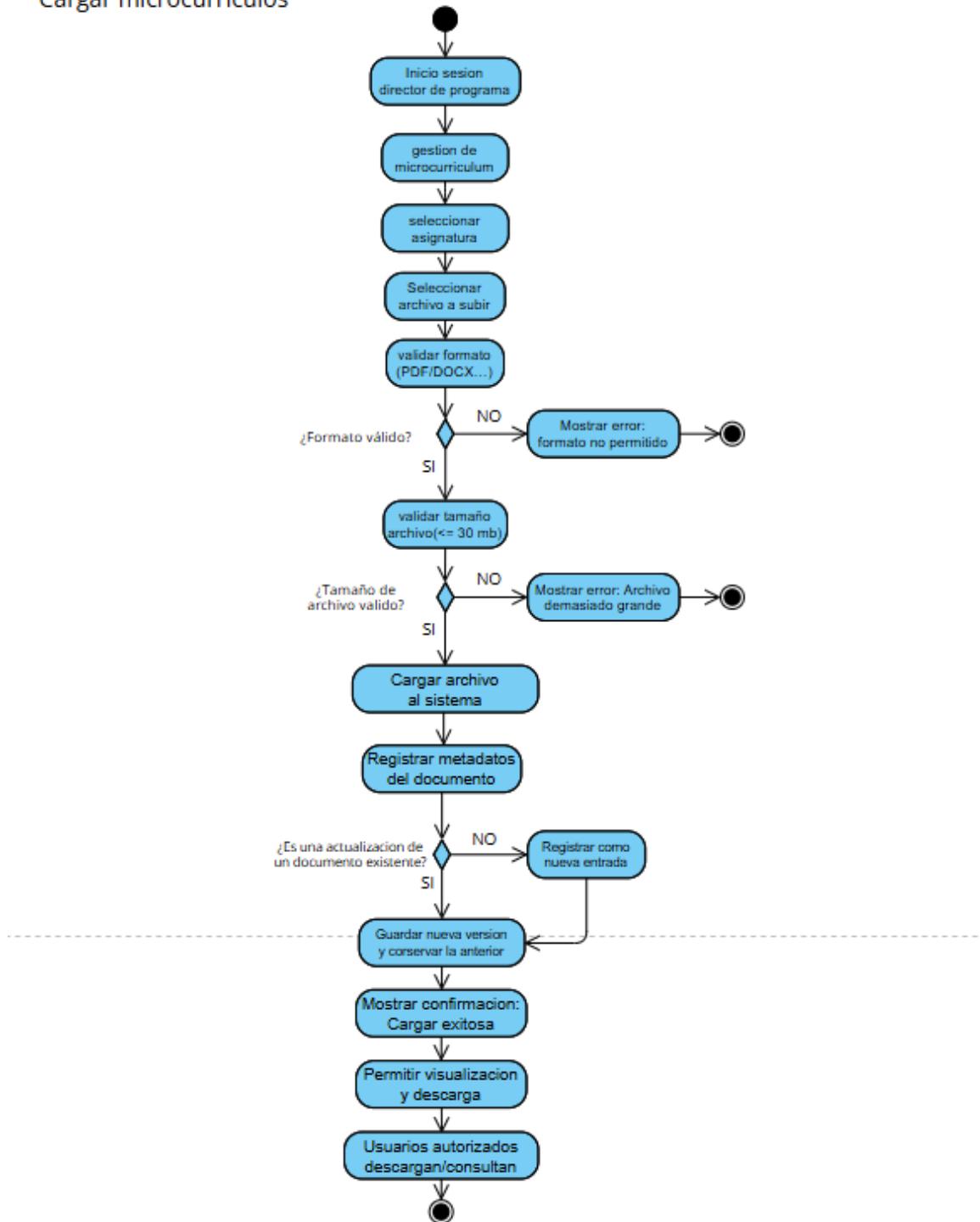
Peticiones para cambios de contenidos



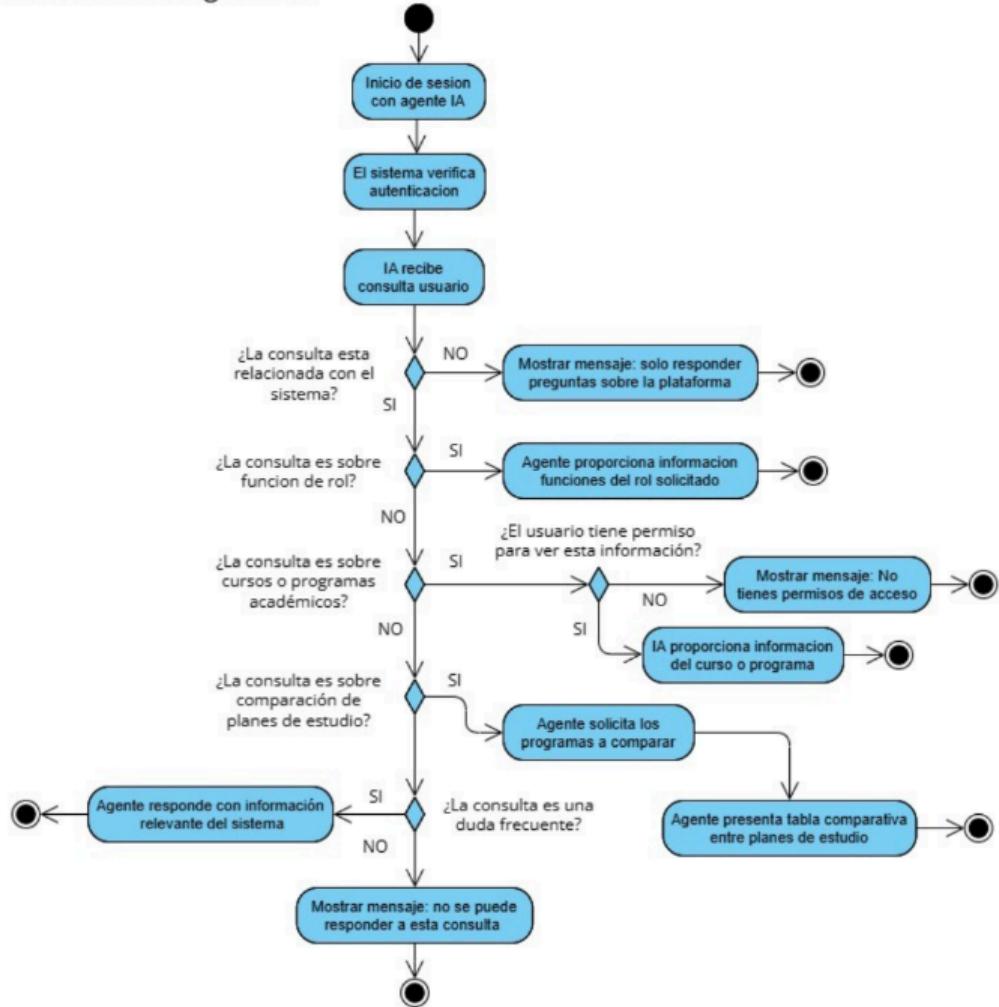
Visualización Gráfica del Plan de Estudios



Cargar microcurrículos

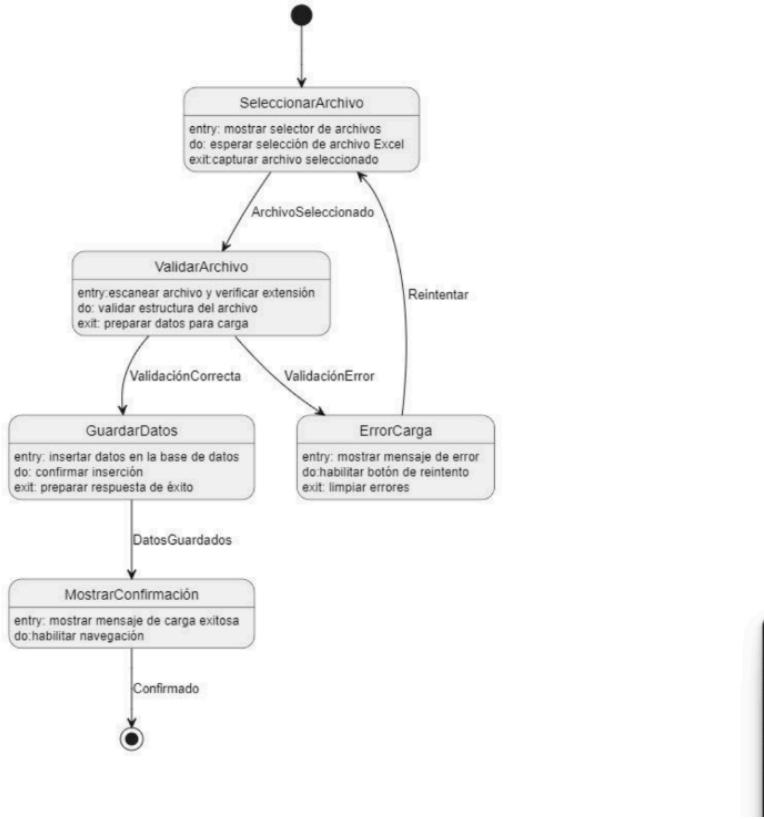


Interaccion con agente IA



6.1.0.2. Diagrama De Estados

6.1.0.2.0. Cargue de archivo excel:



1. Inicio del Proceso

- a. Representado por un círculo negro, marca el comienzo del proceso de carga del archivo.

2. Estado: SeleccionarArchivo

- a. Descripción: El sistema solicita al usuario que seleccione un archivo

Excel.

- b. Transiciones:

- i. Entry: Mostrar el selector de archivos.
- ii. Do: Esperar que el usuario seleccione un archivo.
- iii. Exit: Capturar el archivo seleccionado.

- c. Evento: ArchivoSeleccionado, transición hacia ValidarArchivo.

3. Estado: ValidarArchivo

- a. Descripción: El sistema verifica que el archivo sea válido en formato y

contenido.

- b. Transiciones:

- i. Entry: Escanear el archivo y verificar su extensión.

ii. Do: Validar la estructura interna del archivo.

iii. Exit: Preparar los datos para la carga.

c. Eventos:

i. ValidaciónCorrecta: transición hacia GuardarDatos.

ii. ValidaciónError: transición hacia ErrorCarga.

1. Estado: GuardarDatos

d. Descripción: Los datos del archivo son almacenados en la base de

datos. e. Transiciones:

i. Entry: Insertar los datos en la base de datos.

ii. Do: Confirmar que los datos se han insertado correctamente.

iii. Exit: Preparar respuesta de éxito.

f. Evento: DatosGuardados, transición hacia MostrarConfirmación.

4. Estado: ErrorCarga

a. Descripción: El sistema detecta un error al validar el archivo.

b. Transiciones:

i. Entry: Mostrar mensaje de error al usuario.

ii. Do: Habilitar el botón de "Reintentar" para volver a intentar la
carga. iii. Exit: Limpiar errores al salir del estado.

c. Evento: Reintentar, transición de regreso a SeleccionarArchivo.

5. Estado: MostrarConfirmación

a. Descripción: Se notifica al usuario que la carga fue exitosa.

b. Transiciones:

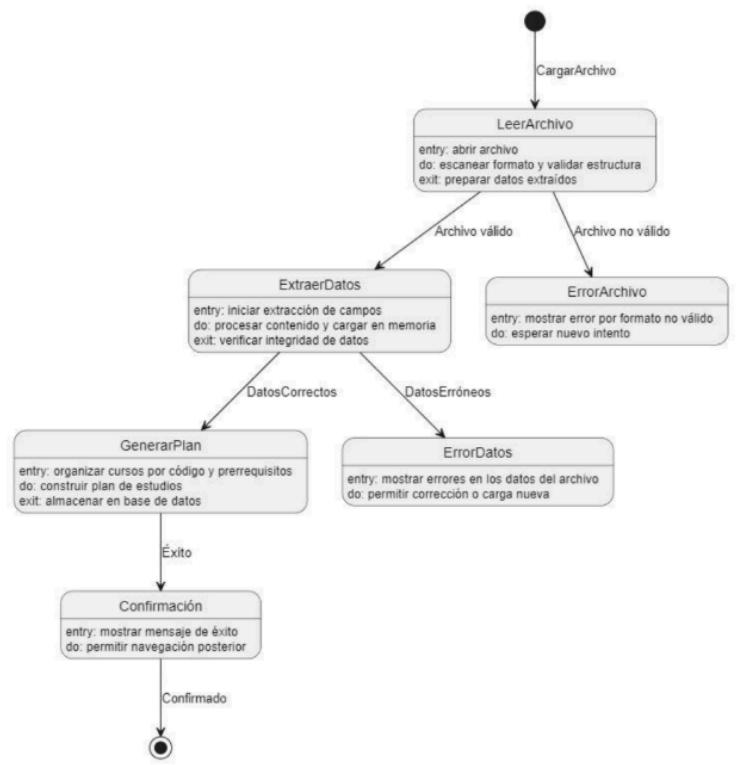
i. Entry: Mostrar un mensaje de éxito.

ii. Do: Habilitar la navegación posterior del usuario.

6. Estado Final

a. Representado por un círculo negro con borde blanco, indicando que el proceso
ha terminado exitosamente.

6.1.0.2.1. Cargue de archivo excel:



1. Inicio del Proceso

a. Representado por un círculo negro, marca el inicio del flujo de carga del

archivo. 2. Estado: CargarArchivo

a. Descripción: Se inicia el proceso de carga de un archivo para su procesamiento posterior.

3. Estado: LeerArchivo

a. Descripción: El sistema abre el archivo y analiza su estructura.

b. Transiciones:

i. Entry: Abrir el archivo.

ii. Do: Escanear el formato y validar su estructura.

iii. Exit: Preparar los datos extraídos.

c. Eventos:

- i. Archivo válido: transición hacia ExtraerDatos.
- ii. Archivo no válido: transición hacia ErrorArchivo.

4. Estado: ExtraerDatos

a. Descripción: El sistema extrae los datos del archivo válido.

b. Transiciones:

- i. Entry: Iniciar extracción de campos.
- ii. Do: Procesar el contenido y cargar en memoria.
- iii. Exit: Verificar la integridad de los datos.

c. Eventos:

- i. DatosCorrectos: transición hacia GenerarPlan.
- ii. DatosErróneos: transición hacia ErrorDatos.

5. Estado: ErrorArchivo

a. Descripción: El sistema detecta que el archivo no es válido en formato. b. Transiciones:

- i. Entry: Mostrar error por formato no válido.
- ii. Do: Esperar nuevo intento de carga.

6. Estado: GenerarPlan

a. Descripción: Se organiza y construye un plan de estudios con los datos correctos. b. Transiciones:

- i. Entry: Organizar cursos por código y prerequisitos.
- ii. Do: Construir el plan de estudios.
- iii. Exit: Almacenar en base de datos.

c. Evento: Éxito, transición hacia Confirmación.

7. Estado: ErrorDatos

a. Descripción: Se detectan errores en los datos extraídos.

b. Transiciones:

- i. Entry: Mostrar errores en los datos del archivo.
- ii. Do: Permitir corrección o carga de un nuevo archivo.

8. Estado: Confirmación

a. Descripción: Se muestra un mensaje de éxito al usuario.

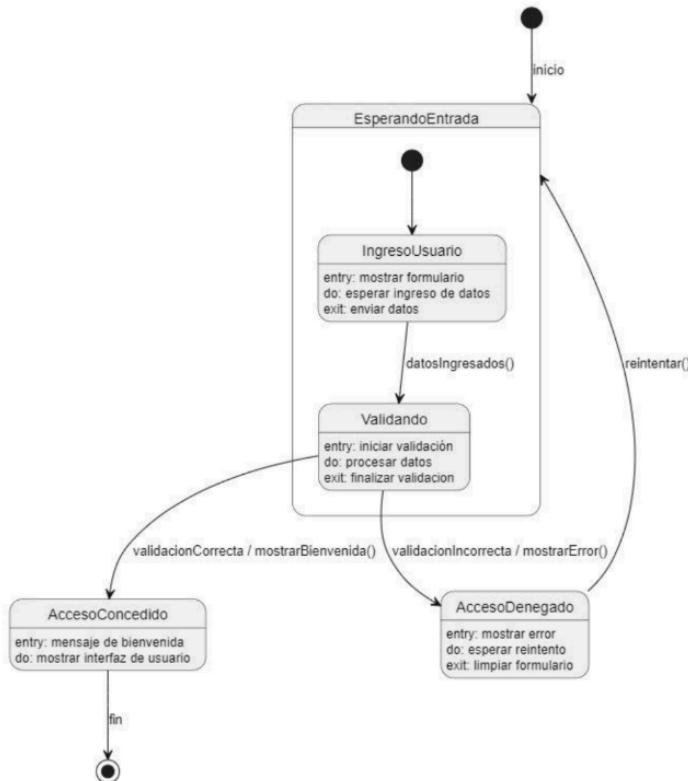
b. Transiciones:

- i. Entry: Mostrar mensaje de éxito.
- ii. Do: Permitir la navegación posterior.

9. Estado Final

a. Representado por un círculo negro con borde blanco, indicando que el proceso terminó satisfactoriamente.

6.1.0.2.2. Gestión y validación de usuarios:



1. Inicio del Proceso

a. Representado por un círculo negro, marca el inicio del flujo de validación de acceso. 2. Estado: EsperandoEntrada (Estado compuesto)

a. Descripción: Agrupa los estados iniciales de interacción del usuario antes de validar el acceso.

b. Comienza automáticamente en IngresoUsuario.

3. Estado: IngresoUsuario

a. Descripción: El sistema muestra el formulario de ingreso para que el usuario introduzca sus datos.

b. Transiciones:

i. Entry: Mostrar formulario.

ii. Do: Esperar el ingreso de datos.

iii. Exit: Enviar datos para validación.

c. Evento:

i. datosIngresados(): transición hacia Validando.

4. Estado: Validando

a. Descripción: Se procesan y validan los datos ingresados por el usuario. b. Transiciones:

i. Entry: Iniciar la validación.

ii. Do: Procesar los datos.

iii. Exit: Finalizar la validación.

c. Eventos:

i. validacionCorrecta / mostrarBienvenida(): transición hacia AccesoConcedido.

ii. validacionIncorrecta / mostrarError(): transición hacia AccesoDenegado. 5. Estado: AccesoConcedido

a. Descripción: El usuario ingresó datos válidos y el acceso es permitido. b. Transiciones:

i. Entry: Mostrar mensaje de bienvenida.

ii. Do: Mostrar la interfaz de usuario.

6. Estado: AccesoDenegado

a. Descripción: Los datos ingresados son incorrectos, y el acceso es denegado.

b. Transiciones:

i. Entry: Mostrar mensaje de error.

ii. Do: Esperar nuevo intento.

iii. Exit: Limpiar el formulario.

c. Evento:

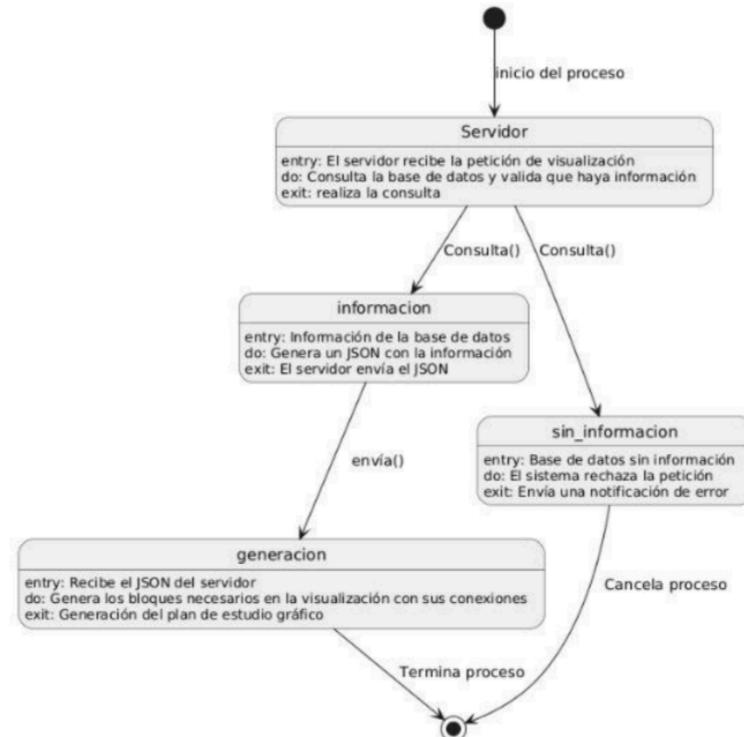
i. reintentar(): transición de regreso a EsperandoEntrada.

7. Estado Final

a. Representado por un círculo negro con borde blanco, indicando que el flujo termina una vez que el acceso ha sido concedido.

6.1.0.2.3. Visualización gráfica de los planes de estudio:

Diagrama de Estados - Revisión de Visualización Gráfica de los Planes de Estudio



1. Inicio del Proceso

a. Indicado por el círculo negro sólido.

2. Estado: Servidor

a. Descripción: El servidor recibe la petición para visualizar el plan de

estudios. b. Acciones:

- i. Entry: Recibe la petición.
- ii. Do: Consulta la base de datos y valida si hay información.
- iii. Exit: Realiza la consulta.

c. Transiciones:

- i. Si hay información, pasa a informacion.
- ii. Si no hay información, pasa a sin_informacion.

3. Estado: informacion

a. Descripción: Se genera un JSON con los datos consultados.

b. Acciones:

- i. Entry: Información de la base de datos.
- ii. Do: Generar JSON.
- iii. Exit: El servidor envía el JSON.

4. Estado: sin_informacion

a. Descripción: Se detecta que no hay datos en la base de

datos. b. Acciones:

- i. Entry: Se reconoce que no hay información.
- ii. Do: El sistema rechaza la petición.
- iii. Exit: Se envía una notificación de error.

c. Resultado: Se cancela el proceso.

5. Estado: generacion

a. Descripción: Se recibe el JSON y se genera la visualización

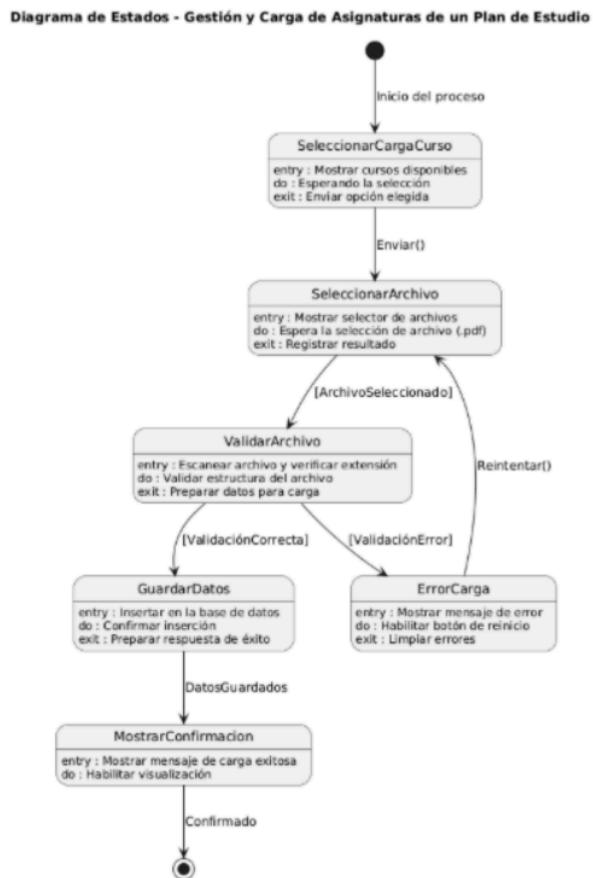
gráfica. b. Acciones:

- i. Entry: Recibe JSON del servidor.
- ii. Do: Generar bloques y conexiones para el plan de estudio gráfico.
- iii. Exit: Generación completada.

6. Estado Final

- a. Representado por el círculo negro con borde blanco:
- b. Termina el proceso (en éxito) o cancela el proceso (en fallo).

6.1.0.2.4. Gestión y carga de asignaturas de un plan de estudio:



1. Inicio del Proceso
 - a. Representado por un círculo negro sólido, indica el arranque de la gestión y carga de asignaturas.
2. Estado: SeleccionarCargaCurso
 - a. Descripción: Se presentan los cursos disponibles para que el usuario seleccione uno.
 - b. Transiciones:
 - i. Entry: Mostrar los cursos disponibles.
 - ii. Do: Esperar a que el usuario haga su elección.
 - iii. Exit: Enviar la opción elegida (enviar()).

3. Estado: SeleccionarArchivo

a. Descripción: Se despliega un selector para que el usuario suba un archivo

.pdf. b. Transiciones:

i. Entry: Mostrar el selector de archivos.

ii. Do: Esperar la selección del archivo.

iii. Exit: Registrar el archivo seleccionado (ArchivoSeleccionado).

4. Estado: ValidarArchivo

a. Descripción: Se valida que el archivo subido sea correcto en estructura y formato. b. Transiciones:

i. Entry: Escanear archivo y verificar su extensión.

ii. Do: Validar la estructura del archivo.

iii. Exit: Preparar los datos para carga:

iv. Si es correcto: transición hacia GuardarDatos

(ValidaciónCorrecta). v. Si hay error: transición hacia ErrorCarga

(ValidaciónError).

5. Estado: GuardarDatos

a. Descripción: Se guardan los datos validados en la base de datos.

b. Transiciones:

i. Entry: Insertar en la base de datos.

ii. Do: Confirmar la inserción.

iii. Exit: Preparar respuesta de éxito (DatosGuardados).

6. Estado: ErrorCarga

a. Descripción: Se maneja el error en caso de que el archivo no sea válido. b. Transiciones:

i. Entry: Mostrar mensaje de error.

ii. Do: Habilitar el botón de reinicio.

iii. Exit: Limpiar errores (Reintentar()).

7. Estado: MostrarConfirmación

a. Descripción: Se confirma al usuario que la carga fue exitosa.

b. Transiciones:

i. Entry: Mostrar mensaje de éxito.

ii. Do: Habilitar visualización de resultados.

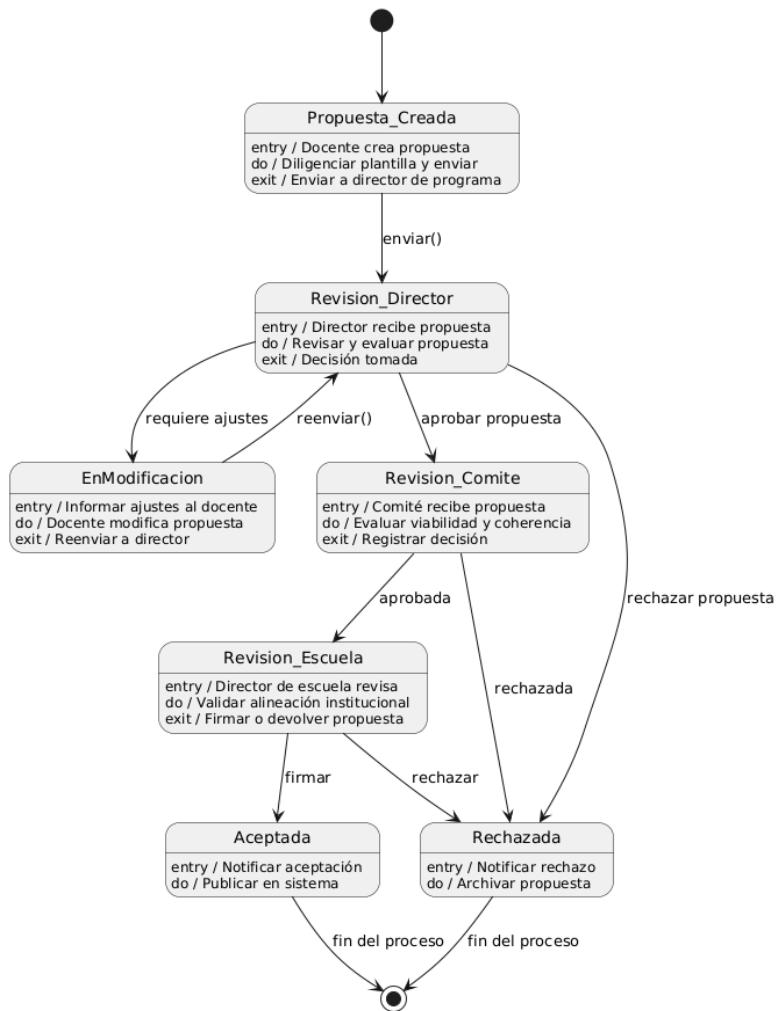
8. Estado Final

a. Representado por un círculo negro con borde blanco, marcando el final del proceso (Confirmado).

6.1.0.2.5.

Peticiones de actualización de microcurrículum:

Diagrama de Estados - Gestión de Propuestas Curriculares



- Inicio del Proceso

Representado por el círculo negro sólido. Indica el inicio del flujo cuando un docente decide enviar una propuesta curricular.

- Estado: Propuesta_Creada

El docente crea la propuesta diligenciando la plantilla correspondiente.
Transiciones:

- entry / Docente crea propuesta
- do / Diligenciar plantilla y enviar
- exit / Enviar a director de programa (enviar())

- Estado: Revision_Director

El Director de Programa recibe la propuesta, la revisa, y decide si requiere ajustes, si es aprobada o si debe ser rechazada.

Transiciones:

- Si la propuesta requiere ajustes: pasa a EnModificacion.
 - Si la propuesta es aprobada: pasa a Revision_Comite.
 - Si la propuesta es rechazada: pasa a Rechazada.
- Estado: EnModificacion

El docente recibe retroalimentación del Director de Programa, realiza las correcciones necesarias y vuelve a enviar la propuesta.

Transiciones:

 - entry / Informar ajustes al docente
 - do / Modificar y corregir propuesta
 - exit / Reenviar propuesta (reenviar()), que regresa a Revision_Director.
- Estado: Revision_Comite

El Comité de Programa evalúa la propuesta para garantizar su coherencia y calidad académica.

Transiciones:

 - Si la propuesta es aprobada: pasa a Revision_Escuela.
 - Si la propuesta es rechazada: pasa a Rechazada.
- Estado: Revision_Escuela

El Director de Escuela revisa la propuesta aprobada por el Comité y decide si la firma para su incorporación al currículo o la rechaza.

Transiciones:

 - Si la propuesta es firmada: pasa a Aceptada.
 - Si la propuesta es rechazada: pasa a Rechazada.
- Estado: Aceptada

Se notifica al docente la aprobación definitiva y se incorpora la propuesta en el sistema curricular.

Transiciones:

 - entry / Notificar aprobación
 - do / Publicar propuesta en el sistema
 - Pasa al estado final como aceptación exitosa del proceso.
- Estado: Rechazada

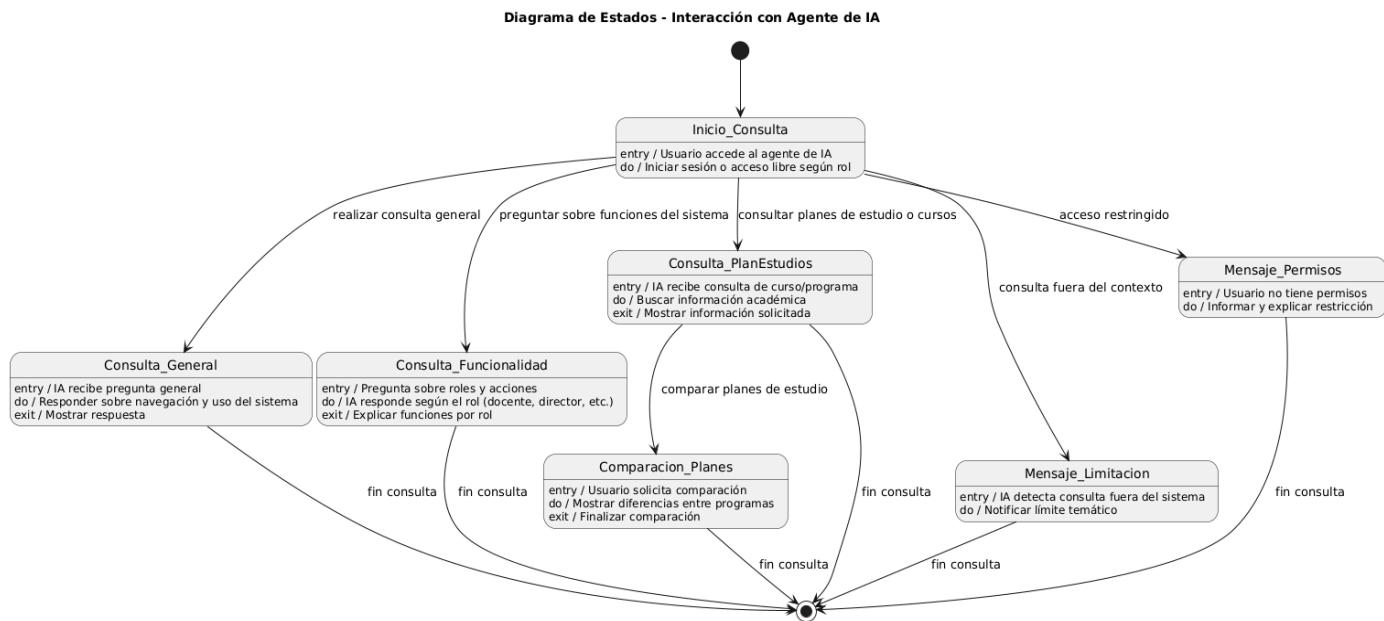
La propuesta fue rechazada en alguna de las etapas anteriores. Se notifica al docente y el proceso se cierra.

Transiciones:

 - entry / Notificar rechazo

- do / Archivar propuesta
 - Pasa al estado final como cierre del proceso por rechazo.
- Estado Final
Representado por el círculo negro con borde blanco.
Indica el fin del proceso ya sea porque la propuesta fue aceptada e incorporada, o rechazada y archivada.

6.1.0.2.6. Interacción con Chat IA



1. **Inicio del Proceso**
 - a. Representado por el círculo negro sólido.
2. **Estado: Inicio_Consulta**
 - a. Descripción: El usuario accede al agente de IA dentro del sistema.
 - b. Transiciones:
 - i. Entry: Usuario accede al agente.
 - ii. Do: Validar acceso (autenticado o no).
 - iii. Exit: Elegir tipo de consulta.
3. **Estado: Consulta_General**
 - a. Descripción: El usuario realiza preguntas generales sobre el uso del sistema.
 - b. Transiciones:
 - i. Entry: IA recibe pregunta.

- ii. Do: Responder sobre navegación y uso.
 - iii. Exit: Mostrar respuesta.
 - iv. Fin de la consulta.
- 4. **Estado: Consulta_Funcionalidad**
 - a. Descripción: El usuario pregunta sobre las funciones permitidas según su rol.
 - b. Transiciones:
 - i. Entry: Pregunta sobre roles y permisos.
 - ii. Do: IA describe funciones según el rol.
 - iii. Exit: Mostrar información.
 - iv. Fin de la consulta.
- 5. **Estado: Consulta_PlanEstudios**
 - a. Descripción: El usuario consulta información de cursos o planes académicos.
 - b. Transiciones:
 - i. Entry: Pregunta sobre curso o programa.
 - ii. Do: IA recupera datos académicos.
 - iii. Exit: Mostrar información solicitada.
 - iv. Puede continuar a Comparacion_Planes o finalizar.
- 6. **Estado: Comparacion_Planes**
 - a. Descripción: El usuario solicita comparar dos o más planes académicos.
 - b. Transiciones:
 - i. Entry: Solicitud de comparación.
 - ii. Do: Mostrar diferencias entre programas.
 - iii. Exit: Mostrar resultado.
 - iv. Fin de la consulta.
- 7. **Estado: Mensaje_Limitacion**
 - a. Descripción: El usuario realiza una consulta fuera del alcance del sistema.
 - b. Transiciones:
 - i. Entry: IA detecta una consulta no válida.
 - ii. Do: Mostrar mensaje indicando limitación temática.
 - iii. Fin de la consulta.
- 8. **Estado: Mensaje_Permisos**
 - a. Descripción: El usuario intenta acceder a información restringida.
 - b. Transiciones:
 - i. Entry: IA detecta falta de permisos.
 - ii. Do: Informar al usuario y explicar la restricción.
 - iii. Fin de la consulta.
- 9. **Estado Final**
 - a. Representado por el círculo negro con borde blanco.
 - b. Indica el fin de la interacción con el agente de IA.

6.1.0.3. Diagramas De Secuencia

PDF

Diagrama de secuencia.pdf

6.1.1. Diagramas De Estructura

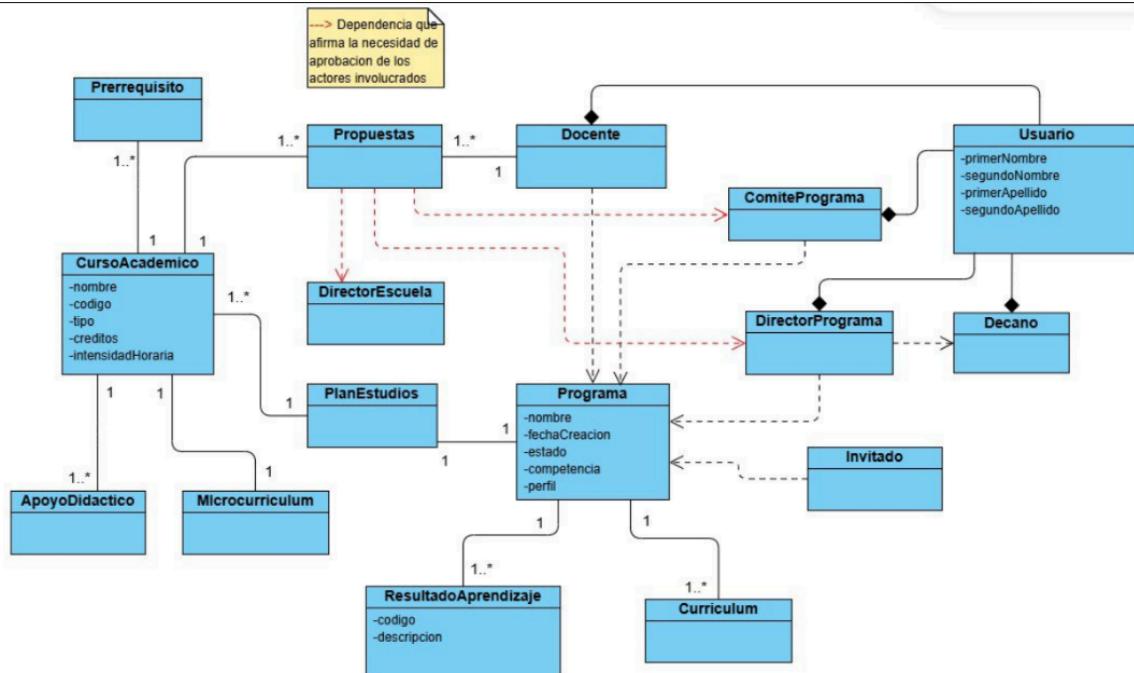
6.1.1.0. Diagrama De Clases

Por fines prácticos el diagrama de clases no se puede mostrar en el manual técnico debido a que sería ilegible gracias a su gran tamaño, para visualizarlo por favor ir al enlace a continuación:

PDF

Diagrama de clases final.pdf

6.1.1.1. Diagrama De Dominio



Entidades principales:

1. Usuario

- Atributos: primerNombre, segundoNombre, primerApellido, segundoApellido.
- Representa a las personas involucradas, como docentes, directores, decanos, invitados, etc.

2. Docente

- Se relaciona 1 a 1 con Usuario.
- Puede proponer propuestas (1..* relación con Propuestas).

3. Propuestas

Documento o iniciativa que requiere revisión/aprobación por varias autoridades.

Relación con:

- · CursoAcademico (1..*): cada propuesta debe tener al menos un curso.
- · Docente (1..*): es quien las genera.
- · DirectorEscuela, ComitéPrograma, DirectorPrograma: líneas punteadas rojas indican necesidad de aprobación (según la nota amarilla).

4. CursoAcademico

- · Atributos: nombre, codigo, tipo, creditos, intensidadHoraria.
- · Requiere:
- · ApoyoDidactico, Microcurriculum (materiales de apoyo y planificación).
- · Prerrequisito (opcional, 1..*).

5. PlanEstudios

- · Es el conjunto de cursos (CursoAcademico), se asocia con el Programa.

6. Programa

- · Atributos: nombre, fechaCreacion, estado, competencia, perfil.
- · Es aprobado o gestionado por:
- · DirectorPrograma
- · ComitePrograma
- · Invitado (relación opcional).
- · PlanEstudios
- · Curriculum y ResultadoAprendizaje.

7. ResultadoAprendizaje y Curriculum

- · Se relacionan directamente con Programa (1..*).
- · Roles de Aprobación (dependencias con líneas punteadas rojas)

El diagrama indica mediante flechas rojas que las siguientes entidades deben aprobar propuestas:

- · DirectorEscuela
- · ComitéPrograma
- · DirectorPrograma

Esto se aclara con la leyenda: “Dependencia que afirma la necesidad de aprobación de los actores involucrados”.

Relaciones importantes

- · Cada CursoAcademico puede tener múltiples Prerrequisitos y estar en múltiples propuestas.
- · Un PlanEstudios puede tener muchos cursos, pero pertenece a un único Programa.
- · Un Programa puede tener múltiples ResultadosAprendizaje y Curriculums.

6.2. Mockups

6.2.0. Plan de estudios

[— Volver](#)

Plan de Estudios de: Ingeniería de Sistemas

Semestre 1

Semestre 2

Semestre 3

COD

nombre_curso

6.2.1. Editar Cursos

Editar Cursos del Programa

Cursos del Programa:

Nombre	Tipo	Créditos	Ciclo	Área	Requisitos

[Guardar Cambios](#)

[Cerrar](#)

6.2.2. Editar plantel

Editor de Plantel

ID	Nombre	Correo	Roles
ejemplo	ejemplo	ejemplo	ejemplo
ejemplo	ejemplo	ejemplo	ejemplo
ejemplo	ejemplo	ejemplo	ejemplo

Rol:

6.2.3.

Login



BIENVENIDO

GRACIAS POR USAR NUESTRO SOFTWARE, PON TUS CREDENCIALES EN LA PARTE DE ABAJO

Contraseña



6.2.4.

Register

REGISTRO DE USUARIO

ejemplo@correo.com

Contraseña

Confirmar contraseña

Registrarse

Volver



6.2.5.

MainScreen

Gestión Curricular

boton_rol

Cerrar sesión

nombre_programa

Ver información

Ver Cursos del Programa

nombre_programa

Ver información

Ver Cursos del Programa

Anterior

Página 1 de 2

Siguiente

6.2.6.

Cursos asignados(docente)

Mis Cursos Asignados

INTRODUCCIÓN A LA INGENIERÍA DE SISTEMAS
Código: 603102
Proponer Micro-Curriculum

CÁLCULO DIFERENCIAL
Código: 603103
Proponer Micro-Curriculum

PROCESOS COMUNICATIVOS
Código: 603105
Proponer Micro-Curriculum

Cerrar

6.2.7. Propuestas

Mis Propuestas de Microcurrículo

Título: Propuesta de microcurrículo para CÁLCULO DIFERENCIAL
Estado: ACEPTADA
Observaciones:
DIRECTOR DE PROGRAMA: ACEPTAR
COMITE DE PROGRAMA: ACEPTAR - Comite aprueba
DIRECTOR DE PROGRAMA: FIRMA OK - ok
DIRECTOR DE ESCUELA: FIRMA OK
[Ver Archivo](#)

Título: Propuesta de microcurrículo para FÍSICA MECÁNICA
Estado: ACEPTADA
Observaciones:
DIRECTOR DE PROGRAMA: ACEPTAR
COMITE DE PROGRAMA: ACEPTAR
DIRECTOR DE ESCUELA: FIRMA OK
DIRECTOR DE PROGRAMA: FIRMA OK

Cerrar

6.2.8. Gestionar programa

Gestionar Programa: Programa test

Actualizar plan de estudios

Editar Cursos del Programa

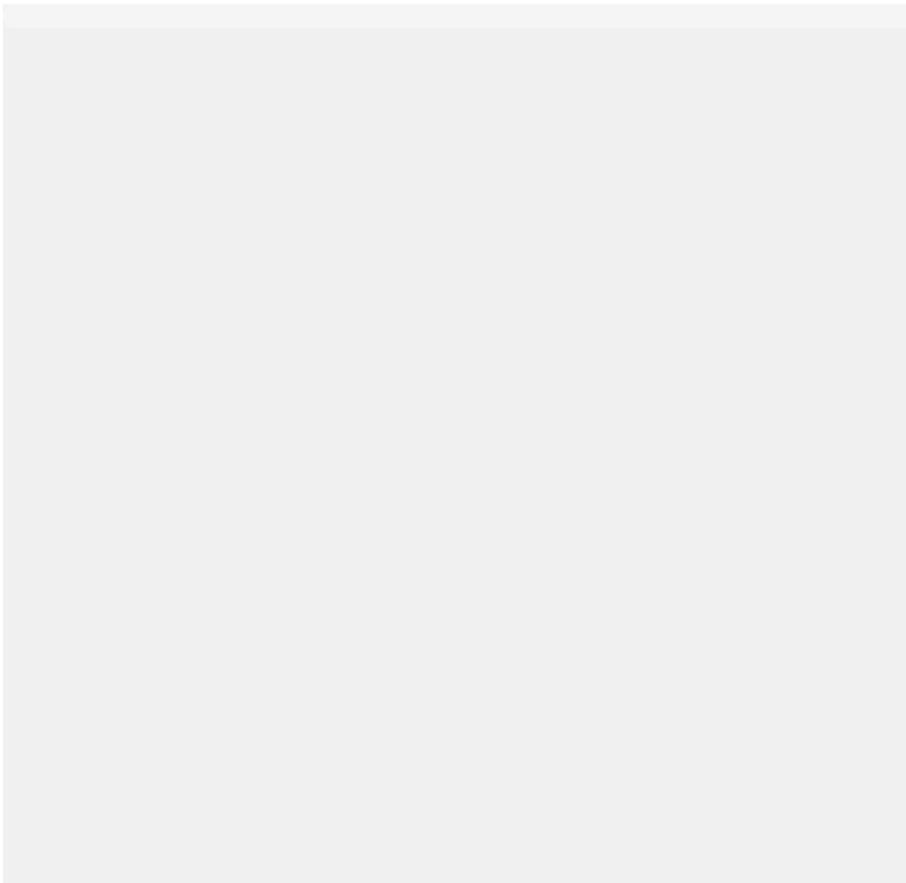
Subir Curriculums

Subir Resultados de Aprendizaje del Programa

Cerrar

6.2.9. Chat con agenteIA

Chat con Asistente IA



Escribe tu mensaje...

Enviar

Cerrar

6.3. Modelo de datos

6.3.0. Modelo Entidad Relación

Por fines prácticos el modelo entidad relación no se puede mostrar en el manual técnico debido a que sería ilegible gracias a su gran tamaño, para visualizarlo por favor ir al enlace a continuación:



MER final.pdf

7. Modelo De Negocios

Por fines prácticos el modelo de negocios no se puede mostrar en el manual técnico debido a que sería ilegible gracias a su gran tamaño, para visualizarlo por favor ir al enlace a continuación:



Lucidchart

8. Documentación de Pruebas



Documentacion_de_pruebas...

9. Conclusiones

- El sistema propuesto logra automatizar y centralizar la gestión de planes de estudio, microcurrículos y guías de laboratorio, lo cual mejora la eficiencia administrativa y garantiza la trazabilidad y validez de las decisiones académicas en la Universidad de los Llanos.
- El diseño del sistema contempla una arquitectura con múltiples roles (decano, directores, docentes, comité, invitados), cada uno con funciones y permisos específicos, lo que promueve un flujo de trabajo estructurado y seguro para la validación de contenidos curriculares.

- Una de las características destacadas es la capacidad de visualizar gráficamente el plan de estudios y su evolución, así como la gestión del historial de versiones y firmas digitales, permitiendo transparencia y seguimiento en los procesos curriculares.
- Aunque inicialmente es una aplicación de escritorio en Java con Spring Boot, el documento considera futuras expansiones hacia plataformas web o móviles, además de la integración con sistemas académicos existentes, lo que evidencia una visión escalable y sostenible del proyecto.