

Especificación de Requisitos según el estándar de IEEE 830

IEEE Std. 830-1998

26 de Mayo de 2025

Resumen

El presente documento detalla la Especificación de Requisitos del Software (ERS) para el desarrollo del Sistema de Gestión Curricular (GECULL), destinado a la Facultad de Ciencias Básicas e Ingeniería de la Universidad de los Llanos. El propósito es definir los requisitos funcionales y no funcionales del sistema, el cual será una aplicación de escritorio construida en Java, con interfaz gráfica en Java Swing y una base de datos PostgreSQL gestionada mediante Supabase.

El documento está estructurado en cuatro secciones: Introducción, Descripción General, Requisitos Específicos y Apéndices. En la descripción general se presentan los usuarios del sistema (estudiantes, docentes y administradores), sus funciones y las restricciones técnicas, incluyendo la necesidad de conexión a internet y compatibilidad con sistemas Windows.

Los requisitos específicos abordan funciones como la gestión del pensum, prerrequisitos, reportes académicos, y asignación de roles, además de criterios de rendimiento, interfaces externas, seguridad, usabilidad, mantenibilidad y escalabilidad. Finalmente, los apéndices incluyen formatos de entrada/salida, restricciones de programación y consideraciones adicionales sobre el entorno técnico y de desarrollo.

Índice

1. Introducción	3
1.1. Propósito	3
1.2. Ámbito del Sistema	3
1.3. Definiciones, Acrónimos y Abreviaturas	3
1.4. Referencias	4
1.5. Visión General del Documento	4
2. Descripción General	4
2.1. Perspectiva del Producto	4
2.2. Funciones del Producto	4
2.3. Características de los Usuarios	5
2.4. Restricciones	5
2.5. Suposiciones y Dependencias	5
2.6. Requisitos Futuros	5
3. Requisitos Específicos	6
3.1. Interfaces Externas	6
3.2. Funciones	6
Las funciones se encuentran organizadas por tipo de usuario:	6
3.3. Requisitos de Rendimiento	6
3.4. Restricciones de Diseño	7
3.5. Atributos del Sistema	7
3.6. Otros Requisitos	7
4. Apéndices	8
4.1. Formatos de Entrada y Salida	8
4.2. Restricciones del Lenguaje de Programación	8
4.3. Otros Datos Relevantes	8
4.4. Historias de usuario	8

1. Introducción

Esta sección proporciona una visión general del documento de Especificación de Requisitos del Software (ERS) y describe el propósito, el alcance, las definiciones utilizadas, las referencias empleadas y la estructura del documento.

1.1. Propósito

Este documento tiene como propósito establecer de forma clara y precisa los requisitos funcionales y no funcionales del Sistema de Gestión Curricular (GECULL), el cual será utilizado en la Facultad de Ciencias Básicas e Ingeniería de la Universidad de los Llanos. Está dirigido a los interesados del proyecto: equipo de desarrollo, analistas, usuarios finales, docentes y personal administrativo.

1.2. Ámbito del Sistema

El GECULL permitirá la gestión completa del pensum académico: registro, modificación y consulta de asignaturas, manejo de prerrequisitos, visualización de mallas curriculares y generación de reportes académicos. No contempla funciones administrativas relacionadas con matrículas, pagos o notas definitivas.

Beneficios esperados:

- Centralización y trazabilidad del contenido curricular.
- Mejora en la toma de decisiones académicas.
- Interfaz sencilla para consulta de estudiantes y docentes.

1.3. Definiciones, Acrónimos y Abreviaturas

- **GECULL:** Sistema de Gestión Curricular.
- **Pensum:** Conjunto estructurado de asignaturas de un programa.
- **CRUD:** Crear, Leer, Actualizar y Eliminar.
- **ERS:** Especificación de Requisitos del Software.
- **Usuario final:** Persona que interactúa directamente con el sistema.
- **Supabase:** Plataforma Backend-as-a-Service basada en PostgreSQL utilizada para la gestión de la base de datos.

1.4. Referencias

- IEEE Std 830-1998, "IEEE Recommended Practice for Software Requirements Specifications"
- Documento: "Análisis de requerimientos " (Universidad de los Llanos)

1.5. Visión General del Documento

Este documento está organizado en tres grandes secciones: Introducción, Descripción General y Requisitos Específicos. La primera contextualiza el sistema y su propósito; la segunda describe el entorno, usuarios y dependencias; la tercera presenta los requisitos detallados que el sistema debe cumplir.

2. Descripción General

Esta sección proporciona una descripción del entorno y contexto del sistema. Se identifican factores técnicos y humanos que afectan a los requisitos, lo que permite una mejor comprensión para su posterior especificación detallada.

2.1. Perspectiva del Producto

El GECULL es una aplicación de escritorio independiente, aunque se podría integrar en el futuro con el sistema institucional de información académica. Utiliza Supabase como plataforma para la gestión de datos en PostgreSQL. No requiere otros componentes externos para su funcionamiento principal.

2.2. Funciones del Producto

- Gestionar pensum: crear, modificar, eliminar y consultar asignaturas.
- Definir y gestionar prerrequisitos.
- Generar reportes de avance curricular.
- Consultar mallas curriculares por parte de los estudiantes.
- Asignación de roles (estudiante, docente, administrador).

2.3. Características de los Usuarios

- **Estudiantes:** Usuarios con conocimientos básicos en informática. Su interacción se enfoca en la consulta de información curricular.
- **Docentes:** Con experiencia en herramientas digitales, pueden actualizar y gestionar asignaturas.
- **Administradores:** Personal con formación técnica o administrativa, encargado de la configuración general y gestión completa del sistema.

2.4. Restricciones

- El sistema debe ser una aplicación de escritorio.
- Compatible con sistemas operativos Windows (preferiblemente desde Windows 10 en adelante).
- Lenguaje de programación: Java.
- Interfaz gráfica desarrollada con Java Swing.
- Base de datos gestionada a través de Supabase (PostgreSQL en la nube).
- Se requiere conexión a internet para acceso a la base de datos.

2.5. Suposiciones y Dependencias

- El sistema operará en computadores institucionales con sistema operativo Windows.
- Los usuarios tendrán acceso a las instalaciones físicas donde esté desplegada la aplicación.
- Se cuenta con acceso estable a Internet para la comunicación con la base de datos Supabase.

2.6. Requisitos Futuros

- Sincronización de datos entre múltiples dispositivos a través de red local o nube.
- Generación de respaldos automáticos desde Supabase.
- Exportación de datos en formatos abiertos (CSV, JSON).

3. Requisitos Específicos

Esta sección contiene los requisitos detallados que permitirán al equipo de diseño implementar el sistema y a los evaluadores verificar su cumplimiento. Los requisitos son claros, completos, verificables y trazables.

3.1. Interfaces Externas

- **Interfaz de Usuario:** Interfaz gráfica de escritorio desarrollada con Java Swing, intuitiva y organizada.
- **Interfaz de Software:** Módulos independientes que se comunican mediante controladores internos.
- **Interfaz de Hardware:** Compatible con computadoras personales con especificaciones básicas.
- **Interfaz con la base de datos:** Conexión mediante API REST y bibliotecas cliente provistas por Supabase.

3.2. Funciones

Las funciones se encuentran organizadas por tipo de usuario:

- **Estudiantes:**
 - Consultar pensum.
 - Consultar historial del pensum.
- **Docentes:**
 - Consultar pensum.
 - Consultar historial del pensum.
 - Proponer cambios curriculares.
- **Administradores:**
 - Crear/modificar/eliminar asignaturas.
 - Establecer prerrequisitos.
 - Asignar roles.
 - Generar reportes globales.

3.3. Requisitos de Rendimiento

- Tiempo de respuesta: < 3 segundos por acción.
- Soporte para múltiples sesiones abiertas en diferentes estaciones.

3 REQUISITOS ESPECÍFICOS

7

- Capacidad de la base de datos para almacenar > 10,000 registros de asignaturas.
- Latencia máxima permitida para conexión a Supabase: 500 ms.

3.4. Restricciones de Diseño

- Uso de arquitectura modular orientada a objetos.
- Interfaz de usuario desarrollada en Java Swing.
- Integración directa con Supabase para la gestión de datos.
- Cumplimiento de estándares de accesibilidad y ergonomía.

3.5. Atributos del Sistema

- **Fiabilidad:** 99.5% disponibilidad en jornadas laborales.
- **Seguridad:** Autenticación basada en roles, manejo seguro de contraseñas y validación de sesiones en Supabase.
- **Usabilidad:** Diseño intuitivo, accesible para todos los perfiles.
- **Mantenibilidad:** Código modular y documentado.
- **Escalabilidad:** Preparado para agregar módulos sin rediseño profundo.

3.6. Otros Requisitos

- Registro de logs de acceso y actividad.
- Manuales de usuario diferenciados por rol.
- Soporte multilenguaje (inicialmente en español).

4. Apéndices

4.1. Formatos de Entrada y Salida

- Formularios de registros de asignaturas, prerequisites y planes de estudio.
- Reportes en pantalla de avance académico y estructuras curriculares.
- Exportación de información formatos PDF y CSV.

4.2. Restricciones del Lenguaje de Programación

- El sistema debe ser desarrollado exclusivamente en Java.
- La interfaz de usuario debe utilizar Java Swing.
- No se permite uso de frameworks web o tecnologías web.

4.3. Otros Datos Relevantes

- La base de datos se gestiona completamente a través de Supabase, plataforma basada en PostgreSQL.
- El diseño de la base de datos contempla integridad referencial entre asignaturas y prerequisites.
- El desarrollo y pruebas se realizarán en entornos Windows, con Java 17 o superior.

4.4. Historias de usuario

Código: HU-01	Usuario: Decano
Título: Validación de decisiones curriculares.	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: Como decano, quiero poder consultar propuestas curriculares generadas por los directores de programa o el comité académico, evaluarlas en detalle y decidir su aprobación o rechazo, de manera que las decisiones se registren automáticamente en el sistema y los cambios curriculares aprobados se actualicen en el pensum y cursos correspondientes. Esto garantizará la trazabilidad y control académico de las decisiones tomadas.	
Criterios de aceptación: <ul style="list-style-type: none"> - Visualización de propuestas: <ul style="list-style-type: none"> • El decano debe tener acceso a un módulo donde se listan todas las propuestas curriculares pendientes de aprobación. • Cada propuesta debe incluir: nombre de la propuesta, fecha de creación, descripción, objetivos del cambio, impacto en el pensum, y usuario que la creó. - Decisión sobre propuestas: <ul style="list-style-type: none"> • El decano debe poder aprobar o rechazar una propuesta mediante botones específicos ("Aprobar" o "Rechazar"). 	

<ul style="list-style-type: none"> Al tomar una decisión, el sistema debe requerir un comentario obligatorio justificando la decisión.
- Registro de decisiones: <ul style="list-style-type: none"> El sistema debe guardar: ID de la propuesta, decisión tomada (aprobada o rechazada), comentario del decano, fecha y hora de la acción, y usuario responsable. Cada registro debe ser accesible para auditoría posterior.
- Actualización de cambios: <ul style="list-style-type: none"> Si la propuesta es aprobada, el sistema debe actualizar automáticamente los elementos curriculares afectados (por ejemplo, actualizar el pensum, añadir/eliminar cursos, modificar contenidos).
- Notificaciones: <ul style="list-style-type: none"> Una vez que el decano toma una decisión, se debe enviar automáticamente una notificación a los usuarios involucrados (director de programa, comité académico, etc.) informando el resultado.

Código: HU-02	Usuario: Decano
Título: Designación de directores de programa.	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: Como decano, quiero asignar, modificar o revocar el rol de director de programa a diferentes profesores registrados en el sistema, de modo que pueda asegurar una correcta gestión académica de cada programa. Cada designación o cambio debe quedar registrado con trazabilidad, y los usuarios afectados deben recibir notificaciones claras y oportunas.	
Criterios de aceptación: <ul style="list-style-type: none"> - Listado de posibles directores: <ul style="list-style-type: none"> El sistema debe permitir al decano ver una lista de todos los profesores activos disponibles para asignación (nombre completo, correo electrónico institucional, programa actual, rol actual). - Asignación y modificación de rol: <ul style="list-style-type: none"> El decano debe poder seleccionar un profesor y asignarle el rol de "Director de Programa" a un programa específico. Si ya existe un director asignado, el sistema debe solicitar confirmación explícita para reemplazar al actual. La acción de asignación o modificación debe requerir una justificación que quede registrada. - Registro de cambios: <ul style="list-style-type: none"> Cada acción de asignación, modificación o revocación debe ser registrada incluyendo: usuario afectado, rol anterior, rol nuevo, programa asociado, fecha y hora de cambio, usuario responsable de la acción. - Notificaciones automáticas: <ul style="list-style-type: none"> Una vez asignado o modificado un rol, el profesor debe recibir una notificación por correo electrónico informándole del cambio de rol, incluyendo: título del nuevo rol, programa asignado y observaciones si las hay. - Historial de roles: <ul style="list-style-type: none"> El sistema debe mantener un historial visible de todos los cambios de rol realizados para cada profesor, accesible solo para usuarios con permisos administrativos. 	

Código: HU-03	Usuario: Director de programa
Título: Generación y actualización del pensum	
Prioridad: Alta	Tipo: Funcionalidad
<p>Descripción: Como director de programa, quiero poder generar nuevos pensums o actualizar versiones existentes en función de las decisiones curriculares aprobadas, asegurando que cada modificación sea registrada, versionada y visible para los usuarios autorizados, manteniendo un historial claro de los cambios realizados.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Generación de pensum: <ul style="list-style-type: none"> • El sistema debe permitir crear un nuevo pensum desde cero o duplicar un pensum existente para modificaciones. • El director debe ingresar: nombre del pensum, año de vigencia, descripción general, listado de cursos asociados. - Actualización de pensum: <ul style="list-style-type: none"> • Al actualizar un pensum, el sistema debe requerir la selección de la versión a modificar y la descripción del cambio realizado. • Cada actualización debe generar automáticamente una nueva versión numerada (v1, v2, etc.), preservando las versiones anteriores para consulta. - Control de cambios: <ul style="list-style-type: none"> • Cada cambio (creación, edición, eliminación de un curso) debe quedar registrado con fecha, usuario responsable y tipo de cambio. - Acceso a versiones anteriores: <ul style="list-style-type: none"> • Los usuarios autorizados deben poder consultar todas las versiones históricas de un pensum. - Notificaciones: <ul style="list-style-type: none"> • El sistema debe notificar automáticamente a los miembros del comité académico y otros usuarios autorizados cuando un nuevo pensum o una actualización sea publicada. 	

Código: HU-04	Usuario: Director de escuela
Título: Supervisión general del programa	
Prioridad: Alta	Tipo: Funcionalidad
<p>Descripción: Como director de escuela, quiero supervisar el estado académico general de los programas bajo mi responsabilidad, pudiendo revisar reportes periódicos, analizar métricas de rendimiento y tomar decisiones informadas sobre ajustes necesarios en los programas.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Acceso a reportes: <ul style="list-style-type: none"> • El director debe poder acceder a informes de: tasa de aprobación por curso, asignaturas críticas, número de titulados, y rendimiento académico general. - Consulta y análisis de información: <ul style="list-style-type: none"> • Cada reporte debe incluir opciones de filtrado por semestre, programa académico, curso o cohorte. - Toma de decisiones: <ul style="list-style-type: none"> • El director debe poder proponer cambios curriculares o administrativos directamente desde la plataforma a partir de los reportes. - Registro y seguimiento: 	

<ul style="list-style-type: none"> Toda decisión tomada debe quedar registrada, indicando el reporte que motivó la decisión, la fecha, y el responsable.
- Notificaciones: <ul style="list-style-type: none"> Al registrar una propuesta de cambio, el sistema debe notificar automáticamente a los directores de programa afectados.

Código: HU-05	Usuario: Director de programa
Título: Supervisión del plan de estudios	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: Como director de programa, quiero acceder a una sección donde pueda visualizar todas las solicitudes de cambios realizadas sobre el plan de estudios de mi programa, junto con la información detallada de cada modificación propuesta, para revisarlas, aprobarlas o rechazarlas según su impacto académico. Quiero que todas mis decisiones sean registradas en el historial del sistema, y que las modificaciones aprobadas actualicen automáticamente la versión vigente del plan de estudios. Además, quiero que los usuarios interesados (profesores y estudiantes) sean notificados de los cambios relevantes en el plan.	
Criterios de aceptación: <ul style="list-style-type: none"> - Consulta de modificaciones: <ul style="list-style-type: none"> El director debe poder ver un listado de todas las solicitudes de modificación pendientes, aprobadas y rechazadas. - Aprobación/Rechazo de cambios: <ul style="list-style-type: none"> El director debe tener la opción de aprobar o rechazar cada modificación, proporcionando un comentario justificativo obligatorio. - Registro de cambios: <ul style="list-style-type: none"> Cada modificación debe ser registrada con estado (pendiente, aprobado, rechazado), fecha de acción, comentario de decisión y usuario responsable. - Notificaciones: <ul style="list-style-type: none"> El sistema debe enviar notificaciones a los usuarios que propusieron los cambios indicando la decisión tomada. 	

Código: HU-06	Usuario: Director de programa
Título: Gestión de información de cursos	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: Como director de programa, quiero cargar la información detallada de cada curso nuevo o actualizado a través de un formulario estructurado, para asegurar que los estudiantes y profesores tengan acceso a contenidos precisos y actualizados. Esta información debe incluir objetivos de aprendizaje, descripción del curso, contenidos temáticos, bibliografía básica, criterios de evaluación y guías de laboratorio si aplica. Además, toda carga o modificación debe quedar registrada con trazabilidad.	
Criterios de aceptación: <ul style="list-style-type: none"> - Formulario de carga: <ul style="list-style-type: none"> El formulario de curso debe incluir los siguientes campos obligatorios: <ul style="list-style-type: none"> Código del curso Nombre del curso Objetivos de aprendizaje 	

<ul style="list-style-type: none"> ○ Descripción general ○ Contenido temático ○ Bibliografía obligatoria y complementaria ○ Perfil de ingreso y egreso relacionado ○ Guías de laboratorio (archivo adjunto opcional) ○ Créditos académicos ○ Duración en semanas
<ul style="list-style-type: none"> - Validaciones del formulario: <ul style="list-style-type: none"> • Todos los campos obligatorios deben ser validados antes de permitir el envío. - Actualización de información: <ul style="list-style-type: none"> • El sistema debe permitir actualizar cualquier campo previamente cargado, generando una nueva versión del curso. - Historial de cambios: <ul style="list-style-type: none"> • Cada actualización debe quedar registrada con fecha, usuario responsable y descripción del cambio. - Visualización de información: <ul style="list-style-type: none"> • Estudiantes y profesores deben poder visualizar siempre la última versión disponible del curso.

Código: HU-07	Usuario: Director de programa
Título: Revisión de propuestas de cambio	
Prioridad: Alta	Tipo: Funcionalidad
<p>Descripción: Como director de programa, quiero recibir automáticamente todas las propuestas de cambios enviadas por profesores o comités (como modificación de contenidos, cambios de créditos, eliminación de cursos) en una bandeja de entrada clasificada, para evaluarlas detalladamente, aprobarlas, rechazarlas o reenviarlas al comité de programa, garantizando un flujo de aprobación organizado y transparente.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Recepción de propuestas: <ul style="list-style-type: none"> • El sistema debe almacenar todas las propuestas en una bandeja de entrada para el director. • Cada propuesta debe mostrar: código del curso, nombre del curso, descripción, créditos, semestre y fecha. - Acciones disponibles: <ul style="list-style-type: none"> • El director debe poder: <ul style="list-style-type: none"> ○ Aprobar propuesta ○ Rechazar propuesta ○ Reenviar propuesta al comité de programa - Registro de acciones: <ul style="list-style-type: none"> • Cada acción tomada debe quedar registrada incluyendo ID de propuesta, acción realizada, comentario obligatorio, fecha y usuario responsable. - Seguimiento y estado: <ul style="list-style-type: none"> • El estado de cada propuesta debe cambiar dinámicamente según la acción realizada ("En revisión", "Aprobada", "Rechazada", "Reenviada"). - Notificaciones: <ul style="list-style-type: none"> • El usuario que propuso el cambio debe ser notificado automáticamente del estado actualizado de su propuesta. 	

Código: HU-08	Usuario: Director de programa
Título: Designación de roles auxiliares.	
Prioridad: Alta	Tipo: Funcionalidad
<p>Descripción: Como director de programa, quiero seleccionar y asignar roles auxiliares, como secretario/a o miembros del comité, directamente desde el sistema, definiendo claramente sus funciones y permisos, para garantizar una correcta delegación de tareas académicas y administrativas. Quiero que toda asignación quede registrada para control administrativo.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Listado de posibles auxiliares: <ul style="list-style-type: none"> • El sistema debe mostrar usuarios elegibles basados en roles previos o criterios administrativos. - Asignación de roles: <ul style="list-style-type: none"> • El director debe seleccionar un auxiliar, asignarle un rol específico y definir las tareas permitidas. - Registro de asignaciones: <ul style="list-style-type: none"> • Cada asignación debe registrarse con usuario asignado, rol, fecha de inicio y fecha de finalización (si aplica). - Notificaciones: <ul style="list-style-type: none"> • El sistema debe notificar automáticamente al usuario designado sobre su nuevo rol. 	

Código: HU-09	Usuario: Secretario del director de programa
Título: Apoyo administrativo y registro de cambios en cursos.	
Prioridad: Alta	Tipo: Funcionalidad e historia
<p>Descripción: Como secretario del director de programa, quiero poder realizar tareas de apoyo, como cargar o modificar la información de cursos y registrar cambios administrativos en el sistema, siguiendo las autorizaciones recibidas, para mantener la información académica actualizada y con trazabilidad completa. Además, quiero que cada acción quede documentada para auditoría interna.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Acceso restringido: <ul style="list-style-type: none"> • El secretario solo puede acceder a la sección de carga/modificación si el director de programa le ha otorgado permisos explícitos. - Carga y modificación de cursos: <ul style="list-style-type: none"> • El secretario debe tener acceso al mismo formulario estructurado usado por el director (ver HU-06), incluyendo todos los campos de curso. • Al actualizar un curso, el sistema debe generar una nueva versión, manteniendo la versión anterior disponible para consultas internas. - Registro detallado de cambios: <ul style="list-style-type: none"> • El sistema debe registrar: <ul style="list-style-type: none"> ○ Curso afectado ○ Tipo de cambio (creación, edición, eliminación) ○ Descripción del cambio ○ Fecha y hora del cambio ○ Usuario responsable (secretario) - Notificaciones de cambios: 	

<ul style="list-style-type: none"> • Cada cambio realizado debe enviar una notificación automática al director de programa y a los profesores asignados al curso modificado.
- Auditoría: <ul style="list-style-type: none"> • El historial de cambios debe ser accesible para usuarios con permisos administrativos.

Código: HU-10	Usuario: Miembro del comité de programa
Título: Revisión del pensum	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: Como miembro del comité de programa, quiero acceder a las propuestas de modificación del pensum enviadas por el director de programa, revisarlas en detalle, emitir mi voto (aprobación o rechazo) y registrar mis observaciones, para asegurar que los cambios curriculares cumplen con los estándares académicos establecidos.	
Criterios de aceptación: <ul style="list-style-type: none"> - Listado de propuestas de modificación: <ul style="list-style-type: none"> • El sistema debe mostrar todas las modificaciones propuestas, incluyendo: nombre del pensum, cursos afectados, tipo de cambio, justificación del cambio, y fecha de propuesta. - Evaluación de propuestas: <ul style="list-style-type: none"> • Cada miembro del comité debe poder: <ul style="list-style-type: none"> ○ Aprobar ○ Rechazar ○ Solicitar cambios (comentario obligatorio) - Registro de votaciones: <ul style="list-style-type: none"> • Cada votación debe registrar: <ul style="list-style-type: none"> ○ Propuesta evaluada ○ Voto emitido (aprobado/rechazado/solicita cambios) ○ Comentario del votante ○ Fecha y hora del voto - Decisión final: <ul style="list-style-type: none"> • La propuesta se considera aprobada si al menos el 60% de los miembros la aprueban. • El sistema debe calcular y mostrar automáticamente el porcentaje de aprobación. - Notificaciones: <ul style="list-style-type: none"> • El director de programa debe ser notificado del resultado final y del porcentaje de aprobación. 	

Código: HU-11	Usuario: Profesor
Título: Consulta de información de cursos	
Prioridad: Media	Tipo: Funcionalidad
Descripción: Como profesor, quiero acceder a toda la información actualizada de los cursos que tengo asignados, incluyendo objetivos de aprendizaje, metodología, contenidos, criterios de evaluación y materiales de apoyo, para planificar mis clases y evaluaciones de manera adecuada.	

<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Acceso personalizado: <ul style="list-style-type: none"> • El profesor debe visualizar únicamente los cursos en los que esté asignado como responsable o co-responsable. - Contenido accesible: <ul style="list-style-type: none"> • Cada curso debe mostrar: <ul style="list-style-type: none"> ○ Nombre y código ○ Objetivos de aprendizaje ○ Contenidos temáticos ○ Metodología de enseñanza ○ Bibliografía básica y complementaria ○ Criterios de evaluación - Exportación de información: <ul style="list-style-type: none"> • El sistema debe permitir exportar el contenido del curso en formato PDF o Excel, seleccionable desde la vista del curso. - Notificaciones de actualización: <ul style="list-style-type: none"> • Cuando la información del curso cambie, el profesor debe recibir una notificación automática indicando qué apartado fue modificado.

Código: HU-12	Usuario: Profesor y/o director de escuela
Título: Envío de propuestas de cambio curricular	
Prioridad: Alta	Tipo: Funcionalidad
<p>Descripción: Como docente o miembro del comité académico, quiero poder registrar y enviar propuestas de cambio curricular desde el sistema (por ejemplo: modificación de contenido temático, número de créditos, creación o eliminación de asignaturas), para que sean revisadas por el director del programa, asegurando una comunicación formal y trazable del proceso de mejora académica.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> - Validaciones: <ul style="list-style-type: none"> • El sistema debe validar que todos los campos obligatorios estén completos antes del envío. • Debe impedir envíos duplicados del mismo usuario sobre el mismo curso hasta que se resuelva la propuesta previa. - Formulario de propuesta: <ul style="list-style-type: none"> • El sistema debe permitir crear una nueva propuesta de cambio desde un formulario estructurado: • El formulario debe incluir los siguientes campos obligatorios: <ul style="list-style-type: none"> ○ Código del curso. ○ Nombre del curso. ○ Créditos. ○ Semestre. ○ Descripción. - Confirmación de envío: <ul style="list-style-type: none"> • El usuario debe recibir un mensaje de confirmación con el código de la propuesta y el estado inicial: “Propuesta guardada exitosamente”. - Seguimiento: <ul style="list-style-type: none"> • El sistema debe permitir a los usuarios (con permisos de evaluar las propuestas enviadas) de ver las propuestas recibidas. 	

<ul style="list-style-type: none"> El historial de estados debe estar disponible cada propuesta: “Enviada” → “En revisión” → “Aprobada / Rechazada / Reenviada”.
- Notificaciones: <ul style="list-style-type: none"> El sistema debe enviar una notificación automática al director del programa cuando una propuesta es enviada. El usuario recibirá notificaciones cuando su propuesta cambie de estado.

Código: HU-13	Usuario: Estudiante
Título: Ingreso al sistema	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: Como estudiante, quiero poder ingresar al sistema utilizando mis credenciales institucionales (correo electrónico y contraseña), de forma segura y protegida contra intentos no autorizados, para acceder a mis datos académicos y administrativos.	
Criterios de aceptación: <ul style="list-style-type: none"> - Formulario de autenticación: <ul style="list-style-type: none"> El estudiante debe ingresar su correo electrónico institucional y contraseña en campos obligatorios. - Validaciones de seguridad: <ul style="list-style-type: none"> El sistema debe validar las credenciales contra una base de datos cifrada. Tras 5 intentos fallidos consecutivos, la cuenta debe ser bloqueada temporalmente (por 30 minutos) y se debe notificar por correo al estudiante. - Recuperación de contraseña: <ul style="list-style-type: none"> Debe existir una opción de "¿Olvidaste tu contraseña?" que permita recuperar el acceso mediante correo de recuperación. - Mensajes de error: <ul style="list-style-type: none"> En caso de error de autenticación, debe mostrarse un mensaje genérico: "Usuario o contraseña incorrectos" (sin revelar cuál fue incorrecto). 	

Código: HU-14	Usuario: Todos los usuarios registrados
Título: Consulta de pensum	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: Como usuario registrado en el sistema, quiero consultar el pensum oficial y actualizado de mi carrera, visualizando la lista de asignaturas organizadas por semestre, los prerrequisitos, la cantidad de créditos y el flujo académico recomendado, para planificar adecuadamente mi trayectoria educativa.	
Criterios de aceptación: <ul style="list-style-type: none"> - Visualización estructurada: <ul style="list-style-type: none"> El pensum debe estar organizado por semestres y mostrar: <ul style="list-style-type: none"> Nombre de la asignatura Código de la asignatura Créditos Prerrequisitos y corequisitos 	

<ul style="list-style-type: none"> ○ Tipología (obligatoria, electiva)
<ul style="list-style-type: none"> - Descarga del pensum: <ul style="list-style-type: none"> • El estudiante debe poder descargar el pensum en formato PDF actualizado. - Actualización automática: <ul style="list-style-type: none"> • El pensum mostrado debe actualizarse automáticamente si se aprueba una nueva versión del plan de estudios. - Historial de cambios (opcional en vista): <ul style="list-style-type: none"> • El estudiante puede consultar una sección que indique las últimas modificaciones realizadas al pensum (campo opcional).

Código: HU-15	Usuario: Todos los usuarios registrados
Título: Consulta del historial de pensum	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: Como usuario registrado en el sistema, quiero poder consultar las versiones históricas de los pensums de mi carrera, ordenadas cronológicamente, para entender cómo ha evolucionado el plan académico desde su creación y comparar versiones anteriores con la actual.	
Criterios de aceptación: <ul style="list-style-type: none"> - Visualización estructurada: <ul style="list-style-type: none"> • El pensum debe estar organizado por semestres y mostrar: <ul style="list-style-type: none"> ○ Nombre de la asignatura ○ Código de la asignatura ○ Créditos ○ Descripción - Actualización automática: <ul style="list-style-type: none"> • El pensum mostrado debe actualizarse automáticamente si se aprueba una nueva versión del plan de estudios. - Historial de cambios (opcional en vista): <ul style="list-style-type: none"> • El estudiante puede consultar una sección que indique las últimas modificaciones realizadas al pensum (campo opcional). 	