

Universidad de La Habana
Facultad de Matemática y Computación



Sistema digital para la gestión de asignaturas electivas en la Universidad de La Habana.

Autor:

Alex Samuel Bas Beovides

Tutores:

Lic. Carlos León González

Dra. C. Lucina García Hernández

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencia de la Computación

31 de mayo de 2025

<https://github.com/thesis-si/course-allocation/tree/main/Electia>

Resumen

El presente trabajo aborda el diseño y desarrollo de un sistema digital para la gestión de asignaturas electivas universitarias, respondiendo a la necesidad de una herramienta institucional que facilite la inscripción, aprobación y administración integral del proceso académico en la Universidad de La Habana. El sistema propuesto se centra en la creación de un entorno digital que permita a profesores, administradores y estudiantes gestionar el ciclo completo de las asignaturas electivas de manera eficiente, desde la propuesta docente hasta la matrícula estudiantil y el seguimiento académico. Se discuten los desafíos y limitaciones de los sistemas existentes de gestión académica, destacando la importancia de una solución adaptada al contexto universitario cubano que integre las particularidades organizativas de la Dirección de Formación de Pregrado.

El desarrollo del sistema se basa en una arquitectura de software robusta y escalable, utilizando tecnologías como ASP.NET Core, React.js, SQL Server y Docker para garantizar su sostenibilidad y rendimiento. En este trabajo se detallan los componentes principales del sistema, incluyendo el diseño de la base de datos relacional, la interfaz de usuario diferenciada por roles y los servicios de gestión de propuestas académicas. Se presentan los resultados de pruebas y validación experimental que demuestran la funcionalidad y eficiencia del sistema en la automatización de procesos administrativos y la generación de documentación académica oficial. Finalmente, se discuten las implicaciones de este trabajo para la comunidad académica universitaria y se proponen futuras líneas de desarrollo que permitan extender la funcionalidad a otros procesos de gestión educativa.

Abstract

This work addresses the design and development of a digital system for managing university elective courses, responding to the need for an institutional tool that facilitates enrollment, approval, and comprehensive administration of the academic process at the *Universidad de La Habana* (University of Havana). The proposed system focuses on the creation of a digital environment that enables professors, administrators, and students to efficiently manage the entire lifecycle of elective courses—from course proposal to student registration and academic monitoring. The challenges and limitations of existing academic management systems are discussed, highlighting the importance of a solution adapted to the Cuban university context and integrating the organizational particularities of the *Dirección de Formación de Pregrado* (Undergraduate Education Office).

The system's development is based on a robust and scalable software architecture, employing technologies such as ASP.NET Core, React.js, SQL Server, and Docker to ensure sustainability and performance. This work details the system's main components, including the design of the relational database, a role-based user interface, and services for managing academic proposals. Test results and experimental validation are presented, demonstrating the system's functionality and efficiency in automating administrative processes and generating official academic documentation. Finally, the implications of this work for the academic university community are discussed, and future lines of development are proposed to extend the system's functionality to other educational management processes

Índice general

Introducción	1
1. Marco teórico-conceptual	4
1.1. Ingeniería de software aplicada a la gestión de procesos académicos	4
1.1.1. Arquitectura de Software	5
1.1.2. DDD: Diseño Orientado al Dominio	6
1.1.3. Arquitectura por Capas	9
1.2. Sistemas de asignación de recursos basados en reglas	13
1.3. Estado del arte	14
2. Concepción y diseño de la solución	17
2.1. Problemática	17
2.2. Determinación de los Requerimientos	19
2.2.1. Requerimientos Funcionales	19
2.2.2. Requerimientos No Funcionales	20
2.2.3. Requerimientos Informacionales	20
2.2.4. Requerimientos de Entorno	22
2.3. Casos de uso y flujos principales	23
2.4. Diseño de la base de datos	27
3. Detalles de Implementación y Experimentación	30
3.1. Tecnologías y herramientas utilizadas para el desarrollo del sistema	30
3.1.1. Sistema de Gestión de Base de Datos: SQL Server	31
3.1.2. Framework de Desarrollo Backend: ASP.NET Core	31
3.1.3. Mapeo Objeto-Relacional: Entity Framework Core	32
3.1.4. Tecnología de Desarrollo Frontend: React.js	32
3.1.5. Plataforma de Contenedorización: Docker	32
3.2. Arquitectura del sistema implementado	33
3.3. Pruebas y validación experimental	33
Bibliografía	34

Índice de figuras

2.1.	Diagrama de casos de uso del Administrador.	24
2.2.	Diagrama de casos de uso del Profesor.	25
2.3.	Diagrama de casos de uso del Estudiante.	26
2.4.	Modelo Entidad-Relacional Extendido de la base de datos.	29

Introducción

La gestión eficiente de las asignaturas electivas en entornos universitarios constituye un desafío clave para garantizar la calidad del proceso formativo, la equidad al acceso de los cursos y el cumplimiento de los objetivos estratégicos de cada institución. Actualmente, la Dirección de Formación de Pregrado (DFP) requiere sistemas que no solo faciliten la inscripción y aprobación de asignaturas por parte del profesorado, sino que además aseguren una adecuada organización, control y seguimiento de la matrícula estudiantil en correspondencia a criterios académicos definidos. Esta necesidad se hace más crítica en contextos donde la oferta electiva es diversa y debe ser gestionada por roles específicos, como administradores, profesores y estudiantes, cada uno con distintos niveles de acceso y funciones.

A nivel internacional, los Sistemas de Gestión del Aprendizaje (LMS, por sus siglas en inglés), como Moodle y Canvas, son ampliamente utilizados para administrar cursos y contenidos educativos en línea. Estas plataformas ofrecen funcionalidades robustas en cuanto a la creación de cursos, la gestión de usuarios y el seguimiento del desempeño académico. Sin embargo, presentan limitaciones cuando se trata de adaptar sus funcionalidades a necesidades particulares de gestión institucional. Por ejemplo, muchos de estos sistemas requieren pagar suscripciones para acceder a módulos avanzados o a servicios personalizados, y no siempre permiten configurar reglas específicas para la asignación automatizada de cursos, tales como restricciones por centro docente, perfil académico o intereses personales. Estas carencias impiden su implementación efectiva en escenarios que demandan alta personalización y autonomía.

Ante las limitaciones identificadas en los sistemas actuales de inscripción y administración de asignaturas electivas, el propósito central de esta tesis es el **diseño y desarrollo de una plataforma digital que permita gestionar de forma integral el proceso de inscripción, aprobación, asignación y matrícula de asignaturas electivas en la Universidad de la Habana**. Este sistema busca dar respuesta a diversas problemáticas institucionales que dificultan la organización eficiente de estas ofertas docentes, entre las que destacan:

1. La inexistencia de una herramienta digital propia que responda a las particula-

ridades organizativas y normativas de la Dirección de Formación de Pregrado.

2. La dispersión del proceso de inscripción y validación de asignaturas electivas, que actualmente depende de métodos manuales o no integrados.
3. La dificultad para aplicar criterios diferenciados de acceso según carrera, año académico u otros parámetros establecidos por la institución.
4. La ausencia de mecanismos automáticos para el seguimiento de la matrícula, control de cupos, generación de actas y obtención de estadísticas útiles para la toma de decisiones.

Como solución, se propone el desarrollo de un sistema web de inscripción y gestión de asignaturas electivas, accesible para todos los actores del proceso educativo universitario, que garantice la transparencia, trazabilidad y eficiencia de cada etapa. Esta solución estará basada en una arquitectura modular y una base de datos relacional, permitiendo escalabilidad, control de acceso según rol, y adaptabilidad a futuros requerimientos institucionales. Para cumplir este propósito general, se plantean los siguientes objetivos específicos:

1. Diseñar una interfaz para que el profesorado proponga nuevas asignaturas electivas, incluyendo todos los datos requeridos por la DFP para su evaluación y aprobación.
2. Implementar un módulo de revisión y validación de asignaturas por parte del rol administrador, permitiendo definir criterios y filtros personalizables.
3. Desarrollar funcionalidades de inscripción estudiantil que respeten la capacidad máxima de cada curso y permitan aplicar reglas específicas por carrera, facultad y/o año académico.
4. Automatizar la asignación de estudiantes, generación de listados, actas académicas y reportes estadísticos, minimizando errores y tareas manuales.
5. Evaluar el desempeño del sistema mediante pruebas funcionales y de usabilidad, con el fin de asegurar su efectividad y aceptación por parte de los usuarios finales.

La presente investigación se organiza en cuatro capítulos, cada uno enfocado en una etapa clave del proceso de desarrollo del sistema propuesto:

- **Capítulo 1: Marco teórico-conceptual.** Se examinan los principios teóricos relacionados con la gestión académica de asignaturas electivas en entornos universitarios, así como la caracterización de plataformas similares, sus funcionalidades, beneficios y limitaciones, con especial atención a las necesidades del contexto cubano.
- **Capítulo 2: Concepción y diseño de la solución.** Se expone la propuesta del sistema informático, especificando los requerimientos funcionales y no funcionales, los roles de usuario, los principales casos de uso y la estructura de datos necesaria para su implementación.
- **Capítulo 3: : Implementación y experimentación.** Se describen los detalles del proceso de implementación, incluyendo las tecnologías utilizadas, la arquitectura del sistema, y los resultados obtenidos a partir de pruebas de funcionamiento, eficiencia y experiencia de usuario.

Finalmente, se incluye la bibliografía consultada, que respalda los contenidos desarrollados y las decisiones de diseño adoptadas a lo largo de la investigación.

Capítulo 1

Marco teórico-conceptual

El presente capítulo tiene como objetivo fundamentar teóricamente la problemática asociada a la gestión de asignaturas electivas en instituciones de educación superior, abordando conceptos claves relacionados con los sistemas de información académica, las arquitecturas de software aplicables y los criterios de diseño que sustentan el desarrollo de soluciones orientadas a mejorar estos procesos. Asimismo, se examinan modelos existentes de gestión educativa y su aplicabilidad en contextos con necesidades institucionales particulares, como es el sistema universitario cubano, particularmente en la Universidad de la Habana.

1.1. Ingeniería de software aplicada a la gestión de procesos académicos

La ingeniería de software constituye una disciplina fundamental orientada al análisis, diseño, implementación y mantenimiento de sistemas computacionales robustos y escalables. Su aplicación trasciende los límites tradicionales de la informática, extendiéndose hacia sectores críticos como la educación superior, donde la gestión eficiente de procesos académicos representa un pilar estratégico para el funcionamiento institucional.

En el contexto universitario contemporáneo, la administración de procesos formativos demanda soluciones tecnológicas que garanticen la integración, automatización y trazabilidad de las actividades académicas. La complejidad inherente a la gestión de asignaturas electivas, matrícula estudiantil, validación docente y generación de reportes institucionales requiere la aplicación de principios sólidos de ingeniería de software que aseguren la confiabilidad, mantenibilidad y escalabilidad de las soluciones desarrolladas.

El desarrollo sostenible de las instituciones educativas exige la implementación de

entornos digitales que faciliten la colaboración entre los diversos actores del proceso formativo, promoviendo la transparencia administrativa y optimizando la asignación de recursos académicos. En el ámbito de la gestión universitaria, la interacción efectiva entre administradores, docentes y estudiantes resulta esencial para el logro de los objetivos institucionales. Sin embargo, una de las principales limitaciones identificadas radica en la ausencia de sistemas integrados que permitan la gestión centralizada y automatizada de los procesos de inscripción y administración de asignaturas electivas.

Aunque existen diversas plataformas de gestión académica en el mercado, muchas de ellas presentan restricciones significativas en términos de personalización, costos de implementación y adaptabilidad a las particularidades normativas de cada institución. La mayoría de estos sistemas comerciales requieren inversiones considerables en licenciamiento y capacitación, además de no ofrecer la flexibilidad necesaria para configurar reglas específicas de asignación según criterios institucionales diferenciados, tales como restricciones por carrera, año académico o disponibilidad de cupos.

Por tanto, resulta imperativo desarrollar soluciones tecnológicas propias que permitan la gestión integral y eficiente de los procesos académicos, incorporando principios de ingeniería de software que garanticen la calidad, seguridad y escalabilidad del sistema resultante. La aplicación de arquitecturas bien definidas, patrones de diseño apropiados y metodologías de desarrollo ágiles constituyen elementos clave para lograr soluciones que respondan efectivamente a las necesidades específicas del entorno universitario cubano.

1.1.1. Arquitectura de Software

La arquitectura de software constituye el conjunto de decisiones estructurales fundamentales que definen la organización, interconexión y comportamiento de los componentes de un sistema informático, estableciendo los principios rectores para su diseño e implementación Bass et al. 2021. En el contexto de los sistemas de gestión académica, la arquitectura determina cómo los diferentes módulos funcionales —gestión de usuarios, procesamiento de inscripciones, validación de asignaturas y generación de reportes— se comunican entre sí para satisfacer los requerimientos operativos de las instituciones educativas.

La arquitectura de software cumple propósitos estratégicos múltiples en el desarrollo de plataformas académicas: facilita el cumplimiento de requisitos técnicos y operativos específicos del dominio educativo, optimiza atributos de calidad críticos como el rendimiento en períodos de alta demanda de inscripciones, garantiza la seguridad en el manejo de datos académicos sensibles, y proporciona escalabilidad para adaptarse al crecimiento institucional y la evolución de las necesidades formativas. Adicionalmente, una arquitectura bien diseñada reduce la complejidad del mantenimiento y permite la incorporación ágil de nuevas funcionalidades sin comprometer la

estabilidad del sistema existente.

Desde una perspectiva estratégica más amplia, la arquitectura de software trasciende las consideraciones puramente técnicas para convertirse en un reflejo de las prioridades organizacionales y los objetivos institucionales de las universidades. Actúa como un puente conceptual entre los procesos de negocio académico —tales como la planificación curricular, la gestión de matrícula y el seguimiento del desempeño estudiantil— y su materialización tecnológica, asegurando que las decisiones arquitectónicas estén alineadas con la misión educativa y los criterios de calidad institucional.

La gestión de la complejidad inherente a los sistemas académicos multifuncionales requiere la aplicación de principios arquitectónicos que consideren tanto los aspectos técnicos como los organizacionales. En este sentido, la Ley de Conway establece que “las organizaciones que diseñan sistemas están restringidas a producir diseños que son copias de las estructuras de comunicación de estas organizaciones” Conway 1968. Esta correspondencia entre estructura organizacional y arquitectura tecnológica resulta particularmente relevante en el contexto universitario cubano, donde los procesos de gestión académica siguen patrones institucionales específicos que deben ser adecuadamente representados en la solución informática.

1.1.2. DDD: Diseño Orientado al Dominio

El Diseño Orientado al Dominio (Domain-Driven Design, DDD) constituye una aproximación metodológica que sitúa el conocimiento del dominio como eje central del proceso de desarrollo de software. Concebida por Eric Evans a comienzos del siglo XXI, esta filosofía busca establecer una correspondencia directa entre los sistemas informáticos y la realidad organizacional que modelan, priorizando la comprensión profunda de los procesos funcionales específicos de cada institución Evans 2004.

Frente a entornos académicos caracterizados por una gran variedad de reglas administrativas, particularidades curriculares y dinámicas institucionales heterogéneas, DDD permite enfrentar la complejidad inherente al modelado de estos sistemas mediante la segmentación lógica y funcional del dominio en áreas bien definidas. En contraste con enfoques centrados exclusivamente en la estructura técnica o la disposición modular, el enfoque orientado al dominio promueve una simbiosis entre el conocimiento contextual del negocio y las decisiones de diseño del software Khononov 2021.

La implementación de esta perspectiva resulta especialmente pertinente en el desarrollo de soluciones informáticas para la gestión universitaria, donde la correcta interpretación de los procesos administrativos —como la inscripción, aprobación y asignación de asignaturas electivas— depende del conocimiento detallado que poseen los distintos actores institucionales. DDD reconoce a estos actores, denominados expertos del dominio, como figuras clave en la construcción del modelo conceptual del

sistema.

Diseño estratégico

Dentro de la propuesta metodológica de DDD, el diseño estratégico representa la fase encargada de definir los límites conceptuales y operativos del sistema a desarrollar. Esta etapa busca identificar los subdominios esenciales del problema, organizándolos en contextos delimitados (bounded contexts) que reflejen la estructura funcional real de la organización.

En el contexto de una universidad, estos subdominios pueden corresponder, por ejemplo, a los procesos de validación docente, inscripción estudiantil o control de matrícula, cada uno con sus propias reglas de negocio, niveles de autorización y procedimientos administrativos. Establecer límites claros entre estos componentes permite reducir el acoplamiento entre subsistemas y evitar conflictos semánticos durante la evolución del software.

La identificación de los subdominios se realiza en estrecha colaboración con los expertos del dominio, quienes poseen un conocimiento tácito y explícito sobre cómo operan los procesos institucionales. Esta interacción sostenida entre desarrolladores y personal académico-administrativo posibilita la construcción de un lenguaje compartido, conocido como lenguaje ubicuo, que permea tanto el código fuente como la documentación del sistema Vernon 2016. Dicho lenguaje unifica los términos utilizados por los distintos actores, evitando ambigüedades y garantizando la coherencia semántica del modelo.

El resultado de este trabajo colaborativo es un modelo de dominio que actúa como representación abstracta y operacional de la realidad organizacional. Este modelo no solo orienta la lógica del software, sino que además se convierte en una herramienta de validación conceptual, capaz de evolucionar a medida que cambian las necesidades institucionales o se introducen nuevas políticas académicas.

El diseño estratégico, por tanto, no es una fase estática del desarrollo, sino un proceso iterativo y adaptativo, que busca alinear continuamente el sistema tecnológico con los objetivos funcionales del entorno educativo. Su correcta aplicación asegura que la solución informática resultante responda a las prioridades del contexto universitario, promueva la flexibilidad operativa y sea capaz de sostener su utilidad a largo plazo

Diseño táctico

El diseño táctico constituye la materialización práctica de los conceptos identificados durante la fase estratégica, proporcionando los mecanismos concretos para transformar el conocimiento del dominio académico en estructuras de código ejecuta-

bles y mantenibles. Esta dimensión operativa de DDD se fundamenta en la aplicación sistemática de patrones específicos que permiten representar fielmente la complejidad inherente a los procesos de gestión universitaria Vernon 2013.

La transición desde el modelo conceptual hacia la implementación técnica requiere el empleo de construcciones arquitectónicas que preserven la semántica del dominio y faciliten la evolución futura del sistema. En el contexto de la gestión de asignaturas electivas, esta traducción debe contemplar tanto las reglas administrativas específicas de cada institución como los flujos de trabajo que vinculan a los diferentes actores del proceso formativo.

Los elementos fundamentales del diseño táctico incluyen estructuras de datos especializadas que reflejan los conceptos centrales del dominio académico:

Entidades: Constituyen los elementos fundamentales que poseen identidad propia y mantienen continuidad a lo largo de su ciclo de vida. En el sistema de gestión universitaria, ejemplos representativos incluyen los estudiantes, profesores y asignaturas electivas, cada uno caracterizado por atributos únicos que los distinguen dentro del contexto institucional y que persisten independientemente de las modificaciones que puedan experimentar sus propiedades secundarias.

Objetos de valor: Representan conceptos que carecen de identidad individual, definiéndose exclusivamente por sus características intrínsecas. En el ámbito académico, elementos como las calificaciones numéricas, los períodos lectivos o las direcciones de contacto ejemplifican esta categoría, ya que su relevancia radica en los valores que contienen más que en su identidad particular.

Agregados: Conforman unidades cohesivas que agrupan entidades y objetos de valor relacionados, estableciendo límites de consistencia y controlando el acceso mediante una entidad raíz. Un proceso de inscripción a asignatura electiva, por ejemplo, podría constituir un agregado que incluya la solicitud estudiantil, los criterios de elegibilidad y el estado de aprobación, todos gestionados a través de la entidad principal del proceso.

Servicios de dominio: Encapsulan operaciones complejas que trascienden las responsabilidades individuales de entidades específicas. En el contexto universitario, la validación de prerrequisitos académicos o el cálculo de disponibilidad de cupos representan servicios que requieren coordinar información proveniente de múltiples fuentes y aplicar reglas institucionales específicas.

Repositorios: Proporcionan abstracción sobre los mecanismos de persistencia, permitiendo que la lógica del dominio opere independientemente de las consideraciones técnicas relacionadas con el almacenamiento de datos. Estos componentes garantizan que los procesos académicos puedan acceder y modificar información sin verse condicionados por las particularidades de la infraestructura tecnológica subyacente.

Eventos de dominio: Capturan acontecimientos significativos que ocurren dentro del sistema y que requieren notificación o procesamiento adicional. La confirma-

ción de una inscripción estudiantil o la aprobación de una nueva asignatura electiva constituyen eventos que pueden desencadenar actualizaciones en otros subsistemas o generar comunicaciones automáticas hacia los usuarios involucrados.

La implementación efectiva del diseño táctico demanda adherencia a principios fundamentales que aseguren la coherencia entre el modelo conceptual y su representación técnica Evans 2004:

La preservación del lenguaje ubicuo en el código fuente resulta esencial para mantener la correspondencia semántica entre los procesos institucionales y su implementación informática. Esta práctica garantiza que los términos empleados por los expertos del dominio académico se reflejen directamente en las estructuras programáticas, facilitando tanto el mantenimiento como la comunicación entre desarrolladores y personal universitario.

El mantenimiento de la coherencia del modelo requiere que todas las construcciones técnicas respeten las reglas y restricciones establecidas por el dominio académico. Esta consistencia implica la validación continua de que las entidades, objetos de valor y agregados implementados preserven las invariantes del negocio y reflejen adecuadamente los procedimientos institucionales.

La separación de la lógica del dominio respecto a las consideraciones infraestructurales constituye un requisito fundamental para asegurar la flexibilidad y mantenibilidad del sistema. Esta segregación se materializa mediante el empleo de patrones arquitectónicos que aíslan los procesos académicos centrales de los detalles relacionados con bases de datos, interfaces de usuario o servicios externos.

Con el fin de incorporar la interacción del usuario sin afectar la estructura del código a largo plazo, es posible utilizar diferentes arquitecturas de software. Tal como se señaló anteriormente, estas arquitecturas proporcionan soluciones estructuradas que cumplen tanto con las necesidades técnicas como operativas, mientras promueven la organización y escalabilidad del código. Por tanto, se abordará una arquitectura de software muy utilizada que puede combinarse con DDD: Arquitectura por Capas.

1.1.3. Arquitectura por Capas

La arquitectura por capas representa uno de los paradigmas estructurales más consolidados en el desarrollo de sistemas informáticos contemporáneos, caracterizándose por su enfoque organizacional basado en la separación funcional de responsabilidades. Su amplia adopción en el ámbito de las aplicaciones empresariales se fundamenta en la simplicidad conceptual, la familiaridad que genera entre los equipos de desarrollo y los costos relativamente reducidos asociados a su implementación Richards y Ford 2020.

Esta aproximación arquitectónica resulta particularmente intuitiva para el desarrollo de sistemas académicos debido a su correspondencia natural con las estructuras

organizacionales típicas de las instituciones educativas. En consonancia con la Ley de Conway previamente mencionada, la división tradicional del personal técnico en especialistas de interfaz de usuario, lógica aplicativa y gestión de datos se alinea eficientemente con la organización por capas, facilitando la distribución de tareas y la coordinación entre equipos multidisciplinarios Richards y Ford 2020.

Sin embargo, esta alineación organizacional puede derivar en patrones arquitectónicos problemáticos si no se gestiona adecuadamente. Entre las principales dificultades identificadas se encuentran el acoplamiento excesivo entre niveles funcionales y la dispersión inadecuada de la lógica de negocio a través de múltiples componentes, lo cual puede comprometer la cohesión del sistema y dificultar su mantenimiento a largo plazo Richards y Ford 2020.

La estructuración fundamental de esta arquitectura se basa en la organización de componentes en estratos horizontales lógicamente diferenciados, donde cada nivel asume responsabilidades específicas dentro del ecosistema aplicativo. Aunque la cantidad de capas puede variar según la complejidad del dominio, la configuración más extendida comprende cuatro niveles principales: Presentación, Lógica de Negocio, Persistencia y Almacenamiento de Datos.

En contextos donde la complejidad del dominio académico es moderada o las reglas de negocio no presentan alta sofisticación, es común observar la consolidación de las capas de Negocio y Persistencia en una única estructura funcional. Esta simplificación resulta especialmente práctica cuando se emplean herramientas de mapeo objeto-relacional (ORM) como TypeORM, Hibernate o Entity Framework, que proporcionan abstracción automática entre los objetos del lenguaje de programación y las estructuras de consulta de la base de datos. En consecuencia, sistemas de gestión universitaria de alcance intermedio pueden operar eficientemente con arquitecturas de tres capas, mientras que plataformas académicas de mayor envergadura pueden requerir cinco o más niveles estructurales.

Capa de Presentación

La capa de presentación constituye el punto de contacto directo entre el sistema de gestión académica y sus usuarios finales, implementando todos los mecanismos de interacción necesarios para facilitar el acceso a las funcionalidades del sistema. En su concepción tradicional, este nivel se asociaba exclusivamente con interfaces gráficas de usuario; no obstante, los sistemas académicos modernos han expandido significativamente este concepto para abarcar múltiples modalidades de interacción, tanto síncronas como asíncronas Khononov 2021.

Las manifestaciones contemporáneas de esta capa incluyen interfaces web responsivas para el acceso estudiantil y docente, aplicaciones móviles para consultas rápidas, interfaces de línea de comandos para tareas administrativas especializadas,

y APIs que permiten la integración programática con otros sistemas institucionales. Adicionalmente, pueden incorporar mecanismos de suscripción a eventos mediante intermediarios de mensajes, facilitando la notificación automática sobre cambios en el estado de inscripciones o aprobaciones de asignaturas.

En el contexto específico de la gestión de asignaturas electivas, esta capa debe proporcionar experiencias de usuario diferenciadas según el rol institucional, asegurando que estudiantes, docentes y administradores accedan únicamente a las funcionalidades pertinentes a sus responsabilidades académicas.

Capa de Lógica de Negocio

Esta capa representa el núcleo conceptual del sistema, encargándose de la implementación y encapsulación de todas las reglas, procedimientos y políticas que rigen los procesos académicos institucionales. En ella residen las decisiones algorítmicas que determinan la elegibilidad estudiantil para asignaturas específicas, los criterios de validación docente, y los mecanismos de asignación automática de cupos Khononov 2021.

Para el sistema de gestión de asignaturas electivas, esta capa debe incorporar la lógica específica que gobierna los procesos de inscripción, validación y asignación, incluyendo reglas diferenciadas por carrera, año académico y disponibilidad de plazas. Su correcta implementación garantiza que todas las operaciones del sistema respeten las políticas institucionales y mantengan la consistencia de los datos académicos.

Capa de Persistencia

La capa de persistencia gestiona la comunicación entre la lógica aplicativa y los diversos mecanismos de almacenamiento empleados por el sistema académico. Esta responsabilidad trasciende el manejo tradicional de bases de datos relacionales, abarcando la integración con servicios de almacenamiento en la nube para documentos académicos, sistemas de mensajería para notificaciones automáticas, y servicios externos necesarios para el funcionamiento integral de la plataforma Khononov 2021.

En el contexto universitario, esta capa debe manejar eficientemente tanto los datos estructurados de inscripciones y calificaciones como los recursos multimedia asociados a las asignaturas, garantizando la disponibilidad y consistencia de la información académica.

Capa de Almacenamiento de Datos

Este nivel fundamental se encarga del acceso directo a las bases de datos y sistemas de almacenamiento, implementando las instrucciones específicas necesarias para la manipulación física de los datos académicos. Su diferenciación respecto a la capa de

persistencia radica en el nivel de abstracción: mientras la persistencia proporciona interfaces conceptuales, esta capa maneja directamente las operaciones de consulta, inserción y actualización en los repositorios de datos Richards y Ford 2020.

Consideraciones Arquitectónicas

La arquitectura por capas se caracteriza por su particionamiento técnico, en contraste con enfoques organizados por dominio funcional. Esta característica implica que los componentes se agrupan según su rol tecnológico específico rather que por su pertenencia a procesos de negocio particulares. Como consecuencia, cualquier funcionalidad académica específica tiende a distribuirse a través de todas las capas arquitectónicas, lo cual puede complicar la implementación de cambios focalizados en procesos particulares y limita la efectividad de aproximaciones de Diseño Orientado al Dominio Richards y Ford 2020.

Alternativamente, algunos enfoques arquitectónicos proponen una configuración simplificada de tres capas, donde las responsabilidades de persistencia y almacenamiento se consolidan en una única Capa de Acceso a Datos. Esta organización se complementa frecuentemente con una Capa de Servicio adicional, que define las operaciones disponibles del sistema y coordina las respuestas aplicativas, exponiendo explícitamente los casos de uso académicos y encapsulando la orquestación requerida entre los niveles subyacentes Khononov 2021.

La comunicación entre capas sigue un patrón unidireccional descendente, donde cada nivel interactúa exclusivamente con la capa inmediatamente inferior, minimizando el acoplamiento y preservando la separación de responsabilidades. Esta disciplina comunicacional resulta fundamental para mantener la integridad arquitectónica y facilitar el mantenimiento evolutivo del sistema.

Es importante reconocer que la nomenclatura de estas capas puede variar según las referencias bibliográficas y las tradiciones de desarrollo institucionales:

- Capa de Presentación equivale a Capa de Interfaz de Usuario
- Capa de Servicio corresponde a Capa de Aplicación
- Capa de Lógica de Negocio se identifica con Capa de Dominio
- Capa de Acceso a Datos se denomina también Capa de Infraestructura

Esta flexibilidad terminológica no afecta los principios arquitectónicos fundamentales, pero requiere claridad conceptual durante la documentación y comunicación del diseño del sistema.

1.2. Sistemas de asignación de recursos basados en reglas

Los sistemas de asignación de recursos basados en reglas constituyen un enfoque ampliamente utilizado en la gestión de procesos que requieren la distribución automatizada de recursos limitados entre un conjunto de usuarios o entidades, según criterios predefinidos. En el contexto académico, estos sistemas son particularmente relevantes para la asignación de plazas en asignaturas electivas, donde es necesario garantizar equidad, transparencia y alineación con los objetivos institucionales. Este tipo de sistemas se fundamenta en la definición de reglas lógicas que determinan la elegibilidad de los usuarios y priorizan su asignación en función de parámetros específicos, como características demográficas, académicas o estratégicas.

En el caso de la gestión de asignaturas electivas, las reglas permiten filtrar a los estudiantes elegibles para inscribirse en un curso, considerando variables como la facultad, la carrera, el año académico o requisitos previos establecidos por la institución. Por ejemplo, una asignatura puede estar dirigida exclusivamente a estudiantes de una carrera específica (e.g., Derecho) o a un grupo académico particular (e.g., estudiantes de cuarto año). Estas reglas de elegibilidad aseguran que la oferta formativa se ajuste a las necesidades y prioridades de la institución, optimizando el acceso a los cursos.

Además, los sistemas basados en reglas pueden incorporar mecanismos de priorización para resolver conflictos cuando la demanda supera la capacidad del curso. Estas reglas de priorización establecen un orden de preferencia, como priorizar a estudiantes de un año académico específico o de una carrera determinada, antes de asignar plazas de manera aleatoria o por otros criterios. Este enfoque no solo automatiza el proceso, reduciendo la intervención manual, sino que también garantiza consistencia y trazabilidad en las decisiones de asignación.

Desde el punto de vista técnico, estos sistemas suelen implementarse mediante algoritmos que evalúan condiciones lógicas (e.g., IF-THEN-ELSE) almacenadas en una base de datos relacional o en estructuras de configuración. Estas reglas son definidas por un rol administrador, lo que permite flexibilidad para adaptarlas a contextos específicos sin necesidad de modificar el código fuente del sistema. En el caso de la plataforma propuesta, las reglas de filtrado y priorización se diseñan para integrarse con los módulos de inscripción y aprobación, asegurando que solo los estudiantes que cumplen con los criterios establecidos puedan visualizar y preinscribirse en las asignaturas, y que las plazas se asignen de manera equitativa según las prioridades definidas por la DFP.

Este enfoque, aunque sencillo, es efectivo para contextos donde los criterios de asignación son claros y no requieren algoritmos complejos de optimización. Comparado con métodos más avanzados, como los basados en inteligencia artificial o aprendizaje automático, los sistemas basados en reglas destacan por su facilidad de implementa-

ción, bajo costo computacional y alta interpretabilidad, lo que los hace idóneos para entornos académicos con recursos tecnológicos limitados, como el de la Universidad de la Habana. Sin embargo, su principal limitación radica en la necesidad de definir manualmente las reglas, lo que puede requerir ajustes frecuentes ante cambios en las políticas institucionales.

1.3. Estado del arte

En el campo de la gestión académica y la asignación de recursos educativos, los sistemas modernos han evolucionado más allá de las plataformas tradicionales de gestión del aprendizaje para abordar problemáticas específicas relacionadas con la distribución y asignación automatizada de recursos limitados. En el contexto educativo, estos recursos incluyen asignaturas electivas con cupos limitados, espacios físicos (aulas, laboratorios), equipamiento especializado, y franjas horarias disponibles. A continuación, se analizan sistemas que implementan mecanismos de asignación de recursos basados en reglas de disponibilidad y criterios de elegibilidad, evaluando su aplicabilidad en diferentes escenarios institucionales.

Clase365

Clase365¹ es una plataforma integral de gestión educativa que se distingue por su enfoque en la automatización de procesos administrativos y la asignación inteligente de recursos académicos. Su arquitectura modular incluye un sistema robusto para la gestión de inscripciones que opera mediante reglas de elegibilidad configurables, permitiendo establecer criterios específicos para determinar qué estudiantes pueden acceder a determinados cursos o recursos en función de su programa académico, año de estudio, prerrequisitos cumplidos y disponibilidad de cupos.

La plataforma implementa un motor de reglas que evalúa automáticamente la disponibilidad de recursos en tiempo real, considerando factores como capacidad máxima, restricciones temporales y criterios de prioridad. Su sistema de asignación permite configurar múltiples niveles de validación, desde verificación de prerrequisitos hasta aplicación de políticas institucionales específicas para la distribución equitativa de recursos limitados.

Entre sus fortalezas destacan la capacidad de manejo de listas de espera automatizadas, generación de reportes de ocupación de recursos y notificaciones automáticas a estudiantes sobre disponibilidad. Sin embargo, su naturaleza comercial implica costos de licenciamiento significativos y dependencia del proveedor para actualizaciones y

¹<https://www.clase365.com/>

mantenimiento, lo que puede representar limitaciones presupuestarias para instituciones con recursos económicos restringidos.

OpenEduCat (sobre Odoo)

OpenEduCat², construido sobre el framework Odoo, representa una solución de código abierto especializada en la gestión integral de instituciones educativas con particular énfasis en la asignación automatizada de recursos académicos. Su arquitectura basada en módulos permite implementar sistemas de reglas complejas para la distribución de asignaturas electivas, considerando múltiples variables como capacidad de aulas, disponibilidad de profesores, horarios institucionales y criterios de elegibilidad estudiantil.

El sistema opera mediante un motor de workflows que evalúa la disponibilidad de recursos en función de reglas predefinidas, permitiendo establecer criterios diferenciados según facultad, carrera o nivel académico. Su módulo de gestión de cursos incluye funcionalidades avanzadas para el control automatizado de cupos, validación de prerrequisitos y asignación basada en prioridades institucionales.

Las ventajas de OpenEduCat incluyen su flexibilidad para personalización debido a su naturaleza de código abierto, integración nativa con otros módulos de gestión administrativa y capacidad para implementar reglas de negocio específicas sin restricciones de licenciamiento. No obstante, su implementación requiere conocimientos técnicos especializados en el framework Odoo, y la ausencia de soporte comercial directo puede representar desafíos para instituciones sin equipos técnicos especializados.

Relevancia de los sistemas de asignación de recursos en el contexto actual

En el contexto universitario cubano, donde la gestión de asignaturas electivas debe optimizar recursos limitados mientras cumple con normativas institucionales específicas, los sistemas de asignación basados en reglas representan una solución más apropiada que las plataformas tradicionales de gestión del aprendizaje. A diferencia de los Learning Management Systems (LMS) convencionales que se enfocan en la entrega de contenido, estos sistemas priorizan la automatización de decisiones de asignación mediante la evaluación sistemática de criterios de elegibilidad y disponibilidad de recursos.

Las soluciones analizadas demuestran que es posible implementar mecanismos sofisticados de asignación que consideren múltiples variables simultáneamente, desde restricciones de capacidad física hasta políticas académicas complejas. Sin embargo, tanto Classe365 como OpenEduCat presentan limitaciones para el contexto específico

²<https://openeducat.org/>

cubano: la primera por sus costos asociados y dependencia comercial, y la segunda por los requerimientos técnicos especializados para su implementación y mantenimiento.

Por estas razones, resulta indispensable considerar el desarrollo de una solución tecnológica propia, diseñada específicamente para responder a las necesidades del contexto educativo cubano. Un sistema de asignación de asignaturas electivas desarrollado localmente permitiría implementar reglas de negocio específicas para el control automatizado de disponibilidad y elegibilidad, ofreciendo una interfaz adaptada a los procesos institucionales existentes y capacidad para generar reportes especializados que apoyen la toma de decisiones académicas, sin las limitaciones económicas o técnicas de las soluciones existentes.

Capítulo 2

Concepción y diseño de la solución

El desarrollo de sistemas digitales de gestión de asignaturas electivas en la educación superior requiere una comprensión profunda de las necesidades institucionales y una traducción precisa de dichas necesidades en especificaciones técnicas funcionales. La fase de análisis y diseño constituye el puente fundamental entre la identificación de problemáticas organizacionales y la materialización de soluciones tecnológicas viables.

Este capítulo presenta el diseño teórico conceptual que soportará la funcionalidad de la plataforma, se aborda el análisis detallado de los requerimientos y se caracterizan los roles y perfiles de usuario del sistema. Asimismo, se establecen los casos de uso principales y la arquitectura de datos necesaria para garantizar que la solución responda de manera integral a las necesidades identificadas en el contexto de la gestión de asignaturas electivas universitarias.

2.1. Problemática

Las instituciones de educación superior enfrentan desafíos significativos en la gestión de asignaturas electivas, particularmente cuando estas deben coordinarse entre múltiples facultades, carreras y niveles académicos. En el contexto de la Universidad de la Habana, este proceso tradicionalmente se ha caracterizado por una alta dependencia de métodos manuales y sistemas fragmentados que no logran integrarse de manera efectiva las diferentes etapas del ciclo de gestión académica. Esta situación genera ineficiencias operativas, limita la transparencia del proceso y dificulta la toma de decisiones basada en datos confiables.

La problemática se manifiesta de manera particular en la Dirección de Formación de Pregrado, donde convergen las necesidades de múltiples actores con diferentes niveles de responsabilidad y acceso a la información. Los profesores requieren mecanismos ágiles para proponer nuevas asignaturas electivas, especificando contenidos, requisitos y modalidades de evaluación. Simultáneamente, los administradores aca-

démicos necesitan herramientas que faciliten la revisión, validación y aprobación de estas propuestas, aplicando criterios institucionales específicos. Por su parte, los estudiantes demandan procesos claros de inscripción, equitativos y que respeten las particularidades de su plan de estudios.

Esta situación se agrava al analizar detenidamente el flujo actual del proceso de gestión de asignaturas electivas. En su forma actual, la Dirección de Formación de Pregrado debe coordinar múltiples fases administrativas con una alta carga operativa y escaso apoyo tecnológico. Los profesores interesados en impartir asignaturas electivas deben presentar una propuesta detallada que incluya información general sobre la asignatura, su fundamentación, objetivos, temario, bibliografía, modalidad de impartición, sistema de evaluación, y una justificación de la modalidad seleccionada. Además, deben registrar datos personales y profesionales tanto del docente principal como de los colaboradores, incluyendo categoría docente, grado científico y medios de contacto.

A esta información se suma la necesidad de especificar elementos logísticos como la capacidad máxima de matrícula, los requisitos mínimos para cursar la asignatura, el lugar exacto donde se impartirá (en caso de ser presencial o híbrida), así como los ejes y sectores estratégicos a los que tributa. Todo este proceso requiere la recopilación y validación manual de una cantidad considerable de datos, muchas veces mediante documentos físicos o formatos no estandarizados, lo que introduce una alta probabilidad de errores, demoras y pérdida de información.

Una vez recibidas las propuestas, la Dirección debe revisarlas y aprobarlas, sin contar con un sistema eficiente de seguimiento o retroalimentación estructurada. Los estudiantes, por su parte, acceden a un listado de asignaturas aprobadas para realizar su inscripción, donde deben proporcionar información detallada que incluye datos personales, académicos y de contacto, además de justificar su interés en matricular la asignatura. No existe actualmente una integración efectiva con sistemas como EVEA, ni un control automatizado sobre la capacidad de las asignaturas o las reglas de priorización por carrera, facultad o año académico.

Los actores involucrados —profesores, estudiantes y administradores— se enfrentan a limitaciones importantes: los primeros, por la complejidad del proceso de solicitud; los segundos, por la opacidad y lentitud del procedimiento de inscripción; y los terceros, por la falta de herramientas que permitan un control eficiente, seguro y centralizado. Además, la elaboración de actas, el seguimiento de las calificaciones, la producción de reportes y estadísticas y la trazabilidad general del proceso resultan fragmentados y altamente dependientes del trabajo manual, dificultando cualquier análisis posterior de calidad o eficiencia.

Esta fragmentación y falta de automatización impide una planificación académica ágil, dificulta la transparencia en la toma de decisiones y retrasa los procesos claves para el desarrollo curricular y la satisfacción de las necesidades formativas de los

estudiantes.

A continuación se detallan los requerimientos específicos que fundamentan el desarrollo de esta solución tecnológica.

2.2. Determinación de los Requerimientos

Los requerimientos del sistema se agrupan en cuatro categorías fundamentales: funcionales, no funcionales, informacionales y de entorno. Estos fueron identificados mediante un análisis detallado de la problemática institucional y las necesidades específicas de la Dirección de Formación de Pregrado en el contexto de la gestión de asignaturas electivas universitarias.

2.2.1. Requerimientos Funcionales

Las siguientes funcionalidades han sido definidas para permitir una gestión integral del proceso de inscripción, aprobación y matrícula de asignaturas electivas:

- **Definición de roles diferenciados:**

Se contemplan tres perfiles de usuario:

- **Personal de la Dirección de Formación de Pregrado:** Acceso completo al sistema, con funciones de supervisión, aprobación y configuración de parámetros institucionales. Aprueba los cursos propuestos por los profesores y es quien determina cuando un curso es puesto como disponible en un semestre.
- **Profesor:** Propone asignaturas, gestiona información académica y administra calificaciones.
- **Estudiante:** Consulta la oferta de asignaturas, realiza su inscripción y accede a su información académica.

- **Gestión del proceso de propuesta de asignaturas:**

La plataforma debe permitir el registro detallado de propuestas, incluyendo datos generales, docentes responsables, contenidos académicos, modalidades, cupos, requisitos y vínculos institucionales estratégicos.

- **Módulo de revisión y aprobación administrativa:**

Herramientas especializadas deben permitir a los administradores evaluar propuestas según criterios institucionales, aprobar asignaturas y habilitar accesos para los profesores responsables.

- **Funcionalidades de inscripción estudiantil:**

Los estudiantes deben acceder al catálogo de asignaturas aprobadas e inscriben según disponibilidad de cupos y reglas configurables por el administrador (facultad, carrera, año, etc.).

- **Generación automatizada de documentación académica:**

El sistema debe generar listados de inscritos, actas académicas y documentos oficiales con estructura y formato requeridos para certificación institucional.

2.2.2. Requerimientos No Funcionales

Para garantizar un correcto funcionamiento, la solución debe cumplir con los siguientes requerimientos no funcionales:

- **Seguridad y control de acceso:** Autenticación y autorización según roles; protección de datos personales y académicos.
- **Usabilidad y experiencia de usuario:** Interfaces intuitivas, accesibles y adecuadas para usuarios con distintos niveles de competencia tecnológica.
- **Administración y mantenimiento:** Herramientas para configurar parámetros, gestionar contenidos y usuarios, y actualizar el sistema con mínima interrupción.
- **Accesibilidad multiplataforma:** Aplicación web compatible con diversos dispositivos y navegadores sin necesidad de instalación adicional.
- **Trazabilidad y auditoría:** Registro detallado de acciones para seguimiento, auditoría y resolución de incidentes.
- **Capacidades de análisis y reporte:** Estadísticas y reportes sobre matrícula, desempeño y uso de recursos para apoyar la toma de decisiones.

2.2.3. Requerimientos Informacionales

El sistema debe proveer información clara, oportuna y contextualizada para apoyar los procesos de gestión, propuesta, matrícula y evaluación de asignaturas electivas. A continuación, se describen los requerimientos informacionales identificados para cada perfil de usuario:

a) Dirección de Formación de Pregrado (Administrador)

- Acceso completo a la información de gestión y supervisión, incluyendo:
 - Listados detallados de todas las propuestas de asignaturas con sus estados (pendientes, aprobadas, rechazadas).
 - Datos completos de los profesores proponentes y colaboradores (información personal, académica y de contacto).
 - Información estadística y cuantitativa de matriculados por asignatura, facultad, carrera y año académico.
 - Reportes de promoción y calificaciones para análisis cualitativo.
 - Histórico de inscripciones y terminaciones por asignatura.
 - Parámetros configurables del sistema (reglas de inscripción, capacidades, prioridades por carrera).

b) Profesor

- Acceso a la información de su propuesta de asignatura (tanto en estado pendiente como aprobada), incluyendo contenido temático, modalidad, bibliografía, criterios de evaluación y capacidad de matrícula.
- Listado actualizado de estudiantes inscritos en su curso, con sus datos de contacto, facultad, carrera y año académico.
- Herramientas para cargar calificaciones y generar actas, con alertas sobre errores o inconsistencias.
- Visualización del estado de aprobación de sus propuestas, y notificaciones automáticas sobre cambios en el proceso de validación institucional.
- Acceso a informes de avance del curso, con indicadores sobre la participación y resultados académicos de los estudiantes inscritos.

c) Estudiante

- Información detallada de cada asignatura disponible para matrícula, incluyendo:
 - Nombre de la asignatura
 - Nombre del profesor
 - Modalidad (virtual, presencial, híbrida)

- Centro, facultad o dirección que propone la electiva
- Texto de presentación del curso
- Lugar de impartición
- Estado de su inscripción (confirmada, pendiente, cerrada por cupo), y acceso al historial de asignaturas electivas cursadas y calificaciones obtenidas.
- Acceso a sus datos académicos registrados en el sistema, con posibilidad de verificar y actualizar su información de contacto.
- Notificaciones automáticas relacionadas con aceptación en cursos, inicio de clases, publicaciones del profesor o cambios en la oferta.

Estos requerimientos informacionales son esenciales para garantizar la transparencia del proceso, la trazabilidad institucional y la autonomía operativa de cada usuario en el sistema.

2.2.4. Requerimientos de Entorno

Para una implementación exitosa y operación estable, se establecen los siguientes requerimientos del entorno:

- **Arquitectura multiplataforma:** Compatibilidad con distintos sistemas operativos y entornos de servidor.
- **Alojamiento en infraestructura nacional:** Uso de servidores nacionales para garantizar autonomía tecnológica y cumplimiento de normativas de protección de datos.
- **Validación de identidad con sistemas institucionales existentes:** El sistema debe permitir la verificación de credenciales institucionales (correo y contraseña universitarios) a través de mecanismos de autenticación con servicios de la universidad, garantizando que los usuarios registrados sean miembros válidos de la comunidad académica.
- **Escalabilidad y rendimiento:** Capacidad de manejar crecimiento en usuarios y datos sin afectar el rendimiento, incluso en periodos de alta demanda.

2.3. Casos de uso y flujos principales

La definición de casos de uso constituye un **enfoque metodológico** fundamental para representar las interacciones entre el sistema y sus diferentes actores, estableciendo un marco conceptual que facilita la comprensión de las funcionalidades requeridas y los flujos de trabajo necesarios para su implementación. A través del análisis de casos de uso se logra identificar de manera precisa las responsabilidades y capacidades de cada rol dentro del sistema, proporcionando una base sólida para el diseño de la arquitectura funcional.

En el contexto del sistema de gestión de asignaturas electivas, se han identificado tres actores principales que interactúan con la plataforma, cada uno con responsabilidades específicas y niveles de acceso diferenciados que reflejan su función dentro del proceso académico institucional.

Actores del sistema

El sistema contempla tres perfiles de usuario claramente diferenciados, que coinciden con las personas descritas en la Sección 2.2.1:

Administrador (DFP): Representa el rol de mayor autoridad dentro del sistema, correspondiente a la Dirección de Formación de Pregrado. Este actor posee privilegios completos para supervisar, configurar y administrar todos los aspectos operativos de la plataforma, incluyendo la aprobación de asignaturas propuestas, la gestión de reglas de inscripción y el acceso a información estadística integral. Además, el administrador puede asumir las funciones propias de un Profesor, interactuando con el sistema en calidad de tal, cuando sea necesario. Es importante destacar que los roles de Profesor y Estudiante no son asignados directamente por el administrador, sino que se determinan automáticamente a partir de la cuenta institucional proporcionada por el departamento de informática de la universidad.

Profesor: Actor responsable de la creación y gestión académica de las asignaturas electivas. Su participación en el sistema abarca desde la propuesta inicial de cursos hasta la administración de calificaciones y generación de documentación académica.

Estudiante: Usuario final que accede al sistema para consultar la oferta académica disponible, realizar inscripciones en asignaturas de su interés y consultar su información académica personal, así como la nota adquirida en cada curso inscrito.

Interacciones del Administrador

El administrador del sistema ejerce funciones de supervisión y control que garantizan el cumplimiento de los estándares académicos institucionales. Además de sus funciones administrativas, el administrador tiene la capacidad de desempeñar todas

las acciones disponibles para el perfil de profesor. Sus interacciones principales incluyen la evaluación y aprobación de propuestas de cursos o asignaturas presentadas por los profesores, aplicando criterios específicos establecidos por la institución.

Estas interacciones se representan en la Figura 2.1, que muestra el diagrama de casos de uso correspondiente al perfil de administrador.

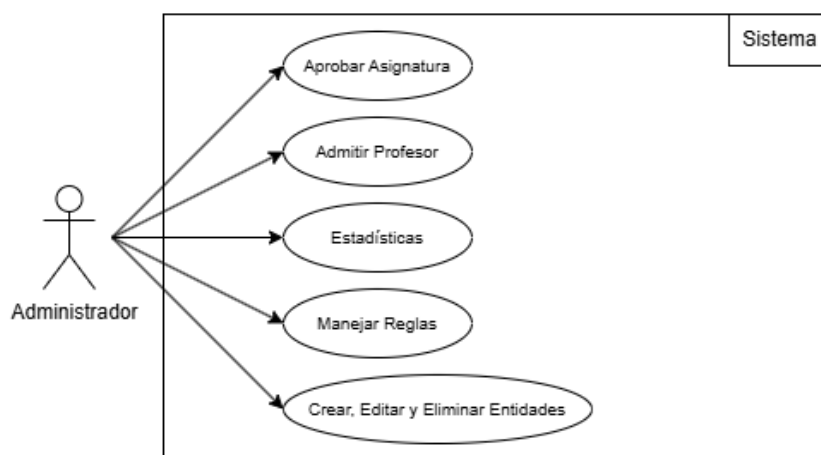


Figura 2.1: Diagrama de casos de uso del Administrador. El caso de uso “Crear, Editar y Eliminar Entidades” agrupa acciones sobre múltiples entidades del sistema, como asignaturas, facultades, centros, carreras y usuarios. Esta agrupación se realizó para mantener la claridad visual del diagrama.

La gestión de entidades constituye una función central del administrador, quien puede crear, editar y eliminar elementos fundamentales del sistema como facultades, centros, carreras y usuarios. Esta capacidad de configuración permite adaptar el sistema a las necesidades específicas de la institución y mantenerlo actualizado ante cambios organizacionales.

El establecimiento y la modificación de las reglas de inscripción representa otra responsabilidad crítica del administrador. Una **regla de inscripción** es un conjunto de condiciones o criterios que determinan si un estudiante puede o no inscribirse en una determinada asignatura. Estas reglas suelen considerar factores como el año académico del estudiante, la facultad a la que pertenece, la carrera que cursa, y otros requisitos previos definidos por la institución. Establecer correctamente estas reglas es fundamental para asegurar una planificación académica ordenada, coherente con los programas de estudio y equitativa para todos los estudiantes.

La generación de estadísticas y reportes proporciona al administrador herramientas analíticas para evaluar el desempeño del sistema, identificar tendencias de matrícula y tomar decisiones informadas sobre la oferta académica. Estos análisis incluyen datos de participación por facultad, carrera y año académico, así como indicadores de rendimiento.

Interacciones del Profesor

Los profesores constituyen los proveedores de contenido académico dentro del sistema, con responsabilidades que abarcan desde la concepción inicial de las asignaturas hasta la evaluación final de los estudiantes participantes.

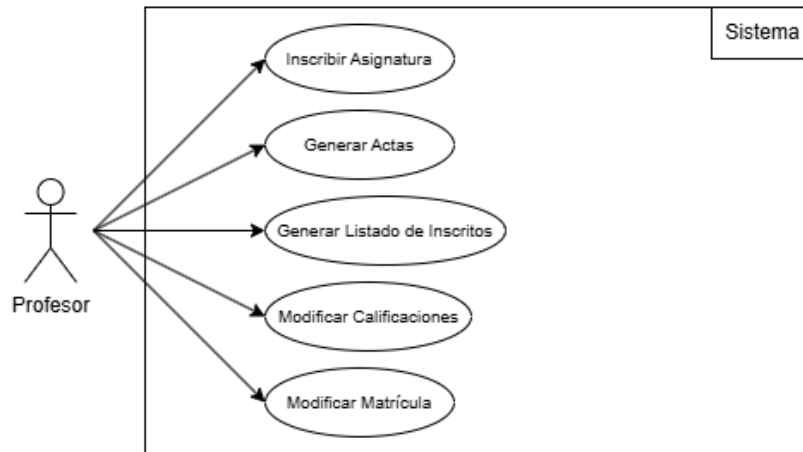


Figura 2.2: Diagrama de casos de uso del Profesor.

La inscripción de asignaturas representa el punto de entrada del profesor al sistema, requiriendo la presentación de información detallada sobre el curso propuesto. Este proceso incluye la especificación de datos generales, información del docente responsable y colaboradores, contenidos académicos, metodología de evaluación y requisitos para los estudiantes inscritos en el curso habilitado.

Una vez aprobada una asignatura, el profesor adquiere capacidades de gestión académica que incluyen la administración de la matrícula estudiantil. Esta función permite realizar ajustes en la lista de participantes según criterios académicos específicos y particulares, respetando siempre los límites de capacidad establecidos y

las reglas institucionales vigentes. Además, el profesor tiene la facultad de asignar y modificar las calificaciones finales de los estudiantes inscritos.

El profesor puede acceder al sistema para generar listados actualizados de estudiantes inscritos en sus asignaturas, así como el acta académica correspondiente a la evaluación final del curso. A través de una interfaz sencilla, el docente selecciona el curso y el sistema genera automáticamente los documentos con el formato institucional requerido, incluyendo espacios destinados a firmas y sellos según la normativa universitaria. Esta funcionalidad facilita al profesor la gestión administrativa y documental del proceso educativo.

Interacciones del Estudiante

Los estudiantes representan el eje central del sistema, con acceso a funcionalidades específicamente diseñadas para facilitar su participación en el proceso de selección e inscripción de asignaturas electivas.

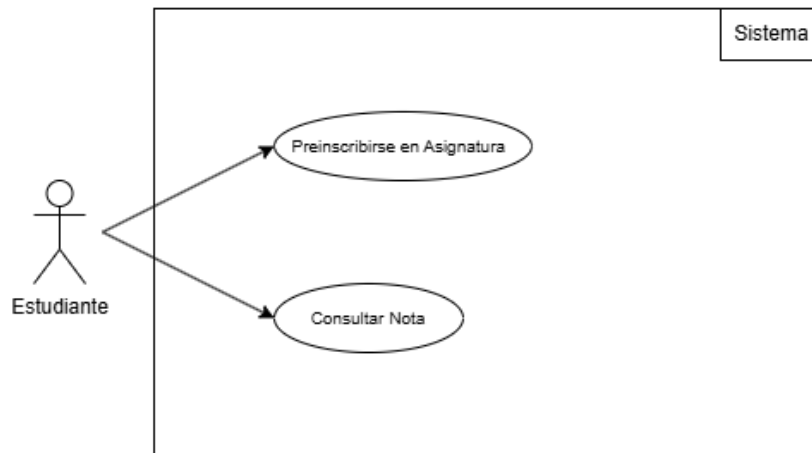


Figura 2.3: Diagrama de casos de uso del Estudiante.

La consulta de asignaturas disponibles proporciona a los estudiantes acceso personalizado al conjunto de cursos electivos que han sido aprobados y habilitados para su perfil académico específico (como carrera, año u otras condiciones), según las reglas definidas por la administración. Esta funcionalidad presenta información detallada sobre cada asignatura, incluyendo objetivos, contenidos, modalidad de impartición, requisitos y disponibilidad de cupos.

El proceso de inscripción en asignaturas permite a los estudiantes formalizar su participación en cursos de su interés, sujeto a la disponibilidad de cupos y el cumplimiento de los requisitos establecidos. El sistema implementa validaciones automáticas que verifican la elegibilidad del estudiante según las reglas configuradas por el administrador.

Además de las funcionalidades relacionadas con la selección e inscripción, el sistema también ofrece herramientas para el seguimiento académico. La consulta de calificaciones proporciona acceso a la información académica personal, permitiendo a los estudiantes monitorear su progreso y desempeño en las asignaturas en las que participan. Esta funcionalidad mantiene un registro histórico completo de todas las evaluaciones recibidas.

Flujos de trabajo principales

Los casos de uso identificados se organizan en flujos de trabajo secuenciales que reflejan el ciclo completo de la gestión de las asignaturas electivas. El flujo principal inicia con la propuesta de asignaturas por parte de los profesores, continúa con la evaluación y aprobación administrativa, prosigue con la habilitación de inscripciones estudiantiles y culmina con la generación de documentación académica oficial.

Este diseño de flujos garantiza la trazabilidad completa del proceso, desde la concepción inicial hasta la certificación final, proporcionando mecanismos de control y validación en cada etapa. La estructura modular de los casos de uso facilita la implementación incremental del sistema y permite la incorporación de nuevas funcionalidades según evolucionen las necesidades institucionales.

2.4. Diseño de la base de datos

Con el fin de estructurar y gestionar eficientemente la información del sistema, se ha diseñado una base de datos relacional que contempla las principales entidades involucradas. Este diseño se basa en los requerimientos derivados de la problemática expuesta en la Sección 2.1, donde se identificaron las necesidades del sistema y los procesos clave a resolver. A continuación, se describen las entidades definidas junto con sus atributos y relaciones clave.

Definición de Entidades:

- **Curso:** ID (PK¹), Título, ID Centro (FK² → Centro), Modalidad, Capacidad de matrícula, ID Profesor principal (FK → Profesor), Justificación del curso, Ob-

¹PK (Primary Key): Llave primaria de la entidad.

²FK (Foreign Key): Llave foránea que referencia a otra tabla.

jetivo general, Objetivos específicos, Temario, Bibliografía básica, Bibliografía complementaria, Sistema de evaluación, Justificación de modalidad, Requisitos básicos, Lugar de encuentro, Ejes estratégicos, Sectores estratégicos, Ruta de carta de autorización, URL de imagen, Fecha de creación, Fecha de actualización, Estatus de aprobación.

- **Curso Aprobado:** ID (PK), ID Curso (FK → Curso), Fecha de inicio, Fecha de fin.
- **Aplicación a Curso:** ID (PK), ID Estudiante (FK → Estudiante), ID Curso Aprobado (FK → Curso Aprobado), Carta de motivación, Año académico, Estatus, Fecha de aplicación.
- **Calificación de Curso:** ID (PK), ID Curso Aprobado (FK → Curso Aprobado), ID Estudiante (FK → Estudiante), Calificación, Comentario.
- **Profesor:** ID Usuario (PK), Nombre completo, Email principal, Categoría docente, Grado académico, Teléfono fijo, Número de teléfono, Email secundario, Estatus de aprobación.
- **Estudiante:** ID Usuario (PK), Nombre completo, Número de identificación, Email principal, Nombre de usuario Enea, Número de teléfono, ID Facultad (FK → Facultad), ID Carrera (FK → Carrera), Email secundario.
- **Centro:** ID (PK), Nombre.
- **Facultad:** ID (PK), Nombre.
- **Carrera:** ID (PK), Nombre.

Para facilitar la comprensión de la estructura de la base de datos, el diagrama de la Figura 2.1 ilustra el Modelo Entidad-Relacional-Extendido (MERX) del sistema, que refleja los conjuntos de entidades principales y sus interrelaciones clave.

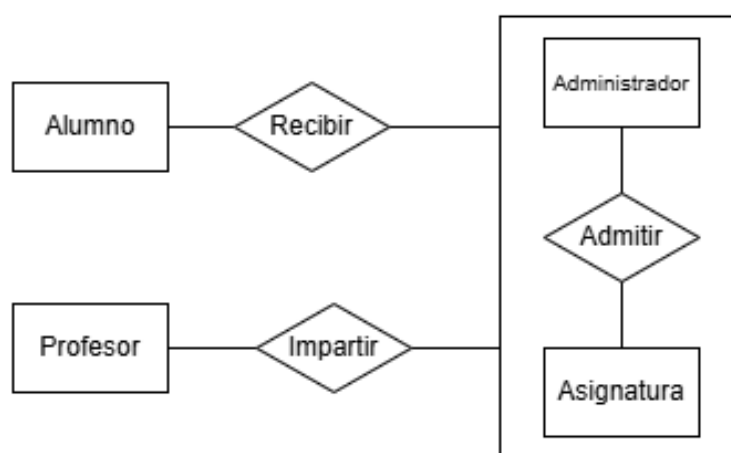


Figura 2.4: Modelo Entidad-Relacional Extendido de la base de datos.

Capítulo 3

Detalles de Implementación y Experimentación

Toda solución computacional, una vez concebida y diseñada, requiere una implementación técnica rigurosa para transformarse en una herramienta funcional y accesible para los usuarios finales. En este capítulo se abordan los aspectos fundamentales del proceso de desarrollo de la plataforma de gestión de asignaturas electivas, desde la selección de tecnologías hasta la validación experimental del sistema implementado. Se detallan las decisiones técnicas adoptadas, justificando la elección de cada componente tecnológico en función de los requerimientos identificados y las particularidades del contexto académico universitario.

El proceso de implementación se estructura en múltiples dimensiones que abarcan desde la arquitectura de datos hasta la experiencia de usuario final. Se presenta un análisis exhaustivo de las herramientas tecnológicas seleccionadas, considerando criterios de escalabilidad, mantenibilidad, seguridad y compatibilidad con el ecosistema tecnológico institucional. Asimismo, se documentan las pruebas experimentales realizadas para validar tanto la funcionalidad técnica como la efectividad operacional de la solución propuesta.

3.1. Tecnologías y herramientas utilizadas para el desarrollo del sistema

La arquitectura tecnológica del sistema incorpora principalmente herramientas de código abierto, buscando optimizar la sostenibilidad económica y facilitar la autonomía tecnológica de la institución, aunque también se ha integrado software propietario como SQL Server por su confiabilidad y soporte en entornos académicos. La selección de cada componente tecnológico se fundamenta en un análisis comparativo que

evalúa criterios de rendimiento, escalabilidad, seguridad, facilidad de mantenimiento y compatibilidad con estándares de desarrollo empresarial.

El enfoque metodológico adoptado para la selección tecnológica considera la necesidad de facilitar la colaboración entre desarrolladores y la incorporación de nuevos miembros al equipo de desarrollo en futuras iteraciones del proyecto. Esta consideración resulta fundamental para garantizar la continuidad del desarrollo y el mantenimiento a largo plazo de la plataforma académica.

3.1.1. Sistema de Gestión de Base de Datos: SQL Server

Para la gestión de datos del sistema se ha seleccionado Microsoft SQL Server como sistema de gestión de base de datos relacional. Esta elección se fundamenta en la robustez y confiabilidad que caracteriza a esta plataforma de base de datos empresarial, así como en su amplia adopción en entornos académicos y corporativos que garantiza la disponibilidad de recursos técnicos especializados.

SQL Server proporciona funcionalidades avanzadas de seguridad, backup y recuperación que resultan esenciales para la protección de datos académicos sensibles. Su arquitectura permite implementar mecanismos de control de acceso granular y auditoría detallada, aspectos fundamentales para el cumplimiento de regulaciones de protección de datos en el contexto educativo. Adicionalmente, las capacidades de escalabilidad horizontal y vertical de la plataforma aseguran el soporte adecuado para el crecimiento proyectado de usuarios y volumen de información académica.

3.1.2. Framework de Desarrollo Backend: ASP.NET Core

Para la implementación de la lógica de negocio y los servicios del sistema se ha adoptado ASP.NET Core como framework de desarrollo backend. Esta tecnología de Microsoft representa una plataforma de desarrollo moderna, multiplataforma y de alto rendimiento que facilita la creación de aplicaciones web empresariales escalables y seguras.

La elección de ASP.NET Core se justifica por su arquitectura modular que permite el desarrollo de servicios web RESTful con capacidades avanzadas de autenticación, autorización y gestión de sesiones. El framework proporciona herramientas integradas para la implementación de patrones de diseño empresariales como la inyección de dependencias, el control de versiones de API y la documentación automática de servicios. Estas características facilitan el mantenimiento del código y la colaboración entre desarrolladores en equipos distribuidos.

3.1.3. Mapeo Objeto-Relacional: Entity Framework Core

Para la abstracción de la capa de acceso a datos se utiliza Entity Framework Core, el mapeador objeto-relacional oficial de Microsoft para .NET. Esta herramienta facilita la interacción con la base de datos mediante el uso de objetos .NET, eliminando la necesidad de escribir código SQL específico para la mayoría de operaciones de acceso a datos.

Entity Framework Core proporciona funcionalidades avanzadas como migraciones automáticas de esquema de base de datos, lazy loading de entidades relacionadas y optimización automática de consultas. Estas características reducen significativamente el tiempo de desarrollo y minimizan la posibilidad de errores en la implementación de operaciones de persistencia de datos. Además, su integración nativa con ASP.NET Core facilita la implementación de patrones arquitectónicos como Repository y Unit of Work.

3.1.4. Tecnología de Desarrollo Frontend: React.js

Para la implementación de la interfaz de usuario se ha seleccionado React.js, una biblioteca de JavaScript desarrollada por Meta que facilita la construcción de interfaces de usuario interactivas y responsivas. React.js se ha consolidado como una de las tecnologías frontend más utilizadas en el desarrollo de aplicaciones web modernas, con un ecosistema maduro y una comunidad activa de desarrolladores.

La arquitectura basada en componentes de React permite la creación de interfaces modulares y reutilizables que facilitan el mantenimiento y la extensión de funcionalidades. Su modelo de gestión de estado y el virtual DOM proporcionan un rendimiento óptimo para aplicaciones con alta interactividad, aspecto fundamental para la experiencia de usuario en procesos académicos que requieren navegación fluida entre múltiples formularios y vistas de datos.

3.1.5. Plataforma de Contenedorización: Docker

Para el empaquetado y despliegue de la aplicación se ha adoptado Docker como plataforma de contenedorización. Esta tecnología permite encapsular la aplicación y todas sus dependencias en contenedores ligeros y portátiles que garantizan la consistencia del entorno de ejecución independientemente de la infraestructura de despliegue.

La utilización de Docker facilita la implementación de prácticas de integración continua y despliegue continuo, aspectos fundamentales para el mantenimiento eficiente de aplicaciones en producción. Los contenedores Docker proporcionan aislamiento de procesos y recursos que mejora la seguridad y estabilidad del sistema, mientras que

su portabilidad garantiza la flexibilidad de migración entre diferentes entornos de infraestructura sin dependencias externas.

La estrategia de contenedorización adoptada contempla la separación de la aplicación en múltiples contenedores especializados para cada componente del sistema, incluyendo el frontend React, el backend ASP.NET Core y la base de datos SQL Server. Esta arquitectura de microservicios facilita el escalado independiente de cada componente según las demandas específicas de carga y proporciona mayor flexibilidad para actualizaciones y mantenimiento del sistema.

3.2. Arquitectura del sistema implementado

3.3. Pruebas y validación experimental

Bibliografía

- Bass, L., Clements, P., & Kazman, R. (2021). *Software Architecture in Practice* (4th). Addison-Wesley Professional. (Vid. pág. 5).
- Conway, M. E. (1968). How do Committees Invent? *Datamation*, 14(4), 28-31 (vid. pág. 6).
- Evans, E. (2004). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Professional. (Vid. págs. 6, 9).
- Khononov, V. (2021). *Learning Domain-Driven Design*. O'Reilly Media. (Vid. págs. 6, 10-12).
- Richards, M., & Ford, N. (2020). *Fundamentals of Software Architecture: An Engineering Approach*. O'Reilly Media. (Vid. págs. 9, 10, 12).
- Vernon, V. (2013). *Implementing Domain-Driven Design* [Vid. págs. 7, 12]. Pearson Education. (Vid. pág. 8).
- Vernon, V. (2016). *Domain-Driven Design Distilled* [Vid. pág. 6]. Addison-Wesley Professional. (Vid. pág. 7).