



Nombre de la institución: Escuela Politécnica Nacional

Nombre de la materia: Fundamentos de Bases de Datos

Nombre del proyecto: CASO 4: SISTEMA DE GESTIÓN ACADÉMICA UNIVERSITARIA

Integrantes: Mathew Verdezoto, Aidan Carrasco, Andrés Merino, Zenán Fernández, Luz Salazar, Alejandro Guanocunga, Kenneth Gaona, Alex Pogo

Nombre del docente: CHANCUSIG ESPIN BERNARDINO

Fecha: 28 de noviembre de 2025

Ciclo / Paralelo: GR1CD

## **Introducción**

### **Descripción Ampliada del Dominio**

El presente proyecto se desarrolla en el contexto universitario, una institución de educación superior que se encuentra en un proceso de transformación digital para optimizar sus procesos académicos y administrativos. La universidad posee una estructura organizacional compleja y de gran escala, compuesta por 8 facultades que administran un total de 35 carreras distribuidas en distintos niveles de formación: Técnico, Tecnológico, Grado (Licenciaturas/Ingenierías) y Posgrado (Maestrías/Especializaciones).

Actualmente, la institución cuenta con una comunidad universitaria conformada por 450 docentes (con diversas categorías y dedicaciones) y 12,000 estudiantes activos. La oferta académica es diversa, impartándose en tres modalidades distintas: Presencial, Semipresencial y Online.

La problemática principal radica en la necesidad de centralizar la gestión de información que actualmente se genera en volúmenes masivos. Esto incluye desde la gestión de recursos humanos y físicos (infraestructura), hasta el control minucioso del ciclo de vida estudiantil (ingreso, avance curricular, titulación) y la producción científica (investigación). La falta de un sistema integral dificulta el control de reglas de negocio críticas, como el cruce de horarios, la validación de prerequisitos complejos, el cálculo de cargas horarias docentes y el seguimiento eficiente de los procesos de titulación.

### **Alcance del Sistema**

El "Sistema de Gestión Académica Universitaria" abarcará el diseño, modelado e implementación de una base de datos relacional robusta que soporte los siguientes módulos y procesos de negocio:

Gestión Organizacional: Administración de la jerarquía universitaria (Facultades, Departamentos y Carreras) y configuración de períodos académicos.

Gestión de Personal Docente: Registro de perfiles, títulos académicos, categorías (Titular, Agregado, Auxiliar), dedicación y control de carga horaria (Docencia, Investigación, Gestión, Vinculación).

Gestión Estudiantil: Administración de perfiles de estudiantes, historial de matrículas, cálculo de GPA acumulado, gestión de becas y control de estados (Activo, Suspendido, Graduado, etc.).

Planificación Académica: Diseño de mallas curriculares versionadas por cohorte, administración de asignaturas con sus pre/correquisitos y planificación de oferta académica (apertura de paralelos y asignación de horarios).

Control de Matrícula y Asistencia: Procesos de inscripción de asignaturas validando reglas de negocio (cruces de horario, cupos, prerrequisitos) y registro de asistencia.

Evaluación y Calificaciones: Registro de notas parciales, finales, supletorios y mejoramientos, asegurando la integridad de la promoción estudiantil.

Procesos de Titulación: Gestión de las modalidades de graduación (Tesis, Proyecto Integrador, Examen Complexivo), asignación de tutores y tribunales.

Gestión de Investigación: Administración de proyectos, presupuestos, equipos de investigación y registro de producción científica (publicaciones).

Gestión de Infraestructura y Biblioteca: Inventario y reserva de espacios físicos (aulas, laboratorios), equipamiento y gestión de préstamos bibliotecarios.

Fuera del Alcance: Este proyecto se limita al diseño de base de datos y lógica de negocio en backend (PL/SQL). No incluye el desarrollo de interfaces de usuario (Frontend), módulos financieros contables externos (facturación electrónica con entes gubernamentales) ni gestión de nómina detallada (pagos bancarios).

## **Objetivos del Proyecto**

### **Objetivo General**

Diseñar e implementar una base de datos relacional optimizada en Oracle Database 19c para la universidad, garantizando la integridad, consistencia y disponibilidad de la información académica y administrativa, cumpliendo estrictamente con las reglas de negocio institucionales.

### **Objetivos Específicos**

1. **Modelar la Arquitectura de Datos:** Elaborar los modelos Conceptual, Lógico y Físico utilizando la herramienta PowerDesigner, identificando correctamente las entidades, atributos y relaciones complejas (recursivas, ternarias y jerárquicas) del dominio universitario.
2. **Asegurar la Integridad de la Información:** Implementar restricciones de integridad referencial y reglas de negocio mediante el uso de Constraints (Primary Keys, Foreign Keys, Checks, Unique) para evitar inconsistencias como horarios solapados o cargas horarias excedidas.
3. **Normalizar la Estructura de Datos:** Aplicar las reglas de normalización hasta la Tercera Forma Normal (3NF) o BCNF para reducir la redundancia de datos y prevenir anomalías de actualización, especialmente en el manejo de historiales académicos y mallas curriculares.

4. **Implementar Lógica de Negocio en Base de Datos:** Desarrollar scripts SQL eficientes para la creación de objetos y la manipulación de datos, incluyendo la validación de requerimientos complejos como el cálculo de eficiencia terminal y la detección de estudiantes en riesgo.
5. **Validar el Sistema:** Ejecutar un plan de pruebas exhaustivo que demuestre la capacidad del sistema para responder a las consultas estratégicas y operativos de la universidad, así como su resistencia ante intentos de violación de las reglas de negocio establecidas.

## **Reglas sobre ESTRUCTURA ORGANIZACIONAL**

### **Regla 1: La universidad tiene FACULTADES que agrupan CARRERAS relacionadas.**

- **Impacto en Diseño:** Se requieren las entidades FACULTAD y CARRERA con una relación de uno a muchos (1:N).
- **Requerimiento Funcional:** El sistema debe permitir la navegación jerárquica: Universidad > Facultad > Carrera.

### **Regla 2: Cada facultad tiene un DECANO (que es un docente de la facultad).**

- **Impacto en Diseño:** Clave Foránea (FK) id\_decano en la tabla FACULTAD que apunta a la tabla DOCENTE.
- **Requerimiento Funcional:** El sistema debe validar que el docente seleccionado como Decano pertenezca activamente a dicha facultad.

### **Regla 3: Una facultad puede tener DEPARTAMENTOS académicos.**

- **Impacto en Diseño:** Entidad DEPARTAMENTO dependiente de FACULTAD.
- **Requerimiento Funcional:** Gestión administrativa de departamentos dentro de cada facultad.

### **Regla 4: Cada carrera pertenece a UNA sola facultad.**

- **Impacto en Diseño:** FK id\_facultad en la tabla CARRERA con restricción NOT NULL.
- **Requerimiento Funcional:** Asegurar integridad referencial para evitar carreras "huérfanas".

### **Regla 5: Las carreras tienen niveles (Técnico, Tecnológico, Grado, Posgrado).**

- **Impacto en Diseño:** Atributo nivel en CARRERA con un Constraint CHECK para limitar los valores permitidos.

- **Requerimiento Funcional:** El sistema debe permitir filtrar la oferta académica según el nivel de formación.

**Regla 6: Cada carrera tiene un COORDINADOR (docente de la carrera).**

- **Impacto en Diseño:** FK id\_coordinador en la tabla CARRERA.
- **Requerimiento Funcional:** Asignar permisos administrativos y de gestión al usuario marcado como Coordinador.

**Regla 7: Una carrera puede ofrecer diferentes modalidades (Presencial, Semipresencial, Online).**

- **Impacto en Diseño:** Atributo modalidad. Si una misma carrera tiene varias modalidades simultáneas, se requiere una tabla detalle.
- **Requerimiento Funcional:** Diferenciar los paralelos y horarios disponibles según la modalidad de la carrera.

**Regla 8: Cada carrera tiene una MALLA CURRICULAR versionada por cohorte.**

- **Impacto en Diseño:** Entidad MALLA\_CURRICULAR con clave compuesta o única que incluya el año de vigencia.
- **Requerimiento Funcional:** El sistema debe asignar automáticamente la malla correcta al estudiante basándose en su fecha de ingreso.

## **Reglas sobre ASIGNATURAS Y MALLAS CURRICULARES**

**Regla A.1: Cada asignatura tiene código único, nombre, número de créditos y tipo.**

- **Impacto en Diseño:** Entidad ASIGNATURA con PK codigo\_asignatura.
- **Requerimiento Funcional:** Mantener un catálogo único de todas las asignaturas ofertadas por la universidad.

**Regla A.2: Tipos de asignatura: OBLIGATORIA, OPTATIVA, LIBRE\_ELECCIÓN.**

- **Impacto en Diseño:** Constraint CHECK en la columna tipo\_asignatura.
- **Requerimiento Funcional:** Controlar la lógica de matrícula (obligatoria vs elección) según este tipo.

**Regla A.3: Una asignatura puede tener PRERREQUISITOS.**

- **Impacto en Diseño:** Relación recursiva N:M sobre la entidad ASIGNATURA (Tabla intermedia PRERREQUISITO).

- **Requerimiento Funcional:** Bloquear la matrícula de una materia si no se han aprobado sus predecesoras.

**Regla A.4: Una asignatura puede tener CORREQUISITOS.**

- **Impacto en Diseño:** Relación recursiva N:M (Tabla intermedia CORREQUISITO).
- **Requerimiento Funcional:** Obligar al estudiante a matricularse simultáneamente en materias vinculadas (ej. Teoría y Laboratorio).

**Regla A.5: El prerrequisito puede ser asignatura específica O aprobación de N créditos.**

- **Impacto en Diseño:** Atributos en la tabla PRERREQUISITO para definir el tipo\_requisito ('ASIGNATURA' o 'CREDITOS') y el valor.
- **Requerimiento Funcional:** El sistema debe ejecutar una lógica condicional compleja para validar si el estudiante cumple el requisito.

**Regla A.6: Las asignaturas OBLIGATORIAS son iguales para todos en la cohorte.**

- **Impacto en Diseño:** Asociación directa entre MALLA y ASIGNATURA.
- **Requerimiento Funcional:** Carga automática de materias sugeridas en la pantalla de matrícula.

**Regla A.7: Las asignaturas OPTATIVAS requieren elegir M de un grupo de N opciones.**

- **Impacto en Diseño:** Entidad GRUPO\_OPTATIVAS o lógica específica en el detalle de la malla.
- **Requerimiento Funcional:** Validar al momento de la graduación que se hayan cumplido los créditos optativos requeridos.

**Regla A.8: Las asignaturas de LIBRE\_ELECCIÓN pueden ser de otras carreras.**

- **Impacto en Diseño:** Relación flexible en la matrícula que permite inscribir asignaturas fuera de la malla base.
- **Requerimiento Funcional:** Permitir al estudiante buscar y seleccionar materias de otras facultades.

**Regla A.9: Una malla curricular define carrera, vigencia, semestres y asignaturas.**

- **Impacto en Diseño:** Tabla intermedia MALLA\_DETALLE relacionando Malla y Asignatura por semestre.
- **Requerimiento Funcional:** Visualización del mapa curricular dividido por semestres o niveles.

**Regla A.10: Detalle de asignatura en malla: semestre recomendado, horas teóricas/prácticas.**

- **Impacto en Diseño:** Atributos propios de la relación MALLA\_DETALLE.
- **Requerimiento Funcional:** Cálculo de horas totales por semestre para reportes académicos.

**Regla A.11: Total de créditos mínimos por nivel (Técnico 120, Grado 240, etc.).**

- **Impacto en Diseño:** Regla de negocio implementada en un Trigger o Procedimiento Almacenado.
- **Requerimiento Funcional:** Validación crítica que impide iniciar el proceso de titulación si la suma de créditos aprobados es inferior al mínimo.

**Reglas sobre PERSONAL DOCENTE**

**Regla 12: Datos del docente (código, cédula, nombres, email, fecha ingreso).**

- **Impacto en Diseño:** Entidad DOCENTE con índices únicos en cédula y email.
- **Requerimiento Funcional:** Gestión de perfiles y unicidad de identidad del personal académico.

**Regla 13: Tipos de dedicación (Tiempo Completo, Medio Tiempo, Honorarios).**

- **Impacto en Diseño:** Atributo dedicacion con Constraint CHECK.
- **Requerimiento Funcional:** Este dato es la base para calcular el límite de horas permitidas.

**Regla 14: Categoría académica (Titular, Agregado, Auxiliar).**

- **Impacto en Diseño:** Atributo categoria con Constraint CHECK.
- **Requerimiento Funcional:** Validar elegibilidad para cargos como Decano o Tutor de Tesis.

**Regla 15: Un docente puede tener MÚLTIPLES títulos registrados.**

- **Impacto en Diseño:** Entidad débil TITULO\_DOCENTE con relación 1:N hacia DOCENTE.
- **Requerimiento Funcional:** Registro detallado de la hoja de vida académica del profesor.

**Regla 16: La carga horaria se distribuye en Docencia, Investigación, Gestión y Vinculación.**

- **Impacto en Diseño:** Tabla CARGA\_HORARIA o distribución de atributos en la planificación docente.

- **Requerimiento Funcional:** Generación de reportes de distribución de actividades.

**Reglas 17-19: Porcentajes mínimos/máximos de docencia según dedicación.**

- **Impacto en Diseño:** Constraint CHECK complejo o Trigger de validación TRG\_VALIDAR\_DISTRIBUCION.
- **Requerimiento Funcional:** Emitir alertas si la planificación de un docente no cumple con los porcentajes estatutarios.

**Regla 20: La carga de docencia se mide en horas semanales por período.**

- **Impacto en Diseño:** Cálculo agregado SUM(horas\_clase) basado en la tabla OFERTA\_ASIGNATURA.
- **Requerimiento Funcional:** Monitoreo en tiempo real de la carga asignada.

**Regla 21: Un docente Tiempo Completo NO puede tener más de 20 horas de clase.**

- **Impacto en Diseño:** Trigger de restricción TRG\_MAX\_HORAS\_DOCENTE.
- **Requerimiento Funcional:** El sistema debe impedir la asignación de una nueva clase si el docente ya suma 20 horas.

**Regla 22: Un docente puede impartir asignaturas de MÚLTIPLES carreras.**

- **Impacto en Diseño:** La relación Docente-Asignatura es independiente de la carrera a la que pertenece el docente.
- **Requerimiento Funcional:** Flexibilidad en la asignación de la planta docente a nivel universidad.

**Reglas sobre ESTUDIANTES**

**Regla 23: Datos del estudiante (matrícula, cédula, nombres, email).**

- **Impacto en Diseño:** Entidad ESTUDIANTE con PK matricula.
- **Requerimiento Funcional:** Identificación única y gestión del perfil estudiantil.

**Regla 24: Un estudiante pertenece a UNA carrera principal.**

- **Impacto en Diseño:** FK id\_carrera en la tabla ESTUDIANTE.
- **Requerimiento Funcional:** Agrupación de estudiantes para reportes por carrera y facultad.

**Regla 25: Estados del estudiante (Activo, Inactivo, Graduado, Suspendido).**

- **Impacto en Diseño:** Atributo estado con lista de valores definida.



- **Requerimiento Funcional:** Controlar el acceso al sistema y la elegibilidad para matricularse.

**Regla 26: Tipos de estudiante (Regular, Especial, Oyente).**

- **Impacto en Diseño:** Atributo tipo\_estudiante.
- **Requerimiento Funcional:** Aplicar reglas de negocio diferenciadas según el tipo (ej. los oyentes no tienen nota oficial).

**Reglas 27-28: Reglas de Promedio y Suspensión (Promedio < 7.0 consecutivo).**

- **Impacto en Diseño:** Procedimiento Almacenado de "Cierre de Ciclo".
- **Requerimiento Funcional:** El sistema debe cambiar automáticamente el estado a SUSPENDIDO si se cumple la condición de bajo rendimiento.

**Reglas 29-31: Gestión de BECAS (Tipos y Acumulación).**

- **Impacto en Diseño:** Relación N:M entre ESTUDIANTE y TIPO\_BECA.
- **Requerimiento Funcional:** Trigger que valide que la suma de porcentajes de beca no exceda el 100%.

**Regla 32: Registro del historial académico completo.**

- **Impacto en Diseño:** Entidad HISTORIAL\_ACADEMICO (separada o vista materializada de matrículas pasadas).
- **Requerimiento Funcional:** Generación del Kárdex o Récord Académico oficial.

**Regla 33: Cálculo del GPA acumulado.**

- **Impacto en Diseño:** Columna calculada o función de base de datos.
- **Requerimiento Funcional:** Indicador principal de rendimiento visible para el estudiante y autoridades.

**Reglas sobre GESTIÓN ACADÉMICA**

**Reglas 34-35: Periodos Académicos y Estados.**

- **Impacto en Diseño:** Entidad PERIODO\_ACADEMICO con atributo estado.
- **Requerimiento Funcional:** Controlar las fases del sistema (ej. solo permitir matrículas cuando el estado sea 'MATRICULA').

**Reglas 36-39: Oferta académica, Paralelos y Horarios.**

- **Impacto en Diseño:** Entidades OFERTA\_ASIGNATURA, PARALELO y HORARIO.

- **Requerimiento Funcional:** Módulo de planificación académica para definir cupos, docentes y aulas por grupo.

**Regla 40: NO puede haber dos paralelos en la misma aula al mismo tiempo.**

- **Impacto en Diseño:** Constraint o Trigger que verifique solapamiento de rangos de fecha/hora para un mismo id\_aula.
- **Requerimiento Funcional:** Validación física de espacios para evitar conflictos.

**Regla 41: Un docente NO puede tener dos clases simultáneas.**

- **Impacto en Diseño:** Constraint o Trigger que verifique solapamiento de horarios para un mismo id\_docente.
- **Requerimiento Funcional:** Validación de disponibilidad de personal.

**Regla 42: Un estudiante NO puede matricularse en asignaturas con cruce de horario.**

- **Impacto en Diseño:** Lógica de validación en el procedimiento de inserción de MATRICULA.
- **Requerimiento Funcional:** Impedir la inscripción si existe conflicto temporal con materias ya registradas.

**Regla 43: Proceso de Matrícula.**

- **Impacto en Diseño:** Entidad transaccional MATRICULA (Cabecera) y MATRICULA\_DETALLE.
- **Requerimiento Funcional:** Registro oficial de las asignaturas que cursará el estudiante en el periodo.

**Regla 44: Validación de Prerrequisitos en Matrícula.**

- **Impacto en Diseño:** Consulta de verificación contra el HISTORIAL\_ACADEMICO.
- **Requerimiento Funcional:** Bloqueo estricto que impide tomar materias avanzadas sin aprobar las previas.

**Reglas 45-46: Carga académica (mínima, normal y máxima condicional).**

- **Impacto en Diseño:** Validación lógica: Si GPA > 8.0 permite hasta 28 créditos, caso contrario máximo 24.
- **Requerimiento Funcional:** Control de carga académica para asegurar un avance adecuado.

**Reglas 47-49: Asistencia y Reprobación.**

- **Impacto en Diseño:** Registro de asistencia y Trigger de fin de periodo.

- **Requerimiento Funcional:** Si la asistencia es  $< 75\%$ , el sistema debe reprobar automáticamente al estudiante (Nota = 0).

## **Reglas sobre EVALUACIONES Y CALIFICACIONES**

### **Regla 50: Ponderación de Evaluaciones (30% Parcial 1, 30% Parcial 2, 40% Final).**

- **Impacto en Diseño:** Columnas específicas nota\_p1, nota\_p2, nota\_final.
- **Requerimiento Funcional:** Cálculo automático de la nota total sumando los ponderados.

### **Regla 51: Nota mínima para aprobar es 7.0.**

- **Impacto en Diseño:** Parámetro global o constante en la lógica del sistema.
- **Requerimiento Funcional:** Determinar el estado APROBADO o REPROBADO de la asignatura.

### **Reglas 52-53: Examen Supletorio (Ponderación 100%).**

- **Impacto en Diseño:** Atributo nota\_supletorio que, si existe, reemplaza a la nota final anterior.
- **Requerimiento Funcional:** Habilitar el ingreso de esta nota solo para estudiantes con promedio final  $< 7.0$ .

### **Reglas 54-56: Mejoramiento (Repetición de materia).**

- **Impacto en Diseño:** Identificación de matrícula como 'MEJORAMIENTO'.
- **Requerimiento Funcional:** Permitir cursar la materia con reglas especiales (sin costo hasta 2 intentos).

### **Regla 57: Nota máxima con Supletorio o Mejoramiento es 7.0.**

- **Impacto en Diseño:** Trigger de ajuste de nota: IF NewNota  $> 7$  THEN NewNota = 7.
- **Requerimiento Funcional:** Asegurar que la recuperación no otorgue una calificación superior a la mínima aprobatoria.

### **Regla 58: Las asignaturas reprobadas DEBEN repetirse obligatoriamente.**

- **Impacto en Diseño:** Validación de matrícula que prioriza materias reprobadas.
- **Requerimiento Funcional:** Bloquear el avance de malla si no se inscriben primero las materias pendientes.

## **Reglas sobre PROCESO DE TITULACIÓN**

### **Regla 59: Requisitos de graduación (Malla, Prácticas, Proyecto).**

- **Impacto en Diseño:** Validaciones booleanas previas a la creación del registro de titulación.
- **Requerimiento Funcional:** Verificación automática de elegibilidad del egresado.

### **Regla 60: Opciones de titulación (Tesis, Proyecto, Complexivo).**

- **Impacto en Diseño:** Atributo modalidad\_titulacion con dominio fijo.
- **Requerimiento Funcional:** Gestión de flujos de trabajo diferenciados según la opción elegida.

### **Regla 64: Etapas del proceso (Anteproyecto, Desarrollo, Defensa, etc.).**

- **Impacto en Diseño:** Tabla HISTORIAL\_ESTADO\_TITULACION para trazabilidad.
- **Requerimiento Funcional:** Workflow de aprobación paso a paso.

### **Regla 65: El tutor debe ser TITULAR o AGREGADO.**

- **Impacto en Diseño:** Validación cruzada con la categoría en la tabla DOCENTE.
- **Requerimiento Funcional:** Asegurar la calidad académica mediante la asignación de tutores calificados.

### **Regla 66: Defensa ante un TRIBUNAL de 3 docentes.**

- **Impacto en Diseño:** Tabla TRIBUNAL (Relación N:M entre Docente y Proceso de Titulación).
- **Requerimiento Funcional:** Asignación y registro de los miembros del jurado evaluador.

## **Reglas sobre PROYECTOS DE INVESTIGACIÓN**

### **Reglas 68-69: Definición y Estados de Proyectos.**

- **Impacto en Diseño:** Entidad PROYECTO\_INVESTIGACION.
- **Requerimiento Funcional:** Gestión del ciclo de vida de los proyectos de investigación.

### **Regla 70: Director y Colaboradores.**

- **Impacto en Diseño:** Relaciones con la entidad DOCENTE (Director como 1:N, Colaboradores como N:M).

- **Requerimiento Funcional:** Conformación de equipos de investigación.

#### **Reglas 72-73: Presupuesto y Ejecución.**

- **Impacto en Diseño:** Entidad PRESUPUESTO\_PROYECTO.
- **Requerimiento Funcional:** Control financiero comparando montos planificados vs. ejecutados.

#### **Regla 76: Reducción del 20% de carga docente por proyectos activos.**

- **Impacto en Diseño:** Lógica crítica que afecta el cálculo de disponibilidad docente (Regla 20).
- **Requerimiento Funcional:** Ajuste automático de la capacidad de horas clase del docente investigador.

### **Reglas sobre INFRAESTRUCTURA**

#### **Reglas 77-79: Aulas y Equipamiento.**

- **Impacto en Diseño:** Entidad AULA con atributos de capacidad y equipamiento (proyector, audio).
- **Requerimiento Funcional:** Búsqueda y asignación óptima de espacios físicos.

#### **Reglas 80-81: Inventario de Laboratorios.**

- **Impacto en Diseño:** Entidad EQUIPO\_LABORATORIO con estado (Operativo, Dañado, etc.).
- **Requerimiento Funcional:** Control de activos y gestión de mantenimiento.

#### **Reglas 82-86: Biblioteca y Préstamos.**

- **Impacto en Diseño:** Entidades LIBRO y PRESTAMO\_LIBRO.
- **Requerimiento Funcional:** Gestión de circulación bibliográfica y cálculo automático de multas por retraso.

#### **Reglas 87-88: Reserva de espacios.**

- **Impacto en Diseño:** Entidad RESERVA\_ESPACIO.
- **Requerimiento Funcional:** Administración de uso de aulas para eventos extraordinarios fuera del horario de clases.

### **ANÁLISIS DE REQUERIMIENTOS: UNA VISIÓN INTEGRAL**

El éxito de este diseño reside en la profunda comprensión de las reglas que rigen la vida universitaria. A continuación, se presenta el análisis de cómo estas reglas se transformaron en requerimientos del sistema, narrado desde una perspectiva funcional.

### **La Jerarquía Institucional**

Para modelar la universidad, fue necesario entender que no se trata de entes aislados. Las Carreras no existen en el vacío, sino que están estrictamente adscritas a una Facultad. Esta dependencia jerárquica es vital; por ejemplo, un Decano solo puede gestionar los recursos de su propia facultad. En el diseño, esto implica que el sistema debe impedir la creación de "carreras huérfanas". Si se intenta registrar una carrera sin vincularla a una facultad existente, el sistema debe rechazar la operación inmediatamente para mantener la coherencia organizacional.

### **Dinámica Curricular y Académica**

El corazón del sistema es la gestión de las asignaturas. Sin embargo, la complejidad radica en que los planes de estudio cambian con el tiempo. Por ello, hemos implementado el concepto de Malla Curricular versionada por Cohorte. Esto significa que el sistema "recuerda" qué plan de estudios estaba vigente cuando un estudiante ingresó, respetando sus derechos académicos. Adicionalmente, las asignaturas tienen una interdependencia lógica: los Prerrequisitos. El sistema no debe permitir que un estudiante avance a "Introducción a Programación III" si no ha aprobado "Teoría de Programación II". Esta validación no es opcional; es un candado de seguridad que garantiza la calidad académica.

### **Control del Personal Docente**

Más allá de almacenar nombres, el sistema actúa como un auditor de la normativa laboral. Un docente a Tiempo Completo tiene un límite físico y legal de horas de trabajo. ¿Cómo lo maneja el sistema? Si un coordinador intenta asignar una carga horaria que supere las 20 horas semanales de docencia a un profesor titular, el sistema debe detectar la violación de la regla y bloquear la asignación. De igual manera, se valida que quienes ocupen cargos directivos o sean tutores de tesis posean la categoría académica requerida (Titular o Agregado), asegurando así la excelencia en la supervisión.

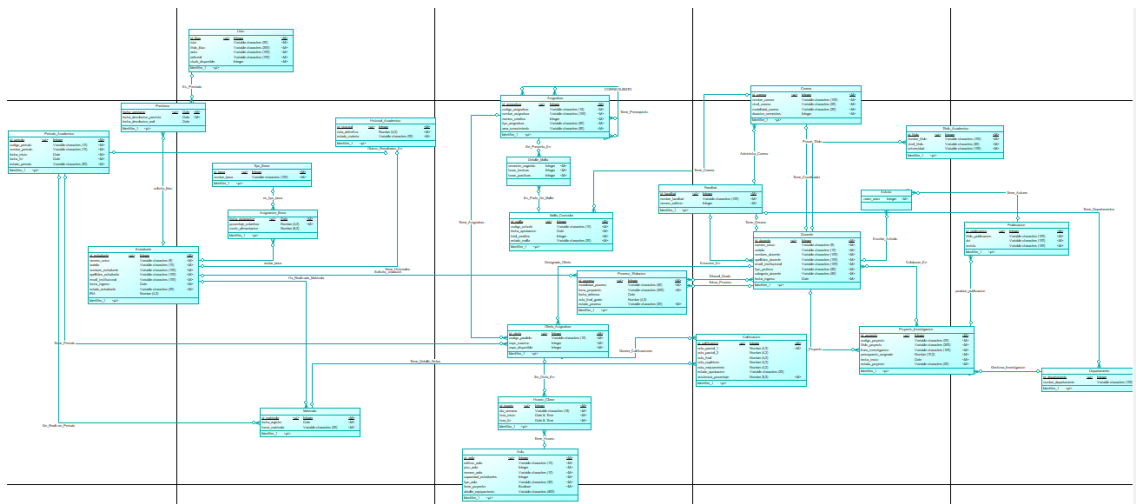
### **El Ciclo de Vida del Estudiante**

Desde que ingresa hasta que se titula, el estudiante pasa por diversos estados (Activo, Suspendido, Graduado). El sistema monitorea constantemente su rendimiento. Un aspecto crítico es la detección de riesgo: si el promedio acumulado (GPA) desciende peligrosamente, el sistema debe alertar o cambiar el estado del alumno automáticamente. Asimismo, en el módulo de Becas, se ha puesto especial énfasis en la integridad financiera, validando que la suma de los beneficios otorgados nunca supere el 100% de los costos, evitando así inconsistencias contables.

## Gestión de Infraestructura y Horarios

Finalmente, el sistema resuelve el problema clásico de la gestión universitaria: el espacio y el tiempo. La validación de horarios es estricta. El sistema cruza tres variables: Estudiante, Docente y Aula. No es posible asignar una clase si el aula está ocupada, si el docente ya está dictando otra materia, o si el estudiante tiene un choque de horario. Esta triangulación garantiza una planificación académica libre de conflictos operativos.

## Modelo Conceptual



**1. FACULTAD** Representa la unidad administrativa y académica de mayor jerarquía dentro de la universidad. Es responsable de agrupar y gestionar tanto las carreras como los departamentos académicos. Almacena información sobre su nombre oficial, ubicación física (edificio) y mantiene una relación directa con el docente que funge como Decano.

**2. CARRERA** Define los programas de estudio que ofrece la institución, clasificados por niveles (Grado, Posgrado, Técnico) y modalidades (Presencial, En Línea). Cada carrera posee su propia estructura curricular y reglas de titulación, y está subordinada administrativamente a una Facultad específica bajo la dirección de un Coordinador.

**3. DEPARTAMENTO** Constituye la unidad académica que agrupa a los docentes según sus áreas de conocimiento (ej. Matemáticas, Física, Programación). Su función principal es gestionar la planta docente y las líneas de investigación, proveyendo personal calificado para dictar las asignaturas requeridas por las diferentes carreras.

**4. PERIODO\_ACADEMICO** Entidad que delimita temporalmente los ciclos de estudio (semestres ordinarios o extraordinarios). Es fundamental para la integridad referencial del sistema, ya que controla la vigencia de las matrículas, la oferta de asignaturas y la asignación de carga horaria docente en un rango de fechas específico.

**5. AULA** Representa los espacios físicos disponibles en el campus para la impartición de clases o eventos. Se ha modelado para incluir no solo aulas teóricas sino también laboratorios, almacenando su capacidad máxima y un detalle simplificado de su equipamiento tecnológico para validar la asignación de horarios.

**6. DOCENTE** Almacena la información del personal académico contratado por la universidad. Esta entidad es central en el sistema pues maneja roles múltiples: imparte clases, dirige proyectos de investigación, tutoriza tesis y ocupa cargos administrativos (Decano/Coordinador), diferenciándolos por su tipo de dedicación y categoría.

**7. TITULO\_ACADEMICO** Entidad creada para normalizar la formación profesional de los docentes, permitiendo registrar múltiples grados académicos (PhD, Maestría, Licenciatura) por profesor. Esto es vital para validar reglas de negocio sobre la elegibilidad de tutores y escalas salariales o de categorización.

**8. ESTUDIANTE** Contiene el perfil socio-académico de los alumnos matriculados. Además de los datos personales, gestiona estados críticos (Activo, Suspendido, Graduado) y métricas de rendimiento como el Índice de Rendimiento Académico (IRA), siendo el eje central de los módulos de matrícula y titulación.

**9. TIPO\_BECA** Catálogo que define los diferentes programas de ayuda económica ofrecidos por la universidad (Mérito Académico, Socioeconómica, Deportiva). Establece las bases normativas para la asignación de beneficios, permitiendo escalar el sistema a nuevos tipos de becas sin alterar la estructura de la base de datos.

**10. ASIGNACION\_BECA** Entidad asociativa que registra la adjudicación de una beca a un estudiante específico. Es necesaria para cumplir la regla de negocio que permite a un alumno tener múltiples becas simultáneas, almacenando individualmente el porcentaje de cobertura, monto de alimentación y fecha de vigencia de cada beneficio.

**11. MALLA\_CURRICULAR** Representa el plan de estudios macro de una carrera, versionado por cohorte generacional. Permite que la universidad actualice sus programas académicos sin afectar el historial de estudiantes antiguos, quienes se rigen por la malla vigente al momento de su ingreso.

**12. ASIGNATURA** Catálogo maestro de todas las materias que se pueden dictar en la universidad. Define las propiedades académicas base como el número de créditos, tipo (Obligatoria/Optativa) y área de conocimiento, sirviendo como plantilla para la planificación de la oferta académica semestral.

**13. DETALLE\_MALLA** Entidad intermedia que vincula las asignaturas con una malla curricular específica. Su función es asignar el "semestre sugerido" y la distribución de horas (teóricas/prácticas) para una materia dentro de un plan de estudios, permitiendo



que una misma asignatura (ej. Cálculo I) se imparta en semestres distintos según la carrera.

**14. OFERTA\_ASIGNATURA** Representa la materialización de una asignatura en un periodo académico específico, conocida comúnmente como "Paralelo" o "Curso". Es la entidad integradora que vincula al docente, la materia y el periodo, gestionando los cupos disponibles para el proceso de matriculación.

**15. HORARIO\_CLASE** Entidad diseñada para gestionar la dimensión temporal y espacial de las clases. Permite asignar múltiples bloques horarios (ej. Lunes y Miércoles) a una misma oferta académica, validando automáticamente que no existan cruces de horario para docentes, estudiantes ni ocupación doble de aulas.

**16. MATRICULA** Constituye la cabecera del registro de inscripción de un estudiante en un periodo académico. Actúa como el contrato formal entre el alumno y la institución para un semestre, centralizando la información sobre la fecha de registro y la modalidad de la matrícula (Ordinaria/Extraordinaria).

**17. CALIFICACION** Almacena el detalle del rendimiento académico del estudiante por asignatura. Esta entidad desglosa las notas parciales, finales, supletorios y mejoramientos, calculando el estado final de aprobación y registrando el porcentaje de asistencia requerido por la normativa.

**18. HISTORIAL\_ACADEMICO** Repositorio histórico y permanente de las notas finales de los estudiantes. A diferencia de la matrícula activa, esta entidad almacena el resultado consolidado (Kárdex) de toda la vida universitaria del alumno, sirviendo de base para la verificación de prerrequisitos y emisión de certificados.

**19. PROYECTO\_INVESTIGACION** Gestiona la información administrativa y financiera de los proyectos científicos desarrollados en la universidad. Vincula las líneas de investigación de los departamentos con los docentes directores, controlando el presupuesto asignado y los estados de ejecución del proyecto.

**20. PUBLICACION** Registra la producción científica (artículos, libros, ponencias) resultante de los proyectos de investigación. Es fundamental para los indicadores de acreditación institucional, almacenando metadatos como DOI, nombre de la revista e indexación (Scopus/Latindex).

**21. AUTORIA** Entidad intermedia que resuelve la relación muchos a muchos entre Docentes y Publicaciones. Permite registrar a todos los coautores de un artículo y, crucialmente, definir el "orden de autoría", dato relevante para la evaluación del desempeño docente.

**22. PROCESO\_TITULACION** Administra el flujo de graduación del estudiante en sus etapas finales. Controla la modalidad elegida (Tesis, Complexivo), el tema propuesto, las fechas de defensa y la calificación final de grado, vinculando al estudiante con sus tutores y tribunales.

**23. LIBRO** Inventario bibliográfico disponible en la biblioteca universitaria. Mantiene el registro de los recursos físicos (título, autor, editorial, ISBN) y controla el stock disponible para préstamos, diferenciándose de los recursos digitales o artículos científicos.

**24. PRESTAMO** Entidad transaccional que gestiona el ciclo de vida del préstamo de libros a estudiantes. Registra las fechas de salida, fecha prevista de retorno y fecha real de devolución, permitiendo el cálculo de multas por mora y el control de activos bibliotecarios no devueltos.

Para el desarrollo del sistema, se identificaron y modelaron **28 entidades principales** que interactúan entre sí. A continuación, se describen las más relevantes alineadas con la implementación final:

1. **FACULTAD:** Unidad administrativa principal. Se relaciona con DOCENTE para la asignación del Decano.
2. **CARRERA:** Programa académico con atributos de nivel y modalidad. Depende de una Facultad.
3. **DOCENTE:** Entidad que centraliza la información del personal académico, sus títulos y su carga horaria.
4. **ESTUDIANTE:** Almacena la información personal, fecha de ingreso, carrera y el índice de rendimiento (GPA/IRA).
5. **MALLA\_CURRICULAR:** Define la versión del plan de estudios vigente para una cohorte específica.
6. **ASIGNATURA:** Catálogo de materias. Posee relaciones recursivas para manejar Prerrequisitos (TIENE\_PRERREQUISITO) y Correquisitos.
7. **PERIODO\_ACADEMICO:** Define los ciclos temporales (semestres) y controla el estado del sistema (Planificación, Matrícula, En Curso).
8. **OFERTA\_ASIGNATURA:** Entidad transaccional que representa la apertura de una materia en un periodo, con un docente y paralelo específico.
9. **MATRICULA:** Registra la inscripción del estudiante y sirve de enlace para las calificaciones.
10. **AULA:** Representa la infraestructura física, controlando capacidad y tipo de equipamiento.
11. **PROYECTO\_INVESTIGACION:** Gestiona los proyectos científicos, presupuestos y vincula a docentes y alumnos.
12. **PROCESO\_TITULACION:** Administra el flujo de graduación, desde la asignación de tutor hasta la defensa ante el tribunal.

## 5. Normalización

Para garantizar la eficiencia y evitar anomalías de datos, el modelo fue sometido a un proceso de normalización hasta la Tercera Forma Normal (3NF). A continuación, se ejemplifican 5 casos clave:

### Caso 1: Docente y Títulos Académicos

Situación Inicial: Una tabla Docente con columnas repetitivas (Titulo1, Univ1, Titulo2, Univ2).

Normalización: Se eliminaron los grupos repetitivos creando la entidad débil TITULO\_ACADEMICO, relacionada 1:N con DOCENTE. Esto permite que un profesor tenga un número ilimitado de títulos sin alterar la estructura.

### Caso 2: Malla Curricular y Asignaturas

Situación Inicial: Atributos de la asignatura (semestre sugerido) mezclados con la definición de la asignatura.

Normalización: Se separó el catálogo de ASIGNATURA (datos fijos como nombre y créditos) de su ubicación en el plan de estudios. Se creó DETALLE\_MALLA para indicar en qué semestre se dicta esa materia para una malla específica.

### Caso 3: Matrícula y Oferta Académica

Situación Inicial: Registrar el docente, horario y aula directamente en la tabla de matrícula del estudiante.

Normalización: Se generaron dependencias transitivas. Se extrajo la información a OFERTA\_ASIGNATURA (que define docente y cupo) y HORARIO\_CLASE (que define tiempo y espacio). La MATRICULA ahora solo vincula al Estudiante con la Oferta, eliminando redundancia.

### Caso 4: Proyectos de Investigación y Publicaciones

Situación Inicial: Lista de publicaciones dentro del registro del proyecto.

Normalización: Se creó la entidad PUBLICACION separada, y una tabla asociativa AUTORIA para vincular a múltiples docentes con una publicación, cumpliendo con la atomicidad de los datos.

### Caso 5: Prerrequisitos (Relación Recursiva)

Situación Inicial: Columnas "Prerrequisito\_1", "Prerrequisito\_2" en la tabla Asignatura.

Normalización: Se creó la tabla TIENE\_PRERREQUISITO, que almacena pares de IDs (Asignatura Padre - Asignatura Hija). Esto permite una flexibilidad total en el árbol de dependencias curriculares.

## **6. Modelo Racional**

El esquema lógico resultante, implementado bajo el usuario SISTEMA\_UNIVERSITARIO en Oracle, presenta una arquitectura robusta interconectada mediante claves primarias y foráneas.

Tablas Eje: ESTUDIANTE y DOCENTE son las entidades maestras que alimentan a los procesos transaccionales.

Tablas Transaccionales: MATRICULA y CALIFICACION registran el día a día académico. MATRICULA conecta a ESTUDIANTE con OFERTA\_ASIGNATURA, mientras que CALIFICACION almacena el detalle del rendimiento.

Tablas de Configuración: PERIODO\_ACADEMICO, TIPO\_BECA, AULA actúan como catálogos que parametrizan el funcionamiento del sistema.

Integridad Referencial: Se establecieron claves foráneas (FK) estrictas. Por ejemplo, la tabla CARRERA tiene una FK hacia FACULTAD; como se demostró en las pruebas, el sistema rechaza la creación de una carrera si el ID de la facultad no existe (Error ORA-02291), garantizando la consistencia de los datos.

## **7. Modelo físico**

La implementación física se realizó en Oracle Database 19c, tomando decisiones técnicas orientadas al rendimiento y la seguridad de los datos.

### **Tipos de Datos y Estándares**

Identificadores: Se utilizó NUMBER(38) para las claves primarias (ID\_DOCENTE, ID\_MATRICULA), gestionadas mediante Secuencias para asegurar unicidad y auto-incremento.

Cadenas de caracteres: Se empleó VARCHAR2 (ej. VARCHAR2(100)) para nombres, direcciones y descripciones, optimizando el espacio de almacenamiento.

Fechas: Uso del tipo DATE para campos como FECHA\_INGRESO y FECHA\_NACIMIENTO, permitiendo cálculos aritméticos de fechas (ej. cálculo de tiempo de titulación).

Numéricos de precisión: Para las calificaciones y el GPA, se definieron campos NUMBER con precisión decimal para manejar notas como "9.55".

## **Restricciones (Constraints)**

Se implementaron restricciones a nivel de base de datos para hacer cumplir las reglas de negocio, tal como se evidenció en el Documento de Pruebas:

Primary Keys (PK): En todas las tablas para garantizar la identidad de fila.

Unique Keys (UK): Aplicadas en campos críticos como CEDULA y EMAIL\_INSTITUCIONAL para evitar duplicidad de personas.

Foreign Keys (FK): Para mantener la integridad relacional (ej. Un estudiante no puede existir sin una carrera asociada).

Check Constraints: Utilizados para validar dominios finitos, como el ESTADO\_ESTUDIANTE (solo permite valores: 'ACTIVO', 'SUSPENDIDO', etc.) y los tipos de aula.

## **Índices y Optimización**

Además de los índices automáticos generados por las claves primarias, se diseñó la base de datos considerando índices en las claves foráneas para optimizar los JOINS en consultas pesadas, como el historial académico o los reportes de eficiencia terminal, mejorando los tiempos de respuesta del sistema.

## **8. Conclusiones y recomendaciones**

### **Conclusiones**

Integridad Garantizada: El diseño implementado ha logrado encapsular las complejas reglas de negocio de la Universidad Politécnica Nacional dentro de la base de datos. Las pruebas de integridad (violación de constraints) demostraron que el sistema es robusto ante intentos de inserción de datos inconsistentes, como carreras huérfanas o duplicidad de identificadores.

Soporte a la Toma de Decisiones: Las consultas SQL desarrolladas permiten extraer información estratégica valiosa, tal como se observó en los reportes de "Estudiantes en Riesgo de Deserción", "Eficiencia Terminal" y "Productividad de Investigación", cumpliendo con los requerimientos de inteligencia de negocios.

Escalabilidad: El modelo normalizado y el uso de tipos de datos adecuados en Oracle aseguran que el sistema pueda escalar para manejar el volumen de 12,000 estudiantes y crecer a futuro sin degradación significativa del rendimiento.

### **Recomendaciones**

Implementación de Auditoría de Datos: La integridad de las calificaciones es el activo más sensible de la universidad. Por ello, recomendamos ir más allá de la seguridad

básica e implementar un sistema de auditoría granular mediante Triggers de tipo AFTER UPDATE/DELETE. El objetivo es crear una "pista de auditoría" inmutable que registre no solo el cambio de una nota (ej. de 6.5 a 8.0), sino también quién realizó el cambio, cuándo ocurrió y desde qué terminal. Esto garantizará la transparencia absoluta ante posibles reclamos o auditorías académicas externas.

Automatización de Lógica Compleja (Stored Procedures): Los Procesos críticos como el "Cierre de Semestre" implican una lógica pesada: verificar promedios de 12,000 estudiantes, detectar reincidencias en reprobación y cambiar estados a 'SUSPENDIDO' masivamente. Se sugiere no ejecutar esto mediante scripts sueltos, sino encapsular esta lógica de negocio dentro de Procedimientos Almacenados (PL/SQL). Además, estos procedimientos deberían programarse como tareas automáticas (Oracle Jobs) para ejecutarse en horarios nocturnos de baja concurrencia, reduciendo así la posibilidad de error humano en tareas administrativas repetitivas.

Optimización de Alto Rendimiento con Vistas Materializadas: Los Reportes estratégicos como el "Ranking Histórico de Estudiantes por GPA" requieren procesar millones de registros de la tabla HISTORIAL\_ACADEMICO, lo cual consume excesivos recursos del servidor. Para evitar que la generación de este reporte ralentice el sistema durante horas pico (como en días de matrícula), recomendamos el uso de Vistas Materializadas. Esta técnica permitiría tener una "foto" pre-calculada del ranking que se refresque periódicamente (ej. cada noche), entregando la información instantáneamente sin impactar el rendimiento transaccional del sistema operativo.

## Referencias

- [1] B. Chancusig Espín, "Tema 3.1 Dependencias Funcionales", apuntes de clase de Fundamento de base de datos, Escuela Politécnica Nacional, Quito, Ecuador, s.f. [En línea]. Disponible: [https://aulasvirtuales.epn.edu.ec/pluginfile.php/15223503/mod\\_resource/content/4/Tema%203.1%20Dependencias%20Funcionales%20.pdf](https://aulasvirtuales.epn.edu.ec/pluginfile.php/15223503/mod_resource/content/4/Tema%203.1%20Dependencias%20Funcionales%20.pdf)
- [2] B. Chancusig Espín, "Capítulo 4: Lenguajes de consulta y programación persistente", apuntes de clase de Fundamento de base de datos, Escuela Politécnica Nacional, Quito, Ecuador, s.f. [En línea]. Disponible: [https://aulasvirtuales.epn.edu.ec/pluginfile.php/14695248/mod\\_resource/content/1/capitulo%204%20Lenguajes%20de%20consulta3%20-programacion%20persistente.pdf](https://aulasvirtuales.epn.edu.ec/pluginfile.php/14695248/mod_resource/content/1/capitulo%204%20Lenguajes%20de%20consulta3%20-programacion%20persistente.pdf)

- [3] Oracle, "Oracle Database SQL Language Reference, 19c," *Oracle Help Center*, 2019.  
[En línea]. Available: <https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/index.html>