

Bases de Datos:

Estrategia de desarrollo

Grupo:

- Nombre: DB_2025
- Número: 30

Integrantes del grupo:

Nombre	Apellido	Legajo
Ramiro	Vera	209.227-0
Nicolas	Tuma	168.792-0
Lucía	Cosenza	202.535-0
Milagros	Marchese	209.069-7

Índice:

Introducción.....	3
Criterios y suposiciones.....	3
Claves elegidas según conveniencia.....	3
Criterio para el JOIN de tablas propias y la Maestra.....	3
Interpretaciones clave.....	5
Objetos.....	5
DER:.....	8

Introducción

En el siguiente documento se detalla la estrategia utilizada por el equipo para desarrollar el trabajo práctico de la materia Bases de Datos, cursada 2C2025.

Como objetivo, se busca generar un script .sql que nos permita, en primera instancia, migrar los datos presentes en la tabla maestra entregada por la cátedra a cada grupo.

Varios datos de la misma se encuentran duplicados dado que en la tabla base se aloca toda la información necesaria para representar a los alumnos, profesores, facturas, pagos, cursos y exámenes de un sistema de gestión de alumnos. El desafío principal para esta entrega fue normalizar la tabla de forma tal que esas duplicaciones se eliminan, quedando un sistema limpio, ordenado y menos complejo para la interacción con él.

Criterios y suposiciones

Detallaremos los criterios y suposiciones que hicimos a lo largo de la resolución del trabajo, para dar contexto de las decisiones que tomamos:

Por enunciado los datos de la tabla jamás se modifican. Estos se toman tal cual se encuentran en la tabla maestra.

Claves elegidas según conveniencia

- Provincia: Como los nombres no varían casi nunca en el tiempo decidimos que el nombre de las provincias es a su vez la PK de la misma.
- Localidad: Por simplicidad, decidimos darle una clave subrogada. De esta forma las entidades que tiene localidad solo tienen como FK hacia esta entidad un atributo “codigo_localidad” que la representa.
- Para los turnos y estados de inscripción tomamos una clave natural dado que es un dato único y simple, que no tiende a cambiar en el tiempo. En el caso de los turnos es el nombre del mismo, y en el caso de los estados lo mismo.

Criterio para el JOIN de tablas propias y la Maestra

Dependiendo de la entidad, decidimos una forma de llevar a cabo el JOIN para así aprovechar los datos ya cargados en nuestras tablas, y evitar los casos donde dichos datos eran NULL. Utilizamos diferentes claves candidatas:

- **Localidad:** Dado que la localidad puede identificar unívocamente mediante su nombre y el de la provincia a la que pertenece, el JOIN con la Maestra se hace mediante estos datos. Dependiendo del caso, el nombre de la localidad proviene de la perteneciente a alguna de las siguientes entidades:
 - Profesor
 - Alumno
 - Sede
- **Profesor:** Decidimos que para identificar en el JOIN con la Maestra a un profesor, de manera unívoca, los datos necesarios son nombre, apellido, mail y dni. De esta manera si dos personas comparten mail, igualmente las podemos identificar por nombre y apellido. Además tenemos en cuenta el caso de un dni duplicado. Se podría haber utilizado también la dirección.
- **Alumno:** Para el alumno solo utilizamos el legajo dado que este debe ser necesariamente único.
- **Categoría:** Solo utilizamos el nombre dado que no había más datos a mano. Además debería ser identificable mediante él, dado que no pueden haber dos distintas con mismo nombre.
- **Sede:** En este caso utilizamos el cuit de la institución a la que pertenece, su nombre y su dirección.
- **Turno:** Solo usamos el nombre del turno, que debería ser suficiente al no poder repetirse.
- **Día:** Usamos el nombre del día como clave dado que es único.
- **Curso:** Usamos el código de curso, que necesariamente debe ser unívoco.
- **Estado de inscripción:** Usamos el nombre del estado dado que es único.
- **Instancia final:** Usamos el código del curso de ese final, la fecha y la hora en que se rinde. Esto es porque son los únicos datos que identificamos que podían ayudarnos a encontrar a cada instancia sin un identificador unívoco. El código de curso podría repetirse para la misma fecha y otro horario, y los otros dos datos podrían compartirse con otros finales tranquilamente.
- **Encuesta:** Usamos el código del curso para el que se generó, y la fecha. Esto es porque dos encuestas podrían ser de la misma fecha pero de diferentes cursos, y podrían haber varias para un mismo curso con diferentes fechas.

- **Pregunta:** Solo usamos el texto de la pregunta porque no había más datos a mano. Dos preguntas distintas no pueden tener el mismo enunciado.
- **Factura:** Utilizamos el número de factura dado que es único.
- **Medio de pago:** Usamos solo el nombre dado que es el único dato a mano, y no puede duplicarse.

Interpretaciones clave

Durante la resolución hubieron interpretaciones del enunciado que fueron clave para el desarrollo del script inicial:

- Las encuestas no son dinámicas. Todas tienen las mismas preguntas a responder, por ende en la tabla Pregunta se encuentran las 4 preguntas que conforman a todas las encuestas por igual, lo que varían son las respuestas de cada alumno y eso se ve reflejado en Encuesta, que representa a la “encuesta respondida/completada”. Tener esta tabla pregunta permite que a la larga se puedan agregar nuevas preguntas sin problema, aunque se tendría que revisar el destino de las respuestas preexistentes.
- Los módulos de los cursos, si bien tienen nombres repetidos, representan cosas distintas. Es decir, el Módulo 1 del curso A no es el mismo Módulo 1 del curso B aún llamándose igual. Esta es la razón por la que el nombre del módulo no es utilizable como unívoca.

Lo interpretamos de esta manera dado que suponemos que cada curso es destinado a una cosa diferente y que por eso se pagan (existen facturas con detalle por curso).

- Las encuestas no guardan referencia al alumno que las completa, solo al curso al que se dirigen. Esto es porque intuimos que eran anónimas, además de que las filas que representaban a las mismas no contaban con los datos del alumno (estaban en NULL).

Objetos

Para concretar la migración desarrollamos un Stored Procedure por tabla a migrar. De esta manera era más fácil probar las migraciones de forma ordenada, a medida que avanzábamos. Además, las responsabilidades estaban bien divididas al ser cada uno dedicado a cada tabla del sistema. Nos fue mucho más fácil detectar errores, que con un único Procedure.

Al final, se lleva a cabo una transacción que ejecuta los SP uno por uno. De esta manera evitamos que el sistema quede inconsistente porque alguna de las migraciones no se completó exitosamente.

Los Procedures, en su respectivo orden de ejecución, son:

DB_2025.migrar_provincia: inserta en Provincia todas las provincias únicas detectadas en la maestra (de sedes, alumnos y profesores). Usa UNION para deduplicar y filtra NULL. Evita repetir nombres y deja lista la FK de Localidad.

DB_2025.migrar_localidad: carga Localidad(nombre, provincia). Se inserta solo si hay nombre y provincia, permitiendo que localidades de igual nombre en distintas provincias convivan. Prepara la FK que consumen Sede, Alumno y Profesor.

DB_2025.migrar_institucion: inserta instituciones únicas por cuit desde la maestra. Solo trae filas con Institucion_Cuit no nulo. Deja la base para relacionar sedes.

DB_2025.migrar_sede: inserta sedes asociándolas a su Institucion (por CUIT) y a Localidad (join por provincia+localidad). Trae nombre, dirección, teléfono y mail. Necesita que Localidad ya esté cargada para resolver codigo_localidad.

DB_2025.migrar_categoria: carga el catálogo Categoria con nombres únicos presentes en cursos de la maestra. Evita categorías repetidas.

DB_2025.migrar_turno: llena la tabla Turno con los turnos distintos usados por cursos. Filtra nulos. Habilita la FK Curso.turno_curso.

DB_2025.migrar_profesor: inserta profesores deduplicandolos (nombre, apellido, dni, mail, etc.) y les resuelve codigo_localidad. El join con Localidad es por (Provincia, Localidad). Prepara la FK que luego usa Curso.

DB_2025.migrar_alumno: inserta alumnos únicos con sus datos personales y codigo_localidad. Similar a profesor: join a Localidad por (Provincia, Localidad). Es base para inscripciones, evaluaciones y facturación.

DB_2025.migrar_curso: inserta cursos y resuelve todas sus FKs: profesor, categoria, sede y turno_curso. Los joins validan que las entidades de las que depende existan antes de crear el curso. Trae además fechas, duración, precio y descripción.

DB_2025.migrar_dia: rellena la tabla Dia con valores únicos de la maestra (Curso_Dia). Filtra nulos. Es la base para la relación curso–día.

DB_2025.migrar_curso_dia: carga la tabla intermedia Curso_Dia relacionando cada curso con sus días de dictado. Requiere Curso y Dia cargados.

DB_2025.migrar_modulo: inserta módulos por curso (nombre y descripción) a partir de la maestra, solo cuando ambos campos están presentes. El join con Curso es por Curso_Codigo. Permite luego cargar evaluaciones por módulo.

DB_2025.migrar_estado_inscripcion: llena el catálogo Estado_Inscripcion con estados únicos presentes en la maestra. Filtra nulos. Normaliza los valores que luego usa Inscripcion.

DB_2025.migrar_inscripcion: inserta inscripciones vinculandose por FK a Alumno, Curso y Estado_Inscripcion. Trae fechas de inscripción y de respuesta cuando existan. Usa DISTINCT para evitar duplicados.

DB_2025.migrar_evaluacion: inserta evaluaciones “regulares” por Modulo y Alumno. Resuelve id_modulo uniendo por (Modulo_Nombre, Curso_Codigo) y valida que haya fecha de evaluación. Trae nota, instancia y presente.

DB_2025.migrar_trabajo_practico: inserta TPs por alumno/curso cuando haya fecha y nota presente. Une a Alumno y Curso por sus claves. Evita registros incompletos filtrando los nulos.

DB_2025.migrar_instancia_final: carga las mesas de final por curso con fecha_final y hora_final. Primero resuelve Curso y filtra fecha y hora nulos. Es la base para inscripciones y evaluaciones de final.

DB_2025.migrar_evaluacion_final: inserta resultados de final vinculando Profesor, Instancia_Final (por curso+fecha+hora) y Alumno. Trae nota, presente y descripción.

DB_2025.migrar_inscripcion_final: inserta inscripciones a finales asociándolas a Instancia_Final y Alumno. El match de instancia es por curso, fecha y hora. Filtra nros y fechas nulos.

DB_2025.migrar_pregunta: inserta preguntas únicas de encuesta (preg1–preg3) en Pregunta. Usa UNION para consolidar columnas paralelas. Filtra nulos para evitar las filas vacías de la Maestra.

DB_2025.migrar_encuesta: carga las Encuestas por curso, guardando la observación cuando exista. No asocia alumno porque la Maestra no lo permite.

DB_2025.migrar_respuesta: inserta las notas de encuesta como respuestas, vinculando Encuesta (por Curso+Fecha_Registro) y Pregunta (por texto). Cubre hasta cuatro columnas de nota (1–4) con UNION. Filtra nulos de nota para no crear respuestas vacías.

DB_2025.migrar_factura: guarda las cabeceras de las de facturas por alumno (número, fechas, total), excluyendo explícitamente filas que son detalle (Detalle_Factura_Importe IS NULL). Evita duplicar cabeceras por la estructura denormalizada de la maestra.

DB_2025.migrar_detalle_factura: inserta ítems de factura (curso, período año/mes, importe) para cada número de factura. Une a Factura por número y a Curso por código. Solo toma filas donde exista importe de detalle.

DB_2025.migrar_medio_pago: llena la tabla Medio_Pago con nombres únicos presentes en la maestra. Filtra nulos. Estandariza el dominio que luego referencia Pago.

DB_2025.migra_pago: guarda pagos vinculándolos a sus respectivas Facturas (por número) y al Medio_Pago (por nombre). Trae fecha e importe y filtra pagos sin importe. Asegura integridad referencial usando las FKs resueltas previamente.

DER

Finalmente, el modelo que desarrollamos fue el siguiente:

Provincia: representa las provincias para normalizar y evitar duplicados.

- provincia_nombre (NVARCHAR(255), PK): nombre único de la provincia. Se usa como clave natural y como FK desde Localidad.

Localidad: representa a las diferentes localidades asociadas a una provincia dentro del sistema. Se relaciona con Alumno, Profesor y Sede.

- codigo_localidad (BIGINT, IDENTITY, PK): clave subrogada.
- nombre (NVARCHAR(255)): nombre de la localidad (puede repetirse entre provincias).
- provincia (NVARCHAR(255), FK → Provincia.provincia_nombre): provincia a la que pertenece.

Institucion: representa a la casa de estudios dueña de las sedes.

- cuit (NVARCHAR(255), PK): CUIT como clave natural.
- nombre (NVARCHAR(255)): nombre comercial.
- razon_social (NVARCHAR(255)): razón social legal.

Sede: representa al establecimiento físico de la institución.

- id_sede (BIGINT, IDENTITY, PK): identificador de sede.
- cuit_institucion (NVARCHAR(255), FK → Institucion.cuit): dueño de la sede.
- nombre (NVARCHAR(255)): nombre/alias de la sede.
- direccion (NVARCHAR(255)): calle y número.
- codigo_localidad (BIGINT, FK → Localidad.codigo_localidad)
- telefono (NVARCHAR(255)): contacto telefónico.
- mail (NVARCHAR(255)): e-mail institucional de la sede.

Categoría: representa a los diferentes tipos de cursos que hay dentro del sistema.

- id_categoria (BIGINT, IDENTITY, PK): identificador.
- nombre_categoria (VARCHAR(255)): nombre de la categoría (Programación, Diseño, etc.).

Turno: franjas (mañana/tarde/noche, etc.) para la cursada.

- nombre (VARCHAR(255), PK): clave natural del turno.

Profesor: representa a los docentes cargados en el sistema.

- id_profesor (BIGINT, IDENTITY, PK): identificador del profesor.
- nombre (NVARCHAR(255)): nombre.
- apellido (NVARCHAR(255)): apellido.
- dni (NVARCHAR(255)): DNI como texto (mantiene ceros a izquierda/formatos).
- fecha_nacimiento (DATETIME2(6)): fecha de nacimiento.
- mail (NVARCHAR(255)): correo de contacto.
- direccion (NVARCHAR(255)): domicilio (texto).
- codigo_localidad (BIGINT, FK → Localidad.codigo_localidad): localidad normalizada.
- telefono (NVARCHAR(255)): teléfono de contacto.

Alumno: representa a los estudiantes cargados en el sistema.

- legajo (BIGINT, PK): identificador académico.
- dni (BIGINT): DNI numérico.
- nombre (VARCHAR(255)): nombre.
- apellido (VARCHAR(255)): apellido.
- fecha_nacimiento (DATETIME2(6)): fecha de nacimiento.
- mail (VARCHAR(255)): e-mail del alumno.
- direccion (VARCHAR(255)): domicilio (texto).
- codigo_localidad (BIGINT, FK → Localidad.codigo_localidad): localidad normalizada.
- telefono (VARCHAR(255)): teléfono de contacto.

Curso: representa a los diferentes cursos del sistema. Es una oferta educativa única.

- codigo_curso (BIGINT, PK): código de curso (de la fuente).
- nombre (VARCHAR(255)): nombre del curso.
- fecha_inicio (DATETIME2(6)): inicio.
- fecha_fin (DATETIME2(6)): fin.
- duracion_meses (BIGINT): duración nominal en meses.
- precio_mensual (DECIMAL(38,2)): arancel mensual.
- profesor (BIGINT, FK → Profesor.id_profesor): docente principal asignado.
- categoria (BIGINT, FK → Categoria.id_categoria): categoría temática.
- sede (BIGINT, FK → Sede.id_sede): sede de dictado.
- turno_curso (VARCHAR(255), FK → Turno.nombre): turno de cursada.
- descripcion (VARCHAR(255)): breve descripción.

Día: días de la semana que se relacionan con Curso para determinar los días de cursada.

- nombre_dia (VARCHAR(10), PK): clave natural (Lunes, Martes, etc. según estandarización).

Curso_Dia: días en los que se dicta cada curso (relación curso–día).

- codigo_curso (BIGINT, PK/FK → Curso.codigo_curso): curso.
- dia (VARCHAR(10), PK/FK → Dia.nombre_dia): día asociado.

Módulo: unidades temáticas internas de un curso.

- id_modulo (BIGINT, IDENTITY, PK): identificador del módulo.
- codigo_curso (BIGINT, FK → Curso.codigo_curso): curso dueño del módulo.
- nombre (VARCHAR(255)): nombre del módulo.
- descripcion (VARCHAR(255)): descripción/resumen.

Estado_Inscripcion: estados posibles de una solicitud de inscripción.

- estado (VARCHAR(255), PK): clave natural (Pendiente, Aprobada, Rechazada, etc.).

Inscripción: solicitudes de alumnos para cursar un curso.

- nro_inscripcion (BIGINT, PK): número único de la inscripción.
- legajo (BIGINT, FK → Alumno.legajo): solicitante.
- codigo_curso (BIGINT, FK → Curso.codigo_curso): curso solicitado.
- fecha_inscripcion (DATETIME2(6)): fecha de alta.
- estado (VARCHAR(255), FK → Estado_Inscripcion.estado): estado actual.
- fecha_respuesta (DATETIME2(6)): fecha de resolución/respuesta.

Evaluación: calificaciones de evaluaciones regulares (parciales, quizzes, etc.) dentro de módulos.

- id_evaluacion (BIGINT, IDENTITY, PK): identificador.
- id_modulo (BIGINT, FK → Modulo.id_modulo): módulo evaluado.
- alumno_legajo (BIGINT, FK → Alumno.legajo): alumno evaluado.
- nota (BIGINT): calificación obtenida.
- fecha (DATETIME2(6)): fecha de evaluación.
- instancia (BIGINT): número de instancia (p.ej., Parcial 1/2).
- presente (BIT): asistencia en la evaluación (1 presente / 0 ausente).

Trabajo_Practico: representa a los trabajos tomados por curso.

- id_trabajo_practico (BIGINT, IDENTITY, PK): identificador del TP.
- legajo_alumno (BIGINT, FK → Alumno.legajo): alumno evaluado.
- codigo_curso (BIGINT, FK → Curso.codigo_curso): curso vinculado al TP.

- nota (BIGINT): calificación del TP.
- fecha_evaluacion (DATETIME2(6)): fecha de evaluación/entrega.

Instancia_Final: representa las mesas/llamados de examen final de cada curso.

- id_instancia (BIGINT, IDENTITY, PK): identificador de la mesa.
- codigo_curso (BIGINT, FK → Curso.codigo_curso): curso al que pertenece la mesa.
- fecha_final (DATETIME2(6)): fecha.
- hora_final (VARCHAR(255)): hora (texto por heterogeneidad en la fuente).

Evaluacion_Final: representa los resultados de finales rendidos en una instancia dada.

- id_examen_final (BIGINT, IDENTITY, PK): identificador.
- id_profesor (BIGINT, FK → Profesor.id_profesor): docente a cargo/firma.
- id_instancia (BIGINT, FK → Instancia_Final.id_instancia): mesa rendida.
- legajo_alumno (BIGINT, FK → Alumno.legajo): alumno evaluado.
- nota (BIGINT): nota final.
- presente (BIT): asistencia a la mesa.
- descripcion (VARCHAR(255)): observaciones (condición, comentarios).

Inscripcion_Final: representa las inscripciones de un alumno a una mesa de final.

- numero_inscripcion (BIGINT, PK): identificador de la inscripción al final.
- legajo_alumno (BIGINT, FK → Alumno.legajo): alumno inscripto.
- id_instancia (BIGINT, FK → Instancia_Final.id_instancia): mesa seleccionada.
- fecha_inscripcion (DATETIME2(6)): fecha de registro.

Pregunta: banco de preguntas de encuestas.

- id_pregunta (BIGINT, IDENTITY, PK): identificador de la pregunta.
- pregunta_texto (VARCHAR(255)): enunciado.

Encuesta: representa a las encuestas que se generan sobre un curso tras su finalización.

- id_encuesta (BIGINT, IDENTITY, PK): identificador.
- codigo_curso (BIGINT, FK → Curso.codigo_curso): curso relevado.
- fecha_registro (DATETIME2(6)): fecha en que se registró.
- observacion (VARCHAR(255), NULL): comentario abierto opcional.

Respuesta: representa las respuestas por pregunta dentro de una encuesta.

- id_respuesta (BIGINT, IDENTITY, PK): identificador.
- id_encuesta (BIGINT, FK → Encuesta.id_encuesta): encuesta a la que pertenece.
- id_pregunta (BIGINT, FK → Pregunta.id_pregunta): pregunta respondida.
- nota_dada (BIGINT): valoración/puntaje.

Factura: representa a los comprobantes emitidos al alumno (la cabecera).

- numero_factura (BIGINT, PK): número único de factura.
- fecha_emision (DATETIME2(6)): fecha de emisión.
- fecha_vencimiento (DATETIME2(6)): fecha de vencimiento.
- legajo_alumno (BIGINT, FK → Alumno.legajo): alumno facturado.
- total (DECIMAL(18,2)): total del comprobante (cuando la fuente trae un total consolidado).

Detalle_Factura: representa a los ítems con períodos y cursos facturados dentro de una factura.

- numero_factura (BIGINT, PK/FK → Factura.numero_factura): cabecera a la que pertenece.
- codigo_curso (BIGINT, PK/FK → Curso.codigo_curso): curso facturado.
- periodo_anio (BIGINT): año del período.
- periodo_mes (BIGINT): mes del período.
- importe (DECIMAL(18,2)): importe del ítem.

PK compuesta (numero_factura, codigo_curso) evita duplicar el mismo curso en la misma factura.

Medio_Pago: representa a todos los medios de pago que ofrece la institución para sus cursos.

- id_medio (BIGINT, IDENTITY, PK): identificador del medio.
- nombre_medio (VARCHAR(255)): nombre (Tarjeta, Transferencia, Efectivo, etc.).

Pago: representa los pagos aplicados a facturas.

- id_pago (BIGINT, IDENTITY, PK): identificador del pago.
- numero_factura (BIGINT, FK → Factura.numero_factura): factura a la que aplica.
- fecha (DATETIME2(6)): fecha del pago.
- importe (DECIMAL(18,2)): monto abonado.
- medio_pago (BIGINT, FK → Medio_Pago.id_medio): medio utilizado (normalizado).

En el caso de la ubicación de los profesores, sedes y alumnos, esta se representa mediante la conjunción entre localidad y dirección.

Imagen:

