

Universidad Finis Terrae
Facultad de Ingeniería

Bases de Datos

Proyecto Semestral

Segundo Semestre 2025

Historial de Revisiones

Versión	Fecha	Descripción	Autor
1.0	23/09/2025	Versión Inicial del Documento	Fernando San Martín W.

Objetivos de Aprendizaje

El proyecto está diseñado para desarrollar las siguientes competencias técnicas y transversales:

Competencias Técnicas Específicas

- **Diseño de Modelos Relacionales:** Diseñar e implementar modelos relacionales normalizados hasta la Tercera Forma Normal (3FN) para resolver problemas de negocio reales, aplicando principios de integridad referencial y optimización.
- **Implementación de Bases de Datos:** Crear estructuras de base de datos eficientes utilizando SQL, incluyendo tablas, vistas, índices y restricciones de integridad.
- **Desarrollo de Consultas SQL:** Implementar consultas SQL eficientes y bien estructuradas para operaciones CRUD, reportes analíticos y manipulación avanzada de datos.
- **Calidad de Código:** Escribir código SQL claro, eficiente y con buenas prácticas, demostrando comprensión de optimización y mantenibilidad.

Competencias Transversales

- **Gestión de Proyectos:** Gestionar un proyecto tecnológico en equipo, aplicando metodologías ágiles básicas para la distribución de tareas y seguimiento de avances.
- **Comunicación Técnica:** Comunicar resultados técnicos de forma clara y profesional mediante informes estructurados y presentaciones efectivas.
- **Trabajo Colaborativo:** Colaborar efectivamente en equipos multidisciplinarios, demostrando habilidades de coordinación y resolución de conflictos.
- **Toma de Decisiones Tecnológicas:** Evaluar y seleccionar herramientas tecnológicas apropiadas basándose en criterios técnicos y de proyecto.

Descripción del Proyecto

Selección del Caso de Estudio

En grupos de 3 o 4 personas, los estudiantes deberán seleccionar un caso de estudio centrado en un problema real susceptible de ser resuelto mediante una solución basada en bases de datos. Esta tarea es crucial para el desarrollo del proyecto, ya que una definición clara y precisa del problema y sus requerimientos es fundamental para el éxito del diseño e implementación de la base de datos.

Descripción del Problema de Negocio:

Los estudiantes deben iniciar con una descripción clara y detallada del problema de negocio. Esto incluye una identificación precisa de las necesidades que deben ser abordadas y cómo la solución de base de datos propuesta puede resolver estas necesidades. Deben considerar lo siguiente:

- **Contexto del Problema:** Describir el entorno en el que se encuentra el problema, incluyendo el tipo de negocio o servicio y los actores principales involucrados (como clientes, empleados, proveedores, etc.).
- **Necesidades y Desafíos:** Identificar específicamente qué necesidades no están siendo satisfechas con los sistemas o procesos actuales y qué desafíos se esperan superar con la implementación de la base de datos.
- **Impacto Esperado:** Describir cómo la implementación de una solución de base de datos impactará positivamente en el negocio o servicio, mejorando procesos, eficiencia, satisfacción del cliente, o cualquier otro resultado clave.

Modelado de Datos

El diseño del modelo debe empezar con la identificación y definición clara de las entidades, atributos y relaciones. Es fundamental asegurar que el modelo sea claro y lógico.

Buenas prácticas en el modelado incluyen:

- Asegurar que cada entidad tenga un identificador único (clave primaria).
- Minimizar la redundancia de datos aplicando la normalización hasta la Tercera Forma Normal (3NF) para evitar anomalías en la inserción, actualización o eliminación de datos.
- Utilizar relaciones claramente definidas entre entidades para preservar la integridad de la base de datos.

Crear consultas SQL básicas para la inserción, actualización, y eliminación de datos, así como consultas complejas para generar reportes y analizar la información almacenada.

Durante los laboratorios a realizar en el curso se verán ejemplos aplicados sobre estos tópicos, es obligatoria la asistencia. Aún así es necesario que los estudiantes sean proactivos en aprender y mejorar sus habilidades en Python, dada su relevancia para el desarrollo de la interfaz y la manipulación de la base de datos.

Documentación

Los estudiantes documentarán todo el proceso de desarrollo del proyecto. Esto incluye la documentación del análisis de requerimientos, los diseños de modelos, detalles de implementación, y las pruebas realizadas. Cada grupo debe compilar esta información en un informe final estructurado que no solo detalle los aspectos técnicos del proyecto, sino que también refleje la evolución del pensamiento y las decisiones tomadas a lo largo del proyecto.

Cronograma del Proyecto

El proyecto se desarrollará en un periodo de 7 semanas, distribuidas en etapas que permiten un avance progresivo y equilibrado. Al final de cada etapa el informe de laboratorio debe evidenciar el entregable correspondiente. Esta estructura considera la complejidad real de cada fase y permite un mejor manejo del tiempo:

- **Semanas 1-2: Definición del Problema y Modelado Conceptual**
 - Identificación y análisis detallado del problema a resolver.
 - Recopilación exhaustiva de requisitos funcionales y no funcionales.
 - Definición clara del alcance del proyecto con entregables específicos.
 - Elaboración del modelo Entidad-Relación completo y documentado.
 - **Entregable:** Documento de requisitos y modelo conceptual (Semana 2)
- **Semanas 3-4: Diseño del Modelo Físico y Normalización**
 - Transformación del modelo conceptual al modelo relacional.
 - Aplicación sistemática de normalización hasta 3FN con documentación del proceso.
 - Definición de claves, restricciones de integridad y optimizaciones.
 - **Entregable:** Modelo físico normalizado y documentación del proceso (Semana 4)
- **Semana 5: Implementación de la Base de Datos**
 - Creación de tablas, vistas, índices y restricciones en SQL.
 - Desarrollo de scripts DDL para la estructura completa.
 - Carga inicial de datos de prueba representativos.

- **Entregable:** Scripts DDL completos y datos de prueba (Semana 5)
- **Semana 6: Desarrollo de Funcionalidades Básicas**
 - Implementación de consultas SQL esenciales (CRUD operations).
 - Desarrollo de la interfaz mínima viable para operaciones básicas.
 - Pruebas de las funcionalidades realizadas.
 - **Entregable:** Aplicación Construida y Probada.
- **Semana 7: Documentación Final y Preparación de Presentación**
 - Redacción del informe final con todos los componentes.
 - Preparación de la demostración para la feria de proyectos.
 - Revisión final y ajustes de calidad.
 - **Entregable:** Todo el material del proyecto (Semana 7)

Entregables

Considerando el desarrollo progresivo del proyecto y la naturaleza del curso, se han definido los siguientes entregables que deberán ser presentados por los grupos al finalizar el semestre:

- **Documento del Proyecto Final:**
 - Documento completo en formato IEEE que incluye:
 - * Descripción detallada del problema y requisitos
 - * Modelo físico normalizado hasta 3FN
 - * Descripción de la implementación de la base de datos (DDL)
 - * Detalle de las consultas SQL utilizadas (DML)
 - * Descripción de la aplicación desarrollada
 - * Pruebas realizadas y resultados obtenidos
 - * Conclusiones y trabajo futuro
- **Aplicación Funcional:**
 - Código fuente completo de la aplicación desarrollada
 - Base de datos implementada con datos de prueba
 - Manual de usuario básico (puede incluirse en el documento final)
 - Instrucciones de instalación y ejecución
- **Exposición en la Feria de Proyectos:**
 - Demostración del sistema funcionando
 - Materiales de apoyo para la exposición (póster, presentación, etc.)
 - Participación activa de todos los integrantes del grupo

Todos los documentos entregados como parte del proyecto final deben cumplir con estándares de formato profesional y académico. Se ofrecen dos opciones para facilitar el proceso de documentación:

Formato para Informe Final

El formato a utilizar corresponde a:

- Portada con información del proyecto y equipo
- Índice con numeración de secciones
- Estructura clara de secciones (Introducción, Desarrollo, Conclusiones)
- Referencias consistentes, utilizando formato APA o IEEE según lo solicitado
- Figuras y tablas numeradas y referenciadas en el texto
- Imágenes y gráficos preferentemente en blanco y negro para facilitar la impresión y mantener uniformidad visual
- Tamaño de letra principal: 11pt o 12pt, fuente Times New Roman o equivalente
- Márgenes estándar (2.5 cm en cada lado)
- Citas textuales y referencias bibliográficas claramente diferenciadas, usando comillas o formato de bloque para citas largas
- Paginación continua en todo el documento
- Encabezados y pies de página con información relevante (nombre del proyecto, grupo, página)

Herramientas y tecnologías sugeridas

A continuación, se presenta una lista de herramientas sugeridas para el desarrollo del proyecto, junto con criterios de selección para ayudar a los estudiantes a elegir la opción más adecuada según su proyecto:

Criterios de Selección de Herramientas

Al elegir las herramientas para su proyecto, consideren:

- **Complejidad del Proyecto:** Proyectos simples pueden usar herramientas más básicas, mientras proyectos complejos pueden requerir frameworks más robustos.
- **Curva de Aprendizaje:** Consideren el tiempo disponible para aprender nuevas herramientas.
- **Requerimientos de Interfaz:** Evalúen si necesitan una interfaz web compleja o una interfaz simple de consola.
- **Integración con Base de Datos:** Verifiquen la facilidad de conexión y manipulación de datos con SQLite.

Herramientas Disponibles

- **SQLite:** Sistema de gestión de bases de datos relacional ligero, ideal para proyectos académicos y aplicaciones pequeñas. **Recomendado para:** Todos los proyectos.
- **SQLiteStudio:** Herramienta gráfica para gestión de bases de datos SQLite. **Recomendado para:** Diseño visual de bases de datos y ejecución de consultas.
- **Visual Studio Code (VSCode):** Editor de código versátil con excelente soporte para Python y SQL. **Recomendado para:** Todos los proyectos.
- **Python 3.11:** Lenguaje principal para desarrollo de la aplicación. **Recomendado para:** Todos los proyectos.

- **Streamlit:** Biblioteca para aplicaciones web simples y interactivas. **Recomendado para:** Proyectos que requieran visualización de datos o interfaces web básicas. **Curva de aprendizaje:** Baja.
- **Taipy:** Framework para aplicaciones web más complejas con componentes interactivos. **Recomendado para:** Proyectos que necesiten interfaces más sofisticadas. **Curva de aprendizaje:** Media.
- **FastHTML:** Herramienta para desarrollo web con Python. **Recomendado para:** Estudiantes con experiencia previa en desarrollo web. **Curva de aprendizaje:** Media-Alta.

Organización del Trabajo, Entregas y Apoyo en Laboratorio

Formación de equipos:

Cada grupo estará conformado por cuatro estudiantes, quienes deberán organizarse para distribuir las tareas del proyecto de forma equilibrada. Es fundamental que exista una coordinación efectiva dentro del equipo y una trazabilidad del trabajo de cada integrante.

Formato de entrega:

Todos los informes del proyecto deberán presentarse en el formato definido en este enunciado. Se espera que los documentos cumplan con los estándares de redacción, organización, citación y presentación visual exigidos por dicho formato.

Plataforma de entregas:

Las entregas deben realizarse exclusivamente a través de la plataforma institucional, en los buzones habilitados para cada fase del proyecto. No se aceptarán entregas por correo electrónico ni por canales informales. La puntualidad en las entregas será parte de la evaluación.

Apoyo en horario de laboratorio:

Durante los bloques de laboratorio se realizarán demostraciones y explicaciones breves sobre herramientas relevantes para el desarrollo del proyecto, como:

- SQLiteStudio: Herramienta para la creación y gestión de bases de datos SQLite.
- Visual Studio Code: Editor de código fuente para desarrollo de aplicaciones.
- Python 3.11: Lenguaje de programación para el desarrollo de aplicaciones web y de escritorio.
- FastHTML: Framework para la creación de páginas web dinámicas con Python.
- Taipy: Herramienta para el desarrollo de aplicaciones de Python.
- Streamlit: Biblioteca para la creación de aplicaciones web interactivas con Python.
- Faker: Biblioteca para la generación de datos falsos en Python.
- Otros recursos y herramientas según las necesidades del proyecto.
- Resolución de dudas y consultas sobre el proyecto.

Sin embargo, el objetivo principal del horario de laboratorio es el trabajo autónomo de cada equipo en su proyecto. Los estudiantes deberán avanzar en las tareas según la planificación de su grupo y resolver dudas específicas con el equipo docente cuando sea necesario.

Informe posterior a cada laboratorio

Al finalizar cada sesión de laboratorio, el equipo deberá subir un informe breve al buzón correspondiente en la plataforma. Este informe debe incluir:

- Resumen de las tareas realizadas durante la sesión.
- Dificultades encontradas y cómo se resolvieron.
- Avances en el proyecto y próximos pasos a seguir.
- Consultas o dudas para el equipo docente.

Estos informes permitirán al equipo docente hacer un seguimiento del progreso de cada grupo y brindar apoyo adicional en caso de ser necesario.

Consideraciones

El documento actual puede ser modificado durante el semestre de acuerdo a situaciones imprevistas. Toda situación no considerada debe ser consultada con el profesor, quien decidirá sobre el asunto. Además, se deben tener en cuenta las siguientes consideraciones adicionales:

- **Cumplimiento de Plazos:** Todos los entregables deben cumplir estrictamente con los plazos establecidos. Cualquier extensión de estos plazos solo se considerará bajo circunstancias excepcionales y debe ser aprobada por el profesor.
- **Integridad Académica:** Se espera que todos los trabajos presentados sean originales y de autoría propia de los estudiantes. Cualquier forma de plagio o deshonestidad académica será penalizada de acuerdo con las políticas de la institución.
- **Colaboración y Trabajo en Equipo:** Aunque se fomenta la colaboración entre los estudiantes para el aprendizaje y la discusión, cada grupo debe entregar su propio trabajo independiente, reflejando sus propias ideas y soluciones.
- **Feria de Proyectos:** La fecha final de la feria de proyectos se anunciará oportunamente previa coordinación con la dirección de la escuela.

Rúbrica de Evaluación del Proyecto

Criterio	No Cumple (1-3 pts)	Cumple (4-5 pts)	Satisfactorio (6-7 pts)	%
Trabajo en Laboratorio				
Avances Semanales	Informes incompletos, poco frecuentes o de baja calidad	Informes completos y regulares con contenido básico	Informes completos, detallados y entregados puntualmente	10%
Documentación Técnica				
Definición del Problema	Descripción vaga, sin contexto claro o necesidades mal definidas	Descripción clara del problema con contexto y necesidades básicas	Descripción precisa, completa y bien fundamentada del problema	8%
Modelo Conceptual	Modelo ER incompleto o con errores graves en entidades y relaciones	Modelo ER básico con entidades y relaciones identificadas correctamente	Modelo ER completo, claro y bien documentado con cardinalidades correctas	10%
Modelo Físico y Normalización	Modelo físico incompleto o con errores graves de integridad y normalización incorrecta con anomalías evidentes	Modelo físico básico con integridad referencial parcial y normalización parcialmente correcta	Modelo físico completo y correcto con integridad referencial adecuada y normalización completamente correcta hasta 3FN	15%
Calidad de Consultas SQL	Consultas con errores sintácticos o lógicos, sin optimización	Consultas funcionales básicas con sintaxis correcta pero sin optimización avanzada	Consultas eficientes, bien estructuradas y optimizadas para el modelo de datos	10%
Pruebas y Resultados	Sin evidencia de pruebas o resultados incompletos	Evidencia básica de pruebas con resultados parciales	Evidencia completa de pruebas exhaustivas con resultados detallados y métricas de calidad	7%
Calidad Documental	Documento con errores graves de formato, redacción y coherencia	Documento con formato adecuado, redacción aceptable y coherencia básica	Documento profesional con estructura clara, redacción excelente y coherencia total	5%
Desarrollo y Funcionalidad				
Calidad del Código	Código desorganizado, sin comentarios y con malas prácticas	Código funcional con estructura básica y algunos comentarios	Código bien estructurado, comentado y con buenas prácticas de programación	10%

Continúa en la siguiente página

Criterio	No Cumple (1-3 pts)	Cumple (4-5 pts)	Satisfactorio (6-7 pts)	%
Funcionalidad del Sistema	Sistema con errores críticos o funcionalidad incompleta	Sistema funcional con características básicas operativas	Sistema completamente funcional con todas las características implementadas y robustas	15%
Manejo de Errores	Sin manejo de errores o manejo inadecuado	Manejo básico de errores con mensajes genéricos	Manejo robusto de errores con mensajes descriptivos y recuperación adecuada	5%
Presentación y Comunicación				
Demostración en Feria	Demostración confusa, mal organizada o con errores técnicos	Demostración clara y organizada con funcionamiento básico	Demostración excelente, bien organizada y sin errores técnicos	5%
Interfaz de Usuario	Interfaz confusa, poco intuitiva o mal integrada	Interfaz básica, intuitiva y parcialmente integrada	Interfaz clara, intuitiva, estéticamente agradable y completamente integrada	5%
Participación en Exposición	Participación desigual o pasiva de los integrantes	Participación adecuada con involucramiento básico de todos	Participación equilibrada y activa de todos los integrantes demostrando conocimiento profundo	5%
Respuesta a Preguntas	Incapacidad para responder preguntas o respuestas incorrectas	Respuestas básicas a preguntas con algunos errores menores	Respuestas claras, precisas y demostrando conocimiento profundo del proyecto	5%
Puntaje Total				100%

Anexos

Ejemplo de Informe de Laboratorio

A continuación se presenta un ejemplo del formato y contenido esperado para los informes que deben ser entregados posterior a cada sesión de laboratorio:

Informe de Laboratorio - 15/10/2025

Proyecto: Sistema de Gestión de Reservas para Restaurantes

- Juan Pérez: Desarrollo del modelo conceptual y diseño de la base de datos.
- María García: Investigación de requisitos funcionales y documentación.
- Carlos López: Implementación inicial en SQLiteStudio y pruebas preliminares de

Resumen de las tareas realizadas durante la sesión:

- Se completó el modelado conceptual del sistema con las entidades principales identificadas.
- Se establecieron las relaciones entre entidades y sus cardinalidades.
- Se inició el diseño preliminar de la base de datos en SQLiteStudio Durante esta sesión de laboratorio
- El equipo se enfocó en completar el modelado conceptual del sistema de gestión de reservas para restaurantes.
- Se trabajó en la definición de las entidades principales (Clientes, Mesas, Reservas, Empleados) y sus atributos.
- Además, se establecieron las relaciones entre estas entidades, identificando las cardinalidades correspondientes.
- Se utilizó SQLiteStudio para comenzar con el diseño preliminar de la base de datos.

Dificultades encontradas y cómo se resolvieron:

- La principal dificultad fue determinar la cardinalidad correcta entre Reservas y Mesas,
- Inicialmente se consideró una relación 1:N, pero al analizar casos especiales como reservas para eventos grandes, se identificó que la relación debía ser N:M.
- Esta dificultad se resolvió mediante una discusión en equipo y la consulta con el profesor durante la sesión, lo que permitió ajustar el modelo correctamente.

Avances en el proyecto y próximos pasos a seguir:

- Se completó exitosamente el modelo entidad-relación conceptual.
- Para la próxima semana, el equipo planea transformar este modelo al modelo relacional físico y aplicar la normalización hasta la Tercera Forma Normal (3NF).
- Además, se iniciará con la creación de las tablas en SQLite basadas en el modelo físico.

Consultas o dudas para el equipo docente:

- ¿Es necesario incluir una entidad separada para manejar los diferentes estados de una reserva (confirmada, pendiente, cancelada) o basta con un atributo en la entidad Reservas?
- Para el manejo de horarios de reserva, ¿es preferible almacenar la fecha y hora en un solo campo DATETIME o separarlos en campos distintos?

Ejemplo de Informe de Casos

A continuación se entregan algunos breves ejemplos específicos de casos que ilustran cómo los conceptos y técnicas discutidos en el curso se pueden aplicar en situaciones reales.

Ejemplo de Caso de Estudio: Sistema de Reservas para Restaurantes

Contexto: Un restaurante local busca mejorar la gestión de sus reservas y la experiencia del cliente mediante la implementación de un sistema de gestión de bases de datos personalizado.

Problema: El restaurante enfrenta desafíos con su sistema de reservas actual que es manual y propenso a errores, resultando en sobre-reservas y clientes insatisfechos.

Solución Propuesta: Desarrollo de una aplicación de gestión y su base de datos de reservas que permita a los clientes reservar mesas en línea, con funciones para manejar automáticamente la disponibilidad y confirmar las reservas en tiempo real.

Impacto Esperado: Se espera que la implementación de esta solución mejore la eficiencia operativa del restaurante, reduzca los errores en las reservas y aumente la satisfacción del cliente.

Ejemplo de Caso de Estudio: Sistema de Gestión de Inventario para una Tienda Minorista

Contexto: Una tienda minorista requiere un sistema robusto para gestionar su inventario y mejorar la precisión del seguimiento de sus productos.

Problema: La tienda actualmente utiliza un método manual para el seguimiento del inventario, lo que lleva a errores frecuentes y problemas de desabastecimiento o exceso de inventario.

Solución Propuesta: Implementación de una base de datos que integre todas las operaciones de inventario, incluyendo entradas, salidas, y ajustes, con capacidades de reporte en tiempo real.

Impacto Esperado: La nueva solución debería reducir significativamente los errores de inventario, permitir un mejor planeamiento de compras y ofrecer una visión clara de la demanda de productos.

Estos ejemplos deben servir como inspiración para los estudiantes al elegir y desarrollar sus propios casos de estudio, asegurándose de que aborden problemas reales con soluciones basadas en bases de datos que pueden ser claramente demostradas y evaluadas. Los contextos y descripciones de problemas que se presenten están resumidos, es responsabilidad de cada grupo completar y detallar estos aspectos en su propio caso de estudio.