Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación


**IIC2115 – Programación como Herramienta para la Ingeniería (II/2025)**

# Laboratorio 6

## Aspectos generales

- **Objetivo:** aplicar los contenidos de bases de datos relacionales para diseñar esquemas, construir bases de datos a partir de datos disponibles, y realizar consultas que permitan analizar y extraer información relevante.
- **Entrega:** Parte 1 lunes 10/11 a las 17:30, Parte 2 domingo 16/11 a las 23:59 hrs. Archivos Python Notebook `L6_1.ipynb` y `L6_2.ipynb` con las soluciones de las partes 1 y 2 del laboratorio, respectivamente. Los archivos deben estar ubicados en la carpeta `L6` del repositorio privado. Entregas que no cumplan con esta especificación no serán corregidas.
- **Formato de entrega:** Utilice múltiples celdas de texto y código para estructurar los archivos. Todas las celdas utilizadas deben estar ejecutadas al momento de entregar el ejercicio, de modo que las salidas generadas sean visibles. **Adjunte además el historial completo de su interacción con asistentes basados en IA, en caso de existir**. Entregas que no cumplan el formato tendrán un descuento de 0,5 puntos.
- **Entregas atrasadas:** el descuento por atraso para la Parte 1 es de 1 punto cada 10 minutos o fracción. El descuento por atraso para la Parte 2 es de 1 punto por cada hora o fracción.
- **Issues:** Las discusiones en las *issues* del Syllabus que sean relevantes para el desarrollo de la evaluación, serán destacadas. Así mismo, el uso de librerías externas que solucionen aspectos fundamentales del problema no podrán ser utilizadas. Solo se podrán utilizar las que han sido aprobadas en las *issues*, previa consulta de los estudiantes.
- **Entregas con errores de sintaxis y/o que generen excepciones en todas las ejecuciones serán calificados con nota 1.0.**

## Descripción del problema

Este laboratorio tiene por objetivo construir y analizar una base de datos relacional consolidada sobre catálogos de servicios de streaming. Deberá integrar la información de múltiples plataformas, modelar su estructura, implementar el esquema correspondiente y ejecutar consultas complejas que permitan comparar y explorar patrones en la oferta de contenidos.

A lo largo del desarrollo, se espera que tome decisiones de diseño justificadas: cómo modelar las relaciones, cómo representar los datos que difieren entre plataformas y cómo abordar los campos que contienen múltiples valores o formatos inconsistentes. Todas las decisiones deben estar documentadas y respaldadas por argumentos técnicos o razonamientos basados en los datos.

## Descripción de los datos

Cada archivo CSV (uno por plataforma) contiene información sobre todos los títulos disponibles. Los archivos tienen formatos similares, con columnas como: `show_id`, `type`, `title`, `director`, `cast`, `country`, `date_added`, `release_year`, `rating`, `duration`, `listed_in`, `description`.

## Parte 1

a) Diseñe un modelo relacional que permita integrar coherentemente toda la información disponible. Defina las entidades y las relaciones que considere necesarias, especificando los atributos, las llaves y las restricciones de integridad que deben cumplirse. Justifique sus decisiones en torno a la normalización, los tipos de datos y la forma en que tratará las diferencias entre plataformas. **(0,75 ptos.)**

b) Implemente el modelo propuesto en una base de datos SQLite creada desde Python. Asegúrese de que la estructura permita representar distintos tipos de relaciones y que las restricciones definidas se cumplan efectivamente. Describa cómo verificó la integridad y consistencia de su esquema. **(0,75 ptos.)**

c) Integre los datos reales en su base de datos, resolviendo los desafíos del proceso. Explique las decisiones adoptadas para limpiar, transformar y estandarizar la información, y discuta cómo estas decisiones podrían afectar los resultados posteriores. Comente además si observa patrones o inconsistencias interesantes durante la carga, tales como la ausencia sistemática de información en ciertas plataformas. **(1,5 ptos.)**

## Parte 2

Para cada una de las siguientes preguntas, escriba una única consulta que entregue la respuesta, utilizando agentes basados en IA de la manera que prefiera (incluso para generar la consulta completa de manera automática). Una vez generada la consulta, escriba usando sus propias palabras la lógica utilizada para construirla y escriba código en Python para validar los resultados de esta, de modo que los resultados coincidan siempre entre ambas versiones. Todas las preguntas tienen el mismo puntaje.

a) Listar todos los títulos junto con la plataforma en la que aparecen y las categorías o géneros asociados, ordenando por título y plataforma, eliminando duplicados derivados de combinaciones redundantes.

b) Para cada plataforma, contar cuántos títulos corresponden a películas y cuántos a series, e indicar además (en la misma salida) el porcentaje que cada tipo representa dentro del catálogo de esa plataforma *y* la diferencia absoluta respecto del promedio global por tipo.

c) Para cada plataforma y para cada mes de incorporación al catálogo, obtener la cantidad de títulos agregados, la variación porcentual respecto del mes anterior y un promedio móvil de tres meses. Reportar también el acumulado anual al mes correspondiente.

d) Identificar los títulos disponibles en, al menos, tres plataformas distintas, informando además la primera y la última fecha de incorporación entre plataformas y el intervalo transcurrido entre ambas (en días o meses). Ordenar del intervalo más corto al más largo.

e) Estimar, por plataforma, la duración media de las películas y el número medio de temporadas de las series, junto con una medida de dispersión (por ejemplo, desviación estándar) y un indicador de valores atípicos simples (conteo de observaciones fuera de un rango intercuartílico).

f) Encontrar las o los diez directores con mayor diversidad geográfica considerando el número de países distintos en los que aparecen sus títulos, incorporando además cuántas plataformas distintas incluyen *al menos* un título de cada persona y el año más reciente en que se incorporó alguna obra suya.

g) Para cada categoría o género, determinar qué plataforma(s) posee(n) la mayor *proporción* de títulos de esa categoría respecto del tamaño total de su propio catálogo (no solo la cuenta absoluta). En caso de empate, mostrar todas las plataformas empatadas.

h) Detectar títulos cuyos metadatos difieren entre plataformas (por ejemplo, clasificaciones por edad distintas o desajustes notables de duración). Para cada caso, mostrar el conjunto de valores observados por plataforma y cuántas discrepancias únicas existen.

i) Para cada plataforma, determinar los tres países con mayor número de incorporaciones en los últimos tres años, e indicar, para cada uno, su variación respecto del trienio inmediatamente anterior. Resolver empates y ordenar por la posición (1, 2, 3) y luego por variación.

j) Listar las quince personas con más títulos distintos en el catálogo unificado, indicando también en cuántas plataformas han tenido presencia y en cuántas categorías diferentes participan. Filtrar solo aquellas que combinan *al menos dos roles* a lo largo de su carrera.

k) Medir la similitud entre catálogos de cada par de plataformas a partir de *dos* vistas: (i) conjunto de categorías y (ii) conjunto de personas. Calcular, para cada vista, un índice basado en intersección sobre unión y reportar ambas métricas lado a lado, ordenando por la media de ambas.

l) Calcular, para cada plataforma, la racha más larga de meses consecutivos con nuevas incorporaciones (islas de continuidad temporal). Para cada racha, informar su longitud, la fecha de inicio y la de término, y la cantidad total de títulos añadidos dentro de ese intervalo.

## Consideraciones sobre la metodología de trabajo

En este laboratorio no solo interesa el resultado final, sino también el proceso seguido para construir la solución. Por ello, se espera que el desarrollo de su entrega se organice a partir de una **división progresiva de cada problema en subproblemas más simples**, siguiendo un enfoque de *divide y vencerás*. La subdivisión debe reflejarse tanto en el diseño de las soluciones, como en la organización del notebook: cada celda debe ser temáticamente coherente, evitando reunir todo el código en una única celda. Todos los supuestos y simplificaciones que se utilicen deben quedar claramente explicitados en el notebook. Además, siempre que sea posible, se recomienda complementar con diagramas.

Otro componente clave es el uso de **prompts estructurados** para interactuar con herramientas de IA. Cada vez que se utilice un asistente para generar código, se debe registrar en una celda de texto el prompt utilizado, incluyendo al menos:

- Propósito y requisitos,
- Entradas y salidas,
- Restricciones y supuestos,
- Hitos intermedios,
- Convenciones.

Este nivel de detalle permite mantener claridad sobre lo que se espera del código y facilita evaluar cómo la IA fue utilizada en el proceso de desarrollo.
El único caso donde está permitido usar código generado por IA sin necesidad de dar estructura al prompt, es cuando este código es luego progresivamente modificado y/o mejorado por el estudiante mismo.

**IMPORTANTE: los prompts estructurados no debe haber sido escritos por un asistente basado en IA, deben ser completamente escritos por el estudiante que entrega el laboratorio**.

Finalmente, todo el código debe estar acompañado de **tests explícitos**. Un test, en este contexto, es simplemente un fragmento de código que demuestra que otro fragmento funciona como se espera, considerando casos no triviales. No se exige un test unitario por cada método, pero sí que se diseñen pruebas convincentes que permitan validar comportamientos relevantes. Para cada caso probado, debe quedar visible el resultado esperado y el resultado obtenido, de modo que cualquier lector pueda verificar si el comportamiento es correcto.
Es importante enfatizar que **adjuntar el historial de interacción con un asistente de IA no constituye por sí mismo una justificación válida ni suficiente**. Si bien dicho historial debe incluirse como material adicional, no sustituye en ningún caso los requisitos de formato, estructuración y testeo descritos anteriormente, los cuales son obligatorios y forman parte de la evaluación.

## Corrección

Para la corrección se revisarán los procedimientos desarrollados para responder las diferentes preguntas y cómo estos cumplen con la materia del capítulo y la metodología de trabajo antes descrita. Dado lo abierto de las preguntas, se espera que todas las respuestas incluyan análisis y visualizaciones que permitan justificar las decisiones tomadas.

## Política de Integridad Académica

**Los/as estudiantes de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los/as estudiantes que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada estudiante conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Docencia de la Escuela de Ingeniería.**

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un/a estudiante para los efectos de la evaluación de un curso debe ser hecho **individualmente** por el/la estudiante, **sin apoyo en material de terceros**. Por "trabajo" se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros.

En particular, si un/a estudiante copia un trabajo, o si a un/a estudiante se le prueba que compró o intentó comprar un trabajo, **obtendrá nota final 1.1 en el curso** y se solicitará a la Dirección de Pregrado de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral.

Por "copia" se entiende incluir en el trabajo presentado como propio, partes hechas por otra persona. En caso que corresponda a "copia" a otros estudiantes, la sanción anterior se aplicará a todos los involucrados. En todos los casos, se informará a la Dirección de Pregrado de la Escuela de Ingeniería para que tome sanciones adicionales si lo estima conveniente.

También se entiende por copia extraer contenido sin modificarlo sustancialmente desde fuentes digitales como Wikipedia o mediante el uso de asistentes inteligentes como ChatGPT, Gemini o Copilot. Se entiende que una modificación sustancial involucra el análisis crítico de la información extraída y en consecuencia todas las modificaciones y mejoras que de este análisis se desprendan. Cualquiera sea el caso, el uso de fuentes bibliográficas, digitales o asistentes debe declararse de forma explícita, y debe indicarse cómo el/la estudiante mejoró la información extraída para cumplir con los objetivos de la actividad evaluativa.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, **siempre y cuando se incluya la referencia correspondiente**.

Lo anterior se entiende como complemento al Reglamento del Estudiante de la Pontificia Universidad Católica de Chile (`https://registrosacademicos.uc.cl/reglamentos/estudiantiles/`). Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.

**Compromiso del Código de Honor**

Este curso suscribe el Código de Honor establecido por la Universidad, el que es vinculante. Todo trabajo evaluado en este curso debe ser propio. En caso que exista colaboración permitida con otros/as estudiantes, el trabajo deberá referenciar y atribuir correctamente dicha contribución a quien corresponda. Como estudiante es un debe conocer el Código de Honor (`https://www.uc.cl/codigo-de-honor/`).