
 <b>Universidad de los Andes</b> Facultad de Ingeniería	Ingeniería de Sistemas y Computación <b>Pregrado</b> ISIS2304 – Sistemas Transaccionales	
---	--	---

## PROYECTO DEL CURSO – ENTREGA 1 – IMPLEMENTACIÓN

### OBJETIVOS

- Desarrollar habilidades en el proceso de generación de un modelo de datos relacional a partir de un modelo conceptual
- Incorporar elementos de calidad del modelo de datos, con respecto a integridad de la información.
- Implementar una aplicación informática de mediana complejidad que involucre bases de datos relacionales.

### HERRAMIENTAS

- Para la documentación del modelo UML puede utilizar GenMyModel u otro
- Para la documentación del modelo E/R puede utilizar Oracle DataModeler
- Para la documentación del modelo relacional debe utilizar la plantilla Excel dispuesta en BN
- Oracle 12c disponible en la infraestructura computacional del Departamento
- SQL Developer, disponible en los laboratorios del Departamento y disponible para descarga en el sitio Web de Oracle
- Debe usar el IDE y framework de desarrollo indicados en el proyecto guiado de Coursera

### CASO DE ESTUDIO

La iteración se desarrolla utilizando el caso **AlpesCab**, que se encuentra en Bloque Neón (BN) - Contenidos.

### ARQUITECTURA DE LA APLICACIÓN

Disponible en la documentación y archivos de base disponibles en BN.

### ACTIVIDADES A DESARROLLAR: DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN

Revisen el caso de estudio propuesto. Identifiquen los elementos fundamentales que hacen parte del negocio que se describe: entidades de negocio, funcionalidades principales, y reglas de negocio.

1. **(15%) Análisis y revisión del modelo de datos.** Analicen la retroalimentación dada por parte del monitor en la Entrega 1 - diseño. A partir de esto:
  - a. Ajusten el modelo conceptual UML e indiquen cuáles clases del modelo son actualizadas o creadas en esta entrega.
  - b. Actualicen el modelo relacional de la BD usando las reglas de transformación vistas en clase. Este debe quedar documentado en el formato Excel disponible para ello en BN.
  - c. Verifiquen que cada relación del modelo esté en la Forma Normal de Boyce-Codd (BCNF). De no estarlo, realicen los ajustes necesarios hasta cumplir con esta forma normal y documente dichos cambios.
  - d. Anexen la documentación y modelos de este entregable.
2. **(20%) Implementación del esquema de la base de datos**
  - e. Utilizando SQL Developer, cree en Oracle las tablas correspondientes al modelo de datos, con sus restricciones de integridad. Para ello, utilicen archivos de scripts con las sentencias SQL correspondientes

- (archivos .sql, generados por SQL Developer). Siga los estándares de nombramiento y buenas prácticas de definición de restricciones.
- f. Anexe a los entregables los archivos .sql utilizados.
3. **(25%) Diseño de sentencias SQL:** Escriba las sentencias SQL para los requerimientos de consulta (RFC1 a RFC4) descritos en el documento marco. En este punto deben entregar:
    - a. Para cada consulta, la documentación que describe las tablas usadas, los atributos de dichas tablas para realizar los joins necesarios de la consulta, y una corta explicación de por qué se escogió el tipo de join usado.
    - b. Los script .sql utilizados.

Tenga en cuenta que todos los requerimientos deben ser probados usando SQL con las tablas pobladas.
  4. **(25%) Desarrollo de la aplicación:** Usando Java Spring y siguiendo la arquitectura de aplicación de ejemplo (ver proyecto guiado de Coursera) que implemente los requerimientos funcionales (RF1 a RF11) y los requerimientos de consulta (RFC1 a RFC4) descritos en el documento marco. Para ello, desarrolle las clases correspondientes a:
    - a. La conexión a la base de datos
    - b. La lógica para la implementación de los RF. Estas deben contener las sentencias SQL de inserción, eliminación y modificación de datos necesarias para la implementación de los RF.
    - c. Las solicitudes SQL para las consultas descritas en los RFC.

Tenga en cuenta que para poder sacar una buena nota en este punto es necesario que los requerimientos puedan ser probados a través de Postman.
  5. **(10%) Población de tablas:** deben poblar las tablas con datos suficientes a través de Postman, SQL Developer o un script para poder verificar los requerimientos funcionales y los requerimientos funcionales de consulta. Asegúrese de incluir como mínimo:
    - a. 100 conductores diferentes
    - b. 200 pasajeros inscritos.
    - c. Una media de 5 viajes por pasajero.
  6. **(5%) Escenarios de prueba:** Realice las inserciones descritas en la siguiente sección para poder realizar pruebas para los RF y RFC a través de postman.
    - a. Implemente los escenarios de prueba para cada requerimiento funcional (ver secciones de Requerimientos funcionales del documento marco del caso de estudio (RF1-RF11), con el fin de verificar la persistencia de los datos y el manejo de las restricciones de integridad por parte el SMBD.
    - b. Incluya la traza de la ejecución de las pruebas (scripts de inserción y pantallazos).
    - c. Anexe los scripts SQL para estas operaciones y pantallazos de lo obtenido.
  7. **(10%) Bono:** Implemente una interfaz gráfica para la aplicación que permita probar los requerimientos funcionales (RF1-RF11) y los requerimientos funcionales de consulta (RFC1-RFC4). Puede utilizar cualquier tecnología para el front que requiera (p.ej Thymeleaf, Angular), sin embargo, recuerde que no hay asistencia en tecnologías diferentes a las usadas en el proyecto guiado.

## ESCENARIOS DE PRUEBA

Diseñe y desarrolle los escenarios de prueba que le permiten asegurar el correcto funcionamiento de la aplicación, así como la corrección y calidad de los datos en la base de datos.

A continuación, damos algunos ejemplos para probar la corrección y calidad del modelo en la base de datos:

- g. Pruebas de unicidad de tuplas: para una de las tablas, por ejemplo, Vehículos, trate de insertar una tupla 1 con una PK nueva. Enseguida trate de insertar una tupla 2, con la misma PK que la tupla 1.
- h. Pruebas de integridad referencial con FK: trate de insertar una tupla 1 en una tabla (por ejemplo, la tabla Vehículos), que tenga una FK cuyo valor se encuentra previamente en la tabla referenciada (por ejemplo, en la tabla Ciudades, ya que todo vehículo está registrado en una ciudad). De igual manera trate de insertar una tupla 1 que tenga una FK que no se encuentra en la tabla referenciada. Haga las pruebas de inserción para cada caso.

1. Pruebas de integridad de acuerdo con restricciones de chequeo: trate de insertar tuplas que violan las restricciones de chequeo de valores establecidas.

Para probar la corrección de la aplicación, incluya pruebas habituales de funcionalidad que revisen las reglas de negocio. No olvide considerar los roles de usuario, casos exitosos, y casos de falla.

## ENTREGA

Fecha y hora límite: **28 de Septiembre de 2025**

- En la actividad correspondiente en BN, envíe la URL del *commit* de la entrega en el repositorio GitHub asignado a su grupo.
- Contenido de la entrega (*commit* de GitHub):
  - Documento de informe, debe estar en formato `.pdf`. Debe incluir:
    - **Análisis y revisión del modelo de datos:** Los modelos UML y relacional y su descripción, indicando las modificaciones hechas. Comprobación de la forma normal de cada relación.
    - **Desarrollo de la aplicación:** Documentación de la arquitectura de la aplicación, las clases de Java Spring, así como pantallazos de las pruebas de los requerimientos funcionales y los requerimientos funcionales de consulta.
    - **Escenarios de prueba:** Traza de la ejecución del plan de pruebas (scripts de inserción y pantallazos). Balance del plan de pruebas (qué se logró hacer y qué no).
  - Nombre del usuario en Oracle en donde se encuentran las tablas del modelo físico del proyecto con el fin de comprobar que están pobladas y que durante la ejecución de las pruebas en Postman se modifican los datos de manera consistente.
  - Proyecto Java con el código fuente y la documentación del proyecto de software descrita antes.
  - Los archivos `.sql` con los scripts utilizados para la creación de los esquemas de la base de datos.
  - Los archivos o scripts utilizados para la población de todas las tablas de la base de datos.
  - Los archivos `.sql` con las sentencias utilizadas para la implementación de los requerimientos funcionales de consulta.
  - La documentación del proyecto debe quedar en una carpeta `docs`, en la estructura principal de directorios del proyecto java.
  - Archivo `readme.txt`, con todos los pasos a seguir para instalar la aplicación.

## DOCUMENTACIÓN

Tutoriales de acceso a los servicios de Oracle en el sitio Web del curso

Documentación en línea de Oracle 12c: <http://www.oracle.com/technetwork/database/enterprise-edition/documentation/index.html>