



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

## PROYECTO DEL CURSO – ENTREGA 1 DEL PROYECTO

### 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 Modeler
- Para la documentación del modelo relacional debe utilizar la plantilla 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 **Hotel de los Andes**, 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

1. Revise el caso de estudio propuesto. Identifique los elementos fundamentales que hacen parte del negocio que se describe: entidades de negocio, funcionalidades principales, y reglas de negocio.
2. **(25%)** Análisis y modelo conceptual
  - a. (10%) Proponga un modelo conceptual en UML que represente el negocio de Hotel de los Andes.
  - b. (15%) Proponga un modelo conceptual en E/R que describa las entidades del modelo de datos para la aplicación que se quiere desarrollar.
3. **(25%)** Diseño de la base de datos
  - a. Desarrolle el modelo de datos relacional correspondiente al modelo conceptual UML propuesto. Aunque está permitido utilizar Oracle modeler para ello, les sugerimos comprobar de manera manual que el proceso de transformación de UML a relacional visto en clase se cumple.
  - b. Determine el nivel de normalización en el que se encuentra su modelo. Para ello adjunte a la entrega la verificación de las formas normales vistas en clase para cada relación. Aviso: si su modelo no está en FN BC es posible que no pueda implementar todos los requerimientos de la aplicación.
  - c. Utilizando SQL Developer, cree en Oracle las tablas correspondientes al modelo de datos, con sus restricciones de integridad. Para ello, sugerimos utilizar 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.

- d. Anexe a los entregables los archivos .sql utilizados.
4. **(45%)<sup>1</sup>** Implemente los requerimientos funcionales de modificación (RF1 a RF11) solicitados en el documento marco de la siguiente manera. Evaluación detallada: 6% para cada requerimiento funcional de modificación (5) y 5% para cada requerimiento funcional de consulta (5).
- Desarrolle las clases que corresponden a la lógica y la conexión a la base de datos. Para el desarrollo, siga la arquitectura de aplicación.
  - Implemente los escenarios de prueba descritos en la sección correspondiente en este documento, para verificar el manejo de los datos persistentes y verificación de restricciones por el SMBD.
  - Incluya la traza de la ejecución de las pruebas.
- (5%)** Escenarios de prueba: realice las inserciones descritas abajo para poder realizar pruebas. Ver secciones de Requerimientos funcionales del documento marco del caso de estudio (RF1, RF2, RF3, RF4, RF5 y RF6) y Escenarios de prueba de este documento. Anexe los scripts SQL para estas operaciones.

## DESARROLLO DEL CASO DE ESTUDIO - HOTEL DE LOS ANDES

### Requerimientos funcionales de modificación:

- RF1, RF2, RF3, RF4, RF5 y RF6:** carga de las tablas con la información de base utilizando sentencias SQL INSERT. La información de base corresponde a los usuarios y la descripción del hotel.
  - RF7 y RF8:** implementar las operaciones de reserva de alojamiento o servicios por parte de un cliente: para cada operación se supone que se cumplen todas las condiciones para poderla realizar.
  - RF9 y RF10:** Implementar las operaciones de llegada de un cliente al hotel y de sus consumos, suponiendo que todas las condiciones de validez se cumplen.
  - RF11:** Implementar la operación de salida de un cliente del hotel, suponiendo que todas las condiciones de validez se cumplen
- ✓ Anexe en el entregable un archivo en texto plano (.sql), con las sentencias SQL utilizadas.

## 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:

- Pruebas de unicidad de tuplas: para una de las tablas, trate de insertar una tupla 1 con una PK conocida y nueva. Enseguida trate de insertar una tupla 2, con la misma PK que la tupla 1.
- Pruebas de integridad con FK: trate de insertar una tupla 1 que tenga una FK que se encuentra en la tabla referenciada. 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
- Pruebas de integridad de acuerdo con restricciones de chequeo: trate de insertar tuplas que violan las restricciones de chequeo 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: **1ero de octubre de 2023**

- En la actividad correspondiente en BN, envíe la URL del *commit* de la entrega en el repositorio GitHub asignado a su grupo.

<sup>1</sup> Note que el porcentaje total es 105%

- Contenido de la entrega (*commit* de GitHub):
  - Documento de informe, debe estar en formato `.pdf`. Debe incluir los modelos solicitados, resultados logrados, resultados no logrados, balance del plan de pruebas, supuestos adicionales sobre las reglas de negocio encontradas en el caso de estudio. El nombre del archivo debe seguir el mismo formato de nombramiento del zip.
  - Proyecto Java con el código fuente y la documentación del proyecto de software.
  - La documentación del proyecto debe quedar en una carpeta `docs`, en la estructura principal de directorios del proyecto java.
  - En la carpeta `docs` del proyecto Java deben estar:
    - Los modelos desarrollados, tanto en el formato fuente como en `pdf`.
    - La documentación de los requerimientos funcionales.
    - Los archivos de scripts, sentencias y pruebas indicados en el enunciado (`.sql`).
  - Archivo `readme.txt`, con todos los pasos a seguir para instalar la aplicación.
- El no seguimiento de los lineamientos de formato de entrega tiene una penalización de 0.5/5.0 puntos en la nota, por cada elemento faltante.

## 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>