
 Universidad de los Andes Facultad de Ingeniería	Ingeniería de Sistemas y Computación Pregrado ISIS2304 – Sistemas Transaccionales	
---	--	---

PROYECTO DEL CURSO - ITERACIÓN 1

OBJETIVOS

- Desarrollar habilidades en el proceso de diseño de una aplicación transaccional, a partir de la descripción de un caso de negocio
- 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, siguiendo la arquitectura de referencia definida para el curso.

METODOLOGÍA

- La iteración se realiza siguiendo los lineamientos indicados en el documento de “Herramientas, recursos, logística de trabajo y evaluación” y el documento de reglas de juego del proyecto del semestre, disponibles en Sicua+.

HERRAMIENTAS

- Para la documentación UML del modelo se recomienda la utilización de *Enterprise Architect*, disponible en los laboratorios. También puede utilizar GenMyModel (<https://api.genmymodel.com/login>), ingresando como usuario su correo de uniandes (válido hasta final de semestre)
- 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
- Máquina virtual con el ambiente de desarrollo instalado, disponible en las máquinas de los laboratorios del Departamento.
- Ejemplo completo desarrollado por los profesores del curso – Parranderos-jdo

CASO DE ESTUDIO

La iteración se desarrolla utilizando el caso **EPSAndes**, que se encuentra en Sicua+ - Contenidos.

ARQUITECTURA DE LA APLICACIÓN

Disponible en la documentación y ejemplo de Parranderos-jdo

RECOMENDACIONES

- ✓ Revise el caso de estudio en su totalidad, pues presenta el contexto particular sobre el cual se trabaja el proyecto y, de forma incremental, las iteraciones siguientes. Revise que comprende la terminología, las reglas de negocio y los servicios que allí se describen.
- ✓ Tenga en cuenta el manejo de usuarios en la arquitectura, el modelo y en la documentación.

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: roles de usuario, entidades de negocio, funcionalidades principales, reglas de negocio.
2. Documente sus resultados en un archivo que **DEBE** seguir el estándar de reportes técnicos establecido para el curso, disponible en Sicua+.
3. **(25%)** Análisis y modelo conceptual
 - a. (10%) Identifique y especifique los requerimientos funcionales que deben ser ofrecidos por la aplicación. Para cada requerimiento, usted debe indicar el nombre, dar una descripción, el usuario para quien aplica el caso, y describir los parámetros de entrada y resultados. Utilice el formato de descripción de requerimientos, disponible en Sicua+, para documentarlos.
 - b. (15%) Proponga un modelo conceptual que represente el negocio objetivo (**EPSAndes**). Utilice un diagrama de clases UML para expresarlo y documentarlo. Indique cuáles clases del modelo conceptual corresponden a entidades que deben ser persistentes y por qué. Integre las reglas de negocio al modelo conceptual, de forma que asegure la calidad del mismo.
4. **(20%)** Diseño de la base de datos
 - a. Desarrolle el modelo de datos relacional correspondiente al modelo conceptual propuesto. Incluya todos los elementos necesarios para evaluar su completitud y calidad. **Debe seguir los estándares definidos para el curso.**
 - b. Genere el modelo de datos utilizando Enterprise Architect o equivalente. Analice este modelo y compárelo con el desarrollado por Ud. en el ítem anterior. Documente las diferencias con su modelo relacional.
 - c. Utilizando SQL Developer, cree en Oracle las tablas correspondientes al modelo de datos, con sus restricciones de integridad. Para ello, utilice 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.
5. **(5%)** Poble las tablas con información de base con datos suficientes para poder realizar pruebas. Ver sección de Requerimientos funcionales del documento marco del caso de estudio (**RF1, RF2, RF3, RF4, RF5 y RF6**).
 - ✓ Anexe en el entregable un archivo en texto plano (.sql), con las sentencias SQL utilizadas.
6. **(55%)¹** **Implemente los requerimientos funcionales de modificación (RF7 a RF9) y los requerimientos funcionales de consulta (RFC1 a RFC5) solicitados en el documento marco. Debe seguir la arquitectura propuesta para el curso.** Evaluación detallada: 10% para cada requerimiento funcional de modificación (3) y 5% para cada requerimiento funcional de consulta (5).
 - a. Desarrolle las clases que corresponden a la lógica y la conexión a la base de datos. Para el desarrollo, siga la arquitectura de la aplicación mostrada en Parranderos-jdo.
 - b. 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**. Puede tomar como ejemplo la aplicación demo de Parranderos-jdo.
 - c. Incluya la traza de la ejecución de las pruebas. Puede tomar como ejemplo la aplicación demo de Parranderos-jdo.
 - ✓ Anexe en el entregable un archivo en texto plano (.sql), con las sentencias SQL utilizadas.
 - ✓ **Para RF7 a RF9:** implementar las operaciones de emisión de órdenes de servicios de salud y la prestación de los mismos: para cada operación se supone que se cumplen todas las condiciones para poderla realizar. **Mediante la ejecución de estos requerimientos, genere un conjunto de datos suficiente para evidenciar el funcionamiento correcto de los requerimientos de consulta.**

Requerimientos no funcionales:

- ✓ Por facilidad de uso, cuando se solicite información al usuario, se le debe presentar únicamente las opciones que son válidas para el usuario y el requerimiento que se está desarrollando.

¹ Note que el porcentaje total es 105%

- ✓ Se debe minimizar la posibilidad de ingreso de datos incorrectos por parte del usuario. Por ejemplo, para las operaciones de un usuario, la aplicación debe revisar cada uno de los valores que ingresa.
- ✓ Los requerimientos solicitados sobre los datos **DEBEN** ser realizados utilizando **el menor número posible de sentencias SQL**, siguiendo buenas prácticas en su desarrollo. NO debe utilizar sentencias PL/SQL o elementos no estándar del lenguaje.
- ✓ Deben garantizarse los requerimientos no funcionales propios al negocio, tal como están descritos en la descripción del caso de negocio (**RNF1 a RNF4**).

ESCENARIOS DE PRUEBA

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.

Para probar la corrección y calidad del modelo en la base de datos, incluya las siguientes pruebas:

1. Pruebas de unicidad de tuplas. Para cada tabla,
 - a. Inserte una tupla 1 con una PK conocida y nueva
 - b. Inserte una tupla 2, con la misma PK que la tupla 1
 - c. Haga las pruebas de resultados de la inserción del primer registro y del segundo registro.
2. Pruebas de integridad con FK
 - a. Inserte una tupla 1 que tenga una FK que se encuentra en la tabla referenciada
 - b. Inserte una tupla 1 que tenga una FK que no se encuentra en la tabla referenciada
 - c. Haga las pruebas de inserción para cada caso
 - d. Haga pruebas de borrado de tuplas maestras y dependientes.
3. Pruebas de integridad de acuerdo con restricciones de chequeo
 - a. Inserte tuplas que cumplen con las restricciones de chequeo establecidas
 - b. Inserte tuplas que violan las restricciones de chequeo establecidas
 - c. Haga las pruebas de inserción y borrado correspondientes.

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

ENTREGA

Fecha y hora límite: **29 de septiembre de 2019, 22:00**

Archivo de la entrega:

- Archivo comprimido en formato .zip. Nombre del archivo: **<It1_L-NN_login1_login2>.zip**. L-NN es el código asignado a su grupo y login1 y 2 son los correspondientes a los integrantes del grupo.
- Contenido del archivo:
 - 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, la documentación del proyecto de software y el bono si fue desarrollado.
 - 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 casos de uso.
 - Los archivos de scripts, sentencias (.sql) y pruebas indicados en el enunciado (.sql y .json).
 - Archivo `readme.txt`, con todos los pasos a seguir para instalar la aplicación.

- El no seguimiento de los lineamientos de formato de entrega del taller tiene una penalización de 0.5/5.0 puntos en la nota final, **por cada elemento faltante**.
- Se aplican las reglas de juego del curso con respecto a contribución al trabajo en grupo, calidad de la presentación de los entregables y seguimiento de estándares. Su incumplimiento puede conllevar penalizaciones, tal como está establecido en las reglas del curso.

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>