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

## ITERACIÓN 2 – MANEJO TRANSACCIONAL DE INFORMACIÓN

### OBJETIVOS

Integrar requerimientos funcionales y no funcionales relacionados con los aspectos ACID de una aplicación transaccional desarrollada en una arquitectura de tres niveles con manejo de persistencia en base de datos.

### PRERREQUISITOS

- La iteración se realiza siguiendo los lineamientos indicados en el documento de “Herramientas, recursos y logística de trabajo y evaluación”.
- Disponibilidad de los recursos computacionales de desarrollo del curso.
- Documento de descripción del caso de estudio.

### HERRAMIENTAS

Las definidas en la iteración 1

### REQUERIMIENTOS SOBRE EL AMBIENTE DE DESARROLLO

Los definidos en la iteración 1

### ARQUITECTURA DE LA APLICACIÓN

La definida en la iteración 1

### 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.
- ✓ Los requerimientos se deben implementar en su totalidad, tal como se describe en el enunciado. Su implementación parcial tendrá una penalización.
- ✓ 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. (2%) Análisis

Si lo requiere, ajuste el modelo del mundo (modelo conceptual: diagrama de clases UML y su respectivo modelo relacional) propuesto en la iteración 1 que representó el caso de estudio. Indique cuáles clases del modelo del mundo fueron actualizadas o creadas en esta iteración.

#### 2. (18 %) Diseño de la aplicación

- (8 %) A partir del diseño existente, analice el impacto que representa la introducción de los nuevos requerimientos y restricciones a nivel del modelo conceptual. Realice los cambios necesarios en su modelo relacional para respetar las reglas de negocio **y asegurar la calidad del mismo**. Tenga en cuenta los comentarios recibidos en la sustentación de la iteración anterior. Documente el diseño y las decisiones tomadas para crear los elementos de la base de datos que da el respaldo de persistencia a la aplicación, a partir del modelo conceptual. Incluya un listado

con las tablas generadas en la base de datos, **utilizando los estándares establecidos y buenas prácticas ilustrados en Parranderos-JDO**. Este listado, resultado de una consulta SQL, debe incluir el nombre de la tabla, el nombre y el tipo de dato de sus campos, así como los nombres de restricciones de llaves primarias, llaves foráneas y de chequeo.

- ✓ Sea claro en mencionar explícitamente los cambios relevantes entre su diseño entregado en la iteración anterior y esta.
- b. Valide que su modelo se encuentra en BCNF y que no presenta anomalías de inserción, borrado o actualización con respecto a las reglas de negocio.
- c. **(10 %)** Documente la lógica de los nuevos requerimientos a desarrollar, descritos en la sección de caso de estudio de este documento. En este punto se requiere definir los mecanismos que utiliza para garantizar las propiedades ACID del requerimiento.

### 3. **(80 %) Construcción de la aplicación**

- a. Ajuste las tablas creadas en Oracle de acuerdo a las decisiones del punto anterior.
- b. Poble las tablas con información suficiente para poder realizar pruebas.
  - i. **Diseño los datos** que le permitan verificar adecuadamente las reglas de negocio. Note que es más importante generar adecuadamente los datos, que obtener un número muy grande de ellos. Así mismo, generar datos de forma totalmente aleatoria puede no ser buena idea, pues no le permite realizar pruebas verificables de forma eficaz y eficiente.
  - ii. Puede escribir un programa de generación automática de datos acorde al diseño establecido para los mismos.
  - iii. Para la población de las tablas, si lo requiere, utilice herramientas de carga masiva como `SQLLoader` o las disponibles en `SQLDeveloper`. Consulte el tutorial disponible en la wiki del curso sobre `SQLLoader`
- c. Desarrolle o ajuste las clases correspondientes a los nuevos requerimientos, de forma que complete o modifique los requerimientos funcionales y cumpla con las restricciones de negocio. En particular, ajuste lo necesario para satisfacer las condiciones **ACID** sobre las operaciones de negocio en los nuevos requerimientos **y en los implementados en la iteración 1**.
  - i. **(10 %)** Desarrollo y/o ajustes para cumplir con los nuevos requerimientos.
  - ii. **(15 %)** Cambios y desarrollo de las transacciones
  - iii. **(15 %)** Cambios y desarrollo en el módulo de persistencia de la aplicación
- ✓ Todos los requerimientos tienen el mismo peso en la evaluación.
- d. **(40 %)** Verifique el comportamiento transaccional de los requerimientos que implican actualización, inserción o borrado de información, mediante la implementación de escenarios de prueba. Genere y documente dichos escenarios, utilizando la arquitectura y pautas de desarrollo similares al que encuentra en el ejemplo de software entregado, para la aplicación Demo. Recuerde que los escenarios de prueba le deben asegurar el correcto funcionamiento de la aplicación y la corrección y calidad de los datos en la base de datos.
  - i. Asegúrese de poner nombres dicientes para cada una de las pruebas que se realice.
  - ii. En un archivo Excel documente claramente, **para cada requerimiento**, cuáles son los datos que le permiten realizar las pruebas, tanto para los casos de terminación exitosa como los fallidos. Indique cuáles son las respuestas esperadas que corresponden a los datos de prueba.

## DESARROLLO DEL CASO DE ESTUDIO - **SUPERANDES**

La iteración se desarrolla **con base en la versión 2 del caso de estudio SuperAndes**. Recuerde que el diseño debe permitir la integración de nuevas funcionalidades y de nuevas unidades de negocio en el futuro.

### Requerimientos funcionales de modificación:

Debe realizar los requerimientos **RF12 a RF17**

- ✓ Anexe en el entregable un archivo en texto plano (`.sql`), con las sentencias SQL utilizadas.

### Requerimientos funcionales de consulta:

Debe realizar los todos los requerimientos de consulta de la iteración anterior **RFC1 a RFC6**.

Adicionalmente debe realizar los requerimientos **RFC7** y **RFC8**

- ✓ Anexe en el entregable un archivo en texto plano (`.sql`), con las sentencias SQL utilizadas.

### Requerimientos no funcionales:

Se tiene como nuevo requerimiento no funcional:

**RNF5.** Transaccionalidad

### Bono:

- ✓ (5%) Al desarrollo del requerimiento RFC9
- ✓ (15%) Al desarrollo de los requerimientos RF18 y RF19

### 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 transaccionalidad, incluya las siguientes pruebas, para cada requerimiento:

1. Pruebas de transacción exitosa. Incluir:
  - a. El estado inicial de la base de datos
  - b. Los datos involucrados en la operación transaccional solicitada
  - c. El estado final de la base de datos
2. Pruebas de transacción NO exitosa. Incluir:
  - a. El estado inicial de la base de datos
  - b. Los datos involucrados en la operación transaccional solicitada
  - c. El estado final de la base de datos

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: **4 de noviembre de 2018, 22:00**

Archivo de la entrega:

- Archivo comprimido en formato `.zip`. Nombre del archivo: **<It2\_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 del formato de entrega del taller tiene una penalización de 0.5/5.0 puntos en la nota final, **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>