



Universidad de  
**los Andes**  
Facultad de Ingeniería

Ingeniería de Sistemas y Computación  
**Pregrado**  
ISIS2304 – Sistemas Transaccionales  
<http://sistemas.uniandes.edu.co/~isis2304>  
Semestre: 2014-2



## ITERACIÓN 3 – MANEJO TRANSACCIONAL DE INFORMACIÓN

### OBJETIVOS

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

### PRERREQUISITOS

- El taller 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.

### REQUERIMIENTOS SOBRE EL AMBIENTE DE DESARROLLO

A continuación se describen requerimientos técnicos para desarrollar la aplicación.

- Lenguaje de programación: Java 1.5 o superior
- La interfaz de la aplicación debe ser web, utilizando *servlets*. Si utiliza otra tecnología deberá registrarlo en el informe a entregar y deberá garantizar su disponibilidad en los laboratorios del Departamento en el momento de la sustentación.
- Sistema de bases de datos: Oracle 10g, instalado en el servidor del curso.
- La conexión a la base de datos es a través de JDBC, que permite independizar la aplicación de la base de datos específica.
- La plataforma de ejecución es JBoss (4.03SP1 ó 4.0.5.GA). Si utiliza una versión posterior debe informarlo al inicio del taller. En cualquier caso, debe ser un ambiente disponible en los laboratorios del Departamento.

### HERRAMIENTAS

- *SQL Developer*
- *SQL Loader*
- Para la documentación UML del modelo se recomienda la utilización de *Enterprise Architect*, disponible en los laboratorios.

### RECOMENDACIONES

1. Recuerde que esta iteración se desarrolla de forma incremental, a partir de lo desarrollado en las iteraciones 1 y 2. Se incluyen nuevos requerimientos funcionales y no funcionales, que redefinen el alcance, funcionalidad, reglas de negocio y restricciones sobre la aplicación.
2. La aplicación que desarrolle debe seguir la arquitectura descrita en la iteración 2 del curso. Debe utilizar las clases *Servlets*, *Fachada*, *ConsultaDAO*, *<RespuestaConsultaValue>* de la misma forma como lo realizó en dicho taller.

## REQUERIMIENTOS PARA IMPLEMENTAR

La iteración se desarrolla utilizando el caso **ValorAndes**. Recuerde que el diseño debe permitir la integración de nuevas funcionalidades y de nuevas unidades de negocio en el futuro. Los requerimientos a desarrollar, **descritos en la versión 3 del caso de estudio**, son:

**RF12.** Recomponer portafolio inversionista.

**RF13.** Retirar intermediario

**RFC5.** Visualizar intermediarios

**RFC6.** Visualizar Oferentes

**RFC7.** Visualizar Inversionistas

**RNF4.** Transaccionalidad. Debe asegurar que los procesos de registro de movimientos de productos y pedidos se realicen de manera transaccional para garantizar la coherencia de la información y de funcionamiento de **ValorAndes**.

## ACTIVIDADES A DESARROLLAR

### 1. (2%) Análisis

Si lo requiere, ajuste el modelo del mundo (modelo conceptual: diagrama de clases UML) propuesto en la iteración 2 que representó el caso de estudio. Indique cuáles clases del modelo del mundo fueron actualizadas o creadas en esta iteración.

### 2. (20%) Diseño de la aplicación

- (7%)** 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 del taller 2. 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, disponibles en la wiki del curso (sección tutoriales)**. Este listado 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 con el taller anterior y este.
- 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.
- (13%)** 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. (78%) Construcción de la aplicación

- Ajuste las tablas creadas en Oracle de acuerdo a las decisiones del punto anterior.
- Pueble las tablas con información suficiente para poder realizar pruebas.
  - 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.
  - Puede escribir un programa de generación automática de datos acorde al diseño establecido para los mismos.
  - Para la población de las tablas utilice herramientas de carga masiva como `SQLLoader` o las disponibles en `SQLDeveloper`. Consulte el tutorial disponible en la wiki del curso sobre `SQLLoader`
- 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. Realice los cambios sobre las clases que corresponden a `servlets`, `fachada`, `<RespuestaConsultaValue>` y `ConsultaDAO`.
  - (5%)** Desarrolle o ajuste los `servlets` correspondientes a la interfaz de la aplicación.

- ii. (5%) Cambios en fachadas y clases `<RespuestaConsultaValue>`.
- iii. (23%) Cambios en `ConsultaDao`.

- ✓ Todos los requerimientos tienen el mismo valor.
- ✓ Los bonos serán acreditados con un máximo de 2% sobre el valor total de la nota.

- d. (20%) Verifique el comportamiento transaccional de los requerimientos que implican actualización, inserción o borrado de información.
- e. (15%) Diseñe un esquema de respaldo de datos. En máximo una página describa los procesos y la infraestructura que usted propone para que el negocio al cual corresponde la aplicación desarrollada pueda tener un esquema confiable de disponibilidad y recuperación de datos.
- f. (10%) Implemente los escenarios de prueba para que le permiten asegurar el correcto funcionamiento de la aplicación y la corrección y calidad de los datos en la base de datos
  - i. En un archivo Excel documente claramente, **para cada caso de uso**, 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.
- g. Automatice los procesos asociados al desarrollo y despliegue utilizando `Ant`. Haga el despliegue de la aplicación en `JBoss`.

### Bonos

- ¿Qué diferencias hay en el manejo transaccional por parte de un contenedor de aplicaciones con respecto al manejo dado por parte del programador de la aplicación? Explique claramente las ventajas y desventajas de cada uno de ellos.
- A la mejor interfaz que tenga mínimo el 80% de la aplicación funcionando.

## ENTREGA

Fecha y hora límite: **Domingo 19 de octubre de 2014 a las 22:00**

Archivo de la entrega:

- Archivo comprimido en formato `.zip`. Nombre del archivo: **`<Taller3_NN_login1_login2>.zip`**. NN es el número del grupo y login1 y 2 son los correspondientes a los integrantes del grupo.
- Contenido del archivo:
  - Proyecto Java con el código fuente, la documentación del proyecto 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. Los documentos que entregue deben estar en formato `".pdf"`.
  - Archivo `integrantes.txt`, con los nombres y códigos de estudiante de los integrantes del grupo
  - Archivo `readme.txt`, con todos los pasos a seguir para instalar y ejecutar la aplicación. Este archivo debe describir la forma de crear los ejecutables de la aplicación y de ejecutar la aplicación. Incluya un ejemplo para obtener resultados al ejecutar cada uno de los requerimientos funcionales implementados en esta entrega.
  - Archivo que describe, para cada caso de uso, los datos de prueba.
- La calificación del proyecto incluye una sustentación, los horarios serán publicados en Sicua+. Si el grupo no se presenta a la sustentación se les asignará una nota de 0. Recuerden la importancia de la puntualidad en las sustentaciones y, en general, las pautas y reglas de juego de evaluación.
- Se espera una contribución individual equitativa del trabajo, dentro de unos rangos esperados. El trabajo por debajo de este rango tiene una penalización proporcional sobre la evaluación de los entregables
- 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: Wiki del curso.

Documentación en línea de Oracle 10g: <http://sistemas.uniandes.edu.co/~manuales/oracle10g/>

Oracle® Database JDBC Developer's Guide and Reference:

[http://download.oracle.com/docs/cd/B14117\\_01/java.101/b10979/toc.htm](http://download.oracle.com/docs/cd/B14117_01/java.101/b10979/toc.htm)