

ITERACIÓN 3 – 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

- Oracle 12c o superior instalado en el servidor `fn3.oracle.virtual.uniandes.edu.co`
- SQL Developer
- SQL Loader
- Para la documentación UML del modelo se recomienda la utilización de *Enterprise Architect*, disponible en los laboratorios, o GenMyModel disponible via WEB
- Máquina virtual con el ambiente de desarrollo y ejecución, disponible en los laboratorios del Departamento.
 - **ESTA MÁQUINA VIRTUAL NO ES PERSISTENTE. ES SU RESPONSABILIDAD TENER COPIA DE SU TRABAJO ANTES DE CERRAR LA SESIÓN.**
- Ejemplo resuelto, disponible en Sicua+

REQUERIMIENTOS SOBRE EL AMBIENTE DE DESARROLLO

A continuación, se describen requerimientos técnicos para desarrollar la aplicación.

- Lenguaje de programación: Java 1.8
- Sistema de bases de datos: Oracle 12c o superior
- La plataforma de ejecución es wildfly-8.2.0.Final (versión disponible en los laboratorios). Si utiliza una tecnología diferente debe informarlo al inicio de la iteración. En cualquier caso, debe ser un ambiente disponible en los laboratorios del Departamento.
- La conexión a la base de datos debe hacerse a través de JDBC. Esto permite desacoplar la aplicación del sistema de base de datos específico.
- Todos los requerimientos se deben exponer como servicios utilizando el esquema REST.
- La capa de presentación (consumo del servicio REST) es opcional.
- Todos los servicios REST se deben consumir y responder en JSON usando Postman

RECOMENDACIONES

1. Revise el caso de estudio en su totalidad. El caso de estudio presenta el contexto particular sobre el cual se trabaja este taller y, de forma incremental, los talleres siguientes. Revise que comprende la terminología, las reglas de negocio y los servicios que allí se describen.
2. Los requerimientos se deben implementar en su totalidad, tal como se describe en el enunciado. Su implementación parcial tendrá una penalización.
3. Tenga en cuenta el manejo de usuarios en la arquitectura, modelo y documentación.

DESARROLLO

La iteración se desarrolla utilizando el caso **RotondAndes, con base en la versión 3 del caso de estudio**. Recuerde que el diseño debe permitir la integración de nuevas funcionalidades y de nuevas unidades de negocio en el futuro.

REQUERIMIENTOS FUNCIONALES DE MODIFICACIÓN

Para **RF9**, garantice que maneja correctamente el pedido de menús

Para **RF10**, garantice que maneja correctamente el servicio de los productos de los menús

Para **RF11 y RF12** debe realizar únicamente la carga de las tablas involucradas, utilizando sentencias SQL INSERT.

Desarrolle los requerimientos funcionales **RF13 a RF17**

REQUERIMIENTOS FUNCIONALES DE CONSULTA

Desarrolle los requerimientos funcionales de consulta **RFC7 y RFC8**.

REQUERIMIENTOS NO FUNCIONALES

Se tiene como nuevo requerimiento no funcional:

RNF5. Transaccionalidad. Debe asegurar que todos los requerimientos funcionales solicitados en las iteraciones 2 y 3 se realicen de manera transaccional para garantizar la coherencia de la información y de funcionamiento de **RotondAndes**.

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. (18 %) Diseño de la aplicación

- a. **(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 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, 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.
 - **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, 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 2**.
 - i. **(10 %)** Desarrollo y/o ajustes a los servicios REST para cumplir con los nuevos requerimientos.
 - ii. **(15 %)** Cambios y desarrollo de las transacciones en **RotondAndesMaster**
 - iii. **(15 %)** Cambios y desarrollo en los `DAO`.
 - ✓ Todos los requerimientos tienen el mismo peso en la evaluación.
 - ✓ Los bonos serán acreditados con un máximo de 2% sobre el valor total de la nota.
 - ✓ **Debe** enviar el archivo de Postman.
- 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, mediante archivos Postman, que le deben asegurar el correcto funcionamiento de la aplicación y la corrección y calidad de los datos en la base de datos.
 - 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.

Bonos

- (5%) ¿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 y cómo se refleja en su arquitectura de software y diseño detallado de **su RotondAndes**.
- (15%) Al desarrollo de una interfaz de usuario, que tenga mínimo el 80% de la aplicación funcionando.

ENTREGA

Fecha y hora límite: **29 de octubre de 2017 a las 22:00**

Archivo de la entrega:

- Archivo comprimido en formato `.zip`. Nombre del archivo: **<Iteracion3_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 `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 o alguno de sus integrantes 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 12c: <http://www.oracle.com/technetwork/database/enterprise-edition/documentation/index.html>

Oracle® Database JDBC Developer's Guide and Reference:

http://download.oracle.com/docs/cd/B14117_01/java.101/b10979/toc.htm