
 <b>Universidad de los Andes</b> Facultad de Ingeniería	Ingeniería de Sistemas y Computación <b>Pregrado</b> ISIS2304 – Sistemas Transaccionales <a href="http://sistemas.uniandes.edu.co/~isis2304">http://sistemas.uniandes.edu.co/~isis2304</a> Semestre: 2017-1	
---	---	---

## ITERACIÓN 2 - DISEÑO E IMPLANTACIÓN DE UNA APLICACIÓN TRANSACCIONAL

### OBJETIVOS

- Realizar el proceso de diseño de una aplicación de características transaccionales, que utiliza un sistema manejador de bases de datos relacional.
- Desarrollar los elementos de arquitectura de software que permiten integrar un SMDB relacional con un ambiente de objetos Java
- Incorporar elementos de calidad del modelo de datos, con respecto a integridad de la información.

### PRERREQUISITOS

- Esta 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+.
- Disponibilidad de los recursos computacionales de desarrollo del curso.
- Documento de descripción del caso de estudio.

### HERRAMIENTAS

- Oracle 112c instalado en el servidor ***fn3.oracle.virtual.uniandes.edu.co***
- SQL Developer, disponible en los laboratorios del Departamento.
- Para la documentación UML del modelo se recomienda la utilización de *Enterprise Architect*, disponible en los laboratorios o GenMyModel, disponible vía WEB
- Eclipse con JBoss Tools instalado, disponible en los laboratorios del Departamento.
- Pugin de Chrome Postman (Consumo REST)
- Ejemplo resuelto, disponible en la wiki del curso (sección Ejemplo Resuelto) y en Sicua+
- Plantilla de desarrollo de aplicaciones Web, disponible en Sicua+.

### REQUERIMIENTOS SOBRE EL AMBIENTE DE DESARROLLO

Los requerimientos descritos a continuación se complementan con los indicados en el enunciado del caso de estudio. Constituyen requerimientos independientes del caso de estudio a desarrollar.

- Lenguaje de programación: Java 1.8
- Sistema de bases de datos: Oracle 12c.
- 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.

## ENUNCIADO

### RECOMENDACIONES

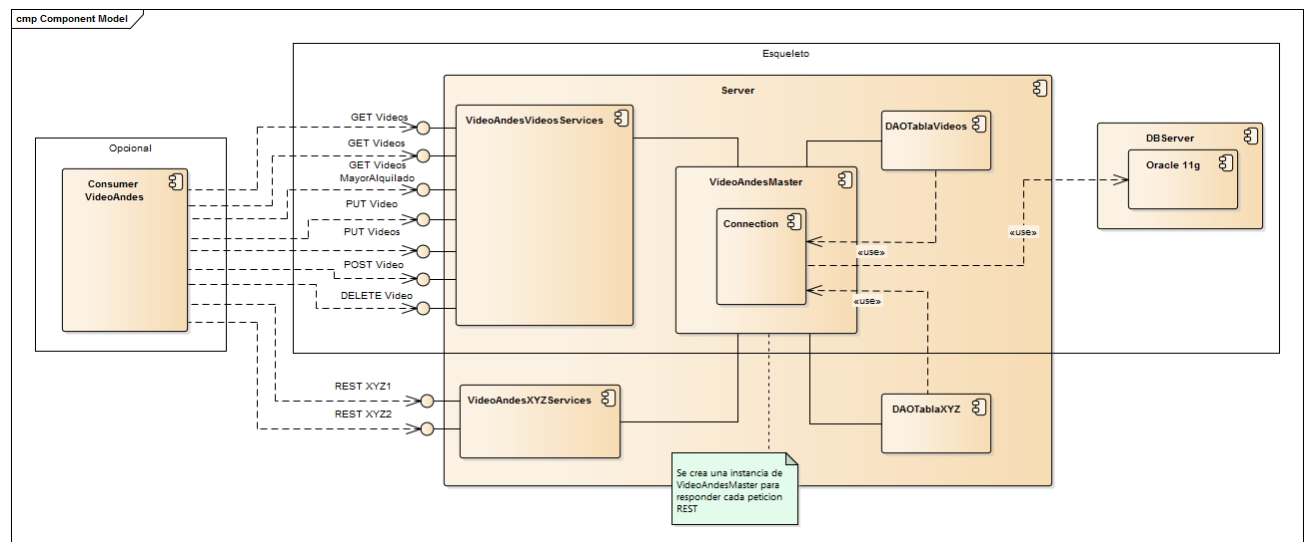
1. Revise el caso de estudio en su totalidad. El caso de estudio presenta el contexto particular sobre el cual se trabaja este taller y, de forma incremental, los talleres siguientes. Revise que comprende la terminología, las reglas de negocio y los servicios que allí se describen.
2. Los requerimientos 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.

### CASO DE ESTUDIO

La iteración se desarrolla utilizando el caso **FestivAndes**, descrito en la versión 2 del caso de estudio. Los documentos de descripción del caso se encuentran en Sicua+ - Contenidos.

### ARQUITECTURA DE LA APLICACIÓN

La aplicación debe seguir la siguiente arquitectura.



### REQUERIMIENTOS FUNCIONALES

#### Requerimientos funcionales de modificación:

Para **RF1 a RF6** debe realizar únicamente la carga de las tablas involucradas, utilizando sentencias SQL INSERT. **RF7, RF8 y RF9** deben ser implementados en su totalidad.

#### Requerimientos funcionales de consulta:

Debe realizar los requerimientos **RFC1 a RFC4**.

Los requerimientos **RFC5 y RFC6** son opcionales, se reconoce su desarrollo como bono en la iteración.

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.

## REQUERIMIENTOS NO FUNCIONALES

Se debe minimizar la posibilidad de ingreso de datos incorrectos por parte del usuario. Por ejemplo, para las operaciones de un usuario, el servidor debe revisar cada uno de los valores que entran en el JSON

Los requerimientos solicitados sobre los datos **DEBEN** ser realizados utilizando **el menor número posible de sentencias SQL**.

Deben garantizarse los requerimientos no funcionales.

## BONO

Realizar los requerimientos funcionales de consulta **RFC5 y RFC6**.

## ESCENARIOS DE PRUEBA

Desarrolle los escenarios de prueba que le permiten asegurar el correcto funcionamiento de la aplicación y la corrección y calidad de los datos en la base de datos. El proceso detallado de pruebas se explica en el siguiente ítem (Actividades a desarrollar)

En particular, 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.

## ACTIVIDADES A DESARROLLAR: CONSTRUCCIÓN DE LA APLICACIÓN

1. (15%) Ajuste los modelos conceptual y lógico a la luz de los requerimientos de la versión 2 del caso de estudio. Cree en Oracle las tablas correspondientes al modelo lógico con sus restricciones de integridad.
2. (5%) Poble las tablas (Ver C1) con datos suficientes para poder realizar pruebas.
3. (20%) 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 que se muestra en este documento. Esto significa, todas las clases Value, las respectivas clases `ConsultaDAO` y la clase `Master` que se encarga de manejar las transacciones. En la plantilla de aplicación adjunta a este enunciado, encuentra un ejemplo de la clase `ConsultaDAO` y `Master`. El desarrollo de los requerimientos funcionales tiene ponderación uniforme.
4. (15%) Desarrolle las clases REST que corresponden a la exposición de los servicios.
5. Haga el despliegue de la aplicación en wildfly.
6. (28%) Pruebe cada uno de los servicios consumiéndolos desde Postman. Debe mandar el archivo postman para consumir todos los servicios de su aplicación. (6% para cada uno de los requerimientos solicitados)
7. (17%) Implemente los escenarios de prueba descritos en la sección escenarios de prueba de este documento, para verificar el manejo de los datos persistentes y **verificación de restricciones por el SMBD**.

## 8. (10%) Implementación y pruebas del bono

### ENTREGA

Fecha y hora límite: **19 de marzo de 2017, 22:00**

Archivo de la entrega:

- Archivo comprimido en formato .zip. Nombre del archivo: **<Iteracion2\_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 la aplicación. La calificación del proyecto no incluye necesariamente una sustentación. 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.
- 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: <http://sistemas.uniandes.edu.co/~isis2304>

Documentación en línea de Oracle 12c: <http://www.oracle.com/technetwork/database/enterprise-edition/documentation/index.html>