
 Universidad de los Andes Facultad de Ingeniería	Ingeniería de Sistemas y Computación Pregrado ISIS2304 – Sistemas Transaccionales Autor: Germán Bravo C.	
---	--	---

PARRANDEROS-JDO – GUÍA DE ESTUDIO

INTRODUCCIÓN GENERAL

Este documento constituye una guía para el entendimiento del proyecto Parranderos JDO. Este entendimiento es la base para el desarrollo posterior del proyecto del curso, el cual debe seguir la misma arquitectura y aplicación de buenas prácticas.

El proyecto ilustra también cómo, a partir de un modelo conceptual, se construye el modelo relacional correspondiente, cómo se genera la base de datos en el SMD Oracle y cómo se desarrolla el programa que manipula los datos de ese modelo de negocio.

El proyecto trabaja sobre un negocio de Parranderos sobre el cual se construyeron dos aplicaciones:

- ParranderosApp, que ilustra la construcción de una aplicación transaccional simple, con requerimientos CRUD básicos
- ParranderosDemo, que ilustra, mediante demostraciones, operaciones transaccionales más avanzadas, basadas en reglas de negocio.

Los requerimientos funcionales implementados están completamente documentados en el código de la aplicación y publicados mediante javadoc.

El proyecto muestra también algunas buenas prácticas para el desarrollo de aplicaciones, tales como manejo de bitácoras, escenarios de prueba y documentación.

GUÍA PASO A PASO

1 - DESCARGAR EL PROYECTO

El proyecto se encuentra disponible en SICUA y en el repositorio del curso. Descargue el archivo parranderos-jdo-est.zip en el directorio de su preferencia

2 - DESCOMPRIMIR

En el directorio de su preferencia

3 - ABRIR EL PROYECTO EN ECLIPSE

El proyecto corre sobre las versiones de eclipse para desarrollo en Java (No corre sobre las versiones para Java EE). En los laboratorios del Departamento funciona con Eclipse Luna, Eclipse Kepler. También corre sobre Eclipse Neon y Eclipse Photon.

Ejecute el eclipse correspondiente y cargue el proyecto.

4 - COMPRENSIÓN DEL ESQUEMA DE PARRANDEROS

Abra y estudie los siguientes archivos

- data/Modelo Conceptual Parranderos.pdf – Modelo de clases para el negocio de Parranderos
- data/Esquema BD Parranderos.pdf – Modelo relacional del negocio de Parranderos
- data/EsquemaParranderos.sql – Sentencias SQL para la creación del esquema de Parranderos

5 - EJECUTAR LAS PRUEBAS DE CONEXIÓN

El proyecto tiene un conjunto de pruebas unitarias que incluyen pruebas de conexión ConexionTest.java y pruebas de funcionamiento sobre la tabla tipoBebida: src/test/java – TipoBebidaTest.java. En este punto nos concentramos en las pruebas de conexión.

Los escenarios de prueba están definidos en archivos de configuración (.json), localizados en src/main/resources/config. Estos archivos de configuración indican la unidad de persistencia deseada (base de datos) y los nombres de las tablas con las que se quiere trabajar. Los 4 escenarios son:

- TablasBD_A.json – Unidad de persistencia Parranderos y tablas con prefijo “A_”. Representan el caso normal de funcionamiento
- TablasBD_B.json – Unidad de persistencia Parranderos y tablas con prefijo “B_”. Representan el caso cuando se puede acceder a la base de datos, pero las tablas buscadas no existen: No se han creado o se crearon con nombres diferentes.
- TablasBD_ErrorDataStore.json – Unidad de persistencia ParranderosErrorDS y tablas con prefijo “X_”. Representan el caso en que la base de datos no existe o los datos de conexión no concuerdan.
- TablasBD_ErrorInvalidUser.json – Unidad de persistencia ParranderosErrorUser y tablas con prefijo “X_”. Representan el caso en que no se logra la conexión a la base de datos porque las credenciales del usuario no son válidas.

Abra el archivo ConexionTest.java y estudie las pruebas implementadas, lo mismo que los archivos de configuración.

En el estado actual del proyecto, donde no se han dado los datos de conexión, deben ser exitosas las pruebas de error de conexión y deben fallar las pruebas de funcionamiento (que suponen que hay una conexión exitosa...).

Ejecute el archivo dado como JUnit Test y compruebe el comportamiento esperado.

Las pruebas realizadas tienen una traza en la consola de Eclipse, pero también tienen una traza en las bitácoras de datanucleus (datanucleus.log y de parranderos.log). Explore el código de las pruebas e identifique la traza de ejecución en los logs correspondientes.

Estas pruebas pueden luego ser extendidas para las otras tablas del proyecto de parranderos.

6 - CONEXIÓN A LA BASE DE DATOS

Los datos de conexión se deben especificar en el archivo src/main/resources/META-INF/persistence.xml. Este archivo contiene las unidades de persistencia a las cuales hacen referencia los archivos json de configuración.

Abra el archivo persistence.xml, estudie las unidades de persistencia definidas y modifique la unidad de persistencia “**Parranderos**” de la siguiente forma:

- En la propiedad “javax.jdo.option.ConnectionUserName” ponga como valor el nombre de usuario que le fue enviado por correo.
- En la propiedad “javax.jdo.option.ConnectionPassword” ponga como valor la contraseña que le fue enviada por correo.

Note que los datos de conexión especificados para esa unidad de persistencia son los que le fueron enviados por correo.

NO MODIFIQUE LAS OTRAS UNIDADES DE PERSISTENCIA

7 - CREE LAS TABLAS (EL ESQUEMA) DE LA BASE DE DATOS

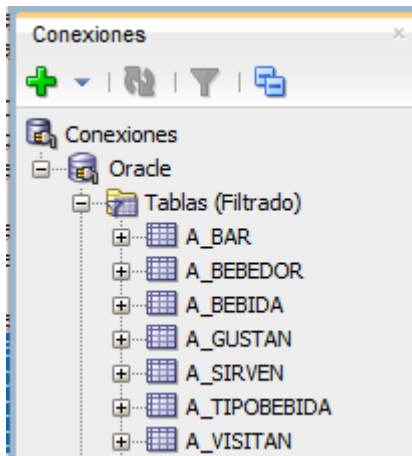
Este proceso se realiza con SQLDeveloper, con el contenido del archivo data/EsquemaParranderos.sql.

- Abra el archivo data/EsquemaParranderos.sql y estudie la creación de tablas y la definición de restricciones

- Abra sqlDeveloper
- Adicione una nueva conexión con los datos que le fueron enviados por correo. El nombre de la conexión lo puede definir a su gusto. **Dado que está en máquinas públicas NO HABILITE LA OPCIÓN DE GUARDAR CONTRASEÑA.**
- Cuando se realiza la conexión, se abre una pestaña SQL. Copie el contenido del archivo EsquemaParranderos.sql en esta ventana y ejecútelo como un script:



- El resultado de esta ejecución la puede observar en el panel de Conexiones:



- Explore las tablas creadas, haciendo clic en el + y también en el nombre de la tabla, que muestra el detalle en el panel de la derecha. Explore las pestañas disponibles.

8 - EJECUTAR NUEVAMENTE LAS PRUEBAS DE CONEXIÓN

Como en este momento ya se cumplen las condiciones de funcionamiento de la aplicación, todas las pruebas deben pasar.

9 - COMPRENSIÓN DE LA ARQUITECTURA DEL PROYECTO

Abra y estudie los siguientes archivos

- data/00-ST-ParranderosJDO.pdf – Presentación general del proyecto, su arquitectura y sus principales características.
- data/ArquitecturaReferencia.pdf – Arquitectura del proyecto, instanciada para el caso de ParranderosApp

Asegúrese de comprender:

- Que la arquitectura del proyecto define la estructura y contenido de los paquetes del programa.
- Cómo es el flujo de datos y el flujo de control entre las diferentes capas de la aplicación.
- Que los VOs protegen los datos del negocio de modificaciones indebidas por parte de las aplicaciones
- Que la persistencia de los datos del negocio tiene un único responsable: la clase PersistenciaParranderos, de la cual sólo puede haber un objeto (patrón SINGLETON). Además, esta es la única clase que conoce las clases SQLxxx (visibilidad por defecto), de manera que nadie más que esta clase puede modificar la información en la base de datos.

10 - EXPLORAR PARRANDEROSAPP

Con el mismo modelo de negocio, se puede definir varias aplicaciones. En particular, para este proyecto se definieron dos aplicaciones diferentes, con objetivos diferentes e interfaz de usuario diferentes: ParranderosApp y ParranderosDemo.

ParranderosApp es una aplicación transaccional que trabaja exclusivamente sobre la tabla TIPOBEBIDA y ParranderosDemo trabaja sobre todas las tablas de parranderos. Presentan una interfaz de menús, teniendo en común los menús de Mantenimiento y de configuración.

Abra, estudie y ejecute las pruebas de funcionamiento definidas en la clase TipoBebidaTest. Ponga especial atención a:

- Las pruebas de funcionamiento incluyen una prueba inicial de conexión. En caso de no haber conexión la prueba se declara como INCOMPLETA.

Ejecute ParranderosApp (src/main/java/uniandes.isis2304.parranderosApp/InterfazParranderosApp.java), observe el comportamiento y estudie el código para cada una de las opciones de menú. El orden sugerido es:

- Mantenimiento: todas las opciones, de arriba hacia abajo. Son interesantes las operaciones de limpieza de los logs y de la base de datos.
- Documentación: todas las opciones, de arriba hacia abajo, teniendo en cuenta que algunos de los documentos referenciados ya han sido estudiados. El documento de presentación general da una visión completa del proyecto, su arquitectura y las características destacadas de este desarrollo.
- TipoBebida: Implementa las operaciones sobre la tabla TIPOBEBIDA (adicionar, listar y borrar), mostrando una traza de ejecución en el panel de datos. Esta traza también se puede ver en los logs de la aplicación. Siga la ejecución desde la clase de interfaz, a las clases Parranderos, PersistenciaParranderos y SQLTipoBebida.
- Ponga especial atención a la definición y uso de los ValueObjects (VOxxxx), que protegen los datos del negocio de posibles modificaciones desde la interfaz.

11 - EXPLORAR PARRANDEROSDEMO

Replique el ejercicio anterior con la aplicación ParranderosDemo.

Esta aplicación trabaja con todas las tablas del modelo de parranderos, haciendo demostración de las operaciones básicas sobre las tablas y de algunas operaciones “interesantes” sobre algunas de ellas. Además de los menús de Mantenimiento y Documentación, hay un submenú para cada una de las tablas del negocio de Parranderos.

Por claridad de las demostraciones, cada demostración supone que empieza con todas las tablas vacías (sin tuplas), adiciona las tuplas de interés para la demostración¹, realiza la operación que se quiere demostrar y luego las borra, para que todo quede listo para las otras demostraciones². Cada demostración hace una traza de su comportamiento en el panel de datos de la interfaz.

Siga la ejecución desde la clase de interfaz, a las clases Parranderos, PersistenciaParranderos y las clases SQLxxxx.

Revise en particular el comportamiento y el código de las demostraciones de los menús:

- Bebida – Además de las operaciones básicas, está la eliminación de tuplas no referenciadas.

¹ Note que la creación de los datos está explícita en el código de los métodos. **ESTO ES VÁLIDO ÚNICAMENTE PORQUE EL OBJETIVO ES FACILITAR LA COMPRENSIÓN DEL FUNCIONAMIENTO DEL NEGOCIO DE PARRANDEROS.** En las aplicaciones “reales”, los datos no pueden hacer parte del código.

² Hay dos casos en los cuales esto no se cumple y debe utilizarse la opción de Limpiar base de datos del menú Mantenimiento de la interfaz

- Bar – Además de las operaciones básicas, está una consulta que involucra varias tablas y una actualización que involucra varias tuplas.
- Bebedor – Además de las operaciones básicas, hay consultas y modificaciones que involucran varias tablas y varias tuplas y las dos últimas opciones realizan la misma operación, pero con dos reglas de negocio diferentes. ATENCIÓN: estas dos últimas demostraciones son las que no dejan limpia la base de datos.
- Ponga especial atención a la definición y uso de los ValueObjects (VOxxxx), que protegen los datos del negocio de posibles modificaciones desde la interfaz.

12 - COMPRENSIÓN DE LA CONSTRUCCIÓN DE LA INTERFAZ DE USUARIO

La interfaz de usuario de las dos aplicaciones ParranderosApp y ParranderosDemo, se construyen con base en los archivos de configuración de interfaz. Esto son archivos JSON que se encuentran en el directorio src/main/resources/config.

Abra y estudie:

- Los archivos interfaceConfigApp.json e interfaceConfigDemo.json. Compare su contenido con los menús disponibles en las aplicaciones.
- El constructor de la clase Interfaz.
- Los métodos openConfig (...) y crearMenu (...)

13 - PROCESO PARA EL DESARROLLO DE UN NUEVO REQUERIMIENTO CON OPERACIONES YA DEFINIDAS EN LA CAPA DE NEGOCIO

Tomando como contexto ParranderosApp y como ejemplo el requerimiento de adicionar una nueva bebida a la base de datos, el proceso a seguir es:

- Modificar el archivo interfaceConfigApp.json, adicionando un nuevo menú (Bebida) y una opción dentro de ese menú (Adicionar bebida). El nombre del evento es el nombre del método en la clase interfaz que responde a dicho requerimiento (adicionarBebida)
- Adicionar el método adicionarBebida (public, void, sin parámetros, con su respectiva documentación) en la clase InterfazParranderosApp.
 - Puede tomar como guía el método adicionarTipoBebida ya existente en la clase.
 - Debe incluir un llamado al método de Parranderos que realiza la operación. Este método ya existe, pues fue definido y usado en la aplicación ParranderosDemo.
 - Haga un buen uso de VOBebida, en particular para mostrar la traza de la ejecución del requerimiento.
 - Tenga en cuenta que, para crear una bebida ya debe existir el tipo de bebida correspondiente y que ese tipo de bebida debe ser especificado por el usuario.
 - Tenga en cuenta el manejo de errores del usuario y de excepciones de la aplicación.
 - Asegúrese de documentar correcta y completamente el método.
- Adicione el método o métodos de prueba para el requerimiento. Para el caso del ejemplo:
 - Cree la clase BebidaTest, tomando como guía las clases de prueba ya existentes.
 - Adicione el método de prueba correspondiente: adicionarBebidaTest
- Genere la documentación del proyecto mediante el comando javadoc, siguiendo las instrucciones del punto 15 de este documento.

14 - PROCESO PARA EL DESARROLLO DE UN NUEVO REQUERIMIENTO CON OPERACIONES NO DEFINIDAS EN LA CAPA DE NEGOCIO

Tomando como contexto ParranderosApp y como ejemplo el requerimiento de cambiar el nombre de una bebida a la base de datos, el proceso a seguir es:

- Modificar el archivo interfaceConfigApp.json, adicionando al menú (Bebida) una nueva opción (Cambiar nombre bebida). El nombre del evento es el nombre del método en la clase interfaz que responde a dicho requerimiento (cambiarNombreBebida)
- Adicionar el método cambiarNombreBebida (public, void, sin parámetros, con su respectiva documentación) en la clase InterfazParranderosApp.
 - Puede tomar como guía el método adicionarTipoBebida ya existente en la clase.
 - Asegúrese de colocarlo junto con los otros métodos de Bebida.
 - Debe incluir un llamado al método de parranderos que realiza la operación. Haga un buen uso de VOBebida, en particular para mostrar la traza de la ejecución del requerimiento.
 - Tenga en cuenta que, para cambiar el nombre de una bebida es necesario conocer su identificador y que ese identificador debe ser dado por el usuario.
 - Tenga en cuenta el manejo de errores del usuario y de excepciones de la aplicación.
 - Asegúrese de documentar correcta y completamente el método.
- Adicionar el método cambiarNombreBebida en la clase Parranderos
 - Puede tomar como guía el método aumentarSedesBaresCiudad ya existente en la clase.
 - Debe incluir un llamado al método de la clase PersistenciaParranderos que realiza la operación.
 - Asegúrese de incluir la traza de la ejecución al log de parranderos.
 - Asegúrese de documentar correcta y completamente el método.
- Adicionar el método cambiarNombreBebida en la clase PersistenciaParranderos
 - Puede tomar como guía el método aumentarSedesBaresCiudad ya existente en la clase.
 - Debe incluir un llamado al método de la clase SQLBar que realiza la operación.
 - Tenga en cuenta el proceso transaccional y el manejo de las excepciones.
 - Asegúrese de documentar correcta y completamente el método.
- Adicionar el método cambiarNombreBebida en la clase SQLBar
 - Puede tomar como guía el método aumentarSedesBaresCiudad ya existente en la clase.
 - Escriba la sentencia SQL que resuelve el problema.
 - Asegúrese de documentar correcta y completamente el método.
- Adicione el método o métodos de prueba para el requerimiento. Para el caso del ejemplo:
 - Cree la clase BebidaTest, si no lo ha hecho anteriormente, tomando como guía las clases de prueba ya existentes.
 - Adicione el método de prueba correspondiente: cambiarNombreBebidaTest
- Genere la documentación del proyecto mediante el comando javadoc, siguiendo las instrucciones del punto 15 de este documento.

Note que se usa el mismo nombre de método en las diferentes clases, pues eso facilita la lectura y comprensión del código. Debe ser claro que la responsabilidad de esos métodos en cada clase es diferente y que entre todos ellos se resuelve el problema completo.

15 - GENERACIÓN DE LA DOCUMENTACIÓN JAVADOC DESDE ECLIPSE

Esto se puede hacer desde eclipse, así:

- Escoja la opción Generate javadoc del menú Project.
- Especifique dónde se encuentra el comando javadoc del ambiente Java: %JAVA_HOME%\bin\javadoc.exe.
- Seleccione el proyecto que va a documentar.
- Seleccione el directorio donde se va a generar la documentación. Generalmente en el directorio doc del proyecto.
- Next.
- Next en la ventana de "Configure Javadoc arguments for standart doclet.
- En Extra Javadoc option especifique **-encoding UTF-8**.
- Finish.

Para finalizar, también toca indicarle a los navegadores que el material generado es UTF-8. Para esto:

- Edite en archivo index.html que encuentra en el directorio especificado para la generación de la documentación del punto anterior
 - En la línea siguiente al tag <head> incluya el tag **<meta charset="UTF-8">**.
 - Guarde y cierre el archivo.
- Abra el archivo index.html desde un navegador y verifique que todo fue creado correctamente.

REFERENCIAS

http://www.datanucleus.org:15080/products/accessplatform_5_2/

<https://www.oracle.com/technetwork/java/index-jsp-135919.html>