"Implementación aplicación informática del caso SuperAndes"

Jaime Andrés Torres Bermejo, Juan Sebastián Alegría Zúñiga ISIS2304 – Iteración 2 – Proyecto de curso - Caso SuperAndes Universidad de los Andes, Bogotá, Colombia

j.torres16@uniandes.edu.co, j.alegria@uniandes.edu.co

Fecha de presentación: octubre 17 de 2022

Tabla de contenido

1 Introducción		
2 Resultados		
	Modelos	
	Resultados logrados	
2.3	Estructura del proyecto	2
2.4	Resultados no logrados	3
2.5	Balance de pruebas	3
2.6	Conclusiones	3
3 Bibliografía		3

1 Introducción

Este documento presentan los resultados obtenidos para la segunda iteración del proyecto SuperAndes en el curso Sistemas Transaccionales de la Universidad de Los Andes.

2 Resultados

2.1 Modelos

A continuación se presenta tanto el modelo conceptual como relacional, luego de haberlo normalizado y agregado atributos útiles para la implementación de los requerimientos del caso:

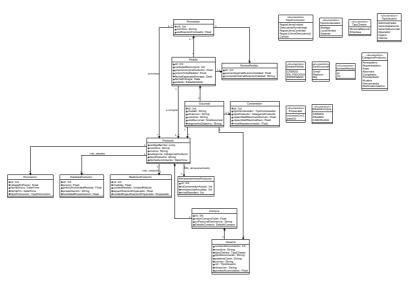


Figura 1: Modelo conceptual

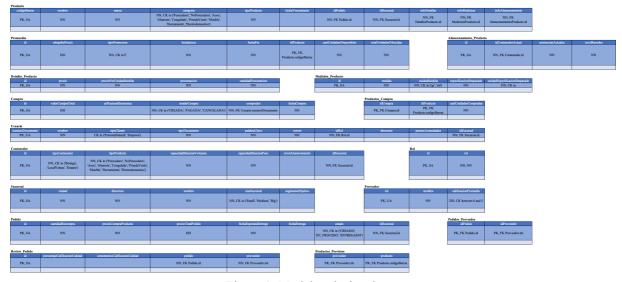


Figura 1: Modelo relacional

2.2 Resultados logrados

- Se logró mejorar significativamente los modelos que representan el negocio de este caso.
- Se añadieron todas las sentencias SQL que permiten a un administrador de bases de datos crear, limpiar, o eliminar todas las tablas, ya sea individualmente o automatizadamente por medio de un script.
- Se crearon todas las sentencias de tipo insert, necesarias para poblar la base de datos con más de 1000 datos de prueba en cada tabla. De este modo, se satisface los requerimientos funcionales 1 a 8 en su totalidad, y del 9 al 14 se insertan datos que más tarde se le permitirá a un usuario ingresar desde la interfaz de aplicación.
- Se crearon todas las sentencias SQL para satisfacer los requerimientos funcionales de consulta (1 a 6) con parámetros de ejemplo, que más tarde son modificados de acuerdo al input dinámico en tiempo de ejecución.
- Ahora bien, por parte de la implementación en JDO, se crearon todas las opciones de menú en la interfaz gráfica junto a las tablas en la unidad de persistencia del proyecto.
- Se modificaron los unit test y credenciales respecto a nuestros propios usuarios.
- Se crearon todas las clases para cada una de las tablas de negocio y persistencia.
- Se añadieron todos los métodos de comunicación con la interfaz, representación de negocio, y la mayoría de la persistencia general.

2.3 Estructura del proyecto

- En 'docs/javadocs/' podrá encontrar la documentación autogenerada del proyecto JDO.
- En `docs/modelos/` podrá encontrar el modelo relacional y conceptual utilizado para este proyecto.
 - En 'docs/sql/create-tables/' podrá encontrar todas las sentencias de creación de tablas.
- En `docs/sql/data` podrá encontrar sentencias de limpieza de base de datos y scripts de creación del esquema completo.
- En `docs/sql/data-population` podrá encontrar sentencias de inserción de registros de prueba para poblar la base de datos y soportar los requerimientos funcionales.
- En `docs/sql/rfc` podrá encontrar sentencias de consulta para soportar los requerimientos funcionales de consulta.

- En 'src/' podrá encontrar el código fuente del proyecto.

2.4 Resultados no logrados

A pesar de que se realizaron todas las consultas SQL necesarias para ejecutar en su totalidad los requerimientos de esta iteración, no se logró crear todos los métodos necesarios para visualizar su funcionamiento en la interfaz gráfica. Se crearon todos los métodos requeridos para satisfacer los requerimientos en la interfaz, interfazSuperAndes, negocio/SuperAndes, la mayoría en PersistenciaSuperAndes, y únicamente quedó faltando los SQL<Entidad> que se encargaban de ejecutar las sentencias SQL ya creadas en docs.

2.5 Balance de pruebas

A pesar de que no se realizaron pruebas unitarias o manejo de QA como se habría esperado del enunciado, se pudo generar una base de datos la cual nos permitiera correr los query especificados con datos relativamente realistas, y a la hora de probar los datos generados por esta mock data y compararlos con lo que nos debería dar una consulta de este tipo, se estableció que las consultas SQL realizadas funcionaban de forma adecuada para solucionar los problemas establecidos. Y esta carga de datos nos implicó el manejo de tuplas con los datos y condiciones establecidas, a partir de los cuales se puede establecer que las pruebas de integridad, de ser construidas formalmente, funcionarían prácticamente a la perfección.

2.6 Conclusiones

Nuestro modelo inicial soportaba, en su mayoría, todas las reglas de negocio y supuestos adicionales de implementación. Solo se necesitó añadir algunos nuevos parámetros para facilitar en su totalidad la eficiencia, privacidad, operabilidad, y demás requerimientos no funcionales

3 Bibliografía

1. *Universidad de los Andes*. [En línea] [Citado el: 1 de Septiembre de 2022.] https://bloqueneon.uniandes.edu.co//content/enforced/140479-UN_202220_ISIS2304_I/Proyecto/ST-Pry-DocMarco-It1.pdf?isCourseFile=true&_&d2lSessionVal=ZXW7mHJ8YHBaf9ZZ0mPgH8UUe&ou=140479.