

Práctica 3. Convertir datos en información

Objetivos de la práctica

- Realizar una implementación de un almacén de datos sobre bases de datos relacionales (ROLAP)
- Poblar el sistema anterior con datos multidimensionales y hacer consultas
- Utilizar el sistema para hacer recomendaciones a los usuarios.
- Crear un Dashboard para mostrar información

Entrega

Se deberá entregar un fichero comprimido en formato ZIP con los siguientes contenidos:

- `/src` : incluirá el código fuente de la práctica, directamente importable como proyecto Maven.
- `pom.xml` : fichero descriptivo de proyectos Maven.
- `autores.txt` : fichero con los datos de los autores de la práctica.
- Se puede añadir un PDF que incluya comentarios o aclaraciones sobre el desarrollo de la práctica.
- Se recomienda incluir también un comentario con el nombre de los autores en todos los ficheros fuente.
- **NO** puede haber ningún fichero con "instrucciones" de instalación, compilado ni ejecución.
- Consultar en el aula virtual la fecha de entrega.

Calificación

Esta práctica representa el 50% del total de la nota de prácticas.

Plantilla de corrección

En general, se evaluarán los siguientes puntos en cada uno de los ejercicios.

- Cumple requisitos técnicos (funciona en MyApps,...) y de formato (nombres de bases de datos, tablas,...)
- Se cumplen los requisitos de cada una de las tareas.
- Diseño de la arquitectura software en forma de capas de abstracción (entidades, repositorios y servicios)
- Funcionamiento general de la aplicación.

Es imprescindible que, antes de entregar la práctica, se verifique que funciona correctamente en el entorno MyApps, con Eclipse JEE y XAMPP. Si una práctica NO FUNCIONA en ese entorno podrá conllevar una calificación de SUSPENSO o rebaja de la nota.

Requisitos

Back-end: se trabajará necesariamente con Spring en Java, MySQL y Maven.

- No se permiten otras tecnologías de servidor como Node.js o similares.
- Se valorará especialmente la implementación del back-end como una API RESTful.
- En caso de que se implemente de este modo, se deberá trabajar necesariamente con Spring REST.

Front-end: la implementación de esta parte se realizará con HTML5 y Javascript.

- No se permite utilizar Angular.

Escenario de la práctica

Entre 2017 y 2018 se dieron de alta en la red **Autónomos Network** 50 profesionales diferentes.

A finales de 2018 la red puso a su disposición un proveedor de materiales y herramientas para toda Europa.

A través de diferentes consultas a la BD transaccional del proveedor se han extraído 3 tablas con todas sus transacciones desde que comenzó a funcionar. Queremos construir un almacén de datos para poder hacer análisis de información.

Para ello construiremos:

- Un esquema ROLAP que acoja la información
- Un ETL para nutrir al almacén de datos
- Un Dashboard para mostrar información
- Un Sistema recomendador que haga sugerencias a los profesionales de la red.

Tareas

1. Esquema estrella del Almacén de datos

- El almacén de datos tendrá como hecho central la compra de un producto, por un cierto importe, por parte de un profesional de la red, desde una capital europea, en una fecha concreta.
- En definitiva tenemos 4 dimensiones (quién, dónde, qué y cuándo) para un hecho central (cuánto)
El almacén de datos se diseña de manera que cada una de estas dimensiones tenga los atributos de la tabla de abajo a la izquierda. Además, tomamos las medidas mostradas a la derecha.
- Además se requiere utilizar claves sustitutas (*surrogate keys*) en todas las tablas del esquema.

<i>Cliente</i>	<i>Lugar</i>	<i>Producto</i>	<i>Tiempo</i>	<i>Compra</i>
Nombre	Capital	Nombre	Día de semana	Importe
Apellido	País	Descripción	Día de mes	Valoración
Correo Electrónico	Núm. de Habitantes		Nombre de mes	
Dominio Correo			Número de mes	
Fecha de alta (día)			Año	
Fecha de alta (mes)			esFinSemana	
Fecha de alta (año)				

2. Sistema ETL

- Queremos construir un sistema ETL que procese los tres ficheros .CSV proporcionados y pueble con ellos el esquema estrella diseñado previamente.
- Para ello conviene realizar primero una inspección de los datos para comprender el formato en el que vienen y el formato al que deben llegar.

3. Sistema Recomendador

- Se quiere construir un sistema recomendador que genere un fichero de texto con 3 recomendaciones para cada uno de los 50 clientes.
- Además se debe mostrar en una tabla de Google Charts el resultado

4. Dashboard

- Realizar un *dashboard* con Google Charts, de diseño libre, siguiendo las recomendaciones de los apuntes.
- Debe haber, al menos, un mapa, un gráfico de sectores y un gráfico de agujas.