Memoria Proyecto Hackaton

Adrián Pradas Gallardo
Emilio Rodrigo Carreira Villalta
Rocio Guzman Arrollo
Juán de Dios Alfaro Lopez

Thursday 17th October, 2024

Tabla de Contenidos

1. Introducción

Este documento relata la elaboración del reto proporcionado en todos sus aspectos, creación de base de datos, carga de datos, depurado de datos, búsqueda de relaciones, elaboración de frontend y backend, etc.

Todo el código es accesible a través de github [1] para su evaluación, pero a lo largo de este documento se mostrarán todas las imágenes necesarias para su explicación

[1] https://github.com/rorro6787/Hackaton.git

2. Backend

Contents

2.1 Creacion de la base de datos

El primer paso para afrontar el desafió consistía en entender el funcionamiento de la base de datos, para ello nuestro compañero Adrián asistió al meet en el que se explicaron los servicios que teníamos disponibles de Oracle, y junto con el video el día antes del evento iniciamos sesión en el cloud de Oracle y creamos la base de datos y la instancia virtual aunque esta al final no terminara siendo usada.

2.2 Carga de datos

Ya en el día del Malakaton nos dividimos en dos grupos, el grupo del frontend cuya parte es explicada en el siguiente capitulo y el del backend que se encargo de la base de datos. Primero teníamos que encargarnos de cargar los datos en la base de datos, esta fue una tarea sencilla excepto por el fichero .tsv, el cual daba fallos en ciertas ocasiones, la solución fue cargar todos los datos como varchar y ya si fuera necesarios se harían casts para manejarlos mejor dentro de las sentencias.

2.3 Creación de relaciones

Con los datos ya cargados en la base de datos comenzaba el proceso de relaciones dentro de las tablas, la tabla agua tenia una clara relación a través del atributo id a la tablas Embalses, por lo que se crearon las claves primarias y foraneas correspondientes a las tablas.

Ahora procedíamos con el paso mas complicado, establecer una relación entre Embalses y Listado, en primer lugar pudimos comprobar que el atributo nombre dentro de Embalses coincidía al cien por cien con el atributo nombre de Listado un 70 por ciento de las veces, por lo que estaba claro que teníamos que crear en primer lugar un atributo extra en Listado que fuera la ID correspondiente a el Embalse.

```
BEGIN
    -- Actualizamos la columna ID en la tabla LISTADO
    FOR r IN (SELECT CODIGO, NOMBRE
              FROM LISTADO)
    LOOP
        -- Intentamos encontrar el ID correspondiente en la tabla EMBALSE
        UPDATE LISTADO l
        SET l.ID = (SELECT e.ID
                    FROM EMBALSE e
                    WHERE e.EMBALSE_NOMBRE = r.NOMBRE)
        WHERE l.CODIGO = r.CODIGO
        AND EXISTS (SELECT 1
                    FROM EMBALSE e
                    WHERE e.EMBALSE_NOMBRE = r.NOMBRE);

        -- Si no hay coincidencia, asignamos 0
        UPDATE LISTADO l
        SET l.ID = 0
        WHERE l.CODIGO = r.CODIGO
        AND NOT EXISTS (SELECT 1
                        FROM EMBALSE e
                        WHERE e.EMBALSE_NOMBRE = r.NOMBRE);
    END LOOP;
END;
/
```

Esta es la funcion utilizada.

Quedaba el 30 por ciento restante y analizando casos nos dimos cuenta que en ocasiones no coincidían simplemente por las tildes, por lo que en la base de datos creamos una función para quitar los acentos, esto supuso pasar de un 70 por ciento de los datos asignados al 85, es decir, un avance considerable.

```
create or replace FUNCTION POLISH_USER.remove_accent(p_texto IN VARCHAR2)
RETURN VARCHAR2
IS
    v_resultado VARCHAR2(4000);
BEGIN
   v_resultado := REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(p_texto, 'Á', 'A'), 'É', 'E'), '
    RETURN v_resultado;
END;
```

Para usarla la aplicamos a todas las columnas de los nombres para convertirlas y asignarlas.

El ultimo paso fue ver un patrón que era ”nombre,articulo” por lo que creamos otra función que elimino la coma y el articulo para poder comparar el nombre con la función LIKE de Oracle para asignar mas Ids.

```
create or replace FUNCTION POLISH_USER.quitarcoma(p_palabra IN VARCHAR2)
RETURN VARCHAR2
```

```
IS
    v_pos_coma NUMBER;
BEGIN
    -- Encontramos la posición de la coma
    v_pos_coma := INSTR(p_palabra, ',');

    -- Si la coma no existe, devolvemos la palabra completa
    IF v_pos_coma = 0 THEN
        RETURN p_palabra;
    ELSE
        -- Devolvemos lo que hay antes de la coma
        RETURN SUBSTR(p_palabra, 1, v_pos_coma - 1);
    END IF;
END;
```

A través de esta función obtuvimos los nombres sin la coma, por lo que usamos después una función igual que la primera usando el comparador LIKE con los datos sin id todavia.

Finalmente los últimos nombres no tenían patrones concretos por lo que los cambiamos a mano ya que se trataba de unos 20 o 30.

Con esto terminamos el procesado de datos y el establecimiento de relaciones.

2.4 End points

Previamente creamos los índices para las ids, coordenadas x e y, nombres y también al establecer las claves Oracle crea automáticamente esos indices por lo que ya esta creado todo.

A través de la propia base de datos podíamos crear end points para consultarlos a través de nuestro frontend con la función de REST en las tablas y vistas, el proceso consistió en crear vistas que tuvieran los datos necesarios para los creadores del frontend, una vez creadas activaríamos el servicio rest y automáticamente obteníamos los comandos para hacer las peticiones necesarias.

2.5 Modelos de prediccion

Hemos intentado crear un modelo de prediccion usando Arima pero no hemos podido lograrlo con exito

3. FrontEnd

Contents

Se trata de la parte visual del proyecto, queríamos que fuera lo mas sencilla e intuitiva posible a la vez que resolvía los problemas. En esta sección trataremos de explicar la interfaz por encima ya que a través del repositorio de github se tiene acceso a todo el código.

3.1 Planteamiento

Para la creación de la pagina nos basamos en un modelo multi página creado con React y css que a la vez fuera Fully Responsive para cualquier dispositivo, con esto planteamos un buen paso inicial a la resolución del problema.

3.2 Página Inicial

Nuestra web tiene en todo momento desplegado en la parte superior de la pantalla un menú para acceder a sus funcionalidades, dentro de la pagina principal hemos incluido el logo de nuestro equipo junto con una bienvenida para después mostrar iconos que te envían también a las funcionalidades, es decir las otras paginas. También hemos incluido un apartado en referencia a nuestro equipo para darnos a ver al publico.

3.3 Página Embalses

En esta parte de la pagina nos encargamos de mostrar los embalses de la base de datos en el mapa, el usuario que acceda a la pagina puede añadir unas coordenadas o ver su propia ubicación para desplegar un radio en el mapa de la distancia que quiera para ver los Embalses que quiera. También si pulsas en cualquier arte del mapa el radio se moverá y se mostraran nuevos embalses, para que el proceso sea también mas sencillo.

3.4 Página Estadísticas

En esta parte podemos seleccionar un Embalse en concreto, el menú permite escribir para obtener cada vez una lista mas filtrada, una vez seleccionado podemos ver su localización, capacidad, cauce, y una gráfica que muestra la evolución de su agua conforme al tiempo.

3.5 Página Predicciones

hemos implementado un fichero de predicciones también pero debido a la falta de tiempo no se pudo crear un modelo de predicción por lo que esta parte no se ha terminado de implementar.