

Projecte

Carga de dades



1	OBJECTIU	2
2	XML	3
3	FORMULARI	3
4	ESTRUCTURA DE DADES	4
4.1	OBJECTES	4
4.2	OBJECTE Manager	5
4.2.1	Carregar dades a estructura de dades	5
4.2.2	Objecte per Carregar dades a Base de dades	5
5	BASE DE DADES	8
5.1	Script creació de taules	8
5.2	Trigger	9
5.3	Un cop carregades Stored Procedure	9
5.4	Cridar un SP des d'una aplicació	9
6	MILLORES	10

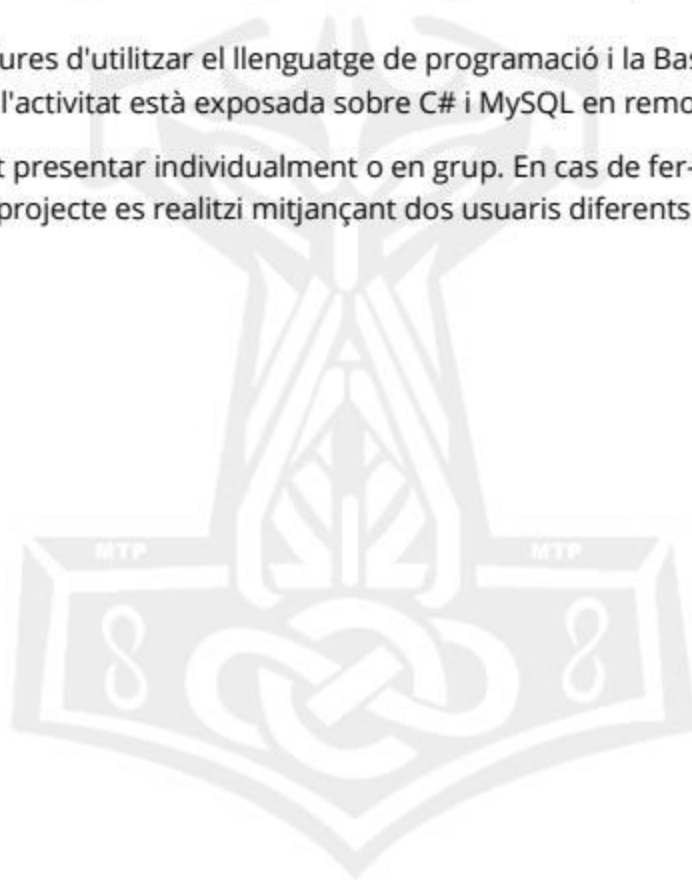
1 OBJECTIU

Aquest projecte engloba la part del Mòdul 2 Base de Dades i el Mòdul 4 de Llenguatges de Marques, tot i que precisem dels coneixement del Mòdul 3 Programació per poder-lo desenvolupar.

L'objectiu principal del projecte és carregar dades provinents de ***diferents fitxers XML amb la mateixa estructura*** a diferents taules en una BBDD, finalment a partir de la informació a la BBDD, realitzar un ***stored procedure*** que retorni un XML diferent a l'original, aquest XML s'ha de descarregar en una carpeta de l'aplicació.

Tot i que, sou lliures d'utilitzar el llenguatge de programació i la Base de dades que més bé us vagi, l'activitat està exposada sobre C# i MySQL en remot.

L'activitat es pot presentar individualment o en grup. En cas de fer-ho en grup s'exigirà que el projecte es realitzi mitjançant dos usuaris diferents en github.



2 XML

El vostres fitxers XML:

- Mínim de 4 nivells (o bé en un mateix nivell diferents tipus de nodes i amb fills)
- Validat sintàcticament.
- Validat mitjançant una plantilla DTD.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dmc SYSTEM "dmc.dtd">
<dmc>
  <championship year="1985" country="UK" city="London">
    <dj local="UK North">
      <nom>Roger Johnson</nom>
      <pos>1</pos>
    </dj>
    <dj local="UK South">
      <nom>Martin Sweeney</nom>
      <pos>2</pos>
    </dj>
    <dj local="Sweden">
      <nom>Roger Tuuri</nom>
      <pos>3</pos>
    </dj>
  </championship>
```

3 FORMULARI

Crear un formulari que ens permeti seleccionar el fitxer XML a carregar. Podeu fer servir l'objecte *OpenFileDialog*.



Carregar Fitxer XML

Carregar Dades

Fitxer XML

Carregar

4 ESTRUCTURA DE DADES

Per emmagatzemar les dades del fitxer XML, primer de tot necessitem una estructura de dades.

En l'exemple, observem tres objectes diferents *dmc*, *championship* i *dj*. Cal tenir en compte que *dmc* pot tenir múltiples nodes *championship* al igual que *dj*. Per tant, és necessari utilitzar algun tipus de col·lecció de dades.

En l'exemple s'ha optat per *List*, ja que que ens proporció ordre, és a dir, cada llista obtindrà el mateix tipus de col·lecció de dades. (P.e. una llista d'objectes *championship*, una llista d'objectes *dj* ...)

4.1 OBJECTES

Cada element de XML ha de ser emmagatzemat en un objecte, amb els seus atributs i mètodes de comportament.

Recordar que podeu utilitzar *getter* i *setter* de forma convencional i explícit o més simplificada i elegant com ens ofereix *C#*.

```
int valor = objeto.getPos(); // Obtén el valor de pos  
objeto.setPos(10); // Establir un nou valor de pos
```

```
int valor = objeto.Pos; // Obtén el valor de pos  
objeto.Pos = 10; // Establir un nou valor de pos
```


4.2 OBJECTE MANAGER

En l'exemple s'ha optat, per crear un objecte amb dos mètodes, un que ens permeti llegir les dades del fitxer XML i emmagatzemar-les en els corresponents objectes i llistes i un segon mètode que ens permetrà un cop carregades les dades bolcar-les a una base de dades remota.

4.2.1 Carregar dades a estructura de dades

Un cop tenim el fitxer XML, l'hem de validar mitjançant un **DTD**, si tot correcte, em de començar a processar-lo, per tant, necessitem un objecte, que ens permeti anar llegint el fitxer XML i anar-lo emmagatzemant els diferents objectes. Per tractar el fitxer XML, podeu utilitzar els diferents objectes que ens proporciona la class **System.XML**.

- **XMLDocument**: Per carregar el document XML.
- **XmlNodeList**: Per obtenir un conjunt de nodes a partir de comandes en **XPath**.
- **XmlNode**: Per tractar un únic node, obtenir el valor del seus atributs o el valor d'un node (text).

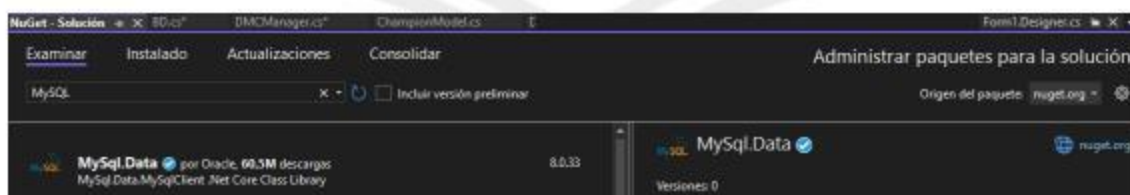
```
public static bool carregarModel(String filePath)
{
    bool bres = false;
    try
    {
        if (DTDValidator.Validate(filePath))
        {
            XmlDocument doc = new XmlDocument();
            doc.Load(filePath);
            XmlNodeList ElementsCampionat = doc.SelectNodes("//championship");

            foreach (XmlNode campionat in ElementsCampionat)
            {
```

4.2.2 Objecte per Carregar dades a Base de dades

Utilitzarem una base de dades remota, per exemple, la que ens proporciona la web <https://db4free.net/>, cal registrar-se i posteriorment validar-se per email.

El tractar-se d'una base de dades **MySQL**, necessitem instal·lar el paquet **MySQL.Data**.



Recorda que necessites **connectionString** per connectar-te a la Base de Dades Remota.

```
private const string server = "db4free.net";
private const string port = "3306";
private const string database = "";
private const string username = "";
private const string password = "";

private const string connectionString = "Server=" + server + ";Port=" + port + ";Database=" + database + ";Uid=" + username + ";Pwd=" + password + ";OldGuids=true;";

private static MySqlConnection? connection;
```

La connexió i desconnexió:

```
private static bool Conectar()
{
    connection = new MySqlConnection(connectionString);
    try
    {
        connection.Open();
        return true;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.ToString());
        connection.Close();
    }
    return false;
}

2 referencias
private static void Desconectar()
{
    if (connection != null) {
        connection.Close();
    }
}
```

Per tal de carregar les dades que ja tindrem emmagatzemades, haurem de recórrer tots els objectes i les seves llistes....

```
public static bool enviarDadesBBDD()
{
    bool bres = false;
    try
    {
        BD.IniciarBD();
        foreach (var campionat in Campionats)
        {
            int idcampionat = BD.InsertarCampionat(campionat);
            foreach (var dj in campionat.getDJs())
            {
                BD.InsertarDJ(dj, idcampionat);
            }
        }
        BD.getdades();
    }
    catch (Exception ex)
    {
        message = "Error el enviar dades a BBDD: " + ex.Message;
    }
    return bres;
}
```

Exemple de inserció d'un registre a la taula Campionat:

```
public static int InsertarCampionat(ChampionModel campionat)
{
    int insertedId = -1;
    try
    {
        using (connection = new MySqlConnection(connectionString))
        {
            connection.Open();

            string query = "INSERT INTO CAMPIONAT (year, country, city) VALUES (@year, @country, @city);";
            string selectQuery = "SELECT LAST_INSERT_ID();";

            using (MySqlCommand command = new MySqlCommand(query, connection))
            {
                command.Parameters.AddWithValue("@year", campionat.Year);
                command.Parameters.AddWithValue("@country", campionat.Country);
                command.Parameters.AddWithValue("@city", campionat.City);

                int rowsAffected = command.ExecuteNonQuery();
                Console.WriteLine("Registres insertats: " + rowsAffected);

                using (MySqlCommand selectCommand = new MySqlCommand(selectQuery, connection))
                {
                    insertedId = Convert.ToInt32(selectCommand.ExecuteScalar());
                    Console.WriteLine("Registre insertat amb ID: " + insertedId);
                }
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error al insertar el registro: " + ex.Message);
    }

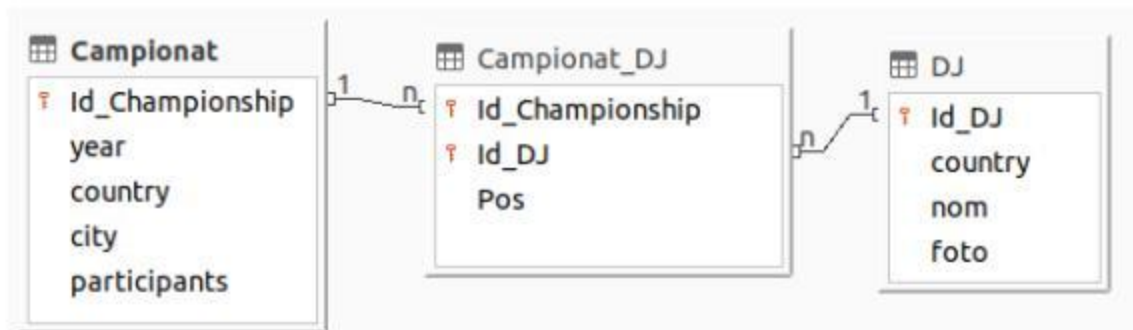
    return insertedId;
}
```



5 BASE DE DADES

Haurem de crear una taula per cada un dels objectes que tinguem, i altres possibles taules que ens permetin relacionar-les.

Aquestes taules seran carregades des de l'aplicació c#.



5.1 SCRIPT CREACIÓ DE TAULES

Cal crear un **script.sql** per tal de crear les taules necessàries per allotjar les dades.

```

DROP TABLE IF EXISTS CAMPIONAT_DJ;
DROP TABLE IF EXISTS DJ;
DROP TABLE IF EXISTS CAMPIONAT;

CREATE TABLE CAMPIONAT (
    Id_Championship INT AUTO_INCREMENT PRIMARY KEY,
    year INT NOT NULL,
    country VARCHAR(50) NOT NULL,
    city VARCHAR(50) NOT NULL,
    participants INT
);

CREATE TABLE DJ (
    Id_DJ INT AUTO_INCREMENT PRIMARY KEY,
    Country VARCHAR(50) NOT NULL,
    nom VARCHAR(50) NOT NULL,
    foto BLOB
);

CREATE TABLE CAMPIONAT_DJ (
    Id_Championship INT NOT NULL,
    Id_DJ INT NOT NULL,
    POS INT,
    PRIMARY KEY (Id_Championship, Id_DJ),
    FOREIGN KEY (Id_Championship) REFERENCES CAMPIONAT (Id_Championship) ON DELETE CASCADE,
    FOREIGN KEY (Id_DJ) REFERENCES DJ (Id_DJ) ON DELETE CASCADE
);

CREATE INDEX IDX_CAMPIONAT_DJ_Championship ON CAMPIONAT_DJ (Id_Championship);
CREATE INDEX IDX_CAMPIONAT_DJ_DJ ON CAMPIONAT_DJ (Id_DJ);
  
```


5.2 TRIGGER

Crear un *Trigger* capaç d'actualitzar el camp **participants** de la taula **Campionat**, amb el número total de **DJ** participants en cada campionat en concret.

5.3 UN COP CARREGADES STORED PROCEDURE

Un cop carregades correctament les dades del XML a les diferents taules, haurem de realitzar un **Stored Procedure** que creï un nou XML, en l'exemple, el que fa es obtenir el DJ que ha guanyat més campionats. (Es valorarà la complexitat de la consulta).

```

DROP PROCEDURE IF EXISTS obtenir_millorDJ;
DELIMITER //

CREATE PROCEDURE obtenir_millorDJ()
BEGIN
    DECLARE xml_result TEXT;

    -- Obtenir el ID del DJ que ha guanyat més campionats
    DECLARE DJ_ID INT;

    SELECT CAMPIONAT_DJ.Id_DJ
    INTO DJ_ID
    FROM CAMPIONAT_DJ
    GROUP BY CAMPIONAT_DJ.Id_DJ
    ORDER BY COUNT(*) DESC
    LIMIT 1;

    -- resultat en un string XML
    SET xml_result = CONCAT('DJ id="', DJ_ID, '"',
        '<DJ>', (SELECT DJ.nom FROM DJ WHERE DJ.Id_DJ = DJ_ID), '</DJ>',
        (SELECT GROUP_CONCAT('<championat>', CAMPIONAT.country, ' - ', CAMPIONAT.nom, '</championat>')
        FROM CAMPIONAT_DJ
        JOIN CAMPIONAT ON CAMPIONAT.DJ.Id_Championship = CAMPIONAT.Id_Championship
        WHERE CAMPIONAT_DJ.Id_DJ = DJ_ID),
        '</DJ>');

    SELECT xml_result AS XML_Result;
END //
DELIMITER ;

```

5.4 CRIDAR UN SP DES D'UNA APLICACIÓ

La forma de recupera la informació retornada per el *stored procedure* anterior, des de la nostre aplicació:

```

public static string getdades()
{
    string resultVariable = string.Empty;
    using (MySqlConnection connection = new MySqlConnection(connectionString))
    {
        connection.Open();

        using (MySqlCommand command = new MySqlCommand("obtener_xml", connection))
        {
            command.CommandType = CommandType.StoredProcedure;

            using (MySqlDataReader reader = command.ExecuteReader())
            {
                if (reader.Read())
                {
                    resultVariable = reader.GetString("XML_Result");
                }
                else
                {
                    resultVariable = string.Empty;
                }
            }
        }

        connection.Close();
    }
    return resultVariable;
}

```

6 MILLORES

Possibles millores:

- Crea un nou formulari, que permeti entrar noves dades a la Base de Dades.
- Segurament la carrega del XML a Base de Dades serà lenta, cronometra quant tarda e intenta accelerar-la.
- Llençar la creació de taules a BBDD des de l'aplicació, de forma que si no existeixen les taules les creï en la BBDD.
- Creació del SP des de l'aplicació.
- Crea un *Trigger* que ompli un dels camps al inserta un nou registre, p.e. el insertar un DJ, si aquest va quedar 1, escriure un text a la taula DJ en un nou camp de DJ com per exemple "El DJ marc va queda primer en el campionat que es va celebrar a Londres".

