















CICLE	DESENVOLUPAMENT D'APLICACIONS MULTIPLATAFORMA	
MÒDUL	M3 - PROGRAMACIÓ	
U.FORMATIVA	UF6 – INTRODUCCIÓ A LA PERSISTÈNCIA EN BASES DE DADES	
PROFESSOR	JAUME FADÓ	

## ClFamiliesSqlServer (Model)

Les classes model normalment s'anomenen amb el nom de la classe d'objecte. És a dir, en aquest cas l'anomenaríem **ClFamilies** però nosaltres, per a fer les coses més entenedores i tenint en compte que el model està lligat a l'SGBD hi afegirem el nom d'aquest SGBD i li direm **ClFamiliesSqlServer**.

ClFamiliesSqlServer	
Clase	
▲ Campos	
 bd : ClBdSqlServer	
 bdAccessible : bool	
▲ Propiedades	
 idFamilia { get; set; } : string	
 nomFamilia { get; set; } : string	
▲ Métodos	
 ~ClFamiliesSqlServer()	
 ClFamiliesSqlServer(ClBdSqlServer xbd)	
 existeixFamilia() : bool	
 getFamilia() : bool	
 llistaFamilies(ref DataSet dset, int n) : void	
 modificarFamilia() : bool	
 novaFamilia() : bool	
 quantesFamilies() : int	
 quantesFamiliesXprefix(string prefix) : int	
 suprimirFamilia() : bool	

Aquesta classe model farà les peticions a la classe ClBdSqlServer (AccessBD) per a accedir a la BD

### Propietats

- **bd** És una propietat **privada** de tipus **ClBdSqlServer** que permetrà la connexió i les operacions sobre la BD

CICLE	DESENVOLUPAMENT D'APLICACIONS MULTIPLATAFORMA	
MÒDUL	M3 - PROGRAMACIÓ	
U.FORMATIVA	UF6 – INTRODUCCIÓ A LA PERSISTÈNCIA EN BASES DE DADES	
PROFESSOR	JAUME FADÓ	

- **bdAccessible** És un booleà que ens indica si tenim accés a la BD, és a dir, si ens hem pogut connectar a la BD i l'hem pogut obrir.
- **idFamilia** És pública. Correspon a una columna de la taula en la BD.
- **nomFamilia** És pública. Correspon a una columna de la taula en la BD.

## Mètodes

- **Constructor** Se li passa com a paràmetre la connexió a la BD que ja haurem fet en el FrmMain. **bdAccessible** contindrà **true** si la BD està oberta.
- **novaFamilia** Insereix una nova família en la taula de famílies i retorna cert o fals segons si s'ha pogut fer l'operació o no. Hem de considerar que les propietats **idFamilia** i **nomFamilia** contenen les dades que hem d'inserir a la base de dades. No cal que fem cap comprovació sobre aquestes dades perquè ja les haurà fet la classe **controlador** que ara no tenim feta però que farem en la pràctica següent.
- **modificarFamilia** Modifica la taula de famílies i retorna cert o fals segons si s'ha pogut fer l'operació o no. Les propietats **idFamilia** i **nomFamilia** contenen les dades necessàries per a la modificació.
- **suprimirFamilia** Fa la supressió en la taula de famílies i retorna cert o fals segons si s'ha pogut fer l'operació o no. El codi que s'ha de suprimir és el que tenim a la propietat **idFamilia**.
- **getFamilia** Consulta a la BD les dades de la família indicada en la propietat **idFamilia**. Retorna cert o fals segons si l'ha trobat o no. En el cas que s'hagi trobat, les altres propietats de la classe contindran les dades obtingudes en la consulta.

CICLE	DESENVOLUPAMENT D'APLICACIONS MULTIPLATAFORMA	
MÒDUL	M3 - PROGRAMACIÓ	
U.FORMATIVA	UF6 – INTRODUCCIÓ A LA PERSISTÈNCIA EN BASES DE DADES	
PROFESSOR	JAUME FADÓ	

- **existeixFamilia** Mira si existeix a la BD la família indicada en la propietat **idFamilia**.  
Retorna cert o fals segons si existeix o no.
- **llistaFamilies** Obté la llista de les famílies. Se li passa un paràmetre **int ordre** que si és **0** vol dir que es volen les dades ordenades per l'**idFamilia** i si és **1** per **nomFamilia**.  
Deixa les dades obtingudes en un DataSet que s'haurà de retornar passant-lo per referència.
- **quantiesFamilies** Retorna un enter amb el número de famílies que hi ha.
- **quantiesFamiliesXprefix** Retorna un enter amb el número de famílies el nom de les quals comença pel prefix que s'ha passat per paràmetre al cridar la funció.

## CONSIDERACIONS GENERALS A TENIR EN COMPTE

1. Cal mirar el diagrama de classes per a esbrinar quins elements són **privats** i quins **públics**.
2. S'han de programar les classes de manera que el codi sàpiga gestionar tota mena d'errors que es puguin produir perquè les dades no són correctes, perquè en falten, perquè s'intenta fer una operació que no es pot realitzar, etc. Si s'utilitzen operacions **try** cal que les excepcions mostrin missatges de tipus **MessageBox** amb les causes de l'excepció i tota la informació que sembli convenient.