

# ENS IV – Tasca 01

Autor: Sergio Blanco Cuaresma

## ***Enunciat***

Se ha d'implementar un Xat en C# utilitzant .NET Remoting com a mecanisme de comunicació client/servidor.

- Es valorarà el disseny i el codi generat.
- El protocol de comunicació hauria de poder ser intercanviable amb altres, com per exemple TCP/IP.
- Utilitzar, a ésser possible, patrons de disseny.
- Incloure un fitxer explicatiu del funcionament de la pràctica.

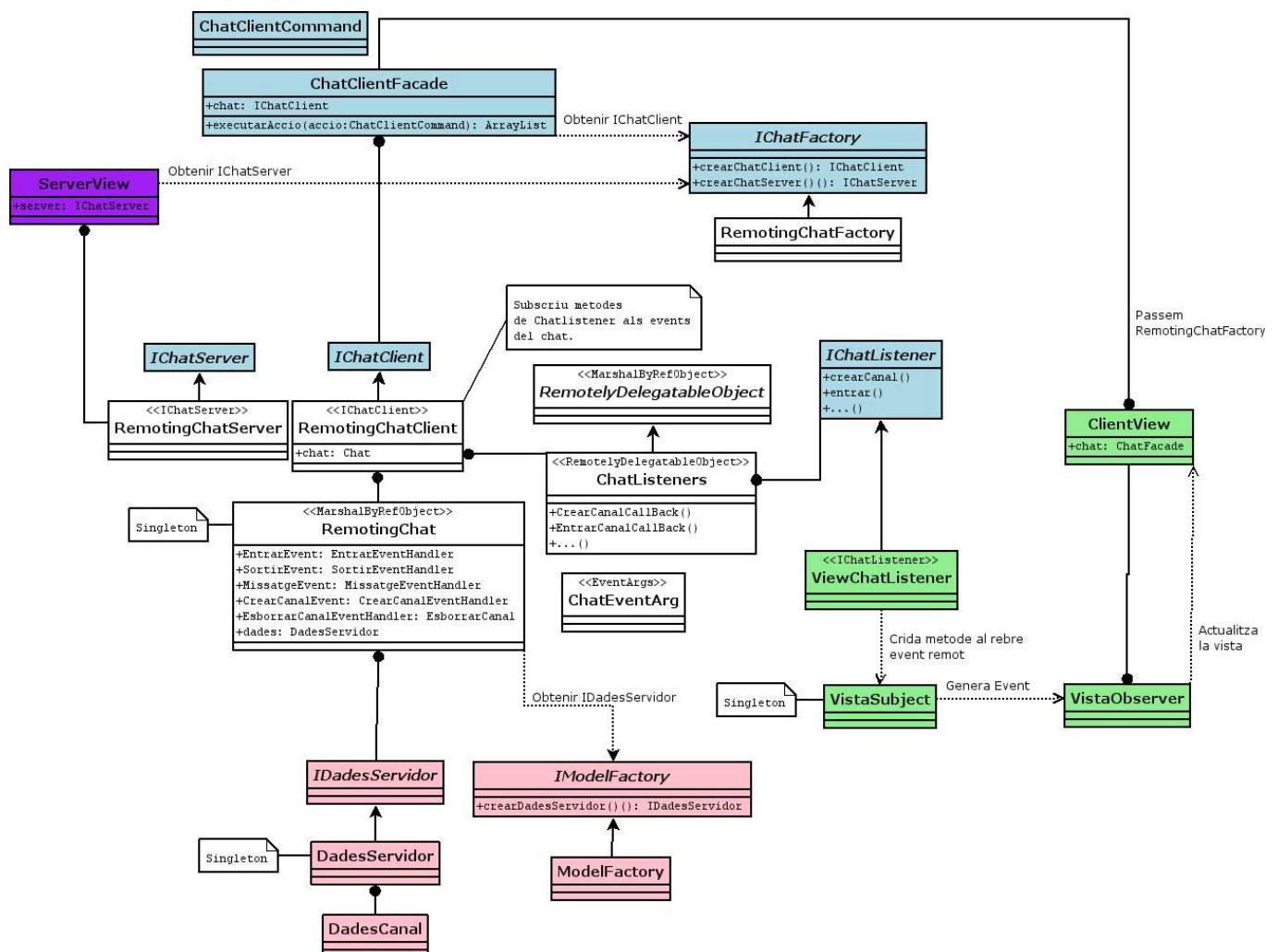
Opcional: que suporti múltiples canals.

## Disseny

Al disseny de l'aplicació s'ha seguit el model MVC (Model-Vista-Control), per tant s'han generat 5 projectes diferents dintre de la solució:

- Controlador: Interfícies d'accés al client i servidor chat. [Color Blau]
- Remoting: Implementació específica del servidor i client chat. [Color Blanc]
- Model: Classes necessàries per guardar la informació del chat a memòria. [Color Rosa]
- VistaServidor [Color Lila]
- VistaClient [Color verd]

Esquema:



Els patrons de disseny utilitzats són:

- MVC: La lògica del programa es troba separat de la interfície i de l'emmagatzemament de la informació, permetent un intercanvi senzill dels diferents mòduls.

- Abstract Factory:
  - En el model s'utilitza Abstract Factory per facilitar l'ús de diferents mètodes d'emmagatzemament de les dades. Només s'ha implementat l'emmagatzemament a memòria, però seria senzill implementar noves classes que guardessin a bases de dades.
  - Al controlador, per poder accedir al client que controla el chat o crear el servidor, s'utilitza un Abstract Factory. D'aquesta forma aconseguim independència del protocols utilitzats, encara que Remoting ja ens ofereix aquesta possibilitat, es possible que ens interessi canviar la implementació Remoting per una amb sockets plans.
- Facade: La interfície del chat client no s'accedeix directament sinó que s'utilitza un Facade, donant llibertat a la implementació de la interfície client.
- Command: El Facade de la interfície client executarà comandes, d'aquesta forma s'aconsegueix major flexibilitat a l'hora d'afegir o treure noves accions del chat.
- Observer: A la vista client, tenim un “assumpte” que es accedit per 2 observadors. En primer lloc tenim l'objecte que rep els events remots, aquest modificarà l'assumpte cridant els seus mètodes. Per altra banda, tenim l'observador de la interfície el qual esta subscript als events de l'assumpte, quan aquest es actualitzat, els events son executats. Aquesta implementació ens permetria tindre més interfícies visualitzant la mateixa informació.
- Singleton:
  - RemotingChat: Només volem un objecte remot en execució. En qualsevol cas, la configuració de Remoting singleton ja assegura això.
  - DadesServidor: Només hi ha un objecte al qual es guarda la informació del servidor.
  - VistaSubject: Només hi ha un “assumpte” amb el que els observadors interactuen.

## ***Implementació***

La implementació de Remoting utilitza fitxers de configuració per establir els paràmetres de comunicació (RemotingServer.config i RemotingClient.config). Aquests poden ser indicats des de la interfície gràfica.

L'objecte remot accedit es RemotingChat el qual serà passat per referència, al igual que ChatListener, el qual servirà per executar els events remots a la màquina client (el servidor cridarà al client). Per a que Remoting permeti passar per referència, cal indicar especificar a Remoting que el nivell de seguretat serà “Full” segons el següent article:

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconautomaticdeserializationinnetremoting.asp>

Això es pot aconseguir en els arxius de configuració del client i el servidor.

A la banda del client gràfic, l'observador que s'encarrega d'actualitzar la interfície (e.g. Entra un nou usuari) no serà executat al mateix thread de la pròpia interfície i per tant, intentar modificar components d'aquesta directament pot provocar errors. Per evitar aquest problema, s'utilitza funcionalitats pròpies de Windows Forms tal i com s'explica al següent article (apartat Wrapping Control.Invoke):

<http://msdn.microsoft.com/library/default.asp?url=/msdnmag/issues/03/02/multithreading/toc.asp>

S'ha implementat també la possibilitat de diferents canals de comunicació tot i que la interfície client només es pot trobar a un canal a la vegada. Característiques:

- El client ha d'indicar el seu nick per poder connectar (es permeten nicks duplicats).
- Al connectar, s'actualitza el llistat de canals disponibles.
- Quan un usuari vol entrar a un canal inexistent, aquest es crea.
- Quan un usuari surt d'un canal, si aquest queda buit, llavors es destrueix.
- L'usuari d'un canal veu quan entren o surten persones, a més, disposa d'un llistat de les persones que hi ha actualment.

