

Pràctiques

Butlletí Pràctica 1

Introducció a Visual Studio i Team Services

Enginyeria del Programari

ETS Enginyeria Informàtica
DSIC – UPV

Curs 2016-2017

1. Objectiu

L'objectiu de la sessió de laboratori és introduir a l'alumne a les principals funcionalitats de *Visual Studio 2015* i la seua relació amb l'eina de gestió de projectes en equip i control de versions *Team Services*. L'alumne haurà de desenvolupar obligatòriament totes les activitats descrites fins al punt 5 (inclusivament). El punt 6 és treball opcional.

2. Creació d'un projecte d'equip de *Team Services* i d'una solució inicial protegida en el servidor

Per a la realització d'aquesta pràctica és necessari haver seguit els passos per a la creació d'un projecte d'equip mitjançant *Team Services* en el núvol, haver donat d'alta als membres de l'equip de pràctiques en aqueix projecte i haver creat una solució en aquest projecte. Es recomana utilitzar com a nom per al servidor de *Team Services* el següent esquema de noms: *2017-upv-isw-<nom grup laboratori>-<nom equip>*. Per exemple, el servidor podria estar allotjat en *2017-upv-isw-3a11-equip1.visualstudio.com*

Aquestes tasques prèvies les realitzarà un únic membre de l'equip (Team Master).

La realització d'aquestes activitats ha sigut descrita en els seminaris de teoria dels temes 2 i 3 (S1 i S2, respectivament). Abans de passar als següents punts s'han d'haver realitzat aquestes activitats (consulte el material existent en Poliformat relatiu a aquests seminaris). En finalitzar aquestes tasques, l'equip de desenvolupament tindrà un projecte d'equip amb una solució Visual Studio que conté les carpetes de solucions *Presentation*, *Library*, *Testing* i, a més, un projecte de modelat tal com es mostra en la Figura 1. **NOTA:** el nom del projecte de modelatge, de la solució i de les captures pot diferir mínimament per variacions en la configuració personal del entorn.

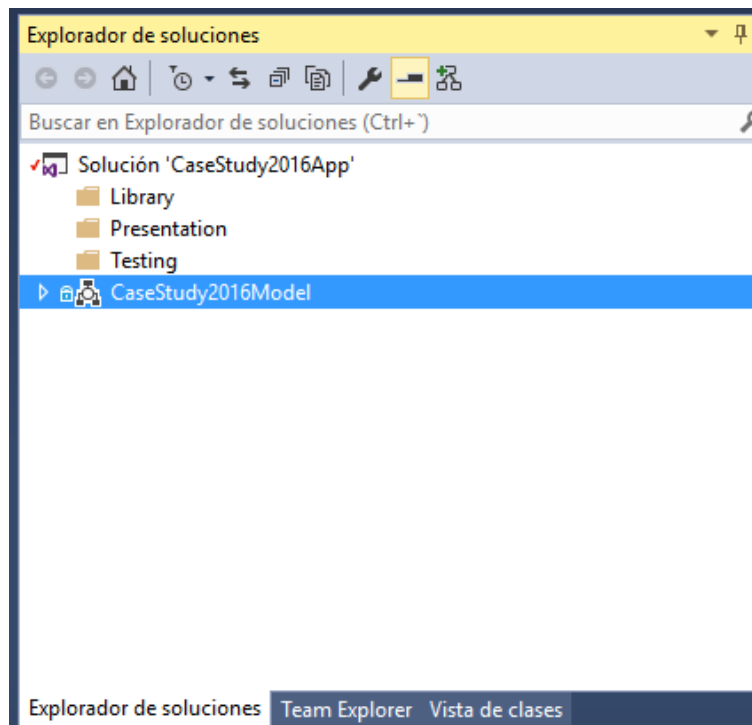


Figura 1. Solució inicial amb les carpetes de solucions i projecte de modelatge

3. Connexió al projecte existent des de Visual Studio

En aquest moment, cada membre de l'equip pot connectar-se des de *Visual Studio* **de forma individual** al projecte de *Team Services*, descarregar del servidor en el núvol una còpia de la solució i assignar-la a un àrea de treball local (directori local privat¹) per a treballar amb ella. D'aquesta manera cada alumne podrà realitzar les següents activitats **de forma individual**. Per a tot açò realitze els següents passos:

- a. Inicie l'eina de desenvolupament *Microsoft Visual Studio 2015 Enterprise*
- b. Inicie sessió amb el seu compte d'usuari Microsoft:
 - *Archivo->Configuración de La cuenta->Iniciar Sesión*
- c. Connecte's al seu projecte d'equip
 - Seleccione *Team Explorer* (marc de la dreta) i a continuació seleccione el botó de connexió a *Team Services* (endoll de color verd). Finalment connecte's al seu projecte d'equip tal com s'observa en la següent Figura 2 prement botó dret de ratolí.

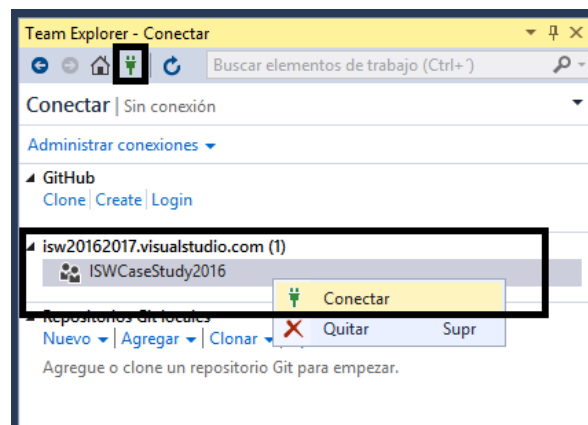


Figura 2. Connexió a projecte d'equip

- Alternativament, en cas que no li aparega el projecte d'equip tal com es mostra en la figura anterior connecte's al servidor de *Team Services* de la següent forma *Team Explorer -> Administrar Conexiones -> Conectar a proyecto de equipo* (veure Figura 3).

¹ **Important:** en els laboratoris del DSIC es **recomana** utilitzar el **directori privat** de cada usuari a partir de la ruta C:\Users\nomUsuari\... però **no** utilitzar la unitat W:, que es una unitat de xarxa en la que pot pedres la connexió, produint errades en la sincronització i errades en Visual Studio.

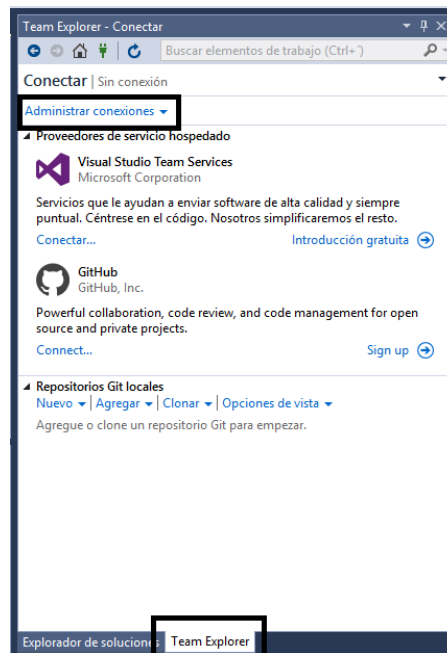


Figura 3. Connexió a servidor

- Seleccione i connecte amb el projecte d'equip (Figura 4).

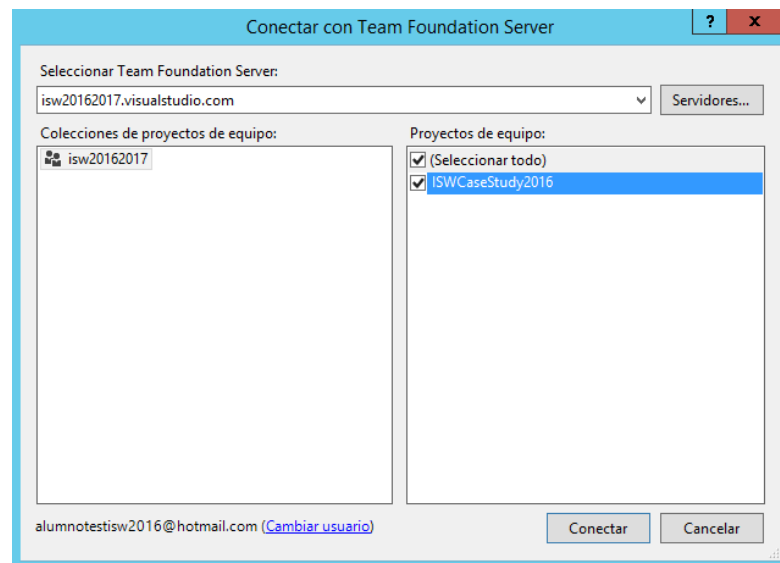


Figura 4. Selecció de projecte d'equip en el servidor

- a. Configure les associacions de l'espai de treball. Per a poder treballar amb una versió local de les solucions que es creuen en el seu projecte d'equip ha de configurar la seua àrea de treball local. Per a açò (veure Figura 5) faça clic en *Home -> configurar Las asignaciones del área de trabajo*, definisca la carpeta local on s'emmagatzemarà la seua àrea de treball i preme el botó *Asignar y obtener* (Figura 5).

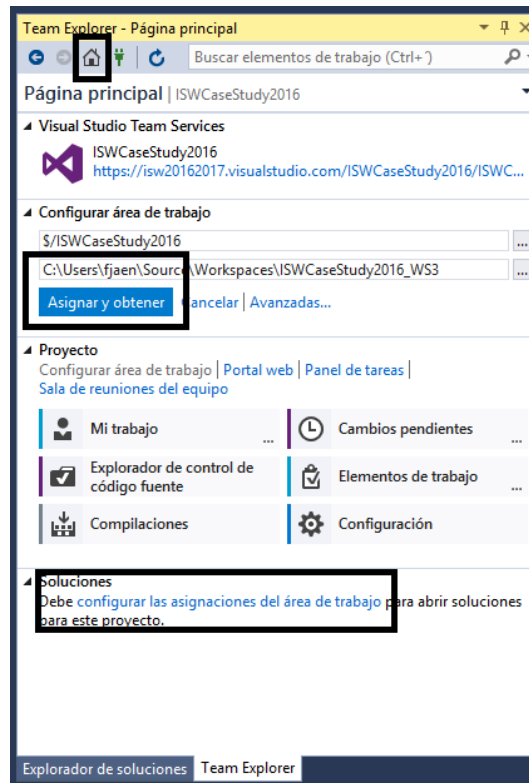


Figura 5. Configuració de l'àrea de treball local

- b. Òbriga la solució creada pel *Team Master*. Una vegada s'ha connectat al projecte d'equip, ha obtingut i assignat la seua àrea de treball local pot obrir la solució que el *Team Master* ha creat prèviament (en el punt 2 d'aquesta guia). D'aquesta forma obtindrà una còpia de la solució en la seua àrea de treball local. Per a açò faça doble clic sobre la solució que es mostra (tal com apareix en la Figura 6)

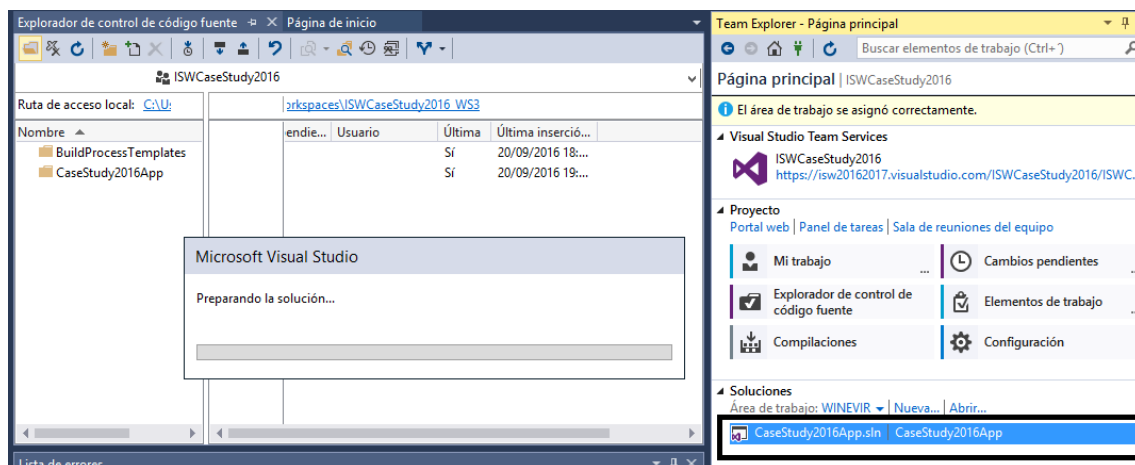


Figura 6. Obrint solució existent en l'àrea de treball local

- c. Vaja a l'explorador de solucions (pestanya "*Explorador de soluciones*" en la part dreta inferior de *Visual Studio*) i comprova que té una solució com la qual el *Team Master* ha creat i protegit en el servidor originalment tal com es mostrava en la Figura 1.

4. Introducció a les eines de depuració de Visual Studio

Les eines de depuració de *Visual Studio* són un element fonamental per al desenvolupament d'aplicacions en *Visual Studio* i la detecció de *bugs* en el codi. Aquestes eines permeten establir punts de ruptura en el codi, executar codi pas a pas, inspeccionar valors de variables en temps d'execució, etc.

4.1 Creació d'un projecte de consola

Per a iniciar-nos en les labors de depuració anem a crear un projecte de consola en la carpeta de solucions **Testing**. Per a açò:

- Creu una subcarpeta en *Testing* denominada *Lab1*
- Botó dret ratolí en carpeta *Lab1* -> *Agregar* -> *Nuevo Proyecto*-> *Aplicación de consola* i cridar a la aplicació **HelloWorldApp**

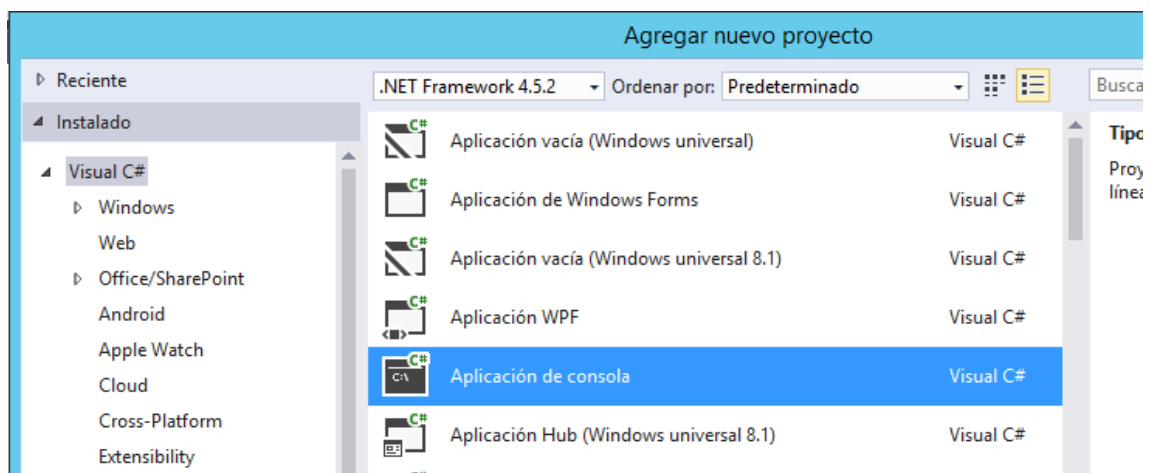


Figura 7. Creació aplicació de consola HelloWorldApp

- Una vegada generada l'aplicació observe les diferents finestres que Visual Studio li proporciona (veure Figura 8): quadre d'eines (esquerra), explorador de solucions (dreta a dalt), propietats (dreta a baix), editor de codi (centre a dalt), llistes d'errors i eixida (centre a baix).
- En el cas que el projecte que acaba de crear no aparega en l'explorador de solucions ressaltat en negreta (projecte d'inici) haurà d'indicar-li a Visual Studio que aquest és el seu projecte d'inici actual a l'efecte de compilació i execució. Per a açò preme el botó dret del ratolí sobre el projecte *HelloWorldApp* i selecciona l'opció de menú **Establecer como proyecto de Inicio**. Observe que el nom del projecte ha passat a estar ressaltat en negreta.

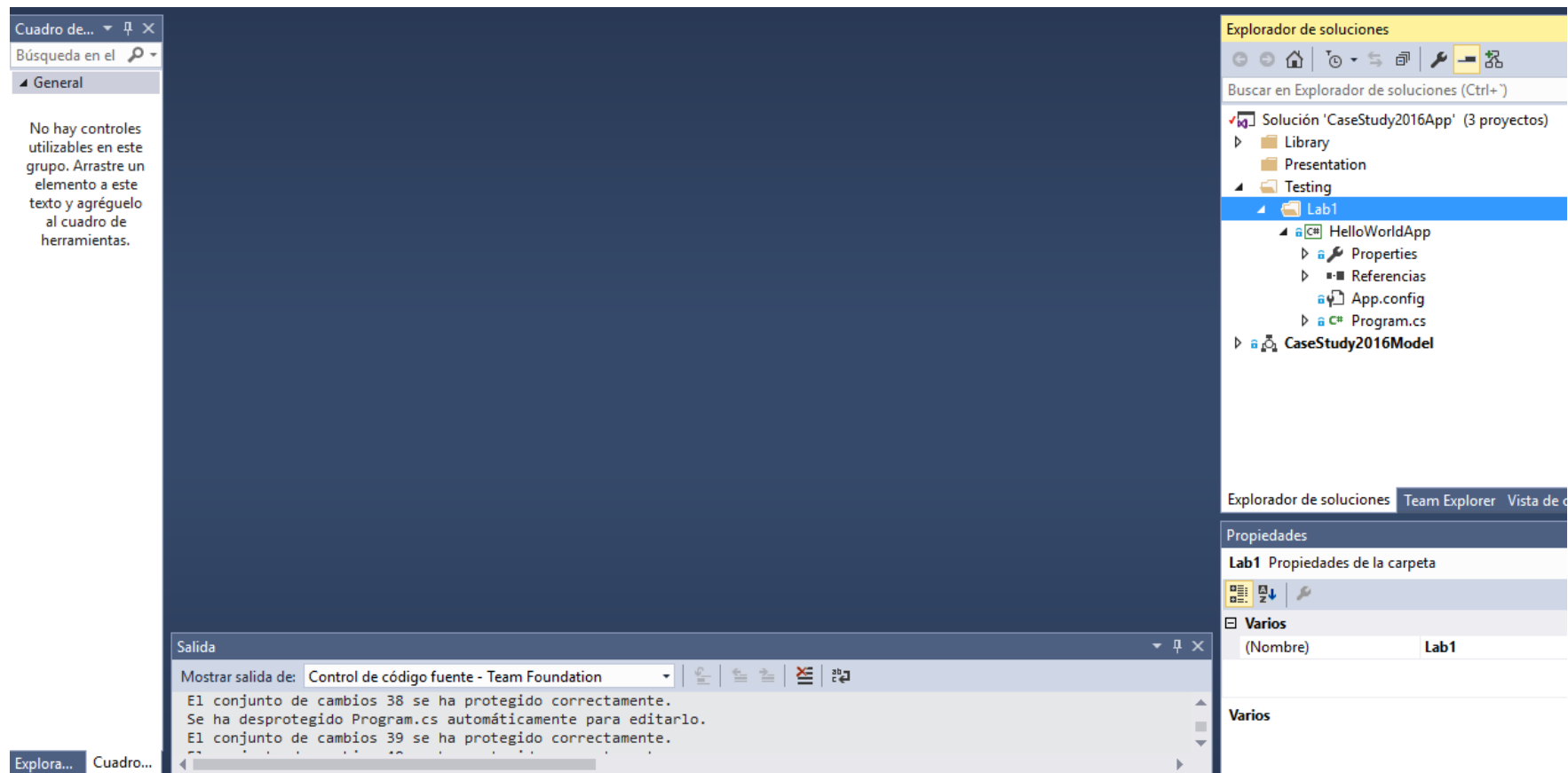


Figura 8. Finestres de treball de Visual Studio

4.2 Compilar i executar

Per a compilar i executar aquesta aplicació preme el triangle verd en la zona superior de l'eina de desenvolupament. Per la finestra d'eixida observarà el resultat de la compilació i de l'execució, encara que el programa actual no realitza cap acció.

4.3 Exploració d'errors de compilació

Òbriga el fitxer *Program.cs* i introduïska un error artificialment en el mètode *main*, per exemple, escrivint la sentència *fadsfsde*. Si torna a compilar el seu programa observarà la llista d'errors en la finestra corresponent. Si fa clic sobre l'error, el cursor apareixerà en la zona del seu programa on es troba el mateix.

4.4 Creant una aplicació HelloWorld en C#

Per a poder practicar les capacitats de depuració anem a generar un senzill programa d'exemple. Obriga el fitxer *Program.cs* i modifiqui el seu contingut perquè quede com segueix.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HelloWorldApp
{
    class Program
    {
        private static List<string> l;
        private static void InitList()
        {
            l = new List<string>()
            {
                "Australian",
                "Mongolian",
                "Russian",
                "Austrian",
                "Brazilian"
            };
        }

        static void Main(string[] args)
        {
            InitList();
            l.Sort();
            foreach (string s in l)
            {
                Console.WriteLine(s);
            }
        }
    }
}
```

Observe que el programa anterior té unes sentències *using* que actuen com a directrius del compilador, fent que es puguin utilitzar tipus definits en l'espai de noms importat sense necessitat d'especificar-los de forma explícita en el codi. A continuació, es defineix un espai de noms *HelloWorldApp* i en ell la classe *Program*. Esta classe conté la definició d'una llista, un mètode d'inicialització de la mateixa i un mètode estàtic *main*. El programa crea una llista de *strings*, l'ordena alfabèticament i després la recorre i visualitza per pantalla el seu contingut.

Aquest codi serà usat per a practicar les capacitats de depuració. Compile i execute el programa. De nou veurà que ha aparegut una finestra de consola, s'ha executat el programa ràpidament però no ha pogut observar l'eixida generada pel mateix.

4.5 Execució d'un programa pas a pas

En la majoria de IDEs actuals és possible incloure punts de ruptura on es deté l'execució per a poder des d'aqueix moment executar sentència a sentència el codi i anar observant el seu comportament de forma controlada. Per a practicar aquest aspecte realitzarem les següents accions

- Inserisca un punt de ruptura en la línia de codi `InitList()`. Per a açò preme sobre el ratolí en la columna de color gris que apareix a l'esquerra del codi, a l'altura de la línia on desitja interrompre l'execució (veure Figura 9)

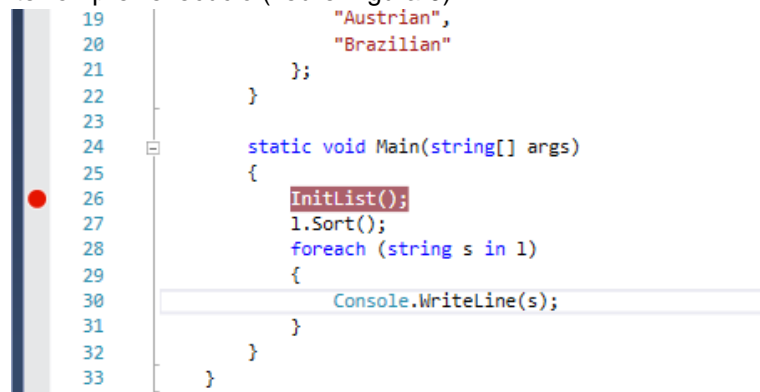


Figura 9. Inserció de punt de ruptura

- Compile i iniciï la depuració del codi (preme el triangle verd) i observe com l'execució es deté en la línia on es va introduir el cercle roig (apareix una fletxa groga indicant on es troba l'execució). Si observa la finestra de consola creada, aquesta apareixerà buida.
- Execute pas a pas l'aplicació. Per a açò utilitze els controls d'execució pas a pas (veure Figura 10). Pot detenir l'execució (quadrat roig), executar la següent instrucció (fletxa vertical cap avall o F11), executar tot un procediment/mètode de forma completa sense entrar en ell (fletxa corbada a la dreta o F10) o executar totes les sentències que hi ha en un procediment per a eixir d'ell (fletxa cap amunt o Majúsc+F11).



Figura 10. Execució pas a pas

4.6 Observant els valors de dades del programa

Mentre el programa es troba en execució és possible observar els valors d'objectes, atributs, variables, etc. Açò és **extremadament útil** quan està depurant el seu programa. Per a observar el valor de qualsevol element situe el cursor sobre el mateix.

- Quan el seu programa arribi a l'execució de la sentència `l.Sort()` pose el cursor sobre la variable `l` i observe el seu contingut (veure Figura 11).

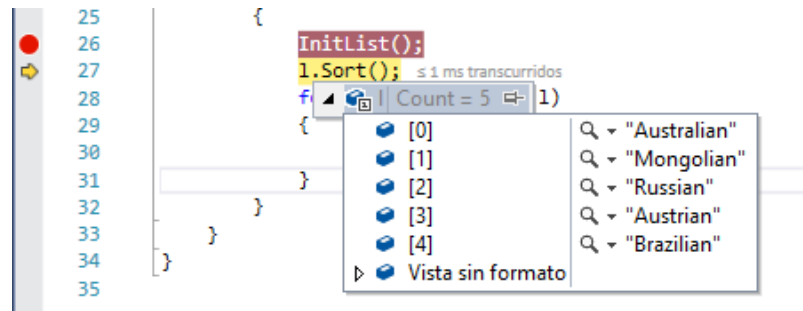


Figura 11. Inspeccionant valors de dades durant l'execució

- b. Execute aqueixa sentència passe a pas (F11) i torne a observar la llista comprovant que s'ha ordenat alfabèticament.
- c. Seguisca executant pas a pas el seu programa i observe com per la finestra de consola apareix el contingut de la llista (Figura 12).

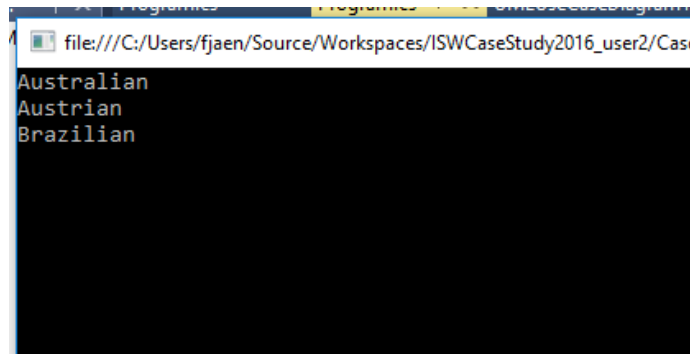


Figura 12. Eixida per consola durant l'execució

- d. Pot tornar cap a arrere en l'execució simplement arrossegant la fletxa groga cap amunt a la línia de codi des d'on desitge repetir l'execució del seu programa.

5. Operacions bàsiques de control de versions

Una dels avantatges d'utilitzar un entorn de desenvolupament en equip com *Team Services* és la possibilitat de tenir control i gestió sobre les diferents versions del projecte que es vagen generant. De fet, quan va crear el seu projecte va indicar què sistema de control de versions utilitzar *GIT* o el control de versions de *Team Foundation*. Un sistema de control de versions és una combinació de tecnologies i pràctiques per a controlar els canvis realitzats en els diferents elements que componen el projecte, en particular en el codi font, documentació, conjunts de proves, etc. És convenient conèixer una nomenclatura prèvia relacionada amb els sistemes de control de versions:

- **commit:** realitzar un canvi en el projecte. Açò suposa emmagatzemar un canvi en el repositori del control de versions de manera que puga ser incorporat en llançaments futurs del projecte.
- **checkout:** el procés d'obtenir una còpia del projecte des del repositori. Un checkout produeix un arbre de directoris anomenat "còpia funcional" des del qual els canvis seran enviats de tornada al repositori original.

- **conjunt de canvis (changeset)**: un conjunt de revisions de fitxers enviats junts com una unitat conceptual.
- **conflicte**: succeeix quan dos o més persones intenten realitzar diferents canvis en la mateixa porció de codi.
- **bloqueig (lock)**: declaració d'un intent exclusiu per a canviar un fitxer o directori en particular.

En la manera més simple de treball, en el repositori de control de versions es genera una nova versió de codi a partir d'un conjunt de canvis cada vegada que un usuari realitza una operació de *commit* (protegir). Per a practicar aquest aspecte:

- Modifique qualsevol part del seu programa. Per exemple, la classe *Program* perquè el primer element de la llista siga un altre (cada membre de l'equip pot triar una cadena diferent ("Italian", "spanish", etc.)
- Vaja a Team Explorer -> Cambios pendientes -> Proteger (afegisca un comentari descriptiu de la nova versió o modificació introduïda).
- Resolució de conflictes: és possible que, si un altre usuari va generar una nova versió abans que vostè i vostè no va descarregar l'última versió existent en el repositori, es produïsquen conflictes que cal resoldre. La resolució de conflictes es pot fer de forma manual decidint què codi conservar en la nova versió a generar mitjançant l'eina de resolució de conflictes (veure Figura 13).

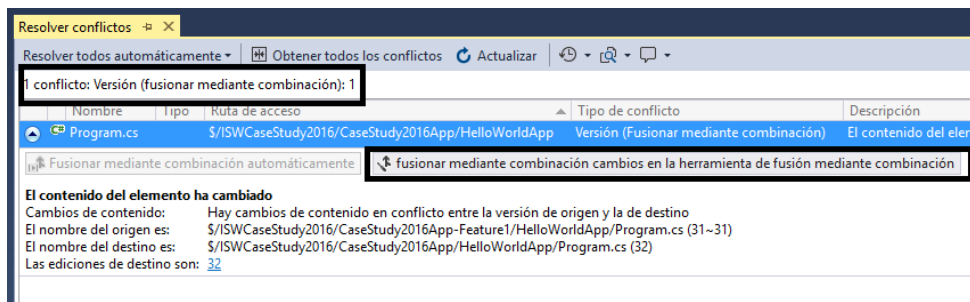


Figura 13. Llistat de conflictes

- S'ha de decidir què versió és la correcta. Com es pot observar en la Figura 14 tenim una versió amb un element en la llista ("Italian"), i una altra versió amb un element ("Australian"). Seleccionem com a correcta una de les versions i finalitzem l'operació de commit (protegir). En fer açò incorporarem els canvis al repositori central generant una nova versió del projecte.

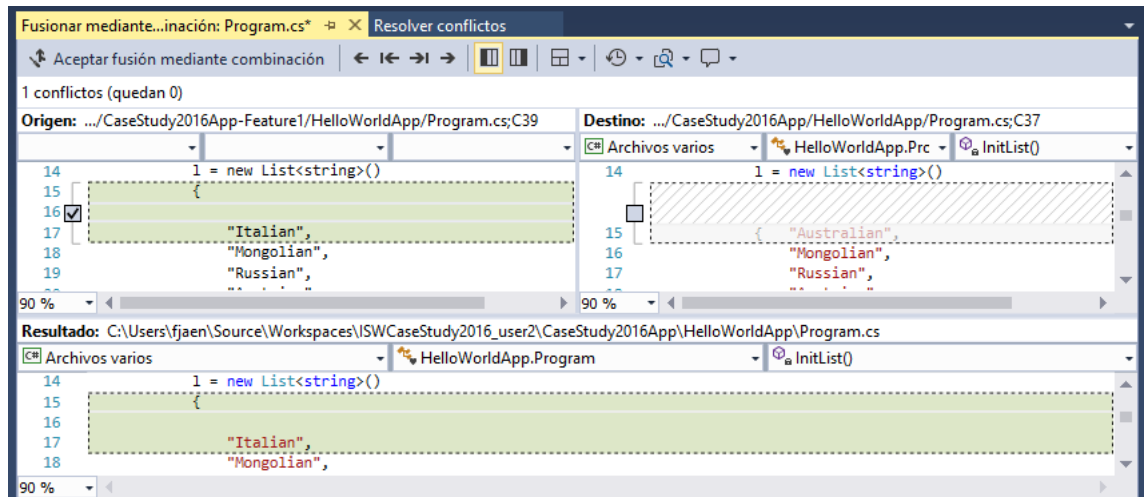


Figura 14. Resolució de conflictes en les versions de codi

- f. Pot observar l'historial de canvis de qualsevol element del seu projecte mitjançant l'Explorador de Versions de codi. Vaja a Team Explorer -> Explorador de control de codigo fuente -> Seleccione en l'explorador l'element a explorar (per exemple el fitxer Program.cs) -> Botó dret de ratolí -> Ver historial
- g. De la mateixa forma pot observar els diferents conjunts de canvis generats en la Web associada al seu projecte de Team Services. Per a açò: Seleccione la pestanya CODE->Pose el ratolí sobre l'element a inspeccionar -> Faça clic sobre la imatge de tres punts suspensius "... " -> View history

6. (Treball Opcional) Operacions Avançades amb l'Explorador de control de versions

En aquesta secció practicarem dos conceptes addicionals relatius al control de versions:

- **branca (branch):** és una còpia del projecte, sota el control de versions, però aïllat, de manera que els canvis realitzats en una branca no afecten a la resta del projecte i viceversa. Les branques també són conegudes com "Líneas de desenvolvament".
- **merge:** moure un canvi d'una branca a una altra.

6.1 Generant una línia de desenvolupament (branca)

Per a generar una línia de desenvolupament pròpia de manera que independitze els canvis realitzats per vostè dels realitzats per uns altres es pot procedir de la següent manera:

- a. El *Team Master* ha de concedir permisos de creació de branques als usuaris del seu equip que estiguen autoritzats. Per a açò, en la Web de gestió del projecte de *Team Services* anirà al panell de control prement en



I a continuació en la pestanya "Versión Control" -> Add User per a afegir permisos específics d'accés per a un membre de l'equip. Concedisca permisos de gestió de branques als usuaris que desitge (veure Figura 15).

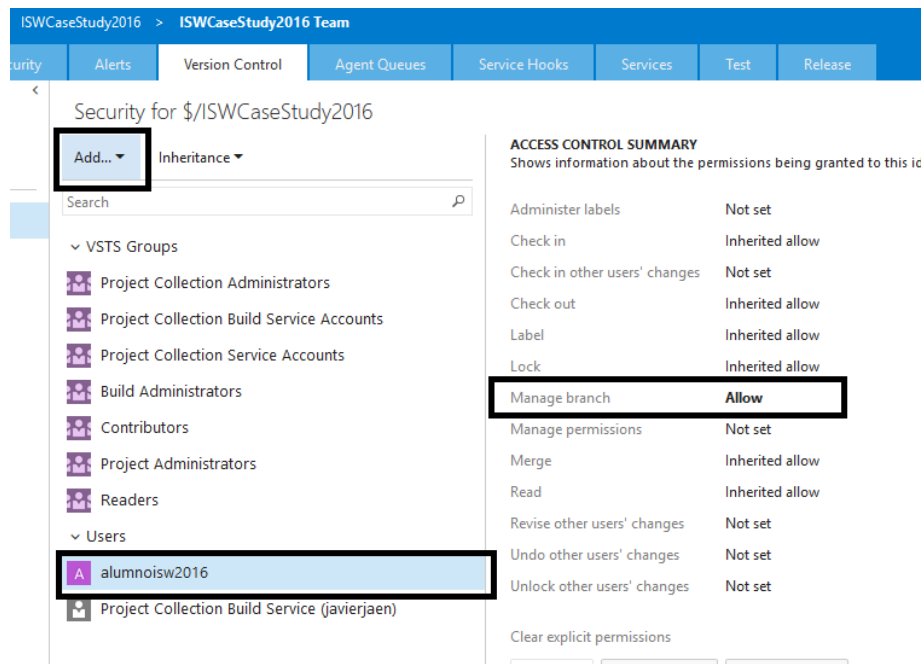


Figura 15. Concessió de permisos de gestió de branques a un usuari

- b. Una vegada concedits aquests permisos, l'usuari en qüestió pot crear branques. En *Team Explorer* obliga l'explorador de control de codi font (veure Figura 16)

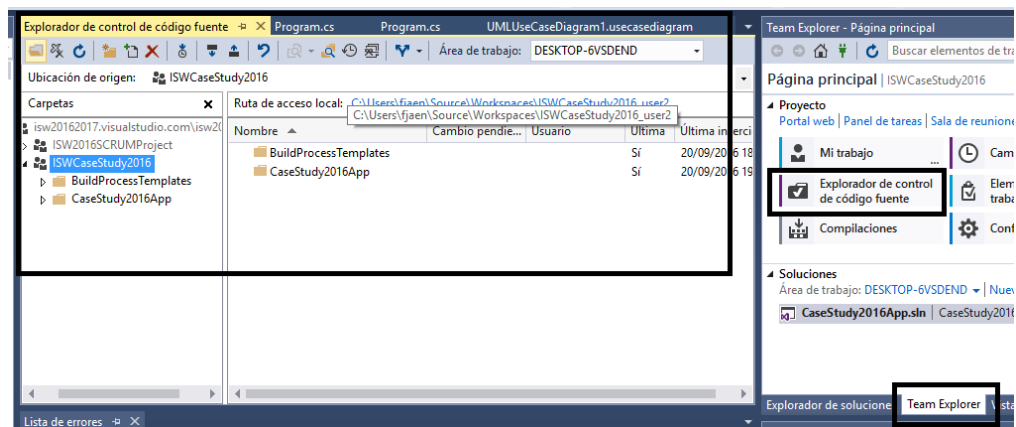


Figura 16. Obrint l'explorador de control de codi font

- c. Seleccione en l'explorador la carpeta que conté la branca des de la qual vol ramificar i genere una ramificació tal com s'observa en la Figura 17.

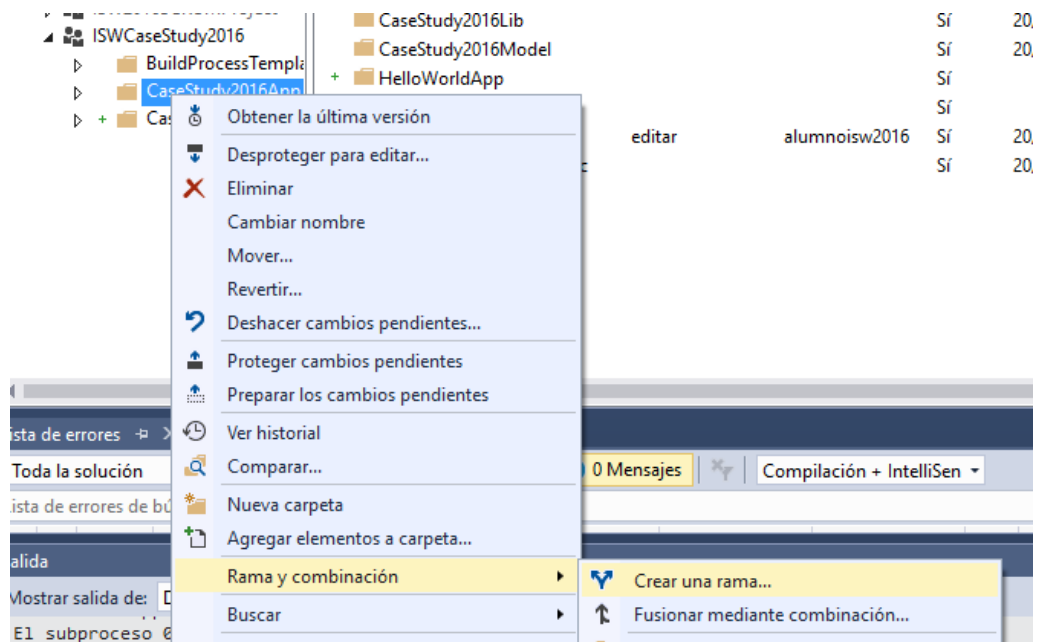


Figura 17. Creant una branca (línia de desenvolupament)

- d. En el nostre cas generarem la nova branca prenent com a versió de codi la que tenim en la nostra àrea de treball local (veure Figura 18) i proporcionem una destinació en el repositori central per a la branca que va a crear. En el nostre cas la branca nova serà situada en el servidor en `$/ISWCaseStudy2016/CaseStudy2016App-Feature1`
 NOTA: En aquesta pràctica cada membre de l'equip pot crear una branca diferent amb noms "...-Feature2", "...-Feature3", etc.

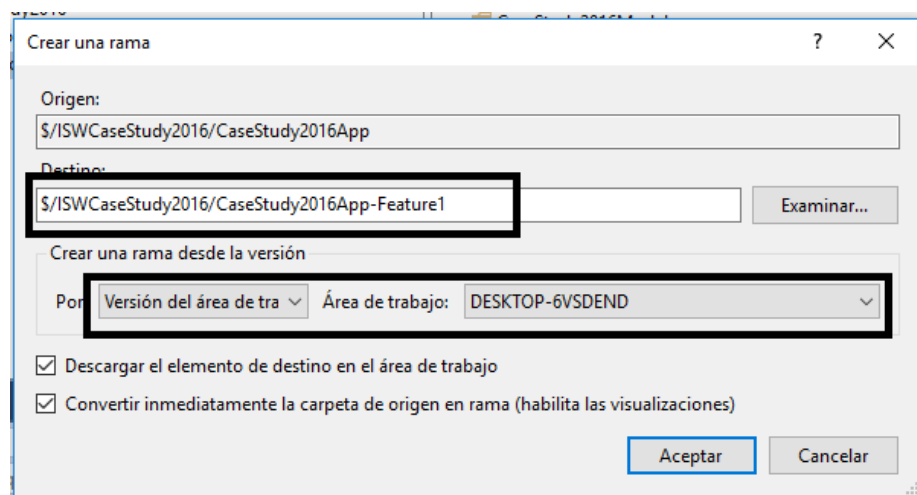


Figura 18. Creació de branca a partir de la versió de codi en àrea de treball local

S'obté una nova ramificació com pot observar-se en la Figura 19.

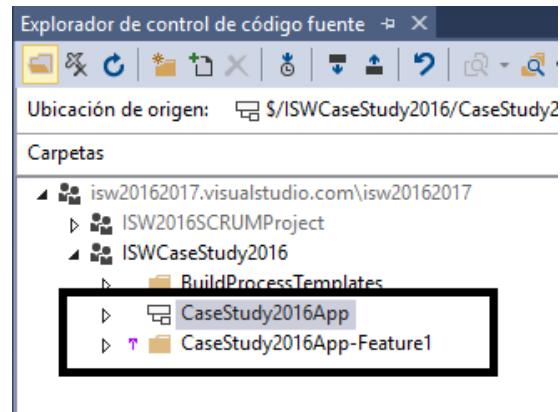


Figura 19. Rama creada

6.2 Protegint una branca (commit)

Com ja s'ha vist anteriorment, en qualsevol moment es pot protegir la versió local associada a una branca de manera que es genere un conjunt de canvis, es pugi la versió al control de versions generant una nova versió del producte. Per a açò n'hi ha prou amb protegir la solució i si escau resoldre els possibles conflictes que sorgisquen amb la versió anterior si aquesta ha sigut modificada per un altre usuari.

- Modifique de nou la classe *Program* perquè l'algun element de la llista siga un altre (cada membre de l'equip pot triar una vegada més una cadena diferent ("Italian", "spanish", etc.)
- En aquest punt anem a protegir la solució de la branca secundària que acabem de crear de manera que tindrem des d'aquest moment aquesta nova versió en el servidor. Associarem un comentari descriptiu per a aquest conjunt de canvis (veure Figura 20).

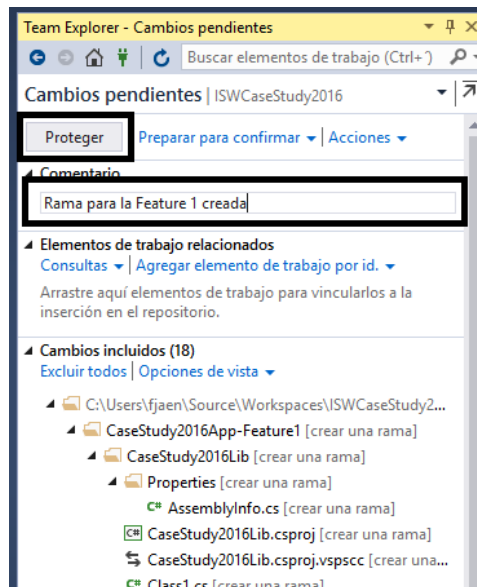


Figura 20. Protecció de la solució

- c. Pot observar les dues branques existents en el servidor a través de l'aplicació Web de *Team Services* i comparar les dues versions de la classe *Program* en les dues branques creades.
- d. Es pot observar la jerarquia de branques existents en aquest moment seleccionant una de les branques en l'explorador de control de codi font en *Visual Studio* tal com es mostra en la Figura 21, i seleccionant *Rama y combinación* -> *Ver jerarquía*

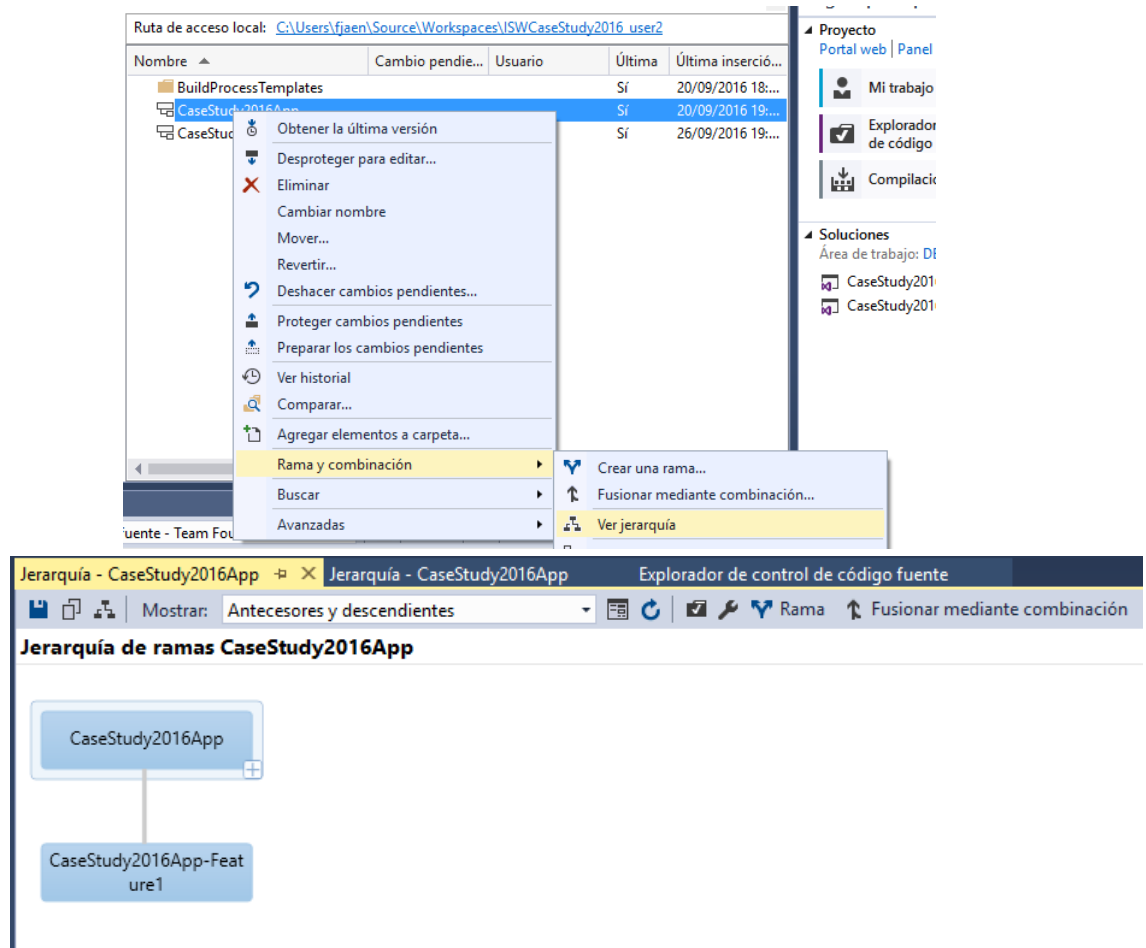


Figura 21. Explorant la jerarquia de branques

5.3 Combinació de branques (merging)

És possible combinar dues branques de manera que els canvis existents en les mateixes es combinen. Açò és útil per exemple per a incorporar els canvis d'una branca de desenvolupament a una branca de *release* o per a combinar dues branques de desenvolupament en una. En aquest cas anem a combinar (fusionar) la branca *CaseStudy2016App-Feature1* (branca origen) amb la branca principal *CaseStudy2016App* (branca de destinació). Per a açò:

- a. Seleccionarem la branca d'origen a combinar en l'explorador de versions de codi de *Visual Studio*, botó dret de ratolí i a continuació l'opció de menú *Rama y Combinación* -> *Fusionar mediante combinación* (Veure Figura 22)

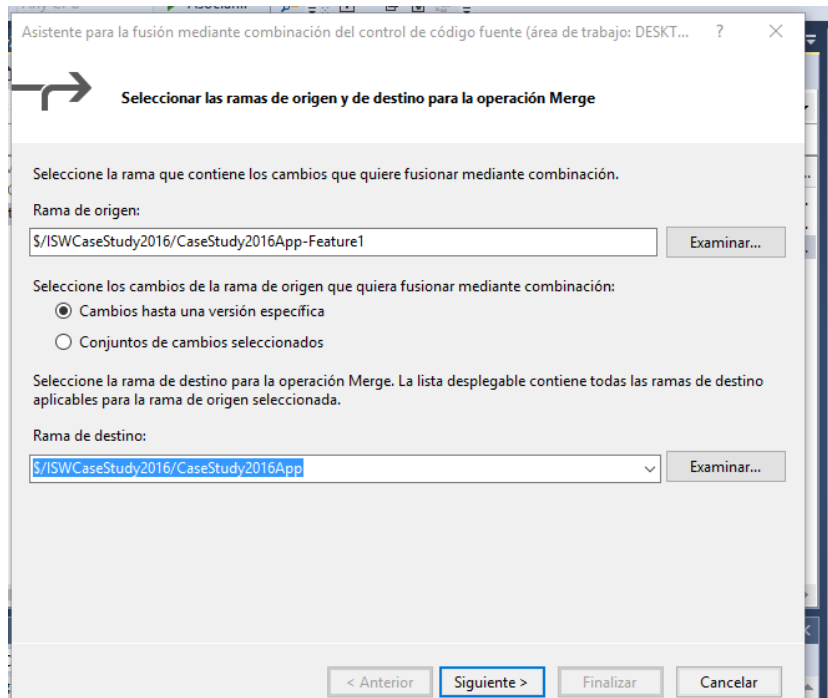


Figura 22. Combinació de dues branques

- b. En realitzar aquesta operació poden aparèixer conflictes. El conflicte es podrà resoldre automàticament o mitjançant l'eina de resolució de conflictes com vam veure anteriorment.
- c. Una vegada resolt el conflicte i realitzada la fusió incorporem els canvis al repositori mitjançant l'opció *Protegir* (veure Figura 23).

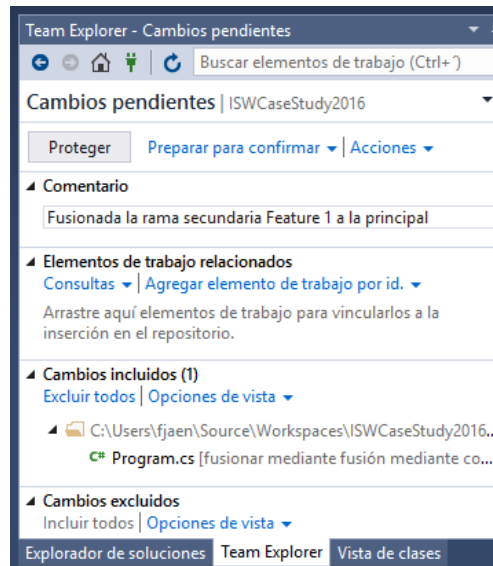


Figura 23. Protecció en el repositori de la fusió de branques

- d. Es pot comprovar mitjançant l'aplicació web que en el repositori central ambdues branques contenen ara la mateixa versió de la classe *Program*

Per a més informació sobre el model de control de versions en forma de branques en *Team Services* pot consultar la informació disponible en:

<https://www.visualstudio.com/nl-nl/docs/tfvc/use-branches-isolate-risk-team-foundation-version-control>

NOTA: L'equip de desenvolupament haurà de decidir quantes branques tenir, només una principal, una de desenvolupament i una altra principal, etc. Es recomana llegir el següent document <https://www.visualstudio.com/nl-nl/docs/tfvc/branch-strategically> per a decidir una adequada estratègia de ramificació.