Seminar

S2

Chapter 3.

Software Architecture

Software Development with Microsoft Visual Studio. Integration with Azure DevOps for Project management

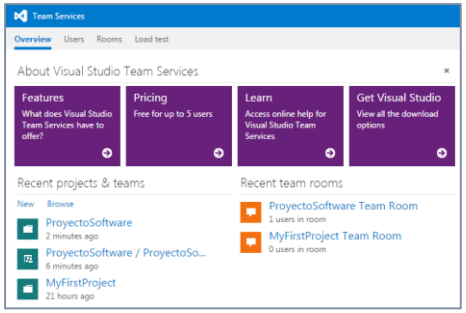
Software Engineering

Computer Science School
DSIC – UPV

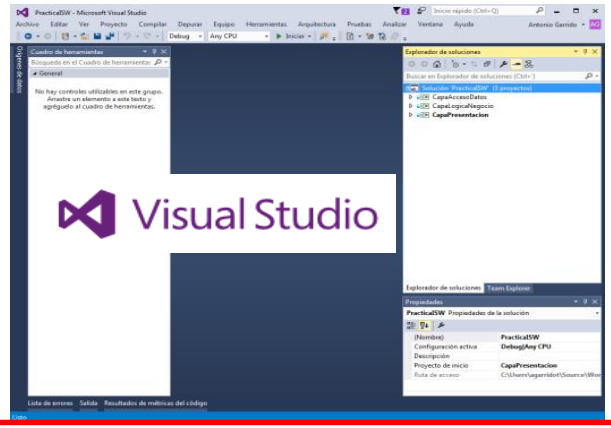
Goals

- Apply an agile methodology for software development using Azure DevOps combined with design and coding tasks with Microsoft Visual Studio

Part 1. Cloud Project Management (Seminar Chapter 2)

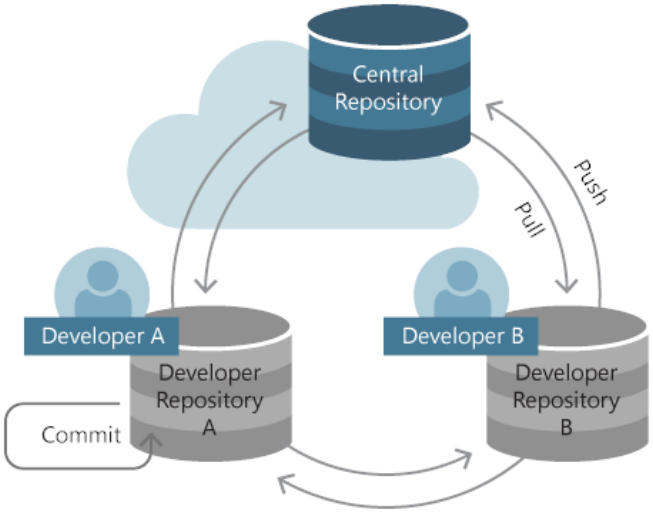


Part 2. Project Development with Visual Studio



Version Control

- Use version control to save your work and coordinate code changes across your team.

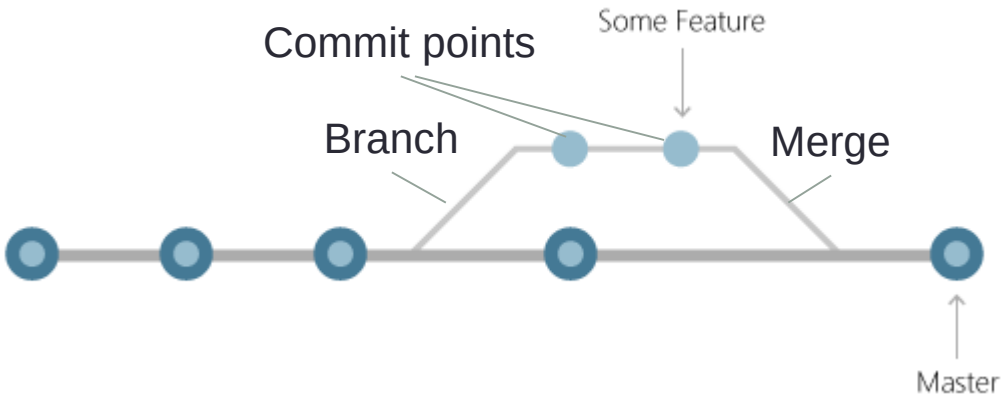


Git and TFS are available for Version Control

- https://docs.microsoft.com/en-us/azure/devops/repos/git/?view=vsts

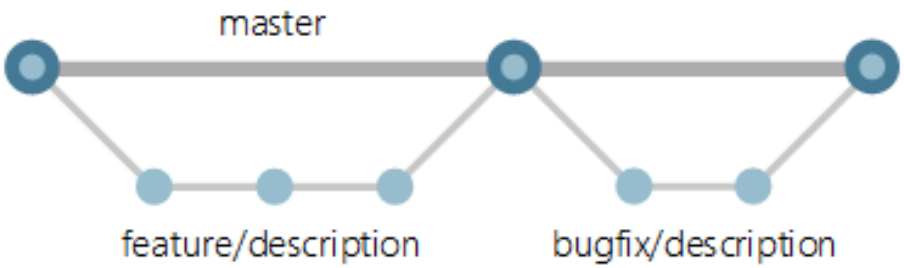
Git WorkFlow

- A normal workflow in Git is:
  - Clone an existing remote repository
  - Create a new branch for your work
  - Do you work on your personal branch
  - Commit your changes on your branch (locally)
  - Push the branch to share it with your team
  - Merge your branch with main branch when code is revised and ready



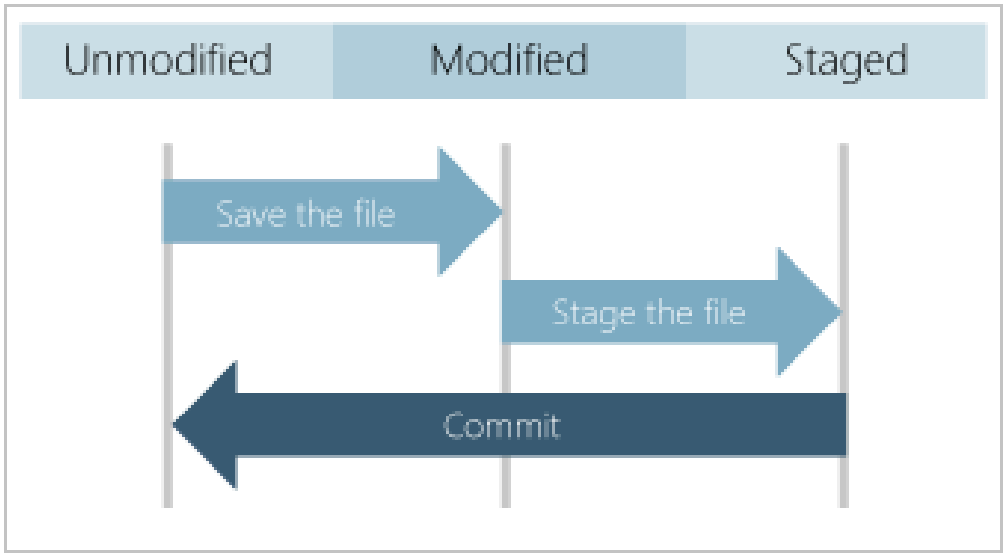
Git Branching Strategy

- Use a consistent naming convention for your feature branches to identify the work done in the branch. For instance
  - users/username/description
  - users/username/workitem
  - bugfix/description
  - features/feature-name
  - features/feature-area/feature-name
  - hotfix/description



How Git tracks changes

- Unmodified files - These files haven't changed since your last commit.
- Modified files - These files have changes since your last commit, but you haven't yet staged for the next commit.
- Staged files - These files have changes that will be added to the next commit.



Project Development with Visual Studio

- Create a software project using *Microsoft Visual Studio*, retrieving (and completing) the Project plan elaborated with *Azure Boards*

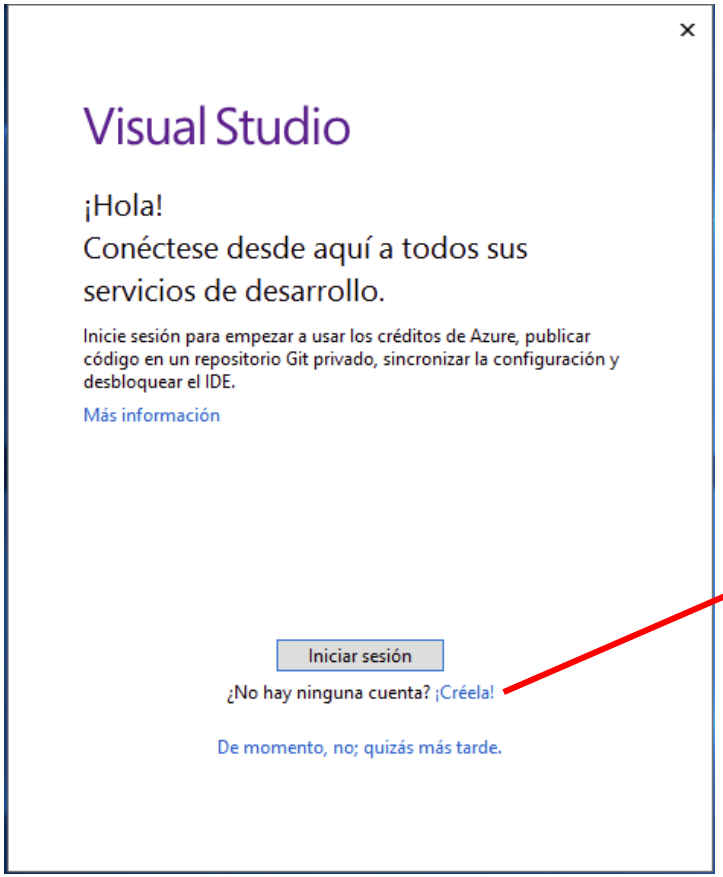
Steps:

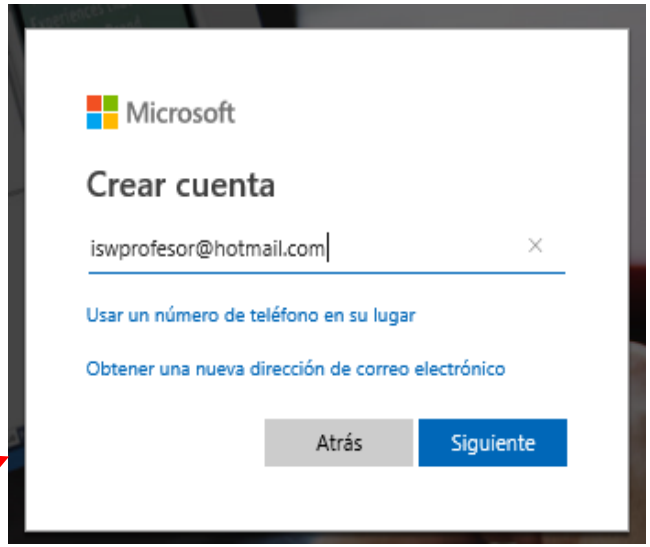
- ✓Create a Microsoft account (if not already done)
- ✓Create a Visual studio Solution project (First time)
- ✓Project Management with *Visual Studio*
- ✓Retrieve the Project from the repository into Visual Studio
- ✓Managing code conflicts

Create Account from Visual Studio

When Visual Studio is started, we have to **log in** with an existing account or **create a new one (same as the one used for Azure DevOps)**.
Once in the environment we can change the account at*: File > Account Settings …*

Start a session with the account used for Azure DevOps





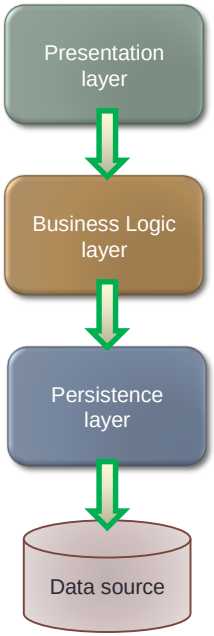
✓ Define Agile Iteration

We start with the following Agile definition for the current iteration.

SoftwareProject01 Team

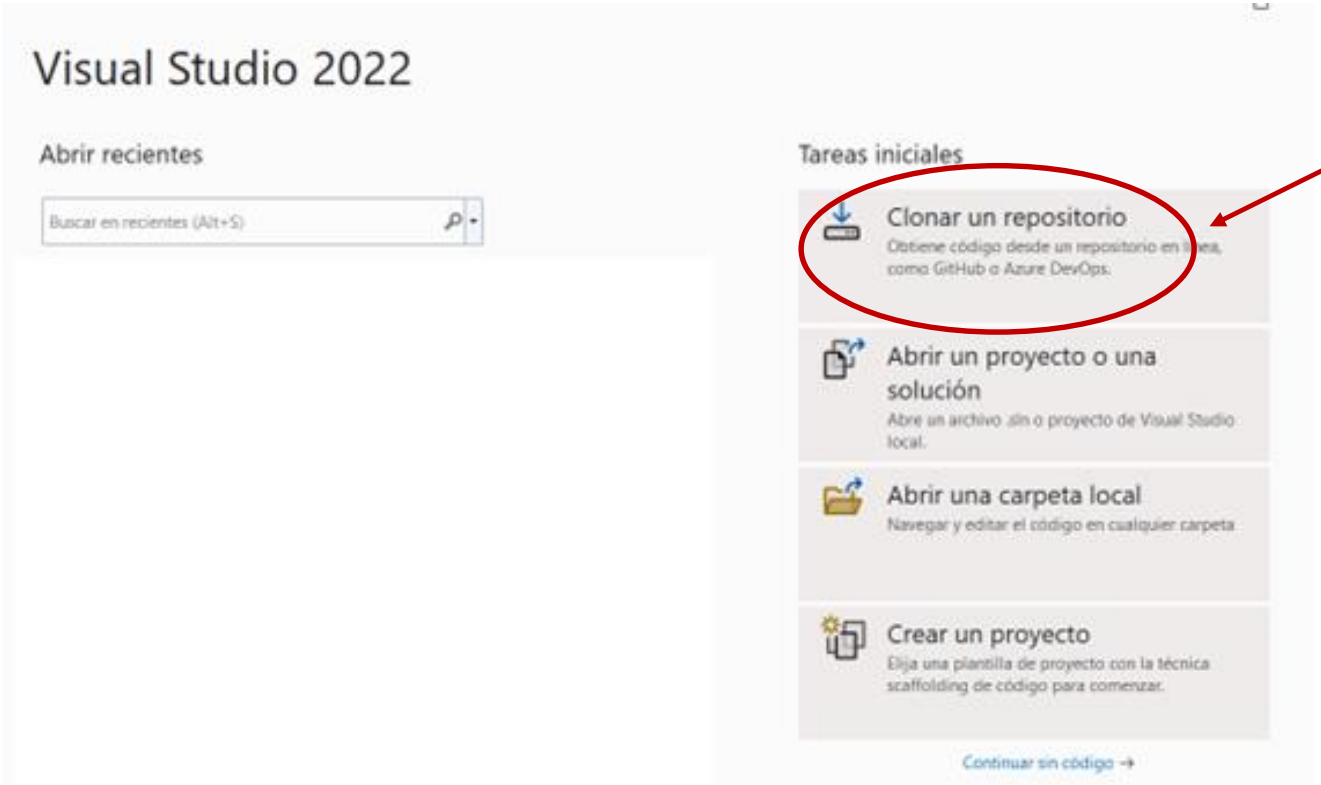
Backlog | Analytics | + New Work Item | View as Board | Column Options | ...

Order	Work Item Type	Title	State	Effor
1	Feature	Architectural Design Definition	New	
	User Story	Presentation Layer	Active	
	Task	Implement Visual Studio Presentation Layer	New	
	User Story	Business Logic Layer	Active	
	Task	Implement Visual Studio Business Logic layer	New	
	User Story	Persistence Layer	Active	
	Task	Implement Visual Studio Persistence Layer	New	
2	Feature	Testing Definition	New	
	User Story	Testing Project	New	
	Task	Create Testing Project	New	



Create Project in Visual Studio

- Main screen shows the most common tasks, including links to most recent projects

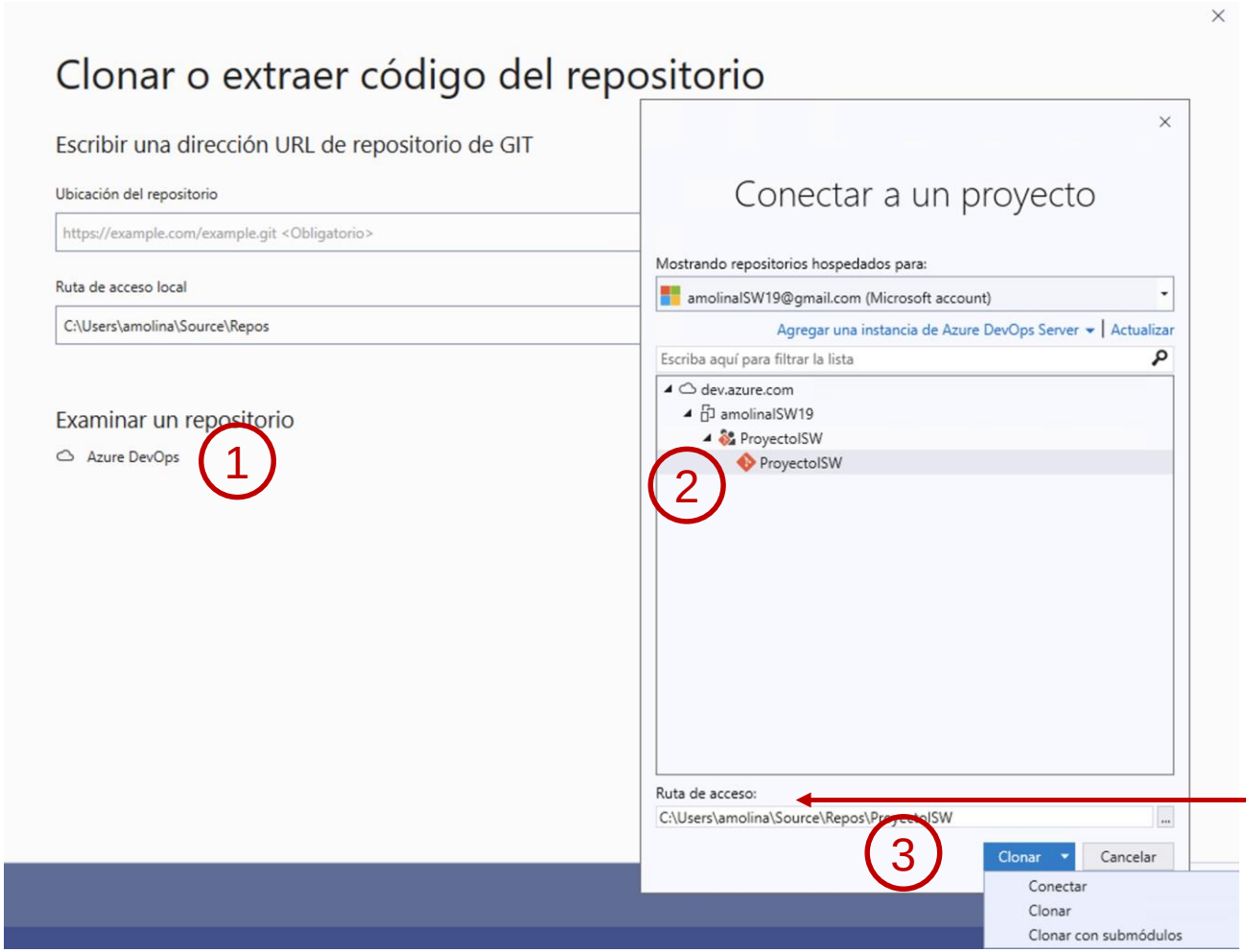


Select cloning repository to connect to our Azure DevOps Project and clone the code

Within VS it is also posible to do this by selecting File > *Clone*

Create Project in Visual Studio

Select the option to explore an Azure DevOps repository



Select the project.

Select the cloning option to connect to the Project and clone the code in a single step.

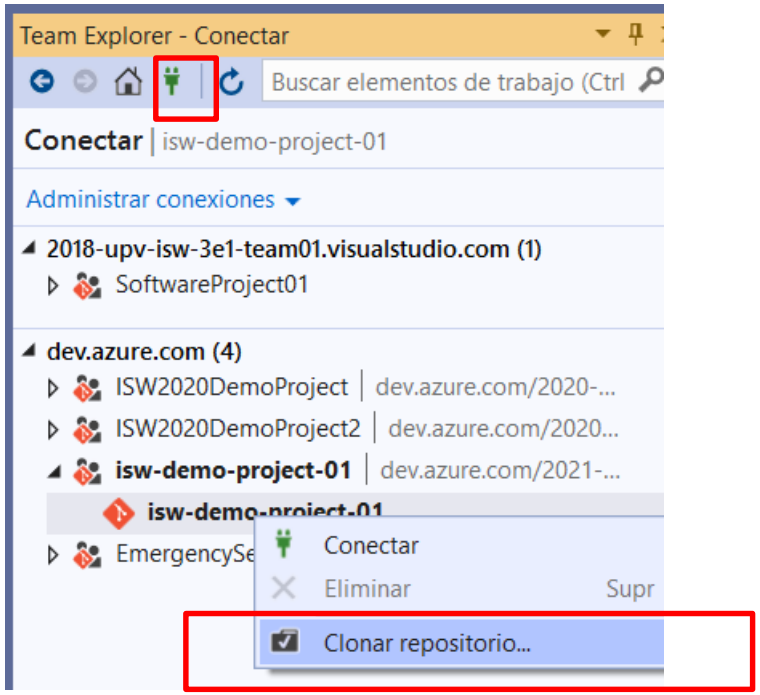
This must only be done by the MANAGER of the project.

The path where the files will be stored locally is shown.

✓ Cloning Repository (Alternative way)

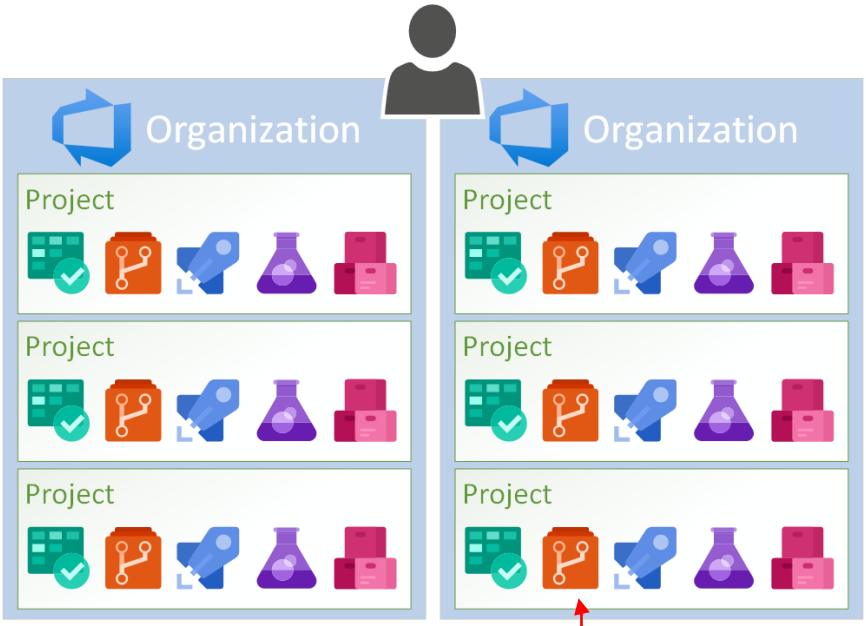
View > Team Explorer

To work with *Azure DevOps projects* from *Visual Studio*

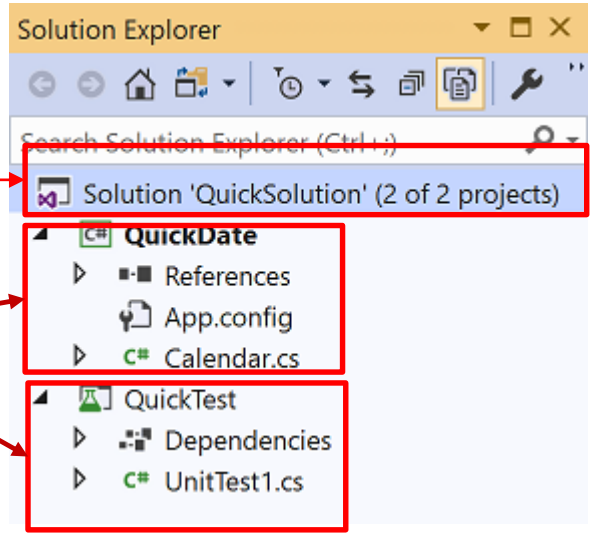


First Time, the Project leader clones the repository (Git Version Control).

Azure DevOps projects vs VStudio projects



In Visual Studio **a Solution is a collection of Projects.** You Will create several projects within the same Solution



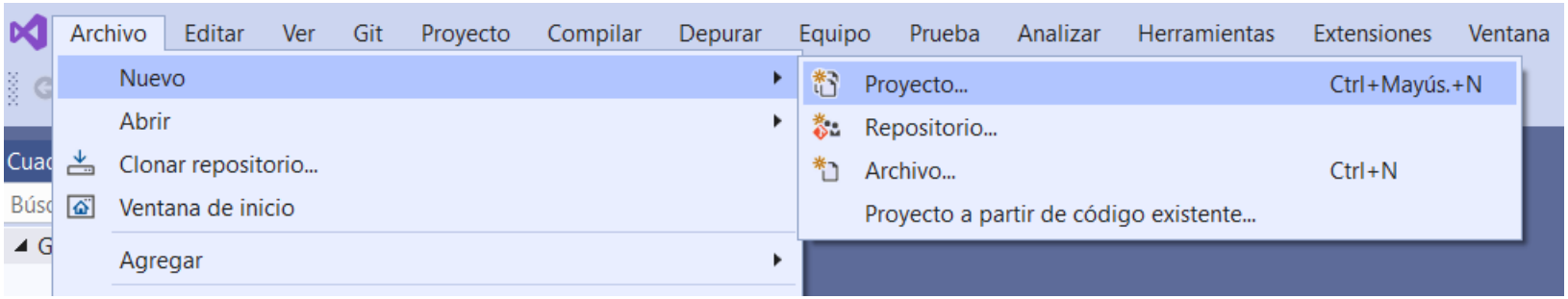
Projects in a solution might be class libraries, some executable applications, and some might be unit test projects or websites.

✓ Configuring workspace

FIle > New > Project

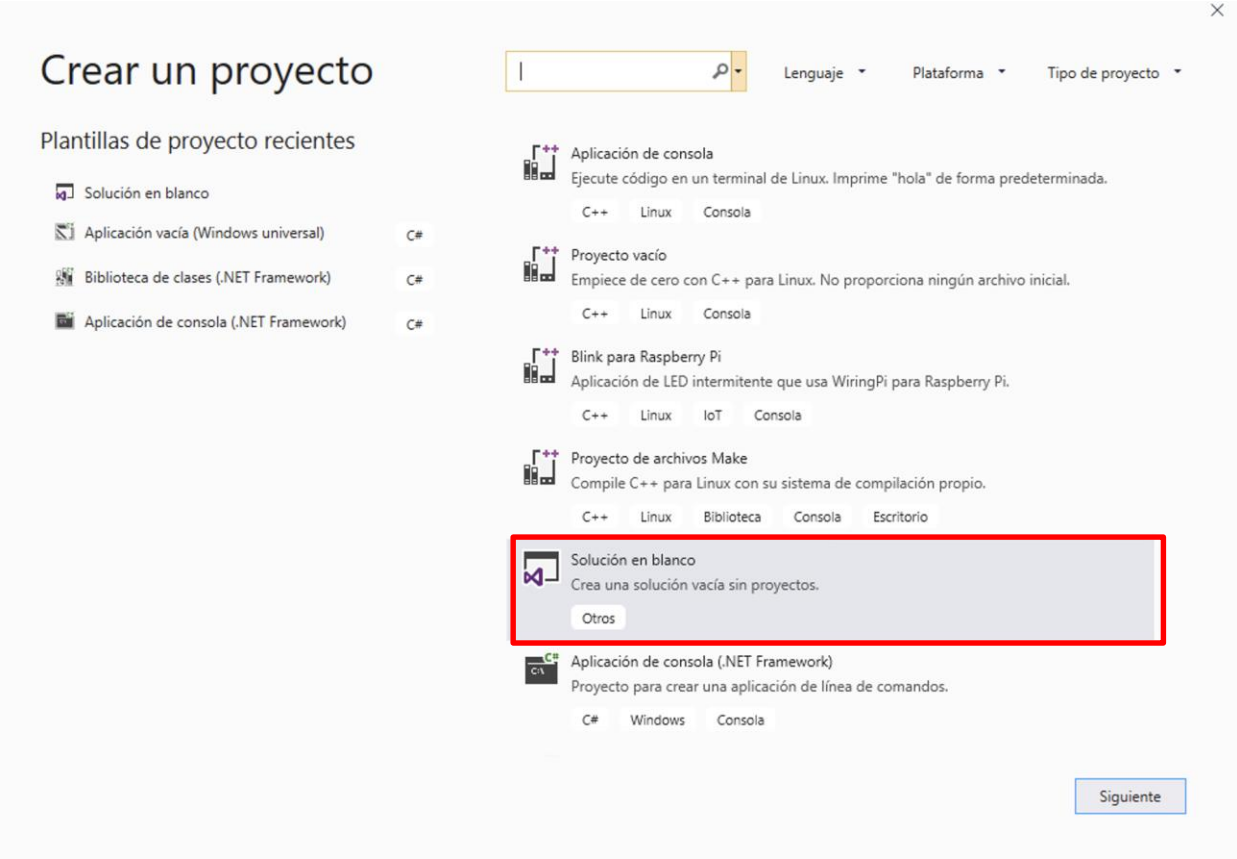
To work with *Azure DevOps projects* from *Visual Studio*

First Time, **the Project leader** creates a new solution.



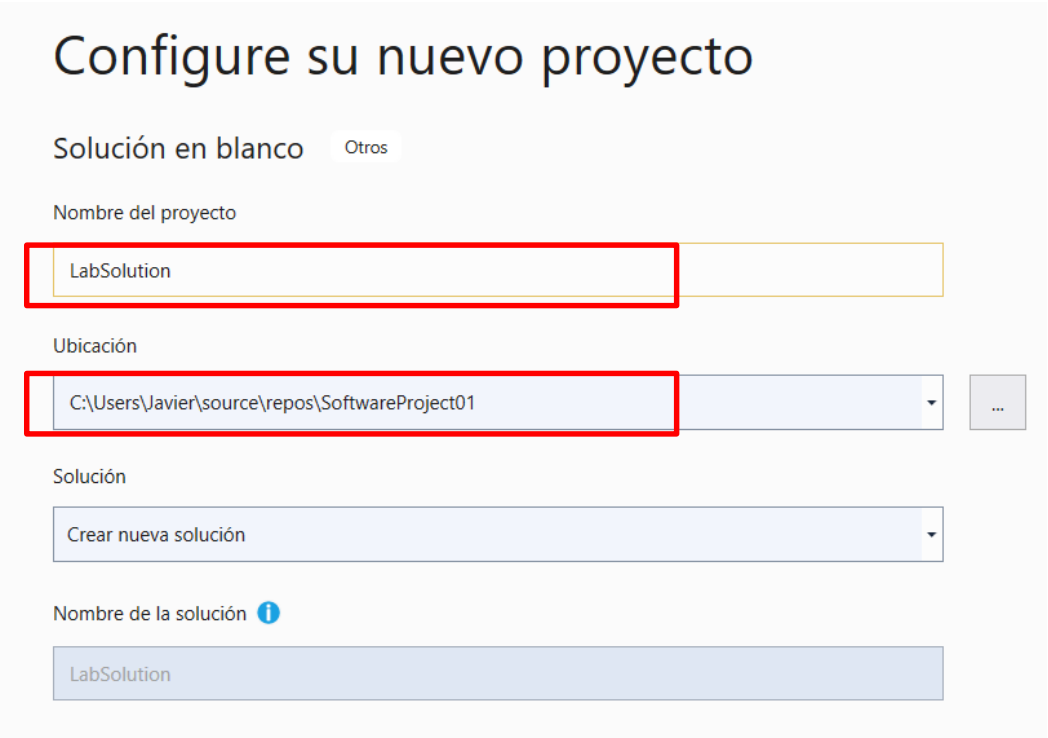
Create VS Project. Create Solution

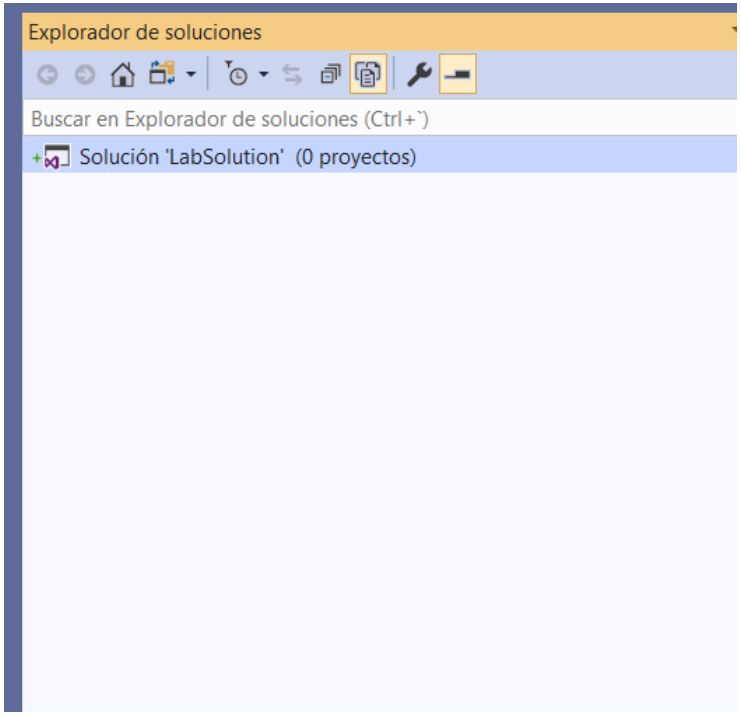
Create a blank (Empty) solution to which we will add different types of projects (Console Apps, Class Libraries, Windows Apps, etc.)



Create VS Project. Create Solution

Give a name to your solution(e.g. LabSolution)

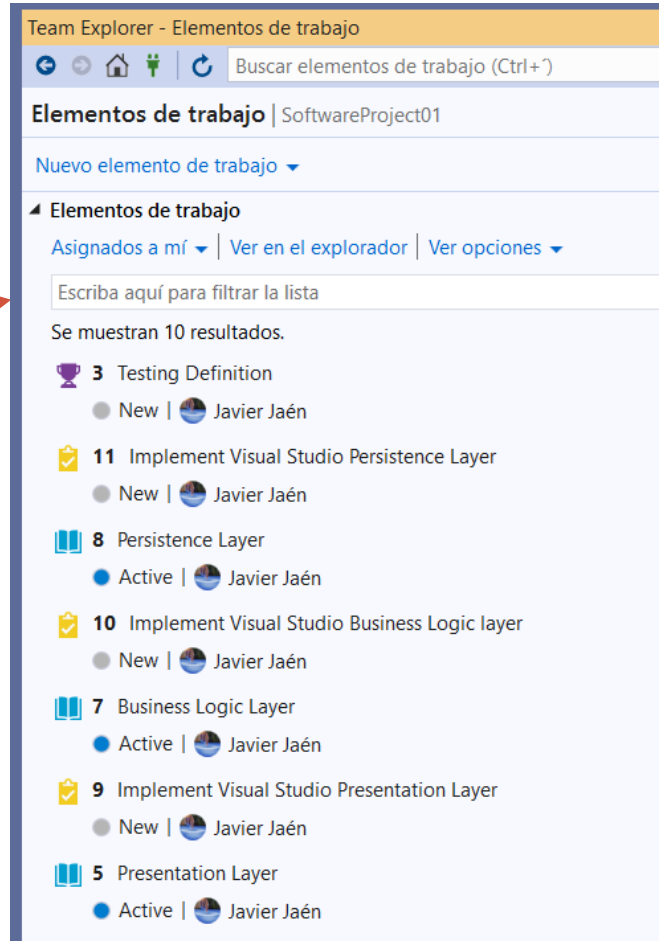




In Solution explorer we may see the empty solution just created

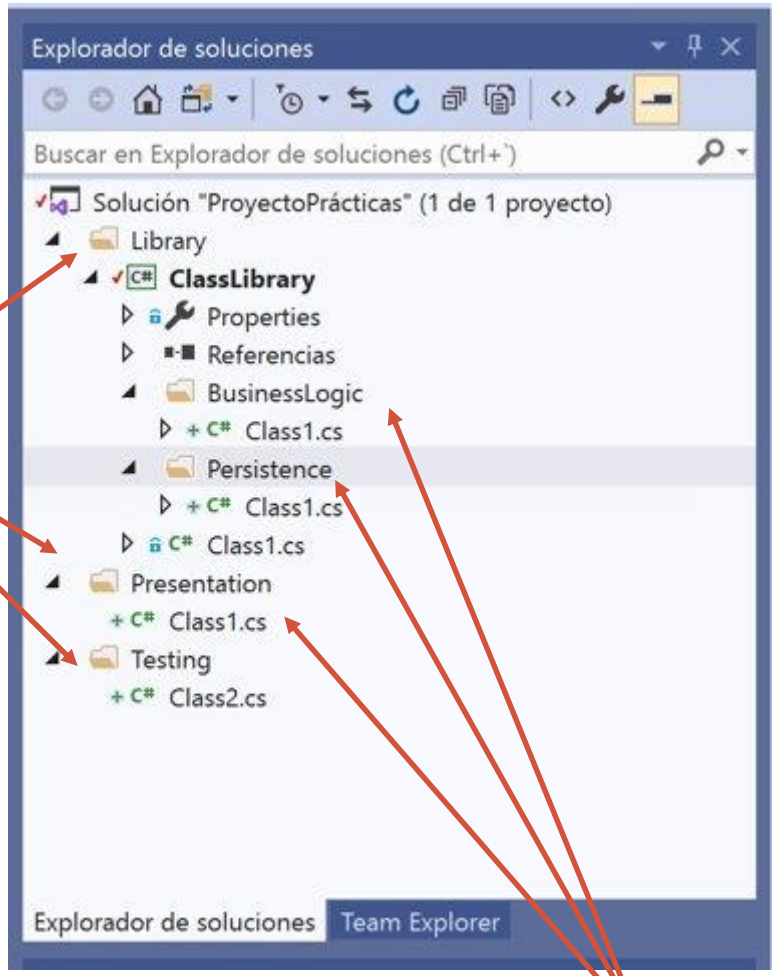
View > Solutions Explorer

Retreiving Work Items.



From Team Explorer all work ítems assigned to us can be displayed.

VStudio Solution Structure



Solution Folders

Architectural Layers

Create Project in Visual Studio

We will create the folder structure of our solution

We will separate the Presentation and the Business Logic+Persistence in two folders

The presentation folder will contain a project with the GUI

The code for the Business Logic and Persistence Layers will be contained in the same class library (dll).

We may add a new solutions folder in the VS menu:

Project > Add new solution folder

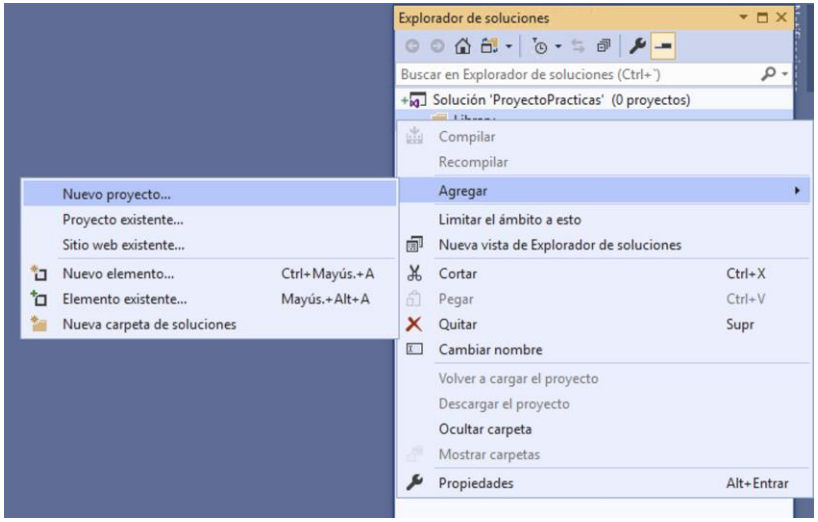
Inside a solutions folder additional folders may be added to organize the code.

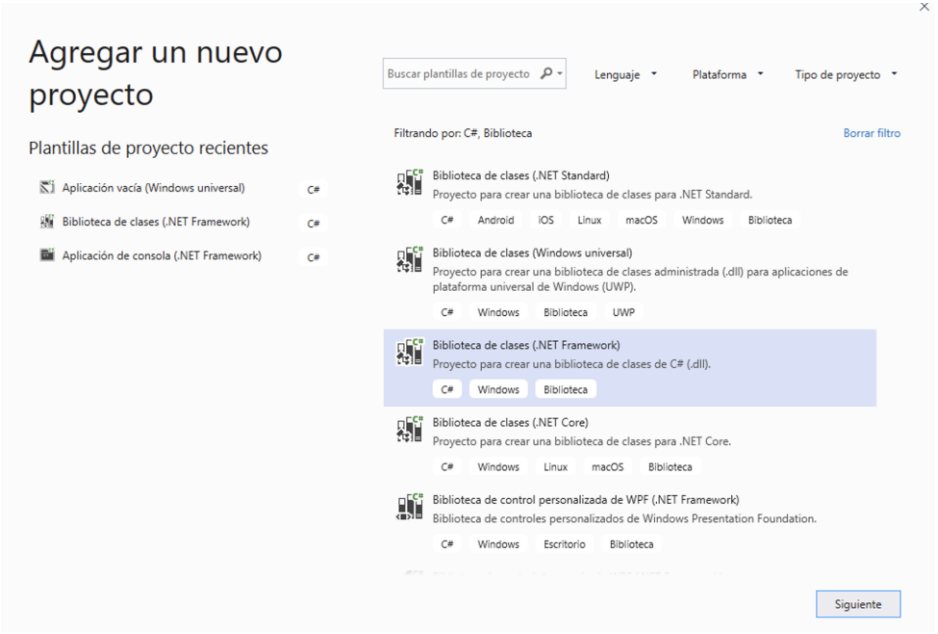
Create Project in Visual Studio

We will handle the work item “Implement Visual Studio Presentation Layer” by adding a Solutions Folder named “**Presentation**”

In the same way we will add another solutions folder called “**Library”.**

In the Solutions explorer we will add to “Library” a new Project of type *Biblioteca de clases (.NET Framework)* named “**ClassLibrary”**.

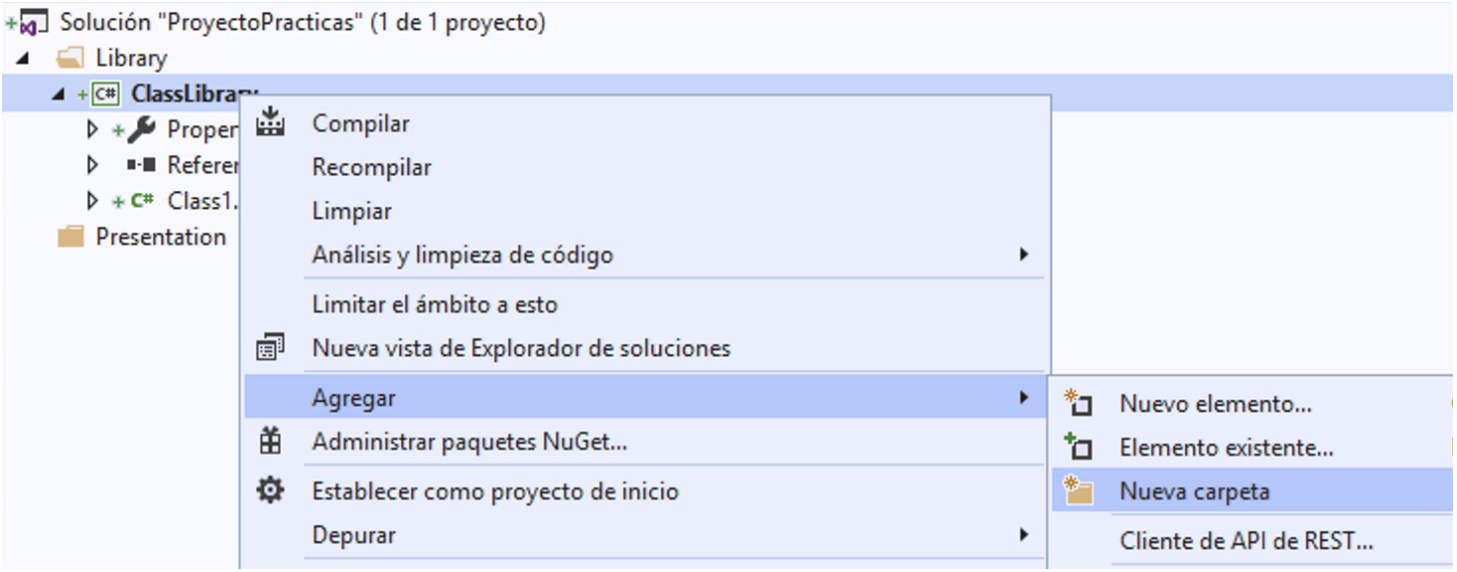




Create Project in Visual Studio

The Project **ClassLibrary** will contain two folders: “**BusinessLogic”** and “**Persistence”.**

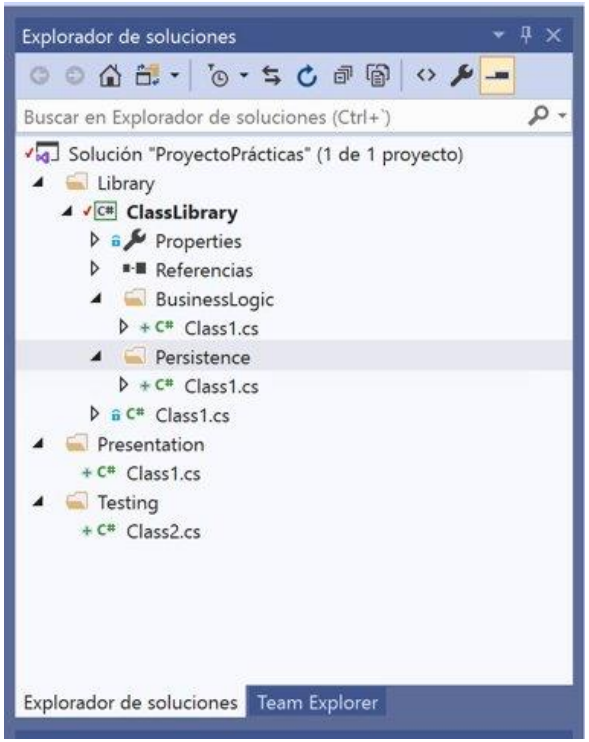
These folders are added in the Solutions Explorer: *Add > New Folder*



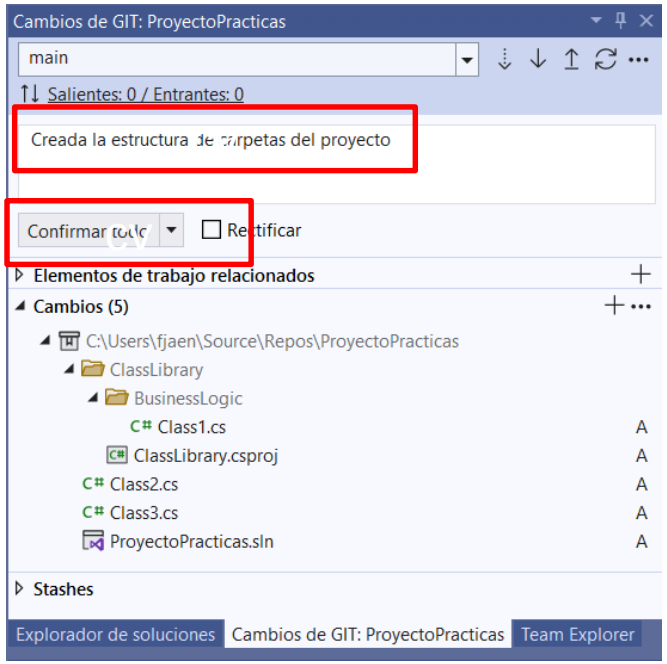
Create Project in Visual Studio

Finally a solutions folder called "**Testing**" has to be added to the solution LabSolution

The Final structure must be as follows:



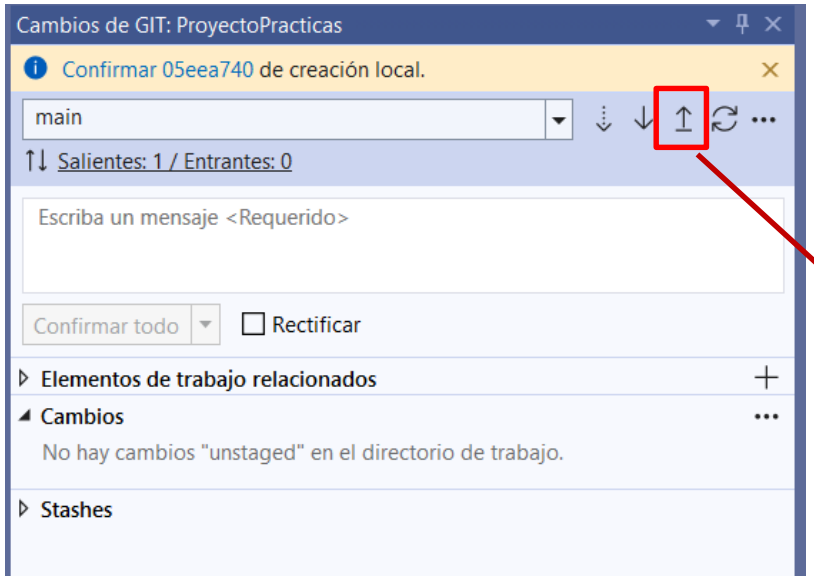
Store your work in the local repository (Commit)

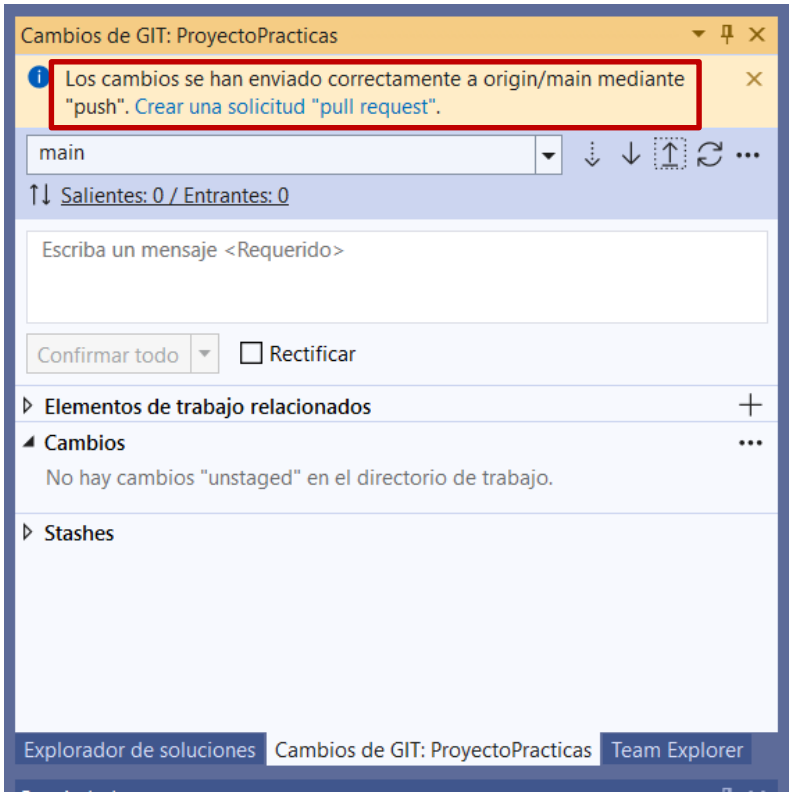


- Each time a significative change occurs:
  - Perform a ***commit*** in your **local repository**
  - Add a descriptive comment indicating the name of the task
- A commit **DOES NOT UPLOAD** your work to the remote repository. Your team mates will not see your changes to the code

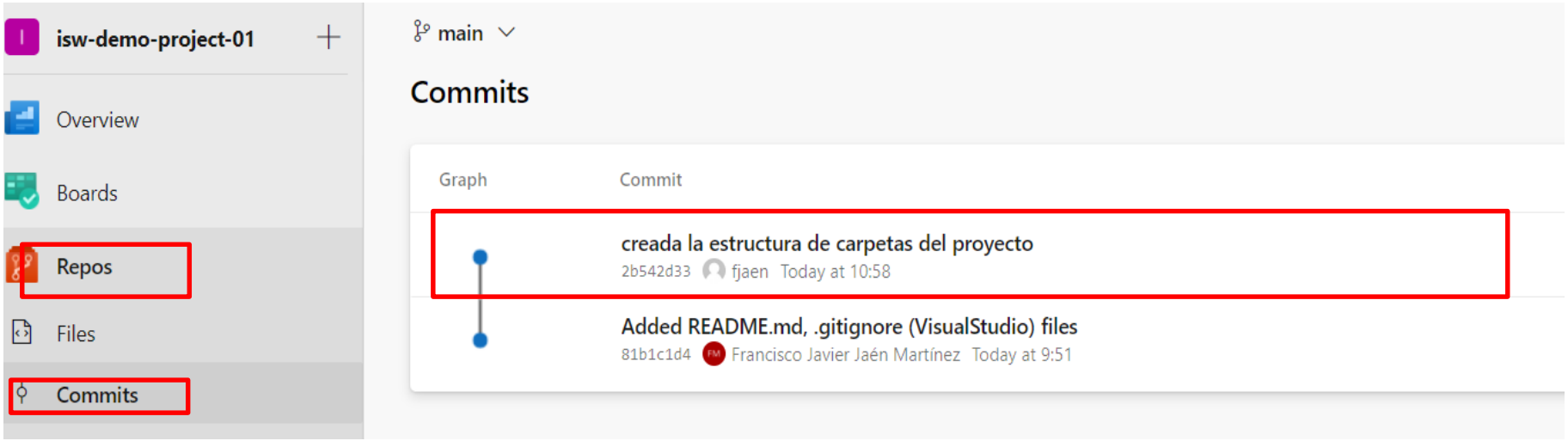
Synchronize: share your work

- Click insertar to perform a ***push*** operation on your work, the local repository will be updated in the remote repository and your work will be visible to the rest of the team

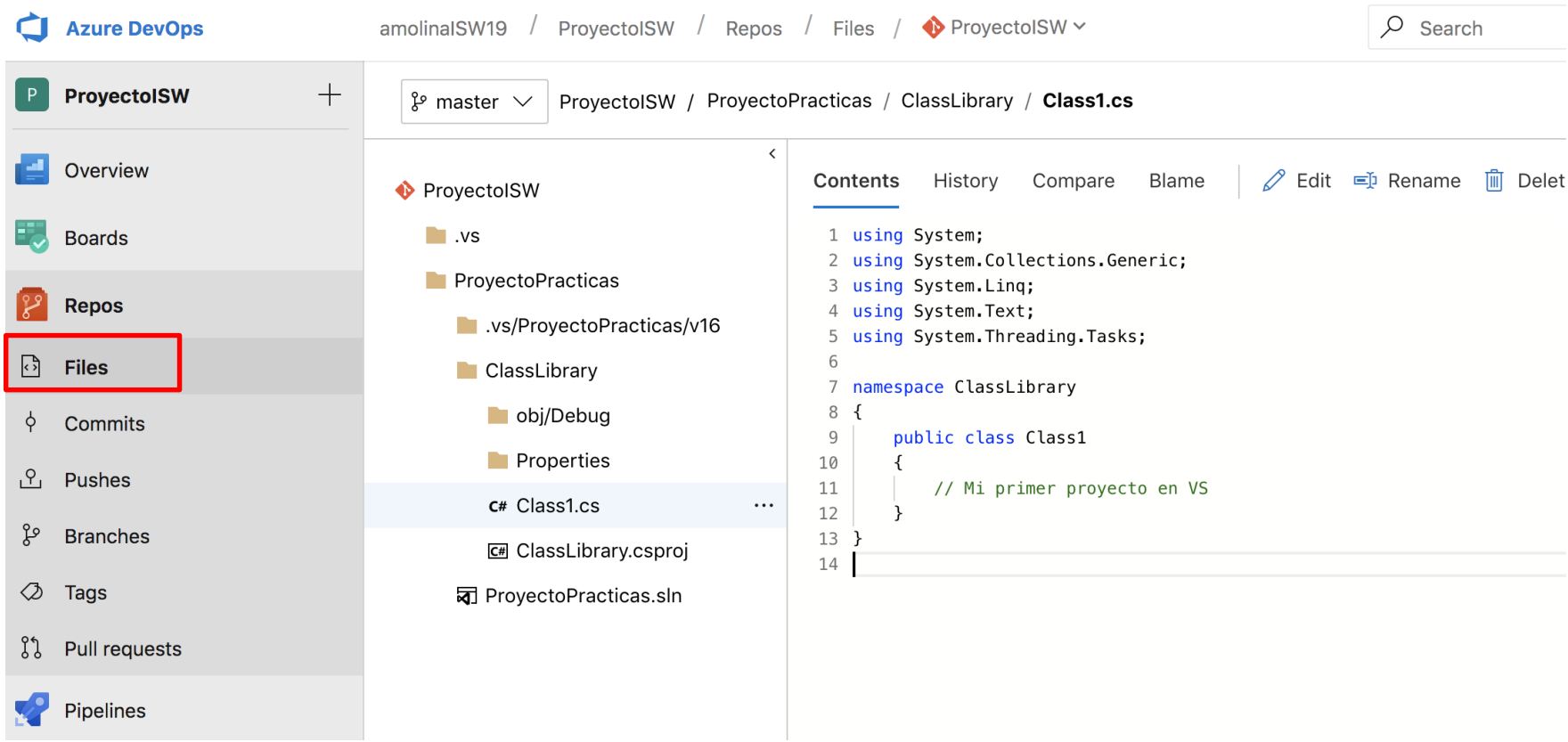




See changes in the repository with Azure DevOps

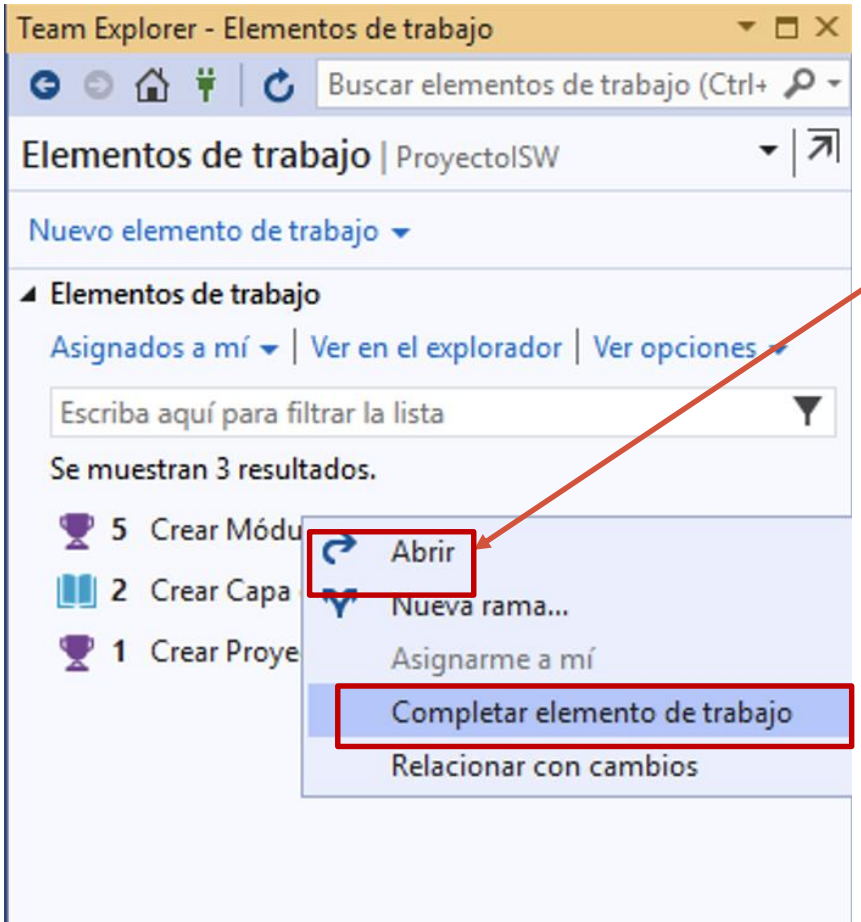


Inspect code with Azure DevOps



Manage your Project in Visual Studio

- In VS the status of the *work items* "stories"/ "tasks" can be controlled and updated as completed (closed) when the tests are successful.



A work item may be directly open in Azure DevOps from VS

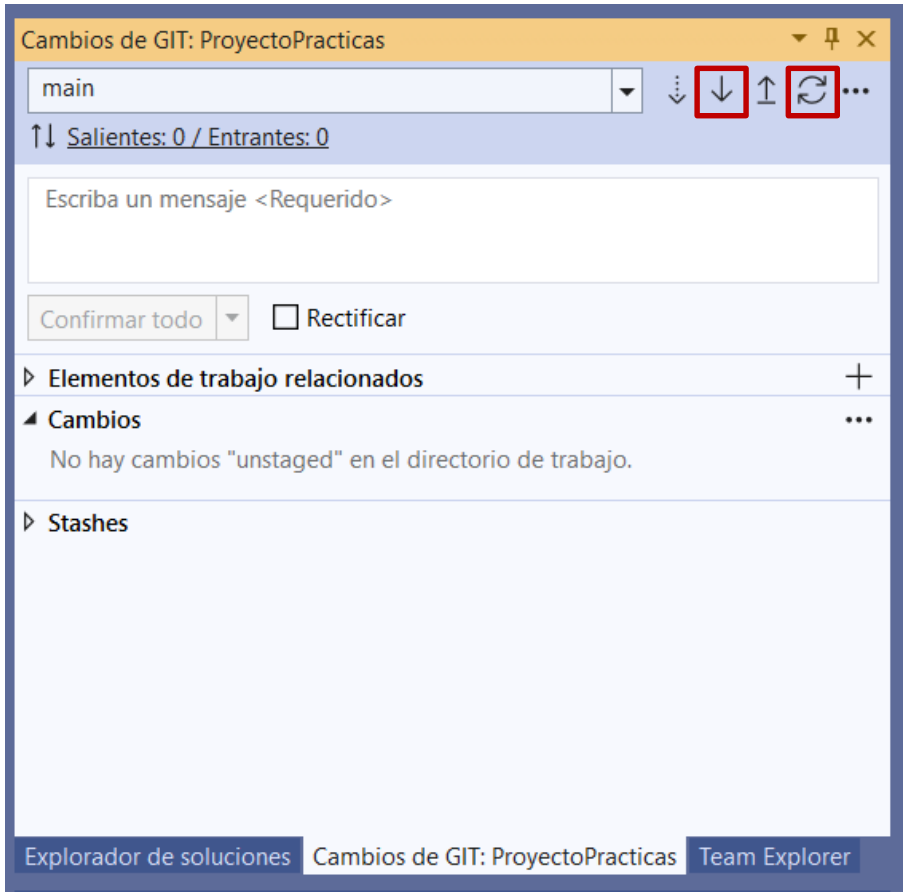
If an item is completed in VS the status Will be updated in the backlog and the board in Azure DevOps

Retrieve the Project from the remote repository into Visual Studio

- To obtain the latest version of the project
  - Clone the latest version of the project
  - Create a local repository in your lab computer

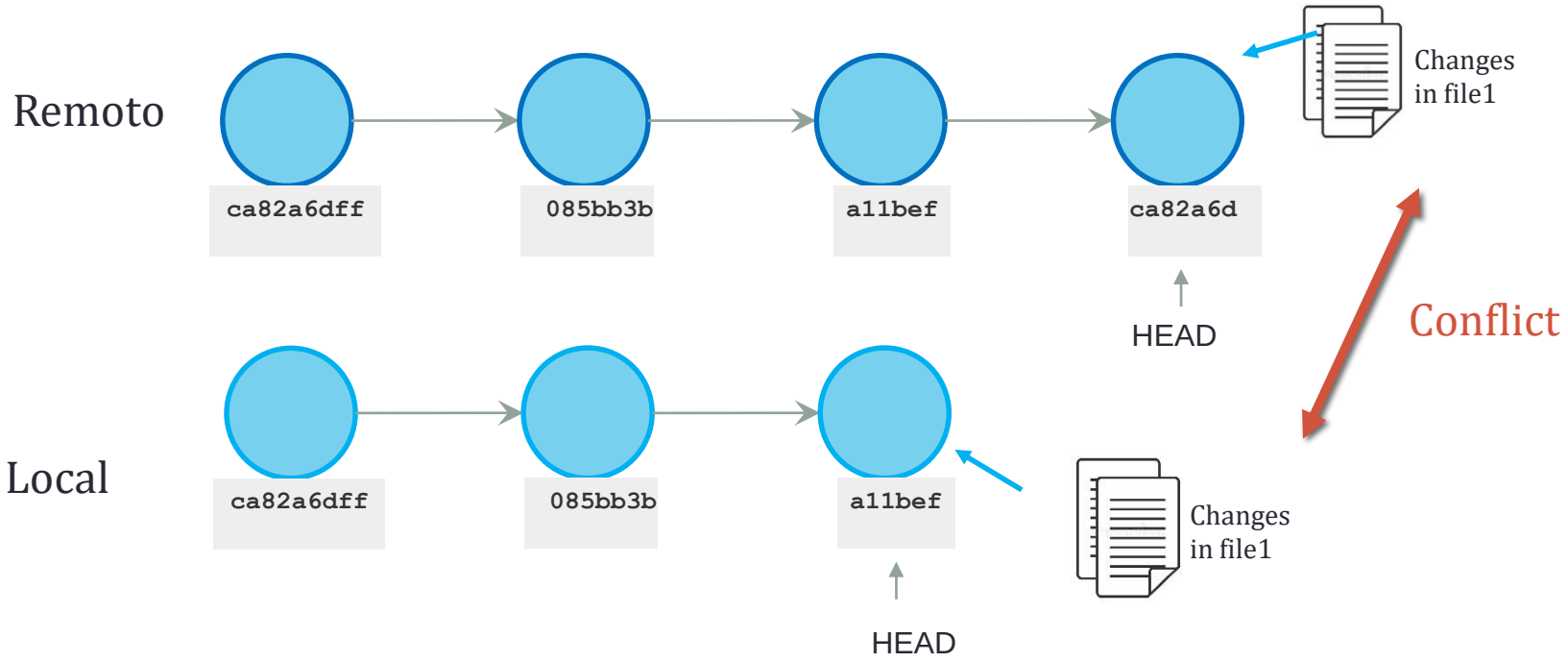
Obtain latest changes

- To incorporate the latest changes to your repository made by other users use the option **Extraer** (pull) or Synch (pull and push)

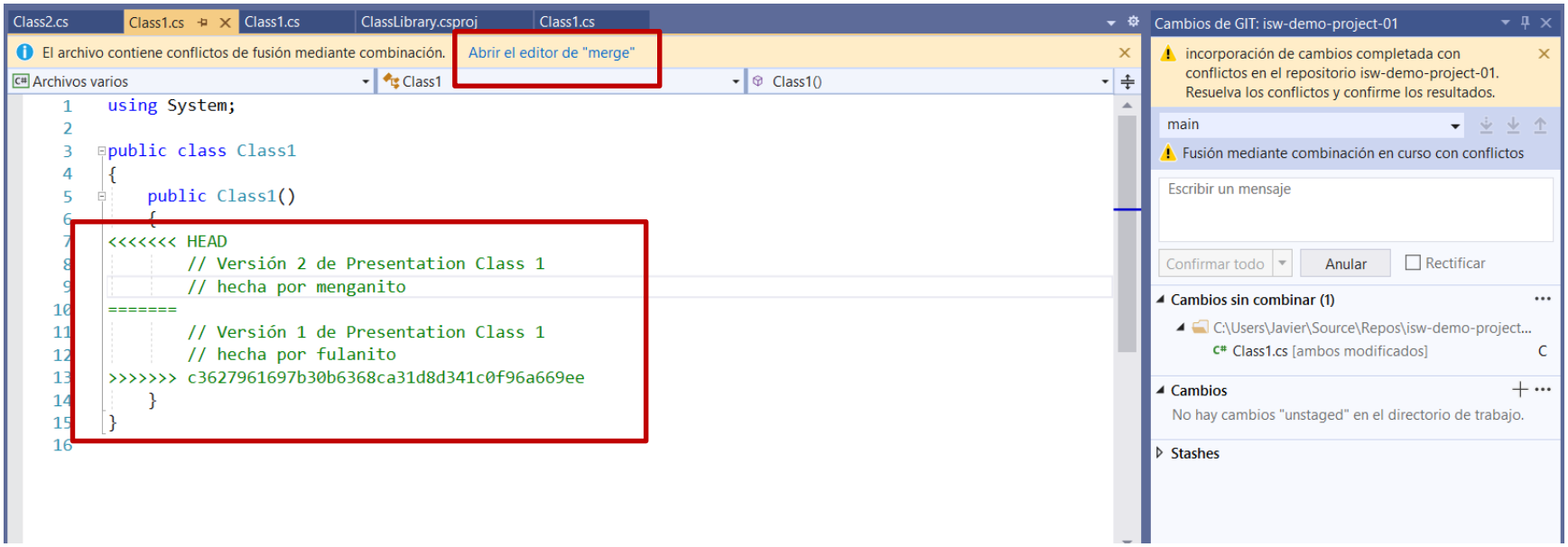


Manage code conflicts

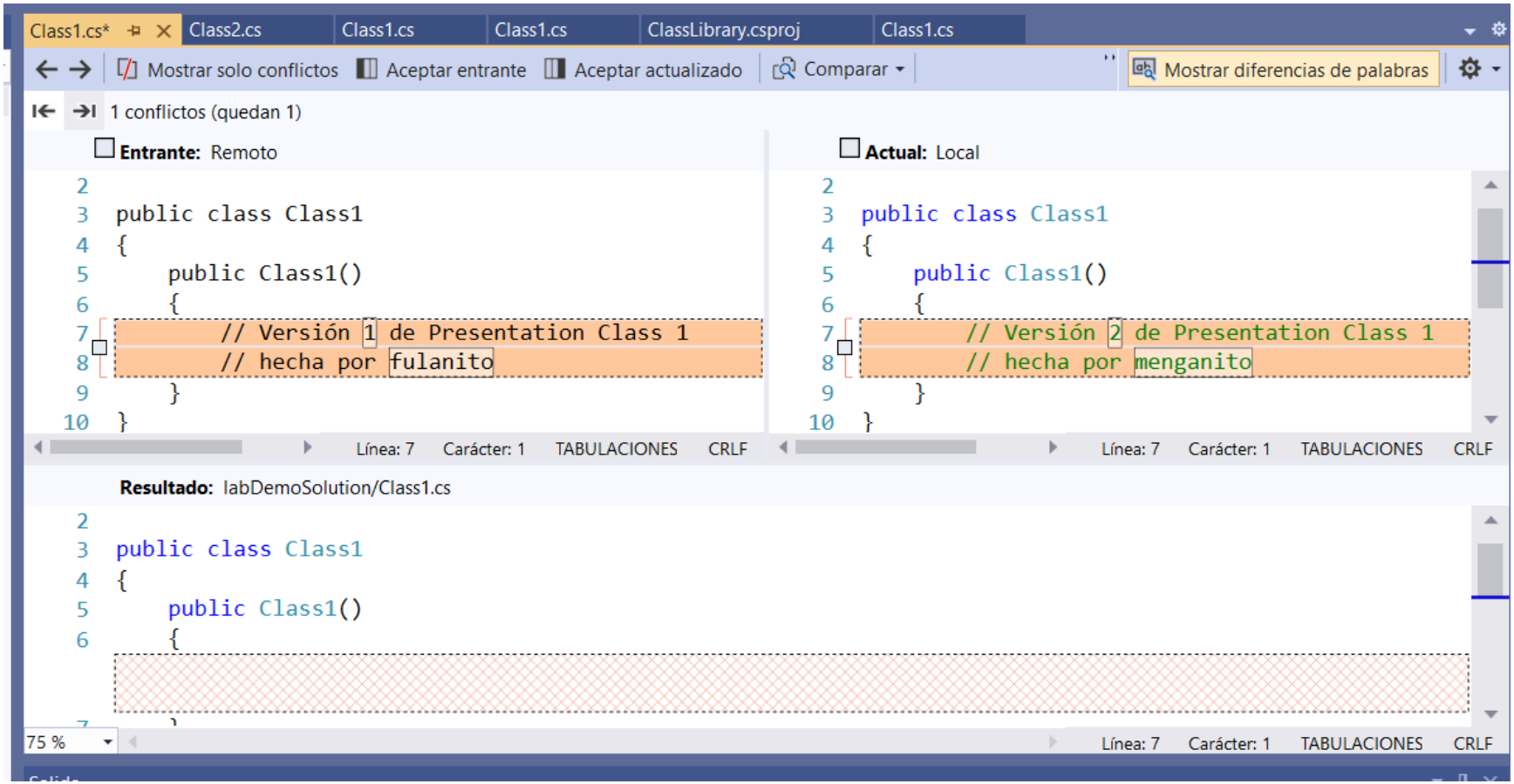
- When two developers work on the same file
  - A push by user 2 in the remote repository has updates in a file commited by user 1 locally



Manage code conflicts



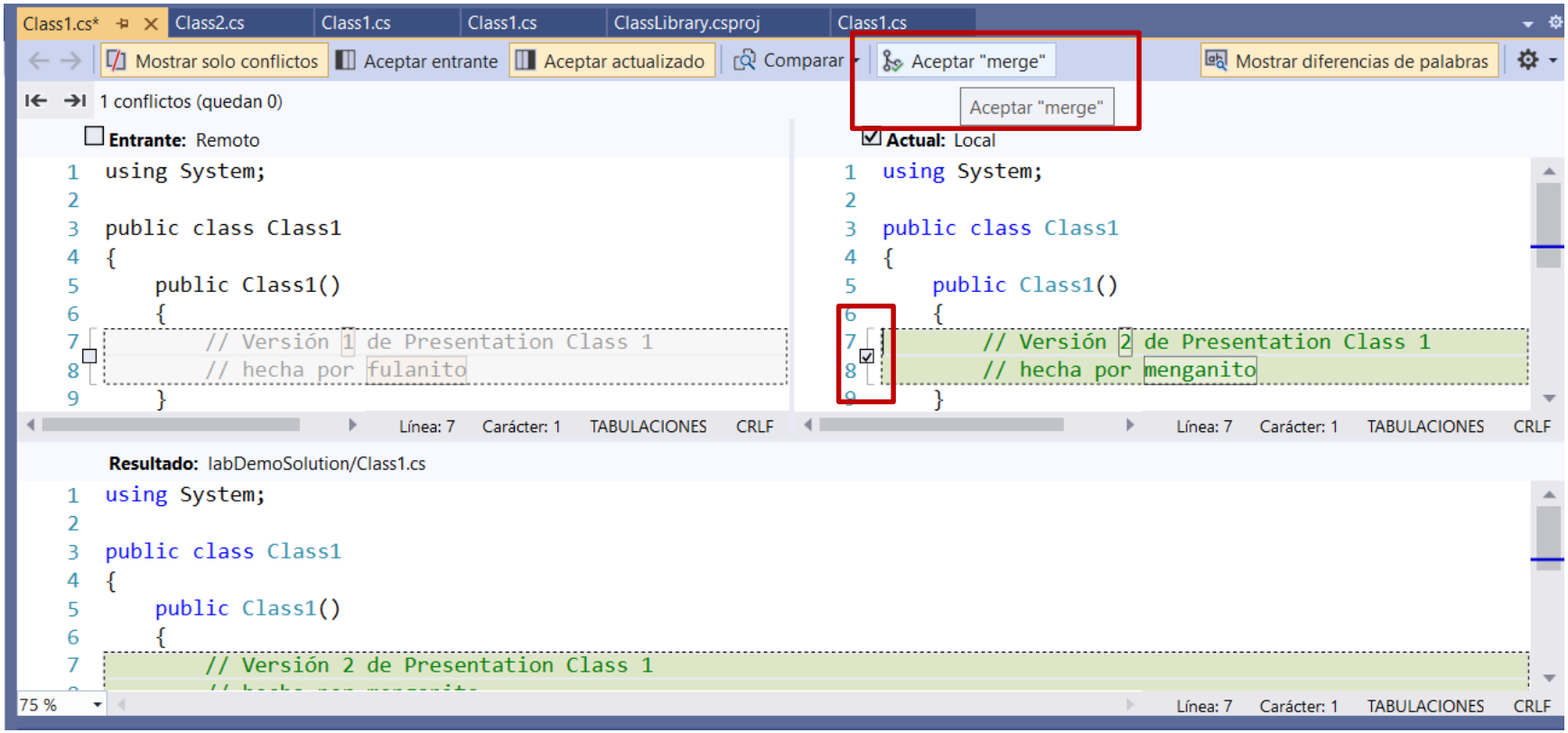
Manage code conflicts



A new commit is created with the right code

Manage code conflicts

Select the correct version or combine both versions indicating the correct code



A new commit is created with the right code

Single Branch Development

- Start by cloning (if there is no local repository) or synchronizing remote and local repository
- Do your work locally
- Commit your work locally
- Pull any commits other teammates may have pushed to the server
- Resolve conflicts
- Push your local repository to the remote server

✓ Conclusions

- Visual Studio complements the work plan designed with Azure DevOps
- It allows us to associate code and changes to the tasks defined in the work plan (correspondence between planned work and implemented code)
- It allows retrieving and protecting code and managing conflicts – free transparent version control in the cloud

Laboratory Virtualization

- Download & Install the UDS Client for your platform
  - https://polilabsvpn.upv.es/uds/page/client-download
- Open a remote desktop connection
  - Server: https://polilabsvpn.upv.es/uds/page/services
    - DSIC Windows image
  - User: Alumno UPVNET (Assigned by UPV when you enrolled)
  - Password: your UPVNET password
- Visual Studio 2022 Enterprise is available in the laboratory virtualization

Learning Resources

- Visual Studio Walkthroughs (English) https://msdn.microsoft.com/es-es/library/szatc41e(v=vs.110).aspx

- Introduction to Azure DevOps. *Donovan Brown*. Microsoft Visual Studio

- Plan Your work with Azure Boards. *Ali Tai.* Microsoft Visual Studio

- Manage and store your code in Azure Repos. Edward Thomson. Microsoft Visual Studio