# Seminar

**S2**

Chapter 3.

## Software Architecture

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

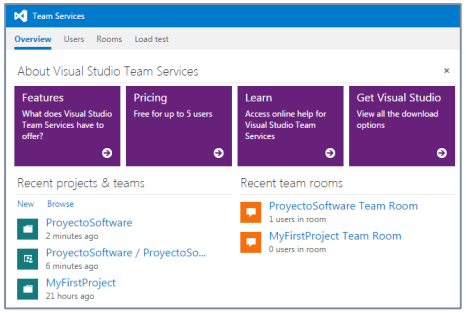
**Software Engineering**

Computer Science School
DSIC – UPV

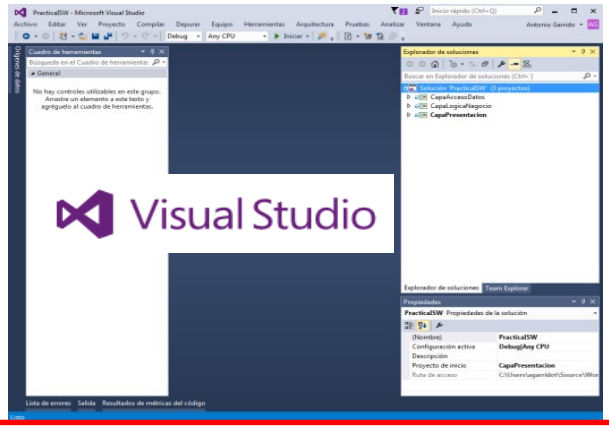
# Goals

- 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)

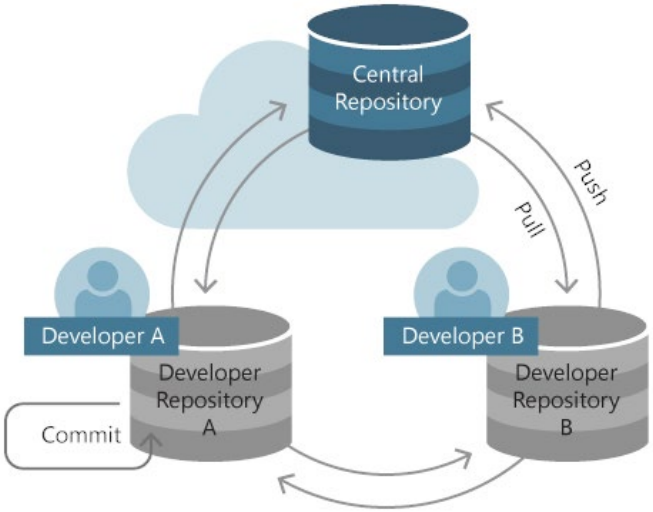



**Part 2.** Project Development with Visual Studio



# Version Control

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

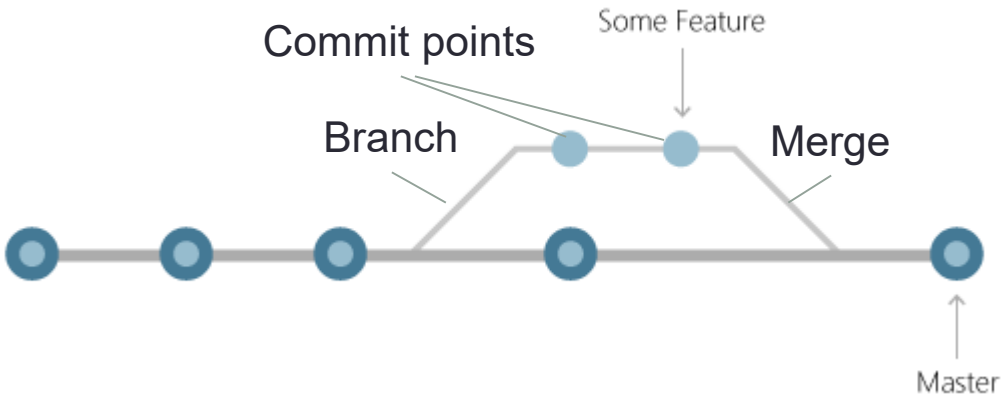



Git and TFS are available for Version Control

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

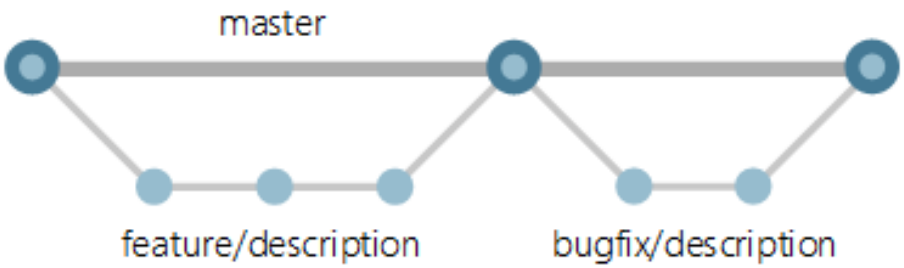
# Git WorkFlow

- 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



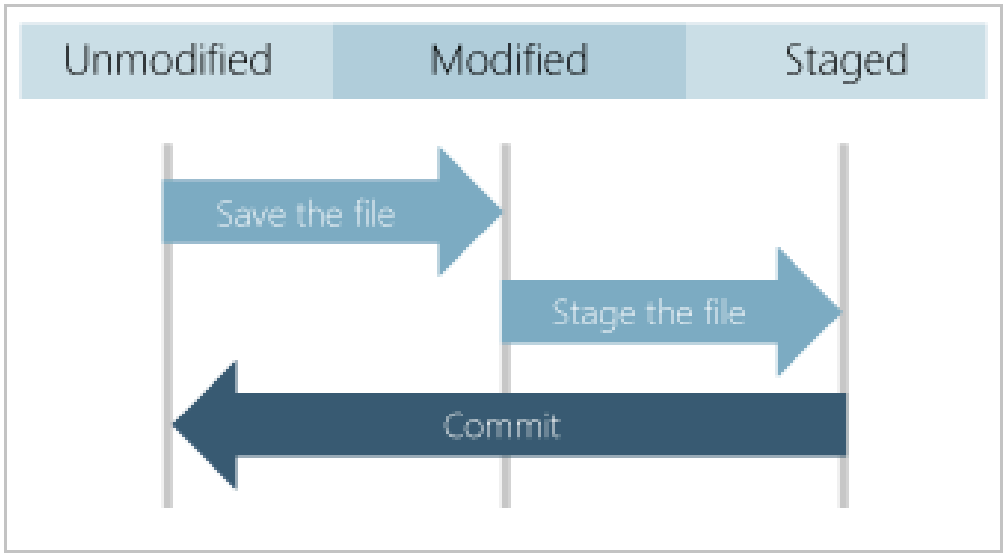
# Git Branching Strategy

- 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

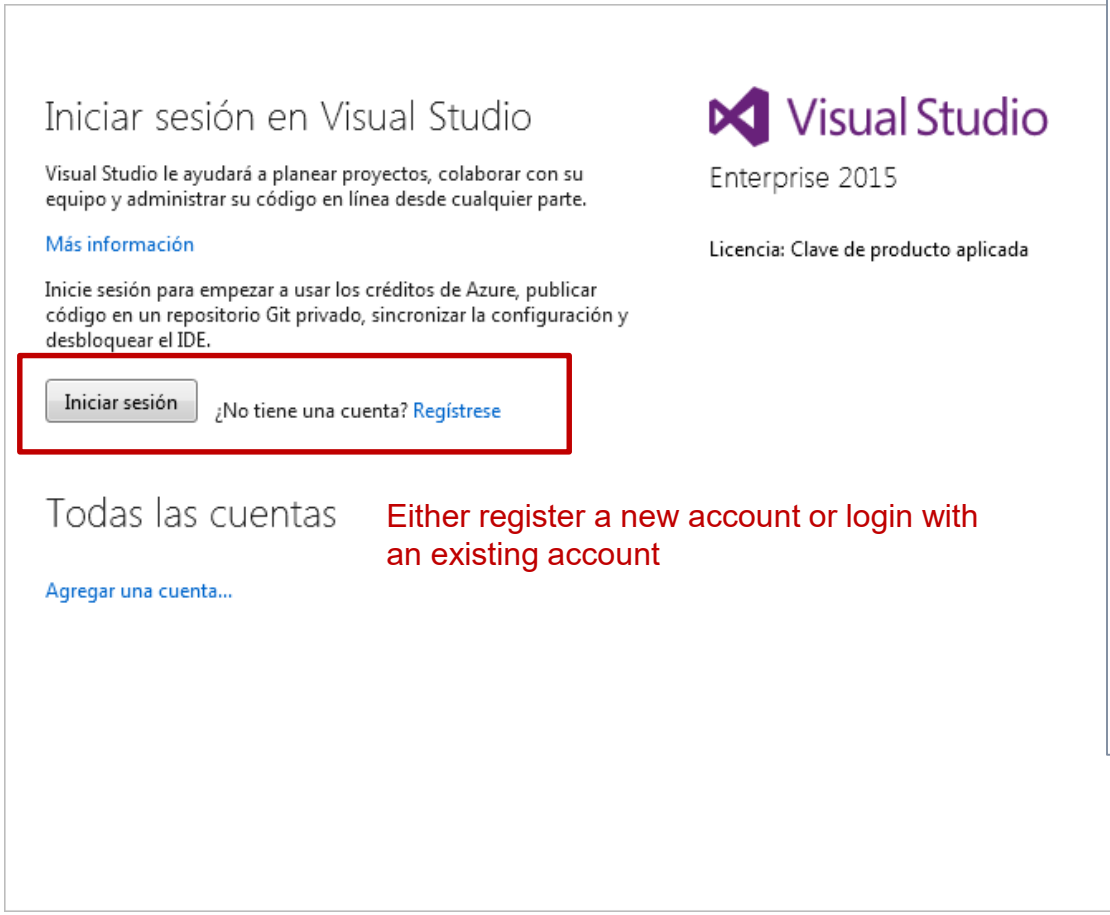
- Create a software project using *Microsoft Visual Studio*, retrieving (and completing) the Project plan elaborated with *Azure Boards*

Steps:

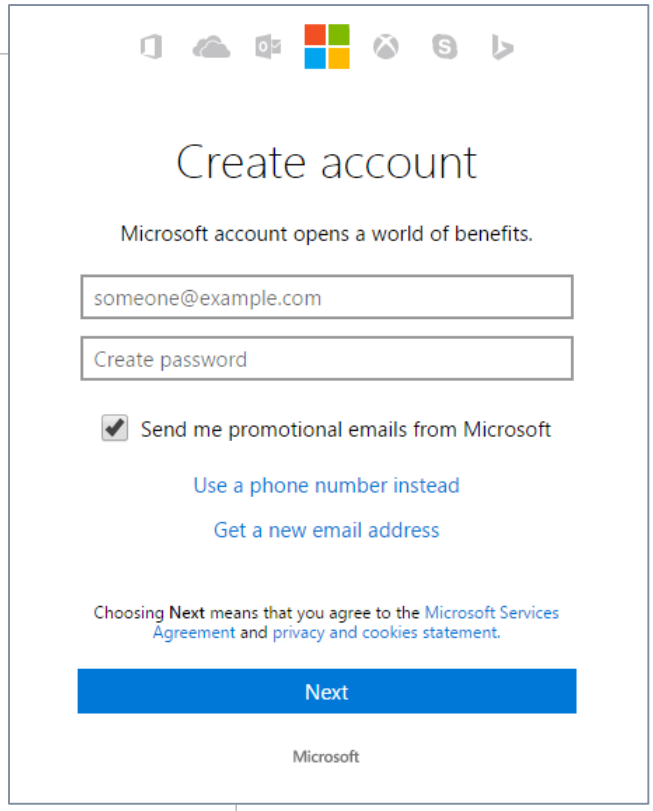
- ✓Create a Microsoft account (if not already done)
- ✓Create a Visual studio Solution project (First time)
- ✓Project Management with *Visual Studio*
- ✓Retrieve the Project from the repository into Visual Studio
- ✓Managing code conflicts

# ✓ LogIn

## File > Account settings...




Either register a new account or login with an existing account




If an account was not previously created follow the steps to create one

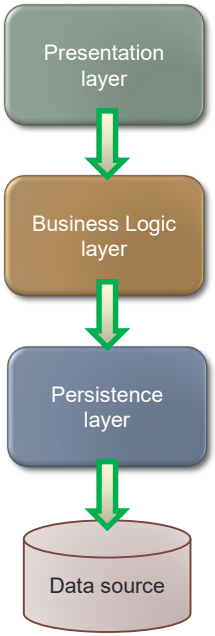
# ✓ Define Agile Iteration

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

SoftwareProject01 Team

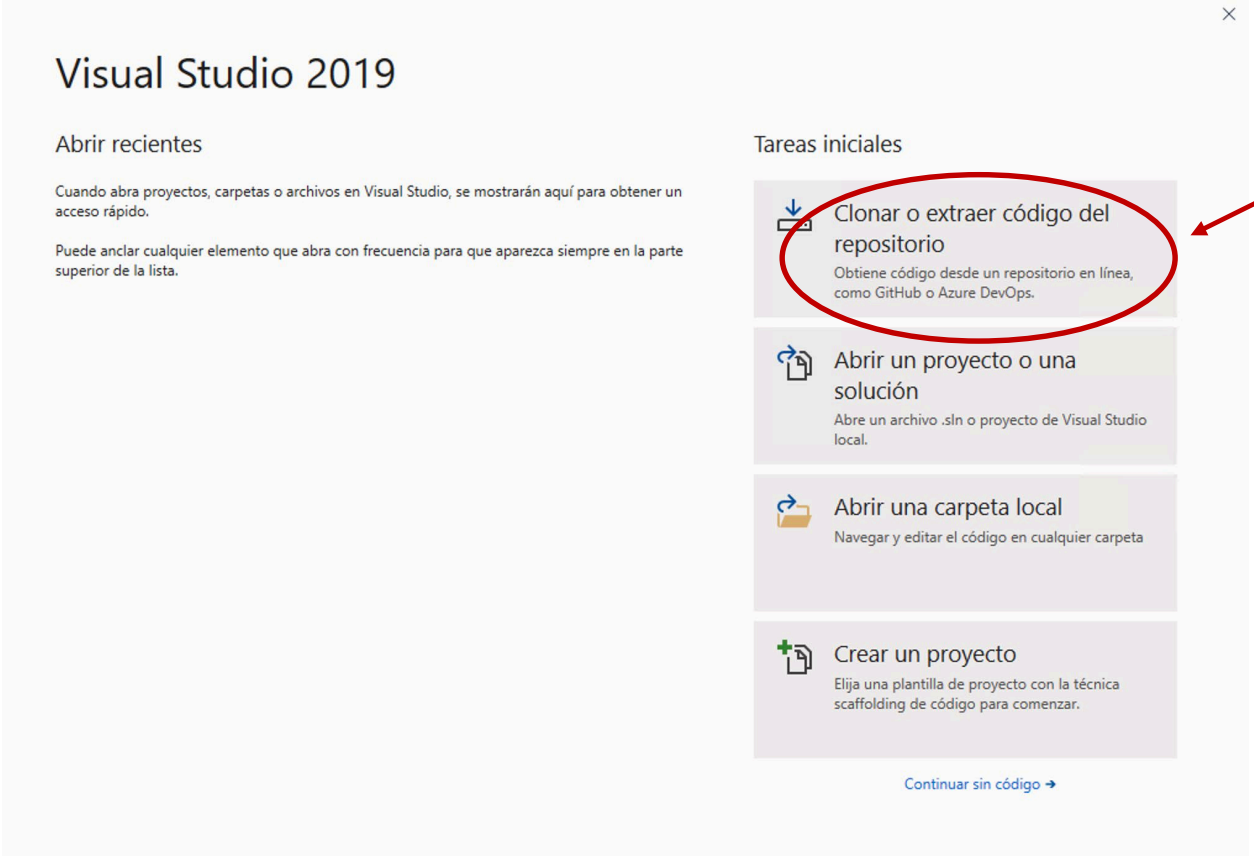
Backlog | Analytics | + New Work Item | View as Board | Column Options | ...

| 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 commontasks, including links to most recent projects

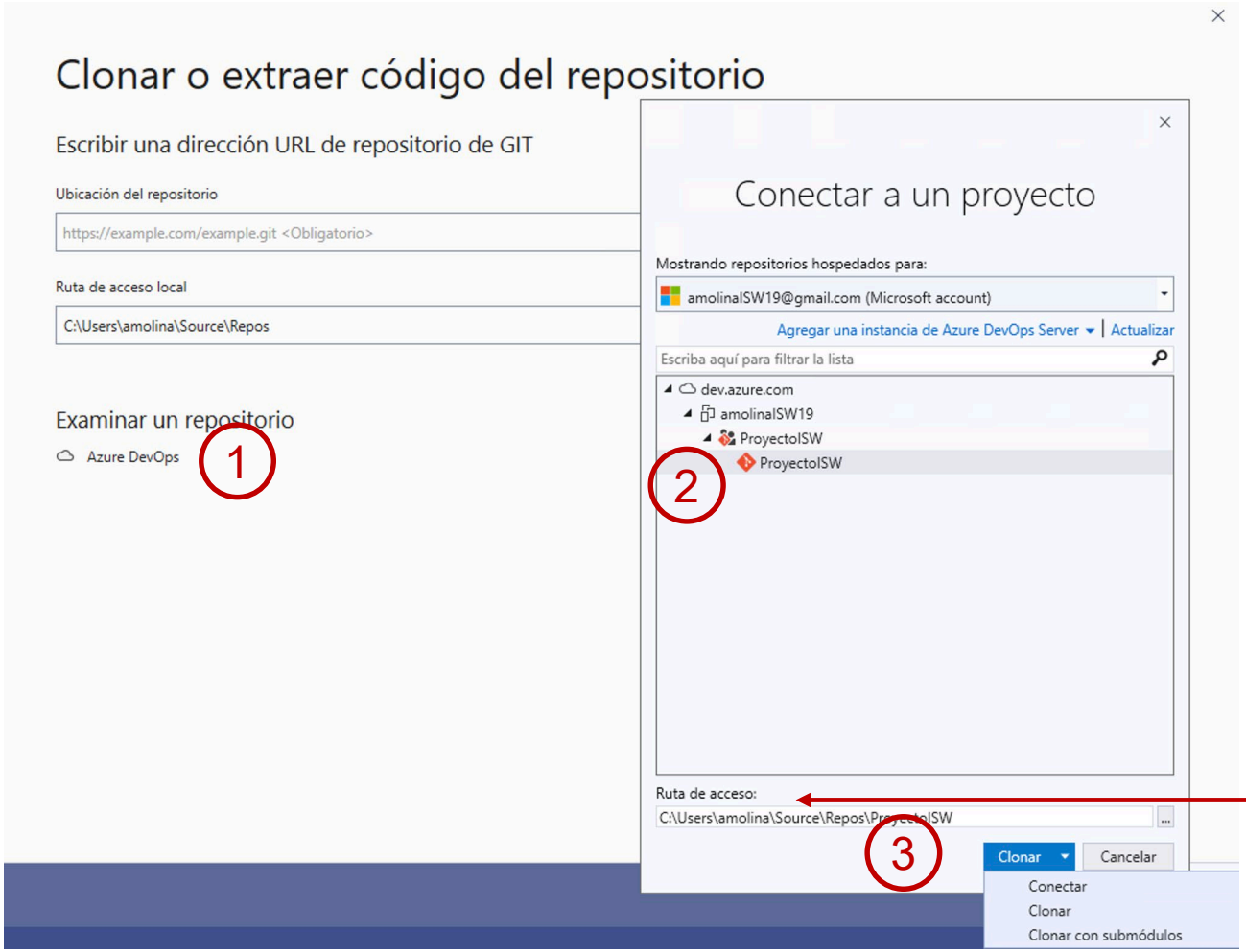



Select cloning repository to connect to our Azure DevOps Project and clone the code

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 eh project(ProyectoISW).

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

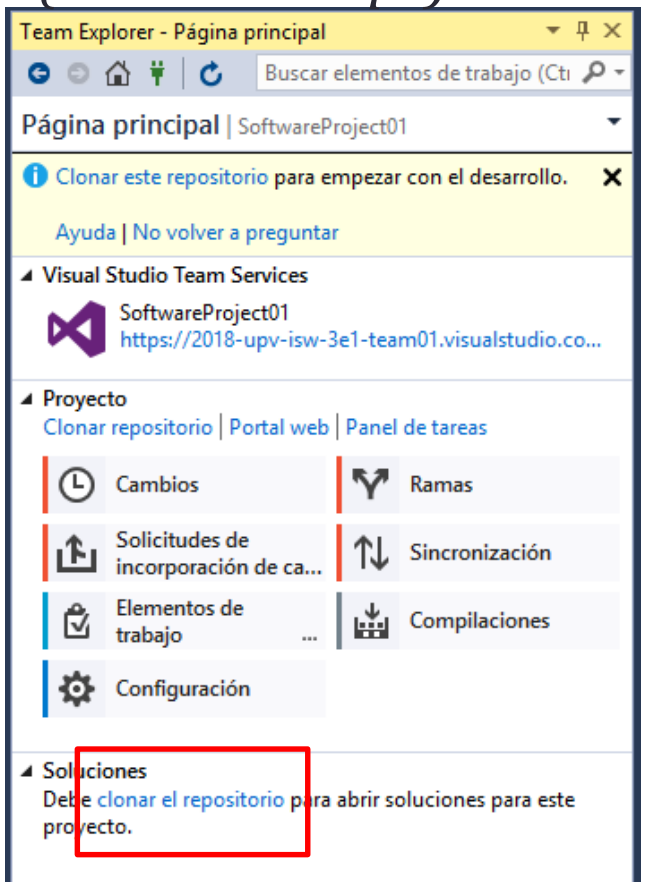
This must only be done by the MANAGER of the project.

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

# ✓ Cloning Repository (Alternative way)

View > Team Explorer

Displays the tools to work with *Visual Studio Team Services (Azure DevOps)* from *Visual Studio*

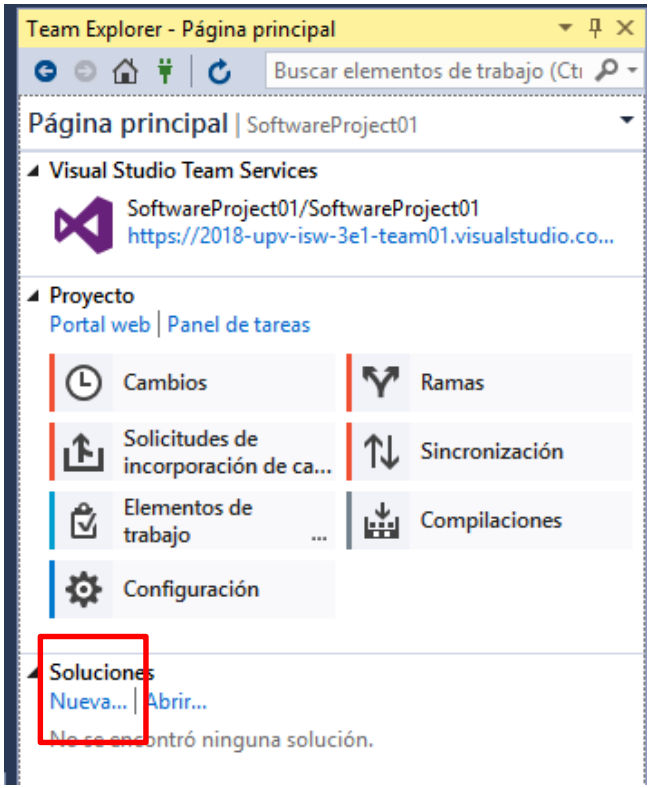



First Time, the Project leader clones the repository (Git Version Control).

# ✓ Configuring workspace

View > Team Explorer

Displays the tools to work with *Visual Studio Team Services (Azure DevOps)* from *Visual Studio*

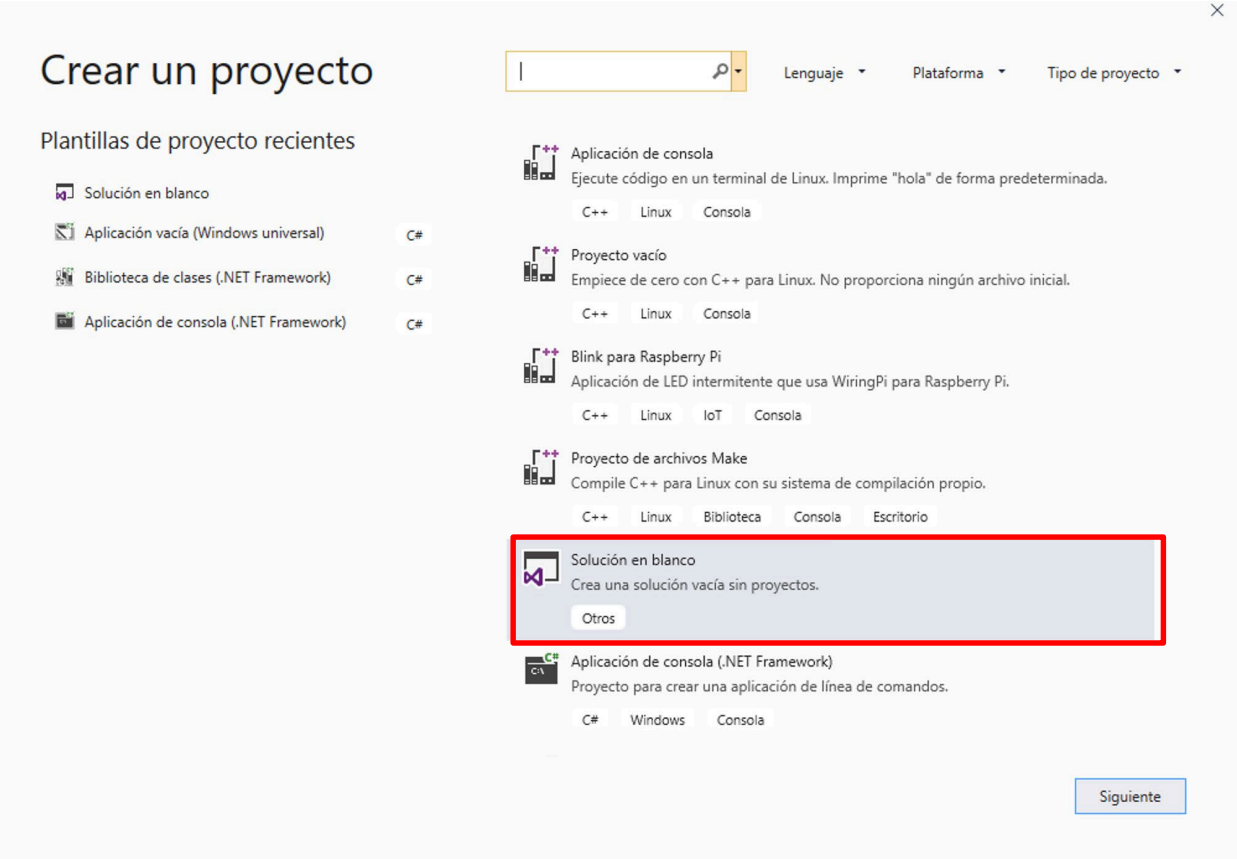



First Time, **the Project leader** creates a new solution.

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

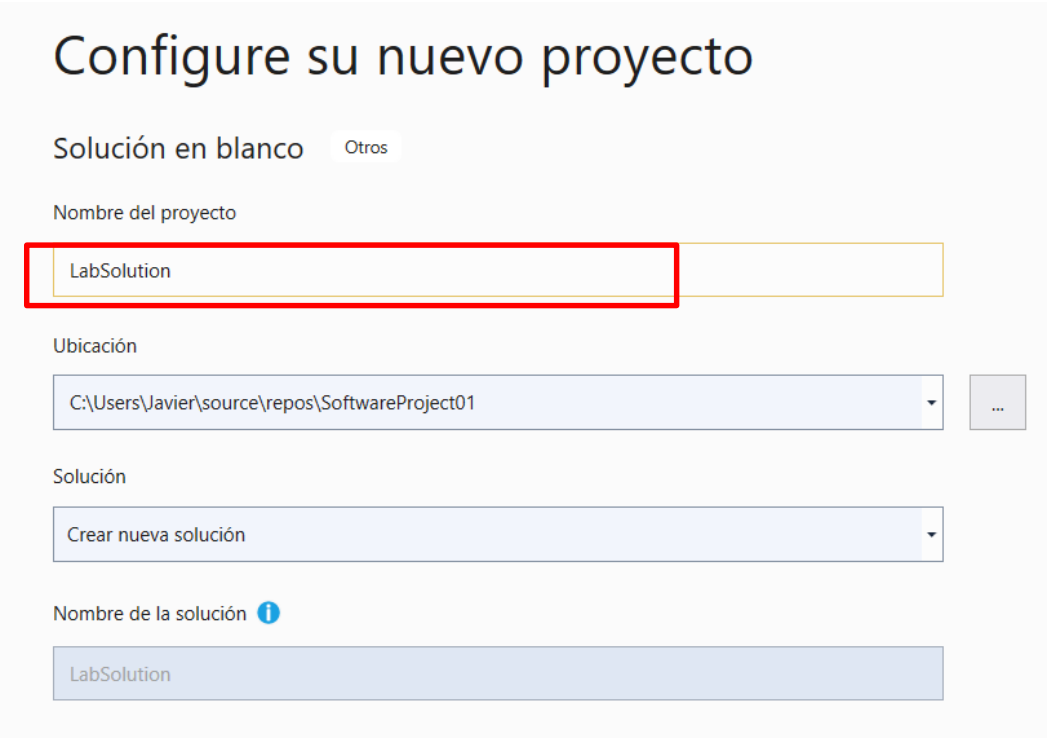
# Create VS Project. Create Solution

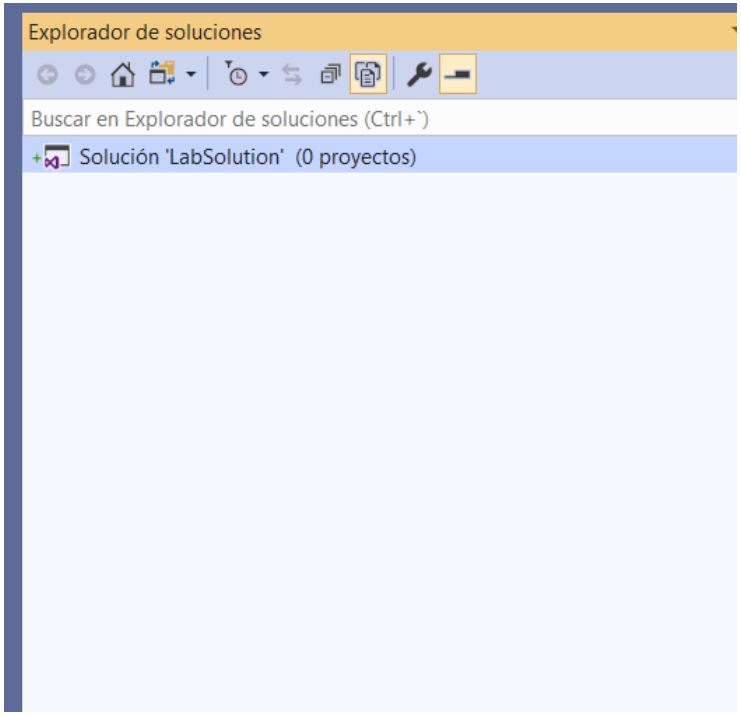
Create a blank (Empty) solution to which we will add different types of projects (Console Apps, Class Libraries, Windows Apps, etc.)



# Create VS Project. Create Solution

Dgive a name to your solution(LabSolution)

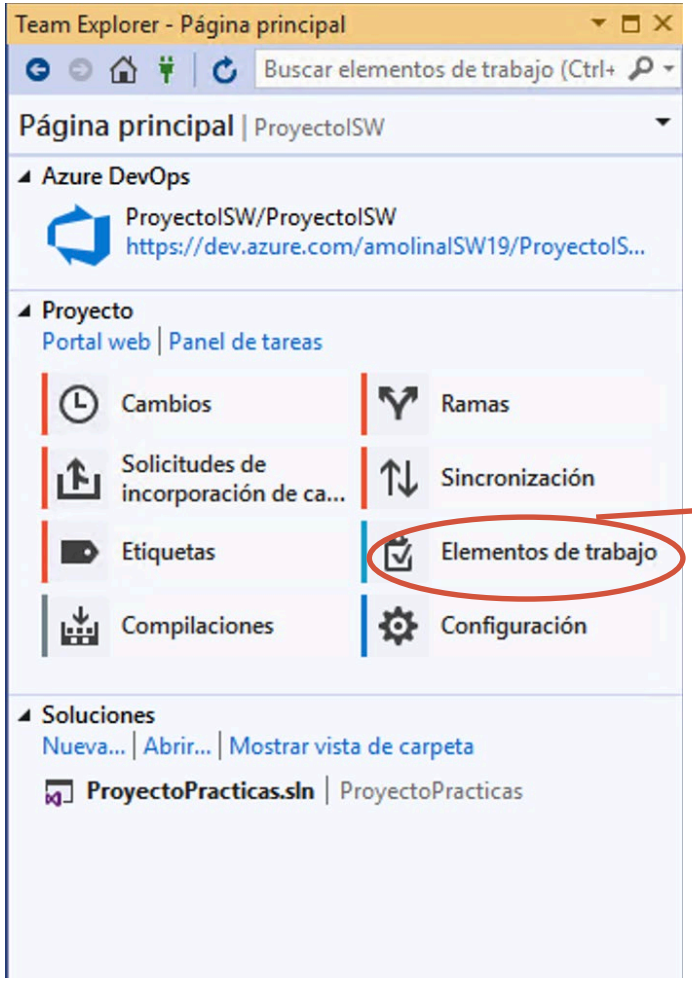


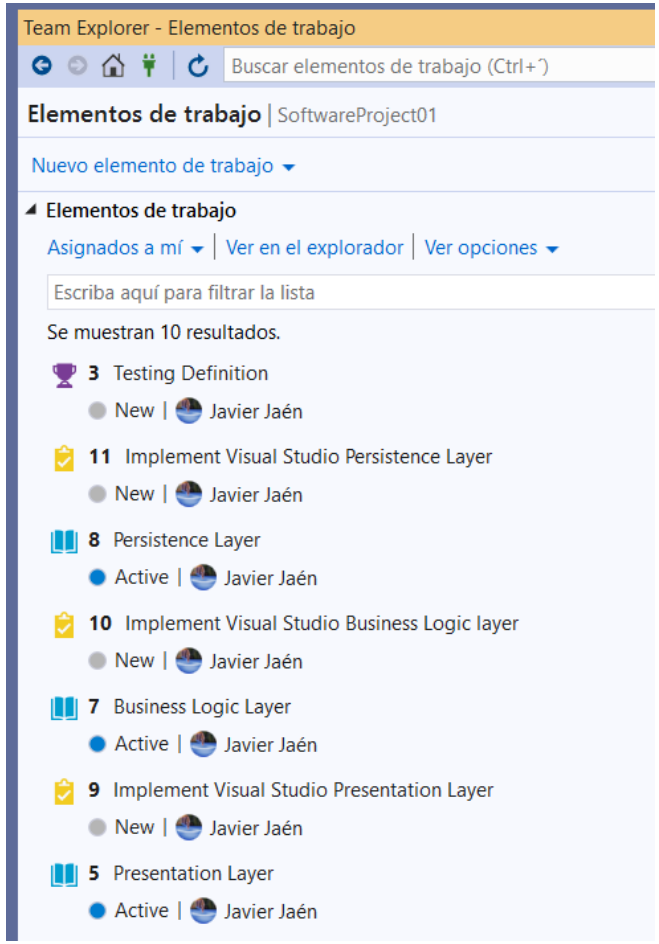




In Solution explorer we may see the empty solution just created

View > Solutions Explorer

# Retreiving Work Items.




FromTeam Explorer all work ítems assigned to us can be displayed.

# 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 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:

Proyecto > Agregar nueva carpeta de soluciones

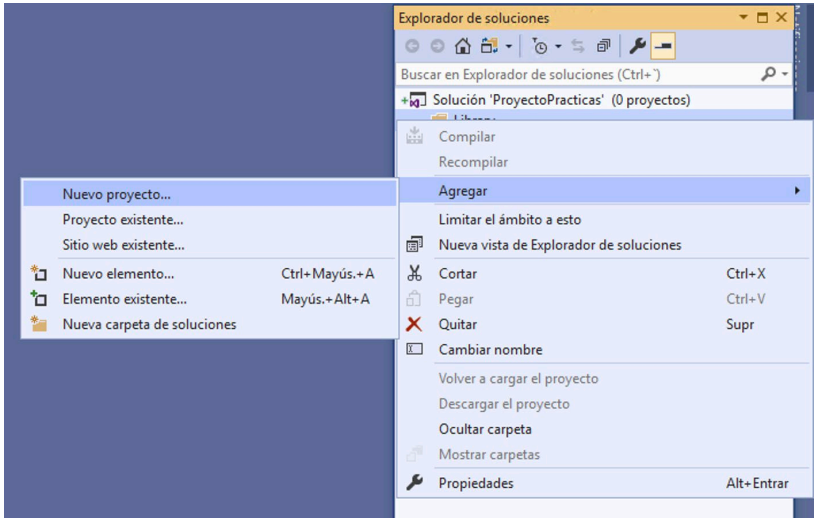
Inside a solutions folder additional folders may be added to organize the code.

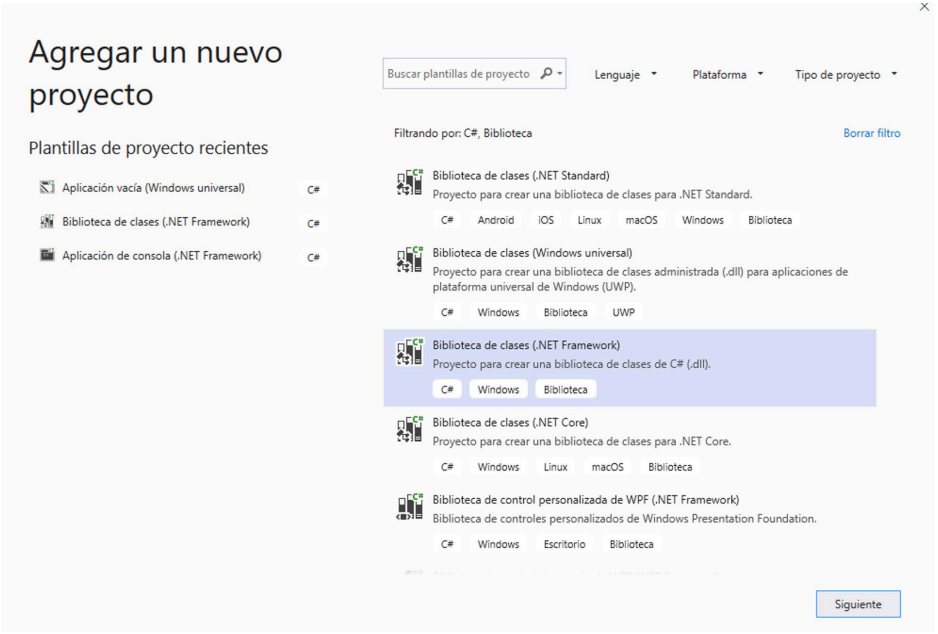
# Create Project in Visual Studio

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”**.

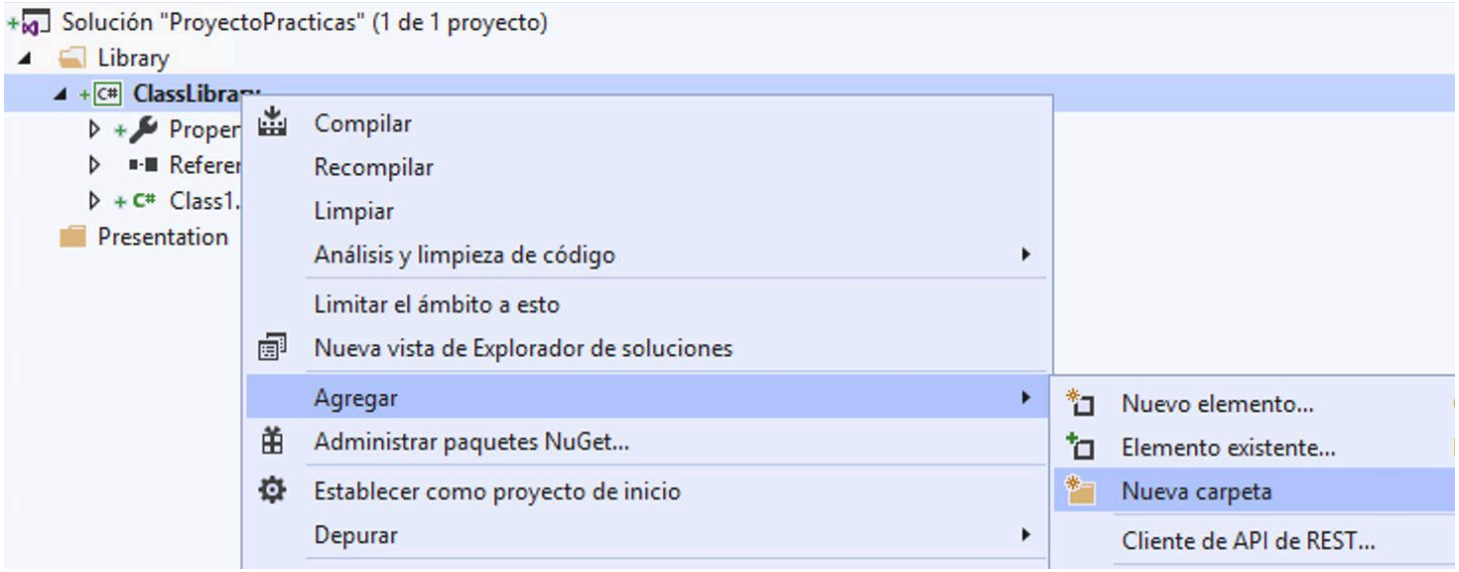


# Create Project in Visual Studio

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

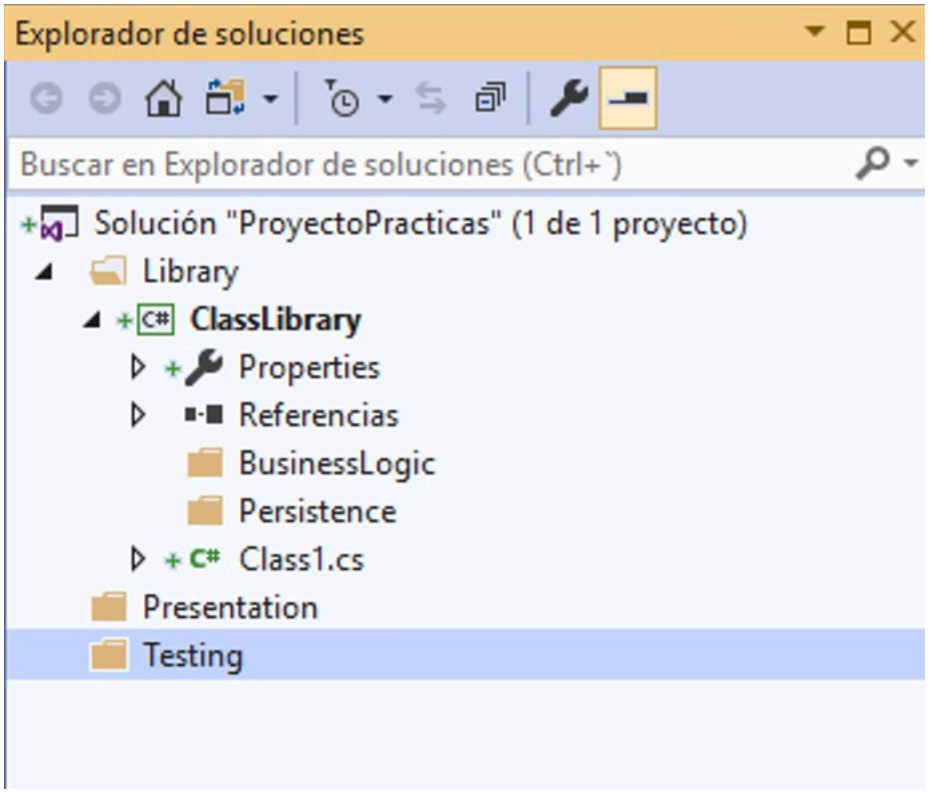
These folders are added in the Solutions Explorer: *Agregar > Nueva carpeta*



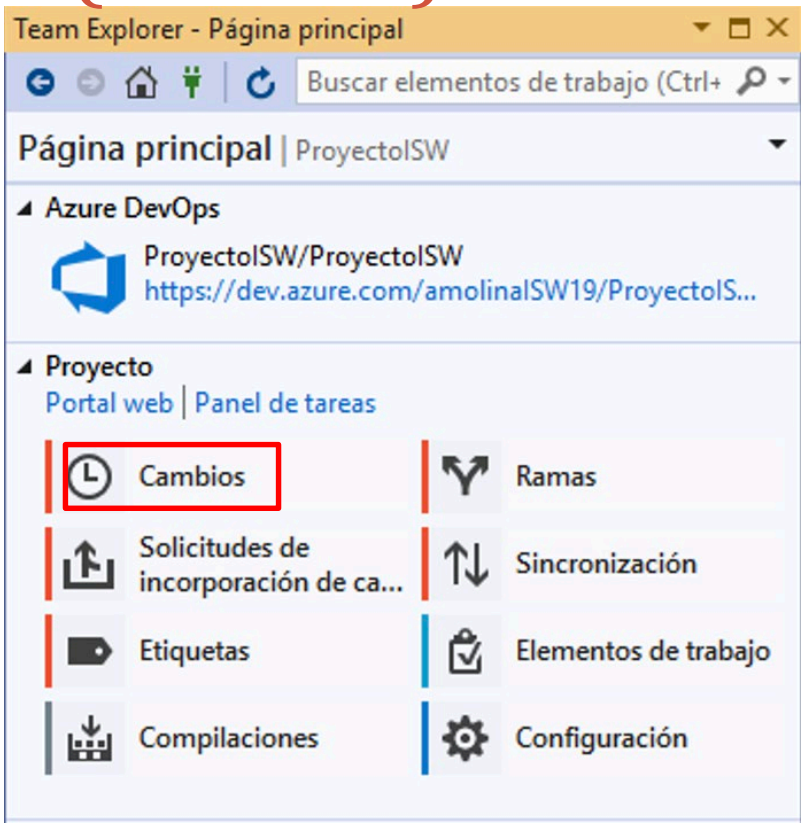
# Create Project in Visual Studio

Finally a solutions folder called “**Testing**” dhas to be added to the solution LabSolution

The Final structure must be as follows:



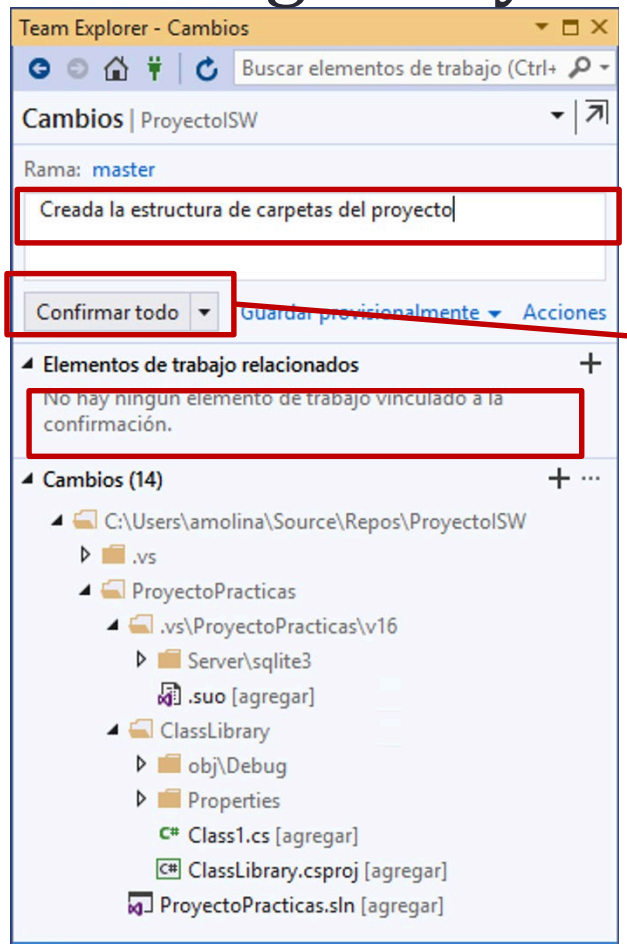
# Store your work in the local repository (Commit)




- 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

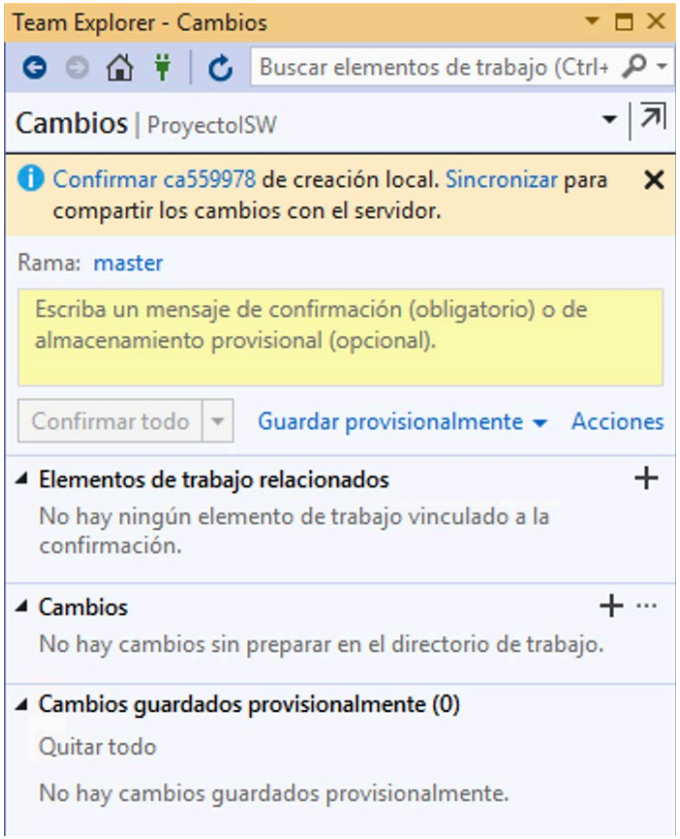
# Store your work in the local repository

- Confirm All: Create a commit with all pending changes in your local repository



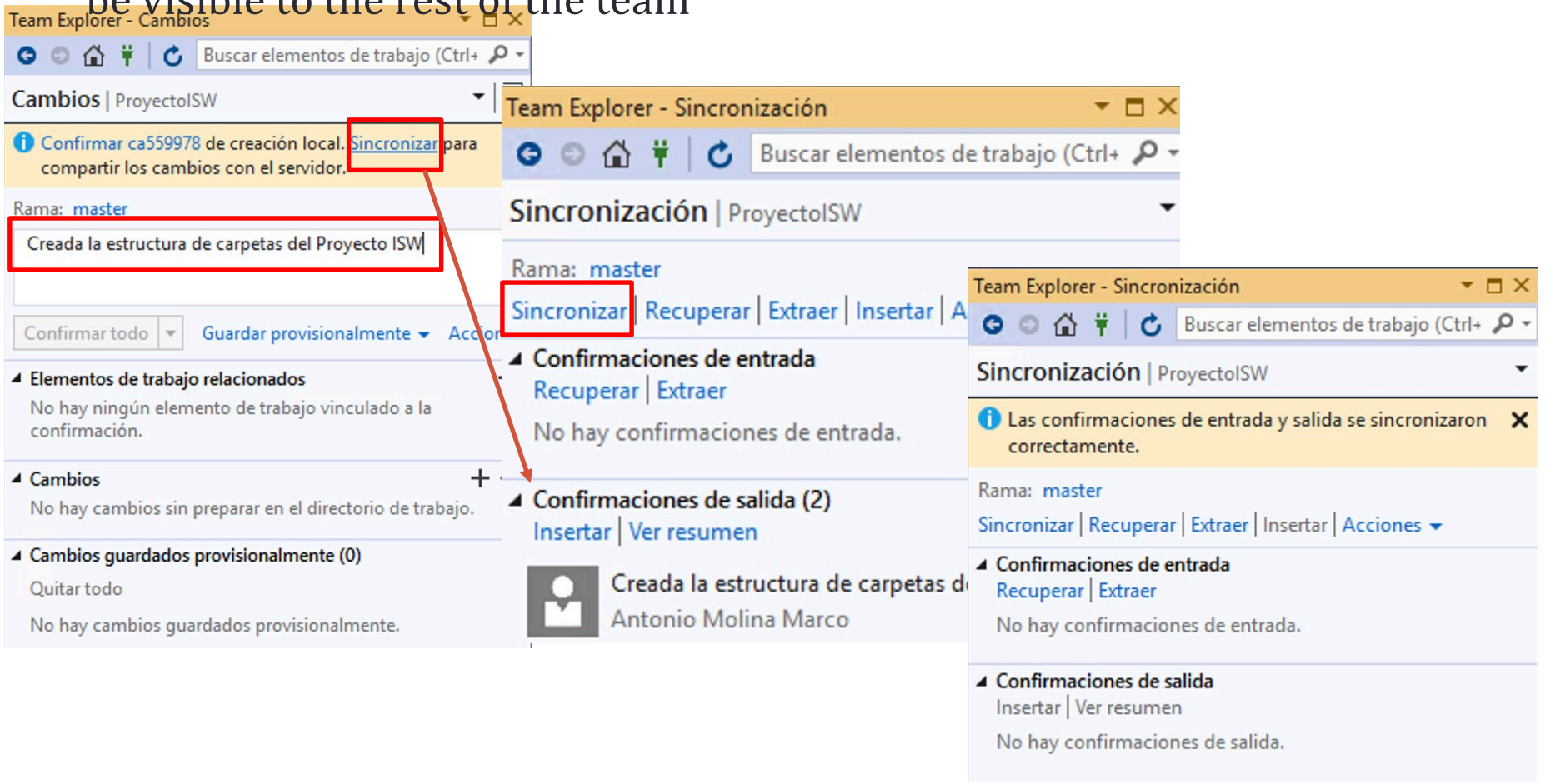

Descriptive text

Tasks from the plan may be associated

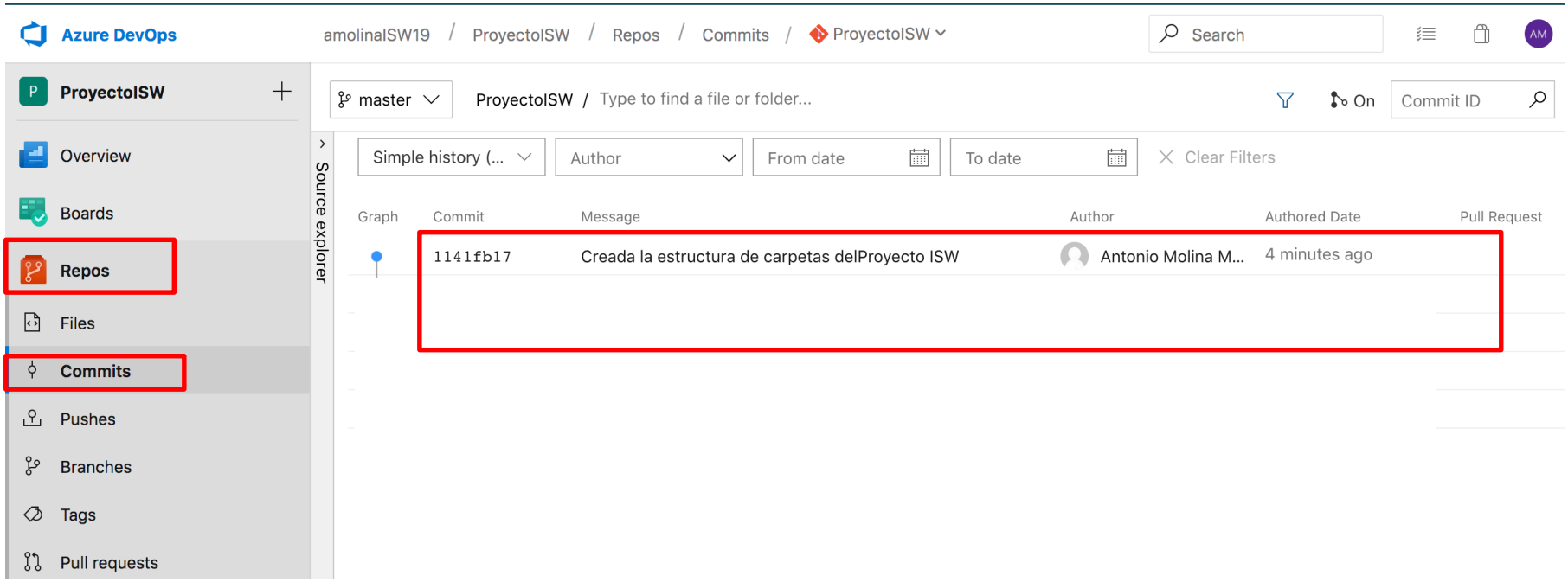


# Synchronize: share your work

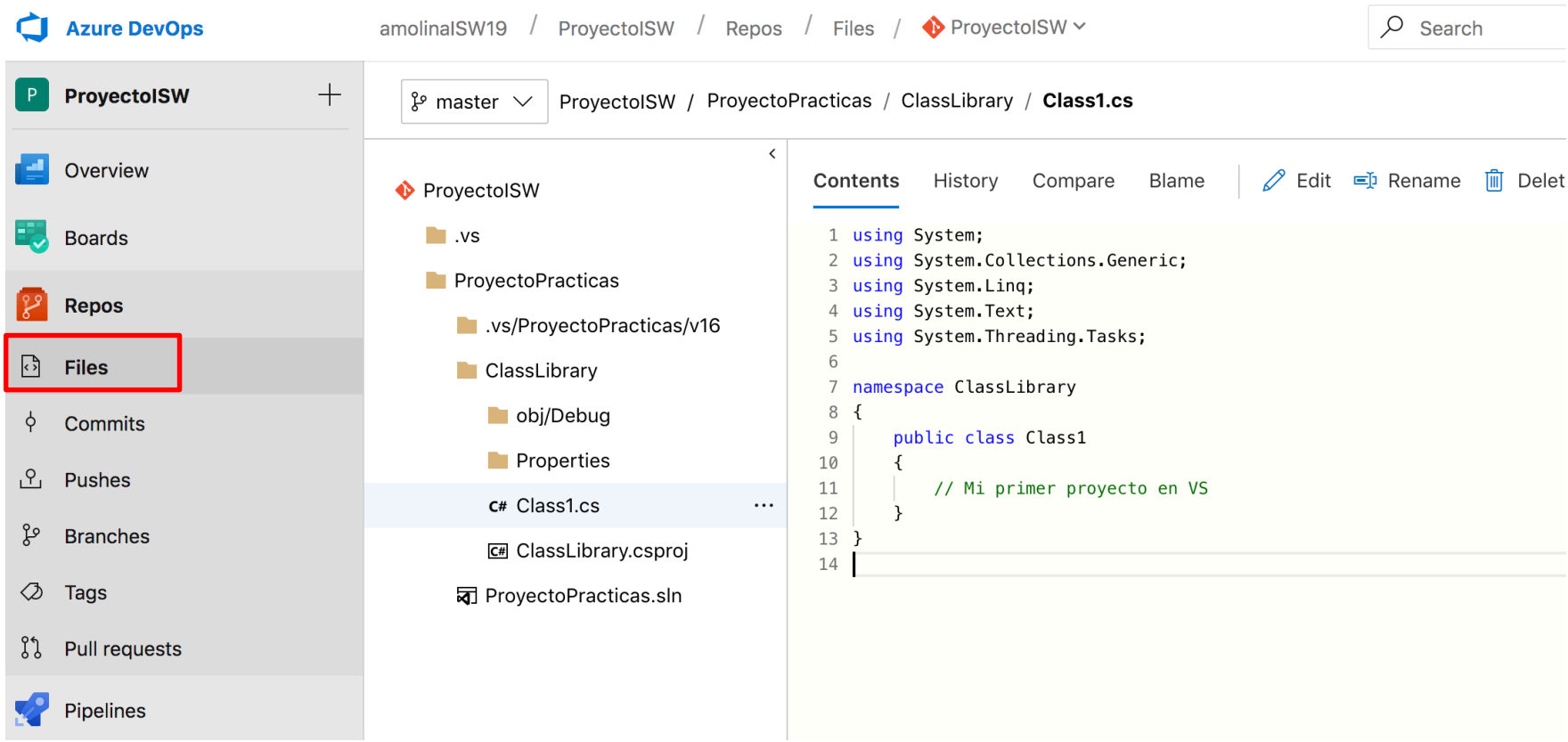
- Click  sincronizar to perform a ***push*** operation on your work, the local repositpory will be updated in the remote repository and your work Will be visible to the rest of the team



# See changes in the repository with Azure DevOps

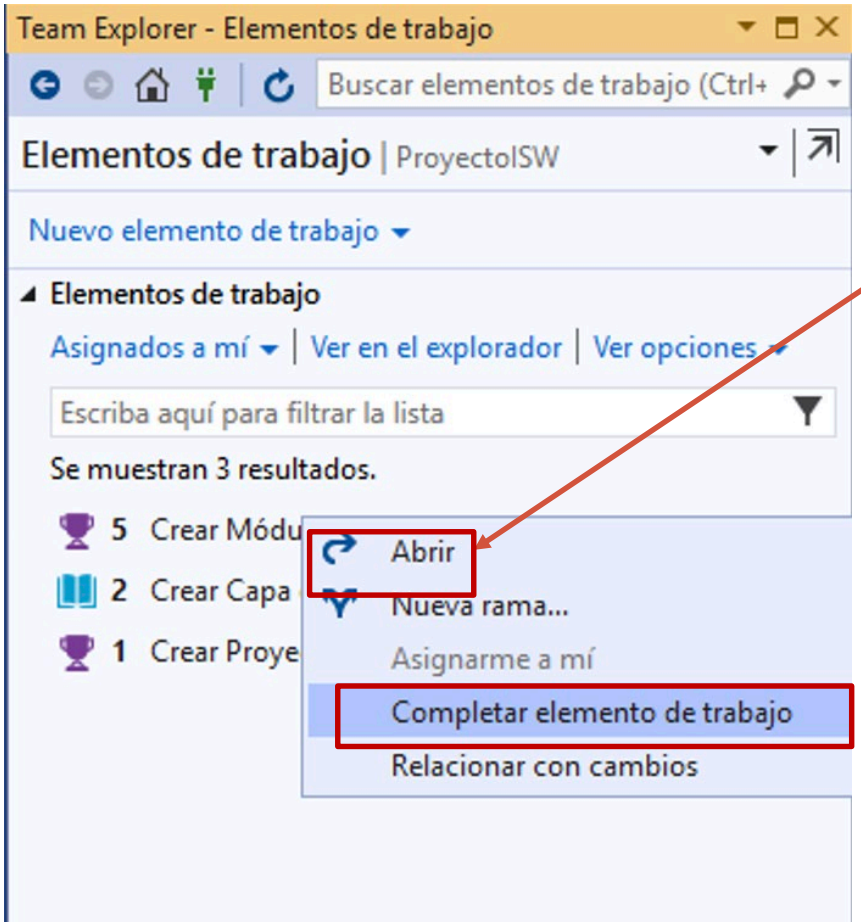


# Inspect code with Azure DevOps



# 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

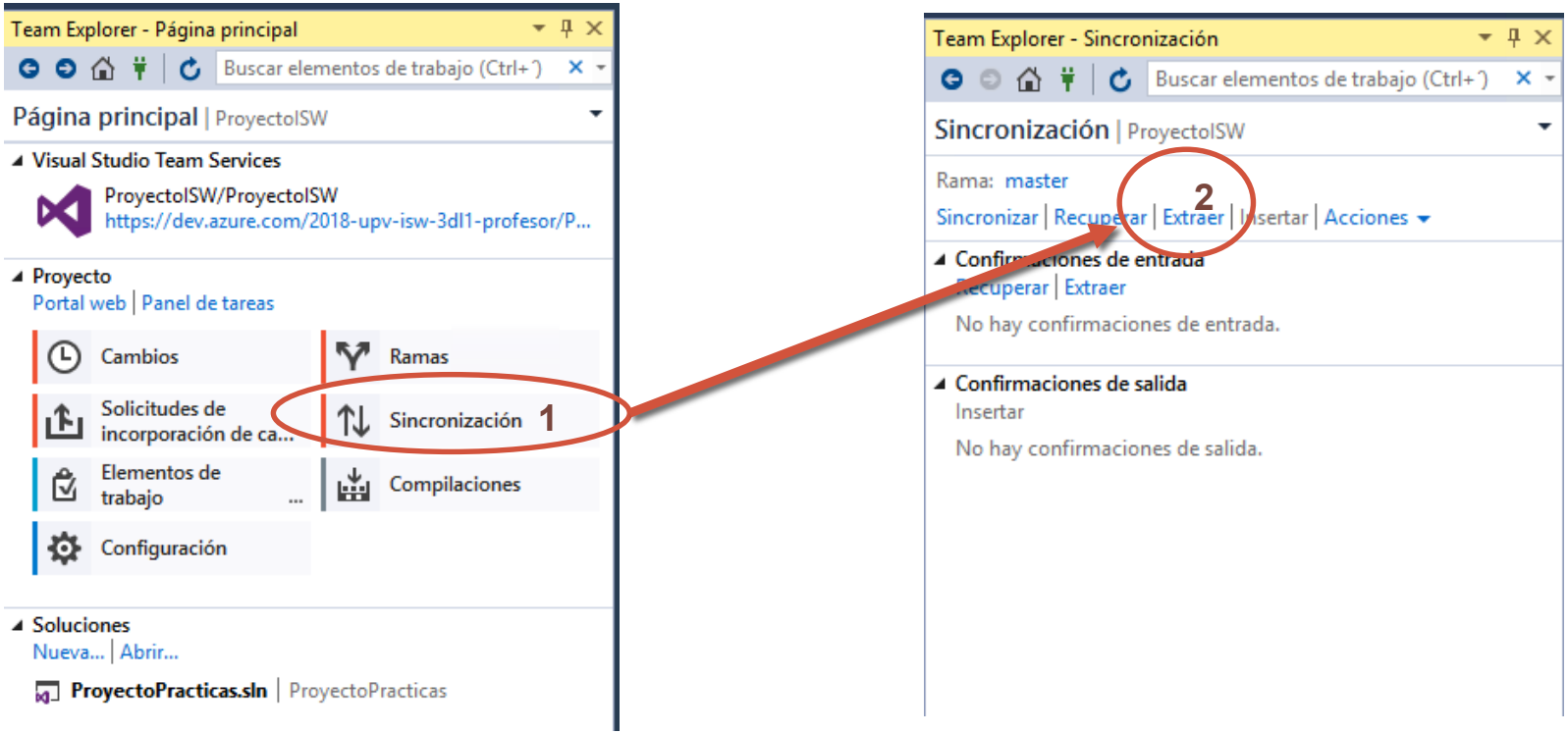
If an item is completed in VS the status Will be updated in the backlog and the board in Azure DevOps

# Retrieve the Project from the remote repository into Visual Studio

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

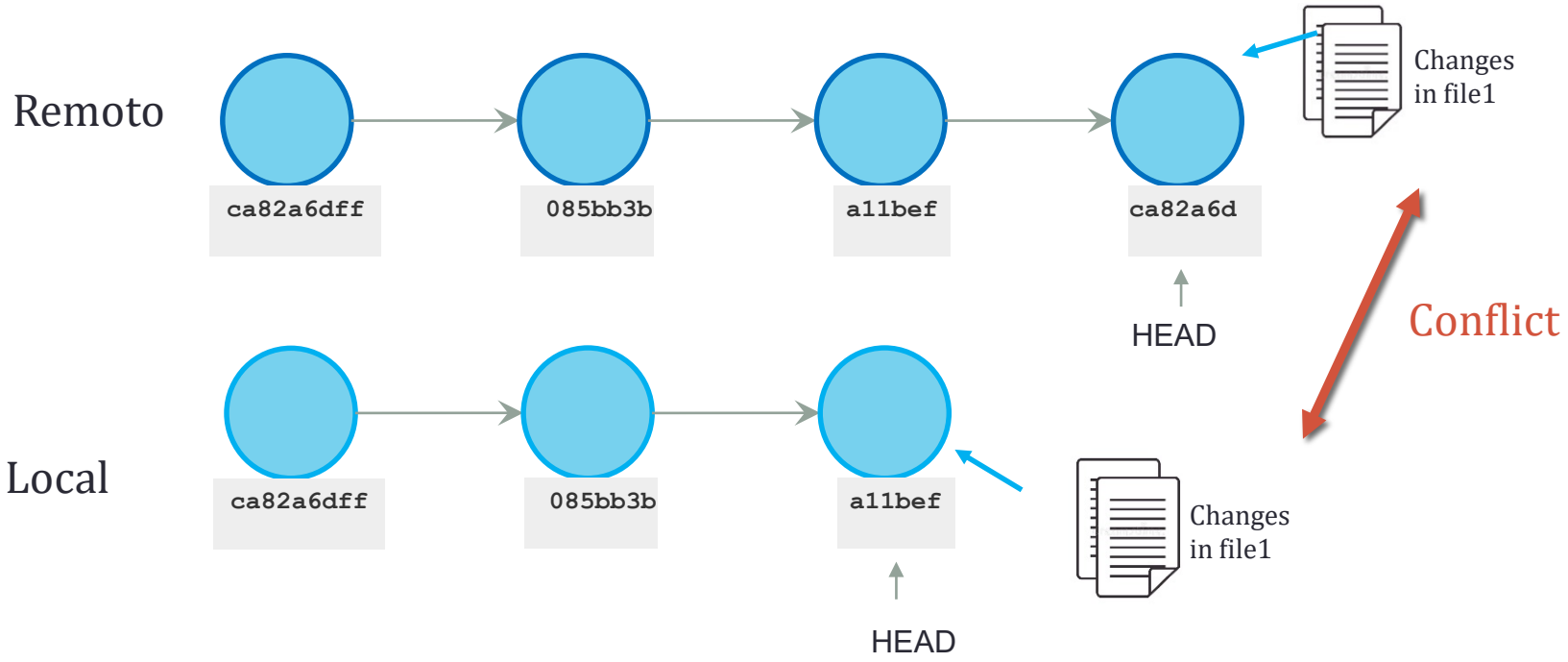
# Obtain latest changes

- To incorpórate the latest changes to your repository made by other users use the option **Extraer** (pull)

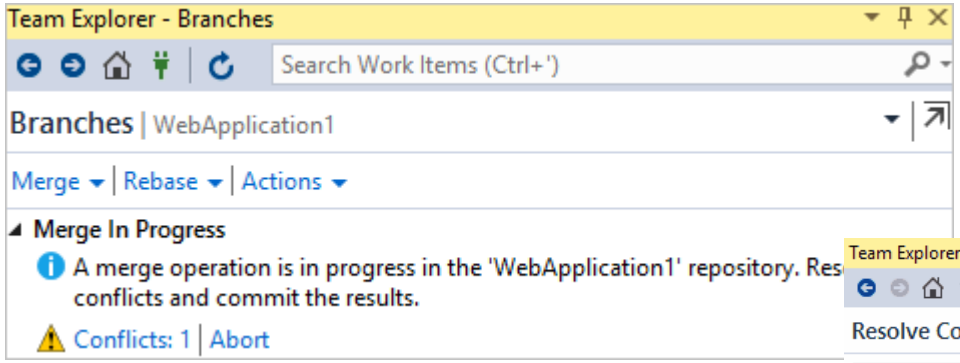


# Manage code conflicts

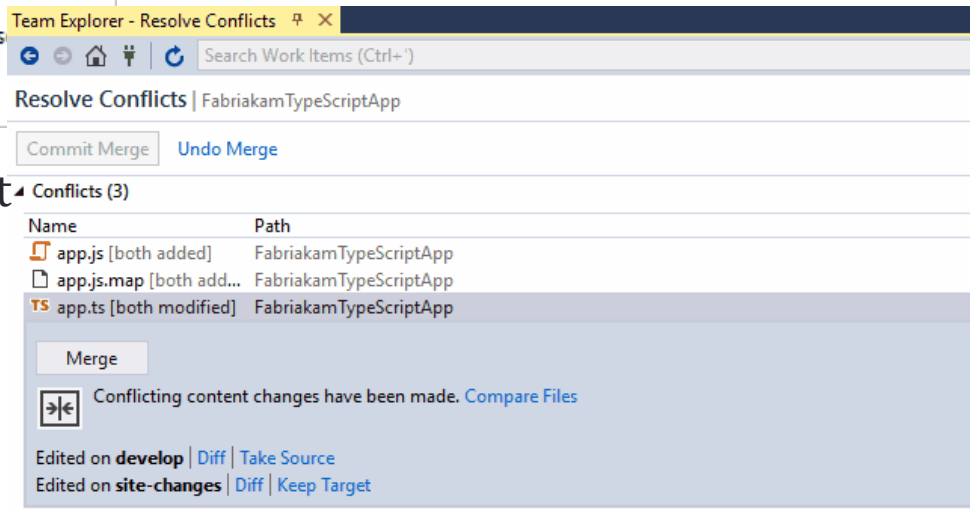
- When two developers work on the same file
  - A push by user 2 in the remote repository has updates in a file commited by user 1 locally



# Manage code conflicts




Select the correct versión 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 Visual Team Services
- 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

- Open a remote desktop connection
  - Server: windesktop.dsic.upv.es
  - User: DSIC\your_user_name (Assigned by UPV when you enrolled)
  - Password: your_password (foreign students your NIE or passport code as written in your official enrolment e.g.: X5567322)
- Visual Studio 2015 Enterprise is available in the laboratory virtualization