# VISUAL STUDIO INTEGRADO CON DEVOPS Y GIT

#### Seminario 1 - Desarrollo de SW

en Visual Studio

#### **DOCENCIA VIRTUAL**

#### Finalidad:

Prestación del servicio Público de educación superior (art. 1 LOU)

#### Responsable:

Universitat Politècnica de València.

Derechos de acceso, rectificación, supresión, portabilidad, limitación u oposición al tratamiento conforme a políticas de privacidad:

http://www.upv.es/contenidos/DPD/

#### Propiedad intelectual:

Uso exclusivo en el entorno de aula virtual.

Queda prohibida la difusión, distribución o divulgación de la grabación de las clases y particularmente su compartición en redes sociales o servicios dedicados a compartir apuntes.

La infracción de esta prohibición puede generar responsabilidad disciplinaria, administrativa o civil





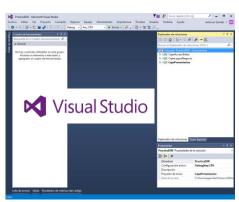
#### Objetivos

 Aplicar una metodología ágil en el desarrollo de software utilizando Azure DevOps (herramienta online) y combinarlo con la herramienta de desarrollo Microsoft Visual Studio

**Parte 1.** Gestión del proyecto con Azure DevOps(<u>herramienta</u> Web)



**Parte 2.** Desarrollo del proyecto desde Visual Studio (S2)



#### Parte 1. Gestión del proyecto desde la Web

 Aprender a elaborar un plan de proyecto software aplicando una metodología ágil, utilizando Azure DevOps

#### • Pasos:

- Crear una organización en AzureDevOps
- Creación de un nuevo proyecto
- Añadir miembros al equipo del proyecto
- Gestión del trabajo

Iniciar sesión

#### Acceder a DevOps

#### Desde la Web de Azure DevOps:

https://azure.microsoft.com/es-es/services/devops/

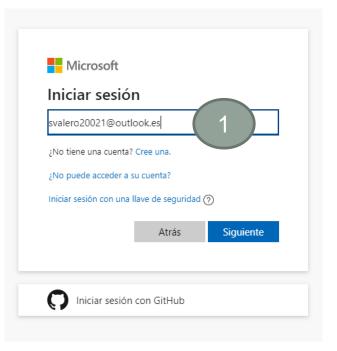


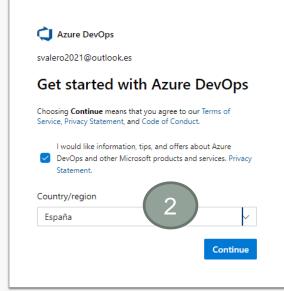


Mi cuenta

## Crear una organización DevOps

- 1. Iniciaremos sesión con una cuenta Microsoft
  - Crearos una cuenta gratuita para la asignatura
  - Podéis usar la cuenta upv.edu de la UPV, pero entonces detectará que pertenece a una organización ya conocida y obliga a que todos los miembros usen el mismo tipo de cuenta. De no hacerlo, se comporta de forma extraña
- 2. Creará una organización por defecto con tu nombre de usuario.
  - Una organización es un equipo que desarrolla proyectos SW

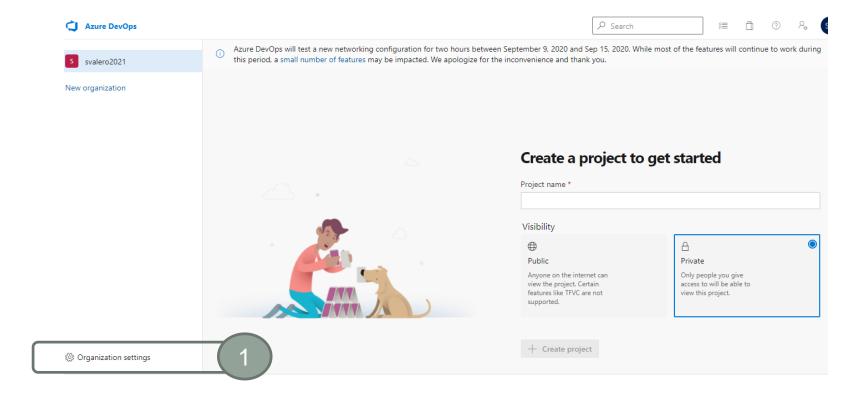






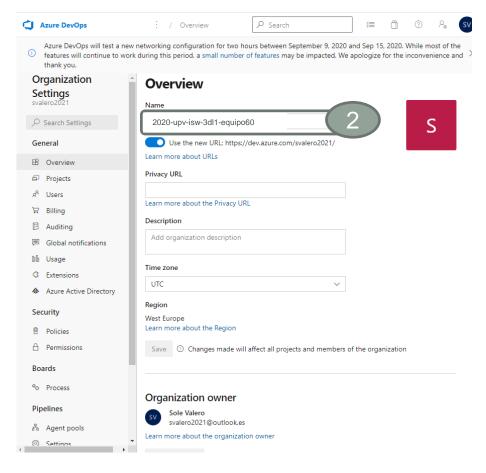
### Modificar una organización DevOps

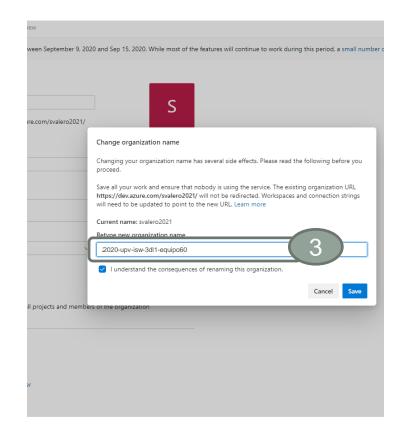
 Vamos a cambiar el nombre por defecto, con nuestro nombre de usuario, por otro con el nombre del equipo.



## Modificar una organización DevOps

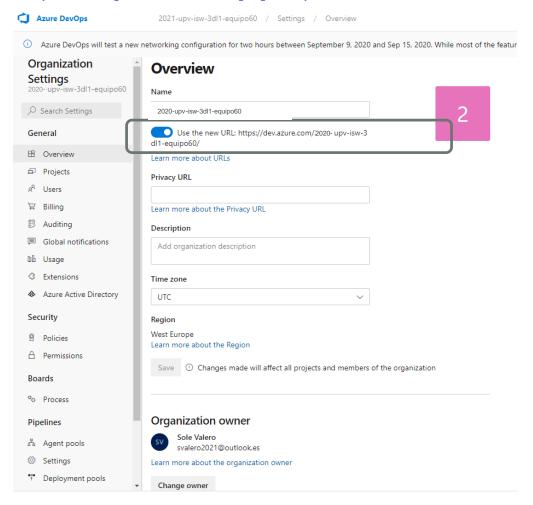
• Nombraremos a nuestro equipo con una combinación que incluye el nombre de la asignatura y el grupo de prácticas. Ej: 2020-upv-isw-3dl1-equipo60 (no se puede repetir)



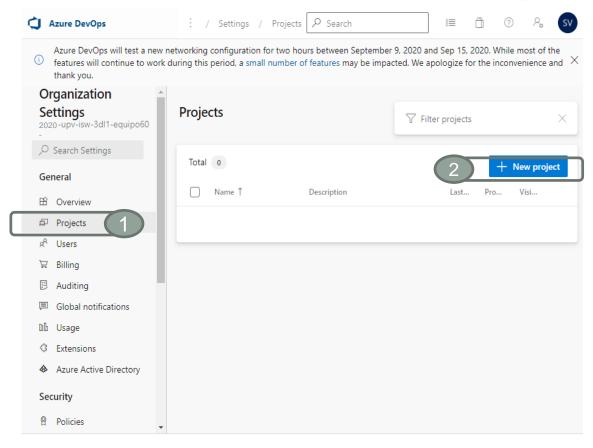


### Modificar una organización DevOps

 A partir de aquí ya dispondremos de un servidor en la para nuestra organización: <a href="https://dev.azure.com/2020-upv-isw-3dl1-equipo60/">https://dev.azure.com/2020-upv-isw-3dl1-equipo60/</a>

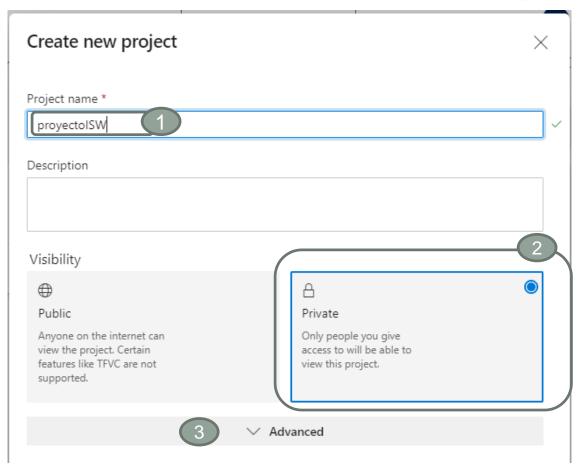


#### Creación de un nuevo proyecto



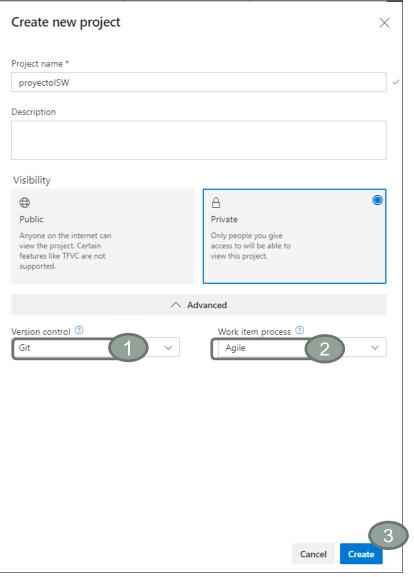
- a) Creamos un nuevo proyecto
- o) Introducimos el nombre y su visibilidad (que será privada)
- c) Desplegamos las características avanzadas
- d) Seleccionado git como control de versiones y Agile como metodología
- e) Creamos el proyecto

#### Creación de un nuevo proyecto



- a) Creamos un nuevo proyecto
- b) Introducimos el nombre y su visibilidad (que será privada)
- c) Desplegamos las características avanzadas
- d) Seleccionado git como control de versiones y Agile como metodología
- e) Creamos el proyecto

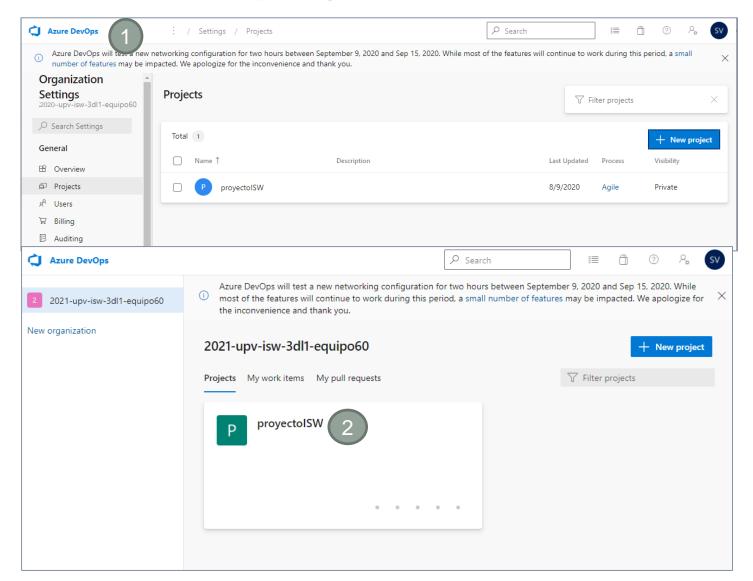
Creación de un nuevo proyecto

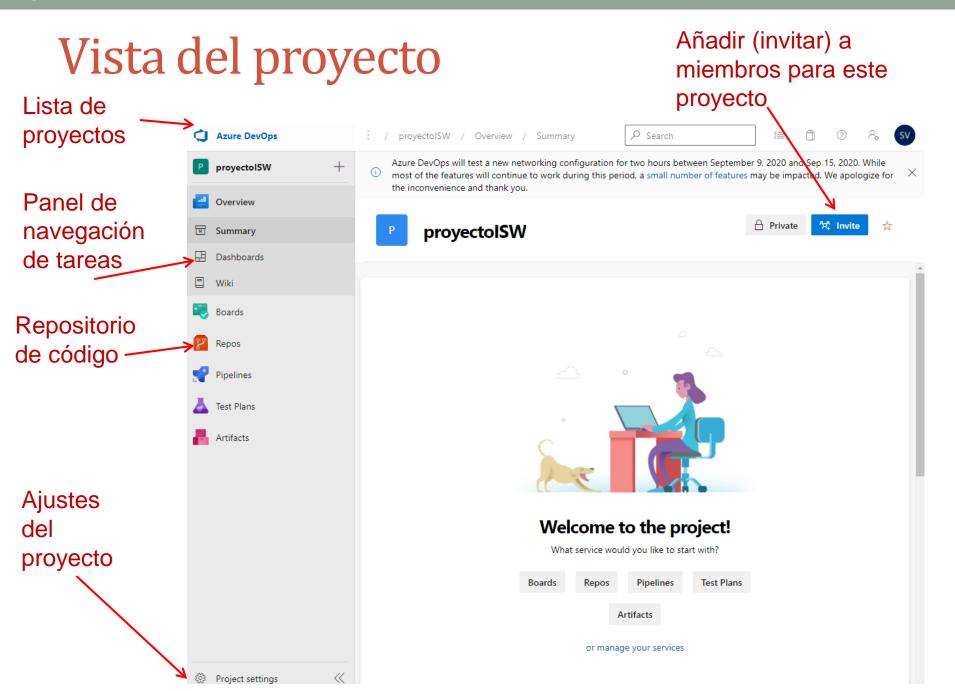


- a) Creamos un nuevo proyecto
- o) Introducimos el nombre y su visibilidad (que será privada)
- c) Desplegamos las características avanzadas
- d) Seleccionado git como control de versiones y Agile como metodología
- e) Creamos el proyecto

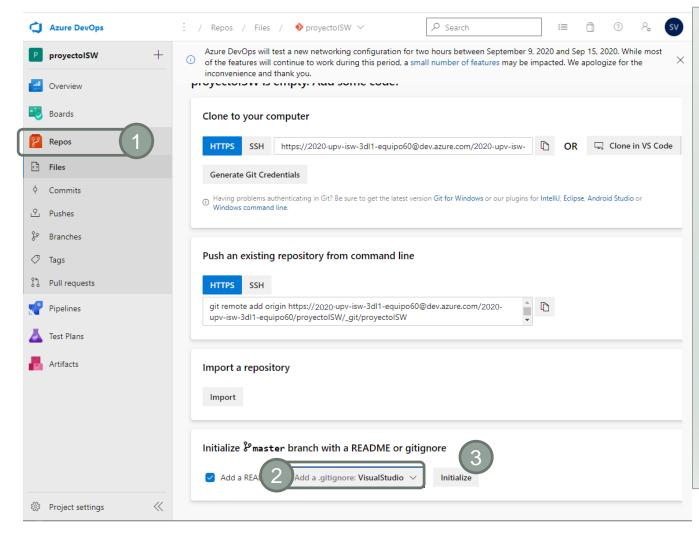


#### Vista del proyecto





### Configurar el repositorio

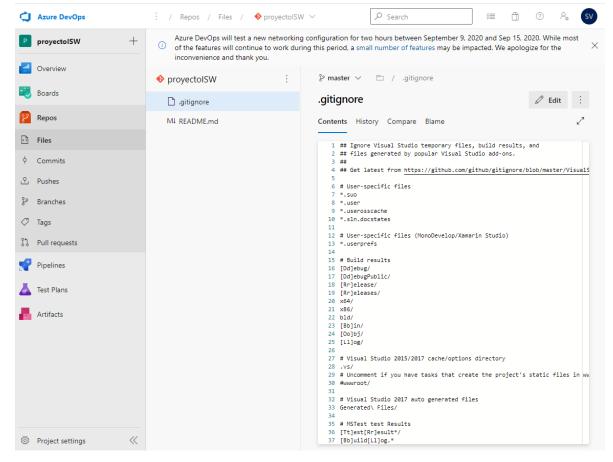


Inicializamos el repositorio git para que trabaje adecuadamente con Visual Studio:

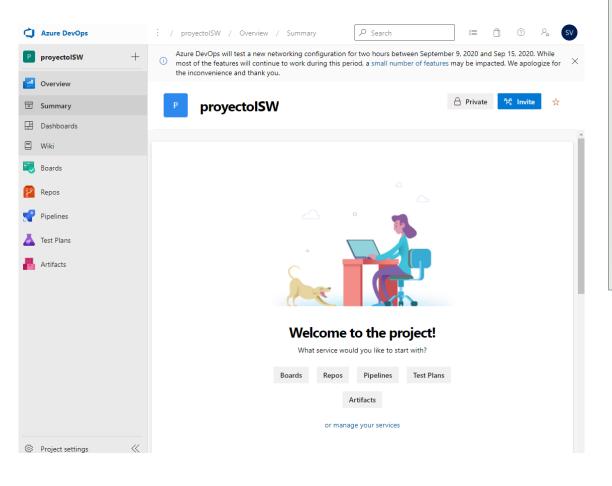
- 1. Seleccionamos Repos
- Seleccionamos
   VisualStudio de
   la pestaña Add a
   .gitignore
- 3. Seleccionamos *Initialize*

#### Configurar el repositorio

 Nos crea un archivo .gitignore y un README.md por defecto.



#### Añadir miembros al proyecto

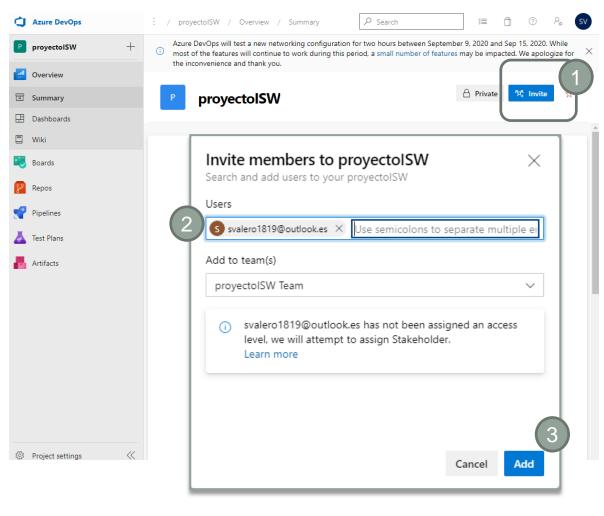


Añadir nuevos miembros al equipo es muy sencillo (5 usuarios básicos gratuitos)

Invitamos a los miembros en base a su correo electrónico. El usuario recibirá un correo de invitación para que se una al proyecto

NOTA: el email para la invitación debe estar asociado a una cuenta de Microsoft, o tendrá que asociarla antes de unirse

#### Añadir miembros al proyecto



Añadir nuevos miembros al equipo es muy sencillo (5 usuarios básicos gratuitos, stakeholder)

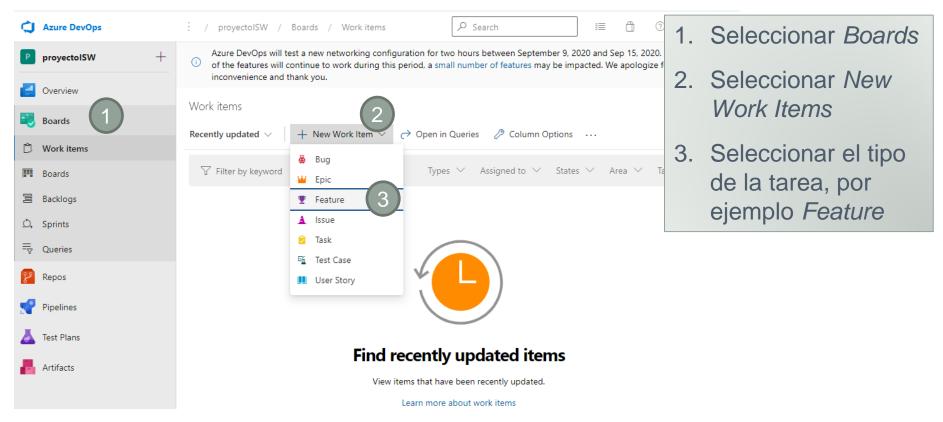
Invitamos a los miembros en base a su correo electrónico. El usuario recibirá un correo de invitación para que se una al proyecto

NOTA: el email para la invitación debe estar asociado a una cuenta de Microsoft, o tendrá que asociarla antes de unirse

 Si os equivocáis al introducir el email de un miembro del equipo, debéis eliminarlo desde Project Setting/Teams antes de añadir el correcto

## Ítems de trabajo: Creación

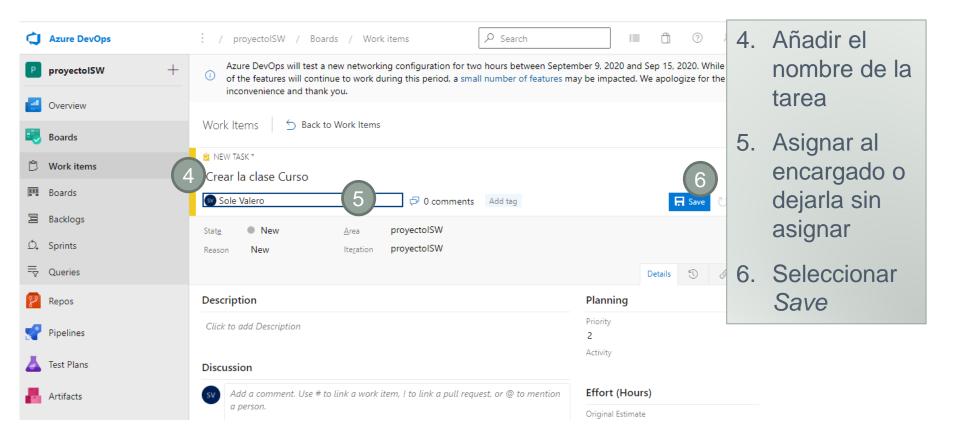
 Podéis crear ítems de trabajo para repartir el trabajo a realizar entre los miembros del equipo



\* Para más información sobre work items, podéis acceder a la documentación de Microsoft

19

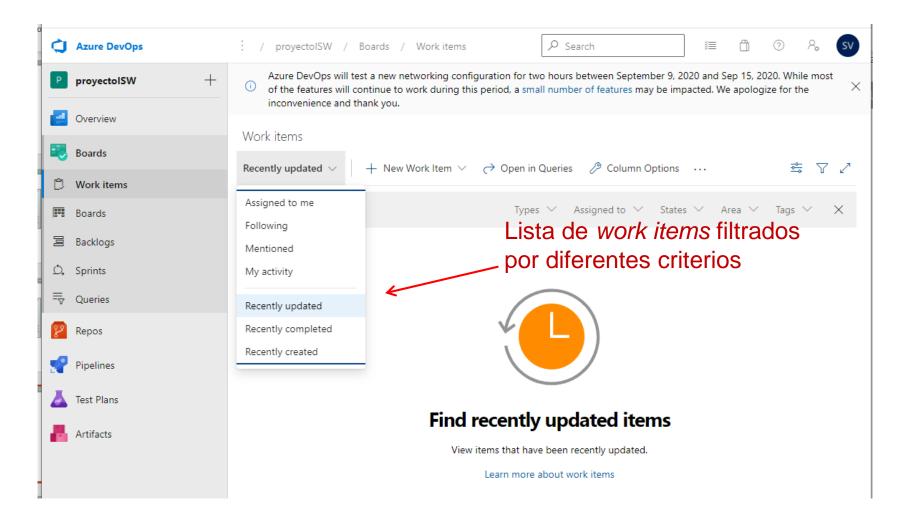
## Ítems de trabajo: Creación



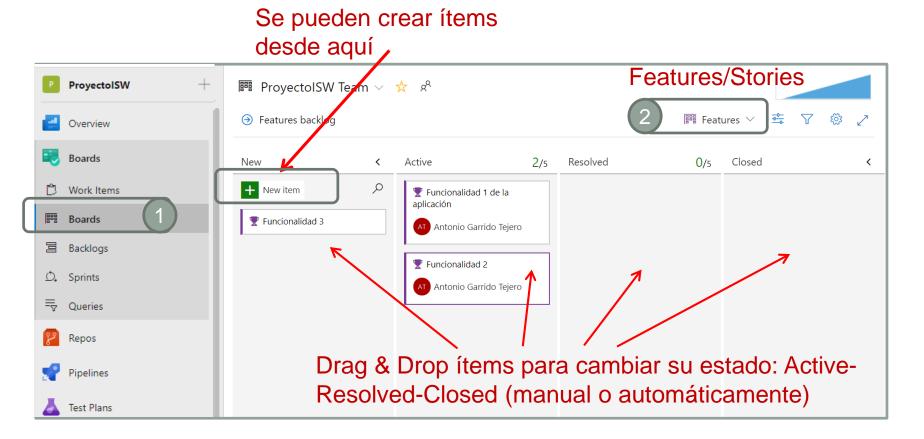
<sup>\*</sup> Para más información sobre work items, podéis acceder a la documentación de Microsoft

20

## Ítems de trabajo: Visualización



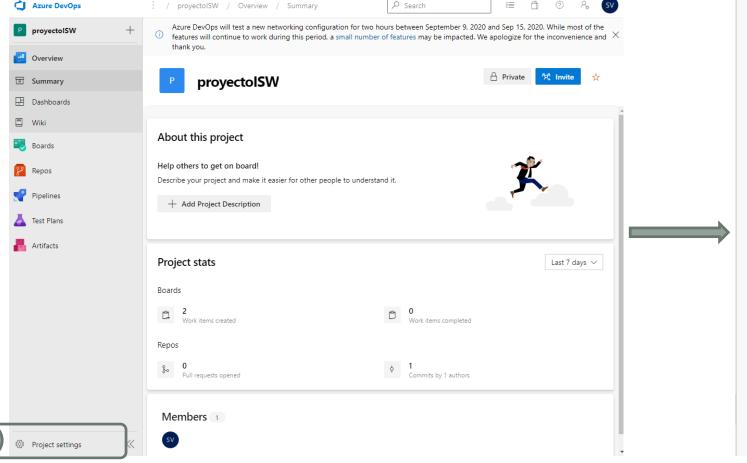
### Ítems de trabajo: Visualización desde el board

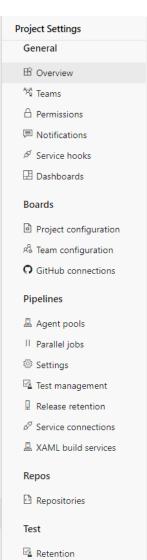


El *board* (lienzo) consta de un conjunto de ítems. Los ítems se usan para compartir información, asignar trabajo a los miembros del proyecto, monitorizar las dependencias, organizar el trabajo, etc.

#### Configuración y ajustes del proyecto

• Se pueden gestionar los permisos de usuarios, notificaciones, equipos, etc.



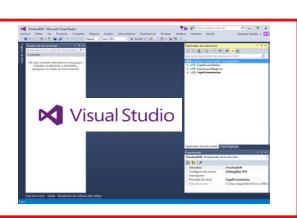


#### Objetivos

 Aplicar una metodología ágil en el desarrollo de software utilizando Azure DevOps (parte Web) y combinarlo con las tareas de diseño y codificación en Microsoft Visual Studio

**Parte 1**. Gestión del proyecto desde la Web (Seminario Tema 2)

**Parte 2.** Desarrollo del proyecto desde Visual Studio

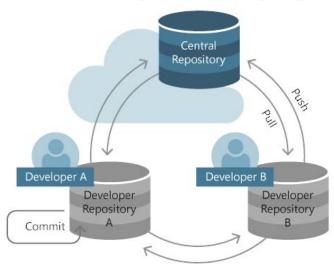


Utilice todos los servicios de DevOps o elija solo lo que necesita para complementar sus fluios de

Planeamiento más inteligente, mejor colaboración y envíos más

#### Control de Versiones en Visual Studio

 Un sistema de control de versiones permite no solo salvar el trabajo realizado, sino coordinar los cambios de código realizados por el equipo



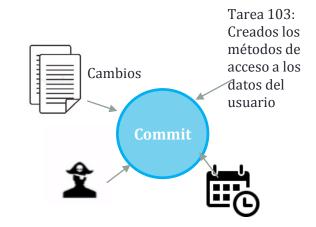
Tenemos disponibles dos sistemas de control de versiones:

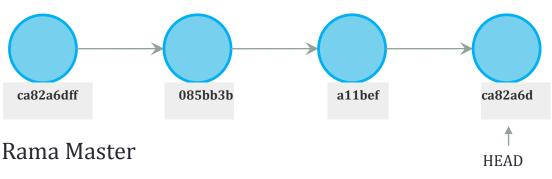
- Git
- TFS

Documentación Azure Git: <a href="https://docs.microsoft.com/en-us/azure/devops/repos/git/?view=azure-devops">https://docs.microsoft.com/en-us/azure/devops/repos/git/?view=azure-devops</a>

#### Flujo de trabajo en Git

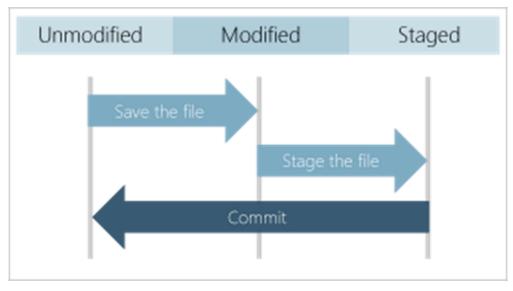
- El flujo de trabajo normal con Git es:
  - Clonar un repositorio remoto existente
  - Trabajar en tus tareas
  - Guardar tu trabajo de forma local, a través de un commit
  - Compartir tus cambios con tu equipo, realizando push





#### ¿Cómo monitoriza Git los cambios?

- Unmodified: ficheros que no han sido modificados desde el último *commit*
- Modified: ficheros que han sido modificados, pero todavía no marcados para ser incluidos en el siguiente commit
- Staged: ficheros modificados y marcados para ser añadidos en el siguiente commit



#### Parte 2. Desarrollo del proyecto desde Visual Studio

Crear un proyecto software mediante Microsoft
 Visual Studio, recuperando (y completando) el plan de proyecto elaborado con Azure Boards

#### • Pasos:

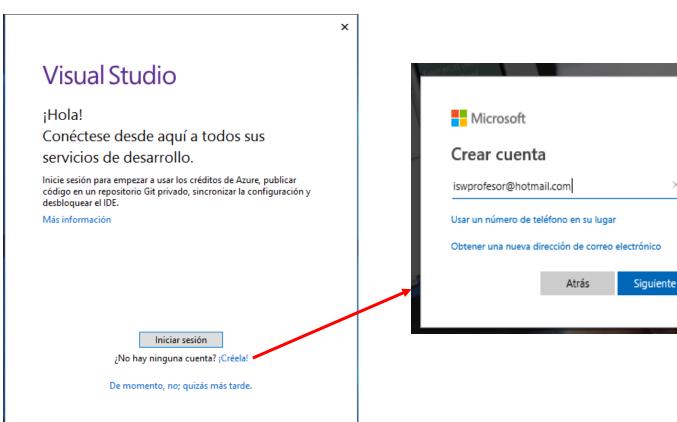
- Crear una cuenta Microsoft (si no se dispone de una)
- Crear el proyecto en Visual Studio (por primera vez)
- Gestionar el proyecto en Visual Studio
- Recuperar el proyecto desde el repositorio hacia Visual Studio
- Gestionar conflictos en el código

#### Crear una cuenta desde Visual Studio



Al Iniciar Visual Studio lo primero que haremos es **iniciar la sesión** con una cuenta existente o **crear una nueva cuenta**.

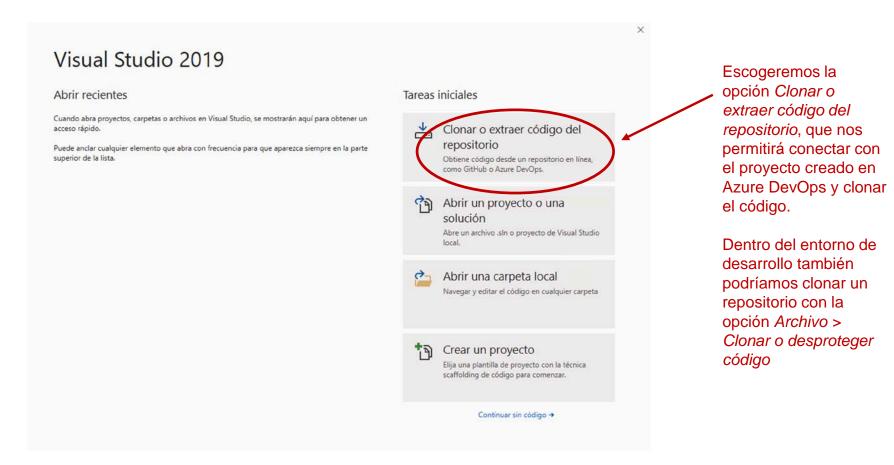
Una vez en el entorno podríamos cambiar de cuenta desde: *Archivo > Configuración de la cuenta ...* 



Iniciamos sesión con la cuenta usada en Azure DevOps

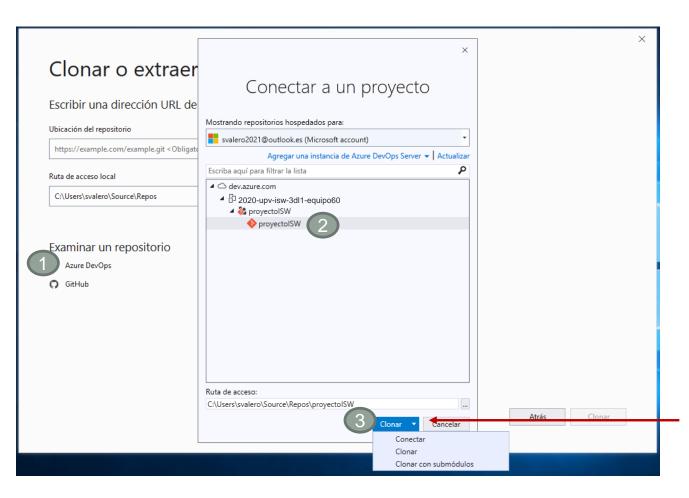
#### Crear el proyecto en Visual Studio

La primera pantalla nos muestra las tareas iniciales más habituales, incluyendo enlaces a los proyectos recientes.



#### Crear el proyecto en Visual Studio. Clonar repositorio.

**Escogeremos** la opción *Examinar un repositorio > Azure Devops* 



Seleccionaremos el proyecto que tendremos alojado en Azure DevOps (ProyectoISW).

Conectaremos y clonaremos el repositorio del proyecto en un solo paso con la opción Clonar. Si escogiéramos la opción Conectar podemos Clonar después dentro de VS.

Por el momento, esta operación la realizará únicamente el RESPONSABLE del proyecto.

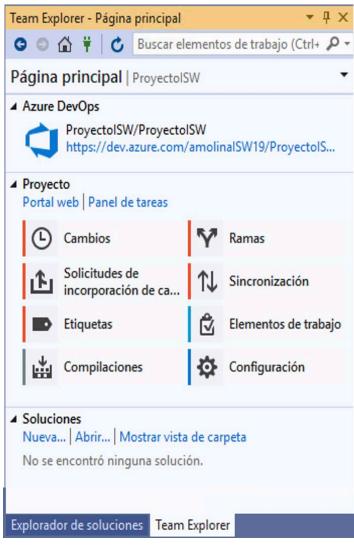
Podemos observar la ruta donde se almacenarán localmente los archivos del proyecto.

# Crear el proyecto en Visual Studio. Clonar repositorio.

A la derecha de la pantalla veremos el explorador de equipo (Team Explorer) desde donde realizaremos las operaciones básicas para la gestión de cambios en los archivos de código fuente, su sincronización con el repositorio Azure DevOps, así como la gestión de los elementos de trabajo.

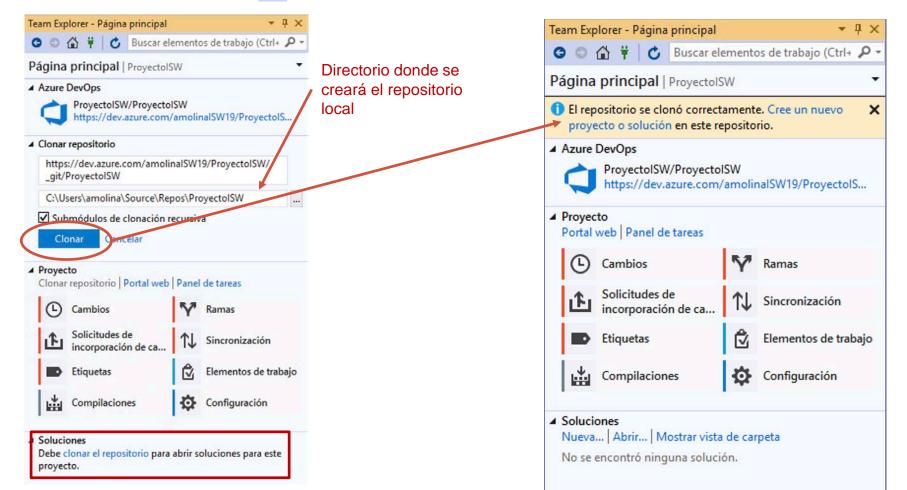
También podemos acceder al explorador desde el menú de VS:



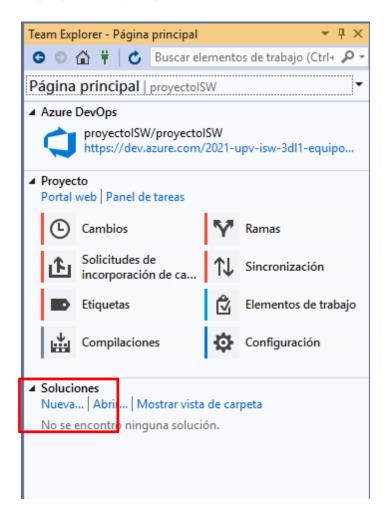


# Crear el proyecto en Visual Studio. Clonar repositorio.

Si no hubiéramos clonado el repositorio del proyecto VS nos advertiría de ello. Con la opción podemos administrar las conexiones a los repositorios.



# Crear el proyecto en Visual Studio. Crear solución.

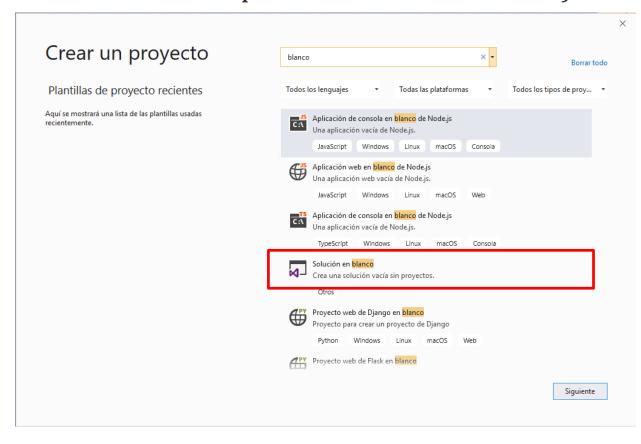


La primera vez, el RESPONSABLE del proyecto crea una nueva solución.

En Visual Studio una
Solución es una colección
de Proyectos. Crearéis
diversos proyectos dentro de
una misma Solución

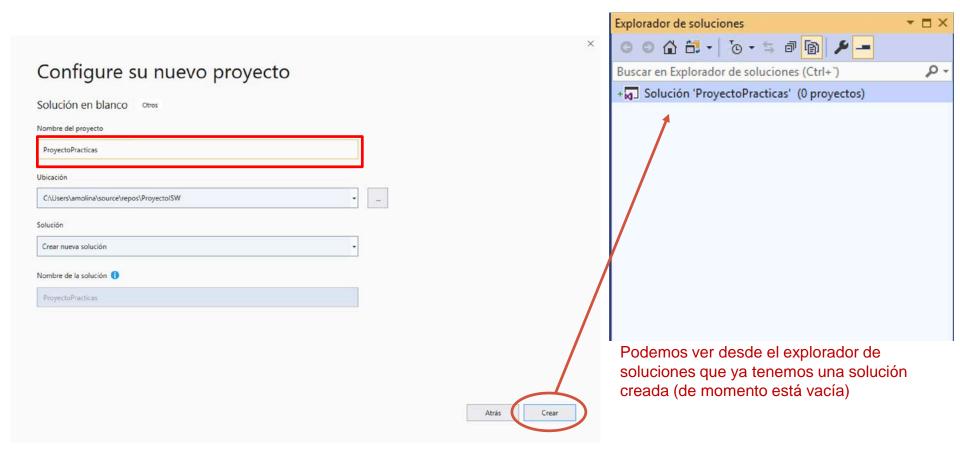
# Crear el proyecto en Visual Studio. Crear solución.

Crearemos una **solución en blanco**, a la que podremos añadir distintos tipos de proyecto para C# y otros lenguajes (Aplicación de Consola, Bibliotecas de Clases, Aplicaciones Windows, etc.)



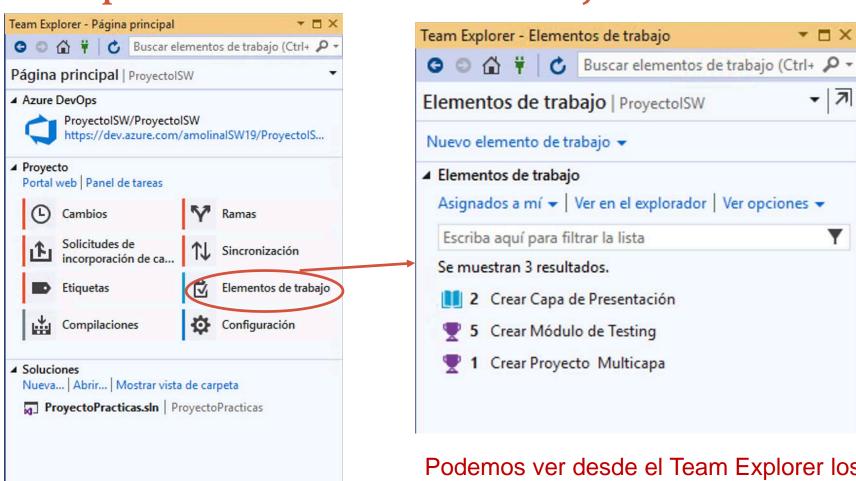
# Crear el proyecto en Visual Studio. Crear solución.

Daremos un nombre a la solución (ProyectoPracticas)





### Crear el proyecto en Visual Studio: Recuperar elementos de trabajo del Plan.



Podemos ver desde el Team Explorer los elementos de trabajo (*work ítems*) que tenemos asignados.

Ahora crearemos la estructura de carpetas de nuestro proyecto.

Separaremos la capa de Presentación de las capas de Lógica y Persistencia.

La carpeta correspondiente a la capa de Presentación contendrá el código de la interfaz de usuario (que podría ser una aplicación de escritorio, web o móvil).

El código de las capas Lógica y Persistencia se ensamblará en una misma biblioteca de clases (dll).

Podemos agregar una nueva carpeta de soluciones desde el menú de VS:

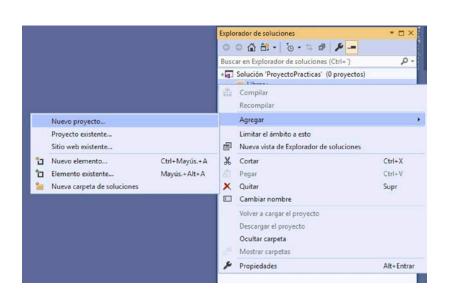


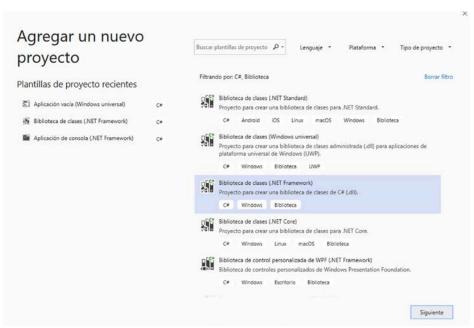
Proyecto > Agregar nueva carpeta de soluciones

Dentro de una carpeta de soluciones podemos crear carpetas para organizar el código.

Iniciamos el elemento de trabajo "Crear Capa de Presentación" agregando una Nueva carpeta de soluciones... denominada "**Presentation**". Añadir dentro un *Nuevo Elemento* de tipo *Clase de Visual C#*.

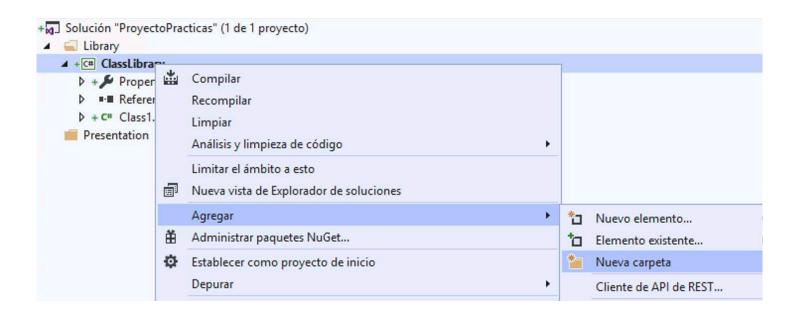
Análogamente agregaremos una Nueva carpeta de soluciones denominada "**Library**". Desde el Explorador de Soluciones agregaremos a Library un Nuevo Proyecto del tipo *Biblioteca de clases (.NET Framework)* al que llamaremos "**ClassLibrary**".





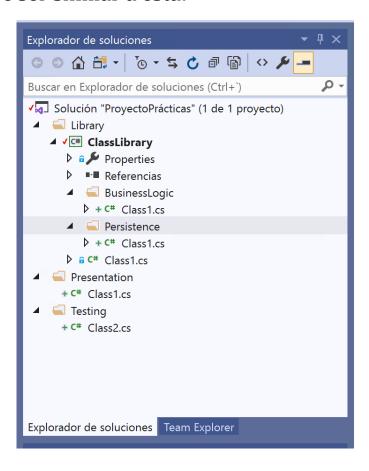
El proyecto **ClassLibrary** contendrá dos nuevas carpetas: "**BusinessLogic**" y "**Persistence**".

Las carpetas se agregan desde el Explorador de Soluciones con la opción *Agregar > Nueva carpeta*. Después, añadir a cada carpeta un *Nuevo elemento* de tipo *Clase*.

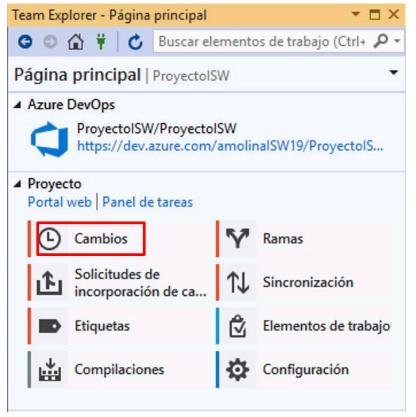


Por último, crearemos una nueva carpeta de soluciones para el módulo de "**Testing**" dentro de la solución ProyectoPracticas, de igual forma que las anteriores.

Nuevamente, añadir un elemento de tipo *Clase de Visual C#* a la carpeta. La estructura final debe ser similar a esta:



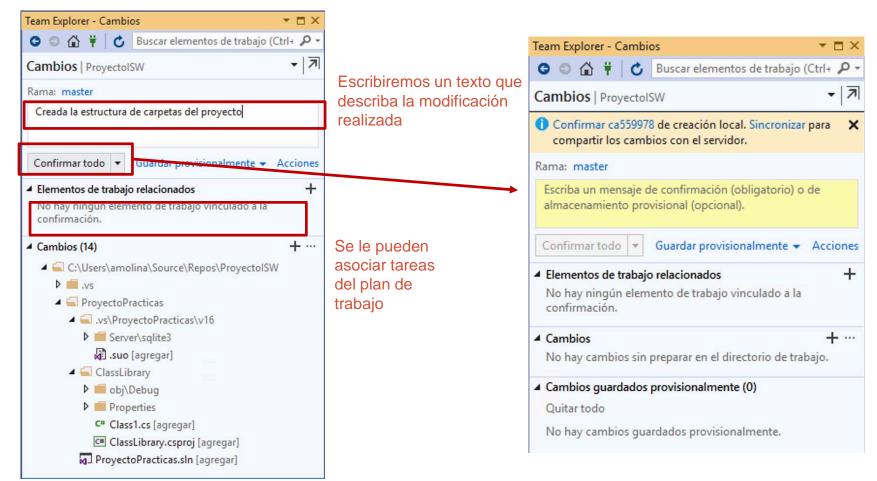
#### Guarda tu trabajo en tu repositorio local



- Cada vez que tengas un cambio significativo:
  - Realiza un *commit* en tu repositorio local
  - Añade un comentario descriptivo de tu trabajo, iniciado con el nombre de tarea
- Un commit NO GUARDA tu trabajo en el repositorio remoto, por tanto tus compañerxs no ven tus cambios

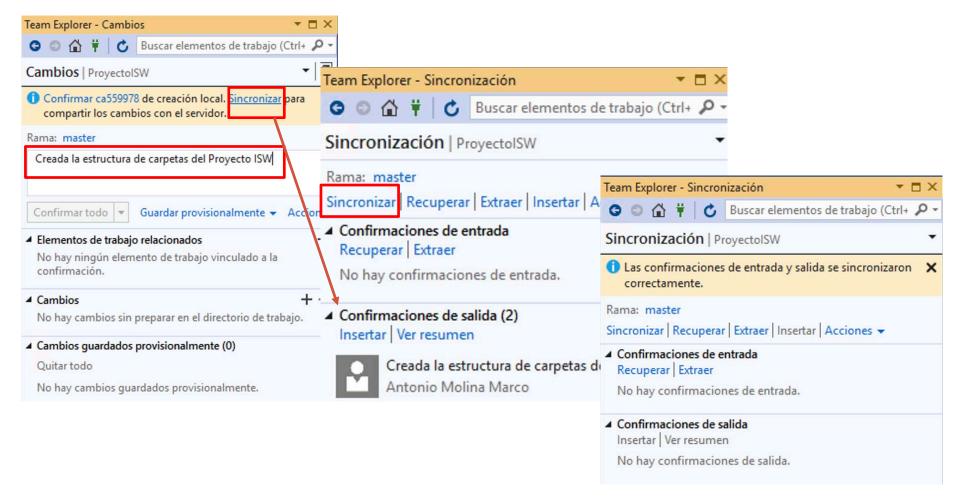
#### Guarda tu trabajo en tu repositorio local

 Confirmar todo: Crea un commit con todos los cambios pendientes en tu repositorio local

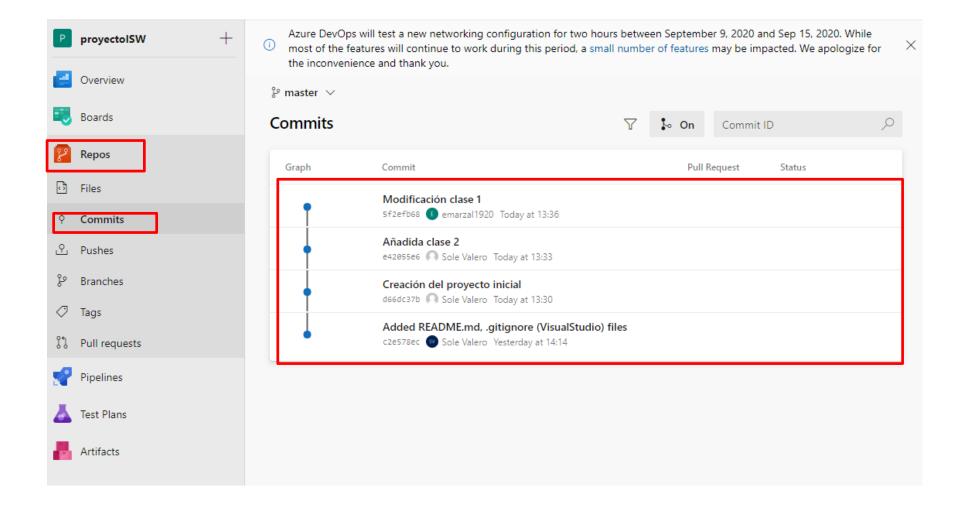


#### Sincronizar: compartir tu trabajo

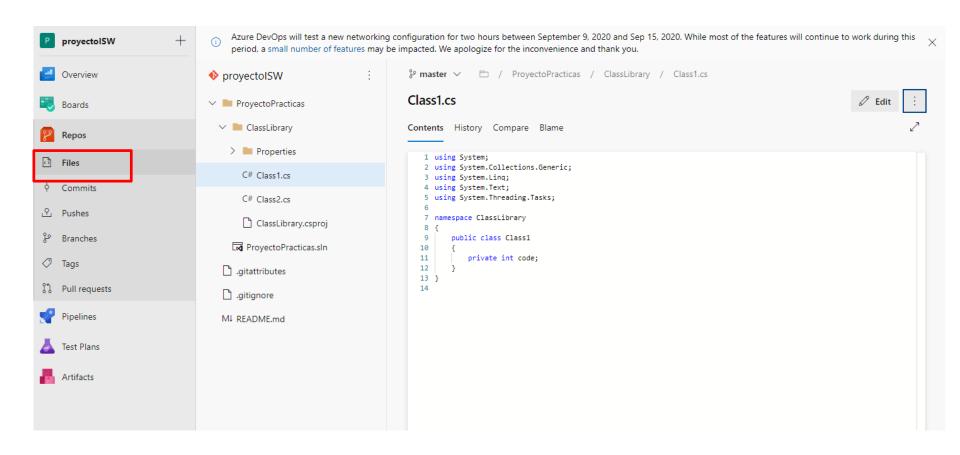
 Pulsa sincronizar para hacer un *push* de tu trabajo, se actualizará el repositorio remoto y se compartirá tus cambios con el resto del equipo



# Ver los cambios en el repositorio desde Azure DevOps

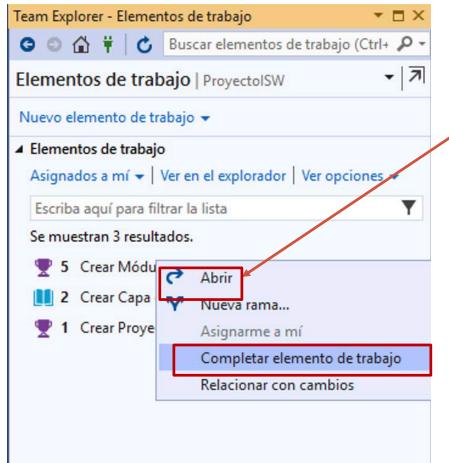


## Ver código desde Azure DevOps



#### Gestionar el proyecto en Visual Studio

 Desde Visual Studio también podemos controlar el estado de los work items "stories" / "tasks" y marcarlas como completadas (closed) cuando todos los tests han tenido éxito.



Podemos abrir el work ítem directamente en Azure DevOps desde VS

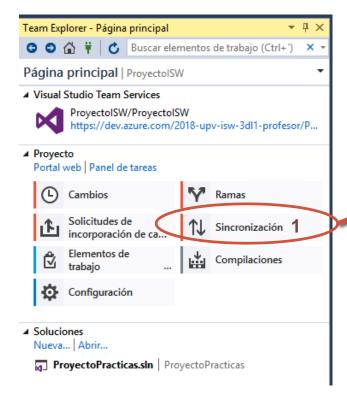
Si completamos un work ítem desde VS, se acualizará el backlog y el board de Azure DevOps con el cambio realizado

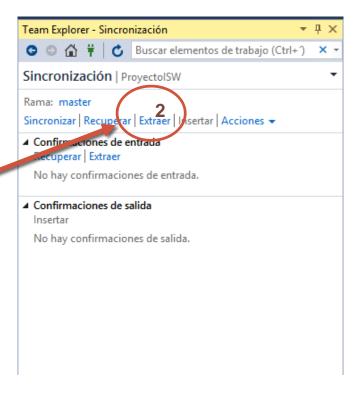
# Recuperar el proyecto desde el repositorio hacia Visual Studio

- Siguiendo los pasos indicados anteriormente cada miembro del proyecto podrá recuperar el proyecto desde el repositorio de Azure DevOps (ver pág. 11 y siguientes)
  - Clona la última versión del proyecto
  - Crea un repositorio local en su máquina

#### Obtener los últimos cambios

 Para incorporar a tu repositorio los cambios (pushes) que ha realizado otro compañero se utiliza la opción Extraer (pull)

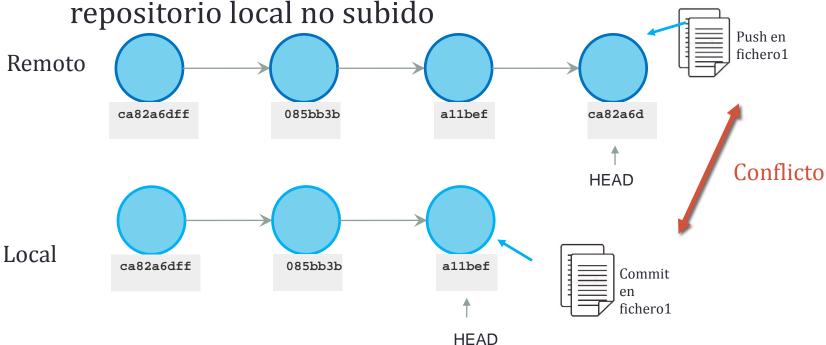




# Gestionar conflictos en el código

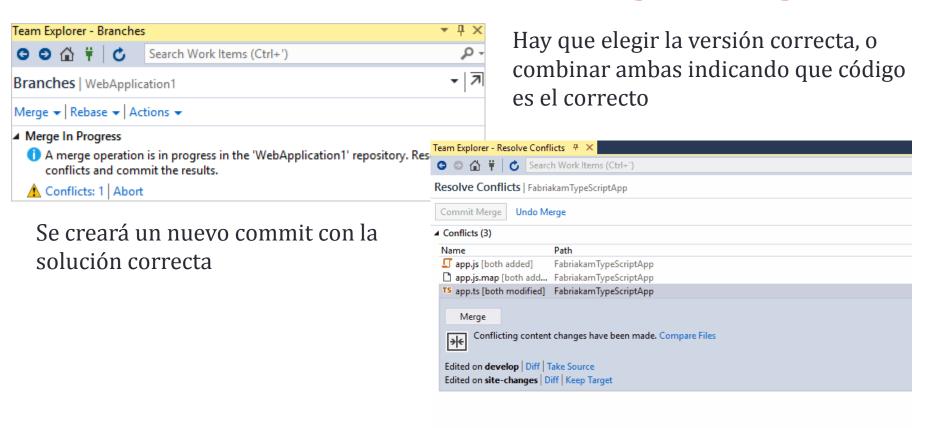
 Cuando dos miembros del equipo trabajan sobre el mismo fichero

• Un nuevo push del repositorio remoto contiene modificaciones en el mismo fichero que un commit del repositorio local no subido



50

#### Gestionar conflictos en el código: merge



## Flujo de trabajo con Git

- Realizar un pull en tu rama: trabajar sobre la última versión
- Trabajar en tus tareas
- Realizar un commit cada vez que se complete algún punto importante de tu tarea
- Cuando tu proyecto esté estable, sincroniza tus cambios (no propagar errores entre miembros del equipo)
- Resolver conflictos: si realizáis una buena división del trabajo, no se darán

#### Conclusiones

- Azure DevOps nos permite gestionar los proyectos de nuestra organización y sus equipos de trabajo,
- Azure DevOps nos da acceso a toda la información de nuestro proyecto para conocer su evolución, conocer el trabajo realizado por los miembros, estadísticas, etc,
- Visual Studio complementa el plan del trabajo diseñado con Azure DevOps.
- Visual Studio nos permite asociar el código y los cambios a las tareas definidas en el plan.
- Visual Studio permite obtener, versionar el código y gestionar conflictos fácilmente – control de versiones transparente para el grupo de trabajo, utilizado un servidor en la nube gratuito

#### Recursos de aprendizaje

- Visual Studio Walkthroughs (English)
   <a href="https://msdn.microsoft.com/es-es/library/szatc41e(v=vs.110).aspx">https://msdn.microsoft.com/es-es/library/szatc41e(v=vs.110).aspx</a>
- 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 <u>Azure Repos</u>. Edward Thomson. Microsoft Visual Studio

