

Cloud application development

with Microsoft Azure

Hands-On Lab | Step-by-Step Guide

Lab 8 – Configuración de integración continua en Visual Studio Team Services

Lab version: 1.0.0

Last updated: September 18, 2016

Important: *An empty Azure Trial subscription is recommended to complete the following exercises. In case your subscription contains previous created elements, you should take care that any of the following steps affect them.*

Contenido

| | |
|--|-----------|
| CONTENIDO | 2 |
| LAB 8: CONFIGURACIÓN DE INTEGRACIÓN CONTINUA EN VISUAL STUDIO TEAM SERVICES..... | 3 |
| EJERCICIO 8.1: CREACIÓN DE UNA DEFINICIÓN DE BUILD PARA INTEGRACIÓN CONTINUA..... | 3 |
| EJERCICIO 8.2: EJECUCIÓN DE UNA DEFINICIÓN DE BUILD DE INTEGRACIÓN CONTINUA..... | 11 |

- Copyright © 2014 by Microsoft Corporation. All rights reserved.

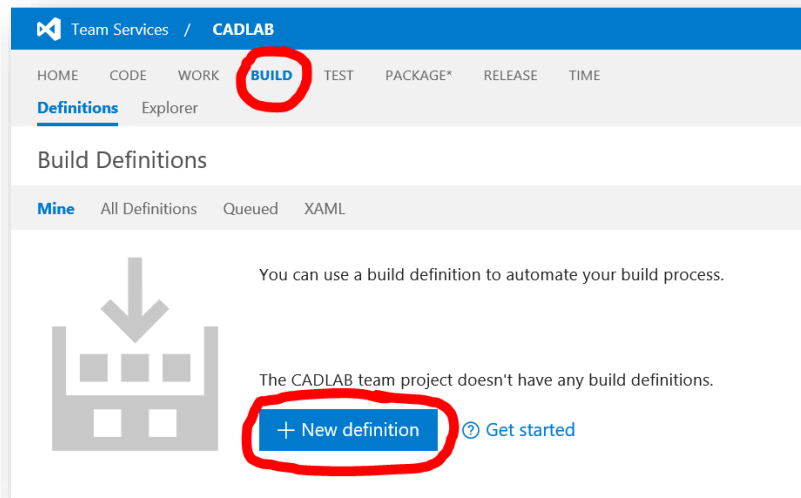
Lab 8: Configuración de integración continua en Visual Studio Team Services

Ejercicio 8.1: Creación de una definición de build para integración continua

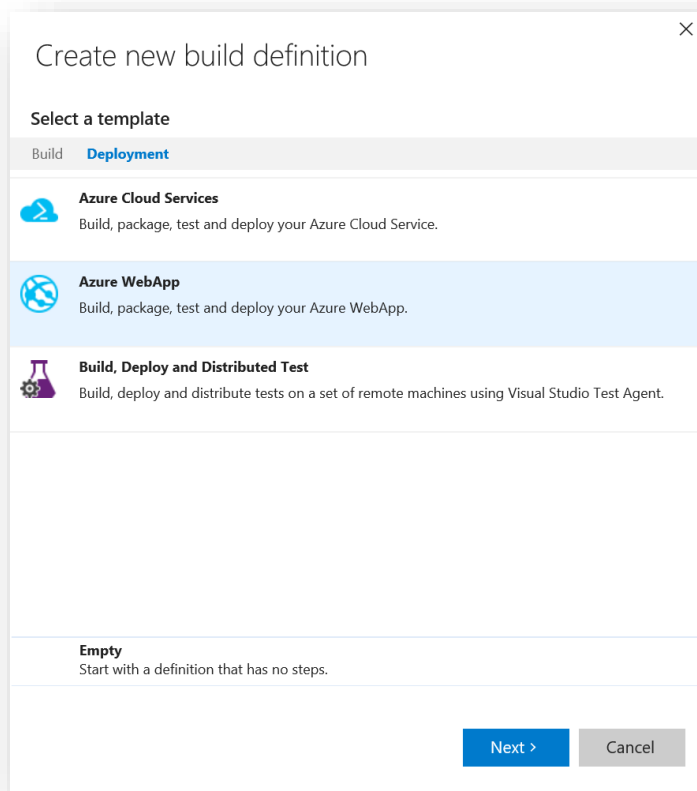
Vamos a crear un proceso de compilación (Build) de nuestra webapp, utilizando nuestro proyecto en equipo de la cuenta Team Services. Hacer este tipo de procesos nos garantiza:

- Que la compilación toma el código fuente más actualizado del repositorio Git
- Que la compilación no depende de la máquina del desarrollador
- Que el proceso se puede extender para realizar otras acciones como pruebas unitarias, cobertura de código, etc., para darnos una idea de la calidad de nuestra aplicación
- Que el proceso es automático y que se ejecutará igual cuantas veces queramos, con lo que nos salvamos la propensión al error que supone la ejecución de acciones manualmente

1. Dentro de nuestro proyecto en equipo de la cuenta Team Server **CADLAB**, acceder al apartado **BUILD** y crear un nuevo proceso de compilación **+New definition**.



2. Seleccionar la plantilla de despliegue (**Deployment**) llamada **Azure WebApp**. Las plantillas de definición de builds ya vienen construidas con una serie de pasos que podemos reaprovechar para construir nuestros proyectos (la otra opción es empezar con una plantilla de definición de build vacía).



3. Seleccionar como repositorio de código el propio del proyecto en equipo con el que estamos trabajando **CADLAB** y marcar el **check de integración continua**; queremos que cada vez que sincronizamos cambios de código en el servidor se lance automáticamente una compilación (build).
Dejar por defecto el agente de compilación que nos propone **Hosted**; el agente de compilación es el servicio que se encargará de realizar la compilación. Team Services proporciona agentes hospedados en Azure, aunque también es posible instalar agentes privadas para realizar esta labor.
Confirmar la creación de esta definición de build.

Create new build definition

Settings

Repository source

CADLAB Team Project GitHub Remote Git Repository Subversion

Repository

CADLAB

Default branch

master

☒ Continuous integration (build whenever this branch is updated)

Default agent queue | [manage queues](#)

Hosted

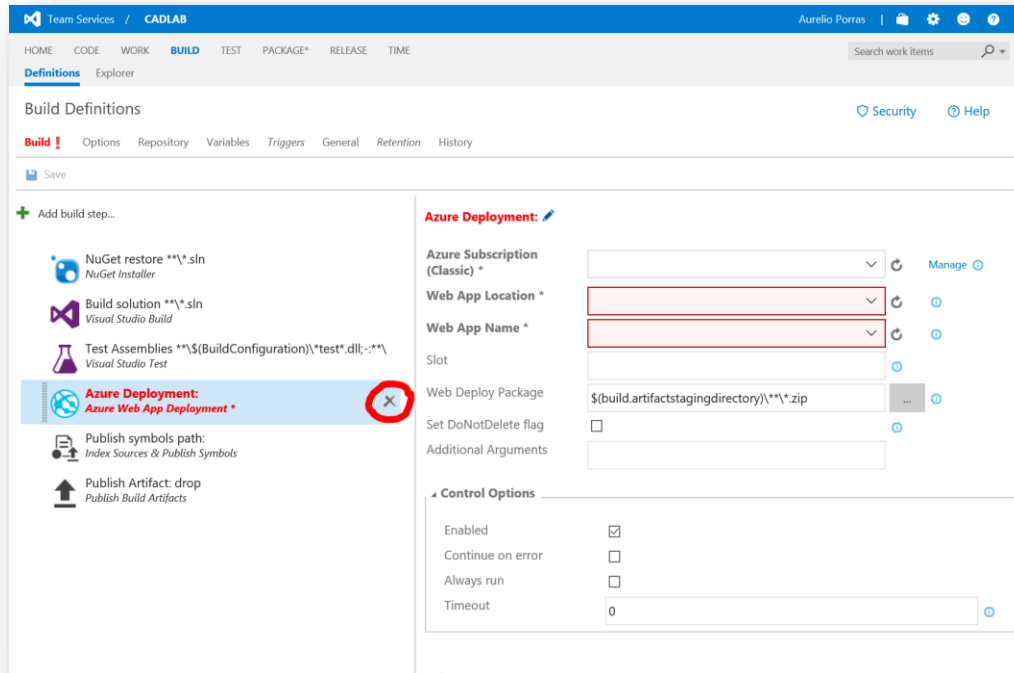
Select folder

\ Choose folder...

< Previous Create Cancel

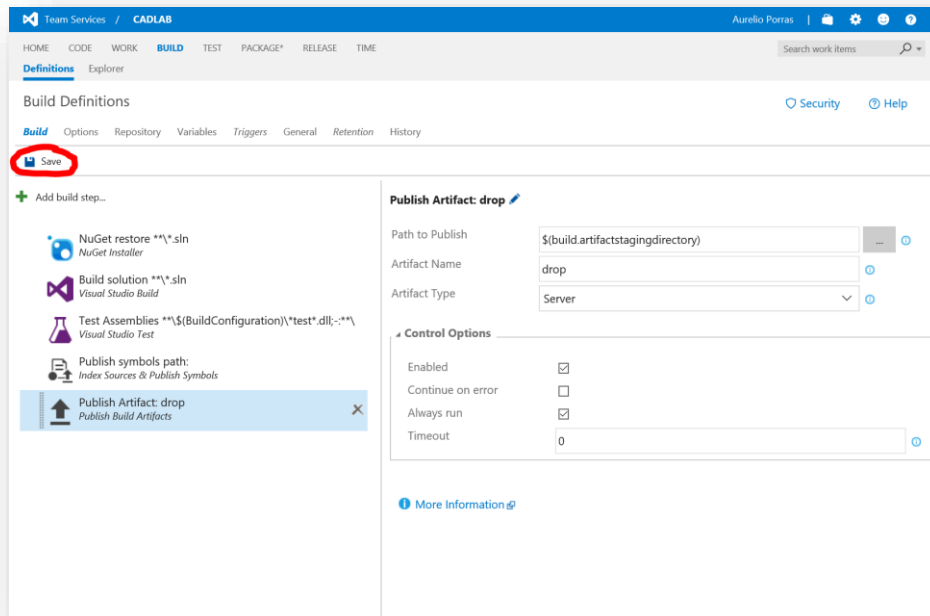
4. Veremos entonces cuáles son las acciones que realizará esta definición de build:
- Bajarse los paquetes de NuGet necesarios, en función de las dependencias de la solución Visual Studio
 - Compilar la solución Visual Studio, con los parámetros para empaquetar la webapp para su posterior despliegue
 - Ejecutar las pruebas que se encuentren en la solución Visual Studio
 - Desplegar la webapp en Azure
 - Publicar los símbolos para depurar la solución Visual Studio
 - Publicar el resultado de la compilación, que constituyen los artefactos para el despliegue, en el repositorio

Borraremos el paso de desplegar la webapp en Azure, para hacerla más adelante, al definir el proceso de despliegue. De esta forma, la definición de build sólo compilará la solución, empaquetará la webapp y se publicarán los artefactos necesarios para hacer el despliegue. Así separamos lo que es la construcción de artefactos del despliegue de los artefactos, y podremos definir diferentes procesos de despliegue sin afectar a la forma de construir los artefactos y sin tener que recompilar cada vez que desplaguemos los mismos artefactos a diferentes entornos.



No cambies nada, pero echa un vistazo a todas las opciones de la definición de build; y echa un vistazo también a cómo se personalizar la definición de build configurando cada paso, añadiendo más pasos, cambiando el orden de los pasos, etc.

5. Salvar la definición de build



6. Pon el nombre **Build CI** a la definición de build, como proceso de compilación de integración continua, añade un comentario para el control de cambios de la definición de build y pulsa **OK**

×

Save

Name

Build CI

Comment

Build CI created

Select folder

\

Choose folder...

OK

Cancel

- Ahora probemos la definición de build, encolando una nueva build de forma manual, aunque esta definición de build se ejecutará cada vez que haya un cambio en el repositorio de código.

←

→

↺

🔒

aurelio:visualstudio.com/CADLAB/_build/definitionEditor?definitionId=13&a=simple-process

📖

☆

≡

🔗

🔔

⋮

Team Services / CADLAB Aurelio Porras | ⚙️ 🔔 🔍

HOME CODE WORK BUILD TEST PACKAGE* RELEASE TIME

Definitions Explorer

Build Definitions / Build CI

Summary Queue new build... Security Help

Build Options Repository Variables Triggers General Retention History

Save Undo

+

Add build step...

NuGet restore ***.sln

NuGet Installer

Build solution ***.sln

Visual Studio Build

Test Assemblies **\\$(BuildConfiguration)*test*.dll;-*\

Visual Studio Test

Publish symbols path:

Index Sources & Publish Symbols

Copy Files to: \$(build.artifactstagingdirectory)

Copy Files

Publish Artifact: drop

Publish Build Artifacts

NuGet restore ***.sln

Path to solution or packages.config

***.sln

Path to NuGet.config

Installation type

☒ Restore
 ☐ Install

Disable local cache

☐

NuGet arguments

Advanced

Control Options

Enabled

☒

Continue on error

☐

Always run

☐

Timeout

0

More Information

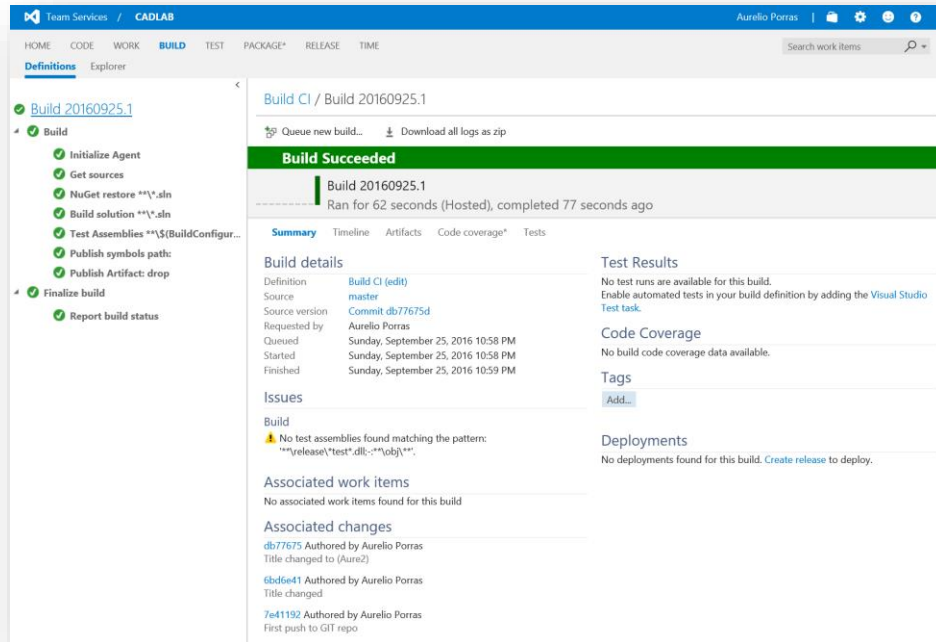
- Team Services nos da la opción de cambiar parámetros de la definición de build cuando realizamos una ejecución manual. Dejamos los parámetros por defecto y confirmamos.

- Esto provoca que el proceso de build se lance y comience la ejecución de los pasos. En todo momento, Team Services muestra qué es lo que está haciendo el agente de compilación, a través de una consola.

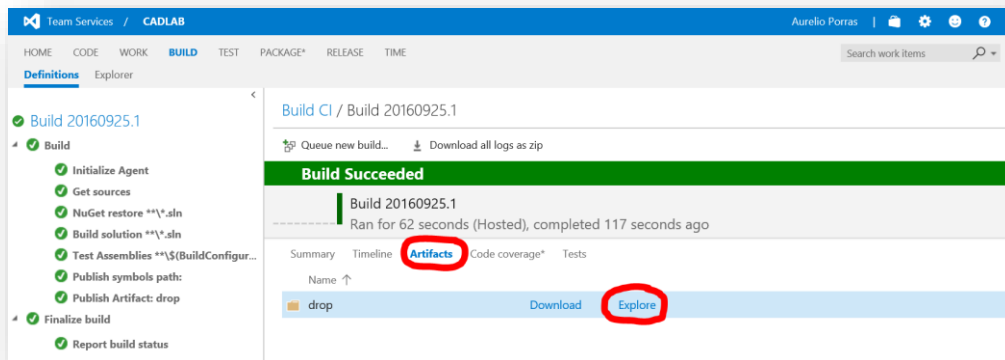
```

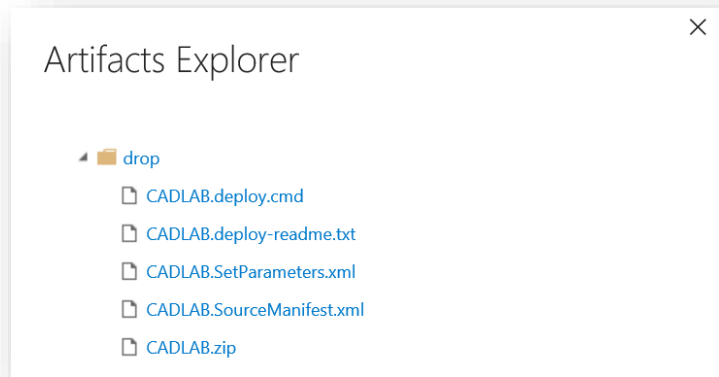
Checking files to be uploaded took 173 milliseconds for 1 files.
Uploading file 'C:\a\1\1\CADLAB.SetParameters.xml' with size 414 (bytes)
Uploading compressed file 'C:\a\1\1\CADLAB.deploy.cmd' with size 2962 (bytes)
File 'C:\a\1\1\CADLAB.SetParameters.xml' took 0.1015936 seconds to upload, upload size 414 (bytes), average upload speed 3.9
88 (KB/s)
Uploading file 'C:\a\1\1\CADLAB.SourceManifest.xml' with size 381 (bytes)
File 'C:\a\1\1\CADLAB.deploy.cmd' took 0.1096646 seconds to upload, upload size 2962 (bytes), average upload speed 17.049 (K
B/s)
File 'C:\a\1\1\CADLAB.SourceManifest.xml' took 0.1181868 seconds to upload, upload size 381 (bytes), average upload speed 3.
148 (KB/s)
Checking files to be uploaded, 1 files will be check at this time.
Checking files to be uploaded took 91 milliseconds for 1 files.
Uploading compressed file 'C:\a\1\1\CADLAB.zip' with size 9250663 (bytes)
Files found locally 5,
Files evaluated 5,
Files left to evaluate 0,,
Files created without upload 0,
Files uploaded 4,
Files left to process 1
-----
File 'C:\a\1\1\CADLAB.zip' took 1.8471626 seconds to upload, upload size 9250663 (bytes), average upload speed 4,000.663 (KB
/s)
Created 0 files without uploading content. Total files processed 5
Uploaded artifact 'C:\a\1\1' to container folder 'drop' of build 76.
Associated artifact 00 with build 76
=====
Finishing task: PublishBuildArtifacts
=====
Finishing Build
=====
Worker Worker-e0b63b1d-1e86-48e4-80c5-34573c0215a9 finished running job e0b63b1d-1e86-48e4-80c5-34573c0215a9
    
```

- Una vez finalizada la build podemos ver el sumario de lo que ha realizado



11. Finalmente, podemos también consultar los **artefactos** de despliegue de la webapp, que usaremos posteriormente.

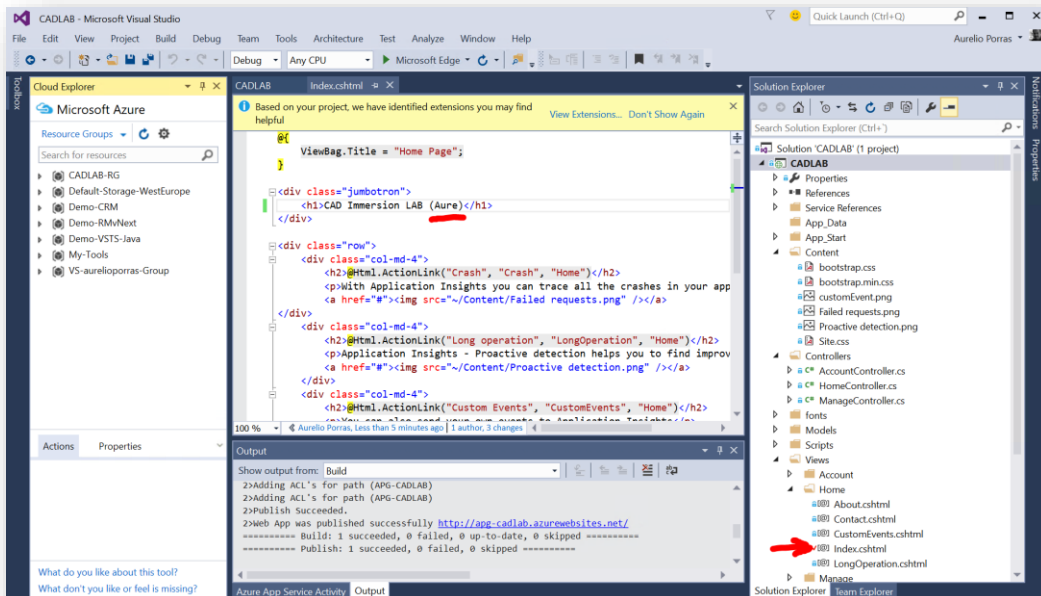
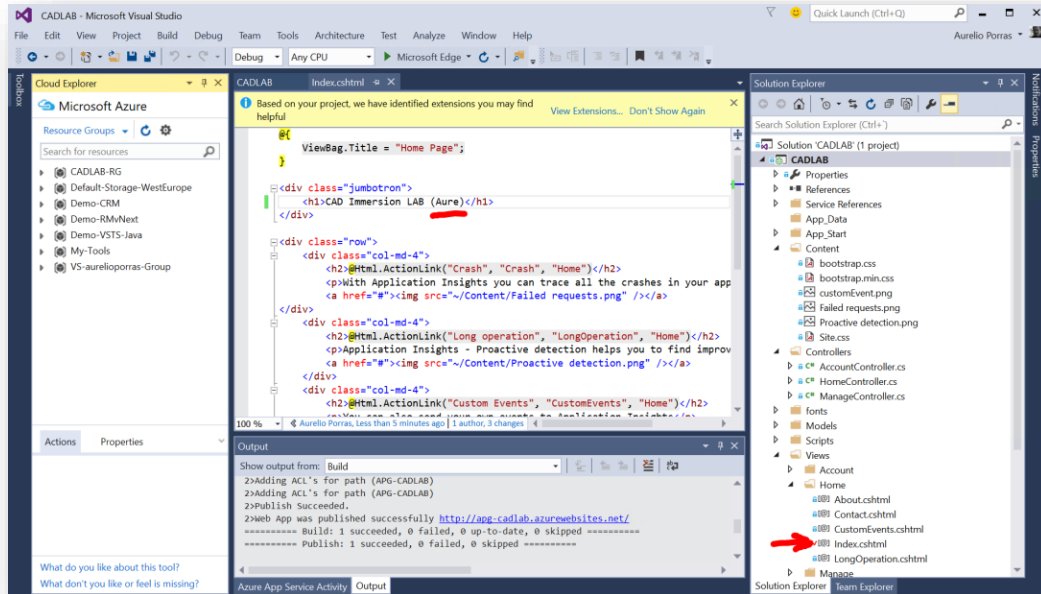




Ejercicio 8.2: Ejecución de una definición de build de integración continua

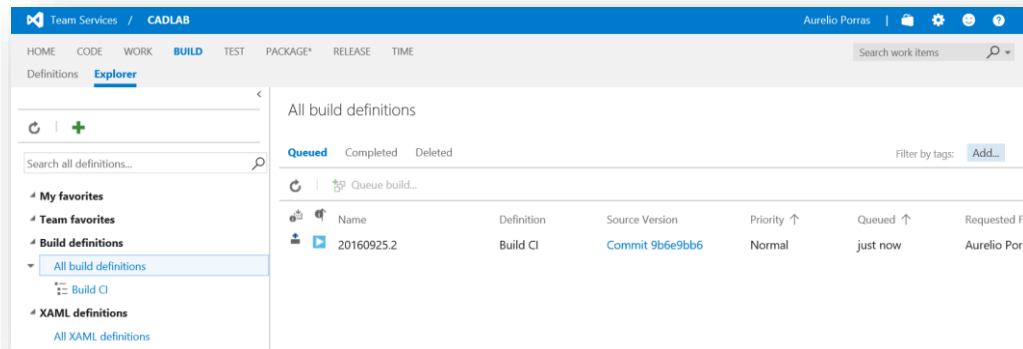
Una vez probada manualmente la definición de build, podemos probar cómo se ejecuta cada vez que se produzca un cambio en el repositorio de código fuente.

1. Desde Visual Studio introduce un cambio en el código fuente; por ejemplo, cambia el título en la página de inicio **CADLAB/Views/Home/index.chtml**, añadiendo tu nombre al final. Sube los cambios al repositorio del servidor con un comentario que recuerdes.



- Desde el navegador con la cuenta Team Services, explora las definiciones de build que están encoladas, y verás que automáticamente al subir el cambio al control de código se han encolado una nueva build.

Puedes consultar su ejecución en directo pulsando en el nombre de la build encolada.



- Una vez completada la build, puedes consultar toda su ejecución en la lista de builds completadas. El símbolo del muñequito indica que la build se lanzó automáticamente a partir de un cambio en el código.

Puedes consultar directamente el cambio en el código a través del enlace al control de versiones, y verás el comentario que pusiste al subir el cambio de código fuente.

