

Guía Práctica BitBucket

Primera Publicación de Fuentes

Empresa: CMPC



Autores: Oscar Salgado

Juan Venegas

Luis Sandoval

Versión del documento: 1.0

Fecha Primera Versión: 30/03/2021

Contenido

1	Introducción	2
2	Primera Publicación de Fuentes	3
	Requisitos	3
	PASO 1. Clonar Repositorio Existente en BitBucket.	3
	PASO 2. Subir fuentes a repositorio clonado (Rama Master).....	8
	PASO 3. Crear Rama Develop	13

1 Introducción

El presente documento es una guía práctica de la primera publicación de fuentes en bitBucket de CMPC. Este documento no busca ser exhaustivo, sino tan solo guiar en la primera publicación permitiendo dejar creadas la rama master y la rama develop de cada aplicativo, bajo la utilización de herramientas gráficas.

Posterior a la subida de los fuentes, el sistema se deberá comenzar a utilizar para todos los incidentes y requerimientos. Esto se llevará a cabo a través del uso de ramas hotfix y feature, dependiendo de si se trabajará con un incidente o un requerimiento, respectivamente.

La subida de fuentes al repositorio después de resolver un incidente o requerimiento, deberá realizarse a la rama develop, sólo se deberá realizar a la master, previo al paso a producción.

Es muy importante que quien suba los fuentes la primera vez, se asegure de que cuenta con la última versión estable de los códigos, dado que con esta versión se cargará la rama master y develop, en caso de existir un desarrollo en curso, estos se deberán cargar en la rama hotfix o feature respectiva, estas ramas deberán ser creadas a través de bitbucket.

Los pasos descritos en este documento utilizan BitBucket, GitHub desktop y para los proyectos visual studio, se describen los pasos en esta herramienta, cabe destacar que la utilización de una u otra herramienta está en relación a las opciones que esta proporciona buscando la mayor simplicidad de la tarea, destacando que la expuesta en este documento no es la única forma de realizar estas actividades.

Este documento fue elaborado con la colaboración de:

Jorge Gonzales

Juan Venegas

Oscar Salgado

Andres Herrera

2 Primera Publicación de Fuentes

Este documento contiene una Guía práctica para la primera subida de archivos fuentes a los repositorios CMPC en bitbucket, estos repositorios estarán previamente creados en Bitbucket por los Jefes de Proyecto CMPC encargados de cada aplicación.

Requisitos

1. Disponer de una cuenta en BitBucket de CMPC.
2. Tener instalado en PC GitHub Desktop, este se puede descargar desde la siguiente URL: <https://desktop.github.com/>
3. Estar creado el repositorio del sistema que se requiere cargar en bitBucket.

PASO 1. Clonar Repositorio Existente en BitBucket.

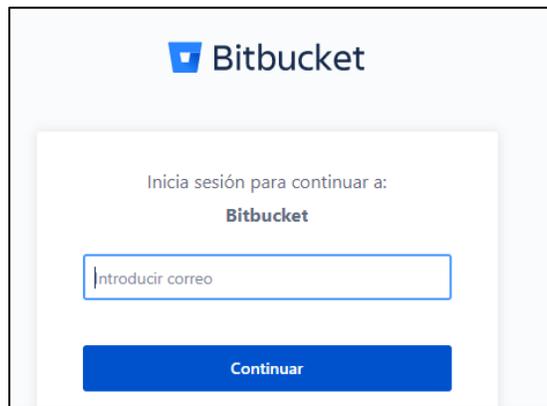
El primer paso, que permita subir los archivos fuentes a la rama master ya existente en BitBucket, es realizar una clonación del repositorio hacia tu repositorio local.

La clonación del repositorio se realizará directamente desde BitBucket.

Para acceder a BitBucket, se debe realizar a través de la siguiente URL:

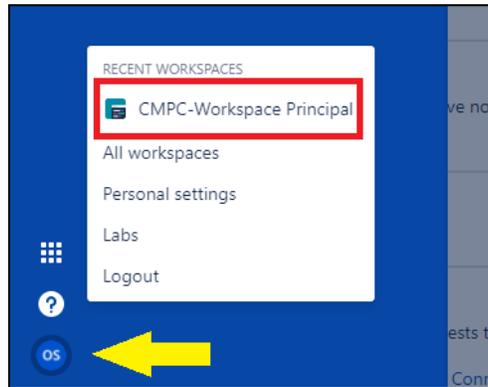
<https://bitbucket.org/dashboard/overview>

PASO 1.1. Ingresar a BitBucket (Correo y Contraseña).

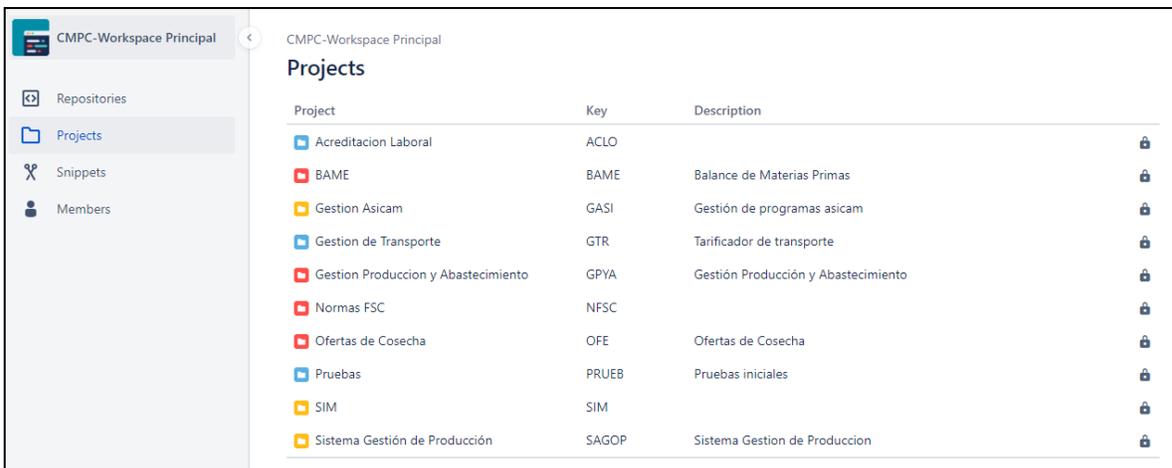


The image shows a screenshot of the Bitbucket login interface. At the top, the Bitbucket logo is displayed. Below it, the text reads "Inicia sesión para continuar a: Bitbucket". Underneath, there is a text input field with the placeholder text "Introducir correo". At the bottom of the form is a blue button labeled "Continuar".

PASO 1.2. Seleccionar el Workspace “CMPC-Workspace Principal” en el menú izquierdo (parte inferior)



PASO 1.3. En la siguiente pantalla seleccionar “Projects” y elegir el sistema que se necesita subir.

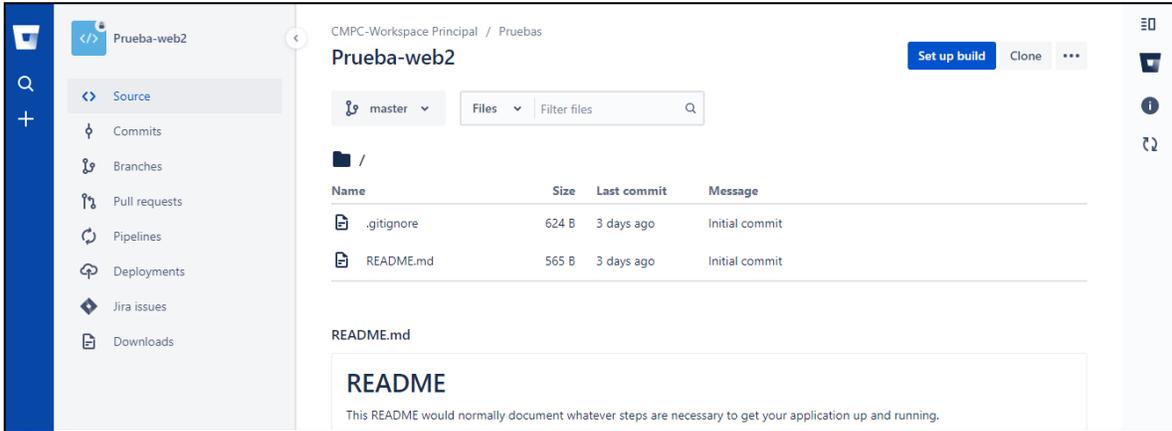


PASO 1.4. Seleccionar el repositorio que corresponde a los archivos fuentes que serán publicados.

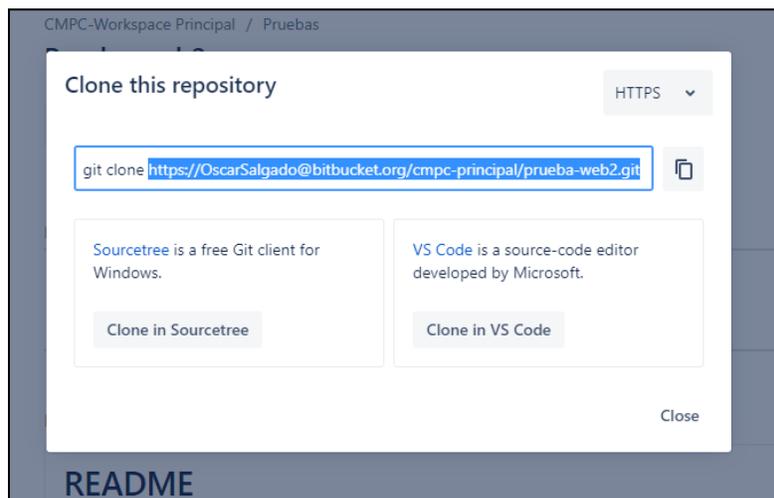
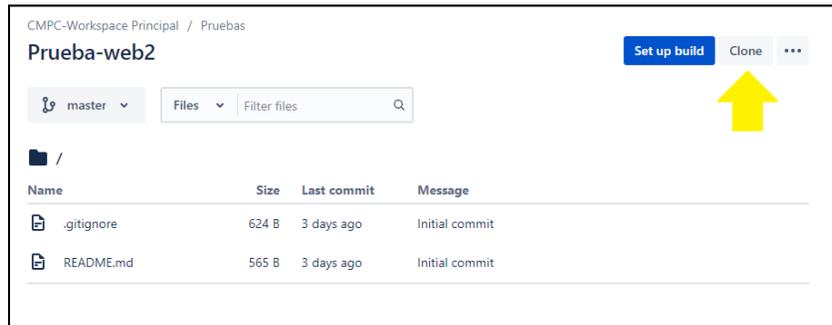
Para el ejemplo se ingresa a repositorio “Prueba-web2”



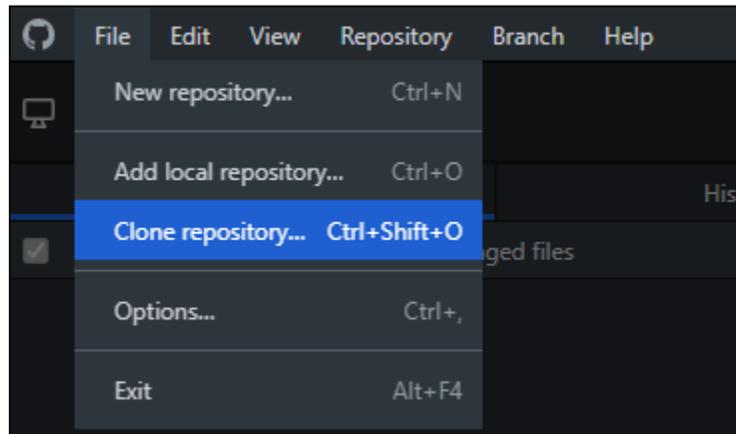
PASO 1.5. Al Ingresar al repositorio se verá algo como lo siguiente:



PASO 1. 6. Luego se debe hacer clic en el botón “Clone” (en la esquina superior derecha), a continuación, se abrirá una ventana pop-up de la cual se debe copiar la URL (sólo la URL sin comandos GIT, es decir desde HTTPS en adelante).



PASO 1.7. Abrir GITHUB DESKTOP e ir al menú FILE → Clone Repository

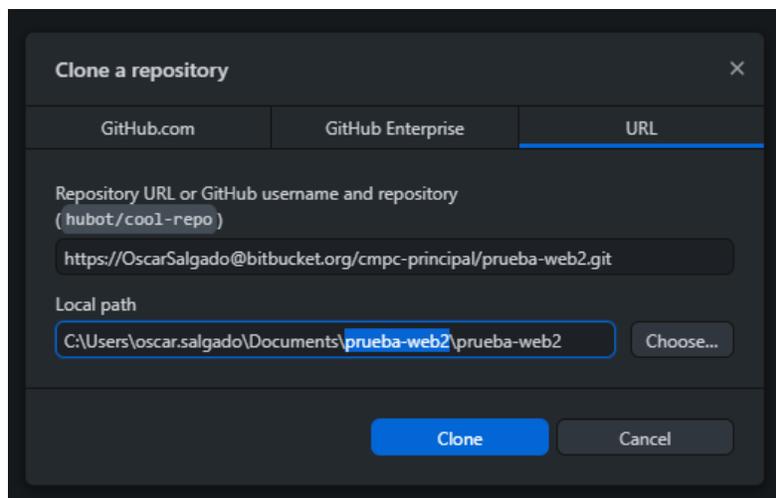


PASO 1.8. Seleccionar la opción “URL” en la parte superior derecha y pegar la url que se acaba de copiar desde BitBucket, a continuación, definir la ruta Local en que se dejará el repositorio.

Esta carpeta es importante dado que será la ruta donde se trabajará en el futuro.

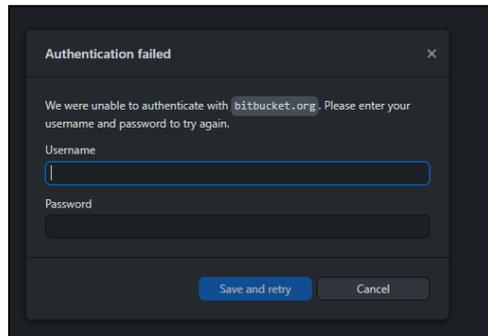
Nota: La carpeta que contendrá el repositorio (carpeta destino), **debe estar vacía**, de lo contrario, GitHub generará conflictos. Se recomienda crear una nueva carpeta con el nombre del proyecto bitbucket a clonar.

Presionar el botón “Clone”, con esto la rama quedará en la carpeta especificada.



Con esta acción ya tendremos clonado el repositorio localmente.

De ser necesario, deberá ingresar sus credenciales de Bitbucket en Github Desktop:



Posterior a esto, es posible ir al explorador de archivos de Windows y confirmar que la rama ya existe en carpeta local; La rama contendrá los siguientes archivos (2):

- a) `.gitignore` : archivo editable en texto plano, en el cual se especificarán todas las rutas y archivos que no se requieran y con ello, el proceso de control de versiones simplemente ignorará esos archivos.
- b) `README.md`: Debe contener un resumen del proyecto, las rutas de las carpetas, su definición y cualquier otro detalle que pueda aportar información a los archivos fuentes.

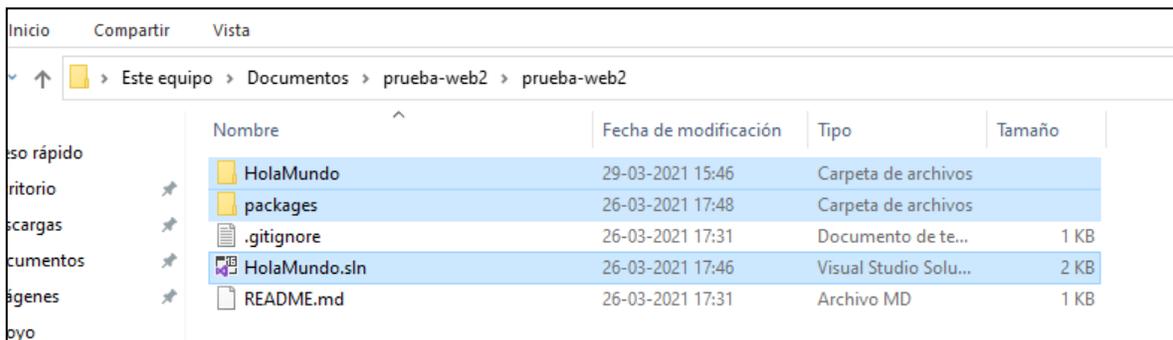


Nota: Considerar que estos archivos son creados por defecto al crear un repositorio en bitbucket, por lo tanto, para el caso de `“.gitignore”` es recomendable borrarlo y crearlo nuevamente con la configuración respectiva de la aplicación.

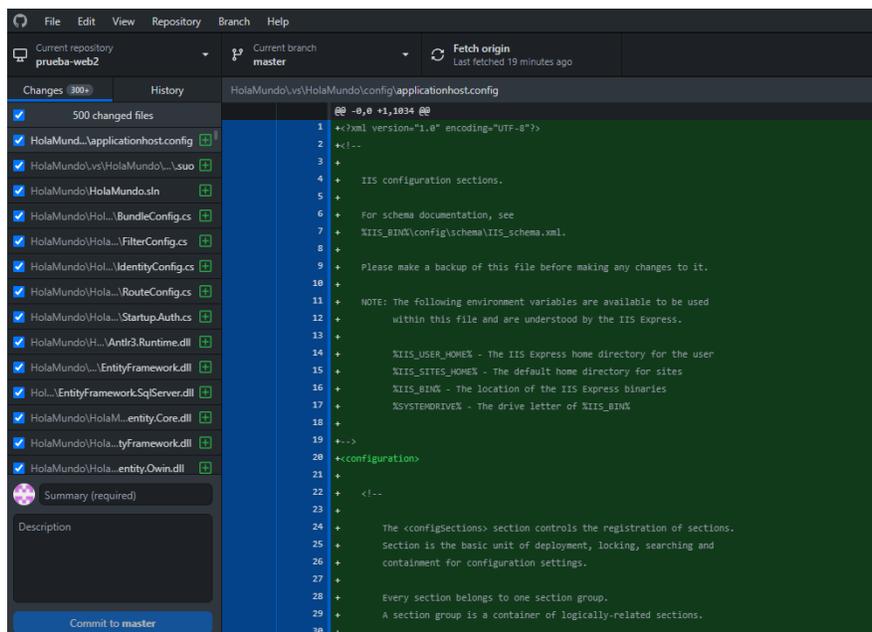
Para detalles de cómo realizar esto en Visual Studio, ir al paso **2.3**.

PASO 2. Subir fuentes a repositorio clonado (Rama Master)

Paso 2.1. Copiar los códigos fuentes del proyecto (asegurando que sea la última versión de producción), a la carpeta local definida en el Paso 1.8. Para este ejemplo se copia el proyecto “HolaMundo”:

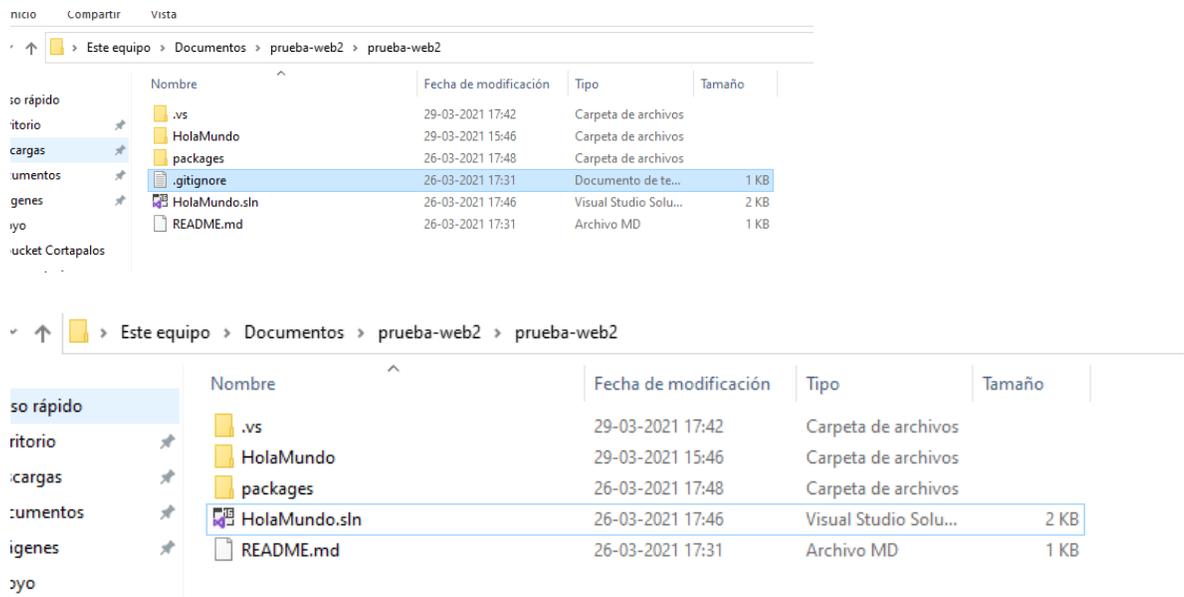


Paso 2.2. Una vez copiados los códigos fuentes, ir nuevamente a GitHub Desktop, en el podremos observar que ya detectó que hemos copiado el proyecto al repositorio, mostrando una lista de cambios.



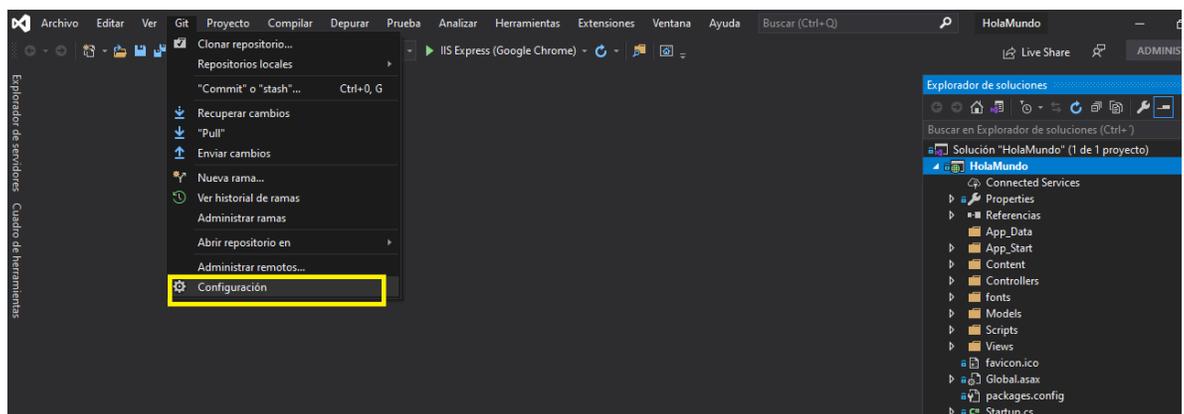
Paso 2.3. IMPORTANTE: Por ser la primera publicación del sistema, es importante que el archivo “.gitignore” que viene creado por defecto desde bitbucket, sea creado de acuerdo a las características de su proyecto. Si su proyecto es un proyecto Visual Studio, le recomendamos eliminarlo y crearlo nuevamente a través de “Visual Studio 2019” (sólo si su proyecto es un proyecto visual studio). Una forma de realizar esto es:

2.3.1. Eliminar el archivo “.gitignore” desde el repositorio local.



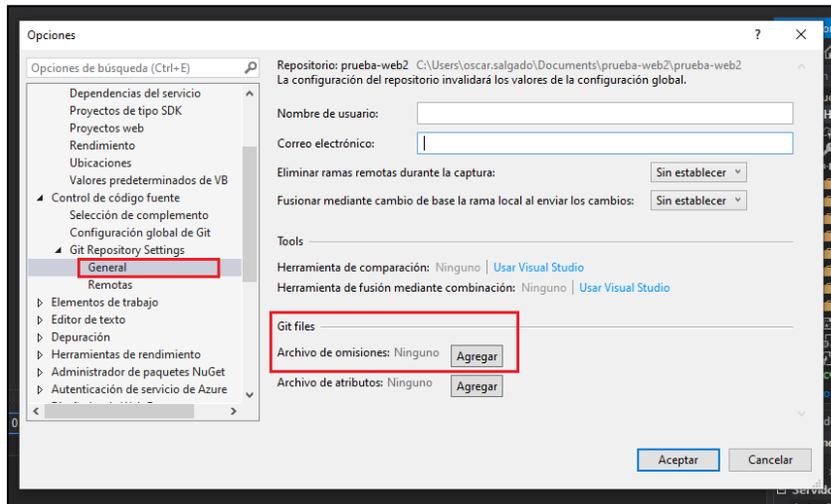
2.3.2. Abrir el proyecto solución o Carpeta en Visual Studio 2019.

En el menú superior seleccionar **GIT** → **Configuración**

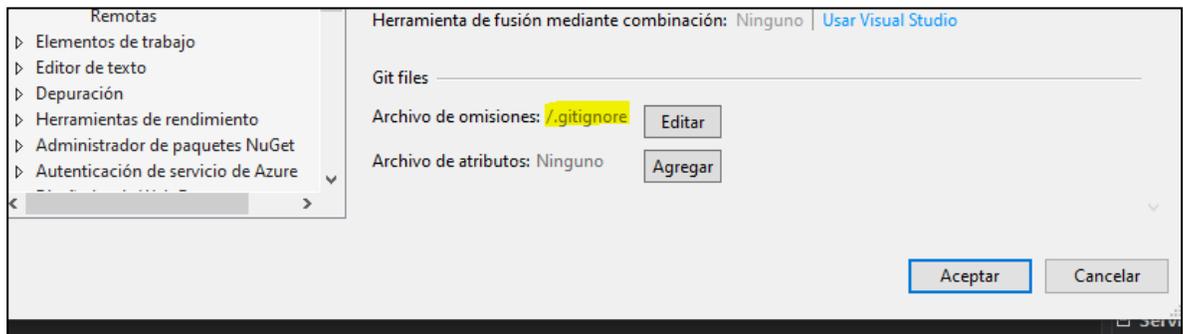


Luego en el panel izquierdo seleccionar:

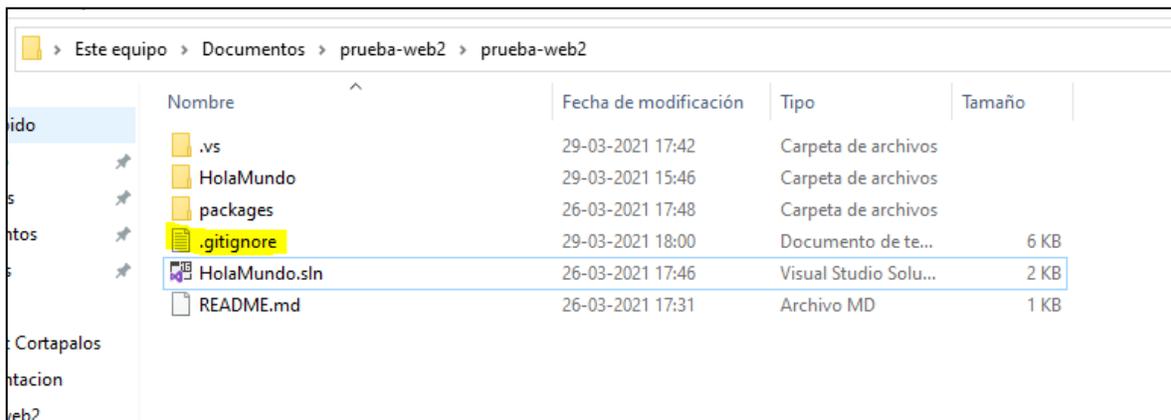
“Control de Código Fuente” → **“Git Repository Settings”** → **“General”** y en la sección derecha **“Git Files”** → **“Archivo de omisiones”** dar click a botón **“Agregar”**, Tal como se indica en la imagen:



Luego de hacer click en agregar, se puede apreciar que archivo **“.gitignore”** ha sido creado.

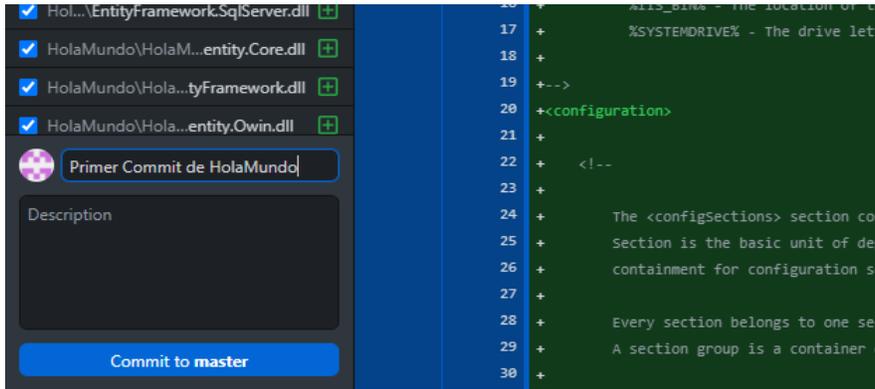


Se valida también en carpeta local:

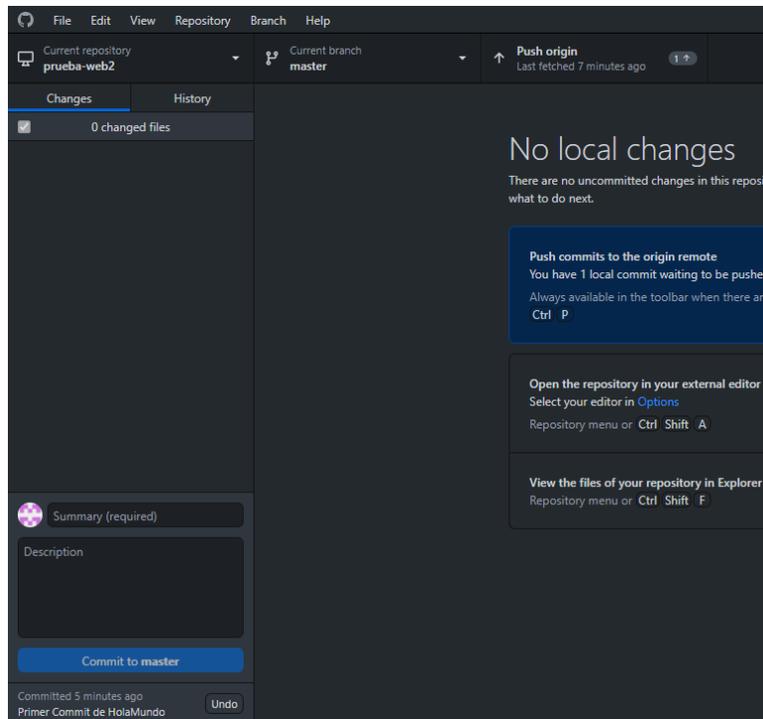


Paso 2.4. A continuación, es necesario confirmar los cambios de manera local, para esto en la esquina inferior izquierda de Github Desktop (donde dice "Summary" y "Description"), escribir un mensaje de confirmación y luego haga clic en "Commit to Master".

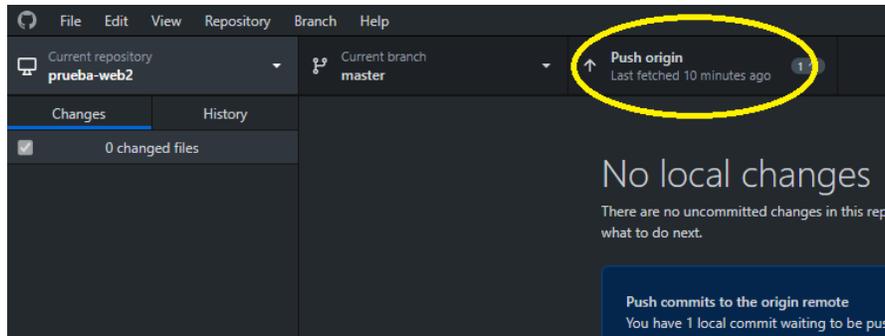
El campo "Summary" es obligatorio.



Paso 2.5. Una vez realizado este primer commit, el panel izquierdo ya no muestra la lista de cambios, por lo tanto, dichos cambios han sido confirmados localmente. (Aún tienen que enviarse al repositorio de origen en bitbucket).



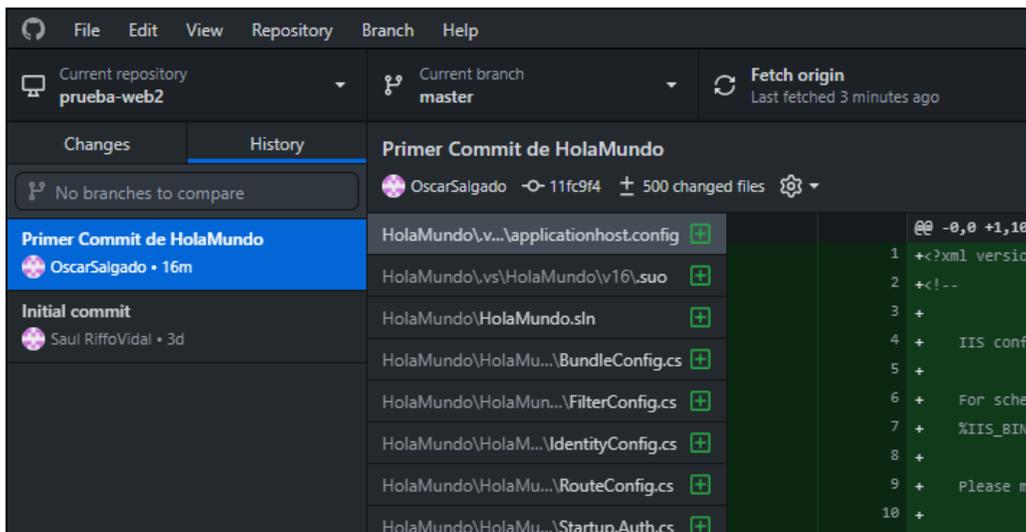
Paso 2.6. En el menú superior, hacer click en el botón “Push Origin” para subir los cambios a repositorio origen en bitbucket.



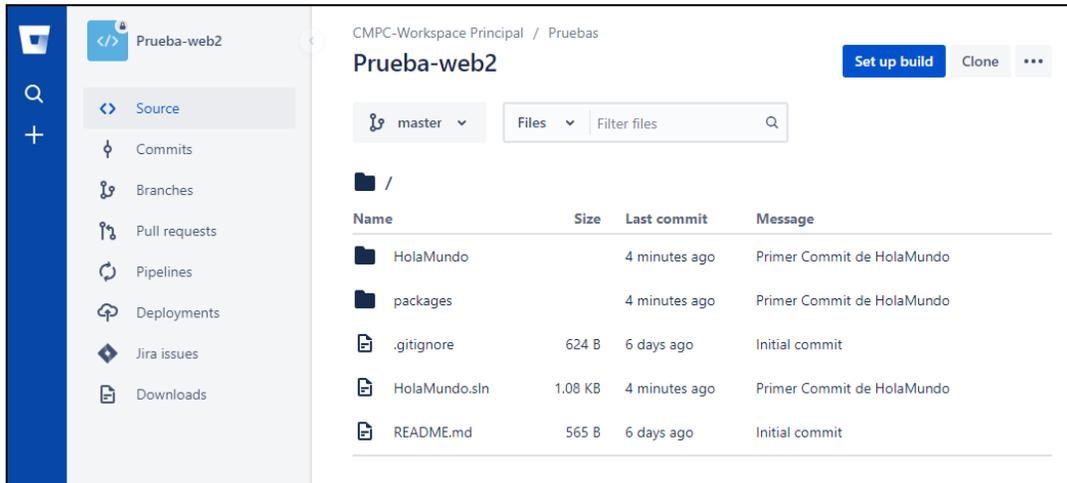
Nota: No olvidar que nunca debemos trabajar nuestros desarrollos en la rama master, esta refleja los fuentes productivos, solo estamos trabajando con la rama master, dado que es la primera publicación de fuentes.

En este punto tenemos creada y cargada la rama master con nuestro sistema, esto puede verificarse en el panel izquierdo, pestaña “History” y/o directamente en **bitbucket**.

En GitHub Desktop:



En Bitbucket:

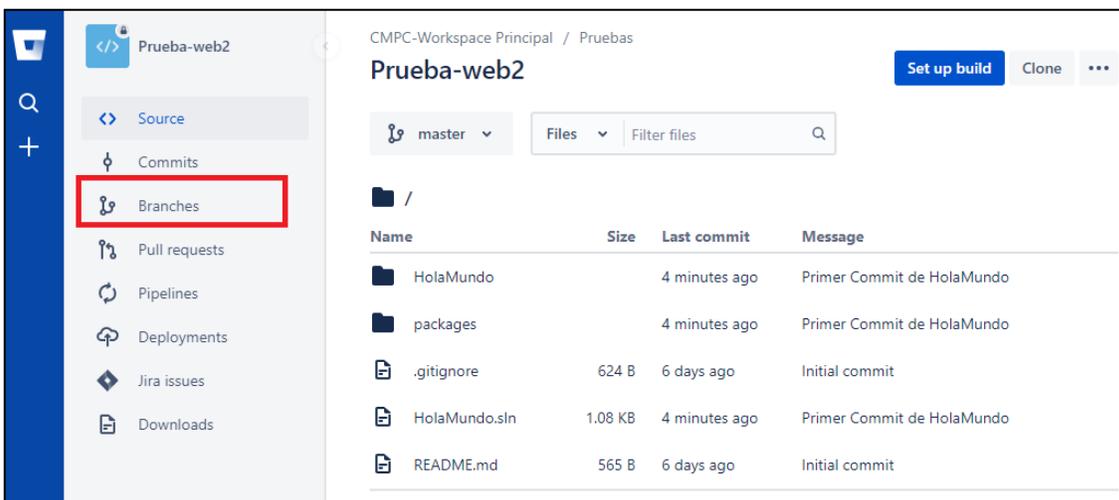


PASO 3. Crear Rama Develop

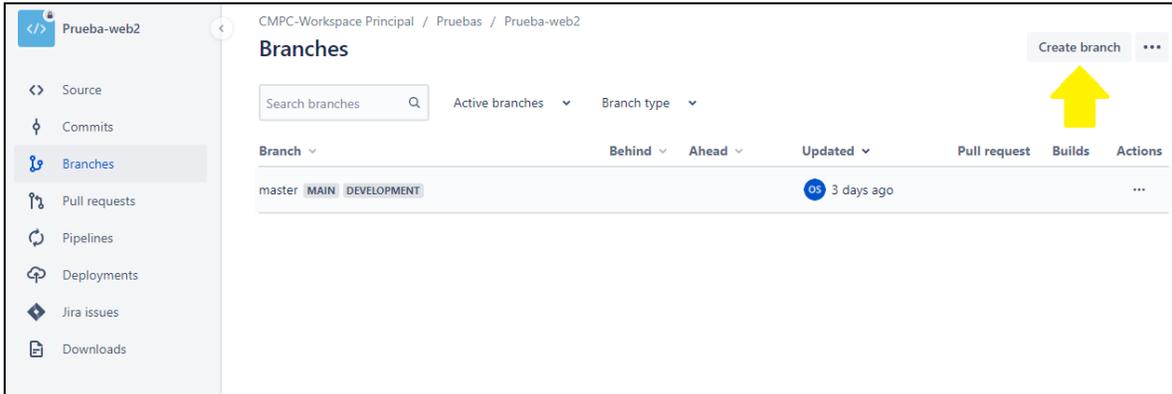
Una vez que ya tenemos la rama master con los códigos fuentes publicados, crearemos la rama develop, que será la que utilizaremos como origen de nuestras ramas de desarrollo.

NOTA: Existen varias formas de crear una nueva rama, pero por definición y acuerdos previos, se realizará SÓLO a través de BitBucket.

Paso 3.1. Situados en el repositorio del proyecto seleccionamos en el menú izquierdo la opción “Branches”:



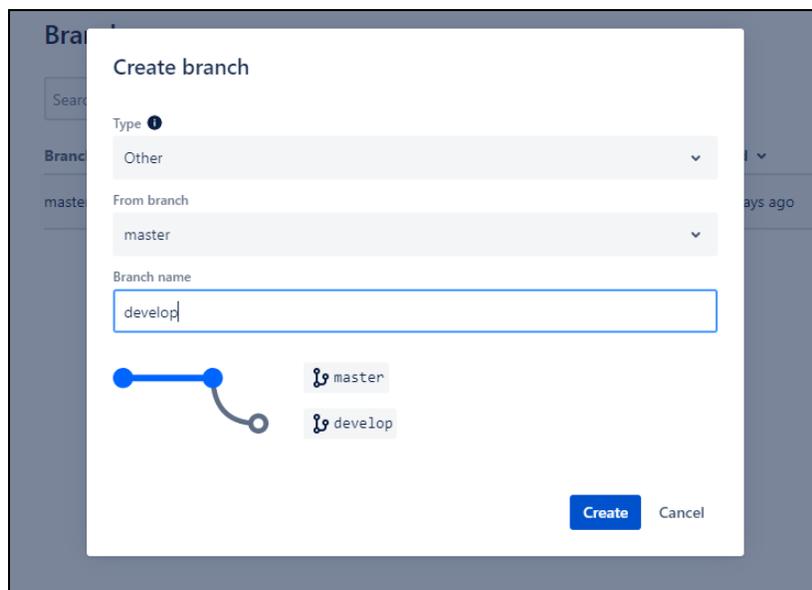
Paso 3.2. A continuación, se mostrará una pantalla con la lista de ramas creadas para este repositorio, se debe hacer click en el botón “Create Branch” en la esquina superior derecha para crear una nueva rama:



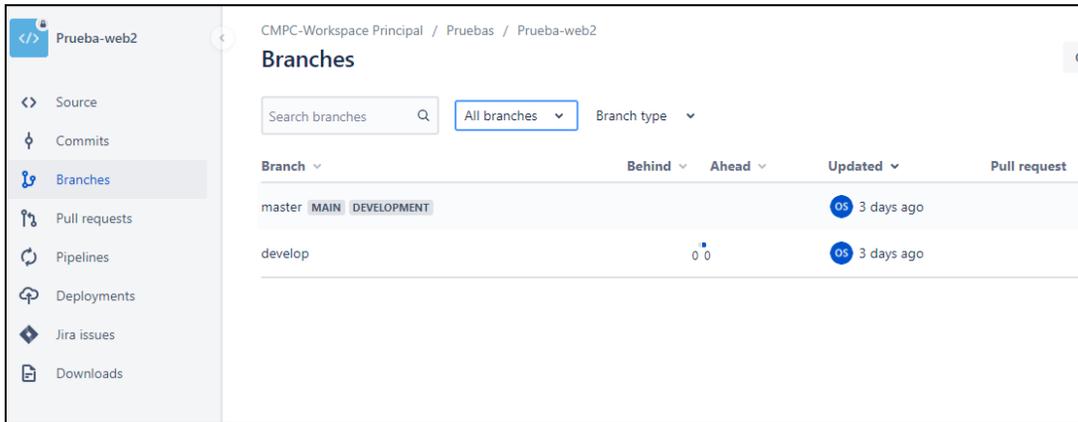
Paso 3.3. Se abrirá una ventana pop-up tal como se muestra en la imagen, se debe seleccionar:

- Tipo de rama: **“Other”**
- Rama a partir de la cual se creará la nueva: **“Master”**
- Nombre de la rama: **“develop”** con todas sus letras en minúscula por convención de nombres.

Hacer click en botón **“Create”** como se muestra en la imagen:



A continuación, podemos ver que la rama ha sido creada:



Si vamos al menú izquierdo “Source”, podemos ver que la nueva rama “develop” ha sido creada:



Si hacemos click en la nueva rama “develop”, veremos que efectivamente contiene los archivos creados a partir de la rama **Master**:

The screenshot shows the BitBucket interface for a repository named 'Prueba-web2'. The left sidebar contains navigation options: Source, Commits, Branches, Pull requests, Pipelines, Deployments, Jira issues, and Downloads. A yellow arrow points to the 'Jira issues' option. The main content area shows the repository path 'CMPC-Workspace Principal / Pruebas' and the repository name 'Prueba-web2'. A red box highlights the 'develop' branch selected in the branch dropdown menu. Below the branch selector is a search bar for files. The file list table is as follows:

Name	Size	Last commit	Message
HolaMundo		12 minutes ago	Primer Commit de HolaMundo
packages		12 minutes ago	Primer Commit de HolaMundo
.gitignore	624 B	6 days ago	Initial commit
HolaMundo.sln	1.08 KB	12 minutes ago	Primer Commit de HolaMundo
README.md	565 B	6 days ago	Initial commit