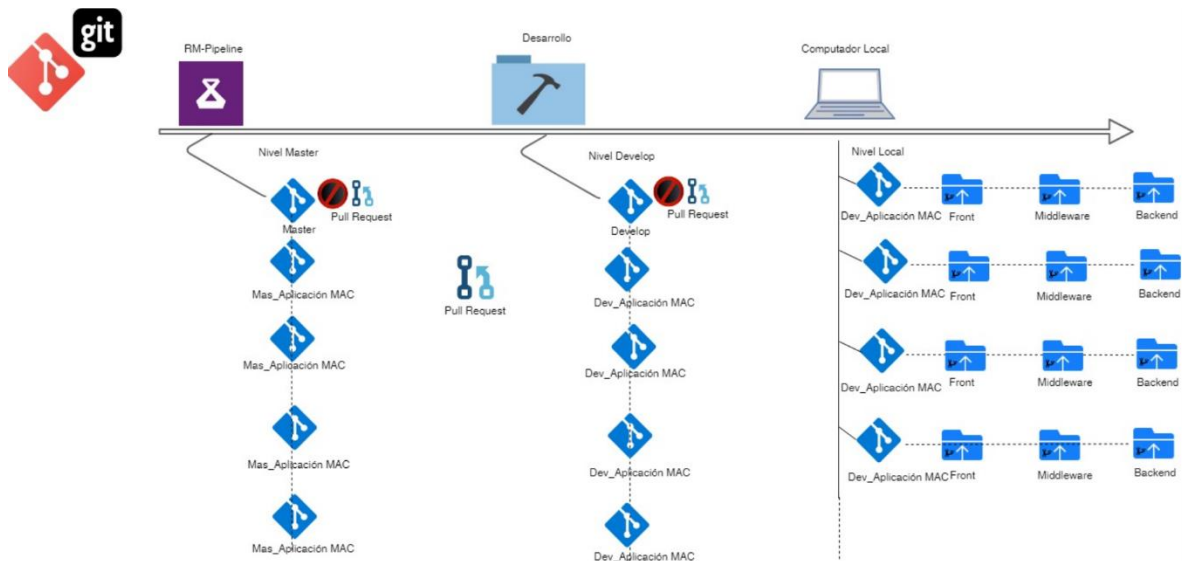


Manual Testware Pruebas Especializadas

Esquema de repositorio

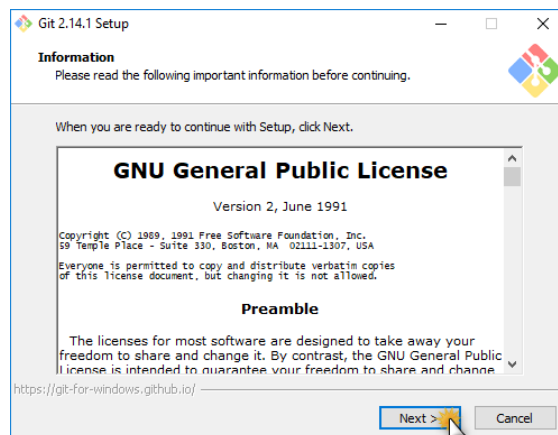
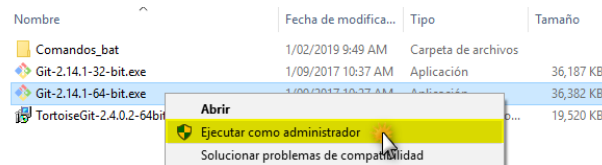


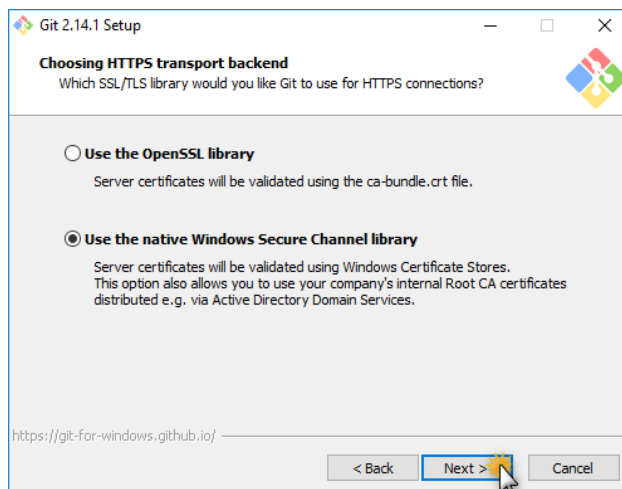
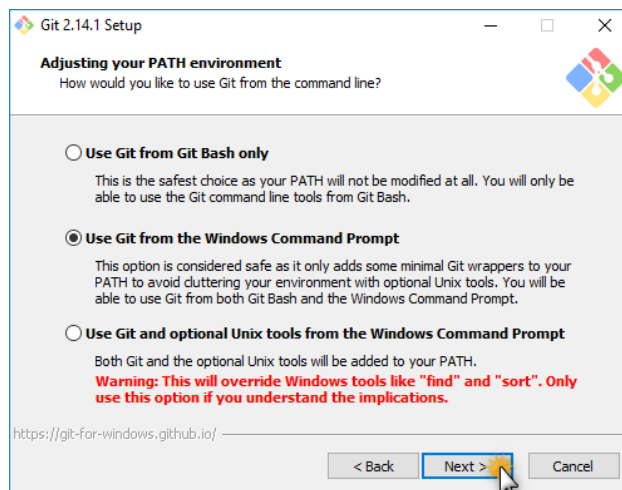
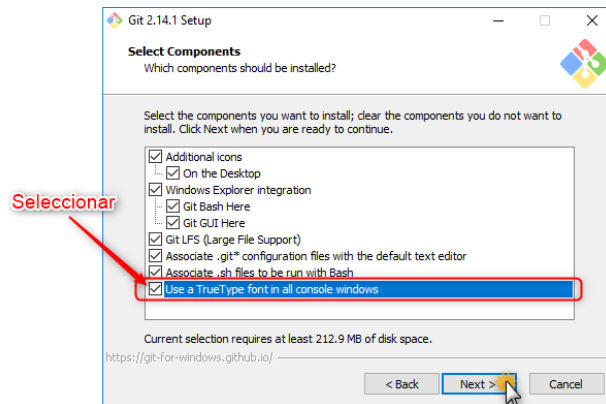
Pre-requisitos:

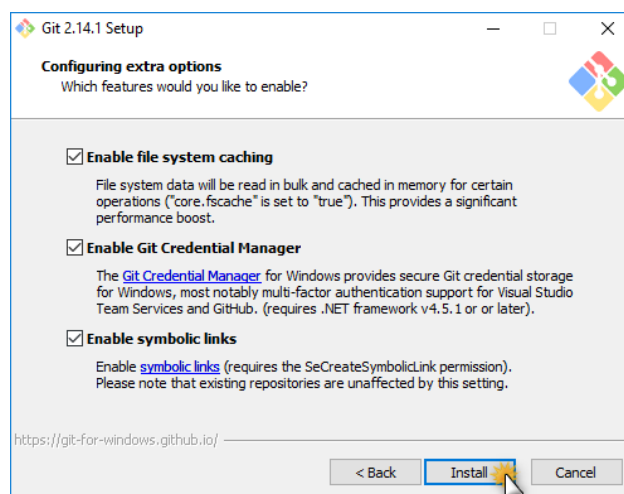
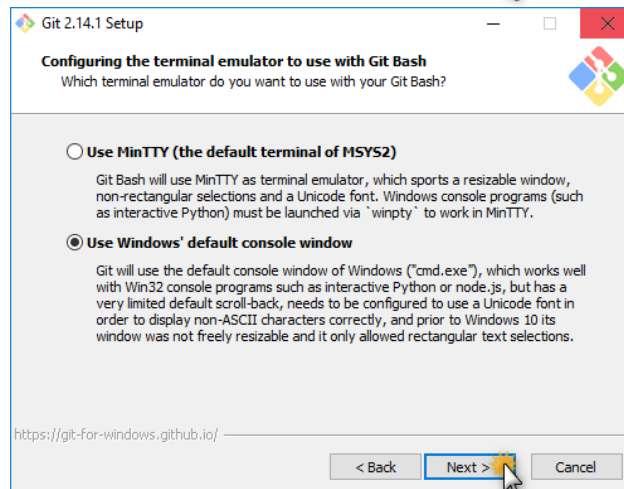
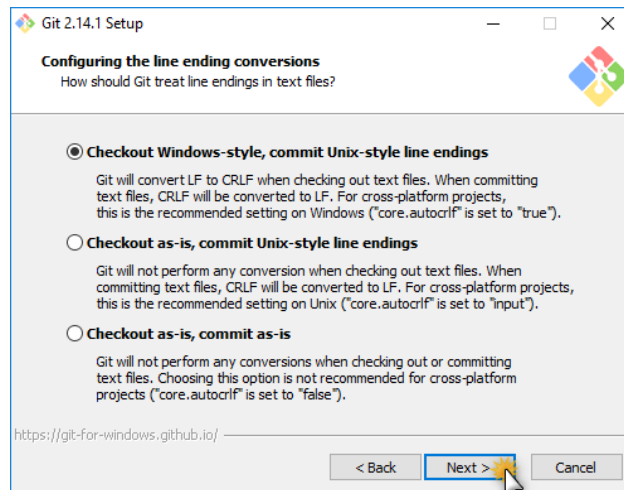
Instalar Git en su máquina local (ejecutar como administrador) y realizar las siguientes configuraciones:

Tomar el .exe de la siguiente ruta:

\\10.8.45.109\\sa-santillana\\Vic Tecnologia\\DIR CERTIFICACION\\GCIA ESTRATEGIA CERTIFICACION\\PRUEBAS ESPECIALIZADAS\\Caja de Herramientas\\Herramientas_Proceso\\Instaladores GIT







Una vez instalado Git se debe realizar la configuración local en cada máquina corriendo el siguiente .bat que se encuentra en la siguiente ruta:

\\10.8.45.109\\sa-santillana\\Vic Tecnologia\\DIR CERTIFICACION\\GCIA ESTRATEGIA CERTIFICACION\\PRUEBAS ESPECIALIZADAS\\Caja de Herramientas\\Herramientas_Proceso\\Instaladores GIT\\Comandos_bat

FICACION > PRUEBAS ESPECIALIZADAS > Caja de Herramientas > Herramientas_Proceso > Instaladores GIT > Comandos_bat				
Nombre	Fecha de modifica...	Tipo	Tamaño	
git clone.bat	19/11/2018 4:28 PM	Archivo por lotes ...	1 KB	
git config Comandos.bat	30/11/2018 3:34 PM	Archivo por lotes ...	1 KB	
git config --global --list.bat	9/11/2018 2:58 PM	Archivo por lotes ...	1 KB	
git config --global.bat	19/11/2018 5:20 PM	Archivo por lotes ...	1 KB	
git config --list.bat	19/11/2018 2:55 PM	Archivo por lotes ...	1 KB	

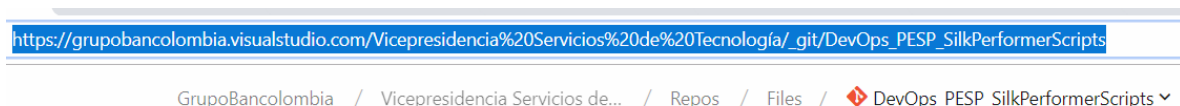
Se debe configurar los siguientes elementos como lo muestra la imagen.

- ✓ Usuario de red
- ✓ Correo corporativo
- ✓ Proxy http
- ✓ Proxy https
- ✓ Deshabilitar SSL

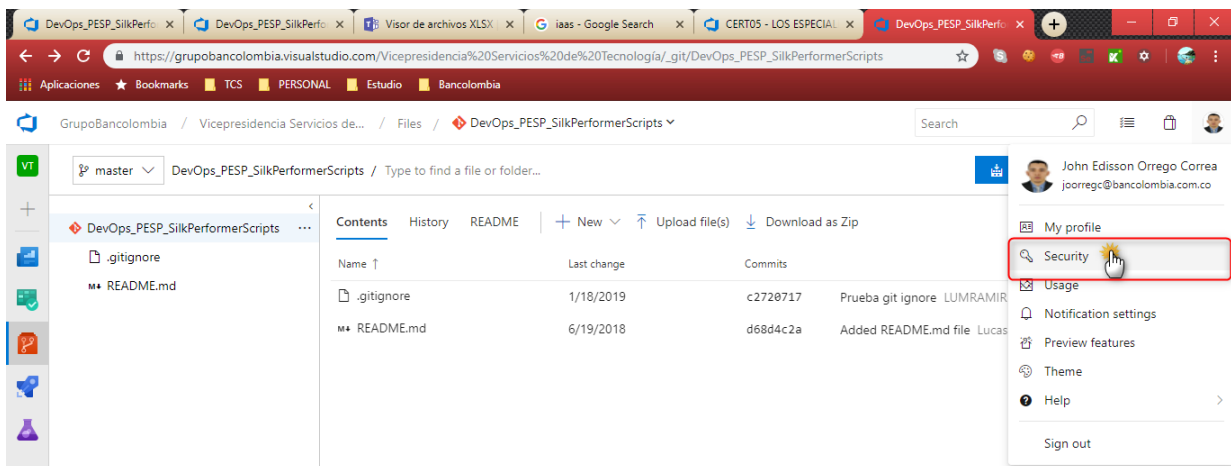
```
type For "Exit (e)"
Usuario Red: LUMRAMIR
Correo Corporativo: lumramir@bancolombia.com.co
Indicar Proxy HTTP, se recomienda(http://proxywsmed.bancolombia.corp:8080): http:proxywsmed.bancolombia.corp:8080
Indicar Proxy HTTPS, se recomienda(https://proxywsmed.bancolombia.corp:8080): https:proxywsmed.bancolombia.corp:8080
Desea deshabilitar SSL, se recomienda (SI) para VSTS: SI
user.name=LUMRAMIR
user.mail=lumramir@bancolombia.com.co
http.proxy=http:proxywsmed.bancolombia.corp:8080
http.sslverify=false
https.proxy=https:proxywsmed.bancolombia.corp:8080
Action:
```

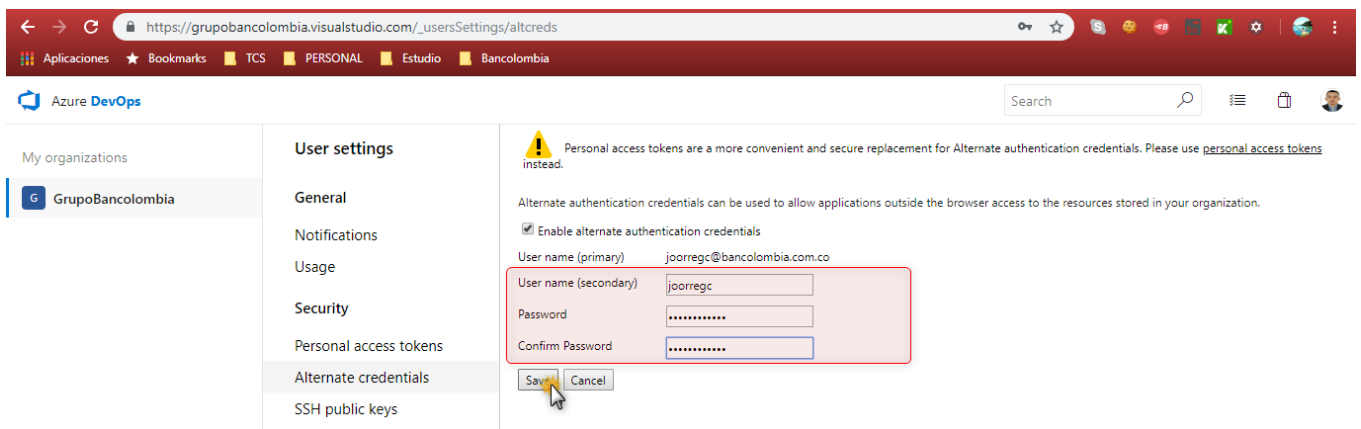
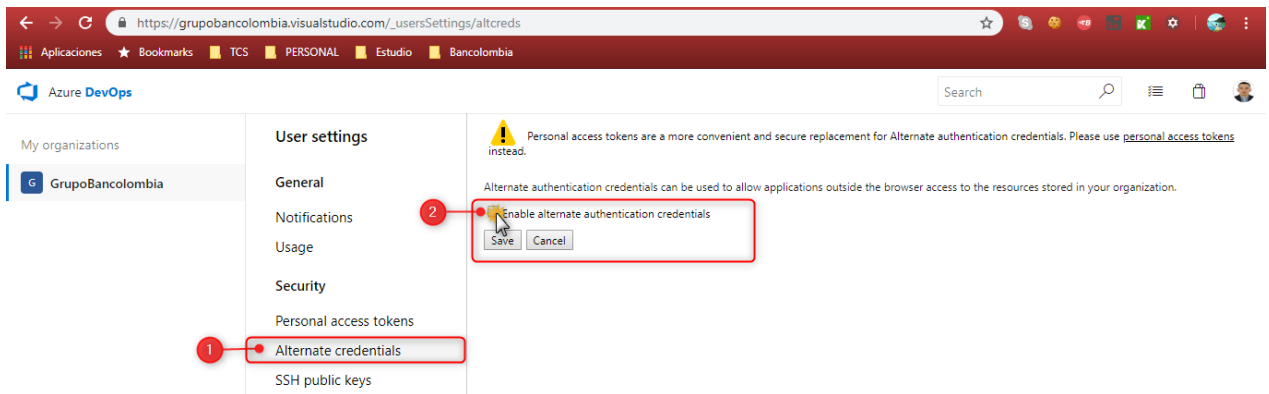
Pasos para la Clonación del Repositorio:

1. Antes de clonar el repositorio en cada máquina local se debe ingresar la siguiente URL en el navegador la cual corresponde al repositorio de PE en VSTS
https://grupobancolombia.visualstudio.com/Vicepresidencia%20Servicios%20de%20Tecnología/_git/DevOps_PESP_SilkPerformerScripts



2. Luego configurar en VSTS una credencial alternativa así:

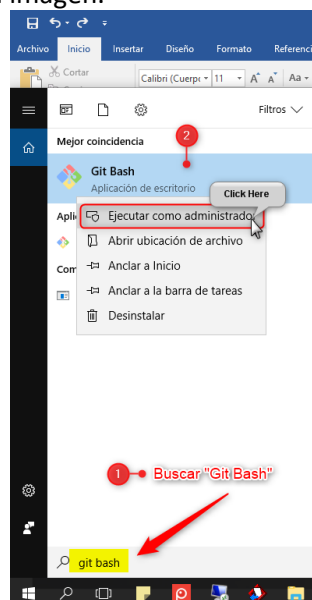


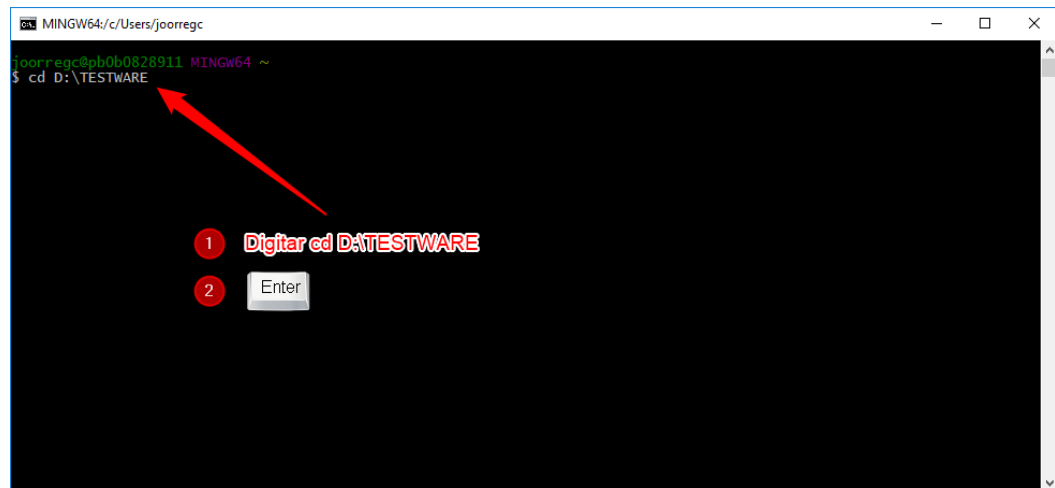


3. Crear una carpeta en cada equipo local así: D:\TESTWARE

Este equipo > Datos (D:) > TESTWARE >

4. Ejecutar "Git Bash" como administrador e ir a la ruta D:\TESTWARE con el comando cd D:\TESTWARE como muestra la imagen.



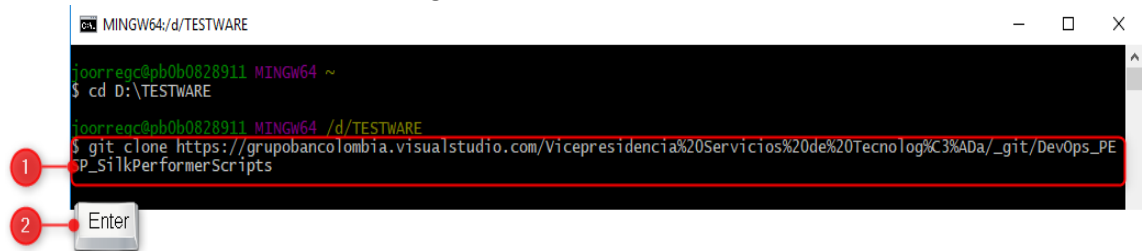


```
MINGW64/c/Users/joorregc
joorregc@pb0b0828911 MINGW64 ~
$ cd D:\TESTWARE
```

1 Digitar **cd D:\TESTWARE**

2 Enter

5. Sobre la ventana que se despliega de **Git Bash** ejecutar el siguiente comando: **git clone https://grupobancolombia.visualstudio.com/Vicepresidencia%20Servicios%20de%20Tecnolog%C3%ADa/_git/DevOps_PESP_SilkPerformerScripts** dentro de la ruta D:\TESTWARE como muestra la imagen:

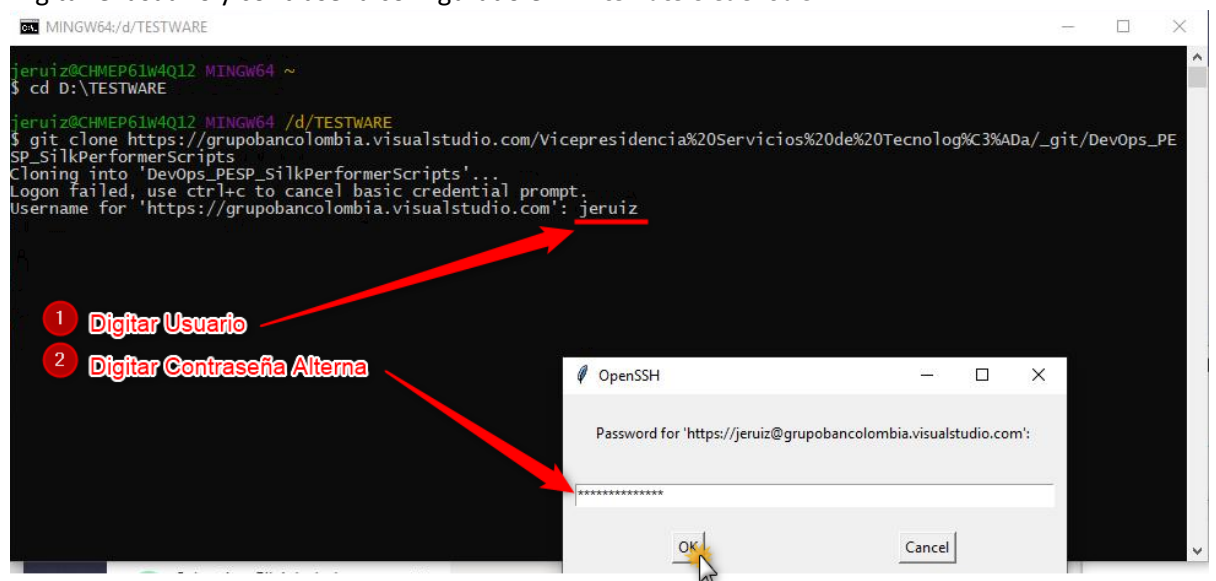


```
MINGW64/d/TESTWARE
joorregc@pb0b0828911 MINGW64 ~
$ cd D:\TESTWARE
joorregc@pb0b0828911 MINGW64 /d/TESTWARE
$ git clone https://grupobancolombia.visualstudio.com/Vicepresidencia%20Servicios%20de%20Tecnolog%C3%ADa/_git/DevOps_PESP_SilkPerformerScripts
```

1 Digitar **git clone https://grupobancolombia.visualstudio.com/Vicepresidencia%20Servicios%20de%20Tecnolog%C3%ADa/_git/DevOps_PESP_SilkPerformerScripts**

2 Enter

6. Digitar el usuario y contraseña configurado en “**Alternate credentials**”



```
MINGW64/d/TESTWARE
jeruiz@CHMEP61W4Q12 MINGW64 ~
$ cd D:\TESTWARE
jeruiz@CHMEP61W4Q12 MINGW64 /d/TESTWARE
$ git clone https://grupobancolombia.visualstudio.com/Vicepresidencia%20Servicios%20de%20Tecnolog%C3%ADa/_git/DevOps_PESP_SilkPerformerScripts
Cloning into 'DevOps_PESP_SilkPerformerScripts'...
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://grupobancolombia.visualstudio.com': jeruiz
```

1 Digitar **Usuario**

2 Digitar **Contraseña Alterna**

OpenSSH

Password for 'https://jeruiz@grupobancolombia.visualstudio.com':

OK Cancel

7. Debe iniciar la clonación del repositorio y completarse al 100% lo cual se puede constatar con la creación de la siguiente carpeta “**DevOps_PESP_SilkPerformerScripts**” y archivos por defecto en la ruta D:\TESTWARE

Nota: la primera vez carga por defecto el Branch master

```
C:\Users\ADMINI~1\AppData\Local\Temp\1\git-20190128-1111\mingw64 /D:/TESTWARE
$ git clone https://grupobancolombia.visualstudio.com/Vicepresidencia%20Servicios%20de%20Tecnolog%C3%ADa/_git/DevOps_PESP_SilkPerformerScripts
Cloning into 'DevOps_PESP_SilkPerformerScripts'...
remote: Azure Repos
remote: Found 509 objects to send. (128 ms)
Receiving objects: 100% (509/509), 35.71 MiB | 504.00 KiB/s, done.
Resolving deltas: 100% (183/183), done.
```

Carpeta: “**DevOps_PESP_SilkPerformerScripts**”

Datos (D:) > TESTWARE			
Nombre	Fecha de modifica...	Tipo	Tamaño
DevOps_PESP_SilkPerformerScripts	28/01/2019 6:47 PM	Carpeta de archivos	

Contenido:

Datos (D:) > TESTWARE > DevOps_PESP_SilkPerformerScripts >			
Nombre	Fecha de modifica...	Tipo	Tamaño
.git	31/01/2019 3:58 PM	Carpeta de archivos	
.gitignore	28/01/2019 6:37 PM	Documento de tex...	1 KB
README.md	28/01/2019 6:37 PM	Archivo MD	1 KB

8. Acceder al archivo **Inventario Testware** y verificar que este nombrada adecuadamente la aplicación de trabajo.

Ruta del archivo:

https://teams.microsoft.com/_#/xlsx/viewer/teams/https%3A~2F~2Fbancolombia.sharepoint.com~2Fteams~2FPECompleto~2FDocumentos%20compartidos~2FGeneral~2FINVENTARIO%20TESTWARE.xlsx?threadId=19%3A3be4d5cb3e8e4c47a9fa4af0f00e64d2%40thread.skype&baseUrl=https%3A~2F~2Fbancolombia.sharepoint.com~2Fteams~2FPECompleto&fileId=925D0FCA-985A-4480-9453-70134EDCB8A5&ctx=files&viewerAction=view

*** Tener en cuenta:**

Aplicativo MAC: Nombre del aplicativo como está registrado en la pestaña “**Matriz de Aplicaciones y Componentes**”, ubicada en el archivo **Inventario Testware**:



Nombre Aplicativo Testware: Corresponde el nombre de la aplicación del campo Aplicativo MAC pero se reemplazan los espacios por _ (guion al piso)

A	B	C	D	E
D	PMO	Aplicativo	Aplicativo MAC	Nombre Aplicativo TestWare
237	PMO27687	ChatBot Jurídico	Chatbot Jurídico	Chatbot_Juridico

9. Teniendo en cuenta el paso anterior donde se estandariza el nombre de la aplicación se procede a crear el primer Branch de trabajo denominado Dev_Nombre_Aplicativo_Testware, para el ejemplo se realiza así:

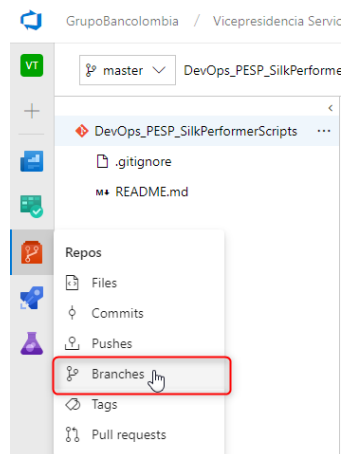
Aplicativo MAC: Chatbot Jurídico

Nombre Aplicativo Testware: Chatbot_Juridico

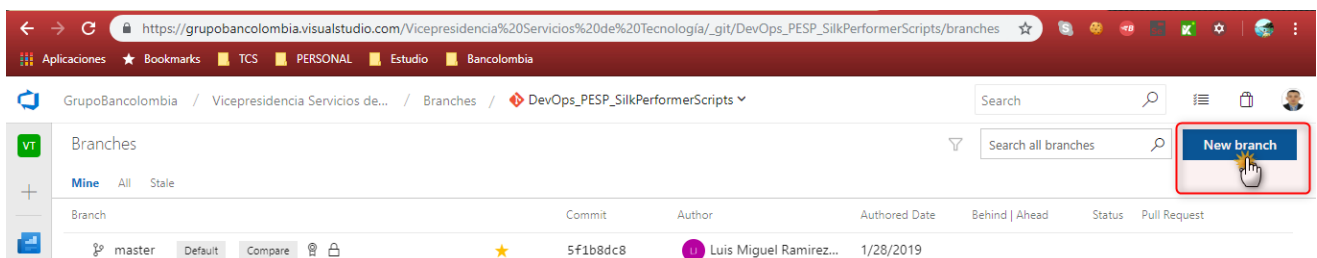
Nombre del Branch a nivel develop: Dev_Chatbot_Juridico

Se debe ir al repositorio en VSTS en la siguiente ruta:

https://grupobancolombia.visualstudio.com/Vicepresidencia%20Servicios%20de%20Tecnolog%C3%ADa/_git/DevOps_PESP_SilkPerformerScripts/branches
o dar click en Repos – Branches

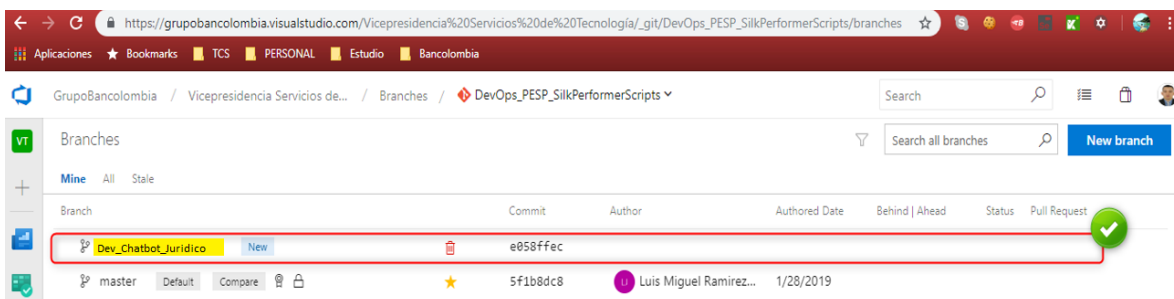
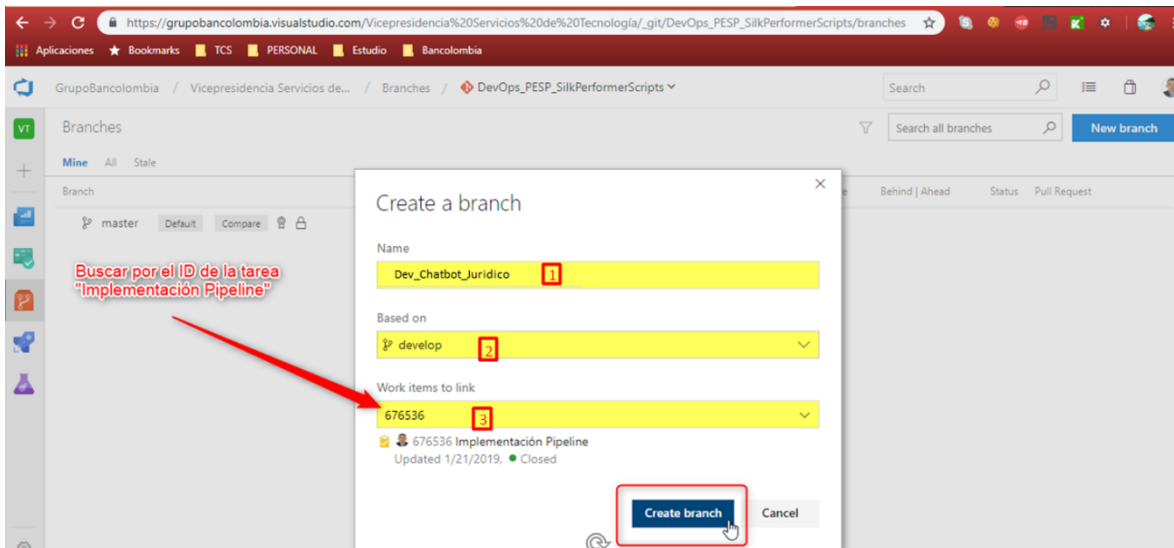


y realizar los siguientes pasos:



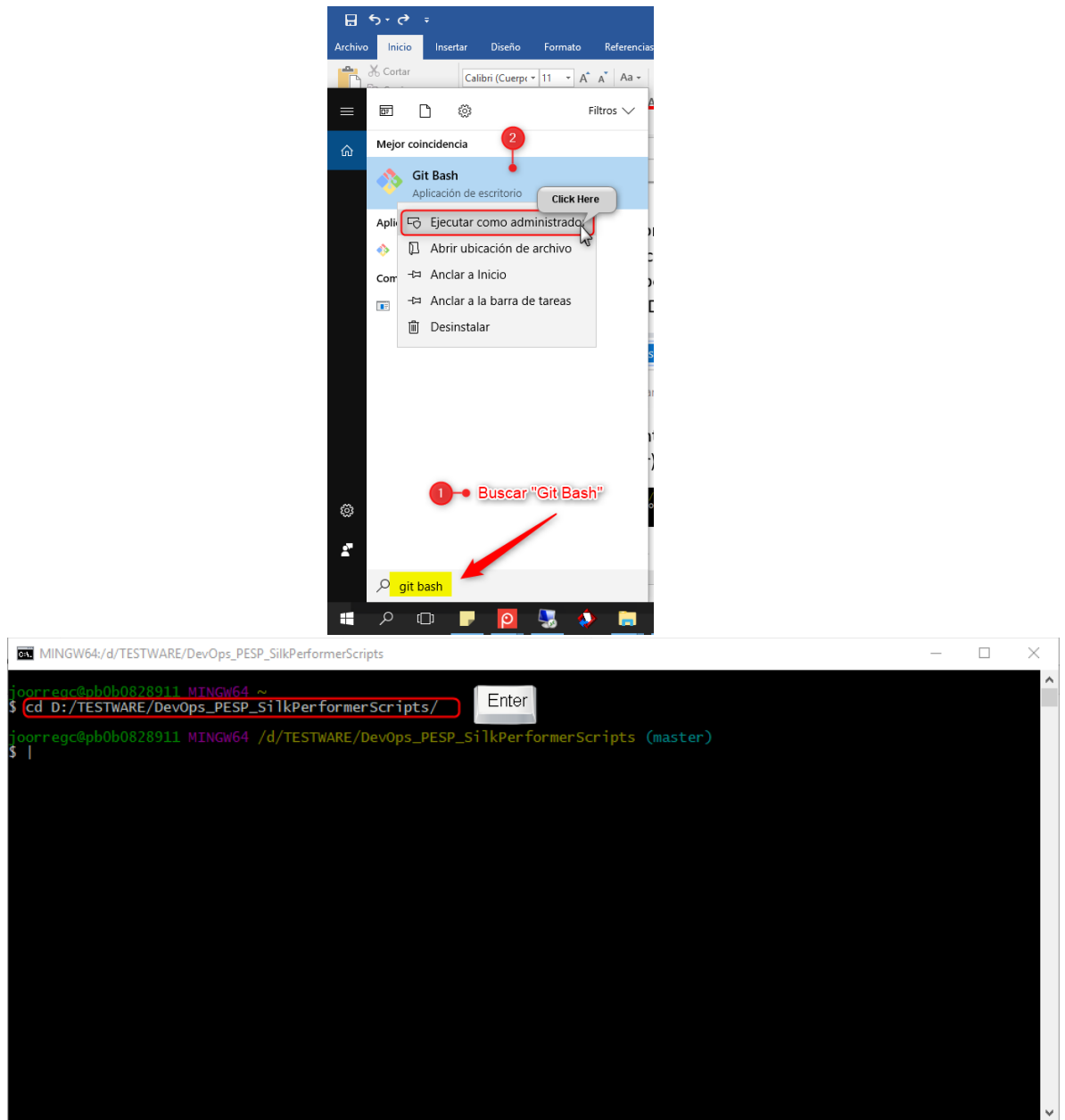
Importante:

1. Tener en cuenta el nombre dado que iniciar por **DEV_**.
2. Tener en cuenta el Branch es basado en la configuración del Branch **develop** (solo se utiliza para configuraciones por defecto en la creación de Branch basados en develop).
3. Tener en cuenta que se debe asociar el Work Item de implementación pipeline del proyecto en el sprint en cual se está realizando la ejecución del proyecto.



10. Una vez creado el Branch de trabajo Dev_ desde VSTS, en el equipo local en la ruta inicial "D:\TESTWARE\DevOps_PESP_SilkPerformerScripts" donde se ejecutó Git, se debe navegar a la ubicación DevOps_PESP_SilkPerformerScripts con el comando CD que corresponde a la carpeta creada en el paso de la clonación de repositorio.

Para esto se debe ejecutar "**Git Bash**" como administrador e ir a la ruta **D:\TESTWARE\DevOps_PESP_SilkPerformerScripts** con el comando **cd D:/TESTWARE/DevOps_PESP_SilkPerformerScripts/** como muestra la imagen.



11. Una vez se ha ingresado a la ruta completa del repositorio local, por defecto git queda en estado checkout al Branch master.

```
joorregc@pb0b0828911 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (master)  
$
```

12. Se debe ejecutar el comando **git checkout Nombre_del_Branch_a_nivel_develop** del Branch recién creado en VSTS para iniciar a trabajar en la carpeta local como muestra la imagen:

```
lunramir@pb0b0828915 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (master)
$ git checkout Dev_Chatbot_Juridico
Switched to a new branch 'Dev_Chatbot_Juridico'
Branch Dev_Chatbot_Juridico set up to track remote branch Dev_Chatbot_Juridico from origin.
lunramir@pb0b0828915 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (Dev_Chatbot_Juridico)
$ _
```

NOTA IMPORTANTE: si al momento de realizar el comando **git checkout** aparece el error (**error: pathspec 'Dev_Aplicación_Testware did not match any file(s) known to git.**) es posible que es su equipo no aparezcan los últimos cambios. Puede validar si aparece si su Branch es visible con el comando **git show-ref** ó **git branch -r**. Sino visualiza su Branch ejecute el comando **git fetch** y luego proceda a ejecutar nuevamente el comando **git checkout**.

Ejemplo realizado con el Branch: Dev_Canal_100_virtual

```
MINGW64/d:/TESTWARE/DevOps_PESP_SilkPerformerScripts
lunramir@pb0b0828911 MINGW64 ~
$ cd D:/TESTWARE/DevOps_PESP_SilkPerformerScripts/
lunramir@pb0b0828911 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (master)
$ git checkout Dev_Canal_100_virtual
error: pathspec 'Dev_Canal_100_virtual' did not match any file(s) known to git.
lunramir@pb0b0828911 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (master)
$ git checkout Dev_Canal_100_virtual
error: pathspec 'Dev_Canal_100_virtual' did not match any file(s) known to git.
lunramir@pb0b0828911 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (master)
$ git show-ref
5f1b8dc8aeb23bbd757a5e3956581abcaadc83f1 refs/heads/master
8abb35b18295a4f42e5b9f9fcdcb48175f0cd3df refs/remotes/origin/APP_Sucursal_Virtual_Personas
cd01f7b4b1f028f45f7d506aee597eb8bbab1e19 refs/remotes/origin/Authentic
e9782be4423153fd2c97eff68e70706242ebc60 refs/remotes/origin/Chatbot_Bancolombia
da5b7c921d465e237445c294c0877b33343f6bb3 refs/remotes/origin/Declaraciones_de_cambio
e2849e4f5c4bbe1e2d043492930ce5479b3c5c refs/remotes/origin/Dev_App_Negocios_Bancolombia
5f1b8dc8aeb23bbd757a5e3956581abcaadc83f1 refs/remotes/origin/Dev_Bizagi
3090c770a502421ed0309a95e3e0a052d46f3719 refs/remotes/origin/Dev_Chatbot_Juridico
8c89dd6074cf87ff990090f4c1079606faf7b8df refs/remotes/origin/Dev_SAS_Studio
d68d4c2a9f70a859a2e9233ad80c76b9bece7c20 refs/remotes/origin/FALCON
5f1b8dc8aeb23bbd757a5e3956581abcaadc83f1 refs/remotes/origin/HEAD
b06113f8941d5dfbdcac3a8878bce56871d31f9 refs/remotes/origin/Mas_Chatbot_Juridico
22ada0b7eb7cec3d1a47daa58effc63be237b5d0 refs/remotes/origin/Murex
b52e39d76147f26ddf9a6010ffed36ac1c9d6cfb refs/remotes/origin/PBI-Power_BI
9255a62e62950e01dcdabfccf5deb53585830203 refs/remotes/origin/PPM-Personal_Financial_Management
192960e3dfe06b637acb9031fad26b9c7932a87c refs/remotes/origin/PSE
b3185533930f1e602780b494ad381654fe8bc243 refs/remotes/origin/SAS_Studio
742ac8f5d77fb725df3fd131be1118df7161ac7 refs/remotes/origin/SISU_SIMULADOR_SUFI_FRONTEND
e058ffec1afasc78d73b3f99afaada00a8d3321 refs/remotes/origin/develop
5f1b8dc8aeb23bbd757a5e3956581abcaadc83f1 refs/remotes/origin/master
```

No se visualiza "Dev_Canal_100_virtual"

```
MINGW64/d/TESTWARE/DevOps_PESP_SilkPerformerScripts
jorregc@pb0b0828911 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (master)
$ git fetch
remote: Azure Repos
remote: Found 219 objects to send. (4 ms)
Receiving objects: 100% (219/219), 2.49 MiB | 272.00 KiB/s, done.
Resolving deltas: 100% (66/66), completed with 6 local objects.
From https://grupobancolombia.visualstudio.com/Vicepresidencia%20de%20Tecnolog%C3%ADa/_git/DevOps_PESP_SilkPerformerScripts
* [new branch] Dev_App_Negocios_Bancolombia -> origin/Dev_App_Negocios_Bancolombia
* [new branch] Dev_Canal_100_virtual -> origin/Dev_Canal_100_virtual
* [new branch] Dev_Chatbot_Bancolombia -> origin/Dev_Chatbot_Bancolombia
* [new branch] Dev_Sucursal_Virtual -> origin/Dev_Sucursal_Virtual
* [new branch] Dev_WOLF -> origin/Dev_WOLF
* [new branch] Mas_App_Negocios_Bancolombia -> origin/Mas_App_Negocios_Bancolombia
* [new branch] Mas_Chatbot_Bancolombia -> origin/Mas_Chatbot_Bancolombia
* [new branch] b06113f..2759c5e Mas_Chatbot_Juridico -> origin/Mas_Chatbot_Juridico
* [new branch] Mas_Sucursal_Virtual -> origin/Mas_Sucursal_Virtual
* [new branch] Mas_WOLF -> origin/Mas_WOLF
jorregc@pb0b0828911 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (master)
$ git branch -r
origin/App_Sucursal_Virtual_Personas
origin/Authentic
origin/Chatbot_Bancolombia
origin/Declaraciones_de_cambio
origin/Dev_App_Negocios_Bancolombia
origin/Dev_Bizagi
origin/Dev_Canal_100_virtual
origin/Dev_Chatbot_Bancolombia
origin/Dev_Chatbot_Juridico
origin/Dev_SAS_Studio
origin/Dev_Sucursal_Virtual
origin/Dev_WOLF
origin/FALCON
origin/HEAD -> origin/master
origin/Mas_App_Negocios_Bancolombia
origin/Mas_Chatbot_Bancolombia
origin/Mas_Chatbot_Juridico
origin/Mas_Sucursal_Virtual
origin/Mas_WOLF
origin/Murex
origin/PBI-Power_BI
origin/PPM-Personal_Financial_Management
origin/PSE
origin/SAS_Studio
origin/SISO_SIMULADOR_SUFI_FRONTEND
origin/develop
origin/master
```

Después de ejecutar el comando **git checkout Nombre_del_Branch_a_nivel_develop** podemos comenzar a trabajar en la carpeta local

```
jumramir@pb0b0828915 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (master)
$ git checkout Dev_Chatbot_Juridico
Switched to a new branch 'Dev_Chatbot_Juridico'
Branch Dev_Chatbot_Juridico set up to track remote branch Dev_Chatbot_Juridico from origin.
jumramir@pb0b0828915 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (Dev_Chatbot_Juridico)
$
```

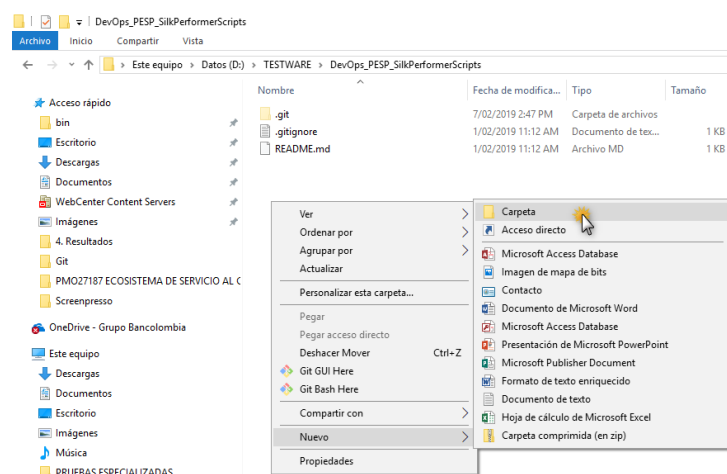
13. En la ruta local del repositorio debe cargar la primera vez las fuentes por defecto que contiene el Branch Dev_ recién creado, en próximas veces deberá cargar el Branch con las fuentes en última versión trabajada tanto local como remoto.

Datos (D:) > TESTWARE > DevOps_PESP_SilkPerformerScripts

Nombre	Fecha de modifica...	Tipo	Tamaño
.git	7/02/2019 2:47 PM	Carpeta de archivos	
.gitignore	1/02/2019 11:12 AM	Documento de tex...	1 KB
README.md	1/02/2019 11:12 AM	Archivo MD	1 KB

14. La primera carpeta a crear corresponde a la capa de aplicación desde la cual se ejecuta las pruebas para la aplicación correspondiente en este caso es Front.

Paso 1: Crear carpeta en la ruta indicada

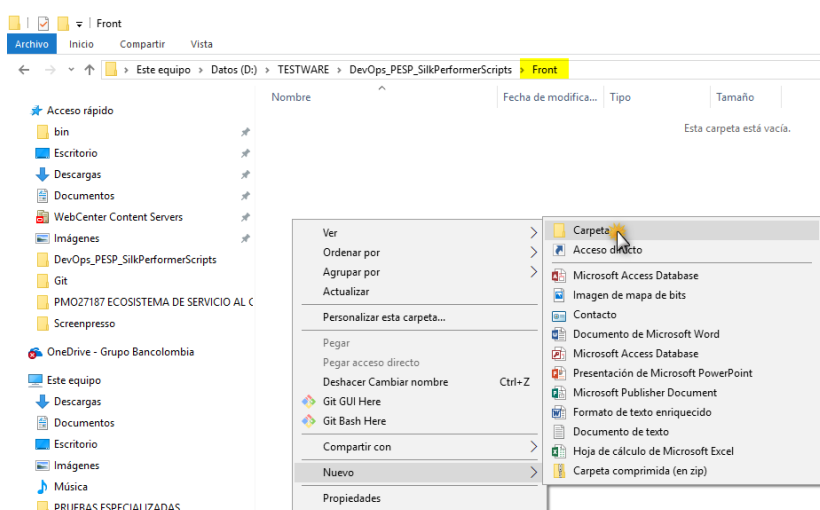


Paso 2: La carpeta a crear debe contener el nombre de la capa de ejecución en este caso **Front**


Datos (D:) > TESTWARE > DevOps_PESP_SilkPerformerScripts >				
Nombre	Fecha de modifica...	Tipo	Tamaño	
.git	25/01/2019 3:54 PM	Carpeta de archivos		
Front	25/01/2019 3:57 PM	Carpeta de archivos		
.gitignore	25/01/2019 3:37 PM	Documento de tex...	1 KB	
README.md	25/01/2019 3:37 PM	Archivo MD	1 KB	

15. Una vez creada la carpeta de capa de ejecución “Front” la siguiente carpeta a crear corresponde al nombre de la aplicación como se indicó en el archivo **Inventario Testware** en la columna **Nombre Aplicación Testware: Chatbot_Juridico**

Paso 1: Se crea la carpeta de **aplicación testware** al interior de la carpeta **capa ejecución**



Paso 2: La carpeta debe ser Nombre de Aplicación Testware en este caso Chatbot_Juridico

Datos (D:) > TESTWARE > DevOps_PESP_SilkPerformerScripts > Front >			
Nombre	Fecha de modifica...	Tipo	Tamaño
 Chatbot_Juridico	25/01/2019 3:57 PM	Carpeta de archivos	

Importante: Se debe crear el proyecto desde la ruta local del Branch en el equipo la cual debe quedar localmente con la siguiente ruta:

D:\TESTWARE\DevOps_PESP_SilkPerformerScripts\“Capa Ejecución”\Nombre Aplicativo Testware

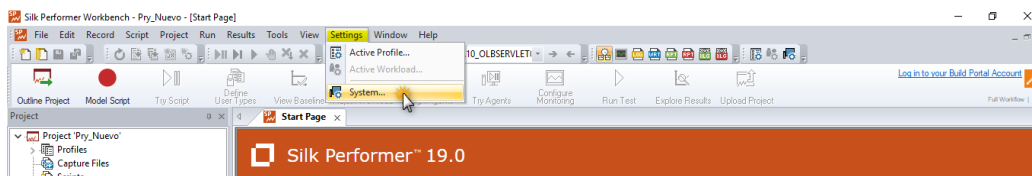
Ejemplo: *D:\TESTWARE\DevOps_PESP_SilkPerformerScripts\Front\Chatbot_Juridico*

Nota: Recuerde estar checkout en el Branch app indicado

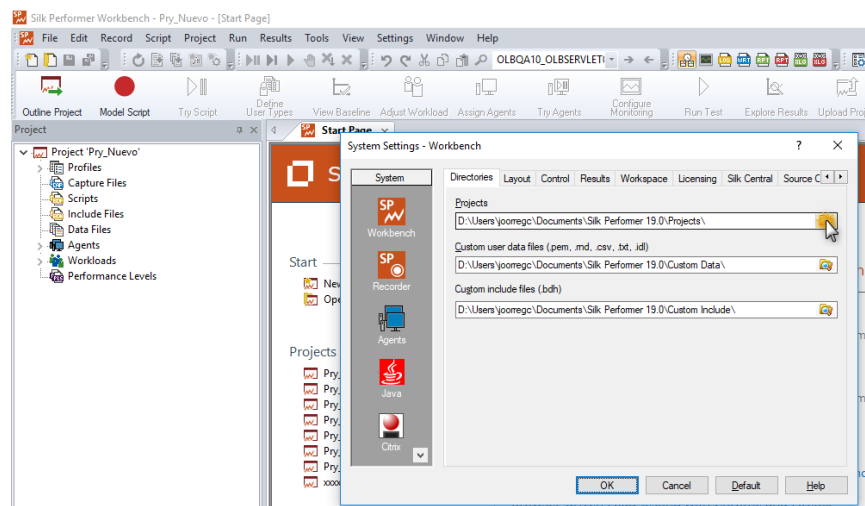
- Una vez se crea la ruta de carpeta local se procede con la creación del proyecto desde la herramienta SilkPerformer como lo indica las siguientes imágenes (**siempre y cuando el proyecto sea nuevo**), donde se renombran la ruta por defecto aplicando la ruta local del repositorio, esto para que coincida con la ruta que tendrá en el servidor de Pruebas Especializadas y no se deban renombrar rutas posteriormente. (**Importante:** solo se indica la ruta hasta la carpeta capa de ejecución)

Nota: Para el caso de Jmeter el archivo .jmx y Csv pueden ser creados en otra ruta y luego ser copiados a la ruta local del repositorio

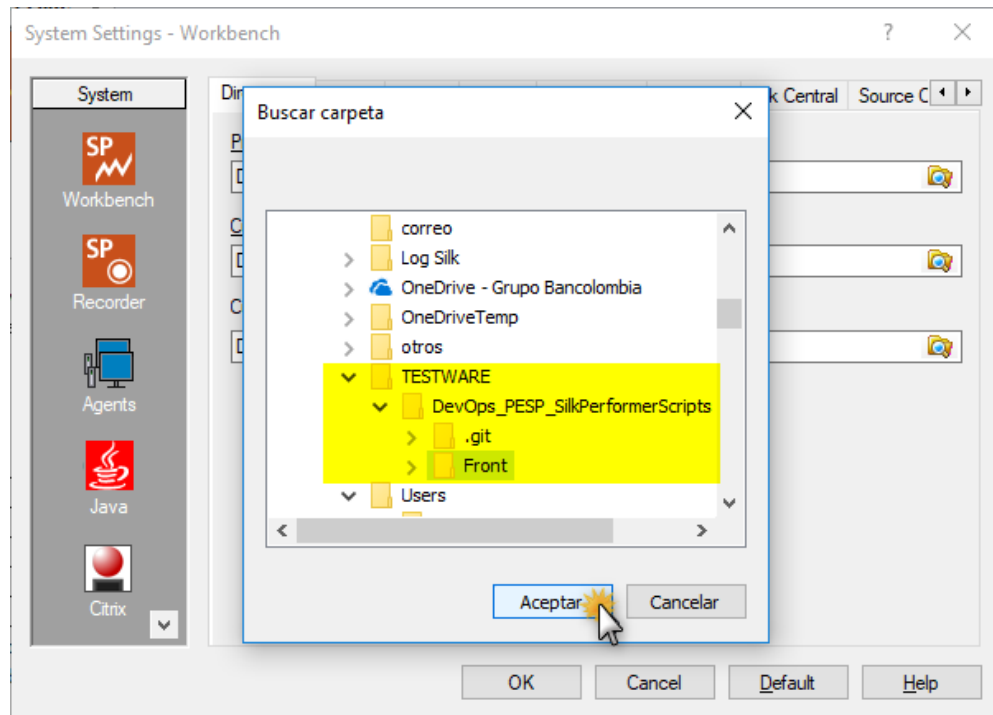
Paso 1: (System)



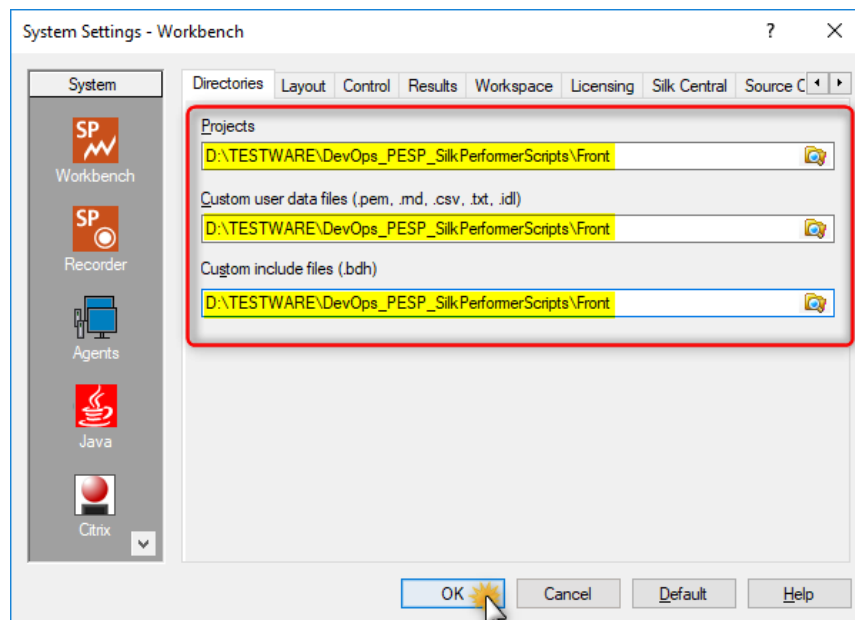
Paso 2: Directories



Paso 3: Ubicar la carpeta de forma local

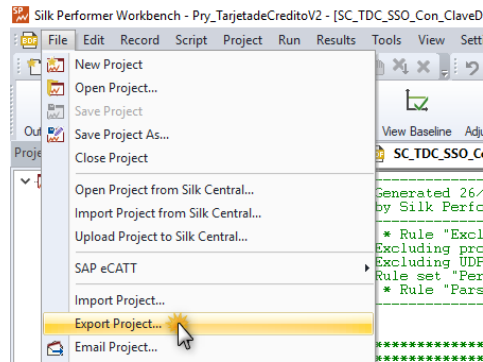


Paso 4: Renombrar los directorios de trabajo en Silk con la ruta del Testware hasta la capa de ejecución.

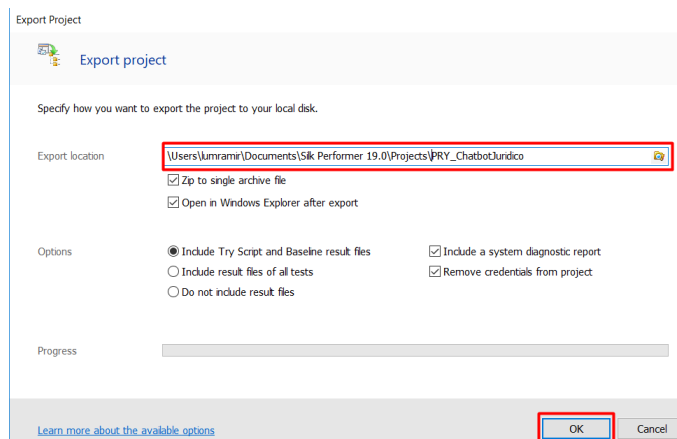


17. Para los proyectos **ya existentes en otra ruta y trabajados antes** se debe hacer los siguientes pasos:

Paso 1: Abrir el proyecto en Silkperformer desde la ruta del equipo en la que se tenga el proyecto, y dar Clic en “Export Project”.



Paso 2: Al exportar el proyecto se debe indicar una ruta donde quedará almacenado el archivo .ltz



Paso 3: Ir a la carpeta donde se generó el .ltz y dar doble clic

> Documentos > Silk Performer 19.0 > Projects > PRY_ChatbotJuridico >				
Nombre	Fecha de modifica...	Tipo	Tamaño	
DT_EndPoint.csv	29/11/2018 4:46 PM	Archivo de valores...	1 KB	
DT_Preguntas_Chatbot.csv	29/11/2018 3:10 PM	Archivo de valores...	1 KB	
DT_Preguntas_ServicioLuis.csv	29/11/2018 6:25 PM	Archivo de valores...	1 KB	
DT_ServicioCognitivo_ServiceAcces.csv	29/11/2018 6:30 PM	Archivo de valores...	1 KB	
PRY_ChatbotJuridico.ltp	26/12/2018 12:04 ...	Silk Performer 19.0...	139 KB	
PRY_ChatbotJuridico.ltz	20/12/2018 11:35 ...	Silk Performer 19.0...	145 KB	
PRY_ChatbotSofy.pew	17/04/2018 7:29 A...	Silk Performance E...	47 KB	
PRY_ChatbotSofy.record.bdf	23/09/2018 9:25 PM	Silk Performer 19.0...	33 KB	
PRY_ChatbotSofy.record.log	23/09/2018 9:25 PM	Documento de tex...	202 KB	
PRY_ChatbotSofy.record.tlz	23/09/2018 9:25 PM	Silk TrueLog Explo...	71 KB	
PRY_ChatbotSofy.spcap	23/09/2018 9:24 PM	Silk Performer 19.0...	291 KB	

Paso 4: Se abrirá una ventana correspondiente a Silk en la cual se le debe indicar la nueva ruta del proyecto

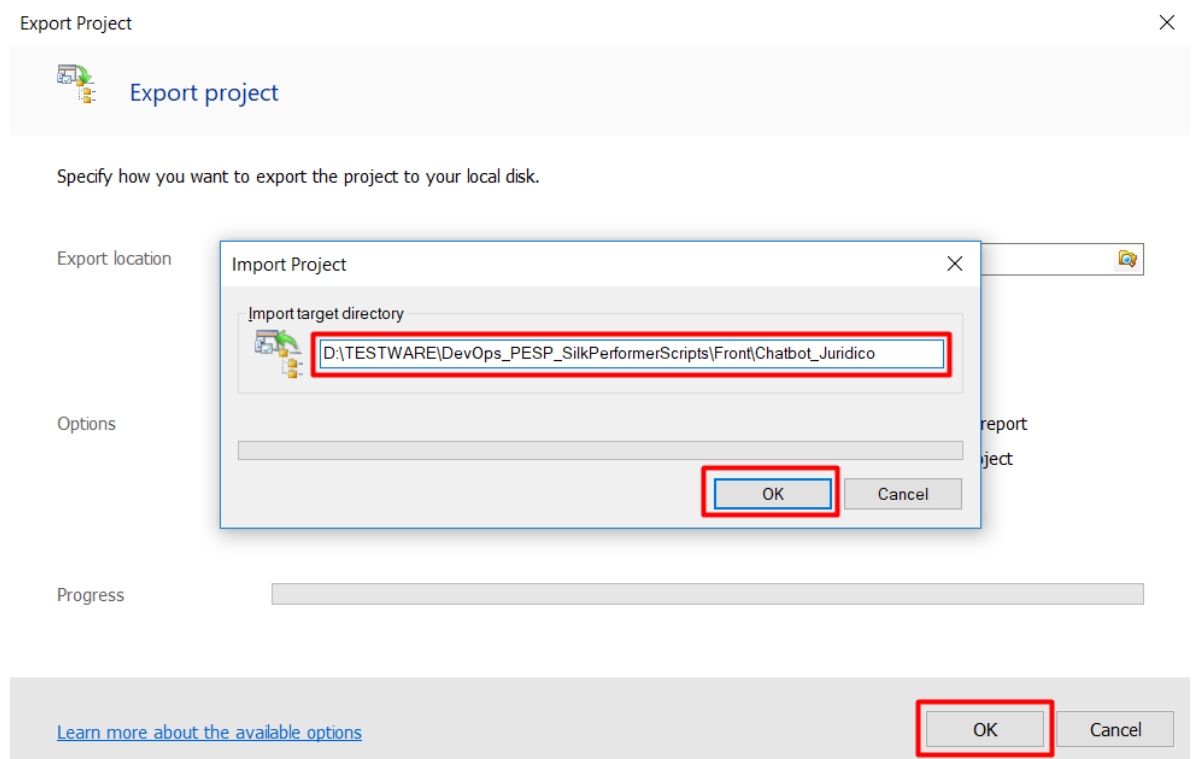
Nota: Exportar e importar actualiza las rutas internas del proyecto, por tal razón copiar y pegar el proyecto de una anterior ubicación a la ruta nueva de Testware, no es un procedimiento válido.

Importante: La ruta donde se va descomprimir el proyecto en la nueva ubicación debe ser la ruta del repositorio local dentro de la carpeta *Nombre Aplicativo Testware*.

D:\TESTWARE\DevOps_PESP_SilkPerformerScripts\“Capa”\ *Nombre Aplicativo Testware*

Ejemplo:

D:\TESTWARE\DevOps_PESP_SilkPerformerScripts\Front\Chatbot_Juridico



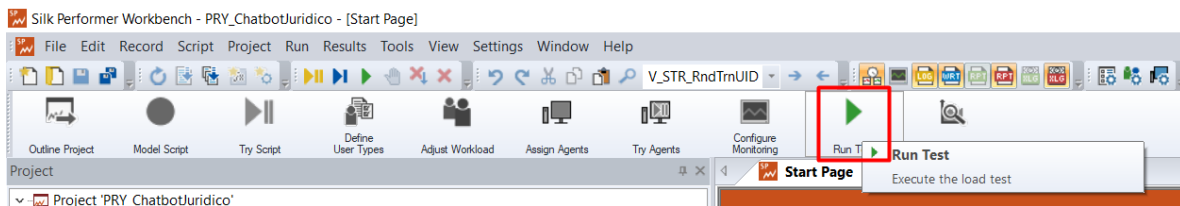
Pre-requisitos para subir cambios en repositorio.

1. Como lo indica la siguiente imagen se recomienda que el contenido de un proyecto de pruebas de performance construido en la herramienta SilkPerformer debe tener la siguiente estructura de carpetas, CSV de datos y fuentes de script. Los archivos .CSV data set se pueden también llevar a una carpeta de datos para una mejor organización.

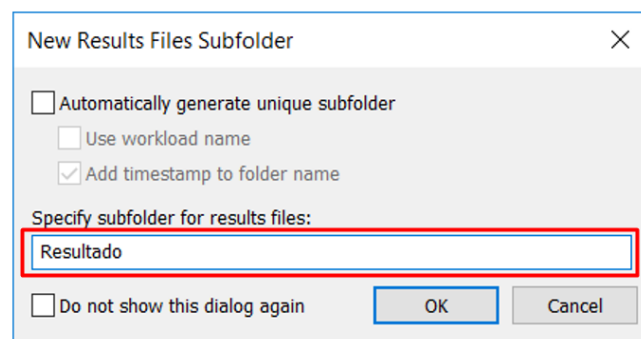
BaselineResults	06/02/2019 11:29 ...	Carpeta de archivos	
Include	06/02/2019 11:29 ...	Carpeta de archivos	
PerformanceLevels	06/02/2019 11:29 ...	Carpeta de archivos	
RecentTryScriptTest	06/02/2019 11:29 ...	Carpeta de archivos	
Resultado	06/02/2019 11:29 ...	Carpeta de archivos	
DT_DirectorioActivo_ServiceAccess.csv	06/02/2019 11:29 ...	Archivo de valores...	2 KB
DT_DirectorioActivoUsers.csv	06/02/2019 11:29 ...	Archivo de valores...	2 KB
DT_EndPoint.csv	06/02/2019 11:29 ...	Archivo de valores...	1 KB
DT_Preguntas_Chatbot.csv	06/02/2019 11:29 ...	Archivo de valores...	1 KB
DT_Preguntas_ServicioLuis.csv	06/02/2019 11:29 ...	Archivo de valores...	1 KB
DT_ServicioCognitivo_ServiceAcces.csv	06/02/2019 11:29 ...	Archivo de valores...	1 KB
PRY_ChatbotJuridico.ltp	06/02/2019 11:29 ...	Silk Performer 19.0...	114 KB
PRY_ChatbotJuridico.ltz	06/02/2019 11:29 ...	Silk Performer 19.0...	438 KB
PRY_ChatbotSofy.pew	06/02/2019 11:29 ...	Silk Performance E...	47 KB
PRY_ChatbotSofy.spcap	06/02/2019 11:29 ...	Silk Performer 19.0...	291 KB
SC_ChatbotJuridico.bdf	06/02/2019 11:29 ...	Silk Performer 19.0...	18 KB
SC_ChatbotJuridico.bex	06/02/2019 11:29 ...	Silk Performer 19.0...	113 KB
SC_DirectorioActivo.bdf	06/02/2019 11:29 ...	Silk Performer 19.0...	4 KB
SC_DirectorioActivo.bex	06/02/2019 11:29 ...	Silk Performer 19.0...	93 KB
SC_ServicioCognitivo.bdf	06/02/2019 11:29 ...	Silk Performer 19.0...	5 KB
SC_ServicioCognitivo.bex	06/02/2019 11:29 ...	Silk Performer 19.0...	94 KB

Nota: Si existen resultados de ejecuciones realizadas localmente, se recomienda que estas queden almacenadas en una carpeta llamada “Resultado”. No deben quedar en una carpeta diferente, a continuación, se indica cómo crear esta carpeta en el momento de ejecutar una prueba para que queden almacenados los resultados allí.

Paso 1: Ejecutar el proyecto como la imagen lo indica.



Paso 2: La ventana que se despliega antes de iniciar la ejecución, en el campo de **SubFolder de Resultados** debe llevar el nombre “Resultado”




Importante: Los resultados de pruebas de ejecuciones locales y anteriores no son almacenables en el repositorio, ya que estos por su tamaño algunas veces superan el límite de transferencia establecido en reglas de firewall para VSTS, por tal motivo cada Branch en su configuración base contiene un archivo .gitignore que ignora la carga de la carpeta “Resultado”

Recomendación: Se recomienda no eliminar el archivo .gitignore el cual contiene configuración por defecto que ignora la carga al repositorio VSTS de elementos de trabajo no sujetos de versionar como la carpeta “Resultado”.

Datos (D:) > TESTWARE > DevOps_PESP_SilkPerformerScripts >				
Nombre	Fecha de modifica...	Tipo	Tamaño	
.git	25/01/2019 3:54 PM	Carpeta de archivos		
.gitignore	25/01/2019 3:37 PM	Documento de tex...	1 KB	
README.md	25/01/2019 3:37 PM	Archivo MD	1 KB	

Estructura interna del archivo .gitignore

 .gitignore: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
# Ignore folder named 'Resultado'
Resultado/
```

18. Una vez se tenga el script de prueba construido o migrado se debe ejecutar el comando git add . para iniciar el proceso de sincronización de cambios del equipo local al repositorio VSTS

```
lumramir@pb0b0828915 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (Dev_Chatbot_Juridico)
$ git add .
```

19. Posterior se ejecuta el comando git commit -m “Comentarios de versión”

```
lumramir@pb0b0828915 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (Dev_Chatbot_Juridico)
$ git commit -m "Version 1.0 Inicial Script App"
[Dev_Chatbot_Juridico d3f218e] Version 1.0 Inicial Script App
Committer: LUMRAMIR <lumramir@bancolombia.com.co>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

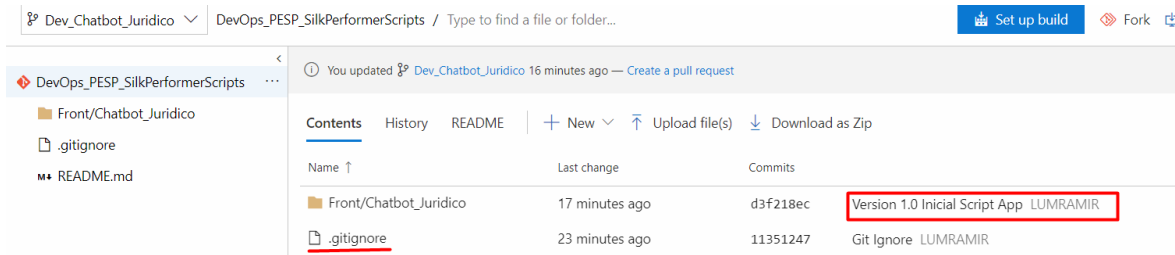
42 files changed, 125346 insertions(+)
create mode 100644 Front/Chatbot_Juridico/BaselineResults/baselineReport_Carga.brp
create mode 100644 Front/Chatbot_Juridico/BaselineResults/baselineReport_Estres 125%.brp
create mode 100644 Front/Chatbot_Juridico/BaselineResults/baselineReport_Estres 150%.brp
create mode 100644 Front/Chatbot_Juridico/BaselineResults/baselineReport_Estres 200%.brp
create mode 100644 Front/Chatbot_Juridico/BaselineResults/baselineReport_Estres 300%.brp
create mode 100644 Front/Chatbot_Juridico/BaselineResults/baselineReport_Lineabase.brp
```

20. Finalmente se ejecuta el comando git push para sincronizar cambios de área de trabajo local al repositorio VSTS

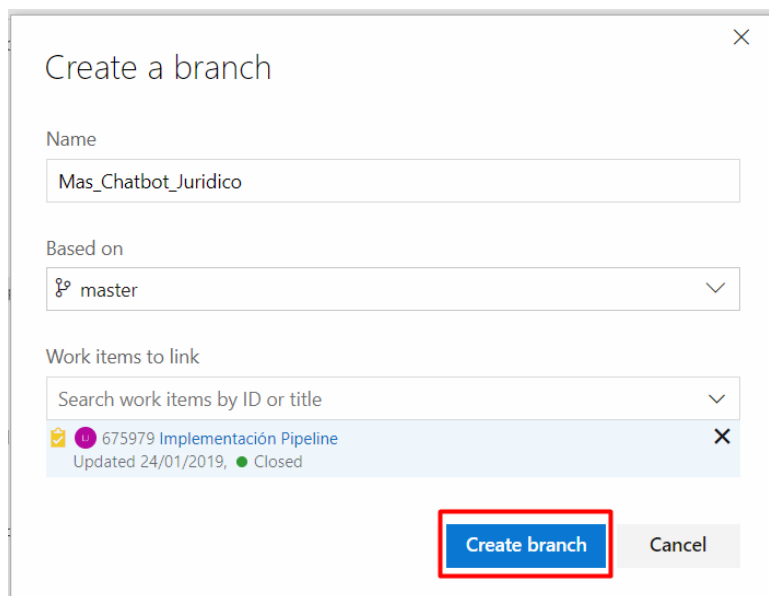
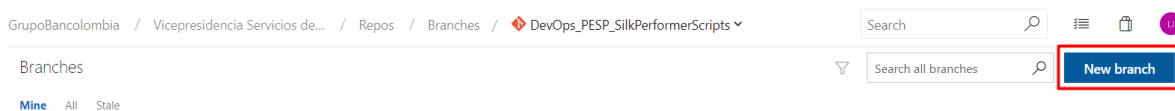
```
lumramir@pb0b0828915 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (Dev_Chatbot_Juridico)
$ git push
```

```
lumramir@pb0b0828915 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (Dev_Chatbot_Juridico)
$ git push
Counting objects: 48, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (33/33), done.
Writing objects: 100% (48/48), 815.64 KiB | 81.56 MiB/s, done.
Total 48 (delta 15), reused 43 (delta 14)
remote: Analyzing objects... (48/48) (577 ms)
remote: Storing packfile... done (92 ms)
remote: Storing index... done (73 ms)
To https://grupobancolombia.visualstudio.com/Vicepresidencia%20Servicios%20de%20Tecnolog%C3%ADa/_git/DevOps_PESP_SilkPerformerScripts
1135124..d3f218e Dev_Chatbot_Juridico -> Dev_Chatbot_Juridico
```

21. Los cambios cargados al repositorio se deben ver en VSTS en el Branch correspondiente con los comentarios indicados en pasos anteriores.



22. En este paso se crea el Branch a nivel Master, desde este Branch se ejecutarán las pruebas de performance en el servidor de PE, el cual tiene como indicador de nombre Mas_Nombre_Aplicativo_Testware





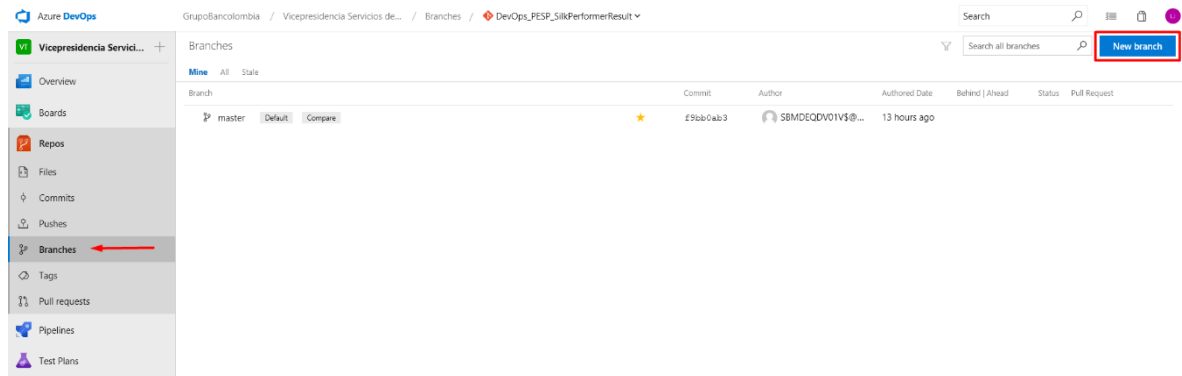
Una vez creado los branch en VSTS se debe actualizar dichos cambios en el computador local haciendo git pull, de esta forma el repositorio local del computador queda actualizado para poder trabajar los branches recién creados

En la imagen se observa como identifica la creación de los Branch realizada anteriormente en VSTS.

```
lunramir@bb0b0828915 MINGW64 /d/TESTWARE/DevOps_PESP_SilkPerformerScripts (master)
$ git pull
remote: Azure Repos
remote: Found 1 objects to send. (1 ms)
Unpacking objects: 100% (1/1), done.
From https://grupobancolombia.visualstudio.com/Vicepresidencia%20Servicios%20de%20Tecnolog%C3%ADa/_git/DevOps_PESP_SilkPerformerScripts
+ d3f218e...e058ffe Dev_Chatbot_Juridico -> origin/Dev_Chatbot_Juridico (forced update)
3ba7515..5f1b8dc Mas_Chatbot_Juridico -> origin/Mas_Chatbot_Juridico
```

23. La integración de cambios entre ramas, para este caso de una rama Dev_ a una rama Mas_ se realiza mediante **Pull Request**. Este pull request se crea como se indica a continuación.

Se ingresa a la opción en menú  Branches y en la ventana que se carga dar clic en 



Recomendaciones del Pull Request

1. El pull request debe ir entre la rama Dev_ a la rama Mas_ que correspondan a la misma aplicación, no se debe realizar pull request entre ramas de aplicaciones diferentes.
2. Se debe indicar una descripción correspondiente a la versión de scripts que se quiere integrar a la rama Mas_ se recomienda que estas versiones sean 1.0, 2.0, 3.0 (dígitos cerrados) los cambios realizados en cada commit sobre la rama Dev_ se recomienda numerarlos con 1.1, 1.2 o 2.1, 2.2 según sea el caso.
3. Se debe incluir aprobadores, se recomienda incluir el aprobador encargado de revisar e integrar cambios del equipo stack en la aplicación de trabajo.
4. Se debe asociar el work ítem correspondiente a la tarea de “Implementación Pipeline” del proyecto en VSTS.

La imagen indica cómo se configura cada recomendación:

 New Pull Request

Dev_Chatbot_Juridico into Mas_Chatbot_Juridico  1

Title *

Pull Request Dev_Chatbot_Juridico a Mas_Chatbot_Juridico V1.0 Script

Add label

Description  2

Mediante Pull Request se integra el script de la línea Develop (Dev_Chatbot_Juridico) a la línea Master (Mas_Chatbot_Juridico) en su versión 1.0 para ser ejecutado en el Pipeline (RM_Proyecto XXXX) correspondiente al ID VSTS (xxxxxx).

Markdown supported.

Aa        @ #  Add commit messages

Mediante Pull Request se integra el script de la línea Develop (Dev_Chatbot_Juridico) a la línea Master (Mas_Chatbot_Juridico) en su versión 1.0 para ser ejecutado en el Pipeline (RM_Proyecto XXXX) correspondiente al ID VSTS (xxxxxx).

Reviewers  3

 Luis Miguel Ramirez Jaramillo  Juan Diego Palacio Restrepo  Search users and groups to add as reviewers

Work Items  4

Search work items by ID or title

 675979 Implementación Pipeline

Create 

Nota: Alguna ayuda adicional para la creación de **pull request** consulta la wiki en VSTS

<https://grupobancolombia.visualstudio.com/Vicepresidencia%20Servicios%20de%20Tecnolog%C3%ADa/wiki/wikis/Vicepresidencia%20Servicios%20de%20Tecnolog%C3%ADa/wiki?wikiVersion=G BwikiMaster&pagePath=%2FDevOps%2FConfiguration%20Management%2FC%C3%B3mo%20crear%20un%20Pull%20Request>

24. Para que el pull request se ejecute en su totalidad se deben cumplir las políticas definidas en las recomendaciones del paso anterior y verificadas en la imagen a continuación



Policies


Required



- ✓ 1 reviewer approved
- ✓ Work items linked
- ✓ All comments resolved


Status


✓ No validation required. Target branch [develop]

Work Items  

 699595 MC060218 Testware TaskGroup...

Reviewers  

 Juan Diego Palacio Restrepo Approved

 Luis Miguel Ramirez Jaramillo Approved

Labels

25. Una vez se cumplan las políticas del **pull request** se debe aprobar el mismo y finalmente **“Complete”** para ver reflejados los cambios en rama Mas_



26. Al dar clic en la opción anterior **“Complete”**, sobre el **“Pull Request”** se genera una ventana adicional en la cual se debe tener en cuenta las siguientes recomendaciones y pasos.

Complete pull request

Deleting the source branch when squashing is recommended

Merged PR 32421: Pull Request Dev_Chatbot_Bancolombia a Mas_Chatbot_Ba

Mediante Pull Request se integra el script de la línea Develop (Dev_Chatbot_Bancolombia) a la línea Master (Mas_Chatbot_Bancolombia) en su versión 1.0 para ser ejecutado en el Pipeline (AW1188_BOT_ControladorChatSofy_CER) correspondiente al ID VSTS (533376)

Related work items: #533376

☒ Complete linked work items after merging

☐ Delete Dev_Chatbot_Bancolombia after merging

☒ Squash changes when merging [Learn more](#)

[Complete merge](#) [Cancel](#)

☒ Complete linked work items after merging

Al seleccionar esta opción se completa la tarea o work ítem de VSTS asociado al pull request

☒ Squash changes when merging [Learn more](#)

En esta opción se lleva el histórico de cambios realizados en la rama Dev_ dado que se requiera en la rama Mas_ hacer gestiones entre versiones del script al ejecutar en pipeline.

Nota: No se recomienda eliminar el Branch Dev luego del **“merge”** que realiza el **pull request** ya que se continuará trabajando sobre el Branch a nivel Dev construcción de futuras versiones del script de pruebas.

27. Una vez se finaliza el proceso de **“pull request”** este deberá quedar en estado aprobado con la información correspondiente del proceso.

Nota: Si no se presentan conflictos entre las dos ramas (Dev y Mas) el **“pull request”** queda en estado completado, de lo contrario se debe tener en cuenta que, si se generan conflictos entre las dos ramas, el **“pull request”** solicitará primero que sean resueltos para ser completado.



28. Por último, se verifica en el Branch Mas_ que queden integrados los cambios realizados mediante el pull request.

The screenshot shows a GitHub repository named 'Mas_Chatbot_Juridico'. The left sidebar displays the file structure with folders 'Front/Chatbot_Juridico' and 'DevOps_PESP_SilkPerformerScripts', and files '.gitignore' and 'README.md'. The main area shows the 'Contents' tab with a table of files and their commit history.

Name	Last change	Commits
Front/Chatbot_Juridico	2 minutes ago	ab7800ce
.gitignore	18/01/2019	c2720717
README.md	19/06/2018	d68d4c2a

Additional information visible: Merged PR 31454: Created by Luis Miguel Ramirez Jaramillo. Pull Request Dev_Chatbot_Juridico a Mas_Ch - Luis Miguel Ramirez Jaramillo.

Nota: Los pull request que se generen quedan asociados a la tarea “implementación pipeline”.

The screenshot shows a Jira task page for 'Implementación Pipeline' (Task ID: 675979). The task is in the 'Closed' state, assigned to 'Luis Miguel Ramirez Jaramillo', and is part of the 'Vicepresidencia Servicios de Tecnología' area. The 'Description' field is empty. The 'Planning' section shows 'Subdominio PRUEBAS ESPECIALIZADAS'. The 'Effort' section shows 'Estimación(opcional) 1'. The 'Development' section shows a link to 'Pull Request Dev_Chatbot_Juridico a Mas...' which is highlighted with a red box. The 'Related Work' section shows a link to '675672 HA PMO27171 STI 22724 - Ejecuc...'.

29. Configuración Pipeline DevOps para pruebas especializadas en RM de proyectos.

A continuación, se muestra la configuración del pipeline de DevOps para que los scripts se ejecuten desde el Testware de Pruebas Especializadas.

Performance E2E Chatbot Juridico
Deployment process

PRY_ChatbotJuridico
Run on agent

LineaBase
Pruebas de Performance con SilkPerf...

Display name * 1
LineaBase

Branch Master * 2
Ej: Mas_Aplicación_Testware

Capa Ejecución * 3
Ej: Frontend / Middleware / Backend

Frecuencia del Job * 4
30

Nombre Aplicativo TestWare * 5
Ej: Aplicacion_Testware

Nombre del Proyecto Silk * 6
PRY_ChatbotJuridico

Nombre del WorkLoad Silk * 7
LineaBase

Repository * 8
grupobancolombia.visualstudio.com/Vicepresidencia%20Servicios%20de%20Tecnolog%C3%ADa/_git/DevOps_PESP_SilkPerformerScripts

Paso 1

Performance E2E Chatbot Juridico
Deployment process

PRY_ChatbotJuridico
Run on agent

Linea Base
Pruebas de Performance con SilkPerformer

Display name *
Linea Base

La tarea de ejecución que se está configurando en el campo “Display name” debe tener el nombre correspondiente al escenario de carga a ejecutar.

Paso 2

GrupoBancolombia / Vicepresidencia Servicios de... / Files / DevOps_PESP_SilkPerformerScripts

Mas_Chatbot_Juridico

DevOps_PESP_SilkPerformerScripts / Type to find a file or folder...

DevOps_PESP_SilkPerformerScripts

Front/Chatbot_Juridico

.gitignore

README.md

Branch Master * 1
Mas_Chatbot_Juridico

Nombre branch a Nivel Master desde la cual se ejecutará la prueba

En este paso se debe configurar el nombre del Branch a nivel Máster desde el cual se ejecutará la prueba en la última versión de script desde Pipeline de DevOps, en este caso se indicó “Mas_Chatbot_Juridico”

Paso 3

DevOps_PESP_SilkPerformerScripts / Type to find a file or folder...

DevOps_PESP_SilkPerformerScripts

- Front/Chatbot_Juridico
 - .gitignore
 - README.md

Capa Ejecución *

Front

Capa desde la cual se ejecuta la prueba para la aplicación

Se debe indicar la capa desde la cual se ejecuta la prueba, esta corresponde a la primera carpeta creada en el Branch y contiene la carpeta de aplicación Testware.

Paso 4

Frecuencia del Job de ejecución en el agente de pruebas, debe ser de 30 segundos para cada tarea.

Frecuencia del Job *

Frecuencia del Job en segundos. (Recomendado >10)

30

Paso 5

En este paso se debe indicar el nombre del aplicativo como se nombró en el archivo de inventario Testware en el campo "Nombre Aplicativo TestWare" y también como aparece en la carpeta posterior a la capa de inyección.

ID	PMO	Aplicativo	Aplicativo MAC	Nombre Aplicativo TestWare
237	PMO27687	ChatBot Jurídico	Chatbot Jurídico	Chatbot_Juridico

DevOps_PESP_SilkPerformerScripts / Type to find a file or folder...

DevOps_PESP_SilkPerformerScripts

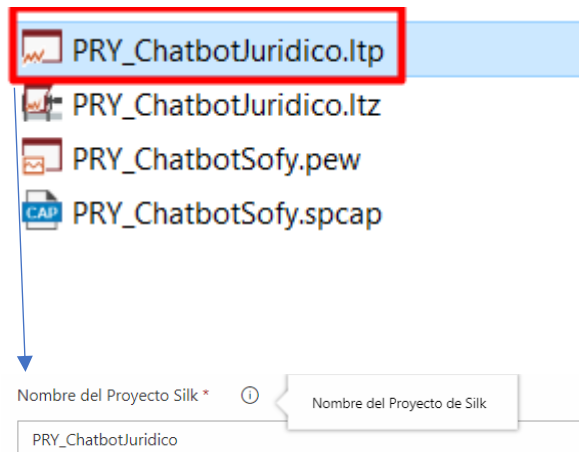
- Front/Chatbot_Juridico
 - .gitignore
 - README.md

Nombre Aplicativo TestWare *

Chatbot_Juridico

Nombre de aplicación según estandarización archivo inventario Testware

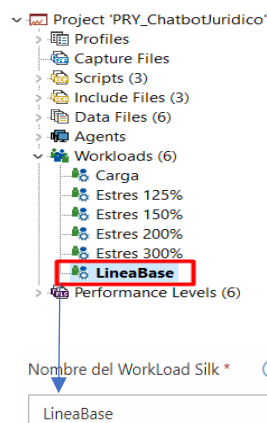
Paso 6



En el campo Nombre del Proyecto Silk, se debe indicar el nombre del proyecto en Silk como aparece el archivo de proyecto con extensión **.ltp** (la extensión no se incluye solo el nombre).

Nota: Este proyecto es pre-existente por eso empieza por PRY, se recomienda solo nombrar los proyectos nuevos en Silk con el del campo del archivo inventario Testware “Nombre de aplicación Testware”.

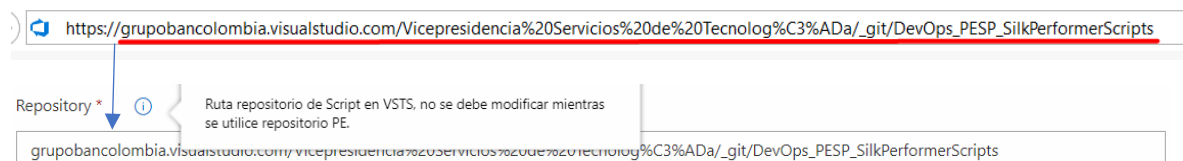
Paso 7



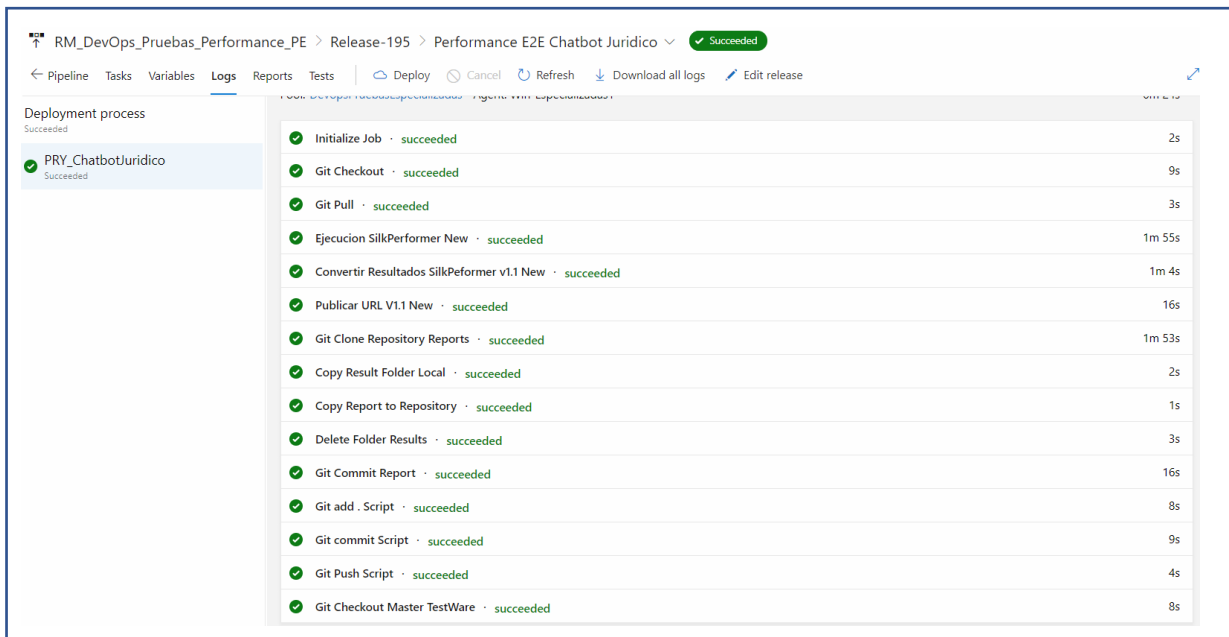
El campo “Nombre del WorkLoad Silk” se debe indicar el nombre del Workload como está configurado en el script para este caso “LineaBase”, este paso se debe realizar para todos los workload en cada tarea del (task) del pipeline.

Paso 8

El campo “Repository” está configurado por defecto con la URL del Repositorio de Pruebas Especializadas en VSTS, no se debe modificar, se diseñó el campo para cuando los scripts de pruebas pertenezcan a un repositorio diferente en cada proyecto.

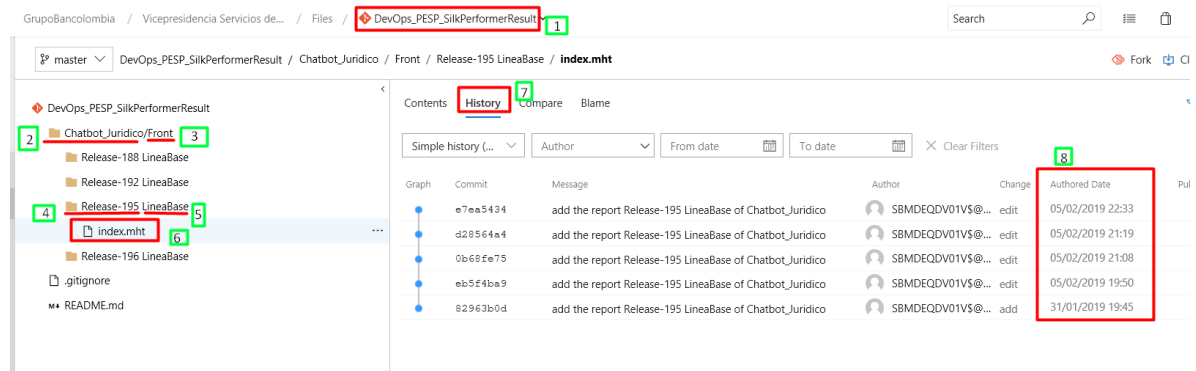


30. Una vez completada la configuración de los pasos en el Pipeline correspondiente al RM de cada proyecto, en el Log de ejecución del Stage Performance E2E las tareas configuradas debe quedar ejecutadas de forma exitosa.



31. Una vez se ejecuta el script de pruebas mediante Pipeline los resultados quedarán almacenados en el siguiente repositorio.

Nota: El proceso de llevar los resultados al repositorio **“DevOps_PESP_SilkPerformerResult”** se realiza de forma automática con Git una vez finaliza la ejecución del script en Pipeline, para este proceso no se debe realizar ninguna configuración adicional a las vistas en los pasos anteriores.

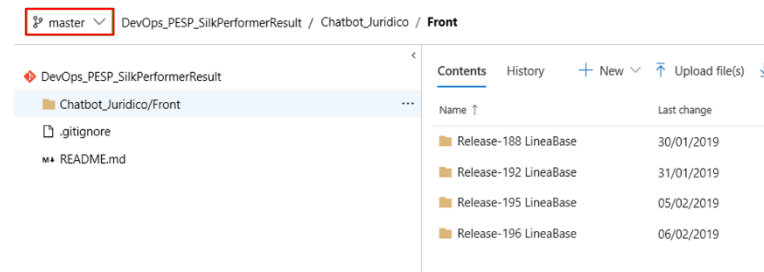


Notas: El repositorio **“DevOps_PESP_SilkPerformerResult”** es aparte al repositorio donde se almacenan los scripts (**DevOps_PESP_SilkPerformerScripts**), configurado para que se almacenen allí los resultados de las pruebas pipeline Devops ejecutado de forma automática.

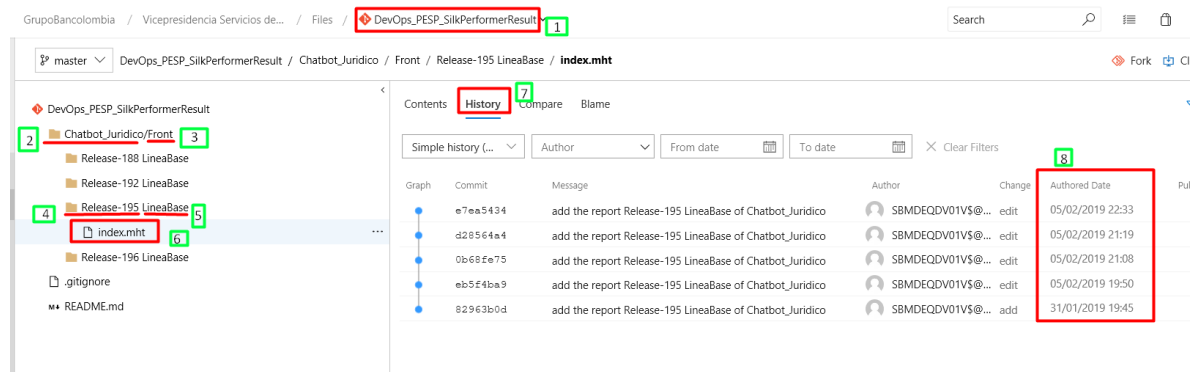
Url repositorio:

https://grupobancolombia.visualstudio.com/Vicepresidencia%20Servicios%20de%20Tecnolog%C3%ADa/_git/DevOps_PESP_SilkPerformerResult

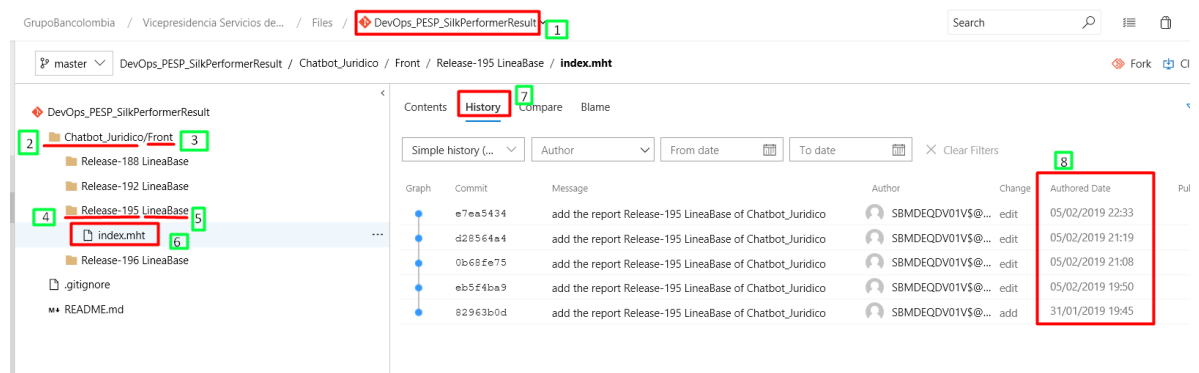
Los resultados se almacenan en la rama **master**



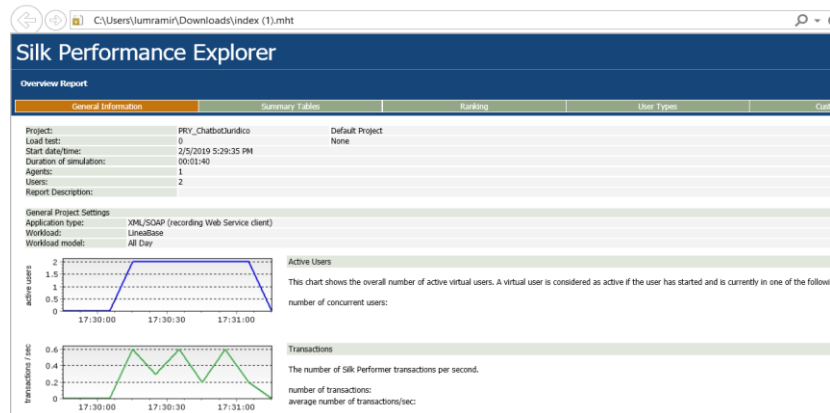
Indicación (2-3): La carpeta de resultados están distribuidas por “Nombre Aplicación Testware”(2) y “Capa de Ejecución”(3)



Indicación (3-4): En la carpeta de “Capa de Ejecución”(3) quedarán almacenadas las subcarpetas correspondientes al **Release Id (4)** de cada proyecto donde se configure el Stage de ejecución, para este caso se realizó el ejercicio en el “**RM_DevOps_Puebas_Performance_PE**” en el Release-195.



Indicación (5-6): En el interior de la carpeta **Release Id Worload(5)** se encuentra el archivo ***index.mht*** (6), el cual si abre por medio del navegador web de modo local visualiza los resultados de la ejecución.



Indicación (7-8): Para los casos en los cuales se realicen varias ejecuciones del mismo Release Id y Worload, en el archivo ***index.mht*** se pueden revisar los diferentes resultados de ejecución versionados por fecha de ejecución.

