

MANUAL DE MODIFICACIÓN DE DIAGRAMAS

A medida que el desarrollo del software avanza, los diferentes requerimientos presentes en la etapa inicial comienzan a adaptarse en razón a los resultados o necesidades del cliente, por eso, en el presente manual se indican los pasos a seguir para realizar la modificación de los diagramas de diseño de software, cuando el código vaya más allá de lo supuesto.

Requisitos:

- Tener instalado Visual Paradigm 15.0 o superior
- Tener instalado sistema de control de versiones Git
- Tener una carpeta creada y designada para trabajar con el repositorio.
- Mantener credenciales vigentes del repositorio web con archivos de diseño (GitHub).
- Software que permita acceso al repositorio web.
 - En este manual utilizaremos Git Bash como medio de acceso al repositorio web.
 - En este manual trabajaremos sobre la rama principal del repositorio.

PASOS A SEGUIR:

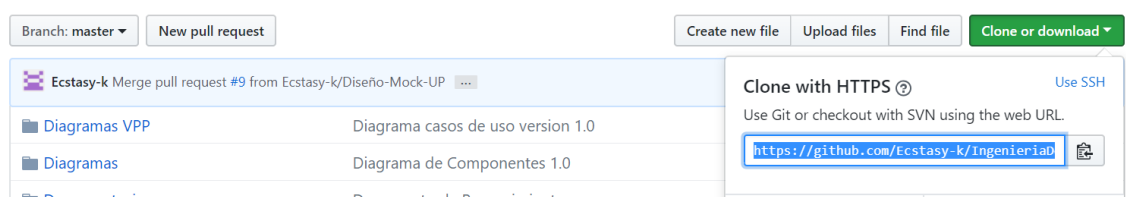
- 1- Mediante Git Bash debemos conectar nuestra carpeta previamente creada al repositorio en GitHub.

- Seleccionamos la carpeta mediante comandos

```
jorge@LAPTOP-ASKPQC2R MINGW64 ~
$ cd Documents/ManualDiagramas/

jorge@LAPTOP-ASKPQC2R MINGW64 ~/Documents/ManualDiagramas
$
```

- Copiamos el acceso al repositorio directo desde GitHub



- Clonamos el contenido del repositorio web, ejecutando los siguientes comandos

```
jorge@LAPTOP-ASKPQC2R MINGW64 ~/Documents/ManualDiagramas
$ git clone https://github.com/Ecstasy-k/IngenieriaDeSoftwareMII.git
Cloning into 'IngenieriaDeSoftwareMII'...
remote: Enumerating objects: 97, done.
remote: Counting objects: 100% (97/97), done.
remote: Compressing objects: 100% (74/74), done.
remote: Total 224 (delta 53), reused 42 (delta 23), pack-reused 127
Receiving objects: 100% (224/224), 81.66 MiB | 3.88 MiB/s, done.
Resolving deltas: 100% (100/100), done.
```

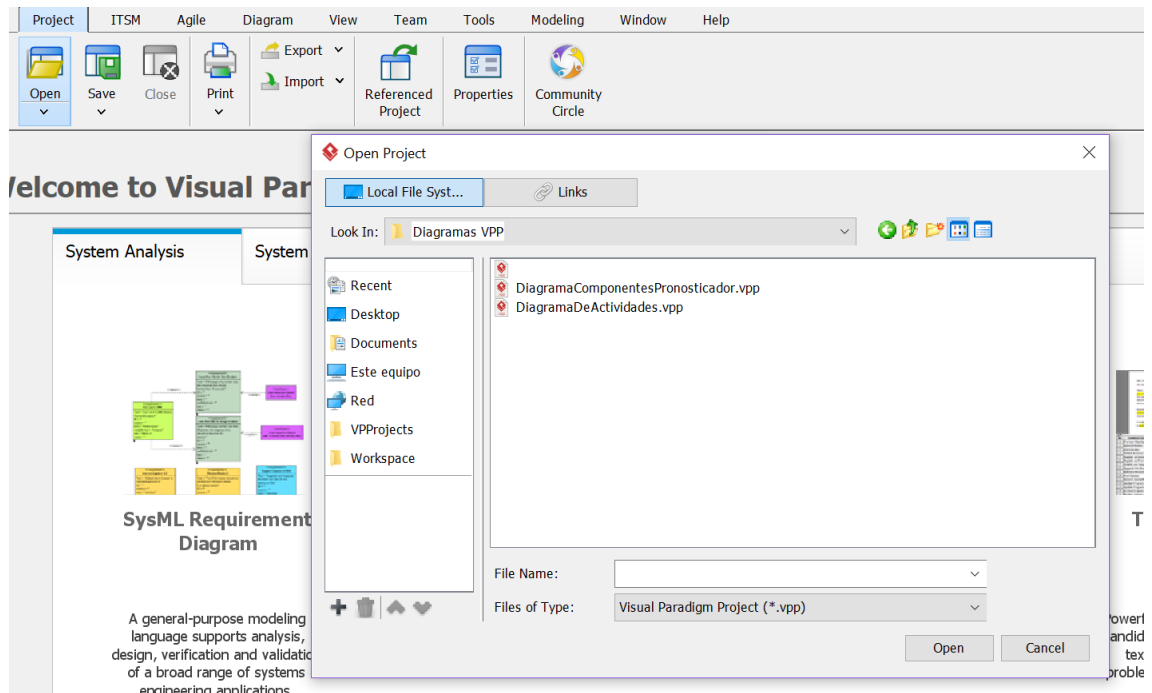
- Ingresamos a la carpeta del repositorio y comprobamos el acceso (pull y push).

```
jorge@LAPTOP-ASKPQC2R MINGW64 ~/Documents/ManualDiagramas
$ cd IngenieriaDeSoftwareMII/

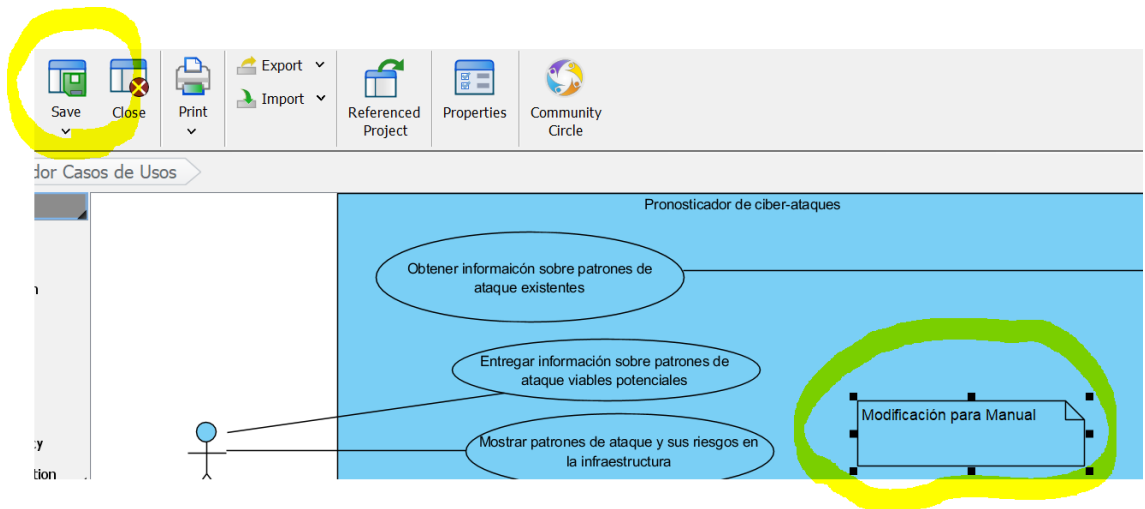
jorge@LAPTOP-ASKPQC2R MINGW64 ~/Documents/ManualDiagramas/IngenieriaDeSoftwareMII
(master)
$ git remote -v
origin https://github.com/Ecstasy-k/IngenieriaDeSoftwareMII.git (fetch)
origin https://github.com/Ecstasy-k/IngenieriaDeSoftwareMII.git (push)
```

2- Mediante Visual Paradigm, abrimos el archivo “.VPP” que queramos modificar, dentro de nuestra carpeta repositorio, modificamos y guardamos.

- Abrimos el archivo .VPP



- Realizamos la modificación, guardamos el archivo y cerramos Visual Paradigm.



3- Cargamos el archivo de vuelta al repositorio web, con las modificaciones guardadas.

- Añadimos el o los archivos “.VPP” modificados a los archivos bajo seguimiento (tracked).

```
jorge@LAPTOP-ASKPQC2R MINGW64 ~/Documents/ManualDiagramas/IngenieriaDeSoftwareMI
I (master)
$ git add Diagramas\ VPP/DiagramaCasosDeUsoPronosticador.vpp
```

- Con git status, podemos comprobar que el archivo ha sido “trackeado” exitosamente, además, podemos observar 2 archivos adicionales (.vpp.bak_000f) que no han sido “trackeados”, los cuales no son necesarios para el posterior uso y modificación del “.VPP”, por lo cual, deben ser ignorados.

```
jorge@LAPTOP-ASKPQC2R MINGW64 ~/Documents/ManualDiagramas/IngenieriaDeSoftwareMI
I (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   Diagramas VPP/DiagramaCasosDeUsoPronosticador.vpp

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Diagramas VPP/DiagramaCasosDeUsoPronosticador.vpp.bak_000f
        Diagramas VPP/DiagramaCasosDeUsoPronosticador.vpp.bak_001d
```

- Luego de realizar la comprobación, procedemos a ejecutar el commit de los archivos añadidos, es necesario agregar un comentario adecuado para mantener el buen registro de modificaciones. Señala la rama en la cual realizamos la modificación (Master).

```
jorge@LAPTOP-ASKPQC2R MINGW64 ~/Documents/ManualDiagramas/IngenieriaDeSoftwareMI
I (master)
$ git commit -m "Modificando Casos de Uso para el MANUAL"
[master 1ab2072] Modificando Casos de Uso para el MANUAL
1 file changed, 0 insertions(+), 0 deletions(-)
```

- Posteriormente necesitamos reflejar los cambios realizados en nuestro repositorio, en el repositorio web, para ello debemos “empujar” los archivos contenidos en este mediante el comando PUSH.

```
jorge@LAPTOP-ASKPQC2R MINGW64 ~/Documents/ManualDiagramas/IngenieriaDeSoftwareMI
I (master)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 12.18 KiB | 779.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0)
To https://github.com/Ecstasy-k/IngenieriaDeSoftwareMII.git 1 objects.
38aedd6..1ab2072 master -> master
```

- Finalmente, la pantalla indicará que el proceso ha terminado con éxito y los archivos deberán reflejarse en el repositorio web, incluyendo la modificación y el comentario añadido en el commit.

Branch: master ▾
New pull request
Create new file
Upload files
Find file
Clone or download ▾

Ecstasy-k Modificando Casos de Uso para el MANUAL Latest commit 1ab2072 6 minutes ago

Diagramas VPP	Modificando Casos de Uso para el MANUAL	6 minutes ago
Diagramas	Diagrama de Componentes 1.0	a month ago

Branch: master ▾
IngenieriaDeSoftwareMII / Diagramas VPP /
Create new file
Upload files
Find file
History

Ecstasy-k Modificando Casos de Uso para el MANUAL Latest commit 1ab2072 7 minutes ago

..		
DiagramaCasosDeUsoPronosticador...	Modificando Casos de Uso para el MANUAL	7 minutes ago
DiagramaComponentesPronosticad...	Diagrama de actividades, agregando leyenda al control Flow	3 days ago
DiagramaDeActividades.vpp	Diagrama de actividades, Version 2(Agregando leyenda Control Flow)	3 days ago