

Manual de Usuario - Solver_Doctorado-2

Tomás Cornejo Zúñiga
Sebastián Montiel Astudillo

Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
`tomas.cornejo.z@mail.pucv.cl`
`sebastian.montiel.a@mail.pucv.cl`

Índice

1. Dependencias	2
1.1 Activar el entorno virtual	2
1.2 Instalar las dependencias	3
1.3 Actualizar las dependencias	3
2 Ramas	5
2.1 Ramas generales	5
2.2 Ramas principales	6
2.3 Resumen de comandos principales	7
2.4 Crear un nuevo repositorio	9
2.5 Mandar enlace por rama	10

El presente manual hace referencia al código disponible en el repositorio <https://github.com/FelipeCisternasCaneo/Solver-Doctorado-version-2.git>

1. Dependencias

Se emplea la versión 3.11.7 de Python en su variante de 64 bits.

Para gestionar las dependencias del proyecto, se recurre a la librería "virtualenv", la cual debe ser instalada mediante el siguiente comando en la consola:

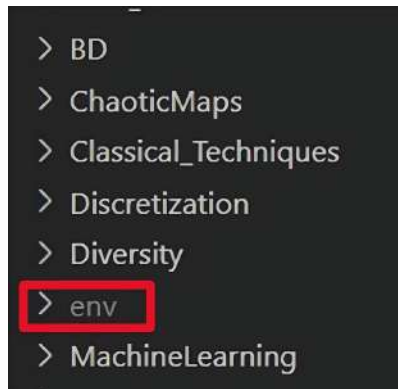
```
pip install virtualenv
```

1.1 Activar el entorno virtual

"virtualenv" es una herramienta de Python que posibilita la creación de entornos virtuales aislados para proyectos específicos. Para utilizarla, es necesario inicialmente crear la carpeta denominada ".env", la cual almacenará localmente en el proyecto las dependencias necesarias. Este proceso se lleva a cabo mediante el siguiente comando:

```
virtualenv -p python3 env
```

Quedando así:



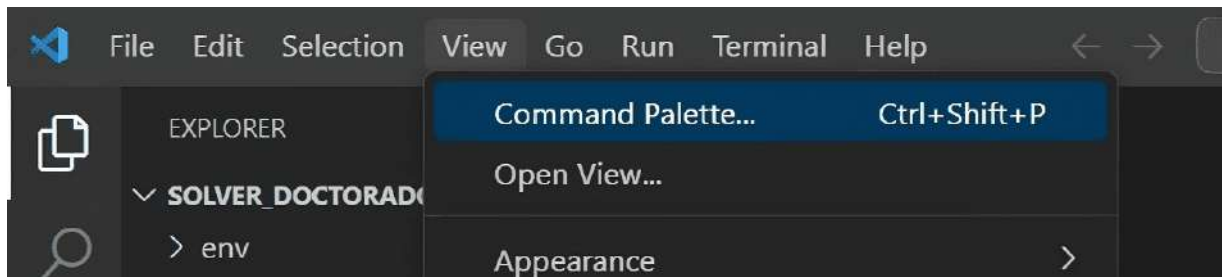
Una vez que la carpeta ".env" esté instalada, es necesario activarla mediante el siguiente comando en la consola:

```
.\env\Scripts\activate
```

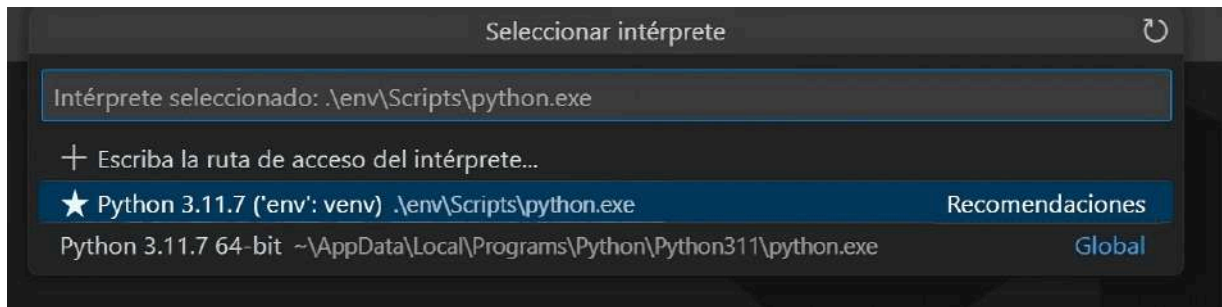
Desde ahora la consola debería verse así:



Para garantizar que Visual Studio Code esté utilizando el intérprete correcto, es necesario ir a la esquina superior izquierda, hacer clic en "View" y luego, en el menú desplegable de "View", seleccionar "Command Palette".



Dentro de la "Command Palette", escriba ">Select Interpreter", presione Enter y revise o elija la opción que incluya ".env"; la opción con una estrella es la actualmente seleccionada.



1.2 Instalar las dependencias

Para instalar las dependencias actuales del proyecto, utilice el siguiente comando en la consola. Asegúrese de tener activado el entorno virtual, consulte la sección *1.1 Activar el entorno virtual*.

```
pip install -r requirements.txt
```

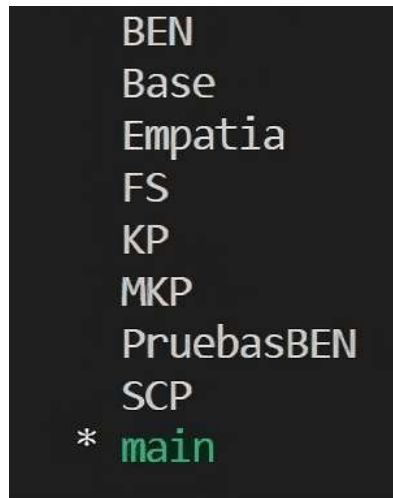
1.3 Actualizar las dependencias

Para actualizar las dependencias, simplemente instálelas con el entorno virtual activado dentro del proyecto. Consulte la sección *1.1 Activar el entorno virtual* para realizar la instalación. Una vez esté seguro de que no hay problemas con las dependencias, puede actualizar el archivo "requirements.txt" utilizando el siguiente comando en la consola:

```
pip freeze > requirements.txt
```

2 Ramas

Para modularizar el repositorio se tiene una división de ramas con la siguiente estructura:

Un diagrama de estructura de ramas de Git. Se muestran las siguientes ramas: BEN, Base, Empatía, FS, KP, MKP, PruebasBEN, SCP, y *main. La rama 'main' está resaltada en verde y precedida por un asterisco, lo que indica que es la rama activa o la rama por defecto.

```
BEN
Base
Empatía
FS
KP
MKP
PruebasBEN
SCP
*main
```

Ramas

2.1 Ramas generales

1. **main:** En esta rama, todos los códigos están consolidados en caso de que se necesiten. Dado que los códigos están agrupados, no resulta adecuada para enviar información de una rama a otra mediante el comando de transferencia de modificaciones. No obstante, esta rama sí tiene la capacidad de recibir modificaciones provenientes de otras ramas.
2. **Base:** En esta rama, se cargarán las metaheurísticas, algoritmos de aprendizaje automático, y otras carpetas que serán compartidas por todas las ramas. El propósito de esta rama es servir como la estructura principal para las demás. Cuando se desee transferir información de esta rama a otras, se debe realizar un commit con los cambios, cambiar a la rama receptora de la base y transferir los cambios. Es crucial tener en cuenta que esta rama base no puede recibir información de otras ramas mediante comandos, ya que esto alteraría su estructura.

- ✓ Solver_Doctorado
 - > ChaoticMaps
 - > Classical_Techniques
 - > Discretization
 - > Diversity
 - > MachineLearning
 - > Metaheuristics
 - > Resultados
 - > util
- ◆ .gitignore
- ≡ requirements.txt

Estructura de las carpetas de rama base

```
git add .
git commit -m "comentario"
git checkout ramaReceptora
git checkout Base -- .
```

3. **PruebasBEN:** Esta rama está destinada a la prueba de metaheurísticas con problemas continuos sin afectar la integridad de la rama BEN original. En caso de que la prueba sea exitosa, se recomienda transferir manualmente la metaheurística a la rama Base para facilitar su implementación en otras ramas. En el escenario en el que la rama PruebasBEN se rompa o se desee repetir el experimento, se aconseja regresar a una versión anterior utilizando los comandos 8 y 9 detallados en la sección 2.3 *Resumen de comandos principales*.

2.2 Ramas principales

1. **BEN:** Rama benchmark.
2. **SCP:** Rama Set covering Problem.
3. **KP:** Rama problema de la mochila.
4. **MKP:** Rama problema mochila multidimensional.
5. **FS:** Rama de selección de características.
6. **Empatia:** Rama de FS empatía.

2.3 Resumen de comandos principales

Los principales comandos para el manejo de ramas son los siguientes:

1. Revisar ramas activas

```
git branch --all
```

2. Cambiar de rama

```
git checkout nombreRama
```

3. Guardar cambios

```
git add .  
git commit -m "comentario"
```

4. Transferir modificaciones de una rama a otra. Implica la adición y modificación de carpetas y archivos, sin eliminar ninguno. Es necesario ubicarse en la rama receptora de los cambios para llevar a cabo este proceso

```
git checkout nombreRama -- .
```

5. Crear una nueva rama

```
git branch nombreRama
```

6. Eliminar una rama. Asegúrese de estar seguro de hacer esto porque los cambios locales no se pueden recuperar

```
git branch -D nombreRama
```

7. Subir rama al repositorio online.

```
git push -u origin nombreRama
```

8. Revisar commits anteriores. Es útil para obtener la identificación de un commit que deseamos revertir (en la imagen adjunta, se destaca en rojo, mostrando el ID). Una vez que hayamos examinado los commits e identificado el que deseamos, debemos presionar la tecla "q" para cerrar la visualización.

```
git log
```



```
commit 9523d7983b90afd58188ae55373a5e152b1b85e1
Author: Tomas Cornejo <tomas.cornejo.z@gmail.pucv.cl>
Date: Tue Jan 16 15:15:06 2024 -0300

first commit base
```

Ejemplo de id de commit

9. **Volver a un commit anterior.** Con la id obtenida del comando anterior podemos volver a una versión anterior del código.

```
git reset --hard idCommit
```

2.4 Crear un nuevo repositorio

Si se desea crear un nuevo repositorio a partir de una rama del proyecto, se debe seguir el siguiente procedimiento:

1. Posicionarse en la rama que desea crear el repositorio y crear un commit de respaldo

```
git checkout nombreRama  
git add .  
git commit -m "comentario"
```

2.1 Para windows: Eliminamos la carpeta oculta del git

```
rd /s /q .git
```

2.2 Para Linux y macOS: Eliminamos la carpeta oculta del git

```
rm -rf .git
```

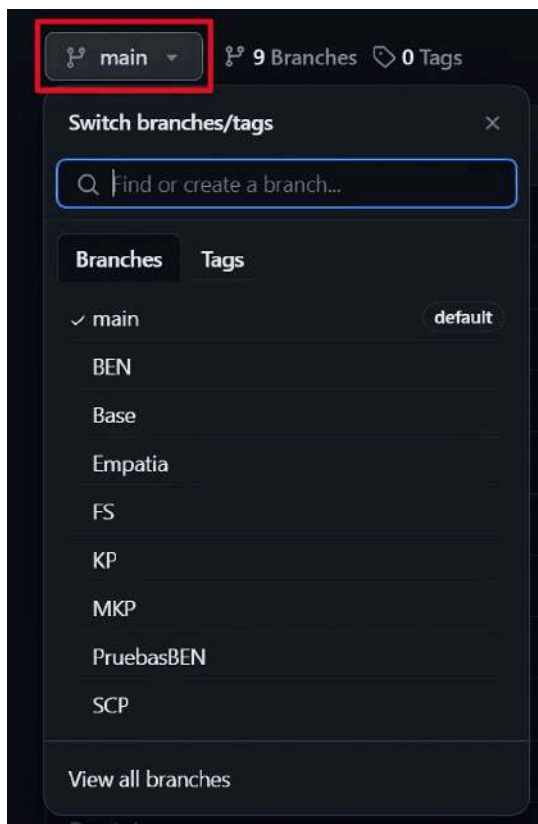
3. Se crea y guarda el nuevo repositorio

```
git init  
git add .  
git commit -m "First commit"  
git remote add origin URL_NUEVO_REPOSITORIO  
git push -u origin master/main
```

Hint: Hay veces que la rama principal se llama master y otras veces se llama main. Revisar con “git branch” cual es el caso.

2.5 Mandar enlace por rama

Cuando se desee compartir únicamente una rama específica del repositorio con otra persona, es necesario acceder al repositorio en GitHub. Desde allí, se debe seleccionar la opción desplegable correspondiente a la rama principal (main):



Desde ese punto, es necesario seleccionar la rama que se desea compartir. Una vez elegida, simplemente queda compartir el enlace del repositorio.

Enlace original:

<https://github.com/FelipeCisternasCaneo/Solver-Doctorado-version-2>

Enlace de la rama BEN:

<https://github.com/FelipeCisternasCaneo/Solver-Doctorado-version-2/tree/BEN>

Al compartir este enlace, se permitirá clonar el repositorio desde la rama específica. Sin embargo, es importante tener en cuenta que compartir este enlace no implica un acceso exclusivo a esa rama en particular. El repositorio completo será enviado, pero con la diferencia de que la rama enviada se establecerá como predeterminada en lugar de la habitual "main".