

Padronização de Softwares

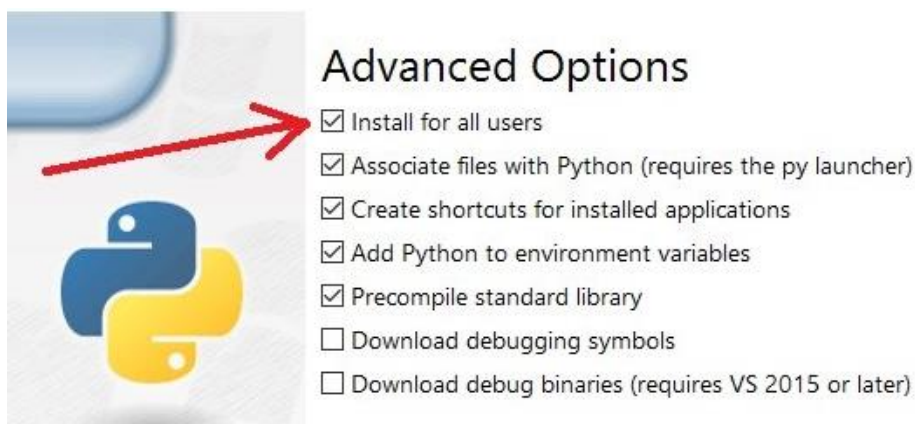
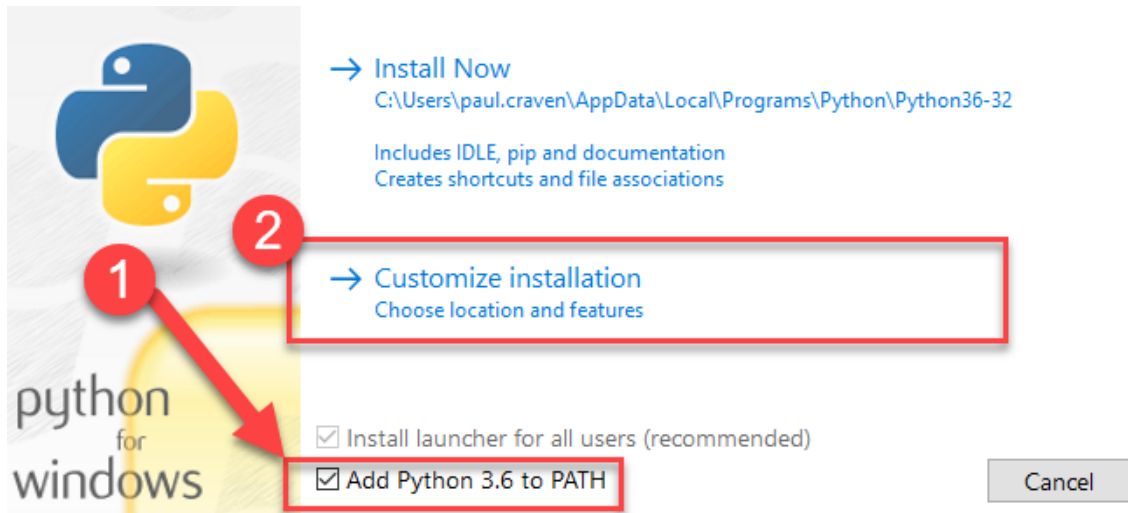
Configurando o computador (Windows)

Luana Vilela e Vitor Soares – 26/03/2020

Esse documento é apenas um guia para ajudar a configurar um computador com sistema operacional Windows com os softwares que adotamos como padrão para o grupo IMAS. Muitas das instruções descritas aqui podem não funcionar corretamente no seu computador. Para instalar alguns dos softwares é preciso ter permissão de administrador.

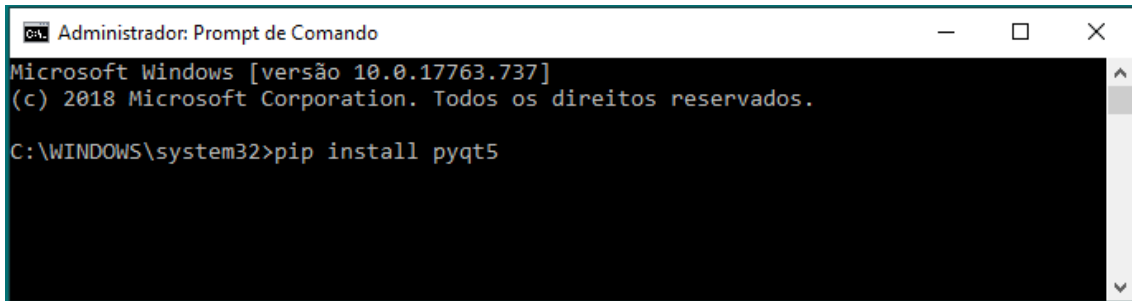
1. Python:

Se o Python não estiver instalado ainda no computador, baixe o instalador em <https://www.python.org/downloads/windows/>, de preferência a versão 3.7, e instale para todos os usuários, conforme mostrado nas figuras abaixo:



2. PyQt:

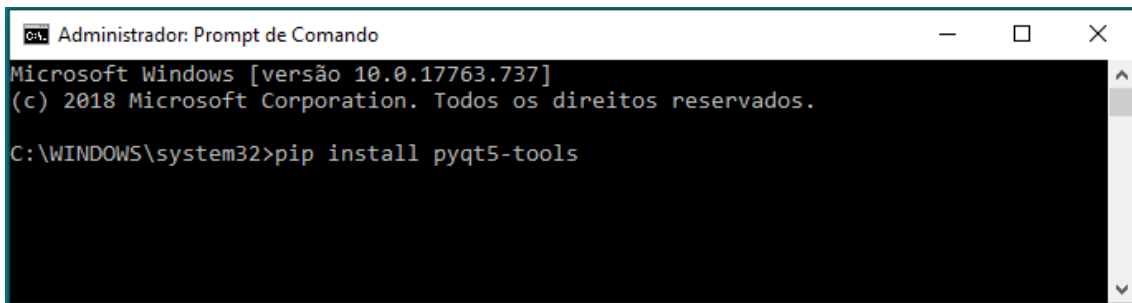
Se nenhuma versão do PyQt estiver instalada, abra o prompt de comando do Windows (cmd) no modo administrador e use o pip para instalar o PyQt5 (comando: ***pip install pyqt5***), como mostrado abaixo:

A screenshot of a Windows Command Prompt window titled "Administrador: Prompt de Comando". The window shows the text "Microsoft Windows [versão 10.0.17763.737]" and "(c) 2018 Microsoft Corporation. Todos os direitos reservados." followed by the command "C:\WINDOWS\system32>pip install pyqt5".

```
Administrador: Prompt de Comando
Microsoft Windows [versão 10.0.17763.737]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32>pip install pyqt5
```

Para poder utilizar ferramentas como o QtDesigner é preciso instalar também o pyqt5-tools (comando ***pip install pyqt5-tools***):

A screenshot of a Windows Command Prompt window titled "Administrador: Prompt de Comando". The window shows the text "Microsoft Windows [versão 10.0.17763.737]" and "(c) 2018 Microsoft Corporation. Todos os direitos reservados." followed by the command "C:\WINDOWS\system32>pip install pyqt5-tools".

```
Administrador: Prompt de Comando
Microsoft Windows [versão 10.0.17763.737]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32>pip install pyqt5-tools
```

Caso você não encontre o caminho para o QtDesigner após a instalação, execute o comando ***designer*** no prompt de comando para abrir o programa.

3. Git:

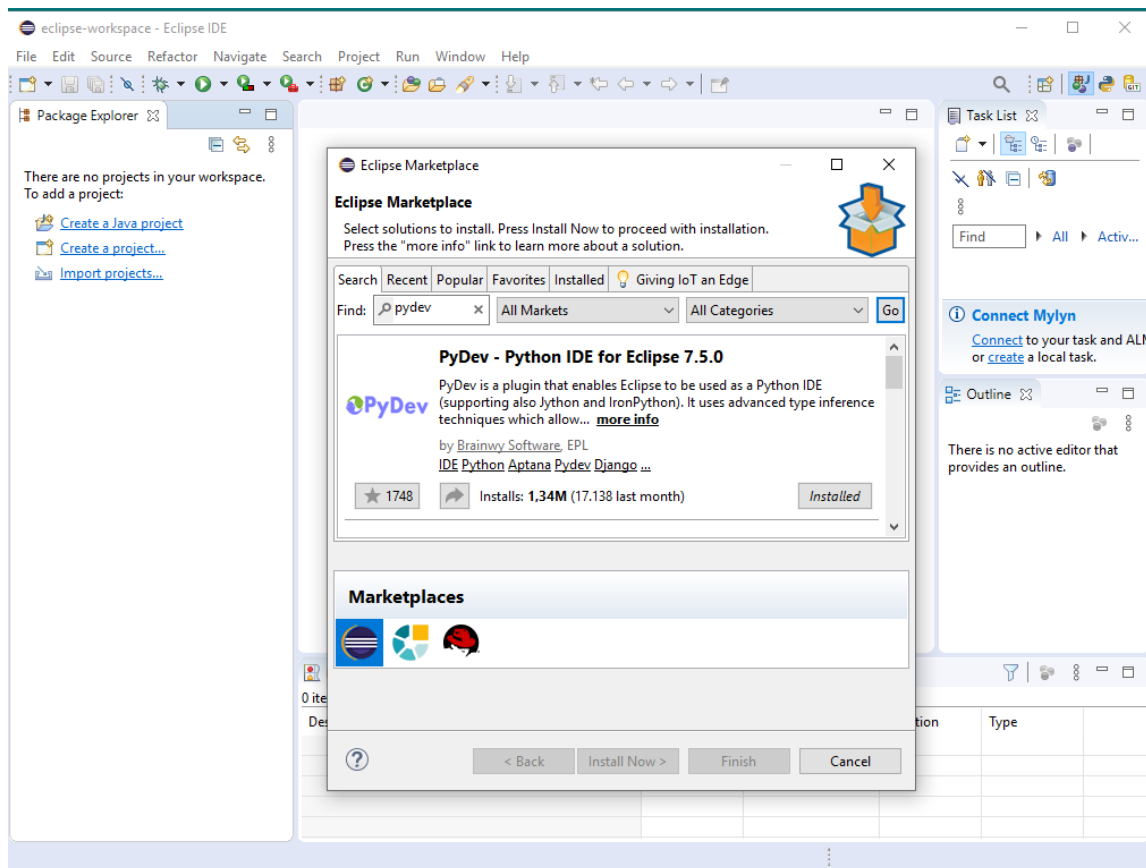
Se o Git não estiver instalado, baixe o instalador para Windows em <https://git-scm.com/downloads> e siga as instruções de instalação.

4. DB Browser:

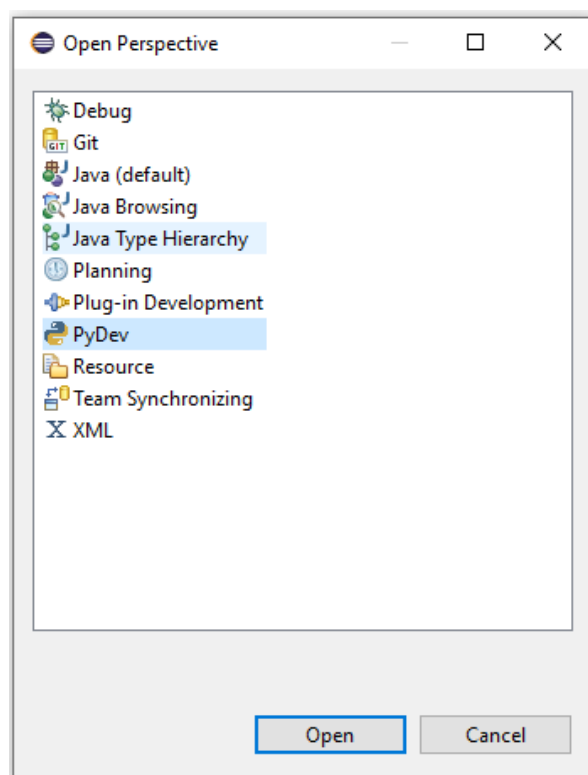
Se for utilizar banco de dados SQLite, baixe o instalador do DB Browser para Windows em <https://sqlitebrowser.org/dl/> e siga as instruções de instalação.

5. IDE:

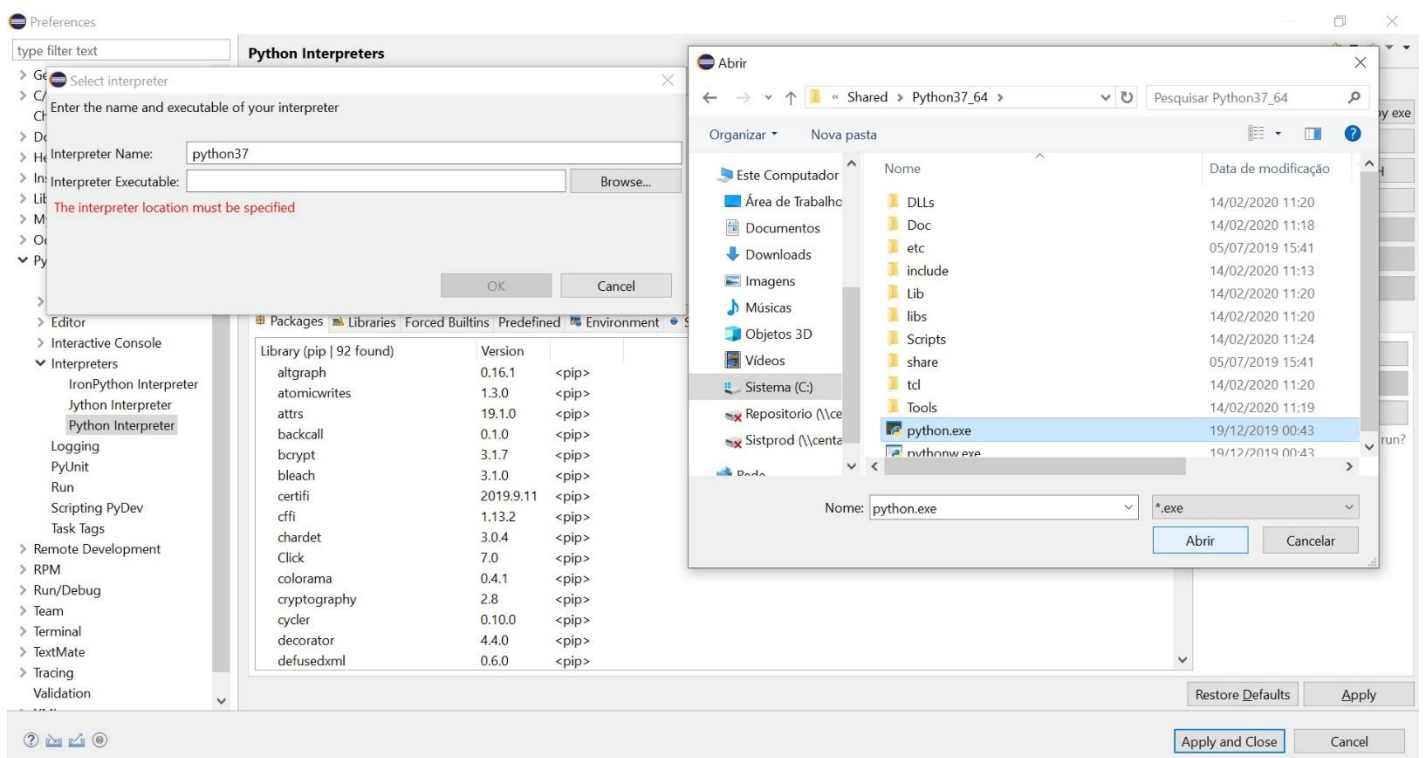
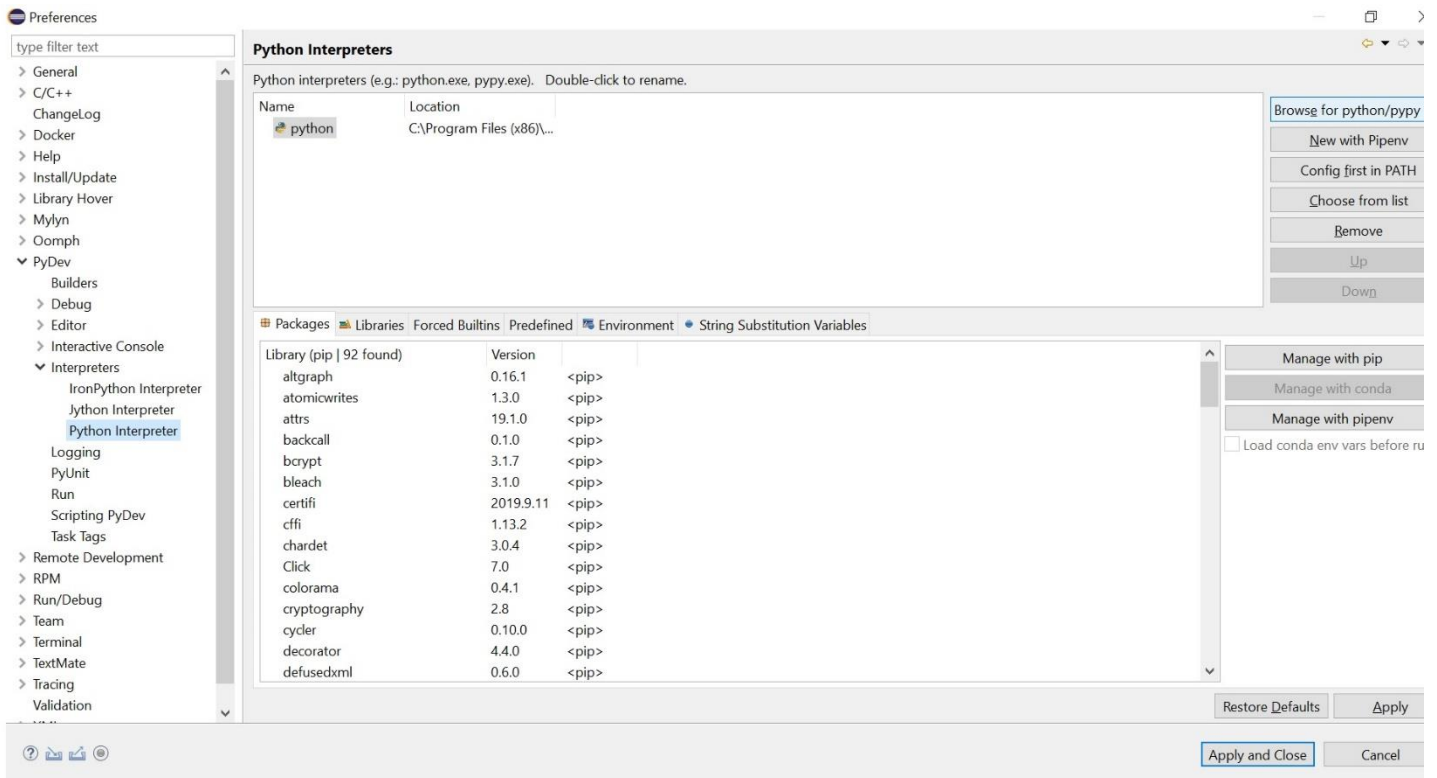
Recomendamos o usar do Eclipse como ambiente de desenvolvimento. Para instalá-lo baixe o instalador em <https://www.eclipse.org/downloads/> e instale a versão C/C++. Em seguida abra o Eclipse e clique no menu ***Help -> Eclipse Marketplace...***, procure por PyDev e siga as instruções de instalação.



Após a instalação do PyDev clique no menu **Window -> Perspective -> Open Perspective -> Other** e selecione a opção PyDev, como mostrado abaixo:

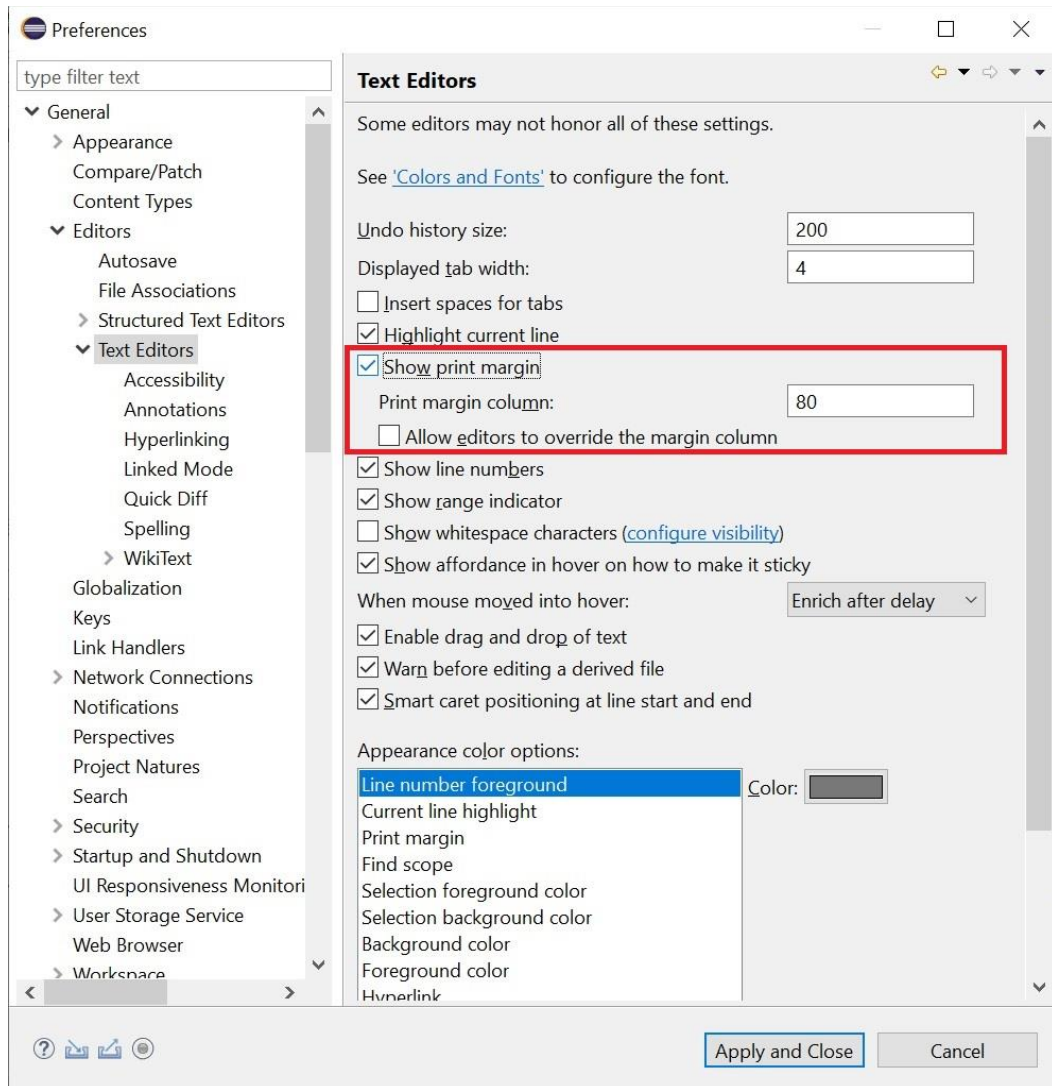


Em seguida é preciso indicar ao PyDev aonde encontrar o interpretador do Python. Isso é feito na janela aberta em **Window -> Preferences**. No novo menu selecione **PyDev -> Interpreters -> Python Interpreter**. No canto superior direito da tela clique em **"Browse for python/pypy exe"**, escolha um nome para o interpretador e selecione o arquivo executável na pasta de instalação do Python, conforme as figuras. Clique em "Ok" e espere enquanto o PyDev procura pelas bibliotecas disponíveis. No canto inferior direito, clique em **"Apply"** para gravar a configuração.

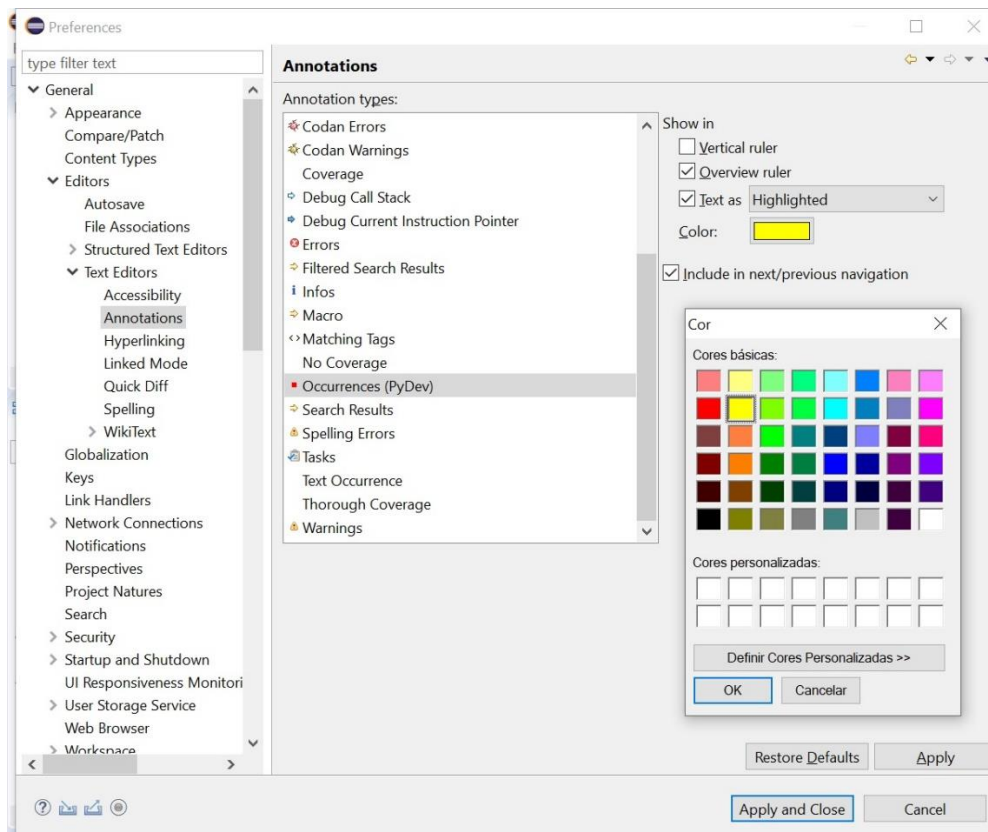


Outras configurações recomendadas a se fazer na janela **Window -> Preferences** são:

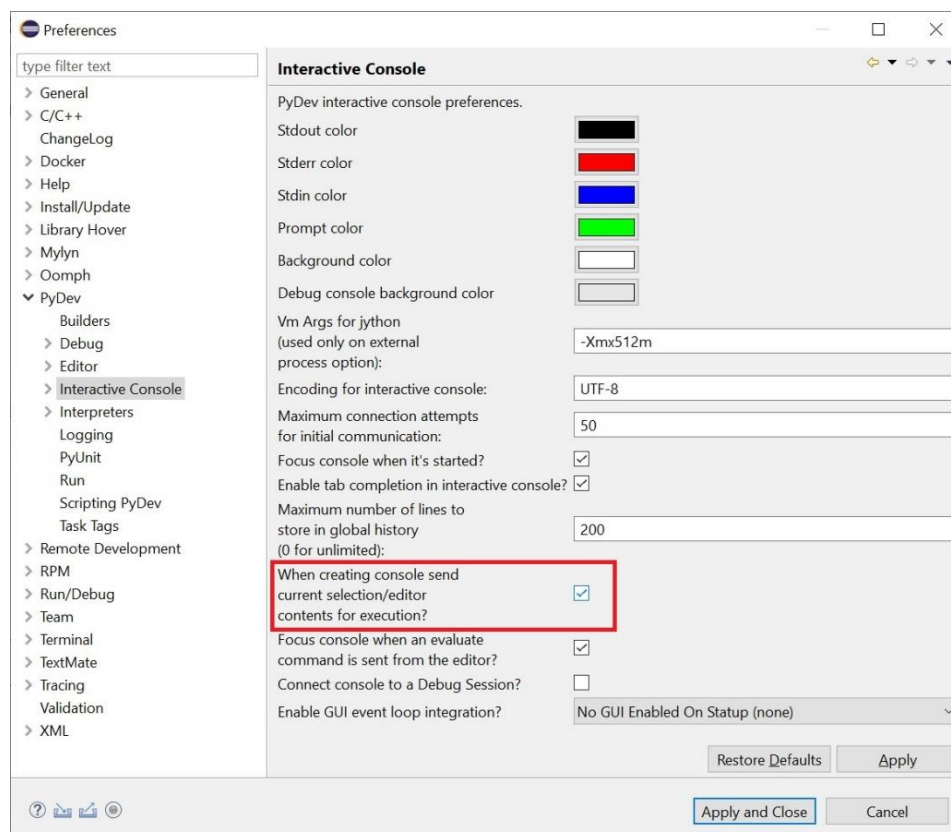
- Habilitar linha de margem no 80º caracter para facilitar respeitar o limite de caracteres do Python: entre no menu **General -> Editors -> Text Editors**, habilite a opção **"Show print margin"** e clique em **"Apply"**.



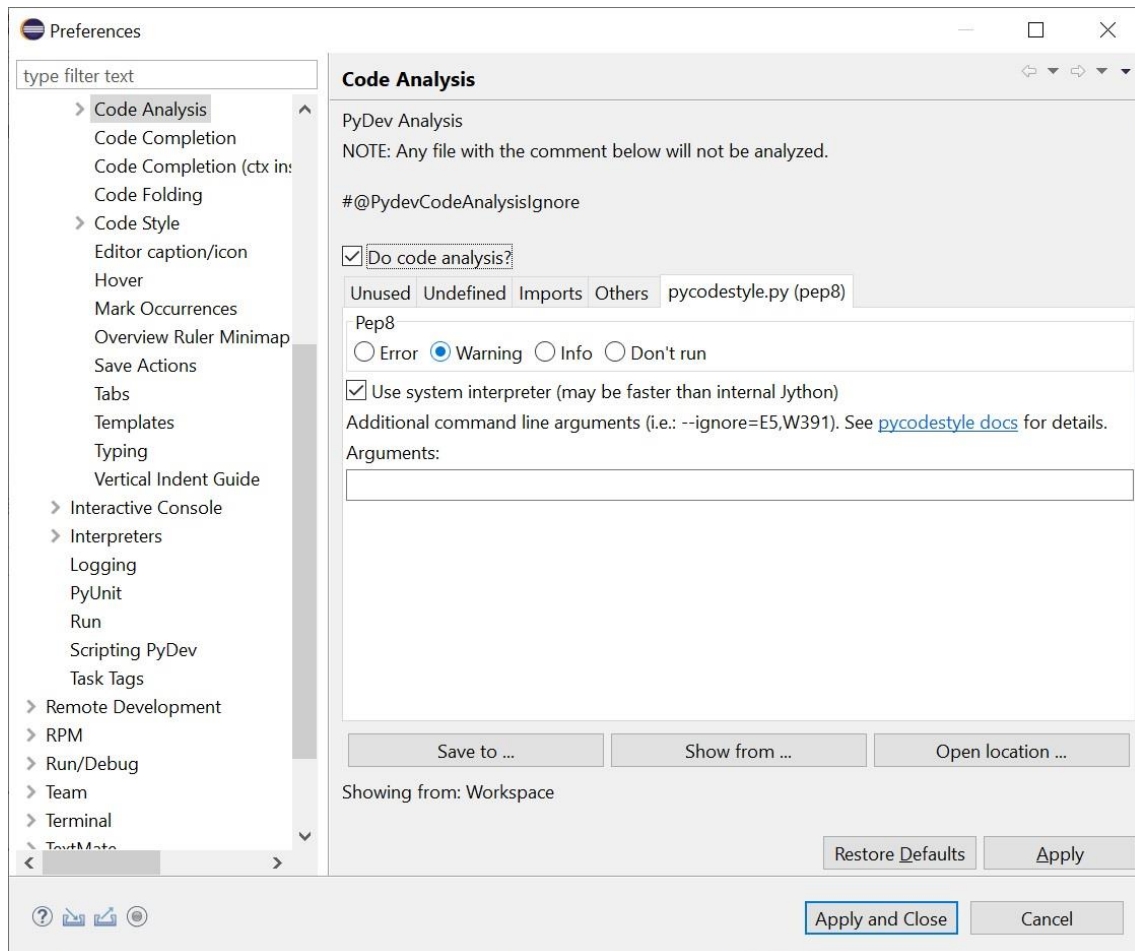
- Alterar a cor das marcações de ocorrência do PyDev para facilitar a visualização na barra lateral: entre no menu **General -> Editors -> Text Editors -> Annotations**, na lista **"Annotation types"** selecione **"Occurrences (PyDev)"**, clique na caixa **"Color"**, selecione segundo amarelo na segunda coluna da paleta de cores, clique em **"Ok"** e depois em **"Apply"**.



- Em **PyDev -> Interactive Console**, habilite a opção **"When creating console send current selection/editor contents for execution"** e clique em **"Apply"**. Isso faz com que não seja necessário clicar em **Ctrl + Alt + Enter** duas vezes quando for rodar um script no Eclipse.



- Ative a análise dos padrões Pep8 no código. Vá em **PyDev -> Editor -> Code Analysis**, habilite o checkbox "**Do code analysis**", vá na aba "**pycodestyle.py**", selecione "**Warning**" em Pep8 e clique em "**Apply**".



É necessário reiniciar o Eclipse para que algumas dessas configurações tenham efeito.

6. Diretório padrão:

Durante as discussões sobre padronização dos softwares foi definido um diretório padrão para os repositórios dos softwares desenvolvidos. Então recomenda-se a criação do seguinte caminho no seu computador, caso este ainda não exista: <C:\Arg\Work At LNLS\eclipse-workspace>. Se estiver utilizando o Eclipse como ambiente de desenvolvimento selecione este caminho como workspace.

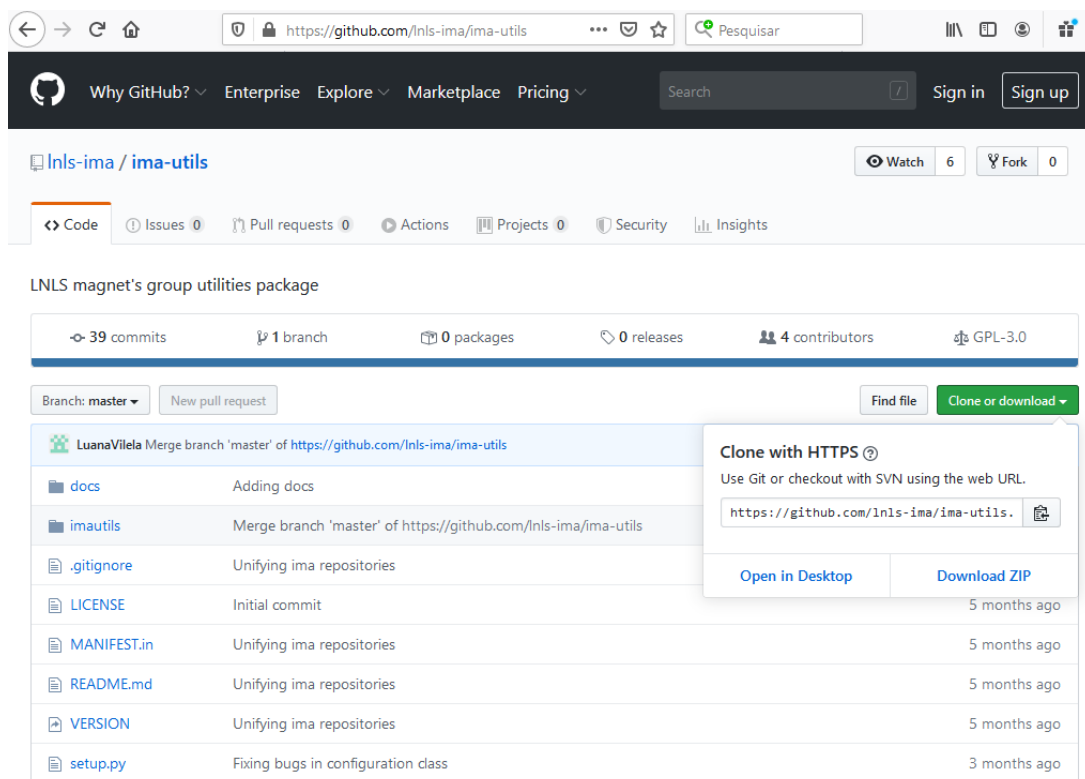
7. Clonando repositórios do GitHub:

Para clonar os repositórios do GitHub do grupo IMAS (<https://github.com/lnls-ima>) abra o prompt de comando (cmd) e vá até a pasta padrão citada anteriormente. Utilize o comando **git clone** com o caminho do repositório que deseja clonar. Veja abaixo o exemplo para o repositório ima-utils:

```
Prompt de Comando
Microsoft Windows [versão 10.0.17763.737]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

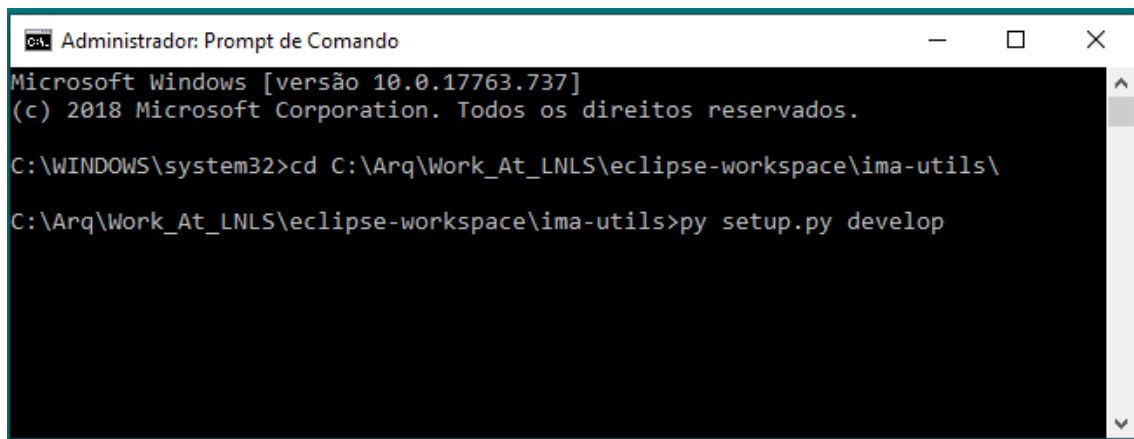
C:\Users\luana.vilela>cd C:\Arq\Work_At_LNLS\eclipse-workspace
C:\Arq\Work_At_LNLS\eclipse-workspace>git clone https://github.com/lnls-ima/ima-utils.git
```

O caminho de cada repositório pode ser encontrado pelo site do GitHub, clicando no botão “Clone or download”.



8. Instalando os pacotes:

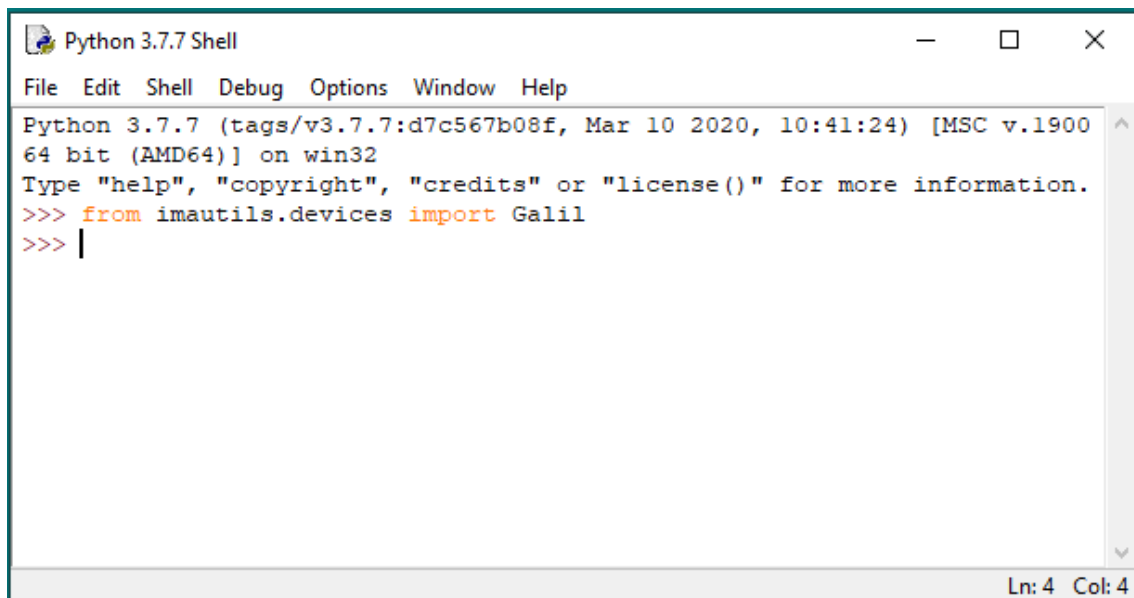
Definimos na padronização que cada repositório de software desenvolvido pelo grupo deve ser um pacote Python que pode ser instalado. Para instalar o pacote abra o prompt de comando no modo administrador, vá até a pasta do repositório e execute o comando ***py setup.py develop***. Esse comando irá tentar instalar o pacote e todas as dependências configuradas no arquivo setup.py.



```
Administrador: Prompt de Comando
Microsoft Windows [versão 10.0.17763.737]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32>cd C:\Arq\Work_At_LNLS\eclipse-workspace\ima-utils\
C:\Arq\Work_At_LNLS\eclipse-workspace\ima-utils>py setup.py develop
```

Após a instalação, o pacote fica disponível para ser importado em qualquer código, assim como outros pacotes do Python, como o numpy ou o matplotlib. Para testar a instalação tente importar o pacote, como mostrado abaixo utilizando o IDLE padrão do Python:



```
Python 3.7.7 Shell
File Edit Shell Debug Options Window Help
Python 3.7.7 (tags/v3.7.7:d7c567b08f, Mar 10 2020, 10:41:24) [MSC v.1900
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from imautils.devices import Galil
>>> |
Ln: 4 Col: 4
```

9. Pronto!

Agora é só fazer as alterações desejadas no software e incluir essas alterações no GitHub para que todos tenham acesso a nova versão. O seguinte tutorial contém algumas explicações básicas sobre a utilização do Git e do GitHub: https://github.com/lnls-ima/documentation/blob/master/2018-01-04_tutorial_git.pdf.