```
  GNU nano 4.8
import funcionesTarea

gato1 = funcionesTarea.gato()
<<<<<<< HEAD
gato1.nombre = "Oreo"
=======
gato1.nombre = "Avena"
>>>>>>> Cambio del nombre al gato1 desde main
gato1.color = "negro"
gato1.edad = 4
gato1.aniadirlista()

gato2 = funcionesTarea.gato()
gato2.nombre = "Pelusa"
gato2.color = "cafe"
gato2.edad = 3
gato2.aniadirlista()

gato3 = funcionesTarea.gato()
gato3.nombre = "Estrella"
gato3.color = "Blnaco"
gato3.edad = 6
gato3.aniadirlista()


funcionesTarea.printlista()
```

```
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ git log --oneline --graph
* 1fa353e (HEAD -> main, origin/main, origin/HEAD) Cambio del nombre al gato1 desde main
* 6158073 (origin/feature, feature) Se crea el objeto gato3
* 4ca823e Cambio del nombre a gato1 desde feature
* 1640128 included gato2 and list printing
* c2b877a corrected mistakes
* d02a6d2 included gato1
* 7ab0adc included functions file
* a13fd08 first commit
```

```
--graph
    Draw a text-based graphical representation of the commit history on the left hand side
    of the output. This may cause extra lines to be printed in between commits, in order
    for the graph history to be drawn properly. Cannot be combined with --no-walk.

    This enables parent rewriting, see History Simplification above.

    This implies the --topo-order option by default, but the --date-order option may also
    be specified.
```

```
--abbrev-commit
    Instead of showing the full 40-byte hexadecimal commit object name, show only a partial
    prefix. Non default number of digits can be specified with "--abbrev=<n>" (which also
    modifies diff output, if it is displayed).

    This should make "--pretty=oneline" a whole lot more readable for people using
    80-column terminals.

--no-abbrev-commit
    Show the full 40-byte hexadecimal commit object name. This negates --abbrev-commit and
    those options which imply it such as "--oneline". It also overrides the
    log.abbrevCommit variable.

--oneline
    This is a shorthand for "--pretty=oneline --abbrev-commit" used together.
```

described below (see **git-config**(1)). Here are the details of the built-in formats:

- oneline

      <hash> <title line>

  This is designed to be as compact as possible.

```
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$ python3 gatosGit/gatos.py
Avena, 4 annos, color: negro
Pelusa, 3 annos, color: cafe
Estrella, 6 annos, color: Blnaco
ccarlos@carlos-c:~/personal_workbench/LaboratorioGit$
```