

Beginning with the pyHed.

1° Step - Environment:

To use pyHed you must configure your environment with:

- [Python 2.5](#): Python interpreter;
- [pyQt4](#): GUI library;
- [sqlAlchemy 0.5.4](#): ORM;
 - [cx_Oracle](#): Driver for oracle;
 - [KinterbasDb](#): Driver for firebird;
 - **Important**: If you have problems to connect on KinterbasDb contact us.
- [ReportLab](#): Library for PDF generation.

Add pyHed path to PYTHONPATH. For example: If pyHed is in directory X:, the string should be "PYTHONPATH;X:\".

OK! Your environment is configured.

2° Step – DataBase:

The database of pyHed application must have a basic structure, with tables of action, profiles, users and others...

Use the script ' BasicDatabase.sql ' to create a basic database. The script it's available on pyHed site.

3° Step – Directory structure , classes and basic files:

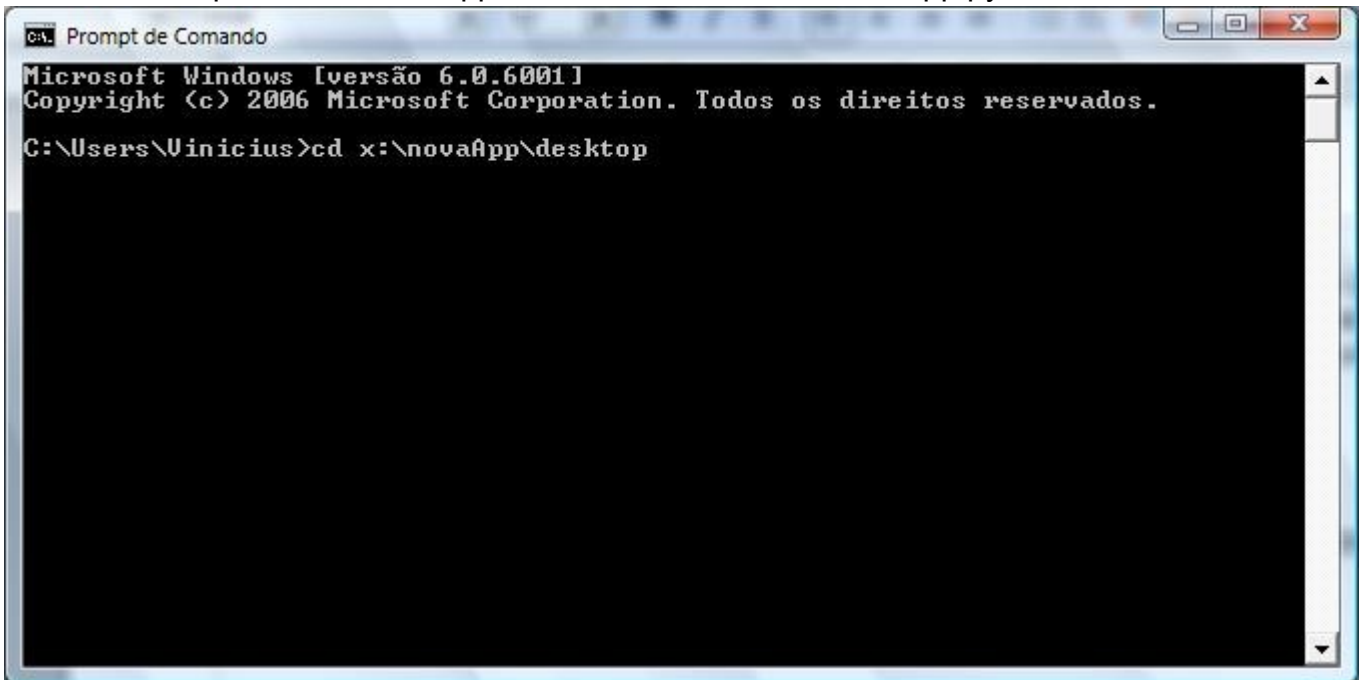
Assume as the default directory the X:/ , PYTHONPATH added previously. Inside X:\ the pyHed and your project must be. By default your project should be the directory structure shown below. On this document we will not detail the basic files, you can see them in the example of application contained on this documentation.

- **novaApp**: Folder of the project;
 - **dbTables**: Here the file that contains database mapper and login validation functions ;
 - **appDbTables.py** : It is not necessary to appoint him as.
 - **desktop**: Project files and image folder;
 - **img**: Image folder;
 - **novaApp.py** : File responsible for execution;
 - **novaAppMain.py**: Main class, inherits frameCustomMain, it will be recorded frames of the system, created the menus and others functions of application home.
 - **FrameLogin**: Amounts to the login screen and makes the validation. inherits frameCustomLogin.
 - **appConsts**: Class that loads constants in the system. It is not necessary, but makes it easier to access constants.
 - **config.ini**: File that contains system constants.

This structure is important because you can create an web application use the pyHew and utilized the same access control and database mappers of desktop.

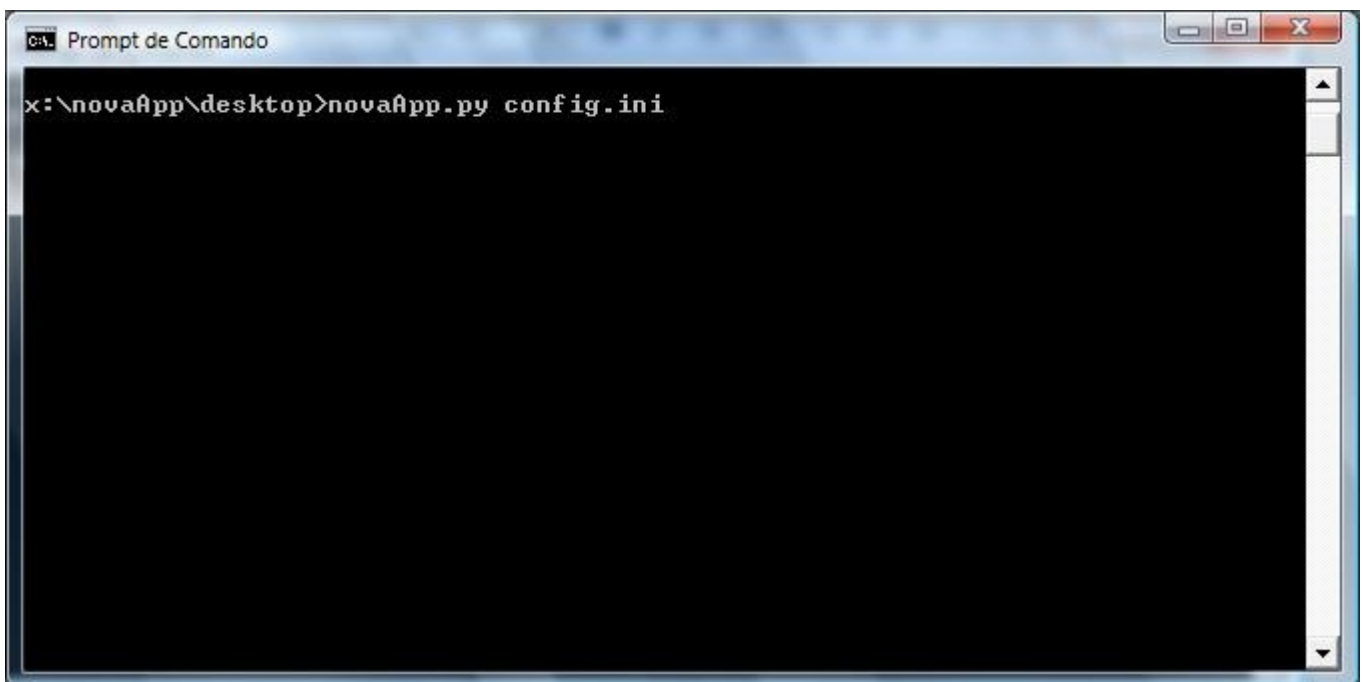
4° Step - Running the application:

To run the application, open the command prompt and go to application directory where is the class responsible for run application. On this case is novaApp.py.



```
C:\Users\Uinicius>cd x:\novaApp\desktop
```

Now execute >> novaApp.py config.ini



```
x:\novaApp\desktop>novaApp.py config.ini
```

novaApp.py: File responsible to run.
config.ini: Necessary argument to run.
Congratulations! You learn to use the pyHed.