

# Guida per scrivere una Guida



## Guida per scrivere una Guida

*Release 0.1 draft*

**Roberto Ceccarelli**

09.04.2017

## Sommario

<b>1</b>	<b>Il progetto</b>	<b>1</b>
1.1	Installare il software necessario	1
1.1.1	Python per Windows	1
1.1.2	Sphinx-build	3
1.1.3	MikTeX	3
1.2	Preparare la cartella di sviluppo	3
1.2.1	Il file di configurazione	5
1.2.2	Il file index	8
1.2.3	Il file di licenza per i PDF	8
<b>2</b>	<b>Scrivere il testo</b>	<b>10</b>
2.1	Corsivo, Grassetto e Sottolineato	10
2.2	Immagini	10
2.3	Capitoli e Sezioni	10
2.4	Link	10
2.5	Blocchi di codice	10
<b>3</b>	<b>Gestire modifiche e versioni</b>	<b>11</b>
3.1	Installare Git	11
3.2	Creare un repository	11
3.2.1	Il file .gitignore	11
3.2.2	Attivare il repository	12
3.2.3	Il primo commit	12
3.3	Utilizzare il repository	14
<b>4</b>	<b>Condividere i sorgenti</b>	<b>15</b>
4.1	Creare un repository su GitHub	15
4.2	Sincronizzare i repository	17
4.2.1	Pull del repository remoto	17
4.2.2	Push del repository locale	17
4.3	Mantenere sincronizzati i repository	19
4.3.1	Da locale a remoto	19
4.3.2	Da remoto a locale	19
4.4	I file di servizio	19
4.4.1	README.md	20
4.4.2	LICENSE.txt	20
<b>5</b>	<b>Pubblicare la guida</b>	<b>21</b>
5.1	Il sito per la documentazione del progetto	21
5.2	Aggiungere i download di altre versioni	22

5.3	Automatizzare il processo . . . . .	23
<b>6</b>	<b>Bibliografia</b>	<b>25</b>
6.1	ReStructuredText . . . . .	25
6.2	Git . . . . .	25
6.3	GitHub . . . . .	25
6.4	Licenze . . . . .	25
<b>7</b>	<b>Ringraziamenti</b>	<b>26</b>
<b>8</b>	<b>Download sorgenti e binari</b>	<b>27</b>
8.1	Versioni compilate . . . . .	27
8.2	Sorgenti . . . . .	27
<b>9</b>	<b>Note legali</b>	<b>28</b>
9.1	Garanzia . . . . .	28
9.2	Riconoscimenti della proprietà . . . . .	28
9.3	Accordi di Copyright e Licenza . . . . .	28

Per realizzare la nostra guida utilizzeremo dei tools che gestiscono un linguaggio di markup denominato [ReStructuredText](#)

Grazie a questo set di strumenti redigeremo il testo con un normalissimo editor (il notepad di Windows può andare benissimo) e questo potrà essere trasformato in un set di pagine Web (o anche una pagina unica se lo preferiamo), un documento Pdf, un ebook e tanti altri formati.

Darò indicazione su come installare i programmi necessari in ambiente Windows, ma gli strumenti presentati sono tutti multiplatforma e si possono utilizzare benissimo anche su macchine Linux o Mac.

## 1.1 Installare il software necessario

Innanzitutto è necessario accertarsi che il proprio PC abbia un interprete Python, cosa comune nelle macchine unix/linux, ma al contrario rara nelle macchine Windows

### 1.1.1 Python per Windows

Può essere scaricato dal sito [python.org](https://python.org) (scegliete l'installer appropriato per il vostro PC, non il file zip).

Lanciamo l'installer



Spuntate entrambe le caselline in basso e poi avviate l'installazione standard confermando sempre le scelte proposte.

Al termine dell'installazione apparirà questo screen:



dove consiglio di cliccare sulla voce Disable path length limit prima di terminare.



### 1.1.2 Sphinx-build

Se avete installato una versione recente di Python (come quella delle figure sopra) dovreste già trovarvi installato pip, il gestore dei pacchetti di Python. Se così non fosse [leggete qui](#) come fare.

Aprire il prompt dei comandi e scrivete:

```
pip install sphinx
```

Verranno scaricati ed installati alcuni pacchetti, al termine scrivete:

```
sphinx-build -h
```

che, se tutto è andato a buon fine, vi mostrerà l'help del tool.

### 1.1.3 MikTeX

Il sistema di build non è in grado di generare direttamente dei file PDF, ma si appoggia ad un altro sistema di composizione dei testi chiamato [LaTeX](#). Dovremo quindi installare un interprete per tale linguaggio: utilizzeremo [MikTeX](#).

L'installazione è banale: basta confermare tutte le opzioni proposte dall'installer.

## 1.2 Preparare la cartella di sviluppo

Siamo finalmente pronti ad iniziare il nostro progetto. Creiamo una cartella per contenerlo, posizioniamoci al suo interno ed apriamo il prompt dei comandi. (Per chi non lo sapesse: tenere premuto il tasto SHIFT e fare click con il tasto destro del mouse; dal menu che appare selezionare *Apri finestra di comando qui*)

Nel prompt comandi scriviamo:

```
sphinx-quickstart
```

A cui seguirà il dialogo riportato di seguito

Listato 1.1: sphinx-quickstart

```
C:\guide_guide>sphinx-quickstart
Welcome to the Sphinx 1.5.3 quickstart utility.

Please enter values for the following settings (just press Enter to
accept a default value, if one is given in brackets).

Enter the root path for documentation.
> Root path for the documentation [.]:

You have two options for placing the build directory for Sphinx output.
Either, you use a directory "_build" within the root path, or you separate
"source" and "build" directories within the root path.
> Separate source and build directories (y/n) [n]:

Inside the root directory, two more directories will be created; "_templates"
for custom HTML templates and "_static" for custom stylesheets and other static
files. You can enter another prefix (such as ".") to replace the underscore.
> Name prefix for templates and static dir [_]:

The project name will occur in several places in the built documentation.
> Project name: guide_guide
> Author name(s): Roberto Ceccarelli
```

Sphinx has the notion of a "version" and a "release" for the software. Each version can have multiple releases. For example, for Python the version is something like 2.5 or 3.0, while the release is something like 2.5.1 or 3.0a1. If you don't need this dual structure, just set both to the same value.

```
> Project version []: 0.1
```

```
> Project release [0.1]:
```

If the documents are to be written in a language other than English, you can select a language here by its language code. Sphinx will then translate text that it generates into that language.

For a list of supported codes, see

<http://sphinx-doc.org/config.html#confval-language>.

```
> Project language [en]: it
```

The file name suffix for source files. Commonly, this is either ".txt" or ".rst". Only files with this suffix are considered documents.

```
> Source file suffix [.rst]:
```

One document is special in that it is considered the top node of the "contents tree", that is, it is the root of the hierarchical structure of the documents. Normally, this is "index", but if your "index" document is a custom template, you can also set this to another filename.

```
> Name of your master document (without suffix) [index]:
```

Sphinx can also add configuration for epub output:

```
> Do you want to use the epub builder (y/n) [n]: y
```

Please indicate if you want to use one of the following Sphinx extensions:

```
> autodoc: automatically insert docstrings from modules (y/n) [n]:
```

```
> doctest: automatically test code snippets in doctest blocks (y/n) [n]:
```

```
> intersphinx: link between Sphinx documentation of different projects (y/n) [n]:
```

```
> todo: write "todo" entries that can be shown or hidden on build (y/n) [n]:
```

```
> coverage: checks for documentation coverage (y/n) [n]:
```

```
> imgmath: include math, rendered as PNG or SVG images (y/n) [n]:
```

```
> mathjax: include math, rendered in the browser by MathJax (y/n) [n]:
```

```
> ifconfig: conditional inclusion of content based on config values (y/n) [n]:
```

```
> viewcode: include links to the source code of documented Python objects (y/n) [n]:
```

```
> githubpages: create .nojekyll file to publish the document on GitHub pages (y/n) [n]: y
```

A Makefile and a Windows command file can be generated for you so that you only have to run e.g. 'make html' instead of invoking sphinx-build directly.

```
> Create Makefile? (y/n) [y]: y
```

```
> Create Windows command file? (y/n) [y]: y
```

```
Creating file .\conf.py.
```

```
Creating file .\index.rst.
```

```
Creating file .\Makefile.
```

```
Creating file .\make.bat.
```

Finished: An initial directory structure has been created.

You should now populate your master file .\index.rst and create other documentation source files. Use the Makefile to build the docs, like so:

```
make builder
```

where "builder" is one of the supported builders, e.g. html, latex or linkcheck.

```
C:\guide_guide>
```

Mi sembra abbastanza semplice capire dove va inserito il vostro nome e quello del progetto; per tutto il

resto potete dare le risposte dell'esempio.

Alla fine della procedura la cartella sarà popolata da alcuni files e directory: dovremo concentrare la nostra attenzione su due di loro.

### 1.2.1 Il file di configurazione

Questo file contiene tutte le impostazioni che regolano il *rendering* della nostra guida sulle varie piattaforme. Non lo analizzeremo tutto, ma solo le parti su cui è necessario intervenire per ottenere lo stile con cui è stata compilata questa guida.

Listato 1.2: config.py - preambolo

```
# All configuration values have a default; values that are commented out
# serve to show the default.

import sys
import os
import shlex

# If extensions (or modules to document with autodoc) are in another directory,
# add these directories to sys.path here. If the directory is relative to the
# documentation root, use os.path.abspath to make it absolute, like shown here.
#sys.path.insert(0, os.path.abspath('.'))
```

Nelle opzioni generali dobbiamo controllare

Listato 1.3: config.py - parametri generali

```
# -- General configuration -----

# If your documentation needs a minimal Sphinx version, state it here.
#needs_sphinx = '1.0'

# Add any Sphinx extension module names here, as strings. They can be
# extensions coming with Sphinx (named 'sphinx.ext.*') or your custom
# ones.
# Install cloud_sptheme with the following command:
# pip install cloud_sptheme
extensions = [
    'sphinx.ext.autodoc',
    'sphinx.ext.mathjax',
    'sphinx.ext.githubpages',
]

# Add any paths that contain templates here, relative to this directory.
templates_path = ['_templates']

# The suffix(es) of source filenames.
# You can specify multiple suffix as a list of string:
# source_suffix = ['.rst', '.md']
source_suffix = '.rst'

# The encoding of source files.
#source_encoding = 'utf-8-sig'

# The master toctree document.
master_doc = 'index'

# General information about the project.
project = 'Guida per scrivere una guida'
copyright = '2017 Roberto Ceccarelli - The Strawberry Field'
```



```
author = 'Roberto Ceccarelli'

# The version info for the project you're documenting, acts as replacement for
# |version| and |release|, also used in various other places throughout the
# built documents.
#
# The short X.Y version.
version = '0.1'
# The full version, including alpha/beta/rc tags.
release = '0.1 draft'

# The language for content autogenerated by Sphinx. Refer to documentation
# for a list of supported languages.
#
# This is also used if you do content translation via gettext catalogs.
# Usually you set "language" from the command line for these cases.
language = 'it'

# There are two options for replacing |today|: either, you set today to some
# non-false value, then it is used:
#today = ''
# Else, today_fmt is used as the format for a strftime call.
today_fmt = '%d.%m.%Y'
```

Nella sezione HTML

Listato 1.4: config.py - HTML

```
# -- Options for HTML output -----

# The theme to use for HTML and HTML Help pages.  See the documentation for
# a list of builtin themes.
html_theme = 'sphinx_rtd_theme'

# Theme options are theme-specific and customize the look and feel of a theme
# further.  For a list of options available for each theme, see the
# documentation.
html_theme_options = {}

# Add any paths that contain custom themes here, relative to this directory.
html_theme_path = []

# The name for this set of Sphinx documents.  If None, it defaults to
# "<project> v<release> documentation".
html_title = "draft"

# A shorter title for the navigation bar.  Default is the same as html_title.
html_short_title = None

# The name of an image file (relative to this directory) to place at the top
# of the sidebar.
html_logo = 'images/logo.png'

# The name of an image file (within the static path) to use as favicon of the
# docs.  This file should be a Windows icon file (.ico) being 16x16 or 32x32
# pixels large.
html_favicon = None

# Add any paths that contain custom static files (such as style sheets) here,
# relative to this directory. They are copied after the builtin static files,
# so a file named "default.css" will overwrite the builtin "default.css".
html_static_path = ['_static']
```

```
# If true, links to the reST sources are added to the pages.
html_show_sourcelink = False

# Language to be used for generating the HTML full-text search index.
# Sphinx supports the following languages:
#   'da', 'de', 'en', 'es', 'fi', 'fr', 'h', 'it', 'ja'
#   'nl', 'no', 'pt', 'ro', 'r', 'sv', 'tr'
html_search_language = 'it'
```

Nella sezione LaTeX

Listato 1.5: config.py - LaTeX

```
# -- Options for LaTeX output -----

latex_elements = {

    'classoptions': ',openany,oneside',
    'babel': '\\usepackage[italian]{babel}',

    # The paper size ('letterpaper' or 'a4paper').
    'papersize': 'a4paper',

    # The font size ('10pt', '11pt' or '12pt').
    'pointsize': '10pt',

    # Additional stuff for the LaTeX preamble.
    'preamble': r'''
\usepackage[default]{lato}
\usepackage{inconsolata}
\usepackage{xmpincl}
\includexmp{license}
\hypersetup{pdfutex,
             pdfauthor={Roberto Ceccarelli - The Strawberry Field},
             pdftitle={Guida per scrivere una Guida},
             pdfsubject={Guida per scrivere una guida con ReStructuredText, Git e GitHub},
             pdfkeywords={Guida, howto, manual ReStructuredText, git, GitHub}}
''',

    # Latex figure (float) alignment
    'figure_align': 'htbp',

    'fncychap': '\\usepackage[Bjornstrup]{fncychap}',

}

# include file for license
latex_additional_files = ['license.xmp']

# Grouping the document tree into LaTeX files. List of tuples
# (source start file, target name, title,
#  author, documentclass [howto, manual, or own class]).
latex_documents = [
    (master_doc, 'guide.tex', 'Guida per scrivere una Guida',
     'Roberto Ceccarelli', 'manual'),
]

#latex_toplevel_sectioning = 'chapter'
```

```
# The name of an image file (relative to this directory) to place at the top of
# the title page.
latex_logo = 'images/cover_image.png'
```

## 1.2.2 Il file index

Questo è il file che elenca tutti i file ReStructuredText che compongono la nostra guida.

Quello che segue è il file indice di questa guida che dovrete adattare per il vostro lavoro indicando (nell'ordine voluto) i file .rst che compongono la guida.

Listato 1.6: index.rst

```
.. TSF Guide Howto master file, created by
   sphinx-quickstart on Tue Dec 22 19:39:50 2015.
   You can adapt this file completely to your liking, but it should at least
   contain the root `toctree` directive.
```

### Una guida per creare guide

```
=====
```

```
.. toctree::
   :maxdepth: 3
   :numbered:
   :name: mastertoc
   :caption: Sommario

   rest
   rest-write
   git
   github
   github-site
   biblio
   acknowledgements
   download
   legal
```

Fate attenzione che l'indentazione è fatta da quattro spazi e non da tabulazioni (che non funzionerebbero). Non necessariamente va utilizzato un file per capitolo, potete organizzarvi come meglio credete.

## 1.2.3 Il file di licenza per i PDF

Non strettamente necessario, ma utile per realizzare file PDF che espongano la loro licenza ai motori di ricerca.

Sul sito di creative Commons potete generarlo utilizzando [questo form](#) oppure potete modificare il file per adattarlo alla licenza che avete scelto.

Questo è il file utilizzato per questa guida:

Listato 1.7: license.xml

```
<?xpacket begin='' id=''>
<x:xmpmeta xmlns:x='adobe:ns:meta/'>
  <rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
    <rdf:Description rdf:about=''
      xmlns:xapRights='http://ns.adobe.com/xap/1.0/rights/'>
      <xapRights:Marked>True</xapRights:Marked>
```

```

</rdf:Description>
<rdf:Description rdf:about=''
    xmlns:xapRights='http://ns.adobe.com/xap/1.0/rights/'>
    <xapRights:WebStatement rdf:resource='https://github.com/strawberryfield/guide_guide'/>
</rdf:Description>
<rdf:Description rdf:about=''
    xmlns:xapRights='http://ns.adobe.com/xap/1.0/rights/'
    >
    <xapRights:UsageTerms>
    <rdf:Alt>
        <rdf:li xml:lang='x-default' >Quest'opera è distribuita con Licenza <a rel='license' href='http://creativecommons.org/licenses/by-nc-sa/4.0/'>Creative Commons
        ↪Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale</a>.</rdf:li>
        <rdf:li xml:lang='it-IT' >Quest'opera è distribuita con Licenza <a rel='license' href='http://creativecommons.org/licenses/by-nc-sa/4.0/'>Creative Commons
        ↪Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale</a>.</rdf:li>
        <rdf:li xml:lang='en' >This work is licensed under a <a rel='license' href='http://creativecommons.org/licenses/by-nc-sa/4.0/'>Creative Commons Attribution-
        ↪NonCommercial-ShareAlike 4.0 International License</a>.</rdf:li>
    </rdf:Alt>
    </xapRights:UsageTerms>
</rdf:Description>
<rdf:Description rdf:about=''
    xmlns:dc='http://purl.org/dc/elements/1.1/'>
    <dc:title>
    <rdf:Alt>
        <rdf:li xml:lang='x-default'>Guida per scrivere una guida</rdf:li>
        <rdf:li xml:lang='it-IT'>Guida per scrivere una guida</rdf:li>
    </rdf:Alt>
    </dc:title>
</rdf:Description>
<rdf:Description rdf:about=''
    xmlns:cc='http://creativecommons.org/ns#'>
    <cc:license rdf:resource='http://creativecommons.org/licenses/by-nc-sa/4.0/'>
</rdf:Description>
<rdf:Description rdf:about=''
    xmlns:cc='http://creativecommons.org/ns#'>
    <cc:attributionName>Roberto Ceccarelli - The Strawberry Field</cc:attributionName>
</rdf:Description>

</rdf:RDF>
</x:xmpmeta>
<?xpacket end='r'?>

```

No, non voglio dirvi che cosa dovete scrivere, ma affronteremo il modo di *marcare* il testo in modo che, una volta elaborato, appaia come desideriamo.

Dovendo il nostro testo essere disponibile in differenti formati occorre un sistema *neutro* per indicare ad esempio che una parola deve essere sottolineata oppure in corsivo. In questo capitolo vedremo proprio questo aspetto.

## 2.1 Corsivo, Grassetto e Sottolineato

## 2.2 Immagini

## 2.3 Capitoli e Sezioni

## 2.4 Link

## 2.5 Blocchi di codice

## Gestire modifiche e versioni

Uno dei vantaggi di gestire la documentazione con dei file di testo e non con formati proprietari come quelli dei word-processor (Word, Libre Office Writer...) è che si possono gestire con degli strumenti di *versioning* che permettono di tenere traccia di tutte le modifiche anche se effettuate da persone diverse.

Questi strumenti, nati per i sorgenti dei programmi per computer, possono essere utilizzati in tanti altri campi: anni fa scrissi una [guida su come utilizzarli per creare contenuti per un simulatore ferroviario](#) in cui gran parte dei files erano testo.

Possono essere utilizzati anche con files binari, ma perdono alcune funzioni applicabili solo ai testi.

Utilizzeremo Git che ci facilita l'integrazione con un [sito per la condivisione](#) e la pubblicazione dei nostri lavori come vedremo nel [prossimo capitolo](#).

### 3.1 Installare Git

L'installer del programma per Windows può essere [scaricato dal sito di Git](#)

L'installer propone diversi pannelli con delle opzioni: per l'uso normale quelle di default vanno benissimo.

### 3.2 Creare un repository

Il repository è una struttura di file (gestita da Git) dove vengono archiviate tutte le modifiche ai file.

#### 3.2.1 Il file .gitignore

Per impostazione predefinita vengono inclusi nel repository tutti i files e le sottocartelle della directory che vogliamo gestire.

Abbiamo visto nel capito precedente che viene creata una cartella `_build` nella quale sono salvati i risultati delle elaborazioni e che non ci interessa includere nel repository. Per gestire queste eccezioni si può creare un file `.gitignore` che contenga la lista dei file e cartelle da escludere.

Questo è il file presente in questo progetto:

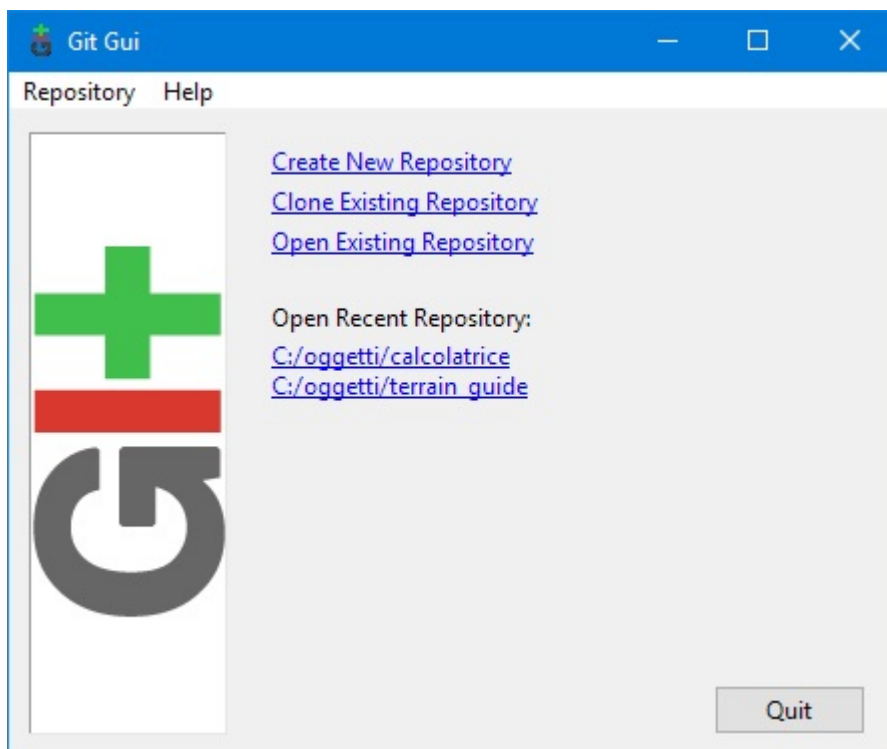
Listato 3.1: .gitignore

```
# exclude build dir
_build/
```

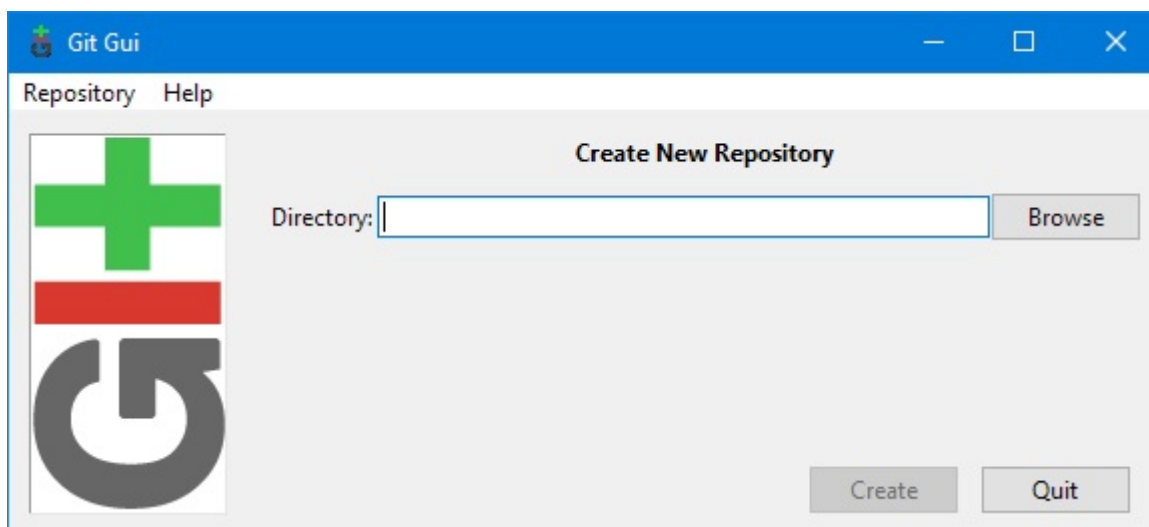


### 3.2.2 Attivare il repository

Clicchiamo con il tasto destro del mouse sulla cartella contenente il nostro progetto e dal menù che appare selezioniamo Git GUI [here](#)



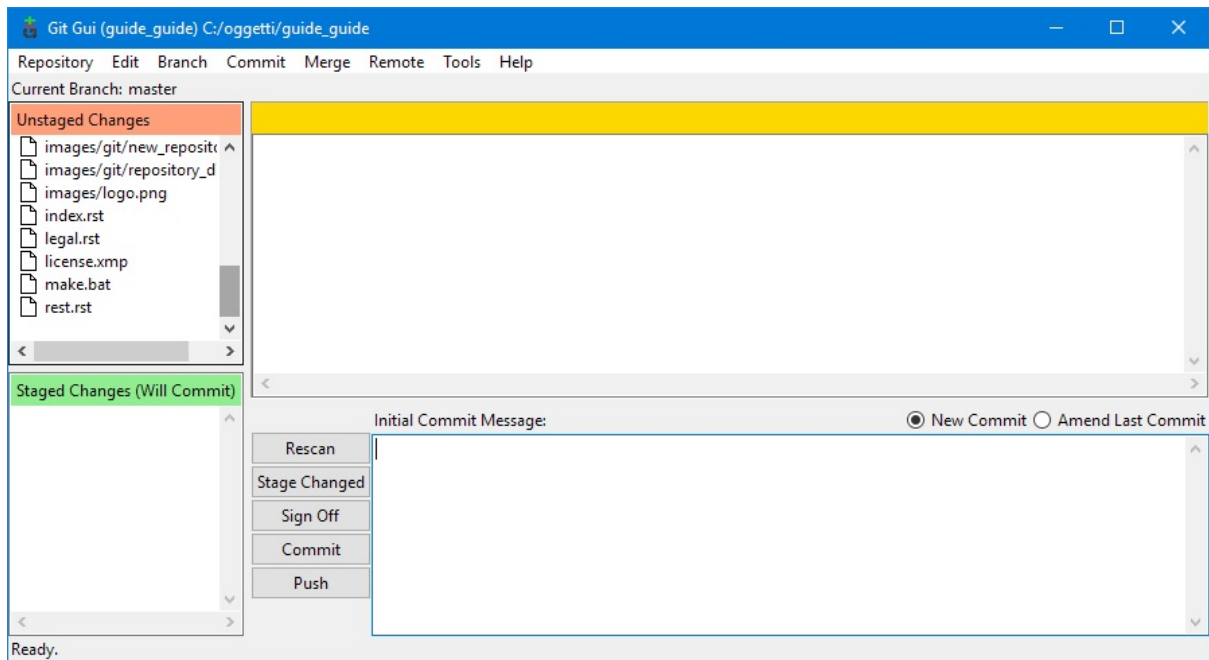
Dalla finestra riportata sopra selezioniamo Create New Repository



Indichiamo la cartella del nostro progetto (possiamo anche selezionarla con il pulsante Browse) e quindi clicchiamo su Create.

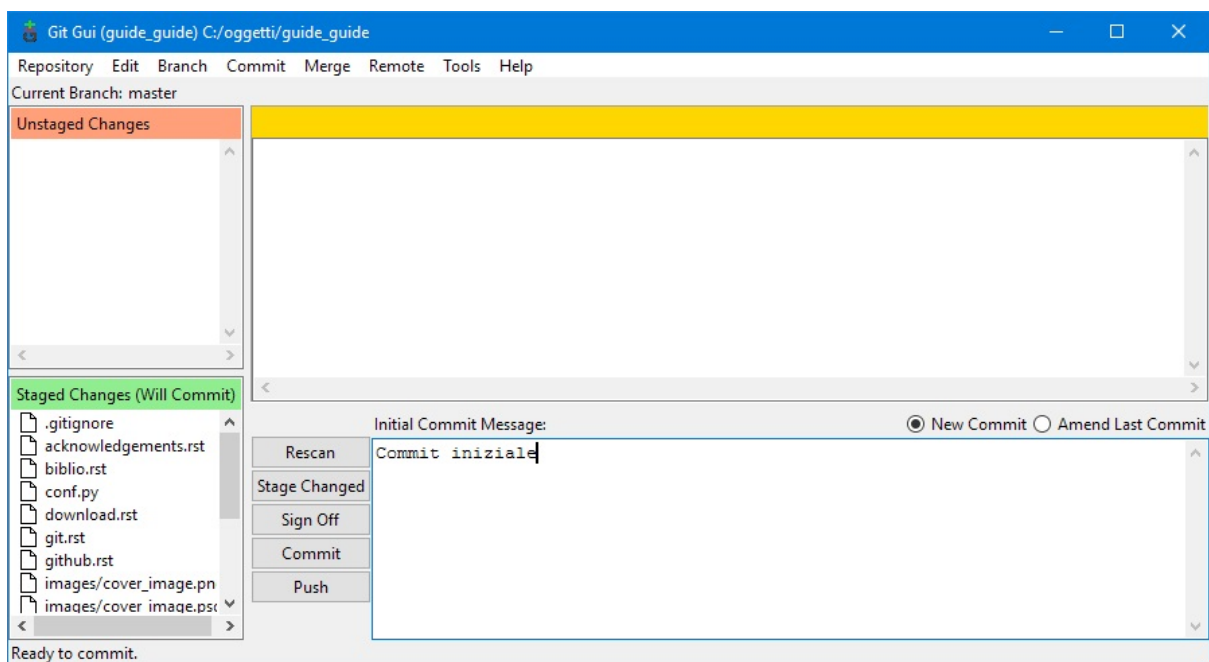
### 3.2.3 Il primo commit

Il sistema ci presenterà questa nuova formata



nella quale elencherà i files presenti nella nostra cartella permettendoci di inserirli nel repository.

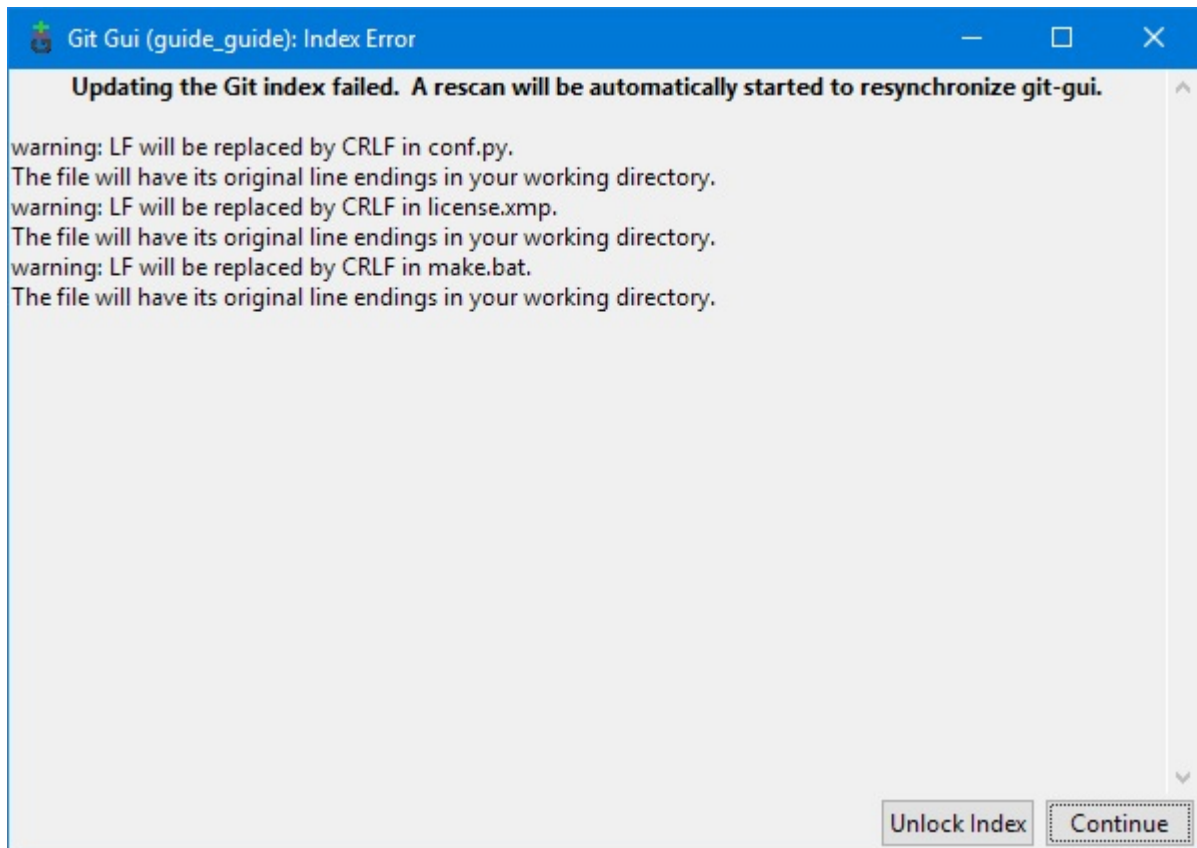
Avendo già impostato le esclusioni possiamo inserirli tutti cliccando sul pulsante Stage changed. I files passeranno nel box sotto:



Per completare l'operazione di inserimento descriviamo i cambiamenti fatti nel box in basso a destra; essendo questo il primo commit scriviamolo come nell'esempio.

Non ci resta che premere il pulsante Commit ed il gioco è fatto.

Molto probabilmente vi apparirà questa segnalazione:



che possiamo tranquillamente ignorare cliccando su Continue (è dovuta alla diversa gestione del *fine riga* tra le macchine Windows e Unix: Git utilizza il metodo di quest'ultimo e ci segnala che effettua una conversione per noi totalmente ininfluente)

### 3.3 Utilizzare il repository

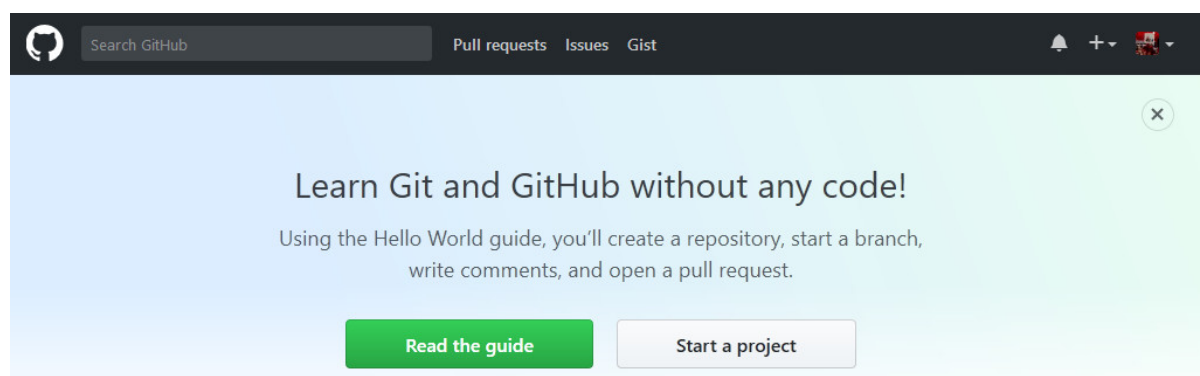
## Condividere i sorgenti

Finora abbiamo gestito la scrittura della nostra guida sul nostro PC: in questo capitolo vedremo come condividere i nostri files sorgenti su una piattaforma internet che ci consentirà di condividerli ed utilizzarli per una scrittura condivisa.

La piattaforma che ci consente questo è GitHub (<https://github.com/>): una volta registrati (gratuitamente) si possono creare infiniti repository per tutti i nostri progetti.

### 4.1 Creare un repository su GitHub

Una volta creato l'account su GitHub ci viene data la possibilità di creare un nuovo repository



clicchiamo su Start a project e completiamo questo form:


## Create a new repository

A repository contains all the files for your project, including the revision history.

---

Owner

Repository name

 strawberryfield ▾

 / 


guide\_guide ✓


Great repository names are short and memorable. Need inspiration? How about **cuddly-tribble**.

Description (optional)

Guida per scrivere una guida

---

☒  **Public**  
Anyone can see this repository. You choose who can commit.


☐  **Private**  
You choose who can see and commit to this repository.

---

☒ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

 | 

Add a license: **None** ▾ 

---

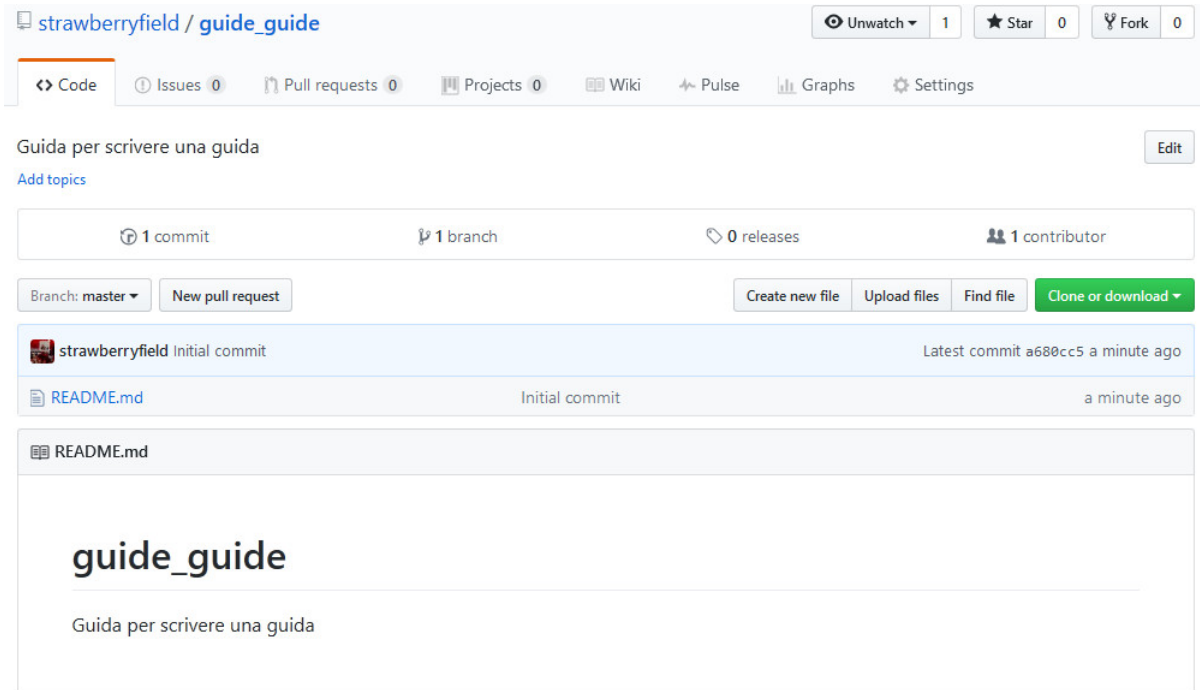
**Create repository**

scegliendo un nome appropriato per il progetto (che comunque potrà essere cambiato in seguito)

Inseriamo una breve, ma significativa, descrizione e abilitiamo la generazione del readme. Non preoccupiamoci del file `.gitignore` (ne abbiamo già uno nel nostro repository locale) e nemmeno della licenza che inseriremo in seguito (le licenze proposte sono rivolte alla scrittura di software e poco adatte alla documentazione).

Il tipo di repository va lasciato su **Public**: in questo modo chiunque, anche non iscritto a GitHub può leggerne e scaricare il contenuto; gli eventuali utenti che autorizzeremo potranno anche aggiornarlo.

Confermiamo il tutto con **Create repository** e GitHub ci presenterà il repository appena creato:



## 4.2 Sincronizzare i repository

In questo momento abbiamo due repository separati: uno sul nostro PC ed uno su GitHub. Dovremo fondere i due repository in modo da poter replicare le modifiche locali sul repository remoto e viceversa.

### 4.2.1 Pull del repository remoto

Apriamo il prompt dei comandi nella nostra cartella di lavoro e scriviamo:

```
git pull guide_guide master --allow-unrelated-histories
```

Dove a guide\_guide dovremo sostituire il nome del nostro repository remoto.

Otterremo in risposta qualcosa di simile a questo:

```
From https://github.com/strawberryfield/guide_guide
* branch      master      -> FETCH_HEAD
Merge made by the 'recursive' strategy.
 README.md | 2 ++
1 file changed, 2 insertions(+)
create mode 100644 README.md
```

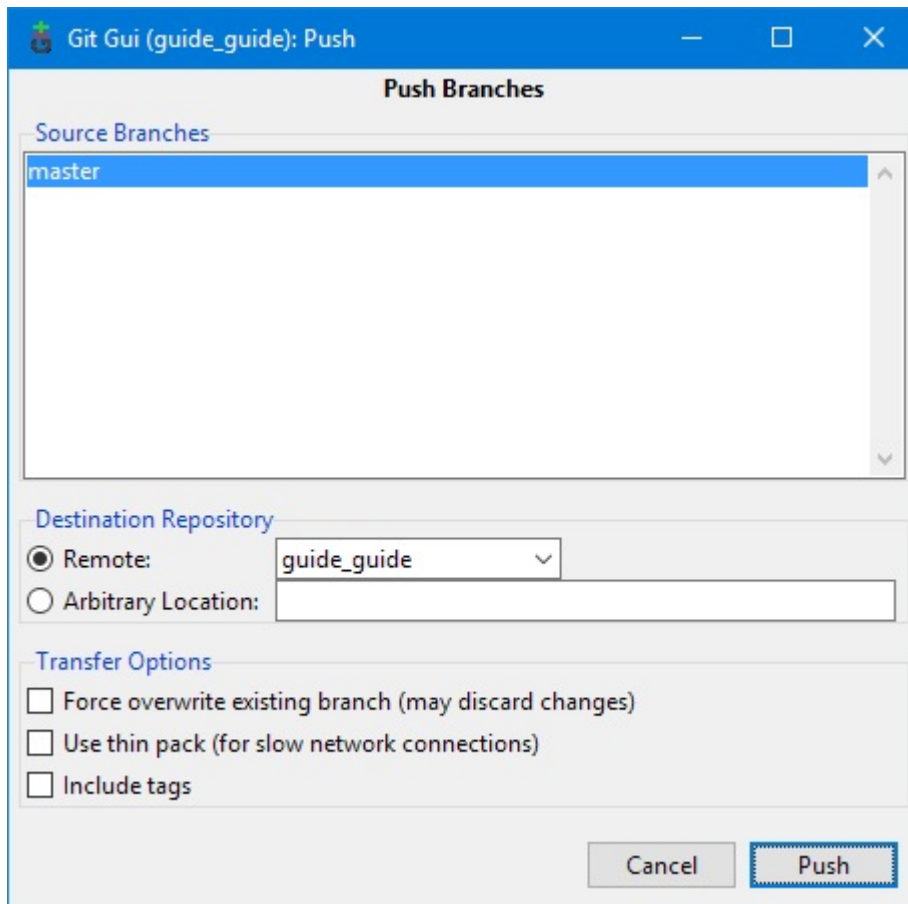
Il repository su github è stato fuso con quello locale: ce ne accorgiamo perchè nella nostra directory di lavoro è apparso il file README.md che era l'unico presente nel repository remoto.

Possiamo chiudere la shell comandi.

### 4.2.2 Push del repository locale

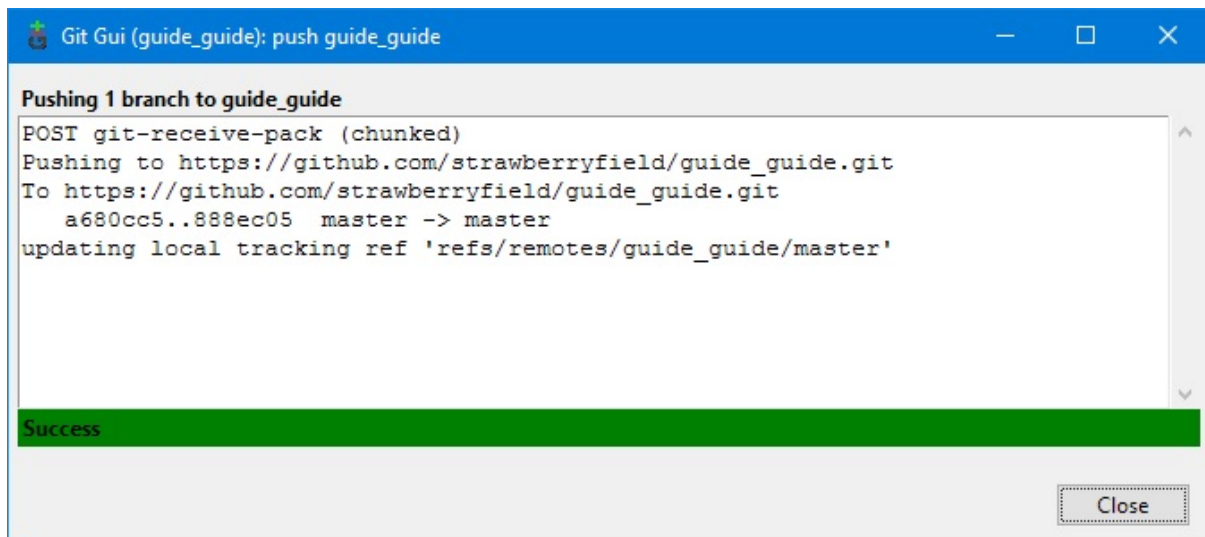
Riapriamo la GUI di Git e premiamo Push





Con questa operazione trasferiamo il contenuto del repository locale su quello remoto. Non c'è nulla da modificare e quindi confermiamo premendo Push.

Al termine del trasferimento riceveremo questa conferma:



Il nostro repository è stato trasferito su GitHub e se controlliamo dal sito possiamo vedere che ci sono tutti i nostri file

strawberryfield / guide\_guide

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Guida per scrivere una guida [Add topics](#) [Edit](#)

5 commits 1 branch 0 releases 1 contributor

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

File	Description	Time
images	linked to github	2 days ago
.gitignore	Commit iniziale	10 days ago
README.md	Initial commit	2 days ago
acknowledgements.rst	Commit iniziale	10 days ago
biblio.rst	Descritte installazioni	2 days ago
conf.py	Commit iniziale	10 days ago
download.rst	Descritte installazioni	2 days ago
git.rst	linked to github	2 days ago

e non solo: vengono sincronizzati anche tutti i commit che abbiamo eseguito sul PC locale

Branch: master

Commits on Apr 2, 2017

- Merge branch 'master' of https://github.com/strawberryfield/guide\_guide  
Roberto Ceccarelli committed 2 days ago [888ec05](#) [Code](#)
- linked to github  
Roberto Ceccarelli committed 2 days ago [d0a6c1c](#) [Code](#)
- Initial commit  
strawberryfield committed 2 days ago [a680cc5](#) [Code](#)
- Descritte installazioni  
Roberto Ceccarelli committed 2 days ago [70a0497](#) [Code](#)

Commits on Mar 25, 2017

- Commit iniziale  
Roberto Ceccarelli committed 10 days ago [34f4f6b](#) [Code](#)

## 4.3 Mantenere sincronizzati i repository

### 4.3.1 Da locale a remoto

### 4.3.2 Da remoto a locale

## 4.4 I file di servizio

E' buona norma includere nei progetti ospitati su GitHub due file: il *readme* e la licenza.

#### 4.4.1 README.md

#### 4.4.2 LICENSE.txt

## Pubblicare la guida

Github permette di gestire delle pagine web dove includere la documentazione del progetto. Per la verità è presente un sito globale legato al proprietario dei repository ed un sito per ogni progetto: noi utilizzeremo quest'ultimo per riversarci dentro le pagine HTML della nostra guida in modo che possa essere letta comodamente online.

### 5.1 Il sito per la documentazione del progetto

Il sito di documentazione è ospitato all'interno del nostro repository nella cartella docs che quindi dovremo creare nella nostra cartella di lavoro. Dalla console scrivete:

```
mkdir docs
```

poi rigeneriamo la guida in HTML:

```
make clean  
make html
```

quindi copiamo il sito generato nella cartella docs:

```
xcopy _build\html docs /S /Y
```

infine aggiungiamo i files al repository locale e sincronizziamo con GitHub:

```
git add .  
git commit -m "Importazione iniziale sito"  
git push guide_guide master
```

Fatte queste operazioni nel repository su GitHub sarà presente la nostra guida compilata in HTML all'interno della cartella docs

Branch: master	guide_guide / docs /	Create new file	Upload files	Find file	History
Roberto Ceccarelli Importazione iniziale sito Latest commit 5c137cf 8 minutes ago					
..					
_images	Importazione iniziale sito	8 minutes ago			
_sources	Importazione iniziale sito	8 minutes ago			
_static	Importazione iniziale sito	8 minutes ago			
.buildinfo	Importazione iniziale sito	8 minutes ago			
.nojekyll	Importazione iniziale sito	8 minutes ago			
acknowledgements.html	Importazione iniziale sito	8 minutes ago			
biblio.html	Importazione iniziale sito	8 minutes ago			
download.html	Importazione iniziale sito	8 minutes ago			
genindex.html	Importazione iniziale sito	8 minutes ago			

Possiamo quindi procedere all'abilitazione del sito.

Andiamo quindi nella sezione Settings del nostro repository su GitHub ed in fondo alla pagina delle Options troveremo la voce per abilitare le GitHub Pages:

### GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

**Source**

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None
Save

Select source

master branch
Use the master branch for GitHub Pages.

master branch /docs folder
Use only the /docs folder for GitHub Pages.

✓ None
Disable GitHub Pages.

Scegliamo l'opzione master branch /docs folder e salviamo.

Il sistema ci comunica dove troveremo le nostre pagine pubblicate

### GitHub Pages

Your site is ready to be published at [https://strawberryfield.github.io/guide\\_guide/](https://strawberryfield.github.io/guide_guide/).

Nel caso di questa guida l'URL è [https://strawberryfield.github.io/guide\\_guide/](https://strawberryfield.github.io/guide_guide/)

## 5.2 Aggiungere i download di altre versioni

Poichè il sistema ci permette di avere differenti tipi di output è buona cosa rendere disponibili all'utente finale (che non è in grado di ricompilarsi la guida) almeno il PDF. Io pubblico anche l'ebook in formato .epub,

ma ognuno è libero di scegliere i formati che ritiene più utili.

Per fare ciò creiamo una cartella bin all'interno di docs nella quale andremo a copiare i file .pdf e .epub che abbiamo preventivamente compilato.

E' anche utile aggiungere un capitolo (o sezione) nel quale linkare tutti possibili file da scaricare come ad esempio questo:

Listato 5.1: download.rst

```
.. _downloads:

*****
Download sorgenti e binari
*****

.. _downloads-bin:

Versioni compilate
=====

`La guida può essere letta on line a questo link <https://strawberryfield.github.io/guide_guide/>`,
`oppure potete scaricare il pdf <https://strawberryfield.github.io/guide_guide/bin/guide_guide.pdf>
↪`,

`o l'e-book <https://strawberryfield.github.io/guide_guide/bin/MSTSORTerrainGuide.epub>`.

.. _downloads-src:

Sorgenti
=====

`Per chi fosse interessato i sorgenti sono disponibili su GitHub <https://github.com/strawberryfield/
↪guide_guide>`.
```

che è quello di questa guida (non dimenticate di inserirlo nel file index.rst).

## 5.3 Automatizzare il processo

Ripetere tutte le volte la procedura di ricompilazione, copia e commit del sito è cosa piuttosto noiosa e fonte di errori. Ho perciò predisposto un piccolo batch che automatizza il processo.

Listato 5.2: makesite.bat

```
@echo off

REM script for building website to host on github

set builddir=_build
set site=docs
set downloads=%site%\bin

call make.bat clean
call make.bat html
xcopy %builddir%\html %site% /S /Y

call make.bat latexpdf
copy /Y %builddir%\latex\*.pdf %downloads%
```



```
call make.bat epub
copy /Y %builddir%\epub\*.epub %downloads%

git add .
git commit -m "Aggiornamento automatico sito"
git push guide_guide master
```

Prima di lanciare questo batch è opportuno fare il commit dei sorgenti in modo da tenere separato l'aggiornamento del sito dalle modifiche alla guida. Non è una cosa obbligatoria, ma aiuta a tenere le cose in ordine.

## 6.1 ReStructuredText

- [Sito di ReStructuredText](#)
- [Guida di Peter Gulyas per la redazione del manuale di OpenRails con ReStructuredText](#)
- [Sito di Python](#)
- [Sito di Sphinx-build](#)
- [Installare Sphinx-build](#)
- [Sito di MikTex](#)
- [Installare MikTex](#)
- [Conoscere LaTeX](#)
- [Sito di Read The Docs](#)

## 6.2 Git

- [Sito di Git](#)
- [Pro Git: la guida di riferimento di Git](#)

## 6.3 GitHub

- [Sito di GitHub](#)
- [Le guide di GitHub](#)

## 6.4 Licenze

- [Le licenze Creative Commons](#)
- [GNU Free Documentation License](#)
- [Licenze in XMP](#)

## Ringraziamenti

Una menzione particolare va a Peter Gulyas che mi ha fatto conoscere [ReStructuredText](#) che ho utilizzato per redigere questa guida.

## Download sorgenti e binari

### 8.1 Versioni compilate

La guida può essere letta on line a questo link,  
oppure potete scaricare il pdf,  
o l'e-book.

### 8.2 Sorgenti

Per chi fosse interessato i sorgenti sono disponibili su GitHub.

## 9.1 Garanzia

NESSUNA GARANZIA: Quest'opera è fornita dall'autore "così com'è" e qualsiasi garanzia espressa o implicita, inclusiva di, ma non limitata a, garanzie implicite di commerciabilità e idoneità ad uno scopo particolare, viene disconosciuta. In nessun caso il possessore di copyright sarà ritenuto responsabile per qualsiasi danno diretto, indiretto, connesso, particolare, esemplare o conseguente (inclusivo di, ma non limitato a, approvvigionamento di beni o servizi alternativi; perdita di utilità, dati o profitti; interruzione di affari) comunque causati e su qualsiasi ipotesi di responsabilità, come da contratto, responsabilità oggettiva, o torto (compresa negligenza o altro) derivante in qualsiasi modo dall'utilizzo di questa opera anche se al corrente della possibilità di tale danno.

Questo lavoro è stato fatto al solo scopo di divertirsi e sperando possa essere utile a qualcuno; se non vi piace semplicemente cancellatelo.

## 9.2 Riconoscimenti della proprietà

Casasoft, The Strawberry Field e le relative rappresentazioni grafiche sono di proprietà di Roberto Ceccarelli. Tutti gli altri marchi e loghi sono proprietà degli aventi diritto e sono utilizzati per identificare società, prodotti e servizi.

## 9.3 Accordi di Copyright e Licenza

© 2017 Roberto Ceccarelli - The Strawberry Field.

Quest'opera è stata rilasciata con licenza Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale.

Per leggere una copia della licenza visita il sito web <http://creativecommons.org/licenses/by-nc-sa/4.0/> o spedisci una lettera a Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.