



Guia de Usuario

Release 2.8.0dev1

Bazaar Developers

19 de September de 2020

Índice general

1. Introducción	1
2. Empezando	3
3. Control de Versionamiento Personal	5
4. Compartiendo con tus pares	7
5. Colaboración en equipo, modo centralizado	9
6. Colaboracion en equipo, modo distribuido	11
7. Un tour breve de los plugins mas populares	13
8. Integrando Bazaar en tu entorno	15
9. Temas varios	17
9.1. Usando <code>bzr version-info</code>	17
10. Apéndice	19

Introducción

Empezando

Control de Versionamiento Personal

Compartiendo con tus pares

Colaboración en equipo, modo centralizado

Colaboracion en equipo, modo distribuido

Un tour breve de los plugins mas populares

Integrando Bazaar en tu entorno

Temas varios

9.1 Usando `bzr version-info`

9.1.1 Repaso General

Este documento describe las formas de usar `bzr version-info` como parte del proceso de embeber la información de versión a un proyecto.

9.1.2 Proyecto Python

TODO: Figure out how to attach into `setup.py`

Si usa un archivo `Makefile` para construir su proyecto, puede generar un archivo con la información de versión tan simple como:

```
library/_version.py:
    bzr version-info --format=python > library/_version.py
```

Eso genera un archivo que contiene 3 diccionarios:

- *version_info*: Un diccionario conteniendo información básica sobre el estado actual
- *revisions*: Un diccionario listando todas las revisiones en el historial del tree, junto con los tiempos y los mensajes de los commits. Esto por defecto está en blanco salvo que use `--all` o `--include-history` es provisto. Esto es útil si quiere seguir que bugs arregla el lanzamiento de esa versión. Para muchos proyectos es más información de la que se va a necesitar.
- *file_revisions*: Un diccionario listando la revisión que modificó por última vez todos los archivos del proyecto. Esto puede ser usado similarmente a como se usan las palabras claves `Id` en los archivos controlados en CVS. La última fecha de modificación puede ser determinada mirando en el mapa de *revisions*. Esto también está vacío por defecto, y habilitado solo por `--all` o `--include-file-revisions`.

9.1.3 Check Clean

La mayoría de la información sobre el contenido del proyecto puede ser determinada a muy bajo costo con solo leer las entradas de revisiones. Sin embargo, puede ser útil si el working tree fue actualizado completamente cuando fue empaquetado, o si hubo alguna modificación local. Al proveer `--all` o `--check-clean`, `bzr` va a inspeccionar

el `working tree`, y definir el `clean flag` en `version_info`, al igual que definir entradas en `file_revisions` como `modified` donde es apropiado.

Apéndice
