

# Mantener un proyecto de software libre:

¿Qué habilidades necesito?

Alberto Fanjul

Diciembre 2018

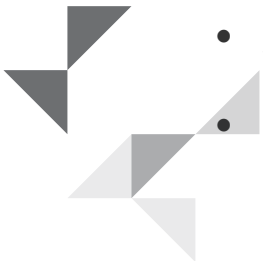


# Introducción



# Un poco sobre mí y GNOME:

- Alberto Fanjul
  - 10 años haciendo software
  - Miembro de la fundación GNOME
  - Mantenedor de gitg (visor de repositorios git)
- GNOME:
  - <https://www.gnome.org/foundation/>
  - Escritorio libre para todo el mundo <https://www.gnome.org/gnome-3/>
    - Gratis
    - Software libre
    - Hecho en colaboración



# Qué es el software libre?

- Poder descargar, compilar, modificar y ejecutar programas
- Todo lo que usas habitualmente lleva opensource
- Permite enfocarte en el proyecto que estás haciendo e indagar a fondo en el funcionamiento del sistema que usas
- Su explosión (en mi opinión) se debe sobre todo a Internet



# Por qué deberías probarlo?

- Hacer las cosas bien te enseñara un montón
    - Discutir la implementación
    - Cualquiera puede opinar sobre lo que sugieres
  - Ver otros modos de crear un proyecto de enseñará un montón
  - Puedes marcarte el objetivo que prefieras:
    - Quieres un nuevo trabajo
    - Quieres aprender a usar un lenguaje
- Necesitas un cambio en software (open source) que utilizas



# Licencias de software

- Dan cobertura a quien hace software libre, a quién acepta nuevos cambios y a quién lo usa
  - GPL: Esta licencia garantiza que lo que se haga a partir de este código será libre
  - LGPL: Esta licencia garantiza que puedas usar el código sin tener que liberarlo (temas comerciales)

Que son las CLA? Contributor license agreement: El que firma esto para un proyecto acepta que todo lo que ha hecho pasa a ser del proyecto y que no lo ha copiado de otro sitio (sin licencia)



# Dar el paso: Cómo comenzar a mantener un proyecto?

- Puede ser simplemente un proyecto que subas a github.com
- Puedes adoptar un proyecto existente:
  - Si el proyecto necesita un nuevo mantenedor
  - Siempre hace falta ayuda, comentalo y te sorprenderás

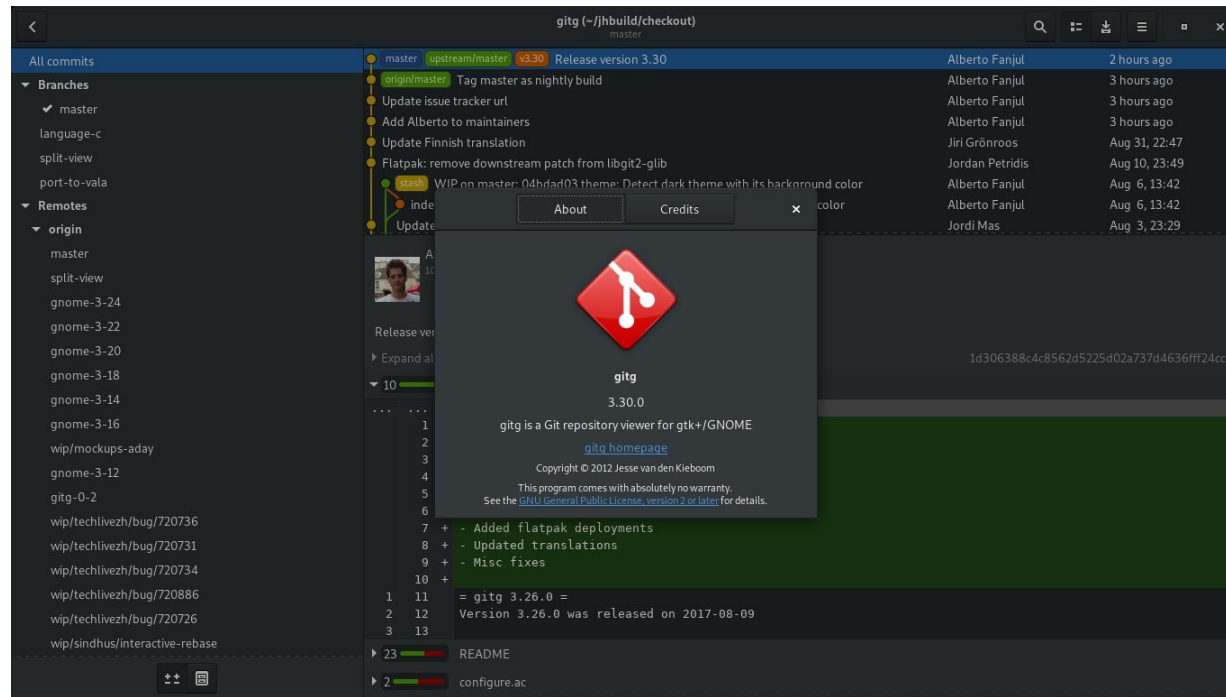


# Manteniendo GNOME Gitg





# Gitg



- Es software que muestra gráficamente un repositorio de git
- Cualquiera puede reportar problemas:
  - Diseñadores de GNOME que deciden cambiar aspectos gráficos
  - Usuarios que quieren nuevas funcionalidades
  - Usuarios que quieren usarlo en diferentes entornos (funciona en Linux, Mac y Windows)
  - Usuarios que tienen algún problema



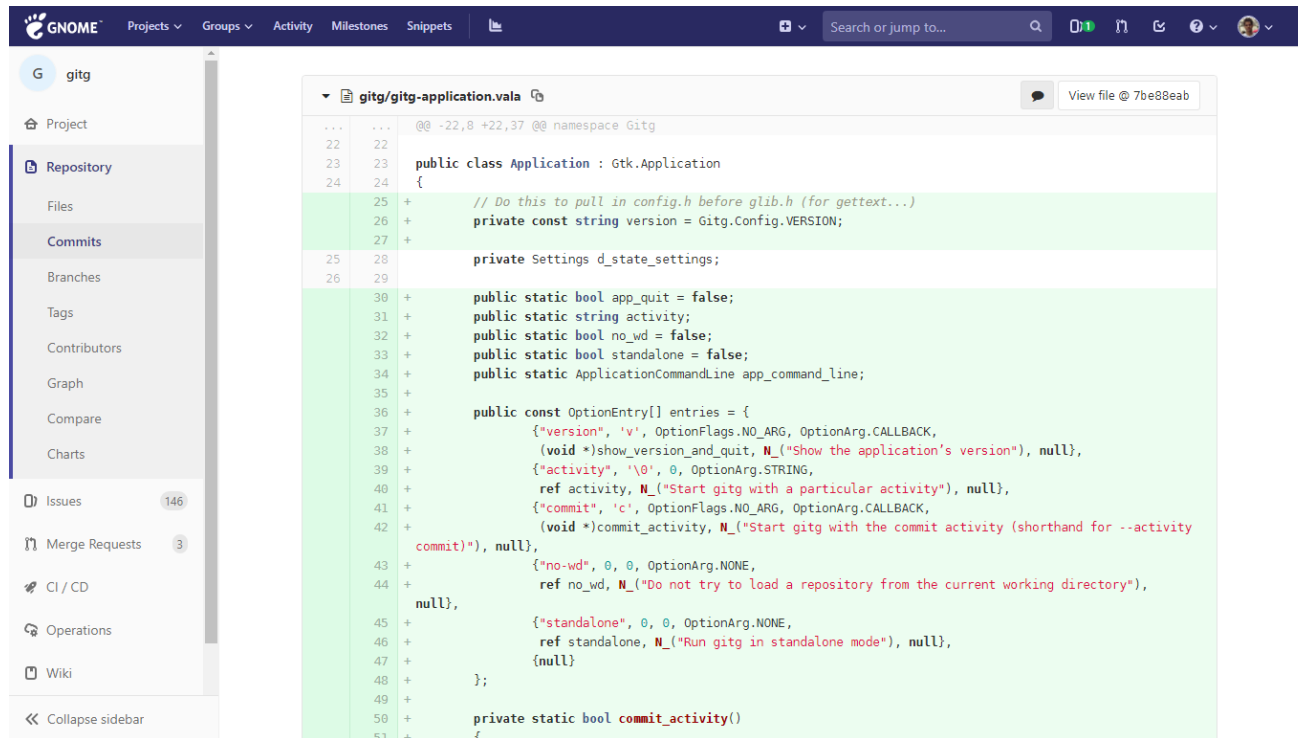
# Control de código

- Saber moverse por el repositorio <https://gitlab.gnome.org/GNOME/gitg>
  - Es importante entender como evoluciona el código
  - moverse por sus cambios
  - Probar diferentes cambios (usar ramas)
  - Probar las sugerencias de otros (Merge Request)
  - Escribir mensajes descriptivos en los commits



# Revisión de código

- Revisar el estilo del código
- Sugerir modificaciones a las contribuciones que otros te hacen
- Todo es público, puedes investigar otras revisiones para entender otras implementaciones

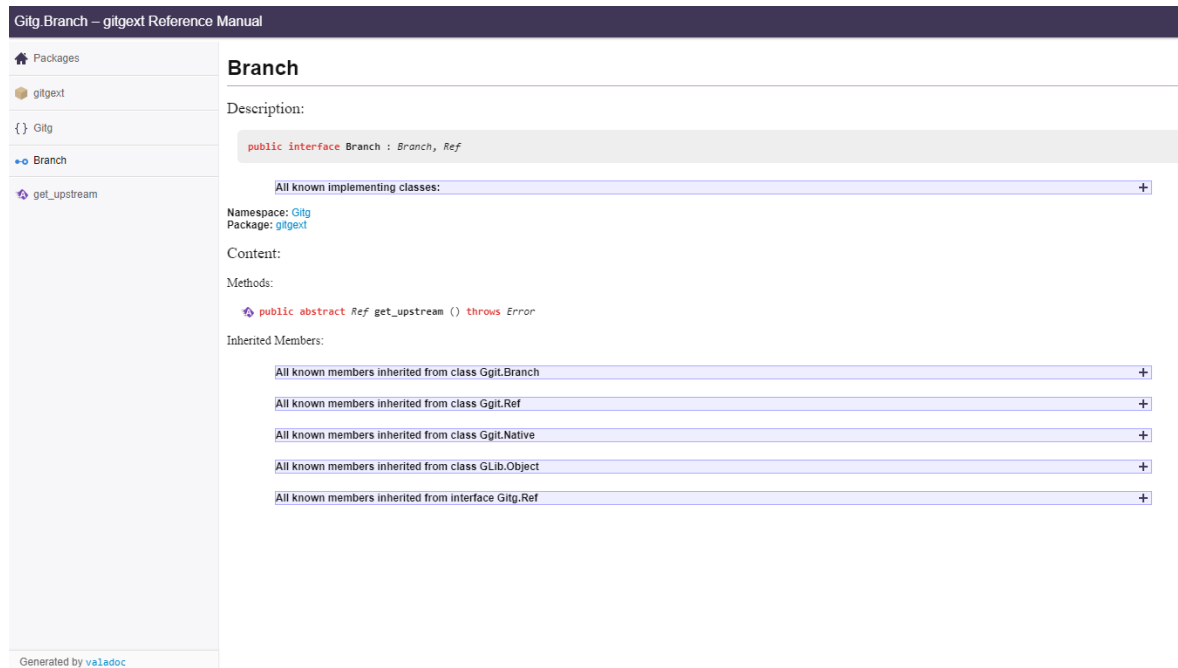


The screenshot shows the GNOME Git GUI interface. On the left, a sidebar contains navigation options: Project, Repository, Files, Commits, Branches, Tags, Contributors, Graph, Compare, Charts, Issues (146), Merge Requests (3), CI/CD, Operations, Wiki, and Collapse sidebar. The main area displays the code for `gitg/gtg-application.vala`. The code is shown with line numbers and highlights for changes. The code includes a `public class Application` that inherits from `Gtk.Application`. It defines a `private const string version` and a `private Settings d_state_settings`. It also defines several public static boolean and string variables, and a `public const OptionEntry[] entries` array. The `entries` array contains several entries for command-line options like `version`, `activity`, `commit`, `no-wd`, and `standalone`. The `commit` entry has a `callback` of `commit_activity`. The `commit_activity` method is defined as `private static bool commit_activity()`.



# Documentación

- Wiki: <https://wiki.gnome.org/Apps/Gitg/>
- Documentación online:  
<https://gnome.pages.gitlab.gnome.org/gitg/gitgext/index.htm>
  - Necesitas conocer lenguajes de marcado (Markdown, herramientas de generación de documentación a partir de comentarios de código)
- Permite que los usuarios y contribuidores puedan resolver sus dudas por si solos



Gitg Branch – gitgext Reference Manual

Packages

- gitgext
- Gitg
- Branch
- get\_upstream

## Branch

Description:

```
public interface Branch : Branch, Ref
```

All known implementing classes: +

Namespace: Glg  
Package: gitgext

Content:

Methods:

```
public abstract Ref get_upstream () throws Error
```

Inherited Members:

- All known members inherited from class Ggit.Branch +
- All known members inherited from class Ggit.Ref +
- All known members inherited from class Ggit.Native +
- All known members inherited from class GLib.Object +
- All known members inherited from interface Gitg.Ref +

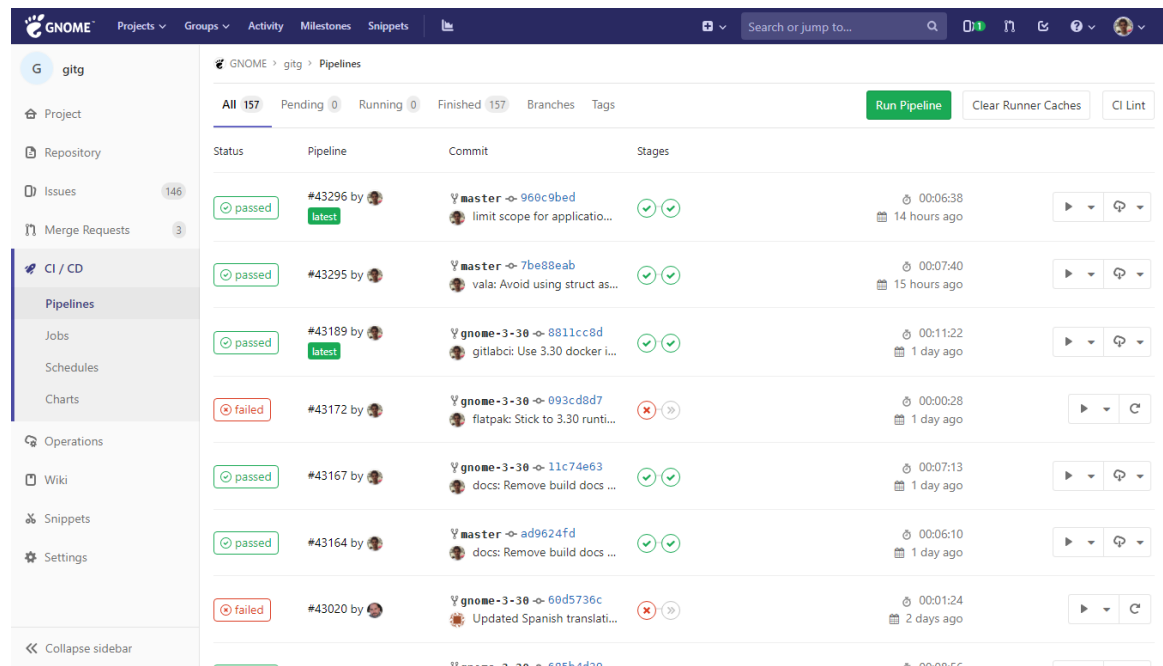
Generated by valadoc



# Integración continua

- Gitlab proporciona runners para ejecutar integración continua <https://gitlab.gnome.org/GNOME/gitg/pipelines>
  - Una ayuda fundamental, ya que evalúa cualquier sugerencia de cambio
  - Solo compilar el proyecto es una métrica perfecta
  - En este caso implica tecnologías como docker, flatpak, meson, ninja

Todas estas tecnologías son sencillas de aprender y muy potentes- Suite de test



| Status | Pipeline         | Commit  | Stages | Duration | Time         |
|--------|------------------|---|--------|----------|--------------|
| passed | #43296 by latest | Y master -> 960c9bed<br>limit scope for applicatio...       | ✓ ✓    | 00:06:38 | 14 hours ago |
| passed | #43295 by latest | Y master -> 7be88eab<br>vala: Avoid using struct as...      | ✓ ✓    | 00:07:40 | 15 hours ago |
| passed | #43189 by latest | Y gnome-3-30 -> 8811cc8d<br>gitlabci: Use 3.30 docker l...  | ✓ ✓    | 00:11:22 | 1 day ago    |
| failed | #43172 by latest | Y gnome-3-30 -> 093cd8d7<br>flatpak: Stick to 3.30 runti... | ✗      | 00:00:28 | 1 day ago    |
| passed | #43167 by latest | Y gnome-3-30 -> 11c74e63<br>docs: Remove build docs ...     | ✓ ✓    | 00:07:13 | 1 day ago    |
| passed | #43164 by latest | Y master -> ad9624fd<br>docs: Remove build docs ...         | ✓ ✓    | 00:06:10 | 1 day ago    |
| failed | #43020 by latest | Y gnome-3-30 -> 60d5736c<br>Updated Spanish translati...    | ✗      | 00:01:24 | 2 days ago   |



# Suite de Testing

- El unicornio del software (gitg no tiene mucho de esto)
  - Lleva mucho mantenimiento
  - Es crucial para controlar un proyecto
- Te ayudara a detectar regresiones
- Cubre muchos aspectos:
  - Test unitarios: Aislados del exterior, solo prueban tu código
  - Test integración: Prueban tu aplicación contra el mundo exterior
  - Test usabilidad: Prueban aplicaciones gráficas

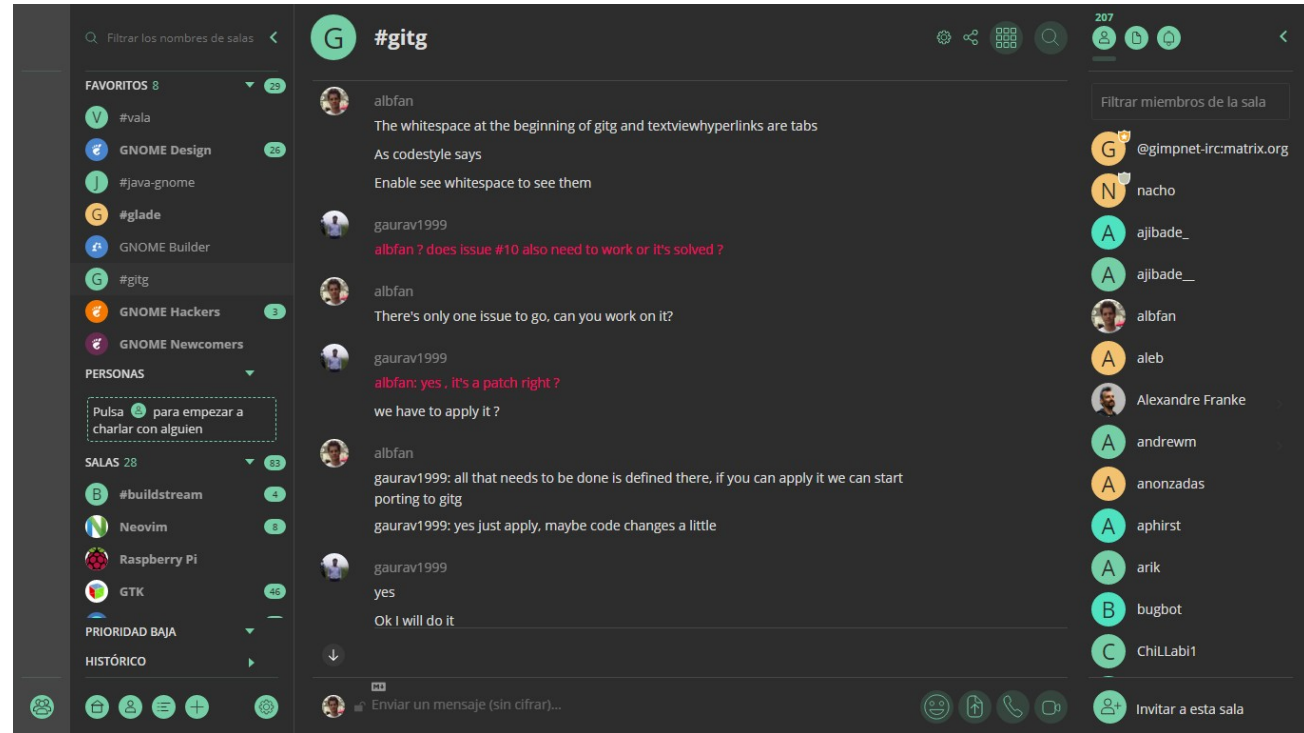


# Upstream/downstream

- Gitg usa proyectos, a los que a veces contribuye. Nos convertimos en usuarios de otros software.
- Gitg es distribuido por diferentes distribuciones y software. Para notificarles que hay nuevas versiones de gitg se usa una lista de correo
  - <https://mail.gnome.org/archives/ftp-release-list/2018-October/msg00045.html>
  - Esta muy automatizado
    - GPG para firmar las versiones
    - Ssh para subir archivos comprimidos con el código fuente



# Habilidades sociales: (INGLÉS)



- **Chat (IRC):** Hablar con otros desarrolladores, resolver temas puntuales, estar en contacto con contribuidores
- **Redes Sociales (Twitter):** Comunicar novedades, pedir colaboración, estar en contacto. Es necesario dar visibilidad al proyecto
- **Reuniones, meetings:** Verse con gente, compartir ideas, risas. Escuchar nuevas ideas.
- **Listas de correo:** Comunicaciones oficiales, temas importantes





# Preguntas





Gracias,  
Nos vemos en IRC!



**GNOME**<sup>™</sup>

[https://riot.im/app/#/room/#\\_gimpnet\\_#gitg:matrix.org](https://riot.im/app/#/room/#_gimpnet_#gitg:matrix.org)



[https://riot.im/app/#/room/#\\_gimpnet\\_#gitg:matrix.org](https://riot.im/app/#/room/#_gimpnet_#gitg:matrix.org)

