

Charla GitHub avanzado 2024

Ramas, comunicación, workflows

Alejandro Barrachina Argudo

2024

Universidad Complutense de Madrid

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Introducción

Charla anterior: <https://github.com/ALK222/charla-git-principiantes-2023>

Se asume que:

- Tenéis cuenta de GitHub
- Sabéis hacer crear repositorios
- Sabéis hacer *commit*, *push*, *pull*

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Ramas

- Mantienen el desarrollo paralelo para distintas partes del programa
- Mantienen separadas las versiones estables de las de desarrollo
- Nos permiten proteger ciertas ramas

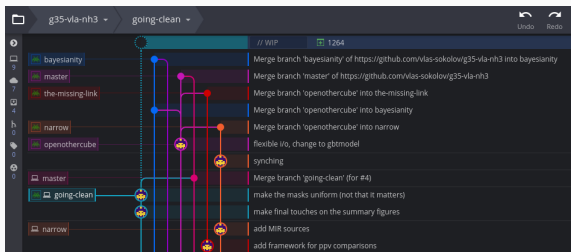


Figure 1: origen de la imagen: <https://stackoverflow.com/questions/46827112/toggling-branch-tree-view-in-gitkraken>

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Main

La rama Main (en proyectos antiguos también se puede encontrar como Master) es la rama principal del proyecto y la primera rama que se crea automáticamente al crear el repositorio.

- Rama (normalmente) protegida para evitar código malicioso (de manera intencional o no) sin supervisión.
- Generalmente es la rama estable del programa, solo se suben versiones definitivas

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Ramas secundarias

A estas ramas les daremos nosotros un nombre al crearlas. Son útiles para desarrollar nuevas características del programa o para hacer arreglos de bugs.

Estas ramas se pueden unir a la principal en cualquier momento del desarrollo para incorporar estos cambios a la rama principal.

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Merge

Un *Merge* nos permite unificar un conjunto de *commits* en un solo historial.

Normalmente esto se hace entre los extremos de dos ramas para fusionarlas en una sola.

Esto puede generar conflictos entre archivos que se pueden resolver con distintos editores de texto o herramientas de gestión de git.

Pull Request

Un *pull request* es un mecanismo de GitHub para poder hacer *merge* en proyectos ajenos o para solicitar feedback antes de hacer un merge en tu propio proyecto.

En ciertos repositorios es obligatorio pasar una *pull request* con varios supervisores para poder contribuir si no eres parte del equipo.

Tras una *pull request* se hace un merge entre dos ramas.

Fork

Un *fork* es un nuevo repositorio que comparte visibilidad y código con un repositorio original.

Nos permite hacer modificaciones a un código ya existente y después mover esos cambios al repositorio original con una *pull request* (o no).

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Github actions es una plataforma de automatización para IC/CD que nos permite crear flujos de trabajo y automatizar pruebas.

Gratis para repositorios públicos, de pago para repositorios privados (2000¹ minutos gratis y 500 MB gratis al mes).

¹Minutos de uso en procesos linux, los procesos en Windows cuentan 2x y en MacOS 10x

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Los archivos para github workflows se guardan en la carpeta `.github/workflows`.

Estos archivos son de formato *yml*

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Estructura básica

Estos archivos siempre tienen que empezar con un nombre para el proceso:

```
1 name: Build LaTeX document
```

Después definimos cuando ejecutar esta acción, puede hacerse en distintas interacciones:

```
1 on: [push, pull_request]
```

Estructura básica

Tras los pasos iniciales, empezamos con los “trabajos”. Los trabajos serán todas las cosas que van a ejecutarse dentro de esta acción. Estos trabajos tendrán un nombre expresado como una variable, un entorno de trabajo y una serie de pasos:

```
1 jobs:  
2   build_latex:  
3     runs-on: ubuntu-latest  
4     steps:
```

Estructura básica

Cada paso tendrá un nombre, un componente que usa para trabajar y posibles variables (dependen del componente):

```
1   - name: Upload PDF
2     uses: actions/upload-artifact@v3
3     with:
4       name: presentacion.pdf
5       path: ./presentacion/Presentacion.pdf
6       if-no-files-found: error # 'warn' or
  'ignore' are also available, defaults to `
  warn`
```

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Estas acciones se pueden usar para automatizar pruebas, hacer versionado de programas, subir archivos o incluso para jugar al ajedrez.

En el marketplace hay miles de acciones ya desarrolladas para hacer casi de todo.

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Repositorios especiales

Github se reserva el nombre de ciertos repositorios para crear lo que se denominan “repositorios especiales”.

Estos nombres son tu nombre de usuario y tu nombre de usuario seguido de “.github.io”. El primero crea una descripción que se verá en la portada de tu perfil, el segundo creará una página web principal para tu perfil de github.

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Github pages

GitHub pages es un servicio de GitHub que nos permite alojar páginas estáticas bajo el dominio “github.io”.

Estas páginas se pueden hacer en cualquier repositorio, posicionarse en distintas ramas y tener dos directorios fuente, la raíz del repositorio o una carpeta “/docs”

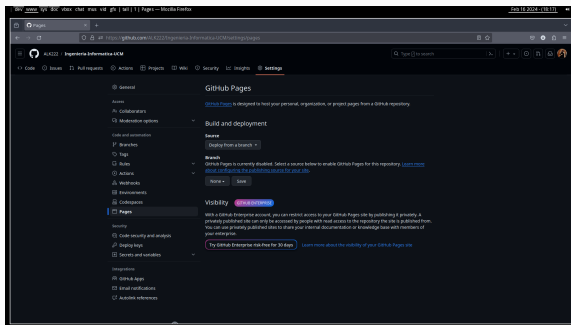


Figure 2: Opciones de GitHub pages antes de publicar la página

Tabla de contenidos

Introducción

Ramas

Main

Ramas secundarias

Pull request y Merge

Workflows / Actions

Archivos

Estructura básica

Usos

Repositorios especiales

Github Pages

Jekyll

Estas páginas de github están construidas con Jekyll, permitiéndonos esto hacer pruebas en local, usar HTML o markdown y utilizar todos los plugins disponibles para Jekyll





Alejandro Barrachina

alejba02@ucm.es

github.com/alk222

