

# Dispositivo de axuda ao peregrinaxe

Sesión 1: Introducción y estandarizado de ferramentas a utilizar



**Oscar Blanco Novoa**  
**Alejandro Taracido Cano**

Este obra está bajo una licencia de [Creative Commons](https://creativecommons.org/licenses/by-sa/4.0/)  
[Reconocimiento-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/)

# ¿Quiénes somos?



**Oscar Blanco Novoa**

Ingeniero en Informática

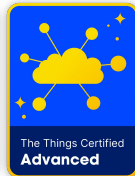
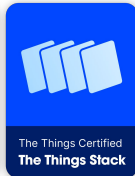
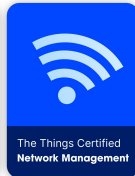
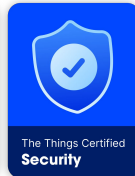
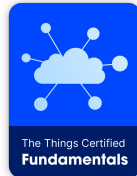
Investigador en el departamento de Ingeniería de Computadores y doctor en Informática en la Universidad de A Coruña.



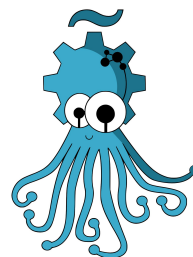
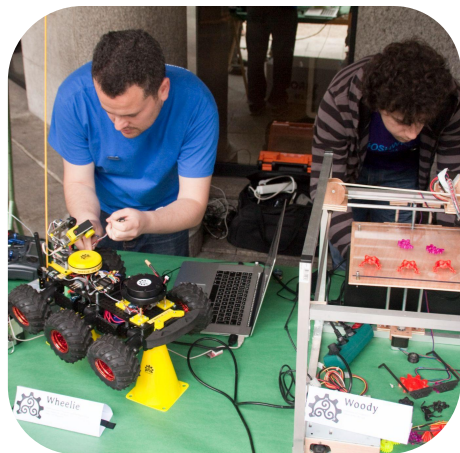
**Alejandro Taracido Cano**

Ing. Tec. Industrial Electrónico

Ingeniero desarrollo I+D+i en ELINSA  
(Grupo Amper)



# Sobre todo makers inquietos



# Introducción a las sesiones

- El proyecto
- En qué podemos ayudaros
- Metodología de trabajo
- Sistema de hitos

# Sesión 1: Intro y estandarizado de herramientas a utilizar

- Platformio
- Visual studio code
- Git básico
- Colaboración en repositorio único
- Organización del código fuente
- Metodología de integración software

Hito para la siguiente sesión:

- Prototipo módulos con cables y breadboard
  - Hello world con cada módulo
  - Integración de todos los módulos
  - Pruebas funcionales preliminares

# Sesión 2: Montaje prototipo perfboard pruebas y MVP

- Revisión integración SW
- MVP
- Orientación prototipado en perfboard
- Creación de esquema en papel
- Creación de software para validación

Hito para la siguiente sesión:

- Prototipo en perfboard
  - Prototipo, validación y esquema
  - Prueba MVP

# Sesión 3: Esquema y elección de componentes

- Captura esquema eléctrico KiCad
- Orientación selección de componentes finales

Hito para la siguiente sesión:

- Diseño eléctrico PCB
  - Diseño eléctrico
  - Sustitución de módulos por componentes definitivos
  - Cierre diseño esquema eléctrico definitivo

# Sesión 4: Diseño PCB

- DFM y reglas de diseño
- Creación y comprobación footprints KiCad
- Dimensionado PCB y colocación de componentes
- Capas
- Ruteo pistas y planos de masa
- Silkscreen
- DRC

Hito para la siguiente sesión:

- Diseño PCB
  - Creación y comprobación footprints
  - Dimensionado placa y colocación de componentes
  - Ruteo
  - DRC
  - Cierre diseño PCB



# Sesión 5: Pedido PCB

- Revisión diseño PCB
- Configuración generación de gerbers
- Como efectuar pedido PCB
- Estrategias de acopio de componentes

Hito para la siguiente sesión:

- Pedido PCB
  - Cierre diseño PCB
  - Generación de gerbers
  - Pedido PCB
  - Acopio componentes faltantes

# Sesión 6: Mejora software

- Mejora del software
- Adición de funcionalidades
- Infraestructura cloud
- Aplicación básica

Hito para la siguiente sesión:

- Versión SW para PCB
  - Pulido del firmware y adición de funcionalidades (red mesh tipo disaster radio, sistema de alerta de caídas, sistema de petición de SOS, conexión por bluetooth con el móvil....)
  - Diseño infraestructura de comunicaciones (cloud)
  - Diseño software básico de aplicación
  - Preparación software test PCB
  - Recepción PCBs

# Sesión 7: Montaje PCB y pruebas

- Soldadura SMD
- PCB bring up
- Pruebas eléctricas
- Pruebas software
- Debugging
- Problem solving
- Evaluación necesidad de nueva iteración

Hito para la siguiente sesión:

- Montaje PCBs
  - Soldadura prototipos
  - Pruebas eléctricas
  - Pruebas software
  - Pruebas de campo
  - Recopilación datos y análisis

# Sesión 8: Diseño del dispositivo final

- Impresión 3D
- Encapsulado final
- Licenciamiento Open Source
- Documentación y publicación del proyecto
- Creación de comunidad

Hito para la siguiente sesión:

- Diseño del dispositivo final
  - Carcasa 3D, medidas, etc.

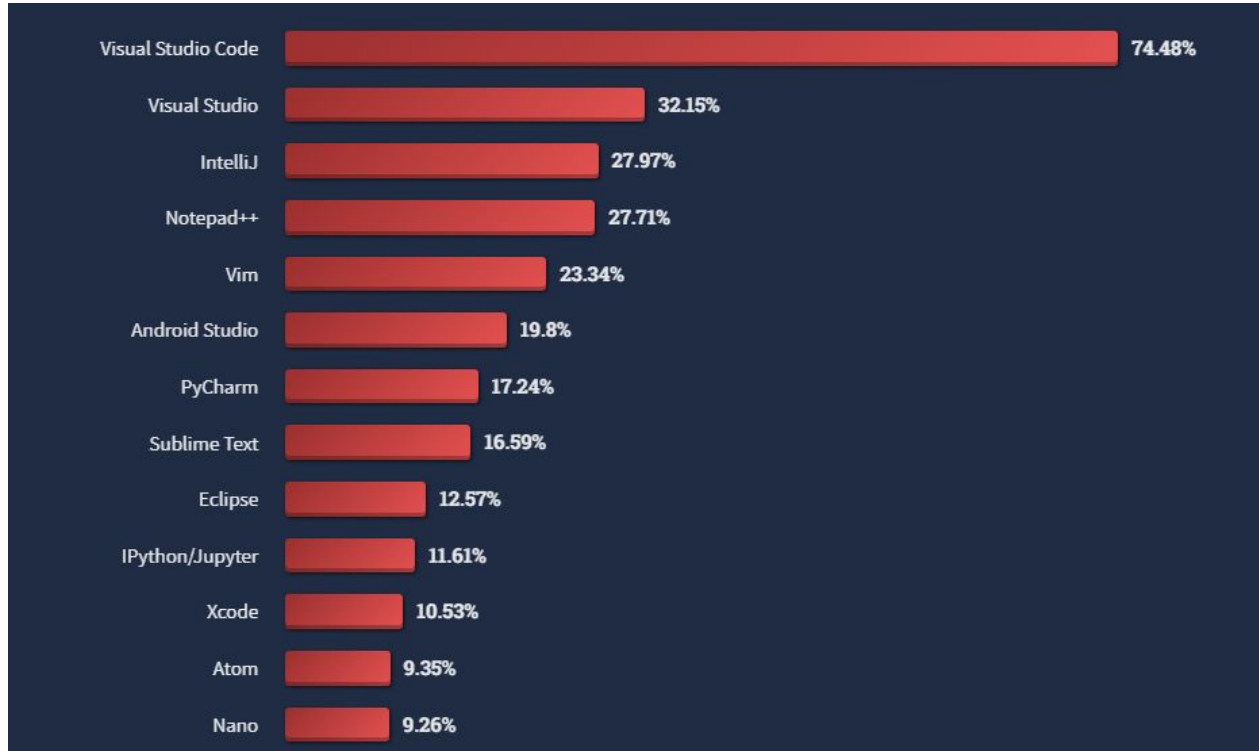
# Visual Studio Code



# Visual Studio Code: Características

- Ligero
- Multiplataforma
- Muy personalizable
- Soporta casi todos los lenguajes de programación
- Posee cientos de extensiones para completar su funcionalidad

# Visual Studio Code



# Visual Studio Code

**Instalación:**

<https://code.visualstudio.com/download>

**Documentación:**

<https://code.visualstudio.com/docs>



# PlatformIO



# Platformio vs Arduino IDE

- Compatible con editores profesionales como VSC
- Mejor gestor de proyectos
- Gestión de librerías y dependencias
- Compatibilidad con distintos controladores y plataformas
- Mayor eficiencia y velocidad de compilado
- Herramientas avanzadas y automatización
- Desarrollo colaborativo
- Soporte para pruebas unitarias

# PlatformIO

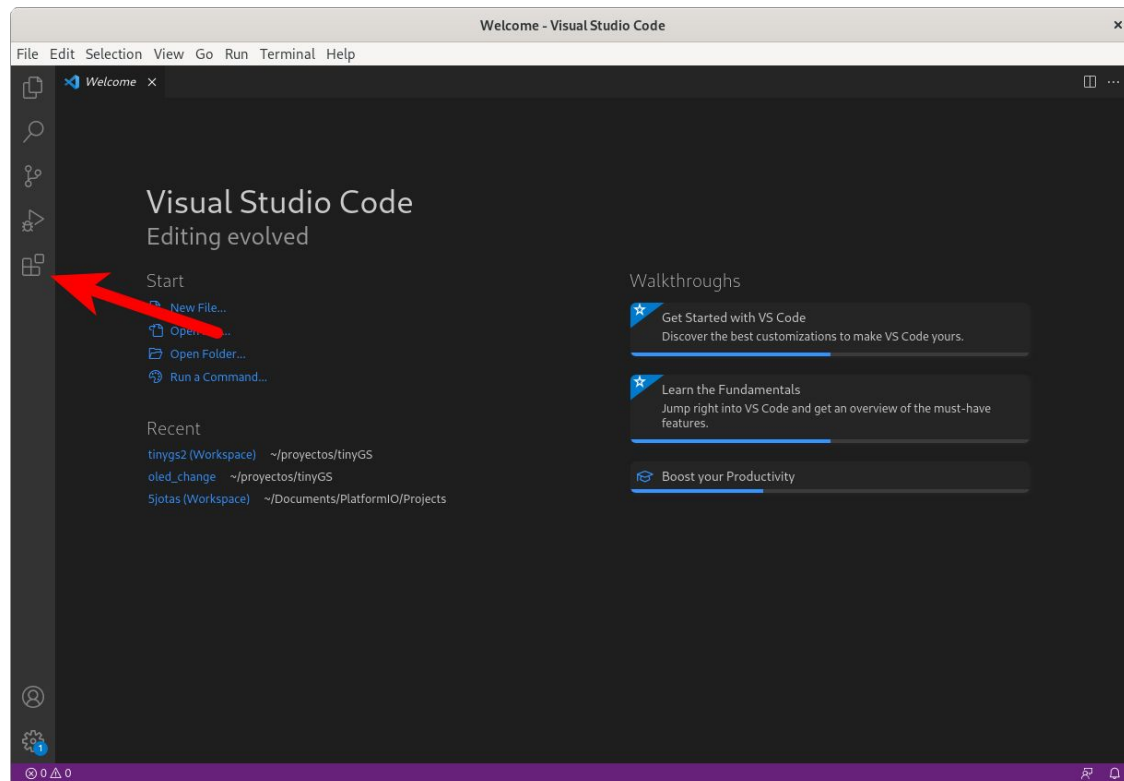
## Instalación:

Mediante menú extensiones en VSC

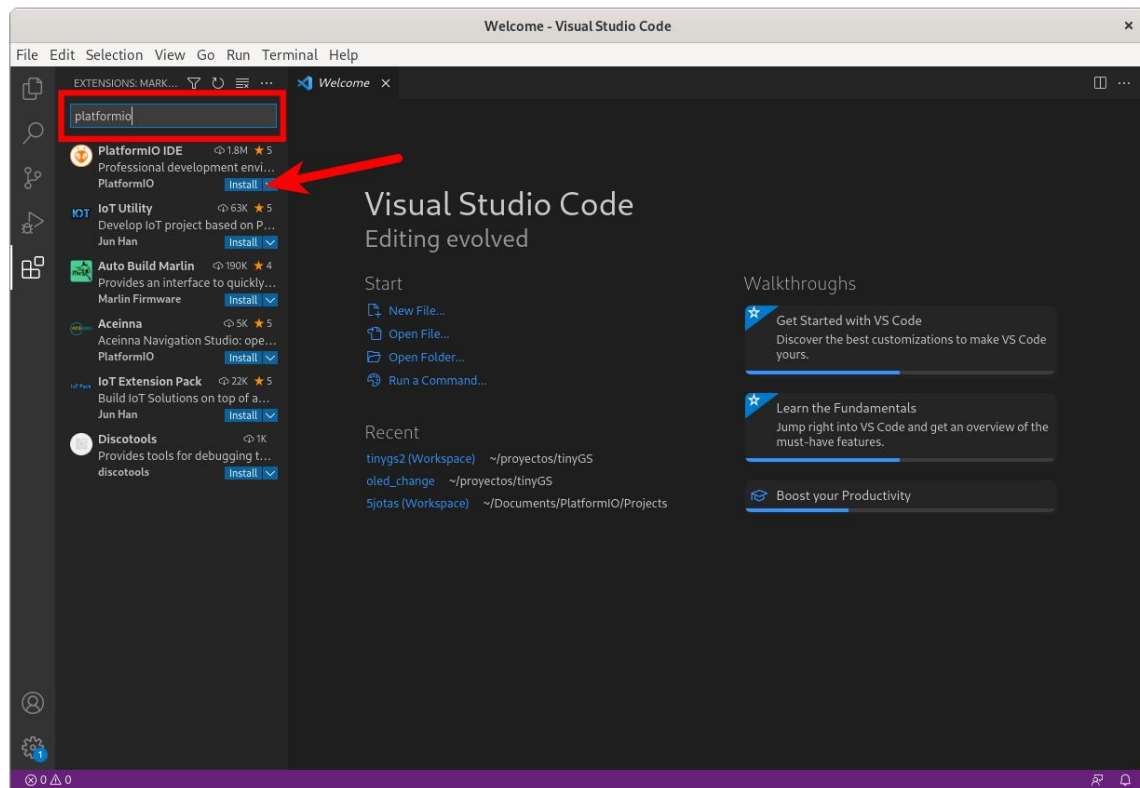
## Documentación:

<https://docs.platformio.org>

# PlatformIO: Instalación



# PlatformIO: Instalación



**GIT**



# ¿Qué es un sistema de control de versiones?

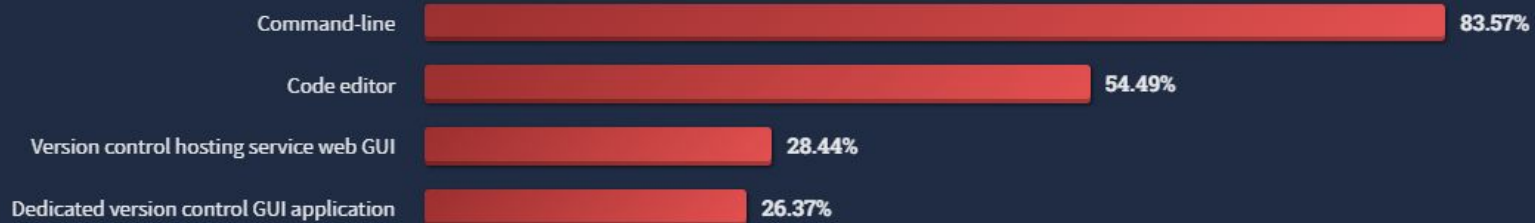
- Tener backups
- Mantener un historial
- Ver cambios pasados
- Hacer experimentos
- Colaborar
- Tipos:
  - Centralizados
  - Descentralizados

# ¿Por qué Git?

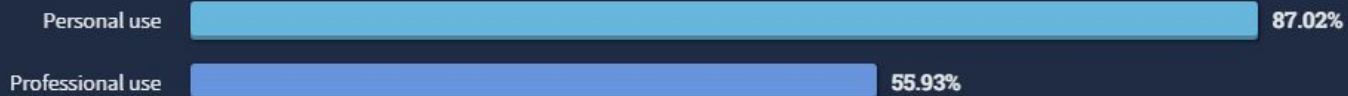
- Distribuido
- Es rápido
- Integridad de datos
- Área de preparación (staging area)
- Open source y gratuito



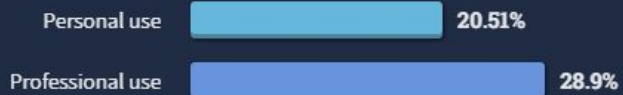




### GitHub



### GitLab

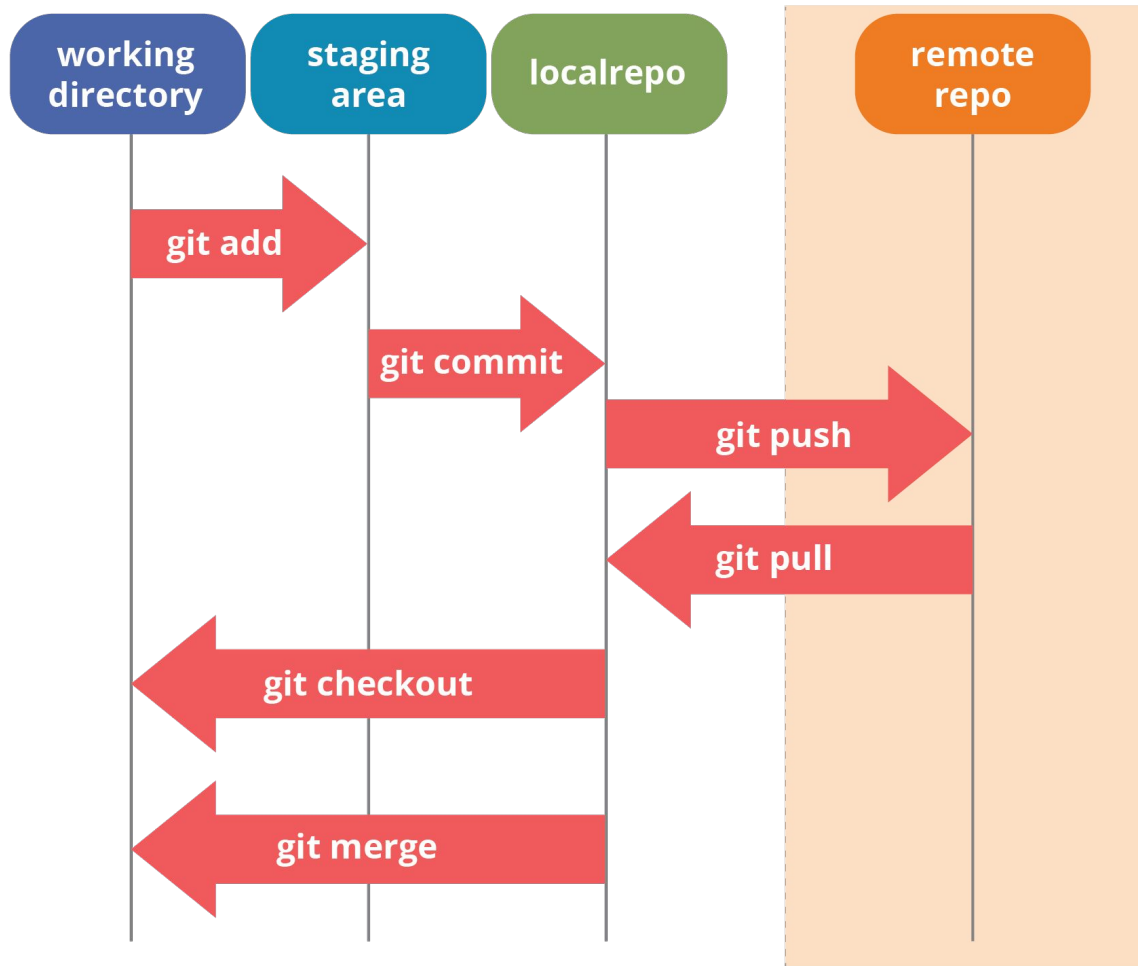


### Bitbucket



## Local

## Remote



# GIT

Instalación:

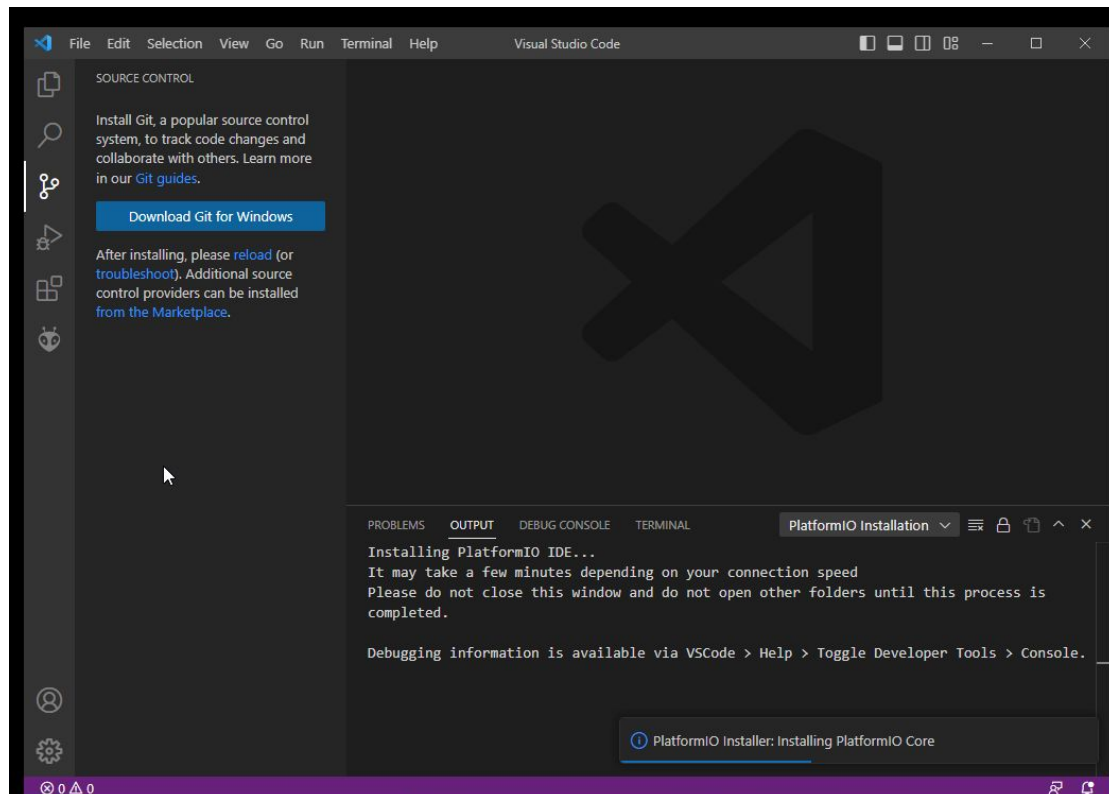
Mediante menú source control en VSC o

<https://git-scm.com/downloads>

Documentación:

<https://git-scm.com/doc>

# GIT: Instalación



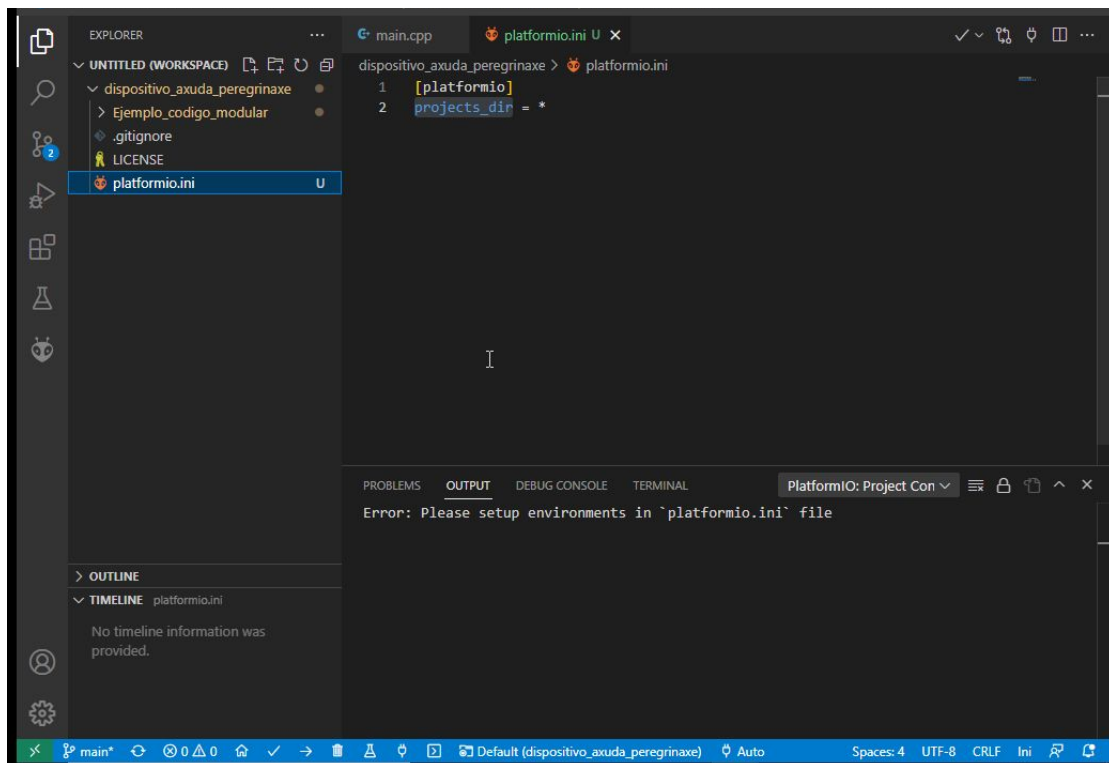
# GIT: Configuración nombre y correo

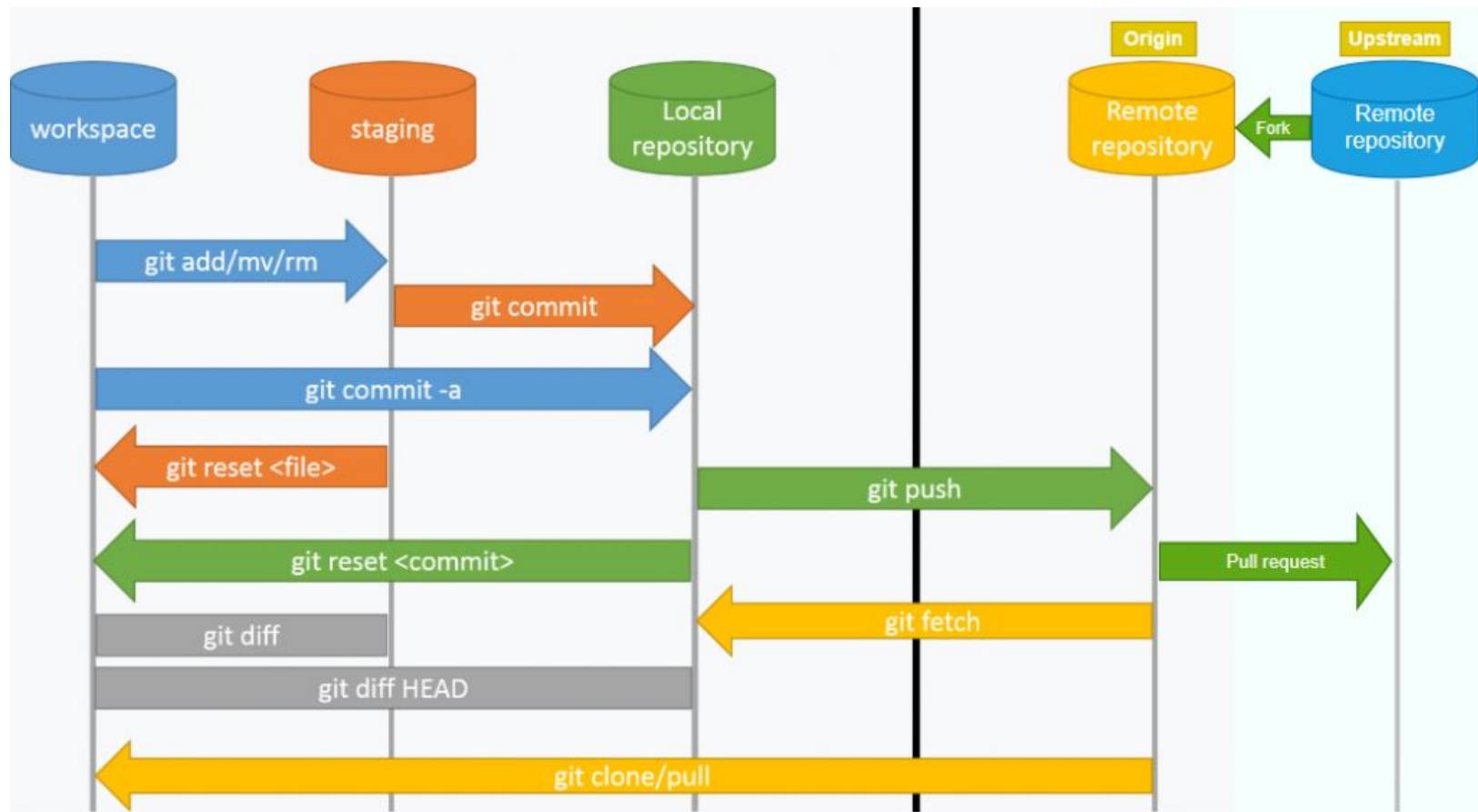
Antes de empezar a utilizar Git, es recomendable configurar nombre y dirección de correo electrónico.

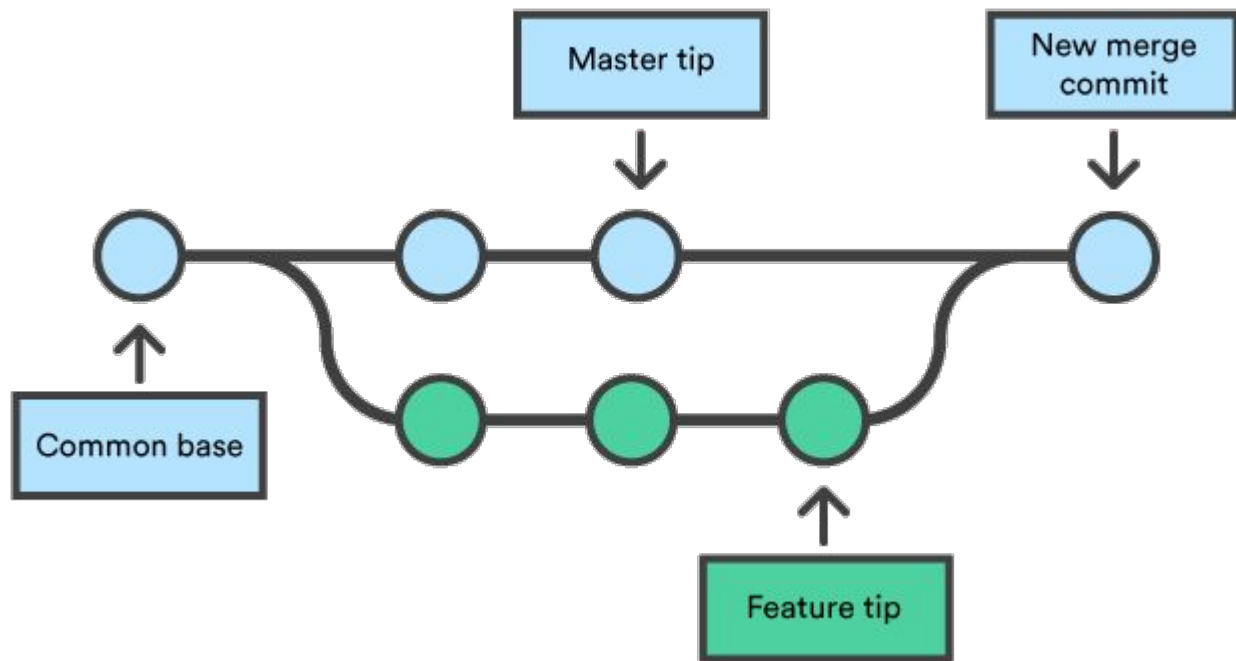
```
git config --global user.name "Tu nombre"
```

```
git config --global user.email "Tu correo electrónico"
```

# GIT: Varios proyectos platformIO en un mismo repo

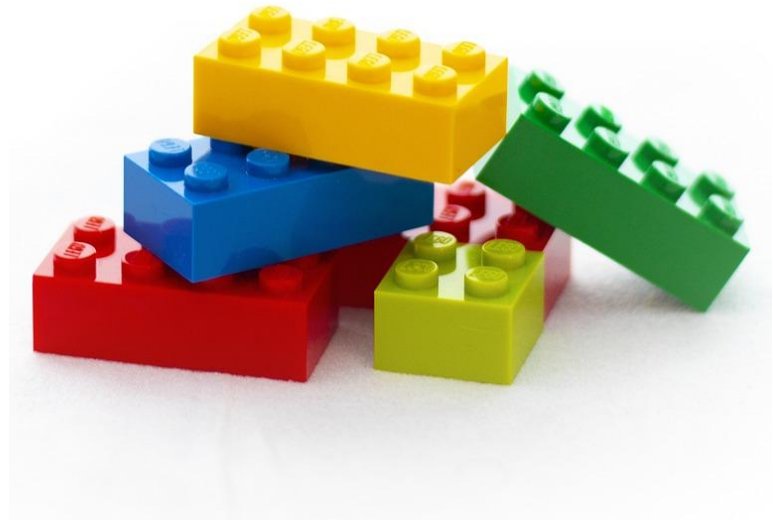








# Organización código fuente



# Organización del código fuente

Repositorio de ejemplo:

[https://github.com/open-workshops/dispositivo\\_axuda\\_peregrinaxe](https://github.com/open-workshops/dispositivo_axuda_peregrinaxe)

# Organización del código fuente

- Separar cada módulo en archivos distintos (.h y .cpp)
- Cada módulo expone sus variables y funciones en el .h
- Las variables siempre lo más local posible
- Programar una función setup y loop en cada módulo
- El main incluye los .h de todos los módulos
- El main debe llamar a todos los setups y loops de los módulos
- Adicionalmente el main hace de coordinador
  - Solicita información a cada módulo
  - Desencadena acciones en los módulos
  - Gestiona temporizadores

Repositorio de ejemplo:

[https://github.com/open-workshops/dispositivo\\_axuda\\_peregrinaxe](https://github.com/open-workshops/dispositivo_axuda_peregrinaxe)

# Metodología integración software (integration hell)



# Metodología integración software

- Antes de comenzar el desarrollo crear archivo compartido de pinado
- Crear proyecto *hello world* por cada periférico manteniendo estructura
- Crear todas las funciones que estimemos sean necesarias
- Tener en cuenta los conflictos de uso de periféricos del micro
  - Varios módulos usando el mismo puerto de comunicación (SPI, I2C, Serie...)
- Una vez listos todos los proyectos independientes se comienza integración
  - Empezar con el proyecto más complejo como base
  - Añadir solo un proyecto/módulo cada vez y comprobar funcionamiento de todas las partes
  - Si la integración de un proyecto genera un conflicto, intentar resolverlo en el proyecto más sencillo (cambio librería, puerto, pin, comunicación...)

# Desarrollo proyecto hello world

- Buscar información del periférico (datasheet)
- Buscar pinado con clave “arduino pinout”
- Buscar ejemplos con con la palabra clave “arduino” o “ESP32”
- Buscar librerías que faciliten el trabajo
  - <https://registry.platformio.org/search>
- Abstraer el uso de librería con estructura de módulos (.cpp .h)
- Otros recursos:
  - StackExchange
  - chatGPT
  - Colaboración de compañeros, amigos, familiares y mascotas

An orange rectangular sticky note is pinned to a light gray surface with a green pushpin. The pushpin is located at the top center of the note. The text "Hitos próxima sesión" is written in black cursive on the note. A shadow is cast by the note and the pushpin onto the surface to the left.

Hitos próxima  
sesión

# Hitos próxima sesión

## Prototipo módulos con cables y breadboard

- Hello world con cada módulo
- Integración de todos los módulos
- Pruebas funcionales preliminares



¡Muchas gracias!