



Cursos Abiertos de Programación de Sistemas Embebidos

Nivel 1 - Introducción a la Programación de Microcontroladores
Repositorios Git

Profesores: Ing. Eric Pernia, Dr. Ing. Pablo Gómez



Asociación Civil para la Investigación,
Promoción y Desarrollo de los
Sistemas Electrónicos Embebidos



1. Repositorio de archivos y control de versiones.
2. Git.
3. Github.
4. Instalación de Git.
5. Integración de Git en Embedded IDE.
6. Ejemplo de uso de git y github.
 - Creación de una cuenta en github.
 - Creación de un repositorio.
 - Clonar el repositorio.
 - Transferir cambios desde el repositorio local al remoto y viceversa.
 - Asignar permisos en el repositorio.
 - Flujo de trabajo para el Trabajo Práctico grupal.



Sistema de control de versiones y Repositorios

Sistemas de control de versiones

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

Una de las claves de trabajar de forma profesional en programación es disponer de un sistema de control de versiones de archivos, ya sea SVN, Git u otros.

¿Qué es un repositorio?

El repositorio es el lugar en el que se almacenan los datos actualizados e históricos de cambios. A veces se le denomina depósito o depot. Puede ser un sistema de archivos en un disco duro, un banco de datos, etc.



git

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

Es un sistema distribuido de control de código fuente o SCM (en inglés Source Code Management).

¿Qué nos aporta git?

- Auditoría del código: saber quién ha modificado qué y cuándo.
- Control sobre cómo ha cambiado nuestro proyecto con el paso del tiempo.
- Volver hacia atrás de una forma rápida.
- Control de versiones a través de etiquetas: versión 1.0, versión 1.0.1, versión 1.1, etc. Sabremos exactamente qué había en cada una de ellas y las diferencias entre cualquiera de ellas dos.
- Seguridad: todas las estructuras internas de datos están firmadas con SHA1. No se puede cambiar el código sin que nos enteremos.
- Mejora nuestra capacidad de trabajar en equipo.
- Merging y branching extremadamente eficientes.



GitHub



GitHub es una plataforma de desarrollo colaborativo de software (se les llama forge) para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails desarrollado por GitHub, Inc. (anteriormente conocida como Logical Awesome).

Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc.

El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta paga.



Instalación de Git

- En linux generalmente ya viene instalado.
- En windows descargar e instalar alguno de los siguientes:
 - <https://git-for-windows.github.io/>
 - <https://git-scm.com/download/win>
 - <https://tortoisegit.org/>



Integración de Git en Embedded IDE

- Embedded IDE desde la versión 0.4.0 se puede integrar git fácilmente desde el menú “Herramientas” donde se puede cargar en dicho menú comandos que pueden ejecutar programas externos.
- En linux la integración es muy sencilla. Se deben agregar los siguientes comandos desde el manejador de herramientas:

Nombre:	Comando:
Git: INIT	git init
Git: CLONE	git clone \${{text: URL}}
Git: PULL	git pull
Git: estado actual	git status
Git: añadir	git add \${{text: Archivos}}
Git: commit	git commit -a -s -m \${{text: Mensaje}}
Git: PUSH	git push
Git: Ver resumen logs	git log --pretty=oneline --abbrev-commit
Git: checkout	git checkout \${{text: branch o hash}}
Git: Logs completos	git log --pretty=fuller
Git: cambios actuales	git diff

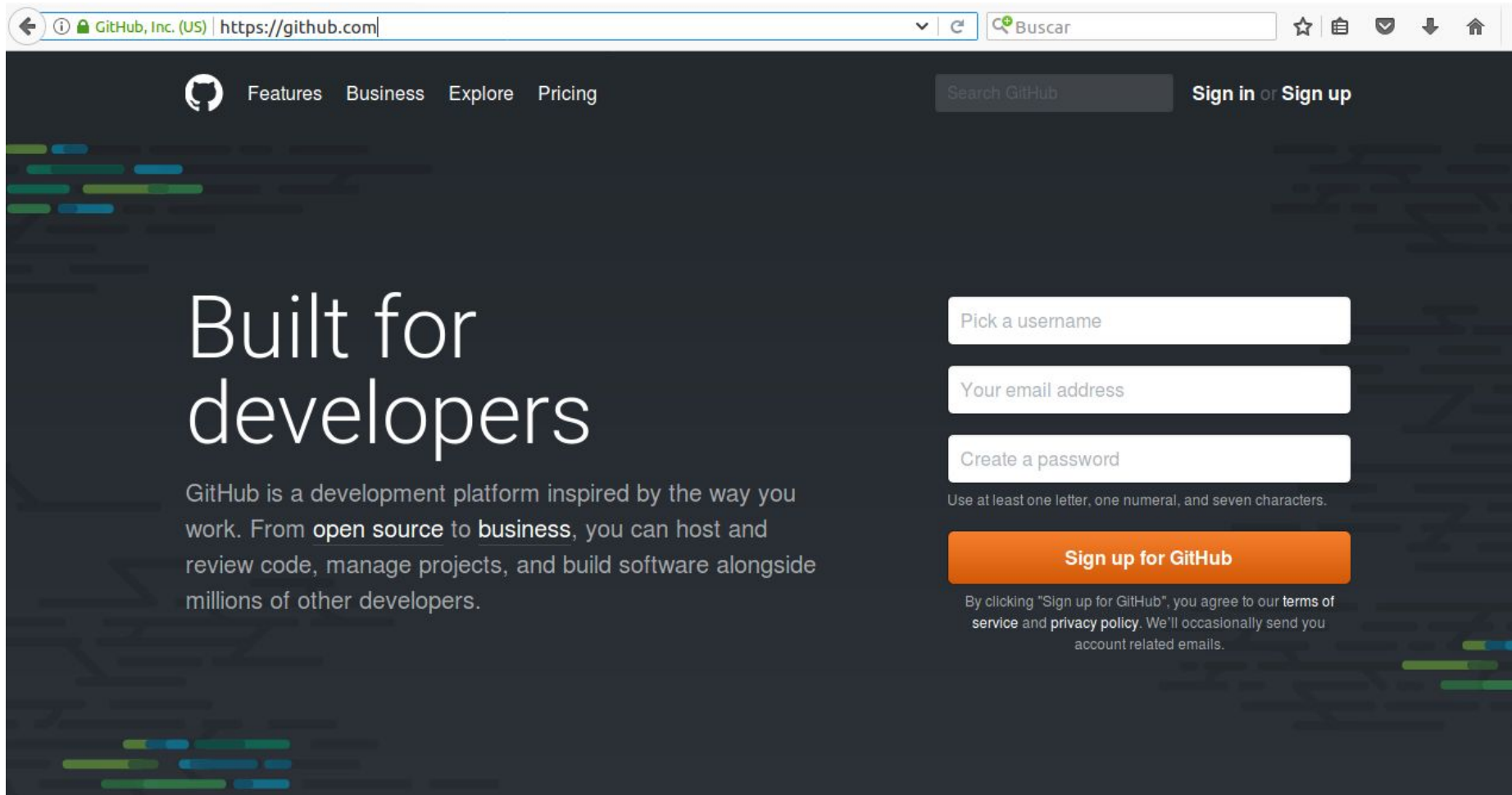
- En windows próximamente veremos como realizarlo. De todas formas siempre se puede utilizar git directamente desde la consola.



Ejemplo de uso de git y github

Creación de una cuenta en github

- Ingresar y registrarse en <https://github.com/>

A screenshot of the GitHub website's sign-up page. The browser's address bar shows 'https://github.com/'. The page has a dark blue background with a pattern of horizontal lines in green and blue. The main heading 'Built for developers' is in large white text. Below it, a paragraph describes GitHub as a development platform. On the right side, there are three input fields for 'Pick a username', 'Your email address', and 'Create a password'. Below these fields is an orange 'Sign up for GitHub' button. At the bottom right, there is a disclaimer about agreeing to terms of service and privacy policy.

GitHub, Inc. (US) | <https://github.com/> | Buscar

Features Business Explore Pricing

Search GitHub Sign in or Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside millions of other developers.

Pick a username

Your email address

Create a password

Use at least one letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails.



Ejemplo de uso de git y github

Creación de un repositorio

Se debe cargar el nombre del repositorio (sin espacios) y una breve descripción del mismo.

← ⓘ GitHub, Inc. (US) | https://github.com/new

Create a new repository

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

Owner **Repository name**

epernia /

Great repository names are short and memorable. Need inspiration? How about **literate-happiness**.

Description (optional)

☒ **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



Ejemplo de uso de git y github

Clonar el repositorio

- Para crear una copia local del repositorio remoto creado en github se debe abrir una terminal e ir a la ruta de la carpeta donde se quiere crear una copia local y ejecutar:

Crea una copia local del repositorio ejecutando

```
git clone /path/to/repository
```

Si utilizas un servidor remoto, ejecuta

```
git clone username@host:/path/to/repository
```

Por ejemplo:

```
git clone https://github.com/epernia/sAPI.git
```

Ejemplo de uso de git y github

Transferir cambios desde el repositorio local al remoto y viceversa

Tu repositorio local esta compuesto por tres "árboles" administrados por git. El primero es tu **Directorio de trabajo** que contiene los archivos, el segundo es el **Index** que actua como una zona intermedia, y el último es el **HEAD** que apunta al último commit realizado.





Ejemplo de uso de git y github

Transferir cambios desde el repositorio local al remoto y viceversa

- Mediante el comando **git pull** se bajan los cambios del **repositorio remoto** al **repositorio local**.
- Con el comando **git add archivo** se añade un archivo al **index** de git. Con **git add .** se añaden al index todos los archivos sin seguimiento en el directorio de trabajo.
- Utilizando el comando **git commit -m"mensaje descriptivo"** se guarda en **head** el cambio realizado.
- Si se quieren subir los cambios del repositorio local al remoto se debe ejecutar **git push**



Ejemplo de uso de git y github

Asignar permisos en el repositorio

- Para que otros puedan realizar push a un repositorio creado desde cierto usuario se deberán agregar como colaboradores:

The screenshot shows the GitHub interface for the repository 'epernia / trabajoNivel1'. The 'Settings' tab is selected, and the 'Collaborators' section is active. The page indicates that no collaborators have been added yet. A search bar is provided to find users by username, full name, or email address. The 'Add collaborator' button is highlighted with a red box.

GitHub, Inc. (US) | <https://github.com/epernia/trabajoNivel1/settings> | Buscar

This repository Search Pull requests Issues Gist

epernia / trabajoNivel1

Unwatch 1 Star 0 Fork 0

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

Collaborators Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

Add collaborator



Ejemplo de uso de git y github

Flujo de trabajo con github para el Trabajo Práctico grupal

- Entre los integrantes del trabajo se deberá armar el esqueleto del programa modularizado en varios archivos .c y .h y se distribuirán la tarea de su realización luego de definir las funciones necesarias y sus parámetros (API).
- Cada integrante del grupo generará una cuenta en github.
- Uno de ellos será el responsable de crear el repositorio para el trabajo práctico y su mantenimiento.
- El responsable del repositorio deberá clonar el repositorio remoto de forma local y cargar el esqueleto, luego deberá subirlo al repositorio remoto.
- El responsable del repositorio debe añadir a sus compañeros como colaboradores.
- Cada integrante deberá trabajar en sus módulos y el responsable del repositorio deberá ser quien los junte en la aplicación final.



Ejemplo de uso de git y github

Flujo de trabajo con github para el Trabajo Práctico grupal

- Una vez realizados los pasos anteriores el resto de los integrantes deberán clonar el repositorio.
- El modo de trabajo colaborativo es el siguiente:
 - Antes de editar cualquier archivo se comienza realizando un pull del repositorio remoto para traer los últimos cambios.
 - Se editan los archivos asignados.
 - Se añaden los nuevos archivos mediante **add** en caso que existan.
 - Se realiza el **commit**.
 - Se envían los cambios al servidor remoto mediante **push**.
- En caso de problemas con git consulte en seguida.





Bibliografía

- Git, la guía sencilla:

<http://rogerdudler.github.io/git-guide/index.es.html>



¡Muchas gracias!

Seguinos:

 /ProyectoCIAA

 /ProyectoCIAA

 @ProyectoCIAA



www.proyecto-ciaa.com.ar