1. Que es GitHub?

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. El c´odigo se almacena de forma p´ublica, aunque

tambi´en se puede hacer de forma privada, creando una cuenta de pago.

2. Para qu´e sirve?

GitHub aloja tu repositorio de c´odigo y te brinda herramientas muy ´utiles para el trabajo

en equipo, dentro de un proyecto.

Adem´as de eso, puedes contribuir a mejorar el software de los dem´as. Para poder alcanzar

esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls.

1

3. Qu´e herramientas proporciona?

En la actualidad, GitHub es mucho m´as que un servicio de alojamiento de c´odigo. Adem´as

de ´este, se ofrecen varias herramientas ´utiles para el trabajo en equipo. Entre ellas, caben

destacar:

a)Una wiki para el mantenimiento de las distintas versiones de las p´aginas.

b)Un sistema de seguimiento de problemas que permiten a los miembros de tu equipo detallar

un problema con tu software o una sugerencia que deseen hacer.

c)Una herramienta de revisi´on de c´odigo, donde se pueden a˜nadir anotaciones en cualquier

punto de un fichero y debatir sobre determinados cambios realizados en un commit espec´ıfico.

d)Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas

de nuestro repositorio.

4. Qu´e uso le daremos?

En nuestra especialidad ”Programaci´on”, fuimos aprendiendo cosas y creando programas de

c´odigo abierto, fomentando el software libre; es por eso que presentamos esta gran herramienta enfocada al crecimiento de proyectos comunitarios y libres.

En esta p´agina podremos crear una cuenta gratuita y comenzar a subir repositorios de c´odigo

2

(o crearlos desde 0), para que con la ayuda de todos ese proyecto mejore; as´ı como tambi´en

fortalecer los proyectos de los dem´as para crecer como grupo.

5. Qu´e es Git?

Git es un sistema de control espec´ıfico de versi´on de fuente abierta creada por Linus Torvalds

en el 2005.

Espec´ıficamente, Git es un sistema de control de versi´on distribuida, lo que quiere decir que

la base del c´odigo entero y su historial se encuentran disponibles en la computadora de todo

desarrollador, lo cual permite un f´acil acceso a las bifurcaciones y fusiones.

6. Necesidad de un control de versiones

El control de versiones es una de las tareas fundamentales para la administraci´on de un proyecto de desarrollo de software en general. Surge de la necesidad de mantener y llevar control

del c´odigo que vamos programando, conservando sus distintos estados. Es absolutamente necesario para el trabajo en equipo, pero resulta ´util incluso a desarrolladores independientes.

Para facilitarnos la vida existen sistemas como Git, Subversion, CVS, etc. que sirven para

controlar las versiones de un software y que deber´ıan ser una obligatoriedad en cualquier

desarrollo. Nos ayudan en muchos ´ambitos fundamentales, como podr´ıan ser:

Comparar el c´odigo de un archivo, de modo que podamos ver las diferencias entre versiones

Restaurar versiones antiguas

3

Fusionar cambios entre distintas versiones

Trabajar con distintas ramas de un proyecto, por ejemplo la de producci´on y desarrollo

7. Tipos de sistemas de control de versiones

Comenzaron a aparecer los sistemas de control del versionado del software all´a por los a˜nos

setenta, aunque al principio eran bastante elementales. Para hacerse una idea, en los primeros

sistemas exist´ıa una restricci´on por la que s´olo una persona pod´ıa estar a la vez tocando el

mismo c´odigo. Es posible imaginarse que cosas semejantes provocaban retraso en los equipos

de trabajo, por ello, a lo largo de los a˜nos fueron surgiendo nuevos sistemas de control de

versiones, siempre evolucionando con el objetivo de resolver las necesidades de los equipos

de desarrollo.

Sistemas de control de versiones locales

Un m´etodo de control de versiones usado por mucha gente es copiar los archivos a otro

directorio (quiz´as indicando la fecha y hora en que lo hicieron, si son avispados). Este enfoque

es muy com´un porque es muy simple, pero tambi´en tremendamente propenso a errores.

Es f´acil olvidar en qu´e directorio te encuentras, y guardar accidentalmente en el archivo

equivocado o sobrescribir archivos que no quer´ıas.

Para hacer frente a este problema, los programadores desarrollaron hace tiempo VCSs locales

que conten´ıan una simple base de datos en la que se llevaba registro de todos los cambios

realizados sobre los archivos.